

Cloudera Data Services on premises 1.5.5

Cloudera Data Services on premises Release Notes

Date published: 2023-12-16

Date modified: 2025-11-08

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Data Services Release Notes.....	4
What's new in Cloudera Data Services on premises 1.5.5.....	4
What's new in Platform Support 1.5.5.....	5
Repository Locations for 1.5.5.....	5
Cloudera Data Services on premises 1.5.5 Release Matrix.....	6
Fixed Issues for the Cloudera Data Services on premises 1.5.5.....	6
Known issues for the Cloudera Data Services on premises 1.5.5.....	7
Fixed CVEs.....	15
Cumulative hotfixes.....	171
Cloudera Data Services on premises 1.5.5-CHF1.....	171
Repository Locations for 1.5.5-CHF1.....	172
Service packs.....	172
Cloudera Data Services on premises 1.5.5 SP1.....	172
What's new in Cloudera Data Services on premises 1.5.5 SP1.....	172
Fixed Issues for the Cloudera Data Services on premises 1.5.5 SP1.....	174
Behavioral issues for the Cloudera Data Services on premises 1.5.5 SP1.....	176
Known issues for the Cloudera Data Services on premises 1.5.5 SP1.....	176
List of fixed Common Vulnerabilities and Exposures in 1.5.5 SP1.....	181

Data Services Release Notes

Cloudera Data Services on premises 1.5.5 includes the Cloudera Management Console, Cloudera Data Catalog, Cloudera Data Warehouse, Cloudera AI, and Cloudera Data Engineering. Learn about the new features and improvements in each of these services.

Release notes for component services

- [Data Catalog](#)
- [Management Console](#)
- [Cloudera Data Warehouse](#)
- [Cloudera AI](#)
- [Cloudera Data Engineering](#)
- [Cloudera Manager](#)
- [Replication Manager](#)

What's new in Cloudera Data Services on premises 1.5.5

Understand the functionalities and improvements to features of Cloudera Control Plane install and upgrade components in Cloudera Data Services on premises 1.5.5.

Certificate Management

Cert-manager is an open-source tool for Kubernetes that automates the provisioning, management, and renewal of TLS certificates. Its documentation at <https://cert-manager.io/docs/> provides comprehensive guidance on installing, configuring, and using cert-manager to secure workloads with trusted X.509 certificates. Cloudera provides out-of-the-box support for Venafi Trust Protection Platform (TPP) as part of the Cloudera Cloudera Embedded Container Service installation. By integrating cert-manager, the Data services achieve secure communication, reduced manual overhead, and compliance with security standards, leveraging its robust automation and flexibility. For more information on setting Cert-manager using Venafi TPP, see [Setting up Certification Manager using Venafi TPP](#).

New upgrade prechecks

New pre-upgrade checks have been added to the list. The additional checks verify if the control plane and the docker registry is ready for upgrade. For more information, see [Pre-upgrade checklist](#).

Quota Management for multiple base cluster support

Quota management enables you to control how resources are allocated within your Cloudera Data Services on premises clusters. In order to prevent a single workload from consuming all available cluster resources, you can limit the number of CPUs, GPUs, and memory allocated by application, user, business units, or Data Service by defining resource pools that define resource limits. Pools are organized in a hierarchical manner by defining nodes in the hierarchy with resource limits, which can then be subdivided as needed to allocate resources for an organization and to allocate resources to cluster or environment wide services such as the monitoring service. For information, see [Quota Management](#).

Creating multiple environments with different base or Data Lake clusters

To register an environment with a data lake cluster managed by a Cloudera Manager that is different from your existing Cloudera Manager, you need to add the certificates of the new Cloudera Manager to the Cloudera Management Console UI. If the existing Cloudera Manager and the new Cloudera Manager share the same root CA,

and the root CA is already uploaded as the data lake certificate, then no additional certificate needs to be added. For more information, see [Creating multiple environments with different base or Data Lake clusters](#).

What's new in Platform Support 1.5.5

You must be aware of the platform support for Cloudera Data Services on premises 1.5.5.

Platform Certifications

- Cloudera Base on premises (7.1.9, 7.1.9 SP1 CHF5, 7.1.7 SP3 CHF10)
- Cloudera Manager 7.13.1.303
- Iceberg v2 GA on Cloudera Data Warehouse, Cloudera Data Engineering, & Cloudera AI with Ozone
- RHEL 8.10, 9.3, 9.4, 9.5
- OEL (RHCK Kernel Only) 8.10, 9.3, 9.4, 9.5
- Rocky Linux 8.10, 9.3, 9.4, 9.5
- K8s RKE2 1.30 and OCP 4.17
- Longhorn version 1.6.4



Note: Cloudera Manager 7.13.1.303 support Cloudera Data Services on premises 1.5.5 release.

Cloudera Manager 7.11.3 CHF8 does not support any Cloudera Data Services on premises release.

Repository Locations for 1.5.5

The URLs for Cloudera Data Services on premises 1.5.5 are listed in the following table:

URL Type	Repository Location
Index	<code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/</code>
Manifest	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/manifest.json</code>
Parcels	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/parcels/</code>

Cloudera Data Services on premises 1.5.5 Release Matrix

You must be familiar with the versions of all the service packs and cumulative hotfixes for Cloudera Data Services on premises 1.5.5.

Cumulative Hotfix or Service Pack	Release Date	Download URL
1.5.5 CHF1	July 28, 2025	https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/
1.5.5 SP1	November 08, 2025	https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h1000/

Fixed Issues for the Cloudera Data Services on premises 1.5.5

You can review the list of reported issues and their fixes in Cloudera Data Services on premises 1.5.5. Fixed issues represent selected issues that were previously logged through Cloudera Support, but are now addressed in the current Cloudera Data Services on premises release. These issues may have been reported in previous versions of Cloudera Data Services on premises as a known issue; meaning they were reported by customers or identified by Cloudera Quality Engineering teams.

OPSX-4308 - Display error in UI if listEnvironments failed

Error is now displayed on the Environments Page of the Cloudera Management Console UI, if an API failure is encountered.

OPSX-6048 - Clean up delete backup Custom Resource (CR) after the job is run

DeleteBackup now removes the backup deletion CR from resource `deletebackuprequests.drs.cdp.cloudera.com`

OPSX-5944 - Issues while uncordoning nodes during restart

The uncordon step was added into Cloudera Manager and is removed from the Cloudera Embedded Container Service parcel.

OPSX-5852 - Remove warn logs for "Unexpected partition in crn" from Cloudera Data Services on premises

"Unexpected partition in crn" log entries are now removed from the logs.

OPSX-5403 - Typecasting fails when truststore password is integer

When truststore password is set to all numbers (integer or float), Cloudera Control Plane installation was failing in both Cloudera Embedded Container Service and OpenShift Container Platform. Safe datatype conversion is done to treat even numbers as string password. Even if numbers are used for truststore passwords, Cloudera Control Plane installation will be successful.

OPSX-5903 - Upgrade failed with rke2-ingress-nginx-controller" exceeded its progress deadline

Automated the manual workaround of scaling down and scaling up the deployment when the earlier rollout or its status check fails.

OPSAPS-72270- Cloudera Embedded Container Service Restart| Start Cloudera Embedded Container Service command fails on uncordon nodes step

To resolve this issue:

1. Ensure the kube-apiserver is up and running for at least 60 seconds before proceeding with the uncordon step.
2. Use the correct target node name, not just the name of the node where the uncordon command is executed.

CDPVC-1739/ OPSX-5307 - Rollback workload username generation changes to pre 1.5.4-SP1 state in Cloudera Data Services on premises

Previously, workload usernames were generated by directly using the username from the IdP. This approach ensured compatibility with services like Ranger, which rely on an exact match between the workload username and the IdP username to sync users correctly.

In version 1.5.4, a change was introduced : workload username generation switched to a pattern-based logic carried over from PbC. As a result, in some cases, the generated workload username no longer matched the IdP username. This mismatch disrupted workloads and broke integrations with services like Ranger.

To resolve these issues, the workload username generation logic is reverted to its pre-1.5.4 state, restoring the direct mapping from IdP.

Known issues for the Cloudera Data Services on premises 1.5.5

You must be aware of the known issues and limitations, the areas of impact, and workaround in Cloudera Data Services on premises 1.5.5 release.

Known Issues in Cloudera Data Services on premises 1.5.5

OBS-8038: When using the Grafana Dashboard URL shortener, the shortened URL defaults to localhost:3000. This behaviour happens because the URL shortener uses the local server address instead of the actual domain name of the Cloudera Observability instance. As a result, users cannot access the shortened URL.

You must not use the shortened URL. To ensure users can access the URL, update it to use the correct Cloudera Observability instance domain name, such as `cp_domain/{shorten_url}{}`.



Note: `cp_domain` refers to the Cloudera Control Plane domain.

ENGESC-31426 - Upgrade fails with pod in ContainerCreating due to Mount Error

During an upgrade, a critical pod (such as vault-0) fails to start and remains in the ContainerCreating status. When you describe the pod (`kubectl describe pod <pod-name>`), you see a FailedMount error in the events section with the message mount point busy. This issue is caused by a stale volume mount on the Kubernetes worker node. The node's kubelet service incorrectly believes the storage volume is already in use, preventing it from being mounted to the new pod.

Effective method to resolve this issue is to perform a rolling, graceful reboot of the worker nodes before starting the upgrade. This action clears any stale in-memory locks or filesystem handles. Perform the following steps, for each worker node, one at a time:

1. Drain the Node: Safely move all workloads off the node.

```
kubectl drain <node-name> --ignore-daemonsets --delete-empty-dir-data
```

2. Reboot the Node: Log in to the node and perform a reboot.

```
sudo reboot
```

3. Uncordon the Node: After the node is back online, make it available for scheduling workloads again.

```
kubectl uncordon <node-name>
```

Performing this rolling maintenance ensures all nodes are in a clean state, preventing this issue from blocking the upgrade process.

DRS automatic backup policy is inconsistent post upgrade

When you update a backup policy manually, the modifications to the policy are not persisted after a Control Plane upgrade.

Ensure that you manually update the policy after the upgrade activity is complete.

CDPQE-32336 - The `nvidia-device-plugin-daemonset` pods fail with `CrashLoopBackoff` error

The `nvidia-device-plugin-daemonset` pods fail with `CrashLoopBackoff` error with logs showing the following error:

```
nvidia-container-cli: initialization error: driver rpc error: timed out
```

To resolve this issue, enable persistence mode through the legacy or daemon modes by following the article at: [Nvidia Persistence Mode Documentation](#).

ENGESC-29640 - Unable to resize the Longhorn volume

Longhorn does not automatically resize the filesystem for RWX volumes backed by NFS when the volume size is expanded.

For more information on explanation of the RWX/NFS limitation and step-by-step instructions to manually resize the filesystem after volume expansion, see [Longhorn Volume Expansion](#).

OPSX-6209 and DWX-20809: Cloudera Data Services on premises installations on RHEL 8.9 or lower versions may encounter issues

You may notice issues when installing Cloudera Data Services on premises on Cloudera Embedded Container Service clusters running on RHEL 8.9 or lower versions. Pod crashloops are noticed with the following error:

```
Warning FailedCreatePodSandBox          1s (x2 over 4s)  kubel
et Failed to create pod
sandbox: rpc error: code = Unknown desc = failed to create containe
r task: failed to create
shim task: OCI runtime create
failed: runc create failed: unable to start container process: u
nable to init seccomp: error
loading seccomp filter into kernel: error loading seccomp filt
er: errno 524: unknown
```

The issue is due to a memory leak with 'seccomp' (Secure Computing Mode) in the Linux kernel. If your kernel version is not on 6.2 or higher versions or if it is not part of the list of versions mentioned [here](#), you may face issues during installation.

To avoid this issue, increase the value of `net.core.bpf_jit_limit` by running the following command on all Cloudera Embedded Container Service hosts:

```
[root@host ~]# sysctl net.core.bpf_jit_limit=528482304
```

However, Cloudera recommends upgrading the Linux kernel to an appropriate version that contains a patch for the memory leak issue. For a list of versions that contain this patch, see this [link](#).



Note: RHEL 9.0, 9.1, and 9.2 will not be supported in 1.5.5 because of https://bugzilla.redhat.com/show_bug.cgi?id=2218682. You will need to upgrade to at least RHEL 9.3 if you are on 9.x line.

COMPX-20705: [153CHF-155] Post Cloudera Embedded Container Service upgrade pods are stuck in ApplicationRejected State

After upgrading the Cloudera data platform installation pods on Kubernetes could be left in a failure state showing "ApplicationRejected". This is caused by a delay in settings being applied to Kubernetes as part of the post upgrade steps.

To resolve this issue, restart the scheduler to pick up the latest settings for Kubernetes. Also, restart YuniKorn using the following commands:

```
kubectl scale deployment yunikorn-scheduler --replicas=0 -n yunikorn
kubectl scale deployment yunikorn-scheduler --replicas=1 -n yunikorn
```

OPSX-6303 - Cloudera Embedded Container Service server went down - 'etcdserver: mvcc: database space exceeded'

Cloudera Embedded Container Service server may fail with error message - "etcdserver: mvcc: database space exceeded" in large clusters.

1. Add to the safety valve for server group:

```
etcd-arg:
- "quota-backend-bytes=4294967296"
```

2. Restart stale services (Select the option re-deploy client configs).
3. The default value for quota-backend-bytes is 2 GB. It can be increased up to 8 GB.

OPSX-6295 - Cloudera Control Plane upgrade failing with cadence-matching and cadence-history

In case of extra cadence-matching and cadence-history pod stuck in Init:CreateContainerError state, Cloudera Embedded Container Service Upgrade to 1.5.5 will be stuck in retry loop because of all pods running validation failure.

You need to manually apply the workaround to proceed further upgrade and get it done successfully. Hence, delete the stuck cadence pods.

OPSX-4391 - External docker cert not base64 encoded

When using Cloudera Data Services on premises on Cloudera Embedded Container Service, in some rare situations, the CA certificate for the Docker registry in the cdp namespace is incorrectly encoded, resulting in TLS errors when connecting to the Docker registry.

Compare and edit the contents of the "cdp-private-installer-docker-cert" secret in the cdp namespace so that it matches the contents of the "cdp-private-installer-docker-cert" secret in other namespaces. The secrets and their corresponding namespaces can be identified using the command:

```
kubectl get secret -A | grep cdp-private-installer-docker-cert
```

Inspect each secret using the command:

```
kubectl get secret -n cdp cdp-private-installer-docker-cert -o yaml
```

Replace "cdp" with the different namespace names. If necessary, modify the secret in the cdp namespace using the command:

```
kubectll edit secret -n cdp cdp-private-installer-docker-cert
```

OPSX-6245 - Airgap | Multiple pods are in pending state on rolling restart

Performing back-to-back rolling restarts on Cloudera Embedded Container Service clusters can intermittently fail during the Vault unseal step. During rapid consecutive rolling restarts, the kube-controller-manager pod may not return to a ready state promptly. This can cause a cascading effect where other critical pods, including Vault, fails to initialize properly. As a result, the unseal Vault step fails.

As a workaround, perform the following steps:

1. Stop the Cloudera Embedded Container Service role that failed.
2. Start the Cloudera Embedded Container Service role again.
3. If required, perform the rolling restart again.

OPSX-4684 - Start ECS command shows green(finished) even though start docker server failed on one of the hosts

Docker service starts with one or more docker roles failed to start because the corresponding host is unhealthy.

Make sure the host is healthy. Start the the docker role in the host.

OPSX-5986 - Cloudera Embedded Container Service fresh install failing with helm-install-rke2-ingress-nginx pod failing to come into Completed state

Cloudera Embedded Container Service fresh install fails at the "Execute command Reapply All Settings to Cluster on service ECS" step due to a timeout waiting for helm-install.

To confirm the issue, run the following kubectll command on the Cloudera Embedded Container Service host to check if the pod is stuck in a running state:

```
kubectll get pods -n kube-system | grep helm-install-rke2-ingress-nginx
```

To resolve the issue, manually delete the pod by running:

```
kubectll delete pod <helm-install-rke2-ingress-nginx-pod-name> -n kube-system
```

Then, click Resume to proceed with the fresh install process on the Cloudera Manager UI.

OPSX-6298 - Issue on service namespace cleanup

There might be cases in which uninstalling services from the Cloudera Data Services on premises UI will fail due to various reasons.

In case uninstallation of a Service fails, trigger again the service uninstall process, and mark "Force Delete" to ensure that all metadata of the service will be removed from Cloudera side. Then, move to the OpenShift UI, and there search for that service namespace / project. On that project/namespace select the Action button on the top right of the screen and choose to Delete the Project.

If you move back to the main Project screen you could see that the project is moving to status "Terminating" after which it will be removed from the OCP platform. Manually terminate the <project_name>-service and <project_name>-monitoring-platform namespaces. This action will ensure that all the entities linked to that project/namespace will also be removed by OpenShift.

OPSX-6265 - Setting inotify max_user_instances config

We cannot recommend an exact value for inotify max_user_instances config. It depends on all workloads that are run in a given node.

With newly introduced features like istio, cert manager, in Cloudera Control Plane, there is a need to set inotify max_user_instancesconfig to 256 instead of 128 to resolve this issue.

COMPX-20362 - Use API to create a pool that has a subset of resource types

The Resource Management UI supports displaying only three resource types: CPU, memory and GPU. The Resource Management UI will always set all three resource types it knows about: CPU, Memory and GPU (K8s resource nvidia.com/gpu) when creating a quota. If no value is chosen for a resource type a value of 0 will be set, blocking the use of that resource type.

To create a pool that has a subset of resource types the REST API must be used as follows:

```
POST /api/v1/compute/createResourcePool
```

Payload:

```
{
  "pool": {
    "path": "root.environment.service.mypool",
    "policy": {
      "allocation": "INELASTIC"
    },
    "quota": {
      "cpu": "100 m",
      "memory": "10 GB"
    }
  }
}
```



Note: Payload needs to be confirmed and checked.

Known issues from previous releases carried in Cloudera Data Services on premises 1.5.5

Known Issues identified in 1.5.4

DOCS-21833: Orphaned replicas/pods are not getting auto cleaned up leading to volume fill-up issues

By default, Longhorn will not automatically delete the orphaned replica directory. One can enable the automatic deletion by setting orphan-auto-deletion to true.

No workaround available.

OPSX-5310: Longhorn engine images were not deployed on Cloudera Embedded Container Service server nodes

Longhorn engine images were not deployed on Cloudera Embedded Container Service nodes due to missing tolerations for Cloudera Control Plane taints. This caused the engine DaemonSet to schedule only on Cloudera Embedded Container Service agent nodes, preventing deployment on Cloudera Control Plane nodes.

1. Check the Engine DaemonSet Status. Run the following command to check if the Longhorn engine DaemonSet is missing on certain nodes:

```
kubectl get ds -n longhorn-system | grep engine
```

- Identify Taints on Affected Nodes. Run the following command to check for taints on affected nodes:

```
kubectl describe node <node-name> | grep Taints
```



Note: If you see, `node-role.kubernetes.io/control-plane=true:NoSchedule`, this confirms the issue.

- Manually Edit the DaemonSet to Add a Toleration. Edit the Longhorn engine DaemonSet YAML:

```
kubectl edit ds -n longhorn-system engine-image-ei-<your-engine-id>
```

- Add the following under tolerations:

```
tolerations:
- effect: NoSchedule
  key: node-role.kubernetes.io/control-plane
  operator: Equal
  value: "true"
```

- Apply the changes and verify deployment. Save and exit the editor. Then, check if the DaemonSet is now running on all necessary nodes:

```
kubectl get pods -n longhorn-system -o wide | grep engine
```

Verify that the engine pods are successfully scheduled on the affected Cloudera Embedded Container Service nodes.

OPSX-5155: OS Upgrade | Pods are not starting after the OS upgrade from RHEL 8.6 to 8.8

After an OS upgrade and start of the Cloudera Embedded Container Service service, pods fail to come up due to stale state.

Restart the Cloudera Embedded Container Service cluster.

OPSX-5055: Cloudera Embedded Container Service upgrade failed at Unseal Vault step

During an Cloudera Embedded Container Service upgrade from 1.5.2 to 1.5.4 release, the vault pod fails to start due to an error caused by the Longhorn volume unable to attach to the host. The error is as below:

```
Warning FailedAttachVolume 3m16s (x166 over 5h26m) attachdetach-controller
AttachVolume.Attach failed for volume "pvc-0ba86385-9064-4ef9-9019-71976b4902a5" :
rpc error: code = Internal desc = volume pvc-0ba86385-9064-4ef9-9019-71976b4902a5
failed to attach to node host-1.cloudera.com with attachmentID
csi-7659ab0e6655d308d2316536269de47b4e66062539f135bf6012bfc8b41fc345: the volume is
currently attached to different node host-2.cloudera.com
```

Follow below steps provided by SUSE to ensure the Longhorn volume is correctly attached to the node where the vault pod is running.

```
# Find out the volume name that is failing to attach to the vault
pod.
For e.g. pvc-bc73e7d3-c7e7-468a-b8e0-afdb8033e40b from the pod
logs.
kubectl edit volumeattachments.longhorn.io -n longhorn-system
pvc-bc73e7d3-c7e7-468a-b8e0-afdb8033e40b

# Update the "spec:" section of the volumeattachment and replace
```



```
attachmentTickets section with {} as shown below and save.
spec:
  attachmentTickets: {}
  volume: pvc-bc73e7d3-c7e7-468a-b8e0-afdb8033e40b

# scale down the vault statefulset to 0 and scale it back up.
kubectl scale sts vault --replicas=0 -n vault-system
kubectl scale sts vault --replicas=1 -n vault-system
```

OPX-4684: Start Cloudera Embedded Container Service command shows green(finished) even though start docker server failed on one of the hosts

The Docker service starts, but one or more Docker roles fail to start because the corresponding host is unhealthy.

Ensure the host is healthy. Start the the Docker role on the host.

OPSX-735: Kerberos service should handle Cloudera Manager downtime

The Cloudera Manager Server in the base cluster operates to generate Kerberos principals for Cloudera on premises. If there is downtime, you may observe Kerberos-related errors.

Resolve downtime on Cloudera Manager. If you encounter Kerberos errors, you can retry the operation (such as retrying creation of the Virtual Warehouse).

Known Issues identified in 1.5.2

OPsx-4594: [ECS Restart Stability] Post rolling restart few volumes are in detached state (vault being one of them)

After rolling restart there may be some volumes in detached state.

1. Open the Longhorn UI to view the detached volumes.
2. Perform the following operations for each volume in a detached state:
 - a. Identify the workload name and type from the volume details.
 - b. Identify the workload and number of replicas using `kubect`l or the Kubernetes UI.
 - c. Scale the workload down to 0.
 - d. Wait for the pods associated with the workload to fully terminate.
 - e. Scale up the workload up to the number of replicas it had originally.

To prevent this issue, use the Longhorn UI to set the number of replicas for the volume to at least 3.

OPSX-4392: Getting the real client IP address in the application

CML has a feature for adding the audit event for each user action ([Monitoring User Events](#)). In Private Cloud, instead of the client IP, we are getting the internal IP, which is logged into the internal DB.

In ECS, add the `enable-real-ip` configuration as true for the nginx ingress controller:

```
apiVersion: v1
data:
  allow-snippet-annotations: "true"
  enable-real-ip: "true"
kind: ConfigMap
metadata:
  annotations:
    meta.helm.sh/release-name: rke2-ingress-nginx
    meta.helm.sh/release-namespace: kube-system
  creationTimestamp: "2023-05-09T04:54:53Z"
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: rke2-ingress-nginx
    app.kubernetes.io/managed-by: Helm
```

```
app.kubernetes.io/name: rke2-ingress-nginx
app.kubernetes.io/part-of: rke2-ingress-nginx
app.kubernetes.io/version: 1.6.4
helm.sh/chart: rke2-ingress-nginx-4.5.201
name: rke2-ingress-nginx-controller
namespace: kube-system
resourceVersion: "162559439"
uid: cca67b0c-bc05-4e1f-8439-7d44323f4624
```

In OpenShift Container Platform, you may be able configure this using [HAproxy with X-forward-for pass to OpenShift 4](#).

CDPVC-1137, CDPAM-4388, COMPX-15083, and COMPX-15418: OpenShift Container Platform version upgrade from 4.10 to 4.11 fails due to a Pod Disruption Budget (PDB) issue

PDB can prevent a node from draining which makes the nodes to report the “Ready,SchedulingDisabled” state. As a result, the node is not updated to correct the Kubernetes version when you upgrade OpenShift Container Platform from 4.10 to 4.11.

To resolve this issue, confirm that the upgrade has failed due to the PDB issue, and then manually delete the PDBs from the Cloudera on premises namespace.

1. Run the following command to check whether the nodes are stuck in the “Ready,SchedulingDisabled” state:

```
oc get nodes
```

2. Get the machine config daemon details of the particular pod as follows:

```
oc get po -n openshift-machine-config-operator -l 'k8s-app=machine-config-daemon' -o wide
```

3. Check the logs of the machine config operator of that particular node as follows:

```
oc logs -f -n openshift-machine-config-operator [***MACHINE-CONFIG-DAEMON-NAME***] -c machine-config-daemon
```

Replace [***MACHINE-CONFIG-DAEMON-NAME***] with the actual machine config daemon name.

You may see one of the following errors in the node logs:

- error when evicting pods/cdp-release-cpx-liftie-****" -n "[***PRIVATE-CLOUD-NAMESPACE***] Cannot evict pod as it would violate the pod's disruption budget
- error when evicting pods/"cdp-release-cluster-proxy-[*****]" -n "[***PRIVATE-CLOUD-NAMESPACE***] Cannot evict pod as it would violate the pod's disruption budget

Delete the PDB from the Cloudera on premises namespace as follows:

- a. Obtain the PDB for the cdp-release-cluster-proxy namespace:

```
oc get pdb -n [***PRIVATE-CLOUD-NAMESPACE***] | grep cdp-release-cluster-proxy
```

- b. Back up the PDB:

```
oc get pdb [***PDB-NAME-OF-CLUSTER-PROXY***] -n [***PRIVATE-CLOUD-NAMESPACE***] -o yaml >> [***BACKUP-FILE-NAME***].yaml
```

- c. Delete the PDB:

```
oc delete pdb [***PDB-NAME-OF-CLUSTER-PROXY***] -n [***PRIVATE-CLOUD-NAMESPACE***]
```

Repeat the steps to delete the cdp-release-cpx-liftie PDB as well.

List of fixed Common Vulnerabilities and Exposures in 1.5.5

Review the Common vulnerabilities and Exposures (CVEs) that were fixed in this release of Cloudera Data Services on premises 1.5.5.

Issue ID	Description
CVE-1999-1237	Multiple buffer overflows in smbvalid/smbval SMB authentication library, as used in Apache::AuthenSmb and poss
CVE-2005-2541	Tar 1.15.1 does not properly warn the user when extracting setuid or setgid files, which may allow local users or rem
CVE-2007-0086	The Apache HTTP Server, when accessed through a TCP connection with a large window size, allows remote attack
CVE-1999-1412	A possible interaction between Apple MacOS X release 1.0 and Apache HTTP server allows remote attackers to cau
CVE-1999-0236	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-2007-5686	initscripts in rPath Linux 1 sets insecure permissions for the /var/log/btmp file, which allows local users to obtain ser
CVE-2007-4559	Directory traversal vulnerability in the (1) extract and (2) extractall functions in the tarfile module in Python allows s
CVE-2009-3490	GNU Wget before 1.12 does not properly handle a '\0' character in a domain name in the Common Name field of an
CVE-2010-2252	GNU Wget 1.12 and earlier uses a server-provided filename instead of the original URL to determine the destination
CVE-2007-0450	Directory traversal vulnerability in Apache HTTP Server and Tomcat 5.x before 5.5.22 and 6.x before 6.0.10, when
CVE-2010-4756	The glob implementation in the GNU C Library (aka glibc or libc6) allows remote authenticated users to cause a den
CVE-2010-5312	Cross-site scripting (XSS) vulnerability in jquery.ui.dialog.js in the Dialog widget in jQuery UI before 1.10.0 allows

CVE-2011-3374	It was found that apt-key in apt, all versions, do not correctly validate gpg keys with the master keyring, leading to a
CVE-2011-4116	_is_safe in the File::Temp module for Perl does not properly handle symlinks.
CVE-2011-3389	The SSL protocol, as used in certain configurations in Microsoft Windows and Microsoft Internet Explorer, Mozilla
CVE-2012-2663	extensions/libxt_tcp.c in iptables through 1.4.21 does not match TCP SYN+FIN packets in --syn rules, which might
CVE-2011-4461	Jetty 8.1.0.RC2 and earlier computes hash values for form parameters without restricting the ability to trigger hash c
CVE-2011-4969	Cross-site scripting (XSS) vulnerability in jQuery before 1.6.3, when using location.hash to select elements, allows r
CVE-2012-6708	jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differ
CVE-2013-4235	shadow: TOCTOU (time-of-check time-of-use) race condition when copying and removing directory trees
CVE-2013-4392	systemd, when updating file permissions, allows local users to change the permissions and SELinux security context
CVE-2014-4877	Absolute path traversal vulnerability in GNU Wget before 1.16, when recursion is enabled, allows remote FTP serve
CVE-2013-7445	The Direct Rendering Manager (DRM) subsystem in the Linux kernel through 4.x mishandles requests for Graphics
CVE-2014-6071	jQuery 1.4.2 allows remote attackers to conduct cross-site scripting (XSS) attacks via vectors related to use of the tex
CVE-2015-20107	In Python (aka CPython) up to 3.10.8, the mailcap module does not add escape characters into commands discovered
CVE-2015-8553	Xen allows guest OS users to obtain sensitive information from uninitialized locations in host OS kernel memory by
CVE-2015-5237	protobuf allows remote authenticated attackers to cause a heap-based buffer overflow.
CVE-2015-7837	The Linux kernel, as used in Red Hat Enterprise Linux 7, kernel-rt, and Enterprise MRG 2 and when booted with UE
CVE-2015-9251	jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is perform
CVE-2016-100027	Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Jav
CVE-2016-20012	OpenSSH through 8.7 allows remote attackers, who have a suspicion that a certain combination of username and pub
CVE-2016-10228	The iconv program in the GNU C Library (aka glibc or libc6) 2.31 and earlier, when invoked with multiple suffixes
CVE-2016-20013	sha256crypt and sha512crypt through 0.6 allow attackers to cause a denial of service (CPU consumption) because th
CVE-2016-2568	pkexec, when used with --user nonpriv, allows local users to escape to the parent session via a crafted TIOCTSTIOCT
CVE-2016-2781	chroot in GNU coreutils, when used with --userspec, allows local users to escape to the parent session via a crafted T
CVE-2016-3709	Possible cross-site scripting vulnerability in libxml after commit 960f0e2.
CVE-2016-7098	Race condition in wget 1.17 and earlier, when used in recursive or mirroring mode to download a single file, might a
CVE-2016-4971	GNU wget before 1.18 allows remote servers to write to arbitrary files by redirecting a request from HTTP to a craft
CVE-2016-7103	Cross-site scripting (XSS) vulnerability in jQuery UI before 1.12.0 might allow remote attackers to inject arbitrary w
CVE-2016-8660	The XFS subsystem in the Linux kernel through 4.8.2 allows local users to cause a denial of service (fdatsync failur
CVE-2017-0537	An information disclosure vulnerability in the kernel USB gadget driver could enable a local malicious application to
CVE-2017-11164	In PCRE 8.41, the OP_KETRMATCH feature in the match function in pcre_exec.c allows stack exhaustion (uncontrolle
CVE-2017-13089	The http.c:skip_short_body() function is called in some circumstances, such as when processing redirects. When the
CVE-2017-13165	An elevation of privilege vulnerability in the kernel file system. Product: Android. Versions: Android kernel. Androi
CVE-2017-13716	The C++ symbol demangler routine in cplus-dem.c in libiberty, as distributed in GNU Binutils 2.29, allows remote a
CVE-2017-13090	The retr.c:fd_read_body() function is called when processing OK responses. When the response is sent chunked in w
CVE-2017-13693	The acpi_ds_create_operands() function in drivers/acpi/acpica/dsutils.c in the Linux kernel through 4.12.9 does not f
CVE-2017-13745	There is a reachable assertion abort in the function jpc_dec_process_sot() in jpc/jpc_dec.c in JasPer 2.0.12 that will l
CVE-2017-16231	In PCRE 8.41, after compiling, a pcretest load test PoC produces a crash overflow in the function match() in pcre_ex
CVE-2017-18018	In GNU Coreutils through 8.29, chown-core.c in chown and chgrp does not prevent replacement of a plain file with a
CVE-2017-16516	In the yajl-ruby gem 1.3.0 for Ruby, when a crafted JSON file is supplied to Yajl::Parser.new.parse, the whole ruby j
CVE-2017-17095	tools/pal2rgb.c in pal2rgb in LibTIFF 4.0.9 allows remote attackers to cause a denial of service (TIFFSetupStrips hea

CVE-2017-3161	The HDFS web UI in Apache Hadoop before 2.7.0 is vulnerable to a cross-site scripting (XSS) attack through an un
CVE-2017-3162	HDFS clients interact with a servlet on the DataNode to browse the HDFS namespace. The NameNode is provided a
CVE-2017-5499	Integer overflow in libjasper/jpc/jpc_dec.c in JasPer 1.900.17 allows remote attackers to cause a denial of service (cr
CVE-2017-3737	OpenSSL 1.0.2 (starting from version 1.0.2b) introduced an "error state" mechanism. The intent was that if a fatal er
CVE-2017-3738	There is an overflow bug in the AVX2 Montgomery multiplication procedure used in exponentiation with 1024-bit m
CVE-2017-3735	While parsing an IPAddressFamily extension in an X.509 certificate, it is possible to do a one-byte overread. This w
CVE-2017-5503	The dec_clnpass function in libjasper/jpc/jpc_t1dec.c in JasPer 1.900.27 allows remote attackers to cause a denial of
CVE-2017-5504	The jpc_undo_roi function in libjasper/jpc/jpc_dec.c in JasPer 1.900.27 allows remote attackers to cause a denial of
CVE-2017-5505	The jas_matrix_asl function in jas_seq.c in JasPer 1.900.27 allows remote attackers to cause a denial of service (inva
CVE-2017-6508	CRLF injection vulnerability in the url_parse function in url.c in Wget through 1.19.1 allows remote attackers to inj
CVE-2017-6519	avahi-daemon in Avahi through 0.6.32 and 0.7 inadvertently responds to IPv6 unicast queries with source addresses
CVE-2017-7245	Stack-based buffer overflow in the pcre32_copy_substring function in pcre_get.c in libpcre1 in PCRE 8.40 allows re
CVE-2017-7246	Stack-based buffer overflow in the pcre32_copy_substring function in pcre_get.c in libpcre1 in PCRE 8.40 allows re
CVE-2017-7475	Cairo version 1.15.4 is vulnerable to a NULL pointer dereference related to the FT_Load_Glyph and FT_Render_GLI
CVE-2017-9782	JasPer 2.0.12 allows remote attackers to cause a denial of service (heap-based buffer over-read and application crash
CVE-2018-0494	GNU Wget before 1.19.5 is prone to a cookie injection vulnerability in the resp_new function in http.c via a '\r\n sequ
CVE-2018-0737	The OpenSSL RSA Key generation algorithm has been shown to be vulnerable to a cache timing side channel attack
CVE-2018-1000021	GIT version 2.15.1 and earlier contains a Input Validation Error vulnerability in Client that can result in problems in
CVE-2018-0739	Constructed ASN.1 types with a recursive definition (such as can be found in PKCS7) could eventually exceed the st
CVE-2018-1000201	ruby-ffi version 1.9.23 and earlier has a DLL loading issue which can be hijacked on Windows OS, when a Symbol
CVE-2018-1000520	ARM mbedTLS version 2.7.0 and earlier contains a Ciphersuite Allows Incorrectly Signed Certificates vulnerability
CVE-2018-1000654	GNU Libtasn1-4.13 libtasn1-4.13 version libtasn1-4.13, libtasn1-4.12 contains a DoS, specifically CPU usage will re
CVE-2018-1000876	binutils version 2.32 and earlier contains a Integer Overflow vulnerability in objdump, bfd_get_dynamic_reloc_upper
CVE-2018-1000879	libarchive version commit 379867ecb330b3a952fb7bfa7bffb7bbd5547205 onwards (release v3.3.0 onwards) contain
CVE-2018-1000880	libarchive version commit 9693801580c0cf7c70e862d305270a16b52826a7 onwards (release v3.2.0 onwards) contain
CVE-2018-10779	TIFFWriteScanline in tif_write.c in LibTIFF 3.8.2 has a heap-based buffer over-read, as demonstrated by bmp2tiff.
CVE-2018-10237	Unbounded memory allocation in Google Guava 11.0 through 24.x before 24.1.1 allows remote attackers to conduct
CVE-2018-10801	TIFFClientOpen in tif_unix.c in LibTIFF 3.8.2 has memory leaks, as demonstrated by bmp2tiff.
CVE-2018-1121	procps-ng, procps is vulnerable to a process hiding through race condition. Since the kernel's proc_pid_readdir() retu
CVE-2018-12641	An issue was discovered in arm_pt in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30. Stack Exh
CVE-2018-12697	A NULL pointer dereference (aka SEGV on unknown address 0x000000000000) was discovered in work_stuff_copy
CVE-2018-12698	demangle_template in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30, allows attackers to trigger
CVE-2018-12699	finish_stab in stabs.c in GNU Binutils 2.30 allows attackers to cause a denial of service (heap-based buffer overflow
CVE-2018-12700	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was wit
CVE-2018-12886	stack_protect_prologue in cfgexpand.c and stack_protect_epilogue in function.c in GNU Compiler Collection (GCC)
CVE-2018-12928	In the Linux kernel 4.15.0, a NULL pointer dereference was discovered in hfs_ext_read_extent in hfs.ko. This can o
CVE-2018-12929	ntfs_read_locked_inode in the ntfs.ko filesystem driver in the Linux kernel 4.15.0 allows attackers to trigger a use-af
CVE-2018-12930	ntfs_end_buffer_async_read in the ntfs.ko filesystem driver in the Linux kernel 4.15.0 allows attackers to trigger a st
CVE-2018-12931	ntfs_attr_find in the ntfs.ko filesystem driver in the Linux kernel 4.15.0 allows attackers to trigger a stack-based out-
CVE-2018-12934	remember_Ktype in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30, allows attackers to trigger e

CVE-2018-15209	ChopUpSingleUncompressedStrip in tif_dirread.c in LibTIFF 4.0.9 allows remote attackers to cause a denial of serv
CVE-2018-15594	arch/x86/kernel/paravirt.c in the Linux kernel before 4.18.1 mishandles certain indirect calls, which makes it easier f
CVE-2018-15919	Remotely observable behaviour in auth-gss2.c in OpenSSH through 7.8 could be used by remote attackers to detect t
CVE-2018-16335	newoffsets handling in ChopUpSingleUncompressedStrip in tif_dirread.c in LibTIFF 4.0.9 allows remote attackers t
CVE-2018-16435	Little CMS (aka Little Color Management System) 2.9 has an integer overflow in the AllocateDataSet function in cn
CVE-2018-16487	A prototype pollution vulnerability was found in lodash <4.17.11 where the functions merge, mergeWith, and default
CVE-2018-16862	A security flaw was found in the Linux kernel in a way that the cleancache subsystem clears an inode after the final
CVE-2018-17101	An issue was discovered in LibTIFF 4.0.9. There are two out-of-bounds writes in cpTags in tools/tiff2bw.c and tools
CVE-2018-17190	In all versions of Apache Spark, its standalone resource manager accepts code to execute on a 'master' host, that then
CVE-2018-17360	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.3
CVE-2018-17794	An issue was discovered in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a NULL poi
CVE-2018-17977	The Linux kernel 4.14.67 mishandles certain interaction among XFRM Netlink messages, IPPROTO_AH packets, a
CVE-2018-17985	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a stack co
CVE-2018-18064	cairo through 1.15.14 has an out-of-bounds stack-memory write during processing of a crafted document by WebKit
CVE-2018-18074	The Requests package before 2.20.0 for Python sends an HTTP Authorization header to an http URI upon receiving
CVE-2018-18309	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.3
CVE-2018-18483	The get_count function in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31, allows remote attacker
CVE-2018-18484	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. Stack Exhaustion
CVE-2018-18605	A heap-based buffer over-read issue was discovered in the function sec_merge_hash_lookup in merge.c in the Binary
CVE-2018-18606	An issue was discovered in the merge_strings function in merge.c in the Binary File Descriptor (BFD) library (aka li
CVE-2018-18607	An issue was discovered in elf_link_input_bfd in elflink.c in the Binary File Descriptor (BFD) library (aka libbfd), a
CVE-2018-18700	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a stack co
CVE-2018-18701	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a stack co
CVE-2018-18873	An issue was discovered in JasPer 2.0.14. There is a NULL pointer dereference in the function ras_putdatastd in ras/
CVE-2018-19139	An issue has been found in JasPer 2.0.14. There is a memory leak in jas_malloc.c when called from jpc_unk_getparr
CVE-2018-19210	In LibTIFF 4.0.9, there is a NULL pointer dereference in the TIFFWriteDirectorySec function in tif_dirwrite.c that v
CVE-2018-19217	In ncurses, possibly a 6.x version, there is a NULL pointer dereference at the function _nc_name_match that will lea
CVE-2018-19211	In ncurses 6.1, there is a NULL pointer dereference at function _nc_parse_entry in parse_entry.c that will lead to a d
CVE-2018-19351	Jupyter Notebook before 5.7.1 allows XSS via an untrusted notebook because nbconvert responses are considered to
CVE-2018-19352	Jupyter Notebook before 5.7.2 allows XSS via a crafted directory name because notebook/static/tree/js/notebooklist.
CVE-2018-19541	An issue was discovered in JasPer 1.900.8, 1.900.9, 1.900.10, 1.900.11, 1.900.12, 1.900.13, 1.900.14, 1.900.15, 1.90
CVE-2018-19539	An issue was discovered in JasPer 2.0.14. There is an access violation in the function jas_image_readcmpt in libjasp
CVE-2018-19540	An issue was discovered in JasPer 1.900.8, 1.900.9, 1.900.10, 1.900.11, 1.900.12, 1.900.13, 1.900.14, 1.900.15, 1.90
CVE-2018-19542	An issue was discovered in JasPer 2.0.14. There is a NULL pointer dereference in the function jp2_decode in libjasp
CVE-2018-19543	An issue was discovered in JasPer 2.0.14. There is a heap-based buffer over-read of size 8 in the function jp2_decod
CVE-2018-19876	cairo 1.16.0, in cairo_ft_apply_variations() in cairo-ft-font.c, would free memory using a free function incompatible
CVE-2018-19932	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils thr
CVE-2018-20002	The _bfd_generic_read_minisymbols function in syms.c in the Binary File Descriptor (BFD) library (aka libbfd), as
CVE-2018-20225	An issue was discovered in pip (all versions) because it installs the version with the highest version number, even if
CVE-2018-20483	set_file_metadata in xattr.c in GNU Wget before 1.20.1 stores a file's origin URL in the user.xdg.origin.url metadata

CVE-2018-20570	jp2_encode in jp2/jp2_enc.c in JasPer 2.0.14 has a heap-based buffer over-read.
CVE-2018-20622	JasPer 2.0.14 has a memory leak in base/jas_malloc.c in libjasper.a when "--output-format jp2" is used.
CVE-2018-20623	In GNU Binutils 2.31.1, there is a use-after-free in the error function in elfcomm.c when called from the process_
CVE-2018-20651	A NULL pointer dereference was discovered in elf_link_add_object_symbols in elflink.c in the Binary File Descript
CVE-2018-20657	The demangle_template function in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31.1, has a mem
CVE-2018-20671	load_specific_debug_section in objdump.c in GNU Binutils through 2.31.1 contains an integer overflow vulnerability
CVE-2018-20673	The demangle_template function in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31.1, contains a
CVE-2018-20786	libvterm through 0+bzr726, as used in Vim and other products, mishandles certain out-of-memory conditions, leadin
CVE-2018-20796	In the GNU C Library (aka glibc or libc6) through 2.29, check_dst_limits_calc_pos_1 in posix/regexec.c has Uncont
CVE-2018-20839	systemd 242 changes the VT1 mode upon a logout, which allows attackers to read cleartext passwords in certain circ
CVE-2018-21030	Jupyter Notebook before 5.5.0 does not use a CSP header to treat served files as belonging to a separate origin. Thus
CVE-2018-3721	lodash node module before 4.17.5 suffers from a Modification of Assumed-Immutable Data (MAID) vulnerability vi
CVE-2018-25032	zlib before 1.2.12 allows memory corruption when deflating (i.e., when compressing) if the input has many distant m
CVE-2018-5360	LibTIFF before 4.0.6 mishandles the reading of TIFF files, as demonstrated by a heap-based buffer over-read in the
CVE-2018-5709	An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n_key_data" in k
CVE-2018-5711	gd_gif_in.c in the GD Graphics Library (aka libgd), as used in PHP before 5.6.33, 7.0.x before 7.0.27, 7.1.x before 7
CVE-2018-6829	cipher/elgama.c in Libgcrypt through 1.8.2, when used to encrypt messages directly, improperly encodes plaintexts,
CVE-2018-6872	The elf_parse_notes function in elf.c in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU
CVE-2018-6952	A double free exists in the another_hunk function in pch.c in GNU patch through 2.7.6.
CVE-2018-7169	An issue was discovered in shadow 4.5. newgidmap (in shadow-utils) is setuid and allows an unprivileged user to be
CVE-2018-7273	In the Linux kernel through 4.15.4, the floppy driver reveals the addresses of kernel functions and global variables u
CVE-2018-8043	The unimac_mdio_probe function in drivers/net/phy/mdio-bcm-unimac.c in the Linux kernel through 4.15.8 does no
CVE-2018-8768	In Jupyter Notebook before 5.4.1, a maliciously forged notebook file can bypass sanitization to execute JavaScript in
CVE-2018-9055	JasPer 2.0.14 allows denial of service via a reachable assertion in the function jpc_firstone in libjasper/jpc/jpc_math.
CVE-2018-9252	JasPer 2.0.14 allows denial of service via a reachable assertion in the function jpc_abstorelstepsize in libjasper/jpc/jp
CVE-2019-1010022	GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass stack guard protection. The
CVE-2019-10086	In Apache Commons Beanutils 1.9.2, a special BeanIntrospector class was added which allows suppressing the abili
CVE-2019-1010023	GNU Libc current is affected by: Re-mapping current loaded library with malicious ELF file. The impact is: In wors
CVE-2019-1010024	GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass ASLR using cache of thead
CVE-2019-1010025	GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may guess the heap addresses of pthread
CVE-2019-1010204	GNU binutils gold v1.11-v1.16 (GNU binutils v2.21-v2.31.1) is affected by: Improper Input Validation, Signed
CVE-2019-10255	An Open Redirect vulnerability for all browsers in Jupyter Notebook before 5.7.7 and some browsers (Chrome, Fire
CVE-2019-1010266	lodash prior to 4.17.11 is affected by: CWE-400: Uncontrolled Resource Consumption. The impact is: Denial of serv
CVE-2019-10856	In Jupyter Notebook before 5.7.8, an open redirect can occur via an empty netloc. This issue exists because of an inc
CVE-2019-10219	A vulnerability was found in Hibernate-Validator. The SafeHtml validator annotation fails to properly sanitize paylo
CVE-2019-11038	When using the gdImageCreateFromXbm() function in the GD Graphics Library (aka LibGD) 2.2.5, as used in the P
CVE-2019-12290	GNU libidn2 before 2.2.0 fails to perform the roundtrip checks specified in RFC3490 Section 4.2 when converting A
CVE-2019-12380	**DISPUTED** An issue was discovered in the efi subsystem in the Linux kernel through 5.1.5. phys_efi_set_virtu
CVE-2019-11358	jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) b
CVE-2019-12900	BZ2_decompress in decompress.c in bzip2 through 1.0.6 has an out-of-bounds write when there are many selectors.

CVE-2019-12904	In Libgcrypt 1.8.4, the C implementation of AES is vulnerable to a flush-and-reload side-channel attack because phy
CVE-2019-12972	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.3
CVE-2019-13115	In libssh2 before 1.9.0, <code>kex_method_diffie_hellman_group_exchange_sha256_key_exchange</code> in <code>kex.c</code> has an integer
CVE-2019-13117	In <code>numbers.c</code> in <code>libxslt</code> 1.1.33, an <code>xsl:number</code> with certain format strings could lead to a uninitialized read in <code>xsltNum</code>
CVE-2019-13118	In <code>numbers.c</code> in <code>libxslt</code> 1.1.33, a type holding grouping characters of an <code>xsl:number</code> instruction was too narrow and a
CVE-2019-13627	It was discovered that there was a ECDSA timing attack in the <code>libgcrypt20</code> cryptographic library. Version affected: 1
CVE-2019-14250	An issue was discovered in GNU <code>libiberty</code> , as distributed in GNU Binutils 2.32. <code>simple_object_elf_match</code> in <code>simple-</code>
CVE-2019-14899	A vulnerability was discovered in Linux, FreeBSD, OpenBSD, MacOS, iOS, and Android that allows a malicious ac
CVE-2019-14855	A flaw was found in the way certificate signatures could be forged using collisions found in the SHA-1 algorithm. A
CVE-2019-15213	An issue was discovered in the Linux kernel before 5.2.3. There is a use-after-free caused by a malicious USB device
CVE-2019-15217	An issue was discovered in the Linux kernel before 5.2.3. There is a NULL pointer dereference caused by a malicious
CVE-2019-15218	An issue was discovered in the Linux kernel before 5.1.8. There is a NULL pointer dereference caused by a malicious
CVE-2019-15219	An issue was discovered in the Linux kernel before 5.1.8. There is a NULL pointer dereference caused by a malicious
CVE-2019-15291	An issue was discovered in the Linux kernel through 5.2.9. There is a NULL pointer dereference caused by a malicious
CVE-2019-15794	Overlays in the Linux kernel and <code>shiftfs</code> , a non-upstream patch to the Linux kernel included in the Ubuntu 5.0 and 5
CVE-2019-1551	There is an overflow bug in the <code>x64_64</code> Montgomery squaring procedure used in exponentiation with 512-bit modul
CVE-2019-15807	In the Linux kernel before 5.1.13, there is a memory leak in <code>drivers/scsi/libsas/sas_expander.c</code> when SAS expander c
CVE-2019-15847	The POWER9 backend in GNU Compiler Collection (GCC) before version 10 could optimize multiple calls of the <code>_</code>
CVE-2019-16089	An issue was discovered in the Linux kernel through 5.2.13. <code>nbd_genl_status</code> in <code>drivers/block/nbd.c</code> does not check t
CVE-2019-16229	<code>drivers/gpu/drm/amd/amdkfd/kfd_interrupt.c</code> in the Linux kernel 5.2.14 does not check the <code>alloc_workqueue</code> return v
CVE-2019-16230	<code>drivers/gpu/drm/radeon/radeon_display.c</code> in the Linux kernel 5.2.14 does not check the <code>alloc_workqueue</code> return valu
CVE-2019-17113	In <code>libopenmpt</code> before 0.3.19 and 0.4.x before 0.4.9, <code>ModPlug_InstrumentName</code> and <code>ModPlug_SampleName</code> in <code>libop</code>
CVE-2019-17498	In <code>libssh2</code> v1.9.0 and earlier versions, the <code>SSH_MSG_DISCONNECT</code> logic in <code>packet.c</code> has an integer overflow in a
CVE-2019-17543	LZ4 before 1.9.2 has a heap-based buffer overflow in <code>LZ4_write32</code> (related to <code>LZ4_compress_destSize</code>), affecting ap
CVE-2019-17567	Apache HTTP Server versions 2.4.6 to 2.4.46 <code>mod_proxy_wstunnel</code> configured on an URL that is not necessarily Up
CVE-2019-18806	A memory leak in the <code>ql_alloc_large_buffers()</code> function in <code>drivers/net/ethernet/qlogic/qla3xxx.c</code> in the Linux kernel l
CVE-2019-19039	<code>__btrfs_free_extent</code> in <code>fs/btrfs/extent-tree.c</code> in the Linux kernel through 5.3.12 calls <code>btrfs_print_leaf</code> in a certain ENO
CVE-2019-19054	A memory leak in the <code>cx23888_ir_probe()</code> function in <code>drivers/media/pci/cx23885/cx23888-ir.c</code> in the Linux kernel th
CVE-2019-19066	A memory leak in the <code>bfad_im_get_stats()</code> function in <code>drivers/scsi/bfa/bfad_attr.c</code> in the Linux kernel through 5.3.11
CVE-2019-19080	Four memory leaks in the <code>nfp_flower_spawn_phy_reprs()</code> function in <code>drivers/net/ethernet/netronome/nfp/flower/main.c</code>
CVE-2019-19076	A memory leak in the <code>nfp_abm_u32_knode_replace()</code> function in <code>drivers/net/ethernet/netronome/nfp/abm/cls.c</code> in the
CVE-2019-19081	A memory leak in the <code>nfp_flower_spawn_vnic_reprs()</code> function in <code>drivers/net/ethernet/netronome/nfp/flower/main.c</code>
CVE-2019-19082	Memory leaks in <code>*create_resource_pool()</code> functions under <code>drivers/gpu/drm/amd/display/dc</code> in the Linux kernel throu
CVE-2019-19083	Memory leaks in <code>*clock_source_create()</code> functions under <code>drivers/gpu/drm/amd/display/dc</code> in the Linux kernel before
CVE-2019-19126	On the x86-64 architecture, the GNU C Library (aka glibc) before 2.31 fails to ignore the <code>LD_PREFER_MAP_32BI</code>
CVE-2019-19244	<code>sqlite3Select</code> in <code>select.c</code> in SQLite 3.30.1 allows a crash if a sub-select uses both <code>DISTINCT</code> and window functions,
CVE-2019-19378	In the Linux kernel 5.0.21, mounting a crafted <code>btrfs</code> filesystem image can lead to slab-out-of-bounds write access in
CVE-2019-19530	In the Linux kernel before 5.2.10, there is a use-after-free bug that can be caused by a malicious USB device in the d
CVE-2019-19462	<code>relay_open</code> in <code>kernel/relay.c</code> in the Linux kernel through 5.4.1 allows local users to cause a denial of service (such as
CVE-2019-19965	In the Linux kernel through 5.4.6, there is a NULL pointer dereference in <code>drivers/scsi/libsas/sas_discover.c</code> because c

CVE-2019-19814	In the Linux kernel 5.0.21, mounting a crafted f2fs filesystem image can cause __remove_dirty_segment slab-out-of-
CVE-2019-20095	mwifiex_tm_cmd in drivers/net/wireless/marvell/mwifiex/cfg80211.c in the Linux kernel before 5.1.6 has some erro
CVE-2019-20426	In the Lustre file system before 2.12.3, the ptlrpc module has an out-of-bounds access and panic due to the lack of va
CVE-2019-20454	An out-of-bounds read was discovered in PCRE before 10.34 when the pattern \X is JIT compiled and used to match
CVE-2019-20633	GNU patch through 2.7.6 contains a free(p_line[p_end]) Double Free vulnerability in the function another_hunk in p
CVE-2019-20794	An issue was discovered in the Linux kernel 4.18 through 5.6.11 when unprivileged user namespaces are allowed. A
CVE-2019-20838	libpcre in PCRE before 8.43 allows a subject buffer over-read in JIT when UTF is disabled, and \X or \R has more th
CVE-2019-2201	In generate_jsimd_ycc_rgb_convert_neon of jsimd_arm64_neon.S, there is a possible out of bounds write due to a m
CVE-2019-25013	The iconv feature in the GNU C Library (aka glibc or libc6) through 2.32, when processing invalid multi-byte input
CVE-2019-25160	In the Linux kernel, the following vulnerability has been resolved: netlabel: fix out-of-bounds memory accesses The
CVE-2019-3844	It was discovered that a systemd service that uses DynamicUser property can get new privileges through the executio
CVE-2019-3843	It was discovered that a systemd service that uses DynamicUser property can create a SUID/SGID binary that would
CVE-2019-5428	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2019-11358. Reason: This candi
CVE-2019-5953	Buffer overflow in GNU Wget 1.20.1 and earlier allows remote attackers to cause a denial-of-service (DoS) or may
CVE-2019-6110	In OpenSSH 7.9, due to accepting and displaying arbitrary stderr output from the server, a malicious server (or Man-
CVE-2019-6128	The TIFFFdOpen function in tif_unix.c in LibTIFF 4.0.10 has a memory leak, as demonstrated by pal2rgb.
CVE-2019-6461	An issue was discovered in cairo 1.16.0. There is an assertion problem in the function _cairo_arc_in_direction in the
CVE-2019-6462	An issue was discovered in cairo 1.16.0. There is an infinite loop in the function _arc_error_normalized in the file ca
CVE-2019-6988	An issue was discovered in OpenJPEG 2.3.0. It allows remote attackers to cause a denial of service (attempted excess
CVE-2019-8457	SQLite3 from 3.6.0 to and including 3.27.2 is vulnerable to heap out-of-bound read in the rtreeNode() function when
CVE-2019-8905	do_core_note in readelf.c in libmagic.a in file 5.35 has a stack-based buffer over-read, related to file_printable, a diff
CVE-2019-7317	png_image_free in png.c in libpng 1.6.x before 1.6.37 has a use-after-free because png_image_free_function is calle
CVE-2019-8906	do_core_note in readelf.c in libmagic.a in file 5.35 has an out-of-bounds read because memcpy is misused.
CVE-2019-9071	An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.32. It is a stack consumption issue in d_
CVE-2019-9077	An issue was discovered in GNU Binutils 2.32. It is a heap-based buffer overflow in process_mips_specific in reade
CVE-2019-9075	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.3
CVE-2019-9192	In the GNU C Library (aka glibc or libc6) through 2.29, check_dst_limits_calc_pos_1 in posix/regexec.c has Uncont
CVE-2019-9644	An XSSI (cross-site inclusion) vulnerability in Jupyter Notebook before 5.7.6 allows inclusion of resources on malic
CVE-2019-9674	Lib/zipfile.py in Python through 3.7.2 allows remote attackers to cause a denial of service (resource consumption) vi
CVE-2019-9923	pax_decode_header in sparse.c in GNU Tar before 1.32 had a NULL pointer dereference when parsing certain archiv
CVE-2019-9936	In SQLite 3.27.2, running fts5 prefix queries inside a transaction could trigger a heap-based buffer over-read in fts5F
CVE-2019-9937	In SQLite 3.27.2, interleaving reads and writes in a single transaction with an fts5 virtual table will lead to a NULL
CVE-2020-0499	In FLAC__bitreader_read_rice_signed_block of bitreader.c, there is a possible out of bounds read due to a heap buff
CVE-2020-10029	The GNU C Library (aka glibc or libc6) before 2.32 could overflow an on-stack buffer during range reduction if an i
CVE-2020-10135	Legacy pairing and secure-connections pairing authentication in Bluetooth BR/EDR Core Specification v5.2 and ear
CVE-2020-10663	The JSON gem through 2.2.0 for Ruby, as used in Ruby 2.4 through 2.4.9, 2.5 through 2.5.7, and 2.6 through 2.6.5, h
CVE-2020-10693	A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables in
CVE-2020-10735	A flaw was found in python. In algorithms with quadratic time complexity using non-binary bases, when using int("t
CVE-2020-10781	A flaw was found in the Linux Kernel before 5.8-rc6 in the ZRAM kernel module, where a user with a local account
CVE-2020-11080	In nghttp2 before version 1.41.0, the overly large HTTP/2 SETTINGS frame payload causes denial of service. The p

CVE-2020-11023	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from
CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after s
CVE-2020-11494	An issue was discovered in slc_bump in drivers/net/can/slcscan.c in the Linux kernel 3.16 through 5.6.2. It allows attac
CVE-2020-11609	An issue was discovered in the stv06xx subsystem in the Linux kernel before 5.6.1. drivers/media/usb/gspca/stv06xx
CVE-2020-12351	Improper input validation in BlueZ may allow an unauthenticated user to potentially enable escalation of privilege vi
CVE-2020-12352	Improper access control in BlueZ may allow an unauthenticated user to potentially enable information disclosure via
CVE-2020-12413	The Raccoon attack is a timing attack on DHE ciphersuites inherit in the TLS specification. To mitigate this vulnerab
CVE-2020-12480	In Play Framework 2.6.0 through 2.8.1, the CSRF filter can be bypassed by making CORS simple requests with cont
CVE-2020-12656	gss_mech_free in net/sunrpc/auth_gss/gss_mech_switch.c in the rpcsec_gss_krb5 implementation in the Linux kernel
CVE-2020-12768	An issue was discovered in the Linux kernel before 5.6. svm_cpu_uninit in arch/x86/kvm/svm.c has a memory leak,
CVE-2020-12769	An issue was discovered in the Linux kernel before 5.4.17. drivers/spi/spi-dw.c allows attackers to cause a panic via
CVE-2020-13790	libjpeg-turbo 2.0.4, and mozjpeg 4.0.0, has a heap-based buffer over-read in get_rgb_row() in rdppm.c via a malformed
CVE-2020-13529	An exploitable denial-of-service vulnerability exists in Systemd 245. A specially crafted DHCP FORCERENEW pac
CVE-2020-13645	In GNOME glib-networking through 2.64.2, the implementation of GTlsClientConnection skips hostname verificatio
CVE-2020-13955	HttpUtils#getURLConnection method disables explicitly hostname verification for HTTPS connections making client
CVE-2020-14304	A memory disclosure flaw was found in the Linux kernel's ethernet drivers, in the way it read data from the EEPROM
CVE-2020-13844	Arm Armv8-A core implementations utilizing speculative execution past unconditional changes in control flow may
CVE-2020-14155	libpcre in PCRE before 8.44 allows an integer overflow via a large number after a (?C substring.
CVE-2020-13956	Apache HttpClient versions prior to version 4.5.13 and 5.0.3 can misinterpret malformed authority component in req
CVE-2020-14390	A flaw was found in the Linux kernel in versions before 5.9-rc6. When changing screen size, an out-of-bounds mem
CVE-2020-14356	A flaw null pointer dereference in the Linux kernel cgroupv2 subsystem in versions before 5.7.10 was found in the v
CVE-2020-14416	In the Linux kernel before 5.4.16, a race condition in tty->disc_data handling in the slip and slcan line discipline cou
CVE-2020-14422	Lib/ipaddress.py in Python through 3.8.3 improperly computes hash values in the IPv4Interface and IPv6Interface cl
CVE-2020-15522	Bouncy Castle BC Java before 1.66, BC C# .NET before 1.8.7, BC-FJA before 1.0.1.2, 1.0.2.1, and BC-FNA before
CVE-2020-16156	CPAN 2.28 allows Signature Verification Bypass.
CVE-2020-16598	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was wit
CVE-2020-1751	An out-of-bounds write vulnerability was found in glibc before 2.31 when handling signal trampolines on PowerPC.
CVE-2020-1752	A use-after-free vulnerability introduced in glibc upstream version 2.14 was found in the way the tilde expansion wa
CVE-2020-18768	There exists one heap buffer overflow in _TIFFmemcpy in tif_unix.c in libtiff 4.0.10, which allows an attacker to ca
CVE-2020-19185	Buffer Overflow vulnerability in one_one_mapping function in progs/dump_entry.c:1373 in ncurses 6.1 allows remo
CVE-2020-19186	Buffer Overflow vulnerability in _nc_find_entry function in tinfo/comp_hash.c:66 in ncurses 6.1 allows remote attac
CVE-2020-19187	Buffer Overflow vulnerability in fmt_entry function in progs/dump_entry.c:1100 in ncurses 6.1 allows remote attack
CVE-2020-19188	Buffer Overflow vulnerability in fmt_entry function in progs/dump_entry.c:1116 in ncurses 6.1 allows remote attack
CVE-2020-19189	Buffer Overflow vulnerability in postprocess_terminfo function in tinfo/parse_entry.c:997 in ncurses 6.1 allows rem
CVE-2020-19190	Buffer Overflow vulnerability in _nc_find_entry in tinfo/comp_hash.c:70 in ncurses 6.1 allows remote attackers to c
CVE-2020-20703	Buffer Overflow vulnerability in VIM v.8.1.2135 allows a remote attacker to execute arbitrary code via the operand
CVE-2020-21469	An issue was discovered in PostgreSQL 12.2 allows attackers to cause a denial of service via repeatedly sending SIG
CVE-2020-21514	An issue was discovered in Fluent-ui v.1.2.2 allows attackers to gain escalated privileges and execute arbitrary code
CVE-2020-21674	Heap-based buffer overflow in archive_string_append_from_wcs() (archive_string.c) in libarchive-3.4.1dev allows r
CVE-2020-23064	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2020-11023. Reason: This cand

CVE-2020-22218	An issue was discovered in function <code>_libssh2_packet_add</code> in <code>libssh2</code> 1.10.0 allows attackers to access out of bounds memory.
CVE-2020-24490	Improper buffer restrictions in BlueZ may allow an unauthenticated user to potentially enable denial of service via a crafted packet.
CVE-2020-24736	Buffer Overflow vulnerability found in SQLite3 v.3.27.1 and before allows a local attacker to cause a denial of service.
CVE-2020-24659	An issue was discovered in GnuTLS before 3.6.15. A server can trigger a NULL pointer dereference in a TLS 1.3 client.
CVE-2020-25219	<code>url::recvline</code> in <code>url.cpp</code> in <code>libproxy</code> 0.4.x through 0.4.15 allows a remote HTTP server to trigger uncontrolled recursion.
CVE-2020-25639	A NULL pointer dereference flaw was found in the Linux kernel's GPU Nouveau driver functionality in versions prior to 5.9-rc7.
CVE-2020-25645	A flaw was found in the Linux kernel in versions before 5.9-rc7. Traffic between two Geneve endpoints may be unauthenticated.
CVE-2020-25658	It was found that <code>python-rsa</code> is vulnerable to Bleichenbacher timing attacks. An attacker can use this flaw via the <code>RSAPublicKey</code> object.
CVE-2020-26215	Jupyter Notebook before version 6.1.5 has an Open redirect vulnerability. A maliciously crafted link to a notebook server can redirect to an arbitrary URL.
CVE-2020-26244	Python oic is a Python OpenID Connect implementation. In Python oic before version 1.2.1, there are several related issues.
CVE-2020-26555	Bluetooth legacy BR/EDR PIN code pairing in Bluetooth Core Specification 1.0B through 5.2 may permit an unauthenticated pairing.
CVE-2020-26558	Bluetooth LE and BR/EDR secure pairing in Bluetooth Core Specification 2.1 through 5.2 may permit a nearby man-in-the-middle attack.
CVE-2020-26939	In Legion of the Bouncy Castle BC before 1.61 and BC-FJA before 1.0.1.2, attackers can obtain sensitive information from the <code>BCFIPS</code> provider.
CVE-2020-27170	An issue was discovered in the Linux kernel before 5.11.8. <code>kernel/bpf/verifier.c</code> performs undesirable out-of-bounds access.
CVE-2020-27171	An issue was discovered in the Linux kernel before 5.11.8. <code>kernel/bpf/verifier.c</code> has an off-by-one error (with a result of zero).
CVE-2020-27534	<code>util/binfmt_misc/check.go</code> in Builder in Docker Engine before 19.03.9 calls <code>os.OpenFile</code> with a potentially unsafe <code>perm</code> argument.
CVE-2020-27618	The <code>iconv</code> function in the GNU C Library (aka <code>glibc</code> or <code>libc6</code>) 2.32 and earlier, when processing invalid multi-byte input, may cause a segmentation fault.
CVE-2020-28097	The <code>vgacon</code> subsystem in the Linux kernel before 5.8.10 mishandles software scrollbar. There is a <code>vgacon_scrollback</code> buffer overflow.
CVE-2020-29374	An issue was discovered in the Linux kernel before 5.7.3, related to <code>mm/gup.c</code> and <code>mm/huge_memory.c</code> . The <code>get_user</code> function may cause a segmentation fault.
CVE-2020-29582	In JetBrains Kotlin before 1.4.21, a vulnerable Java API was used for temporary file and folder creation. An attacker can cause a denial of service.
CVE-2020-35493	A flaw exists in <code>binutils</code> in <code>bfd/pef.c</code> . An attacker who is able to submit a crafted PEF file to be parsed by <code>objdump</code> can cause a denial of service.
CVE-2020-35494	There's a flaw in <code>binutils</code> <code>/opcodes/tic4x-dis.c</code> . An attacker who is able to submit a crafted input file to be processed by the <code>objdump</code> can cause a denial of service.
CVE-2020-35495	There's a flaw in <code>binutils</code> <code>/bfd/pef.c</code> . An attacker who is able to submit a crafted input file to be processed by the <code>objdump</code> can cause a denial of service.
CVE-2020-35496	There's a flaw in <code>bfd_pef_scan_start_address()</code> of <code>bfd/pef.c</code> in <code>binutils</code> which could allow an attacker who is able to submit a crafted input file to be processed by the <code>objdump</code> to cause a denial of service.
CVE-2020-35501	A flaw was found in the Linux kernels implementation of audit rules, where a <code>syscall</code> can unexpectedly not be corrected.
CVE-2020-35507	There's a flaw in <code>bfd_pef_parse_function_stubs</code> of <code>bfd/pef.c</code> in <code>binutils</code> in versions prior to 2.34 which could allow an attacker to cause a denial of service.
CVE-2020-35512	A use-after-free flaw was found in D-Bus Development branch <= 1.13.16, <code>dbus-1.12.x</code> stable branch <= 1.12.18, and <code>dbus-1.10.x</code> stable branch <= 1.10.18.
CVE-2020-35525	In SQLite 3.31.1, a potential null pointer dereference was found in the INTERSECT query processing.
CVE-2020-35527	In SQLite 3.31.1, there is an out of bounds access problem through ALTER TABLE for views that have a nested FROM clause.
CVE-2020-35538	A crafted input file could cause a null pointer dereference in <code>jpeg_sample_rows()</code> when processed by <code>libjpeg-turbo</code> .
CVE-2020-36311	An issue was discovered in the Linux kernel before 5.9. <code>arch/x86/kvm/svm/sev.c</code> allows attackers to cause a denial of service via a crafted packet.
CVE-2020-36789	In the Linux kernel, the following vulnerability has been resolved: <code>can: dev: can_get_echo_skb()</code> : prevent call to <code>kfree_skb()</code> on a non-allocated skb.
CVE-2020-36518	<code>jackson-databind</code> before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.
CVE-2020-36790	In the Linux kernel, the following vulnerability has been resolved: <code>nvmet: fix a memory leak</code> We forgot to free <code>new_nvmf_request</code> in <code>nvmet_req_init</code> .
CVE-2020-36791	In the Linux kernel, the following vulnerability has been resolved: <code>net_sched: keep alloc_hash updated after hash allocation</code> .
CVE-2020-36843	The implementation of EdDSA in EdDSA-Java (aka <code>ed25519-java</code>) through 0.3.0 exhibits signature malleability and a denial of service.
CVE-2020-7656	<code>jquery</code> prior to 1.9.0 allows Cross-site Scripting attacks via the <code>load</code> method. The <code>load</code> method fails to recognize and escape <code><script></code> tags.
CVE-2020-3702	Specifically timed and handcrafted traffic can cause internal errors in a WLAN device that lead to improper layer 2 security.
CVE-2020-6096	An exploitable signed comparison vulnerability exists in the ARMv7 <code>memcpy()</code> implementation of GNU <code>glibc</code> 2.30.0.
CVE-2020-8130	There is an OS command injection vulnerability in Ruby Rake < 12.3.3 in <code>Rake::FileList</code> when supplying a filename to the <code>add</code> method.

CVE-2020-8169	curl 7.62.0 through 7.70.0 is vulnerable to an information disclosure vulnerability that can lead to a partial password
CVE-2020-8177	curl 7.20.0 through 7.70.0 is vulnerable to improper restriction of names for files and other resources that can lead to
CVE-2020-8231	Due to use of a dangling pointer, libcurl 7.29.0 through 7.71.1 can use the wrong connection when sending data.
CVE-2020-8284	A malicious server can use the FTP PASV response to trick curl 7.73.0 and earlier into connecting back to a given IP
CVE-2020-8285	curl 7.21.0 to and including 7.73.0 is vulnerable to uncontrolled recursion due to a stack overflow issue in FTP wild
CVE-2020-8286	curl 7.41.0 through 7.73.0 is vulnerable to an improper check for certificate revocation due to insufficient verificatio
CVE-2020-8554	Kubernetes API server in all versions allow an attacker who is able to create a ClusterIP service and set the spec.exte
CVE-2020-8492	Python 2.7 through 2.7.17, 3.5 through 3.5.9, 3.6 through 3.6.10, 3.7 through 3.7.6, and 3.8 through 3.8.1 allows an l
CVE-2020-8559	The Kubernetes kube-apiserver in versions v1.6-v1.15, and versions prior to v1.16.13, v1.17.9 and v1.18.6 are vulne
CVE-2020-8561	A security issue was discovered in Kubernetes where actors that control the responses of MutatingWebhookConfigu
CVE-2020-8562	As mitigations to a report from 2019 and CVE-2020-8555, Kubernetes attempts to prevent proxied connections from
CVE-2020-8564	In Kubernetes clusters using a logging level of at least 4, processing a malformed docker config file will result in the
CVE-2020-8565	In Kubernetes, if the logging level is set to at least 9, authorization and bearer tokens will be written to log files. This
CVE-2021-0561	In append_to_verify_fifo_interleaved_ of stream_encoder.c, there is a possible out of bounds write due to a missing
CVE-2020-8908	A temp directory creation vulnerability exists in all versions of Guava, allowing an attacker with access to the machi
CVE-2021-20231	A flaw was found in gnutls. A use after free issue in client sending key_share extension may lead to memory corrupt
CVE-2021-20193	A flaw was found in the src/list.c of tar 1.33 and earlier. This flaw allows an attacker who can submit a crafted input
CVE-2021-20232	A flaw was found in gnutls. A use after free issue in client_send_params in lib/ext/pre_shared_key.c may lead to me
CVE-2021-20305	A flaw was found in Nettle in versions before 3.7.2, where several Nettle signature verification functions (GOST DS
CVE-2021-20320	A flaw was found in s390 eBPF JIT in bpf_jit_insn in arch/s390/net/bpf_jit_comp.c in the Linux kernel. In this flaw,
CVE-2021-21284	In Docker before versions 9.03.15, 20.10.3 there is a vulnerability involving the --users-remap option in which acce
CVE-2020-8694	Insufficient access control in the Linux kernel driver for some Intel(R) Processors may allow an authenticated user to
CVE-2021-21285	In Docker before versions 9.03.15, 20.10.3 there is a vulnerability in which pulling an intentionally malformed Dock
CVE-2021-22569	An issue in protobuf-java allowed the interleaving of com.google.protobuf.UnknownFieldSet fields in such a way th
CVE-2021-22876	curl 7.1.1 to and including 7.75.0 is vulnerable to an "Exposure of Private Personal Information to an Unauthorized A
CVE-2021-22570	Nullptr dereference when a null char is present in a proto symbol. The symbol is parsed incorrectly, leading to an un
CVE-2021-22890	curl 7.63.0 to and including 7.75.0 includes vulnerability that allows a malicious HTTPS proxy to MITM a connectio
CVE-2021-23437	The package pillow 5.2.0 and before 8.3.2 are vulnerable to Regular Expression Denial of Service (ReDoS) via the g
CVE-2021-22946	A user can tell curl >= 7.20.0 and <= 7.78.0 to require a successful upgrade to TLS when speaking to an IMAP, POP
CVE-2021-22898	curl 7.7 through 7.76.1 suffers from an information disclosure when the `-t` command line option, known as `CURL
CVE-2021-22947	When curl >= 7.20.0 and <= 7.78.0 connects to an IMAP or POP3 server to retrieve data using STARTTLS to upgra
CVE-2021-22924	libcurl keeps previously used connections in a connection pool for subsequent transfers to reuse, if one of them match
CVE-2021-24032	Beginning in v1.4.1 and prior to v1.4.9, due to an incomplete fix for CVE-2021-24031, the Zstandard command-line
CVE-2021-24031	In the Zstandard command-line utility prior to v1.4.1, output files were created with default permissions. Correct file
CVE-2021-23841	The OpenSSL public API function X509_issuer_and_serial_hash() attempts to create a unique hash value based on t
CVE-2021-23840	Calls to EVP_CipherUpdate, EVP_EncryptUpdate and EVP_DecryptUpdate may overflow the output length argume
CVE-2021-25219	In BIND 9.3.0 -> 9.11.35, 9.12.0 -> 9.16.21, and versions 9.9.3-S1 -> 9.11.35-S1 and 9.16.8-S1 -> 9.16.21-S1 of BI
CVE-2021-25317	A Incorrect Default Permissions vulnerability in the packaging of cups of SUSE Linux Enterprise Server 11-SP4-LT
CVE-2021-25735	A security issue was discovered in kube-apiserver that could allow node updates to bypass a Validating Admission V
CVE-2021-25220	BIND 9.11.0 -> 9.11.36 9.12.0 -> 9.16.26 9.17.0 -> 9.18.0 BIND Supported Preview Editions: 9.11.4-S1 -> 9.11.36-

CVE-2021-25736	Kube-proxy on Windows can unintentionally forward traffic to local processes listening on the same port ("spec.port"
CVE-2021-25737	A security issue was discovered in Kubernetes where a user may be able to redirect pod traffic to private networks on
CVE-2021-25740	A security issue was discovered with Kubernetes that could enable users to send network traffic to locations they wo
CVE-2021-25743	kubectl does not neutralize escape, meta or control sequences contained in the raw data it outputs to a terminal. This
CVE-2021-25741	A security issue was discovered in Kubernetes where a user may be able to create a container with subpath volume m
CVE-2021-26934	An issue was discovered in the Linux kernel 4.18 through 5.10.16, as used by Xen. The backend allocation (aka be-a
CVE-2021-27212	In OpenLDAP through 2.4.57 and 2.5.x through 2.5.1alpha, an assertion failure in slapd can occur in the issuerAndT
CVE-2021-27645	The nameserver caching daemon (nscd) in the GNU C Library (aka glibc or libc6) 2.29 through 2.33, when processin
CVE-2021-28170	In the Jakarta Expression Language implementation 3.0.3 and earlier, a bug in the ELParserTokenManager enables i
CVE-2021-28711	Rogue backends can cause DoS of guests via high frequency events T[his CNA information record relates to multipl
CVE-2021-28168	Eclipse Jersey 2.28 to 2.33 and Eclipse Jersey 3.0.0 to 3.0.1 contains a local information disclosure vulnerability. Th
CVE-2021-28712	Rogue backends can cause DoS of guests via high frequency events T[his CNA information record relates to multipl
CVE-2021-28713	Rogue backends can cause DoS of guests via high frequency events T[his CNA information record relates to multipl
CVE-2021-28972	In drivers/pci/hotplug/rpadlpar_sysfs.c in the Linux kernel through 5.11.8, the RPA PCI Hotplug driver has a user-to
CVE-2021-3178	fs/nfsd/nfs3xdr.c in the Linux kernel through 5.10.8, when there is an NFS export of a subdirectory of a filesystem, a
CVE-2021-29657	arch/x86/kvm/svm/nested.c in the Linux kernel before 5.11.12 has a use-after-free in which an AMD KVM guest can
CVE-2021-28861	Python 3.x through 3.10 has an open redirection vulnerability in lib/http/server.py due to no protection against multi
CVE-2021-31879	GNU Wget through 1.21.1 does not omit the Authorization header upon a redirect to a different origin, a related issu
CVE-2021-3236	vim 8.2.2348 is affected by null pointer dereference, allows local attackers to cause a denial of service (DoS) via the
CVE-2021-32797	JupyterLab is a user interface for Project Jupyter which will eventually replace the classic Jupyter Notebook. In affe
CVE-2021-32798	The Jupyter notebook is a web-based notebook environment for interactive computing. In affected versions untrusted
CVE-2021-3326	The iconv function in the GNU C Library (aka glibc or libc6) 2.32 and earlier, when processing invalid input sequen
CVE-2021-33430	A Buffer Overflow vulnerability exists in NumPy 1.9.x in the PyArray_NewFromDescr_int function of ctors.c when
CVE-2021-33294	In elfutils 0.183, an infinite loop was found in the function handle_syntab in readelf.c .Which allows attackers to ca
CVE-2021-33574	The mq_notify function in the GNU C Library (aka glibc) versions 2.32 and 2.33 has a use-after-free. It may use the
CVE-2021-33560	Libgcrypt before 1.8.8 and 1.9.x before 1.9.3 mishandles ElGamal encryption because it lacks exponent blinding to a
CVE-2021-33624	In kernel/bpf/verifier.c in the Linux kernel before 5.12.13, a branch can be mispredicted (e.g., because of type confu
CVE-2021-3411	A flaw was found in the Linux kernel in versions prior to 5.10. A violation of memory access was found while detec
CVE-2021-33910	basic/unit-name.c in systemd prior to 246.15, 247.8, 248.5, and 249.1 has a Memory Allocation with an Excessive S
CVE-2021-34141	An incomplete string comparison in the numpy.core component in NumPy before 1.22.0 allows attackers to trigger s
CVE-2021-3443	A NULL pointer dereference flaw was found in the way Jasper versions before 2.0.27 handled component references
CVE-2021-34428	For Eclipse Jetty versions <= 9.4.40, <= 10.0.2, <= 11.0.2, if an exception is thrown from the SessionListener#sessio
CVE-2021-3444	The bpf verifier in the Linux kernel did not properly handle mod32 destination register truncation when the source re
CVE-2021-34556	In the Linux kernel through 5.13.7, an unprivileged BPF program can obtain sensitive information from kernel mem
CVE-2021-3467	A NULL pointer dereference flaw was found in the way Jasper versions before 2.0.26 handled component references
CVE-2021-34693	net/can/bcm.c in the Linux kernel through 5.12.10 allows local users to obtain sensitive information from kernel stac
CVE-2021-3449	An OpenSSL TLS server may crash if sent a maliciously crafted renegotiation ClientHello message from a client. If
CVE-2021-34981	Linux Kernel Bluetooth CMTP Module Double Free Privilege Escalation Vulnerability. This vulnerability allows loc
CVE-2021-35477	In the Linux kernel through 5.13.7, an unprivileged BPF program can obtain sensitive information from kernel mem
CVE-2021-3520	There's a flaw in lz4. An attacker who submits a crafted file to an application linked with lz4 may be able to trigger a

CVE-2021-3572	A flaw was found in python-pip in the way it handled Unicode separators in git references. A remote attacker could p
CVE-2021-3580	A flaw was found in the way nettle's RSA decryption functions handled specially crafted ciphertext. An attacker cou
CVE-2021-35937	A race condition vulnerability was found in rpm. A local unprivileged user could use this flaw to bypass the checks t
CVE-2021-35939	It was found that the fix for CVE-2017-7500 and CVE-2017-7501 was incomplete: the check was only implemented
CVE-2021-35938	A symbolic link issue was found in rpm. It occurs when rpm sets the desired permissions and credentials after install
CVE-2021-35942	The wordexp function in the GNU C Library (aka glibc) through 2.33 may crash or read arbitrary memory in parse_
CVE-2021-36084	The CIL compiler in SELinux 3.2 has a use-after-free in __cil_verify_classperms (called from __cil_verify_classper
CVE-2021-36085	The CIL compiler in SELinux 3.2 has a use-after-free in __cil_verify_classperms (called from __verify_map_perm_
CVE-2021-36086	The CIL compiler in SELinux 3.2 has a use-after-free in cil_reset_classpermission (called from cil_reset_classperms
CVE-2021-36087	The CIL compiler in SELinux 3.2 has a heap-based buffer over-read in ebitmap_match_any (called indirectly from c
CVE-2021-36222	ec_verify in kdc/kdc_preauth_ec.c in the Key Distribution Center (KDC) in MIT Kerberos 5 (aka krb5) before 1.18.x
CVE-2021-3634	A flaw has been found in libssh in versions prior to 0.9.6. The SSH protocol keeps track of two shared secrets during
CVE-2021-3655	A vulnerability was found in the Linux kernel in versions prior to v5.14-rc1. Missing size validations on inbound SC
CVE-2021-3711	In order to decrypt SM2 encrypted data an application is expected to call the API function EVP_PKEY_decrypt(). T
CVE-2021-3712	ASN.1 strings are represented internally within OpenSSL as an ASN1_STRING structure which contains a buffer ho
CVE-2021-3714	A flaw was found in the Linux kernels memory deduplication mechanism. Previous work has shown that memory de
CVE-2021-3733	There's a flaw in urllib's AbstractBasicAuthHandler class. An attacker who controls a malicious HTTP server that an
CVE-2021-3737	A flaw was found in python. An improperly handled HTTP response in the HTTP client code of python may allow a
CVE-2021-37600	An integer overflow in util-linux through 2.37.1 can potentially cause a buffer overflow if an attacker were able to us
CVE-2021-38153	Some components in Apache Kafka use `Arrays.equals` to validate a password or key, which is vulnerable to timing
CVE-2021-37750	The Key Distribution Center (KDC) in MIT Kerberos 5 (aka krb5) before 1.18.5 and 1.19.x before 1.19.3 has a NUL
CVE-2021-38166	In kernel/bpf/hashtab.c in the Linux kernel through 5.13.8, there is an integer overflow and out-of-bounds write whe
CVE-2021-38198	arch/x86/kvm/mmu/paging_tmpl.h in the Linux kernel before 5.12.11 incorrectly computes the access permissions o
CVE-2021-38199	fs/nfs/nfs4client.c in the Linux kernel before 5.13.4 has incorrect connection-setup ordering, which allows operators
CVE-2021-38206	The mac80211 subsystem in the Linux kernel before 5.12.13, when a device supporting only 5 GHz is used, allows a
CVE-2021-3826	Heap/stack buffer overflow in the dlang_lname function in d-demangle.c in libiberty allows attackers to potentially c
CVE-2021-38593	Qt 5.x before 5.15.6 and 6.x through 6.1.2 has an out-of-bounds write in QOutlineMapper::convertPath (called from
CVE-2021-3864	A flaw was found in the way the dumpable flag setting was handled when certain SUID binaries executed its descen
CVE-2021-3927	vim is vulnerable to Heap-based Buffer Overflow
CVE-2021-39537	An issue was discovered in ncurses through v6.2-1. _nc_captainfo in captainfo.c has a heap-based buffer overflow.
CVE-2021-3974	vim is vulnerable to Use After Free
CVE-2021-3997	A flaw was found in systemd. An uncontrolled recursion in systemd-tmpfiles may lead to a denial of service at boot
CVE-2021-3999	A flaw was found in glibc. An off-by-one buffer overflow and underflow in getcwd() may lead to memory corruption
CVE-2021-4001	A race condition was found in the Linux kernel's ebpf verifier between bpf_map_update_elem and bpf_map_freeze c
CVE-2021-4044	Internally libssl in OpenSSL calls X509_verify_cert() on the client side to verify a certificate supplied by a server. T
CVE-2021-40490	A race condition was discovered in ext4_write_inline_data_end in fs/ext4/inline.c in the ext4 subsystem in the Linux
CVE-2021-40528	The ElGamal implementation in Libgcrypt before 1.9.4 allows plaintext recovery because, during interaction between
CVE-2021-4095	A NULL pointer dereference was found in the Linux kernel's KVM when dirty ring logging is enabled without an ac
CVE-2021-41091	Moby is an open-source project created by Docker to enable software containerization. A bug was found in Moby (D
CVE-2021-41182	jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `altField` op

CVE-2021-41183	jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of various `*Text`
CVE-2021-4135	A memory leak vulnerability was found in the Linux kernel's eBPF for the Simulated networking device driver in the
CVE-2021-41184	jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `of` option o
CVE-2021-41495	Null Pointer Dereference vulnerability exists in numpy.sort in NumPy <1.19 and 1.19 in the PyArray_DescrNew functi
CVE-2021-41496	Buffer overflow in the array_from_pyobj function of fortranobject.c in NumPy < 1.19, which allows attackers to con
CVE-2021-4159	A vulnerability was found in the Linux kernel's EBPF verifier when handling internal data structures. Internal memo
CVE-2021-4160	There is a carry propagation bug in the MIPS32 and MIPS64 squaring procedure. Many EC algorithms are affected,
CVE-2021-4166	vim is vulnerable to Out-of-bounds Read
CVE-2021-41720	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was wit
CVE-2021-4189	A flaw was found in Python, specifically in the FTP (File Transfer Protocol) client library in PASV (passive) mode.
CVE-2021-4217	A flaw was found in unzip. The vulnerability occurs due to improper handling of Unicode strings, which can lead to
CVE-2021-4209	A NULL pointer dereference flaw was found in GnuTLS. As Nettle's hash update functions internally call memcpy,
CVE-2021-42357	When using Apache Knox SSO prior to 1.6.1, a request could be crafted to redirect a user to a malicious page due to
CVE-2021-43618	GNU Multiple Precision Arithmetic Library (GMP) through 6.2.1 has an mpz/inp_raw.c integer overflow and result
CVE-2021-43797	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high p
CVE-2021-4440	In the Linux kernel, the following vulnerability has been resolved: x86/xen: Drop USERGS_SYSRET64 paravirt cal
CVE-2021-4439	In the Linux kernel, the following vulnerability has been resolved: isdn: cpai: check ctr->cnr to avoid array index ou
CVE-2021-4442	In the Linux kernel, the following vulnerability has been resolved: tcp: add sanity tests to TCP_QUEUE_SEQ Qingy
CVE-2021-45078	stab_xcoff_builtin_type in stabs.c in GNU Binutils through 2.37 allows attackers to cause a denial of service (heap-b
CVE-2021-45261	An Invalid Pointer vulnerability exists in GNU patch 2.7 via the another_hunk function, which causes a Denial of Se
CVE-2021-46195	GCC v12.0 was discovered to contain an uncontrolled recursion via the component libiberty/rust-demangle.c. This v
CVE-2021-46877	jackson-databind 2.10.x through 2.12.x before 2.12.6 and 2.13.x before 2.13.1 allows attackers to cause a denial of s
CVE-2021-45402	The check_alu_op() function in kernel/bpf/verifier.c in the Linux kernel through v5.16-rc5 did not properly update b
CVE-2021-46904	In the Linux kernel, the following vulnerability has been resolved: net: hso: fix null-ptr-deref during tty device unreg
CVE-2021-46907	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2021-46908	In the Linux kernel, the following vulnerability has been resolved: bpf: Use correct permission flag for mixed signed
CVE-2021-46909	In the Linux kernel, the following vulnerability has been resolved: ARM: footbridge: fix PCI interrupt mapping Sinc
CVE-2021-46910	In the Linux kernel, the following vulnerability has been resolved: ARM: 9063/1: mm: reduce maximum number of
CVE-2021-46915	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_limit: avoid possible divide error in
CVE-2021-46917	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: fix wq cleanup of WQCFG regi
CVE-2021-46920	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix clobbering of SWERR over
CVE-2021-46918	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: clear MSIX permission entry or
CVE-2021-46921	In the Linux kernel, the following vulnerability has been resolved: locking/qrwlock: Fix ordering in queued_write_lo
CVE-2021-46925	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix kernel panic caused by race of smc_s
CVE-2021-46926	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: intel-sdw-acpi: harden detection of c
CVE-2021-46929	In the Linux kernel, the following vulnerability has been resolved: sctp: use call_rcu to free endpoint This patch is to
CVE-2021-46932	In the Linux kernel, the following vulnerability has been resolved: Input: appletouch - initialize work before device r
CVE-2021-46937	In the Linux kernel, the following vulnerability has been resolved: mm/damon/dbgfs: fix 'struct pid' leaks in 'dbgfs_t
CVE-2021-46939	In the Linux kernel, the following vulnerability has been resolved: tracing: Restructure trace_clock_global() to never
CVE-2021-46960	In the Linux kernel, the following vulnerability has been resolved: cifs: Return correct error code from smb2_get_en

CVE-2021-46961	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Do not enable irqs when handling
CVE-2021-46963	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix crash in qla2xxx_mqueuecomm
CVE-2021-46972	In the Linux kernel, the following vulnerability has been resolved: ovl: fix leaked dentry Since commit 6815f479ca9
CVE-2021-46973	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: Avoid potential use after free in MHI se
CVE-2021-46976	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix crash in auto_retire The retire logic
CVE-2021-46984	In the Linux kernel, the following vulnerability has been resolved: kyber: fix out of bounds access when preempted
CVE-2021-46978	In the Linux kernel, the following vulnerability has been resolved: KVM: nVMX: Always make an attempt to map e
CVE-2021-46988	In the Linux kernel, the following vulnerability has been resolved: userfaultfd: release page in error path to avoid BU
CVE-2021-46997	In the Linux kernel, the following vulnerability has been resolved: arm64: entry: always set GIC_PRIO_PSR_I_SET
CVE-2021-46999	In the Linux kernel, the following vulnerability has been resolved: sctp: do asoc update earlier in sctp_sf_do_dupcoo
CVE-2021-47001	In the Linux kernel, the following vulnerability has been resolved: xprtrdma: Fix cwnd update ordering After a recor
CVE-2021-47003	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix potential null dereference o
CVE-2021-47011	In the Linux kernel, the following vulnerability has been resolved: mm: memcontrol: slab: fix obtain a reference to a
CVE-2021-47010	In the Linux kernel, the following vulnerability has been resolved: net: Only allow init netns to set default tcp cong t
CVE-2021-47012	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Fix a use after free in siw_alloc_mr
CVE-2021-47014	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_ct: fix wild memory access when c
CVE-2021-47017	In the Linux kernel, the following vulnerability has been resolved: ath10k: Fix a use after free in ath10k_htc_send_b
CVE-2021-47018	In the Linux kernel, the following vulnerability has been resolved: powerpc/64: Fix the definition of the fixmap area
CVE-2021-47020	In the Linux kernel, the following vulnerability has been resolved: soundwire: stream: fix memory leak in stream cor
CVE-2021-47024	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: free queued packets when closing so
CVE-2021-47038	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: avoid deadlock between hci_dev->loc
CVE-2021-47036	In the Linux kernel, the following vulnerability has been resolved: udp: skip L4 aggregation for UDP tunnel packets
CVE-2021-47045	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix null pointer dereference in lpfc_pre
CVE-2021-47041	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: fix incorrect locking in state_change s
CVE-2021-47069	In the Linux kernel, the following vulnerability has been resolved: ipc/mqueue, msg, sem: avoid relying on a stack re
CVE-2021-47070	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Fix another memory leak in erro
CVE-2021-47071	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Fix a memory leak in error handl
CVE-2021-47073	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-smbios-wmi: Fix oops on rmm
CVE-2021-47074	In the Linux kernel, the following vulnerability has been resolved: nvme-loop: fix memory leak in nvme_loop_creat
CVE-2021-47075	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix memory leak in nvmet_alloc_ctrl() WH
CVE-2021-47077	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Add pointer checks in qedf_update_lin
CVE-2021-47078	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Clear all QP fields if creation failed
CVE-2021-47079	In the Linux kernel, the following vulnerability has been resolved: platform/x86: ideapad-laptop: fix a NULL pointer
CVE-2021-47092	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Always clear vmx->fail on emulatio
CVE-2021-47094	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Don't advance iterator after rest
CVE-2021-47097	In the Linux kernel, the following vulnerability has been resolved: Input: elantech - fix stack out of bound access in
CVE-2021-47101	In the Linux kernel, the following vulnerability has been resolved: asix: fix uninit-value in asix_mdio_read() asix_re
CVE-2021-47103	In the Linux kernel, the following vulnerability has been resolved: inet: fully convert sk->sk_rx_dst to RCU rules sy
CVE-2021-47107	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix READDIR buffer overflow If a client
CVE-2021-47116	In the Linux kernel, the following vulnerability has been resolved: ext4: fix memory leak in ext4_mb_init_backend c

CVE-2021-47120	In the Linux kernel, the following vulnerability has been resolved: HID: magicmouse: fix NULL-deref on disconnect
CVE-2021-47133	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: Fix memory leak in amd_sfh_wor
CVE-2021-47134	In the Linux kernel, the following vulnerability has been resolved: efi/fdt: fix panic when no valid fdt found setup_a
CVE-2021-47136	In the Linux kernel, the following vulnerability has been resolved: net: zero-initialize tc skb extension on allocation
CVE-2021-47138	In the Linux kernel, the following vulnerability has been resolved: cxgb4: avoid accessing registers when clearing fi
CVE-2021-47140	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Clear DMA ops when switching dor
CVE-2021-47141	In the Linux kernel, the following vulnerability has been resolved: gve: Add NULL pointer checks when freeing irq
CVE-2021-47143	In the Linux kernel, the following vulnerability has been resolved: net/smc: remove device from smcd_dev_list after
CVE-2021-47144	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2021-47146	In the Linux kernel, the following vulnerability has been resolved: mld: fix panic in mld_newpack() mld_newpack()
CVE-2021-47149	In the Linux kernel, the following vulnerability has been resolved: net: fujitsu: fix potential null-ptr-deref In fmvj18
CVE-2021-47162	In the Linux kernel, the following vulnerability has been resolved: tipc: skb_linearize the head skb when reassemblin
CVE-2021-47163	In the Linux kernel, the following vulnerability has been resolved: tipc: wait and exit until all work queues are done
CVE-2021-47167	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix an Oopsable condition in __nfs_pageio
CVE-2021-47168	In the Linux kernel, the following vulnerability has been resolved: NFS: fix an incorrect limit in filelayout_decode_l
CVE-2021-47170	In the Linux kernel, the following vulnerability has been resolved: USB: usbfs: Don't WARN about excessively larg
CVE-2021-47173	In the Linux kernel, the following vulnerability has been resolved: misc/uss720: fix memory leak in uss720_probe us
CVE-2021-47176	In the Linux kernel, the following vulnerability has been resolved: s390/dasd: add missing discipline function Fix cra
CVE-2021-47177	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix sysfs leak in alloc_iommu() iom
CVE-2021-47178	In the Linux kernel, the following vulnerability has been resolved: scsi: target: core: Avoid smp_processor_id() in pr
CVE-2021-47182	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix scsi_mode_sense() buffer length ha
CVE-2021-47183	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix link down processing to address NU
CVE-2021-47191	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Fix out-of-bound read in resp_re
CVE-2021-47186	In the Linux kernel, the following vulnerability has been resolved: tipc: check for null after calling kmemdup kmem
CVE-2021-47194	In the Linux kernel, the following vulnerability has been resolved: cfg80211: call cfg80211_stop_ap when switch fro
CVE-2021-47196	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Set send and receive CQ before forw
CVE-2021-47198	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix use-after-free in lpfc_unreg_rpi() r
CVE-2021-47197	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: nullify cq->dbg pointer in mlx5_debu
CVE-2021-47199	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: CT, Fix multiple allocations and mem
CVE-2021-47201	In the Linux kernel, the following vulnerability has been resolved: iavf: free q_vectors before queues in iavf_disable
CVE-2021-47207	In the Linux kernel, the following vulnerability has been resolved: ALSA: gus: fix null pointer dereference on point
CVE-2021-47211	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: fix null pointer dereference on
CVE-2021-47210	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tipd: Remove WARN_ON in tps6598
CVE-2021-47215	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: kTLS, Fix crash in RX resync flow Fo
CVE-2021-47217	In the Linux kernel, the following vulnerability has been resolved: x86/hyperv: Fix NULL deref in set_hv_tscchange
CVE-2021-47218	In the Linux kernel, the following vulnerability has been resolved: selinux: fix NULL-pointer dereference when has
CVE-2021-47219	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Fix out-of-bound read in resp_re
CVE-2021-47222	In the Linux kernel, the following vulnerability has been resolved: net: bridge: fix vlan tunnel dst refcnt when egress
CVE-2021-47223	In the Linux kernel, the following vulnerability has been resolved: net: bridge: fix vlan tunnel dst null pointer deref
CVE-2021-47225	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix deadlock in AP/VLAN handling S

CVE-2021-47226	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Invalidate FPU state after a failed XRST
CVE-2021-47227	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Prevent state corruption in __fpu_restor
CVE-2021-47230	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Immediately reset the MMU context
CVE-2021-47236	In the Linux kernel, the following vulnerability has been resolved: net: cdc_eem: fix tx fixup skb leak when usbnet t
CVE-2021-47238	In the Linux kernel, the following vulnerability has been resolved: net: ipv4: fix memory leak in ip_mc_add1_src BL
CVE-2021-47239	In the Linux kernel, the following vulnerability has been resolved: net: usb: fix possible use-after-free in smsc75xx_
CVE-2021-47241	In the Linux kernel, the following vulnerability has been resolved: ethtool: strset: fix message length calculation Out
CVE-2021-47242	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix soft lookup in subflow_error_report()
CVE-2021-47243	In the Linux kernel, the following vulnerability has been resolved: sch_cake: Fix out of bounds when parsing TCP o
CVE-2021-47245	In the Linux kernel, the following vulnerability has been resolved: netfilter: synproxy: Fix out of bounds when parsin
CVE-2021-47246	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix page reclaim for dead peer hairpin
CVE-2021-47247	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix use-after-free of encap entry in nei
CVE-2021-47248	In the Linux kernel, the following vulnerability has been resolved: udp: fix race between close() and udp_abort() Ka
CVE-2021-47251	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix skb length check in ieee80211_sca
CVE-2021-47250	In the Linux kernel, the following vulnerability has been resolved: net: ipv4: fix memory leak in netlbl_cipsov4_add
CVE-2021-47254	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix use-after-free in gfs2_glock_shrink_sca
CVE-2021-47255	In the Linux kernel, the following vulnerability has been resolved: kvm: LAPIC: Restore guard to prevent illegal AP
CVE-2021-47257	In the Linux kernel, the following vulnerability has been resolved: net: ieee802154: fix null deref in parse dev addr F
CVE-2021-47258	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix error handling of scsi_host_alloc()
CVE-2021-47259	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix use-after-free in nfs4_init_client() KAS
CVE-2021-47260	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix a potential NULL dereference in nfs_ge
CVE-2021-47261	In the Linux kernel, the following vulnerability has been resolved: IB/mlx5: Fix initializing CQ fragments buffer Th
CVE-2021-47264	In the Linux kernel, the following vulnerability has been resolved: ASoC: core: Fix Null-point-dereference in fmt_si
CVE-2021-47265	In the Linux kernel, the following vulnerability has been resolved: RDMA: Verify port when creating flow rule Vali
CVE-2021-47262	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Ensure liveness of nested VM-Enter
CVE-2021-47266	In the Linux kernel, the following vulnerability has been resolved: RDMA/ipoib: Fix warning caused by destroying
CVE-2021-47276	In the Linux kernel, the following vulnerability has been resolved: ftrace: Do not blindly read the ip address in ftrace
CVE-2021-47277	In the Linux kernel, the following vulnerability has been resolved: kvm: avoid speculation-based attacks from out-of
CVE-2021-47281	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: Fix race of snd_seq_timer_open() Th
CVE-2021-47280	In the Linux kernel, the following vulnerability has been resolved: drm: Fix use-after-free read in drm_getunique() T
CVE-2021-47284	In the Linux kernel, the following vulnerability has been resolved: isdn: mISDN: netjet: Fix crash in nj_probe: 'nj_se
CVE-2021-47287	In the Linux kernel, the following vulnerability has been resolved: driver core: auxiliary bus: Fix memory leak when
CVE-2021-47289	In the Linux kernel, the following vulnerability has been resolved: ACPI: fix NULL pointer dereference Commit 71f
CVE-2021-47290	In the Linux kernel, the following vulnerability has been resolved: scsi: target: Fix NULL dereference on XCOPY co
CVE-2021-47295	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix memory leak in tcindex_partial_de
CVE-2021-47293	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_skbmod: Skip non-Ethernet packet
CVE-2021-47298	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix potential memory leak on unlik
CVE-2021-47301	In the Linux kernel, the following vulnerability has been resolved: igb: Fix use-after-free error during reset Cleans th
CVE-2021-47302	In the Linux kernel, the following vulnerability has been resolved: igc: Fix use-after-free error during reset Cleans th
CVE-2021-47304	In the Linux kernel, the following vulnerability has been resolved: tcp: fix tcp_init_transfer() to not reset icsk_ca_ini

CVE-2021-47303	In the Linux kernel, the following vulnerability has been resolved: bpf: Track subprog poke descriptors correctly and
CVE-2021-47305	In the Linux kernel, the following vulnerability has been resolved: dma-buf/sync_file: Don't leak fences on merge fa
CVE-2021-47307	In the Linux kernel, the following vulnerability has been resolved: cifs: prevent NULL deref in cifs_compose_moun
CVE-2021-47308	In the Linux kernel, the following vulnerability has been resolved: scsi: libfc: Fix array index out of bound exception
CVE-2021-47310	In the Linux kernel, the following vulnerability has been resolved: net: ti: fix UAF in tlan_remove_one priv is netdev
CVE-2021-47311	In the Linux kernel, the following vulnerability has been resolved: net: qcom/emac: fix UAF in emac_remove_adpt is
CVE-2021-47313	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Fix potential memleak in cppc_cp
CVE-2021-47316	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix NULL dereference in nfs3svc_encode_g
CVE-2021-47317	In the Linux kernel, the following vulnerability has been resolved: powerpc/bpf: Fix detecting BPF atomic instructio
CVE-2021-47319	In the Linux kernel, the following vulnerability has been resolved: virtio-blk: Fix memory leak among suspend/resur
CVE-2021-47320	In the Linux kernel, the following vulnerability has been resolved: nfs: fix acl memory leak of posix_acl_create() W
CVE-2021-47321	In the Linux kernel, the following vulnerability has been resolved: watchdog: Fix possible use-after-free by calling d
CVE-2021-47322	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Fix an Oops in pnfs_mark_request_commn
CVE-2021-47325	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Fix arm_smmu_device refcou
CVE-2021-47327	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Fix arm_smmu_device refcou
CVE-2021-47332	In the Linux kernel, the following vulnerability has been resolved: ALSA: usx2y: Don't call free_pages_exact() with
CVE-2021-47333	In the Linux kernel, the following vulnerability has been resolved: misc: alcor_pci: fix null-ptr-deref when there is n
CVE-2021-47338	In the Linux kernel, the following vulnerability has been resolved: fbmem: Do not delete the mode that is still in use
CVE-2021-47339	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-core: explicitly clear ioctl input data
CVE-2021-47342	In the Linux kernel, the following vulnerability has been resolved: ext4: fix possible UAF when remounting r/o a m
CVE-2021-47349	In the Linux kernel, the following vulnerability has been resolved: mwifiex: bring down link before deleting interfac
CVE-2021-47343	In the Linux kernel, the following vulnerability has been resolved: dm btree remove: assign new_root only when rem
CVE-2021-47350	In the Linux kernel, the following vulnerability has been resolved: powerpc/mm: Fix lockup on kernel exec fault The
CVE-2021-47352	In the Linux kernel, the following vulnerability has been resolved: virtio-net: Add validation for used length This ad
CVE-2021-47353	In the Linux kernel, the following vulnerability has been resolved: udf: Fix NULL pointer dereference in udf_symlin
CVE-2021-47356	In the Linux kernel, the following vulnerability has been resolved: mISDN: fix possible use-after-free in HFC_clean
CVE-2021-47359	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix soft lockup during fsstress Below traces
CVE-2021-47360	In the Linux kernel, the following vulnerability has been resolved: binder: make sure fd closes complete During BC
CVE-2021-47367	In the Linux kernel, the following vulnerability has been resolved: virtio-net: fix pages leaking when building skb in
CVE-2021-47370	In the Linux kernel, the following vulnerability has been resolved: mptcp: ensure tx skbs always have the MPTCP ex
CVE-2021-47374	In the Linux kernel, the following vulnerability has been resolved: dma-debug: prevent an error message from causin
CVE-2021-47376	In the Linux kernel, the following vulnerability has been resolved: bpf: Add oversize check before call kvccalloc() Co
CVE-2021-47380	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: Fix potential NULL pointer deref
CVE-2021-47373	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3-its: Fix potential VPE leak on erro
CVE-2021-47381	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Fix DSP oops stack dump output co
CVE-2021-47383	In the Linux kernel, the following vulnerability has been resolved: tty: Fix out-of-bound vmalloc access in imageblit
CVE-2021-47384	In the Linux kernel, the following vulnerability has been resolved: hwmon: (w83793) Fix NULL pointer dereference
CVE-2021-47386	In the Linux kernel, the following vulnerability has been resolved: hwmon: (w83791d) Fix NULL pointer dereference
CVE-2021-47385	In the Linux kernel, the following vulnerability has been resolved: hwmon: (w83792d) Fix NULL pointer dereference
CVE-2021-47387	In the Linux kernel, the following vulnerability has been resolved: cpufreq: schedutil: Use kobject release() method t

CVE-2021-47388	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix use-after-free in CCMP/GCMP R2
CVE-2021-47389	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: fix missing sev_decommission in sev
CVE-2021-47395	In the Linux kernel, the following vulnerability has been resolved: mac80211: limit injected vht mcs/nss in ieee80211
CVE-2021-47393	In the Linux kernel, the following vulnerability has been resolved: hwmon: (mlxreg-fan) Return non-zero value whe
CVE-2021-47397	In the Linux kernel, the following vulnerability has been resolved: sctp: break out if skb_header_pointer returns NU
CVE-2021-47398	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Fix kernel pointer leak Pointers sho
CVE-2021-47404	In the Linux kernel, the following vulnerability has been resolved: HID: betop: fix slab-out-of-bounds Write in betop
CVE-2021-47407	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Handle SRCU initialization failure du
CVE-2021-47408	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: serialize hash resizes and clea
CVE-2021-47412	In the Linux kernel, the following vulnerability has been resolved: block: don't call rq_qos_ops->done_bio if the bio
CVE-2021-47413	In the Linux kernel, the following vulnerability has been resolved: usb: chipidea: ci_hdrc_imx: Also search for 'phys
CVE-2021-47415	In the Linux kernel, the following vulnerability has been resolved: iwlfwifi: mvmm: Fix possible NULL dereference In
CVE-2021-47416	In the Linux kernel, the following vulnerability has been resolved: phy: mdio: fix memory leak Syzbot reported mem
CVE-2021-47419	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_taprio: properly cancel timer from
CVE-2021-47420	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: fix a potential ttm->sg memory leak
CVE-2021-47421	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: handle the case of pci_channel_io_
CVE-2021-47424	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix freeing of uninitialized misc IRQ vector
CVE-2021-47426	In the Linux kernel, the following vulnerability has been resolved: bpf, s390: Fix potential memory leak about jit_da
CVE-2021-47427	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi: Fix iscsi_task use after free Commit d3
CVE-2021-47431	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix gart.bo pin_count leak gmc_v{9
CVE-2021-47432	In the Linux kernel, the following vulnerability has been resolved: lib/generic-radix-tree.c: Don't overflow in peek()
CVE-2021-47438	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix memory leak in mlx5_core_destro
CVE-2021-47441	In the Linux kernel, the following vulnerability has been resolved: mlxsw: thermal: Fix out-of-bounds memory acces
CVE-2021-47449	In the Linux kernel, the following vulnerability has been resolved: ice: fix locking for Tx timestamp tracking flush C
CVE-2021-47452	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: skip netdev events generated
CVE-2021-47453	In the Linux kernel, the following vulnerability has been resolved: ice: Avoid crash from unnecessary IDA free In th
CVE-2021-47455	In the Linux kernel, the following vulnerability has been resolved: ptp: Fix possible memory leak in ptp_clock_regis
CVE-2021-47456	In the Linux kernel, the following vulnerability has been resolved: can: peak_pci: peak_pci_remove(): fix UAF Whe
CVE-2021-47461	In the Linux kernel, the following vulnerability has been resolved: userfaultfd: fix a race between writeprotect and e
CVE-2021-47462	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: do not allow illegal MPOL_F_N
CVE-2021-47464	In the Linux kernel, the following vulnerability has been resolved: audit: fix possible null-pointer dereference in aud
CVE-2021-47465	In the Linux kernel, the following vulnerability has been resolved: KVM: PPC: Book3S HV: Fix stack handling in ic
CVE-2021-47466	In the Linux kernel, the following vulnerability has been resolved: mm, slub: fix potential memoryleak in kmem_cac
CVE-2021-47468	In the Linux kernel, the following vulnerability has been resolved: isdn: mISDN: Fix sleeping function called from in
CVE-2021-47469	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2021-47481	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Initialize the ODP xarray when cre
CVE-2021-47483	In the Linux kernel, the following vulnerability has been resolved: regmap: Fix possible double-free in regcache_rbt
CVE-2021-47488	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2021-47490	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: fix memleak in ttm_transferred_destroy
CVE-2021-47491	In the Linux kernel, the following vulnerability has been resolved: mm: khugepaged: skip huge page collapse for spe

CVE-2021-47492	In the Linux kernel, the following vulnerability has been resolved: mm, thp: bail out early in collapse_file for writeb
CVE-2021-47494	In the Linux kernel, the following vulnerability has been resolved: cfg80211: fix management registrations locking T
CVE-2021-47495	In the Linux kernel, the following vulnerability has been resolved: usbnet: sanity check for maxpacket maxpacket of
CVE-2021-47496	In the Linux kernel, the following vulnerability has been resolved: net/tls: Fix flipped sign in tls_err_abort() calls sk-
CVE-2021-47506	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix use-after-free due to delegation race A d
CVE-2021-47507	In the Linux kernel, the following vulnerability has been resolved: nfsd: Fix nfsd startup race (again) Commit bd5ae
CVE-2021-47497	In the Linux kernel, the following vulnerability has been resolved: nvmmem: Fix shift-out-of-bound (UBSAN) with by
CVE-2021-47517	In the Linux kernel, the following vulnerability has been resolved: ethtool: do not perform operations on net devices
CVE-2021-47514	In the Linux kernel, the following vulnerability has been resolved: devlink: fix netns refcount leak in devlink_nl_cm
CVE-2021-47527	In the Linux kernel, the following vulnerability has been resolved: serial: core: fix transmit-buffer reset and memleak
CVE-2021-47529	In the Linux kernel, the following vulnerability has been resolved: iwlfwifi: Fix memory leaks in error handling path
CVE-2021-47536	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix wrong list_del in smc_lgr_cleanup_e
CVE-2021-47548	In the Linux kernel, the following vulnerability has been resolved: ethernet: hisilicon: hns: hns_dsaf_misc: fix a poss
CVE-2021-47550	In the Linux kernel, the following vulnerability has been resolved: drm/amd/amdgpu: fix potential memleak In funct
CVE-2021-47552	In the Linux kernel, the following vulnerability has been resolved: blk-mq: cancel blk-mq dispatch work in both blk_
CVE-2021-47554	In the Linux kernel, the following vulnerability has been resolved: vdpa_sim: avoid putting an uninitialized iova_dor
CVE-2021-47557	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_ets: don't peek at classes beyond 'r
CVE-2021-47558	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: Disable Tx queues when reconfiguri
CVE-2021-47559	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix NULL pointer dereferencing in smc_
CVE-2021-47560	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum: Protect driver from buggy firm
CVE-2021-47562	In the Linux kernel, the following vulnerability has been resolved: ice: fix vsi->txq_map sizing The approach of hav
CVE-2021-47565	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix kernel panic during drive powe
CVE-2021-47563	In the Linux kernel, the following vulnerability has been resolved: ice: avoid bpf_prog refcount underflow Ice driver
CVE-2021-47566	In the Linux kernel, the following vulnerability has been resolved: proc/vmcore: fix clearing user buffer by properly
CVE-2021-47576	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Sanity check block descriptor le
CVE-2021-47582	In the Linux kernel, the following vulnerability has been resolved: USB: core: Make do_proc_control() and do_proc
CVE-2021-47579	In the Linux kernel, the following vulnerability has been resolved: ovl: fix warning in ovl_create_real() Syzbot trigge
CVE-2021-47578	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Don't call kcalloc() if size arg is
CVE-2021-47588	In the Linux kernel, the following vulnerability has been resolved: sit: do not call ipip6_dev_free() from sit_init_net
CVE-2021-47591	In the Linux kernel, the following vulnerability has been resolved: mptcp: remove tcp ulp setsockopt support TCP_U
CVE-2021-47595	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_ets: don't remove idle classes from
CVE-2021-47594	In the Linux kernel, the following vulnerability has been resolved: mptcp: never allow the PM to close a listener sub
CVE-2021-47597	In the Linux kernel, the following vulnerability has been resolved: inet_diag: fix kernel-infoleak for UDP sockets KB
CVE-2021-47600	In the Linux kernel, the following vulnerability has been resolved: dm btree remove: fix use after free in rebalance_c
CVE-2021-47606	In the Linux kernel, the following vulnerability has been resolved: net: netlink: af_netlink: Prevent empty skb by ad
CVE-2021-47603	In the Linux kernel, the following vulnerability has been resolved: audit: improve robustness of the audit queue han
CVE-2021-47607	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix kernel address leakage in atomic cmpxch
CVE-2021-47608	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix kernel address leakage in atomic fetch T
CVE-2021-47609	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scpi: Fix string overflow in SCPI
CVE-2021-47615	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.

CVE-2021-47616	In the Linux kernel, the following vulnerability has been resolved: RDMA: Fix use-after-free in rxe_queue_cleanup
CVE-2021-47617	In the Linux kernel, the following vulnerability has been resolved: PCI: pciehp: Fix infinite loop in IRQ handler upon
CVE-2021-47619	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix queues reservation for XDP When XDP
CVE-2021-47620	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: refactor malicious adv data check Che
CVE-2021-47623	In the Linux kernel, the following vulnerability has been resolved: powerpc/fixmap: Fix VM debug warning on unm
CVE-2021-47624	In the Linux kernel, the following vulnerability has been resolved: net/sunrpc: fix reference count leaks in rpc_sysfs_
CVE-2021-47634	In the Linux kernel, the following vulnerability has been resolved: ubi: Fix race condition between ctrl_cdev_ioctl and
CVE-2021-47639	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Zap _all_ roots when unmapping
CVE-2021-47646	In the Linux kernel, the following vulnerability has been resolved: Revert "Revert "block, bfq: honor already-setup q
CVE-2021-47650	In the Linux kernel, the following vulnerability has been resolved: ASoC: soc-compress: prevent the potentially use
CVE-2021-47657	In the Linux kernel, the following vulnerability has been resolved: drm/virtio: Ensure that objs is not NULL in virtio
CVE-2021-47658	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: fix a potential gpu_metrics_table m
CVE-2021-47659	In the Linux kernel, the following vulnerability has been resolved: drm/plane: Move range check for format_count e
CVE-2021-47668	In the Linux kernel, the following vulnerability has been resolved: can: dev: can_restart: fix use after free bug After
CVE-2021-47670	In the Linux kernel, the following vulnerability has been resolved: can: peak_usb: fix use after free bugs After callin
CVE-2022-0171	A flaw was found in the Linux kernel. The existing KVM SEV API has a vulnerability that allows a non-root (host)
CVE-2022-0264	A vulnerability was found in the Linux kernel's eBPF verifier when handling internal data structures. Internal memor
CVE-2022-0235	node-fetch is vulnerable to Exposure of Sensitive Information to an Unauthorized Actor
CVE-2022-0351	Access of Memory Location Before Start of Buffer in GitHub repository vim/vim prior to 8.2.
CVE-2022-0391	A flaw was found in Python, specifically within the urllib.parse module. This module helps break Uniform Resource
CVE-2022-0400	An out-of-bounds read vulnerability was discovered in linux kernel in the smc protocol stack, causing remote dos.
CVE-2022-0529	A flaw was found in Unzip. The vulnerability occurs during the conversion of a wide string to a local string that lead
CVE-2022-0480	A flaw was found in the filelock_init in fs/locks.c function in the Linux kernel. This issue can lead to host memory e
CVE-2022-0530	A flaw was found in Unzip. The vulnerability occurs during the conversion of a wide string to a local string that lead
CVE-2022-0563	A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library
CVE-2022-0778	The BN_mod_sqrt() function, which computes a modular square root, contains a bug that can cause it to loop forever
CVE-2022-1056	Out-of-bounds Read error in tiffcrop in libtiff 4.3.0 allows attackers to cause a denial-of-service via a crafted tiff file
CVE-2022-0995	An out-of-bounds (OOB) memory write flaw was found in the Linux kernel's watch_queue event notification subsystem
CVE-2022-1154	Use after free in utf_ptr2char in GitHub repository vim/vim prior to 8.2.4646.
CVE-2022-1205	A NULL pointer dereference flaw was found in the Linux kernel's Amateur Radio AX.25 protocol functionality in the
CVE-2022-1247	An issue found in linux-kernel that leads to a race condition in rose_connect(). The rose driver uses rose_neigh->use
CVE-2022-1271	An arbitrary file write vulnerability was found in GNU gzip's zgrep utility. When zgrep is applied on the attacker's cl
CVE-2022-1304	An out-of-bounds read/write vulnerability was found in e2fsprogs 1.46.5. This issue leads to a segmentation fault and
CVE-2022-1292	The c_rehash script does not properly sanitise shell metacharacters to prevent command injection. This script is distr
CVE-2022-1354	A heap buffer overflow flaw was found in Libtiffs' tiffinfo.c in TIFFReadRawDataStriped() function. This flaw allow
CVE-2022-1471	SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializin
CVE-2022-1552	A flaw was found in PostgreSQL. There is an issue with incomplete efforts to operate safely when a privileged user
CVE-2022-1586	An out-of-bounds read vulnerability was discovered in the PCRE2 library in the compile_xclass_matchingpath() fun
CVE-2022-1587	An out-of-bounds read vulnerability was discovered in the PCRE2 library in the get_recurse_data_length() function
CVE-2022-1619	Heap-based Buffer Overflow in function cmdline_erase_chars in GitHub repository vim/vim prior to 8.2.4899. This

CVE-2022-1621	Heap buffer overflow in vim_strncpy find_word in GitHub repository vim/vim prior to 8.2.4919. This vulnerability is
CVE-2022-1629	Buffer Over-read in function find_next_quote in GitHub repository vim/vim prior to 8.2.4925. This vulnerabilities ar
CVE-2022-1665	A set of pre-production kernel packages of Red Hat Enterprise Linux for IBM Power architecture can be booted by t
CVE-2022-1720	Buffer Over-read in function grab_file_name in GitHub repository vim/vim prior to 8.2.4956. This vulnerability is c
CVE-2022-1785	Out-of-bounds Write in GitHub repository vim/vim prior to 8.2.4977.
CVE-2022-1897	Out-of-bounds Write in GitHub repository vim/vim prior to 8.2.
CVE-2022-1927	Buffer Over-read in GitHub repository vim/vim prior to 8.2.
CVE-2022-1972	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: CVE-2022-2078. Reason: This candid
CVE-2022-20154	In lock_sock_nested of sock.c, there is a possible use after free due to a race condition. This could lead to local escal
CVE-2022-2047	In Eclipse Jetty versions 9.4.0 thru 9.4.46, and 10.0.0 thru 10.0.9, and 11.0.0 thru 11.0.9 versions, the parsing of the
CVE-2022-20566	In l2cap_chan_put of l2cap_core, there is a possible use after free due to improper locking. This could lead to local e
CVE-2022-2068	In addition to the c_rehash shell command injection identified in CVE-2022-1292, further circumstances where the c
CVE-2022-2097	AES OCB mode for 32-bit x86 platforms using the AES-NI assembly optimised implementation will not encrypt the
CVE-2022-2124	Buffer Over-read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2125	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 8.2.
CVE-2022-2126	Out-of-bounds Read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2129	Out-of-bounds Write in GitHub repository vim/vim prior to 8.2.
CVE-2022-2175	Buffer Over-read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2182	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 8.2.
CVE-2022-2183	Out-of-bounds Read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2206	Out-of-bounds Read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2207	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 8.2.
CVE-2022-2208	NULL Pointer Dereference in GitHub repository vim/vim prior to 8.2.5163.
CVE-2022-2210	Out-of-bounds Write in GitHub repository vim/vim prior to 8.2.
CVE-2022-2255	A vulnerability was found in mod_wsgi. The X-Client-IP header is not removed from a request from an untrusted pr
CVE-2022-22576	An improper authentication vulnerability exists in curl 7.33.0 to and including 7.82.0 which might allow reuse OAU
CVE-2022-22815	path_getbbox in path.c in Pillow before 9.0.0 improperly initializes ImagePath.Path.
CVE-2022-22816	path_getbbox in path.c in Pillow before 9.0.0 has a buffer over-read during initialization of ImagePath.Path.
CVE-2022-22817	PIL.ImageMath.eval in Pillow before 9.0.0 allows evaluation of arbitrary expressions, such as ones that use the Pyth
CVE-2022-2284	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.
CVE-2022-2286	Out-of-bounds Read in GitHub repository vim/vim prior to 9.0.
CVE-2022-2285	Integer Overflow or Wraparound in GitHub repository vim/vim prior to 9.0.
CVE-2022-2287	Out-of-bounds Read in GitHub repository vim/vim prior to 9.0.
CVE-2022-23218	The deprecated compatibility function svcunix_create in the sunrpc module of the GNU C Library (aka glibc) throug
CVE-2022-23219	The deprecated compatibility function clnt_create in the sunrpc module of the GNU C Library (aka glibc) through 2.
CVE-2022-2343	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0044.
CVE-2022-2344	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0045.
CVE-2022-23990	Expat (aka libexpat) before 2.4.4 has an integer overflow in the doProlog function.
CVE-2022-2345	Use After Free in GitHub repository vim/vim prior to 9.0.0046.
CVE-2022-24303	Pillow before 9.0.1 allows attackers to delete files because spaces in temporary pathnames are mishandled.

CVE-2022-24407	In Cyrus SASL 2.1.17 through 2.1.27 before 2.1.28, plugins/sql.c does not escape the password for a SQL INSERT c
CVE-2022-24329	In JetBrains Kotlin before 1.6.0, it was not possible to lock dependencies for Multiplatform Gradle Projects.
CVE-2022-24758	The Jupyter notebook is a web-based notebook environment for interactive computing. Prior to version 6.4.9, unauth
CVE-2022-24795	yajl-ruby is a C binding to the YAJL JSON parsing and generation library. The 1.x branch and the 2.x branch of `yajl
CVE-2022-24769	Moby is an open-source project created by Docker to enable and accelerate software containerization. A bug was fou
CVE-2022-24823	Netty is an open-source, asynchronous event-driven network application framework. The package `io.netty:netty-coo
CVE-2022-2509	A vulnerability found in gnutils. This security flaw happens because of a double free error occurs during verification
CVE-2022-2522	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0061.
CVE-2022-25836	Bluetooth® Low Energy Pairing in Bluetooth Core Specification v4.0 through v5.3 may permit an unauthenticated M
CVE-2022-2526	A use-after-free vulnerability was found in systemd. This issue occurs due to the on_stream_io() function and dns_st
CVE-2022-2625	A vulnerability was found in PostgreSQL. This attack requires permission to create non-temporary objects in at least
CVE-2022-27664	In net/http in Go before 1.18.6 and 1.19.x before 1.19.1, attackers can cause a denial of service because an HTTP/2 c
CVE-2022-27774	An insufficiently protected credentials vulnerability exists in curl 4.9 to and include curl 7.82.0 are affected that coul
CVE-2022-27781	libcurl provides the `CURLOPT_CERTINFO` option to allow applications to request details to be returned about a se
CVE-2022-27776	A insufficiently protected credentials vulnerability in fixed in curl 7.83.0 might leak authentication or cookie header
CVE-2022-27943	libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle_const, as demonstrated by nm-r
CVE-2022-27782	libcurl would reuse a previously created connection even when a TLS or SSH-related option had been changed that sh
CVE-2022-2819	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0211.
CVE-2022-2795	By flooding the target resolver with queries exploiting this flaw an attacker can significantly impair the resolver's pe
CVE-2022-2845	Improper Validation of Specified Quantity in Input in GitHub repository vim/vim prior to 9.0.0218.
CVE-2022-2879	Reader.Read does not set a limit on the maximum size of file headers. A maliciously crafted archive could cause Rea
CVE-2022-2849	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0220.
CVE-2022-2905	An out-of-bounds memory read flaw was found in the Linux kernel's BPF subsystem in how a user calls the bpf_tail
CVE-2022-2880	Requests forwarded by ReverseProxy include the raw query parameters from the inbound request, including unparsa
CVE-2022-29155	In OpenLDAP 2.x before 2.5.12 and 2.6.x before 2.6.2, a SQL injection vulnerability exists in the experimental back
CVE-2022-29238	Jupyter Notebook is a web-based notebook environment for interactive computing. Prior to version 6.4.12, authentic
CVE-2022-2923	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.0240.
CVE-2022-29458	ncurses 6.3 before patch 20220416 has an out-of-bounds read and segmentation violation in convert_strings in tinfor
CVE-2022-2946	Use After Free in GitHub repository vim/vim prior to 9.0.0246.
CVE-2022-2961	A use-after-free flaw was found in the Linux kernel's PLP Rose functionality in the way a user triggers a race condit
CVE-2022-2963	A vulnerability found in jasper. This security vulnerability happens because of a memory leak bug in function cmdop
CVE-2022-2980	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.0259.
CVE-2022-3037	Use After Free in GitHub repository vim/vim prior to 9.0.0322.
CVE-2022-29824	In libxml2 before 2.9.14, several buffer handling functions in buf.c (xmlBuf*) and tree.c (xmlBuffer*) don't check fo
CVE-2022-3094	Sending a flood of dynamic DNS updates may cause `named` to allocate large amounts of memory. This, in turn, ma
CVE-2022-31015	Waitress is a Web Server Gateway Interface server for Python 2 and 3. Waitress versions 2.1.0 and 2.1.1 may termin
CVE-2022-31023	Play Framework is a web framework for Java and Scala. Versions prior to 2.8.16 are vulnerable to generation of error
CVE-2022-3114	An issue was discovered in the Linux kernel through 5.16-rc6. imx_register_uart_clocks in drivers/clock/imx/clock.c lack
CVE-2022-31160	jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of jQuery. Versions
CVE-2022-3153	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.0404.

CVE-2022-3171	A parsing issue with binary data in protobuf-java core and lite versions prior to 3.21.7, 3.20.3, 3.19.6 and 3.16.3 can
CVE-2022-32149	An attacker may cause a denial of service by crafting an Accept-Language header which ParseAcceptLanguage will
CVE-2022-32190	JoinPath and URL.JoinPath do not remove ../ path elements appended to a relative path. For example, JoinPath("http
CVE-2022-3219	GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of sign
CVE-2022-32206	curl < 7.84.0 supports "chained" HTTP compression algorithms, meaning that a serverresponse can be compressed m
CVE-2022-32208	When curl < 7.84.0 does FTP transfers secured by krb5, it handles message verification failures wrongly. This flaw r
CVE-2022-32221	When doing HTTP(S) transfers, libcurl might erroneously use the read callback (CURLOPT_READFUNCTION)
CVE-2022-3234	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0483.
CVE-2022-3235	Use After Free in GitHub repository vim/vim prior to 9.0.0490.
CVE-2022-3238	A double-free flaw was found in the Linux kernel's NTFS3 subsystem in how a user triggers remount and umount si
CVE-2022-3256	Use After Free in GitHub repository vim/vim prior to 9.0.0530.
CVE-2022-3296	Stack-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0577.
CVE-2022-33064	An off-by-one error in function wav_read_header in src/wav.c in Libsndfile 1.1.0, results in a write out of bound, wh
CVE-2022-33070	Protobuf-c v1.4.0 was discovered to contain an invalid arithmetic shift via the function parse_tag_and_wiretype in p
CVE-2022-3344	A flaw was found in the KVM's AMD nested virtualization (SVM). A malicious L1 guest could purposely fail to int
CVE-2022-3352	Use After Free in GitHub repository vim/vim prior to 9.0.0614.
CVE-2022-3424	A use-after-free flaw was found in the Linux kernel's SGI GRU driver in the way the first gru_file_unlocked_ioctl fu
CVE-2022-34903	GnuPG through 2.3.6, in unusual situations where an attacker possesses any secret-key information from a victim's k
CVE-2022-3509	A parsing issue similar to CVE-2022-3171, but with textformat in protobuf-java core and lite versions prior to 3.21.7
CVE-2022-3510	A parsing issue similar to CVE-2022-3171, but with Message-Type Extensions in protobuf-java core and lite version
CVE-2022-3523	A vulnerability was found in Linux Kernel. It has been classified as problematic. Affected is an unknown function o
CVE-2022-35252	When curl is used to retrieve and parse cookies from a HTTP(S) server, itaccepts cookies using control codes that wh
CVE-2022-3534	A vulnerability classified as critical has been found in Linux Kernel. Affected is the function btf_dump_name_dups
CVE-2022-3542	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was wit
CVE-2022-3554	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was wit
CVE-2022-3555	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was wit
CVE-2022-3570	Multiple heap buffer overflows in tiffcrop.c utility in libtiff library Version 4.4.0 allows attacker to trigger unsafe or
CVE-2022-35737	SQLite 1.0.12 through 3.39.x before 3.39.2 sometimes allows an array-bounds overflow if billions of bytes are used
CVE-2022-3598	LibTIFF 4.4.0 has an out-of-bounds write in extractContigSamplesShifted24bits in tools/tiffcrop.c:3604, allowing at
CVE-2022-3599	LibTIFF 4.4.0 has an out-of-bounds read in writeSingleSection in tools/tiffcrop.c:7345, allowing attackers to cause a
CVE-2022-3606	A vulnerability was found in Linux Kernel. It has been classified as problematic. This affects the function find_prog
CVE-2022-36280	An out-of-bounds(OOB) memory access vulnerability was found in vmwgfx driver in drivers/gpu/vmxgfx/vmxgfx_k
CVE-2022-3705	A vulnerability was found in vim and classified as problematic. Affected by this issue is the function qf_update_buff
CVE-2022-3715	A flaw was found in the bash package, where a heap-buffer overflow can occur in valid parameter_transform. This is
CVE-2022-36227	In libarchive before 3.6.2, the software does not check for an error after calling calloc function that can return with a
CVE-2022-37434	zlib through 1.2.12 has a heap-based buffer over-read or buffer overflow in inflate in inflate.c via a large gzip header
CVE-2022-38177	By spoofing the target resolver with responses that have a malformed ECDSA signature, an attacker can trigger a sm
CVE-2022-3821	An off-by-one Error issue was discovered in Systemd in format_timespan() function of time-util.c. An attacker coul
CVE-2022-38178	By spoofing the target resolver with responses that have a malformed EdDSA signature, an attacker can trigger a sm
CVE-2022-38457	A use-after-free(UAF) vulnerability was found in function 'vmw_cmd_res_check' in drivers/gpu/vmxgfx/vmxgfx_ex

CVE-2022-3857	Rejected reason: Maintainer contacted. This is a false-positive. The flaw does not actually exist and was erroneously
CVE-2022-38533	In GNU Binutils before 2.40, there is a heap-buffer-overflow in the error function bfd_getl32 when called from the s
CVE-2022-3903	An incorrect read request flaw was found in the Infrared Transceiver USB driver in the Linux kernel. This issue occur
CVE-2022-39327	Azure CLI is the command-line interface for Microsoft Azure. In versions previous to 2.40.0, Azure CLI contains a
CVE-2022-39379	Fluentd collects events from various data sources and writes them to files, RDBMS, NoSQL, IaaS, SaaS, Hadoop an
CVE-2022-3965	A vulnerability classified as problematic was found in ffmpeg. This vulnerability affects the function smc_encode_st
CVE-2022-3964	A vulnerability classified as problematic has been found in ffmpeg. This affects an unknown part of the file libavcod
CVE-2022-40090	An issue was discovered in function TIFFReadDirectory libtiff before 4.4.0 allows attackers to cause a denial of serv
CVE-2022-40133	A use-after-free(UAF) vulnerability was found in function 'vmw_execbuf_tie_context' in drivers/gpu/vmxgfx/vmxgfx
CVE-2022-40303	An issue was discovered in libxml2 before 2.10.3. When parsing a multi-gigabyte XML document with the XML_Pa
CVE-2022-40304	An issue was discovered in libxml2 before 2.10.3. Certain invalid XML entity definitions can corrupt a hash table ke
CVE-2022-40735	The Diffie-Hellman Key Agreement Protocol allows use of long exponents that arguably make certain calculations u
CVE-2022-40755	JasPer 3.0.6 allows denial of service via a reachable assertion in the function inttobits in libjasper/base/jas_image.c.
CVE-2022-40897	Python Packaging Authority (PyPA) setuptools before 65.5.1 allows remote attackers to cause a denial of service via
CVE-2022-40898	An issue discovered in Python Packaging Authority (PyPA) Wheel 0.37.1 and earlier allows remote attackers to caus
CVE-2022-41404	An issue in the fetch() method in the BasicProfile class of org.ini4j through version v0.5.4 allows attackers to cause
CVE-2022-41409	Integer overflow vulnerability in pcre2test before 10.41 allows attackers to cause a denial of service or other unspeci
CVE-2022-41715	Programs which compile regular expressions from untrusted sources may be vulnerable to memory exhaustion or de
CVE-2022-41716	Due to unsanitized NUL values, attackers may be able to maliciously set environment variables on Windows. In sys
CVE-2022-41717	An attacker can cause excessive memory growth in a Go server accepting HTTP/2 requests. HTTP/2 server connecti
CVE-2022-41720	On Windows, restricted files can be accessed via os.DirFS and http.Dir. The os.DirFS function and http.Dir type pro
CVE-2022-41721	A request smuggling attack is possible when using MaxBytesHandler. When using MaxBytesHandler, the body of an
CVE-2022-41723	A maliciously crafted HTTP/2 stream could cause excessive CPU consumption in the HPACK decoder, sufficient to
CVE-2022-41722	A path traversal vulnerability exists in filepath.Clean on Windows. On Windows, the filepath.Clean function could t
CVE-2022-41724	Large handshake records may cause panics in crypto/tls. Both clients and servers may send large TLS handshake rec
CVE-2022-41725	A denial of service is possible from excessive resource consumption in net/http and mime/multipart. Multipart form
CVE-2022-41848	drivers/char/pcmcia/synclink_cs.c in the Linux kernel through 5.19.12 has a race condition and resultant use-after-fre
CVE-2022-41850	roccat_report_event in drivers/hid/hid-roccat.c in the Linux kernel through 5.19.12 has a race condition and resultant
CVE-2022-41862	In PostgreSQL, a modified, unauthenticated server can send an unterminated string during the establishment of Kerb
CVE-2022-42003	In FasterXML jackson-databind before versions 2.13.4.1 and 2.12.17.1, resource exhaustion can occur because of a l
CVE-2022-42004	In FasterXML jackson-databind before 2.13.4, resource exhaustion can occur because of a lack of a check in BeanDe
CVE-2022-42011	An issue was discovered in D-Bus before 1.12.24, 1.13.x and 1.14.x before 1.14.4, and 1.15.x before 1.15.2. An auth
CVE-2022-42010	An issue was discovered in D-Bus before 1.12.24, 1.13.x and 1.14.x before 1.14.4, and 1.15.x before 1.15.2. An auth
CVE-2022-42898	PAC parsing in MIT Kerberos 5 (aka krb5) before 1.19.4 and 1.20.x before 1.20.1 has integer overflows that may lea
CVE-2022-42012	An issue was discovered in D-Bus before 1.12.24, 1.13.x and 1.14.x before 1.14.4, and 1.15.x before 1.15.2. An auth
CVE-2022-4292	Use After Free in GitHub repository vim/vim prior to 9.0.0882.
CVE-2022-4293	Floating Point Comparison with Incorrect Operator in GitHub repository vim/vim prior to 9.0.0804.
CVE-2022-4304	A timing based side channel exists in the OpenSSL RSA Decryption implementation which could be sufficient to rec
CVE-2022-43552	A use after free vulnerability exists in curl <7.87.0. Curl can be asked to *tunnel* virtually all protocols it supports t
CVE-2022-44032	An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/cm4000_cs.c has a race condition an

CVE-2022-44033	An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/cm4040_cs.c has a race condition and
CVE-2022-44034	An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/scr24x_cs.c has a race condition and
CVE-2022-4415	A vulnerability was found in systemd. This security flaw can cause a local information leak due to systemd-coredump
CVE-2022-4450	The function PEM_read_bio_ex() reads a PEM file from a BIO and parses and decodes the "name" (e.g. "CERTIFIC
CVE-2022-44840	Heap buffer overflow vulnerability in binutils readelf before 2.40 via function find_section_in_set in file readelf.c.
CVE-2022-45061	An issue was discovered in Python before 3.11.1. An unnecessary quadratic algorithm exists in one path when proces
CVE-2022-45198	Pillow before 9.2.0 performs Improper Handling of Highly Compressed GIF Data (Data Amplification).
CVE-2022-45199	Pillow before 9.3.0 allows denial of service via SAMPLESPERPIXEL.
CVE-2022-4543	A flaw named "EntryBleed" was found in the Linux Kernel Page Table Isolation (KPTI). This issue could allow a lo
CVE-2022-45885	An issue was discovered in the Linux kernel through 6.0.9. drivers/media/dvb-core/dvb_frontend.c has a race conditi
CVE-2022-45888	An issue was discovered in the Linux kernel through 6.0.9. drivers/char/xillybus/xillyusb.c has a race condition and
CVE-2022-45939	GNU Emacs through 28.2 allows attackers to execute commands via shell metacharacters in the name of a source-co
CVE-2022-46337	A cleverly devised username might bypass LDAP authentication checks. In LDAP-authenticated Derby installations
CVE-2022-46908	SQLite through 3.40.0, when relying on --safe for execution of an untrusted CLI script, does not properly implement
CVE-2022-47007	An issue was discovered function stab_demangle_v3_arg in stabs.c in Binutils 2.34 thru 2.38, allows attackers to cau
CVE-2022-47008	An issue was discovered function make_tmpdir, and make_tmpname in bucomm.c in Binutils 2.34 thru 2.38, allow
CVE-2022-47010	An issue was discovered function pr_function_type in prdbg.c in Binutils 2.34 thru 2.38, allows attackers to cause a
CVE-2022-47011	An issue was discovered function parse_stab_struct_fields in stabs.c in Binutils 2.34 thru 2.38, allows attackers to ca
CVE-2022-48064	GNU Binutils before 2.40 was discovered to contain an excessive memory consumption vulnerability via the function
CVE-2022-48303	GNU Tar through 1.34 has a one-byte out-of-bounds read that results in use of uninitialized memory for a conditiona
CVE-2022-48337	GNU Emacs through 28.2 allows attackers to execute commands via shell metacharacters in the name of a source-co
CVE-2022-48338	An issue was discovered in GNU Emacs through 28.2. In ruby-mode.el, the ruby-find-library-file function has a loca
CVE-2022-48339	An issue was discovered in GNU Emacs through 28.2. htmlfontify.el has a command injection vulnerability. In the h
CVE-2022-48560	A use-after-free exists in Python through 3.9 via heappushpop in heapq.
CVE-2022-48564	read_ints in plistlib.py in Python through 3.9.1 is vulnerable to a potential DoS attack via CPU and RAM exhaustion
CVE-2022-48565	An XML External Entity (XXE) issue was discovered in Python through 3.9.1. The plistlib module no longer accept
CVE-2022-48619	An issue was discovered in drivers/input/input.c in the Linux kernel before 5.17.10. An attacker can cause a denial o
CVE-2022-48628	In the Linux kernel, the following vulnerability has been resolved: ceph: drop messages from MDS when unmounting
CVE-2022-48631	In the Linux kernel, the following vulnerability has been resolved: ext4: fix bug in extents parsing when eh_entries =
CVE-2022-48632	In the Linux kernel, the following vulnerability has been resolved: i2c: mlxbf: prevent stack overflow in mlxbf_i2c_
CVE-2022-48633	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: Fix WARN_ON(lock->magic != lo
CVE-2022-48634	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: Fix BUG: sleeping function called
CVE-2022-48635	In the Linux kernel, the following vulnerability has been resolved: fsdax: Fix infinite loop in dax_iomap_rw() I got a
CVE-2022-48637	In the Linux kernel, the following vulnerability has been resolved: bnxt: prevent skb UAF after handing over to PTP
CVE-2022-48638	In the Linux kernel, the following vulnerability has been resolved: cgroup: cgroup_get_from_id() must check the loc
CVE-2022-48639	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix possible refcount leak in tc_new_t
CVE-2022-48643	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fix nft_counters_enabled unde
CVE-2022-48641	In the Linux kernel, the following vulnerability has been resolved: netfilter: ebtables: fix memory leak when blob is
CVE-2022-48640	In the Linux kernel, the following vulnerability has been resolved: bonding: fix NULL deref in bond_rr_gen_slave_i
CVE-2022-48646	In the Linux kernel, the following vulnerability has been resolved: sfc/siena: fix null pointer dereference in efx_har

CVE-2022-48647	In the Linux kernel, the following vulnerability has been resolved: sfc: fix TX channel offset when using legacy inter
CVE-2022-48651	In the Linux kernel, the following vulnerability has been resolved: ipvlan: Fix out-of-bound bugs caused by unset sk
CVE-2022-48648	In the Linux kernel, the following vulnerability has been resolved: sfc: fix null pointer dereference in efx_hard_start
CVE-2022-48652	In the Linux kernel, the following vulnerability has been resolved: ice: Fix crash by keep old cfg when update TCs m
CVE-2022-48653	In the Linux kernel, the following vulnerability has been resolved: ice: Don't double unplug aux on peer initiated res
CVE-2022-48658	In the Linux kernel, the following vulnerability has been resolved: mm: slub: fix flush_cpu_slab()/__free_slab() inv
CVE-2022-48659	In the Linux kernel, the following vulnerability has been resolved: mm/slub: fix to return errno if kcalloc() fails in c
CVE-2022-48662	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gem: Really move i915_gem_context.li
CVE-2022-48666	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix a use-after-free There are two .exit
CVE-2022-48667	In the Linux kernel, the following vulnerability has been resolved: smb3: fix temporary data corruption in insert rang
CVE-2022-48668	In the Linux kernel, the following vulnerability has been resolved: smb3: fix temporary data corruption in collapse r
CVE-2022-48673	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix possible access to freed memory in l
CVE-2022-48672	In the Linux kernel, the following vulnerability has been resolved: of: fdt: fix off-by-one error in unflatten_dt_nodes
CVE-2022-48686	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix UAF when detecting digest errors V
CVE-2022-48687	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: fix out-of-bounds read when setting HM
CVE-2022-48690	In the Linux kernel, the following vulnerability has been resolved: ice: Fix DMA mappings leak Fix leak, when user
CVE-2022-48691	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: clean up hook list when offloa
CVE-2022-48694	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix drain SQ hang with no comple
CVE-2022-48696	In the Linux kernel, the following vulnerability has been resolved: regmap: spi: Reserve space for register address/pa
CVE-2022-48697	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix a use-after-free Fix the following use-a
CVE-2022-48701	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix an out-of-bounds bug in __
CVE-2022-48703	In the Linux kernel, the following vulnerability has been resolved: thermal/int340x_thermal: handle data_vault when
CVE-2022-48702	In the Linux kernel, the following vulnerability has been resolved: ALSA: emu10k1: Fix out of bounds access in snd
CVE-2022-48705	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921e: fix crash in chip reset fail In
CVE-2022-48704	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: add a force flush to delay work when
CVE-2022-48706	In the Linux kernel, the following vulnerability has been resolved: vdpa: ifcvf: Do proper cleanup if IFCVF init fails
CVE-2022-48709	In the Linux kernel, the following vulnerability has been resolved: ice: switch: fix potential memleak in ice_add_adv
CVE-2022-48710	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix a possible null pointer dereferenc
CVE-2022-48711	In the Linux kernel, the following vulnerability has been resolved: tipc: improve size validations for received domain
CVE-2022-48713	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/pt: Fix crash with stop filters in sin
CVE-2022-48714	In the Linux kernel, the following vulnerability has been resolved: bpf: Use VM_MAP instead of VM_ALLOC for r
CVE-2022-48715	In the Linux kernel, the following vulnerability has been resolved: scsi: bnx2fc: Make bnx2fc_recv_frame() mp safe
CVE-2022-48721	In the Linux kernel, the following vulnerability has been resolved: net/smc: Forward wakeup to smc socket waitque
CVE-2022-48724	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix potential memory leak in intel_s
CVE-2022-48725	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Fix refcounting leak in siw_create_q
CVE-2022-48728	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix AIP early init panic An early failure
CVE-2022-48729	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix panic with larger ipoib send_queue_s
CVE-2022-48730	In the Linux kernel, the following vulnerability has been resolved: dma-buf: heaps: Fix potential spectre v1 gadget It
CVE-2022-48731	In the Linux kernel, the following vulnerability has been resolved: mm/kmemleak: avoid scanning potential huge ho
CVE-2022-48732	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: fix off by one in BIOS boundary ch

CVE-2022-48735	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: Fix UAF of leds class devs at unbind
CVE-2022-48736	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-48737	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-48743	In the Linux kernel, the following vulnerability has been resolved: net: amd-xgbe: Fix skb data length underflow The
CVE-2022-48744	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Avoid field-overflowing memcpy() In
CVE-2022-48745	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Use del_timer_sync in fw reset flow of
CVE-2022-48747	In the Linux kernel, the following vulnerability has been resolved: block: Fix wrong offset in bio_truncate() bio_trun
CVE-2022-48746	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix handling of wrong devices during
CVE-2022-48748	In the Linux kernel, the following vulnerability has been resolved: net: bridge: vlan: fix memory leak in __allowed_i
CVE-2022-48751	In the Linux kernel, the following vulnerability has been resolved: net/smc: Transitional solution for clcsock race iss
CVE-2022-48752	In the Linux kernel, the following vulnerability has been resolved: powerpc/perf: Fix power_pmu_disable to call clea
CVE-2022-48755	In the Linux kernel, the following vulnerability has been resolved: powerpc64/bpf: Limit 'ldbrx' to processors compl
CVE-2022-48754	In the Linux kernel, the following vulnerability has been resolved: phylib: fix potential use-after-free Commit bafbd
CVE-2022-48758	In the Linux kernel, the following vulnerability has been resolved: scsi: bnx2fc: Flush destroy_work queue before ca
CVE-2022-48757	In the Linux kernel, the following vulnerability has been resolved: net: fix information leakage in /proc/net/ptype In
CVE-2022-48760	In the Linux kernel, the following vulnerability has been resolved: USB: core: Fix hang in usb_kill_urb by adding m
CVE-2022-48763	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Forcibly leave nested virt when SMM
CVE-2022-48761	In the Linux kernel, the following vulnerability has been resolved: usb: xhci-plat: fix crash when suspend if remote v
CVE-2022-48764	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Free kvm_cpuid_entry2 array on post
CVE-2022-48766	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Wrap dcn301_calculate_wm_ar
CVE-2022-48767	In the Linux kernel, the following vulnerability has been resolved: ceph: properly put ceph_string reference after asy
CVE-2022-48769	In the Linux kernel, the following vulnerability has been resolved: efi: runtime: avoid EFIv2 runtime services on Ap
CVE-2022-48771	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Fix stale file descriptors on failed u
CVE-2022-48770	In the Linux kernel, the following vulnerability has been resolved: bpf: Guard against accessing NULL pt_regs in bp
CVE-2022-48772	In the Linux kernel, the following vulnerability has been resolved: media: lgdt3306a: Add a check against null-point
CVE-2022-48773	In the Linux kernel, the following vulnerability has been resolved: xprtrdma: fix pointer derefs in error cases of rprc
CVE-2022-48778	In the Linux kernel, the following vulnerability has been resolved: mtd: rawnand: gpmi: don't leak PM reference in e
CVE-2022-48780	In the Linux kernel, the following vulnerability has been resolved: net/smc: Avoid overwriting the copies of clcsock
CVE-2022-48783	In the Linux kernel, the following vulnerability has been resolved: net: dsa: lantiq_gswip: fix use after free in gswip_
CVE-2022-48784	In the Linux kernel, the following vulnerability has been resolved: cfg80211: fix race in netlink owner interface dest
CVE-2022-48786	In the Linux kernel, the following vulnerability has been resolved: vsock: remove vsock from connected table when
CVE-2022-48787	In the Linux kernel, the following vulnerability has been resolved: iwlfwifi: fix use-after-free If no firmware was pres
CVE-2022-48788	In the Linux kernel, the following vulnerability has been resolved: nvme-rdma: fix possible use-after-free in transpor
CVE-2022-48789	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix possible use-after-free in transport
CVE-2022-48790	In the Linux kernel, the following vulnerability has been resolved: nvme: fix a possible use-after-free in controller re
CVE-2022-48793	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: nSVM: fix potential NULL dereferenc
CVE-2022-48796	In the Linux kernel, the following vulnerability has been resolved: iommu: Fix potential use-after-free during probe
CVE-2022-48799	In the Linux kernel, the following vulnerability has been resolved: perf: Fix list corruption in perf_cgroup_switch()
CVE-2022-48800	In the Linux kernel, the following vulnerability has been resolved: mm: vmscan: remove deadlock due to throttling f
CVE-2022-48801	In the Linux kernel, the following vulnerability has been resolved: iio: buffer: Fix file related error handling in IIO_I

CVE-2022-48802	In the Linux kernel, the following vulnerability has been resolved: fs/proc: task_mmu.c: don't read mapcount for mig
CVE-2022-48804	In the Linux kernel, the following vulnerability has been resolved: vt_ioctl: fix array_index_nospec in vt_setactivate
CVE-2022-48806	In the Linux kernel, the following vulnerability has been resolved: eeprom: ee1004: limit i2c reads to I2C_SMBUS_
CVE-2022-48805	In the Linux kernel, the following vulnerability has been resolved: net: usb: ax88179_178a: Fix out-of-bounds access
CVE-2022-48809	In the Linux kernel, the following vulnerability has been resolved: net: fix a memleak when uncloning an skb dst and
CVE-2022-48807	In the Linux kernel, the following vulnerability has been resolved: ice: Fix KASAN error in LAG NETDEV_UNRE
CVE-2022-48810	In the Linux kernel, the following vulnerability has been resolved: ipmr,ip6mr: acquire RTNL before calling ip[6]mr
CVE-2022-48811	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: don't release napi in __ibmvnic_open()
CVE-2022-48816	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: lock against ->sock changing during sy
CVE-2022-48823	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Fix refcount issue when LOGO is rece
CVE-2022-48825	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Add stag_work to all the vports Call tr
CVE-2022-48827	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix the behavior of READ near OFFSET_
CVE-2022-48828	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix ia_size underflow iattr::ia_size is a lo
CVE-2022-48829	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix NFSv3 SETATTR/CREATE's handlin
CVE-2022-48834	In the Linux kernel, the following vulnerability has been resolved: usb: usbtmc: Fix bug in pipe direction for control
CVE-2022-48835	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Page fault in reply q processing A p
CVE-2022-48836	In the Linux kernel, the following vulnerability has been resolved: Input: aiptek - properly check endpoint type Syzb
CVE-2022-48837	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: rndis: prevent integer overflow in m
CVE-2022-48839	In the Linux kernel, the following vulnerability has been resolved: net/packet: fix slab-out-of-bounds access in packe
CVE-2022-48841	In the Linux kernel, the following vulnerability has been resolved: ice: fix NULL pointer dereference in ice_update_
CVE-2022-48842	In the Linux kernel, the following vulnerability has been resolved: ice: Fix race condition during interface enslave C
CVE-2022-48843	In the Linux kernel, the following vulnerability has been resolved: drm/vrr: Set VRR capable prop only if it is attach
CVE-2022-48844	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Fix leaking sent_cmd skb se
CVE-2022-48846	In the Linux kernel, the following vulnerability has been resolved: block: release rq qos structures for queue without
CVE-2022-48848	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Do not unregister events twice N
CVE-2022-48849	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: bypass tiling flag check in virtual d
CVE-2022-48850	In the Linux kernel, the following vulnerability has been resolved: net-sysfs: add check for netdevice being present t
CVE-2022-48853	In the Linux kernel, the following vulnerability has been resolved: swiotlb: fix info leak with DMA_FROM_DEVIC
CVE-2022-48861	In the Linux kernel, the following vulnerability has been resolved: vdpa: fix use-after-free on vp_vdpa_remove Whe
CVE-2022-48858	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix a race on command flush flow Fix a
CVE-2022-48862	In the Linux kernel, the following vulnerability has been resolved: vhost: fix hung thread due to erroneous iotlb entri
CVE-2022-48863	In the Linux kernel, the following vulnerability has been resolved: mISDN: Fix memory leak in dsp_pipeline_build()
CVE-2022-48864	In the Linux kernel, the following vulnerability has been resolved: vdpa/mlx5: add validation for VIRTIO_NET_CT
CVE-2022-48865	In the Linux kernel, the following vulnerability has been resolved: tipc: fix kernel panic when enabling bearer When
CVE-2022-48866	In the Linux kernel, the following vulnerability has been resolved: HID: hid-thrustmaster: fix OOB read in thrustmas
CVE-2022-48868	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Let probe fail when workqueue
CVE-2022-48875	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: sdata can be NULL during AMP
CVE-2022-48876	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix initialization of rx->link and
CVE-2022-48887	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Remove rcu locks from user resour
CVE-2022-48889	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: sof-nau8825: fix module alias overfl

CVE-2022-48890	In the Linux kernel, the following vulnerability has been resolved: scsi: storvsc: Fix swiotlb bounce buffer leak in co
CVE-2022-48896	In the Linux kernel, the following vulnerability has been resolved: ixgbe: fix pci device refcount leak As the comm
CVE-2022-48899	In the Linux kernel, the following vulnerability has been resolved: drm/virtio: Fix GEM handle creation UAF Usersp
CVE-2022-48900	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-48904	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix I/O page table memory leak The
CVE-2022-48905	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: free reset-work-item when flushing Fix
CVE-2022-48909	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix connection leak There's a potential le
CVE-2022-48906	In the Linux kernel, the following vulnerability has been resolved: mptcp: Correctly set DATA_FIN timeout when n
CVE-2022-48911	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_queue: fix possible use-after-free Eric
CVE-2022-48910	In the Linux kernel, the following vulnerability has been resolved: net: ipv6: ensure we call ipv6_mc_down() at mos
CVE-2022-48913	In the Linux kernel, the following vulnerability has been resolved: blktrace: fix use after free for struct blk_trace Wh
CVE-2022-48914	In the Linux kernel, the following vulnerability has been resolved: xen/netfront: destroy queues before real_num_tx_
CVE-2022-48915	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Fix TZ_GET_TRIP NULL pointer
CVE-2022-48916	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix double list_add when enabling V
CVE-2022-48917	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-48919	In the Linux kernel, the following vulnerability has been resolved: cifs: fix double free race when mount fails in cifs,
CVE-2022-48929	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix crash due to out of bounds access into reg
CVE-2022-48931	In the Linux kernel, the following vulnerability has been resolved: configs: fix a race in configs_{,un}register_sub
CVE-2022-48936	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-48933	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fix memory leak during statel
CVE-2022-48938	In the Linux kernel, the following vulnerability has been resolved: CDC-NCM: avoid overflow in sanity checking A
CVE-2022-48939	In the Linux kernel, the following vulnerability has been resolved: bpf: Add schedule points in batch ops syzbot repo
CVE-2022-48943	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: make apf token non-zero to fix
CVE-2022-48953	In the Linux kernel, the following vulnerability has been resolved: rtc: cmos: Fix event handler registration ordering
CVE-2022-48951	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Check bounds for second channel in s
CVE-2022-48946	In the Linux kernel, the following vulnerability has been resolved: udf: Fix preallocation discarding at indirect exten
CVE-2022-48961	In the Linux kernel, the following vulnerability has been resolved: net: mdio: fix unbalanced fwnode reference count
CVE-2022-48950	In the Linux kernel, the following vulnerability has been resolved: perf: Fix perf_pending_task() UaF Per syzbot it is
CVE-2022-48969	In the Linux kernel, the following vulnerability has been resolved: xen-netfront: Fix NULL string after live migration
CVE-2022-48971	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix not cleanup led when bt_init fails l
CVE-2022-48976	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable_offload: fix using __this_cpu_
CVE-2022-48975	In the Linux kernel, the following vulnerability has been resolved: gpiolib: fix memory leak in gpiochip_setup_dev()
CVE-2022-48972	In the Linux kernel, the following vulnerability has been resolved: mac802154: fix missing INIT_LIST_HEAD in ie
CVE-2022-48977	In the Linux kernel, the following vulnerability has been resolved: can: af_can: fix NULL pointer dereference in can
CVE-2022-48978	In the Linux kernel, the following vulnerability has been resolved: HID: core: fix shift-out-of-bounds in hid_report_
CVE-2022-48979	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix array index out of bound err
CVE-2022-48982	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix crash when replugging CSR fake c
CVE-2022-48985	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix race on per-CQ variable napi work
CVE-2022-48988	In the Linux kernel, the following vulnerability has been resolved: memcg: fix possible use-after-free in memcg_wri
CVE-2022-48986	In the Linux kernel, the following vulnerability has been resolved: mm/gup: fix gup_pud_range() for dax For dax pu

CVE-2022-48987	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-dv-timings.c: fix too strict blanking s
CVE-2022-48990	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix use-after-free during gpu recover
CVE-2022-4899	A vulnerability was found in zstd v1.4.10, where an attacker can supply empty string as an argument to the command
CVE-2022-48994	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: Fix function prototype mismatch in s
CVE-2022-48998	In the Linux kernel, the following vulnerability has been resolved: powerpc/bpf/32: Fix Oops on tail call tests test_b
CVE-2022-48991	In the Linux kernel, the following vulnerability has been resolved: mm/khugepaged: invoke MMU notifiers in shmer
CVE-2022-49005	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Fix bounds check for _sx controls For
CVE-2022-49003	In the Linux kernel, the following vulnerability has been resolved: nvme: fix SRCU protection of nvme_ns_head list
CVE-2022-49011	In the Linux kernel, the following vulnerability has been resolved: hwmon: (coretemp) fix pci device refcount leak in
CVE-2022-49013	In the Linux kernel, the following vulnerability has been resolved: sctp: fix memory leak in sctp_stream_outq_migra
CVE-2022-49006	In the Linux kernel, the following vulnerability has been resolved: tracing: Free buffers when a used dynamic event t
CVE-2022-49014	In the Linux kernel, the following vulnerability has been resolved: net: tun: Fix use-after-free in tun_detach() syzbot
CVE-2022-49017	In the Linux kernel, the following vulnerability has been resolved: tipc: re-fetch skb cb after tipc_msg_validate As th
CVE-2022-49021	In the Linux kernel, the following vulnerability has been resolved: net: phy: fix null-ptr-deref while probe() failed I g
CVE-2022-49022	In the Linux kernel, the following vulnerability has been resolved: wifi: mac8021: fix possible oob access in ieee802
CVE-2022-49027	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix error handling in iavf_init_module() The
CVE-2022-49028	In the Linux kernel, the following vulnerability has been resolved: ixgbev: Fix resource leak in ixgbev_init_module
CVE-2022-49029	In the Linux kernel, the following vulnerability has been resolved: hwmon: (ibmpex) Fix possible UAF when ibmpe
CVE-2022-49034	In the Linux kernel, the following vulnerability has been resolved: sh: cpuinfo: Fix a warning for CONFIG_CPUMA
CVE-2022-49043	xmlXIncludeAddNode in xinclude.c in libxml2 before 2.11.0 has a use-after-free.
CVE-2022-49044	In the Linux kernel, the following vulnerability has been resolved: dm integrity: fix memory corruption when tag_siz
CVE-2022-49045	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49048	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix panic when forwarding a pkt with no in6
CVE-2022-49052	In the Linux kernel, the following vulnerability has been resolved: mm: fix unexpected zeroed page mapping with zr
CVE-2022-49053	In the Linux kernel, the following vulnerability has been resolved: scsi: target: tcmu: Fix possible page UAF tcmu_tr
CVE-2022-49057	In the Linux kernel, the following vulnerability has been resolved: block: null_blk: end timed out poll request When
CVE-2022-49058	In the Linux kernel, the following vulnerability has been resolved: cifs: potential buffer overflow in handling symlin
CVE-2022-49060	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix NULL pointer dereference in smc_p
CVE-2022-49063	In the Linux kernel, the following vulnerability has been resolved: ice: arfs: fix use-after-free when freeing @rx_cpu
CVE-2022-49065	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix the svc_deferred_event trace class
CVE-2022-49066	In the Linux kernel, the following vulnerability has been resolved: veth: Ensure eth header is in skb's linear part Afte
CVE-2022-49069	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix by adding FPU protection f
CVE-2022-4907	Uninitialized Use in FFmpeg in Google Chrome prior to 108.0.5359.71 allowed a remote attacker to execute arbitrar
CVE-2022-49070	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix unregistering of framebuffers without
CVE-2022-49072	In the Linux kernel, the following vulnerability has been resolved: gpio: Restrict usage of GPIO chip irq members be
CVE-2022-49074	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Fix GICR_CTLR.RWP polling It
CVE-2022-49076	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Fix use-after-free bug for mm struct
CVE-2022-49077	In the Linux kernel, the following vulnerability has been resolved: mmmremap.c: avoid pointless invalidate_range_s
CVE-2022-49078	In the Linux kernel, the following vulnerability has been resolved: lz4: fix LZ4_decompress_safe_partial read out of
CVE-2022-49080	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: fix mpol_new leak in shared_po

CVE-2022-49081	In the Linux kernel, the following vulnerability has been resolved: highmem: fix checks in __kmap_local_sched_{in
CVE-2022-49084	In the Linux kernel, the following vulnerability has been resolved: qede: confirm skb is allocated before using qede_
CVE-2022-49082	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix use after free in _scsih_expand
CVE-2022-49086	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix leak of nested actions While
CVE-2022-49089	In the Linux kernel, the following vulnerability has been resolved: IB/rdmavt: add lock to call to rvt_error_qp to prev
CVE-2022-49094	In the Linux kernel, the following vulnerability has been resolved: net/tls: fix slab-out-of-bounds bug in decrypt_inte
CVE-2022-49096	In the Linux kernel, the following vulnerability has been resolved: net: sfc: add missing xdp queue reinitialization A
CVE-2022-49097	In the Linux kernel, the following vulnerability has been resolved: NFS: Avoid writeback threads getting stuck in m
CVE-2022-49101	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49100	In the Linux kernel, the following vulnerability has been resolved: virtio_console: eliminate anonymous module_ini
CVE-2022-49103	In the Linux kernel, the following vulnerability has been resolved: NFSv4.2: fix reference count leaks in _nfs42_pro
CVE-2022-49108	In the Linux kernel, the following vulnerability has been resolved: clk: mediatek: Fix memory leaks on probe Handle
CVE-2022-49107	In the Linux kernel, the following vulnerability has been resolved: ceph: fix memory leak in ceph_readaddr when note
CVE-2022-49109	In the Linux kernel, the following vulnerability has been resolved: ceph: fix inode reference leakage in ceph_get_sn
CVE-2022-49110	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: revisit gc autotuning as of co
CVE-2022-49111	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix use after free in hci_send_acl This
CVE-2022-49113	In the Linux kernel, the following vulnerability has been resolved: powerpc/secvar: fix refcount leak in format_show
CVE-2022-49116	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: use memset avoid memory leaks Use m
CVE-2022-49118	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Free irq vectors in order for v3 HW
CVE-2022-49123	In the Linux kernel, the following vulnerability has been resolved: ath11k: Fix frames flush failure caused by deadlo
CVE-2022-49124	In the Linux kernel, the following vulnerability has been resolved: x86/mce: Work around an erratum on fast string c
CVE-2022-49125	In the Linux kernel, the following vulnerability has been resolved: drm/sprid: fix potential NULL dereference 'drm' c
CVE-2022-49126	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix memory leaks Fix memory leak
CVE-2022-49127	In the Linux kernel, the following vulnerability has been resolved: ref_tracker: implement use-after-free detection W
CVE-2022-49129	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921: fix crash when startup fails. If the
CVE-2022-49130	In the Linux kernel, the following vulnerability has been resolved: ath11k: mhi: use mhi_sync_power_up() If amss.b
CVE-2022-49132	In the Linux kernel, the following vulnerability has been resolved: ath11k: pci: fix crash on suspend if board file is n
CVE-2022-49133	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: svm range restore work deadlock w
CVE-2022-49134	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum: Guard against invalid local por
CVE-2022-49138	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: Ignore multiple conn comp
CVE-2022-49136	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: Fix queuing commands whe
CVE-2022-49139	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix null ptr deref on hci_sync_conn_co
CVE-2022-49140	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49143	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49145	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Avoid out of bounds access when p
CVE-2022-49155	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Suppress a kernel complaint in qla_
CVE-2022-49154	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: fix panic on out-of-bounds guest IR
CVE-2022-49156	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix scheduling while atomic The d
CVE-2022-49157	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix premature hw access after PCI
CVE-2022-49159	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Implement ref count for SRB The t

CVE-2022-49158	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix warning message due to adisc b
CVE-2022-49160	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix crash during module load unloa
CVE-2022-49161	In the Linux kernel, the following vulnerability has been resolved: ASoC: mediatek: Fix error handling in mt8183_d
CVE-2022-49167	In the Linux kernel, the following vulnerability has been resolved: btrfs: do not double complete bio on errors during
CVE-2022-49171	In the Linux kernel, the following vulnerability has been resolved: ext4: don't BUG if someone dirty pages without a
CVE-2022-49172	In the Linux kernel, the following vulnerability has been resolved: parisc: Fix non-access data TLB cache flush fault
CVE-2022-49174	In the Linux kernel, the following vulnerability has been resolved: ext4: fix ext4_mb_mark_bb() with flex_bg with f
CVE-2022-49173	In the Linux kernel, the following vulnerability has been resolved: spi: fsi: Implement a timeout for polling status Th
CVE-2022-49175	In the Linux kernel, the following vulnerability has been resolved: PM: core: keep irq flags in device_pm_check_cal
CVE-2022-49176	In the Linux kernel, the following vulnerability has been resolved: bfq: fix use-after-free in bfq_dispatch_request KA
CVE-2022-49178	In the Linux kernel, the following vulnerability has been resolved: memstick/mspro_block: fix handling of read-only
CVE-2022-49177	In the Linux kernel, the following vulnerability has been resolved: hwrng: cavium - fix NULL but dereferenced cocc
CVE-2022-49179	In the Linux kernel, the following vulnerability has been resolved: block, bfq: don't move oom_bfqq Our test report t
CVE-2022-49180	In the Linux kernel, the following vulnerability has been resolved: LSM: general protection fault in legacy_parse_pa
CVE-2022-49183	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_ct: fix ref leak when switching zon
CVE-2022-49193	In the Linux kernel, the following vulnerability has been resolved: ice: fix 'scheduling while atomic' on aux critical e
CVE-2022-49190	In the Linux kernel, the following vulnerability has been resolved: kernel/resource: fix kfree() of bootmem memory
CVE-2022-49196	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Fix use after free in remove_phb
CVE-2022-49197	In the Linux kernel, the following vulnerability has been resolved: af_netlink: Fix shift out of bounds in group mask
CVE-2022-49198	In the Linux kernel, the following vulnerability has been resolved: mptcp: Fix crash due to tcp_tsorted_anchor was in
CVE-2022-49199	In the Linux kernel, the following vulnerability has been resolved: RDMA/nldev: Prevent underflow in nldev_stat_s
CVE-2022-49201	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: fix race between xmit and reset There is
CVE-2022-49203	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix double free during GPU res
CVE-2022-49204	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix more uncharged while msg has
CVE-2022-49205	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix double uncharge the mem of sh
CVE-2022-49206	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix memory leak in error flow for s
CVE-2022-49207	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix memleak in sk_psock_queue_r
CVE-2022-49208	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Prevent some integer underflows f
CVE-2022-49209	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix memleak in tcp_bpf_sendmsg
CVE-2022-49220	In the Linux kernel, the following vulnerability has been resolved: dax: make sure inodes are flushed before destroy
CVE-2022-49215	In the Linux kernel, the following vulnerability has been resolved: xsk: Fix race at socket teardown Fix a race in the
CVE-2022-49218	In the Linux kernel, the following vulnerability has been resolved: drm/dp: Fix OOB read when handling Post Cursor
CVE-2022-49225	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921s: fix a possible memory leak in mt
CVE-2022-49226	In the Linux kernel, the following vulnerability has been resolved: net: asix: add proper error handling of usb read er
CVE-2022-49227	In the Linux kernel, the following vulnerability has been resolved: igc: avoid kernel warning when changing RX ring
CVE-2022-49234	In the Linux kernel, the following vulnerability has been resolved: net: dsa: Avoid cross-chip syncing of VLAN filter
CVE-2022-49231	In the Linux kernel, the following vulnerability has been resolved: rtw88: fix memory overrun and memory leak dur
CVE-2022-49235	In the Linux kernel, the following vulnerability has been resolved: ath9k_htc: fix uninit value bugs Syzbot reported 2
CVE-2022-49236	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix UAF due to race between btf_try_get_me
CVE-2022-49237	In the Linux kernel, the following vulnerability has been resolved: ath11k: add missing of_node_put() to avoid leak

CVE-2022-49245	In the Linux kernel, the following vulnerability has been resolved: ASoC: rockchip: Fix PM usage reference of rockchip
CVE-2022-49248	In the Linux kernel, the following vulnerability has been resolved: ALSA: firewire-lib: fix uninitialized flag for AV/
CVE-2022-49259	In the Linux kernel, the following vulnerability has been resolved: block: don't delete queue kobject before its children
CVE-2022-49263	In the Linux kernel, the following vulnerability has been resolved: brcmfmac: pcie: Release firmwares in the brcmf_
CVE-2022-49264	In the Linux kernel, the following vulnerability has been resolved: exec: Force single empty string when argv is empty
CVE-2022-49265	In the Linux kernel, the following vulnerability has been resolved: PM: domains: Fix sleep-in-atomic bug caused by
CVE-2022-49266	In the Linux kernel, the following vulnerability has been resolved: block: fix rq-qos breakage from skipping rq_qos_
CVE-2022-49267	In the Linux kernel, the following vulnerability has been resolved: mmc: core: use sysfs_emit() instead of sprintf() sp
CVE-2022-49268	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: Fix NULL ptr dereference wh
CVE-2022-49271	In the Linux kernel, the following vulnerability has been resolved: cifs: prevent bad output lengths in smb2_ioctl_qu
CVE-2022-49272	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix potential AB/BA lock with buffe
CVE-2022-49280	In the Linux kernel, the following vulnerability has been resolved: NFSD: prevent underflow in nfssvc_decode_writ
CVE-2022-49281	In the Linux kernel, the following vulnerability has been resolved: cifs: fix handlecache and multiuser In multiuser e
CVE-2022-49286	In the Linux kernel, the following vulnerability has been resolved: tpm: use try_get_ops() in tpm-space.c As part of t
CVE-2022-49287	In the Linux kernel, the following vulnerability has been resolved: tpm: fix reference counting for struct tpm_chip T
CVE-2022-49288	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix races among concurrent preallo
CVE-2022-49290	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix potential double free on mesh join
CVE-2022-49291	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix races among concurrent hw_par
CVE-2022-49293	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: initialize registers in nft_do_c
CVE-2022-49294	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check if modulo is 0 before div
CVE-2022-49295	In the Linux kernel, the following vulnerability has been resolved: nbd: call genl_unregister_family() first in nbd_cl
CVE-2022-49296	In the Linux kernel, the following vulnerability has been resolved: ceph: fix possible deadlock when holding Fwb to
CVE-2022-49297	In the Linux kernel, the following vulnerability has been resolved: nbd: fix io hung while disconnecting device In ou
CVE-2022-49300	In the Linux kernel, the following vulnerability has been resolved: nbd: fix race between nbd_alloc_config() and mo
CVE-2022-49303	In the Linux kernel, the following vulnerability has been resolved: drivers: staging: rtl8192eu: Fix deadlock in rtw_j
CVE-2022-49307	In the Linux kernel, the following vulnerability has been resolved: tty: synclink_gt: Fix null-pointer-dereference in s
CVE-2022-49306	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: host: Stop setting the ACPI companio
CVE-2022-49308	In the Linux kernel, the following vulnerability has been resolved: extcon: Modify extcon device to be created after
CVE-2022-49314	In the Linux kernel, the following vulnerability has been resolved: tty: Fix a possible resource leak in icom_probe W
CVE-2022-49316	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Don't hold the layoutget locks across mul
CVE-2022-49317	In the Linux kernel, the following vulnerability has been resolved: f2fs: avoid infinite loop to flush node pages xfste
CVE-2022-49321	In the Linux kernel, the following vulnerability has been resolved: xprtrdma: treat all calls not a bcall when bc_serv
CVE-2022-49322	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix sleeping function called from invalid
CVE-2022-49323	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: fix possible null-ptr-deref in a
CVE-2022-49325	In the Linux kernel, the following vulnerability has been resolved: tcp: add accessors to read/set tp->snd_cwnd We h
CVE-2022-49330	In the Linux kernel, the following vulnerability has been resolved: tcp: fix tcp_mtup_probe_success vs wrong snd_c
CVE-2022-49333	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: E-Switch, pair only capable devices OF
CVE-2022-49339	In the Linux kernel, the following vulnerability has been resolved: net: ipv6: unexport __init-annotated seg6_hmac_
CVE-2022-49341	In the Linux kernel, the following vulnerability has been resolved: bpf, arm64: Clear prog->jited_len along prog->jit
CVE-2022-49342	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: bgmac: Fix refcount leak in bcma_r

CVE-2022-49343	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid cycles in directory h-tree A malicious
CVE-2022-49344	In the Linux kernel, the following vulnerability has been resolved: af_unix: Fix a data-race in unix_dgram_peer_wa
CVE-2022-49348	In the Linux kernel, the following vulnerability has been resolved: ext4: filter out EXT4_FC_REPLAY from on-disk
CVE-2022-49345	In the Linux kernel, the following vulnerability has been resolved: net: xfrm: unexport __init-annotated xfrm4_prot
CVE-2022-49349	In the Linux kernel, the following vulnerability has been resolved: ext4: fix use-after-free in ext4_rename_dir_prepa
CVE-2022-49350	In the Linux kernel, the following vulnerability has been resolved: net: mdio: unexport __init-annotated mdio_bus_in
CVE-2022-49352	In the Linux kernel, the following vulnerability has been resolved: ext4: fix warning in ext4_handle_inode_extensio
CVE-2022-49355	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49353	In the Linux kernel, the following vulnerability has been resolved: powerpc/papr_scm: don't requests stats with '0' si
CVE-2022-49356	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Trap RDMA segment overflows Preve
CVE-2022-49357	In the Linux kernel, the following vulnerability has been resolved: efi: Do not import certificates from UEFI Secure
CVE-2022-49358	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: memleak flow rule from com
CVE-2022-49359	In the Linux kernel, the following vulnerability has been resolved: drm/panfrost: Job should reference MMU not file
CVE-2022-49370	In the Linux kernel, the following vulnerability has been resolved: firmware: dmi-sysfs: Fix memory leak in dmi_sys
CVE-2022-49371	In the Linux kernel, the following vulnerability has been resolved: driver core: fix deadlock in __device_attach In __
CVE-2022-49372	In the Linux kernel, the following vulnerability has been resolved: tcp: tcp_rtx_synack() can be called from process
CVE-2022-49374	In the Linux kernel, the following vulnerability has been resolved: tipc: check attribute length for bearer name syzbo
CVE-2022-49377	In the Linux kernel, the following vulnerability has been resolved: blk-mq: don't touch ->tagset in blk_mq_get_sq_h
CVE-2022-49378	In the Linux kernel, the following vulnerability has been resolved: sfc: fix considering that all channels have TX que
CVE-2022-49384	In the Linux kernel, the following vulnerability has been resolved: md: fix double free of io_acct_set bioset Now io_
CVE-2022-49385	In the Linux kernel, the following vulnerability has been resolved: driver: base: fix UAF when driver_attach failed V
CVE-2022-49388	In the Linux kernel, the following vulnerability has been resolved: ubi: ubi_create_volume: Fix use-after-free when v
CVE-2022-49390	In the Linux kernel, the following vulnerability has been resolved: macsec: fix UAF bug for real_dev Create a new m
CVE-2022-49393	In the Linux kernel, the following vulnerability has been resolved: misc: fastrpc: fix list iterator in fastrpc_req_mem
CVE-2022-49394	In the Linux kernel, the following vulnerability has been resolved: blk-iolatency: Fix inflight count imbalances and I
CVE-2022-49395	In the Linux kernel, the following vulnerability has been resolved: um: Fix out-of-bounds read in LDT setup syscall
CVE-2022-49400	In the Linux kernel, the following vulnerability has been resolved: md: Don't set mddev private to NULL in raid0 pe
CVE-2022-49401	In the Linux kernel, the following vulnerability has been resolved: mm/page_owner: use strscpy() instead of strncpy()
CVE-2022-49402	In the Linux kernel, the following vulnerability has been resolved: ftrace: Clean up hash direct_functions on register
CVE-2022-49404	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Fix potential integer multiplication o
CVE-2022-49407	In the Linux kernel, the following vulnerability has been resolved: dlm: fix plock invalid read This patch fixes an inv
CVE-2022-49409	In the Linux kernel, the following vulnerability has been resolved: ext4: fix bug_on in __es_tree_search Hulk Robot
CVE-2022-49410	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix potential double free in create_var_re
CVE-2022-49412	In the Linux kernel, the following vulnerability has been resolved: bfq: Avoid merging queues with different parents
CVE-2022-49411	In the Linux kernel, the following vulnerability has been resolved: bfq: Make sure bfqg for which we are queueing re
CVE-2022-49413	In the Linux kernel, the following vulnerability has been resolved: bfq: Update cgroup information before merging b
CVE-2022-49414	In the Linux kernel, the following vulnerability has been resolved: ext4: fix race condition between ext4_write and e
CVE-2022-49418	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Fix free of uninitialized nfs4_label on ref
CVE-2022-49420	In the Linux kernel, the following vulnerability has been resolved: net: annotate races around sk->sk_bound_dev_if
CVE-2022-49416	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix use-after-free in chanctx cod

CVE-2022-49422	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix the error handling path in ic
CVE-2022-49429	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Prevent panic when SDMA is disab
CVE-2022-49434	In the Linux kernel, the following vulnerability has been resolved: PCI: Avoid pci_dev_lock() AB/BA deadlock with
CVE-2022-49432	In the Linux kernel, the following vulnerability has been resolved: powerpc/xics: fix refcount leak in icp_opal_init()
CVE-2022-49433	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Prevent use of lock before it is initia
CVE-2022-49436	In the Linux kernel, the following vulnerability has been resolved: powerpc/papr_scm: Fix leaking nvdimm_events_
CVE-2022-49437	In the Linux kernel, the following vulnerability has been resolved: powerpc/xive: Fix refcount leak in xive_spapr_in
CVE-2022-49440	In the Linux kernel, the following vulnerability has been resolved: powerpc/rtas: Keep MSR[RI] set when calling RT
CVE-2022-49442	In the Linux kernel, the following vulnerability has been resolved: drivers/base/node.c: fix compaction sysfs file leak
CVE-2022-49441	In the Linux kernel, the following vulnerability has been resolved: tty: fix deadlock caused by calling printk() under
CVE-2022-49443	In the Linux kernel, the following vulnerability has been resolved: list: fix a data-race around ep->rdllist ep_poll() fi
CVE-2022-49455	In the Linux kernel, the following vulnerability has been resolved: misc: ocxl: fix possible double free in ocxl_file_r
CVE-2022-49447	In the Linux kernel, the following vulnerability has been resolved: ARM: hisi: Add missing of_node_put after of_fin
CVE-2022-49456	In the Linux kernel, the following vulnerability has been resolved: bonding: fix missed rcu protection When removing
CVE-2022-49459	In the Linux kernel, the following vulnerability has been resolved: thermal/drivers/broadcom: Fix potential NULL de
CVE-2022-49465	In the Linux kernel, the following vulnerability has been resolved: blk-throttle: Set BIO_THROTTLED when bio has
CVE-2022-49468	In the Linux kernel, the following vulnerability has been resolved: thermal/core: Fix memory leak in __thermal_cool
CVE-2022-49471	In the Linux kernel, the following vulnerability has been resolved: rtw89: cfo: check mac_id to avoid out-of-bounds
CVE-2022-49469	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix anon_dev leak in create_subvol() When
CVE-2022-49472	In the Linux kernel, the following vulnerability has been resolved: net: phy: micrel: Allow probing without .driver_d
CVE-2022-49474	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix dangling sco_conn and use-after-fr
CVE-2022-49476	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921: fix kernel crash at mt7921_pci_re
CVE-2022-49479	In the Linux kernel, the following vulnerability has been resolved: mt76: fix tx status related use-after-free race on st
CVE-2022-49484	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7915: fix possible NULL pointer derefer
CVE-2022-49478	In the Linux kernel, the following vulnerability has been resolved: media: pvrusb2: fix array-index-out-of-bounds in
CVE-2022-49492	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: fix a NULL pointer dereference in nvme
CVE-2022-49493	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt5645: Fix erroneous cleanup order Ther
CVE-2022-49496	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: prevent kernel crash wh
CVE-2022-49498	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Check for null pointer of pointer sub
CVE-2022-49497	In the Linux kernel, the following vulnerability has been resolved: net: remove two BUG() from skb_checksum_help
CVE-2022-49501	In the Linux kernel, the following vulnerability has been resolved: usbnet: Run unregister_netdev() before unbind() a
CVE-2022-49503	In the Linux kernel, the following vulnerability has been resolved: ath9k_htc: fix potential out of bounds access with
CVE-2022-49504	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Inhibit aborts if external loopback plug
CVE-2022-49508	In the Linux kernel, the following vulnerability has been resolved: HID: elan: Fix potential double free in elan_input
CVE-2022-49511	In the Linux kernel, the following vulnerability has been resolved: fbdev: defio: fix the pagelist corruption Easily hit
CVE-2022-49516	In the Linux kernel, the following vulnerability has been resolved: ice: always check VF VSI pointer values The ice_
CVE-2022-49518	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc3-topology: Correct get_control_
CVE-2022-49519	In the Linux kernel, the following vulnerability has been resolved: ath10k: skip ath10k_halt during suspend for drive
CVE-2022-49521	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix resource leak in lpfc_sli4_send_seq
CVE-2022-49520	In the Linux kernel, the following vulnerability has been resolved: arm64: compat: Do not treat syscall number as ES

CVE-2022-49524	In the Linux kernel, the following vulnerability has been resolved: media: pci: cx23885: Fix the error handling in cx23885_remove()
CVE-2022-49526	In the Linux kernel, the following vulnerability has been resolved: md/bitmap: don't set sb values if can't pass sanity check
CVE-2022-49528	In the Linux kernel, the following vulnerability has been resolved: media: i2c: dw9714: Disable the regulator when the device is not powered
CVE-2022-49529	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm: fix the null pointer while the smu is not initialized
CVE-2022-49531	In the Linux kernel, the following vulnerability has been resolved: loop: implement ->free_disk Ensure that the loop device is not freed while it is in use
CVE-2022-49532	In the Linux kernel, the following vulnerability has been resolved: drm/virtio: fix NULL pointer dereference in virtio_gpu_ioctl
CVE-2022-49533	In the Linux kernel, the following vulnerability has been resolved: ath11k: Change max no of active probe SSID and BSSID
CVE-2022-49534	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Protect memory leak for NPIV ports session
CVE-2022-49535	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix null pointer dereference after failing to allocate
CVE-2022-49536	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix SCSI I/O completion and abort handling
CVE-2022-49538	In the Linux kernel, the following vulnerability has been resolved: ALSA: jack: Access input_dev under mutex It is not safe to access input_dev under mutex
CVE-2022-49539	In the Linux kernel, the following vulnerability has been resolved: rtw89: ser: fix CAM leaks occurring in L2 reset
CVE-2022-49542	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Move cfg_log_verbose check before calling
CVE-2022-49543	In the Linux kernel, the following vulnerability has been resolved: ath11k: fix the warning of dev_wake in mhi_pm_deinit
CVE-2022-49545	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Cancel pending work at closing
CVE-2022-49547	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix deadlock between concurrent dio writes
CVE-2022-49548	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix potential array overflow in bpf_trampoline_v2
CVE-2022-49549	In the Linux kernel, the following vulnerability has been resolved: x86/MCE/AMD: Fix memory leak when threshold is reached
CVE-2022-49552	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix combination of jit blinding and pointers to memory
CVE-2022-49554	In the Linux kernel, the following vulnerability has been resolved: zsmalloc: fix races between asynchronous zspage allocation and deallocation
CVE-2022-49556	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Use kzalloc for sev ioctl interfaces
CVE-2022-49557	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: KVM: Set the base guest FPU uABI size
CVE-2022-49558	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: double hook unregistration in nf_tables_remove_hook
CVE-2022-49559	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Drop WARNs that assert a triple fault
CVE-2022-49561	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: re-fetch conntrack after insert
CVE-2022-49562	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Use __try_cmpxchg_user() to update
CVE-2022-49563	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - add param check for RSA Reject request
CVE-2022-49564	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - add param check for DH Reject request
CVE-2022-49565	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/lbr: Fix unchecked MSR access error
CVE-2022-49566	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - fix memory leak in RSA When an RSA key is not found
CVE-2022-49567	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: fix uninit-value in mpol_rebind
CVE-2022-49571	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_max_reorder
CVE-2022-49572	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_slow_start
CVE-2022-49573	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_early_retrans
CVE-2022-49574	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_recovery. W
CVE-2022-49575	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_thin_linear
CVE-2022-49577	In the Linux kernel, the following vulnerability has been resolved: udp: Fix a data-race around sysctl_udp_l3mdev_a
CVE-2022-49576	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix data-races around sysctl_fib_multipath
CVE-2022-49578	In the Linux kernel, the following vulnerability has been resolved: ip: Fix data-races around sysctl_ip_prot_sock. sys
CVE-2022-49579	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix data-races around sysctl_fib_multipath

CVE-2022-49580	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix a data-race around sysctl_fib_multipath
CVE-2022-49581	In the Linux kernel, the following vulnerability has been resolved: be2net: Fix buffer overflow in be_get_module_e
CVE-2022-49583	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix handling of dummy receive descriptors
CVE-2022-49584	In the Linux kernel, the following vulnerability has been resolved: ixgbe: Add locking to prevent panic when setting
CVE-2022-49585	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_fastopen_b
CVE-2022-49586	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_fastopen. W
CVE-2022-49587	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_notsent_lo
CVE-2022-49589	In the Linux kernel, the following vulnerability has been resolved: igmp: Fix data-races around sysctl_igmp_qrv. W
CVE-2022-49590	In the Linux kernel, the following vulnerability has been resolved: igmp: Fix data-races around sysctl_igmp_llm_rep
CVE-2022-49592	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: fix dma queue left shift overflow iss
CVE-2022-49593	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_probe_inter
CVE-2022-49595	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_probe_thre
CVE-2022-49596	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_min_snd_m
CVE-2022-49598	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_mtu_probin
CVE-2022-49597	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_base_mss. V
CVE-2022-49599	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_l3mdev_acc
CVE-2022-49601	In the Linux kernel, the following vulnerability has been resolved: tcp/dccp: Fix a data-race around sysctl_tcp_fwma
CVE-2022-49602	In the Linux kernel, the following vulnerability has been resolved: ip: Fix a data-race around sysctl_fwmark_reflect.
CVE-2022-49603	In the Linux kernel, the following vulnerability has been resolved: ip: Fix data-races around sysctl_ip_fwd_update_p
CVE-2022-49604	In the Linux kernel, the following vulnerability has been resolved: ip: Fix data-races around sysctl_ip_fwd_use_pmt
CVE-2022-49605	In the Linux kernel, the following vulnerability has been resolved: igc: Reinstate IGC_REMOVED logic and implem
CVE-2022-49606	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix sleep from invalid context BU
CVE-2022-49610	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Prevent RSB underflow before vme
CVE-2022-49607	In the Linux kernel, the following vulnerability has been resolved: perf/core: Fix data race between perf_event_set_c
CVE-2022-49611	In the Linux kernel, the following vulnerability has been resolved: x86/speculation: Fill RSB on vmexit for IBRS Pr
CVE-2022-49612	In the Linux kernel, the following vulnerability has been resolved: power: supply: core: Fix boundary conditions in i
CVE-2022-49613	In the Linux kernel, the following vulnerability has been resolved: serial: 8250: Fix PM usage_count for console han
CVE-2022-49615	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt711-sdca: fix kernel NULL pointer deref
CVE-2022-49616	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt7*-sdw: harden jack_detect_handler Rea
CVE-2022-49617	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: sof_sdw: handle errors on card regis
CVE-2022-49620	In the Linux kernel, the following vulnerability has been resolved: net: tipc: fix possible refcount leak in tipc_sk_cre
CVE-2022-49622	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: avoid skb access on nf_stolen
CVE-2022-49623	In the Linux kernel, the following vulnerability has been resolved: powerpc/xive/spapr: correct bitmap allocation siz
CVE-2022-49624	In the Linux kernel, the following vulnerability has been resolved: net: atlantic: remove aq_nic_deinit() when resum
CVE-2022-49625	In the Linux kernel, the following vulnerability has been resolved: sfc: fix kernel panic when creating VF When crea
CVE-2022-49626	In the Linux kernel, the following vulnerability has been resolved: sfc: fix use after free when disabling sriov Use af
CVE-2022-49630	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_ecn_fallba
CVE-2022-49632	In the Linux kernel, the following vulnerability has been resolved: icmp: Fix a data-race around sysctl_icmp_errors_
CVE-2022-49635	In the Linux kernel, the following vulnerability has been resolved: drm/i915/selftests: fix subtraction overflow bug C
CVE-2022-49636	In the Linux kernel, the following vulnerability has been resolved: vlan: fix memory leak in vlan_newlink() Blamed

CVE-2022-49638	In the Linux kernel, the following vulnerability has been resolved: icmp: Fix data-races around sysctl. While reading
CVE-2022-49639	In the Linux kernel, the following vulnerability has been resolved: cipso: Fix data-races around sysctl. While reading
CVE-2022-49641	In the Linux kernel, the following vulnerability has been resolved: sysctl: Fix data races in proc_douintvec(). A sysctl
CVE-2022-49640	In the Linux kernel, the following vulnerability has been resolved: sysctl: Fix data races in proc_douintvec_minmax()
CVE-2022-49642	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: dwc-qos: Disable split header for Te
CVE-2022-49643	In the Linux kernel, the following vulnerability has been resolved: ima: Fix a potential integer overflow in ima_appri
CVE-2022-49644	In the Linux kernel, the following vulnerability has been resolved: drm/i915: fix a possible refcount leak in intel_dp_
CVE-2022-49646	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix queue selection for mesh/OC
CVE-2022-49647	In the Linux kernel, the following vulnerability has been resolved: cgroup: Use separate src/dst nodes when preloadi
CVE-2022-49651	In the Linux kernel, the following vulnerability has been resolved: srcu: Tighten cleanup_srcu_struct() GP checks C
CVE-2022-49653	In the Linux kernel, the following vulnerability has been resolved: i2c: piix4: Fix a memory leak in the EFCH MMIO
CVE-2022-49656	In the Linux kernel, the following vulnerability has been resolved: ARM: meson: Fix refcount leak in meson_smp_p
CVE-2022-49657	In the Linux kernel, the following vulnerability has been resolved: usbnet: fix memory leak in error case usbnet_writ
CVE-2022-49658	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix insufficient bounds propagation from adj
CVE-2022-49660	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49663	In the Linux kernel, the following vulnerability has been resolved: tunnels: do not assume mac header is set in skb_t
CVE-2022-49664	In the Linux kernel, the following vulnerability has been resolved: tipc: move bc link creation back to tipc_node_cre
CVE-2022-49667	In the Linux kernel, the following vulnerability has been resolved: net: bonding: fix use-after-free after 802.3ad slave
CVE-2022-49670	In the Linux kernel, the following vulnerability has been resolved: linux/dim: Fix divide by 0 in RDMA DIM Fix a c
CVE-2022-49671	In the Linux kernel, the following vulnerability has been resolved: RDMA/cm: Fix memory leak in ib_cm_insert_lis
CVE-2022-49672	In the Linux kernel, the following vulnerability has been resolved: net: tun: unlink NAPI from device on destruction
CVE-2022-49673	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix KASAN warning in raid5_add_disks
CVE-2022-49674	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix accesses beyond end of raid member
CVE-2022-49675	In the Linux kernel, the following vulnerability has been resolved: tick/nohz: unexport __init-annotated tick_nohz_fu
CVE-2022-49679	In the Linux kernel, the following vulnerability has been resolved: ARM: Fix refcount leak in axxia_boot_secondary
CVE-2022-49687	In the Linux kernel, the following vulnerability has been resolved: virtio_net: fix xdp_rxq_info bug after suspend/res
CVE-2022-49690	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49691	In the Linux kernel, the following vulnerability has been resolved: erspan: do not assume transport header is always
CVE-2022-49696	In the Linux kernel, the following vulnerability has been resolved: tipc: fix use-after-free Read in tipc_named_reinit
CVE-2022-49697	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix request_sock leak in sk lookup helpers A
CVE-2022-49695	In the Linux kernel, the following vulnerability has been resolved: igb: fix a use-after-free issue in igb_clean_tx_ring
CVE-2022-49698	In the Linux kernel, the following vulnerability has been resolved: netfilter: use get_random_u32 instead of prandom
CVE-2022-49699	In the Linux kernel, the following vulnerability has been resolved: filemap: Handle sibling entries in filemap_get_re
CVE-2022-49700	In the Linux kernel, the following vulnerability has been resolved: mm/slub: add missing TID updates on slab deacti
CVE-2022-49703	In the Linux kernel, the following vulnerability has been resolved: scsi: ibmvfc: Store vhost pointer during subcrq al
CVE-2022-49701	In the Linux kernel, the following vulnerability has been resolved: scsi: ibmvfc: Allocate/free queue resource only d
CVE-2022-49708	In the Linux kernel, the following vulnerability has been resolved: ext4: fix bug_on ext4_mb_use_inode_pa Hulk Ro
CVE-2022-49710	In the Linux kernel, the following vulnerability has been resolved: dm mirror log: round up region bitmap size to BI
CVE-2022-49715	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Fix refcount leak in gic_populate
CVE-2022-49716	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Fix error handling in gic_populate

CVE-2022-49720	In the Linux kernel, the following vulnerability has been resolved: block: Fix handling of offline queues in blk_mq_
CVE-2022-49722	In the Linux kernel, the following vulnerability has been resolved: ice: Fix memory corruption in VF driver Disable
CVE-2022-49725	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix call trace in setup_tx_descriptors After l
CVE-2022-49728	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix signed integer overflow in __ip6_appen
CVE-2022-49731	In the Linux kernel, the following vulnerability has been resolved: ata: libata-core: fix NULL pointer deref in ata_ho
CVE-2022-49730	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Resolve NULL ptr dereference after an
CVE-2022-49739	In the Linux kernel, the following vulnerability has been resolved: gfs2: Always check inode size of inline inodes Cl
CVE-2022-49740	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Check the count value of channe
CVE-2022-49742	In the Linux kernel, the following vulnerability has been resolved: f2fs: initialize locks earlier in f2fs_fill_super() sy
CVE-2022-49749	In the Linux kernel, the following vulnerability has been resolved: i2c: designware: use casting of u64 in clock multi
CVE-2022-49743	In the Linux kernel, the following vulnerability has been resolved: ovl: Use "buf" flexible array for memcpy() destin
CVE-2022-49748	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd: fix potential integer overflow on sh
CVE-2022-49750	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Add u64 casts to avoid overflowi
CVE-2022-49752	In the Linux kernel, the following vulnerability has been resolved: device property: fix of node refcount leak in fwno
CVE-2022-49753	In the Linux kernel, the following vulnerability has been resolved: dmaengine: Fix double increment of client_count
CVE-2022-49759	In the Linux kernel, the following vulnerability has been resolved: VMCI: Use threaded irqs instead of tasklets The v
CVE-2022-49766	In the Linux kernel, the following vulnerability has been resolved: netlink: Bounds-check struct nlmsgerr creation In
CVE-2022-49764	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent bpf program recursion for raw tracep
CVE-2022-49769	In the Linux kernel, the following vulnerability has been resolved: gfs2: Check sb_bsize_shift after reading superblo
CVE-2022-49770	In the Linux kernel, the following vulnerability has been resolved: ceph: avoid putting the realm twice when decodin
CVE-2022-49771	In the Linux kernel, the following vulnerability has been resolved: dm ioctl: fix misbehavior if list_versions races wi
CVE-2022-49773	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix optc2_configure warning o
CVE-2022-49772	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Drop snd_BUG_ON() from snd
CVE-2022-49779	In the Linux kernel, the following vulnerability has been resolved: kprobes: Skip clearing aggrprobe's post_handler i
CVE-2022-49776	In the Linux kernel, the following vulnerability has been resolved: macvlan: enforce a consistent minimal mtu macvl
CVE-2022-49780	In the Linux kernel, the following vulnerability has been resolved: scsi: target: tcm_loop: Fix possible name leak in t
CVE-2022-49781	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd: Fix crash due to race between amd
CVE-2022-49783	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Drop fpregs lock before inheriting FPU
CVE-2022-49782	In the Linux kernel, the following vulnerability has been resolved: perf: Improve missing SIGTRAP checking To cat
CVE-2022-49784	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd/uncore: Fix memory leak for events
CVE-2022-49785	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Add overflow check in sgx_validate_off
CVE-2022-49788	In the Linux kernel, the following vulnerability has been resolved: misc/vmw_vmci: fix an infoleak in vmci_host_de
CVE-2022-49789	In the Linux kernel, the following vulnerability has been resolved: scsi: zfcp: Fix double free of FSF request when q
CVE-2022-49802	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix null pointer dereference in ftrace_add
CVE-2022-49803	In the Linux kernel, the following vulnerability has been resolved: netdevsim: Fix memory leak of nsim_dev->fa_co
CVE-2022-49810	In the Linux kernel, the following vulnerability has been resolved: netfs: Fix missing xas_retry() calls in xarray itera
CVE-2022-49812	In the Linux kernel, the following vulnerability has been resolved: bridge: switchdev: Fix memory leaks when chang
CVE-2022-49813	In the Linux kernel, the following vulnerability has been resolved: net: ena: Fix error handling in ena_init() The ena
CVE-2022-49817	In the Linux kernel, the following vulnerability has been resolved: net: mhi: Fix memory leak in mhi_net_dellink() M
CVE-2022-49821	In the Linux kernel, the following vulnerability has been resolved: mISDN: fix possible memory leak in mISDN_dsp

CVE-2022-49822	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix connections leak when tlink setup failed
CVE-2022-49823	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix error handling in ata_tdev
CVE-2022-49825	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix error handling in ata_tpor
CVE-2022-49824	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix error handling in ata_tlin
CVE-2022-49826	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix double ata_host_put() in
CVE-2022-49828	In the Linux kernel, the following vulnerability has been resolved: hugetlbfs: don't delete error page from pagecache
CVE-2022-49829	In the Linux kernel, the following vulnerability has been resolved: drm/scheduler: fix fence ref counting We leaked
CVE-2022-49832	In the Linux kernel, the following vulnerability has been resolved: pinctrl: devicetree: fix null pointer dereferencing
CVE-2022-49833	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: clone zoned device info when cloning
CVE-2022-49835	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: fix potential memleak in 'add_widge
CVE-2022-49837	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix memory leaks in __check_func_call kme
CVE-2022-49839	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_transport_sas: Fix error handling in sas
CVE-2022-49838	In the Linux kernel, the following vulnerability has been resolved: sctp: clear out_curr if all frag chunks of current m
CVE-2022-49840	In the Linux kernel, the following vulnerability has been resolved: bpf, test_run: Fix alignment problem in bpf_prog
CVE-2022-49842	In the Linux kernel, the following vulnerability has been resolved: ASoC: core: Fix use-after-free in snd_soc_exit()
CVE-2022-49846	In the Linux kernel, the following vulnerability has been resolved: udf: Fix a slab-out-of-bounds write bug in udf_fin
CVE-2022-49858	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: Fix SQE threshold checking Current
CVE-2022-49853	In the Linux kernel, the following vulnerability has been resolved: net: macvlan: fix memory leaks of macvlan_comr
CVE-2022-49862	In the Linux kernel, the following vulnerability has been resolved: tipc: fix the msg->req tlv len check in tipc_nl_cor
CVE-2022-49864	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix NULL pointer dereference in sv
CVE-2022-49863	In the Linux kernel, the following vulnerability has been resolved: can: af_can: fix NULL pointer dereference in can
CVE-2022-49865	In the Linux kernel, the following vulnerability has been resolved: ipv6: addrlabel: fix infoleak when sending struct
CVE-2022-49869	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix possible crash in bnxt_hwrng_set_co
CVE-2022-49871	In the Linux kernel, the following vulnerability has been resolved: net: tun: Fix memory leaks of napi_get_frags kme
CVE-2022-49872	In the Linux kernel, the following vulnerability has been resolved: net: gso: fix panic on frag_list with mixed head al
CVE-2022-49870	In the Linux kernel, the following vulnerability has been resolved: capabilities: fix undefined behavior in bit shift fo
CVE-2022-49873	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix wrong reg type conversion in release_ref
CVE-2022-49874	In the Linux kernel, the following vulnerability has been resolved: HID: hyperv: fix possible memory leak in mouse
CVE-2022-49876	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix general-protection-fault in ie
CVE-2022-49879	In the Linux kernel, the following vulnerability has been resolved: ext4: fix BUG_ON() when directory entry has inv
CVE-2022-49878	In the Linux kernel, the following vulnerability has been resolved: bpf, verifier: Fix memory leak in array reallocation
CVE-2022-49880	In the Linux kernel, the following vulnerability has been resolved: ext4: fix warning in 'ext4_da_release_space' Syzk
CVE-2022-49881	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: fix memory leak in query_regdb_
CVE-2022-49884	In the Linux kernel, the following vulnerability has been resolved: KVM: Initialize gfn_to_pfn_cache locks in dedica
CVE-2022-49883	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: smm: number of GPRs in the SMRA
CVE-2022-49885	In the Linux kernel, the following vulnerability has been resolved: ACPI: APEI: Fix integer overflow in ghes_estatu
CVE-2022-49886	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Panic on bad configs that #VE on "privat
CVE-2022-49888	In the Linux kernel, the following vulnerability has been resolved: arm64: entry: avoid kprobe recursion The cortex_
CVE-2022-49889	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Check for NULL cpu_buffer in ring_
CVE-2022-49890	In the Linux kernel, the following vulnerability has been resolved: capabilities: fix potential memleak on error path f

CVE-2022-49892	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix use-after-free for dynamic ftrace_ops l
CVE-2022-49900	In the Linux kernel, the following vulnerability has been resolved: i2c: piix4: Fix adapter not be removed in piix4_re
CVE-2022-49901	In the Linux kernel, the following vulnerability has been resolved: blk-mq: Fix kmemleak in blk_mq_init_allocated
CVE-2022-49903	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix WARNING in ip6_route_net_exit_late()
CVE-2022-49905	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix possible leaked pernet namespace in
CVE-2022-49906	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Free rwi on reset success Free the rwi st
CVE-2022-49907	In the Linux kernel, the following vulnerability has been resolved: net: mdio: fix undefined behavior in bit shift for _
CVE-2022-49908	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix memory leak in vhci_wri
CVE-2022-49910	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix use-after-free caused by l
CVE-2022-49909	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: fix use-after-free in l2cap_co
CVE-2022-49915	In the Linux kernel, the following vulnerability has been resolved: mISDN: fix possible memory leak in mISDN_reg
CVE-2022-49917	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix WARNING in ip_vs_app_net_cleanup()
CVE-2022-49918	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix WARNING in __ip_vs_cleanup_batch()
CVE-2022-49921	In the Linux kernel, the following vulnerability has been resolved: net: sched: Fix use after free in red_enqueue() We
CVE-2022-49919	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: release flow rule object from
CVE-2022-49925	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix null-ptr-deref in ib_core_cleanu
CVE-2022-49927	In the Linux kernel, the following vulnerability has been resolved: nfs4: Fix kmemleak when allocate slot failed If o
CVE-2022-49928	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix null-ptr-deref when xps sysfs alloc
CVE-2022-49929	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix mr leak in RESPST_ERR_RNR
CVE-2022-49931	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Correctly move list in sc_disable() Com
CVE-2022-49932	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Do _all_ initialization before expos
CVE-2022-49933	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49934	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Fix UAF in ieee80211_scan_rx()
CVE-2022-49935	In the Linux kernel, the following vulnerability has been resolved: dma-buf/dma-resv: check if the new fence is reall
CVE-2022-49936	In the Linux kernel, the following vulnerability has been resolved: USB: core: Prevent nested device-reset calls Auto
CVE-2022-49937	In the Linux kernel, the following vulnerability has been resolved: media: mceusb: Use new usb_control_msg_*() ro
CVE-2022-49938	In the Linux kernel, the following vulnerability has been resolved: cifs: fix small mempool leak in SMB2_negotiate()
CVE-2022-49940	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: add sanity check for gsm->receive in
CVE-2022-49941	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-49942	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Don't finalize CSA in IBSS mod
CVE-2022-49943	In the Linux kernel, the following vulnerability has been resolved: USB: gadget: Fix obscure lockdep violation for u
CVE-2022-49944	In the Linux kernel, the following vulnerability has been resolved: Revert "usb: typec: ucsi: add a common function
CVE-2022-49948	In the Linux kernel, the following vulnerability has been resolved: vt: Clear selection before changing the font When
CVE-2022-49955	In the Linux kernel, the following vulnerability has been resolved: powerpc/rtas: Fix RTAS MSR[HV] handling for
CVE-2022-49949	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: Fix memory leak in firmware up
CVE-2022-49958	In the Linux kernel, the following vulnerability has been resolved: net/sched: fix netdevice reference leaks in attach_
CVE-2022-49959	In the Linux kernel, the following vulnerability has been resolved: openvswitch: fix memory leak at failed datapath c
CVE-2022-49961	In the Linux kernel, the following vulnerability has been resolved: bpf: Do mark_chain_precision for ARG_CONST
CVE-2022-49962	In the Linux kernel, the following vulnerability has been resolved: xhci: Fix null pointer dereference in remove if xH
CVE-2022-49964	In the Linux kernel, the following vulnerability has been resolved: arm64: cacheinfo: Fix incorrect assignment of sig

CVE-2022-49963	In the Linux kernel, the following vulnerability has been resolved: drm/i915/ttm: fix CCS handling Crucible + recen
CVE-2022-49965	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: add missing ->fini_xxxx interfaces
CVE-2022-49966	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: add missing ->fini_microcode inter
CVE-2022-49967	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix a data-race around bpf_jit_limit. While r
CVE-2022-49970	In the Linux kernel, the following vulnerability has been resolved: bpf, cgroup: Fix kernel BUG in purge_effective_j
CVE-2022-49971	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Fix a potential gpu_metrics_table m
CVE-2022-49972	In the Linux kernel, the following vulnerability has been resolved: xsk: Fix corrupted packets for XDP_SHARED_U
CVE-2022-49973	In the Linux kernel, the following vulnerability has been resolved: skmsg: Fix wrong last sg check in sk_msg_recvm
CVE-2022-49974	In the Linux kernel, the following vulnerability has been resolved: HID: nintendo: fix rumble worker null pointer de
CVE-2022-49975	In the Linux kernel, the following vulnerability has been resolved: bpf: Don't redirect packets with invalid pkt_len S
CVE-2022-49977	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix NULL pointer dereference in is_ftrace
CVE-2022-49979	In the Linux kernel, the following vulnerability has been resolved: net: fix refcount bug in sk_psock_get (2) Syzkalle
CVE-2022-49980	In the Linux kernel, the following vulnerability has been resolved: USB: gadget: Fix use-after-free Read in usb_udc_
CVE-2022-49981	In the Linux kernel, the following vulnerability has been resolved: HID: hidraw: fix memory leak in hidraw_release()
CVE-2022-49983	In the Linux kernel, the following vulnerability has been resolved: udmabuf: Set the DMA mask for the udmabuf dev
CVE-2022-49982	In the Linux kernel, the following vulnerability has been resolved: media: pvrusb2: fix memory leak in pvr_probe TH
CVE-2022-49985	In the Linux kernel, the following vulnerability has been resolved: bpf: Don't use tnum_range on array range checkin
CVE-2022-49986	In the Linux kernel, the following vulnerability has been resolved: scsi: storvsc: Remove WQ_MEM_RECLAIM fro
CVE-2022-49987	In the Linux kernel, the following vulnerability has been resolved: md: call __md_stop_writes in md_stop From the
CVE-2022-49989	In the Linux kernel, the following vulnerability has been resolved: xen/privcmd: fix error exit of privcmd_ioctl_dm_
CVE-2022-49990	In the Linux kernel, the following vulnerability has been resolved: s390: fix double free of GS and RI CBs on fork()
CVE-2022-49993	In the Linux kernel, the following vulnerability has been resolved: loop: Check for overflow while configuring loop
CVE-2022-49997	In the Linux kernel, the following vulnerability has been resolved: net: lantiq_xrx200: restore buffer if memory alloc
CVE-2022-50000	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable: fix stuck flows on cleanup du
CVE-2022-50001	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_tproxy: restrict to prerouting hook T
CVE-2022-50002	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: LAG, fix logic over MLX5_LAG_FLAG
CVE-2022-50003	In the Linux kernel, the following vulnerability has been resolved: ice: xsk: prohibit usage of non-balanced queue id
CVE-2022-50007	In the Linux kernel, the following vulnerability has been resolved: xfrm: fix refcount leak in __xfrm_policy_check()
CVE-2022-50008	In the Linux kernel, the following vulnerability has been resolved: kprobes: don't call disarm_kprobe() for disabled k
CVE-2022-50009	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix null-ptr-deref in f2fs_get_dnode_of_data
CVE-2022-50012	In the Linux kernel, the following vulnerability has been resolved: powerpc/64: Init jump labels before parse_early_
CVE-2022-50016	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: cnl: Do not process IPC reply
CVE-2022-50015	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda-ipc: Do not process IPC re
CVE-2022-50021	In the Linux kernel, the following vulnerability has been resolved: ext4: block range must be validated before use in
CVE-2022-50022	In the Linux kernel, the following vulnerability has been resolved: drivers/md:fix a potential use-after-free bug In lin
CVE-2022-50020	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid resizing to a partial cluster size This p
CVE-2022-50027	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix possible memory leak when failing
CVE-2022-50025	In the Linux kernel, the following vulnerability has been resolved: cxl: Fix a memory leak in an error handling path
CVE-2022-50030	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Prevent buffer overflow crashes in debu
CVE-2022-50031	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi: Fix HW conn removal use after free If

CVE-2022-50037	In the Linux kernel, the following vulnerability has been resolved: drm/i915/ttm: don't leak the ccs state The kernel o
CVE-2022-50039	In the Linux kernel, the following vulnerability has been resolved: stmmac: intel: Add a missing clk_disable_unprep
CVE-2022-50041	In the Linux kernel, the following vulnerability has been resolved: ice: Fix call trace with null VSI during VF reset D
CVE-2022-50042	In the Linux kernel, the following vulnerability has been resolved: net: genl: fix error path memory leak in policy du
CVE-2022-50044	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: start MHI channel after endpoit creation
CVE-2022-50045	In the Linux kernel, the following vulnerability has been resolved: powerpc/pci: Fix get_phb_number() locking The
CVE-2022-50048	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: possible module reference unc
CVE-2022-50046	In the Linux kernel, the following vulnerability has been resolved: net/sunrpc: fix potential memory leaks in rpc_sys
CVE-2022-50049	In the Linux kernel, the following vulnerability has been resolved: ASoC: DPCM: Don't pick up BE without substrea
CVE-2022-50050	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda: Fix potential buffer overf
CVE-2022-50051	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: debug: Fix potential buffer overflow
CVE-2022-50052	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: avs: Fix potential buffer overflow by
CVE-2022-50053	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix reset error handling Do not call iavf_clo
CVE-2022-50054	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix NULL pointer dereference in iavf_get_li
CVE-2022-50055	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix adminq error handling iavf_alloc_asq_b
CVE-2022-50059	In the Linux kernel, the following vulnerability has been resolved: ceph: don't leak snap_rwsem in handle_cap_grant
CVE-2022-50065	In the Linux kernel, the following vulnerability has been resolved: virtio_net: fix memory leak inside XPD_TX with
CVE-2022-50068	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: Fix dummy res NULL ptr deref bug Che
CVE-2022-50070	In the Linux kernel, the following vulnerability has been resolved: mptcp: do not queue data on closed subflows Dip
CVE-2022-50071	In the Linux kernel, the following vulnerability has been resolved: mptcp: move subflow cleanup in mptcp_destroy_
CVE-2022-50072	In the Linux kernel, the following vulnerability has been resolved: NFSv4/pnfs: Fix a use-after-free bug in open If so
CVE-2022-50073	In the Linux kernel, the following vulnerability has been resolved: net: tap: NULL pointer derefence in dev_parse_h
CVE-2022-50079	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check correct bounds for stre
CVE-2022-50080	In the Linux kernel, the following vulnerability has been resolved: tee: add overflow check in register_shm_helper()
CVE-2022-50082	In the Linux kernel, the following vulnerability has been resolved: ext4: fix warning in ext4_iomap_begin as race be
CVE-2022-50083	In the Linux kernel, the following vulnerability has been resolved: ext4: add EXT4_INODE_HAS_XATTR_SPACE
CVE-2022-50084	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix address sanitizer warning in raid_stat
CVE-2022-50085	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix address sanitizer warning in raid_res
CVE-2022-50086	In the Linux kernel, the following vulnerability has been resolved: block: don't allow the same type rq_qos add more
CVE-2022-50087	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scp: Ensure scpi_info is not assign
CVE-2022-50090	In the Linux kernel, the following vulnerability has been resolved: btrfs: replace BTRFS_MAX_EXTENT_SIZE with
CVE-2022-50088	In the Linux kernel, the following vulnerability has been resolved: mm/damon/reclaim: fix potential memory leak in
CVE-2022-50092	In the Linux kernel, the following vulnerability has been resolved: dm thin: fix use-after-free crash in dm_sm_regist
CVE-2022-50098	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix crash due to stale SRB access a
CVE-2022-50095	In the Linux kernel, the following vulnerability has been resolved: posix-cpu-timers: Cleanup CPU timers before fre
CVE-2022-50100	In the Linux kernel, the following vulnerability has been resolved: sched/core: Do not requeue task on CPU excluded
CVE-2022-50103	In the Linux kernel, the following vulnerability has been resolved: sched, cpuset: Fix dl_cpu_busy() panic due to em
CVE-2022-50104	In the Linux kernel, the following vulnerability has been resolved: powerpc/xive: Fix refcount leak in xive_get_max
CVE-2022-50110	In the Linux kernel, the following vulnerability has been resolved: watchdog: sp5100_tco: Fix a memory leak of EFC
CVE-2022-50111	In the Linux kernel, the following vulnerability has been resolved: ASoC: mt6359: Fix refcount leak bug In mt6359_

CVE-2022-50113	In the Linux kernel, the following vulnerability has been resolved: ASoc: audio-graph-card2: Fix refcount leak bug in
CVE-2022-50116	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: fix deadlock and link starvation in ou
CVE-2022-50118	In the Linux kernel, the following vulnerability has been resolved: powerpc/perf: Optimize clearing the pending PM
CVE-2022-50126	In the Linux kernel, the following vulnerability has been resolved: jbd2: fix assertion 'jh->b_frozen_data == NULL'
CVE-2022-50127	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix error unwind in rxe_create_qp(Q
CVE-2022-50129	In the Linux kernel, the following vulnerability has been resolved: RDMA/srpt: Fix a use-after-free Change the LIO
CVE-2022-50130	In the Linux kernel, the following vulnerability has been resolved: staging: fbft: core: set smem_len before fb_defer
CVE-2022-50133	In the Linux kernel, the following vulnerability has been resolved: usb: xhci_plat_remove: avoid NULL dereference
CVE-2022-50135	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix BUG: KASAN: null-ptr-deref in
CVE-2022-50134	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: fix potential memory leak in setup_
CVE-2022-50137	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix a window for use-after-free D
CVE-2022-50138	In the Linux kernel, the following vulnerability has been resolved: RDMA/qedr: Fix potential memory leak in __qed
CVE-2022-50144	In the Linux kernel, the following vulnerability has been resolved: soundwire: revisit driver bind/unbind and callbac
CVE-2022-50148	In the Linux kernel, the following vulnerability has been resolved: kernfs: fix potential NULL dereference in __kern
CVE-2022-50149	In the Linux kernel, the following vulnerability has been resolved: driver core: fix potential deadlock in __driver_att
CVE-2022-50150	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2022-50153	In the Linux kernel, the following vulnerability has been resolved: usb: host: Fix refcount leak in ehci_hcd_ppc_of_p
CVE-2022-50159	In the Linux kernel, the following vulnerability has been resolved: of: check previous kernel's ima-kexec-buffer agai
CVE-2022-50163	In the Linux kernel, the following vulnerability has been resolved: ax25: fix incorrect dev_tracker usage While inves
CVE-2022-50166	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: When HCI work queue is drained, onl
CVE-2022-50167	In the Linux kernel, the following vulnerability has been resolved: bpf: fix potential 32-bit overflow when accessing
CVE-2022-50170	In the Linux kernel, the following vulnerability has been resolved: kunit: executor: Fix a memory leak on failure in k
CVE-2022-50168	In the Linux kernel, the following vulnerability has been resolved: bpf, x86: fix freeing of not-finalized bpf_prog_pa
CVE-2022-50174	In the Linux kernel, the following vulnerability has been resolved: net: hinic: avoid kernel hung in hinic_get_stats64
CVE-2022-50176	In the Linux kernel, the following vulnerability has been resolved: drm/mcde: Fix refcount leak in mcde_dsi_bind Ev
CVE-2022-50178	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: 8852a: rfk: fix div 0 exception The D
CVE-2022-50179	In the Linux kernel, the following vulnerability has been resolved: ath9k: fix use-after-free in ath9k_hif_usb_rx_cb S
CVE-2022-50181	In the Linux kernel, the following vulnerability has been resolved: virtio-gpu: fix a missing check to avoid NULL de
CVE-2022-50185	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix potential buffer overflow in ni_se
CVE-2022-50186	In the Linux kernel, the following vulnerability has been resolved: ath11k: fix missing skb drop on htc_tx_completi
CVE-2022-50187	In the Linux kernel, the following vulnerability has been resolved: ath11k: fix netdev open race Make sure to allocat
CVE-2022-50189	In the Linux kernel, the following vulnerability has been resolved: tools/power turbostat: Fix file pointer leak Curren
CVE-2022-50195	In the Linux kernel, the following vulnerability has been resolved: ARM: dts: qcom: replace gcc PXO with pxo_boar
CVE-2022-50201	In the Linux kernel, the following vulnerability has been resolved: selinux: fix memleak in security_read_state_kern
CVE-2022-50197	In the Linux kernel, the following vulnerability has been resolved: cpufreq: zynq: Fix refcount leak in zynq_get_revi
CVE-2022-50202	In the Linux kernel, the following vulnerability has been resolved: PM: hibernate: defer device probing when resum
CVE-2022-50203	In the Linux kernel, the following vulnerability has been resolved: ARM: OMAP2+: display: Fix refcount leak bug I
CVE-2022-50211	In the Linux kernel, the following vulnerability has been resolved: md-raid10: fix KASAN warning There's a KASA
CVE-2022-50212	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: do not allow CHAIN_ID to re
CVE-2022-50213	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: do not allow SET_ID to refer

CVE-2022-50215	In the Linux kernel, the following vulnerability has been resolved: scsi: sg: Allow waiting for commands to complete
CVE-2022-50218	In the Linux kernel, the following vulnerability has been resolved: iio: light: isl29028: Fix the warning in isl29028_r
CVE-2022-50219	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix KASAN use-after-free Read in compute
CVE-2022-50220	In the Linux kernel, the following vulnerability has been resolved: usbnet: Fix linkwatch use-after-free on disconnect
CVE-2022-50222	In the Linux kernel, the following vulnerability has been resolved: tty: vt: initialize unicode screen buffer syzbot rep
CVE-2022-50221	In the Linux kernel, the following vulnerability has been resolved: drm/fb-helper: Fix out-of-bounds access Clip men
CVE-2022-50224	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Treat NX as a valid SPTE bit fo
CVE-2022-50226	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp - Use kzalloc for sev ioctl interfaces t
CVE-2022-50228	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Don't BUG if userspace injects an in
CVE-2022-50229	In the Linux kernel, the following vulnerability has been resolved: ALSA: bcd2000: Fix a UAF bug on the error path
CVE-2022-50230	In the Linux kernel, the following vulnerability has been resolved: arm64: set UXN on swapper page tables [This is
CVE-2022-50232	In the Linux kernel, the following vulnerability has been resolved: arm64: set UXN on swapper page tables [This is
CVE-2023-0030	A use-after-free flaw was found in the Linux kernel's nouveau driver in how a user triggers a memory overflow that
CVE-2023-0045	The current implementation of the prctl syscall does not issue an IBPB immediately during the syscall. The ib_prctl_
CVE-2023-0054	Out-of-bounds Write in GitHub repository vim/vim prior to 9.0.1145.
CVE-2023-0049	Out-of-bounds Read in GitHub repository vim/vim prior to 9.0.1143.
CVE-2023-0160	A deadlock flaw was found in the Linux kernel's BPF subsystem. This flaw allows a local user to potentially crash th
CVE-2023-0215	The public API function BIO_new_NDEF is a helper function used for streaming ASN.1 data via a BIO. It is primar
CVE-2023-0286	There is a type confusion vulnerability relating to X.400 address processing inside an X.509 GeneralName. X.400 ad
CVE-2023-0288	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1189.
CVE-2023-0433	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1225.
CVE-2023-0464	A security vulnerability has been identified in all supported versions of OpenSSL related to the verification of X.509
CVE-2023-0361	A timing side-channel in the handling of RSA ClientKeyExchange messages was discovered in GnuTLS. This side-c
CVE-2023-0465	Applications that use a non-default option when verifying certificates may be vulnerable to an attack from a malicio
CVE-2023-0466	The function X509_VERIFY_PARAM_add0_policy() is documented to implicitly enable the certificate policy check
CVE-2023-0512	Divide By Zero in GitHub repository vim/vim prior to 9.0.1247.
CVE-2023-0645	An out of bounds read exists in libjxl. An attacker using a specifically crafted file could cause an out of bounds read
CVE-2023-0795	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3488, allowing attackers to cause a denial-of-s
CVE-2023-0797	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in libtiff/tif_unix.c:368, invoked by tools/tiffcrop.c:2903 and too
CVE-2023-0796	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3592, allowing attackers to cause a denial-of-s
CVE-2023-0798	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3400, allowing attackers to cause a denial-of-s
CVE-2023-0799	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3701, allowing attackers to cause a denial-of-s
CVE-2023-1076	A flaw was found in the Linux Kernel. The tun/tap sockets have their socket UID hardcoded to 0 due to a type confu
CVE-2023-1127	Divide By Zero in GitHub repository vim/vim prior to 9.0.1367.
CVE-2023-1077	In the Linux kernel, pick_next_rt_entity() may return a type confused entry, not detected by the BUG_ON condition
CVE-2023-1170	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1376.
CVE-2023-1175	Incorrect Calculation of Buffer Size in GitHub repository vim/vim prior to 9.0.1378.
CVE-2023-1193	A use-after-free flaw was found in setup_async_work in the KSMBD implementation of the in-kernel samba server a
CVE-2023-1264	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.1392.
CVE-2023-1370	[Json-smart](https://netplex.github.io/json-smart/) is a performance focused, JSON processor lib. When reaching a '[

CVE-2023-1428	There exists an vulnerability causing an abort() to be called in gRPC. The following headers cause gRPC's C++ impl
CVE-2023-1380	A slab-out-of-bound read problem was found in brcmf_get_assoc_ies in drivers/net/wireless/broadcom/brcm80211/b
CVE-2023-1667	A NULL pointer dereference was found In libssh during re-keying with algorithm guessing. This issue may allow an
CVE-2023-1916	A flaw was found in tiffcrop, a program distributed by the libtiff package. A specially crafted tiff file can lead to an
CVE-2023-1932	A flaw was found in hibernate-validator's 'isValid' method in the org.hibernate.validator.internal.constraintvalidators
CVE-2023-1972	A potential heap based buffer overflow was found in _bfd_elf_slurp_version_tables() in bfd/elf.c. This may lead to l
CVE-2023-2007	The specific flaw exists within the DPT I2O Controller driver. The issue results from the lack of proper locking whe
CVE-2023-20859	In Spring Vault, versions 3.0.x prior to 3.0.2 and versions 2.3.x prior to 2.3.3 and older versions, an application is vu
CVE-2023-21930	Vulnerability in the Oracle Java SE, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: JSS
CVE-2023-20863	In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a speci
CVE-2023-21937	Vulnerability in the Oracle Java SE, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Net
CVE-2023-21938	Vulnerability in the Oracle Java SE, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Lib
CVE-2023-21939	Vulnerability in the Oracle Java SE, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Sw
CVE-2023-21954	Vulnerability in the Oracle Java SE, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Ho
CVE-2023-21967	Vulnerability in the Oracle Java SE, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: JSS
CVE-2023-22656	Out-of-bounds read in Intel(R) Media SDK and some Intel(R) oneVPL software before version 23.3.5 may allow an
CVE-2023-2253	A flaw was found in the `/v2/_catalog` endpoint in distribution/distribution, which accepts a parameter to control the
CVE-2023-2283	A vulnerability was found in libssh, where the authentication check of the connecting client can be bypassed in the`p
CVE-2023-22946	In Apache Spark versions prior to 3.4.0, applications using spark-submit can specify a 'proxy-user' to run as, limiting
CVE-2023-24056	In pkgconf through 1.9.3, variable duplication can cause unbounded string expansion due to incorrect checks in libpl
CVE-2023-23916	An allocation of resources without limits or throttling vulnerability exists in curl <v7.88.0 based on the "chained" HT
CVE-2023-2431	A security issue was discovered in Kubelet that allows pods to bypass the seccomp profile enforcement. Pods that us
CVE-2023-24329	An issue in the urllib.parse component of Python before 3.11.4 allows attackers to bypass blocklisting methods by su
CVE-2023-24532	The ScalarMult and ScalarBaseMult methods of the P256 Curve may return an incorrect result if called with some sp
CVE-2023-24534	HTTP and MIME header parsing can allocate large amounts of memory, even when parsing small inputs, potentially
CVE-2023-24537	Calling any of the Parse functions on Go source code which contains //line directives with very large line numbers ca
CVE-2023-24536	Multipart form parsing can consume large amounts of CPU and memory when processing form inputs containing ve
CVE-2023-24538	Templates do not properly consider backticks (`) as Javascript string delimiters, and do not escape them as expected.
CVE-2023-24539	Angle brackets (<>) are not considered dangerous characters when inserted into CSS contexts. Templates containing
CVE-2023-2454	schema_element defeats protective search_path changes; It was found that certain database calls in PostgreSQL coul
CVE-2023-24540	Not all valid JavaScript whitespace characters are considered to be whitespace. Templates containing whitespace cha
CVE-2023-2455	Row security policies disregard user ID changes after inlining; PostgreSQL could permit incorrect policies to be app
CVE-2023-24607	Qt before 6.4.3 allows a denial of service via a crafted string when the SQL ODBC driver plugin is used and the size
CVE-2023-2491	A flaw was found in the Emacs text editor. Processing a specially crafted org-mode code with the "org-babel-execute
CVE-2023-25433	libtiff 4.5.0 is vulnerable to Buffer Overflow via /libtiff/tools/tiffcrop.c:8499. Incorrect updating of buffer size after r
CVE-2023-25434	libtiff 4.5.0 is vulnerable to Buffer Overflow via extractContigSamplesBytes() at /libtiff/tools/tiffcrop.c:3215.
CVE-2023-25435	libtiff 4.5.0 is vulnerable to Buffer Overflow via extractContigSamplesShifted8bits() at /libtiff/tools/tiffcrop.c:3753.
CVE-2023-25584	An out-of-bounds read flaw was found in the parse_module function in bfd/vms-alpha.c in Binutils.
CVE-2023-25585	A flaw was found in Binutils. The use of an uninitialized field in the struct module *module may lead to application
CVE-2023-25588	A flaw was found in Binutils. The field `the_bfd` of `asymbol` struct is uninitialized in the `bfd_mach_o_get_synt

CVE-2023-26048	Jetty is a java based web server and servlet engine. In affected versions servlets with multipart support (e.g. annotate
CVE-2023-26049	Jetty is a java based web server and servlet engine. Nonstandard cookie parsing in Jetty may allow an attacker to sm
CVE-2023-2609	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.1531.
CVE-2023-2610	Integer Overflow or Wraparound in GitHub repository vim/vim prior to 9.0.1532.
CVE-2023-26242	afu_mmio_region_get_by_offset in drivers/fpga/dfi-afu-region.c in the Linux kernel through 6.1.12 has an integer ov
CVE-2023-2650	Issue summary: Processing some specially crafted ASN.1 object identifiers or data containing them may be very slow
CVE-2023-26604	systemd before 247 does not adequately block local privilege escalation for some Sudo configurations, e.g., plausible
CVE-2023-26965	loadImage() in tools/tiffcrop.c in LibTIFF through 4.5.0 has a heap-based use after free via a crafted TIFF image.
CVE-2023-26966	libtiff 4.5.0 is vulnerable to Buffer Overflow in uv_encode() when libtiff reads a corrupted little-endian TIFF file and
CVE-2023-27043	The email module of Python through 3.11.3 incorrectly parses e-mail addresses that contain a special character. The
CVE-2023-2727	Users may be able to launch containers using images that are restricted by ImagePolicyWebhook when using ephem
CVE-2023-2728	Users may be able to launch containers that bypass the mountable secrets policy enforced by the ServiceAccount ad
CVE-2023-27533	A vulnerability in input validation exists in curl <8.0 during communication using the TELNET protocol may allow
CVE-2023-27534	A path traversal vulnerability exists in curl <8.0.0 SFTP implementation causes the tilde (~) character to be wrongly
CVE-2023-27536	An authentication bypass vulnerability exists libcurl <8.0.0 in the connection reuse feature which can reuse previous
CVE-2023-27535	An authentication bypass vulnerability exists in libcurl <8.0.0 in the FTP connection reuse feature that can result in v
CVE-2023-27538	An authentication bypass vulnerability exists in libcurl prior to v8.0.0 where it reuses a previously established SSH c
CVE-2023-2828	Every `named` instance configured to run as a recursive resolver maintains a cache database holding the responses to
CVE-2023-28484	In libxml2 before 2.10.4, parsing of certain invalid XSD schemas can lead to a NULL pointer dereference and subse
CVE-2023-28322	An information disclosure vulnerability exists in curl <v8.1.0 when doing HTTP(S) transfers, libcurl might erroneou
CVE-2023-28617	org-babel-execute:latex in ob-latex.el in Org Mode through 9.6.1 for GNU Emacs allows attackers to execute arbitra
CVE-2023-28321	An improper certificate validation vulnerability exists in curl <v8.1.0 in the way it supports matching of wildcard pa
CVE-2023-28746	Information exposure through microarchitectural state after transient execution from some register files for some Inte
CVE-2023-29400	Templates containing actions in unquoted HTML attributes (e.g. "attr={.}")) executed with empty input can result
CVE-2023-28842	Moby) is an open source container framework developed by Docker Inc. that is distributed as Docker, Mirantis Cont
CVE-2023-29383	In Shadow 4.13, it is possible to inject control characters into fields provided to the SUID program chfn (change finge
CVE-2023-28840	Moby is an open source container framework developed by Docker Inc. that is distributed as Docker, Mirantis Conta
CVE-2023-28841	Moby is an open source container framework developed by Docker Inc. that is distributed as Docker, Mirantis Conta
CVE-2023-29402	The go command may generate unexpected code at build time when using cgo. This may result in unexpected behav
CVE-2023-29405	The go command may execute arbitrary code at build time when using cgo. This may occur when running "go get" o
CVE-2023-29403	On Unix platforms, the Go runtime does not behave differently when a binary is run with the setuid/setgid bits. This
CVE-2023-29404	The go command may execute arbitrary code at build time when using cgo. This may occur when running "go get" o
CVE-2023-29406	The HTTP/1 client does not fully validate the contents of the Host header. A maliciously crafted Host header can inj
CVE-2023-29469	An issue was discovered in libxml2 before 2.10.4. When hashing empty dict strings in a crafted XML document, xm
CVE-2023-29491	ncurses before 6.4 20230408, when used by a setuid application, allows local users to trigger security-relevant memo
CVE-2023-29409	Extremely large RSA keys in certificate chains can cause a client/server to expend significant CPU time verifying sig
CVE-2023-29499	A flaw was found in GLib. GVariant deserialization fails to validate that the input conforms to the expected format,
CVE-2023-29483	eventlet before 0.35.2, as used in dnspython before 2.6.0, allows remote attackers to interfere with DNS name resolu
CVE-2023-2976	Use of Java's default temporary directory for file creation in `FileBackedOutputStream` in Google Guava versions 1.
CVE-2023-2953	A vulnerability was found in openldap. This security flaw causes a null pointer dereference in ber_memalloc_x() fun

CVE-2023-3022	A flaw was found in the IPv6 module of the Linux kernel. The arg.result was not used consistently in fib6_rule_lookup.
CVE-2023-30086	Buffer Overflow vulnerability found in Libtiff V.4.0.7 allows a local attacker to cause a denial of service via the tiffcrop utility.
CVE-2023-3006	A known cache speculation vulnerability, known as Branch History Injection (BHI) or Spectre-BHB, becomes actual when the CPU is in a certain state.
CVE-2023-30775	A vulnerability was found in the libtiff library. This security flaw causes a heap buffer overflow in extractContigSamples.
CVE-2023-30774	A vulnerability was found in the libtiff library. This flaw causes a heap buffer overflow issue via the TIFFTAG_INK.
CVE-2023-31082	An issue was discovered in drivers/tty/n_gsm.c in the Linux kernel 6.2. There is a sleeping function called from an interrupt context.
CVE-2023-31085	An issue was discovered in drivers/mtd/ubi/cdev.c in the Linux kernel 6.2. There is a divide-by-zero error in do_divide.
CVE-2023-31437	An issue was discovered in systemd 253. An attacker can modify a sealed log file such that, in some views, not all events are visible.
CVE-2023-31438	An issue was discovered in systemd 253. An attacker can truncate a sealed log file and then resume log sealing such that the log is not properly sealed.
CVE-2023-31484	CPAN.pm before 2.35 does not verify TLS certificates when downloading distributions over HTTPS.
CVE-2023-31439	An issue was discovered in systemd 253. An attacker can modify the contents of past events in a sealed log file and then resume log sealing.
CVE-2023-31486	HTTP::Tiny before 0.083, a Perl core module since 5.13.9 and available standalone on CPAN, has an insecure default configuration.
CVE-2023-3159	A use after free issue was discovered in driver/firewire in outbound_phy_packet_callback in the Linux Kernel. In this case, the driver uses a pointer that has been freed.
CVE-2023-3164	A heap-buffer-overflow vulnerability was found in LibTIFF, in extractImageSection() at tools/tiffcrop.c:7916 and tools/tiffcrop.c:7917.
CVE-2023-32636	A flaw was found in glib, where the gvariant deserialization code is vulnerable to a denial of service introduced by a crafted GVariant.
CVE-2023-32611	A flaw was found in GLib. GVariant deserialization is vulnerable to a slowdown issue where a crafted GVariant can cause a denial of service.
CVE-2023-32665	A flaw was found in GLib. GVariant deserialization is vulnerable to an exponential blowup issue where a crafted GVariant can cause a denial of service.
CVE-2023-32681	Requests is a HTTP library. Since Requests 2.3.0, Requests has been leaking Proxy-Authorization headers to destination servers.
CVE-2023-32731	When gRPC HTTP2 stack raised a header size exceeded error, it skipped parsing the rest of the HPACK frame. This can lead to a denial of service.
CVE-2023-32732	gRPC contains a vulnerability whereby a client can cause a termination of connection between a HTTP2 proxy and a server.
CVE-2023-33053	Memory corruption in Kernel while parsing metadata.
CVE-2023-3316	A NULL pointer dereference in TIFFClose() is caused by a failure to open an output file (non-existent path or a path that is not writable).
CVE-2023-33201	Bouncy Castle For Java before 1.74 is affected by an LDAP injection vulnerability. The vulnerability only affects applications that use the LDAP module.
CVE-2023-3341	The code that processes control channel messages sent to 'named' calls certain functions recursively during packet processing.
CVE-2023-33202	Bouncy Castle for Java before 1.73 contains a potential Denial of Service (DoS) issue within the Bouncy Castle.org's Bouncy Castle Java API.
CVE-2023-3358	A null pointer dereference was found in the Linux kernel's Integrated Sensor Hub (ISH) driver. This issue could allow a local attacker to cause a denial of service.
CVE-2023-34462	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance network servers and clients.
CVE-2023-3446	Issue summary: Checking excessively long DH keys or parameters may be very slow. Impact summary: Application may be slow to respond.
CVE-2023-3576	A memory leak flaw was found in Libtiff's tiffcrop utility. This issue occurs when tiffcrop operates on a TIFF image that is not a standard TIFF image.
CVE-2023-34969	D-Bus before 1.15.6 sometimes allows unprivileged users to crash dbus-daemon. If a privileged user with control over the bus is able to cause a crash.
CVE-2023-35790	An issue was discovered in dec_patch_dictionary.cc in libjxl before 0.8.2. An integer underflow in patch decoding can lead to a denial of service.
CVE-2023-36054	lib/kadm5/kadm_rpc_xdr.c in MIT Kerberos 5 (aka krb5) before 1.20.2 and 1.21.x before 1.21.1 frees an uninitialized memory buffer.
CVE-2023-3618	A flaw was found in libtiff. A specially crafted tiff file can lead to a segmentation fault due to a buffer overflow in the tiffcrop utility.
CVE-2023-3635	GzipSource does not handle an exception that might be raised when parsing a malformed gzip buffer. This may lead to a denial of service.
CVE-2023-3640	A possible unauthorized memory access flaw was found in the Linux kernel's cpu_entry_area mapping of X86 CPU registers.
CVE-2023-36478	Eclipse Jetty provides a web server and servlet container. In versions 11.0.0 through 11.0.15, 10.0.0 through 10.0.15, and 9.4.0 through 9.4.53, there is a denial of service vulnerability.
CVE-2023-36617	A ReDoS issue was discovered in the URI component before 0.12.2 for Ruby. The URI parser mishandles invalid URI characters.
CVE-2023-3676	A security issue was discovered in Kubernetes where a user that can create pods on Windows nodes may be able to escape the container.
CVE-2023-37454	An issue was discovered in the Linux kernel through 6.4.2. A crafted UDF filesystem image causes a use-after-free vulnerability.
CVE-2023-37769	stress-test master commit e4c878 was discovered to contain a FPE vulnerability via the component combine_inner and combine_outer.

CVE-2023-3817	Issue summary: Checking excessively long DH keys or parameters may be very slow. Impact summary: Application
CVE-2023-38546	This flaw allows an attacker to insert cookies at will into a running program using libcurl, if the specific series of con
CVE-2023-38709	Faulty input validation in the core of Apache allows malicious or exploitable backend/content generators to split HT
CVE-2023-39318	The html/template package does not properly handle HTML-like "" comment tokens, nor hashbang "#!" comment to
CVE-2023-3899	A vulnerability was found in subscription-manager that allows local privilege escalation due to inadequate authoriza
CVE-2023-39319	The html/template package does not apply the proper rules for handling occurrences of "<script", "<!--", and "</scrip
CVE-2023-39323	Line directives ("//line") can be used to bypass the restrictions on "//go:cgo_" directives, allowing blocked linker and
CVE-2023-39325	A malicious HTTP/2 client which rapidly creates requests and immediately resets them can cause excessive server re
CVE-2023-39326	A malicious HTTP sender can use chunk extensions to cause a receiver reading from a request or response body to r
CVE-2023-39328	A vulnerability was found in OpenJPEG similar to CVE-2019-6988. This flaw allows an attacker to bypass existing
CVE-2023-39329	A flaw was found in OpenJPEG. A resource exhaustion can occur in the opj_t1_decode_cblks function in tcd.c throu
CVE-2023-39410	When deserializing untrusted or corrupted data, it is possible for a reader to consume memory beyond the allowed co
CVE-2023-39417	IN THE EXTENSION SCRIPT, a SQL Injection vulnerability was found in PostgreSQL if it uses @extowner@, @
CVE-2023-3955	A security issue was discovered in Kubernetes where a user that can create pods on Windows nodes may be able to e
CVE-2023-39615	Xmlsoft Libxml2 v2.11.0 was discovered to contain an out-of-bounds read via the xmlSAX2StartElement() function
CVE-2023-3978	Text nodes not in the HTML namespace are incorrectly literally rendered, causing text which should be escaped to n
CVE-2023-39804	In GNU tar before 1.35, mishandled extension attributes in a PAX archive can lead to an application crash in xheade
CVE-2023-4010	A flaw was found in the USB Host Controller Driver framework in the Linux kernel. The usb_giveback_urb function
CVE-2023-4016	Under some circumstances, this weakness allows a user who has access to run the "ps" utility on a machine, the abil
CVE-2023-40167	Jetty is a Java based web server and servlet engine. Prior to versions 9.4.52, 10.0.16, 11.0.16, and 12.0.1, Jetty accep
CVE-2023-40217	An issue was discovered in Python before 3.8.18, 3.9.x before 3.9.18, 3.10.x before 3.10.13, and 3.11.x before 3.11.5
CVE-2023-4039	**DISPUTED** A failure in the -fstack-protector feature in GCC-based toolchains that target AArch64 allows an att
CVE-2023-40403	The issue was addressed with improved memory handling. This issue is fixed in macOS Ventura 13.6, tvOS 17, iOS
CVE-2023-40745	LibTIFF is vulnerable to an integer overflow. This flaw allows remote attackers to cause a denial of service (applicat
CVE-2023-41175	A vulnerability was found in libtiff due to multiple potential integer overflows in raw2tiff.c. This flaw allows remote
CVE-2023-4133	A use-after-free vulnerability was found in the cxgb4 driver in the Linux kernel. The bug occurs when the cxgb4 dev
CVE-2023-4156	A heap out-of-bounds read flaw was found in builtin.c in the gawk package. This issue may lead to a crash and could
CVE-2023-4194	A flaw was found in the Linux kernel's TUN/TAP functionality. This issue could allow a local user to bypass network
CVE-2023-42282	The ip package before 1.1.9 for Node.js might allow SSRF because some IP addresses (such as 0x7f.1) are improper
CVE-2023-42503	Improper Input Validation, Uncontrolled Resource Consumption vulnerability in Apache Commons Compress in TA
CVE-2023-43804	urllib3 is a user-friendly HTTP client library for Python. urllib3 doesn't treat the `Cookie` HTTP header special or pr
CVE-2023-4408	The DNS message parsing code in `named` includes a section whose computational complexity is overly high. It doe
CVE-2023-44271	An issue was discovered in Pillow before 10.0.0. It is a Denial of Service that uncontrollably allocates memory to pr
CVE-2023-44981	Authorization Bypass Through User-Controlled Key vulnerability in Apache ZooKeeper. If SASL Quorum Peer auth
CVE-2023-4504	Due to failure in validating the length provided by an attacker-crafted PPD PostScript document, CUPS and libppd a
CVE-2023-45142	OpenTelemetry-Go Contrib is a collection of third-party packages for OpenTelemetry-Go. A handler wrapper out of
CVE-2023-44487	The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can rese
CVE-2023-4527	A flaw was found in glibc. When the getaddrinfo function is called with the AF_UNSPEC address family and the sy
CVE-2023-45283	The filepath package does not recognize paths with a \\??\ prefix as special. On Windows, a path beginning with \\??\
CVE-2023-45284	On Windows, The IsLocal function does not correctly detect reserved device names in some cases. Reserved names

CVE-2023-45285	Using go get to fetch a module with the ".git" suffix may unexpectedly fallback to the insecure "git://" protocol if the
CVE-2023-45287	Before Go 1.20, the RSA based TLS key exchanges used the math/big library, which is not constant time. RSA blind
CVE-2023-45288	An attacker may cause an HTTP/2 endpoint to read arbitrary amounts of header data by sending an excessive number
CVE-2023-45289	When following an HTTP redirect to a domain which is not a subdomain match or exact match of the initial domain,
CVE-2023-45290	When parsing a multipart form (either explicitly with Request.ParseMultipartForm or implicitly with Request.FormV
CVE-2023-45322	libxml2 through 2.11.5 has a use-after-free that can only occur after a certain memory allocation fails. This occurs in
CVE-2023-45803	urllib3 is a user-friendly HTTP client library for Python. urllib3 previously wouldn't remove the HTTP request body
CVE-2023-45853	MiniZip in zlib through 1.3 has an integer overflow and resultant heap-based buffer overflow in zipOpenNewFileInZ
CVE-2023-45857	An issue discovered in Axios 1.5.1 inadvertently reveals the confidential XSRF-TOKEN stored in cookies by includ
CVE-2023-45896	ntfs3 in the Linux kernel through 6.8.0 allows a physically proximate attacker to read kernel memory by mounting a
CVE-2023-45918	Rejected reason: DO NOT USE THIS CANDIDATE NUMBER. ConsultIDs: none. Reason: This candidate was with
CVE-2023-45960	Rejected reason: DO NOT USE THIS CVE RECORD. ConsultIDs: none. Reason: This record was withdrawn by its
CVE-2023-46218	This flaw allows a malicious HTTP server to set "super cookies" in curl that are then passed back to more origins tha
CVE-2023-46246	Vim is an improved version of the good old UNIX editor Vi. Heap-use-after-free in memory allocated in the function
CVE-2023-46136	Werkzeug is a comprehensive WSGI web application library. If an upload of a file that starts with CR or LF and then
CVE-2023-4641	A flaw was found in shadow-utils. When asking for a new password, shadow-utils asks the password twice. If the pa
CVE-2023-46749	Apache Shiro before 1.13.0 or 2.0.0-alpha-4, may be susceptible to a path traversal attack that results in an authentic
CVE-2023-46750	URL Redirection to Untrusted Site ('Open Redirect') vulnerability when "form" authentication is used in Apache Shi
CVE-2023-47038	A vulnerability was found in perl 5.30.0 through 5.38.0. This issue occurs when a crafted regular expression is comp
CVE-2023-47233	The brcm80211 component in the Linux kernel through 6.5.10 has a brcmf_cfg80211_detach use-after-free in the de
CVE-2023-47108	OpenTelemetry-Go Contrib is a collection of third-party packages for OpenTelemetry-Go. Prior to version 0.46.0, th
CVE-2023-47282	Out-of-bounds write in Intel(R) Media SDK all versions and some Intel(R) oneVPL software before version 23.3.5 n
CVE-2023-4733	Use After Free in GitHub repository vim/vim prior to 9.0.1840.
CVE-2023-4734	Integer Overflow or Wraparound in GitHub repository vim/vim prior to 9.0.1846.
CVE-2023-4735	Out-of-bounds Write in GitHub repository vim/vim prior to 9.0.1847.
CVE-2023-4738	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1848.
CVE-2023-4750	Use After Free in GitHub repository vim/vim prior to 9.0.1857.
CVE-2023-4751	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1331.
CVE-2023-4752	Use After Free in GitHub repository vim/vim prior to 9.0.1858.
CVE-2023-4781	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1873.
CVE-2023-4813	A flaw was found in glibc. In an uncommon situation, the gaih_inet function may use memory that has been freed, re
CVE-2023-4806	A flaw was found in glibc. In an extremely rare situation, the getaddrinfo function may access memory that has been
CVE-2023-48161	Buffer Overflow vulnerability in GifLib Project GifLib v.5.2.1 allows a local attacker to obtain sensitive information
CVE-2023-48231	Vim is an open source command line text editor. When closing a window, vim may try to access already freed windo
CVE-2023-48232	Vim is an open source command line text editor. A floating point exception may occur when calculating the line offs
CVE-2023-48233	Vim is an open source command line text editor. If the count after the :s command is larger than what fits into a (sign
CVE-2023-48234	Vim is an open source command line text editor. When getting the count for a normal mode z command, it may over
CVE-2023-48235	Vim is an open source command line text editor. When parsing relative ex addresses one may unintentionally cause a
CVE-2023-48236	Vim is an open source command line text editor. When using the z= command, the user may overflow the count with
CVE-2023-48237	Vim is an open source command line text editor. In affected versions when shifting lines in operator pending mode a

CVE-2023-4863	Heap buffer overflow in libwebp in Google Chrome prior to 116.0.5845.187 and libwebp 1.3.2 allowed a remote attacker to execute arbitrary code via a crafted image.
CVE-2023-48706	Vim is a UNIX editor that, prior to version 9.0.2121, has a heap-use-after-free vulnerability. When executing a `:s` command, Vim would free a buffer that was still referenced by another buffer, leading to a crash or, in some cases, arbitrary code execution.
CVE-2023-48713	Knative Serving builds on Kubernetes to support deploying and serving of applications and functions as serverless components. A vulnerability was discovered in Knative Serving that could allow an attacker to execute arbitrary code on the host.
CVE-2023-48727	NULL pointer dereference in some Intel(R) oneVPL software before version 23.3.5 may allow an authenticated user to execute arbitrary code or cause a denial of service (crash) on the host.
CVE-2023-49501	Buffer Overflow vulnerability in Ffmpeg v.n6.1-3-g466799d4f5 allows a local attacker to execute arbitrary code via a crafted input.
CVE-2023-4911	A buffer overflow was discovered in the GNU C Library's dynamic loader ld.so while processing the GLIBC_TUNABLES environment variable.
CVE-2023-49502	Buffer Overflow vulnerability in Ffmpeg v.n6.1-3-g466799d4f5 allows a local attacker to execute arbitrary code via a crafted input.
CVE-2023-48795	The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows an attacker to execute arbitrary code on the host.
CVE-2023-49528	Buffer Overflow vulnerability in FFmpeg version n6.1-3-g466799d4f5, allows a local attacker to execute arbitrary code via a crafted input.
CVE-2023-50007	FFmpeg v.n6.1-3-g466799d4f5 allows an attacker to trigger use of a parameter of negative size in the av_samples_get_buffer_size function.
CVE-2023-50008	FFmpeg v.n6.1-3-g466799d4f5 allows memory consumption when using the colorcorrect filter, in the av_malloc function.
CVE-2023-50387	Certain DNSSEC aspects of the DNS protocol (in RFC 4033, 4034, 4035, 6840, and related RFCs) allow remote attackers to execute arbitrary code on the host.
CVE-2023-50447	Pillow through 10.1.0 allows PIL.ImageMath.eval Arbitrary Code Execution via the environment parameter, a different vector than CVE-2023-50495.
CVE-2023-50495	NCurse v6.4-20230418 was discovered to contain a segmentation fault via the component _nc_wrap_entry().
CVE-2023-50868	The Closest Encloser Proof aspect of the DNS protocol (in RFC 5155 when RFC 9276 guidance is skipped) allows remote attackers to execute arbitrary code on the host.
CVE-2023-5090	A flaw was found in KVM. An improper check in svm_set_x2apic_msr_interception() may allow direct access to host memory.
CVE-2023-51074	json-path v2.8.0 was discovered to contain a stack overflow via the Criteria.parse() method.
CVE-2023-51257	An invalid memory write issue in Jasper-Software Jasper v.4.1.1 and before allows a local attacker to execute arbitrary code on the host.
CVE-2023-5156	A flaw was found in the GNU C Library. A recent fix for CVE-2023-4806 introduced the potential for a memory leak.
CVE-2023-51767	OpenSSH through 9.6, when common types of DRAM are used, might allow row hammer attacks (for authentication) via a crafted input.
CVE-2023-52323	PyCryptodome and pycryptodomex before 3.19.1 allow side-channel leakage for OAEP decryption, exploitable for a denial of service (crash) or, in some cases, arbitrary code execution.
CVE-2023-52355	An out-of-memory flaw was found in libtiff that could be triggered by passing a crafted tiff file to the TIFFRasterScan function.
CVE-2023-52356	A segment fault (SEGV) flaw was found in libtiff that could be triggered by passing a crafted tiff file to the TIFFReadRawStrip function.
CVE-2023-52428	In Connect2id Nimbus JOSE+JWT before 9.37.2, an attacker can cause a denial of service (resource consumption) via a crafted input.
CVE-2023-52435	In the Linux kernel, the following vulnerability has been resolved: net: prevent mss overflow in skb_segment() Once a packet is segmented, the mss of the resulting segments must not exceed the mss of the original packet.
CVE-2023-52447	In the Linux kernel, the following vulnerability has been resolved: bpf: Defer the free of inner map when necessary When a BPF map is freed, the inner map must be freed only after the outer map has been freed.
CVE-2023-52450	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/uncore: Fix NULL pointer dereference in intel_uncore_event_init() The intel_uncore_event_init() function was called with a NULL pointer, leading to a NULL pointer dereference.
CVE-2023-52451	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries/memhp: Fix access beyond end of memory In the powerpc/pseries/memhp driver, an access beyond the end of memory was allowed, leading to a crash or, in some cases, arbitrary code execution.
CVE-2023-52452	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix accesses to uninit stack slots Privileged programs can access uninit stack slots, leading to a crash or, in some cases, arbitrary code execution.
CVE-2023-52458	In the Linux kernel, the following vulnerability has been resolved: block: add check that partition length needs to be less than or equal to the disk size In the block layer, a check was added to ensure that the partition length is less than or equal to the disk size.
CVE-2023-52454	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: Fix a kernel panic when host sends an invalid command In the nvmet-tcp driver, a kernel panic was triggered when the host sends an invalid command.
CVE-2023-52463	In the Linux kernel, the following vulnerability has been resolved: efivarfs: force RO when remounting if SetVariable() fails In the efivarfs driver, the read-only (RO) flag was not set when remounting if the SetVariable() function failed.
CVE-2023-52465	In the Linux kernel, the following vulnerability has been resolved: power: supply: Fix null pointer dereference in smbus_i2c_adapter_remove() The smbus_i2c_adapter_remove() function was called with a NULL pointer, leading to a NULL pointer dereference.
CVE-2023-52464	In the Linux kernel, the following vulnerability has been resolved: EDAC/thunderx: Fix possible out-of-bounds string access In the EDAC/thunderx driver, a possible out-of-bounds string access was allowed, leading to a crash or, in some cases, arbitrary code execution.
CVE-2023-52466	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52468	In the Linux kernel, the following vulnerability has been resolved: class: fix use-after-free in class_register() The class_register() function was called with a NULL pointer, leading to a use-after-free error.
CVE-2023-52469	In the Linux kernel, the following vulnerability has been resolved: drivers/amd/pm: fix a use-after-free in kv_parse() The kv_parse() function was called with a NULL pointer, leading to a use-after-free error.
CVE-2023-52470	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: check the alloc_workqueue return value In the drm/radeon driver, the alloc_workqueue() function was not checked for a NULL return value.
CVE-2023-52471	In the Linux kernel, the following vulnerability has been resolved: ice: Fix some null pointer dereference issues in ice_vc_tx_desc() The ice_vc_tx_desc() function was called with a NULL pointer, leading to a NULL pointer dereference.
CVE-2023-52474	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix bugs with non-PAGE_SIZE-end multi-page RDMA writes In the IB/hfi1 driver, several bugs were fixed related to non-PAGE_SIZE-end multi-page RDMA writes.

CVE-2023-52475	In the Linux kernel, the following vulnerability has been resolved: Input: powermate - fix use-after-free in powermate
CVE-2023-52476	In the Linux kernel, the following vulnerability has been resolved: perf/x86/lbr: Filter vsyscall addresses We found t
CVE-2023-52478	In the Linux kernel, the following vulnerability has been resolved: HID: logitech-hidpp: Fix kernel crash on receiver
CVE-2023-52481	In the Linux kernel, the following vulnerability has been resolved: arm64: errata: Add Cortex-A520 speculative unpr
CVE-2023-52485	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Wake DMCUB before sending
CVE-2023-52482	In the Linux kernel, the following vulnerability has been resolved: x86/srso: Add SRSO mitigation for Hygon proces
CVE-2023-52486	In the Linux kernel, the following vulnerability has been resolved: drm: Don't unref the same fb many times by mista
CVE-2023-52490	In the Linux kernel, the following vulnerability has been resolved: mm: migrate: fix getting incorrect page mapping
CVE-2023-52492	In the Linux kernel, the following vulnerability has been resolved: dmaengine: fix NULL pointer in channel unregist
CVE-2023-52498	In the Linux kernel, the following vulnerability has been resolved: PM: sleep: Fix possible deadlocks in core system
CVE-2023-52508	In the Linux kernel, the following vulnerability has been resolved: nvme-fc: Prevent null pointer dereference in nvme
CVE-2023-52516	In the Linux kernel, the following vulnerability has been resolved: dma-debug: don't call __dma_entry_alloc_check
CVE-2023-52519	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: ipc: Disable and reenabl
CVE-2023-52522	In the Linux kernel, the following vulnerability has been resolved: net: fix possible store tearing in neigh_periodic_w
CVE-2023-52523	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Reject sk_msg egress redirects to n
CVE-2023-52530	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix potential key use-after-free
CVE-2023-52532	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix TX CQE error handling For an unl
CVE-2023-52560	In the Linux kernel, the following vulnerability has been resolved: mm/damon/vaddr-test: fix memory leak in damon
CVE-2023-52561	In the Linux kernel, the following vulnerability has been resolved: arm64: dts: qcom: sdm845-db845c: Mark cont sp
CVE-2023-52568	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Resolves SECS reclaim vs. page fault fo
CVE-2023-52569	In the Linux kernel, the following vulnerability has been resolved: btrfs: remove BUG() after failure to insert delaye
CVE-2023-52570	In the Linux kernel, the following vulnerability has been resolved: vfio/mdev: Fix a null-ptr-deref bug for mdev_unr
CVE-2023-52575	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52576	In the Linux kernel, the following vulnerability has been resolved: x86/mm, kexec, ima: Use memblock_free_late()
CVE-2023-52582	In the Linux kernel, the following vulnerability has been resolved: netfs: Only call folio_start_fscache() one time for
CVE-2023-52583	In the Linux kernel, the following vulnerability has been resolved: ceph: fix deadlock or deadcode of misusing dget(
CVE-2023-52585	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix possible NULL dereference in a
CVE-2023-52586	In the Linux kernel, the following vulnerability has been resolved: drm/msm/dpu: Add mutex lock in control vblank
CVE-2023-52587	In the Linux kernel, the following vulnerability has been resolved: IB/ipoib: Fix mcast list locking Releasing the `pri
CVE-2023-52589	In the Linux kernel, the following vulnerability has been resolved: media: rkisp1: Fix IRQ disable race issue In rkisp
CVE-2023-52590	In the Linux kernel, the following vulnerability has been resolved: ocfs2: Avoid touching renamed directory if paren
CVE-2023-52593	In the Linux kernel, the following vulnerability has been resolved: wifi: wfx: fix possible NULL pointer dereference
CVE-2023-52591	In the Linux kernel, the following vulnerability has been resolved: reiserfs: Avoid touching renamed directory if par
CVE-2023-52605	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52614	In the Linux kernel, the following vulnerability has been resolved: PM / devfreq: Fix buffer overflow in trans_stat_sl
CVE-2023-52615	In the Linux kernel, the following vulnerability has been resolved: hwrng: core - Fix page fault dead lock on mmap-p
CVE-2023-52619	In the Linux kernel, the following vulnerability has been resolved: pstore/ram: Fix crash when setting number of cpu
CVE-2023-52621	In the Linux kernel, the following vulnerability has been resolved: bpf: Check rcu_read_lock_trace_held() before cal
CVE-2023-52622	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid online resizing failures due to oversiz
CVE-2023-52623	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix a suspicious RCU usage warning I

CVE-2023-52624	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Wake DMCUB before executing
CVE-2023-52625	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Refactor DMCUB enter/exit id
CVE-2023-52626	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix operation precedence bug in port t
CVE-2023-52629	In the Linux kernel, the following vulnerability has been resolved: sh: push-switch: Reorder cleanup operations to av
CVE-2023-52632	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix lock dependency warning with
CVE-2023-52634	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix disable_otg_wa logic [Why
CVE-2023-52639	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: vsie: fix race during shadow creation
CVE-2023-52642	In the Linux kernel, the following vulnerability has been resolved: media: rc: bpf attach/detach requires write permis
CVE-2023-52646	In the Linux kernel, the following vulnerability has been resolved: aio: fix mmap after fork null-deref Commit e4a
CVE-2023-52648	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Unmap the surface before resetting
CVE-2023-52653	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: fix a memleak in gss_import_v2_cont
CVE-2023-52657	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/amd/pm: resolve reboot exception fo
CVE-2023-52658	In the Linux kernel, the following vulnerability has been resolved: Revert "net/mlx5: Block entering switchdev mode
CVE-2023-52660	In the Linux kernel, the following vulnerability has been resolved: media: rkisp1: Fix IRQ handling due to shared in
CVE-2023-52662	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: fix a memleak in vmw_gmrid_man
CVE-2023-52663	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: amd: Fix memory leak in amd_sof_a
CVE-2023-52664	In the Linux kernel, the following vulnerability has been resolved: net: atlantic: eliminate double free in error handli
CVE-2023-52667	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: fix a potential double-free in fs_any_c
CVE-2023-52669	In the Linux kernel, the following vulnerability has been resolved: crypto: s390/aes - Fix buffer overread in CTR mo
CVE-2023-52671	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix hang/underflow when trans
CVE-2023-52673	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix a debugfs null pointer error
CVE-2023-52674	In the Linux kernel, the following vulnerability has been resolved: ALSA: scarlett2: Add clamp() in scarlett2_mixer
CVE-2023-52675	In the Linux kernel, the following vulnerability has been resolved: powerpc/imc-pmu: Add a null pointer check in up
CVE-2023-52676	In the Linux kernel, the following vulnerability has been resolved: bpf: Guard stack limits against 32bit overflow Th
CVE-2023-52679	In the Linux kernel, the following vulnerability has been resolved: of: Fix double free in of_parse_phandle_with_arg
CVE-2023-52680	In the Linux kernel, the following vulnerability has been resolved: ALSA: scarlett2: Add missing error checks to *_c
CVE-2023-52682	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to wait on block writeback for post_read
CVE-2023-52683	In the Linux kernel, the following vulnerability has been resolved: ACPI: LPIT: Avoid u32 multiplication overflow
CVE-2023-52689	In the Linux kernel, the following vulnerability has been resolved: ALSA: scarlett2: Add missing mutex lock around
CVE-2023-52686	In the Linux kernel, the following vulnerability has been resolved: powerpc/powernv: Add a null pointer check in op
CVE-2023-52697	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: sof_sdw_rt_sdca_jack_common: ctx
CVE-2023-52698	In the Linux kernel, the following vulnerability has been resolved: calipso: fix memory leak in netlbl_calipso_add_p
CVE-2023-52700	In the Linux kernel, the following vulnerability has been resolved: tipc: fix kernel warning when sending SYN mess
CVE-2023-52701	In the Linux kernel, the following vulnerability has been resolved: net: use a bounce buffer for copying skb->mark s
CVE-2023-52702	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix possible memory leak in ov
CVE-2023-52703	In the Linux kernel, the following vulnerability has been resolved: net/usb: kalmia: Don't pass act_len in usb_bulk_m
CVE-2023-52707	In the Linux kernel, the following vulnerability has been resolved: sched/psi: Fix use-after-free in ep_remove_wait
CVE-2023-52730	In the Linux kernel, the following vulnerability has been resolved: mmc: sdio: fix possible resource leaks in some er
CVE-2023-52731	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix invalid page access after closing deferr
CVE-2023-52732	In the Linux kernel, the following vulnerability has been resolved: ceph: blocklist the kclient when receiving corrupt

CVE-2023-52735	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Don't let sock_map_{close,destroy}
CVE-2023-52736	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: Do not unset preset when cleaning up
CVE-2023-52737	In the Linux kernel, the following vulnerability has been resolved: btrfs: lock the inode in shared mode before starting
CVE-2023-52738	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/fence: Fix oops due to non-matching
CVE-2023-52739	In the Linux kernel, the following vulnerability has been resolved: Fix page corruption caused by racy check in __fr
CVE-2023-52741	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix use-after-free in rdata->read_into_pagest
CVE-2023-52743	In the Linux kernel, the following vulnerability has been resolved: ice: Do not use WQ_MEM_RECLAIM flag for w
CVE-2023-52744	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix potential NULL-ptr-dereference
CVE-2023-52745	In the Linux kernel, the following vulnerability has been resolved: IB/IPoIB: Fix legacy IPoIB due to wrong number
CVE-2023-52747	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Restore allocated resources on failed cop
CVE-2023-52749	In the Linux kernel, the following vulnerability has been resolved: spi: Fix null dereference on suspend A race condi
CVE-2023-52750	In the Linux kernel, the following vulnerability has been resolved: arm64: Restrict CPU_BIG_ENDIAN to GNU as
CVE-2023-52751	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix use-after-free in smb2_query_info
CVE-2023-52756	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52754	In the Linux kernel, the following vulnerability has been resolved: media: imon: fix access to invalid resource for the
CVE-2023-52760	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix slab-use-after-free in gfs2_qd_dealloc In
CVE-2023-52758	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52761	In the Linux kernel, the following vulnerability has been resolved: riscv: VMAP_STACK overflow detection thread
CVE-2023-52762	In the Linux kernel, the following vulnerability has been resolved: virtio-blk: fix implicit overflow on virtio_max_dr
CVE-2023-52764	In the Linux kernel, the following vulnerability has been resolved: media: gspca: cpia1: shift-out-of-bounds in set_fl
CVE-2023-52775	In the Linux kernel, the following vulnerability has been resolved: net/smc: avoid data corruption caused by decline
CVE-2023-52777	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix gtk offload status event locking
CVE-2023-52781	In the Linux kernel, the following vulnerability has been resolved: usb: config: fix iteration issue in 'usb_get_bos_de
CVE-2023-52782	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Track xmit submission to PTP WQ aft
CVE-2023-52784	In the Linux kernel, the following vulnerability has been resolved: bonding: stop the device in bond_setup_by_slave
CVE-2023-52788	In the Linux kernel, the following vulnerability has been resolved: i915/perf: Fix NULL deref bugs with drm_dbg()
CVE-2023-52791	In the Linux kernel, the following vulnerability has been resolved: i2c: core: Run atomic i2c xfer when !preemptible
CVE-2023-52796	In the Linux kernel, the following vulnerability has been resolved: ipvlan: add ipvlan_route_v6_outbound() helper In
CVE-2023-52798	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix dfs radar event locking The ath1
CVE-2023-52800	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix htt pktlog locking The ath11k ac
CVE-2023-52803	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix RPC client cleaned up the freed pi
CVE-2023-52806	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: Fix possible null-ptr-deref when assi
CVE-2023-52808	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Set debugfs_dir pointer to NULL a
CVE-2023-52809	In the Linux kernel, the following vulnerability has been resolved: scsi: libfc: Fix potential NULL pointer dereference
CVE-2023-52811	In the Linux kernel, the following vulnerability has been resolved: scsi: ibmvfc: Remove BUG_ON in the case of an
CVE-2023-52813	In the Linux kernel, the following vulnerability has been resolved: crypto: pccrypt - Fix hungtask for PADATA_RES
CVE-2023-52814	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential null pointer dereference
CVE-2023-52815	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/vkms: fix a possible null pointer der
CVE-2023-52817	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix a null pointer access when the s
CVE-2023-52818	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Fix UBSAN array-index-out-of-bounds

CVE-2023-52819	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Fix UBSAN array-index-out-of-bounds
CVE-2023-52831	In the Linux kernel, the following vulnerability has been resolved: cpu/hotplug: Don't offline the last non-isolated CPU
CVE-2023-52829	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix possible out-of-bound write in ath12k
CVE-2023-52832	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: don't return unset power in ieee80211
CVE-2023-52833	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btusb: Add date->evt_skb is NULL check
CVE-2023-52835	In the Linux kernel, the following vulnerability has been resolved: perf/core: Bail out early if the request AUX area is not
CVE-2023-52834	In the Linux kernel, the following vulnerability has been resolved: atl1c: Work around the DMA RX overflow issue
CVE-2023-52837	In the Linux kernel, the following vulnerability has been resolved: nbd: fix uaf in nbd_open Commit 4af5f2e03013 (
CVE-2023-52840	In the Linux kernel, the following vulnerability has been resolved: Input: synaptics-rmi4 - fix use after free in rmi_un
CVE-2023-52845	In the Linux kernel, the following vulnerability has been resolved: tipc: Change nla_policy for bearer-related names
CVE-2023-52847	In the Linux kernel, the following vulnerability has been resolved: media: btvtv: fix use after free error due to btv->tin
CVE-2023-52851	In the Linux kernel, the following vulnerability has been resolved: IB/mlx5: Fix init stage error handling to avoid do
CVE-2023-52854	In the Linux kernel, the following vulnerability has been resolved: padata: Fix refcnt handling in padata_free_shell()
CVE-2023-52857	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: Fix coverity issue with unintention
CVE-2023-52864	In the Linux kernel, the following vulnerability has been resolved: platform/x86: wmi: Fix opening of char device S
CVE-2023-52867	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: possible buffer overflow Buffer 'afm
CVE-2023-52872	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: fix race condition in status line chang
CVE-2023-52877	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tcpm: Fix NULL pointer dereference i
CVE-2023-52878	In the Linux kernel, the following vulnerability has been resolved: can: dev: can_put_echo_skb(): don't crash kernel
CVE-2023-52879	In the Linux kernel, the following vulnerability has been resolved: tracing: Have trace_event_file have ref counters T
CVE-2023-52880	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: require CAP_NET_ADMIN to attach
CVE-2023-52881	In the Linux kernel, the following vulnerability has been resolved: tcp: do not accept ACK of bytes we never sent TH
CVE-2023-52882	In the Linux kernel, the following vulnerability has been resolved: clk: sunxi-ng: h6: Reparent CPUX during PLL Cl
CVE-2023-52883	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix possible null pointer dereferenc
CVE-2023-52884	In the Linux kernel, the following vulnerability has been resolved: Input: cyapa - add missing input core locking to s
CVE-2023-52885	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix UAF in svc_tcp_listen_data_ready
CVE-2023-52886	In the Linux kernel, the following vulnerability has been resolved: USB: core: Fix race by not overwriting udev->des
CVE-2023-52887	In the Linux kernel, the following vulnerability has been resolved: net: can: j1939: enhanced error handling for tight
CVE-2023-52888	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Only free buffer VA th
CVE-2023-52889	In the Linux kernel, the following vulnerability has been resolved: apparmor: Fix null pointer deref when receiving s
CVE-2023-52898	In the Linux kernel, the following vulnerability has been resolved: xhci: Fix null pointer dereference when host dies
CVE-2023-52901	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Check endpoint is valid before derefer
CVE-2023-52904	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix possible NULL pointer der
CVE-2023-52905	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: Fix resource leakage in VF driver u
CVE-2023-52906	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_mpls: Fix warning during failed att
CVE-2023-52908	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential NULL dereference Fix
CVE-2023-52910	In the Linux kernel, the following vulnerability has been resolved: iommu/iova: Fix alloc iova overflows issue In _
CVE-2023-52912	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fixed bug on error when unloading
CVE-2023-52913	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix potential context UAFs gem_conte
CVE-2023-52915	In the Linux kernel, the following vulnerability has been resolved: media: dvb-usb-v2: af9035: Fix null-ptr-deref in a

CVE-2023-52916	In the Linux kernel, the following vulnerability has been resolved: media: aspeed: Fix memory overwrite if timing is
CVE-2023-52917	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52918	In the Linux kernel, the following vulnerability has been resolved: media: pci: cx23885: check cx23885_vdev_init()
CVE-2023-52920	In the Linux kernel, the following vulnerability has been resolved: bpf: support non-r10 register spill/fill to/from stack
CVE-2023-52921	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix possible UAF in amdgpu_cs_pa
CVE-2023-52922	In the Linux kernel, the following vulnerability has been resolved: can: bcm: Fix UAF in bcm_proc_show() BUG: K
CVE-2023-52923	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: adapt set backend to use GC t
CVE-2023-52925	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: don't fail inserts if duplicate h
CVE-2023-52924	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: don't skip expired elements du
CVE-2023-52926	In the Linux kernel, the following vulnerability has been resolved: IORING_OP_READ did not correctly consume the
CVE-2023-52927	In the Linux kernel, the following vulnerability has been resolved: netfilter: allow exp not to be removed in nf_ct_fir
CVE-2023-52928	In the Linux kernel, the following vulnerability has been resolved: bpf: Skip invalid kfunc call in backtrack_insn The
CVE-2023-52931	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Avoid potential vm use-after-free Addi
CVE-2023-52932	In the Linux kernel, the following vulnerability has been resolved: mm/swapfile: add cond_resched() in get_swap_pa
CVE-2023-52935	In the Linux kernel, the following vulnerability has been resolved: mm/khugepaged: fix ->anon_vma race If an ->an
CVE-2023-52936	In the Linux kernel, the following vulnerability has been resolved: kernel/irq/irqdomain.c: fix memory leak with usin
CVE-2023-52937	In the Linux kernel, the following vulnerability has been resolved: HV: hv_balloon: fix memory leak with using deb
CVE-2023-52938	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Don't attempt to resume the ports
CVE-2023-52939	In the Linux kernel, the following vulnerability has been resolved: mm: memcg: fix NULL pointer in mem_cgroup_
CVE-2023-52942	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: Fix wrong check in update_parent
CVE-2023-52973	In the Linux kernel, the following vulnerability has been resolved: vc_screen: move load of struct vc_data pointer in
CVE-2023-52974	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi_tcp: Fix UAF during login when access
CVE-2023-52975	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi_tcp: Fix UAF during logout when acces
CVE-2023-52976	In the Linux kernel, the following vulnerability has been resolved: efi: fix potential NULL deref in efi_mem_reserve
CVE-2023-52981	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix request ref counting during error ca
CVE-2023-52983	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix uaf for bfqq in bic_set_bfqq() After
CVE-2023-52984	In the Linux kernel, the following vulnerability has been resolved: net: phy: dp83822: Fix null pointer access on DP8
CVE-2023-52987	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-mtrace: prevent underflow in so
CVE-2023-52986	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Check for any of tcp_bpf_protos wh
CVE-2023-52988	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda/via: Avoid potential array out-of-bou
CVE-2023-52989	In the Linux kernel, the following vulnerability has been resolved: firewire: fix memory leak for payload of request s
CVE-2023-52990	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-52992	In the Linux kernel, the following vulnerability has been resolved: bpf: Skip task with pid=1 in send_signal_common
CVE-2023-52993	In the Linux kernel, the following vulnerability has been resolved: x86/i8259: Mark legacy PIC interrupts with IRQ_
CVE-2023-52994	In the Linux kernel, the following vulnerability has been resolved: acpi: Fix suspend with Xen PV Commit f1e52500
CVE-2023-52996	In the Linux kernel, the following vulnerability has been resolved: ipv4: prevent potential spectre v1 gadget in fib_m
CVE-2023-52997	In the Linux kernel, the following vulnerability has been resolved: ipv4: prevent potential spectre v1 gadget in ip_m
CVE-2023-53000	In the Linux kernel, the following vulnerability has been resolved: netlink: prevent potential spectre v1 gadgets Most
CVE-2023-53001	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-53002	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix a memory leak with reused mmap_

CVE-2023-53005	In the Linux kernel, the following vulnerability has been resolved: trace_events_hist: add check for return value of 'c
CVE-2023-53006	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix oops due to uncleared server->smbd_cor
CVE-2023-53007	In the Linux kernel, the following vulnerability has been resolved: tracing: Make sure trace_printk() can output as so
CVE-2023-53008	In the Linux kernel, the following vulnerability has been resolved: cifs: fix potential memory leaks in session setup M
CVE-2023-53009	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Add sync after creating vram bo Th
CVE-2023-53010	In the Linux kernel, the following vulnerability has been resolved: bnxt: Do not read past the end of test names Test
CVE-2023-53011	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: enable all safety features by default
CVE-2023-53013	In the Linux kernel, the following vulnerability has been resolved: ptdma: pt_core_execute_cmd() should use spinloc
CVE-2023-53015	In the Linux kernel, the following vulnerability has been resolved: HID: betop: check shape of output reports betopff
CVE-2023-53016	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix possible deadlock in rfcomm_sk_s
CVE-2023-53019	In the Linux kernel, the following vulnerability has been resolved: net: mdio: validate parameter addr in mdiobus_ge
CVE-2023-53020	In the Linux kernel, the following vulnerability has been resolved: l2tp: close all race conditions in l2tp_tunnel_regis
CVE-2023-53024	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix pointer-leak due to insufficient speculativ
CVE-2023-53026	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix ib block iterator counter overflo
CVE-2023-53031	In the Linux kernel, the following vulnerability has been resolved: powerpc/imc-pmu: Fix use of mutex in IRQs disa
CVE-2023-53032	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: Fix overflow before widen in the b
CVE-2023-53034	In the Linux kernel, the following vulnerability has been resolved: ntb_hw_switctec: Fix shift-out-of-bounds in swit
CVE-2023-53036	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix call trace warning and hang wh
CVE-2023-53037	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Bad drive in topology results kerne
CVE-2023-53038	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Check kzalloc() in lpfc_sli4_cgn_param
CVE-2023-53039	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: ipc: Fix potential use-after-fre
CVE-2023-53041	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Perform lockless command comple
CVE-2023-53042	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Do not set DRR on pipe Comm
CVE-2023-53044	In the Linux kernel, the following vulnerability has been resolved: dm stats: check for and propagate alloc_percpu fa
CVE-2023-53046	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix race condition in hci_cmd_sync_c
CVE-2023-53048	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tcpm: fix warning when handle discov
CVE-2023-53049	In the Linux kernel, the following vulnerability has been resolved: usb: ucsi: Fix NULL pointer deref in ucsi_connec
CVE-2023-53050	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Fix memory leak in margining Mem
CVE-2023-53051	In the Linux kernel, the following vulnerability has been resolved: dm crypt: add cond_resched() to dmccrypt_write()
CVE-2023-53052	In the Linux kernel, the following vulnerability has been resolved: cifs: fix use-after-free bug in refresh_cache_work
CVE-2023-53053	In the Linux kernel, the following vulnerability has been resolved: erspan: do not use skb_mac_header() in ndo_start
CVE-2023-53056	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Synchronize the IOCB count to be
CVE-2023-53057	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: HCI: Fix global-out-of-bounds To loop
CVE-2023-53058	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: E-Switch, Fix an Oops in error handling
CVE-2023-53060	In the Linux kernel, the following vulnerability has been resolved: igb: revert rtnl_lock() that causes deadlock The c
CVE-2023-53062	In the Linux kernel, the following vulnerability has been resolved: net: usb: smsc95xx: Limit packet length to skb->l
CVE-2023-53063	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-53064	In the Linux kernel, the following vulnerability has been resolved: iavf: fix hang on reboot with ice When a system v
CVE-2023-53065	In the Linux kernel, the following vulnerability has been resolved: perf/core: Fix perf_output_begin parameter is inc
CVE-2023-53066	In the Linux kernel, the following vulnerability has been resolved: qed/qed_sriov: guard against NULL derefs from c

CVE-2023-53067	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Only call get_timer_irq() once in con
CVE-2023-53068	In the Linux kernel, the following vulnerability has been resolved: net: usb: lan78xx: Limit packet length to skb->len
CVE-2023-53070	In the Linux kernel, the following vulnerability has been resolved: ACPI: PPTT: Fix to avoid sleep in the atomic con
CVE-2023-53071	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: do not run mt76_unregister_device()
CVE-2023-53072	In the Linux kernel, the following vulnerability has been resolved: mptcp: use the workqueue to destroy unaccepted
CVE-2023-53073	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd/core: Always clear status for idx TH
CVE-2023-53074	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix ttm_bo calltrace warning in psp
CVE-2023-53075	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix invalid address access in lookup_rec()
CVE-2023-53076	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2023-53077	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix shift-out-of-bounds in Calcu
CVE-2023-53078	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_dh_alua: Fix memleak for 'qdata' in alua
CVE-2023-53079	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix steering rules cleanup vport's mc, u
CVE-2023-53080	In the Linux kernel, the following vulnerability has been resolved: xsk: Add missing overflow check in xdp_umem_
CVE-2023-53084	In the Linux kernel, the following vulnerability has been resolved: drm/shmem-helper: Remove another errant put in
CVE-2023-53085	In the Linux kernel, the following vulnerability has been resolved: drm/edid: fix info leak when failing to get panel i
CVE-2023-53086	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: connac: do not check WED status for
CVE-2023-53088	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix UaF in listener shutdown As reported b
CVE-2023-53089	In the Linux kernel, the following vulnerability has been resolved: ext4: fix task hung in ext4_xattr_delete_inode Sys
CVE-2023-53090	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix an illegal memory access In the
CVE-2023-53091	In the Linux kernel, the following vulnerability has been resolved: ext4: update s_journal_inum if it changes after jo
CVE-2023-53093	In the Linux kernel, the following vulnerability has been resolved: tracing: Do not let histogram values have some m
CVE-2023-53095	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: Fix a NULL pointer dereference The LR
CVE-2023-53097	In the Linux kernel, the following vulnerability has been resolved: powerpc/iommu: fix memory leak with using deb
CVE-2023-53098	In the Linux kernel, the following vulnerability has been resolved: media: rc: gpio-ir-recv: add remove function In ca
CVE-2023-53100	In the Linux kernel, the following vulnerability has been resolved: ext4: fix WARNING in ext4_update_inline_data
CVE-2023-53101	In the Linux kernel, the following vulnerability has been resolved: ext4: zero i_disksize when initializing the bootloa
CVE-2023-53103	In the Linux kernel, the following vulnerability has been resolved: bonding: restore bond's IFF_SLAVE flag if a non
CVE-2023-53105	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix cleanup null-ptr deref on encaps lo
CVE-2023-53108	In the Linux kernel, the following vulnerability has been resolved: net/iucv: Fix size of interrupt data iucv_irq_data n
CVE-2023-53102	In the Linux kernel, the following vulnerability has been resolved: ice: xsk: disable txq irq before flushing hw ice_qp
CVE-2023-53109	In the Linux kernel, the following vulnerability has been resolved: net: tunnels: annotate lockless accesses to dev->n
CVE-2023-53112	In the Linux kernel, the following vulnerability has been resolved: drm/i915/sseu: fix max_subslices array-index-out
CVE-2023-53113	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: fix NULL-ptr deref in offchan che
CVE-2023-53111	In the Linux kernel, the following vulnerability has been resolved: loop: Fix use-after-free issues do_req_filebacked
CVE-2023-53110	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix NULL sndbuf_desc in smc_cdc_tx_h
CVE-2023-53114	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix kernel crash during reboot when adapter
CVE-2023-53115	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix memory leaks in mpi3mr_init_
CVE-2023-53117	In the Linux kernel, the following vulnerability has been resolved: fs: prevent out-of-bounds array speculation when
CVE-2023-53118	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix a procfs host directory removal reg
CVE-2023-53116	In the Linux kernel, the following vulnerability has been resolved: nvmet: avoid potential UAF in nvmet_req_compl

CVE-2023-53120	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix config page DMA memory leak
CVE-2023-53121	In the Linux kernel, the following vulnerability has been resolved: tcp: tcp_make_synack() can be called from process context
CVE-2023-53124	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix NULL pointer access in mpt3sas_scsi_dev
CVE-2023-53123	In the Linux kernel, the following vulnerability has been resolved: PCI: s390: Fix use-after-free of PCI resources with pci_dev
CVE-2023-53125	In the Linux kernel, the following vulnerability has been resolved: net: usb: smsc75xx: Limit packet length to skb->len
CVE-2023-53126	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix sas_hba.phy memory leak in mpi3mr_sas_hba
CVE-2023-53127	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix expander node leak in mpi3mr_sas_hba
CVE-2023-53128	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix throttle_groups memory leak in mpi3mr_sas_hba
CVE-2023-53131	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix a server shutdown leak Fix a race condition
CVE-2023-53132	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix mpi3mr_hba_port memory leak in mpi3mr_sas_hba
CVE-2023-53133	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix an infinite loop error when len is 0
CVE-2023-53134	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Avoid order-5 memory allocation for TPA
CVE-2023-53137	In the Linux kernel, the following vulnerability has been resolved: ext4: Fix possible corruption when moving a directory
CVE-2023-53140	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Remove the /proc/scsi/\${proc_name} directory
CVE-2023-53142	In the Linux kernel, the following vulnerability has been resolved: ice: copy last block omitted in ice_get_module_eeprom
CVE-2023-53143	In the Linux kernel, the following vulnerability has been resolved: ext4: fix another off-by-one fsmap error on 1k block
CVE-2023-53145	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: bttdio: fix use after free bug in bttdio_remove
CVE-2023-53146	In the Linux kernel, the following vulnerability has been resolved: media: dw2102: Fix null-ptr-deref in dw2102_i2c_driver
CVE-2023-5344	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1969.
CVE-2023-5441	NULL Pointer Dereference in GitHub repository vim/vim prior to 20d161ace307e28690229b68584f2d84556f8960.
CVE-2023-5455	A Cross-site request forgery vulnerability exists in ipa/session/login_password in all supported versions of IPA. This vulnerability
CVE-2023-5528	A security issue was discovered in Kubernetes where a user that can create pods and persistent volumes on Windows can create
CVE-2023-5535	Use After Free in GitHub repository vim/vim prior to v9.0.2010.
CVE-2023-5574	A use-after-free flaw was found in xorg-x11-server-Xvfb. This issue occurs in Xvfb with a very specific and legacy configuration
CVE-2023-5678	Issue summary: Generating excessively long X9.42 DH keys or checking excessively long X9.42 DH keys or parameters
CVE-2023-5752	When installing a package from a Mercurial VCS URL (ie "pip install hg+...") with pip prior to v23.3, the specified URL
CVE-2023-5868	A memory disclosure vulnerability was found in PostgreSQL that allows remote users to access sensitive information
CVE-2023-5869	A flaw was found in PostgreSQL that allows authenticated database users to execute arbitrary code through missing
CVE-2023-5870	A flaw was found in PostgreSQL involving the pg_cancel_backend role that signals background workers, including
CVE-2023-5981	A vulnerability was found that the response times to malformed ciphertexts in RSA-PSK ClientKeyExchange differ
CVE-2023-6004	A flaw was found in libssh. By utilizing the ProxyCommand or ProxyJump feature, users can exploit unchecked host
CVE-2023-6040	An out-of-bounds access vulnerability involving netfilter was reported and fixed as: f1082dd31fe4 (netfilter: nf_tables
CVE-2023-6129	Issue summary: The POLY1305 MAC (message authentication code) implementation contains a bug that might corrupt
CVE-2023-6135	Multiple NSS NIST curves were susceptible to a side-channel attack known as "Minerva". This attack could potentially
CVE-2023-6228	An issue was found in the tiffcp utility distributed by the libtiff package where a crafted TIFF file on processing may
CVE-2023-6237	Issue summary: Checking excessively long invalid RSA public keys may take a long time. Impact summary: Applications
CVE-2023-6238	A buffer overflow vulnerability was found in the NVM Express (NVMe) driver in the Linux kernel. Only privileged
CVE-2023-6277	An out-of-memory flaw was found in libtiff. Passing a crafted tiff file to TIFFOpen() API may allow a remote attacker
CVE-2023-6378	A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount
CVE-2023-6597	An issue was found in the CPython `tempfile.TemporaryDirectory` class affecting versions 3.12.1, 3.11.7, 3.10.13, 3.9.13, 3.8.13

CVE-2023-6601	A flaw was found in FFmpeg's HLS demuxer. This vulnerability allows bypassing unsafe file extension checks and t
CVE-2023-6602	A flaw was found in FFmpeg's TTY Demuxer. This vulnerability allows possible data exfiltration via improper pars
CVE-2023-6603	A flaw was found in FFmpeg's HLS playlist parsing. This vulnerability allows a denial of service via a maliciously c
CVE-2023-6604	A flaw was found in FFmpeg. This vulnerability allows unexpected additional CPU load and storage consumption, p
CVE-2023-6605	A flaw was found in FFmpeg's DASH playlist support. This vulnerability allows arbitrary HTTP GET requests to be
CVE-2023-6610	An out-of-bounds read vulnerability was found in smb2_dump_detail in fs/smb/client/smb2ops.c in the Linux Kerne
CVE-2023-6918	A flaw was found in the libssh implements abstract layer for message digest (MD) operations implemented by differ
CVE-2023-7008	A vulnerability was found in systemd-resolved. This issue may allow systemd-resolved to accept records of DNSSE
CVE-2023-7042	A null pointer dereference vulnerability was found in ath10k_wmi_tlv_op_pull_mgmt_tx_compl_ev() in drivers/net/
CVE-2023-7104	A vulnerability was found in SQLite SQLite3 up to 3.43.0 and classified as critical. This issue affects the function se
CVE-2023-7216	A path traversal vulnerability was found in the CPIO utility. This issue could allow a remote unauthenticated attacker
CVE-2024-0132	NVIDIA Container Toolkit 1.16.1 or earlier contains a Time-of-check Time-of-Use (TOCTOU) vulnerability when t
CVE-2024-0133	NVIDIA Container Toolkit 1.16.1 or earlier contains a vulnerability in the default mode of operation allowing a spec
CVE-2024-0232	A heap use-after-free issue has been identified in SQLite in the jsonParseAddNodeArray() function in sqlite3.c. This
CVE-2024-0243	With the following crawler configuration: ``python from bs4 import BeautifulSoup as Soup url = "https://example.c
CVE-2024-0397	A defect was discovered in the Python "ssl" module where there is a memory race condition with the ssl.SSLContext
CVE-2024-0450	An issue was found in the CPython `zipfile` module affecting versions 3.12.1, 3.11.7, 3.10.13, 3.9.18, and 3.8.18 and
CVE-2024-0553	A vulnerability was found in GnuTLS. The response times to malformed ciphertexts in RSA-PSK ClientKeyExchang
CVE-2024-0564	A flaw was found in the Linux kernel's memory deduplication mechanism. The max page sharing of Kernel Samepag
CVE-2024-0562	A use-after-free flaw was found in the Linux Kernel. When a disk is removed, bdi_unregister is called to stop further
CVE-2024-0567	A vulnerability was found in GnuTLS, where a cockpit (which uses gnuTLS) rejects a certificate chain with distrib
CVE-2024-0607	A flaw was found in the Netfilter subsystem in the Linux kernel. The issue is in the nft_byteorder_eval() function, w
CVE-2024-0639	A denial of service vulnerability due to a deadlock was found in sctp_auto_asconf_init in net/sctp/socket.c in the Lin
CVE-2024-0641	A denial of service vulnerability was found in tipc_crypto_key_revoke in net/tipc/crypto.c in the Linux kernel's TIP
CVE-2024-0727	Issue summary: Processing a maliciously formatted PKCS12 file may lead OpenSSL to crash leading to a potential I
CVE-2024-0793	A flaw was found in kube-controller-manager. This issue occurs when the initial application of a HPA config YAM
CVE-2024-0985	Late privilege drop in REFRESH MATERIALIZED VIEW CONCURRENTLY in PostgreSQL allows an object cre
CVE-2024-10041	A vulnerability was found in PAM. The secret information is stored in memory, where the attacker can trigger the vi
CVE-2024-10096	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-10220	The Kubernetes kubelet component allows arbitrary command execution via specially crafted gitRepo volumes. This
CVE-2024-10524	Applications that use Wget to access a remote resource using shorthand URLs and pass arbitrary user credentials in t
CVE-2024-10940	A vulnerability in langchain-core versions >=0.1.17,<0.1.53, >=0.2.0,<0.2.43, and >=0.3.0,<0.3.15 allows unauthori
CVE-2024-10963	A flaw was found in pam_access, where certain rules in its configuration file are mistakenly treated as hostnames. Th
CVE-2024-10976	Incomplete tracking in PostgreSQL of tables with row security allows a reused query to view or change different ro
CVE-2024-10977	Client use of server error message in PostgreSQL allows a server not trusted under current SSL or GSS settings to fu
CVE-2024-10978	Incorrect privilege assignment in PostgreSQL allows a less-privileged application user to view or change different r
CVE-2024-10979	Incorrect control of environment variables in PostgreSQL PL/Perl allows an unprivileged database user to change se
CVE-2024-11041	vllm-project vllm version v0.6.2 contains a vulnerability in the MessageQueue.dequeue() API function. The function
CVE-2024-11053	When asked to both use a `.netrc` file for credentials and to follow HTTP redirects, curl could leak the password use
CVE-2024-11168	The urllib.parse.urlsplit() and urlparse() functions improperly validated bracketed hosts ('[]'), allowing hosts that we

CVE-2024-11187	It is possible to construct a zone such that some queries to it will generate responses containing numerous records in
CVE-2024-11392	Hugging Face Transformers MobileViTV2 Deserialization of Untrusted Data Remote Code Execution Vulnerability
CVE-2024-11393	Hugging Face Transformers MaskFormer Model Deserialization of Untrusted Data Remote Code Execution Vulnerability
CVE-2024-11394	Hugging Face Transformers Trax Model Deserialization of Untrusted Data Remote Code Execution Vulnerability. T
CVE-2024-11403	There exists an out of bounds read/write in LibJXL versions prior to commit 9cc451b91b74ba470fd72bd48c121e9f3
CVE-2024-11498	There exists a stack buffer overflow in libjxl. A specifically-crafted file can cause the JPEG XL decoder to use large
CVE-2024-1151	A vulnerability was reported in the Open vSwitch sub-component in the Linux Kernel. The flaw occurs when a recur
CVE-2024-12084	A heap-based buffer overflow flaw was found in the rsync daemon. This issue is due to improper handling of attacke
CVE-2024-12085	A flaw was found in rsync which could be triggered when rsync compares file checksums. This flaw allows an attack
CVE-2024-12086	A flaw was found in rsync. It could allow a server to enumerate the contents of an arbitrary file from the client's mac
CVE-2024-12087	A path traversal vulnerability exists in rsync. It stems from behavior enabled by the `--inc-recursive` option, a default
CVE-2024-12088	A flaw was found in rsync. When using the `--safe-links` option, the rsync client fails to properly verify if a symboli
CVE-2024-12133	A flaw in libtasn1 causes inefficient handling of specific certificate data. When processing a large number of elemen
CVE-2024-12243	A flaw was found in GnuTLS, which relies on libtasn1 for ASN.1 data processing. Due to an inefficient algorithm in
CVE-2024-12539	An issue was discovered where improper authorization controls affected certain queries that could allow a malicious
CVE-2024-12254	Starting in Python 3.12.0, the <code>asyncio.SelectorSocketTransport.writelines()</code> method would not "pause" writing and
CVE-2024-12705	Clients using DNS-over-HTTPS (DoH) can exhaust a DNS resolver's CPU and/or memory by flooding it with crafted
CVE-2024-12718	Allows modifying some file metadata (e.g. last modified) with <code>filter="data"</code> or file permissions (<code>chmod</code>) with <code>filter="</code>
CVE-2024-12720	A Regular Expression Denial of Service (ReDoS) vulnerability was identified in the huggingface/transformers library
CVE-2024-12747	A flaw was found in rsync. This vulnerability arises from a race condition during rsync's handling of symbolic links.
CVE-2024-12745	A SQL injection in the Amazon Redshift Python Connector v2.1.4 allows a user to gain escalated privileges via the g
CVE-2024-12797	Issue summary: Clients using RFC7250 Raw Public Keys (RPKs) to authenticate a server may fail to notice that the
CVE-2024-12798	ACE vulnerability in JaninoEventEvaluator by QOS.CH logback-core upto including version 0.1 to 1.3.14 and 1.4.0
CVE-2024-12801	Server-Side Request Forgery (SSRF) in SaxEventRecorder by QOS.CH logback version 0.1 to 1.3.14 and 1.4.0 to 1.
CVE-2024-13009	In Eclipse Jetty versions 9.4.0 to 9.4.56 a buffer can be incorrectly released when confronted with a gzip error when
CVE-2024-13176	Issue summary: A timing side-channel which could potentially allow recovering the private key exists in the ECDSA
CVE-2024-1455	A vulnerability in the langchain-ai/langchain repository allows for a Billion Laughs Attack, a type of XML External
CVE-2024-1597	pgjdbc, the PostgreSQL JDBC Driver, allows attacker to inject SQL if using <code>PreferQueryMode=SIMPLE</code> . Note this
CVE-2024-1737	Resolver caches and authoritative zone databases that hold significant numbers of RRs for the same hostname (of an
CVE-2024-1975	If a server hosts a zone containing a "KEY" Resource Record, or a resolver DNSSEC-validates a "KEY" Resource R
CVE-2024-20696	Windows libarchive Remote Code Execution Vulnerability
CVE-2024-21011	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21012	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21068	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21094	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21131	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21138	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21140	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21145	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2024-21147	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac

CVE-2024-21208	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle
CVE-2024-21210	Vulnerability in Oracle Java SE (component: Hotspot). Supported versions that are affected are Oracle Java SE: 8u4
CVE-2024-21217	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle
CVE-2024-21235	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle
CVE-2024-21272	Vulnerability in the MySQL Connectors product of Oracle MySQL (component: Connector/Python). Supported vers
CVE-2024-21520	Versions of the package djangoestframework before 3.15.2 are vulnerable to Cross-site Scripting (XSS) via the bre
CVE-2024-21538	Versions of the package cross-spawn before 6.0.6, from 7.0.0 and before 7.0.5 are vulnerable to Regular Expression
CVE-2024-21626	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. In runc 1.1.11 a
CVE-2024-21634	Amazon Ion is a Java implementation of the Ion data notation. Prior to version 1.10.5, a potential denial-of-service is
CVE-2024-21823	Hardware logic with insecure de-synchronization in Intel(R) DSA and Intel(R) IAA for some Intel(R) 4th or 5th gen
CVE-2024-2201	A cross-privilege Spectre v2 vulnerability allows attackers to bypass all deployed mitigations, including the recent F
CVE-2024-22018	A vulnerability has been identified in Node.js, affecting users of the experimental permission model when the --allow
CVE-2024-22020	A security flaw in Node.js allows a bypass of network import restrictions. By embedding non-network imports in da
CVE-2024-22099	NULL Pointer Dereference vulnerability in Linux Linux kernel kernel on Linux, x86, ARM (net, bluetooth modules)
CVE-2024-22243	Applications that use UriComponentsBuilder to parse an externally provided URL (e.g. through a query parameter) a
CVE-2024-22259	Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through
CVE-2024-22262	Applications that use UriComponentsBuilder to parse an externally provided URL (e.g. through a query parameter) a
CVE-2024-2236	A timing-based side-channel flaw was found in libgcrypt's RSA implementation. This issue may allow a remote attac
CVE-2024-22513	djangoestframework-simplejwt version 5.3.1 and before is vulnerable to information disclosure. A user can access v
CVE-2024-22365	linux-pam (aka Linux PAM) before 1.6.0 allows attackers to cause a denial of service (blocked login process) via m
CVE-2024-22667	Vim before 9.0.2142 has a stack-based buffer overflow because did_set_langmap in map.c calls sprintf to write to th
CVE-2024-23196	A race condition was found in the Linux kernel's sound/hda device driver in snd_hdac_regmap_sync() function. This
CVE-2024-23337	jq is a command-line JSON processor. In versions up to and including 1.7.1, an integer overflow arises when assigni
CVE-2024-23848	In the Linux kernel through 6.7.1, there is a use-after-free in cec_queue_msg_fh, related to drivers/media/cec/core/ce
CVE-2024-23944	Information disclosure in persistent watchers handling in Apache ZooKeeper due to missing ACL check. It allows an
CVE-2024-23945	Signing cookies is an application security feature that adds a digital signature to cookie data to verify its authenticity
CVE-2024-23953	Use of Arrays.equals() in LlapSignerImpl in Apache Hive to compare message signatures allows attacker to forge a
CVE-2024-2398	When an application tells libcurl it wants to allow HTTP/2 server push, and the amount of received headers for the p
CVE-2024-2408	The openssl_private_decrypt function in PHP, when using PKCS1 padding (OPENSSL_PKCS1_PADDING, which
CVE-2024-24557	Moby is an open-source project created by Docker to enable software containerization. The classic builder cache sys
CVE-2024-24783	Verifying a certificate chain which contains a certificate with an unknown public key algorithm will cause Certificat
CVE-2024-24784	The ParseAddressList function incorrectly handles comments (text within parentheses) within display names. Since
CVE-2024-24785	If errors returned from MarshalJSON methods contain user controlled data, they may be used to break the contextual
CVE-2024-24786	The protojson.Unmarshal function can enter an infinite loop when unmarshaling certain forms of invalid JSON. This
CVE-2024-24788	A malformed DNS message in response to a query can cause the Lookup functions to get stuck in an infinite loop.
CVE-2024-24789	The archive/zip package's handling of certain types of invalid zip files differs from the behavior of most zip impleme
CVE-2024-24790	The various Is methods (IsPrivate, IsLoopback, etc) did not work as expected for IPv4-mapped IPv6 addresses, retur
CVE-2024-24791	The net/http HTTP/1.1 client mishandled the case where a server responds to a request with an "Expect: 100-contin
CVE-2024-24795	HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious re
CVE-2024-24855	A race condition was found in the Linux kernel's scsi device driver in lpfc_unregister_fcf_rescan() function. This can

CVE-2024-24857	A race condition was found in the Linux kernel's net/bluetooth device driver in conn_info_{min,max}_age_set() function.
CVE-2024-24858	A race condition was found in the Linux kernel's net/bluetooth in {conn,adv}_{min,max}_interval_set() function. This can lead to a denial of service.
CVE-2024-24859	A race condition was found in the Linux kernel's net/bluetooth in sniff_{min,max}_interval_set() function. This can lead to a denial of service.
CVE-2024-25062	An issue was discovered in libxml2 before 2.11.7 and 2.12.x before 2.12.5. When using the XML Reader interface with a maliciously crafted XML document, a buffer overflow can occur, leading to a denial of service or potential code execution.
CVE-2024-2511	Issue summary: Some non-default TLS server configurations can cause unbounded memory growth when processing large requests.
CVE-2024-25126	Rack is a modular Ruby web server interface. Carefully crafted content type headers can cause Rack's media type parser to enter an infinite loop, leading to a denial of service.
CVE-2024-25260	elfutils v0.189 was discovered to contain a NULL pointer dereference via the handle_verdef() function at readelf.c.
CVE-2024-25638	dnsjava is an implementation of DNS in Java. Records in DNS replies are not checked for their relevance to the query, which can lead to a denial of service.
CVE-2024-25739	create_empty_lvol in drivers/mtd/ubi/vtbl.c in the Linux kernel through 6.7.4 can attempt to allocate zero bytes, and this can lead to a denial of service.
CVE-2024-25710	Loop with Unreachable Exit Condition ('Infinite Loop') vulnerability in Apache Commons Compress. This issue affects versions 1.25.0 through 1.25.1.
CVE-2024-25740	A memory leak flaw was found in the UBI driver in drivers/mtd/ubi/attach.c in the Linux kernel through 6.7.4 for UBI volumes.
CVE-2024-25741	printer_write in drivers/usb/gadget/function/f_printer.c in the Linux kernel through 6.7.4 does not properly call usb_ep_wake_up() when the endpoint is stalled, which can lead to a denial of service.
CVE-2024-25744	In the Linux kernel before 6.6.7, an untrusted VMM can trigger int80 syscall handling at any given point. This is related to the KVM module.
CVE-2024-26130	cryptography is a package designed to expose cryptographic primitives and recipes to Python developers. Starting in version 40.0.0, it includes a vulnerability that allows for a denial of service.
CVE-2024-26141	Rack is a modular Ruby web server interface. Carefully crafted Range headers can cause a server to respond with an infinite loop, leading to a denial of service.
CVE-2024-26146	Rack is a modular Ruby web server interface. Carefully crafted headers can cause header parsing in Rack to take an infinite loop, leading to a denial of service.
CVE-2024-26308	Allocation of Resources Without Limits or Throttling vulnerability in Apache Commons Compress. This issue affects versions 1.25.0 through 1.25.1.
CVE-2024-26458	Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in /krb5/src/lib/krb5/pmap_rmt.c.
CVE-2024-26461	Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/lib/gssapi/krb5/k5sealv3.c.
CVE-2024-26462	Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in /krb5/src/kdc/ndr.c.
CVE-2024-26584	In the Linux kernel, the following vulnerability has been resolved: net: tls: handle backlogging of crypto requests. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26583	In the Linux kernel, the following vulnerability has been resolved: tls: fix race between async notify and socket close. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26585	In the Linux kernel, the following vulnerability has been resolved: tls: fix race between tx work scheduling and socket close. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26586	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcaml: Fix stack corruption. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26589	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject variable offset alu on PTR_TO_FLOW. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26591	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix re-attachment branch in bpf_tracing_prog. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26595	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcaml: Fix NULL pointer dereference. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26598	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: vgic-its: Avoid potential UAF in LPI. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26600	In the Linux kernel, the following vulnerability has been resolved: phy: ti: phy-omap-usb2: Fix NULL pointer dereference. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26605	In the Linux kernel, the following vulnerability has been resolved: PCI/ASPM: Fix deadlock when enabling ASPM. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26611	In the Linux kernel, the following vulnerability has been resolved: xsk: fix usage of multi-buffer BPF helpers for ZC. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26607	In the Linux kernel, the following vulnerability has been resolved: drm/bridge: sii902x: Fix probing race issue. A null pointer dereference can occur during probing.
CVE-2024-26614	In the Linux kernel, the following vulnerability has been resolved: tcp: make sure init the accept_queue's spinlocks. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26621	In the Linux kernel, the following vulnerability has been resolved: mm: huge_memory: don't force huge page alignment. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26633	In the Linux kernel, the following vulnerability has been resolved: ip6_tunnel: fix NEXTHDR_FRAGMENT handling. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26638	In the Linux kernel, the following vulnerability has been resolved: nbd: always initialize struct msghdr completely. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26640	In the Linux kernel, the following vulnerability has been resolved: tcp: add sanity checks to rx zerocopy TCP rx zero-copy. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26641	In the Linux kernel, the following vulnerability has been resolved: ip6_tunnel: make sure to pull inner header in __ip6_tunnel_rcv. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26645	In the Linux kernel, the following vulnerability has been resolved: tracing: Ensure visibility when inserting an element. This affects versions 6.6.0 through 6.6.7.
CVE-2024-26646	In the Linux kernel, the following vulnerability has been resolved: thermal: intel: hfi: Add syscore callbacks for syscore. This affects versions 6.6.0 through 6.6.7.

CVE-2024-26647	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix late derefrence 'dsc' check i
CVE-2024-26648	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix variable deferencing before
CVE-2024-26649	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix the null pointer when load rlc fi
CVE-2024-26656	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix use-after-free bug The bug can b
CVE-2024-26660	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Implement bounds check for str
CVE-2024-26658	In the Linux kernel, the following vulnerability has been resolved: bcachefs: grab s_umount only if snapshotting Wh
CVE-2024-26661	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL test for 'timing gen
CVE-2024-26662	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix 'panel_cntl' could be null in
CVE-2024-26663	In the Linux kernel, the following vulnerability has been resolved: tipc: Check the bearer type before calling tipc_ud
CVE-2024-26665	In the Linux kernel, the following vulnerability has been resolved: tunnels: fix out of bounds access when building I
CVE-2024-26668	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_limit: reject configurations that caus
CVE-2024-26669	In the Linux kernel, the following vulnerability has been resolved: net/sched: flower: Fix chain template offload Wh
CVE-2024-26672	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix variable 'mca_funcs' dereferenc
CVE-2024-26677	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix delayed ACKs to not set the reference s
CVE-2024-26675	In the Linux kernel, the following vulnerability has been resolved: ppp_async: limit MRU to 64K syzbot triggered a
CVE-2024-26680	In the Linux kernel, the following vulnerability has been resolved: net: atlantic: Fix DMA mapping for PTP hwts rin
CVE-2024-26681	In the Linux kernel, the following vulnerability has been resolved: netdevsim: avoid potential loop in nsim_dev_trap
CVE-2024-26686	In the Linux kernel, the following vulnerability has been resolved: fs/proc: do_task_stat: use sig->stats_lock to gathe
CVE-2024-26687	In the Linux kernel, the following vulnerability has been resolved: xen/events: close evtchn after mapping cleanup sl
CVE-2024-26689	In the Linux kernel, the following vulnerability has been resolved: ceph: prevent use-after-free in encode_cap_msg()
CVE-2024-26688	In the Linux kernel, the following vulnerability has been resolved: fs,hugetlb: fix NULL pointer dereference in huge
CVE-2024-26691	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Fix circular locking dependency Th
CVE-2024-26695	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp - Fix null pointer dereference in __sev
CVE-2024-26698	In the Linux kernel, the following vulnerability has been resolved: hv_netvsc: Fix race condition between netvsc_pro
CVE-2024-26699	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix array-index-out-of-bounds
CVE-2024-26700	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix MST Null Ptr for RV The c
CVE-2024-26704	In the Linux kernel, the following vulnerability has been resolved: ext4: fix double-free of blocks due to wrong exten
CVE-2024-26713	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-26714	In the Linux kernel, the following vulnerability has been resolved: interconnect: qcom: sc8180x: Mark CO0 BCM ke
CVE-2024-26715	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: Fix NULL pointer dereference
CVE-2024-26718	In the Linux kernel, the following vulnerability has been resolved: dm-crypt, dm-verity: disable tasklets Tasklets hav
CVE-2024-26717	In the Linux kernel, the following vulnerability has been resolved: HID: i2c-hid-of: fix NULL-deref on failed power
CVE-2024-26719	In the Linux kernel, the following vulnerability has been resolved: nouveau: offload fence uevents work to workque
CVE-2024-26720	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-26721	In the Linux kernel, the following vulnerability has been resolved: drm/i915/dsc: Fix the macro that calculates DSCC
CVE-2024-26726	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't drop extent_map for free space inode
CVE-2024-26733	In the Linux kernel, the following vulnerability has been resolved: arp: Prevent overflow in arp_req_get(). syzkaller
CVE-2024-26735	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: fix possible use-after-free and null-ptr-de
CVE-2024-26740	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_mirred: use the backlog for mirred
CVE-2024-26747	In the Linux kernel, the following vulnerability has been resolved: usb: roles: fix NULL pointer issue when put mod

CVE-2024-26742	In the Linux kernel, the following vulnerability has been resolved: scsi: smartpqi: Fix disable_managed_interrupts C
CVE-2024-26751	In the Linux kernel, the following vulnerability has been resolved: ARM: ep93xx: Add terminator to gpiod_lookup_
CVE-2024-26756	In the Linux kernel, the following vulnerability has been resolved: md: Don't register sync_thread for reshape directl
CVE-2024-26757	In the Linux kernel, the following vulnerability has been resolved: md: Don't ignore read-only array in md_check_re
CVE-2024-26758	In the Linux kernel, the following vulnerability has been resolved: md: Don't ignore suspended array in md_check_r
CVE-2024-26759	In the Linux kernel, the following vulnerability has been resolved: mm/swap: fix race when skipping swapcache Wh
CVE-2024-26763	In the Linux kernel, the following vulnerability has been resolved: dm-crypt: don't modify the data when using auth
CVE-2024-26767	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fixed integer types and null che
CVE-2024-26769	In the Linux kernel, the following vulnerability has been resolved: nvmet-fc: avoid deadlock on delete association pa
CVE-2024-26770	In the Linux kernel, the following vulnerability has been resolved: HID: nvidia-shield: Add missing null pointer che
CVE-2024-26771	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: edma: Add some null pointer che
CVE-2024-26772	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid allocating blocks from corrupted grou
CVE-2024-26773	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid allocating blocks from corrupted grou
CVE-2024-26775	In the Linux kernel, the following vulnerability has been resolved: aoe: avoid potential deadlock at set_capacity Mov
CVE-2024-26777	In the Linux kernel, the following vulnerability has been resolved: fbdev: sis: Error out if pixclock equals zero The u
CVE-2024-26778	In the Linux kernel, the following vulnerability has been resolved: fbdev: savage: Error out if pixclock equals zero T
CVE-2024-26782	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix double-free on socket dismantle when
CVE-2024-26800	In the Linux kernel, the following vulnerability has been resolved: tls: fix use-after-free on failed backlog decryption
CVE-2024-26801	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Avoid potential use-after-free in hci_e
CVE-2024-26802	In the Linux kernel, the following vulnerability has been resolved: stmmac: Clear variable when destroying workque
CVE-2024-26803	In the Linux kernel, the following vulnerability has been resolved: net: veth: clear GRO when clearing XDP even wh
CVE-2024-26804	In the Linux kernel, the following vulnerability has been resolved: net: ip_tunnel: prevent perpetual headroom growt
CVE-2024-26807	In the Linux kernel, the following vulnerability has been resolved: Both cadence-quadspi ->runtime_suspend() and -
CVE-2024-26810	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Lock external INTx masking ops Mask c
CVE-2024-26812	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Create persistent INTx handler A vulner
CVE-2024-26816	In the Linux kernel, the following vulnerability has been resolved: x86, relocs: Ignore relocations in .notes section W
CVE-2024-26822	In the Linux kernel, the following vulnerability has been resolved: smb: client: set correct id, uid and cruid for multi
CVE-2024-26826	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix data re-injection from stale subflow W
CVE-2024-26828	In the Linux kernel, the following vulnerability has been resolved: cifs: fix underflow in parse_server_interfaces() In
CVE-2024-26830	In the Linux kernel, the following vulnerability has been resolved: i40e: Do not allow untrusted VF to remove admin
CVE-2024-26837	In the Linux kernel, the following vulnerability has been resolved: net: bridge: switchdev: Skip MDB replays of defe
CVE-2024-26838	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix KASAN issue with tasklet KA
CVE-2024-26840	In the Linux kernel, the following vulnerability has been resolved: cachefiles: fix memory leak in cachefiles_add_ca
CVE-2024-26842	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Fix shift issue in ufshcd_clear_cm
CVE-2024-26843	In the Linux kernel, the following vulnerability has been resolved: efi: runtime: Fix potential overflow of soft-reserv
CVE-2024-26844	In the Linux kernel, the following vulnerability has been resolved: block: Fix WARNING in _copy_from_iter Syzka
CVE-2024-26845	In the Linux kernel, the following vulnerability has been resolved: scsi: target: core: Add TMF to tmr_list handling A
CVE-2024-26846	In the Linux kernel, the following vulnerability has been resolved: nvme-fc: do not wait in vain when unloading mod
CVE-2024-26851	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_conntrack_h323: Add protection for
CVE-2024-26852	In the Linux kernel, the following vulnerability has been resolved: net/ipv6: avoid possible UAF in ip6_route_mpath

CVE-2024-26853	In the Linux kernel, the following vulnerability has been resolved: igc: avoid returning frame twice in XDP_REDIRECT
CVE-2024-26855	In the Linux kernel, the following vulnerability has been resolved: net: ice: Fix potential NULL pointer dereference
CVE-2024-26857	In the Linux kernel, the following vulnerability has been resolved: geneve: make sure to pull inner header in geneve
CVE-2024-26859	In the Linux kernel, the following vulnerability has been resolved: net/bnx2x: Prevent access to a freed page in page
CVE-2024-26866	In the Linux kernel, the following vulnerability has been resolved: spi: lpspi: Avoid potential use-after-free in probe
CVE-2024-26869	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to truncate meta inode pages forcibly
CVE-2024-26870	In the Linux kernel, the following vulnerability has been resolved: NFSv4.2: fix nfs4_listxattr kernel BUG at mm/us
CVE-2024-26876	In the Linux kernel, the following vulnerability has been resolved: drm/bridge: adv7511: fix crash on irq during prob
CVE-2024-26878	In the Linux kernel, the following vulnerability has been resolved: quota: Fix potential NULL pointer dereference
CVE-2024-26880	In the Linux kernel, the following vulnerability has been resolved: dm: call the resume method on internal suspend
CVE-2024-26882	In the Linux kernel, the following vulnerability has been resolved: net: ip_tunnel: make sure to pull inner header
CVE-2024-26883	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix stackmap overflow check on 32-bit arch
CVE-2024-26886	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: af_bluetooth: Fix deadlock Attempting
CVE-2024-26893	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix double free in SMC tran
CVE-2024-26894	In the Linux kernel, the following vulnerability has been resolved: ACPI: processor_idle: Fix memory leak in acpi_p
CVE-2024-26896	In the Linux kernel, the following vulnerability has been resolved: wifi: wfx: fix memory leak when starting AP Km
CVE-2024-26900	In the Linux kernel, the following vulnerability has been resolved: md: fix kmemleak of rdev->serial If kobject_add
CVE-2024-26902	In the Linux kernel, the following vulnerability has been resolved: perf: RISCv: Fix panic on pmu overflow handler
CVE-2024-26906	In the Linux kernel, the following vulnerability has been resolved: x86/mm: Disallow vsyscall page read for copy_fr
CVE-2024-26903	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: rfcomm: Fix null-ptr-deref in rfcomm
CVE-2024-26907	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix fortify source warning while ac
CVE-2024-26908	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-26915	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Reset IH OVERFLOW_CLEAR bi
CVE-2024-26921	In the Linux kernel, the following vulnerability has been resolved: inet: inet_defrag: prevent sk release while still in
CVE-2024-26916	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/amd: flush any delayed gfxoff on sus
CVE-2024-26922	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: validate the parameters of bo mapp
CVE-2024-26923	In the Linux kernel, the following vulnerability has been resolved: af_unix: Fix garbage collector racing against con
CVE-2024-26924	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: do not free live element
CVE-2024-26926	In the Linux kernel, the following vulnerability has been resolved: binder: check offset alignment in binder_get_obje
CVE-2024-26925	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: release mutex after nft_gc_sec
CVE-2024-26927	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Add some bounds checking to firmw
CVE-2024-26928	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in cifs_debug_files
CVE-2024-26929	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-26930	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix double free of the ha->vp_map
CVE-2024-26931	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix command flush on cable pull S
CVE-2024-26935	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix unremoved procfs host directory re
CVE-2024-26936	In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate request buffer size in smb2_alloc
CVE-2024-26938	In the Linux kernel, the following vulnerability has been resolved: drm/i915/bios: Tolerate devdata==NULL in intel
CVE-2024-26939	In the Linux kernel, the following vulnerability has been resolved: drm/i915/vma: Fix UAF on destroy against retire
CVE-2024-26940	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Create debugfs ttm_resource_mana

CVE-2024-26945	In the Linux kernel, the following vulnerability has been resolved: crypto: iaa - Fix nr_cpus < nr_iaa case If nr_cpus
CVE-2024-26947	In the Linux kernel, the following vulnerability has been resolved: ARM: 9359/1: flush: check if the folio is reserved
CVE-2024-26948	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add a dc_state NULL check in
CVE-2024-26953	In the Linux kernel, the following vulnerability has been resolved: net: esp: fix bad handling of pages from page_poo
CVE-2024-26954	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-out-of-bounds in smb_strndup_fr
CVE-2024-26958	In the Linux kernel, the following vulnerability has been resolved: nfs: fix UAF in direct writes In production we hav
CVE-2024-26960	In the Linux kernel, the following vulnerability has been resolved: mm: swap: fix race between free_swap_and_cach
CVE-2024-26961	In the Linux kernel, the following vulnerability has been resolved: mac802154: fix llsec key resources release in mac
CVE-2024-26962	In the Linux kernel, the following vulnerability has been resolved: dm-raid456, md/raid456: fix a deadlock for dm-ra
CVE-2024-26974	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - resolve race condition during AER re
CVE-2024-26976	In the Linux kernel, the following vulnerability has been resolved: KVM: Always flush async #PF workqueue when
CVE-2024-26980	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-out-of-bounds in smb2_allocate_
CVE-2024-26982	In the Linux kernel, the following vulnerability has been resolved: Squashfs: check the inode number is not the inval
CVE-2024-26983	In the Linux kernel, the following vulnerability has been resolved: bootconfig: use memblock_free_late to free xbc m
CVE-2024-26984	In the Linux kernel, the following vulnerability has been resolved: nouveau: fix instmem race condition around ptr st
CVE-2024-26999	In the Linux kernel, the following vulnerability has been resolved: serial/pmac_zilog: Remove flawed mitigation for
CVE-2024-27004	In the Linux kernel, the following vulnerability has been resolved: clk: Get runtime PM before walking tree during d
CVE-2024-27002	In the Linux kernel, the following vulnerability has been resolved: clk: mediatek: Do a runtime PM get on controller
CVE-2024-27005	In the Linux kernel, the following vulnerability has been resolved: interconnect: Don't access req_list while it's being
CVE-2024-27008	In the Linux kernel, the following vulnerability has been resolved: drm: nv04: Fix out of bounds access When Outpu
CVE-2024-27010	In the Linux kernel, the following vulnerability has been resolved: net/sched: Fix mirred deadlock on device recursio
CVE-2024-27011	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fix memleak in map from abo
CVE-2024-27012	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: restore set elements when dele
CVE-2024-27013	In the Linux kernel, the following vulnerability has been resolved: tun: limit printing rate when illegal packet receive
CVE-2024-27014	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Prevent deadlock while disabling aRF
CVE-2024-27017	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: walk over current view o
CVE-2024-27020	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: Fix potential data-race in __n
CVE-2024-27019	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: Fix potential data-race in __n
CVE-2024-27022	In the Linux kernel, the following vulnerability has been resolved: fork: defer linking file vma until vma is fully initi
CVE-2024-27023	In the Linux kernel, the following vulnerability has been resolved: md: Fix missing release of 'active_io' for flush sub
CVE-2024-27032	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid potential panic during recovery
CVE-2024-27035	In the Linux kernel, the following vulnerability has been resolved: f2fs: compress: fix to guarantee persisting compre
CVE-2024-27025	In the Linux kernel, the following vulnerability has been resolved: nbd: null check for nla_nest_start nla_nest_start()
CVE-2024-27041	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix NULL checks for adev->dm
CVE-2024-27042	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential out-of-bounds access
CVE-2024-27043	In the Linux kernel, the following vulnerability has been resolved: media: edia: dvbdev: fix a use-after-free In dvb_r
CVE-2024-27056	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: ensure offloading TID queue
CVE-2024-27055	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-27057	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-pcm: Workaround for crashed
CVE-2024-27062	In the Linux kernel, the following vulnerability has been resolved: nouveau: lock the client object tree. It appears the

CVE-2024-27072	In the Linux kernel, the following vulnerability has been resolved: media: usbtv: Remove useless locks in usbtv_vid
CVE-2024-27065	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: do not compare internal table
CVE-2024-27073	In the Linux kernel, the following vulnerability has been resolved: media: ttpci: fix two memleaks in budget_av_atta
CVE-2024-27075	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: avoid stack overflow warn
CVE-2024-27351	In Django 3.2 before 3.2.25, 4.2 before 4.2.11, and 5.0 before 5.0.3, the django.utils.text.Truncator.words() method
CVE-2024-27388	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: fix some memleaks in gssx_dec_option
CVE-2024-27389	In the Linux kernel, the following vulnerability has been resolved: pstore: inode: Only d_invalidate() is needed Unlo
CVE-2024-27395	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: Fix Use-After-Free in ovs_ct_e
CVE-2024-27397	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: use timestamp to check for se
CVE-2024-27398	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix use-after-free bugs caused by sco_
CVE-2024-27399	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: l2cap: fix null-ptr-deref in l2cap_chan
CVE-2024-27400	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: once more fix the call oder in amdg
CVE-2024-27403	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_flow_offload: reset dst in route objec
CVE-2024-27401	In the Linux kernel, the following vulnerability has been resolved: firewire: nosy: ensure user_length is taken into ac
CVE-2024-27407	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fixed overflow check in mi_enum_attr()
CVE-2024-27408	In the Linux kernel, the following vulnerability has been resolved: dmaengine: dw-edma: eDMA: Add sync read bef
CVE-2024-27410	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: reject iftype change with mesh ID
CVE-2024-27413	In the Linux kernel, the following vulnerability has been resolved: efi/capsule-loader: fix incorrect allocation size gc
CVE-2024-27415	In the Linux kernel, the following vulnerability has been resolved: netfilter: bridge: confirm multicast packets before
CVE-2024-27417	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix potential "struct net" leak in inet6_rtm_g
CVE-2024-27418	In the Linux kernel, the following vulnerability has been resolved: net: mctp: take ownership of skb in mctp_local_o
CVE-2024-27431	In the Linux kernel, the following vulnerability has been resolved: cpumap: Zero-initialise xdp_rxq_info struct before
CVE-2024-27434	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: don't set the MFP flag for the
CVE-2024-27435	In the Linux kernel, the following vulnerability has been resolved: nvme: fix reconnection fail due to reserved tag all
CVE-2024-27436	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Stop parsing channels bits whe
CVE-2024-27437	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Disable auto-enable of exclusive INTx I
CVE-2024-28085	wall in util-linux through 2.40, often installed with setgid tty permissions, allows escape sequences to be sent to othe
CVE-2024-28088	LangChain through 0.1.10 allows ../ directory traversal by an actor who is able to control the final part of the path pa
CVE-2024-27454	orjson.loads in orjson before 3.9.15 does not limit recursion for deeply nested JSON documents.
CVE-2024-28180	Package jose aims to provide an implementation of the Javascript Object Signing and Encryption set of standards. A
CVE-2024-28182	nghttp2 is an implementation of the Hypertext Transfer Protocol version 2 in C. The nghttp2 library prior to version
CVE-2024-28219	In _imagingcms.c in Pillow before 10.3.0, a buffer overflow exists because strcpy is used instead of strncpy.
CVE-2024-28834	A flaw was found in GnuTLS. The Minerva attack is a cryptographic vulnerability that exploits deterministic behavi
CVE-2024-28835	A flaw has been discovered in GnuTLS where an application crash can be induced when attempting to verify a speci
CVE-2024-28956	Exposure of Sensitive Information in Shared Microarchitectural Structures during Transient Execution for some Inte
CVE-2024-28863	node-tar is a Tar for Node.js. node-tar prior to version 6.2.1 has no limit on the number of sub-folders created in the
CVE-2024-29025	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high p
CVE-2024-29131	Out-of-bounds Write vulnerability in Apache Commons Configuration.This issue affects Apache Commons Configur
CVE-2024-29018	Moby is an open source container framework that is a key component of Docker Engine, Docker Desktop, and other
CVE-2024-29133	Out-of-bounds Write vulnerability in Apache Commons Configuration.This issue affects Apache Commons Configur

CVE-2024-2961	The iconv() function in the GNU C Library versions 2.39 and older may overflow the output buffer passed to it by up
CVE-2024-2965	A Denial-of-Service (DoS) vulnerability exists in the `SitemapLoader` class of the `langchain-ai/langchain` repository
CVE-2024-29869	Hive creates a credentials file to a temporary directory in the file system with permissions 644 by default when the fi
CVE-2024-29857	An issue was discovered in ECCurve.java and ECCurve.cs in Bouncy Castle Java (BC Java) before 1.78, BC Java L
CVE-2024-30171	An issue was discovered in Bouncy Castle Java TLS API and JSSE Provider before 1.78. Timing-based leakage may
CVE-2024-30172	An issue was discovered in Bouncy Castle Java Cryptography APIs before 1.78. An Ed25519 verification code infin
CVE-2024-30204	In Emacs before 29.3, LaTeX preview is enabled by default for e-mail attachments.
CVE-2024-30203	In Emacs before 29.3, Gnus treats inline MIME contents as trusted.
CVE-2024-30205	In Emacs before 29.3, Org mode considers contents of remote files to be trusted. This affects Org Mode before 9.6.2
CVE-2024-3095	A Server-Side Request Forgery (SSRF) vulnerability exists in the Web Research Retriever component of langchain-
CVE-2024-3056	A flaw was found in Podman. This issue may allow an attacker to create a specially crafted container that, when con
CVE-2024-31033	JJWT (aka Java JWT) through 0.12.5 ignores certain characters and thus a user might falsely conclude that they have
CVE-2024-31141	Files or Directories Accessible to External Parties, Improper Privilege Management vulnerability in Apache Kafka C
CVE-2024-31076	In the Linux kernel, the following vulnerability has been resolved: genirq/cpuhotplug, x86/vector: Prevent vector lea
CVE-2024-31578	FFmpeg version n6.1.1 was discovered to contain a heap use-after-free via the av_hwframe_ctx_init function.
CVE-2024-31582	FFmpeg version n6.1 was discovered to contain a heap buffer overflow vulnerability in the draw_block_rectangle fun
CVE-2024-31755	cJSON v1.7.17 was discovered to contain a segmentation violation, which can trigger through the second parameter
CVE-2024-3177	A security issue was discovered in Kubernetes where users may be able to launch containers that bypass the mounta
CVE-2024-32002	Git is a revision control system. Prior to versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4, repository
CVE-2024-32004	Git is a revision control system. Prior to versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4, an attacker
CVE-2024-32021	Git is a revision control system. Prior to versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4, when clone
CVE-2024-32020	Git is a revision control system. Prior to versions 2.45.1, 2.44.1, 2.43.4, 2.42.2, 2.41.1, 2.40.2, and 2.39.4, local clone
CVE-2024-32228	FFmpeg 7.0 is vulnerable to Buffer Overflow. There is a SEGV at libavcodec/hevcdec.c:2947:22 in hevc_frame_end
CVE-2024-32230	FFmpeg 7.0 is vulnerable to Buffer Overflow. There is a negative-size-param bug at libavcodec/mpegvideo_enc.c:12
CVE-2024-32465	Git is a revision control system. The Git project recommends to avoid working in untrusted repositories, and instead
CVE-2024-32473	Moby is an open source container framework that is a key component of Docker Engine, Docker Desktop, and other
CVE-2024-33599	nscd: Stack-based buffer overflow in netgroup cache If the Name Service Cache Daemon's (nscd) fixed size cache is
CVE-2024-33600	nscd: Null pointer crashes after notfound response If the Name Service Cache Daemon's (nscd) cache fails to add a n
CVE-2024-33601	nscd: netgroup cache may terminate daemon on memory allocation failure The Name Service Cache Daemon's (nscd)
CVE-2024-33602	nscd: netgroup cache assumes NSS callback uses in-buffer strings The Name Service Cache Daemon's (nscd) netgro
CVE-2024-33847	In the Linux kernel, the following vulnerability has been resolved: f2fs: compress: don't allow unaligned truncation o
CVE-2024-33621	In the Linux kernel, the following vulnerability has been resolved: ipvlan: Dont Use skb->sk in ipvlan_process_v{4,
CVE-2024-34027	In the Linux kernel, the following vulnerability has been resolved: f2fs: compress: fix to cover {reserve,release}_con
CVE-2024-34062	tqdm is an open source progress bar for Python and CLI. Any optional non-boolean CLI arguments (e.g. `--delim`, `--
CVE-2024-34069	Werkzeug is a comprehensive WSGI web application library. The debugger in affected versions of Werkzeug can al
CVE-2024-34155	Calling any of the Parse functions on Go source code which contains deeply nested literals can cause a panic due to s
CVE-2024-34156	Calling Decoder.Decode on a message which contains deeply nested structures can cause a panic due to stack exhaus
CVE-2024-34158	Calling Parse on a "// +build" build tag line with deeply nested expressions can cause a panic due to stack exhaustio
CVE-2024-34447	An issue was discovered in the Bouncy Castle Crypto Package For Java before BC TLS Java 1.0.19 (ships with BC J
CVE-2024-34459	An issue was discovered in xmllint (from libxml2) before 2.11.8 and 2.12.x before 2.12.7. Formatting error message

CVE-2024-34777	In the Linux kernel, the following vulnerability has been resolved: dma-mapping: benchmark: fix node id validation
CVE-2024-35176	REXML is an XML toolkit for Ruby. The REXML gem before 3.2.6 has a denial of service vulnerability when it par
CVE-2024-35195	Requests is a HTTP library. Prior to 2.32.0, when making requests through a Requests `Session`, if the first request i
CVE-2024-35235	OpenPrinting CUPS is an open source printing system for Linux and other Unix-like operating systems. In versions
CVE-2024-35247	In the Linux kernel, the following vulnerability has been resolved: fpga: region: add owner module and take its refco
CVE-2024-35255	Azure Identity Libraries and Microsoft Authentication Library Elevation of Privilege Vulnerability
CVE-2024-35365	FFmpeg version n6.1.1 has a double-free vulnerability in the fftools/ffmpeg_mux_init.c component of FFmpeg, spec
CVE-2024-35366	FFmpeg n6.1.1 is Integer Overflow. The vulnerability exists in the parse_options function of sbgdec.c within the liba
CVE-2024-35367	FFmpeg n6.1.1 has an Out-of-bounds Read via libavcodec/ppc/vp8dsp_altivec.c, static const vec_s8 h_subpel_filters
CVE-2024-35368	FFmpeg n7.0 is affected by a Double Free via the rkmpv_retrieve_frame function within libavcodec/rkmpvdec.c.
CVE-2024-3571	langchain-ai/langchain is vulnerable to path traversal due to improper limitation of a pathname to a restricted directo
CVE-2024-35789	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: check/clear fast rx for non-4addr
CVE-2024-35784	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix deadlock with fiemap and extent locking
CVE-2024-35790	In the Linux kernel, the following vulnerability has been resolved: usb: typec: altmodes/displayport: create sysfs nod
CVE-2024-35791	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Flush pages under kvm->lock to fix
CVE-2024-35799	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Prevent crash when disable stre
CVE-2024-35801	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Keep xfd_state in sync with MSR_IA32
CVE-2024-35802	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-35794	In the Linux kernel, the following vulnerability has been resolved: dm-raid: really frozen sync_thread during suspens
CVE-2024-35803	In the Linux kernel, the following vulnerability has been resolved: x86/efistub: Call mixed mode boot services on the
CVE-2024-35805	In the Linux kernel, the following vulnerability has been resolved: dm snapshot: fix lockup in dm_exception_table_e
CVE-2024-35808	In the Linux kernel, the following vulnerability has been resolved: md/dm-raid: don't call md_reap_sync_thread() di
CVE-2024-35807	In the Linux kernel, the following vulnerability has been resolved: ext4: fix corruption during on-line resize We obs
CVE-2024-35809	In the Linux kernel, the following vulnerability has been resolved: PCI/PM: Drain runtime-idle callbacks before driv
CVE-2024-35810	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Fix the lifetime of the bo cursor me
CVE-2024-35814	In the Linux kernel, the following vulnerability has been resolved: swiotlb: Fix double-allocation of slots due to brok
CVE-2024-35817	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: amdgpu_ttm_gart_bind set gtt boun
CVE-2024-35823	In the Linux kernel, the following vulnerability has been resolved: vt: fix unicode buffer corruption when deleting ch
CVE-2024-35824	In the Linux kernel, the following vulnerability has been resolved: misc: lis3lv02d_i2c: Fix regulators getting en-/dis
CVE-2024-35826	In the Linux kernel, the following vulnerability has been resolved: block: Fix page refcounts for unaligned buffers in
CVE-2024-35832	In the Linux kernel, the following vulnerability has been resolved: bcachefs: kvfree bch_fs::snapshots in bch2_fs_sn
CVE-2024-35838	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix potential sta-link leak When
CVE-2024-35835	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: fix a double-free in arfs_create_group
CVE-2024-35839	In the Linux kernel, the following vulnerability has been resolved: netfilter: bridge: replace physindev with physinif
CVE-2024-35840	In the Linux kernel, the following vulnerability has been resolved: mptcp: use OPTION_MPTCP_MPJ_SYNACK in
CVE-2024-35843	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Use device rbtree in iopf reporting p
CVE-2024-35845	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: dbg-tlv: ensure NUL termination Th
CVE-2024-35847	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3-its: Prevent double free on error Th
CVE-2024-35848	In the Linux kernel, the following vulnerability has been resolved: eeprom: at24: fix memory corruption race conditi
CVE-2024-35852	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcam: Fix memory leak wh

CVE-2024-35853	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcam: Fix memory leak dur
CVE-2024-35854	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcam: Fix possible use-after
CVE-2024-35860	In the Linux kernel, the following vulnerability has been resolved: bpf: support deferring bpf_link dealloc to after R
CVE-2024-35855	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcam: Fix possible use-after
CVE-2024-35861	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in cifs_signal_cifs
CVE-2024-35862	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in smb2_is_network
CVE-2024-35863	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in is_valid_oplock
CVE-2024-35864	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in smb2_is_valid_
CVE-2024-35865	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in smb2_is_valid_
CVE-2024-35868	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in cifs_stats_proc_
CVE-2024-35869	In the Linux kernel, the following vulnerability has been resolved: smb: client: guarantee refcounted children from p
CVE-2024-35870	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix UAF in smb2_reconnect_server()
CVE-2024-35876	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-35875	In the Linux kernel, the following vulnerability has been resolved: x86/coco: Require seeding RNG with RDRAND
CVE-2024-35877	In the Linux kernel, the following vulnerability has been resolved: x86/mm/pat: fix VM_PAT handling in COW map
CVE-2024-35878	In the Linux kernel, the following vulnerability has been resolved: of: module: prevent NULL pointer dereference in
CVE-2024-35884	In the Linux kernel, the following vulnerability has been resolved: udp: do not accept non-tunnel GSO skbs landing
CVE-2024-35885	In the Linux kernel, the following vulnerability has been resolved: mlxbf_gige: stop interface during shutdown The n
CVE-2024-35886	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix infinite recursion in fib6_dump_done().
CVE-2024-35887	In the Linux kernel, the following vulnerability has been resolved: ax25: fix use-after-free bugs caused by ax25_ds_
CVE-2024-35888	In the Linux kernel, the following vulnerability has been resolved: erspan: make sure erspan_base_hdr is present in s
CVE-2024-35890	In the Linux kernel, the following vulnerability has been resolved: gro: fix ownership transfer If packets are GROed
CVE-2024-35893	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_skbmod: prevent kernel-infoleak sy
CVE-2024-35895	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Prevent lock inversion deadlock in
CVE-2024-35896	In the Linux kernel, the following vulnerability has been resolved: netfilter: validate user input for expected length I
CVE-2024-35897	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: discard table flag update with
CVE-2024-35898	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: Fix potential data-race in __n
CVE-2024-35899	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: flush pending destroy work be
CVE-2024-35900	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: reject new basechain after tab
CVE-2024-35901	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix Rx DMA datasize and skb_over_p
CVE-2024-35904	In the Linux kernel, the following vulnerability has been resolved: selinux: avoid dereference of garbage after mount
CVE-2024-35905	In the Linux kernel, the following vulnerability has been resolved: bpf: Protect against int overflow for stack access
CVE-2024-35907	In the Linux kernel, the following vulnerability has been resolved: mlxbf_gige: call request_irq() after NAPI initializ
CVE-2024-35908	In the Linux kernel, the following vulnerability has been resolved: tls: get psock ref after taking rxlock to avoid leak
CVE-2024-35911	In the Linux kernel, the following vulnerability has been resolved: ice: fix memory corruption bug with suspend and
CVE-2024-35910	In the Linux kernel, the following vulnerability has been resolved: tcp: properly terminate timers for kernel sockets V
CVE-2024-35912	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: mvm: rfi: fix potential response lea
CVE-2024-35924	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Limit read size on v1.2 Between
CVE-2024-35922	In the Linux kernel, the following vulnerability has been resolved: fbmon: prevent division by zero in fb_videomode
CVE-2024-35925	In the Linux kernel, the following vulnerability has been resolved: block: prevent division by zero in blk_rq_stat_sur

CVE-2024-35926	In the Linux kernel, the following vulnerability has been resolved: crypto: iaa - Fix async_disable descriptor leak Th
CVE-2024-35927	In the Linux kernel, the following vulnerability has been resolved: drm: Check output polling initialized before disab
CVE-2024-35928	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-35929	In the Linux kernel, the following vulnerability has been resolved: rcu/nocb: Fix WARN_ON_ONCE() in rcu_no
CVE-2024-35931	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Skip do PCI error slot reset during
CVE-2024-35930	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix possible memory leak in lpfc_rcv_p
CVE-2024-35932	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: don't check if plane->state->fb == state-
CVE-2024-35933	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btintel: Fix null ptr deref in btintel_rea
CVE-2024-35934	In the Linux kernel, the following vulnerability has been resolved: net/smc: reduce rtnl pressure in smc_pnet_create_
CVE-2024-35938	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: decrease MHI channel buffer length
CVE-2024-35937	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: check A-MSDU format more care
CVE-2024-35939	In the Linux kernel, the following vulnerability has been resolved: dma-direct: Leak pages on dma_set_decrypted() f
CVE-2024-35942	In the Linux kernel, the following vulnerability has been resolved: pmdomain: imx8mp-blk-ctrl: imx8mp_blk: Add f
CVE-2024-35944	In the Linux kernel, the following vulnerability has been resolved: VMCI: Fix memcpy() run-time warning in dg_dis
CVE-2024-35945	In the Linux kernel, the following vulnerability has been resolved: net: phy: phy_device: Prevent nullptr exceptions
CVE-2024-35946	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix null pointer access when abort sca
CVE-2024-35947	In the Linux kernel, the following vulnerability has been resolved: dyndbg: fix old BUG_ON in >control parser Fix a
CVE-2024-35948	In the Linux kernel, the following vulnerability has been resolved: bcachefs: Check for journal entries overrunning en
CVE-2024-35950	In the Linux kernel, the following vulnerability has been resolved: drm/client: Fully protect modes[] with dev->mode
CVE-2024-35949	In the Linux kernel, the following vulnerability has been resolved: btrfs: make sure that WRITTEN is set on all meta
CVE-2024-35951	In the Linux kernel, the following vulnerability has been resolved: drm/panfrost: Fix the error path in panfrost_mmu
CVE-2024-35952	In the Linux kernel, the following vulnerability has been resolved: drm/ast: Fix soft lockup There is a while-loop in a
CVE-2024-35956	In the Linux kernel, the following vulnerability has been resolved: btrfs: qgroup: fix qgroup prealloc rsv leak in subv
CVE-2024-35957	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix WARN_ON in iommu probe pat
CVE-2024-35958	In the Linux kernel, the following vulnerability has been resolved: net: ena: Fix incorrect descriptor free behavior EN
CVE-2024-35959	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix mlx5e_priv_init() cleanup flow W
CVE-2024-3596	RADIUS Protocol under RFC 2865 is susceptible to forgery attacks by a local attacker who can modify any valid Re
CVE-2024-35960	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Properly link new fs rules into the tree
CVE-2024-35961	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Register devlink first under devlink loc
CVE-2024-35962	In the Linux kernel, the following vulnerability has been resolved: netfilter: complete validation of user input In my
CVE-2024-35963	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sock: Fix not validating setsockopt
CVE-2024-35965	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix not validating setsockopt
CVE-2024-35966	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: Fix not validating setsock
CVE-2024-35967	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SCO: Fix not validating setsockopt use
CVE-2024-35969	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix race condition between ipv6_get_ifaddr
CVE-2024-35971	In the Linux kernel, the following vulnerability has been resolved: net: ks8851: Handle softirqs at the end of IRQ thr
CVE-2024-35976	In the Linux kernel, the following vulnerability has been resolved: xsk: validate user input for XDP_{UMEM COM
CVE-2024-35978	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix memory leak in hci_req_sync_con
CVE-2024-35979	In the Linux kernel, the following vulnerability has been resolved: raid1: fix use-after-free for original bio in raid1_v
CVE-2024-35983	In the Linux kernel, the following vulnerability has been resolved: bounds: Use the right number of bits for power-of

CVE-2024-35989	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix oops during rmmmod on sing
CVE-2024-35995	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Use access_width over bit_width fo
CVE-2024-35999	In the Linux kernel, the following vulnerability has been resolved: smb3: missing lock when picking channel Coveri
CVE-2024-35998	In the Linux kernel, the following vulnerability has been resolved: smb3: fix lock ordering potential deadlock in cif
CVE-2024-36000	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: fix missing hugetlb_lock for resv un
CVE-2024-36003	In the Linux kernel, the following vulnerability has been resolved: ice: fix LAG and VF lock dependency in ice_rese
CVE-2024-36004	In the Linux kernel, the following vulnerability has been resolved: i40e: Do not use WQ_MEM_RECLAIM flag for
CVE-2024-36005	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: honor table dormant flag from
CVE-2024-36006	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcam: Fix incorrect list API
CVE-2024-36007	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_tcam: Fix warning during r
CVE-2024-36009	In the Linux kernel, the following vulnerability has been resolved: ax25: Fix netdev refcount issue The dev_tracker i
CVE-2024-36010	In the Linux kernel, the following vulnerability has been resolved: igb: Fix string truncation warnings in igb_set_fw
CVE-2024-36012	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: msft: fix slab-use-after-free in msft_do
CVE-2024-36015	In the Linux kernel, the following vulnerability has been resolved: ppdev: Add an error check in register_device In r
CVE-2024-36013	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix slab-use-after-free in l2ca
CVE-2024-36014	In the Linux kernel, the following vulnerability has been resolved: drm/arm/malidp: fix a possible null pointer derefe
CVE-2024-36016	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: fix possible out-of-bounds in gsm0_re
CVE-2024-36017	In the Linux kernel, the following vulnerability has been resolved: rtnetlink: Correct nested IFLA_VF_VLAN_LIST
CVE-2024-36020	In the Linux kernel, the following vulnerability has been resolved: i40e: fix vf may be used uninitialized in this func
CVE-2024-36021	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when devlink reload du
CVE-2024-36022	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-36024	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Disable idle reallow as part of c
CVE-2024-36025	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix off by one in qla_edif_app_get
CVE-2024-36026	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: fixes a random hang in S4 for SMU
CVE-2024-36032	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: qca: fix info leak when fetching fw bu
CVE-2024-36031	In the Linux kernel, the following vulnerability has been resolved: keys: Fix overwrite of key expiration on instantia
CVE-2024-36114	Aircompressor is a library with ports of the Snappy, LZO, LZ4, and Zstandard compression algorithms to Java. All o
CVE-2024-36137	A vulnerability has been identified in Node.js, affecting users of the experimental permission model when the --allow
CVE-2024-36244	In the Linux kernel, the following vulnerability has been resolved: net/sched: taprio: extend minimum interval restric
CVE-2024-36347	Improper signature verification in AMD CPU ROM microcode patch loader may allow an attacker with local admini
CVE-2024-36286	In the Linux kernel, the following vulnerability has been resolved: netfilter: nfnetlink_queue: acquire rcu_read_lock
CVE-2024-36270	In the Linux kernel, the following vulnerability has been resolved: netfilter: tproxy: bail out if IP has been disabled o
CVE-2024-36350	A transient execution vulnerability in some AMD processors may allow an attacker to infer data from previous store
CVE-2024-36357	A transient execution vulnerability in some AMD processors may allow an attacker to infer data in the L1D cache, p
CVE-2024-36476	In the Linux kernel, the following vulnerability has been resolved: RDMA/rtrs: Ensure 'ib_sge list' is accessible Mov
CVE-2024-36478	In the Linux kernel, the following vulnerability has been resolved: null_blk: fix null-ptr-dereference while configurin
CVE-2024-36479	In the Linux kernel, the following vulnerability has been resolved: fpga: bridge: add owner module and take its refco
CVE-2024-36484	In the Linux kernel, the following vulnerability has been resolved: net: relax socket state check at accept time. Chris
CVE-2024-36489	In the Linux kernel, the following vulnerability has been resolved: tls: fix missing memory barrier in tls_init In tls_in
CVE-2024-3651	A vulnerability was identified in the kjd/idna library, specifically within the `idna.encode()` function, affecting versi

CVE-2024-36613	FFmpeg n6.1.1 has a vulnerability in the DXA demuxer of the libavformat library allowing for an integer overflow, which can be used to trigger a crash.
CVE-2024-36616	An integer overflow in the component /libavformat/westwood_vqa.c of FFmpeg n6.1.1 allows attackers to cause a denial of service (crash) or execute arbitrary code.
CVE-2024-36617	FFmpeg n6.1.1 has an integer overflow vulnerability in the FFmpeg CAF decoder.
CVE-2024-36618	FFmpeg n6.1.1 has a vulnerability in the AVI demuxer of the libavformat library which allows for an integer overflow, which can be used to trigger a crash.
CVE-2024-36619	FFmpeg n6.1.1 has a vulnerability in the WAVARC decoder of the libavcodec library which allows for an integer overflow, which can be used to trigger a crash.
CVE-2024-36623	moby through v25.0.3 has a Race Condition vulnerability in the streamformatter package which can be used to trigger a crash.
CVE-2024-36621	moby v25.0.5 is affected by a Race Condition in builder/builder-next/adapters/snapshot/layer.go. The vulnerability can be used to trigger a crash.
CVE-2024-36880	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: qca: add missing firmware sanity check
CVE-2024-36881	In the Linux kernel, the following vulnerability has been resolved: mm/userfaultfd: reset ptes when close() for wr-pr
CVE-2024-36882	In the Linux kernel, the following vulnerability has been resolved: mm: use memalloc_nofs_save() in page_cache_ra
CVE-2024-36883	In the Linux kernel, the following vulnerability has been resolved: net: fix out-of-bounds access in ops_init net_alloc
CVE-2024-36884	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Use the correct type in nvidia
CVE-2024-36885	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-36886	In the Linux kernel, the following vulnerability has been resolved: tipc: fix UAF in error path Sam Page (sam4k) wo
CVE-2024-36889	In the Linux kernel, the following vulnerability has been resolved: mptcp: ensure snd_nxt is properly initialized on c
CVE-2024-36893	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tcpc: Check for port partner validity b
CVE-2024-36894	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_fs: Fix race between aio_cancel() a
CVE-2024-36898	In the Linux kernel, the following vulnerability has been resolved: gpiolib: cdev: fix uninitialised kfifo If a line is rec
CVE-2024-36896	In the Linux kernel, the following vulnerability has been resolved: USB: core: Fix access violation during port device
CVE-2024-36899	In the Linux kernel, the following vulnerability has been resolved: gpiolib: cdev: Fix use after free in lineinfo_chang
CVE-2024-36900	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when devlink reload du
CVE-2024-36901	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent NULL dereference in ip6_output() .
CVE-2024-36902	In the Linux kernel, the following vulnerability has been resolved: ipv6: fib6_rules: avoid possible NULL dereference
CVE-2024-36903	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix potential uninit-value access in __ip6_n
CVE-2024-36904	In the Linux kernel, the following vulnerability has been resolved: tcp: Use refcount_inc_not_zero() in tcp_twsk_uni
CVE-2024-36906	In the Linux kernel, the following vulnerability has been resolved: ARM: 9381/1: kasan: clear stale stack poison We
CVE-2024-36905	In the Linux kernel, the following vulnerability has been resolved: tcp: defer shutdown(SEND_SHUTDOWN) for T
CVE-2024-36909	In the Linux kernel, the following vulnerability has been resolved: Drivers: hv: vmbus: Don't free ring buffers that co
CVE-2024-36907	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-36910	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Don't free decrypted memory In
CVE-2024-36911	In the Linux kernel, the following vulnerability has been resolved: hv_netvsc: Don't free decrypted memory In CoCo
CVE-2024-36912	In the Linux kernel, the following vulnerability has been resolved: Drivers: hv: vmbus: Track decrypted status in vm
CVE-2024-36914	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Skip on writeback when it's not
CVE-2024-36913	In the Linux kernel, the following vulnerability has been resolved: Drivers: hv: vmbus: Leak pages if set_memory_e
CVE-2024-36915	In the Linux kernel, the following vulnerability has been resolved: nfc: llcp: fix nfc_llcp_setsockopt() unsafe copies
CVE-2024-36916	In the Linux kernel, the following vulnerability has been resolved: blk-iocost: avoid out of bounds shift UBSAN cat
CVE-2024-36917	In the Linux kernel, the following vulnerability has been resolved: block: fix overflow in blk_ioctl_discard() There is
CVE-2024-36918	In the Linux kernel, the following vulnerability has been resolved: bpf: Check bloom filter map value size This patch
CVE-2024-36919	In the Linux kernel, the following vulnerability has been resolved: scsi: bnx2fc: Remove spin_lock_bh while releasin
CVE-2024-36920	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Avoid memcpy field-spanning writ

CVE-2024-36921	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: guard against invalid STA ID
CVE-2024-36922	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: read txq->read_ptr under lock If we
CVE-2024-36924	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Release hbalock before calling lpfc_wc
CVE-2024-36923	In the Linux kernel, the following vulnerability has been resolved: fs/9p: fix uninitialized values during inode evict I
CVE-2024-36925	In the Linux kernel, the following vulnerability has been resolved: swiotlb: initialise restricted pool list_head when S
CVE-2024-36928	In the Linux kernel, the following vulnerability has been resolved: s390/qeth: Fix kernel panic after setting hsnid Syn
CVE-2024-36927	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix uninit-value access in __ip_make_skb()
CVE-2024-36929	In the Linux kernel, the following vulnerability has been resolved: net: core: reject skb_copy(_expand) for fraglist G
CVE-2024-36931	In the Linux kernel, the following vulnerability has been resolved: s390/cio: Ensure the copied buf is NUL terminate
CVE-2024-36933	In the Linux kernel, the following vulnerability has been resolved: nsh: Restore skb->{protocol,data,mac_header} fo
CVE-2024-36934	In the Linux kernel, the following vulnerability has been resolved: bna: ensure the copied buf is NUL terminated Cu
CVE-2024-36937	In the Linux kernel, the following vulnerability has been resolved: xdp: use flags field to disambiguate broadcast red
CVE-2024-36939	In the Linux kernel, the following vulnerability has been resolved: nfs: Handle error of rpc_proc_register() in nfs_ne
CVE-2024-36938	In the Linux kernel, the following vulnerability has been resolved: bpf, skmsg: Fix NULL pointer dereference in sk_
CVE-2024-36940	In the Linux kernel, the following vulnerability has been resolved: pinctrl: core: delete incorrect free in pinctrl_enabl
CVE-2024-36944	In the Linux kernel, the following vulnerability has been resolved: Reapply "drm/qxl: simplify qxl_fence_wait" This
CVE-2024-36941	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: don't free NULL coalescing rule If
CVE-2024-36945	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix neighbour and rtable leak in smc_ib_
CVE-2024-36946	In the Linux kernel, the following vulnerability has been resolved: phonet: fix rtm_phonet_notify() skb allocation fil
CVE-2024-36947	In the Linux kernel, the following vulnerability has been resolved: qibfs: fix dentry leak simple_recursive_removal()
CVE-2024-36948	In the Linux kernel, the following vulnerability has been resolved: drm/xe/xe_migrate: Cast to output precision befor
CVE-2024-36949	In the Linux kernel, the following vulnerability has been resolved: amd/amdkfd: sync all devices to wait all processe
CVE-2024-36950	In the Linux kernel, the following vulnerability has been resolved: firewire: ohci: mask bus reset interrupts between
CVE-2024-36951	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: range check cp bad op exception int
CVE-2024-36952	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Move NPIV's transport unregistration t
CVE-2024-36953	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: vgic-v2: Check for non-NULL vCI
CVE-2024-36954	In the Linux kernel, the following vulnerability has been resolved: tipc: fix a possible memleak in tipc_buf_append
CVE-2024-36955	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: intel-sdw-acpi: fix usage of device_g
CVE-2024-36957	In the Linux kernel, the following vulnerability has been resolved: octeontx2-af: avoid off-by-one read from userspa
CVE-2024-36959	In the Linux kernel, the following vulnerability has been resolved: pinctrl: devicetree: fix refcount leak in pinctrl_dt
CVE-2024-36960	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Fix invalid reads in fence signaled
CVE-2024-36962	In the Linux kernel, the following vulnerability has been resolved: net: ks8851: Queue RX packets in IRQ handler in
CVE-2024-36964	In the Linux kernel, the following vulnerability has been resolved: fs/9p: only translate RWX permissions for plain 9
CVE-2024-36965	In the Linux kernel, the following vulnerability has been resolved: remoteproc: mediatek: Make sure IPI buffer fits i
CVE-2024-36966	In the Linux kernel, the following vulnerability has been resolved: erofs: reliably distinguish block based and fscache
CVE-2024-36967	In the Linux kernel, the following vulnerability has been resolved: KEYS: trusted: Fix memory leak in tpm2_key_en
CVE-2024-36968	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix div-by-zero in l2cap_le_f
CVE-2024-36970	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: Use request_module_nowait This ap
CVE-2024-36969	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix division by zero in setup_d
CVE-2024-36971	In the Linux kernel, the following vulnerability has been resolved: net: fix __dst_negative_advice() race __dst_negat

CVE-2024-36972	In the Linux kernel, the following vulnerability has been resolved: af_unix: Update unix_sk(sk)->oob_skb under sk_
CVE-2024-36974	In the Linux kernel, the following vulnerability has been resolved: net/sched: taprio: always validate TCA_TAPRIO
CVE-2024-36975	In the Linux kernel, the following vulnerability has been resolved: KEYS: trusted: Do not use WARN when encode
CVE-2024-36978	In the Linux kernel, the following vulnerability has been resolved: net: sched: sch_multiq: fix possible OOB write in
CVE-2024-36979	In the Linux kernel, the following vulnerability has been resolved: net: bridge: mst: fix vlan use-after-free syzbot rep
CVE-2024-37021	In the Linux kernel, the following vulnerability has been resolved: fpga: manager: add owner module and take its ref
CVE-2024-37078	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential kernel bug due to lack of writ
CVE-2024-37353	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-37354	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix crash on racing fsync and size-extending
CVE-2024-37356	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix shift-out-of-bounds in dctcp_update_alph
CVE-2024-37370	In MIT Kerberos 5 (aka krb5) before 1.21.3, an attacker can modify the plaintext Extra Count field of a confidential
CVE-2024-37371	In MIT Kerberos 5 (aka krb5) before 1.21.3, an attacker can cause invalid memory reads during GSS message token
CVE-2024-37891	urllib3 is a user-friendly HTTP client library for Python. When using urllib3's proxy support with `ProxyManager`, t
CVE-2024-38306	In the Linux kernel, the following vulnerability has been resolved: btrfs: protect folio::private when attaching exten
CVE-2024-38381	In the Linux kernel, the following vulnerability has been resolved: nfc: nci: Fix uninit-value in nci_rx_work syzbot r
CVE-2024-38384	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: fix list corruption from reorder of WB
CVE-2024-38388	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda/cs_dsp_ctl: Use private_free for contr
CVE-2024-38473	Encoding problem in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows request URLs with incorrect enc
CVE-2024-38428	url.c in GNU Wget through 1.24.5 mishandles semicolons in the userinfo subcomponent of a URI, and thus there ma
CVE-2024-38474	Substitution encoding issue in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows attacker to execute scr
CVE-2024-38475	Improper escaping of output in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to map UR
CVE-2024-38476	Vulnerability in core of Apache HTTP Server 2.4.59 and earlier are vulnerably to information disclosure, SSRF or lo
CVE-2024-38477	null pointer dereference in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows an attacker to crash the serv
CVE-2024-38538	In the Linux kernel, the following vulnerability has been resolved: net: bridge: xmit: make sure we have at least eth
CVE-2024-38540	In the Linux kernel, the following vulnerability has been resolved: bnxt_re: avoid shift undefined behavior in bnxt_q
CVE-2024-38543	In the Linux kernel, the following vulnerability has been resolved: lib/test_hmm.c: handle src_pfns and dst_pfns allo
CVE-2024-38541	In the Linux kernel, the following vulnerability has been resolved: of: module: add buffer overflow check in of_mod
CVE-2024-38544	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix seg fault in rxe_comp_queue_pk
CVE-2024-38545	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix UAF for cq async event The refc
CVE-2024-38546	In the Linux kernel, the following vulnerability has been resolved: drm: vc4: Fix possible null pointer dereference In
CVE-2024-38547	In the Linux kernel, the following vulnerability has been resolved: media: atomisp: ssh_css: Fix a null-pointer derefe
CVE-2024-38548	In the Linux kernel, the following vulnerability has been resolved: drm: bridge: cdns-mhdp8546: Fix possible null po
CVE-2024-38549	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: Add 0 size check to mtk_drm_gen
CVE-2024-38550	In the Linux kernel, the following vulnerability has been resolved: ASoC: kirkwood: Fix potential NULL dereferenc
CVE-2024-38552	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix potential index out of bound
CVE-2024-38553	In the Linux kernel, the following vulnerability has been resolved: net: fec: remove .ndo_poll_controller to avoid dea
CVE-2024-38554	In the Linux kernel, the following vulnerability has been resolved: ax25: Fix reference count leak issue of net_device
CVE-2024-38555	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Discard command completions in interr
CVE-2024-38556	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Add a timeout to acquire the command
CVE-2024-38557	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Reload only IB representors upon lag d

CVE-2024-38558	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix overwriting ct original tuple
CVE-2024-38559	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Ensure the copied buf is NUL terminat
CVE-2024-38560	In the Linux kernel, the following vulnerability has been resolved: scsi: bfa: Ensure the copied buf is NUL terminat
CVE-2024-38561	In the Linux kernel, the following vulnerability has been resolved: kunit: Fix kthread reference There is a race condi
CVE-2024-38564	In the Linux kernel, the following vulnerability has been resolved: bpf: Add BPF_PROG_TYPE_CGROUP_SKB ar
CVE-2024-38565	In the Linux kernel, the following vulnerability has been resolved: wifi: ar5523: enable proper endpoint verification
CVE-2024-38567	In the Linux kernel, the following vulnerability has been resolved: wifi: carl9170: add a proper sanity check for endp
CVE-2024-38570	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix potential glock use-after-free on unmount
CVE-2024-38571	In the Linux kernel, the following vulnerability has been resolved: thermal/drivers/tsens: Fix null pointer dereference
CVE-2024-38573	In the Linux kernel, the following vulnerability has been resolved: cpcp_cpufreq: Fix possible null pointer dereferen
CVE-2024-38575	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: pcie: handle randbuf allocation f
CVE-2024-38577	In the Linux kernel, the following vulnerability has been resolved: rcu-tasks: Fix show_rcu_tasks_trace_gp_kthread
CVE-2024-38578	In the Linux kernel, the following vulnerability has been resolved: ecryptfs: Fix buffer size for tag 66 packet The 'TA
CVE-2024-38579	In the Linux kernel, the following vulnerability has been resolved: crypto: bcm - Fix pointer arithmetic In spu2_dum
CVE-2024-38580	In the Linux kernel, the following vulnerability has been resolved: epoll: be better about file lifetimes epoll can call
CVE-2024-38581	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/mes: fix use-after-free issue Delete
CVE-2024-38582	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential hang in nilfs_detach_log_writ
CVE-2024-38583	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix use-after-free of timer for log writer thr
CVE-2024-38586	In the Linux kernel, the following vulnerability has been resolved: r8169: Fix possible ring buffer corruption on fragm
CVE-2024-38588	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix possible use-after-free issue in ftrace_l
CVE-2024-38587	In the Linux kernel, the following vulnerability has been resolved: speakup: Fix sizeof() vs ARRAY_SIZE() bug The
CVE-2024-38589	In the Linux kernel, the following vulnerability has been resolved: netrom: fix possible dead-lock in nr_rt_ioctl() syz
CVE-2024-38590	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Modify the print level of CQE error
CVE-2024-38591	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix deadlock on SRQ async events.
CVE-2024-38594	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: move the EST lock to struct stmmac
CVE-2024-38596	In the Linux kernel, the following vulnerability has been resolved: af_unix: Fix data races in unix_release_sock/unix
CVE-2024-38597	In the Linux kernel, the following vulnerability has been resolved: eth: sungem: remove .ndo_poll_controller to avoid
CVE-2024-38598	In the Linux kernel, the following vulnerability has been resolved: md: fix resync softlockup when bitmap size is les
CVE-2024-38599	In the Linux kernel, the following vulnerability has been resolved: jffs2: prevent xattr node from overflowing the era
CVE-2024-38601	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix a race between readers and resize
CVE-2024-38600	In the Linux kernel, the following vulnerability has been resolved: ALSA: Fix deadlocks with kctl removals at disco
CVE-2024-38602	In the Linux kernel, the following vulnerability has been resolved: ax25: Fix reference count leak issues of ax25_dev
CVE-2024-38605	In the Linux kernel, the following vulnerability has been resolved: ALSA: core: Fix NULL module pointer assignme
CVE-2024-38607	In the Linux kernel, the following vulnerability has been resolved: macintosh/via-macii: Fix "BUG: sleeping functio
CVE-2024-38608	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix netif state handling mlx5e_suspen
CVE-2024-38610	In the Linux kernel, the following vulnerability has been resolved: drivers/virt/acrn: fix PFNMAP PTE checks in acrn
CVE-2024-38611	In the Linux kernel, the following vulnerability has been resolved: media: i2c: et8ek8: Don't strip remove function w
CVE-2024-38612	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: fix invalid unregister error path The erro
CVE-2024-38613	In the Linux kernel, the following vulnerability has been resolved: m68k: Fix spinlock race in kernel thread creation
CVE-2024-38615	In the Linux kernel, the following vulnerability has been resolved: cpufreq: exit() callback is optional The exit() call

CVE-2024-38618	In the Linux kernel, the following vulnerability has been resolved: ALSA: timer: Set lower bound of start tick time C
CVE-2024-38619	In the Linux kernel, the following vulnerability has been resolved: usb-storage: alauda: Check whether the media is s
CVE-2024-38621	In the Linux kernel, the following vulnerability has been resolved: media: stk1160: fix bounds checking in stk1160_
CVE-2024-38623	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Use variable length array instead of fixe
CVE-2024-38624	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Use 64 bit variable to avoid 32 bit overfl
CVE-2024-38625	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Check 'folio' pointer for NULL. It can be
CVE-2024-38627	In the Linux kernel, the following vulnerability has been resolved: stm class: Fix a double free in stm_register_devic
CVE-2024-38628	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: u_audio: Fix race condition use of co
CVE-2024-38630	In the Linux kernel, the following vulnerability has been resolved: watchdog: cpu5wdt.c: Fix use-after-free bug caus
CVE-2024-38632	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: fix potential memory leak in vfio_intx_e
CVE-2024-38633	In the Linux kernel, the following vulnerability has been resolved: serial: max3100: Update uart_driver_registered o
CVE-2024-38634	In the Linux kernel, the following vulnerability has been resolved: serial: max3100: Lock port->lock when calling u
CVE-2024-38635	In the Linux kernel, the following vulnerability has been resolved: soundwire: cadence: fix invalid PDI offset For so
CVE-2024-38637	In the Linux kernel, the following vulnerability has been resolved: greybus: lights: check return of get_channel_from
CVE-2024-38659	In the Linux kernel, the following vulnerability has been resolved: enic: Validate length of nl attributes in enic_set_v
CVE-2024-38661	In the Linux kernel, the following vulnerability has been resolved: s390/ap: Fix crash in AP internal function modify
CVE-2024-38662	In the Linux kernel, the following vulnerability has been resolved: bpf: Allow delete from sockmap/sockhash only if
CVE-2024-38663	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: fix list corruption from resetting io st
CVE-2024-38667	In the Linux kernel, the following vulnerability has been resolved: riscv: prevent pt_regs corruption for secondary id
CVE-2024-38780	In the Linux kernel, the following vulnerability has been resolved: dma-buf/sw-sync: don't enable IRQ from sync_pr
CVE-2024-38808	In Spring Framework versions 5.3.0 - 5.3.38 and older unsupported versions, it is possible for a user to provide a spe
CVE-2024-38809	Applications that parse ETags from "If-Match" or "If-None-Match" request headers are vulnerable to DoS attack. Us
CVE-2024-38820	The fix for CVE-2022-22968 made disallowedFields patterns in DataBinder case insensitive. However, String.toLow
CVE-2024-38827	The usage of String.toLowerCase() and String.toUpperCase() has some Locale dependent exceptions that could pote
CVE-2024-39276	In the Linux kernel, the following vulnerability has been resolved: ext4: fix mb_cache_entry's e_refcnt leak in ext4_
CVE-2024-39277	In the Linux kernel, the following vulnerability has been resolved: dma-mapping: benchmark: handle NUMA_NO_N
CVE-2024-39282	In the Linux kernel, the following vulnerability has been resolved: net: wwan: t7xx: Fix FSM command timeout issu
CVE-2024-39292	In the Linux kernel, the following vulnerability has been resolved: um: Add winch to winch_handlers before register
CVE-2024-39293	In the Linux kernel, the following vulnerability has been resolved: Revert "xsk: Support redirect to any socket bound
CVE-2024-39298	In the Linux kernel, the following vulnerability has been resolved: mm/memory-failure: fix handling of dissolved bu
CVE-2024-39301	In the Linux kernel, the following vulnerability has been resolved: net/9p: fix uninit-value in p9_client_rpc() Syzbot
CVE-2024-39331	In Emacs before 29.4, org-link-expand-abbrev in lisp/ol.el expands a %(...) link abbrev even when it specifies an uns
CVE-2024-39463	In the Linux kernel, the following vulnerability has been resolved: 9p: add missing locking around taking dentry fid
CVE-2024-39466	In the Linux kernel, the following vulnerability has been resolved: thermal/drivers/qcom/lmh: Check for SCM availa
CVE-2024-39467	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on i_xattr_nid in sanit
CVE-2024-39468	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix deadlock in smb2_find_smb_tcon
CVE-2024-39469	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix nilfs_empty_dir() misjudgment and lon
CVE-2024-39471	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: add error handle to avoid out-of-bo
CVE-2024-39472	In the Linux kernel, the following vulnerability has been resolved: xfs: fix log recovery buffer allocation for the lega
CVE-2024-39473	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-topology: Fix input format que

CVE-2024-39475	In the Linux kernel, the following vulnerability has been resolved: fbdev: savage: Handle err return when savagefb_
CVE-2024-39476	In the Linux kernel, the following vulnerability has been resolved: md/raid5: fix deadlock that raid5d() wait for itself
CVE-2024-39484	In the Linux kernel, the following vulnerability has been resolved: mmc: davinci: Don't strip remove function when
CVE-2024-39479	In the Linux kernel, the following vulnerability has been resolved: drm/i915/hwmon: Get rid of devm When both hw
CVE-2024-39480	In the Linux kernel, the following vulnerability has been resolved: kdb: Fix buffer overflow during tab-complete Cu
CVE-2024-39487	In the Linux kernel, the following vulnerability has been resolved: bonding: Fix out-of-bounds read in bond_option_
CVE-2024-39488	In the Linux kernel, the following vulnerability has been resolved: arm64: asm-bug: Add .align 2 to the end of __BU
CVE-2024-39490	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: fix missing sk_buff release in seg6_inpu
CVE-2024-39493	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - Fix ADF_DEV_RESET_SYNC men
CVE-2024-39489	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: fix memleak in seg6_hmac_init_algo seg
CVE-2024-39495	In the Linux kernel, the following vulnerability has been resolved: greybus: Fix use-after-free bug in gb_interface_re
CVE-2024-39494	In the Linux kernel, the following vulnerability has been resolved: ima: Fix use-after-free on a dentry's dname.name
CVE-2024-39497	In the Linux kernel, the following vulnerability has been resolved: drm/shmem-helper: Fix BUG_ON() on mmap(PR
CVE-2024-39499	In the Linux kernel, the following vulnerability has been resolved: vmci: prevent speculation leaks by sanitizing ever
CVE-2024-39500	In the Linux kernel, the following vulnerability has been resolved: sock_map: avoid race between sock_map_close a
CVE-2024-39496	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: fix use-after-free due to race with de
CVE-2024-39501	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-39502	In the Linux kernel, the following vulnerability has been resolved: ionic: fix use after netif_napi_del() When queues
CVE-2024-39503	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: Fix race between namespace clean
CVE-2024-39507	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash problem in concurrent
CVE-2024-39505	In the Linux kernel, the following vulnerability has been resolved: drm/komeda: check for error-valued pointer komed
CVE-2024-39506	In the Linux kernel, the following vulnerability has been resolved: liquidio: Adjust a NULL pointer handling path in
CVE-2024-39508	In the Linux kernel, the following vulnerability has been resolved: io_uring/io-wq: Use set_bit() and test_bit() at wor
CVE-2024-39509	In the Linux kernel, the following vulnerability has been resolved: HID: core: remove unnecessary WARN_ON() in
CVE-2024-39573	Potential SSRF in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to cause unsafe Rewrite
CVE-2024-39689	Certifi is a curated collection of Root Certificates for validating the trustworthiness of SSL certificates while verifyin
CVE-2024-39908	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.1 has some DoS vulnerabilities when it parses an
CVE-2024-39920	The TCP protocol in RFC 9293 has a timing side channel that makes it easier for remote attackers to infer the conten
CVE-2024-4032	The “ipaddress” module contained incorrect information about whether certain IPv4 and IPv6 addresses were design
CVE-2024-40635	containerd is an open-source container runtime. A bug was found in containerd prior to versions 1.6.38, 1.7.27, and 2
CVE-2024-40900	In the Linux kernel, the following vulnerability has been resolved: cachefiles: remove requests from xarray during fl
CVE-2024-40901	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Avoid test/set_bit() operating in no
CVE-2024-40902	In the Linux kernel, the following vulnerability has been resolved: jfs: xattr: fix buffer overflow for invalid xattr Wh
CVE-2024-40905	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix possible race in __fib6_drop_pcpu_from
CVE-2024-40904	In the Linux kernel, the following vulnerability has been resolved: USB: class: cdc-wdm: Fix CPU lockup caused by
CVE-2024-40906	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Always stop health timer during driver
CVE-2024-40908	In the Linux kernel, the following vulnerability has been resolved: bpf: Set run context for rawtp test_run callback sy
CVE-2024-40911	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: Lock wiphy in cfg80211_get_stat
CVE-2024-40910	In the Linux kernel, the following vulnerability has been resolved: ax25: Fix refcount imbalance on inbound connect
CVE-2024-40914	In the Linux kernel, the following vulnerability has been resolved: mm/huge_memory: don't unpoison huge_zero_fo

CVE-2024-40912	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Fix deadlock in ieee80211_sta_p
CVE-2024-40915	In the Linux kernel, the following vulnerability has been resolved: riscv: rewrite __kernel_map_pages() to fix sleepi
CVE-2024-40916	In the Linux kernel, the following vulnerability has been resolved: drm/exynos: hdmi: report safe 640x480 mode as a
CVE-2024-40918	In the Linux kernel, the following vulnerability has been resolved: parisc: Try to fix random segmentation faults in p
CVE-2024-40919	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Adjust logging of firmware messages in
CVE-2024-40923	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: disable rx data ring on dma allocation f
CVE-2024-40924	In the Linux kernel, the following vulnerability has been resolved: drm/i915/dpt: Make DPT object unshrinkable In s
CVE-2024-40929	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: check n_ssids before accessin
CVE-2024-40927	In the Linux kernel, the following vulnerability has been resolved: xhci: Handle TD clearing for multiple streams cas
CVE-2024-40931	In the Linux kernel, the following vulnerability has been resolved: mptcp: ensure snd_una is properly initialized on c
CVE-2024-40932	In the Linux kernel, the following vulnerability has been resolved: drm/exynos/vidi: fix memory leak in .get_modes()
CVE-2024-40934	In the Linux kernel, the following vulnerability has been resolved: HID: logitech-dj: Fix memory leak in logi_dj_rec
CVE-2024-40937	In the Linux kernel, the following vulnerability has been resolved: gve: Clear napi->skb before dev_kfree_skb_any()
CVE-2024-40940	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix tainted pointer delete is case of flow
CVE-2024-40942	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: mesh: Fix leak of mesh_preq_qu
CVE-2024-40943	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix races between hole punching and AIO+
CVE-2024-40941	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: don't read past the mfuart not
CVE-2024-40945	In the Linux kernel, the following vulnerability has been resolved: iommu: Return right value in iommu_sva_bind_d
CVE-2024-40953	In the Linux kernel, the following vulnerability has been resolved: KVM: Fix a data race on last_boosted_vcpu in kv
CVE-2024-40954	In the Linux kernel, the following vulnerability has been resolved: net: do not leave a dangling sk pointer, when sock
CVE-2024-40956	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix possible Use-After-Free in
CVE-2024-40957	In the Linux kernel, the following vulnerability has been resolved: seg6: fix parameter passing when calling NF_HO
CVE-2024-40958	In the Linux kernel, the following vulnerability has been resolved: netns: Make get_net_ns() handle zero refcount ne
CVE-2024-40959	In the Linux kernel, the following vulnerability has been resolved: xfrm6: check ip6_dst_idev() return value in xfrm
CVE-2024-40961	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent possible NULL deref in fib6_nh_in
CVE-2024-40960	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent possible NULL dereference in rt6_p
CVE-2024-40963	In the Linux kernel, the following vulnerability has been resolved: mips: bmips: BCM6358: make sure CBR is corre
CVE-2024-40964	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: cs35l41: Possible null pointer derefer
CVE-2024-40965	In the Linux kernel, the following vulnerability has been resolved: i2c: lpi2c: Avoid calling clk_get_rate during trans
CVE-2024-40966	In the Linux kernel, the following vulnerability has been resolved: tty: add the option to have a tty reject a new ldisc
CVE-2024-40967	In the Linux kernel, the following vulnerability has been resolved: serial: imx: Introduce timeout when waiting on tra
CVE-2024-40968	In the Linux kernel, the following vulnerability has been resolved: MIPS: Octeon: Add PCIe link status check The st
CVE-2024-40970	In the Linux kernel, the following vulnerability has been resolved: Avoid hw_desc array overrun in dw-axi-dmac I h
CVE-2024-40972	In the Linux kernel, the following vulnerability has been resolved: ext4: do not create EA inode under buffer lock ex
CVE-2024-40971	In the Linux kernel, the following vulnerability has been resolved: f2fs: remove clear SB_INLINECRYPT flag in de
CVE-2024-40974	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Enforce hcall result buffer valid
CVE-2024-40973	In the Linux kernel, the following vulnerability has been resolved: media: mtk-vcodec: potential null pointer deferen
CVE-2024-40975	In the Linux kernel, the following vulnerability has been resolved: platform/x86: x86-android-tablets: Unregister dev
CVE-2024-40976	In the Linux kernel, the following vulnerability has been resolved: drm/lima: mask irqs in timeout path before hard r
CVE-2024-40977	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921s: fix potential hung tasks durin

CVE-2024-40979	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix kernel crash during resume Curr
CVE-2024-40978	In the Linux kernel, the following vulnerability has been resolved: scsi: qedi: Fix crash while reading debugfs attribu
CVE-2024-40980	In the Linux kernel, the following vulnerability has been resolved: drop_monitor: replace spin_lock by raw_spin_lo
CVE-2024-40981	In the Linux kernel, the following vulnerability has been resolved: batman-adv: bypass empty buckets in batadv_pur
CVE-2024-40983	In the Linux kernel, the following vulnerability has been resolved: tipc: force a dst refcount before doing decryption
CVE-2024-40984	In the Linux kernel, the following vulnerability has been resolved: ACPICA: Revert "ACPICA: avoid Info: mapping
CVE-2024-40987	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix UBSAN warning in kv_dpm.c A
CVE-2024-40988	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix UBSAN warning in kv_dpm.c A
CVE-2024-40990	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Add check for srq max_sge attribu
CVE-2024-40994	In the Linux kernel, the following vulnerability has been resolved: ptp: fix integer overflow in max_vclocks_store O
CVE-2024-40995	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_api: fix possible infinite loop in tcf
CVE-2024-40989	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Disassociate vcpus from redistribu
CVE-2024-40997	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: fix memory leak on CPU EF
CVE-2024-40998	In the Linux kernel, the following vulnerability has been resolved: ext4: fix uninitialized ratelimit_state->lock access
CVE-2024-40999	In the Linux kernel, the following vulnerability has been resolved: net: ena: Add validation for completion descriptor
CVE-2024-41000	In the Linux kernel, the following vulnerability has been resolved: block/ioctl: prefer different overflow check Runn
CVE-2024-41001	In the Linux kernel, the following vulnerability has been resolved: io_uring/sqpoll: work around a potential audit me
CVE-2024-41002	In the Linux kernel, the following vulnerability has been resolved: crypto: hisilicon/sec - Fix memory leak for sec res
CVE-2024-41004	In the Linux kernel, the following vulnerability has been resolved: tracing: Build event generation tests only as modu
CVE-2024-41005	In the Linux kernel, the following vulnerability has been resolved: netpoll: Fix race condition in netpoll_owner_activ
CVE-2024-41006	In the Linux kernel, the following vulnerability has been resolved: netrom: Fix a memory leak in nr_heartbeat_expir
CVE-2024-41007	In the Linux kernel, the following vulnerability has been resolved: tcp: avoid too many retransmit packets If a TCP s
CVE-2024-41008	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: change vm->task_info handling Th
CVE-2024-41009	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix overrunning reservations in ringbuf The
CVE-2024-41010	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix too early release of tcx_entry Pedro Pinto
CVE-2024-41012	In the Linux kernel, the following vulnerability has been resolved: filelock: Remove locks reliably when fcntl/close
CVE-2024-41011	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: don't allow mapping the MMIO HD
CVE-2024-41013	In the Linux kernel, the following vulnerability has been resolved: xfs: don't walk off the end of a directory data bloc
CVE-2024-41014	In the Linux kernel, the following vulnerability has been resolved: xfs: add bounds checking to xlog_recover_proces
CVE-2024-41015	In the Linux kernel, the following vulnerability has been resolved: ocfs2: add bounds checking to ocfs2_check_dir_e
CVE-2024-41016	In the Linux kernel, the following vulnerability has been resolved: ocfs2: strict bound check before memcmp in ocfs
CVE-2024-41017	In the Linux kernel, the following vulnerability has been resolved: jfs: don't walk off the end of ealist Add a check b
CVE-2024-41019	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Validate ff offset This adds sanity check
CVE-2024-41022	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix signedness bug in sdma_v4_0_
CVE-2024-41020	In the Linux kernel, the following vulnerability has been resolved: filelock: Fix fcntl/close race recovery compat pat
CVE-2024-41023	In the Linux kernel, the following vulnerability has been resolved: sched/deadline: Fix task_struct reference leak Du
CVE-2024-41027	In the Linux kernel, the following vulnerability has been resolved: Fix userfaultfd_api to return EINVAL as expected
CVE-2024-41030	In the Linux kernel, the following vulnerability has been resolved: ksmbd: discard write access to the directory open
CVE-2024-41031	In the Linux kernel, the following vulnerability has been resolved: mm/filemap: skip to create PMD-sized page cach
CVE-2024-41034	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix kernel bug on rename operation of brok

CVE-2024-41036	In the Linux kernel, the following vulnerability has been resolved: net: ks8851: Fix deadlock with the SPI chip varia
CVE-2024-41035	In the Linux kernel, the following vulnerability has been resolved: USB: core: Fix duplicate endpoint bug by clearing
CVE-2024-41037	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda: fix null deref on system s
CVE-2024-41038	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Prevent buffer overrun when p
CVE-2024-41039	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix overflow checking of wmf
CVE-2024-41041	In the Linux kernel, the following vulnerability has been resolved: udp: Set SOCK_RCU_FREE earlier in udp_lib_g
CVE-2024-41040	In the Linux kernel, the following vulnerability has been resolved: net/sched: Fix UAF when resolving a clash KASA
CVE-2024-41042	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: prefer nft_chain_validate nft_
CVE-2024-41044	In the Linux kernel, the following vulnerability has been resolved: ppp: reject claimed-as-LCP but actually malforme
CVE-2024-41045	In the Linux kernel, the following vulnerability has been resolved: bpf: Defer work in bpf_timer_cancel_and_free C
CVE-2024-41046	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: lantiq_etop: fix double free in detach
CVE-2024-41047	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix XDP program unloading while removing
CVE-2024-41048	In the Linux kernel, the following vulnerability has been resolved: skmsg: Skip zero length skb in sk_msg_recvmmsg
CVE-2024-41049	In the Linux kernel, the following vulnerability has been resolved: filelock: fix potential use-after-free in posix_lock
CVE-2024-41050	In the Linux kernel, the following vulnerability has been resolved: cachefiles: cyclic allocation of msg_id to avoid re
CVE-2024-41055	In the Linux kernel, the following vulnerability has been resolved: mm: prevent dereferencing NULL ptr in pfn_section
CVE-2024-41056	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Use strlen() on name fields in
CVE-2024-41059	In the Linux kernel, the following vulnerability has been resolved: hfsplus: fix uninit-value in copy_name [syzbot rep
CVE-2024-41060	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: check bo_va->bo is non-NULL before
CVE-2024-41062	In the Linux kernel, the following vulnerability has been resolved: bluetooth/l2cap: sync sock recv cb and release Th
CVE-2024-41063	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: cancel all works upon hci_un
CVE-2024-41064	In the Linux kernel, the following vulnerability has been resolved: powerpc/eeh: avoid possible crash when edev->p
CVE-2024-41065	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Whitelist dtl slub object for copy
CVE-2024-41067	In the Linux kernel, the following vulnerability has been resolved: btrfs: scrub: handle RST lookup error correctly [E
CVE-2024-41068	In the Linux kernel, the following vulnerability has been resolved: s390/scslp: Fix scslp_init() cleanup on failure If scl
CVE-2024-41069	In the Linux kernel, the following vulnerability has been resolved: ASoC: topology: Fix references to freed memory
CVE-2024-41066	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Add tx check to prevent skb leak Below
CVE-2024-41070	In the Linux kernel, the following vulnerability has been resolved: KVM: PPC: Book3S HV: Prevent UAF in kvm_s
CVE-2024-41071	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-41072	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: wext: add extra SIOCSIWSCAN
CVE-2024-41074	In the Linux kernel, the following vulnerability has been resolved: cachefiles: Set object to close if ondemand_id < C
CVE-2024-41073	In the Linux kernel, the following vulnerability has been resolved: nvme: avoid double free special payload If a disc
CVE-2024-41075	In the Linux kernel, the following vulnerability has been resolved: cachefiles: add consistency check for copen/cread
CVE-2024-41076	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Fix memory leak in nfs4_set_security_lai
CVE-2024-41077	In the Linux kernel, the following vulnerability has been resolved: null_blk: fix validation of block size Block size sh
CVE-2024-41078	In the Linux kernel, the following vulnerability has been resolved: btrfs: qgroup: fix quota root leak after quota disab
CVE-2024-41079	In the Linux kernel, the following vulnerability has been resolved: nvmet: always initialize cq.e.result The spec does
CVE-2024-41080	In the Linux kernel, the following vulnerability has been resolved: io_uring: fix possible deadlock in io_register_iow
CVE-2024-41081	In the Linux kernel, the following vulnerability has been resolved: ila: block BH in ila_output() As explained in com
CVE-2024-41082	In the Linux kernel, the following vulnerability has been resolved: nvme-fabrics: use reserved tag for reg read/write

CVE-2024-41087	In the Linux kernel, the following vulnerability has been resolved: ata: libata-core: Fix double free on error If e.g. the
CVE-2024-41088	In the Linux kernel, the following vulnerability has been resolved: can: mcp251xfd: fix infinite loop when xmit fails
CVE-2024-41089	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/dispnv04: fix null pointer dereferen
CVE-2024-41090	In the Linux kernel, the following vulnerability has been resolved: tap: add missing verification for short frame The
CVE-2024-41091	In the Linux kernel, the following vulnerability has been resolved: tun: add missing verification for short frame The
CVE-2024-41092	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gt: Fix potential UAF by revoke of fenc
CVE-2024-41093	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: avoid using null object of framebuf
CVE-2024-41095	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/dispnv04: fix null pointer dereferen
CVE-2024-41097	In the Linux kernel, the following vulnerability has been resolved: usb: atm: cxacru: fix endpoint checking in cxacru
CVE-2024-41098	In the Linux kernel, the following vulnerability has been resolved: ata: libata-core: Fix null pointer dereference on er
CVE-2024-41110	Moby is an open-source project created by Docker for software containerization. A security vulnerability has been d
CVE-2024-41932	In the Linux kernel, the following vulnerability has been resolved: sched: fix warning in sched_setaffinity Commit 8
CVE-2024-41123	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.2 has some DoS vulnerabilities when it parses an
CVE-2024-41935	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to shrink read extent node in batches We
CVE-2024-41946	REXML is an XML toolkit for Ruby. The REXML gem 3.3.2 has a DoS vulnerability when it parses an XML that h
CVE-2024-41957	Vim is an open source command line text editor. Vim < v9.1.0647 has double free in src/alloc.c:616. When closing a
CVE-2024-41965	Vim is an open source command line text editor. double-free in dialog_changed() in Vim < v9.1.0648. When aband
CVE-2024-41996	Validating the order of the public keys in the Diffie-Hellman Key Agreement Protocol, when an approved safe prime
CVE-2024-42063	In the Linux kernel, the following vulnerability has been resolved: bpf: Mark bpf prog stack with kmsan_unposion_r
CVE-2024-42065	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Add a NULL check in xe_ttm_stolen_mg
CVE-2024-42067	In the Linux kernel, the following vulnerability has been resolved: bpf: Take return from set_memory_rox() into acc
CVE-2024-42069	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix possible double free in error handl
CVE-2024-42068	In the Linux kernel, the following vulnerability has been resolved: bpf: Take return from set_memory_ro() into acco
CVE-2024-42070	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fully validate NFT_DATA_V
CVE-2024-42076	In the Linux kernel, the following vulnerability has been resolved: net: can: j1939: Initialize unused data in j1939_se
CVE-2024-42077	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix DIO failure due to insufficient transacti
CVE-2024-42079	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix NULL pointer dereference in gfs2_log_
CVE-2024-42080	In the Linux kernel, the following vulnerability has been resolved: RDMA/restrack: Fix potential invalid address acc
CVE-2024-42082	In the Linux kernel, the following vulnerability has been resolved: xdp: Remove WARN() from __xdp_reg_mem_m
CVE-2024-42084	In the Linux kernel, the following vulnerability has been resolved: ftruncate: pass a signed offset The old ftruncate()
CVE-2024-42085	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: core: remove lock of otg mode during
CVE-2024-42086	In the Linux kernel, the following vulnerability has been resolved: iio: chemical: bme680: Fix overflows in compens
CVE-2024-42087	In the Linux kernel, the following vulnerability has been resolved: drm/panel: ilitek-ili9881c: Fix warning with GPI
CVE-2024-42089	In the Linux kernel, the following vulnerability has been resolved: ASoC: fsl-asoc-card: set priv->pdev before using
CVE-2024-42090	In the Linux kernel, the following vulnerability has been resolved: pinctrl: fix deadlock in create_pinctrl() when han
CVE-2024-42091	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Check pat.ops before dumping PAT setti
CVE-2024-42092	In the Linux kernel, the following vulnerability has been resolved: gpio: davinci: Validate the obtained number of IR
CVE-2024-42093	In the Linux kernel, the following vulnerability has been resolved: net/dpaa2: Avoid explicit cpumask var allocation
CVE-2024-42094	In the Linux kernel, the following vulnerability has been resolved: net/iucv: Avoid explicit cpumask var allocation o
CVE-2024-42095	In the Linux kernel, the following vulnerability has been resolved: serial: 8250_omap: Implementation of Errata i23

CVE-2024-42096	In the Linux kernel, the following vulnerability has been resolved: x86: stop playing stack games in profile_pc() The
CVE-2024-42097	In the Linux kernel, the following vulnerability has been resolved: ALSA: emux: improve patch ioctl data validation
CVE-2024-42098	In the Linux kernel, the following vulnerability has been resolved: crypto: ecdh - explicitly zeroize private_key priva
CVE-2024-42101	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: fix null pointer dereference in nouv
CVE-2024-42102	In the Linux kernel, the following vulnerability has been resolved: Revert "mm/writeback: fix possible divide-by-zer
CVE-2024-42104	In the Linux kernel, the following vulnerability has been resolved: nilfs2: add missing check for inode numbers on d
CVE-2024-42105	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix inode number range checks Patch serie
CVE-2024-42106	In the Linux kernel, the following vulnerability has been resolved: inet_diag: Initialize pad field in struct inet_diag_r
CVE-2024-42107	In the Linux kernel, the following vulnerability has been resolved: ice: Don't process extts if PTP is disabled The ice
CVE-2024-42109	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: unconditionally flush pending
CVE-2024-42114	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: restrict NL80211_ATTR_TXQ_C
CVE-2024-42110	In the Linux kernel, the following vulnerability has been resolved: net: ntb_netdev: Move ntb_netdev_rx_handler() t
CVE-2024-42116	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-42115	In the Linux kernel, the following vulnerability has been resolved: jffs2: Fix potential illegal address access in jffs2_
CVE-2024-42117	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: ASSERT when failing to find i
CVE-2024-42118	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Do not return negative stream i
CVE-2024-42119	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Skip finding free audio for unk
CVE-2024-42120	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check pipe offset before setting
CVE-2024-42121	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check index msg_id before rea
CVE-2024-42122	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL pointer check for k
CVE-2024-42124	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Make qedf_execute_tmf() non-preemp
CVE-2024-42125	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fw: scan offload prohibit all 6 GHz c
CVE-2024-42126	In the Linux kernel, the following vulnerability has been resolved: powerpc: Avoid nmi_enter/nmi_exit in real mode
CVE-2024-42127	In the Linux kernel, the following vulnerability has been resolved: drm/lima: fix shared irq handling on driver remov
CVE-2024-42128	In the Linux kernel, the following vulnerability has been resolved: leds: an30259a: Use devm_mutex_init() for mutex
CVE-2024-42129	In the Linux kernel, the following vulnerability has been resolved: leds: mlxreg: Use devm_mutex_init() for mutex i
CVE-2024-42130	In the Linux kernel, the following vulnerability has been resolved: nfc/nci: Add the inconsistency check between the
CVE-2024-42131	In the Linux kernel, the following vulnerability has been resolved: mm: avoid overflows in dirty throttling logic The
CVE-2024-42134	In the Linux kernel, the following vulnerability has been resolved: virtio-pci: Check if is_avq is NULL [bug] In the v
CVE-2024-42135	In the Linux kernel, the following vulnerability has been resolved: vhost_task: Handle SIGKILL by flushing work ar
CVE-2024-42136	In the Linux kernel, the following vulnerability has been resolved: cdrom: rearrange last_media_change check to av
CVE-2024-42137	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: qca: Fix BT enable failure again for Q
CVE-2024-42139	In the Linux kernel, the following vulnerability has been resolved: ice: Fix improper extts handling Extts events are e
CVE-2024-42140	In the Linux kernel, the following vulnerability has been resolved: riscv: kexec: Avoid deadlock in kexec crash path
CVE-2024-42145	In the Linux kernel, the following vulnerability has been resolved: IB/core: Implement a limit on UMAP receive Lis
CVE-2024-42147	In the Linux kernel, the following vulnerability has been resolved: crypto: hisilicon/debugfs - Fix debugfs uninit pro
CVE-2024-42148	In the Linux kernel, the following vulnerability has been resolved: bnx2x: Fix multiple UBSAN array-index-out-of-b
CVE-2024-42151	In the Linux kernel, the following vulnerability has been resolved: bpf: mark bpf_dummy_struct_ops.test_1 paramet
CVE-2024-42152	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix a possible leak when destroy a ctrl dur
CVE-2024-42153	In the Linux kernel, the following vulnerability has been resolved: i2c: pnx: Fix potential deadlock warning from del

CVE-2024-42154	In the Linux kernel, the following vulnerability has been resolved: tcp_metrics: validate source addr length I don't se
CVE-2024-42155	In the Linux kernel, the following vulnerability has been resolved: s390/pkey: Wipe copies of protected- and secure-
CVE-2024-42156	In the Linux kernel, the following vulnerability has been resolved: s390/pkey: Wipe copies of clear-key structures on
CVE-2024-42157	In the Linux kernel, the following vulnerability has been resolved: s390/pkey: Wipe sensitive data on failure Wipe se
CVE-2024-42158	In the Linux kernel, the following vulnerability has been resolved: s390/pkey: Use kfree_sensitive() to fix Coccinelle
CVE-2024-42159	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Sanitise num_phys Information is s
CVE-2024-42160	In the Linux kernel, the following vulnerability has been resolved: f2fs: check validation of fault attrs in f2fs_build_
CVE-2024-42161	In the Linux kernel, the following vulnerability has been resolved: bpf: Avoid uninitialized value in BPF_CORE_RE
CVE-2024-42223	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: tda10048: Fix integer over
CVE-2024-42224	In the Linux kernel, the following vulnerability has been resolved: net: dsa: mv88e6xxx: Correct check for empty lis
CVE-2024-42225	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: replace skb_put with skb_put_zero Av
CVE-2024-42226	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-42228	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Using uninitialized value *size whe
CVE-2024-42229	In the Linux kernel, the following vulnerability has been resolved: crypto: aead,cipher - zeroize key buffer after use
CVE-2024-42230	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Fix scv instruction crash with ke
CVE-2024-42236	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: configs: Prevent OOB read/write in
CVE-2024-42232	In the Linux kernel, the following vulnerability has been resolved: libceph: fix race between delayed_work() and cep
CVE-2024-42237	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Validate payload length before
CVE-2024-42238	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Return error if block header ov
CVE-2024-42239	In the Linux kernel, the following vulnerability has been resolved: bpf: Fail bpf_timer_cancel when callback is being
CVE-2024-42240	In the Linux kernel, the following vulnerability has been resolved: x86/bhi: Avoid warning in #DB handler due to BI
CVE-2024-42244	In the Linux kernel, the following vulnerability has been resolved: USB: serial: mos7840: fix crash on resume Since
CVE-2024-42245	In the Linux kernel, the following vulnerability has been resolved: Revert "sched/fair: Make sure to try to detach at l
CVE-2024-42247	In the Linux kernel, the following vulnerability has been resolved: wireguard: allowedips: avoid unaligned 64-bit me
CVE-2024-42246	In the Linux kernel, the following vulnerability has been resolved: net, sunrpc: Remap EPERM in case of connection
CVE-2024-42252	In the Linux kernel, the following vulnerability has been resolved: closures: Change BUG_ON() to WARN_ON() If
CVE-2024-42253	In the Linux kernel, the following vulnerability has been resolved: gpio: pca953x: fix pca953x_irq_bus_sync_unlock
CVE-2024-42259	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gem: Fix Virtual Memory mapping bou
CVE-2024-42265	In the Linux kernel, the following vulnerability has been resolved: protect the fetch of ->fd[fd] in do_dup2() from m
CVE-2024-42267	In the Linux kernel, the following vulnerability has been resolved: riscv/mm: Add handling for VM_FAULT_SIGSE
CVE-2024-42269	In the Linux kernel, the following vulnerability has been resolved: netfilter: iptables: Fix potential null-ptr-deref in ip
CVE-2024-42268	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix missing lock on sync reset reload C
CVE-2024-42270	In the Linux kernel, the following vulnerability has been resolved: netfilter: iptables: Fix null-ptr-deref in iptable_na
CVE-2024-42271	In the Linux kernel, the following vulnerability has been resolved: net/iucv: fix use after free in iucv_sock_close() iu
CVE-2024-42272	In the Linux kernel, the following vulnerability has been resolved: sched: act_ct: take care of padding in struct zones
CVE-2024-42273	In the Linux kernel, the following vulnerability has been resolved: f2fs: assign CURSEG_ALL_DATA_ATGC if bl
CVE-2024-42274	In the Linux kernel, the following vulnerability has been resolved: Revert "ALSA: firewire-lib: operate for period el
CVE-2024-42276	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: add missing condition check for existe
CVE-2024-42277	In the Linux kernel, the following vulnerability has been resolved: iommu: sprd: Avoid NULL deref in sprd_iommu
CVE-2024-42283	In the Linux kernel, the following vulnerability has been resolved: net: nexthop: Initialize all fields in dumped nexth

CVE-2024-42281	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix a segment issue when downgrading gso_
CVE-2024-42280	In the Linux kernel, the following vulnerability has been resolved: mISDN: Fix a use after free in hfcmulti_tx() Don't
CVE-2024-42284	In the Linux kernel, the following vulnerability has been resolved: tipc: Return non-zero value from tipc_udp_addr2
CVE-2024-42285	In the Linux kernel, the following vulnerability has been resolved: RDMA/iwcm: Fix a use-after-free related to destr
CVE-2024-42288	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix for possible memory corruption
CVE-2024-42286	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: validate nvme_local_port correctly
CVE-2024-42287	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Complete command early within lo
CVE-2024-42289	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: During vport delete send async log
CVE-2024-42290	In the Linux kernel, the following vulnerability has been resolved: irqchip/imx-irqsteer: Handle runtime power mana
CVE-2024-42295	In the Linux kernel, the following vulnerability has been resolved: nilfs2: handle inconsistent state in nilfs_bnode_c
CVE-2024-42292	In the Linux kernel, the following vulnerability has been resolved: kobject_uevent: Fix OOB access within zap_mod
CVE-2024-42291	In the Linux kernel, the following vulnerability has been resolved: ice: Add a per-VF limit on number of FDIR filter
CVE-2024-42296	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix return value of f2fs_convert_inline_inod
CVE-2024-42297	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to don't dirty inode for readonly filesystem
CVE-2024-42299	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Update log->page_{mask,bits} if log->p
CVE-2024-42302	In the Linux kernel, the following vulnerability has been resolved: PCI/DPC: Fix use-after-free on concurrent DPC a
CVE-2024-42301	In the Linux kernel, the following vulnerability has been resolved: dev/parport: fix the array out-of-bounds risk Fixe
CVE-2024-42304	In the Linux kernel, the following vulnerability has been resolved: ext4: make sure the first directory block is not a h
CVE-2024-42305	In the Linux kernel, the following vulnerability has been resolved: ext4: check dot and dotdot of dx_root before mak
CVE-2024-42306	In the Linux kernel, the following vulnerability has been resolved: udf: Avoid using corrupted block bitmap buffer V
CVE-2024-42310	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: fix null pointer dereference in cdv_
CVE-2024-42309	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: fix null pointer dereference in psb_
CVE-2024-42311	In the Linux kernel, the following vulnerability has been resolved: hfs: fix to initialize fields of hfs_inode_info after
CVE-2024-42312	In the Linux kernel, the following vulnerability has been resolved: sysctl: always initialize i_uid/i_gid Always initial
CVE-2024-42313	In the Linux kernel, the following vulnerability has been resolved: media: venus: fix use after free in vdec_close The
CVE-2024-42318	In the Linux kernel, the following vulnerability has been resolved: landlock: Don't lose track of restrictions on cred_
CVE-2024-42315	In the Linux kernel, the following vulnerability has been resolved: exfat: fix potential deadlock on __exfat_get_dentr
CVE-2024-42320	In the Linux kernel, the following vulnerability has been resolved: s390/dasd: fix error checks in dasd_copy_pair_sto
CVE-2024-42319	In the Linux kernel, the following vulnerability has been resolved: mailbox: mtk-cmdq: Move devm_mbox_controlle
CVE-2024-42321	In the Linux kernel, the following vulnerability has been resolved: net: flow_dissector: use DEBUG_NET_WARN_
CVE-2024-42322	In the Linux kernel, the following vulnerability has been resolved: ipv6: properly dereference pe in ip_vs_add_servic
CVE-2024-42367	aiohttp is an asynchronous HTTP client/server framework for asyncio and Python. In versions on the 3.10 branch pri
CVE-2024-42516	HTTP response splitting in the core of Apache HTTP Server allows an attacker who can manipulate the Content-Typ
CVE-2024-43098	In the Linux kernel, the following vulnerability has been resolved: i3c: Use i3cdev->desc->info instead of calling i3c
CVE-2024-43204	SSRF in Apache HTTP Server with mod_proxy loaded allows an attacker to send outbound proxy requests to a URL
CVE-2024-43374	The UNIX editor Vim prior to version 9.1.0678 has a use-after-free error in argument list handling. When adding a n
CVE-2024-43382	Snowflake JDBC driver versions >= 3.2.6 and <= 3.19.1 have an Incorrect Security Setting that can result in data be
CVE-2024-43398	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.6 has a DoS vulnerability when it parses an XML
CVE-2024-43598	LightGBM Remote Code Execution Vulnerability
CVE-2024-43799	Send is a library for streaming files from the file system as a http response. Send passes untrusted user input to Send

CVE-2024-43800	serve-static serves static files. serve-static passes untrusted user input - even after sanitizing it - to redirect() may exe
CVE-2024-43802	Vim is an improved version of the unix vi text editor. When flushing the typeahead buffer, Vim moves the current po
CVE-2024-43805	jupyterlab is an extensible environment for interactive and reproducible computing, based on the Jupyter Notebook A
CVE-2024-43817	In the Linux kernel, the following vulnerability has been resolved: net: missing check virtio Two missing check in vi
CVE-2024-43819	In the Linux kernel, the following vulnerability has been resolved: kvm: s390: Reject memory region operations for
CVE-2024-43823	In the Linux kernel, the following vulnerability has been resolved: PCI: keystone: Fix NULL pointer dereference in
CVE-2024-43821	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix a possible null pointer dereference
CVE-2024-43824	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: pci-epf-test: Make use of cached 'e
CVE-2024-43829	In the Linux kernel, the following vulnerability has been resolved: drm/qxl: Add check for drm_cvt_mode Add chec
CVE-2024-43828	In the Linux kernel, the following vulnerability has been resolved: ext4: fix infinite loop when replaying fast_comm
CVE-2024-43830	In the Linux kernel, the following vulnerability has been resolved: leds: trigger: Unregister sysfs attributes before ca
CVE-2024-43831	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Handle invalid decoder
CVE-2024-43832	In the Linux kernel, the following vulnerability has been resolved: s390/uv: Don't call folio_wait_writeback() withou
CVE-2024-43834	In the Linux kernel, the following vulnerability has been resolved: xdp: fix invalid wait context of page_pool_destro
CVE-2024-43839	In the Linux kernel, the following vulnerability has been resolved: bna: adjust 'name' buf size of bna_tcb and bna_cc
CVE-2024-43835	In the Linux kernel, the following vulnerability has been resolved: virtio_net: Fix napi_skb_cache_put warning After
CVE-2024-43841	In the Linux kernel, the following vulnerability has been resolved: wifi: virt_wifi: avoid reporting connection succes
CVE-2024-43842	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: Fix array index mistake in rtw89_sta
CVE-2024-43844	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: wow: fix GTK offload H2C skbuff is
CVE-2024-43846	In the Linux kernel, the following vulnerability has been resolved: lib: objagg: Fix general protection fault The libr
CVE-2024-43849	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: pdr: protect locator_addr with the mai
CVE-2024-43853	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: Prevent UAF in proc_cpuset_show
CVE-2024-43854	In the Linux kernel, the following vulnerability has been resolved: block: initialize integrity buffer to zero before wr
CVE-2024-43856	In the Linux kernel, the following vulnerability has been resolved: dma: fix call order in dmam_free_coherent dmam
CVE-2024-43860	In the Linux kernel, the following vulnerability has been resolved: remoteproc: imx_rproc: Skip over memory region
CVE-2024-43858	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix array-index-out-of-bounds in diFree
CVE-2024-43861	In the Linux kernel, the following vulnerability has been resolved: net: usb: qmi_wwan: fix memory leak for not ip p
CVE-2024-43863	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Fix a deadlock in dma buf fence po
CVE-2024-43866	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Always drain health in shutdown callba
CVE-2024-43867	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: prime: fix refcount underflow Calli
CVE-2024-43869	In the Linux kernel, the following vulnerability has been resolved: perf: Fix event leak upon exec and file release Th
CVE-2024-43870	In the Linux kernel, the following vulnerability has been resolved: perf: Fix event leak upon exit When a task is sche
CVE-2024-43871	In the Linux kernel, the following vulnerability has been resolved: devres: Fix memory leakage caused by driver AP
CVE-2024-43872	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix soft lockup under heavy CEQE l
CVE-2024-43875	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: Clean up error handling in vpci_sc
CVE-2024-43873	In the Linux kernel, the following vulnerability has been resolved: vhost/vsock: always initialize seqpacket_allow TL
CVE-2024-43879	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: handle 2x996 RU allocation in cf
CVE-2024-43880	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_acl_erp: Fix object nesting war
CVE-2024-43882	In the Linux kernel, the following vulnerability has been resolved: exec: Fix ToCToU between perm check and set-u
CVE-2024-43883	In the Linux kernel, the following vulnerability has been resolved: usb: vhci-hcd: Do not drop references before new

CVE-2024-43884	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Add error handling to pair_de
CVE-2024-43889	In the Linux kernel, the following vulnerability has been resolved: padata: Fix possible divide-by-0 panic in padata_
CVE-2024-43890	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix overflow in get_free_elt() "tracing_m
CVE-2024-43892	In the Linux kernel, the following vulnerability has been resolved: memcg: protect concurrent access to mem_cgrou
CVE-2024-43893	In the Linux kernel, the following vulnerability has been resolved: serial: core: check uartclk for zero to avoid divide
CVE-2024-43894	In the Linux kernel, the following vulnerability has been resolved: drm/client: fix null pointer dereference in drm_cli
CVE-2024-43898	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-43899	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null pointer deref in dcn20_
CVE-2024-43900	In the Linux kernel, the following vulnerability has been resolved: media: xc2028: avoid use-after-free in load_firmw
CVE-2024-43902	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null checker before passing
CVE-2024-43904	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null checks for 'stream' and
CVE-2024-43905	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Fix the null pointer dereference for
CVE-2024-43907	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm: Fix the null pointer dereference
CVE-2024-43906	In the Linux kernel, the following vulnerability has been resolved: drm/admgpu: fix dereferencing null pointer conte
CVE-2024-43908	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix the null pointer dereference to n
CVE-2024-43909	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/pm: Fix the null pointer dereference
CVE-2024-43910	In the Linux kernel, the following vulnerability has been resolved: bpf: add missing check_func_arg_reg_off() to pre
CVE-2024-43911	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix NULL dereference at band cl
CVE-2024-43912	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: disallow setting special AP channel
CVE-2024-43914	In the Linux kernel, the following vulnerability has been resolved: md/raid5: avoid BUG_ON() while continue resha
CVE-2024-44931	In the Linux kernel, the following vulnerability has been resolved: gpio: prevent potential speculation leaks in gpio_
CVE-2024-44932	In the Linux kernel, the following vulnerability has been resolved: idpf: fix UAFs when destroying the queues The s
CVE-2024-44935	In the Linux kernel, the following vulnerability has been resolved: sctp: Fix null-ptr-deref in reuseport_add_sock(). s
CVE-2024-44938	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix shift-out-of-bounds in dbDiscardAG Whe
CVE-2024-44934	In the Linux kernel, the following vulnerability has been resolved: net: bridge: mcast: wait for previous gc cycles wh
CVE-2024-44939	In the Linux kernel, the following vulnerability has been resolved: jfs: fix null ptr deref in dtInsertEntry [syzbot repo
CVE-2024-44940	In the Linux kernel, the following vulnerability has been resolved: fou: remove warn in gue_gro_receive on unsuppo
CVE-2024-44942	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on F2FS_INLINE_DA
CVE-2024-44947	In the Linux kernel, the following vulnerability has been resolved: fuse: Initialize beyond-EOF page contents before
CVE-2024-44944	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: use helper function to calculat
CVE-2024-44948	In the Linux kernel, the following vulnerability has been resolved: x86/mtrr: Check if fixed MTRRs exist before sav
CVE-2024-44946	In the Linux kernel, the following vulnerability has been resolved: kcm: Serialise kcm_sendmsg() for the same socke
CVE-2024-44950	In the Linux kernel, the following vulnerability has been resolved: serial: sc16is7xx: fix invalid FIFO access with sp
CVE-2024-44954	In the Linux kernel, the following vulnerability has been resolved: ALSA: line6: Fix racy access to midibuf There ca
CVE-2024-44951	In the Linux kernel, the following vulnerability has been resolved: serial: sc16is7xx: fix TX fifo corruption Sometim
CVE-2024-44957	In the Linux kernel, the following vulnerability has been resolved: xen: privcmd: Switch from mutex to spinlock for
CVE-2024-44952	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-44958	In the Linux kernel, the following vulnerability has been resolved: sched/smt: Fix unbalance sched_smt_present decr
CVE-2024-44961	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Forward soft recovery errors to use
CVE-2024-44960	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: core: Check for unset descriptor Mak

CVE-2024-44963	In the Linux kernel, the following vulnerability has been resolved: btrfs: do not BUG_ON() when freeing tree block
CVE-2024-44962	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btmxpurt: Shutdown timer and preven
CVE-2024-44964	In the Linux kernel, the following vulnerability has been resolved: idpf: fix memory leaks and crashes while perform
CVE-2024-44966	In the Linux kernel, the following vulnerability has been resolved: binfmt_flat: Fix corruption when not offsetting da
CVE-2024-44965	In the Linux kernel, the following vulnerability has been resolved: x86/mm: Fix pti_clone_pgtable() alignment assur
CVE-2024-44967	In the Linux kernel, the following vulnerability has been resolved: drm/mgag200: Bind I2C lifetime to DRM device
CVE-2024-44969	In the Linux kernel, the following vulnerability has been resolved: s390/sclp: Prevent release of buffer in I/O When a
CVE-2024-44971	In the Linux kernel, the following vulnerability has been resolved: net: dsa: bcm_sf2: Fix a possible memory leak in
CVE-2024-44970	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: SHAMPO, Fix invalid WQ linked list
CVE-2024-44972	In the Linux kernel, the following vulnerability has been resolved: btrfs: do not clear page dirty inside extent_write_
CVE-2024-44977	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Validate TA binary size Add TA bi
CVE-2024-44974	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: avoid possible UaF when selecting en
CVE-2024-44982	In the Linux kernel, the following vulnerability has been resolved: drm/msm/dpu: cleanup FB if dpu_format_popular
CVE-2024-44983	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable: validate vlan header Ensure th
CVE-2024-44985	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent possible UAF in ip6_xmit() If skb_
CVE-2024-44986	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix possible UAF in ip6_finish_output2() If
CVE-2024-44987	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent UAF in ip6_send_skb() syzbot repo
CVE-2024-44988	In the Linux kernel, the following vulnerability has been resolved: net: dsa: mv88e6xxx: Fix out-of-bound access If a
CVE-2024-44989	In the Linux kernel, the following vulnerability has been resolved: bonding: fix xfrm real_dev null pointer dereferen
CVE-2024-44990	In the Linux kernel, the following vulnerability has been resolved: bonding: fix null pointer deref in bond_ipsec_offl
CVE-2024-44998	In the Linux kernel, the following vulnerability has been resolved: atm: idt77252: prevent use after free in dequeue_
CVE-2024-44995	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix a deadlock problem when config TC
CVE-2024-44999	In the Linux kernel, the following vulnerability has been resolved: gtp: pull network headers in gtp_dev_xmit() syzb
CVE-2024-45001	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix RX buf alloc_size alignment and a
CVE-2024-45003	In the Linux kernel, the following vulnerability has been resolved: vfs: Don't evict inode under the inode lru traversi
CVE-2024-45005	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: fix validity interception issue when g
CVE-2024-45006	In the Linux kernel, the following vulnerability has been resolved: xhci: Fix Panther point NULL pointer deref at full
CVE-2024-45007	In the Linux kernel, the following vulnerability has been resolved: char: xillybus: Don't destroy workqueue from wor
CVE-2024-45008	In the Linux kernel, the following vulnerability has been resolved: Input: MT - limit max slots syzbot is reporting too
CVE-2024-45009	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: only decrement add_addr_accepted fo
CVE-2024-45011	In the Linux kernel, the following vulnerability has been resolved: char: xillybus: Check USB endpoints when probin
CVE-2024-45010	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: only mark 'subflow' endp as available
CVE-2024-45012	In the Linux kernel, the following vulnerability has been resolved: nouveau/firmware: use dma non-coherent allocato
CVE-2024-45015	In the Linux kernel, the following vulnerability has been resolved: drm/msm/dpu: move dpu_encoder's connector ass
CVE-2024-45016	In the Linux kernel, the following vulnerability has been resolved: netem: fix return value if duplicate enqueue fails
CVE-2024-45018	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable: initialise extack before use Fi
CVE-2024-45019	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Take state lock during tx timeout repo
CVE-2024-45021	In the Linux kernel, the following vulnerability has been resolved: memcg_write_event_control(): fix a user-triggera
CVE-2024-45025	In the Linux kernel, the following vulnerability has been resolved: fix bitmap corruption on close_range() with CLO
CVE-2024-45026	In the Linux kernel, the following vulnerability has been resolved: s390/dasd: fix error recovery leading to data corru

CVE-2024-45028	In the Linux kernel, the following vulnerability has been resolved: mmc: mmc_test: Fix NULL dereference on allocat
CVE-2024-45033	Insufficient Session Expiration vulnerability in Apache Airflow Fab Provider. This issue affects Apache Airflow Fab
CVE-2024-45231	An issue was discovered in Django v5.1.1, v5.0.9, and v4.2.16. The django.contrib.auth.forms.PasswordResetForm c
CVE-2024-45296	path-to-regexp turns path strings into a regular expressions. In certain cases, path-to-regexp will output a regular exp
CVE-2024-45306	Vim is an open source, command line text editor. Patch v9.1.0038 optimized how the cursor position is calculated an
CVE-2024-45310	runc is a CLI tool for spawning and running containers according to the OCI specification. runc 1.1.13 and earlier, as
CVE-2024-45336	The HTTP client drops sensitive headers after following a cross-domain redirect. For example, a request to a.com/ co
CVE-2024-45337	Applications and libraries which misuse connection.serverAuthenticate (via callback field ServerConfig.PublicKeyC
CVE-2024-45338	An attacker can craft an input to the Parse functions that would be processed non-linearly with respect to its length, r
CVE-2024-45339	When logs are written to a widely-writable directory (the default), an unprivileged attacker may predict a privileged
CVE-2024-45341	A certificate with a URI which has a IPv6 address with a zone ID may incorrectly satisfy a URI name constraint that
CVE-2024-45490	An issue was discovered in libexpat before 2.6.3. xmlparse.c does not reject a negative length for XML_ParseBuffer
CVE-2024-45491	An issue was discovered in libexpat before 2.6.3. dtdCopy in xmlparse.c can have an integer overflow for nDefaultA
CVE-2024-45492	An issue was discovered in libexpat before 2.6.3. nextScaffoldPart in xmlparse.c can have an integer overflow for m
CVE-2024-45828	In the Linux kernel, the following vulnerability has been resolved: i3c: mipi-i3c-hci: Mask ring interrupts before ring
CVE-2024-4603	Issue summary: Checking excessively long DSA keys or parameters may be very slow. Impact summary: Application
CVE-2024-46672	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: cfg80211: Handle SSID based pr
CVE-2024-46673	In the Linux kernel, the following vulnerability has been resolved: scsi: aacraid: Fix double-free on probe failure aac
CVE-2024-46675	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: core: Prevent USB core invalid event
CVE-2024-46676	In the Linux kernel, the following vulnerability has been resolved: nfc: pn533: Add poll mod list filling check In cas
CVE-2024-46677	In the Linux kernel, the following vulnerability has been resolved: gtp: fix a potential NULL pointer dereference Wh
CVE-2024-46678	In the Linux kernel, the following vulnerability has been resolved: bonding: change ipsec_lock from spin lock to mu
CVE-2024-46679	In the Linux kernel, the following vulnerability has been resolved: ethtool: check device is present when getting link
CVE-2024-46681	In the Linux kernel, the following vulnerability has been resolved: pktgen: use cpus_read_lock() in pg_net_init() I ha
CVE-2024-46685	In the Linux kernel, the following vulnerability has been resolved: pinctrl: single: fix potential NULL dereference in
CVE-2024-46689	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: cmd-db: Map shared memory as WC,
CVE-2024-46695	In the Linux kernel, the following vulnerability has been resolved: selinux,smack: don't bypass permissions check in
CVE-2024-46702	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Mark XDomain as unplugged when
CVE-2024-46705	In the Linux kernel, the following vulnerability has been resolved: drm/xe: reset mmio mappings with devm Set our
CVE-2024-46707	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Make ICC_*SGI*_EL1 undef in th
CVE-2024-46710	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Prevent unmapping active read buf
CVE-2024-46713	In the Linux kernel, the following vulnerability has been resolved: perf/aux: Fix AUX buffer serialization Ole report
CVE-2024-46714	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Skip wbscl_set_scaler_filter if
CVE-2024-46715	In the Linux kernel, the following vulnerability has been resolved: driver: iio: add missing checks on iio_info's callb
CVE-2024-46716	In the Linux kernel, the following vulnerability has been resolved: dmaengine: altera-msgdma: properly free descrip
CVE-2024-46718	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Don't overmap identity VRAM mapping
CVE-2024-46717	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: SHAMPO, Fix incorrect page release
CVE-2024-46719	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Fix null pointer dereference in tr
CVE-2024-46720	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix dereference after null check che
CVE-2024-46721	In the Linux kernel, the following vulnerability has been resolved: apparmor: fix possible NULL pointer dereference

CVE-2024-46722	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix mc_data out-of-bounds read warning
CVE-2024-46723	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix ucode out-of-bounds read warning
CVE-2024-46724	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix out-of-bounds read of df_v1_7_0
CVE-2024-46725	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix out-of-bounds write warning Cl
CVE-2024-46726	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Ensure index calculation will n
CVE-2024-46727	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add otg_master NULL check v
CVE-2024-46728	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check index for aux_rd_interv
CVE-2024-46729	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix incorrect size calculation fo
CVE-2024-46730	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Ensure array index tg_inst won
CVE-2024-46731	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: fix the Out-of-bounds read warning
CVE-2024-46732	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Assign linear_pitch_alignment
CVE-2024-46733	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix qgroup reserve leaks in cow_file_range
CVE-2024-46737	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: fix kernel crash if commands allocatio
CVE-2024-46738	In the Linux kernel, the following vulnerability has been resolved: VMCI: Fix use-after-free when removing resource
CVE-2024-46739	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Fix kernel NULL pointer derefer
CVE-2024-46740	In the Linux kernel, the following vulnerability has been resolved: binder: fix UAF caused by offsets overwrite Bind
CVE-2024-46742	In the Linux kernel, the following vulnerability has been resolved: smb/server: fix potential null-ptr-deref of lease_c
CVE-2024-46743	In the Linux kernel, the following vulnerability has been resolved: of/irq: Prevent device address out-of-bounds read
CVE-2024-46744	In the Linux kernel, the following vulnerability has been resolved: Squashfs: sanity check symbolic link size Syzkill
CVE-2024-46745	In the Linux kernel, the following vulnerability has been resolved: Input: uinput - reject requests with unreasonable r
CVE-2024-46746	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: free driver_data after destroying h
CVE-2024-46747	In the Linux kernel, the following vulnerability has been resolved: HID: cougar: fix slab-out-of-bounds Read in coug
CVE-2024-46748	In the Linux kernel, the following vulnerability has been resolved: cachefiles: Set the max subreq size for cache writ
CVE-2024-46749	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btmxpuart: Fix Null pointer dereferenc
CVE-2024-46752	In the Linux kernel, the following vulnerability has been resolved: btrfs: replace BUG_ON() with error handling at u
CVE-2024-46750	In the Linux kernel, the following vulnerability has been resolved: PCI: Add missing bridge lock to pci_bus_lock() C
CVE-2024-46753	In the Linux kernel, the following vulnerability has been resolved: btrfs: handle errors from btrfs_dec_ref() properly
CVE-2024-46754	In the Linux kernel, the following vulnerability has been resolved: bpf: Remove tst_run from lwt_seg6local_prog_op
CVE-2024-46755	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Do not return unused priv in mwifi
CVE-2024-46758	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-46756	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-46759	In the Linux kernel, the following vulnerability has been resolved: hwmon: (adc128d818) Fix underflows seen when
CVE-2024-46760	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: usb: schedule rx work after everything
CVE-2024-46761	In the Linux kernel, the following vulnerability has been resolved: pci/hotplug/pnv_php: Fix hotplug driver crash on
CVE-2024-46762	In the Linux kernel, the following vulnerability has been resolved: xen: privcmd: Fix possible access to a freed kirqf
CVE-2024-46763	In the Linux kernel, the following vulnerability has been resolved: fou: Fix null-ptr-deref in GRO. We observed a nu
CVE-2024-46765	In the Linux kernel, the following vulnerability has been resolved: ice: protect XDP configuration with a mutex The
CVE-2024-46770	In the Linux kernel, the following vulnerability has been resolved: ice: Add netif_device_attach/detach into PF reset
CVE-2024-46771	In the Linux kernel, the following vulnerability has been resolved: can: bcm: Remove proc entry when dev is unregist
CVE-2024-46774	In the Linux kernel, the following vulnerability has been resolved: powerpc/rtas: Prevent Spectre v1 gadget construct

CVE-2024-46775	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Validate function returns [WHA
CVE-2024-46777	In the Linux kernel, the following vulnerability has been resolved: udf: Avoid excessive partition lengths Avoid mou
CVE-2024-46778	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check UnboundedRequestEnab
CVE-2024-46780	In the Linux kernel, the following vulnerability has been resolved: nilfs2: protect references to superblock parameter
CVE-2024-46781	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix missing cleanup on rollforward recover
CVE-2024-46782	In the Linux kernel, the following vulnerability has been resolved: ila: call nf_unregister_net_hooks() sooner syzbot
CVE-2024-46783	In the Linux kernel, the following vulnerability has been resolved: tcp_bpf: fix return value of tcp_bpf_sendmsg() W
CVE-2024-46784	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix error handling in mana_create_txq
CVE-2024-46787	In the Linux kernel, the following vulnerability has been resolved: userfaultfd: fix checks for huge PMDs Patch serie
CVE-2024-46788	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Use a cpumask to know what thr
CVE-2024-46791	In the Linux kernel, the following vulnerability has been resolved: can: mcp251x: fix deadlock if an interrupt occurs
CVE-2024-46794	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Fix data leak in mmio_read() The mmio
CVE-2024-46795	In the Linux kernel, the following vulnerability has been resolved: ksmbd: unset the binding mark of a reused connec
CVE-2024-46798	In the Linux kernel, the following vulnerability has been resolved: ASoC: dapm: Fix UAF for snd_soc_pcm_runtime
CVE-2024-46800	In the Linux kernel, the following vulnerability has been resolved: sch/netem: fix use after free in netem_dequeue If
CVE-2024-46802	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: added NULL check at start of d
CVE-2024-46803	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Check debug trap enable before wri
CVE-2024-46804	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add array index check for hdcp
CVE-2024-46805	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix the waring dereferencing hive C
CVE-2024-46806	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix the warning division or modulor
CVE-2024-46807	In the Linux kernel, the following vulnerability has been resolved: drm/amd/amdgpu: Check tbo resource pointer Va
CVE-2024-46808	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add missing NULL pointer che
CVE-2024-46809	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check BIOS images before it is
CVE-2024-46810	In the Linux kernel, the following vulnerability has been resolved: drm/bridge: tc358767: Check if fully initialized b
CVE-2024-46811	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index may exceed array ran
CVE-2024-46812	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Skip inactive planes within Mo
CVE-2024-46814	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check msg_id before processin
CVE-2024-46813	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check link_index before access
CVE-2024-46815	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check num_valid_sets before a
CVE-2024-46817	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Stop amdgpu_dm initialize whe
CVE-2024-46818	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check gpio_id before used as a
CVE-2024-46820	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/vcn: remove irq disabling in vcn 5 s
CVE-2024-46819	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: the warning dereferencing obj for n
CVE-2024-46821	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Fix negative array index read Avoi
CVE-2024-46822	In the Linux kernel, the following vulnerability has been resolved: arm64: acpi: Harden get_cpu_for_acpi_id() again
CVE-2024-46823	In the Linux kernel, the following vulnerability has been resolved: kunit/overflow: Fix UB in overflow_allocation_te
CVE-2024-46825	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: mvmm: use IWL_FW_CHECK for li
CVE-2024-46826	In the Linux kernel, the following vulnerability has been resolved: ELF: fix kernel.randomize_va_space double read
CVE-2024-46827	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix firmware crash due to invalid pe
CVE-2024-46828	In the Linux kernel, the following vulnerability has been resolved: sched: sch_cake: fix bulk flow accounting logic f

CVE-2024-46829	In the Linux kernel, the following vulnerability has been resolved: rtmutex: Drop rt_mutex::wait_lock before schedu
CVE-2024-46830	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Acquire kvm->srcu when handling K
CVE-2024-46832	In the Linux kernel, the following vulnerability has been resolved: MIPS: cevt-r4k: Don't call get_c0_compare_int if
CVE-2024-46833	In the Linux kernel, the following vulnerability has been resolved: net: hns3: void array out of bound when loop tnl_
CVE-2024-46834	In the Linux kernel, the following vulnerability has been resolved: ethtool: fail closed if we can't get max channel us
CVE-2024-46835	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix smatch static checker warning a
CVE-2024-46836	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: aspeed_udc: validate endpoint index
CVE-2024-46839	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-46840	In the Linux kernel, the following vulnerability has been resolved: btrfs: clean up our handling of refs == 0 in snaps
CVE-2024-46841	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't BUG_ON on ENOMEM from btrfs_lo
CVE-2024-46842	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Handle mailbox timeouts in lpfc_get_st
CVE-2024-46844	In the Linux kernel, the following vulnerability has been resolved: um: line: always fill *error_out in setup_one_line
CVE-2024-46843	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Remove SCSI host only if added I
CVE-2024-46845	In the Linux kernel, the following vulnerability has been resolved: tracing/timerlat: Only clear timer if a kthread exists
CVE-2024-46848	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: Limit the period on Haswell Runni
CVE-2024-46849	In the Linux kernel, the following vulnerability has been resolved: ASoC: meson: axg-card: fix 'use-after-free' Buffer
CVE-2024-46852	In the Linux kernel, the following vulnerability has been resolved: dma-buf: heaps: Fix off-by-one in CMA heap fault
CVE-2024-46853	In the Linux kernel, the following vulnerability has been resolved: spi: nxp-fspi: fix the KASAN report out-of-bound
CVE-2024-46854	In the Linux kernel, the following vulnerability has been resolved: net: dpaa: Pad packets to ETH_ZLEN When send
CVE-2024-46855	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_socket: fix sk refcount leaks We mus
CVE-2024-46857	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix bridge mode operations when there
CVE-2024-46858	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: Fix uaf in __timer_delete_sync There
CVE-2024-46859	In the Linux kernel, the following vulnerability has been resolved: platform/x86: panasonic-laptop: Fix SINF array o
CVE-2024-46860	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921: fix NULL pointer access in m
CVE-2024-46864	In the Linux kernel, the following vulnerability has been resolved: x86/hyperv: fix kexec crash due to VP assist page
CVE-2024-46861	In the Linux kernel, the following vulnerability has been resolved: usbnet: ipheth: do not stop RX on failing RX call
CVE-2024-46870	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Disable DMCUB timeout for D
CVE-2024-46865	In the Linux kernel, the following vulnerability has been resolved: fou: fix initialization of grc The grc must be initial
CVE-2024-46871	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Correct the defined value for A
CVE-2024-47081	Requests is a HTTP library. Due to a URL parsing issue, Requests releases prior to 2.32.4 may leak .netrc credential
CVE-2024-47141	In the Linux kernel, the following vulnerability has been resolved: pinmux: Use sequential access to access desc->pi
CVE-2024-47143	In the Linux kernel, the following vulnerability has been resolved: dma-debug: fix a possible deadlock on radix_lock
CVE-2024-47175	CUPS is a standards-based, open-source printing system, and `libppd` can be used for legacy PPD file support. The
CVE-2024-47220	An issue was discovered in the WEBrick toolkit through 1.8.1 for Ruby. It allows HTTP request smuggling by provi
CVE-2024-47252	Insufficient escaping of user-supplied data in mod_ssl in Apache HTTP Server 2.4.63 and earlier allows an untrusted
CVE-2024-47408	In the Linux kernel, the following vulnerability has been resolved: net/smc: check smcd_v2_ext_offset when receivi
CVE-2024-47535	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high p
CVE-2024-4741	Issue summary: Calling the OpenSSL API function SSL_free_buffers may cause memory to be accessed that was pr
CVE-2024-47554	Uncontrolled Resource Consumption vulnerability in Apache Commons IO. The org.apache.commons.io.input.XmlIS
CVE-2024-47561	Schema parsing in the Java SDK of Apache Avro 1.11.3 and previous versions allows bad actors to execute arbitrary

CVE-2024-47658	In the Linux kernel, the following vulnerability has been resolved: crypto: stm32/cryp - call finalize with bh disabled
CVE-2024-47659	In the Linux kernel, the following vulnerability has been resolved: smack: tcp: ipv4, fix incorrect labeling Currently,
CVE-2024-47660	In the Linux kernel, the following vulnerability has been resolved: fsnotify: clear PARENT_WATCHED flags lazily
CVE-2024-47661	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Avoid overflow from uint32_t
CVE-2024-47662	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Remove register from DCN35
CVE-2024-47663	In the Linux kernel, the following vulnerability has been resolved: staging: iio: frequency: ad9834: Validate frequen
CVE-2024-47664	In the Linux kernel, the following vulnerability has been resolved: spi: hisi-kunpeng: Add verification for the max_f
CVE-2024-47665	In the Linux kernel, the following vulnerability has been resolved: i3c: mipi-i3c-hci: Error out instead on BUG_ON
CVE-2024-47667	In the Linux kernel, the following vulnerability has been resolved: PCI: keystone: Add workaround for Errata #1203
CVE-2024-47666	In the Linux kernel, the following vulnerability has been resolved: scsi: pm80xx: Set phy->enable_completion only
CVE-2024-47668	In the Linux kernel, the following vulnerability has been resolved: lib/generic-radix-tree.c: Fix rare race in __genrad
CVE-2024-47669	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix state management in error path of log v
CVE-2024-47670	In the Linux kernel, the following vulnerability has been resolved: ocfs2: add bounds checking to ocfs2_xattr_find_e
CVE-2024-47671	In the Linux kernel, the following vulnerability has been resolved: USB: usbtmc: prevent kernel-usb-infoleak The sy
CVE-2024-47672	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: don't wait for tx queues if fir
CVE-2024-47673	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: pause TCM when the firmwa
CVE-2024-47674	In the Linux kernel, the following vulnerability has been resolved: mm: avoid leaving partial pfn mappings around in
CVE-2024-47675	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix use-after-free in bpf_uprobe_multi_link_
CVE-2024-47677	In the Linux kernel, the following vulnerability has been resolved: exfat: resolve memory leak from exfat_create_up
CVE-2024-47678	In the Linux kernel, the following vulnerability has been resolved: icmp: change the order of rate limits ICMP messa
CVE-2024-47679	In the Linux kernel, the following vulnerability has been resolved: vfs: fix race between evic_inodes() and find_ino
CVE-2024-47681	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: fix NULL pointer dereferenc
CVE-2024-47682	In the Linux kernel, the following vulnerability has been resolved: scsi: sd: Fix off-by-one error in sd_read_block_cl
CVE-2024-47684	In the Linux kernel, the following vulnerability has been resolved: tcp: check skb is non-NULL in tcp_rto_delta_us
CVE-2024-47683	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Skip Recompute DSC Params i
CVE-2024-47686	In the Linux kernel, the following vulnerability has been resolved: ep93xx: clock: Fix off by one in ep93xx_div_reca
CVE-2024-47685	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_reject_ipv6: fix nf_reject_ip6_tcpdr
CVE-2024-47687	In the Linux kernel, the following vulnerability has been resolved: vdpa/mlx5: Fix invalid mr resource destroy Certai
CVE-2024-47688	In the Linux kernel, the following vulnerability has been resolved: driver core: Fix a potential null-ptr-deref in modu
CVE-2024-47689	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to don't set SB_RDONLY in f2fs_handle
CVE-2024-47690	In the Linux kernel, the following vulnerability has been resolved: f2fs: get rid of online repaire on corrupted directo
CVE-2024-47691	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid use-after-free in f2fs_stop_gc_th
CVE-2024-47692	In the Linux kernel, the following vulnerability has been resolved: nfsd: return -EINVAL when namelen is 0 When v
CVE-2024-47693	In the Linux kernel, the following vulnerability has been resolved: IB/core: Fix ib_cache_setup_one error flow clean
CVE-2024-47695	In the Linux kernel, the following vulnerability has been resolved: RDMA/rtrs-clt: Reset cid to con_num - 1 to stay i
CVE-2024-47697	In the Linux kernel, the following vulnerability has been resolved: drivers: media: dvb-frontends/rtl2830: fix an out-
CVE-2024-47696	In the Linux kernel, the following vulnerability has been resolved: RDMA/iwcm: Fix WARNING:at_kernel/workqu
CVE-2024-47698	In the Linux kernel, the following vulnerability has been resolved: drivers: media: dvb-frontends/rtl2832: fix an out-
CVE-2024-47699	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential null-ptr-deref in nilfs_btree_in
CVE-2024-47700	In the Linux kernel, the following vulnerability has been resolved: ext4: check stripe size compatibility on remount a

CVE-2024-47701	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid OOB when system.data xattr changes
CVE-2024-47702	In the Linux kernel, the following vulnerability has been resolved: bpf: Fail verification for sign-extension of packet
CVE-2024-47703	In the Linux kernel, the following vulnerability has been resolved: bpf, lsm: Add check for BPF LSM return value A
CVE-2024-47704	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check link_res->hpo_dp_link_
CVE-2024-47705	In the Linux kernel, the following vulnerability has been resolved: block: fix potential invalid pointer dereference in
CVE-2024-47706	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix possible UAF for bfqq->bic with m
CVE-2024-47707	In the Linux kernel, the following vulnerability has been resolved: ipv6: avoid possible NULL deref in rt6_uncached
CVE-2024-47709	In the Linux kernel, the following vulnerability has been resolved: can: bcm: Clear bo->bcm_proc_read after remove
CVE-2024-47710	In the Linux kernel, the following vulnerability has been resolved: sock_map: Add a cond_resched() in sock_hash_f
CVE-2024-47711	In the Linux kernel, the following vulnerability has been resolved: af_unix: Don't return OOB skb in manage_oob().
CVE-2024-47712	In the Linux kernel, the following vulnerability has been resolved: wifi: wilc1000: fix potential RCU dereference iss
CVE-2024-47713	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: use two-phase skb reclamation in
CVE-2024-47714	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: use hweight16 to get correct
CVE-2024-47715	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: fix oops on non-dbdc mt7980
CVE-2024-47716	In the Linux kernel, the following vulnerability has been resolved: ARM: 9410/1: vfp: Use asm volatile in fmxr/fmxr
CVE-2024-47719	In the Linux kernel, the following vulnerability has been resolved: iommufd: Protect against overflow of ALIGN() d
CVE-2024-47718	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: always wait for both firmware loadin
CVE-2024-47720	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for set_output_
CVE-2024-47723	In the Linux kernel, the following vulnerability has been resolved: jfs: fix out-of-bounds in dbNextAG() and diAlloc
CVE-2024-47726	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to wait dio completion It should wait all
CVE-2024-47727	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Fix "in-kernel MMIO" check TDX only
CVE-2024-47728	In the Linux kernel, the following vulnerability has been resolved: bpf: Zero former ARG_PTR_TO_{LONG,INT} &
CVE-2024-47729	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Use reserved copy engine for user binds
CVE-2024-47730	In the Linux kernel, the following vulnerability has been resolved: crypto: hisilicon/qm - inject error before stopping
CVE-2024-47731	In the Linux kernel, the following vulnerability has been resolved: drivers/perf: Fix ali_drw_pmu driver interrupt sta
CVE-2024-47732	In the Linux kernel, the following vulnerability has been resolved: crypto: iaa - Fix potential use after free bug The f
CVE-2024-47733	In the Linux kernel, the following vulnerability has been resolved: netfs: Delete subtree of 'fs/netfs' when netfs modu
CVE-2024-47735	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix spin_unlock_irqrestore() called v
CVE-2024-47734	In the Linux kernel, the following vulnerability has been resolved: bonding: Fix unnecessary warnings and logs from
CVE-2024-47736	In the Linux kernel, the following vulnerability has been resolved: erofs: handle overlapped pclusters out of crafted i
CVE-2024-47737	In the Linux kernel, the following vulnerability has been resolved: nfsd: call cache_put if xdr_reserve_space returns
CVE-2024-47738	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: don't use rate mask for offchanne
CVE-2024-47739	In the Linux kernel, the following vulnerability has been resolved: padata: use integer wrap around to prevent deadlo
CVE-2024-47740	In the Linux kernel, the following vulnerability has been resolved: f2fs: Require FMODE_WRITE for atomic write i
CVE-2024-47741	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix race setting file private on concurrent lse
CVE-2024-47742	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: Block path traversal Most firmv
CVE-2024-47743	In the Linux kernel, the following vulnerability has been resolved: KEYS: prevent NULL pointer dereference in find
CVE-2024-47745	In the Linux kernel, the following vulnerability has been resolved: mm: call the security_mmap_file() LSM hook in
CVE-2024-47744	In the Linux kernel, the following vulnerability has been resolved: KVM: Use dedicated mutex to protect kvm_usag
CVE-2024-47747	In the Linux kernel, the following vulnerability has been resolved: net: seq: Fix use after free vulnerability in ether3

CVE-2024-47749	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: Added NULL check for lookup_a
CVE-2024-47748	In the Linux kernel, the following vulnerability has been resolved: vhost_vdpa: assign irq bypass producer token cor
CVE-2024-47750	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix Use-After-Free of rsv_qp on HIB
CVE-2024-47751	In the Linux kernel, the following vulnerability has been resolved: PCI: kirin: Fix buffer overflow in kirin_pcie_pars
CVE-2024-47752	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix H264 stateless deco
CVE-2024-47753	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix VP8 stateless deco
CVE-2024-47754	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix H264 multi stateles
CVE-2024-47756	In the Linux kernel, the following vulnerability has been resolved: PCI: keystone: Fix if-statement expression in ks_
CVE-2024-47757	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential oob read in nilfs_btree_check
CVE-2024-47764	cookie is a basic HTTP cookie parser and serializer for HTTP servers. The cookie name could be used to set other fi
CVE-2024-47809	In the Linux kernel, the following vulnerability has been resolved: dlm: fix possible lkb_resource null dereference T
CVE-2024-47794	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent tailcall infinite loop caused by frepla
CVE-2024-47814	Vim is an open source, command line text editor. A use-after-free was found in Vim < 9.1.0764. When closing a buf
CVE-2024-47874	Starlette is an Asynchronous Server Gateway Interface (ASGI) framework/toolkit. Prior to version 0.40.0, Starlette t
CVE-2024-48873	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: check return value of ieee80211_pro
CVE-2024-48875	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't take dev_replace rwsem on task alrea
CVE-2024-48876	In the Linux kernel, the following vulnerability has been resolved: stackdepot: fix stack_depot_save_flags() in NMI
CVE-2024-48881	In the Linux kernel, the following vulnerability has been resolved: bcache: revert replacing IS_ERR_OR_NULL wit
CVE-2024-48957	execute_filter_audio in archive_read_support_format_rar.c in libarchive before 3.7.5 allows out-of-bounds access vi
CVE-2024-48958	execute_filter_delta in archive_read_support_format_rar.c in libarchive before 3.7.5 allows out-of-bounds access via
CVE-2024-49568	In the Linux kernel, the following vulnerability has been resolved: net/smc: check v2_ext_offset/eid_cnt/ism_gid_cn
CVE-2024-49569	In the Linux kernel, the following vulnerability has been resolved: nvme-rdma: unquiesce admin_q before destroy it
CVE-2024-49571	In the Linux kernel, the following vulnerability has been resolved: net/smc: check iparea_offset and ipv6_prefixes_c
CVE-2024-49750	The Snowflake Connector for Python provides an interface for developing Python applications that can connect to S
CVE-2024-49761	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.9 has a ReDoS vulnerability when it parses an X
CVE-2024-49766	Werkzeug is a Web Server Gateway Interface web application library. On Python < 3.11 on Windows, os.path.isabs
CVE-2024-49767	Werkzeug is a Web Server Gateway Interface web application library. Applications using `werkzeug.formparser.Mu
CVE-2024-49768	Waitress is a Web Server Gateway Interface server for Python 2 and 3. A remote client may send a request that is ex
CVE-2024-49769	Waitress is a Web Server Gateway Interface server for Python 2 and 3. When a remote client closes the connection b
CVE-2024-49850	In the Linux kernel, the following vulnerability has been resolved: bpf: correctly handle malformed BPF_CORE_TY
CVE-2024-49851	In the Linux kernel, the following vulnerability has been resolved: tpm: Clean up TPM space after command failure
CVE-2024-49852	In the Linux kernel, the following vulnerability has been resolved: scsi: elx: libefc: Fix potential use after free in efc
CVE-2024-49853	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix double free in OPTEE tr
CVE-2024-49855	In the Linux kernel, the following vulnerability has been resolved: nbd: fix race between timeout and normal comple
CVE-2024-49856	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Fix deadlock in SGX NUMA node search
CVE-2024-49858	In the Linux kernel, the following vulnerability has been resolved: efistub/tpm: Use ACPI reclaim memory for event
CVE-2024-49859	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to check atomic_file in f2fs_ioctl interf
CVE-2024-49860	In the Linux kernel, the following vulnerability has been resolved: ACPI: sysfs: validate return type of _STR method
CVE-2024-49861	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix helper writes to read-only maps Lonia
CVE-2024-49862	In the Linux kernel, the following vulnerability has been resolved: powercap: intel_rapl: Fix off by one in get_rpi() T

CVE-2024-49863	In the Linux kernel, the following vulnerability has been resolved: vhost/scsi: null-ptr-dereference in vhost_scsi_get
CVE-2024-49864	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix a race between socket set up and I/O th
CVE-2024-49865	In the Linux kernel, the following vulnerability has been resolved: drm/xe/vm: move xa_alloc to prevent UAF Evil u
CVE-2024-49866	In the Linux kernel, the following vulnerability has been resolved: tracing/timerlat: Fix a race during cpuhp processi
CVE-2024-49867	In the Linux kernel, the following vulnerability has been resolved: btrfs: wait for fixup workers before stopping clea
CVE-2024-49868	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix a NULL pointer dereference when failed
CVE-2024-49871	In the Linux kernel, the following vulnerability has been resolved: Input: adp5589-keys - fix NULL pointer derefere
CVE-2024-49870	In the Linux kernel, the following vulnerability has been resolved: cachefiles: fix dentry leak in cachefiles_open_file
CVE-2024-49874	In the Linux kernel, the following vulnerability has been resolved: i3c: master: svc: Fix use after free vulnerability in
CVE-2024-49875	In the Linux kernel, the following vulnerability has been resolved: nfsd: map the EBADMSG to nfserr_io to avoid w
CVE-2024-49876	In the Linux kernel, the following vulnerability has been resolved: drm/xe: fix UAF around queue destruction We cu
CVE-2024-49877	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix possible null-ptr-deref in ocfs2_set_buf
CVE-2024-49878	In the Linux kernel, the following vulnerability has been resolved: resource: fix region_intersects() vs add_memory_
CVE-2024-49879	In the Linux kernel, the following vulnerability has been resolved: drm: omapdrm: Add missing check for alloc_orde
CVE-2024-49880	In the Linux kernel, the following vulnerability has been resolved: ext4: fix off by one issue in alloc_flex_gd() Wesl
CVE-2024-49881	In the Linux kernel, the following vulnerability has been resolved: ext4: update orig_path in ext4_find_extent() In ex
CVE-2024-49882	In the Linux kernel, the following vulnerability has been resolved: ext4: fix double brelse() the buffer of the extents p
CVE-2024-49883	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid use-after-free in ext4_ext_insert_exten
CVE-2024-49885	In the Linux kernel, the following vulnerability has been resolved: mm, slub: avoid zeroing kcalloc redzone Since c
CVE-2024-49884	In the Linux kernel, the following vulnerability has been resolved: ext4: fix slab-use-after-free in ext4_split_extent_a
CVE-2024-49886	In the Linux kernel, the following vulnerability has been resolved: platform/x86: ISST: Fix the KASAN report slab-o
CVE-2024-49888	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix a sdiv overflow issue Zac Ecob reported
CVE-2024-49889	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid use-after-free in ext4_ext_show_leaf(
CVE-2024-49890	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: ensure the fw_info is not null before
CVE-2024-49891	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Validate hdwq pointers before derefere
CVE-2024-49892	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize get_bytes_per_element
CVE-2024-49893	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check stream_status before it is
CVE-2024-49895	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index out of bounds in DCN
CVE-2024-49894	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index out of bounds in dega
CVE-2024-49896	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check stream before comparing
CVE-2024-49897	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check phantom_stream before
CVE-2024-49898	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null-initialized variables
CVE-2024-49900	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix uninit-value access of new_ea in ea_buffe
CVE-2024-49899	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize denominators' default
CVE-2024-49901	In the Linux kernel, the following vulnerability has been resolved: drm/msm/adreno: Assign msm_gpu->pdev earlier
CVE-2024-49902	In the Linux kernel, the following vulnerability has been resolved: jfs: check if leafidx greater than num leaves per d
CVE-2024-49903	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix uaf in dbFreeBits [syzbot reported] =====
CVE-2024-49904	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: add list empty check to avoid null p
CVE-2024-49906	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointer before try to
CVE-2024-49905	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for 'afb' in amd

CVE-2024-49908	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for 'afb' in amd
CVE-2024-49907	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before using
CVE-2024-49909	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function
CVE-2024-49910	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function
CVE-2024-49911	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function
CVE-2024-49912	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Handle null 'stream_status' in 'p
CVE-2024-49913	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for top_pipe_to
CVE-2024-49914	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for pipe_ctx->p
CVE-2024-49915	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr
CVE-2024-49917	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr
CVE-2024-49916	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr
CVE-2024-49918	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for head_pipe i
CVE-2024-49919	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for head_pipe i
CVE-2024-49920	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before mul
CVE-2024-49921	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before usec
CVE-2024-49922	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointers before usin
CVE-2024-49923	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Pass non-null to dcn20_validate
CVE-2024-49924	In the Linux kernel, the following vulnerability has been resolved: fbdev: pxaafb: Fix possible use after free in pxaafb_
CVE-2024-49925	In the Linux kernel, the following vulnerability has been resolved: fbdev: efifb: Register sysfs groups through driver
CVE-2024-49926	In the Linux kernel, the following vulnerability has been resolved: rcu-tasks: Fix access non-existent percpu rtcpv va
CVE-2024-49928	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid reading out of bounds when loc
CVE-2024-49927	In the Linux kernel, the following vulnerability has been resolved: x86/ioapic: Handle allocation failures gracefully
CVE-2024-49929	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: avoid NULL pointer derefere
CVE-2024-49930	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix array out-of-bound access in SoC
CVE-2024-49931	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix array out-of-bound access in SoC
CVE-2024-49932	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't readahead the relocation inode on RS
CVE-2024-49933	In the Linux kernel, the following vulnerability has been resolved: blk_iocost: fix more out of bound shifts Recently
CVE-2024-49934	In the Linux kernel, the following vulnerability has been resolved: fs/inode: Prevent dump_mapping() accessing inva
CVE-2024-49935	In the Linux kernel, the following vulnerability has been resolved: ACPI: PAD: fix crash in exit_round_robin() The
CVE-2024-49936	In the Linux kernel, the following vulnerability has been resolved: net/xen-netback: prevent UAF in xenvif_flush_ha
CVE-2024-49937	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: Set correct chandef when starting
CVE-2024-49938	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k_htc: Use __skb_set_length() for resett
CVE-2024-49939	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid to add interface to list twice wh
CVE-2024-49940	In the Linux kernel, the following vulnerability has been resolved: l2tp: prevent possible tunnel refcount underflow
CVE-2024-49942	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Prevent null pointer access in xe_migrate
CVE-2024-49944	In the Linux kernel, the following vulnerability has been resolved: sctp: set sk_state back to CLOSED if autobind fai
CVE-2024-49945	In the Linux kernel, the following vulnerability has been resolved: net/ncsi: Disable the ncsi work before freeing the
CVE-2024-49946	In the Linux kernel, the following vulnerability has been resolved: ppp: do not assume bh is held in ppp_channel_bri
CVE-2024-49947	In the Linux kernel, the following vulnerability has been resolved: net: test for not too small csum_start in virtio_net
CVE-2024-49948	In the Linux kernel, the following vulnerability has been resolved: net: add more sanity checks to qdisc_pkt_len_init

CVE-2024-49949	In the Linux kernel, the following vulnerability has been resolved: net: avoid potential underflow in qdisc_pkt_len_i
CVE-2024-49950	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix uaf in l2cap_connect [Syz
CVE-2024-49951	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix possible crash on mgmt_
CVE-2024-49953	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix crash caused by calling __xfrm_st
CVE-2024-49955	In the Linux kernel, the following vulnerability has been resolved: ACPI: battery: Fix possible crash when unregister
CVE-2024-49954	In the Linux kernel, the following vulnerability has been resolved: static_call: Replace pointless WARN_ON() in sta
CVE-2024-49952	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: prevent nf_skb_duplicated co
CVE-2024-49957	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix null-ptr-deref when journal load failed.
CVE-2024-49961	In the Linux kernel, the following vulnerability has been resolved: media: i2c: ar0521: Use cansleep version of gpio
CVE-2024-49960	In the Linux kernel, the following vulnerability has been resolved: ext4: fix timer use-after-free on failed mount Syz
CVE-2024-49959	In the Linux kernel, the following vulnerability has been resolved: jbd2: stop waiting for space when jbd2_cleanu
CVE-2024-49958	In the Linux kernel, the following vulnerability has been resolved: ocfs2: reserve space for inline xattr before attachi
CVE-2024-49962	In the Linux kernel, the following vulnerability has been resolved: ACPICA: check null return of ACPI_ALLOCAT
CVE-2024-49963	In the Linux kernel, the following vulnerability has been resolved: mailbox: bcm2835: Fix timeout during suspend m
CVE-2024-49967	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-49965	In the Linux kernel, the following vulnerability has been resolved: ocfs2: remove unreasonable unlock in ocfs2_read
CVE-2024-49966	In the Linux kernel, the following vulnerability has been resolved: ocfs2: cancel dq_i_sync_work before freeing oinfo
CVE-2024-49968	In the Linux kernel, the following vulnerability has been resolved: ext4: filesystems without casefold feature cannot
CVE-2024-49969	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix index out of bounds in DC
CVE-2024-49970	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Implement bounds check for str
CVE-2024-49971	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Increase array size of dummy_
CVE-2024-49972	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Deallocate DML memory if all
CVE-2024-49973	In the Linux kernel, the following vulnerability has been resolved: r8169: add tally counter fields added with RTL81
CVE-2024-49975	In the Linux kernel, the following vulnerability has been resolved: uprobes: fix kernel info leak via "[uprobes]" vma
CVE-2024-49974	In the Linux kernel, the following vulnerability has been resolved: NFSD: Limit the number of concurrent async CO
CVE-2024-49976	In the Linux kernel, the following vulnerability has been resolved: tracing/timerlat: Drop interface_lock in stop_kthr
CVE-2024-49977	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: Fix zero-division error when disabli
CVE-2024-49978	In the Linux kernel, the following vulnerability has been resolved: gso: fix udp gso fraglist segmentation after pull fr
CVE-2024-49980	In the Linux kernel, the following vulnerability has been resolved: vrf: revert "vrf: Remove unnecessary RCU-bh cri
CVE-2024-49981	In the Linux kernel, the following vulnerability has been resolved: media: venus: fix use after free bug in venus_rem
CVE-2024-49982	In the Linux kernel, the following vulnerability has been resolved: aoe: fix the potential use-after-free problem in mo
CVE-2024-49983	In the Linux kernel, the following vulnerability has been resolved: ext4: drop ppath from ext4_ext_replay_update_ex
CVE-2024-49985	In the Linux kernel, the following vulnerability has been resolved: i2c: stm32f7: Do not prepare/unprepare clock dur
CVE-2024-49986	In the Linux kernel, the following vulnerability has been resolved: platform/x86: x86-android-tablets: Fix use after fr
CVE-2024-49987	In the Linux kernel, the following vulnerability has been resolved: bpftool: Fix undefined behavior in qsort(NULL, C
CVE-2024-49989	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix double free issue during am
CVE-2024-49988	In the Linux kernel, the following vulnerability has been resolved: ksmbd: add refcnt to ksmbd_conn struct When se
CVE-2024-49990	In the Linux kernel, the following vulnerability has been resolved: drm/xehdcp: Check GSC structure validity Some
CVE-2024-49991	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: amd_kfd_free_gtt_mem clear the cor
CVE-2024-49992	In the Linux kernel, the following vulnerability has been resolved: drm/stm: Avoid use-after-free issues with crtc and

CVE-2024-49994	In the Linux kernel, the following vulnerability has been resolved: block: fix integer overflow in BLKSECDISCARD
CVE-2024-49993	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-49995	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-49996	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix buffer overflow when parsing NFS repair
CVE-2024-49997	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: lantiq_etop: fix memory disclosure
CVE-2024-49998	In the Linux kernel, the following vulnerability has been resolved: net: dsa: improve shutdown sequence Alexander S
CVE-2024-49999	In the Linux kernel, the following vulnerability has been resolved: afs: Fix the setting of the server responding flag I
CVE-2024-50000	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix NULL deref in mlx5e_tir_builder
CVE-2024-50001	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix error path in multi-packet WQE tra
CVE-2024-50002	In the Linux kernel, the following vulnerability has been resolved: static_call: Handle module init failure correctly in
CVE-2024-50003	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix system hang while resume
CVE-2024-50004	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: update DML2 policy Enhanced
CVE-2024-50005	In the Linux kernel, the following vulnerability has been resolved: mac802154: Fix potential RCU dereference issue
CVE-2024-50006	In the Linux kernel, the following vulnerability has been resolved: ext4: fix i_data_sem unlock order in ext4_ind_mi
CVE-2024-50007	In the Linux kernel, the following vulnerability has been resolved: ALSA: asihpi: Fix potential OOB array access AS
CVE-2024-50009	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: add check for cpufreq_cpu_
CVE-2024-50008	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Fix memcpy() field-spanning write
CVE-2024-50010	In the Linux kernel, the following vulnerability has been resolved: exec: don't WARN for racy path_noexec check B
CVE-2024-50012	In the Linux kernel, the following vulnerability has been resolved: cpufreq: Avoid a bad reference count on CPU no
CVE-2024-50013	In the Linux kernel, the following vulnerability has been resolved: exfat: fix memory leak in exfat_load_bitmap() If
CVE-2024-50015	In the Linux kernel, the following vulnerability has been resolved: ext4: dax: fix overflowing extents beyond inode s
CVE-2024-50014	In the Linux kernel, the following vulnerability has been resolved: ext4: fix access to uninitialised lock in fc replay p
CVE-2024-50017	In the Linux kernel, the following vulnerability has been resolved: x86/mm/ident_map: Use gbpages only where full
CVE-2024-50018	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-50019	In the Linux kernel, the following vulnerability has been resolved: kthread: unpark only parked kthread Calling into
CVE-2024-50020	In the Linux kernel, the following vulnerability has been resolved: ice: Fix improper handling of refcount in ice_srio
CVE-2024-50021	In the Linux kernel, the following vulnerability has been resolved: ice: Fix improper handling of refcount in ice_dpll
CVE-2024-50022	In the Linux kernel, the following vulnerability has been resolved: device-dax: correct pgoff align in dax_set_mappin
CVE-2024-50023	In the Linux kernel, the following vulnerability has been resolved: net: phy: Remove LED entry from LEDs list on u
CVE-2024-50024	In the Linux kernel, the following vulnerability has been resolved: net: Fix an unsafe loop on the list The kernel may
CVE-2024-50025	In the Linux kernel, the following vulnerability has been resolved: scsi: fnic: Move flush_work initialization out of i
CVE-2024-50026	In the Linux kernel, the following vulnerability has been resolved: scsi: wd33c93: Don't use stale scsi_pointer value
CVE-2024-50027	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Free tzp copy along with the therm
CVE-2024-50028	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Reference count the zone in therm
CVE-2024-50030	In the Linux kernel, the following vulnerability has been resolved: drm/xen/ct: prevent UAF in send_recv() Ensure we
CVE-2024-50029	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: Fix UAF in hci_enhanced_s
CVE-2024-50031	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Stop the active perfmon before being de
CVE-2024-50033	In the Linux kernel, the following vulnerability has been resolved: slip: make slhc_remember() more robust against r
CVE-2024-50034	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix lacks of icsk_syn_mss with IPPROTO
CVE-2024-50035	In the Linux kernel, the following vulnerability has been resolved: ppp: fix ppp_async_encode() illegal access syzbo

CVE-2024-50036	In the Linux kernel, the following vulnerability has been resolved: net: do not delay dst_entries_add() in dst_release()
CVE-2024-50038	In the Linux kernel, the following vulnerability has been resolved: netfilter: xttables: avoid NFPROTO_UNSPEC when
CVE-2024-50039	In the Linux kernel, the following vulnerability has been resolved: net/sched: accept TCA_STAB only for root qdisc
CVE-2024-50040	In the Linux kernel, the following vulnerability has been resolved: igb: Do not bring the device up after non-fatal error
CVE-2024-50041	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix macvlan leak by synchronizing access to
CVE-2024-50042	In the Linux kernel, the following vulnerability has been resolved: ice: Fix increasing MSI-X on VF Increasing MSI-X
CVE-2024-50044	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: FIX possible deadlock in
CVE-2024-50045	In the Linux kernel, the following vulnerability has been resolved: netfilter: br_netfilter: fix panic with metadata_dst
CVE-2024-50046	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Prevent NULL-pointer dereference in nfs
CVE-2024-50048	In the Linux kernel, the following vulnerability has been resolved: fbcon: Fix a NULL pointer dereference issue in fb
CVE-2024-50047	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix UAF in async decryption Doing a
CVE-2024-50049	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointer before deref
CVE-2024-50051	In the Linux kernel, the following vulnerability has been resolved: spi: mpc52xx: Add cancel_work_sync before mo
CVE-2024-50055	In the Linux kernel, the following vulnerability has been resolved: driver core: bus: Fix double free in driver API bus
CVE-2024-50056	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: uvc: Fix ERR_PTR dereference in u
CVE-2024-50057	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tipd: Free IRQ only if it was requeste
CVE-2024-50058	In the Linux kernel, the following vulnerability has been resolved: serial: protect uart_port_dtr_rts() in uart_shutdown
CVE-2024-50059	In the Linux kernel, the following vulnerability has been resolved: ntb: ntb_hw_switchtec: Fix use after free vulnerab
CVE-2024-50060	In the Linux kernel, the following vulnerability has been resolved: io_uring: check if we need to reschedule during o
CVE-2024-50061	In the Linux kernel, the following vulnerability has been resolved: i3c: master: cdns: Fix use after free vulnerability
CVE-2024-50062	In the Linux kernel, the following vulnerability has been resolved: RDMA/rtrs-srv: Avoid null pointer deref during p
CVE-2024-50063	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent tail call between progs attached to di
CVE-2024-50065	In the Linux kernel, the following vulnerability has been resolved: ntfs3: Change to non-blocking allocation in ntfs_
CVE-2024-50064	In the Linux kernel, the following vulnerability has been resolved: zram: free secondary algorithms names We need
CVE-2024-50066	In the Linux kernel, the following vulnerability has been resolved: mm/mremap: fix move_normal_pmd/retract_page
CVE-2024-50067	In the Linux kernel, the following vulnerability has been resolved: uprobe: avoid out-of-bounds memory access of fe
CVE-2024-50068	In the Linux kernel, the following vulnerability has been resolved: mm/damon/tests/sysfs-kunit.h: fix memory leak in
CVE-2024-50069	In the Linux kernel, the following vulnerability has been resolved: pinctrl: apple: check devm_kasprintf() returned v
CVE-2024-50070	In the Linux kernel, the following vulnerability has been resolved: pinctrl: stm32: check devm_kasprintf() returned v
CVE-2024-50072	In the Linux kernel, the following vulnerability has been resolved: x86/bugs: Use code segment selector for VERW
CVE-2024-50073	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: Fix use-after-free in gsm_cleanup_mu
CVE-2024-50074	In the Linux kernel, the following vulnerability has been resolved: parport: Proper fix for array out-of-bounds access
CVE-2024-50075	In the Linux kernel, the following vulnerability has been resolved: xhci: tegra: fix checked USB2 port number If US
CVE-2024-50076	In the Linux kernel, the following vulnerability has been resolved: vt: prevent kernel-infoleak in con_font_get() font
CVE-2024-50077	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: ISO: Fix multiple init when debugfs is
CVE-2024-50078	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Call iso_exit() on module unload If iso
CVE-2024-50080	In the Linux kernel, the following vulnerability has been resolved: ublk: don't allow user copy for unprivileged device
CVE-2024-50082	In the Linux kernel, the following vulnerability has been resolved: blk-rq-qos: fix crash on rq_qos_wait vs. rq_qos_v
CVE-2024-50083	In the Linux kernel, the following vulnerability has been resolved: tcp: fix mptcp DSS corruption due to large pmtu
CVE-2024-50084	In the Linux kernel, the following vulnerability has been resolved: net: microchip: vcap api: Fix memory leaks in vca

CVE-2024-50085	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: fix UaF read in mptcp_pm_nl_rm_ad
CVE-2024-50086	In the Linux kernel, the following vulnerability has been resolved: ksmdb: fix user-after-free from session log off Th
CVE-2024-50087	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix uninitialized pointer free on read_alloc_
CVE-2024-50088	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix uninitialized pointer free in add_inode_r
CVE-2024-50090	In the Linux kernel, the following vulnerability has been resolved: drm/xe/oa: Fix overflow in oa batch buffer By de
CVE-2024-50091	In the Linux kernel, the following vulnerability has been resolved: dm vdo: don't refer to dedupe_context after releas
CVE-2024-50095	In the Linux kernel, the following vulnerability has been resolved: RDMA/mad: Improve handling of timed out WR
CVE-2024-50093	In the Linux kernel, the following vulnerability has been resolved: thermal: intel: int340x: processor: Fix warning du
CVE-2024-50096	In the Linux kernel, the following vulnerability has been resolved: nouveau/dmem: Fix vulnerability in migrate_to_r
CVE-2024-50098	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Set SDEV_OFFLINE when UFS i
CVE-2024-50099	In the Linux kernel, the following vulnerability has been resolved: arm64: probes: Remove broken LDR (literal) upr
CVE-2024-50101	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix incorrect pci_for_each_dma_alia
CVE-2024-50102	In the Linux kernel, the following vulnerability has been resolved: x86: fix user address masking non-canonical spec
CVE-2024-50103	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: Fix NULL Dereference in asoc_qco
CVE-2024-50104	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: sdm845: add missing soundwire ru
CVE-2024-50105	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: sc7280: Fix missing Soundwire run
CVE-2024-50106	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix race between laundromat and free_statei
CVE-2024-50107	In the Linux kernel, the following vulnerability has been resolved: platform/x86/intel/pmc: Fix pmc_core_iounmap t
CVE-2024-50108	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Disable PSR-SU on Parade 08-
CVE-2024-50110	In the Linux kernel, the following vulnerability has been resolved: xfrm: fix one more kernel-infoleak in algo dumpi
CVE-2024-50111	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Enable IRQ if do_ale() triggered in i
CVE-2024-50112	In the Linux kernel, the following vulnerability has been resolved: x86/lam: Disable ADDRESS_MASKING in mos
CVE-2024-50115	In the Linux kernel, the following vulnerability has been resolved: KVM: nSVM: Ignore nCR3[4:0] when loading P
CVE-2024-50116	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix kernel bug due to missing clearing of b
CVE-2024-50117	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Guard against bad data for ATIF ACPI
CVE-2024-50118	In the Linux kernel, the following vulnerability has been resolved: btrfs: reject ro->rw reconfiguration if there are ha
CVE-2024-50120	In the Linux kernel, the following vulnerability has been resolved: smb: client: Handle kstrdup failures for password
CVE-2024-50121	In the Linux kernel, the following vulnerability has been resolved: nfsd: cancel nfsd_shrinker_work using sync mode
CVE-2024-50124	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: ISO: Fix UAF on iso_sock_timeout co
CVE-2024-50125	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SCO: Fix UAF on sco_sock_timeout c
CVE-2024-50126	In the Linux kernel, the following vulnerability has been resolved: net: sched: use RCU read-side critical section in t
CVE-2024-50127	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix use-after-free in taprio_change() In
CVE-2024-50128	In the Linux kernel, the following vulnerability has been resolved: net: wwan: fix global oob in wwan_rtnl_policy Th
CVE-2024-50130	In the Linux kernel, the following vulnerability has been resolved: netfilter: bpf: must hold reference on net namespa
CVE-2024-50133	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Don't crash in stack_top() for tasks w
CVE-2024-50131	In the Linux kernel, the following vulnerability has been resolved: tracing: Consider the NULL character when valid
CVE-2024-50134	In the Linux kernel, the following vulnerability has been resolved: drm/vboxvideo: Replace fake VLA at end of vbva
CVE-2024-50135	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: fix race condition between reset and nv
CVE-2024-50136	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Unregister notifier on eswitch init failu
CVE-2024-50137	In the Linux kernel, the following vulnerability has been resolved: reset: starfive: jh71x0: Fix accessing the empty m

CVE-2024-50138	In the Linux kernel, the following vulnerability has been resolved: bpf: Use raw_spinlock_t in ringbuf The function
CVE-2024-50139	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Fix shift-out-of-bounds bug Fix a
CVE-2024-50140	In the Linux kernel, the following vulnerability has been resolved: sched/core: Disable page allocation in task_tick_r
CVE-2024-50141	In the Linux kernel, the following vulnerability has been resolved: ACPI: PRM: Find EFI_MEMORY_RUNTIME b
CVE-2024-50142	In the Linux kernel, the following vulnerability has been resolved: xfrm: validate new SA's prefixlen using SA famil
CVE-2024-50143	In the Linux kernel, the following vulnerability has been resolved: udf: fix uninit-value use in udf_get_fileshead Cl
CVE-2024-50145	In the Linux kernel, the following vulnerability has been resolved: octeon_ep: Add SKB allocation failures handling
CVE-2024-50146	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Don't call cleanup on profile rollback
CVE-2024-50147	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix command bitmask initialization Co
CVE-2024-50148	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: bnep: fix wild-memory-access in proto
CVE-2024-50150	In the Linux kernel, the following vulnerability has been resolved: usb: typec: altmode should keep reference to pare
CVE-2024-50151	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix OOBs when building SMB2_IOC
CVE-2024-50152	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix possible double free in smb2_set
CVE-2024-50153	In the Linux kernel, the following vulnerability has been resolved: scsi: target: core: Fix null-ptr-deref in target_alloc
CVE-2024-50154	In the Linux kernel, the following vulnerability has been resolved: tcp/dccp: Don't use timer_pending() in reqsk_que
CVE-2024-50155	In the Linux kernel, the following vulnerability has been resolved: netdevsim: use cond_resched() in nsim_dev_trap
CVE-2024-50156	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Avoid NULL dereference in msm_disp
CVE-2024-50157	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Avoid CPU lockups due fifo occ
CVE-2024-50158	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix out of bound check Driver e
CVE-2024-50159	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix the double free in scmi
CVE-2024-50160	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda/cs8409: Fix possible NULL dereferen
CVE-2024-50162	In the Linux kernel, the following vulnerability has been resolved: bpf: devmap: provide rxq after redirect rxq contain
CVE-2024-50163	In the Linux kernel, the following vulnerability has been resolved: bpf: Make sure internal and UAPI bpf_redirect fla
CVE-2024-50164	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix overloading of MEM_UNINIT's meaning
CVE-2024-50166	In the Linux kernel, the following vulnerability has been resolved: fsl/fman: Fix refcount handling of fman-related d
CVE-2024-50167	In the Linux kernel, the following vulnerability has been resolved: be2net: fix potential memory leak in be_xmit() Th
CVE-2024-50168	In the Linux kernel, the following vulnerability has been resolved: net/sun3_82586: fix potential memory leak in sun
CVE-2024-50170	In the Linux kernel, the following vulnerability has been resolved: net: bcmasp: fix potential memory leak in bcmasp
CVE-2024-50169	In the Linux kernel, the following vulnerability has been resolved: vsock: Update rx_bytes on read_skb() Make sure
CVE-2024-50171	In the Linux kernel, the following vulnerability has been resolved: net: systemport: fix potential memory leak in bcm
CVE-2024-50172	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix a possible memory leak In b
CVE-2024-50175	In the Linux kernel, the following vulnerability has been resolved: media: qcom: camss: Remove use_count guard in
CVE-2024-50176	In the Linux kernel, the following vulnerability has been resolved: remoteproc: k3-r5: Fix error handling when powe
CVE-2024-50178	In the Linux kernel, the following vulnerability has been resolved: cpufreq: loongson3: Use raw_smp_processor_id(
CVE-2024-50179	In the Linux kernel, the following vulnerability has been resolved: ceph: remove the incorrect Fw reference check w
CVE-2024-50180	In the Linux kernel, the following vulnerability has been resolved: fbdev: sisfb: Fix strbuf array overflow The values
CVE-2024-50182	In the Linux kernel, the following vulnerability has been resolved: secretmem: disable memfd_secret() if arch cannot
CVE-2024-50183	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Ensure DA_ID handling completion be
CVE-2024-50184	In the Linux kernel, the following vulnerability has been resolved: virtio_pmem: Check device status before requesti
CVE-2024-50185	In the Linux kernel, the following vulnerability has been resolved: mptcp: handle consistently DSS corruption Bugg

CVE-2024-50186	In the Linux kernel, the following vulnerability has been resolved: net: explicitly clear the sk pointer, when pf->creat
CVE-2024-50187	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: Stop the active perfmon before being de
CVE-2024-50188	In the Linux kernel, the following vulnerability has been resolved: net: phy: dp83869: fix memory corruption when e
CVE-2024-50189	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: Switch to device-managed dman
CVE-2024-50191	In the Linux kernel, the following vulnerability has been resolved: ext4: don't set SB_RDONLY after filesystem erro
CVE-2024-50192	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v4: Don't allow a VMOVP on a dying
CVE-2024-50193	In the Linux kernel, the following vulnerability has been resolved: x86/entry_32: Clear CPU buffers after register res
CVE-2024-50194	In the Linux kernel, the following vulnerability has been resolved: arm64: probes: Fix uprobes for big-endian kernel
CVE-2024-50195	In the Linux kernel, the following vulnerability has been resolved: posix-clock: Fix missing timespec64 check in pc_
CVE-2024-50196	In the Linux kernel, the following vulnerability has been resolved: pinctrl: ocelot: fix system hang on level based int
CVE-2024-50197	In the Linux kernel, the following vulnerability has been resolved: pinctrl: intel: platform: fix error path in device_fo
CVE-2024-50198	In the Linux kernel, the following vulnerability has been resolved: iio: light: vml6030: fix IIO device retrieval from
CVE-2024-50199	In the Linux kernel, the following vulnerability has been resolved: mm/swapfile: skip HugeTLB pages for unuse_vm
CVE-2024-50200	In the Linux kernel, the following vulnerability has been resolved: maple_tree: correct tree corruption on spanning st
CVE-2024-50201	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: Fix encoder->possible_clones Includ
CVE-2024-50202	In the Linux kernel, the following vulnerability has been resolved: nilfs2: propagate directory read errors from nilfs_
CVE-2024-50203	In the Linux kernel, the following vulnerability has been resolved: bpf, arm64: Fix address emission with tag-based l
CVE-2024-50205	In the Linux kernel, the following vulnerability has been resolved: ALSA: firewire-lib: Avoid division by zero in ap
CVE-2024-50206	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk_eth_soc: fix memory corruption
CVE-2024-50207	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix reader locking when changing the
CVE-2024-50208	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix a bug while setting up Level
CVE-2024-50209	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Add a check for memory allocat
CVE-2024-50210	In the Linux kernel, the following vulnerability has been resolved: posix-clock: posix-clock: Fix unbalanced locking
CVE-2024-50211	In the Linux kernel, the following vulnerability has been resolved: udf: refactor inode_bmap() to handle error Refact
CVE-2024-50215	In the Linux kernel, the following vulnerability has been resolved: nvmet-auth: assign dh_key to NULL after kfree_s
CVE-2024-50216	In the Linux kernel, the following vulnerability has been resolved: xfs: fix finding a last resort AG in xfs_filestream
CVE-2024-50217	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free of block device file in __b
CVE-2024-50218	In the Linux kernel, the following vulnerability has been resolved: ocfs2: pass u64 to ocfs2_truncate_inline maybe o
CVE-2024-50219	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-50220	In the Linux kernel, the following vulnerability has been resolved: fork: do not invoke uffd on fork if error occurs Pa
CVE-2024-50221	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Vangogh: Fix kernel memory out o
CVE-2024-50222	In the Linux kernel, the following vulnerability has been resolved: iov_iter: fix copy_page_from_iter_atomic() if KM
CVE-2024-50223	In the Linux kernel, the following vulnerability has been resolved: sched/numa: Fix the potential null pointer derefer
CVE-2024-50224	In the Linux kernel, the following vulnerability has been resolved: spi: spi-fsl-dspi: Fix crash when not using GPIO c
CVE-2024-50225	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix error propagation of split bios The purpo
CVE-2024-50226	In the Linux kernel, the following vulnerability has been resolved: cxl/port: Fix use-after-free, permit out-of-order de
CVE-2024-50228	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-50230	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix kernel bug due to missing clearing of c
CVE-2024-50229	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential deadlock with newly created s
CVE-2024-50231	In the Linux kernel, the following vulnerability has been resolved: iio: gts-helper: Fix memory leaks in iio_gts_build

CVE-2024-50232	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ad7124: fix division by zero in ad7124_s
CVE-2024-50233	In the Linux kernel, the following vulnerability has been resolved: staging: iio: frequency: ad9832: fix division by ze
CVE-2024-50234	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlegacy: Clear stale interrupts before resum
CVE-2024-50235	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: clear wdev->cqm_config pointer
CVE-2024-50236	In the Linux kernel, the following vulnerability has been resolved: wifi: ath10k: Fix memory leak in management tx
CVE-2024-50237	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: do not pass a stopped vif to the d
CVE-2024-50238	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usbc: fix NULL-deref on runtime
CVE-2024-50239	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb-legacy: fix NULL-deref on r
CVE-2024-50240	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb: fix NULL-deref on runtime
CVE-2024-50242	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Additional check in ntfs_file_release
CVE-2024-50243	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fix general protection fault in run_is_ma
CVE-2024-50244	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Additional check in ni_clear() Checking
CVE-2024-50245	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fix possible deadlock in mi_read Mutex
CVE-2024-50246	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Add rough attr alloc_size check
CVE-2024-50248	In the Linux kernel, the following vulnerability has been resolved: ntfs3: Add bounds checking to mi_enum_attr() A
CVE-2024-50247	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Check if more than chunk-size bytes are
CVE-2024-50249	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Make rmw_lock a raw_spin_lock T
CVE-2024-50250	In the Linux kernel, the following vulnerability has been resolved: fsdax: dax_unshare_iter needs to copy entire bloc
CVE-2024-50251	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_payload: sanitize offset and length b
CVE-2024-50252	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_ipip: Fix memory leak when ch
CVE-2024-50255	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci: fix null-ptr-deref in hci_read_sup
CVE-2024-50256	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_reject_ipv6: fix potential crash in nf_
CVE-2024-50257	In the Linux kernel, the following vulnerability has been resolved: netfilter: Fix use-after-free in get_info() ip6table_
CVE-2024-50258	In the Linux kernel, the following vulnerability has been resolved: net: fix crash when config small gso_max_size/gso
CVE-2024-50259	In the Linux kernel, the following vulnerability has been resolved: netdevsim: Add trailing zero to terminate the strin
CVE-2024-50261	In the Linux kernel, the following vulnerability has been resolved: macsec: Fix use-after-free while sending the offlo
CVE-2024-50262	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix out-of-bounds write in trie_get_next_key
CVE-2024-50263	In the Linux kernel, the following vulnerability has been resolved: fork: only invoke khugepaged, ksm hooks if no er
CVE-2024-50264	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: Initialization of the dangling pointer
CVE-2024-50265	In the Linux kernel, the following vulnerability has been resolved: ocfs2: remove entry once instead of null-ptr-deref
CVE-2024-50267	In the Linux kernel, the following vulnerability has been resolved: USB: serial: io_edgeport: fix use after free in deb
CVE-2024-50268	In the Linux kernel, the following vulnerability has been resolved: usb: typec: fix potential out of bounds in ucsi_ccg
CVE-2024-50270	In the Linux kernel, the following vulnerability has been resolved: mm/damon/core: avoid overflow in damon_feed_
CVE-2024-50269	In the Linux kernel, the following vulnerability has been resolved: usb: musb: sunxi: Fix accessing an released usb p
CVE-2024-50271	In the Linux kernel, the following vulnerability has been resolved: signal: restore the override_rlimit logic Prior to c
CVE-2024-50272	In the Linux kernel, the following vulnerability has been resolved: filemap: Fix bounds checking in filemap_read() I
CVE-2024-50273	In the Linux kernel, the following vulnerability has been resolved: btrfs: reinitialize delayed ref list after deleting it f
CVE-2024-50274	In the Linux kernel, the following vulnerability has been resolved: idpf: avoid vport access in idpf_get_link_ksetting
CVE-2024-50275	In the Linux kernel, the following vulnerability has been resolved: arm64/sve: Discard stale CPU state when handlin
CVE-2024-50276	In the Linux kernel, the following vulnerability has been resolved: net: vertexcom: mse102x: Fix possible double fre

CVE-2024-50277	In the Linux kernel, the following vulnerability has been resolved: dm: fix a crash if blk_alloc_disk fails If blk_alloc
CVE-2024-50279	In the Linux kernel, the following vulnerability has been resolved: dm cache: fix out-of-bounds access to the dirty bi
CVE-2024-50278	In the Linux kernel, the following vulnerability has been resolved: dm cache: fix potential out-of-bounds access on th
CVE-2024-50280	In the Linux kernel, the following vulnerability has been resolved: dm cache: fix flushing uninitialized delayed_worl
CVE-2024-50282	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: add missing size check in amdgpu_
CVE-2024-50283	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-use-after-free in smb3_preauth_h
CVE-2024-50284	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Fix the missing xa_store error check xa_s
CVE-2024-50285	In the Linux kernel, the following vulnerability has been resolved: ksmbd: check outstanding simultaneous SMB ope
CVE-2024-50286	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-use-after-free in ksmbd_smb2_se
CVE-2024-50287	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-tpg: prevent the risk of a division by
CVE-2024-50288	In the Linux kernel, the following vulnerability has been resolved: media: vivid: fix buffer overwrite when using > 3
CVE-2024-50289	In the Linux kernel, the following vulnerability has been resolved: media: av7110: fix a spectre vulnerability As war
CVE-2024-50290	In the Linux kernel, the following vulnerability has been resolved: media: cx24116: prevent overflows on SNR calcul
CVE-2024-50291	In the Linux kernel, the following vulnerability has been resolved: media: dvb-core: add missing buffer index check
CVE-2024-50292	In the Linux kernel, the following vulnerability has been resolved: ASoC: stm32: spdifrx: fix dma channel release in
CVE-2024-50294	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix missing locking causing hanging calls
CVE-2024-50296	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when uninstalling drive
CVE-2024-50295	In the Linux kernel, the following vulnerability has been resolved: net: arc: fix the device for dma_map_single/dma_
CVE-2024-50297	In the Linux kernel, the following vulnerability has been resolved: net: xilinx: axienet: Enqueue Tx packets in dql be
CVE-2024-50298	In the Linux kernel, the following vulnerability has been resolved: net: enetc: allocate vf_state during PF probes In th
CVE-2024-50299	In the Linux kernel, the following vulnerability has been resolved: sctp: properly validate chunk size in sctp_sf_ootb
CVE-2024-50300	In the Linux kernel, the following vulnerability has been resolved: regulator: rtq2208: Fix uninitialized use of regula
CVE-2024-50301	In the Linux kernel, the following vulnerability has been resolved: security/keys: fix slab-out-of-bounds in key_task
CVE-2024-50302	In the Linux kernel, the following vulnerability has been resolved: HID: core: zero-initialize the report buffer Since
CVE-2024-50303	In the Linux kernel, the following vulnerability has been resolved: resource,kexec: walk_system_ram_res_rev must t
CVE-2024-50304	In the Linux kernel, the following vulnerability has been resolved: ipv4: ip_tunnel: Fix suspicious RCU usage warni
CVE-2024-50349	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high
CVE-2024-50602	An issue was discovered in libexpat before 2.6.4. There is a crash within the XML_ResumeParser function because
CVE-2024-50379	Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability during JSP compilation in Apache Tomcat per
CVE-2024-50613	libsndfile through 1.2.2 has a reachable assertion, that may lead to application exit, in mpeg_l3_encode.c mpeg_l3_e
CVE-2024-51504	When using IPAuthenticationProvider in ZooKeeper Admin Server there is a possibility of Authentication Bypass by
CVE-2024-51744	golang-jwt is a Go implementation of JSON Web Tokens. Unclear documentation of the error behavior in `ParseWith
CVE-2024-52003	Traefik (pronounced traffic) is an HTTP reverse proxy and load balancer. There is a vulnerability in Traefik that allo
CVE-2024-5187	A vulnerability in the `download_model_with_test_data` function of the onnx/onnx framework, version 1.16.0, allow
CVE-2024-52005	Git is a source code management tool. When cloning from a server (or fetching, or pushing), informational or error m
CVE-2024-52006	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high
CVE-2024-52046	The ObjectSerializationDecoder in Apache MINA uses Java's native deserialization protocol to process incoming se
CVE-2024-5206	A sensitive data leakage vulnerability was identified in scikit-learn's TfidfVectorizer, specifically in versions up to a
CVE-2024-52303	aiohttp is an asynchronous HTTP client/server framework for asyncio and Python. In versions starting with 3.10.6 an
CVE-2024-52304	aiohttp is an asynchronous HTTP client/server framework for asyncio and Python. Prior to version 3.10.11, the Pytho

CVE-2024-52332	In the Linux kernel, the following vulnerability has been resolved: igb: Fix potential invalid memory access in igb_in
CVE-2024-52533	gio/gsockets4aproxy.c in GNOME GLib before 2.82.1 has an off-by-one error and resultant buffer overflow because S
CVE-2024-52560	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Mark inode as bad as soon as error detec
CVE-2024-52559	In the Linux kernel, the following vulnerability has been resolved: drm/msm/gem: prevent integer overflow in msm_
CVE-2024-52616	A flaw was found in the Avahi-daemon, where it initializes DNS transaction IDs randomly only once at startup, incr
CVE-2024-52615	A flaw was found in Avahi-daemon, which relies on fixed source ports for wide-area DNS queries. This issue simpli
CVE-2024-52804	Tornado is a Python web framework and asynchronous networking library. The algorithm used for parsing HTTP co
CVE-2024-53042	In the Linux kernel, the following vulnerability has been resolved: ipv4: ip_tunnel: Fix suspicious RCU usage warni
CVE-2024-53043	In the Linux kernel, the following vulnerability has been resolved: mctp i2c: handle NULL header address daddr can
CVE-2024-53044	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_api: fix xa_insert() error path in tc
CVE-2024-53045	In the Linux kernel, the following vulnerability has been resolved: ASoC: dapm: fix bounds checker error in dapm_v
CVE-2024-53047	In the Linux kernel, the following vulnerability has been resolved: mptcp: init: protect sched with rcu_read_lock Ena
CVE-2024-53046	In the Linux kernel, the following vulnerability has been resolved: arm64: dts: imx8ulp: correct the flexspi compatib
CVE-2024-53048	In the Linux kernel, the following vulnerability has been resolved: ice: fix crash on probe for DPLL enabled E810 L
CVE-2024-53050	In the Linux kernel, the following vulnerability has been resolved: drm/i915/hdcp: Add encoder check in hdcp2_get_
CVE-2024-53051	In the Linux kernel, the following vulnerability has been resolved: drm/i915/hdcp: Add encoder check in intel_hdcp
CVE-2024-53052	In the Linux kernel, the following vulnerability has been resolved: io_uring/rw: fix missing NOWAIT check for O_I
CVE-2024-53053	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Fix another deadlock during RTC
CVE-2024-53054	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-53055	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: fix 6 GHz scan construction l
CVE-2024-53056	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: Fix potential NULL dereference in
CVE-2024-53058	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: TSO: Fix unbalanced DMA map/un
CVE-2024-53057	In the Linux kernel, the following vulnerability has been resolved: net/sched: stop qdisc_tree_reduce_backlog on TC
CVE-2024-53060	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: prevent NULL pointer dereference
CVE-2024-53059	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: Fix response handling in iwl_
CVE-2024-53061	In the Linux kernel, the following vulnerability has been resolved: media: s5p-jpeg: prevent buffer overflows The cu
CVE-2024-53062	In the Linux kernel, the following vulnerability has been resolved: media: mgb4: protect driver against spectre Frequ
CVE-2024-53063	In the Linux kernel, the following vulnerability has been resolved: media: dvbdev: prevent the risk of out of memory
CVE-2024-53064	In the Linux kernel, the following vulnerability has been resolved: idpf: fix idpf_vc_core_init error path In an event
CVE-2024-53066	In the Linux kernel, the following vulnerability has been resolved: nfs: Fix KMSAN warning in decode_getfattr_attr
CVE-2024-53067	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Start the RTC update work later Th
CVE-2024-53072	In the Linux kernel, the following vulnerability has been resolved: platform/x86/amd/pmc: Detect when STB is not a
CVE-2024-53068	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix slab-use-after-free in scr
CVE-2024-53073	In the Linux kernel, the following vulnerability has been resolved: NFSD: Never decrement pending_async_copies c
CVE-2024-53079	In the Linux kernel, the following vulnerability has been resolved: mm/thp: fix deferred split unqueue naming and lo
CVE-2024-53076	In the Linux kernel, the following vulnerability has been resolved: iio: gts-helper: Fix memory leaks for the error pat
CVE-2024-53081	In the Linux kernel, the following vulnerability has been resolved: media: ar0521: don't overflow when checking PL
CVE-2024-53082	In the Linux kernel, the following vulnerability has been resolved: virtio_net: Add hash_key_length check Add hash
CVE-2024-53083	In the Linux kernel, the following vulnerability has been resolved: usb: typec: qcom-pmic: init value of hdr_len/txbu
CVE-2024-53084	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: Break an object reference loop

CVE-2024-53085	In the Linux kernel, the following vulnerability has been resolved: tpm: Lock TPM chip in tpm_pm_suspend() first S
CVE-2024-53086	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Drop VM dma-resv lock on xe_sync_in_
CVE-2024-53087	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix possible exec queue leak in exec IOC
CVE-2024-53088	In the Linux kernel, the following vulnerability has been resolved: i40e: fix race condition by adding filter's intermed
CVE-2024-53089	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Mark hrtimer to expire in har
CVE-2024-53090	In the Linux kernel, the following vulnerability has been resolved: afs: Fix lock recursion afs_wake_up_async_call()
CVE-2024-53091	In the Linux kernel, the following vulnerability has been resolved: bpf: Add sk_is_inet and IS_ICSK check in tls_sw
CVE-2024-53093	In the Linux kernel, the following vulnerability has been resolved: nvme-multipath: defer partition scanning We need
CVE-2024-53094	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Add sendpage_ok() check to disable
CVE-2024-53095	In the Linux kernel, the following vulnerability has been resolved: smb: client: Fix use-after-free of network namesp
CVE-2024-53096	In the Linux kernel, the following vulnerability has been resolved: mm: resolve faulty mmap_region() error path beh
CVE-2024-53097	In the Linux kernel, the following vulnerability has been resolved: mm: krealloc: Fix MTE false alarm in __do_kreal
CVE-2024-53098	In the Linux kernel, the following vulnerability has been resolved: drm/xe/ufence: Prefetch ufence addr to catch bog
CVE-2024-53099	In the Linux kernel, the following vulnerability has been resolved: bpf: Check validity of link->type in bpf_link_sho
CVE-2024-53100	In the Linux kernel, the following vulnerability has been resolved: nvme: tcp: avoid race between queue_lock lock a
CVE-2024-53101	In the Linux kernel, the following vulnerability has been resolved: fs: Fix uninitialized value issue in from_kuid and
CVE-2024-53102	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-53103	In the Linux kernel, the following vulnerability has been resolved: hv_sock: Initializing vsk->trans to NULL to prev
CVE-2024-53104	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Skip parsing frames of type UV
CVE-2024-53105	In the Linux kernel, the following vulnerability has been resolved: mm: page_alloc: move mlocked flag clearance in
CVE-2024-53106	In the Linux kernel, the following vulnerability has been resolved: ima: fix buffer overrun in ima_eventdigest_init_c
CVE-2024-53107	In the Linux kernel, the following vulnerability has been resolved: fs/proc/task_mmu: prevent integer overflow in pa
CVE-2024-53108	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Adjust VSDB parser for replay
CVE-2024-53109	In the Linux kernel, the following vulnerability has been resolved: nommu: pass NULL argument to vma_iter_preall
CVE-2024-53110	In the Linux kernel, the following vulnerability has been resolved: vp_vdpa: fix id_table array not null terminated er
CVE-2024-53111	In the Linux kernel, the following vulnerability has been resolved: mm/mremap: fix address wraparound in move_pa
CVE-2024-53112	In the Linux kernel, the following vulnerability has been resolved: ocfs2: uncache inode which has failed entering th
CVE-2024-53113	In the Linux kernel, the following vulnerability has been resolved: mm: fix NULL pointer dereference in alloc_page
CVE-2024-53114	In the Linux kernel, the following vulnerability has been resolved: x86/CPU/AMD: Clear virtualized VMLOAD/VM
CVE-2024-53115	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: avoid null_ptr_deref in vmw_frame
CVE-2024-53117	In the Linux kernel, the following vulnerability has been resolved: virtio/vsock: Improve MSG_ZEROCOPY error h
CVE-2024-53118	In the Linux kernel, the following vulnerability has been resolved: vsock: Fix sk_error_queue memory leak Kernel q
CVE-2024-53119	In the Linux kernel, the following vulnerability has been resolved: virtio/vsock: Fix accept_queue memory leak As ti
CVE-2024-53120	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: CT: Fix null-ptr-deref in add rule err f
CVE-2024-53121	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: fs, lock FTE when checking if active Th
CVE-2024-53122	In the Linux kernel, the following vulnerability has been resolved: mptcp: cope racing subflow creation in mptcp_rcv
CVE-2024-53123	In the Linux kernel, the following vulnerability has been resolved: mptcp: error out earlier on disconnect Eric report
CVE-2024-53124	In the Linux kernel, the following vulnerability has been resolved: net: fix data-races around sk->sk_forward_alloc S
CVE-2024-53125	In the Linux kernel, the following vulnerability has been resolved: bpf: sync_linked_regs() must preserve subreg_de
CVE-2024-53126	In the Linux kernel, the following vulnerability has been resolved: vdpa: solidrun: Fix UB bug with devres In psnet_

CVE-2024-53127	In the Linux kernel, the following vulnerability has been resolved: Revert "mmc: dw_mmc: Fix IDMAC operation w
CVE-2024-53128	In the Linux kernel, the following vulnerability has been resolved: sched/task_stack: fix object_is_on_stack() for KA
CVE-2024-53129	In the Linux kernel, the following vulnerability has been resolved: drm/rockchip: vop: Fix a dereferenced before che
CVE-2024-53130	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix null-ptr-deref in block_dirty_buffer tra
CVE-2024-53131	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix null-ptr-deref in block_touch_buffer tra
CVE-2024-53133	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Handle dml allocation failure to
CVE-2024-53134	In the Linux kernel, the following vulnerability has been resolved: pmdomain: imx93-blk-ctrl: correct remove path T
CVE-2024-53135	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Bury Intel PT virtualization (guest/
CVE-2024-53138	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: kTLS, Fix incorrect page refcounting
CVE-2024-53136	In the Linux kernel, the following vulnerability has been resolved: mm: revert "mm: shmem: fix data-race in shmem
CVE-2024-53139	In the Linux kernel, the following vulnerability has been resolved: sctp: fix possible UAF in sctp_v6_available() A l
CVE-2024-53141	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: add missing range check in bitmap
CVE-2024-53140	In the Linux kernel, the following vulnerability has been resolved: netlink: terminate outstanding dump on socket clo
CVE-2024-53142	In the Linux kernel, the following vulnerability has been resolved: initramfs: avoid filename buffer overrun The initr
CVE-2024-53144	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: Align BR/EDR JUST_WO
CVE-2024-53145	In the Linux kernel, the following vulnerability has been resolved: um: Fix potential integer overflow during physme
CVE-2024-53147	In the Linux kernel, the following vulnerability has been resolved: exfat: fix out-of-bounds access of directory entrie
CVE-2024-53146	In the Linux kernel, the following vulnerability has been resolved: NFSD: Prevent a potential integer overflow If the
CVE-2024-53148	In the Linux kernel, the following vulnerability has been resolved: comedi: Flush partial mappings in error case If so
CVE-2024-53154	In the Linux kernel, the following vulnerability has been resolved: clk: clk-apple-nco: Add NULL check in applnco_
CVE-2024-53150	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix out of bounds reads when
CVE-2024-53151	In the Linux kernel, the following vulnerability has been resolved: svcrdma: Address an integer overflow Dan Carpe
CVE-2024-53155	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix uninitialized value in ocfs2_file_read_i
CVE-2024-53156	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k: add range check for conn_rsp_epid in
CVE-2024-53158	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: geni-se: fix array underflow in geni_s
CVE-2024-53160	In the Linux kernel, the following vulnerability has been resolved: rcu/kvfree: Fix data-race in __mod_timer / kvfree
CVE-2024-53157	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scpi: Check the DVFS OPP count
CVE-2024-53161	In the Linux kernel, the following vulnerability has been resolved: EDAC/bluefield: Fix potential integer overflow T
CVE-2024-53162	In the Linux kernel, the following vulnerability has been resolved: crypto: qat/qat_4xxx - fix off by one in uof_get_n
CVE-2024-53164	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix ordering of qlen adjustment Chang
CVE-2024-53163	In the Linux kernel, the following vulnerability has been resolved: crypto: qat/qat_420xx - fix off by one in uof_get_
CVE-2024-53166	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix bfqq uaf in bfq_limit_depth() Set n
CVE-2024-53165	In the Linux kernel, the following vulnerability has been resolved: sh: intc: Fix use-after-free bug in register_intc_co
CVE-2024-53169	In the Linux kernel, the following vulnerability has been resolved: nvme-fabrics: fix kernel crash while shutting dow
CVE-2024-53170	In the Linux kernel, the following vulnerability has been resolved: block: fix uaf for flush rq while iterating tags blk_
CVE-2024-53172	In the Linux kernel, the following vulnerability has been resolved: ubi: fastmap: Fix duplicate slab cache names whi
CVE-2024-53171	In the Linux kernel, the following vulnerability has been resolved: ubifs: authentication: Fix use-after-free in ubifs_t
CVE-2024-53168	In the Linux kernel, the following vulnerability has been resolved: sunrpc: fix one UAF issue caused by sunrpc kerno
CVE-2024-53173	In the Linux kernel, the following vulnerability has been resolved: NFSv4.0: Fix a use-after-free problem in the asyn
CVE-2024-53174	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: make sure cache entry active before ca

CVE-2024-53175	In the Linux kernel, the following vulnerability has been resolved: ipc: fix memleak if msg_init_ns failed in create_i
CVE-2024-53176	In the Linux kernel, the following vulnerability has been resolved: smb: During unmount, ensure all cached dir instan
CVE-2024-53177	In the Linux kernel, the following vulnerability has been resolved: smb: prevent use-after-free due to open_cached_c
CVE-2024-53178	In the Linux kernel, the following vulnerability has been resolved: smb: Don't leak cfid when reconnect races with o
CVE-2024-53180	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Add sanity NULL check for the defa
CVE-2024-53179	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix use-after-free of signing key Cust
CVE-2024-53181	In the Linux kernel, the following vulnerability has been resolved: um: vector: Do not use drvdata in release The drv
CVE-2024-53183	In the Linux kernel, the following vulnerability has been resolved: um: net: Do not use drvdata in release The drvdat
CVE-2024-53184	In the Linux kernel, the following vulnerability has been resolved: um: ubd: Do not use drvdata in release The drvda
CVE-2024-53187	In the Linux kernel, the following vulnerability has been resolved: io_uring: check for overflows in io_pin_pages W
CVE-2024-53185	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix NULL ptr deref in crypto_aead_s
CVE-2024-53190	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: Drastically reduce the attempts to re
CVE-2024-53188	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix crash when unbinding If there is
CVE-2024-53194	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix use-after-free of slot->bus on hot remov
CVE-2024-53191	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix warning when unbinding If there
CVE-2024-53195	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Get rid of userspace_irqchip_in_us
CVE-2024-53196	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Don't retire aborted MMIO instruct
CVE-2024-53197	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix potential out-of-bound acc
CVE-2024-53198	In the Linux kernel, the following vulnerability has been resolved: xen: Fix the issue of resource not being properly
CVE-2024-53200	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null check for pipe_ctx->pl
CVE-2024-53201	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null check for pipe_ctx->pl
CVE-2024-53202	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: Fix possible resource leak in fw
CVE-2024-53203	In the Linux kernel, the following vulnerability has been resolved: usb: typec: fix potential array underflow in ucsi_c
CVE-2024-53206	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix use-after-free of nreq in reqsk_timer_han
CVE-2024-53209	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix receive ring space parameters when
CVE-2024-53207	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix possible deadlocks This f
CVE-2024-53208	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix slab-use-after-free Read i
CVE-2024-5321	A security issue was discovered in Kubernetes clusters with Windows nodes where BUILTIN\Users may be able to o
CVE-2024-53210	In the Linux kernel, the following vulnerability has been resolved: s390/iucv: MSG_PEEK causes memory leak in iu
CVE-2024-53213	In the Linux kernel, the following vulnerability has been resolved: net: usb: lan78xx: Fix double free issue with inter
CVE-2024-53214	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Properly hide first-in-list PCIe extended
CVE-2024-53215	In the Linux kernel, the following vulnerability has been resolved: svcrdma: fix miss destroy percpu_counter in svc_
CVE-2024-53216	In the Linux kernel, the following vulnerability has been resolved: nfsd: release svc_expkey/svc_export with rcu_wc
CVE-2024-53217	In the Linux kernel, the following vulnerability has been resolved: NFSD: Prevent NULL dereference in nfsd4_proc
CVE-2024-53218	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix race in concurrent f2fs_stop_gc_thread In
CVE-2024-53220	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to account dirty data in __get_secs_requi
CVE-2024-53219	In the Linux kernel, the following vulnerability has been resolved: virtiofs: use pages instead of pointer for kernel di
CVE-2024-53221	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix null-ptr-deref in f2fs_submit_page_bio()
CVE-2024-53222	In the Linux kernel, the following vulnerability has been resolved: zram: fix NULL pointer in comp_algorithm_show
CVE-2024-53226	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix NULL pointer dereference in hns

CVE-2024-53223	In the Linux kernel, the following vulnerability has been resolved: clk: ralink: mtmips: fix clocks probe order in older
CVE-2024-53224	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Move events notifier registration to
CVE-2024-53227	In the Linux kernel, the following vulnerability has been resolved: scsi: bfa: Fix use-after-free in bfad_im_module_e
CVE-2024-53228	In the Linux kernel, the following vulnerability has been resolved: riscv: kvm: Fix out-of-bounds array access In kvm
CVE-2024-53229	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix the qp flush warnings in req Wh
CVE-2024-53231	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Fix possible null-ptr-deref for cpy
CVE-2024-53232	In the Linux kernel, the following vulnerability has been resolved: iommu/s390: Implement blocking domain This fi
CVE-2024-53230	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Fix possible null-ptr-deref for cpy
CVE-2024-53233	In the Linux kernel, the following vulnerability has been resolved: unicode: Fix utf8_load() error path utf8_load() re
CVE-2024-53234	In the Linux kernel, the following vulnerability has been resolved: erofs: handle NONHEAD !delta[1] lclusters grace
CVE-2024-53236	In the Linux kernel, the following vulnerability has been resolved: xsk: Free skb when TX metadata options are inva
CVE-2024-53237	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix use-after-free in device_for_each_c
CVE-2024-53239	In the Linux kernel, the following vulnerability has been resolved: ALSA: 6fire: Release resources at card release Th
CVE-2024-53240	In the Linux kernel, the following vulnerability has been resolved: xen/netfront: fix crash when removing device Wh
CVE-2024-53241	In the Linux kernel, the following vulnerability has been resolved: x86/xen: don't do PV irect hypercall through hyper
CVE-2024-53259	quic-go is an implementation of the QUIC protocol in Go. An off-path attacker can inject an ICMP Packet Too Large
CVE-2024-53427	decNumberCopy in decNumber.c in jq through 1.7.1 does not properly consider that NaN is interpreted as numeric, r
CVE-2024-53580	iperf v3.17.1 was discovered to contain a segmentation violation via the iperf_exchange_parameters() function.
CVE-2024-53680	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix UB due to uninitialized stack access in ip
CVE-2024-53685	In the Linux kernel, the following vulnerability has been resolved: ceph: give up on paths longer than PATH_MAX
CVE-2024-53687	In the Linux kernel, the following vulnerability has been resolved: riscv: Fix IPIs usage in kfence_protect_page() flu
CVE-2024-53690	In the Linux kernel, the following vulnerability has been resolved: nilfs2: prevent use of deleted inode syzbot reporte
CVE-2024-53861	pyjwt is a JSON Web Token implementation in Python. An incorrect string comparison is run for `iss` checking, res
CVE-2024-53899	virtualenv before 20.26.6 allows command injection through the activation scripts for a virtual environment. Magic
CVE-2024-53920	In elisp-mode.el in GNU Emacs before 30.1, a user who chooses to invoke elisp-completion-at-point (for code comp
CVE-2024-53981	python-multipart is a streaming multipart parser for Python. When parsing form data, python-multipart skips line bre
CVE-2024-54193	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix WARN in ivpu_ipc_send_receive
CVE-2024-54455	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix general protection fault in ivpu_b
CVE-2024-54456	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix potential buffer overflow in nfs_sysfs_li
CVE-2024-54460	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: iso: Fix circular lock in iso_listen_bis
CVE-2024-54534	The issue was addressed with improved memory handling. This issue is fixed in watchOS 11.2, visionOS 2.2, tvOS
CVE-2024-54683	In the Linux kernel, the following vulnerability has been resolved: netfilter: IDLETIMER: Fix for possible ABBA de
CVE-2024-55069	ffmpeg 7.1 is vulnerable to Null Pointer Dereference in function iamf_read_header in /libavformat/iamfdec.c.
CVE-2024-5535	Issue summary: Calling the OpenSSL API function SSL_select_next_proto with an empty supported client protocols
CVE-2024-55459	An issue in keras 3.7.0 allows attackers to write arbitrary files to the user's machine via downloading a crafted tar fil
CVE-2024-55549	xsltGetInheritedNsList in libxslt before 1.1.43 has a use-after-free issue related to exclusion of result prefixes.
CVE-2024-5569	A Denial of Service (DoS) vulnerability exists in the jaraco/zip library, affecting all versions prior to 3.19.1. The v
CVE-2024-55639	In the Linux kernel, the following vulnerability has been resolved: net: renesas: rswitch: avoid use-after-put for a dev
CVE-2024-55881	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Play nice with protected guests in con
CVE-2024-55916	In the Linux kernel, the following vulnerability has been resolved: Drivers: hv: util: Avoid accessing a ringbuffer no

CVE-2024-56128	Incorrect Implementation of Authentication Algorithm in Apache Kafka's SCRAM implementation. Issue Summary:
CVE-2024-56171	libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a use-after-free in xmlSchemaIDCFillNodeTables and xmlSchema
CVE-2024-56201	Jinja is an extensible templating engine. In versions on the 3.x branch prior to 3.1.5, a bug in the Jinja compiler allow
CVE-2024-56326	Jinja is an extensible templating engine. Prior to 3.1.5, An oversight in how the Jinja sandboxed environment detects
CVE-2024-56337	Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability in Apache Tomcat. This issue affects Apache
CVE-2024-56368	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix overflow in __rb_map_vma An o
CVE-2024-56369	In the Linux kernel, the following vulnerability has been resolved: drm/modes: Avoid divide by zero harder in drm_
CVE-2024-56372	In the Linux kernel, the following vulnerability has been resolved: net: tun: fix tun_napi_alloc_frags() syzbot reporte
CVE-2024-56374	An issue was discovered in Django 5.1 before 5.1.5, 5.0 before 5.0.11, and 4.2 before 4.2.18. Lack of upper-bound li
CVE-2024-56406	A heap buffer overflow vulnerability was discovered in Perl. Release branches 5.34, 5.36, 5.38 and 5.40 are affected
CVE-2024-56433	shadow-utils (aka shadow) 4.4 through 4.17.0 establishes a default /etc/subuid behavior (e.g., uid 100000 through 16
CVE-2024-5642	CPython 3.9 and earlier doesn't disallow configuring an empty list ([""]) for SSLContext.set_npn_protocols() which
CVE-2024-56531	In the Linux kernel, the following vulnerability has been resolved: ALSA: caiaq: Use snd_card_free_when_closed()
CVE-2024-56532	In the Linux kernel, the following vulnerability has been resolved: ALSA: us122l: Use snd_card_free_when_closed()
CVE-2024-56533	In the Linux kernel, the following vulnerability has been resolved: ALSA: usx2y: Use snd_card_free_when_closed()
CVE-2024-56538	In the Linux kernel, the following vulnerability has been resolved: drm: zynqmp_kms: Unplug DRM device before r
CVE-2024-56539	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Fix memcpy() field-spanning write
CVE-2024-56540	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Prevent recovery invocation during pr
CVE-2024-56543	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Skip Rx TID cleanup for self peer D
CVE-2024-56544	In the Linux kernel, the following vulnerability has been resolved: udmabuf: change folios array from kmalloc to kv
CVE-2024-56546	In the Linux kernel, the following vulnerability has been resolved: drivers: soc: xilinx: add the missing kfree in xlnx
CVE-2024-56545	In the Linux kernel, the following vulnerability has been resolved: HID: hyperv: streamline driver probe to avoid dev
CVE-2024-56548	In the Linux kernel, the following vulnerability has been resolved: hfsplus: don't query the device logical block size
CVE-2024-56549	In the Linux kernel, the following vulnerability has been resolved: cachefiles: Fix NULL pointer dereference in obje
CVE-2024-56550	In the Linux kernel, the following vulnerability has been resolved: s390/stacktrace: Use break instead of return stat
CVE-2024-56551	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix usage slab after free [+0.00002
CVE-2024-56552	In the Linux kernel, the following vulnerability has been resolved: drm/xe/guc_submit: fix race around suspend_pen
CVE-2024-56557	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ad7923: Fix buffer overflow for tx_buf a
CVE-2024-56558	In the Linux kernel, the following vulnerability has been resolved: nfsd: make sure exp active before svc_export_sho
CVE-2024-56561	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: Fix PCI domain ID release in pci_
CVE-2024-56565	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to drop all discards after creating snapshe
CVE-2024-56562	In the Linux kernel, the following vulnerability has been resolved: i3c: master: Fix miss free init_dyn_addr at i3c_m
CVE-2024-56566	In the Linux kernel, the following vulnerability has been resolved: mm/slub: Avoid list corruption when removing a
CVE-2024-56567	In the Linux kernel, the following vulnerability has been resolved: ad7780: fix division by zero in ad7780_write_raw
CVE-2024-56568	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Defer probe of clients after sm
CVE-2024-56569	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix regression with module command in st
CVE-2024-56570	In the Linux kernel, the following vulnerability has been resolved: ovl: Filter invalid inodes with missing lookup fun
CVE-2024-56572	In the Linux kernel, the following vulnerability has been resolved: media: platform: allegro-dvt: Fix possible memor
CVE-2024-56573	In the Linux kernel, the following vulnerability has been resolved: efi/libstub: Free correct pointer on failure cmdline
CVE-2024-56575	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Ensure power suppliers be susp

CVE-2024-56574	In the Linux kernel, the following vulnerability has been resolved: media: ts2020: fix null-ptr-deref in ts2020_probe
CVE-2024-56576	In the Linux kernel, the following vulnerability has been resolved: media: i2c: tc358743: Fix crash in the probe error
CVE-2024-56577	In the Linux kernel, the following vulnerability has been resolved: media: mtk-jpeg: Fix null-ptr-deref during unload
CVE-2024-56578	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Set video drvdata before register
CVE-2024-56579	In the Linux kernel, the following vulnerability has been resolved: media: amphion: Set video drvdata before register
CVE-2024-56580	In the Linux kernel, the following vulnerability has been resolved: media: qcom: camss: fix error path on configurati
CVE-2024-56581	In the Linux kernel, the following vulnerability has been resolved: btrfs: ref-verify: fix use-after-free after invalid ref
CVE-2024-56582	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free in btrfs_encoded_read_en
CVE-2024-56583	In the Linux kernel, the following vulnerability has been resolved: sched/deadline: Fix warning in migrate_enable fo
CVE-2024-56584	In the Linux kernel, the following vulnerability has been resolved: io_uring/tctx: work around xa_store() allocation e
CVE-2024-56586	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix f2fs_bug_on when uninstalling filesystem
CVE-2024-56587	In the Linux kernel, the following vulnerability has been resolved: leds: class: Protect brightness_show() with led_co
CVE-2024-56588	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Create all dump files during debug
CVE-2024-56589	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Add cond_resched() for no forced p
CVE-2024-56590	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Fix not checking skb length
CVE-2024-56591	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: Use disable_delayed_work
CVE-2024-56592	In the Linux kernel, the following vulnerability has been resolved: bpf: Call free_htab_elem() after htab_unlock_buc
CVE-2024-56593	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Fix oops due to NULL pointer de
CVE-2024-56594	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: set the right AMDGPU sg segment
CVE-2024-56595	In the Linux kernel, the following vulnerability has been resolved: jfs: add a check to prevent array-index-out-of-bou
CVE-2024-56596	In the Linux kernel, the following vulnerability has been resolved: jfs: fix array-index-out-of-bounds in jfs_readdir T
CVE-2024-56597	In the Linux kernel, the following vulnerability has been resolved: jfs: fix shift-out-of-bounds in dbSplit When dmt_
CVE-2024-56598	In the Linux kernel, the following vulnerability has been resolved: jfs: array-index-out-of-bounds fix in dtReadFirst
CVE-2024-56599	In the Linux kernel, the following vulnerability has been resolved: wifi: ath10k: avoid NULL pointer error during sd
CVE-2024-56601	In the Linux kernel, the following vulnerability has been resolved: net: inet: do not leave a dangling sk pointer in ine
CVE-2024-56600	In the Linux kernel, the following vulnerability has been resolved: net: inet6: do not leave a dangling sk pointer in in
CVE-2024-56602	In the Linux kernel, the following vulnerability has been resolved: net: ieee802154: do not leave a dangling sk point
CVE-2024-56603	In the Linux kernel, the following vulnerability has been resolved: net: af_can: do not leave a dangling sk pointer in
CVE-2024-56604	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: avoid leaving dangling sk
CVE-2024-56605	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: do not leave dangling sk poin
CVE-2024-56606	In the Linux kernel, the following vulnerability has been resolved: af_packet: avoid erroring out after sock_init_data
CVE-2024-56607	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix atomic calls in ath12k_mac_op_
CVE-2024-56608	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix out-of-bounds access in 'dc
CVE-2024-56609	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: use ieee80211_purge_tx_queue() to p
CVE-2024-56610	In the Linux kernel, the following vulnerability has been resolved: kcsan: Turn report_filterlist_lock into a raw_spin
CVE-2024-56611	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: fix migrate_to_node() assuming
CVE-2024-56613	In the Linux kernel, the following vulnerability has been resolved: sched/numa: fix memory leak due to the overwrit
CVE-2024-56614	In the Linux kernel, the following vulnerability has been resolved: xsk: fix OOB map writes when deleting elements
CVE-2024-56615	In the Linux kernel, the following vulnerability has been resolved: bpf: fix OOB devmap writes when deleting eleme
CVE-2024-56616	In the Linux kernel, the following vulnerability has been resolved: drm/dp_mst: Fix MST sideband message body len

CVE-2024-56617	In the Linux kernel, the following vulnerability has been resolved: cacheinfo: Allocate memory during CPU hotplug
CVE-2024-56619	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential out-of-bounds memory access
CVE-2024-56620	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: qcom: Only free platform MSIs when E
CVE-2024-56621	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Cancel RTC work during ufshcd_r
CVE-2024-56622	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: sysfs: Prevent div by zero Prevent
CVE-2024-56623	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix use after free on unload System
CVE-2024-56625	In the Linux kernel, the following vulnerability has been resolved: can: dev: can_set_termination(): allow sleeping G
CVE-2024-56626	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix Out-of-Bounds Write in ksmbd_vfs_st
CVE-2024-56627	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix Out-of-Bounds Read in ksmbd_vfs_st
CVE-2024-56629	In the Linux kernel, the following vulnerability has been resolved: HID: wacom: fix when get product name maybe n
CVE-2024-56630	In the Linux kernel, the following vulnerability has been resolved: ocfs2: free inode when ocfs2_get_init_inode() fai
CVE-2024-56631	In the Linux kernel, the following vulnerability has been resolved: scsi: sg: Fix slab-use-after-free read in sg_release
CVE-2024-56632	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix the memleak while create new ctrl
CVE-2024-56633	In the Linux kernel, the following vulnerability has been resolved: tcp_bpf: Fix the sk_mem_uncharge logic in tcp_b
CVE-2024-56634	In the Linux kernel, the following vulnerability has been resolved: gpio: grgpio: Add NULL check in grgpio_probe
CVE-2024-56635	In the Linux kernel, the following vulnerability has been resolved: net: avoid potential UAF in default_operstate() sy
CVE-2024-56636	In the Linux kernel, the following vulnerability has been resolved: geneve: do not assume mac header is set in geneve
CVE-2024-56637	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: Hold module reference while requ
CVE-2024-56638	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_inner: incorrect percpu area handling
CVE-2024-56639	In the Linux kernel, the following vulnerability has been resolved: net: hsr: must allocate more bytes for RedBox sup
CVE-2024-56640	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix LGR and link use-after-free issue We
CVE-2024-56641	In the Linux kernel, the following vulnerability has been resolved: net/smc: initialize close_work early to avoid warn
CVE-2024-56642	In the Linux kernel, the following vulnerability has been resolved: tipc: Fix use-after-free of kernel socket in cleanup
CVE-2024-56643	In the Linux kernel, the following vulnerability has been resolved: dccp: Fix memory leak in dccp_feat_change_recv
CVE-2024-56644	In the Linux kernel, the following vulnerability has been resolved: net/ipv6: release expired exception dst cached in s
CVE-2024-56645	In the Linux kernel, the following vulnerability has been resolved: can: j1939: j1939_session_new(): fix skb referenc
CVE-2024-56647	In the Linux kernel, the following vulnerability has been resolved: net: Fix icmp host relookup triggering ip_rt_bug
CVE-2024-56648	In the Linux kernel, the following vulnerability has been resolved: net: hsr: avoid potential out-of-bound access in fi
CVE-2024-56649	In the Linux kernel, the following vulnerability has been resolved: net: enetc: Do not configure preemptible TCs if S
CVE-2024-56650	In the Linux kernel, the following vulnerability has been resolved: netfilter: x_tables: fix LED ID check in led_tg_ch
CVE-2024-56651	In the Linux kernel, the following vulnerability has been resolved: can: hi311x: hi3110_can_ist(): fix potential use-a
CVE-2024-56652	In the Linux kernel, the following vulnerability has been resolved: drm/xe/reg_sr: Remove register pool That pool in
CVE-2024-56653	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btmtk: avoid UAF in btmtk_process_c
CVE-2024-56654	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: Fix using rcu_read_(un)loc
CVE-2024-56656	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix aggregation ID mask to prevent oop
CVE-2024-56657	In the Linux kernel, the following vulnerability has been resolved: ALSA: control: Avoid WARN() for symlink error
CVE-2024-56658	In the Linux kernel, the following vulnerability has been resolved: net: defer final 'struct net' free in netns dismantl
CVE-2024-56659	In the Linux kernel, the following vulnerability has been resolved: net: lapb: increase LAPB_HEADER_LEN It is un
CVE-2024-56660	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: DR, prevent potential error pointer dere
CVE-2024-56662	In the Linux kernel, the following vulnerability has been resolved: acpi: nfit: vmlalloc-out-of-bounds Read in acpi_nf

CVE-2024-56663	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: fix NL80211_ATTR_MLO_LINK
CVE-2024-56664	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix race between element replace a
CVE-2024-56665	In the Linux kernel, the following vulnerability has been resolved: bpf,perf: Fix invalid prog_array access in perf_ev
CVE-2024-56667	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix NULL pointer dereference in captu
CVE-2024-56670	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: u_serial: Fix the issue that gs_start i
CVE-2024-56671	In the Linux kernel, the following vulnerability has been resolved: gpio: graniterapids: Fix vGPIO driver crash Move
CVE-2024-56672	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: Fix UAF in blkcg_unpin_online() blk
CVE-2024-56674	In the Linux kernel, the following vulnerability has been resolved: virtio_net: correct netdev_tx_reset_queue() invoc
CVE-2024-56675	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix UAF via mismatching bpf_prog/attachme
CVE-2024-56677	In the Linux kernel, the following vulnerability has been resolved: powerpc/fadump: Move fadump_cma_init to setu
CVE-2024-56678	In the Linux kernel, the following vulnerability has been resolved: powerpc/mm/fault: Fix kfence page fault reportin
CVE-2024-56681	In the Linux kernel, the following vulnerability has been resolved: crypto: bcm - add error check in the ahash_hmac
CVE-2024-56679	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in
CVE-2024-56683	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: hdmi: Avoid hang with debug registers
CVE-2024-56687	In the Linux kernel, the following vulnerability has been resolved: usb: musb: Fix hardware lockup on first Rx endp
CVE-2024-56685	In the Linux kernel, the following vulnerability has been resolved: ASoC: mediatek: Check num_codecs is not zero t
CVE-2024-56688	In the Linux kernel, the following vulnerability has been resolved: sunrpc: clear XPRT_SOCKET_UPD_TIMEOUT wh
CVE-2024-56689	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: epf-mhi: Avoid NULL dereference
CVE-2024-56690	In the Linux kernel, the following vulnerability has been resolved: crypto: pccrypt - Call crypto layer directly when p
CVE-2024-56691	In the Linux kernel, the following vulnerability has been resolved: mfd: intel_soc_pmic_bxtwc: Use IRQ domain for
CVE-2024-56692	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on node blkaddr in tru
CVE-2024-56693	In the Linux kernel, the following vulnerability has been resolved: brd: defer automatic disk creation until module in
CVE-2024-56694	In the Linux kernel, the following vulnerability has been resolved: bpf: fix recursive lock when verdict program retu
CVE-2024-56698	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: Fix looping of queued SG entr
CVE-2024-56700	In the Linux kernel, the following vulnerability has been resolved: media: wl128x: Fix atomicity violation in fmc_se
CVE-2024-56701	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Fix dtl_access_lock to be a rw_
CVE-2024-56702	In the Linux kernel, the following vulnerability has been resolved: bpf: Mark raw_tp arguments with PTR_MAYBE
CVE-2024-56703	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix soft lockups in fib6_select_path under h
CVE-2024-56705	In the Linux kernel, the following vulnerability has been resolved: media: atomisp: Add check for rgb_data memor
CVE-2024-56704	In the Linux kernel, the following vulnerability has been resolved: 9p/xen: fix release of IRQ Kernel logs indicate an
CVE-2024-56707	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in
CVE-2024-56708	In the Linux kernel, the following vulnerability has been resolved: EDAC/igen6: Avoid segmentation fault on modul
CVE-2024-56709	In the Linux kernel, the following vulnerability has been resolved: io_uring: check if iowq is killed before queuing ta
CVE-2024-56710	In the Linux kernel, the following vulnerability has been resolved: ceph: fix memory leak in ceph_direct_read_writes
CVE-2024-56712	In the Linux kernel, the following vulnerability has been resolved: udmabuf: fix memory leak on last export_udmabu
CVE-2024-56715	In the Linux kernel, the following vulnerability has been resolved: ionic: Fix netdev notifier unregister on failure If r
CVE-2024-56716	In the Linux kernel, the following vulnerability has been resolved: netdevsim: prevent bad user input in nsim_dev_h
CVE-2024-56717	In the Linux kernel, the following vulnerability has been resolved: net: msc: ocelot: fix incorrect IFH SRC_PORT f
CVE-2024-56718	In the Linux kernel, the following vulnerability has been resolved: net/smc: protect link down work from execute aft
CVE-2024-56719	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: fix TSO DMA API usage causing o

CVE-2024-56720	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Several fixes to bpf_msg_pop_data
CVE-2024-56721	In the Linux kernel, the following vulnerability has been resolved: x86/CPU/AMD: Terminate the erratum_1386_mi
CVE-2024-56722	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix cpu stuck caused by printings du
CVE-2024-56723	In the Linux kernel, the following vulnerability has been resolved: mfd: intel_soc_pmic_bxtwc: Use IRQ domain for
CVE-2024-56724	In the Linux kernel, the following vulnerability has been resolved: mfd: intel_soc_pmic_bxtwc: Use IRQ domain for
CVE-2024-56725	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in
CVE-2024-56726	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in
CVE-2024-56727	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in
CVE-2024-56728	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in
CVE-2024-56729	In the Linux kernel, the following vulnerability has been resolved: smb: Initialize cfid->tcon before performing netw
CVE-2024-56739	In the Linux kernel, the following vulnerability has been resolved: rtc: check if __rtc_read_time was successful in rtc
CVE-2024-56744	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid potential deadlock in f2fs_recor
CVE-2024-56742	In the Linux kernel, the following vulnerability has been resolved: vfio/mlx5: Fix an unwind issue in mlx5vf_add_m
CVE-2024-56745	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix reset_method_store() memory leak In re
CVE-2024-56746	In the Linux kernel, the following vulnerability has been resolved: fbdev: sh7760fb: Fix a possible memory leak in s
CVE-2024-56747	In the Linux kernel, the following vulnerability has been resolved: scsi: qedi: Fix a possible memory leak in qedi_all
CVE-2024-56751	In the Linux kernel, the following vulnerability has been resolved: ipv6: release nexthop on device removal The CI i
CVE-2024-56748	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Fix a possible memory leak in qedf_all
CVE-2024-56752	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/gr/gf100: Fix missing unlock in gf1
CVE-2024-56755	In the Linux kernel, the following vulnerability has been resolved: netfs/fscache: Add a memory barrier for FSCACHE
CVE-2024-56754	In the Linux kernel, the following vulnerability has been resolved: crypto: caam - Fix the pointer passed to caam_qi
CVE-2024-56756	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: fix freeing of the HMB descriptor table
CVE-2024-56757	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btusb: mediatek: add intf release flow
CVE-2024-56758	In the Linux kernel, the following vulnerability has been resolved: btrfs: check folio mapping after unlock in relocat
CVE-2024-56759	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free when COWing tree bock a
CVE-2024-56760	In the Linux kernel, the following vulnerability has been resolved: PCI/MSI: Handle lack of irqdomain gracefully Al
CVE-2024-56761	In the Linux kernel, the following vulnerability has been resolved: x86/fred: Clear WFE in missing-ENDBRANCH #
CVE-2024-56763	In the Linux kernel, the following vulnerability has been resolved: tracing: Prevent bad count for tracing_cpumask_v
CVE-2024-56764	In the Linux kernel, the following vulnerability has been resolved: ublk: detach gendisk from ublk device if add_dish
CVE-2024-56765	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries/vas: Add close() callback in vas_
CVE-2024-56767	In the Linux kernel, the following vulnerability has been resolved: dmaengine: at_xdmac: avoid null_prt_deref in at_
CVE-2024-56769	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: dib3000mb: fix uninit-valu
CVE-2024-56770	In the Linux kernel, the following vulnerability has been resolved: net/sched: netem: account for backlog updates fro
CVE-2024-56771	In the Linux kernel, the following vulnerability has been resolved: mtd: spinand: winbond: Fix 512GW, 01GW, 01JW
CVE-2024-56772	In the Linux kernel, the following vulnerability has been resolved: kunit: string-stream: Fix a UAF bug in kunit_init
CVE-2024-56773	In the Linux kernel, the following vulnerability has been resolved: kunit: Fix potential null dereference in kunit_dev
CVE-2024-56774	In the Linux kernel, the following vulnerability has been resolved: btrfs: add a sanity check for btrfs root in btrfs_se
CVE-2024-56775	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix handling of plane refcount
CVE-2024-56776	In the Linux kernel, the following vulnerability has been resolved: drm/sti: avoid potential dereference of error point
CVE-2024-56777	In the Linux kernel, the following vulnerability has been resolved: drm/sti: avoid potential dereference of error point

CVE-2024-56778	In the Linux kernel, the following vulnerability has been resolved: drm/sti: avoid potential dereference of error pointer
CVE-2024-56779	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix nfs4_openowner leak when concurrent m
CVE-2024-56780	In the Linux kernel, the following vulnerability has been resolved: quota: flush quota_release_work upon quota write
CVE-2024-56781	In the Linux kernel, the following vulnerability has been resolved: powerpc/prom_init: Fixup missing powermac #si
CVE-2024-56782	In the Linux kernel, the following vulnerability has been resolved: ACPI: x86: Add adev NULL check to acpi_quirk
CVE-2024-56783	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_socket: remove WARN_ON_ONCE
CVE-2024-56784	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Adding array index check to pr
CVE-2024-56785	In the Linux kernel, the following vulnerability has been resolved: MIPS: Loongson64: DTS: Really fix PCIe port n
CVE-2024-56786	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-56787	In the Linux kernel, the following vulnerability has been resolved: soc: imx8m: Probe the SoC driver as platform dri
CVE-2024-56788	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: oa_tc6: fix tx skb race condition bet
CVE-2024-57360	https://www.gnu.org/software/binutils/ nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation
CVE-2024-57699	A security issue was found in Netplex Json-smart 2.5.0 through 2.5.1. When loading a specially crafted JSON input,
CVE-2024-5742	A vulnerability was found in GNU Nano that allows a possible privilege escalation through an insecure temporary fi
CVE-2024-57791	In the Linux kernel, the following vulnerability has been resolved: net/smc: check return value of sock_recvmg
CVE-2024-57792	In the Linux kernel, the following vulnerability has been resolved: power: supply: gpio-charger: Fix set charge curre
CVE-2024-57793	In the Linux kernel, the following vulnerability has been resolved: virt: tdx-guest: Just leak decrypted memory on un
CVE-2024-57795	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Remove the direct link to net_device
CVE-2024-57798	In the Linux kernel, the following vulnerability has been resolved: drm/dp_mst: Ensure mst_primary pointer is valid
CVE-2024-57801	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Skip restore TC rules for vport rep wit
CVE-2024-57802	In the Linux kernel, the following vulnerability has been resolved: netrom: check buffer length before accessing it S
CVE-2024-57804	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix corrupt config pages PHY state
CVE-2024-57805	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda-dai: Do not release the lin
CVE-2024-57807	In the Linux kernel, the following vulnerability has been resolved: scsi: megaraid_sas: Fix for a potential deadlock T
CVE-2024-57806	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix transaction atomicity bug when enabling
CVE-2024-57809	In the Linux kernel, the following vulnerability has been resolved: PCI: imx6: Fix suspend/resume support on i.MX6
CVE-2024-57834	In the Linux kernel, the following vulnerability has been resolved: media: vidtv: Fix a null-ptr-deref in vidtv_mux_s
CVE-2024-57838	In the Linux kernel, the following vulnerability has been resolved: s390/entry: Mark IRQ entries to fix stack depot w
CVE-2024-57841	In the Linux kernel, the following vulnerability has been resolved: net: fix memory leak in tcp_conn_request() If ine
CVE-2024-57843	In the Linux kernel, the following vulnerability has been resolved: virtio-net: fix overflow inside virtnet_rq_alloc W
CVE-2024-57849	In the Linux kernel, the following vulnerability has been resolved: s390/cpum_sf: Handle CPU hotplug remove durin
CVE-2024-57850	In the Linux kernel, the following vulnerability has been resolved: jffs2: Prevent rtime decompress memory corrupti
CVE-2024-57852	In the Linux kernel, the following vulnerability has been resolved: firmware: qcom: scm: smc: Handle missing SCM
CVE-2024-57857	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Remove direct link to net_device Do
CVE-2024-57872	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: pltfm: Dellocate HBA during ufshcd_p
CVE-2024-57874	In the Linux kernel, the following vulnerability has been resolved: arm64: ptrace: fix partial SETREGSET for NT_A
CVE-2024-57875	In the Linux kernel, the following vulnerability has been resolved: block: RCU protect disk->conv_zones_bitmap Er
CVE-2024-57876	In the Linux kernel, the following vulnerability has been resolved: drm/dp_mst: Fix resetting msg rx state after topol
CVE-2024-57879	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: iso: Always release hdev at the end of
CVE-2024-57882	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix TCP options overflow. Syzbot reporte

CVE-2024-57884	In the Linux kernel, the following vulnerability has been resolved: mm: vmscan: account for free pages to prevent in
CVE-2024-57883	In the Linux kernel, the following vulnerability has been resolved: mm: hugetlb: independent PMD page table shared
CVE-2024-57885	In the Linux kernel, the following vulnerability has been resolved: mm/kmemleak: fix sleeping function called from
CVE-2024-57888	In the Linux kernel, the following vulnerability has been resolved: workqueue: Do not warn when cancelling WQ_M
CVE-2024-57887	In the Linux kernel, the following vulnerability has been resolved: drm: adv7511: Fix use-after-free in adv7533_atta
CVE-2024-57889	In the Linux kernel, the following vulnerability has been resolved: pinctrl: mcp23s08: Fix sleeping in atomic context
CVE-2024-57890	In the Linux kernel, the following vulnerability has been resolved: RDMA/uverbs: Prevent integer overflow issue In
CVE-2024-57892	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix slab-use-after-free due to dangling poin
CVE-2024-57893	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: oss: Fix races at processing SysEx m
CVE-2024-57894	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-57896	In the Linux kernel, the following vulnerability has been resolved: btrfs: flush delalloc workers queue before stoppin
CVE-2024-57895	In the Linux kernel, the following vulnerability has been resolved: ksmbd: set ATTR_CTIME flags when setting mtm
CVE-2024-57898	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: clear link ID from bitmap during
CVE-2024-57897	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Correct the migration DMA map di
CVE-2024-57899	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix mbss changed flags corruptio
CVE-2024-57900	In the Linux kernel, the following vulnerability has been resolved: ila: serialize calls to nf_register_net_hooks() syzbu
CVE-2024-57902	In the Linux kernel, the following vulnerability has been resolved: af_packet: fix vlan_get_tci() vs MSG_PEEK Blan
CVE-2024-57903	In the Linux kernel, the following vulnerability has been resolved: net: restrict SO_REUSEPORT to inet sockets Aft
CVE-2024-57904	In the Linux kernel, the following vulnerability has been resolved: iio: adc: at91: call input_free_device() on allocate
CVE-2024-57901	In the Linux kernel, the following vulnerability has been resolved: af_packet: fix vlan_get_protocol_dgram() vs MSG
CVE-2024-57906	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ti-ads8688: fix information leak in trigge
CVE-2024-57907	In the Linux kernel, the following vulnerability has been resolved: iio: adc: rockchip_saradc: fix information leak in
CVE-2024-57908	In the Linux kernel, the following vulnerability has been resolved: iio: imu: kmx61: fix information leak in triggered
CVE-2024-57910	In the Linux kernel, the following vulnerability has been resolved: iio: light: vcnl4035: fix information leak in trigge
CVE-2024-57911	In the Linux kernel, the following vulnerability has been resolved: iio: dummy: iio_simply_dummy_buffer: fix infor
CVE-2024-57912	In the Linux kernel, the following vulnerability has been resolved: iio: pressure: zpa2326: fix information leak in trig
CVE-2024-57913	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_fs: Remove WARN_ON in functio
CVE-2024-57916	In the Linux kernel, the following vulnerability has been resolved: misc: microchip: pci1xxx: Resolve kernel panic
CVE-2024-57922	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add check for granularity in dr
CVE-2024-57917	In the Linux kernel, the following vulnerability has been resolved: topology: Keep the cpumask unchanged when pri
CVE-2024-57924	In the Linux kernel, the following vulnerability has been resolved: fs: relax assertions on failure to encode file handl
CVE-2024-57925	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix a missing return value check bug In th
CVE-2024-57926	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: Set private->all_drm_private[i]->c
CVE-2024-57930	In the Linux kernel, the following vulnerability has been resolved: tracing: Have process_string() also allow arrays I
CVE-2024-57931	In the Linux kernel, the following vulnerability has been resolved: selinux: ignore unknown extended permissions W
CVE-2024-57929	In the Linux kernel, the following vulnerability has been resolved: dm array: fix releasing a faulty array block twice
CVE-2024-57932	In the Linux kernel, the following vulnerability has been resolved: gve: guard XDP xmit NDO on existence of xdp q
CVE-2024-57938	In the Linux kernel, the following vulnerability has been resolved: net/sctp: Prevent autoclose integer overflow in sc
CVE-2024-57933	In the Linux kernel, the following vulnerability has been resolved: gve: guard XSK operations on the existence of qu
CVE-2024-57939	In the Linux kernel, the following vulnerability has been resolved: riscv: Fix sleeping in invalid context in die() die()

CVE-2024-57940	In the Linux kernel, the following vulnerability has been resolved: exfat: fix the infinite loop in exfat_readdir() If the
CVE-2024-57945	In the Linux kernel, the following vulnerability has been resolved: riscv: mm: Fix the out of bound issue of vmemman
CVE-2024-57946	In the Linux kernel, the following vulnerability has been resolved: virtio-blk: don't keep queue frozen during system
CVE-2024-57947	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_set_pipapo: fix initial map fill The in
CVE-2024-57948	In the Linux kernel, the following vulnerability has been resolved: mac802154: check local interfaces before deleting
CVE-2024-57949	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3-its: Don't enable interrupts in its_in
CVE-2024-57950	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize denominator defaults
CVE-2024-57951	In the Linux kernel, the following vulnerability has been resolved: hrtimers: Handle CPU state correctly on hotplug
CVE-2024-57973	In the Linux kernel, the following vulnerability has been resolved: rdma/cxgb4: Prevent potential integer overflow o
CVE-2024-57952	In the Linux kernel, the following vulnerability has been resolved: Revert "libfs: fix infinite directory reads for offse
CVE-2024-57970	libarchive through 3.7.7 has a heap-based buffer over-read in header_gnu_longlink in archive_read_support_format
CVE-2024-57974	In the Linux kernel, the following vulnerability has been resolved: udp: Deal with race between UDP socket address
CVE-2024-57975	In the Linux kernel, the following vulnerability has been resolved: btrfs: do proper folio cleanup when run_delalloc_
CVE-2024-57978	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Fix potential error pointer deref
CVE-2024-57976	In the Linux kernel, the following vulnerability has been resolved: btrfs: do proper folio cleanup when cow_file_ran
CVE-2024-57979	In the Linux kernel, the following vulnerability has been resolved: pps: Fix a use-after-free On a board running ntpd
CVE-2024-57977	In the Linux kernel, the following vulnerability has been resolved: memcg: fix soft lockup in the OOM process A so
CVE-2024-57980	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Fix double free in error path If t
CVE-2024-57981	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Fix NULL pointer dereference on certa
CVE-2024-57982	In the Linux kernel, the following vulnerability has been resolved: xfrm: state: fix out-of-bounds read during lookup
CVE-2024-57986	In the Linux kernel, the following vulnerability has been resolved: HID: core: Fix assumption that Resolution Multip
CVE-2024-57984	In the Linux kernel, the following vulnerability has been resolved: i3c: dw: Fix use-after-free in dw_i3c_master driv
CVE-2024-57993	In the Linux kernel, the following vulnerability has been resolved: HID: hid-thrustmaster: Fix warning in thrustmast
CVE-2024-57996	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: don't allow 1 packet limit Th
CVE-2024-57999	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries/iommu: IOMMU incorrectly ma
CVE-2024-58002	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Remove dangling pointers Whe
CVE-2024-58001	In the Linux kernel, the following vulnerability has been resolved: ocfs2: handle a symlink read error correctly Patch
CVE-2024-58005	In the Linux kernel, the following vulnerability has been resolved: tpm: Change to kcalloc() in eventlog/acpi.c The f
CVE-2024-58007	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: socinfo: Avoid out of bounds read of s
CVE-2024-58010	In the Linux kernel, the following vulnerability has been resolved: binfmt_flat: Fix integer overflow bug on 32 bit sy
CVE-2024-58011	In the Linux kernel, the following vulnerability has been resolved: platform/x86: int3472: Check for adev == NULL
CVE-2024-58012	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda-dai: Ensure DAI widget is
CVE-2024-58013	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix slab-use-after-free Read i
CVE-2024-58014	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmsmac: add gain range check to wlc_phy
CVE-2024-58015	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix for out-of bound access error Se
CVE-2024-58016	In the Linux kernel, the following vulnerability has been resolved: safesetid: check size of policy writes syzbot att
CVE-2024-58017	In the Linux kernel, the following vulnerability has been resolved: printk: Fix signed integer overflow when defining
CVE-2024-58018	In the Linux kernel, the following vulnerability has been resolved: nvkm: correctly calculate the available space of th
CVE-2024-58019	In the Linux kernel, the following vulnerability has been resolved: nvkm/gsp: correctly advance the read pointer of C
CVE-2024-58034	In the Linux kernel, the following vulnerability has been resolved: memory: tegra20-emc: fix an OF node reference l

CVE-2024-58020	In the Linux kernel, the following vulnerability has been resolved: HID: multitouch: Add NULL check in mt_input_
CVE-2024-58051	In the Linux kernel, the following vulnerability has been resolved: ipmi: ipmb: Add check devm_kasprintf() returned
CVE-2024-58052	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential NULL pointer derefer
CVE-2024-58053	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix handling of received connection abort
CVE-2024-58055	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_tcm: Don't free command immedi
CVE-2024-58054	In the Linux kernel, the following vulnerability has been resolved: staging: media: max96712: fix kernel oops when
CVE-2024-58057	In the Linux kernel, the following vulnerability has been resolved: idpf: convert workqueues to unbound When a wo
CVE-2024-58061	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: prohibit deactivating all links In
CVE-2024-58058	In the Linux kernel, the following vulnerability has been resolved: ubifs: skip dumping tnc tree when zroot is null Cl
CVE-2024-58063	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: fix memory leaks and invalid access
CVE-2024-58069	In the Linux kernel, the following vulnerability has been resolved: rtc: pcf85063: fix potential OOB write in PCF850
CVE-2024-58071	In the Linux kernel, the following vulnerability has been resolved: team: prevent adding a device which is already a
CVE-2024-58076	In the Linux kernel, the following vulnerability has been resolved: clk: qcom: gcc-sm6350: Add missing parent_map
CVE-2024-58072	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: remove unused check_buddy_priv C
CVE-2024-58078	In the Linux kernel, the following vulnerability has been resolved: misc: misc_minor_alloc to use ida for all dynamic
CVE-2024-58077	In the Linux kernel, the following vulnerability has been resolved: ASoC: soc-pcm: don't use soc_pcm_ret() on .prep
CVE-2024-58079	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Fix crash during unbind if gpio
CVE-2024-58083	In the Linux kernel, the following vulnerability has been resolved: KVM: Explicitly verify target vCPU is online in l
CVE-2024-58087	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix racy issue from session lookup and ex
CVE-2024-58085	In the Linux kernel, the following vulnerability has been resolved: tomooyo: don't emit warning in tomooyo_write_con
CVE-2024-58086	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Stop active perfmon if it is being destro
CVE-2024-58089	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix double accounting race when btrfs_run_
CVE-2024-58090	In the Linux kernel, the following vulnerability has been resolved: sched/core: Prevent rescheduling when interrupts
CVE-2024-58092	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix legacy client tracking initialization Get r
CVE-2024-58094	In the Linux kernel, the following vulnerability has been resolved: jfs: add check read-only before truncation in jfs_t
CVE-2024-58095	In the Linux kernel, the following vulnerability has been resolved: jfs: add check read-only before txBeginAnon() ca
CVE-2024-58093	In the Linux kernel, the following vulnerability has been resolved: PCI/ASPM: Fix link state exit during switch upstr
CVE-2024-58096	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: add srng->lock for ath11k_hal_srng
CVE-2024-58097	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix RCU stall while reaping monitor
CVE-2024-58098	In the Linux kernel, the following vulnerability has been resolved: bpf: track changes_pkt_data property for global fu
CVE-2024-58099	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: Fix packet corruption in vmxnet3_xdp_
CVE-2024-58100	In the Linux kernel, the following vulnerability has been resolved: bpf: check changes_pkt_data property for extensi
CVE-2024-58237	In the Linux kernel, the following vulnerability has been resolved: bpf: consider that tail calls invalidate packet point
CVE-2024-5998	A vulnerability in the FAISS.deserialize_from_bytes function of langchain-ai/langchain allows for pickle deserializa
CVE-2024-6119	Issue summary: Applications performing certificate name checks (e.g., TLS clients checking server certificates) may
CVE-2024-6104	go-retryablehttp prior to 0.7.7 did not sanitize urls when writing them to its log file. This could lead to go-retryableh
CVE-2024-6221	A vulnerability in corydolphin/flask-cors version 4.0.1 allows the `Access-Control-Allow-Private-Network` CORS h
CVE-2024-6232	There is a MEDIUM severity vulnerability affecting CPython. Regular expressions that allowed excessive backtrack
CVE-2024-6345	A vulnerability in the package_index module of pypa/setuptools versions up to 69.1.1 allows for remote code execut
CVE-2024-6602	A mismatch between allocator and deallocator could have led to memory corruption. This vulnerability affects Firefo

CVE-2024-6655	A flaw was found in the GTK library. Under certain conditions, it is possible for a library to be injected into a GTK application.
CVE-2024-6763	Eclipse Jetty is a lightweight, highly scalable, Java-based web server and Servlet engine . It includes a utility class, <code>HttpServer</code> .
CVE-2024-6839	corydolphin/flask-cors version 4.0.1 contains an improper regex path matching vulnerability. The plugin prioritizes <code>OPTIONS</code> requests.
CVE-2024-6827	Gunicorn version 21.2.0 does not properly validate the value of the 'Transfer-Encoding' header as specified in the RFC 7230.
CVE-2024-6844	A vulnerability in corydolphin/flask-cors version 4.0.1 allows for inconsistent CORS matching due to the handling of <code>OPTIONS</code> requests.
CVE-2024-6866	corydolphin/flask-cors version 4.0.1 contains a vulnerability where the request path matching is case-insensitive due to the handling of <code>OPTIONS</code> requests.
CVE-2024-6923	There is a MEDIUM severity vulnerability affecting CPython. The email module didn't properly quote newlines for email headers.
CVE-2024-7006	A null pointer dereference flaw was found in Libtiff via <code>`tif_dirinfo.c`</code> . This issue may allow an attacker to trigger a crash or denial of service.
CVE-2024-7055	A vulnerability was found in FFmpeg up to 7.0.1. It has been classified as critical. This affects the function <code>pnm_decode</code> .
CVE-2024-7254	Any project that parses untrusted Protocol Buffers data containing an arbitrary number of nested groups / series of <code>SerializedData</code> objects.
CVE-2024-7264	libcurl's ASN1 parser code has the <code>`GTime2str()</code> ` function, used for parsing an ASN.1 Generalized Time field. If given an invalid input, it can cause a crash.
CVE-2024-7348	Time-of-check Time-of-use (TOCTOU) race condition in <code>pg_dump</code> in PostgreSQL allows an object creator to execute arbitrary SQL commands.
CVE-2024-7531	Calling <code>`PK11_Encrypt()</code> ` in NSS using CKM_CHACHA20 and the same buffer for input and output can result in a crash or denial of service.
CVE-2024-7592	There is a LOW severity vulnerability affecting CPython, specifically the 'http.cookies' standard library module. When parsing cookies, it can cause a crash.
CVE-2024-7776	A vulnerability in the <code>`download_model`</code> function of the onnx/onnx framework, before and including version 1.16.1, allows an attacker to trigger a crash or denial of service.
CVE-2024-7804	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-8088	There is a HIGH severity vulnerability affecting the CPython "zipfile" module affecting "zipfile.Path". Note that the vulnerability is only present in the <code>Path</code> class.
CVE-2024-8096	When curl is told to use the Certificate Status Request TLS extension, often referred to as OCSP stapling, to verify the certificate, it can cause a crash or denial of service.
CVE-2024-8176	A stack overflow vulnerability exists in the libexpat library due to the way it handles recursive entity expansion in XML documents.
CVE-2024-8184	There exists a security vulnerability in Jetty's ThreadLimitHandler.getRemote() which can be exploited by unauthorized users to cause a denial of service.
CVE-2024-8309	A vulnerability in the GraphCypherQACHain class of langchain-ai/langchain version 0.2.5 allows for SQL injection attacks.
CVE-2024-8768	A flaw was found in the vLLM library. A completions API request with an empty prompt will crash the vLLM API server.
CVE-2024-8805	BlueZ HID over GATT Profile Improper Access Control Remote Code Execution Vulnerability. This vulnerability allows an attacker to execute arbitrary code on a remote device.
CVE-2024-9042	This CVE affects only Windows worker nodes. Your worker node is vulnerable to this issue if it is running one of the affected versions.
CVE-2024-9052	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2024-9053	vllm-project vllm version 0.6.0 contains a vulnerability in the AsyncEngineRPCServer() RPC server endpoints. This allows an attacker to trigger a crash or denial of service.
CVE-2024-9143	Issue summary: Use of the low-level GF(2^m) elliptic curve APIs with untrusted explicit values for the field polynomial can lead to a crash or denial of service.
CVE-2024-9287	A vulnerability has been found in the CPython <code>`venv`</code> module and CLI where path names provided when creating a virtual environment can be exploited.
CVE-2024-9681	When curl is asked to use HSTS, the expiry time for a subdomain might overwrite a parent domain's cache entry, making the subdomain's cache entry invalid.
CVE-2025-0167	When asked to use a <code>`.netrc`</code> file for credentials <code>**and**</code> to follow HTTP redirects, curl could leak the password used for authentication.
CVE-2024-9823	There exists a security vulnerability in Jetty's DosFilter which can be exploited by unauthorized users to cause a denial of service.
CVE-2025-0291	Type Confusion in V8 in Google Chrome prior to 131.0.6778.264 allowed a remote attacker to execute arbitrary code on the victim's machine.
CVE-2025-0395	When the <code>assert()</code> function in the GNU C Library versions 2.13 to 2.40 fails, it does not allocate enough space for the message, leading to a crash or denial of service.
CVE-2025-0426	A security issue was discovered in Kubernetes where a large number of container checkpoint requests made to the <code>ContainerCheckpointManager</code> could cause a denial of service.
CVE-2025-0434	Out of bounds memory access in V8 in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially execute arbitrary code on the victim's machine.
CVE-2025-0435	Inappropriate implementation in Navigation in Google Chrome on Android prior to 132.0.6834.83 allowed a remote attacker to potentially execute arbitrary code on the victim's machine.
CVE-2025-0436	Integer overflow in Skia in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow to execute arbitrary code on the victim's machine.
CVE-2025-0437	Out of bounds read in Metrics in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow to execute arbitrary code on the victim's machine.
CVE-2025-0438	Stack buffer overflow in Tracing in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow to execute arbitrary code on the victim's machine.
CVE-2025-0439	Race in Frames in Google Chrome prior to 132.0.6834.83 allowed a remote attacker who convinced a user to engage in a phishing attack to execute arbitrary code on the victim's machine.

CVE-2025-0440	Inappropriate implementation in Fullscreen in Google Chrome on Windows prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0441	Inappropriate implementation in Fenced Frames in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0442	Inappropriate implementation in Payments in Google Chrome prior to 132.0.6834.83 allowed a remote attacker who could interact with the browser to potentially exploit heap overflow
CVE-2025-0443	Insufficient data validation in Extensions in Google Chrome prior to 132.0.6834.83 allowed a remote attacker who could interact with the browser to potentially exploit heap overflow
CVE-2025-0444	Use after free in Skia in Google Chrome prior to 133.0.6943.53 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0445	Use after free in V8 in Google Chrome prior to 133.0.6943.53 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0446	Inappropriate implementation in Extensions in Google Chrome prior to 132.0.6834.83 allowed a remote attacker who could interact with the browser to potentially exploit heap overflow
CVE-2025-0447	Inappropriate implementation in Navigation in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0451	Inappropriate implementation in Extensions API in Google Chrome prior to 133.0.6943.53 allowed a remote attacker who could interact with the browser to potentially exploit heap overflow
CVE-2025-0518	Unchecked Return Value, Out-of-bounds Read vulnerability in FFmpeg allows Read Sensitive Constants Within an array
CVE-2025-0448	Inappropriate implementation in Compositing in Google Chrome prior to 132.0.6834.83 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0611	Object corruption in V8 in Google Chrome prior to 132.0.6834.110 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0612	Out of bounds memory access in V8 in Google Chrome prior to 132.0.6834.110 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0665	libcurl would wrongly close the same eventfd file descriptor twice when taking down a connection channel after having been used to connect to a server
CVE-2025-0762	Use after free in DevTools in Google Chrome prior to 132.0.6834.159 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0725	When libcurl is asked to perform automatic gzip decompression of content-encoded HTTP responses with the `CURLOPT_HTTP_TRANSFER_ENCODING` option, it would fail to properly handle responses that are not gzip encoded
CVE-2025-0840	A vulnerability, which was classified as problematic, was found in GNU Binutils up to 2.43. This affects the function <code>binutils_elf_relocate_section</code>
CVE-2025-0913	<code>os.OpenFile(path, os.O_CREATE O_EXCL)</code> behaved differently on Unix and Windows systems when the target path exists and is a directory
CVE-2025-0927	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority. Filesystem bugs due to race conditions
CVE-2025-0938	The Python standard library functions <code>`urllib.parse.urlsplit`</code> and <code>`urlparse`</code> accepted domain names that included square brackets
CVE-2025-0995	Use after free in V8 in Google Chrome prior to 133.0.6943.98 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0996	Inappropriate implementation in Browser UI in Google Chrome on Android prior to 133.0.6943.98 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0997	Use after free in Navigation in Google Chrome prior to 133.0.6943.98 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-0999	Heap buffer overflow in V8 in Google Chrome prior to 133.0.6943.126 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-1006	Use after free in Network in Google Chrome prior to 133.0.6943.126 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-1094	Improper neutralization of quoting syntax in PostgreSQL libpq functions <code>PQescapeLiteral()</code> , <code>PQescapeIdentifier()</code> , <code>PQescapeString()</code> , and <code>PQescapeStringConn()</code>
CVE-2025-1097	A security issue was discovered in ingress-nginx https://github.com/kubernetes/ingress-nginx where the <code>`auth-tls-map`</code> annotation was not properly validated
CVE-2025-1098	A security issue was discovered in ingress-nginx https://github.com/kubernetes/ingress-nginx where the <code>`mirror-target`</code> annotation was not properly validated
CVE-2025-1150	A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function <code>binutils_elf_relocate_section</code>
CVE-2025-1152	A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function <code>xstrdup</code> of the <code>binutils_elf_relocate_section</code>
CVE-2025-1153	A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function <code>binutils_elf_relocate_section</code>
CVE-2025-1151	A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function <code>xmbrlcpy</code>
CVE-2025-1176	A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function <code>_bfd_elf_get_relocation_type</code>
CVE-2025-1178	A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function <code>binutils_elf_relocate_section</code>
CVE-2025-1181	A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function <code>_bfd_elf_get_relocation_type</code>
CVE-2025-1182	A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function <code>bfd_elf_relocate_section</code>
CVE-2025-1194	A Regular Expression Denial of Service (ReDoS) vulnerability was identified in the huggingface/transformers library
CVE-2025-1215	A vulnerability classified as problematic was found in vim up to 9.1.1096. This vulnerability affects unknown code of the <code>vim</code> editor
CVE-2025-1244	A command injection flaw was found in the text editor Emacs. It could allow a remote, unauthenticated attacker to execute arbitrary commands
CVE-2025-1352	A vulnerability has been found in GNU elfutils 0.192 and classified as critical. This vulnerability affects the function <code>elfutils_elf_relocate_section</code>

CVE-2025-1365	A vulnerability, which was classified as critical, was found in GNU elfutils 0.192. This affects the function process_
CVE-2025-1371	A vulnerability has been found in GNU elfutils 0.192 and classified as problematic. This vulnerability affects the fun
CVE-2025-1372	A vulnerability was found in GNU elfutils 0.192. It has been declared as critical. Affected by this vulnerability is the
CVE-2025-1376	A vulnerability classified as problematic was found in GNU elfutils 0.192. This vulnerability affects the function elf
CVE-2025-1377	A vulnerability, which was classified as problematic, has been found in GNU elfutils 0.192. This issue affects the fun
CVE-2025-1390	The PAM module pam_cap.so of libcap configuration supports group names starting with “@”, during actual parsing
CVE-2025-1426	Heap buffer overflow in GPU in Google Chrome on Android prior to 133.0.6943.126 allowed a remote attacker to p
CVE-2025-1550	The Keras Model.load_model function permits arbitrary code execution, even with safe_mode=True, through a man
CVE-2025-1594	A vulnerability, which was classified as critical, was found in FFmpeg up to 7.1. This affects the function ff_aac_sea
CVE-2025-1632	A vulnerability was found in libarchive up to 3.7.7. It has been classified as problematic. This affects the function lis
CVE-2025-1767	This CVE only affects Kubernetes clusters that utilize the in-tree gitRepo volume to clone git repositories from other
CVE-2025-1795	During an address list folding when a separating comma ends up on a folded line and that line is to be unicode-enco
CVE-2025-1914	Out of bounds read in V8 in Google Chrome prior to 134.0.6998.35 allowed a remote attacker to perform out of bou
CVE-2025-1915	Improper Limitation of a Pathname to a Restricted Directory in DevTools in Google Chrome on Windows prior to 13
CVE-2025-1916	Use after free in Profiles in Google Chrome prior to 134.0.6998.35 allowed an attacker who convinced a user to insta
CVE-2025-1917	Inappropriate implementation in Browser UI in Google Chrome on Android prior to 134.0.6998.35 allowed a remote
CVE-2025-1918	Out of bounds read in PDFium in Google Chrome prior to 134.0.6998.35 allowed a remote attacker to potentially pe
CVE-2025-1919	Out of bounds read in Media in Google Chrome prior to 134.0.6998.35 allowed a remote attacker to potentially perfo
CVE-2025-1920	Type Confusion in V8 in Google Chrome prior to 134.0.6998.88 allowed a remote attacker to potentially exploit hea
CVE-2025-1921	Inappropriate implementation in Media Stream in Google Chrome prior to 134.0.6998.35 allowed a remote attacker
CVE-2025-1922	Inappropriate implementation in Selection in Google Chrome on Android prior to 134.0.6998.35 allowed a remote a
CVE-2025-1974	A security issue was discovered in Kubernetes where under certain conditions, an unauthenticated attacker with acce
CVE-2025-1923	Inappropriate implementation in Permission Prompts in Google Chrome prior to 134.0.6998.35 allowed an attacker v
CVE-2025-1979	Versions of the package ray before 2.43.0 are vulnerable to Insertion of Sensitive Information into Log File where th
CVE-2025-2135	Type Confusion in V8 in Google Chrome prior to 134.0.6998.88 allowed a remote attacker to potentially exploit hea
CVE-2025-2136	Use after free in Inspector in Google Chrome prior to 134.0.6998.88 allowed a remote attacker to potentially exploit
CVE-2025-2137	Out of bounds read in V8 in Google Chrome prior to 134.0.6998.88 allowed a remote attacker to perform out of bou
CVE-2025-21502	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2025-21587	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2025-21613	go-git is a highly extensible git implementation library written in pure Go. An argument injection vulnerability was c
CVE-2025-21614	go-git is a highly extensible git implementation library written in pure Go. A denial of service (DoS) vulnerability w
CVE-2025-21629	In the Linux kernel, the following vulnerability has been resolved: net: reenable NETIF_F_IPV6_CSUM offload for
CVE-2025-21631	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix waker_bfq UAF after bfq_split_b
CVE-2025-21632	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Ensure shadow stack is active before "ge
CVE-2025-21634	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: remove kernfs active break A war
CVE-2025-21635	In the Linux kernel, the following vulnerability has been resolved: rds: sysctl: rds_tcp_{rcv,snd}buf: avoid using cur
CVE-2025-21636	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: plpmtud_probe_interval: avoid using
CVE-2025-21637	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: udp_port: avoid using current->nspro
CVE-2025-21638	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: auth_enable: avoid using current->ns
CVE-2025-21639	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: rto_min/max: avoid using current->n

CVE-2025-21640	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: cookie_hmac_alg: avoid using current
CVE-2025-21642	In the Linux kernel, the following vulnerability has been resolved: mptcp: sysctl: sched: avoid using current->nsprox
CVE-2025-21643	In the Linux kernel, the following vulnerability has been resolved: netfs: Fix kernel async DIO Netfslib needs to be a
CVE-2025-21645	In the Linux kernel, the following vulnerability has been resolved: platform/x86/amd/pmc: Only disable IRQ1 wake
CVE-2025-21646	In the Linux kernel, the following vulnerability has been resolved: afs: Fix the maximum cell name length The kafs
CVE-2025-21647	In the Linux kernel, the following vulnerability has been resolved: sched: sch_cake: add bounds checks to host bulk
CVE-2025-21648	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: clamp maximum hashtable si
CVE-2025-21649	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when 1588 is sent on F
CVE-2025-21650	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fixed hclge_fetch_pf_reg accesses bar s
CVE-2025-21651	In the Linux kernel, the following vulnerability has been resolved: net: hns3: don't auto enable misc vector Currently
CVE-2025-21652	In the Linux kernel, the following vulnerability has been resolved: ipvlan: Fix use-after-free in ipvlan_get_iflink(). s
CVE-2025-21653	In the Linux kernel, the following vulnerability has been resolved: net_sched: cls_flow: validate TCA_FLOW_RSH
CVE-2025-21654	In the Linux kernel, the following vulnerability has been resolved: ovl: support encoding fid from inode with no alias
CVE-2025-21655	In the Linux kernel, the following vulnerability has been resolved: io_uring/eventfd: ensure io_eventfd_signal() defe
CVE-2025-21656	In the Linux kernel, the following vulnerability has been resolved: hwmon: (drivetemp) Fix driver producing garbaga
CVE-2025-21660	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix unexpectedly changed path in ksmbd_
CVE-2025-21659	In the Linux kernel, the following vulnerability has been resolved: netdev: prevent accessing NAPI instances from ar
CVE-2025-21658	In the Linux kernel, the following vulnerability has been resolved: btrfs: avoid NULL pointer dereference if no valid
CVE-2025-21662	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix variable not being completed when
CVE-2025-21663	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: dwmac-tegra: Read iommu stream i
CVE-2025-21664	In the Linux kernel, the following vulnerability has been resolved: dm thin: make get_first_thin use rcu-safe list first
CVE-2025-21665	In the Linux kernel, the following vulnerability has been resolved: filemap: avoid truncating 64-bit offset to 32 bits C
CVE-2025-21666	In the Linux kernel, the following vulnerability has been resolved: vsock: prevent null-ptr-deref in vsock_*[has_data
CVE-2025-21667	In the Linux kernel, the following vulnerability has been resolved: iomap: avoid avoid truncating 64-bit offset to 32
CVE-2025-21668	In the Linux kernel, the following vulnerability has been resolved: pmdomain: imx8mp-blk-ctrl: add missing loop br
CVE-2025-21669	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: discard packets if the transport chan
CVE-2025-21671	In the Linux kernel, the following vulnerability has been resolved: zram: fix potential UAF of zram table If zram_me
CVE-2025-21670	In the Linux kernel, the following vulnerability has been resolved: vsock/bpf: return early if transport is not assigned
CVE-2025-21672	In the Linux kernel, the following vulnerability has been resolved: afs: Fix merge preference rule failure condition sy
CVE-2025-21673	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix double free of TCP_Server_Info::
CVE-2025-21674	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix inversion dependency warning wh
CVE-2025-21678	In the Linux kernel, the following vulnerability has been resolved: gtp: Destroy device along with udp socket's netns
CVE-2025-21675	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Clear port select structure when fail to c
CVE-2025-21676	In the Linux kernel, the following vulnerability has been resolved: net: fec: handle page_pool_dev_alloc_pages error
CVE-2025-21680	In the Linux kernel, the following vulnerability has been resolved: pktgen: Avoid out-of-bounds access in get_imix_
CVE-2025-21681	In the Linux kernel, the following vulnerability has been resolved: openvswitch: fix lockup on tx to unregistering net
CVE-2025-21682	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: always recalculate features after XDP c
CVE-2025-21683	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix bpf_sk_select_reuseport() memory leak &
CVE-2025-21684	In the Linux kernel, the following vulnerability has been resolved: gpio: xilinx: Convert gpio_lock to raw spinlock in
CVE-2025-21687	In the Linux kernel, the following vulnerability has been resolved: vfio/platform: check the bounds of read/write sys

CVE-2025-21689	In the Linux kernel, the following vulnerability has been resolved: USB: serial: quatech2: fix null-ptr-deref in qt2_pr
CVE-2025-21690	In the Linux kernel, the following vulnerability has been resolved: scsi: storvsc: Ratelimit warning logs to prevent V
CVE-2025-21691	In the Linux kernel, the following vulnerability has been resolved: cachestat: fix page cache statistics permission che
CVE-2025-21693	In the Linux kernel, the following vulnerability has been resolved: mm: zswap: properly synchronize freeing resourc
CVE-2025-21692	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix ets qdisc OOB Indexing Haowei Y
CVE-2025-21694	In the Linux kernel, the following vulnerability has been resolved: fs/proc: fix softlockup in __read_vmcore (part 2)
CVE-2025-21696	In the Linux kernel, the following vulnerability has been resolved: mm: clear uffd-wp PTE/PMD state on mremap()
CVE-2025-21697	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Ensure job pointer is set to NULL after
CVE-2025-21700	In the Linux kernel, the following vulnerability has been resolved: net: sched: Disallow replacing of child qdisc from
CVE-2025-21699	In the Linux kernel, the following vulnerability has been resolved: gfs2: Truncate address space when flipping GFS2
CVE-2025-21701	In the Linux kernel, the following vulnerability has been resolved: net: avoid race between device unregistration and
CVE-2025-21702	In the Linux kernel, the following vulnerability has been resolved: pfifo_tail_enqueue: Drop new packet when sch->
CVE-2025-21704	In the Linux kernel, the following vulnerability has been resolved: usb: cdc-acm: Check control transfer buffer size b
CVE-2025-21703	In the Linux kernel, the following vulnerability has been resolved: netem: Update sch->q.qlen before qdisc_tree_red
CVE-2025-21707	In the Linux kernel, the following vulnerability has been resolved: mptcp: consolidate suboption status MPTCP main
CVE-2025-21709	In the Linux kernel, the following vulnerability has been resolved: kernel: be more careful about dup_mmap() failure
CVE-2025-21708	In the Linux kernel, the following vulnerability has been resolved: net: usb: rtl8150: enable basic endpoint checking
CVE-2025-21711	In the Linux kernel, the following vulnerability has been resolved: net/rose: prevent integer overflows in rose_setso
CVE-2025-21712	In the Linux kernel, the following vulnerability has been resolved: md/md-bitmap: Synchronize bitmap_get_stats() v
CVE-2025-21714	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix implicit ODP use after free Pre
CVE-2025-21718	In the Linux kernel, the following vulnerability has been resolved: net: rose: fix timer races against user threads Ros
CVE-2025-21715	In the Linux kernel, the following vulnerability has been resolved: net: davicom: fix UAF in dm9000_drv_remove d
CVE-2025-21719	In the Linux kernel, the following vulnerability has been resolved: ipmr: do not call mr_mfc_uses_dev() for unres en
CVE-2025-21721	In the Linux kernel, the following vulnerability has been resolved: nilfs2: handle errors that nilfs_prepare_chunk() m
CVE-2025-21722	In the Linux kernel, the following vulnerability has been resolved: nilfs2: do not force clear folio if buffer is referenc
CVE-2025-21726	In the Linux kernel, the following vulnerability has been resolved: padata: avoid UAF for reorder_work Although th
CVE-2025-21723	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix possible crash when setting up
CVE-2025-21728	In the Linux kernel, the following vulnerability has been resolved: bpf: Send signals asynchronously if !preemptible
CVE-2025-21727	In the Linux kernel, the following vulnerability has been resolved: padata: fix UAF in padata_reorder A bug was fou
CVE-2025-21729	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix race between cancel_hw_scan and
CVE-2025-2173	A vulnerability was found in libzvbi up to 0.2.43. It has been classified as problematic. Affected is the function vbi_
CVE-2025-21730	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid to init mgnt_entry list twice wh
CVE-2025-21732	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix a race for an ODP MR which l
CVE-2025-21731	In the Linux kernel, the following vulnerability has been resolved: nbd: don't allow reconnect after disconnect Follow
CVE-2025-21733	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Fix resetting of tracepoints If a ti
CVE-2025-21734	In the Linux kernel, the following vulnerability has been resolved: misc: fastrpc: Fix copy buffer page size For non-r
CVE-2025-21735	In the Linux kernel, the following vulnerability has been resolved: NFC: nci: Add bounds checking in nci_hci_create
CVE-2025-21738	In the Linux kernel, the following vulnerability has been resolved: ata: libata-sff: Ensure that we cannot write outside
CVE-2025-21736	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix possible int overflows in nilfs_fiemap()
CVE-2025-21739	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Fix use-after free in init error and

CVE-2025-2174	A vulnerability was found in libzvb up to 0.2.43. It has been declared as problematic. Affected by this vulnerability
CVE-2025-21744	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: fix NULL pointer dereference in
CVE-2025-21745	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: Fix class @block_class's subsystem n
CVE-2025-21748	In the Linux kernel, the following vulnerability has been resolved: ksmdb: fix integer overflows on 32 bit systems O
CVE-2025-21749	In the Linux kernel, the following vulnerability has been resolved: net: rose: lock the socket in rose_bind() syzbot re
CVE-2025-2175	A vulnerability was found in libzvb up to 0.2.43. It has been rated as problematic. Affected by this issue is the funct
CVE-2025-21750	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Check the return value of of_pro
CVE-2025-21751	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: HWS, change error flow on matcher dis
CVE-2025-21752	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't use btrfs_set_item_key_safe on RAID
CVE-2025-21753	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free when attempting to join an
CVE-2025-21756	In the Linux kernel, the following vulnerability has been resolved: vsock: Keep the binding until socket destruction
CVE-2025-21758	In the Linux kernel, the following vulnerability has been resolved: ipv6: mcast: add RCU protection to mld_newpack
CVE-2025-21759	In the Linux kernel, the following vulnerability has been resolved: ipv6: mcast: extend RCU protection in igmp6_ser
CVE-2025-2176	A vulnerability classified as critical has been found in libzvb up to 0.2.43. This affects the function vbi_capture_sim
CVE-2025-21760	In the Linux kernel, the following vulnerability has been resolved: ndisc: extend RCU protection in ndisc_send_skb()
CVE-2025-21761	In the Linux kernel, the following vulnerability has been resolved: openvswitch: use RCU protection in ovs_vport_c
CVE-2025-21762	In the Linux kernel, the following vulnerability has been resolved: arp: use RCU protection in arp_xmit() arp_xmit()
CVE-2025-21763	In the Linux kernel, the following vulnerability has been resolved: neighbour: use RCU protection in __neigh_notify
CVE-2025-21764	In the Linux kernel, the following vulnerability has been resolved: ndisc: use RCU protection in ndisc_alloc_skb() n
CVE-2025-21765	In the Linux kernel, the following vulnerability has been resolved: ipv6: use RCU protection in ip6_default_advmss
CVE-2025-21766	In the Linux kernel, the following vulnerability has been resolved: ipv4: use RCU protection in __ip_rt_update_pmtu
CVE-2025-21767	In the Linux kernel, the following vulnerability has been resolved: clocksource: Use migrate_disable() to avoid callin
CVE-2025-21768	In the Linux kernel, the following vulnerability has been resolved: net: ipv6: fix dst ref loops in rpl, seg6 and ioam6
CVE-2025-2177	A vulnerability classified as critical was found in libzvb up to 0.2.43. This vulnerability affects the function vbi_sea
CVE-2025-21772	In the Linux kernel, the following vulnerability has been resolved: partitions: mac: fix handling of bogus partition ta
CVE-2025-21776	In the Linux kernel, the following vulnerability has been resolved: USB: hub: Ignore non-compliant devices with too
CVE-2025-21779	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Reject Hyper-V's SEND_IPI hyperca
CVE-2025-21780	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: avoid buffer overflow attach in smu
CVE-2025-21782	In the Linux kernel, the following vulnerability has been resolved: oragefs: fix a oob in oragefs_debug_write I got
CVE-2025-21781	In the Linux kernel, the following vulnerability has been resolved: batman-adv: fix panic during interface removal R
CVE-2025-21785	In the Linux kernel, the following vulnerability has been resolved: arm64: cacheinfo: Avoid out-of-bounds write to c
CVE-2025-21786	In the Linux kernel, the following vulnerability has been resolved: workqueue: Put the pwq after detaching the rescu
CVE-2025-21787	In the Linux kernel, the following vulnerability has been resolved: team: better TEAM_OPTION_TYPE_STRING v
CVE-2025-21791	In the Linux kernel, the following vulnerability has been resolved: vrf: use RCU protection in l3mdev_l3_out() l3mdev
CVE-2025-21792	In the Linux kernel, the following vulnerability has been resolved: ax25: Fix refcount leak caused by setting SO_BI
CVE-2025-21795	In the Linux kernel, the following vulnerability has been resolved: NFSD: fix hang in nfsd4_shutdown_callback If n
CVE-2025-21796	In the Linux kernel, the following vulnerability has been resolved: nfsd: clear acl_access/acl_default after releasing
CVE-2025-21799	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: ti: am65-cpsw: fix freeing IRQ in an
CVE-2025-21801	In the Linux kernel, the following vulnerability has been resolved: net: ravb: Fix missing rtnl lock in suspend/resume
CVE-2025-21802	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix oops when unload drivers parallelin

CVE-2025-21804	In the Linux kernel, the following vulnerability has been resolved: PCI: rcar-ep: Fix incorrect variable used when cal
CVE-2025-21811	In the Linux kernel, the following vulnerability has been resolved: nilfs2: protect access to buffers with no active ref
CVE-2025-21806	In the Linux kernel, the following vulnerability has been resolved: net: let net.core.dev_weight always be non-zero T
CVE-2025-21812	In the Linux kernel, the following vulnerability has been resolved: ax25: rcu protect dev->ax25_ptr syzbot found a lo
CVE-2025-21814	In the Linux kernel, the following vulnerability has been resolved: ptp: Ensure info->enable callback is always set T
CVE-2025-21816	In the Linux kernel, the following vulnerability has been resolved: hrtimers: Force migrate away hrtimers queued aft
CVE-2025-21817	In the Linux kernel, the following vulnerability has been resolved: block: mark GFP_NOIO around sysfs ->store() sy
CVE-2025-21819	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/amd/display: Use HW lock mgr for l
CVE-2025-21821	In the Linux kernel, the following vulnerability has been resolved: fbdev: omap: use threaded IRQ for LCD DMA W
CVE-2025-21820	In the Linux kernel, the following vulnerability has been resolved: tty: xilinx_uartps: split sysrq handling lockdep de
CVE-2025-21823	In the Linux kernel, the following vulnerability has been resolved: batman-adv: Drop unmanaged ELP metric worke
CVE-2025-21825	In the Linux kernel, the following vulnerability has been resolved: bpf: Cancel the running bpf_timer through kwork
CVE-2025-21826	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: reject mismatching sum of fie
CVE-2025-21829	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix the warning "__rxe_cleanup+0x1
CVE-2025-21828	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: don't flush non-uploaded STAs l
CVE-2025-21830	In the Linux kernel, the following vulnerability has been resolved: landlock: Handle weird files A corrupted filesystem
CVE-2025-21831	In the Linux kernel, the following vulnerability has been resolved: PCI: Avoid putting some root ports into D3 on TU
CVE-2025-21832	In the Linux kernel, the following vulnerability has been resolved: block: don't revert iter for -EIOCBQUEUED blk
CVE-2025-21833	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Avoid use of NULL after WARN_O
CVE-2025-21835	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_midi: fix MIDI Streaming descript
CVE-2025-21838	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: core: flush gadget workqueue after d
CVE-2025-21844	In the Linux kernel, the following vulnerability has been resolved: smb: client: Add check for next_buffer in receive
CVE-2025-21839	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Load DR6 with guest value only befo
CVE-2025-21846	In the Linux kernel, the following vulnerability has been resolved: acct: perform last write from workqueue In [1] it
CVE-2025-21847	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: stream-ipc: Check for cstream nullit
CVE-2025-21848	In the Linux kernel, the following vulnerability has been resolved: nfp: bpf: Add check for nfp_app_ctrl_msg_alloc
CVE-2025-21853	In the Linux kernel, the following vulnerability has been resolved: bpf: avoid holding freeze_mutex during mmap op
CVE-2025-21855	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Don't reference skb after sending to VIC
CVE-2025-21857	In the Linux kernel, the following vulnerability has been resolved: net/sched: cls_api: fix error handling causing NU
CVE-2025-21858	In the Linux kernel, the following vulnerability has been resolved: geneve: Fix use-after-free in geneve_find_dev().
CVE-2025-21859	In the Linux kernel, the following vulnerability has been resolved: USB: gadget: f_midi: f_midi_complete to call que
CVE-2025-21861	In the Linux kernel, the following vulnerability has been resolved: mm/migrate_device: don't add folio to be freed to
CVE-2025-21862	In the Linux kernel, the following vulnerability has been resolved: drop_monitor: fix incorrect initialization order Sy
CVE-2025-21863	In the Linux kernel, the following vulnerability has been resolved: io_uring: prevent opcode speculation sqe->opcode
CVE-2025-21865	In the Linux kernel, the following vulnerability has been resolved: gtp: Suppress list corruption splat in gtp_net_exit
CVE-2025-21866	In the Linux kernel, the following vulnerability has been resolved: powerpc/code-patching: Fix KASAN hit by not fl
CVE-2025-21870	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-topology: Harden loops for loo
CVE-2025-21871	In the Linux kernel, the following vulnerability has been resolved: tee: optee: Fix supplicant wait loop OP-TEE supp
CVE-2025-21872	In the Linux kernel, the following vulnerability has been resolved: efi: Don't map the entire mokvar table to determin
CVE-2025-21873	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: bsg: Fix crash when arpmmb comm

CVE-2025-21875	In the Linux kernel, the following vulnerability has been resolved: mptcp: always handle address removal under msk
CVE-2025-21877	In the Linux kernel, the following vulnerability has been resolved: usbnet: gl620a: fix endpoint checking in genelink
CVE-2025-21878	In the Linux kernel, the following vulnerability has been resolved: i2c: npcm: disable interrupt enable bit before dev
CVE-2025-21880	In the Linux kernel, the following vulnerability has been resolved: drm/xe/userptr: fix EFAULT handling Currently
CVE-2025-21881	In the Linux kernel, the following vulnerability has been resolved: uprobes: Reject the shared zeropage in uprobe_w
CVE-2025-21883	In the Linux kernel, the following vulnerability has been resolved: ice: Fix deinitializing VF in error path If ice_ena_
CVE-2025-21884	In the Linux kernel, the following vulnerability has been resolved: net: better track kernel sockets lifetime While ker
CVE-2025-21885	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix the page details for the srq c
CVE-2025-21886	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix implicit ODP hang on parent d
CVE-2025-21887	In the Linux kernel, the following vulnerability has been resolved: ovl: fix UAF in ovl_dentry_update_reval by mov
CVE-2025-21888	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix a WARN during dereg_mr for
CVE-2025-21889	In the Linux kernel, the following vulnerability has been resolved: perf/core: Add RCU read lock protection to perf_
CVE-2025-21890	In the Linux kernel, the following vulnerability has been resolved: idpf: fix checksums set in idpf_rx_rsc() idpf_rx_r
CVE-2025-21891	In the Linux kernel, the following vulnerability has been resolved: ipvlan: ensure network headers are in skb linear p
CVE-2025-21894	In the Linux kernel, the following vulnerability has been resolved: net: enetc: VFs do not support HWTSTAMP_TX
CVE-2025-21892	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix the recovery flow of the UMR
CVE-2025-21895	In the Linux kernel, the following vulnerability has been resolved: perf/core: Order the PMU list to fix warning abou
CVE-2025-21898	In the Linux kernel, the following vulnerability has been resolved: ftrace: Avoid potential division by zero in functio
CVE-2025-21899	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix bad hist from corrupting named_trigge
CVE-2025-21903	In the Linux kernel, the following vulnerability has been resolved: mctp i3c: handle NULL header address daddr can
CVE-2025-21904	In the Linux kernel, the following vulnerability has been resolved: caif_virtio: fix wrong pointer check in cfv_probec
CVE-2025-21907	In the Linux kernel, the following vulnerability has been resolved: mm: memory-failure: update ttu flag inside unma
CVE-2025-21905	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: limit printed string from FW file Th
CVE-2025-21908	In the Linux kernel, the following vulnerability has been resolved: NFS: fix nfs_release_folio() to not deadlock via k
CVE-2025-21909	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: reject cooked mode if it is set along
CVE-2025-21910	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: regulatory: improve invalid hints
CVE-2025-21911	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: avoid deadlock on fence release
CVE-2025-21912	In the Linux kernel, the following vulnerability has been resolved: gpio: rcar: Use raw_spinlock to protect register ac
CVE-2025-21913	In the Linux kernel, the following vulnerability has been resolved: x86/amd_nb: Use rdmsr_safe() in amd_get_mmco
CVE-2025-21914	In the Linux kernel, the following vulnerability has been resolved: slimbus: messaging: Free transaction ID in delaye
CVE-2025-21915	In the Linux kernel, the following vulnerability has been resolved: cdx: Fix possible UAF error in driver_override_sl
CVE-2025-21916	In the Linux kernel, the following vulnerability has been resolved: usb: atm: cxacr: fix a flaw in existing endpoint c
CVE-2025-21917	In the Linux kernel, the following vulnerability has been resolved: usb: renesas_usbhs: Flush the notify_hotplug_wo
CVE-2025-21918	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Fix NULL pointer access Resour
CVE-2025-21919	In the Linux kernel, the following vulnerability has been resolved: sched/fair: Fix potential memory corruption in ch
CVE-2025-21920	In the Linux kernel, the following vulnerability has been resolved: vlan: enforce underlying device type Currently, V
CVE-2025-21922	In the Linux kernel, the following vulnerability has been resolved: ppp: Fix KMSAN uninit-value warning with bpf f
CVE-2025-21924	In the Linux kernel, the following vulnerability has been resolved: net: hns3: make sure ptp clock is unregister and fr
CVE-2025-21925	In the Linux kernel, the following vulnerability has been resolved: llc: do not use skb_get() before dev_queue_xmit()
CVE-2025-21926	In the Linux kernel, the following vulnerability has been resolved: net: gso: fix ownership in __udp_gso_segment In

CVE-2025-21927	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix potential memory corruption in nvme
CVE-2025-21928	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: Fix use-after-free issue in ishtp
CVE-2025-21929	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: Fix use-after-free issue in hid_
CVE-2025-21931	In the Linux kernel, the following vulnerability has been resolved: hwpoison, memory_hotplug: lock folio before un
CVE-2025-21930	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: don't try to talk to a dead firm
CVE-2025-21934	In the Linux kernel, the following vulnerability has been resolved: rapidio: fix an API misues when rio_add_net() fa
CVE-2025-21936	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Add check for mgmt_alloc_skb() in m
CVE-2025-21935	In the Linux kernel, the following vulnerability has been resolved: rapidio: add check for rio_add_net() in rio_scan_
CVE-2025-21937	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Add check for mgmt_alloc_skb() in m
CVE-2025-21938	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix 'scheduling while atomic' in mptcp_pm
CVE-2025-21941	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null check for pipe_ctx->pl
CVE-2025-21944	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix bug on trap in smb2_lock If lock coun
CVE-2025-21943	In the Linux kernel, the following vulnerability has been resolved: gpio: aggregator: protect driver attr handlers again
CVE-2025-21945	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in smb2_lock If smb_lo
CVE-2025-21946	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix out-of-bounds in parse_sec_desc() If d
CVE-2025-21947	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix type confusion via race condition whe
CVE-2025-21948	In the Linux kernel, the following vulnerability has been resolved: HID: appleir: Fix potential NULL dereference at
CVE-2025-21949	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Set hugetlb mmap base address align
CVE-2025-21951	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: host: pci_generic: Use pci_try_reset_fu
CVE-2025-21950	In the Linux kernel, the following vulnerability has been resolved: drivers: virt: acrn: hsm: Use kzalloc to avoid info
CVE-2025-21955	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent connection release during oplock
CVE-2025-21956	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Assign normalized_pix_clk wh
CVE-2025-21957	In the Linux kernel, the following vulnerability has been resolved: scsi: qla1280: Fix kernel oops when debug level >
CVE-2025-21960	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: do not update checksum in bnxt_xdp_b
CVE-2025-21961	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: fix truesize for mb-xdp-pass case When
CVE-2025-21959	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_conncount: Fully initialize struct nf_c
CVE-2025-21962	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix integer overflow while processing closet
CVE-2025-21963	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix integer overflow while processing acdir
CVE-2025-21966	In the Linux kernel, the following vulnerability has been resolved: dm-flakey: Fix memory corruption in optional co
CVE-2025-21964	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix integer overflow while processing acreg
CVE-2025-21967	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in ksmbd_free_work_st
CVE-2025-21968	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix slab-use-after-free on hdc
CVE-2025-21969	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix slab-use-after-free Read i
CVE-2025-21970	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Bridge, fix the crash caused by LAG sta
CVE-2025-21971	In the Linux kernel, the following vulnerability has been resolved: net_sched: Prevent creation of classes with TC_H
CVE-2025-21972	In the Linux kernel, the following vulnerability has been resolved: net: mctp: unshare packets when reassembling E
CVE-2025-21975	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: handle errors in mlx5_chains_create_ta
CVE-2025-21976	In the Linux kernel, the following vulnerability has been resolved: fbdev: hyperv_fb: Allow graceful removal of fram
CVE-2025-21978	In the Linux kernel, the following vulnerability has been resolved: drm/hyperv: Fix address space leak when Hyper-
CVE-2025-21977	In the Linux kernel, the following vulnerability has been resolved: fbdev: hyperv_fb: Fix hang in kdump kernel whe

CVE-2025-21979	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: cancel wiphy_work before freeing
CVE-2025-21980	In the Linux kernel, the following vulnerability has been resolved: sched: address a potential NULL pointer dereference
CVE-2025-21981	In the Linux kernel, the following vulnerability has been resolved: ice: fix memory leak in aRFS after reset Fix aRFS
CVE-2025-21982	In the Linux kernel, the following vulnerability has been resolved: pinctrl: nuvoton: npcm8xx: Add NULL check in n
CVE-2025-21984	In the Linux kernel, the following vulnerability has been resolved: mm: fix kernel BUG when userfaultfd_move enc
CVE-2025-21985	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix out-of-bound accesses [WH
CVE-2025-21986	In the Linux kernel, the following vulnerability has been resolved: net: switchdev: Convert blocking notification cha
CVE-2025-21991	In the Linux kernel, the following vulnerability has been resolved: x86/microcode/AMD: Fix out-of-bounds on syste
CVE-2025-21992	In the Linux kernel, the following vulnerability has been resolved: HID: ignore non-functional sensor in HP 5MP Ca
CVE-2025-21993	In the Linux kernel, the following vulnerability has been resolved: iscsi_ibft: Fix UBSAN shift-out-of-bounds warni
CVE-2025-21994	In the Linux kernel, the following vulnerability has been resolved: ksmdb: fix incorrect validation for num_acces fiel
CVE-2025-21995	In the Linux kernel, the following vulnerability has been resolved: drm/sched: Fix fence reference count leak The las
CVE-2025-21996	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix uninitialized size issue in radeon_
CVE-2025-21997	In the Linux kernel, the following vulnerability has been resolved: xsk: fix an integer overflow in xp_create_and_ass
CVE-2025-21999	In the Linux kernel, the following vulnerability has been resolved: proc: fix UAF in proc_get_inode() Fix race betwe
CVE-2025-22001	In the Linux kernel, the following vulnerability has been resolved: accel/qaic: Fix integer overflow in qaic_validate_
CVE-2025-22002	In the Linux kernel, the following vulnerability has been resolved: netfs: Call `invalidate_cache` only if implemente
CVE-2025-22003	In the Linux kernel, the following vulnerability has been resolved: can: ucan: fix out of bound read in strscpy() sourc
CVE-2025-22004	In the Linux kernel, the following vulnerability has been resolved: net: atm: fix use after free in lec_send() The ->ser
CVE-2025-22005	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix memleak of nhc_pcpu_rth_output in fib
CVE-2025-22007	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix error code in chan_alloc_skb_cb()
CVE-2025-22008	In the Linux kernel, the following vulnerability has been resolved: regulator: check that dummy regulator has been p
CVE-2025-22009	In the Linux kernel, the following vulnerability has been resolved: regulator: dummy: force synchronous probing So
CVE-2025-22010	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix soft lockup during bt pages loop
CVE-2025-22011	In the Linux kernel, the following vulnerability has been resolved: ARM: dts: bcm2711: Fix xHCI power-domain Du
CVE-2025-22013	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Unconditionally save+flush host F
CVE-2025-22014	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: pdr: Fix the potential deadlock When
CVE-2025-22015	In the Linux kernel, the following vulnerability has been resolved: mm/migrate: fix shmem xarray update during mig
CVE-2025-22016	In the Linux kernel, the following vulnerability has been resolved: dpll: fix xa_alloc_cyclic() error handling In ca
CVE-2025-22017	In the Linux kernel, the following vulnerability has been resolved: devlink: fix xa_alloc_cyclic() error handling In ca
CVE-2025-22018	In the Linux kernel, the following vulnerability has been resolved: atm: Fix NULL pointer dereference When MPOA
CVE-2025-22019	In the Linux kernel, the following vulnerability has been resolved: bcachefs: bch2_ioctl_subvolume_destroy() fixes l
CVE-2025-22020	In the Linux kernel, the following vulnerability has been resolved: memstick: rtsx_usb_ms: Fix slab-use-after-free in
CVE-2025-22021	In the Linux kernel, the following vulnerability has been resolved: netfilter: socket: Lookup orig tuple for IPv6 SNA
CVE-2025-22022	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Apply the link chain quirk on NEC iso
CVE-2025-22025	In the Linux kernel, the following vulnerability has been resolved: nfsd: put dl_std if fail to queue dl_recall Before c
CVE-2025-22026	In the Linux kernel, the following vulnerability has been resolved: nfsd: don't ignore the return code of svc_proc_reg
CVE-2025-22027	In the Linux kernel, the following vulnerability has been resolved: media: streamzap: fix race between device discon
CVE-2025-22028	In the Linux kernel, the following vulnerability has been resolved: media: vimc: skip .s_stream() for stopped entities
CVE-2025-22029	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.

CVE-2025-22033	In the Linux kernel, the following vulnerability has been resolved: arm64: Don't call NULL in do_compat_alignmen
CVE-2025-22035	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix use-after-free in print_graph_function
CVE-2025-22036	In the Linux kernel, the following vulnerability has been resolved: exfat: fix random stack corruption after get_block
CVE-2025-22037	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix null pointer dereference in alloc_prea
CVE-2025-22038	In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate zero num_subauth before sub_auth
CVE-2025-22039	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix overflow in dacloffset bounds check T
CVE-2025-22040	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix session use-after-free in multichannel
CVE-2025-22042	In the Linux kernel, the following vulnerability has been resolved: ksmbd: add bounds check for create lease context
CVE-2025-22041	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in ksmbd_sessions_dere
CVE-2025-22043	In the Linux kernel, the following vulnerability has been resolved: ksmbd: add bounds check for durable handle cont
CVE-2025-22044	In the Linux kernel, the following vulnerability has been resolved: acpi: nfit: fix narrowing conversion in acpi_nfit_c
CVE-2025-22045	In the Linux kernel, the following vulnerability has been resolved: x86/mm: Fix flush_tlb_range() when used for zap
CVE-2025-22050	In the Linux kernel, the following vulnerability has been resolved: usbnet:fix NPE during rx_complete Missing usbn
CVE-2025-22053	In the Linux kernel, the following vulnerability has been resolved: net: ibmveth: make veth_pool_store stop hanging
CVE-2025-22055	In the Linux kernel, the following vulnerability has been resolved: net: fix geneve_opt length integer overflow struct
CVE-2025-22054	In the Linux kernel, the following vulnerability has been resolved: arcnet: Add NULL check in com20020pci_prober
CVE-2025-22056	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_tunnel: fix geneve_opt type confusio
CVE-2025-22057	In the Linux kernel, the following vulnerability has been resolved: net: decrease cached dst counters in dst_release U
CVE-2025-22058	In the Linux kernel, the following vulnerability has been resolved: udp: Fix memory accounting leak. Matt Dowling
CVE-2025-22060	In the Linux kernel, the following vulnerability has been resolved: net: mvpp2: Prevent parser TCAM memory corru
CVE-2025-22062	In the Linux kernel, the following vulnerability has been resolved: sctp: add mutual exclusion in proc_sctp_do_udp
CVE-2025-22063	In the Linux kernel, the following vulnerability has been resolved: netlabel: Fix NULL pointer exception caused by
CVE-2025-22064	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: don't unregister hook when ta
CVE-2025-22065	In the Linux kernel, the following vulnerability has been resolved: idpf: fix adapter NULL pointer dereference on rel
CVE-2025-22066	In the Linux kernel, the following vulnerability has been resolved: ASoC: imx-card: Add NULL check in imx_card
CVE-2025-22068	In the Linux kernel, the following vulnerability has been resolved: ublk: make sure ubq->canceling is set when queu
CVE-2025-22070	In the Linux kernel, the following vulnerability has been resolved: fs/9p: fix NULL pointer dereference on mkdir WL
CVE-2025-22071	In the Linux kernel, the following vulnerability has been resolved: spufs: fix a leak in spufs_create_context() Leak fi
CVE-2025-22072	In the Linux kernel, the following vulnerability has been resolved: spufs: fix gang directory lifetimes prior to "[POW
CVE-2025-22073	In the Linux kernel, the following vulnerability has been resolved: spufs: fix a leak on spufs_new_file() failure It's ca
CVE-2025-22075	In the Linux kernel, the following vulnerability has been resolved: rtnetlink: Allocate vfinfo size for VF GUIDs whe
CVE-2025-22077	In the Linux kernel, the following vulnerability has been resolved: Revert "smb: client: fix TCP timers deadlock after
CVE-2025-22079	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate l_tree_depth to avoid out-of-boun
CVE-2025-22080	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Prevent integer overflow in hdr_first_def
CVE-2025-22083	In the Linux kernel, the following vulnerability has been resolved: vhost-scsi: Fix handling of multiple calls to vhost
CVE-2025-22081	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fix a couple integer overflows on 32bit s
CVE-2025-22086	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix mlx5_poll_one() cur_qp update
CVE-2025-22090	In the Linux kernel, the following vulnerability has been resolved: x86/mm/pat: Fix VM_PAT handling when fork()
CVE-2025-22089	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Don't expose hw_counters outside c
CVE-2025-22088	In the Linux kernel, the following vulnerability has been resolved: RDMA/erdma: Prevent use-after-free in erdma_a

CVE-2025-22095	In the Linux kernel, the following vulnerability has been resolved: PCI: brcmstb: Fix error path after a call to regulat
CVE-2025-22097	In the Linux kernel, the following vulnerability has been resolved: drm/vkms: Fix use after free and double free on in
CVE-2025-22101	In the Linux kernel, the following vulnerability has been resolved: net: libwx: fix Tx L4 checksum The hardware onl
CVE-2025-22103	In the Linux kernel, the following vulnerability has been resolved: net: fix NULL pointer dereference in l3mdev_l3_
CVE-2025-22102	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btmxpuart: Fix kernel panic during FW
CVE-2025-22104	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Use kernel helpers for hex dumps Previo
CVE-2025-22105	In the Linux kernel, the following vulnerability has been resolved: bonding: check xdp prog when set bond mode Fo
CVE-2025-22106	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: unregister xdp rxq info in the reset path
CVE-2025-22107	In the Linux kernel, the following vulnerability has been resolved: net: dsa: sja1105: fix kasan out-of-bounds warnin
CVE-2025-22108	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Mask the bd_cnt field in the TX BD pro
CVE-2025-22109	In the Linux kernel, the following vulnerability has been resolved: ax25: Remove broken autobind Binding AX25 so
CVE-2025-22111	In the Linux kernel, the following vulnerability has been resolved: net: Remove RTNL dance for SIOCBRADDIF an
CVE-2025-22113	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid journaling sb update on error if journa
CVE-2025-22115	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix block group refcount race in btrfs_creat
CVE-2025-22116	In the Linux kernel, the following vulnerability has been resolved: idpf: check error for register_netdev() on init Cur
CVE-2025-22120	In the Linux kernel, the following vulnerability has been resolved: ext4: goto right label 'out_mmap_sem' in ext4_se
CVE-2025-22121	In the Linux kernel, the following vulnerability has been resolved: ext4: fix out-of-bound read in ext4_xattr_inode_d
CVE-2025-22124	In the Linux kernel, the following vulnerability has been resolved: md/md-bitmap: fix wrong bitmap_limit for cluster
CVE-2025-22125	In the Linux kernel, the following vulnerability has been resolved: md/raid1,raid10: don't ignore IO flags If blk-wbt
CVE-2025-22127	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix potential deadlock in prepare_compress
CVE-2025-22126	In the Linux kernel, the following vulnerability has been resolved: md: fix mddev uaf while iterating all_mddevs list
CVE-2025-22128	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Clear affinity hint before calling ath
CVE-2025-22227	In some specific scenarios with chained redirects, Reactor Netty HTTP client leaks credentials. In order for this to ha
CVE-2025-22134	When switching to other buffers using the :all command and visual mode still being active, this may cause a heap-bu
CVE-2025-22150	Undici is an HTTP/1.1 client. Starting in version 4.5.0 and prior to versions 5.28.5, 6.21.1, and 7.2.3, undici uses `M
CVE-2025-22228	BCryptPasswordEncoder.matches(CharSequence,String) will incorrectly return true for passwords larger than 72 cha
CVE-2025-22233	CVE-2024-38820 ensured Locale-independent, lowercase conversion for both the configured disallowedFields patter
CVE-2025-22235	EndpointRequest.to() creates a matcher for null/** if the actuator endpoint, for which the EndpointRequest has been
CVE-2025-22868	An attacker can pass a malicious malformed token which causes unexpected memory to be consumed during parsing
CVE-2025-22869	SSH servers which implement file transfer protocols are vulnerable to a denial of service attack from clients which c
CVE-2025-22866	Due to the usage of a variable time instruction in the assembly implementation of an internal function, a small numb
CVE-2025-22870	Matching of hosts against proxy patterns can improperly treat an IPv6 zone ID as a hostname component. For exampl
CVE-2025-22872	The tokenizer incorrectly interprets tags with unquoted attribute values that end with a solidus character (/) as self-cl
CVE-2025-22871	The net/http package improperly accepts a bare LF as a line terminator in chunked data chunk-size lines. This can pe
CVE-2025-22919	A reachable assertion in FFmpeg git-master commit N-113007-g8d24a28d06 allows attackers to cause a Denial of S
CVE-2025-22874	Calling Verify with a VerifyOptions.KeyUsages that contains ExtKeyUsageAny unintentionally disabledpolicy valid
CVE-2025-22921	FFmpeg git-master,N-113007-g8d24a28d06 was discovered to contain a segmentation violation via the component /
CVE-2025-23048	In some mod_ssl configurations on Apache HTTP Server 2.4.35 through to 2.4.63, an access control bypass by truste
CVE-2025-23083	With the aid of the diagnostics_channel utility, an event can be hooked into whenever a worker thread is created. Th
CVE-2025-23085	A memory leak could occur when a remote peer abruptly closes the socket without sending a GOAWAY notification

CVE-2025-2312	A flaw was found in cifs-utils. When trying to obtain Kerberos credentials, the cifs.upcall program from the cifs-utils
CVE-2025-23129	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: Clear affinity hint before calling ath
CVE-2025-23130	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid panic once fallocation fails for p
CVE-2025-23131	In the Linux kernel, the following vulnerability has been resolved: dlm: prevent NPD when writing a positive value
CVE-2025-23132	In the Linux kernel, the following vulnerability has been resolved: f2fs: quota: fix to avoid warning in dquot_writeba
CVE-2025-23133	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: update channel list in reg notifier ins
CVE-2025-23135	In the Linux kernel, the following vulnerability has been resolved: RISC-V: KVM: Teardown riscv specific bits after
CVE-2025-23136	In the Linux kernel, the following vulnerability has been resolved: thermal: int340x: Add NULL check for adev Not
CVE-2025-23138	In the Linux kernel, the following vulnerability has been resolved: watch_queue: fix pipe accounting mismatch Curr
CVE-2025-23140	In the Linux kernel, the following vulnerability has been resolved: misc: pci_endpoint_test: Avoid issue of interrupts
CVE-2025-23141	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Acquire SRCU in KVM_GET_MP_S
CVE-2025-23142	In the Linux kernel, the following vulnerability has been resolved: sctp: detect and prevent references to a freed trans
CVE-2025-23143	In the Linux kernel, the following vulnerability has been resolved: net: Fix null-ptr-deref by sock_lock_init_class_ar
CVE-2025-23145	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix NULL pointer in can_accept_new_sub
CVE-2025-23144	In the Linux kernel, the following vulnerability has been resolved: backlight: led_bl: Hold led_access lock when call
CVE-2025-23146	In the Linux kernel, the following vulnerability has been resolved: mfd: ene-kb3930: Fix a potential NULL pointer d
CVE-2025-23148	In the Linux kernel, the following vulnerability has been resolved: soc: samsung: exynos-chipid: Add NULL pointer
CVE-2025-23147	In the Linux kernel, the following vulnerability has been resolved: i3c: Add NULL pointer check in i3c_master_que
CVE-2025-23150	In the Linux kernel, the following vulnerability has been resolved: ext4: fix off-by-one error in do_split Syzkaller de
CVE-2025-23149	In the Linux kernel, the following vulnerability has been resolved: tpm: do not start chip while suspended Checking
CVE-2025-23151	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: host: Fix race between unprepare and q
CVE-2025-23155	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: Fix accessing freed irq affinity_hin
CVE-2025-23156	In the Linux kernel, the following vulnerability has been resolved: media: venus: hfi_parser: refactor hfi packet pars
CVE-2025-23157	In the Linux kernel, the following vulnerability has been resolved: media: venus: hfi_parser: add check to avoid out
CVE-2025-23158	In the Linux kernel, the following vulnerability has been resolved: media: venus: hfi: add check to handle incorrect c
CVE-2025-23159	In the Linux kernel, the following vulnerability has been resolved: media: venus: hfi: add a check to handle OOB in
CVE-2025-23160	In the Linux kernel, the following vulnerability has been resolved: media: mediatek: vcodec: Fix a resource leak rela
CVE-2025-23161	In the Linux kernel, the following vulnerability has been resolved: PCI: vmd: Make vmd_dev::cfg_lock a raw_spinlo
CVE-2025-23162	In the Linux kernel, the following vulnerability has been resolved: drm/xe/vf: Don't try to trigger a full GT reset if V
CVE-2025-23165	In Node.js, the `ReadFileUtf8` internal binding leaks memory due to a corrupted pointer in `uv_fs_s.file`: a UTF-16
CVE-2025-23163	In the Linux kernel, the following vulnerability has been resolved: net: vlan: don't propagate flags on open With the
CVE-2025-23166	The C++ method SignTraits::DeriveBits() may incorrectly call ThrowException() based on user-supplied inputs whe
CVE-2025-23167	A flaw in Node.js 20's HTTP parser allows improper termination of HTTP/1 headers using `r\n\rX` instead of the re
CVE-2025-23184	A potential denial of service vulnerability is present in versions of Apache CXF before 3.5.10, 3.6.5 and 4.0.6. In so
CVE-2025-24014	Vim is an open source, command line text editor. A segmentation fault was found in Vim before 9.1.1043. In silent l
CVE-2025-24023	Flask-AppBuilder is an application development framework. Prior to 4.5.3, Flask-AppBuilder allows unauthenticated
CVE-2025-24357	vLLM is a library for LLM inference and serving. vllm/model_executor/weight_utils.py implements hf_model_weig
CVE-2025-24358	gorilla/csrif provides Cross Site Request Forgery (CSRF) prevention middleware for Go web applications & services
CVE-2025-24513	A security issue was discovered in ingress-nginx https://github.com/kubernetes/ingress-nginx where attacker-provide
CVE-2025-24514	A security issue was discovered in ingress-nginx https://github.com/kubernetes/ingress-nginx where the `auth-url` In

CVE-2025-2476	Use after free in Lens in Google Chrome prior to 134.0.6998.117 allowed a remote attacker to potentially exploit heap overflow
CVE-2025-24789	Snowflake JDBC provides a JDBC type 4 driver that supports core functionality, allowing Java program to connect to Snowflake
CVE-2025-24790	Snowflake JDBC provides a JDBC type 4 driver that supports core functionality, allowing Java program to connect to Snowflake
CVE-2025-24793	The Snowflake Connector for Python provides an interface for developing Python applications that can connect to Snowflake
CVE-2025-24794	The Snowflake Connector for Python provides an interface for developing Python applications that can connect to Snowflake
CVE-2025-24795	The Snowflake Connector for Python provides an interface for developing Python applications that can connect to Snowflake
CVE-2025-24855	numbers.c in libxslt before 1.1.43 has a use-after-free because, in nested XPath evaluations, an XPath context node can be freed
CVE-2025-24813	Path Equivalence: 'file.Name' (Internal Dot) leading to Remote Code Execution and/or Information disclosure and/or Denial of Service
CVE-2025-24970	Netty, an asynchronous, event-driven network application framework, has a vulnerability starting in version 4.1.91.Final
CVE-2025-24928	libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a stack-based buffer overflow in xmlSnprintfElements in valid.c
CVE-2025-25184	Rack provides an interface for developing web applications in Ruby. Prior to versions 2.2.11, 3.0.12, and 3.1.10, Rack::Session::Cookie
CVE-2025-25183	vLLM is a high-throughput and memory-efficient inference and serving engine for LLMs. Maliciously constructed requests can cause
CVE-2025-25186	Net::IMAP implements Internet Message Access Protocol (IMAP) client functionality in Ruby. Starting in version 0.9.2, Net::IMAP
CVE-2025-25473	FFmpeg git master before commit c08d30 was discovered to contain a NULL pointer dereference via the component avcodec
CVE-2025-25193	Netty, an asynchronous, event-driven network application framework, has a vulnerability in versions up to and including 4.1.91.Final
CVE-2025-25724	list_item_verbose in tar/util.c in libarchive through 3.7.7 does not check an strftime return value, which can lead to a denial of service
CVE-2025-2588	A vulnerability has been found in Hercules Augeas 1.14.1 and classified as problematic. This vulnerability affects the Augeas
CVE-2025-26465	A vulnerability was found in OpenSSH when the VerifyHostKeyDNS option is enabled. A machine-in-the-middle attack can be performed
CVE-2025-26519	musl libc 0.9.13 through 1.2.5 before 1.2.6 has an out-of-bounds write vulnerability when an attacker can trigger iconv
CVE-2025-26597	A buffer overflow flaw was found in X.Org and Xwayland. If XkbChangeTypesOfKey() is called with a 0 group, it can cause a denial of service
CVE-2025-26603	Vim is a greatly improved version of the good old UNIX editor Vi. Vim allows to redirect screen messages using the :redir command
CVE-2025-26699	An issue was discovered in Django 5.1 before 5.1.7, 5.0 before 5.0.13, and 4.2 before 4.2.20. The django.utils.text.truncate_words
CVE-2025-27111	Rack is a modular Ruby web server interface. The Rack::Sendfile middleware logs unsanitised header values from the application
CVE-2025-27018	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') vulnerability in Apache ActiveMQ
CVE-2025-27113	libxml2 before 2.12.10 and 2.13.x before 2.13.6 has a NULL pointer dereference in xmlPatMatch in pattern.c.
CVE-2025-27144	Go JOSE provides an implementation of the Javascript Object Signing and Encryption set of standards in Go, including JWS, JWE, and JWK
CVE-2025-27152	axios is a promise based HTTP client for the browser and node.js. The issue occurs when passing absolute URLs rather than relative
CVE-2025-27219	In the CGI gem before 0.4.2 for Ruby, the CGI::Cookie.parse method in the CGI library contains a potential Denial of Service
CVE-2025-27220	In the CGI gem before 0.4.2 for Ruby, a Regular Expression Denial of Service (ReDoS) vulnerability exists in the URI module
CVE-2025-27221	In the URI gem before 1.0.3 for Ruby, the URI handling methods (URI.join, URI#merge, URI#++) have an inadvertent Denial of Service
CVE-2025-27363	An out of bounds write exists in FreeType versions 2.13.0 and below (newer versions of FreeType are not vulnerable to this issue)
CVE-2025-27496	Snowflake, a platform for using artificial intelligence in the context of cloud computing, has a vulnerability in the Snowflake
CVE-2025-27516	Jinja is an extensible templating engine. Prior to 3.1.6, an oversight in how the Jinja sandboxed environment interacts with the
CVE-2025-27587	OpenSSL 3.0.0 through 3.3.2 on the PowerPC architecture is vulnerable to a Minerva attack, exploitable by measuring the execution
CVE-2025-27610	Rack provides an interface for developing web applications in Ruby. Prior to versions 2.2.13, 3.0.14, and 3.1.12, Rack::Session::Cookie
CVE-2025-27613	Gitk is a Tcl/Tk based Git history browser. Starting with 1.7.0, when a user clones an untrusted repository and runs gitk, a denial of service
CVE-2025-27614	Gitk is a Tcl/Tk based Git history browser. Starting with 2.41.0, a Git repository can be crafted in such a way that when gitk is
CVE-2025-27789	Babel is a compiler for writing next generation JavaScript. When using versions of Babel prior to 7.26.10 and 8.0.0-alpha.4, a
CVE-2025-27817	A possible arbitrary file read and SSRF vulnerability has been identified in Apache Kafka Client. Apache Kafka Client versions
CVE-2025-27820	A bug in PSL validation logic in Apache HttpClient 5.4.x disables domain checks, affecting cookie management and other

CVE-2025-2783	Incorrect handle provided in unspecified circumstances in Mojo in Google Chrome on Windows prior to 134.0.6998.110
CVE-2025-29087	In SQLite 3.44.0 through 3.49.0 before 3.49.1, the concat_ws() SQL function can cause memory to be written beyond the end of the buffer
CVE-2025-2828	A Server-Side Request Forgery (SSRF) vulnerability exists in the RequestsToolkit component of the langchain-community
CVE-2025-29088	In SQLite 3.49.0 before 3.49.1, certain argument values to sqlite3_db_config (in the C-language API) can cause a denial of service
CVE-2025-2953	A vulnerability, which was classified as problematic, has been found in PyTorch 2.6.0+cu124. Affected by this issue is the PyTorch
CVE-2025-29768	Vim, a text editor, is vulnerable to potential data loss with zip.vim and special crafted zip files in versions prior to 9.0.0
CVE-2025-29770	vLLM is a high-throughput and memory-efficient inference and serving engine for LLMs. The outlines library is one of the core dependencies
CVE-2025-29786	Expr is an expression language and expression evaluation for Go. Prior to version 1.17.0, if the Expr expression parser encounters a malformed expression, it can cause a denial of service
CVE-2025-30065	Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code
CVE-2025-30165	vLLM is an inference and serving engine for large language models. In a multi-node vLLM deployment using the vLLM
CVE-2025-30167	Jupyter Core is a package for the core common functionality of Jupyter projects. When using Jupyter Core prior to v0.1.0, a user can cause a denial of service
CVE-2025-30202	vLLM is a high-throughput and memory-efficient inference and serving engine for LLMs. Versions starting from 0.5.0, vLLM is vulnerable to a denial of service
CVE-2025-30258	In GnuPG before 2.5.5, if a user chooses to import a certificate with certain crafted subkey data that lacks a valid back signature, a denial of service can be achieved
CVE-2025-30204	golang-jwt is a Go implementation of JSON Web Tokens. Starting in version 3.2.0 and prior to versions 5.2.2 and 4.5.1, a denial of service can be achieved
CVE-2025-30370	jupyterlab-git is a JupyterLab extension for version control using Git. On many platforms, a third party can create a denial of service
CVE-2025-30473	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') vulnerability in Apache Airflow 2.10.0 and previous versions
CVE-2025-3066	Use after free in Site Isolation in Google Chrome prior to 135.0.7049.84 allowed a remote attacker to potentially exploit a browser vulnerability
CVE-2025-3068	Inappropriate implementation in Intents in Google Chrome on Android prior to 135.0.7049.52 allowed a remote attacker to potentially exploit a browser vulnerability
CVE-2025-3067	Inappropriate implementation in Custom Tabs in Google Chrome on Android prior to 135.0.7049.52 allowed a remote attacker to potentially exploit a browser vulnerability
CVE-2025-3069	Inappropriate implementation in Extensions in Google Chrome prior to 135.0.7049.52 allowed a remote attacker to potentially exploit a browser vulnerability
CVE-2025-30691	Vulnerability in Oracle Java SE (component: Compiler). Supported versions that are affected are Oracle Java SE: 21.0.4, 21.0.3, 21.0.2, 21.0.1, 21.0.0, 21.0.0-rc, 21.0.0-rc1, 21.0.0-rc2, 21.0.0-rc3, 21.0.0-rc4, 21.0.0-rc5, 21.0.0-rc6, 21.0.0-rc7, 21.0.0-rc8, 21.0.0-rc9, 21.0.0-rc10, 21.0.0-rc11, 21.0.0-rc12, 21.0.0-rc13, 21.0.0-rc14, 21.0.0-rc15, 21.0.0-rc16, 21.0.0-rc17, 21.0.0-rc18, 21.0.0-rc19, 21.0.0-rc20, 21.0.0-rc21, 21.0.0-rc22, 21.0.0-rc23, 21.0.0-rc24, 21.0.0-rc25, 21.0.0-rc26, 21.0.0-rc27, 21.0.0-rc28, 21.0.0-rc29, 21.0.0-rc30, 21.0.0-rc31, 21.0.0-rc32, 21.0.0-rc33, 21.0.0-rc34, 21.0.0-rc35, 21.0.0-rc36, 21.0.0-rc37, 21.0.0-rc38, 21.0.0-rc39, 21.0.0-rc40, 21.0.0-rc41, 21.0.0-rc42, 21.0.0-rc43, 21.0.0-rc44, 21.0.0-rc45, 21.0.0-rc46, 21.0.0-rc47, 21.0.0-rc48, 21.0.0-rc49, 21.0.0-rc50, 21.0.0-rc51, 21.0.0-rc52, 21.0.0-rc53, 21.0.0-rc54, 21.0.0-rc55, 21.0.0-rc56, 21.0.0-rc57, 21.0.0-rc58, 21.0.0-rc59, 21.0.0-rc60, 21.0.0-rc61, 21.0.0-rc62, 21.0.0-rc63, 21.0.0-rc64, 21.0.0-rc65, 21.0.0-rc66, 21.0.0-rc67, 21.0.0-rc68, 21.0.0-rc69, 21.0.0-rc70, 21.0.0-rc71, 21.0.0-rc72, 21.0.0-rc73, 21.0.0-rc74, 21.0.0-rc75, 21.0.0-rc76, 21.0.0-rc77, 21.0.0-rc78, 21.0.0-rc79, 21.0.0-rc80, 21.0.0-rc81, 21.0.0-rc82, 21.0.0-rc83, 21.0.0-rc84, 21.0.0-rc85, 21.0.0-rc86, 21.0.0-rc87, 21.0.0-rc88, 21.0.0-rc89, 21.0.0-rc90, 21.0.0-rc91, 21.0.0-rc92, 21.0.0-rc93, 21.0.0-rc94, 21.0.0-rc95, 21.0.0-rc96, 21.0.0-rc97, 21.0.0-rc98, 21.0.0-rc99, 21.0.0-rc100, 21.0.0-rc101, 21.0.0-rc102, 21.0.0-rc103, 21.0.0-rc104, 21.0.0-rc105, 21.0.0-rc106, 21.0.0-rc107, 21.0.0-rc108, 21.0.0-rc109, 21.0.0-rc110, 21.0.0-rc111, 21.0.0-rc112, 21.0.0-rc113, 21.0.0-rc114, 21.0.0-rc115, 21.0.0-rc116, 21.0.0-rc117, 21.0.0-rc118, 21.0.0-rc119, 21.0.0-rc120, 21.0.0-rc121, 21.0.0-rc122, 21.0.0-rc123, 21.0.0-rc124, 21.0.0-rc125, 21.0.0-rc126, 21.0.0-rc127, 21.0.0-rc128, 21.0.0-rc129, 21.0.0-rc130, 21.0.0-rc131, 21.0.0-rc132, 21.0.0-rc133, 21.0.0-rc134, 21.0.0-rc135, 21.0.0-rc136, 21.0.0-rc137, 21.0.0-rc138, 21.0.0-rc139, 21.0.0-rc140, 21.0.0-rc141, 21.0.0-rc142, 21.0.0-rc143, 21.0.0-rc144, 21.0.0-rc145, 21.0.0-rc146, 21.0.0-rc147, 21.0.0-rc148, 21.0.0-rc149, 21.0.0-rc150, 21.0.0-rc151, 21.0.0-rc152, 21.0.0-rc153, 21.0.0-rc154, 21.0.0-rc155, 21.0.0-rc156, 21.0.0-rc157, 21.0.0-rc158, 21.0.0-rc159, 21.0.0-rc160, 21.0.0-rc161, 21.0.0-rc162, 21.0.0-rc163, 21.0.0-rc164, 21.0.0-rc165, 21.0.0-rc166, 21.0.0-rc167, 21.0.0-rc168, 21.0.0-rc169, 21.0.0-rc170, 21.0.0-rc171, 21.0.0-rc172, 21.0.0-rc173, 21.0.0-rc174, 21.0.0-rc175, 21.0.0-rc176, 21.0.0-rc177, 21.0.0-rc178, 21.0.0-rc179, 21.0.0-rc180, 21.0.0-rc181, 21.0.0-rc182, 21.0.0-rc183, 21.0.0-rc184, 21.0.0-rc185, 21.0.0-rc186, 21.0.0-rc187, 21.0.0-rc188, 21.0.0-rc189, 21.0.0-rc190, 21.0.0-rc191, 21.0.0-rc192, 21.0.0-rc193, 21.0.0-rc194, 21.0.0-rc195, 21.0.0-rc196, 21.0.0-rc197, 21.0.0-rc198, 21.0.0-rc199, 21.0.0-rc200, 21.0.0-rc201, 21.0.0-rc202, 21.0.0-rc203, 21.0.0-rc204, 21.0.0-rc205, 21.0.0-rc206, 21.0.0-rc207, 21.0.0-rc208, 21.0.0-rc209, 21.0.0-rc210, 21.0.0-rc211, 21.0.0-rc212, 21.0.0-rc213, 21.0.0-rc214, 21.0.0-rc215, 21.0.0-rc216, 21.0.0-rc217, 21.0.0-rc218, 21.0.0-rc219, 21.0.0-rc220, 21.0.0-rc221, 21.0.0-rc222, 21.0.0-rc223, 21.0.0-rc224, 21.0.0-rc225, 21.0.0-rc226, 21.0.0-rc227, 21.0.0-rc228, 21.0.0-rc229, 21.0.0-rc230, 21.0.0-rc231, 21.0.0-rc232, 21.0.0-rc233, 21.0.0-rc234, 21.0.0-rc235, 21.0.0-rc236, 21.0.0-rc237, 21.0.0-rc238, 21.0.0-rc239, 21.0.0-rc240, 21.0.0-rc241, 21.0.0-rc242, 21.0.0-rc243, 21.0.0-rc244, 21.0.0-rc245, 21.0.0-rc246, 21.0.0-rc247, 21.0.0-rc248, 21.0.0-rc249, 21.0.0-rc250, 21.0.0-rc251, 21.0.0-rc252, 21.0.0-rc253, 21.0.0-rc254, 21.0.0-rc255, 21.0.0-rc256, 21.0.0-rc257, 21.0.0-rc258, 21.0.0-rc259, 21.0.0-rc260, 21.0.0-rc261, 21.0.0-rc262, 21.0.0-rc263, 21.0.0-rc264, 21.0.0-rc265, 21.0.0-rc266, 21.0.0-rc267, 21.0.0-rc268, 21.0.0-rc269, 21.0.0-rc270, 21.0.0-rc271, 21.0.0-rc272, 21.0.0-rc273, 21.0.0-rc274, 21.0.0-rc275, 21.0.0-rc276, 21.0.0-rc277, 21.0.0-rc278, 21.0.0-rc279, 21.0.0-rc280, 21.0.0-rc281, 21.0.0-rc282, 21.0.0-rc283, 21.0.0-rc284, 21.0.0-rc285, 21.0.0-rc286, 21.0.0-rc287, 21.0.0-rc288, 21.0.0-rc289, 21.0.0-rc290, 21.0.0-rc291, 21.0.0-rc292, 21.0.0-rc293, 21.0.0-rc294, 21.0.0-rc295, 21.0.0-rc296, 21.0.0-rc297, 21.0.0-rc298, 21.0.0-rc299, 21.0.0-rc300, 21.0.0-rc301, 21.0.0-rc302, 21.0.0-rc303, 21.0.0-rc304, 21.0.0-rc305, 21.0.0-rc306, 21.0.0-rc307, 21.0.0-rc308, 21.0.0-rc309, 21.0.0-rc310, 21.0.0-rc311, 21.0.0-rc312, 21.0.0-rc313, 21.0.0-rc314, 21.0.0-rc315, 21.0.0-rc316, 21.0.0-rc317, 21.0.0-rc318, 21.0.0-rc319, 21.0.0-rc320, 21.0.0-rc321, 21.0.0-rc322, 21.0.0-rc323, 21.0.0-rc324, 21.0.0-rc325, 21.0.0-rc326, 21.0.0-rc327, 21.0.0-rc328, 21.0.0-rc329, 21.0.0-rc330, 21.0.0-rc331, 21.0.0-rc332, 21.0.0-rc333, 21.0.0-rc334, 21.0.0-rc335, 21.0.0-rc336, 21.0.0-rc337, 21.0.0-rc338, 21.0.0-rc339, 21.0.0-rc340, 21.0.0-rc341, 21.0.0-rc342, 21.0.0-rc343, 21.0.0-rc344, 21.0.0-rc345, 21.0.0-rc346, 21.0.0-rc347, 21.0.0-rc348, 21.0.0-rc349, 21.0.0-rc350, 21.0.0-rc351, 21.0.0-rc352, 21.0.0-rc353, 21.0.0-rc354, 21.0.0-rc355, 21.0.0-rc356, 21.0.0-rc357, 21.0.0-rc358, 21.0.0-rc359, 21.0.0-rc360, 21.0.0-rc361, 21.0.0-rc362, 21.0.0-rc363, 21.0.0-rc364, 21.0.0-rc365, 21.0.0-rc366, 21.0.0-rc367, 21.0.0-rc368, 21.0.0-rc369, 21.0.0-rc370, 21.0.0-rc371, 21.0.0-rc372, 21.0.0-rc373, 21.0.0-rc374, 21.0.0-rc375, 21.0.0-rc376, 21.0.0-rc377, 21.0.0-rc378, 21.0.0-rc379, 21.0.0-rc380, 21.0.0-rc381, 21.0.0-rc382, 21.0.0-rc383, 21.0.0-rc384, 21.0.0-rc385, 21.0.0-rc386, 21.0.0-rc387, 21.0.0-rc388, 21.0.0-rc389, 21.0.0-rc390, 21.0.0-rc391, 21.0.0-rc392, 21.0.0-rc393, 21.0.0-rc394, 21.0.0-rc395, 21.0.0-rc396, 21.0.0-rc397, 21.0.0-rc398, 21.0.0-rc399, 21.0.0-rc400, 21.0.0-rc401, 21.0.0-rc402, 21.0.0-rc403, 21.0.0-rc404, 21.0.0-rc405, 21.0.0-rc406, 21.0.0-rc407, 21.0.0-rc408, 21.0.0-rc409, 21.0.0-rc410, 21.0.0-rc411, 21.0.0-rc412, 21.0.0-rc413, 21.0.0-rc414, 21.0.0-rc415, 21.0.0-rc416, 21.0.0-rc417, 21.0.0-rc418, 21.0.0-rc419, 21.0.0-rc420, 21.0.0-rc421, 21.0.0-rc422, 21.0.0-rc423, 21.0.0-rc424, 21.0.0-rc425, 21.0.0-rc426, 21.0.0-rc427, 21.0.0-rc428, 21.0.0-rc429, 21.0.0-rc430, 21.0.0-rc431, 21.0.0-rc432, 21.0.0-rc433, 21.0.0-rc434, 21.0.0-rc435, 21.0.0-rc436, 21.0.0-rc437, 21.0.0-rc438, 21.0.0-rc439, 21.0.0-rc440, 21.0.0-rc441, 21.0.0-rc442, 21.0.0-rc443, 21.0.0-rc444, 21.0.0-rc445, 21.0.0-rc446, 21.0.0-rc447, 21.0.0-rc448, 21.0.0-rc449, 21.0.0-rc450, 21.0.0-rc451, 21.0.0-rc452, 21.0.0-rc453, 21.0.0-rc454, 21.0.0-rc455, 21.0.0-rc456, 21.0.0-rc457, 21.0.0-rc458, 21.0.0-rc459, 21.0.0-rc460, 21.0.0-rc461, 21.0.0-rc462, 21.0.0-rc463, 21.0.0-rc464, 21.0.0-rc465, 21.0.0-rc466, 21.0.0-rc467, 21.0.0-rc468, 21.0.0-rc469, 21.0.0-rc470, 21.0.0-rc471, 21.0.0-rc472, 21.0.0-rc473, 21.0.0-rc474, 21.0.0-rc475, 21.0.0-rc476, 21.0.0-rc477, 21.0.0-rc478, 21.0.0-rc479, 21.0.0-rc480, 21.0.0-rc481, 21.0.0-rc482, 21.0.0-rc483, 21.0.0-rc484, 21.0.0-rc485, 21.0.0-rc486, 21.0.0-rc487, 21.0.0-rc488, 21.0.0-rc489, 21.0.0-rc490, 21.0.0-rc491, 21.0.0-rc492, 21.0.0-rc493, 21.0.0-rc494, 21.0.0-rc495, 21.0.0-rc496, 21.0.0-rc497, 21.0.0-rc498, 21.0.0-rc499, 21.0.0-rc500, 21.0.0-rc501, 21.0.0-rc502, 21.0.0-rc503, 21.0.0-rc504, 21.0.0-rc505, 21.0.0-rc506, 21.0.0-rc507, 21.0.0-rc508, 21.0.0-rc509, 21.0.0-rc510, 21.0.0-rc511, 21.0.0-rc512, 21.0.0-rc513, 21.0.0-rc514, 21.0.0-rc515, 21.0.0-rc516, 21.0.0-rc517, 21.0.0-rc518, 21.0.0-rc519, 21.0.0-rc520, 21.0.0-rc521, 21.0.0-rc522, 21.0.0-rc523, 21.0.0-rc524, 21.0.0-rc525, 21.0.0-rc526, 21.0.0-rc527, 21.0.0-rc528, 21.0.0-rc529, 21.0.0-rc530, 21.0.0-rc531, 21.0.0-rc532, 21.0.0-rc533, 21.0.0-rc534, 21.0.0-rc535, 21.0.0-rc536, 21.0.0-rc537, 21.0.0-rc538, 21.0.0-rc539, 21.0.0-rc540, 21.0.0-rc541, 21.0.0-rc542, 21.0.0-rc543, 21.0.0-rc544, 21.0.0-rc545, 21.0.0-rc546, 21.0.0-rc547, 21.0.0-rc548, 21.0.0-rc549, 21.0.0-rc550, 21.0.0-rc551, 21.0.0-rc552, 21.0.0-rc553, 21.0.0-rc554, 21.0.0-rc555, 21.0.0-rc556, 21.0.0-rc557, 21.0.0-rc558, 21.0.0-rc559, 21.0.0-rc560, 21.0.0-rc561, 21.0.0-rc562, 21.0.0-rc563, 21.0.0-rc564, 21.0.0-rc565, 21.0.0-rc566, 21.0.0-rc567, 21.0.0-rc568, 21.0.0-rc569, 21.0.0-rc570, 21.0.0-rc571, 21.0.0-rc572, 21.0.0-rc573, 21.0.0-rc574, 21.0.0-rc575, 21.0.0-rc576, 21.0.0-rc577, 21.0.0-rc578, 21.0.0-rc579, 21.0.0-rc580, 21.0.0-rc581, 21.0.0-rc582, 21.0.0-rc583, 21.0.0-rc584, 21.0.0-rc585, 21.0.0-rc586, 21.0.0-rc587, 21.0.0-rc588, 21.0.0-rc589, 21.0.0-rc590, 21.0.0-rc591, 21.0.0-rc592, 21.0.0-rc593, 21.0.0-rc594, 21.0.0-rc595, 21.0.0-rc596, 21.0.0-rc597, 21.0.0-rc598, 21.0.0-rc599, 21.0.0-rc600, 21.0.0-rc601, 21.0.0-rc602, 21.0.0-rc603, 21.0.0-rc604, 21.0.0-rc605, 21.0.0-rc606, 21.0.0-rc607, 21.0.0-rc608, 21.0.0-rc609, 21.0.0-rc610, 21.0.0-rc611, 21.0.0-rc612, 21.0.0-rc613, 21.0.0-rc614, 21.0.0-rc615, 21.0.0-rc616, 21.0.0-rc617, 21.0.0-rc618, 21.0.0-rc619, 21.0.0-rc620, 21.0.0-rc621, 21.0.0-rc622, 21.0.0-rc623, 21.0.0-rc624, 21.0.0-rc625, 21.0.0-rc626, 21.0.0-rc627, 21.0.0-rc628, 21.0.0-rc629, 21.0.0-rc630, 21.0.0-rc631, 21.0.0-rc632, 21.0.0-rc633, 21.0.0-rc634, 21.0.0-rc635, 21.0.0-rc636, 21.0.0-rc637, 21.0.0-rc638, 21.0.0-rc639, 21.0.0-rc640, 21.0.0-rc641, 21.0.0-rc642, 21.0.0-rc643, 21.0.0-rc644, 21.0.0-rc645, 21.0.0-rc646, 21.0.0-rc647, 21.0.0-rc648, 21.0.0-rc649, 21.0.0-rc650, 21.0.0-rc651, 21.0.0-rc652, 21.0.0-rc653, 21.0.0-rc654, 21.0.0-rc655, 21.0.0-rc656, 21.0.0-rc657, 21.0.0-rc658, 21.0.0-rc659, 21.0.0-rc660, 21.0.0-rc661, 21.0.0-rc662, 21.0.0-rc663, 21.0.0-rc664, 21.0.0-rc665, 21.0.0-rc666, 21.0.0-rc667, 21.0.0-rc668, 21.0.0-rc669, 21.0.0-rc670, 21.0.0-rc671, 21.0.0-rc672, 21.0.0-rc673, 21.0.0-rc674, 21.0.0-rc675, 21.0.0-rc676, 21.0.0-rc677, 21.0.0-rc678, 21.0.0-rc679, 21.0.0-rc680, 21.0.0-rc681, 21.0.0-rc682, 21.0.0-rc683, 21.0.0-rc684, 21.0.0-rc685, 21.0.0-rc686, 21.0.0-rc687, 21.0.0-rc688, 21.0.0-rc689, 21.0.0-rc690, 21.0.0-rc691, 21.0.0-rc692, 21.0.0-rc693, 21.0.0-rc694, 21.0.0-rc695, 21.0.0-rc696, 21.0.0-rc697, 21.0.0-rc698, 21.0.0-rc699, 21.0.0-rc700, 21.0.0-rc701, 21.0.0-rc702, 21.0.0-rc703, 21.0.0-rc704, 21.0.0-rc705, 21.0.0-rc706, 21.0.0-rc707, 21.0.0-rc708, 21.0.0-rc709, 21.0.0-rc710, 21.0.0-rc711, 21.0.0-rc712, 21.0.0-rc713, 21.0.0-rc714, 21.0.0-rc715, 21.0.0-rc716, 21.0.0-rc717, 21.0.0-rc718, 21.0.0-rc719, 21.0.0-rc720, 21.0.0-rc721, 21.0.0-rc722, 21.0.0-rc723, 21.0.0-rc724, 21.0.0-rc725, 21.0.0-rc726, 21.0.0-rc727, 21.0.0-rc728, 21.0.0-rc729, 21.0.0-rc730, 21.0.0-rc731, 21.0.0-rc732, 21.0.0-rc733, 21.0.0-rc734, 21.0.0-rc735, 21.0.0-rc736, 21.0.0-rc737, 21.0.0-rc738, 21.0.0-rc739, 21.0.0-rc740, 21.0.0-rc741, 21.0.0-rc742, 21.0.0-rc743, 21.0.0-rc744, 21.0.0-rc745, 21.0.0-rc746, 21.0.0-rc747, 21.0.0-rc748, 21.0.0-rc749, 21.0.0-rc750, 21.0.0-rc751, 21.0.0-rc752, 21.0.0-rc753, 21.0.0-rc754, 21.0.0-rc755, 21.0.0-rc756, 21.0.0-rc757, 21.0.0-rc758, 21.0.0-rc759, 21.0.0-rc760, 21.0.0-rc761, 21.0.0-rc762, 21.0.0-rc763, 21.0.0-rc764, 21.0.0-rc765, 21.0.0-rc766, 21.0.0-rc767, 21.0.0-rc768, 21.0.0-rc769, 21.0.0-rc770, 21.0.0-rc771, 21.0.0-rc772, 21.0.0-rc773, 21.0.0-rc774, 21.0.0-rc775, 21.0.0-rc776, 21.0.0-rc777, 21.0.0-rc778, 21.0.0-rc779, 21.0.0-rc780, 21.0.0-rc781, 21.0.0-rc782, 21.0.0-rc783, 21.0.0-rc784, 21.0.0-rc785, 21.0.0-rc786, 21.0.0-rc787, 21.0.0-rc788, 21.0.0-rc789, 21.0.0-rc790, 21.0.0-rc791, 21.0.0-rc792, 21.0.0-rc793, 21.0.0-rc794, 21.0.0-rc795, 21.0.0-rc796, 21.0.0-rc797, 21.0.0-rc798, 21.0.0-rc799, 21.0.0-rc800, 21.0.0-rc801, 21.0.0-rc802, 21.0.0-rc803, 21.0.0-rc804, 21.0.0-rc805, 21.0.0-rc806, 21.0.0-rc807, 21.0.0-rc808, 21.0.0-rc809, 21.0.0-rc810, 21.0.0-rc811, 21.0.0-rc812, 21.0.0-rc813, 21.0.0-rc814, 21.0.0-rc815, 21.0.0-rc816, 21.0.0-rc817, 21.0.0-rc818, 21.0.0-rc819, 21.0.0-rc820, 21.0.0-rc821, 21.0.0-rc822, 21.0.0-rc823, 21.0.0-rc824, 21.0.0-rc825, 21.0.0-rc826, 21.0.0-rc827, 21.0.0-rc828, 21.0.0-rc829, 21.0.0-rc830, 21.0.0-rc831, 21.0.0-rc832, 21.0.0-rc833, 21.0.0-rc834, 21.0.0-rc835, 21.0.0-rc836, 21.0.0-rc837, 21.0.0-rc838, 21.0.0-rc839, 21.0.0-rc840, 21.0.0-rc841, 21.0.0-rc8

CVE-2025-32415	In libxml2 before 2.13.8 and 2.14.x before 2.14.2, xmlSchemaIDCFillNodeTables in xmlschemas.c has a heap-based
CVE-2025-32434	PyTorch is a Python package that provides tensor computation with strong GPU acceleration and deep neural network
CVE-2025-32444	vLLM is a high-throughput and memory-efficient inference and serving engine for LLMs. Versions starting from 0.6
CVE-2025-32462	Sudo before 1.9.17p1, when used with a sudoers file that specifies a host that is neither the current host nor ALL, all
CVE-2025-32463	Sudo before 1.9.17p1 allows local users to obtain root access because /etc/nsswitch.conf from a user-controlled direc
CVE-2025-3262	A Regular Expression Denial of Service (ReDoS) vulnerability was discovered in the huggingface/transformers repo
CVE-2025-3263	A Regular Expression Denial of Service (ReDoS) vulnerability was discovered in the Hugging Face Transformers lib
CVE-2025-3264	A Regular Expression Denial of Service (ReDoS) vulnerability was discovered in the Hugging Face Transformers lib
CVE-2025-32728	In sshd in OpenSSH before 10.0, the DisableForwarding directive does not adhere to the documentation stating that
CVE-2025-3277	An integer overflow can be triggered in SQLite's `concat_ws()` function. The resulting, truncated integer is then use
CVE-2025-32873	An issue was discovered in Django 4.2 before 4.2.21, 5.1 before 5.1.9, and 5.2 before 5.2.1. The django.utils.html.str
CVE-2025-32912	A flaw was found in libsoup, where SoupAuthDigest is vulnerable to a NULL pointer dereference. The HTTP server
CVE-2025-32962	Flask-AppBuilder is an application development framework built on top of Flask. Versions prior to 4.6.2 would allo
CVE-2025-32988	A flaw was found in GnuTLS. A double-free vulnerability exists in GnuTLS due to incorrect ownership handling in
CVE-2025-32989	A heap-buffer-overread vulnerability was found in GnuTLS in how it handles the Certificate Transparency (CT) Sign
CVE-2025-32990	A heap-buffer-overflow (off-by-one) flaw was found in the GnuTLS software in the template parsing logic within th
CVE-2025-3360	A flaw was found in GLib. An integer overflow and buffer under-read occur when parsing a long invalid ISO 8601 t
CVE-2025-35036	Hibernate Validator before 6.2.0 and 7.0.0, by default and depending how it is used, may interpolate user-supplied in
CVE-2025-3576	A vulnerability in the MIT Kerberos implementation allows GSSAPI-protected messages using RC4-HMAC-MD5 t
CVE-2025-3588	A vulnerability, which was classified as problematic, has been found in joelittlejohn jsonschema2pojo 1.2.2. This iss
CVE-2025-3619	Heap buffer overflow in Codecs in Google Chrome on Windows prior to 135.0.7049.95 allowed a remote attacker to
CVE-2025-3620	Use after free in USB in Google Chrome prior to 135.0.7049.95 allowed a remote attacker to potentially exploit heap
CVE-2025-3730	A vulnerability, which was classified as problematic, was found in PyTorch 2.6.0. Affected is the function torch.nn.f
CVE-2025-37738	In the Linux kernel, the following vulnerability has been resolved: ext4: ignore xattrs past end Once inside 'ext4_xatt
CVE-2025-37739	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid out-of-bounds access in f2fs_tru
CVE-2025-37740	In the Linux kernel, the following vulnerability has been resolved: jfs: add sanity check for agwidth in dbMount The
CVE-2025-37741	In the Linux kernel, the following vulnerability has been resolved: jfs: Prevent copying of nlink with value 0 from di
CVE-2025-37742	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix uninit-value access of imap allocated in th
CVE-2025-37743	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Avoid memory leak while enabling
CVE-2025-37744	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix memory leak in ath12k_pci_rem
CVE-2025-37745	In the Linux kernel, the following vulnerability has been resolved: PM: hibernate: Avoid deadlock in hibernate_com
CVE-2025-37746	In the Linux kernel, the following vulnerability has been resolved: perf/dwc_pcie: fix duplicate pci_dev devices Dur
CVE-2025-37747	In the Linux kernel, the following vulnerability has been resolved: perf: Fix hang while freeing sigtrap event Perf ca
CVE-2025-37748	In the Linux kernel, the following vulnerability has been resolved: iommu/mediatek: Fix NULL pointer dereference in
CVE-2025-37749	In the Linux kernel, the following vulnerability has been resolved: net: ppp: Add bound checking for skb data on ppp
CVE-2025-37752	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: move the limit validation It is
CVE-2025-37754	In the Linux kernel, the following vulnerability has been resolved: drm/i915/huc: Fix fence not released on early pro
CVE-2025-37755	In the Linux kernel, the following vulnerability has been resolved: net: libwx: handle page_pool_dev_alloc_pages er
CVE-2025-37756	In the Linux kernel, the following vulnerability has been resolved: net: tls: explicitly disallow disconnect syzbot disc
CVE-2025-37757	In the Linux kernel, the following vulnerability has been resolved: tipc: fix memory leak in tipc_link_xmit In case th

CVE-2025-37758	In the Linux kernel, the following vulnerability has been resolved: ata: pata_pxa: Fix potential NULL pointer dereference
CVE-2025-37759	In the Linux kernel, the following vulnerability has been resolved: ublk: fix handling recovery & reissue in ublk_abort
CVE-2025-37763	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: take paired job reference For paired jobs
CVE-2025-37761	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix an out-of-bounds shift when invalidating
CVE-2025-37764	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: fix firmware memory leaks Free
CVE-2025-37765	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: prime: fix ttm_bo_delayed_delete
CVE-2025-37766	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user c
CVE-2025-37767	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user c
CVE-2025-37768	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user c
CVE-2025-37769	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm/smu11: Prevent division by zero The
CVE-2025-3777	Hugging Face Transformers versions up to 4.49.0 are affected by an improper input validation vulnerability in the `in
CVE-2025-37770	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user c
CVE-2025-37771	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user c
CVE-2025-37772	In the Linux kernel, the following vulnerability has been resolved: RDMA/cma: Fix workqueue crash in cma_netevent
CVE-2025-37773	In the Linux kernel, the following vulnerability has been resolved: virtiofs: add filesystem context source name chec
CVE-2025-37775	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix the warning from __kernel_write_iter
CVE-2025-37776	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in smb_break_all_levII
CVE-2025-37777	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in __smb2_lease_break
CVE-2025-37778	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Fix dangling pointer in krb_authenticate k
CVE-2025-37780	In the Linux kernel, the following vulnerability has been resolved: isofs: Prevent the use of too small fid syzbot repo
CVE-2025-37781	In the Linux kernel, the following vulnerability has been resolved: i2c: cros-ec-tunnel: defer probe if parent EC is no
CVE-2025-37784	In the Linux kernel, the following vulnerability has been resolved: net: ti: icss-iep: Fix possible NULL pointer deref
CVE-2025-37786	In the Linux kernel, the following vulnerability has been resolved: net: dsa: free routing table on probe failure If com
CVE-2025-37785	In the Linux kernel, the following vulnerability has been resolved: ext4: fix OOB read when checking dotdot dir Mo
CVE-2025-37787	In the Linux kernel, the following vulnerability has been resolved: net: dsa: mv88e6xxx: avoid unregistering devlink
CVE-2025-37788	In the Linux kernel, the following vulnerability has been resolved: cxgb4: fix memory leak in cxgb4_init_ethtool_fil
CVE-2025-37789	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix nested key length validation
CVE-2025-37790	In the Linux kernel, the following vulnerability has been resolved: net: mctp: Set SOCK_RCU_FREE Bind lookup r
CVE-2025-37792	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btrtl: Prevent potential NULL derefer
CVE-2025-37794	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Purge vif txq in ieee80211_do_s
CVE-2025-37793	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: avs: Fix null-ptr-deref in avs_compo
CVE-2025-37795	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2025-37797	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Fix a UAF vulnerability in class
CVE-2025-37796	In the Linux kernel, the following vulnerability has been resolved: wifi: at76c50x: fix use after free access in at76_d
CVE-2025-37798	In the Linux kernel, the following vulnerability has been resolved: codell: remove sch->q.qlen check before qdisc_tre
CVE-2025-37799	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: Fix malformed packet sizing in vmxnet
CVE-2025-37800	In the Linux kernel, the following vulnerability has been resolved: driver core: fix potential NULL pointer dereferen
CVE-2025-37801	In the Linux kernel, the following vulnerability has been resolved: spi: spi-imx: Add check for spi_imx_setupxfer() a
CVE-2025-37802	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix WARNING "do not call blocking ops
CVE-2025-37803	In the Linux kernel, the following vulnerability has been resolved: udmabuf: fix a buf size overflow issue during udr

CVE-2025-37806	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Keep write operations atomic syzbot rep
CVE-2025-37805	In the Linux kernel, the following vulnerability has been resolved: sound/virtio: Fix cancel_sync warnings on uniniti
CVE-2025-37807	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix kmemleak warning for percpu hashmap V
CVE-2025-37808	In the Linux kernel, the following vulnerability has been resolved: crypto: null - Use spin lock instead of mutex As t
CVE-2025-37809	In the Linux kernel, the following vulnerability has been resolved: usb: typec: class: Fix NULL pointer access Concu
CVE-2025-37810	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: check that event count does no
CVE-2025-37811	In the Linux kernel, the following vulnerability has been resolved: usb: chipidea: ci_hdrc_imx: fix usbmisc handling
CVE-2025-37812	In the Linux kernel, the following vulnerability has been resolved: usb: cdns3: Fix deadlock when using NCM gadge
CVE-2025-37815	In the Linux kernel, the following vulnerability has been resolved: misc: microchip: pci1xxx: Fix Kernel panic duri
CVE-2025-37816	In the Linux kernel, the following vulnerability has been resolved: mei: vsc: Fix fortify-panic caused by invalid coun
CVE-2025-37813	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Fix invalid pointer dereference in Etron
CVE-2025-37817	In the Linux kernel, the following vulnerability has been resolved: mcb: fix a double free bug in chameleon_parse_g
CVE-2025-37819	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v2m: Prevent use after free of gicv2m
CVE-2025-37820	In the Linux kernel, the following vulnerability has been resolved: xen-netfront: handle NULL returned by xdp_conv
CVE-2025-37822	In the Linux kernel, the following vulnerability has been resolved: riscv: uprobes: Add missing fence.i after building
CVE-2025-37823	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Fix a potential UAF in hfsc_dequ
CVE-2025-37824	In the Linux kernel, the following vulnerability has been resolved: tipc: fix NULL pointer dereference in tipc_mon_i
CVE-2025-37826	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Add NULL check in ufshcd_mcq_
CVE-2025-37827	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: return EIO on RAID1 block group w
CVE-2025-37828	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: mcq: Add NULL check in ufshcd_mcq_
CVE-2025-37829	In the Linux kernel, the following vulnerability has been resolved: cpufreq: scpi: Fix null-ptr-deref in scpi_cpufreq_
CVE-2025-37830	In the Linux kernel, the following vulnerability has been resolved: cpufreq: scmi: Fix null-ptr-deref in scmi_cpufreq
CVE-2025-37831	In the Linux kernel, the following vulnerability has been resolved: cpufreq: apple-soc: Fix null-ptr-deref in apple_so
CVE-2025-37833	In the Linux kernel, the following vulnerability has been resolved: net/niu: Niu requires MSIX ENTRY_DATA field
CVE-2025-37834	In the Linux kernel, the following vulnerability has been resolved: mm/vmscan: don't try to reclaim hwpoison folio S
CVE-2025-37838	In the Linux kernel, the following vulnerability has been resolved: HSI: ssi_protocol: Fix use after free vulnerability
CVE-2025-37836	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix reference leak in pci_register_host_bridg
CVE-2025-37839	In the Linux kernel, the following vulnerability has been resolved: jbd2: remove wrong sb->s_sequence check Journ
CVE-2025-37840	In the Linux kernel, the following vulnerability has been resolved: mtd: rawnand: brcmnand: fix PM resume warning
CVE-2025-37841	In the Linux kernel, the following vulnerability has been resolved: pm: cpupower: bench: Prevent NULL dereference
CVE-2025-37842	In the Linux kernel, the following vulnerability has been resolved: spi: fsl-qspi: use devm function instead of driver
CVE-2025-37844	In the Linux kernel, the following vulnerability has been resolved: cifs: avoid NULL pointer dereference in dbg call
CVE-2025-37846	In the Linux kernel, the following vulnerability has been resolved: arm64: mops: Do not dereference src reg for a set
CVE-2025-37849	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Tear down vGIC on failed vCPU c
CVE-2025-37850	In the Linux kernel, the following vulnerability has been resolved: pwm: mediatek: Prevent divide-by-zero in pwm_
CVE-2025-37851	In the Linux kernel, the following vulnerability has been resolved: fbdev: omapfb: Add 'plane' value check Function
CVE-2025-37852	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: handle amdgpu_cgs_create_device
CVE-2025-37853	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: debugfs hang_hws skip GPU with M
CVE-2025-37855	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Guard Possible Null Pointer De
CVE-2025-37854	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix mode1 reset crash issue If HW s

CVE-2025-37856	In the Linux kernel, the following vulnerability has been resolved: btrfs: harden block_group::bg_list against list_del
CVE-2025-37857	In the Linux kernel, the following vulnerability has been resolved: scsi: st: Fix array overflow in st_setup() Change t
CVE-2025-37858	In the Linux kernel, the following vulnerability has been resolved: fs/jfs: Prevent integer overflow in AG size calcula
CVE-2025-37861	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Synchronous access b/w reset and t
CVE-2025-37863	In the Linux kernel, the following vulnerability has been resolved: ovl: don't allow datadir only In theory overlaysfs c
CVE-2025-37862	In the Linux kernel, the following vulnerability has been resolved: HID: pidff: Fix null pointer dereference in pidff_f
CVE-2025-37859	In the Linux kernel, the following vulnerability has been resolved: page_pool: avoid infinite loop to schedule delayed
CVE-2025-37860	In the Linux kernel, the following vulnerability has been resolved: sfc: fix NULL dereferences in ef100_process_des
CVE-2025-37864	In the Linux kernel, the following vulnerability has been resolved: net: dsa: clean up FDB, MDB, VLAN entries on t
CVE-2025-37869	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Use local fence in error path of xe_migra
CVE-2025-37870	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: prevent hang on link training fa
CVE-2025-37865	In the Linux kernel, the following vulnerability has been resolved: net: dsa: mv88e6xxx: fix -ENOENT when deletin
CVE-2025-37867	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Silence oversized kvmalloc() warni
CVE-2025-37872	In the Linux kernel, the following vulnerability has been resolved: net: txgbe: fix memory leak in txgbe_probe() erro
CVE-2025-37873	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: fix missing ring index trim on error pat
CVE-2025-37875	In the Linux kernel, the following vulnerability has been resolved: igc: fix PTM cycle trigger logic Writing to clear t
CVE-2025-37874	In the Linux kernel, the following vulnerability has been resolved: net: ngbe: fix memory leak in ngbe_probe() erro
CVE-2025-37876	In the Linux kernel, the following vulnerability has been resolved: netfs: Only create /proc/fs/netfs with CONFIG_P
CVE-2025-37877	In the Linux kernel, the following vulnerability has been resolved: iommu: Clear iommu-dma ops on cleanup If iom
CVE-2025-37878	In the Linux kernel, the following vulnerability has been resolved: perf/core: Fix WARN_ON(!ctx) in __free_event(
CVE-2025-37879	In the Linux kernel, the following vulnerability has been resolved: 9p/net: fix improper handling of bogus negative r
CVE-2025-37880	In the Linux kernel, the following vulnerability has been resolved: um: work around sched_yield not yielding in time
CVE-2025-37881	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: aspeed: Add NULL pointer check in
CVE-2025-37882	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Fix isochronous Ring Underrun/Overru
CVE-2025-37883	In the Linux kernel, the following vulnerability has been resolved: s390/sclp: Add check for get_zeroed_page() Add
CVE-2025-37884	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix deadlock between rcu_tasks_trace and ev
CVE-2025-37885	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Reset IRTE to host control if *new* r
CVE-2025-37886	In the Linux kernel, the following vulnerability has been resolved: pds_core: make wait_context part of q_info Make
CVE-2025-37887	In the Linux kernel, the following vulnerability has been resolved: pds_core: handle unsupported PDS_CORE_CMD
CVE-2025-37889	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Consistently treat platform_max as co
CVE-2025-37890	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Fix a UAF vulnerability in class
CVE-2025-37891	In the Linux kernel, the following vulnerability has been resolved: ALSA: ump: Fix buffer overflow at UMP SysEx
CVE-2025-37892	In the Linux kernel, the following vulnerability has been resolved: mtd: infatcore: Add error check for infat_read_oob
CVE-2025-37897	In the Linux kernel, the following vulnerability has been resolved: wifi: plfxc: Remove erroneous assert in plfxc_m
CVE-2025-37899	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in session logoff The ses
CVE-2025-37900	In the Linux kernel, the following vulnerability has been resolved: iommu: Fix two issues in iommu_copy_struct_fro
CVE-2025-37901	In the Linux kernel, the following vulnerability has been resolved: irqchip/qcom-mpm: Prevent crash when trying to
CVE-2025-37903	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix slab-use-after-free in hdcp
CVE-2025-37905	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Balance device refcount wh
CVE-2025-37906	In the Linux kernel, the following vulnerability has been resolved: ublk: fix race between io_uring_cmd_complete_i

CVE-2025-37909	In the Linux kernel, the following vulnerability has been resolved: net: lan743x: Fix memleak issue when GSO enabled
CVE-2025-37907	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix locking order in ivpu_job_submit
CVE-2025-37911	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix out-of-bound memcpy() during ethtool
CVE-2025-37912	In the Linux kernel, the following vulnerability has been resolved: ice: Check VF VSI Pointer Value in ice_vc_add
CVE-2025-37913	In the Linux kernel, the following vulnerability has been resolved: net_sched: qfq: Fix double list add in class with n
CVE-2025-37914	In the Linux kernel, the following vulnerability has been resolved: net_sched: ets: Fix double list add in class with n
CVE-2025-37915	In the Linux kernel, the following vulnerability has been resolved: net_sched: drr: Fix double list add in class with n
CVE-2025-37916	In the Linux kernel, the following vulnerability has been resolved: pds_core: remove write-after-free of client_id A u
CVE-2025-37917	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk-star-emas: fix spinlock recursio
CVE-2025-37920	In the Linux kernel, the following vulnerability has been resolved: xsk: Fix race condition in AF_XDP generic RX p
CVE-2025-37918	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btusb: avoid NULL pointer dereferenc
CVE-2025-37921	In the Linux kernel, the following vulnerability has been resolved: vxlan: vnifilter: Fix unlocked deletion of default l
CVE-2025-37922	In the Linux kernel, the following vulnerability has been resolved: book3s64/radix : Align section vmemmap start ad
CVE-2025-37923	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix oob write in trace_seq_to_buffer() sy
CVE-2025-37924	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in kerberos authentication
CVE-2025-37925	In the Linux kernel, the following vulnerability has been resolved: jfs: reject on-disk inodes of an unsupported type S
CVE-2025-37926	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in ksmbd_session_rpc_c
CVE-2025-37927	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix potential buffer overflow in pars
CVE-2025-37928	In the Linux kernel, the following vulnerability has been resolved: dm-bufio: don't schedule in atomic context A BU
CVE-2025-37930	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: Fix WARN_ON in nouveau_fence
CVE-2025-37932	In the Linux kernel, the following vulnerability has been resolved: sch_htb: make htb_qlen_notify() idempotent htb_
CVE-2025-37933	In the Linux kernel, the following vulnerability has been resolved: octeon_ep: Fix host hang issue during device rebo
CVE-2025-37931	In the Linux kernel, the following vulnerability has been resolved: btrfs: adjust subpage bit start based on sectorsize
CVE-2025-37935	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk_eth_soc: fix SER panic with 4G
CVE-2025-37936	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: KVM: Mask PEBS_ENABLE load
CVE-2025-37937	In the Linux kernel, the following vulnerability has been resolved: objtool, media: dib8000: Prevent divide-by-zero i
CVE-2025-37940	In the Linux kernel, the following vulnerability has been resolved: ftrace: Add cond_resched() to ftrace_graph_set_h
CVE-2025-37938	In the Linux kernel, the following vulnerability has been resolved: tracing: Verify event formats that have "%*p.." T
CVE-2025-37942	In the Linux kernel, the following vulnerability has been resolved: HID: pidff: Make sure to fetch pool before checki
CVE-2025-37943	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix invalid data access in ath12k_dp
CVE-2025-37944	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix invalid entry fetch in ath12k_dp
CVE-2025-37947	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent out-of-bounds stream writes by v
CVE-2025-37945	In the Linux kernel, the following vulnerability has been resolved: net: phy: allow MDIO bus PM ops to start/stop st
CVE-2025-37948	In the Linux kernel, the following vulnerability has been resolved: arm64: bpf: Add BHB mitigation to the epilogue
CVE-2025-37949	In the Linux kernel, the following vulnerability has been resolved: xenbus: Use kref to track req lifetime Marek repo
CVE-2025-37951	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Add job to pending list if the reset was s
CVE-2025-37952	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Fix UAF in __close_file_table_ids A use-
CVE-2025-37954	In the Linux kernel, the following vulnerability has been resolved: smb: client: Avoid race in open_cached_dir with
CVE-2025-37956	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent rename with empty string Client c
CVE-2025-37957	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Forcibly leave SMM mode on SHU

CVE-2025-37960	In the Linux kernel, the following vulnerability has been resolved: memblock: Accept allocated memory before use
CVE-2025-37959	In the Linux kernel, the following vulnerability has been resolved: bpf: Scrub packet on bpf_redirect_peer When bpf
CVE-2025-37958	In the Linux kernel, the following vulnerability has been resolved: mm/huge_memory: fix dereferencing invalid pmo
CVE-2025-37961	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix uninit-value for saddr in do_output_rout
CVE-2025-37963	In the Linux kernel, the following vulnerability has been resolved: arm64: bpf: Only mitigate cBPF programs loaded
CVE-2025-37969	In the Linux kernel, the following vulnerability has been resolved: iio: imu: st_lsm6dsx: fix possible lockup in st_lsm
CVE-2025-37967	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: displayport: Fix deadlock This p
CVE-2025-37970	In the Linux kernel, the following vulnerability has been resolved: iio: imu: st_lsm6dsx: fix possible lockup in st_lsm
CVE-2025-37968	In the Linux kernel, the following vulnerability has been resolved: iio: light: opt3001: fix deadlock due to concurr
CVE-2025-37972	In the Linux kernel, the following vulnerability has been resolved: Input: mtk-pmic-keys - fix possible null pointer d
CVE-2025-37973	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: fix out-of-bounds access during n
CVE-2025-37977	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: exynos: Disable ioccc if dma-coherent pr
CVE-2025-37974	In the Linux kernel, the following vulnerability has been resolved: s390/pci: Fix missing check for zpci_create_devic
CVE-2025-37975	In the Linux kernel, the following vulnerability has been resolved: riscv: module: Fix out-of-bounds relocation acces
CVE-2025-37978	In the Linux kernel, the following vulnerability has been resolved: block: integrity: Do not call set_page_dirty_lock(
CVE-2025-37979	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: Fix sc7280 lpass potential buffer ov
CVE-2025-37980	In the Linux kernel, the following vulnerability has been resolved: block: fix resource leak in blk_register_queue() e
CVE-2025-37982	In the Linux kernel, the following vulnerability has been resolved: wifi: wl1251: fix memory leak in wl1251_tx_wor
CVE-2025-37983	In the Linux kernel, the following vulnerability has been resolved: qibfs: fix _another_ leak failure to allocate inode
CVE-2025-37984	In the Linux kernel, the following vulnerability has been resolved: crypto: ecdsa - Harden against integer overflows
CVE-2025-37985	In the Linux kernel, the following vulnerability has been resolved: USB: wdm: close race between wdm_open and w
CVE-2025-37986	In the Linux kernel, the following vulnerability has been resolved: usb: typec: class: Invalidate USB device pointers
CVE-2025-37987	In the Linux kernel, the following vulnerability has been resolved: pds_core: Prevent possible adminq overflow/stuck
CVE-2025-37988	In the Linux kernel, the following vulnerability has been resolved: fix a couple of races in MNT_TREE_BENEATH
CVE-2025-37989	In the Linux kernel, the following vulnerability has been resolved: net: phy: leds: fix memory leak A network restart
CVE-2025-37990	In the Linux kernel, the following vulnerability has been resolved: wifi: brcm80211: fmac: Add error handling for br
CVE-2025-37991	In the Linux kernel, the following vulnerability has been resolved: parisc: Fix double SIGFPE crash Camm noticed t
CVE-2025-37992	In the Linux kernel, the following vulnerability has been resolved: net_sched: Flush gso_skb list too during ->chang
CVE-2025-37994	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: displayport: Fix NULL pointer a
CVE-2025-37995	In the Linux kernel, the following vulnerability has been resolved: module: ensure that kobject_put() is safe for mod
CVE-2025-37997	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: fix region locking in hash types Re
CVE-2025-37998	In the Linux kernel, the following vulnerability has been resolved: openvswitch: Fix unsafe attribute parsing in outpu
CVE-2025-38000	In the Linux kernel, the following vulnerability has been resolved: sch_hfsc: Fix qlen accounting bug when using pe
CVE-2025-38001	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Address reentrant enqueue addin
CVE-2025-38003	In the Linux kernel, the following vulnerability has been resolved: can: bcm: add missing rcu read protection for pro
CVE-2025-38004	In the Linux kernel, the following vulnerability has been resolved: can: bcm: add locking for bcm_op runtime update
CVE-2025-38005	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: k3-udma: Add missing locking Re
CVE-2025-38006	In the Linux kernel, the following vulnerability has been resolved: net: mctp: Don't access ifa_index when missing In
CVE-2025-38007	In the Linux kernel, the following vulnerability has been resolved: HID: ucllogic: Add NULL check in ucllogic_input
CVE-2025-38008	In the Linux kernel, the following vulnerability has been resolved: mm/page_alloc: fix race condition in unaccepted

CVE-2025-38009	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: disable napi on driver removal A warn
CVE-2025-38010	In the Linux kernel, the following vulnerability has been resolved: phy: tegra: xusb: Use a bitmask for UTMI pad po
CVE-2025-38011	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: csa unmap use uninterruptible lock
CVE-2025-38013	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Set n_channels after allocating s
CVE-2025-38014	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Refactor remove call with idxd
CVE-2025-38015	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: fix memory leak in error handli
CVE-2025-38017	In the Linux kernel, the following vulnerability has been resolved: fs/eventpoll: fix endless busy loop after timeout h
CVE-2025-38018	In the Linux kernel, the following vulnerability has been resolved: net/tls: fix kernel panic when alloc_page failed W
CVE-2025-38019	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_router: Fix use-after-free when
CVE-2025-38022	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix "KASAN: slab-use-after-free R
CVE-2025-38020	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Disable MACsec offload for uplink re
CVE-2025-38024	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix slab-use-after-free Read in rxe_q
CVE-2025-38023	In the Linux kernel, the following vulnerability has been resolved: nfs: handle failure of nfs_get_lock_context in unl
CVE-2025-38027	In the Linux kernel, the following vulnerability has been resolved: regulator: max20086: fix invalid memory access i
CVE-2025-38029	In the Linux kernel, the following vulnerability has been resolved: kasan: avoid sleepable page allocation from atom
CVE-2025-38031	In the Linux kernel, the following vulnerability has been resolved: padata: do not leak refcount in reorder_work A re
CVE-2025-38033	In the Linux kernel, the following vulnerability has been resolved: x86/Kconfig: make CFI_AUTO_DEFAULT depe
CVE-2025-38034	In the Linux kernel, the following vulnerability has been resolved: btrfs: correct the order of prelim_ref arguments in
CVE-2025-38035	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: don't restore null sk_state_change que
CVE-2025-38036	In the Linux kernel, the following vulnerability has been resolved: drm/xe/vf: Perform early GT MMIO initialization
CVE-2025-38037	In the Linux kernel, the following vulnerability has been resolved: vxlan: Annotate FDB data races The 'used' and 'up
CVE-2025-38038	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: Remove unnecessary driver_
CVE-2025-38039	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Avoid WARN_ON when configuring
CVE-2025-38040	In the Linux kernel, the following vulnerability has been resolved: serial: mctrl_gpio: split disable_ms into sync and
CVE-2025-38041	In the Linux kernel, the following vulnerability has been resolved: clk: sunxi-ng: h616: Reparent GPU clock during
CVE-2025-38042	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: k3-udma-glue: Drop skip_fdq argu
CVE-2025-38043	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_ffa: Set dma_mask for ffa devices
CVE-2025-38044	In the Linux kernel, the following vulnerability has been resolved: media: cx231xx: set device_caps for 417 The vid
CVE-2025-38045	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: fix debug actions order The order o
CVE-2025-38046	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2025-38047	In the Linux kernel, the following vulnerability has been resolved: x86/fred: Fix system hang during S4 resume with
CVE-2025-38048	In the Linux kernel, the following vulnerability has been resolved: virtio_ring: Fix data race by tagging event_trigger
CVE-2025-38051	In the Linux kernel, the following vulnerability has been resolved: smb: client: Fix use-after-free in cifs_fill_dirent T
CVE-2025-38052	In the Linux kernel, the following vulnerability has been resolved: net/tipc: fix slab-use-after-free Read in tipc_aead
CVE-2025-38053	In the Linux kernel, the following vulnerability has been resolved: idpf: fix null-ptr-deref in idpf_features_check idp
CVE-2025-38055	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: Fix segfault with PEBS-via-PT wi
CVE-2025-38057	In the Linux kernel, the following vulnerability has been resolved: espintcp: fix skb leaks A few error paths are miss
CVE-2025-38058	In the Linux kernel, the following vulnerability has been resolved: __legitimize_mnt(): check for MNT_SYNC_UM
CVE-2025-38059	In the Linux kernel, the following vulnerability has been resolved: btrfs: avoid NULL pointer dereference if no valid
CVE-2025-38060	In the Linux kernel, the following vulnerability has been resolved: bpf: copy_verifier_state() should copy 'loop_entr

CVE-2025-38061	In the Linux kernel, the following vulnerability has been resolved: net: pktgen: fix access outside of user given buffer
CVE-2025-38062	In the Linux kernel, the following vulnerability has been resolved: genirq/msi: Store the IOMMU IOVA directly in msi_desc
CVE-2025-38063	In the Linux kernel, the following vulnerability has been resolved: dm: fix unconditional IO throttle caused by REQ_FLUSH
CVE-2025-38064	In the Linux kernel, the following vulnerability has been resolved: virtio: break and reset virtio devices on device_shutdown
CVE-2025-38066	In the Linux kernel, the following vulnerability has been resolved: dm cache: prevent BUG_ON by blocking retries
CVE-2025-38065	In the Linux kernel, the following vulnerability has been resolved: orangesfs: Do not truncate file size 'len' is used to
CVE-2025-38067	In the Linux kernel, the following vulnerability has been resolved: rseq: Fix segfault on registration when rseq_cs is
CVE-2025-38068	In the Linux kernel, the following vulnerability has been resolved: crypto: lzo - Fix compression buffer overrun Unli
CVE-2025-38069	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: pci-epf-test: Fix double free that c
CVE-2025-38070	In the Linux kernel, the following vulnerability has been resolved: ASoC: sma1307: Add NULL check in sma1307_
CVE-2025-38072	In the Linux kernel, the following vulnerability has been resolved: libnvdimm/labels: Fix divide error in nd_label_da
CVE-2025-38071	In the Linux kernel, the following vulnerability has been resolved: x86/mm: Check return value from memblock_phy
CVE-2025-38073	In the Linux kernel, the following vulnerability has been resolved: block: fix race between set_blocksize and read pa
CVE-2025-38074	In the Linux kernel, the following vulnerability has been resolved: vhost-scsi: protect vq->log_used with vq->mutex
CVE-2025-38075	In the Linux kernel, the following vulnerability has been resolved: scsi: target: iscsi: Fix timeout on deleted connecti
CVE-2025-38078	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix race of buffer access at PCM OS
CVE-2025-38077	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-wmi-sysman: Avoid buffer ove
CVE-2025-38079	In the Linux kernel, the following vulnerability has been resolved: crypto: algif_hash - fix double free in hash_accept
CVE-2025-38080	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Increase block_sequence array
CVE-2025-38082	In the Linux kernel, the following vulnerability has been resolved: gpio: virtuser: fix potential out-of-bound write If
CVE-2025-38081	In the Linux kernel, the following vulnerability has been resolved: spi-rockchip: Fix register out of bounds access De
CVE-2025-38083	In the Linux kernel, the following vulnerability has been resolved: net_sched: prio: fix a race in prio_tune() Gerrard
CVE-2025-38084	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: unshare page tables during VMA sp
CVE-2025-38085	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: fix huge_pmd_unshare() vs GUP-fa
CVE-2025-38086	In the Linux kernel, the following vulnerability has been resolved: net: ch9200: fix uninitialised access during mii_n
CVE-2025-38087	In the Linux kernel, the following vulnerability has been resolved: net/sched: fix use-after-free in taprio_dev_notifie
CVE-2025-38088	In the Linux kernel, the following vulnerability has been resolved: powerpc/powernv/memtrace: Fix out of bounds is
CVE-2025-38089	In the Linux kernel, the following vulnerability has been resolved: sunrpc: handle SVC_GARBAGE during svc auth
CVE-2025-38090	In the Linux kernel, the following vulnerability has been resolved: drivers/rapidio/rio_cm.c: prevent possible heap ov
CVE-2025-38091	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: check stream id dml21 wrapper
CVE-2025-38092	In the Linux kernel, the following vulnerability has been resolved: ksmbd: use list_first_entry_or_null for opinfo_ge
CVE-2025-38094	In the Linux kernel, the following vulnerability has been resolved: net: cadence: macb: Fix a possible deadlock in m
CVE-2025-38095	In the Linux kernel, the following vulnerability has been resolved: dma-buf: insert memory barrier before updating n
CVE-2025-38096	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: don't warn when if there is a FW er
CVE-2025-38097	In the Linux kernel, the following vulnerability has been resolved: espintcp: remove encaps socket caching to avoid r
CVE-2025-38098	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Don't treat wb connector as phy
CVE-2025-38099	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Disable SCO support if READ_VOIC
CVE-2025-38100	In the Linux kernel, the following vulnerability has been resolved: x86/iopl: Cure TIF_IO_BITMAP inconsistencies
CVE-2025-38102	In the Linux kernel, the following vulnerability has been resolved: VMCI: fix race between vmci_host_setup_notify
CVE-2025-38101	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix buffer locking in ring_buffer_sub

CVE-2025-38103	In the Linux kernel, the following vulnerability has been resolved: HID: usbhid: Eliminate recurrent out-of-bounds b
CVE-2025-38104	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Replace Mutex with Spinlock for R
CVE-2025-38105	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Kill timer properly at removal
CVE-2025-38107	In the Linux kernel, the following vulnerability has been resolved: net_sched: ets: fix a race in ets_qdisc_change() G
CVE-2025-38108	In the Linux kernel, the following vulnerability has been resolved: net_sched: red: fix a race in __red_change() Gerra
CVE-2025-38109	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix ECVF vports unload on shutdown f
CVE-2025-38110	In the Linux kernel, the following vulnerability has been resolved: net/mdionbus: Fix potential out-of-bounds clause 4
CVE-2025-38111	In the Linux kernel, the following vulnerability has been resolved: net/mdionbus: Fix potential out-of-bounds read/wr
CVE-2025-38112	In the Linux kernel, the following vulnerability has been resolved: net: Fix TOCTOU issue in sk_is_readable() sk->s
CVE-2025-38113	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Fix NULL pointer dereference whe
CVE-2025-38115	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: fix a potential crash on gso_s
CVE-2025-38117	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Protect mgmt_pending list w
CVE-2025-38118	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix UAF on mgmt_remove_a
CVE-2025-38120	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_set_pipapo_avx2: fix initial map fill
CVE-2025-38119	In the Linux kernel, the following vulnerability has been resolved: scsi: core: ufs: Fix a hang in the error handler ufs
CVE-2025-38122	In the Linux kernel, the following vulnerability has been resolved: gve: add missing NULL check for gve_alloc_pen
CVE-2025-38123	In the Linux kernel, the following vulnerability has been resolved: net: wwan: t7xx: Fix napi rx poll issue When driv
CVE-2025-38124	In the Linux kernel, the following vulnerability has been resolved: net: fix udp gso skb_segment after pull from frag
CVE-2025-38125	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: make sure that ptp_rate is not 0 befo
CVE-2025-38126	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: make sure that ptp_rate is not 0 befo
CVE-2025-38127	In the Linux kernel, the following vulnerability has been resolved: ice: fix Tx scheduler error handling in XDP callba
CVE-2025-38129	In the Linux kernel, the following vulnerability has been resolved: page_pool: Fix use-after-free in page_pool_recyc
CVE-2025-38131	In the Linux kernel, the following vulnerability has been resolved: coresight: prevent deactivate active config while
CVE-2025-38132	In the Linux kernel, the following vulnerability has been resolved: coresight: holding cscfg_csdev_lock while remov
CVE-2025-38135	In the Linux kernel, the following vulnerability has been resolved: serial: Fix potential null-ptr-deref in mlb_usio_pr
CVE-2025-38136	In the Linux kernel, the following vulnerability has been resolved: usb: renesas_usbhs: Reorder clock handling and p
CVE-2025-38138	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: Add NULL check in udma_probe()
CVE-2025-38140	In the Linux kernel, the following vulnerability has been resolved: dm: limit swapping tables for devices with zone v
CVE-2025-38142	In the Linux kernel, the following vulnerability has been resolved: hwmon: (asus-ec-sensors) check sensor index in r
CVE-2025-38143	In the Linux kernel, the following vulnerability has been resolved: backlight: pm8941: Add NULL check in wled_co
CVE-2025-38145	In the Linux kernel, the following vulnerability has been resolved: soc: aspeed: Add NULL check in aspeed_lpc_ena
CVE-2025-38146	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: Fix the dead loop of MPLS par
CVE-2025-38147	In the Linux kernel, the following vulnerability has been resolved: calipso: Don't call calipso functions for AF_INET
CVE-2025-38148	In the Linux kernel, the following vulnerability has been resolved: net: phy: msc: Fix memory leak when using one
CVE-2025-38152	In the Linux kernel, the following vulnerability has been resolved: remoteproc: core: Clear table_sz when rproc_shut
CVE-2025-38149	In the Linux kernel, the following vulnerability has been resolved: net: phy: clear phydev->devlink when the link is
CVE-2025-38153	In the Linux kernel, the following vulnerability has been resolved: net: usb: aqc111: fix error handling of usbnet reac
CVE-2025-38154	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Avoid using sk_socket after free w
CVE-2025-38155	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: Fix null-ptr-deref in mt7915_
CVE-2025-38156	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: Fix null-ptr-deref in mt7996_

CVE-2025-38157	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k_htc: Abort software beacon handling
CVE-2025-38158	In the Linux kernel, the following vulnerability has been resolved: hisi_acc_vfio_pci: fix XQE dma address error Th
CVE-2025-38159	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: fix the 'para' buffer size to avoid read
CVE-2025-38160	In the Linux kernel, the following vulnerability has been resolved: clk: bcm: rpi: Add NULL check in raspberrypi_cl
CVE-2025-38161	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix error flow upon firmware failur
CVE-2025-38162	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: prevent overflow in look
CVE-2025-38163	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on sbi->total_valid_bl
CVE-2025-38164	In the Linux kernel, the following vulnerability has been resolved: f2fs: zone: fix to avoid inconsistence in between s
CVE-2025-38165	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix panic when calling skb_linear
CVE-2025-38166	In the Linux kernel, the following vulnerability has been resolved: bpf: fix ktls panic with sockmap [2172.936997] -
CVE-2025-38167	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: handle hdr_first_de() return value The h
CVE-2025-38168	In the Linux kernel, the following vulnerability has been resolved: perf: arm-ni: Unregister PMUs on probe failure W
CVE-2025-38169	In the Linux kernel, the following vulnerability has been resolved: arm64/fpsimd: Avoid clobbering kernel FPSIMD
CVE-2025-38170	In the Linux kernel, the following vulnerability has been resolved: arm64/fpsimd: Discard stale CPU state when han
CVE-2025-38173	In the Linux kernel, the following vulnerability has been resolved: crypto: marvell/cesa - Handle zero-length skciph
CVE-2025-38174	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Do not double dequeue a configurati
CVE-2025-38177	In the Linux kernel, the following vulnerability has been resolved: sch_hfsc: make hfsc_qlen_notify() idempotent hf
CVE-2025-38180	In the Linux kernel, the following vulnerability has been resolved: net: atm: fix /proc/net/atm/lec handling /proc/net/
CVE-2025-38181	In the Linux kernel, the following vulnerability has been resolved: calipso: Fix null-ptr-deref in calipso_req_{set,del
CVE-2025-38182	In the Linux kernel, the following vulnerability has been resolved: ublk: santizize the arguments from userspace whe
CVE-2025-38183	In the Linux kernel, the following vulnerability has been resolved: net: lan743x: fix potential out-of-bounds write in
CVE-2025-38184	In the Linux kernel, the following vulnerability has been resolved: tipc: fix null-ptr-deref when acquiring remote ip o
CVE-2025-38187	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: fix a use-after-free in r535_gsp_rpc
CVE-2025-38185	In the Linux kernel, the following vulnerability has been resolved: atm: atmtcp: Free invalid length skb in atmtcp_c
CVE-2025-38188	In the Linux kernel, the following vulnerability has been resolved: drm/msm/a7xx: Call CP_RESET_CONTEXT_ST
CVE-2025-38189	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Avoid NULL pointer dereference in `v3
CVE-2025-38190	In the Linux kernel, the following vulnerability has been resolved: atm: Revert atm_account_tx() if copy_from_iter_
CVE-2025-38192	In the Linux kernel, the following vulnerability has been resolved: net: clear the dst when changing skb protocol A n
CVE-2025-38191	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix null pointer dereference in destroy_pro
CVE-2025-38193	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: reject invalid perturb period C
CVE-2025-38194	In the Linux kernel, the following vulnerability has been resolved: jffs2: check that raw node were preallocated befo
CVE-2025-38197	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell_rbu: Fix list usage Pass the cor
CVE-2025-38198	In the Linux kernel, the following vulnerability has been resolved: fbcon: Make sure modelist not set on unregistered
CVE-2025-38199	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix memory leak due to multiple rx
CVE-2025-38200	In the Linux kernel, the following vulnerability has been resolved: i40e: fix MMIO write access to an invalid page in
CVE-2025-38201	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: clamp maximum map bu
CVE-2025-38202	In the Linux kernel, the following vulnerability has been resolved: bpf: Check rcu_read_lock_trace_held() in bpf_ma
CVE-2025-38203	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix null-ptr-deref in jfs_ioc_trim [Syzkaller E
CVE-2025-38204	In the Linux kernel, the following vulnerability has been resolved: jfs: fix array-index-out-of-bounds read in add_mi
CVE-2025-38205	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Avoid divide by zero by initiali

CVE-2025-38206	In the Linux kernel, the following vulnerability has been resolved: exfat: fix double free in delayed_free The double
CVE-2025-38207	In the Linux kernel, the following vulnerability has been resolved: mm: fix uprobe pte be overwritten when expanding
CVE-2025-38208	In the Linux kernel, the following vulnerability has been resolved: smb: client: add NULL check in automount_fullp
CVE-2025-38210	In the Linux kernel, the following vulnerability has been resolved: configs-tsm-report: Fix NULL dereference of tsr
CVE-2025-38211	In the Linux kernel, the following vulnerability has been resolved: RDMA/iwcm: Fix use-after-free of work objects
CVE-2025-38212	In the Linux kernel, the following vulnerability has been resolved: ipc: fix to protect IPCS lookups using RCU syzbo
CVE-2025-38213	In the Linux kernel, the following vulnerability has been resolved: vgacon: Add check for vc_origin address range in
CVE-2025-38214	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix fb_set_var to prevent null-ptr-deref in
CVE-2025-38217	In the Linux kernel, the following vulnerability has been resolved: hwmon: (ftstetates) Fix TOCTOU race in fts_rea
CVE-2025-38215	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix do_register_framebuffer to prevent nul
CVE-2025-38218	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on sit_bitmap_size w/
CVE-2025-38219	In the Linux kernel, the following vulnerability has been resolved: f2fs: prevent kernel warning due to negative i_nli
CVE-2025-38220	In the Linux kernel, the following vulnerability has been resolved: ext4: only dirty folios when data journaling regul
CVE-2025-38222	In the Linux kernel, the following vulnerability has been resolved: ext4: inline: fix len overflow in ext4_prepare_inli
CVE-2025-38225	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Cleanup after an allocation erro
CVE-2025-38227	In the Linux kernel, the following vulnerability has been resolved: media: vidtv: Terminating the subsequent process
CVE-2025-38226	In the Linux kernel, the following vulnerability has been resolved: media: vivid: Change the size of the composing s
CVE-2025-38229	In the Linux kernel, the following vulnerability has been resolved: media: cxusb: no longer judge rbuf when the writ
CVE-2025-38230	In the Linux kernel, the following vulnerability has been resolved: jfs: validate AG parameters in dbMount() to prev
CVE-2025-38231	In the Linux kernel, the following vulnerability has been resolved: nfsd: Initialize ssc before laundromat_work to pre
CVE-2025-38234	In the Linux kernel, the following vulnerability has been resolved: sched/rt: Fix race in push_rt_task Overview ====
CVE-2025-38232	In the Linux kernel, the following vulnerability has been resolved: NFSD: fix race between nfsd registration and exp
CVE-2025-38236	In the Linux kernel, the following vulnerability has been resolved: af_unix: Don't leave consecutive consumed OOB
CVE-2025-38237	In the Linux kernel, the following vulnerability has been resolved: media: platform: exynos4-is: Add hardware sync
CVE-2025-38239	In the Linux kernel, the following vulnerability has been resolved: scsi: megaraid_sas: Fix invalid node index On a s
CVE-2025-38242	In the Linux kernel, the following vulnerability has been resolved: mm: userfaultfd: fix race of userfaultfd_move and
CVE-2025-38240	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: dp: drm_err => dev_err in HPD pa
CVE-2025-38244	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential deadlock when reconnected
CVE-2025-38245	In the Linux kernel, the following vulnerability has been resolved: atm: Release atm_dev_mutex after removing proc
CVE-2025-38246	In the Linux kernel, the following vulnerability has been resolved: bnxt: properly flush XDP redirect lists We encour
CVE-2025-38249	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix out-of-bounds read in snd_
CVE-2025-38248	In the Linux kernel, the following vulnerability has been resolved: bridge: mcast: Fix use-after-free during router por
CVE-2025-38250	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Fix use-after-free in vhci_flu
CVE-2025-38251	In the Linux kernel, the following vulnerability has been resolved: atm: clip: prevent NULL deref in clip_push() Bla
CVE-2025-38253	In the Linux kernel, the following vulnerability has been resolved: HID: wacom: fix crash in wacom_aes_battery_ha
CVE-2025-38257	In the Linux kernel, the following vulnerability has been resolved: s390/pkey: Prevent overflow in size calculation fo
CVE-2025-38255	In the Linux kernel, the following vulnerability has been resolved: lib/group_cpus: fix NULL pointer dereference fro
CVE-2025-38258	In the Linux kernel, the following vulnerability has been resolved: mm/damon/sysfs-schemes: free old damon_sysfs
CVE-2025-38259	In the Linux kernel, the following vulnerability has been resolved: ASoC: codecs: wcd9335: Fix missing free of regu
CVE-2025-38260	In the Linux kernel, the following vulnerability has been resolved: btrfs: handle csum tree error with rescue=ibadroo

CVE-2025-38262	In the Linux kernel, the following vulnerability has been resolved: tty: serial: uartlite: register uart driver in init Whe
CVE-2025-38263	In the Linux kernel, the following vulnerability has been resolved: bcache: fix NULL pointer in cache_set_flush() 1.
CVE-2025-38261	In the Linux kernel, the following vulnerability has been resolved: riscv: save the SR_SUM status over switches Wh
CVE-2025-38265	In the Linux kernel, the following vulnerability has been resolved: serial: jsm: fix NPE during jsm_uart_port_init No
CVE-2025-38264	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: sanitize request list handling Validate
CVE-2025-38269	In the Linux kernel, the following vulnerability has been resolved: btrfs: exit after state insertion failure at btrfs_con
CVE-2025-38272	In the Linux kernel, the following vulnerability has been resolved: net: dsa: b53: do not enable EEE on bcm63xx BC
CVE-2025-38273	In the Linux kernel, the following vulnerability has been resolved: net: tipc: fix refcount warning in tipc_aead_encrypt
CVE-2025-38274	In the Linux kernel, the following vulnerability has been resolved: fpga: fix potential null pointer deref in fpga_mgr_
CVE-2025-38275	In the Linux kernel, the following vulnerability has been resolved: phy: qcom-qmp-usb: Fix an NULL vs IS_ERR() b
CVE-2025-38277	In the Linux kernel, the following vulnerability has been resolved: mtd: nand: ecc-mxic: Fix use of uninitialized vari
CVE-2025-38278	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: QOS: Refactor TC_HTB_LEAF_D
CVE-2025-38279	In the Linux kernel, the following vulnerability has been resolved: bpf: Do not include stack ptr register in precision
CVE-2025-38282	In the Linux kernel, the following vulnerability has been resolved: kernfs: Relax constraint in draining guard The act
CVE-2025-38280	In the Linux kernel, the following vulnerability has been resolved: bpf: Avoid __bpf_prog_ret0_warn when jit fails s
CVE-2025-38283	In the Linux kernel, the following vulnerability has been resolved: hisi_acc_vfio_pci: bugfix live migration function
CVE-2025-38285	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix WARN() in get_bpf_raw_tp_regs syzkal
CVE-2025-38286	In the Linux kernel, the following vulnerability has been resolved: pinctrl: at91: Fix possible out-of-boundary access
CVE-2025-38289	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Avoid potential ndlp use-after-free in d
CVE-2025-38290	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix node corruption in ar->arvifs list
CVE-2025-38292	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix invalid access to memory In ath1
CVE-2025-38293	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix node corruption in ar->arvifs list
CVE-2025-38298	In the Linux kernel, the following vulnerability has been resolved: EDAC/skx_common: Fix general protection fault
CVE-2025-38299	In the Linux kernel, the following vulnerability has been resolved: ASoC: mediatek: mt8195: Set ETDM1/2 IN/OUT
CVE-2025-38295	In the Linux kernel, the following vulnerability has been resolved: perf/amlogic: Replace smp_processor_id() with r
CVE-2025-38300	In the Linux kernel, the following vulnerability has been resolved: crypto: sun8i-ce-cipher - fix error handling in sun
CVE-2025-38303	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: eir: Fix possible crashes on eir_create_
CVE-2025-38304	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix NULL pointer deference on eir_ge
CVE-2025-38305	In the Linux kernel, the following vulnerability has been resolved: ptp: remove ptp->n_vclocks check logic in ptp_vc
CVE-2025-38307	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: avs: Verify content returned by pars
CVE-2025-38310	In the Linux kernel, the following vulnerability has been resolved: seg6: Fix validation of nexthop addresses The ker
CVE-2025-38311	In the Linux kernel, the following vulnerability has been resolved: iavf: get rid of the crit lock Get rid of the crit lock
CVE-2025-38312	In the Linux kernel, the following vulnerability has been resolved: fbdev: core: fbcvt: avoid division by 0 in fb_cvt_
CVE-2025-38313	In the Linux kernel, the following vulnerability has been resolved: bus: fsl-mc: fix double-free on mc_dev The blame
CVE-2025-38319	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pp: Fix potential NULL pointer derefere
CVE-2025-38320	In the Linux kernel, the following vulnerability has been resolved: arm64/ptrace: Fix stack-out-of-bounds read in reg
CVE-2025-38323	In the Linux kernel, the following vulnerability has been resolved: net: atm: add lec_mutex syzbot found its way in n
CVE-2025-38321	In the Linux kernel, the following vulnerability has been resolved: smb: Log an error when close_all_cached_dirs fa
CVE-2025-38324	In the Linux kernel, the following vulnerability has been resolved: mpls: Use rcu_dereference_rtnl() in mpls_route_i
CVE-2025-38326	In the Linux kernel, the following vulnerability has been resolved: aoe: clean device rq_list in aoedev_downdev() Ar

CVE-2025-38328	In the Linux kernel, the following vulnerability has been resolved: jffs2: check jffs2_prealloc_raw_node_refs() result
CVE-2025-38329	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix OOB memory read access
CVE-2025-38330	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix OOB memory read access
CVE-2025-38331	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: cortina: Use TOE/TSO on all TCP P
CVE-2025-38332	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Use memcpy() for BIOS version The st
CVE-2025-38333	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to bail out in get_new_segment() -----
CVE-2025-38335	In the Linux kernel, the following vulnerability has been resolved: Input: gpio-keys - fix a sleep while atomic with P
CVE-2025-38336	In the Linux kernel, the following vulnerability has been resolved: ata: pata_via: Force PIO for ATAPI devices on V
CVE-2025-38334	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Prevent attempts to reclaim poisoned pa
CVE-2025-38337	In the Linux kernel, the following vulnerability has been resolved: jbd2: fix data-race and null-ptr-deref in jbd2_jour
CVE-2025-38338	In the Linux kernel, the following vulnerability has been resolved: fs/nfs/read: fix double-unlock bug in nfs_return_e
CVE-2025-38340	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix OOB memory read access
CVE-2025-38343	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: drop fragments with multicas
CVE-2025-38342	In the Linux kernel, the following vulnerability has been resolved: software node: Correct a OOB check in software_
CVE-2025-38344	In the Linux kernel, the following vulnerability has been resolved: ACPICA: fix acpi parse and parseext cache leaks
CVE-2025-38345	In the Linux kernel, the following vulnerability has been resolved: ACPICA: fix acpi operand cache leak in dswstate
CVE-2025-38346	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix UAF when lookup kallsym after ftrace
CVE-2025-38347	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on ino and xnid syzbo
CVE-2025-38348	In the Linux kernel, the following vulnerability has been resolved: wifi: p54: prevent buffer-overflow in p54_rx_eepr
CVE-2025-38575	In the Linux kernel, the following vulnerability has been resolved: ksmbd: use aead_request_free to match aead_req
CVE-2025-38637	In the Linux kernel, the following vulnerability has been resolved: net_sched: skbprio: Remove overly strict queue a
CVE-2025-3933	A Regular Expression Denial of Service (ReDoS) vulnerability was discovered in the Hugging Face Transformers lib
CVE-2025-39728	In the Linux kernel, the following vulnerability has been resolved: clk: samsung: Fix UBSAN panic in samsung_clk
CVE-2025-39735	In the Linux kernel, the following vulnerability has been resolved: jfs: fix slab-out-of-bounds read in ea_get() During
CVE-2025-39989	In the Linux kernel, the following vulnerability has been resolved: x86/mce: use is_copy_from_user() to determine c
CVE-2025-40014	In the Linux kernel, the following vulnerability has been resolved: objtool, spi: amd: Fix out-of-bounds stack access
CVE-2025-40114	In the Linux kernel, the following vulnerability has been resolved: iio: light: Add check for array bounds in veml607
CVE-2025-40325	In the Linux kernel, the following vulnerability has been resolved: md/raid10: wait barrier before returning discard r
CVE-2025-4035	A flaw was found in libsoup. When handling cookies, libsoup clients mistakenly allow cookies to be set for public st
CVE-2025-40364	In the Linux kernel, the following vulnerability has been resolved: io_uring: fix io_req_prep_async with provided bu
CVE-2025-4050	Out of bounds memory access in DevTools in Google Chrome prior to 136.0.7103.59 allowed a remote attacker who co
CVE-2025-4051	Insufficient data validation in DevTools in Google Chrome prior to 136.0.7103.59 allowed a remote attacker who co
CVE-2025-4052	Inappropriate implementation in DevTools in Google Chrome prior to 136.0.7103.59 allowed a remote attacker who
CVE-2025-40909	Perl threads have a working directory race condition where file operations may target unintended paths. If a directory
CVE-2025-4096	Heap buffer overflow in HTML in Google Chrome prior to 136.0.7103.59 allowed a remote attacker to potentially e
CVE-2025-41234	Description In Spring Framework, versions 6.0.x as of 6.0.5, versions 6.1.x and 6.2.x, an application is vulnerable to
CVE-2025-4138	Allows the extraction filter to be ignored, allowing symlink targets to point outside the destination directory, and the
CVE-2025-4207	Buffer over-read in PostgreSQL GB18030 encoding validation allows a database input provider to achieve temporar
CVE-2025-4330	Allows the extraction filter to be ignored, allowing symlink targets to point outside the destination directory, and the
CVE-2025-4372	Use after free in WebAudio in Google Chrome prior to 136.0.7103.92 allowed a remote attacker to potentially explo

CVE-2025-4373	A flaw was found in GLib, which is vulnerable to an integer overflow in the <code>g_string_insert_unichar()</code> function. When
CVE-2025-43857	Net::IMAP implements Internet Message Access Protocol (IMAP) client functionality in Ruby. Prior to versions 0.5
CVE-2025-43859	h11 is a Python implementation of HTTP/1.1. Prior to version 0.16.0, a leniency in h11's parsing of line terminators
CVE-2025-4435	When using a TarFile.errorlevel = 0 and extracting with a filter the documented behavior is that any filtered member
CVE-2025-4476	A denial-of-service vulnerability has been identified in the libsoup HTTP client library. This flaw can be triggered w
CVE-2025-4516	There is an issue in CPython when using <code>`bytes.decode("unicode_escape", error="ignore replace")`</code> . If you are not us
CVE-2025-4517	Allows arbitrary filesystem writes outside the extraction directory during extraction with filter="data". You are affect
CVE-2025-45582	GNU Tar through 1.35 allows file overwrite via directory traversal in crafted TAR archives, with a certain two-step p
CVE-2025-4563	A vulnerability exists in the NodeRestriction admission controller where nodes can bypass dynamic resource allocati
CVE-2025-4565	Any project that uses Protobuf Pure-Python backend to parse untrusted Protocol Buffers data containing an arbitrary
CVE-2025-4598	A vulnerability was found in systemd-coredump. This flaw allows an attacker to force a SUID process to crash and r
CVE-2025-4575	Issue summary: Use of -adreject option with the openssl x509 application adds a trusted use instead of a rejected us
CVE-2025-46392	Uncontrolled Resource Consumption vulnerability in Apache Commons Configuration 1.x. There are a number of is
CVE-2025-46570	vLLM is an inference and serving engine for large language models (LLMs). Prior to version 0.9.0, when a new prom
CVE-2025-46560	vLLM is a high-throughput and memory-efficient inference and serving engine for LLMs. Versions starting from 0.8
CVE-2025-4664	Insufficient policy enforcement in Loader in Google Chrome prior to 136.0.7103.113 allowed a remote attacker to le
CVE-2025-46701	Improper Handling of Case Sensitivity vulnerability in Apache Tomcat's GCI servlet allows security constraint bypa
CVE-2025-46722	vLLM is an inference and serving engine for large language models (LLMs). In versions starting from 0.7.0 to befor
CVE-2025-46727	Rack is a modular Ruby web server interface. Prior to versions 2.2.14, 3.0.16, and 3.1.14, <code>`Rack::QueryParser`</code> parse
CVE-2025-4673	Proxy-Authorization and Proxy-Authenticate headers persisted on cross-origin redirects potentially leaking sensitive
CVE-2025-46835	Git GUI allows you to use the Git source control management tools via a GUI. When a user clones an untrusted repo
CVE-2025-46762	Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to exe
CVE-2025-47273	setuptools is a package that allows users to download, build, install, upgrade, and uninstall Python packages. A path
CVE-2025-47277	vLLM, an inference and serving engine for large language models (LLMs), has an issue in versions 0.6.5 through 0.8
CVE-2025-47287	Tornado is a Python web framework and asynchronous networking library. When Tornado's <code>`multipart/form-data`</code>
CVE-2025-47291	containerd is an open-source container runtime. A bug was found in the containerd's CRI implementation where con
CVE-2025-4802	Untrusted LD_LIBRARY_PATH environment variable vulnerability in the GNU C Library version 2.27 to 2.38 allo
CVE-2025-48060	jq is a command-line JSON processor. In versions up to and including 1.7.1, a heap-buffer-overflow is present in fun
CVE-2025-48371	OpenFGA is an authorization/permission engine. OpenFGA versions 1.8.0 through 1.8.12 (corresponding to Helm ch
CVE-2025-48379	Pillow is a Python imaging library. In versions 11.2.0 to before 11.3.0, there is a heap buffer overflow when writing
CVE-2025-48384	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high
CVE-2025-48385	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high
CVE-2025-48387	tar-fs provides filesystem bindings for tar-stream. Versions prior to 3.0.9, 2.1.3, and 1.16.5 have an issue where an ex
CVE-2025-48386	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high
CVE-2025-48432	An issue was discovered in Django 5.2 before 5.2.3, 5.1 before 5.1.11, and 4.2 before 4.2.23. Internal HTTP respons
CVE-2025-48734	Improper Access Control vulnerability in Apache Commons. A special BeanIntrospector class was added in version
CVE-2025-4878	A vulnerability was found in libssh, where an uninitialized variable exists under certain conditions in the privatekey.
CVE-2025-48887	vLLM, an inference and serving engine for large language models (LLMs), has a Regular Expression Denial of Serv
CVE-2025-48924	Uncontrolled Recursion vulnerability in Apache Commons Lang. This issue affects Apache Commons Lang: Startin
CVE-2025-48942	vLLM is an inference and serving engine for large language models (LLMs). In versions 0.8.0 up to but excluding 0.

CVE-2025-48943	vLLM is an inference and serving engine for large language models (LLMs). Version 0.8.0 up to but excluding 0.9.0
CVE-2025-48944	vLLM is an inference and serving engine for large language models (LLMs). In version 0.8.0 up to but excluding 0.9.0
CVE-2025-48976	Allocation of resources for multipart headers with insufficient limits enabled a DoS vulnerability in Apache Commo
CVE-2025-48988	Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat. This issue affects Apache To
CVE-2025-49007	Rack is a modular Ruby web server interface. Starting in version 3.1.0 and prior to version 3.1.16, there is a denial o
CVE-2025-49125	Authentication Bypass Using an Alternate Path or Channel vulnerability in Apache Tomcat. When using PreResourc
CVE-2025-49128	Jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson D
CVE-2025-4945	A flaw was found in the cookie parsing logic of the libsoup HTTP library, used in GNOME applications and other so
CVE-2025-4947	libcurl accidentally skips the certificate verification for QUIC connections when connecting to a host specified as an
CVE-2025-49630	In certain proxy configurations, a denial of service attack against Apache HTTP Server versions 2.4.26 through to 2.
CVE-2025-4969	A vulnerability was found in the libsoup package. This flaw stems from its failure to correctly verify the termination
CVE-2025-49795	A NULL pointer dereference vulnerability was found in libxml2 when processing XPath XML expressions. This flaw
CVE-2025-49794	A use-after-free vulnerability was found in libxml2. This issue occurs when parsing XPath elements under certain ci
CVE-2025-49796	A vulnerability was found in libxml2. Processing certain sch:name elements from the input XML file can trigger a m
CVE-2025-49812	In some mod_ssl configurations on Apache HTTP Server versions through to 2.4.63, an HTTP desynchronisation att
CVE-2025-50059	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2025-50106	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Orac
CVE-2025-50181	urllib3 is a user-friendly HTTP client library for Python. Prior to 2.5.0, it is possible to disable redirects for all requ
CVE-2025-50182	urllib3 is a user-friendly HTTP client library for Python. Starting in version 2.2.0 and prior to 2.5.0, urllib3 does not
CVE-2025-50213	Failure to Sanitize Special Elements into a Different Plane (Special Element Injection) vulnerability in Apache Airfl
CVE-2025-5063	Use after free in Compositing in Google Chrome prior to 137.0.7151.55 allowed a remote attacker to potentially exp
CVE-2025-5064	Inappropriate implementation in Background Fetch API in Google Chrome prior to 137.0.7151.55 allowed a remote
CVE-2025-5065	Inappropriate implementation in FileSystemAccess API in Google Chrome prior to 137.0.7151.55 allowed a remote
CVE-2025-5066	Inappropriate implementation in Messages in Google Chrome on Android prior to 137.0.7151.55 allowed a remote a
CVE-2025-5067	Inappropriate implementation in Tab Strip in Google Chrome prior to 137.0.7151.55 allowed a remote attacker to pe
CVE-2025-5068	Use after free in Blink in Google Chrome prior to 137.0.7151.68 allowed a remote attacker to potentially exploit hea
CVE-2025-5222	A stack buffer overflow was found in Internationl components for unicode (ICU). While running the genrb binary, t
CVE-2025-5245	A vulnerability classified as critical has been found in GNU Binutils up to 2.44. This affects the function debug_type
CVE-2025-5278	A flaw was found in GNU Coreutils. The sort utility's begfield() function is vulnerable to a heap buffer under-read. T
CVE-2025-5279	When the Amazon Redshift Python Connector is configured with the BrowserAzureOAuth2CredentialsProvider plug
CVE-2025-5280	Out of bounds write in V8 in Google Chrome prior to 137.0.7151.55 allowed a remote attacker to potentially exploit
CVE-2025-5281	Inappropriate implementation in BFCache in Google Chrome prior to 137.0.7151.55 allowed a remote attacker to po
CVE-2025-5283	Use after free in libvpx in Google Chrome prior to 137.0.7151.55 allowed a remote attacker to potentially exploit he
CVE-2025-52968	xdg-open in xdg-utils through 1.2.1 can send requests containing SameSite=Strict cookies, which can facilitate CSRF
CVE-2025-52999	jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson D
CVE-2025-53020	Late Release of Memory after Effective Lifetime vulnerability in Apache HTTP Server. This issue affects Apache H
CVE-2025-5318	A flaw was found in the libssh library. An out-of-bounds read can be triggered in the sftp_handle function due to an
CVE-2025-5351	A flaw was found in the key export functionality of libssh. The issue occurs in the internal function responsible for c
CVE-2025-53547	Helm is a package manager for Charts for Kubernetes. Prior to 3.18.4, a specially crafted Chart.yaml file along with
CVE-2025-53643	AIOHTTP is an asynchronous HTTP client/server framework for asyncio and Python. Prior to version 3.12.14, the P

CVE-2025-5372	A flaw was found in libssh versions built with OpenSSL versions older than 3.0, specifically in the <code>ssh_kdf()</code> function.
CVE-2025-53864	Connect2id Nimbus JOSE + JWT before 10.0.2 allows a remote attacker to cause a denial of service via a deeply nested object.
CVE-2025-53905	Vim is an open source, command line text editor. Prior to version 9.1.1552, a path traversal issue in Vim's <code>tar.vim</code> plugin.
CVE-2025-53906	Vim is an open source, command line text editor. Prior to version 9.1.1551, a path traversal issue in Vim's <code>zip.vim</code> plugin.
CVE-2025-5419	Out of bounds read and write in V8 in Google Chrome prior to 137.0.7151.68 allowed a remote attacker to potentially execute arbitrary code.
CVE-2025-5455	An issue was found in the private API function <code>qDecodeDataUrl()</code> in QtCore, which is used in <code>QTextDocument</code> and <code>QTextStream</code> .
CVE-2025-5683	When loading a specifically crafted ICNS format image file in QImage then it will trigger a crash. This issue affects Qt versions 5.15.0 through 6.4.0.
CVE-2025-5702	The <code>stricmp</code> implementation optimized for the Power10 processor in the GNU C Library version 2.39 and later writes to memory.
CVE-2025-5889	A vulnerability was found in juliangruber brace-expansion up to 1.1.11/2.0.1/3.0.0/4.0.0. It has been rated as problematic.
CVE-2025-5914	A vulnerability has been identified in the libarchive library, specifically within the <code>archive_read_format_rar_seek_data</code> function.
CVE-2025-5915	A vulnerability has been identified in the libarchive library. This flaw can lead to a heap buffer over-read due to the <code>archive_read_data_block</code> function.
CVE-2025-5916	A vulnerability has been identified in the libarchive library. This flaw involves an integer overflow that can be triggered by a specially crafted archive.
CVE-2025-5917	A vulnerability has been identified in the libarchive library. This flaw involves an 'off-by-one' miscalculation when handling archive data.
CVE-2025-5918	A vulnerability has been identified in the libarchive library. This flaw can be triggered when file streams are piped into the <code>archive_read_data_block</code> function.
CVE-2025-5958	Use after free in Media in Google Chrome prior to 137.0.7151.103 allowed a remote attacker to potentially exploit heap memory.
CVE-2025-5959	Type Confusion in V8 in Google Chrome prior to 137.0.7151.103 allowed a remote attacker to execute arbitrary code.
CVE-2025-5987	A flaw was found in libssh when using the ChaCha20 cipher with the OpenSSL library. If an attacker manages to exploit this, they could potentially execute arbitrary code.
CVE-2025-6020	A flaw was found in linux-pam. The module <code>pam_namespace</code> may use access user-controlled paths without proper permissions.
CVE-2025-6021	A flaw was found in libxml2's <code>xmlBuildQName</code> function, where integer overflows in buffer size calculations can lead to a denial of service.
CVE-2025-6052	A flaw was found in how GLib's GString manages memory when adding data to strings. If a string is already very large, it can lead to a denial of service.
CVE-2025-6069	The <code>html.parser.HTMLParser</code> class had worse-case quadratic complexity when processing certain crafted malformed HTML documents.
CVE-2025-6170	A flaw was found in the interactive shell of the <code>xmlint</code> command-line tool, used for parsing XML files. When a user enters a command, it can lead to a denial of service.
CVE-2025-6199	A flaw was found in the GIF parser of Gdlib's LZW decoder. When an invalid symbol is encountered during decoding, it can lead to a denial of service.
CVE-2025-6297	It was discovered that <code>dpkg-deb</code> does not properly sanitize directory permissions when extracting a control member from a .deb file.
CVE-2025-6395	A NULL pointer dereference flaw was found in the GnuTLS software in <code>_gnutls_figure_common_ciphersuite()</code> .
CVE-2025-6442	Ruby WEBrick <code>read_header</code> HTTP Request Smuggling Vulnerability. This vulnerability allows remote attackers to smuggle HTTP requests.
CVE-2025-6965	There exists a vulnerability in SQLite versions before 3.50.2 where the number of aggregate terms could exceed the stack size.
CVE-2025-7339	on-headers is a node.js middleware for listening to when a response writes headers. A bug in on-headers versions < 2.0.0 can lead to a denial of service.
CVE-2025-7345	A flaw exists in <code>gdk#pixbuf</code> within the <code>gdk_pixbuf__jpeg_image_load_increment</code> function (<code>io-jpeg.c</code>) and in <code>glib's glib</code> .
CVE-2025-7424	A flaw was found in the libxslt library. The same memory field, <code>psvi</code> , is used for both stylesheet and input data, which can lead to a denial of service.
CVE-2025-7425	A flaw was found in libxslt where the attribute type, <code>atype</code> , flags are modified in a way that corrupts internal memory.
CVE-2025-7545	A vulnerability classified as problematic was found in GNU Binutils 2.45. Affected by this vulnerability is the <code>binutils</code> function.
CVE-2025-7546	A vulnerability, which was classified as problematic, has been found in GNU Binutils 2.45. Affected by this issue is the <code>binutils</code> function.
DLA-3112-1	bzip2 - bugfix update
DLA-3085-1	curl - security update
DLA-3134-1	tzdata - new timezone database
DLA-3161-1	tzdata - new timezone database
DLA-3263-1	libtasn1-6 - security update
DLA-3288-1	curl - security update
DLA-3325-1	openssl - security update

DLA-3341-1	curl - security update
DLA-3366-1	tzdata - new timezone database
DLA-3398-1	curl - security update
DLA-3412-1	tzdata - new timezone database
DLA-3449-1	openssl - security update
DLA-3613-1	curl - security update
DLA-3482-1	debian-archive-keyring - security update
DLA-3530-1	openssl - security update
DLA-3684-1	tzdata - new timezone database
DLA-3692-1	curl - security update
DLA-3755-1	tar - security update
DLA-3782-1	util-linux - security update
DLA-3763-1	curl - security update
DLA-3788-1	tzdata - new timezone database
DLA-3942-1	openssl - security update
DLA-4061-1	libtasn1-6 - security update
DLA-3972-1	tzdata - new timezone database
DLA-4085-1	tzdata - new timezone database
DLA-4105-1	tzdata - new timezone database
DLA-4176-1	openssl - security update
DSA-4855-1	openssl - security update
DSA-4881-1	curl - security update
DSA-4875-1	openssl - security update
DSA-5103-1	openssl - security update
DSA-4963-1	openssl - security update
DSA-5139-1	openssl - security update
DSA-5147-1	dpkg - security update
DSA-5122-1	gzip - security update
DSA-5169-1	openssl - security update
DSA-5417-1	openssl - security update
GHSA-2c7c-3mj9-8fqh	The go-jose package is subject to a "billion hashes attack" causing denial-of-service when decrypting JWE inputs. The
GHSA-2x5j-vhc8-9cwm	Impact The CIRCL implementation of FourQ fails to validate user-supplied low-order points during Diffie-Hellman
GHSA-4fx9-vc88-q2xc	JpegImagePlugin may append an EOF marker to the end of a truncated file, so that the last segment of the data will s
GHSA-5w6v-399v-w3cc	Summary Nokogiri v1.18.8 upgrades its dependency libxml2 to v2.13.8. libxml2 v2.13.8 addresses: CVE-2025-3241
GHSA-58qw-p7qm-5rvh	From the reporter XmlParser is vulnerable to XML external entity (XXE) vulnerability. XmlParser is being used wh
GHSA-6xv5-86q9-7xr8	Impact For Windows users of github.com/cyphar/filepath-securejoin, until v0.2.4 it was possible for certain rootfs an
GHSA-fv92-fjc5-jj9h	Summary Use of this library in a security-critical context may result in leaking sensitive information, if used to proc
GHSA-ggpf-24jw-3fcw	Description GHSA-rh4j-5rhw-hr54 reported a vulnerability where loading a malicious model could result in code ex
GHSA-h4gh-qq45-vh27	pyca/cryptography's wheels include a statically linked copy of OpenSSL. The versions of OpenSSL included in cryp
GHSA-hf3c-wxg2-49q9	Impact This report is to highlight a vulnerability in XGrammar, a library used by the structured output feature in vLL

GHSA-jq35-85cj-fj4p	Intel's RAPL (Running Average Power Limit) feature, introduced by the Sandy Bridge microarchitecture, provides s
GHSA-j828-28rj-hfhp	Summary A recent review identified several regular expressions in the vllm codebase that are susceptible to Regular
GHSA-m425-mq94-257g	Impact In affected releases of gRPC-Go, it is possible for an attacker to send HTTP/2 requests, cancel them, and sen
GHSA-mrxw-mxhj-p664	Summary Nokogiri v1.18.4 upgrades its dependency libxslt to v1.1.43. libxslt v1.1.43 resolves: CVE-2025-24855: F
GHSA-r4pg-vg54-wxx4	Impact cert-manager packages which call the standard library pem.Decode() function can take a long time to process
GHSA-r95h-9x8f-r3f7	Summary Nokogiri v1.16.5 upgrades its dependency libxml2 to 2.12.7 from 2.12.6. libxml2 v2.12.7 addresses CVE-
GHSA-vvfq-8hwr-qm4m	Summary Nokogiri v1.18.3 upgrades its dependency libxml2 to v2.13.6. libxml2 v2.13.6 addresses: CVE-2025-2492
GHSA-xmmx-7jpf-fx42	Impact In the OCI Distribution Specification version 1.0.0 and prior and in the OCI Image Specification version 1.0.
GHSA-xpw8-rcwv-8f8p	A client might overload the server by issue frequent RST frames. This can cause a massive amount of load on the re
RHBA-2022:2065	Description For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.6 Release Notes
RHSA-2022:1065	Description OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TL
RHSA-2022:1537	Description The gzip packages contain the gzip (GNU zip) data compression utility. gzip is used to compress regul
RHBA-2024:6680	Description Network Security Services (NSS) is a set of libraries designed to support the cross-platform developmen
RHBA-2024:5736	Description The ca-certificates package contains a set of Certificate Authority (CA) certificates chosen by the Mozil
RHSA-2022:1891	Description The libpq package provides the PostgreSQL client library, which allows client programs to connect to P
RHSA-2022:1642	Description The zlib packages provide a general-purpose lossless data compression library that is used by many diff
RHSA-2022:2201	Description The rsync utility enables the users to copy and synchronize files locally or across a network. Synchroniz
RHSA-2022:2031	Description libssh is a library which implements the SSH protocol. It can be used to implement client and server app
RHSA-2022:4807	Description PostgreSQL is an advanced object-relational database management system (DBMS).The following pack
RHSA-2022:4991	Description XZ Utils is an integrated collection of user-space file compression utilities based on the Lempel-Ziv-Ma
RHSA-2022:5311	Description The libgcrypt library provides general-purpose implementations of various cryptographic algorithms.Sec
RHSA-2022:5313	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using
RHSA-2022:5314	Description Expat is a C library for parsing XML documents.Security Fix(es): expat: stack exhaustion in doctype par
RHSA-2022:5317	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.S
RHSA-2022:5818	Description OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TL
RHSA-2022:5809	Description The pcre2 package contains a new generation of the Perl Compatible Regular Expression libraries for im
RHSA-2022:6159	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using
RHSA-2022:6180	Description The rsync utility enables the users to copy and synchronize files locally or across a network. Synchroniz
RHSA-2022:6206	Description The systemd packages contain systemd, a system and service manager for Linux, compatible with the Sys
RHSA-2022:6463	Description The GNU Privacy Guard (GnuPG or GPG) is a tool for encrypting data and creating digital signatures, c
RHSA-2022:6878	Description Expat is a C library for parsing XML documents.Security Fix(es): expat: a use-after-free in the doConter
RHSA-2022:7089	Description KSBA (pronounced Kasbah) is a library to make X.509 certificates as well as the CMS easily accessible
RHSA-2022:7105	Description The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cr
RHSA-2022:7106	Description The zlib packages provide a general-purpose lossless data compression library that is used by many diff
RHSA-2022:7128	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): po
RHSA-2022:7704	Description WebKitGTK is the port of the portable web rendering engine WebKit to the GTK platform.GLib provid
RHSA-2022:7715	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.S
RHSA-2022:7793	Description The rsync utility enables the users to copy and synchronize files locally or across a network. Synchroniz
RHSA-2023:0096	Description D-Bus is a system for sending messages between applications. It is used both for the system-wide messa
RHSA-2023:0116	Description A library that provides Abstract Syntax Notation One (ASN.1, as specified by the X.680 ITU-T recomm

RHSA-2023:0100	Description The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV init system.
RHSA-2023:0103	Description Expat is a C library for parsing XML documents.Security Fix(es): expat: use-after free caused by overread
RHSA-2023:0173	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.Security Fix(es): libxml2: heap overflow in xmlSave
RHSA-2023:0625	Description KSBA (pronounced Kasbah) is a library to make X.509 certificates as well as the CMS easily accessible.
RHSA-2023:0837	Description The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV init system.
RHSA-2023:1140	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols.
RHSA-2023:0842	Description The GNU tar program can save multiple files in an archive and restore files from an archive.Security Fix(es): tar: buffer overflow in tar_read
RHSA-2023:1405	Description OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.
RHSA-2023:1569	Description The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic protocols.
RHSA-2023:2963	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols.
RHSA-2023:3018	Description The libarchive programming library can create and read several different streaming archive formats, including tar, zip, bzip2, and xz.
RHSA-2023:3106	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols.
RHSA-2023:3837	Description The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV init system.
RHSA-2023:3839	Description libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2023:4524	Description Libcap is a library for getting and setting POSIX.1e (formerly POSIX 6) draft 15 capabilities.Security Fix(es): libcap: buffer overflow in cap_get_proc
RHSA-2023:4498	Description D-Bus is a system for sending messages between applications. It is used both for the system-wide message bus and for inter-process communication.
RHSA-2023:4529	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.Security Fix(es): libxml2: heap overflow in xmlSave
RHSA-2023:4535	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): postgresql: buffer overflow in postgresql
RHSA-2023:4523	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols.
RHSA-2023:4539	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): postgresql: buffer overflow in postgresql
RHSA-2023:4706	Description The subscription-manager packages provide programs and libraries to allow users to manage subscription information.
RHSA-2023:5252	Description The dmidecode packages provide utilities for extracting Intel 64 and Intel Itanium hardware information.
RHSA-2023:5455	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard C++ libraries (libstdc++), and other standard libraries.
RHSA-2023:7016	Description The libpq package provides the PostgreSQL client library, which allows client programs to connect to PostgreSQL databases.
RHSA-2023:6944	Description The protobuf-c packages provide C bindings for Google's Protocol Buffers.Security Fix(es): protobuf-c: buffer overflow in protobuf-c
RHSA-2023:5249	Description The ncurses (new curses) library routines are a terminal-independent method of updating character screen content.
RHSA-2023:7112	Description The shadow-utils packages include programs for converting UNIX password files to the shadow password format.
RHSA-2023:7166	Description The tpm2-tss packages provide the Intel implementation of the Trusted Platform Module (TPM) 2.0 System Software Stack.
RHSA-2023:7174	Description HTTP::Tiny is a small and simple HTTP/1.1 client written in Perl.Security Fix(es): http-tiny: insecure TLS connection
RHSA-2023:7187	Description The procps-ng packages contain a set of system utilities that provide system information, including ps, free, and top.
RHSA-2023:7790	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): postgresql: buffer overflow in postgresql
RHSA-2023:7714	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): postgresql: buffer overflow in postgresql
RHSA-2023:7877	Description OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.
RHSA-2024:0155	Description The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic protocols.
RHSA-2024:0627	Description The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic protocols.
RHSA-2024:0628	Description libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2024:0119	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.Security Fix(es): libxml2: heap overflow in xmlSave
RHSA-2024:0647	Description The RPM Package Manager (RPM) is a command-line driven package management system capable of installing, upgrading, querying, and erasing packages.
RHSA-2024:0768	Description The libmaxminddb package contains the MaxMind DB library.Security Fix(es): libmaxminddb: improper memory management
RHSA-2024:0956	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): postgresql: buffer overflow in postgresql

RHSA-2024:0974	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): po
RHSA-2024:0786	Description Network Security Services (NSS) is a set of libraries designed to support the cross-platform development
RHSA-2024:10379	Description Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the
RHSA-2024:10785	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): po
RHSA-2024:1601	Description The curl packages provide the libcurl library and the curl utility for downloading files from servers using
RHSA-2024:1610	Description The "less" utility is a text file browser that resembles "more", but allows users to move backwards in the
RHSA-2024:1615	Description Expat is a C library for parsing XML documents.Security Fix(es): expat: parsing large tokens can trigger
RHSA-2024:1784	Description The gnutls package provide the GNU Transport Layer Security (GnuTLS) library,which implements cry
RHSA-2024:2722	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard
RHSA-2024:3214	Description The gmp packages contain GNU MP, a library for arbitrary precision arithmetics, signed integers operat
RHSA-2024:3203	Description The systemd packages contain systemd, a system and service manager for Linux, compatible with the Sy
RHSA-2024:3163	Description Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the
RHSA-2024:3233	Description libssh is a library which implements the SSH protocol. It can be used to implement client and server app
RHSA-2024:3344	Description The glibc packages provide the standard C libraries (libc), POSIX threadlibraries (libpthread), standard
RHSA-2024:4197	Description The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.S
RHSA-2024:3626	Description The libxml2 library is a development toolbox providing the implementation ofvarious XML standards.S
RHSA-2024:4247	Description libuv is a multi-platform support library with a focus on asynchronous I/O. Security Fix(es): libuv: Impr
RHSA-2024:4256	Description The "less" utility is a text file browser that resembles "more", but allows users to move backwards in the
RHSA-2024:4720	Description The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.S
RHSA-2024:5079	Description The libtiff packages contain a library of functions for manipulating Tagged Image File Format (TIFF) fi
RHSA-2024:4264	Description OpenLDAP is an open-source suite of Lightweight Directory Access Protocol (LDAP) applications and
RHSA-2024:5193	Description The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.S
RHSA-2024:5299	Description The wget packages provide the GNU Wget file retrieval utility for HTTP, HTTPS, and FTP protocols.Se
RHSA-2024:5654	## The curl packages provide the libcurl library and the curl utility for downloading files from servers using various
RHSA-2024:5814	Description Node.js is a software development platform for building fast and scalable network applications in the Jay
RHSA-2024:6000	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): po
RHSA-2024:5962	Description Python is an interpreted, interactive, object-oriented programming language, which includes modules, cl
RHSA-2024:6989	Description Expat is a C library for parsing XML documents.Security Fix(es): libexpat: Negative Length Parsing Vu
RHSA-2024:7848	Description OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TL
RHSA-2024:8117	Description The java-1.8.0-openjdk packages provide the OpenJDK 8 Java Runtime Environment and the OpenJDK
RHSA-2024:8359	Description Python is an interpreted, interactive, object-oriented programming language, which includes modules, cl
RHSA-2024:8833	Description The libtiff packages contain a library of functions for manipulating Tagged Image File Format (TIFF) fi
RHSA-2024:8922	Description The bzip2 packages contain a freely available, high-quality data compressor. It provides both standalone
RHSA-2024:9502	Description Expat is a C library for parsing XML documents.Security Fix(es): libexpat: expat: DoS via XML_Resum
RHSA-2025:0325	Description The rsync utility enables the users to copy and synchronize files locally or across a network. Synchroniz
RHSA-2024:9689	Description The binutils packages provide a collection of binary utilities for the manipulation of object code in vario
RHSA-2025:10027	Description Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the
RHSA-2025:10698	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.S
RHSA-2025:11327	Description GLib provides the core application building blocks for libraries and applications written in C. It provides
RHSA-2025:1351	Description Node.js is a software development platform for building fast and scalable network applications in the Jay

RHSA-2025:1301	Description The gcc packages provide compilers for C, C++, Java, Fortran, Objective C, and Ada 95 GNU, as well as
RHSA-2025:1517	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.S
RHSA-2025:1737	Description The libpq package provides the PostgreSQL client library, which allows client programs to connect to P
RHSA-2025:2600	Description The rsync utility enables the users to copy and synchronize files locally or across a network. Synchroniz
RHSA-2025:2686	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.S
RHSA-2025:3421	Description FreeType is a free, high-quality, portable font engine that can open and manage font files. FreeType load
RHSA-2025:3082	Description PostgreSQL is an advanced object-relational database management system (DBMS).Security Fix(es): po
RHSA-2025:3615	Description libxslt is a library for transforming XML files into other textual formats (including HTML, plain text, an
RHSA-2025:3828	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard
RHSA-2025:3913	Description Expat is a C library for parsing XML documents.Security Fix(es): libexpat: expat: Improper Restriction
RHSA-2025:3845	Description The java-1.8.0-openjdk packages provide the OpenJDK 8 Java Runtime Environment and the OpenJDK
RHSA-2025:4049	Description A library that provides Abstract Syntax Notation One (ASN.1, as specified by the X.680 ITU-T recomm
RHSA-2025:4051	Description The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cr
RHSA-2025:4461	Description Node.js is a software development platform for building fast and scalable network applications in the Ja
RHSA-2025:4244	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard
RHSA-2025:4560	Description The libsoup packages provide an HTTP client and server library for GNOME.Security Fix(es): libsoup:
RHSA-2025:4658	Description The libtiff packages contain a library of functions for manipulating Tagged Image File Format (TIFF) fi
RHSA-2025:4791	Description Python is an interpreted, interactive, object-oriented programming language, which includes modules, cl
RHSA-2025:7540	Description The libjpeg-turbo packages contain a library of functions for manipulating JPEG images. They also cont
RHSA-2025:8132	Description The libsoup packages provide an HTTP client and server library for GNOME.Security Fix(es): libsoup:
RHSA-2025:8395	Description The rsync utility enables the users to copy and synchronize files locally or across a network. Synchroniz
RHSA-2025:8414	Description Git is a distributed revision control system with a decentralized architecture. As opposed to centralized v
RHSA-2025:8514	Description Node.js is a software development platform for building fast and scalable network applications in the Ja
RHSA-2025:8686	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard
RHSA-2025:8655	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard
RHSA-2025:8958	Description The libxml2 library is a development toolbox providing the implementation of various XML standards.S
RHSA-2025:8676	Description libxslt is a library for transforming XML files into other textual formats (including HTML, plain text, an
RHSA-2025:9877	Description The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard
SUSE-SU-2025:01702-1	This update for glibc fixes the following issues: CVE-2025-4802: possible execution of attacker controlled code whe
SUSE-SU-2025:01776-1	This update for iputils fixes the following issues: CVE-2025-47268: Fixed integer overflow in RTT calculation can l
SUSE-SU-2025:02027-1	This update for perl fixes the following issues: CVE-2025-40909: Do not change the current directory when cloning
SUSE-SU-2025:02013-1	This update for pam fixes the following issues: CVE-2025-6018: pam_env: Change the default to not read the user .p
SUSE-SU-2025:02080-1	This update for pam-config fixes the following issues: CVE-2025-6018: Stop adding pam_env in AUTH stack, and b
SUSE-SU-2025:02244-1	This update for systemd fixes the following issues: CVE-2025-4598: Fixed race condition that allows a local attacke
SUSE-SU-2025:02259-1	This update for gpg2 fixes the following issues: CVE-2025-30258: Fixed DoS due to a malicious subkey in the keyri
SUSE-SU-2025:0283-1	This update for nginx fixes the following issues: CVE-2023-44487: Mitigate HTTP/2 Rapid Reset Attack (bsc#1216
SUSE-SU-2025:0291-1	This update for iperf fixes the following issues: Update to version 3.18 CVE-2024-53580: Fixed a segmentation viol
SUSE-SU-2025:0369-1	This update for curl fixes the following issues: CVE-2025-0725: Fixed gzip integer overflow (bsc#1236590) CVE-2
SUSE-SU-2025:0401-1	This update for crypto-policies and krb5 fixes the following issues: Security issue fixed: CVE-2025-24528: Fixed ou
SUSE-SU-2025:0548-1	This update for libtasn1 fixes the following issues: CVE-2024-12133: the processing of input DER data containing a

SUSE-SU-2025:0430-1	This update for openssl-3 fixes the following issues: CVE-2024-13176: Fixed timing side-channel in ECDSA signature verification.
SUSE-SU-2025:0582-1	This update for glibc fixes the following issues: CVE-2025-0395: Fix underallocation of abort_msg_s struct (bsc#1234567).
SUSE-SU-2025:0741-1	This update for procs fixes the following issues: Integer overflow due to incomplete fix for CVE-2023-4016 can lead to denial of service.
SUSE-SU-2025:1334-1	This update for pam fixes the following issues: CVE-2024-10041: sensitive data exposure while performing authentication.
SUSE-SU-2025:1381-1	This update for cifs-utils fixes the following issues: CVE-2025-2312: Fixed cifs.upcall making an upcall to the wrong host.
SUSE-SU-2025:1550-1	This update for openssl-3 fixes the following issues: Security: CVE-2025-27587: Timing side channel vulnerability in RSA decryption.

Cumulative hotfixes

Review the cumulative hotfixes for Cloudera Data Services on premises 1.5.5-CHF1.

Cloudera Data Services on premises 1.5.5-CHF1 supports:

- Cloudera Base on premises 7.1.9, 7.1.9 SP1 CHF5, 7.1.7 SP3 CHF10, and 7.3.1.400
- Cloudera Manager 7.13.1.402

Cloudera Data Services on premises 1.5.5-CHF1

There are no new features introduced in cumulative hotfix 1.5.5-CHF1.



Note: Before upgrading to Cloudera Data Services on premises 1.5.5 CHF1 and above with Datalake 7.3.1, follow the [Cloudera Data Engineering Prerequisites](#) to avoid breaking [Data Engineering's Spark support matrix](#).

Cloudera Data Services on premises 1.5.5 existing known issues are carried into Cloudera Data Services on premises 1.5.5-CHF1. For more details, see [Known Issues](#).

Known issue identified in 1.5.5-CHF1

The following are the known issues identified in 1.5.5 CHF1:

COMPX-20437 - Handle DB connection failures in running RPM and CAM

When a customer upgrades from version 1.5.5 to any 1.5.5 hotfix release, the cluster-access-manager and resource-pool-manager pods can enter a CrashLoopBackOff state if they are not restarted during the upgrade.

Restart the cluster-access-manager pod and then restart the resource-pool-manager pod (order of restart is very important). Commands to restart:

```
kubectl rollout restart deployment <cluster-access-manager-deployment-name> -n <namespace>
kubectl rollout restart deployment <resource-pool-manager-deployment-name> -n <namespace>
```

Repository Locations for 1.5.5-CHF1

The URLs for Cloudera Data Services on premises 1.5.5-CHF1 are listed in the following table:

URL Type	Repository Location
Index	<code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/</code>
Manifest	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/manifest.json</code>
Parcels	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2/parcels/</code>

Service packs

The service pack issues that have been shipped for Cloudera Data Services on premises 1.5.5-SP1.

Cloudera Data Services on premises 1.5.5 SP1

Review new features, behavioural changes, known issues, and fixed issues introduced in service pack 1.5.5 SP1.

What's new in Cloudera Data Services on premises 1.5.5 SP1

Understand the functionalities and improvements to features of Cloudera Data Services on premises 1.5.5 SP1.



Note: The Cloudera Data Services on premises UI now utilizes advanced browser capabilities introduced in modern Firefox releases. To ensure full functionality, optimal performance, and UI compatibility, upgrade your Firefox browser to version 77 or later.

Upgrade pre-checks

A new host inspection called Host Prerequisite Inspection is added to the upgrade pre-checks. It verifies that the minimum requirements for the system configurations are met on the hosts. For more information, see [Pre-upgrade checklist](#).

Post Upgrade Validation

This is a set of health checks added as the last step in the upgrade to verify that the upgrade was successful and the cluster is healthy. This includes verifying that the services are healthy, the hosts are upgraded to the new version, and that the Cloudera Control Plane is in a healthy state after upgrade. For the Cloudera Control Plane to be determined as good health, it will verify that RKE2, Longhorn, and Vault are healthy. It will also check that the critical pods in the Cloudera Control Plane namespace are up and running. For more information, see [Upgrading Cloudera Data Services on premises using Cloudera Embedded Container Service](#).

SAML enhancement in Cloudera Management Console

New settings such as Signing and encryption-decryption are added in Cloudera Management Console UI. For more information, see [Setting SAML as IdP](#).

Data Recovery Service (DRS) functionalities

Cloudera Data Services on premises 1.5.5 SP1 and higher versions provide the following DRS features:

- You can view the Backup and Restore Manager on the Cloudera Management Console Backup Manager Control Plane page.
- You can use the S3 locations in Ozone, in AWS buckets, or in both as an external backup location to back up and restore the Cloudera Control Plane.
- DRS automatically backs up Virtual Warehouse configurations from the database along with the namespace backup. You can restore the backup, when required.

For more information, see [Data Recovery Service overview](#).

Cloudera Embedded Container Service containerd And kubelet health tests

Implemented health test Runners for both the Ecs Server and Ecs Agent roles within the Cloudera Embedded Container Service. These runners executed by SMON on each Ecs Server and Ecs Agent to read metrics and determine the health status of containerd and kubelet, which is then displayed on the Cloudera Embedded Container Service Roles page in the Cloudera Manager UI.

Metrics are collected (60 sec of interval) by process running alongside the Ecs Server and Ecs Agent process and sent to SMON.

For more information on enabling Ecs Server/Agent containerd health test and Ecs Server/Agent kubelet health test, see [ECS Agent Health test](#) and [ECS Server Health tests](#).

What's new in Platform Support 1.5.5 SP1

You must be aware of the platform support for Cloudera Data Services on premises 1.5.5 SP1.

Platform Certifications

- Cloudera Base on premises (7.1.9 SP1 CHF11, 7.1.7 SP3 CHF10, 7.3.1.500)
- Cloudera Manager 7.13.1.501
- Iceberg v2 GA on Cloudera Data Warehouse, Cloudera Data Engineering, & Cloudera AI with Ozone
- RHEL 8.10, 9.3, 9.4, 9.5, 9.6
- OEL (RHCK Kernel Only) 8.10, 9.3, 9.4, 9.5, 9.6
- Rocky Linux 8.10, 9.3, 9.4, 9.5, 9.6
- K8s RKE2 1.31 and OCP 4.18
- Longhorn version 1.6.4



Note:

Cloudera Manager 7.11.3 CHF8 does not support any Cloudera Data Services on premises release.

Repository Locations for 1.5.5-SP1

The URLs for Cloudera Data Services on premises 1.5.5-SP1 are listed in the following table:

URL Type	Repository Location
Index	<code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h1000/</code>
Manifest	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h1000/manifest.json</code>

URL Type	Repository Location
Parcels	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h1000/parcels/</code>

Fixed Issues for the Cloudera Data Services on premises 1.5.5 SP1

You can review the list of reported issues and their fixes in Cloudera Data Services on premises 1.5.5 SP1. Fixed issues represent selected issues that were previously logged through Cloudera Support, but are now addressed in the current Cloudera Data Services on premises release. These issues may have been reported in previous versions of Cloudera Data Services on premises as a known issue; meaning they were reported by customers or identified by Cloudera Quality Engineering teams.

OPSX-6589 - Cloudera Embedded Container Service 1.5.4 or 1.5.5, rke2_launch startup script fails to run because root is disallowed to use sudo

Startup failure in Cloudera Embedded Container Service 1.5.4 or 1.5.5 when sudo is disallowed for root. In 1.5.4 SP2, the rke2_launch startup script was updated to use sudo for setting the default NFSv4 configuration. This change caused Cloudera Embedded Container Service startup failures in environments where sudo is disabled for all users (including root).

The rke2_launch installer script has been updated so that inline sudo commands are no longer required and startup now proceeds without needing sudo.

OPSX-6049 - Diagnostics service did not purge logs from /data/env

Previously, the diagnostics service only purged logs under /data/bundles and /data/dp. Log-purging functionality has been extended to also include the /data/env directory.

OPSX-6602 - Break the dependency between unseal vault and restart validations steps

The restart validation step waits for pods in the cdp namespace to become Ready. However, some pods in this namespace depend on vault being unsealed first. Since the vault unsealing occurs after this step, the validation process waits until it times out unless vault is unsealed manually.

To prevent this issue, the validation of the cdp namespace is excluded from the restart validation to eliminate the manual unseal dependency.

OPSX-6349 - Upgrade logic missing for nvidia-device-plugin in Cloudera Embedded Container Service

After upgrading from Cloudera Embedded Container Service 1.5.2 to 1.5.4 (or later), Nvidia pods continued using legacy images because only install logic (not upgrade logic) was present.

The installer now includes a pre-upgrade step that upgrades the nvidia-device-plugin. After this change, upgrades of Cloudera Data Services on premises will ensure the plugin is updated too.

OPSX-6305 - Removal of SUID bit from /opt/cni/bin/install after Cloudera Embedded Container Service upgrade

The issue was identified during upgrades from older Calico versions, which left the binary with SUID permissions, posing a security risk. The fix involved removal of the SUID bit from the /opt/cni/bin/install binary after Cloudera Embedded Container Service upgrades to ensure compliance with security policies.



Note: The change applies specifically to Cloudera Embedded Container Service environments and does not impact OCP, with ongoing verification for Kubernetes 1.31 clusters.



Note: Latest updates will remove the binary entirely in newer Calico versions (v3.28.0+), but current upgrades from older versions require manual or scripted cleanup.

OPSX-6245 - Airgap | Multiple pods are in pending state on rolling restart

When performing consecutive rolling restarts on Cloudera Embedded Container Service clusters, the kube-controller-manager pod sometimes fails to become Ready promptly. This prevents other critical pods (including Vault) from initializing, causing the Vault-unseal step to fail.

The installer now implements pre-stop and post-start actions for Cloudera Embedded Container Service restarts in air-gapped environments to ensure pods return to Ready before proceeding.

OPSX-6241 - rke2-ingress-nginx-controller pod fails after Cloudera Embedded Container Service upgrade

After upgrading Cloudera Embedded Container Service, the ingress controller pod failed to start reliably. The restart sequence for the ingress component now includes forced termination of ingress pods, followed by scaling down and scaling up of the rke2-ingress-nginx-controller pod to ensure correct startup.

OPSX-5216 - Unsupported 'internal alias for registry' option

The internal alias for registry setting in Cloudera Embedded Container Service was enabled, but this configuration is not supported when running Cloudera AI workloads on fresh clusters.

Do not set the internal alias for registry flag if you intend to run Cloudera AI workloads.

OPSX-4763 - Environment state not reset after failed creation

In EnvironmentServiceAsyncUtils, when hive warehouse external directory creation failed, the isAvailable flag was set to false. While the subsequent API calls succeeded, the environment remained in a non-AVAILABLE state. The fix ensures that after failure and retry, the environment can be correctly marked as AVAILABLE when all tasks succeed.



Note: A pod#restart of the environment service may be required in earlier deployments.

OPSX-3323 - Custom log redaction does not work for JSON files in the diag bundle

Redaction rules (configured through the admin console) were not applied to JSON files in the diagnostic bundle, although they worked for .log and .txt files.

The diagnostic-bundle logic has been refactored so that JSON files are now included in log redaction. No behavioural change; only the file coverage is extended.

OPSAPS-74598 - Removal of Internal Alias Configuration from Cloudera Embedded Container Service Installation Wizard

ECS Internal Alias configuration is removed from Cloudera Embedded Container Service installation wizard, so that you do not enable this configuration.

OPSAPS-75358 - Configure the 'worker-shutdown-timeout' value for nginx ingress controller to 15 minutes

Based on the analysis for sample workloads, the worker-shutdown-timeout parameter for the nginx ingress controller has been lowered to 15 minutes to ensure graceful shutdown.

OPSX-6227 - Environment should be marked as FAILED if one of the async tasks fails

Previously, if one asynchronous task failed during environment creation, the environment remained in a CREATED state rather than FAILED, causing downstream workflows to proceed incorrectly.

The environment service now transitions the environment to FAILED status if any async task fails during creation.

OPSX-3953 - Upgrade postgresql image used in cdp-embedded-db

The embedded DB PostgreSQL component is upgraded from version 10.x to 17.4 as part of the Cloudera Control Plane upgrade. When Cloudera Control Plane 1.5.5 SP1 is installed or Cloudera Control Plane 1.5.x is upgraded to 1.5.5 SP1, cdp-embedded-db will participate in the Cloudera Control Plane upgrade. If the prior version is 1.5.x, then the data is automatically migrated to the new PostgreSQL 16 database.

Behavioral issues for the Cloudera Data Services on premises 1.5.5 SP1

Behavioral changes denote a marked change in behavior from the previously released version to this version of Cloudera Data Services on premises release.

Summary:

Implement Longhorn Recurring Job for Prometheus Snapshot Cleanup

Previous behaviour:

Longhorn's automatic snapshot generation, even without configured recurring jobs, can result in high-volume utilization for highly volatile data, such as Prometheus metrics. Prometheus frequently writes new metrics and cleans up old metrics, which is not an ideal write pattern for Longhorn's snapshot management. This can cause volume utilization to exceed configured limits, triggering volume filling up or above 100% alerts in the Cloudera Manager UI, when the actual issue is accumulated snapshots.

New behaviour:

In Cloudera Data Services on premises 1.5.5 SP1, a snapshot cleanup job is automatically configured for all Prometheus volumes. This new Recurring Job runs daily at 1:00 A.M. The timezone setting matches the Cloudera Manager Server's timezone.

You can reconfigure the schedule or completely disable this job through the Longhorn UI.

Reference:

1. [Snapshot documentation](#)
2. [Recurring Snapshot Cleanup](#)

Known issues for the Cloudera Data Services on premises 1.5.5 SP1

You must be aware of the known issues and limitations, the areas of impact, and workaround in Cloudera Data Services on premises 1.5.5 SP1 release.

Cloudera Data Services on premises 1.5.5 existing known issues are carried into Cloudera Data Services on premises 1.5.5 SP1. For more details, see [Known Issues](#).

Known issue identified in 1.5.5 SP1

The following are the known issues identified in 1.5.5 SP1:

OPSAPS-70612: Invalid URL error while installing Cloudera Data Services on premises 1.5.5 SP1 from Cloudera Manager 7.13.1.501

When attempting to install Cloudera Data Services on premises using a Cloudera Manager 7.13.1.501 hotfix, the installation may fail.

The installation fails to process the repository URL correctly in this specific scenario, resulting in an "Invalid URL" error, which blocks the installation.

Perform the following steps to mitigate this issue:

1. Login to Cloudera Manager and click **Hosts Add Hosts**.
2. In the **Add Hosts** page, select the **Add hosts to Cloudera Manager** option and then click **Continue**.
3. Follow the instructions on the **Setup Auto-TLS** page and then click **Continue**.
4. In the **Specify Hosts** page, enter a host name or pattern to search for new hosts to add to the cluster, and then click **Continue**. You can click the **Patterns** link for more information.

A list of matching hosts are displayed.

5. Select the hosts that you want to add and click **Continue**.
6. Select the repository location where Cloudera Manager can access the software required to install on the new hosts. Choose either the **Public Cloudera Repository** or a **Custom Repository**, and provide the URL of the custom repository available on your local network.

7. Click Continue.

After completing the above tasks, you can proceed with the Cloudera Data Services on premises installation.

OPSX-6794 - Upgrading from Cloudera Data Services on premises 1.5.4 to 1.5.5 Service Pack 1 fails with a Upgrade embedded DB error message midway through the upgrade process.

When attempting to upgrade from Cloudera Data Services on premises 1.5.4 to 1.5.5 Service Pack 1, the upgrade process could fail.

If you have performed a manual **Expand Volume** operation on the cdp-embedded-db-backend Persistent Volume Claim (PVC) of the cdp-embedded-db service in **cdp** namespace, it can cause the upgrade process to fail. The upgrade workflow attempts to apply the original value that was used during the initial cluster set up. As Longhorn does not support reducing the volume size, the upgrade process fails.

To avoid stalling the ongoing upgrade process, you must manually exclude the PVC for the cdp-embedded-db-backend in the upgrade process.



Important: You must abort the upgrade process before applying this workaround.

Perform the following steps:

```
#Login to ECS master node
ssh root@ecs_master_node

#Switch to embedded db directory in the ECS parcel
cd /opt/cloudera/parcels/ECS-1.5.5-h1000-b167-ecs-1.5.5-h1000-b167.p0.72522329/installer/helm/cdp-embedded-db

#Create a tmp working directory
mkdir temp-cdp-embedded-db-1.5.5-h1000-b167

#Extract the tgz contents into the temp working directory
tar -xzf cdp-embedded-db-1.5.5-h1000-b167.tgz -C ./temp-cdp-embedded-db-1.5.5-h1000-b167

#Manual changes
vi temp-cdp-embedded-db-1.5.5-h1000-b167/templates/persistentVolumeClaim.yaml

#This file will have two PVCs. cdp-embedded-db-backend and cdp-embedded-db-backend-v17. Remove cdp-embedded-db-backend PVC from the file. Save the file and exit

#Take a backup of the old tgz file
mv cdp-embedded-db-1.5.5-h1000-b167.tgz backup-cdp-embedded-db-1.5.5-h1000-b167.tgz

#Switch to temp working directory
cd temp-cdp-embedded-db-1.5.5-h1000-b167

#Prepare an updated tgz file
tar -czvf ../cdp-embedded-db-1.5.5-h1000-b167.tgz .

#Switch to previous directory and check the tgz file
cd ..
ls -lrth

#Move the backup tgz to a different directory
mv backup-cdp-embedded-db-1.5.5-h1000-b167.tgz /tmp/some_directory
```

```
#Cleanup the temp working directory
rm -rf temp-cdp-embedded-db-1.5.5-h1000-b167/

#Backup the following secret in cdp namespace to 'some_directory'
'. Make sure to replace 'some_directory' with an appropriate path
kubectl get secret cdp-thunderheaduserpreference-db-secret -n
cdp -o yaml > /tmp/some_directory/backup-cdp-thunderheaduserpref
erence-db-secret

#Delete the following secret in cdp namespace
kubectl delete secret cdp-thunderheaduserpreference-db-secret -n
cdp
```

Once you complete these steps, you can resume the upgrade process.

OPSX-6797 - A possible upgrade failure resulting from expanding the cdp-embedded-db volume from Longhorn UI after the initial installation of your Data Services on premises cluster

If you have expanded cdp-embedded-db volume from Longhorn UI after the initial installation of your Data Services on premises cluster, you must complete the workaround steps before planning your upgrade to Cloudera Data Services on premises 1.5.5 Service Pack 1 to avoid a potential upgrade failure.

Login to your ECS master node on your cluster and perform the following these steps:

```
# Find the size of cdp-embedded-db-backend PVC, look for the val
ue in CAPACITY column
kubectl get pvc -n cdp | grep cdp-embedded-db-backend
# Find the latest/current copy of cdp-private-installer-values
# If the cluster is on 1.5.5-b212, then the secret to use will be
cdp-private-installer-values-1.5.5-b212
kubectl get secret -n cdp | grep cdp-private-installer-values
# For this example, we will use 'cdp-private-installer-values-1
.5.5-b212'
# Please use the correct one from your cluster

# Backup the existing values, in case you need to restore it late
r
kubectl get secret cdp-private-installer-values-1.5.5-b212 -n
cdp -o yaml > backup-cdp-private-installer-values-1.5.5-b212.yaml

# Make a new copy before updating the values
cp backup-cdp-private-installer-values-1.5.5-b212.yaml updated-c
dp-private-installer-values-1.5.5-b212.yaml
# Edit 'updated-cdp-private-installer-values-1.5.5-b212.yaml' u
sing the following instructions:
# 1. Base64decode the value present in 'values.yaml.merged' att
ribute
# 2. Update Database.EmbeddedDbStorage to the latest value obtain
ed in the first step
# 3. Base64encode the updated yaml string and update it in 'val
ues.yaml.merged' attribute
# 4. Delete metadata.creationTimestamp, metadata.name, metadata.r
esourceVersion and metadata.uid attributes
# 5. Save these changes to updated-cdp-private-installer-values-1
.5.5-b212.yaml

# Delete the existing secret
kubectl delete secret cdp-private-installer-values-1.5.5-b212 -n
cdp

# Create a new secret using the updated file
kubectl apply -f updated-cdp-private-installer-values-1.5.5-b212
.yaml
```

```
# Check if the secret got created
kubectl get secret -n cdp | grep cdp-private-installer-values

# Verify if the value got updated
kubectl get secret cdp-private-installer-values-1.5.5-b212 -n
cdp -o jsonpath='{.data.values\.yaml\.merged}' | base64 --decode
```

COMPX-20437 - DB connection failures causing RPM and CAM pods to CrashLoopBackOff

During an upgrade from version 1.5.5 to any 1.5.5 hotfix release, the cluster-access-manager (CAM) and resource-pool-manager (RPM) pods can enter a CrashLoopBackOff state if they are not automatically restarted during the upgrade.

After upgrade, manually restart the CAM pod and then restart the RPM pod (order of restart is very important).

Commands to restart:

```
kubectl rollout restart deployment <cluster-access-manager-depl
oyment-name> -n <namespace>
kubectl rollout restart deployment <resource-pool-manager-deplo
yment-name> -n <namespace>
```

OBS-9491 - Prometheus configuration exceeds size limit in large environments

In environments with a large number of namespaces (approximately 300 or more per environment), the Prometheus configuration for Cloudera Monitoring might exceed the 1 MB Kubernetes Secret size limit. If the total size, which depends on factors such as the number of namespaces, the length of namespace names, their variability, and the size of the certificate store, exceeds 1 MB, the new Prometheus configuration will not be applied, and new namespaces will not be monitored. As a result, the telemetry data will not be collected from those namespaces and will not be reflected on the corresponding Grafana charts.

To resolve this issue, you must enable Prometheus configuration compression at the Cloudera Control Plane level.

1. Upgrade to Cloudera Data Services on premises 1.5.5 SP1 or a higher version.
2. Back up the secret from the monitoring namespace to allow for rollback and disabling compression later.
3. Set the environment variable `ENABLE_ENVIRONMENT_PROMETHEUS_CONFIG_COMPRESSION` to "true" on the `cdp-release-monitoring-pvcservice` deployment in the Cloudera Control Plane namespace.



Note: The upper bound for monitoring is based on the number of namespaces per environment, the length of the namespace name, the variability of the names, and the size of the cert store.

- For example, about 700 workload namespaces in a single environment with a short, consistent pattern ("aaaa-bbbb-XXXX") generate around 150 KB of configuration, which fits within the 1 MB limit.
- About 3000 namespaces in a single environment with a longer, consistent pattern ("aaaa-cccc-dddd-XXXXX") generate about 900 KB.

Rolling back the workaround:

1. Set the environment variable `ENABLE_ENVIRONMENT_PROMETHEUS_CONFIG_COMPRESSION` to "false" on the `cdp-release-monitoring-pvcservice` deployment in the Cloudera Control Plane namespace.
2. Reapply the secret from the backup.

OPSX-6618 - In Cloudera Embedded Container Service upgrade not all volumes are upgraded to the latest longhorn version

During restart of the Cloudera Embedded Container Service cluster from 1.5.5 to 1.5.5 SP1, the upgrade failed due to longhorn health issues. This is because one of the volumes was degraded.

Follow the steps to resolve the issue:

- Identify the problematic volumes (degraded state).
- Set the value of `spec.numberOfReplicas` of the volume to the number of active replicas. For example, set the value to 3 if two replicas are active.
- Apply the fix before or during the upgrade as per the workaround instructions (refer to Longhorn issue #11825). Longhorn Engineering is addressing in v1.11.0 and will back-port the fix. The workaround is included as part of the upgrade. However, if the issue is noticed even after upgrade please follow the workaround steps documented here: <https://github.com/longhorn/longhorn/issues/11825>

COMPX-23842 / COMPX-24130/ COMPX-23319 - Pod status shown as OutOfcpu, OutOfmemory, or Pending after a cluster restart

During a cluster restart, due to an upgrade or normal maintenance, all nodes in the cluster are restarted. During this process, the cluster operates with reduced resource capacity. When this occurs, pod placement can be rejected, resulting in some pods entering `OutOfcpu` or `OutOfmemory` state.

Do not restart failed pods until the cluster restart process has fully completed. Once the cluster is fully operational:

- Restart any pods that are in a failed (`OutOfcpu` or `OutOfmemory`) state, so they can be rescheduled properly.
- If any pods remain in a pending state after the restart, restart the YuniKorn scheduler to refresh scheduling:

```
kubectl rollout restart deployment yunikorn-scheduler -n yunikorn
```

OPSX-6566 - Cloudera Embedded Container Service restart fails with etcd connectivity issues

Restart of the Cloudera Embedded Container Service server fails with etcd error: "error reading from server"

1. Identify the server role that failed.
2. Restart only the Cloudera Embedded Container Service server role which failed.
3. Once the server role is restarted and healthy, proceed with the remainder of the Cloudera Embedded Container Service server role restart sequence.

OPSX-6401 - Istio ingress-default-cert is not created in the upgrade scenario

After upgrading to 1.5.5 SP1, the Secret `ingress-default-cert` is not created in the `istio-ingress` namespace. Because this certificate is expected, failing to create it causes components like CAII & MR provisioning to fail.

To resolve this issue:

- After upgrading from an older version to 1.5.5 SP1, check for the secret called `ingress-default-cert` in the `istio-ingress` namespace (ns).

For example:

```
kubectl get secret ingress-default-cert -n istio-ingress
```

- If missing, copy the identical secret from `kube-system` namespace (or from the pre-upgrade environment) into `istio-ingress` namespace with the same name and contents.
- This will restore the expected certificate and allow CAII & MR provisioning to proceed.

OPSX-6645 - Cloudera Embedded Container Service upgrade failure at restart step

When the Cloudera Embedded Container Service role fails to start after a node reboot or service restart, the root cause can be that the etcd defragmentation process which runs on startup takes longer than the component timeout thresholds. As a result:

- The kubelet service may fail to start or time out.
- The kube-apiserver, kube-scheduler or kube-controller-manager roles may remain in a NotReady state.
- etcd may perform automatic defragmentation at startup.
- The API server may fail to connect to etcd (connection refused or timeout).

To fix this issue:

1. Resume the Cloudera Embedded Container Service start/restart from the Cloudera Manager UI once etcd has come up.
2. Ensure etcd meets production hardware and configuration requirements: For more information, see [etcd requirements](#) and [Knowledge Base](#).

OPSX-6767 - Cloudera Embedded Container Service cluster has stale configuration after Cloudera Manager upgrade to 7.13.1.501-b2 from 7.11.3.24

After upgrading Cloudera Manager to version 7.13.1.501, the Cloudera Embedded Container Service shows a staleness indicator. This occurs due to configuration changes applied by the upgrade:

- worker-shutdown-timeout: reduced from 24 hours (86,400 s) to 15 minutes (900 s).
- smon_host: a new monitoring configuration added.
- smon_port: a new monitoring port configuration (9997).

No action required — the staleness is expected following this upgrade and can be safely ignored. The indicator will automatically clear once Cloudera Embedded Container Service is upgraded to version 1.5.5 SP1 or later.

If you prefer to clear the staleness indicator right away, you may manually refresh the Cloudera Embedded Container Service service through the Cloudera Manager UI.



Note: Performing a refresh will restart the ingress controller and may cause a brief service interruption.

OPSX-6638 - Post rolling restart many pods are stuck in pending state

Pods remain in Pending state and fail to schedule with etcd performance warnings.

Restart the Cloudera Embedded Container Service master role which has etcd performance issues. For more information, see [etcd hardware recommendations](#) and [Knowledge Base](#).

List of fixed Common Vulnerabilities and Exposures in 1.5.5 SP1

Review the Common vulnerabilities and Exposures (CVEs) that were fixed in this release of Cloudera Data Services on premises 1.5.5 SP1.

CVE ID	Description
CVE-2001-1268	Directory traversal vulnerability in Info-ZIP UnZip 5.42 and earlier allows attackers to overwrite arbitrary files during archive extraction via a .. (dot dot) in an extracted filename.
CVE-2001-1269	Info-ZIP UnZip 5.42 and earlier allows attackers to overwrite arbitrary files during archive extraction via filenames in the archive that begin with the '/' (slash) character.
CVE-2005-0602	Unzip 5.51 and earlier does not properly warn the user when extracting setuid or setgid files, which may allow local users to gain privileges.
CVE-2005-2541	Tar 1.15.1 does not properly warn the user when extracting setuid or setgid files, which may allow local users or remote attackers to gain privileges.

CVE ID	Description
CVE-2007-5686	initscripts in rPath Linux 1 sets insecure permissions for the /var/log/btmp file, which allows local users to obtain sensitive information regarding authentication attempts. NOTE: because sshd detects the insecure permissions and does not log certain events, this also prevents sshd from logging failed authentication attempts by remote attackers.
CVE-2010-4756	The glob implementation in the GNU C Library (aka glibc or libc6) allows remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in STAT commands to an FTP daemon, a different vulnerability than CVE-2010-2632.
CVE-2010-5312	Cross-site scripting (XSS) vulnerability in jquery.ui.dialog.js in the Dialog widget in jQuery UI before 1.10.0 allows remote attackers to inject arbitrary web script or HTML via the title option.
CVE-2011-3374	It was found that apt-key in apt, all versions, do not correctly validate gpg keys with the master keyring, leading to a potential man-in-the-middle attack.
CVE-2011-3389	The SSL protocol, as used in certain configurations in Microsoft Windows and Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and other products, encrypts data by using CBC mode with chained initialization vectors, which allows man-in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack (BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket API, (2) the Java URLConnection API, or (3) the Silverlight WebClient API, aka a "BEAST" attack.
CVE-2011-4116	_is_safe in the File::Temp module for Perl does not properly handle symlinks.
CVE-2011-4461	Jetty 8.1.0.RC2 and earlier computes hash values for form parameters without restricting the ability to trigger hash collisions predictably, which allows remote attackers to cause a denial of service (CPU consumption) by sending many crafted parameters.
CVE-2011-4969	Cross-site scripting (XSS) vulnerability in jQuery before 1.6.3, when using location.hash to select elements, allows remote attackers to inject arbitrary web script or HTML via a crafted tag.
CVE-2012-2663	extensions/libxt_tcp.c in iptables through 1.4.21 does not match TCP SYN+FIN packets in --syn rules, which might allow remote attackers to bypass intended firewall restrictions via crafted packets. NOTE: the CVE-2012-6638 fix makes this issue less relevant.
CVE-2012-6708	jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.
CVE-2013-4235	shadow: TOCTOU (time-of-check time-of-use) race condition when copying and removing directory trees
CVE-2013-4392	systemd, when updating file permissions, allows local users to change the permissions and SELinux security contexts for arbitrary files via a symlink attack on unspecified files.
CVE-2013-7445	The Direct Rendering Manager (DRM) subsystem in the Linux kernel through 4.x mishandles requests for Graphics Execution Manager (GEM) objects, which allows context-dependent attackers to cause a denial of service (memory consumption) via an application that processes graphics data, as demonstrated by JavaScript code that creates many CANVAS elements for rendering by Chrome or Firefox.
CVE-2014-6071	jQuery 1.4.2 allows remote attackers to conduct cross-site scripting (XSS) attacks via vectors related to use of the text method inside after.
CVE-2014-9471	The parse_datetime function in GNU coreutils allows remote attackers to cause a denial of service (crash) or possibly execute arbitrary code via a crafted date string, as demonstrated by the "--date=TZ='123'345" @1" string to the touch or date command.
CVE-2015-4041	The keycompare_mb function in sort.c in sort in GNU Coreutils through 8.23 on 64-bit platforms performs a size calculation without considering the number of bytes occupied by multibyte characters, which allows attackers to cause a denial of service (heap-based buffer overflow and application crash) or possibly have unspecified other impact via long UTF-8 strings.
CVE-2015-4042	Integer overflow in the keycompare_mb function in sort.c in sort in GNU Coreutils through 8.23 might allow attackers to cause a denial of service (application crash) or possibly have unspecified other impact via long strings.
CVE-2015-5237	protobuf allows remote authenticated attackers to cause a heap-based buffer overflow.

CVE ID	Description
CVE-2015-7837	The Linux kernel, as used in Red Hat Enterprise Linux 7, kernel-rt, and Enterprise MRG 2 and when booted with UEFI Secure Boot enabled, allows local users to bypass intended securelevel/secureboot restrictions by leveraging improper handling of secure_boot flag across kexec reboot.
CVE-2015-8553	Xen allows guest OS users to obtain sensitive information from uninitialized locations in host OS kernel memory by not enabling memory and I/O decoding control bits. NOTE: this vulnerability exists because of an incomplete fix for CVE-2015-0777.
CVE-2015-9251	jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.
CVE-2016-1000027	Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.
CVE-2016-2781	chroot in GNU coreutils, when used with --userspec, allows local users to escape to the parent session via a crafted TIOCTSTIOCTL call, which pushes characters to the terminal's input buffer.
CVE-2016-4971	GNU wget before 1.18 allows remote servers to write to arbitrary files by redirecting a request from HTTP to a crafted FTP resource.
CVE-2016-7098	Race condition in wget 1.17 and earlier, when used in recursive or mirroring mode to download a single file, might allow remote servers to bypass intended access list restrictions by keeping an HTTP connection open.
CVE-2016-7103	Cross-site scripting (XSS) vulnerability in jQuery UI before 1.12.0 might allow remote attackers to inject arbitrary web script or HTML via the closeText parameter of the dialog function.
CVE-2016-8660	The XFS subsystem in the Linux kernel through 4.8.2 allows local users to cause a denial of service (fdatasync failure and system hang) by using the vfs syscall group in the trinity program, related to a "page lock order bug in the XFS seek hole/data implementation."
CVE-2017-0537	An information disclosure vulnerability in the kernel USB gadget driver could enable a local malicious application to access data outside of its permission levels. This issue is rated as Moderate because it first requires compromising a privileged process. Product: Android. Versions: Kernel-3.18. Android ID: A-31614969.
CVE-2017-11164	In PCRE 8.41, the OP_KETRMATCH feature in the match function in pcre_exec.c allows stack exhaustion (uncontrolled recursion) when processing a crafted regular expression.
CVE-2017-13165	An elevation of privilege vulnerability in the kernel file system. Product: Android. Versions: Android kernel. Android ID A-31269937.
CVE-2017-13693	The acpi_ds_create_operands() function in drivers/acpi/acpica/dsutils.c in the Linux kernel through 4.12.9 does not flush the operand cache and causes a kernel stack dump, which allows local users to obtain sensitive information from kernel memory and bypass the KASLR protection mechanism (in the kernel through 4.9) via a crafted ACPI table.
CVE-2017-13716	The C++ symbol demangler routine in cplusplus-dem.c in libiberty, as distributed in GNU Binutils 2.29, allows remote attackers to cause a denial of service (excessive memory allocation and application crash) via a crafted file, as demonstrated by a call from the Binary File Descriptor (BFD) library (aka libbfd).
CVE-2017-13745	There is a reachable assertion abort in the function jpc_dec_process_sot() in jpc/jpc_dec.c in JasPer 2.0.12 that will lead to a remote denial of service attack by triggering an unexpected jpc_ppmstabtostreams return value, a different vulnerability than CVE-2018-9154.
CVE-2017-16231	In PCRE 8.41, after compiling, a pcretest load test PoC produces a crash overflow in the function match() in pcre_exec.c because of a self-recursive call. NOTE: third parties dispute the relevance of this report, noting that there are options that can be used to limit the amount of stack that is used
CVE-2017-16516	In the yajl-ruby gem 1.3.0 for Ruby, when a crafted JSON file is supplied to Yajl::Parser.new.parse, the whole ruby process crashes with a SIGABRT in the yajl_string_decode function in yajl_encode.c. This results in the whole ruby process terminating and potentially a denial of service.
CVE-2017-18018	In GNU Coreutils through 8.29, chown-core.c in chown and chgrp does not prevent replacement of a plain file with a symlink during use of the POSIX "-R -L" options, which allows local users to modify the ownership of arbitrary files by leveraging a race condition.
CVE-2017-3161	The HDFS web UI in Apache Hadoop before 2.7.0 is vulnerable to a cross-site scripting (XSS) attack through an unescaped query parameter.

CVE ID	Description
CVE-2017-3162	HDFS clients interact with a servlet on the DataNode to browse the HDFS namespace. The NameNode is provided as a query parameter that is not validated in Apache Hadoop before 2.7.0.
CVE-2017-3735	While parsing an IPAddressFamily extension in an X.509 certificate, it is possible to do a one-byte overread. This would result in an incorrect text display of the certificate. This bug has been present since 2006 and is present in all versions of OpenSSL before 1.0.2m and 1.1.0g.
CVE-2017-3736	There is a carry propagating bug in the x86_64 Montgomery squaring procedure in OpenSSL before 1.0.2m and 1.1.0 before 1.1.0g. No EC algorithms are affected. Analysis suggests that attacks against RSA and DSA as a result of this defect would be very difficult to perform and are not believed likely. Attacks against DH are considered just feasible (although very difficult) because most of the work necessary to deduce information about a private key may be performed offline. The amount of resources required for such an attack would be very significant and likely only accessible to a limited number of attackers. An attacker would additionally need online access to an unpatched system using the target private key in a scenario with persistent DH parameters and a private key that is shared between multiple clients. This only affects processors that support the BMI1, BMI2 and ADX extensions like Intel Broadwell (5th generation) and later or AMD Ryzen.
CVE-2017-3737	OpenSSL 1.0.2 (starting from version 1.0.2b) introduced an "error state" mechanism. The intent was that if a fatal error occurred during a handshake then OpenSSL would move into the error state and would immediately fail if you attempted to continue the handshake. This works as designed for the explicit handshake functions (SSL_do_handshake(), SSL_accept() and SSL_connect()), however due to a bug it does not work correctly if SSL_read() or SSL_write() is called directly. In that scenario, if the handshake fails then a fatal error will be returned in the initial function call. If SSL_read()/SSL_write() is subsequently called by the application for the same SSL object then it will succeed and the data is passed without being decrypted/encrypted directly from the SSL/TLS record layer. In order to exploit this issue an application bug would have to be present that resulted in a call to SSL_read()/SSL_write() being issued after having already received a fatal error. OpenSSL version 1.0.2b-1.0.2m are affected. Fixed in OpenSSL 1.0.2n. OpenSSL 1.1.0 is not affected.
CVE-2017-3738	There is an overflow bug in the AVX2 Montgomery multiplication procedure used in exponentiation with 1024-bit moduli. No EC algorithms are affected. Analysis suggests that attacks against RSA and DSA as a result of this defect would be very difficult to perform and are not believed likely. Attacks against DH1024 are considered just feasible, because most of the work necessary to deduce information about a private key may be performed offline. The amount of resources required for such an attack would be significant. However, for an attack on TLS to be meaningful, the server would have to share the DH1024 private key among multiple clients, which is no longer an option since CVE-2016-0701. This only affects processors that support the AVX2 but not ADX extensions like Intel Haswell (4th generation). Note: The impact from this issue is similar to CVE-2017-3736, CVE-2017-3732 and CVE-2015-3193. OpenSSL version 1.0.2-1.0.2m and 1.1.0-1.1.0g are affected. Fixed in OpenSSL 1.0.2n. Due to the low severity of this issue we are not issuing a new release of OpenSSL 1.1.0 at this time. The fix will be included in OpenSSL 1.1.0h when it becomes available. The fix is also available in commit e502cc86d in the OpenSSL git repository.
CVE-2017-5499	Integer overflow in libjasper/jpc/jpc_dec.c in JasPer 1.900.17 allows remote attackers to cause a denial of service (crash) via a crafted file.
CVE-2017-5503	The dec_clnpass function in libjasper/jpc/jpc_t1dec.c in JasPer 1.900.27 allows remote attackers to cause a denial of service (invalid memory write and crash) or possibly have unspecified other impact via a crafted image.
CVE-2017-5504	The jpc_undo_roi function in libjasper/jpc/jpc_dec.c in JasPer 1.900.27 allows remote attackers to cause a denial of service (invalid memory read and crash) via a crafted image.
CVE-2017-5505	The jas_matrix_asl function in jas_seq.c in JasPer 1.900.27 allows remote attackers to cause a denial of service (invalid memory read and crash) via a crafted image.
CVE-2017-6519	avahi-daemon in Avahi through 0.6.32 and 0.7 inadvertently responds to IPv6 unicast queries with source addresses that are not on-link, which allows remote attackers to cause a denial of service (traffic amplification) and may cause information leakage by obtaining potentially sensitive information from the responding device via port-5353 UDP packets. NOTE: this may overlap CVE-2015-2809.
CVE-2017-7245	Stack-based buffer overflow in the pcre32_copy_substring function in pcre_get.c in libpcre1 in PCRE 8.40 allows remote attackers to cause a denial of service (WRITE of size 4) or possibly have unspecified other impact via a crafted file.
CVE-2017-7246	Stack-based buffer overflow in the pcre32_copy_substring function in pcre_get.c in libpcre1 in PCRE 8.40 allows remote attackers to cause a denial of service (WRITE of size 268) or possibly have unspecified other impact via a crafted file.

CVE ID	Description
CVE-2017-7475	Cairo version 1.15.4 is vulnerable to a NULL pointer dereference related to the FT_Load_Glyph and FT_Render_Glyph resulting in an application crash.
CVE-2017-9782	JasPer 2.0.12 allows remote attackers to cause a denial of service (heap-based buffer over-read and application crash) via a crafted image, related to the jp2_decode function in libjasper/jp2/jp2_dec.c.
CVE-2018-0732	During key agreement in a TLS handshake using a DH(E) based ciphersuite a malicious server can send a very large prime value to the client. This will cause the client to spend an unreasonably long period of time generating a key for this prime resulting in a hang until the client has finished. This could be exploited in a Denial Of Service attack. Fixed in OpenSSL 1.1.0i-dev (Affected 1.1.0-1.1.0h). Fixed in OpenSSL 1.0.2p-dev (Affected 1.0.2-1.0.2o).
CVE-2018-0734	The OpenSSL DSA signature algorithm has been shown to be vulnerable to a timing side channel attack. An attacker could use variations in the signing algorithm to recover the private key. Fixed in OpenSSL 1.1.1a (Affected 1.1.1). Fixed in OpenSSL 1.1.0j (Affected 1.1.0-1.1.0i). Fixed in OpenSSL 1.0.2q (Affected 1.0.2-1.0.2p).
CVE-2018-0737	The OpenSSL RSA Key generation algorithm has been shown to be vulnerable to a cache timing side channel attack. An attacker with sufficient access to mount cache timing attacks during the RSA key generation process could recover the private key. Fixed in OpenSSL 1.1.0i-dev (Affected 1.1.0-1.1.0h). Fixed in OpenSSL 1.0.2p-dev (Affected 1.0.2b-1.0.2o).
CVE-2018-0739	Constructed ASN.1 types with a recursive definition (such as can be found in PKCS7) could eventually exceed the stack given malicious input with excessive recursion. This could result in a Denial Of Service attack. There are no such structures used within SSL/TLS that come from untrusted sources so this is considered safe. Fixed in OpenSSL 1.1.0h (Affected 1.1.0-1.1.0g). Fixed in OpenSSL 1.0.2o (Affected 1.0.2b-1.0.2n).
CVE-2018-1000021	GIT version 2.15.1 and earlier contains a Input Validation Error vulnerability in Client that can result in problems including messing up terminal configuration to RCE. This attack appear to be exploitable via The user must interact with a malicious git server, (or have their traffic modified in a MITM attack).
CVE-2018-1000201	ruby-ffi version 1.9.23 and earlier has a DLL loading issue which can be hijacked on Windows OS, when a Symbol is used as DLL name instead of a String This vulnerability appears to have been fixed in v1.9.24 and later.
CVE-2018-1000520	ARM mbedTLS version 2.7.0 and earlier contains a Ciphersuite Allows Incorrectly Signed Certificates vulnerability in mbedtls_ssl_get_verify_result() that can result in ECDSA-signed certificates are accepted, when only RSA-signed ones should be.. This attack appear to be exploitable via Peers negotiate a TLS-ECDH-RSA-* ciphersuite. Any of the peers can then provide an ECDSA-signed certificate, when only an RSA-signed one should be accepted..
CVE-2018-1000654	GNU Libtasn1-4.13 libtasn1-4.13 version libtasn1-4.13, libtasn1-4.12 contains a DoS, specifically CPU usage will reach 100% when running asn1Paser against the POC due to an issue in _asn1_expand_object_id(p_tree), after a long time, the program will be killed. This attack appears to be exploitable via parsing a crafted file.
CVE-2018-1000876	binutils version 2.32 and earlier contains a Integer Overflow vulnerability in objdump, bfd_get_dynamic_reloc_upper_bound, bfd_canonicalize_dynamic_reloc that can result in Integer overflow trigger heap overflow. Successful exploitation allows execution of arbitrary code.. This attack appear to be exploitable via Local. This vulnerability appears to have been fixed in after commit 3a551c7a1b80fca579461774860574eabfd7f18f.
CVE-2018-1000879	libarchive version commit 379867ecb330b3a952fb7bfa7bffb7bbd5547205 onwards (release v3.3.0 onwards) contains a CWE-476: NULL Pointer Dereference vulnerability in ACL parser - libarchive/ archive_acl.c, archive_acl_from_text_l() that can result in Crash/DoS. This attack appear to be exploitable via the victim must open a specially crafted archive file.
CVE-2018-1000880	libarchive version commit 9693801580c0cf7c70e862d305270a16b52826a7 onwards (release v3.2.0 onwards) contains a CWE-20: Improper Input Validation vulnerability in WARC parser - libarchive/ archive_read_support_format_warc.c, _warc_read() that can result in DoS - quasi-infinite run time and disk usage from tiny file. This attack appear to be exploitable via the victim must open a specially crafted WARC file.
CVE-2018-10237	Unbounded memory allocation in Google Guava 11.0 through 24.x before 24.1.1 allows remote attackers to conduct denial of service attacks against servers that depend on this library and deserialize attacker-provided data, because the AtomicDoubleArray class (when serialized with Java serialization) and the CompoundOrdering class (when serialized with GWT serialization) perform eager allocation without appropriate checks on what a client has sent and whether the data size is reasonable.
CVE-2018-10779	TIFFWriteScanline in tif_write.c in LibTIFF 3.8.2 has a heap-based buffer over-read, as demonstrated by bmp2tiff.

CVE ID	Description
CVE-2018-10801	TIFFClientOpen in tif_unix.c in LibTIFF 3.8.2 has memory leaks, as demonstrated by bmp2tiff.
CVE-2018-1121	procpns-ng, procpns is vulnerable to a process hiding through race condition. Since the kernel's proc_pid_readdir() returns PID entries in ascending numeric order, a process occupying a high PID can use inotify events to determine when the process list is being scanned, and fork/exec to obtain a lower PID, thus avoiding enumeration. An unprivileged attacker can hide a process from procpns-ng's utilities by exploiting a race condition in reading /proc/PID entries. This vulnerability affects procpns and procpns-ng up to version 3.3.15, newer versions might be affected also.
CVE-2018-12641	An issue was discovered in arm_pt in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30. Stack Exhaustion occurs in the C++ demangling functions provided by libiberty, and there are recursive stack frames: demangle_arm_hp_template, demangle_class_name, demangle_fund_type, do_type, do_arg, demangle_args, and demangle_nested_args. This can occur during execution of nm-new.
CVE-2018-12697	A NULL pointer dereference (aka SEGV on unknown address 0x000000000000) was discovered in work_stuff_copy_to_from in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30. This can occur during execution of objdump.
CVE-2018-12698	demangle_template in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30, allows attackers to trigger excessive memory consumption (aka OOM) during the "Create an array for saving the template argument values" XNEWVEC call. This can occur during execution of objdump.
CVE-2018-12928	In the Linux kernel 4.15.0, a NULL pointer dereference was discovered in hfs_ext_read_extent in hfs.ko. This can occur during a mount of a crafted hfs filesystem.
CVE-2018-12929	ntfs_read_locked_inode in the ntfs.ko filesystem driver in the Linux kernel 4.15.0 allows attackers to trigger a use-after-free read and possibly cause a denial of service (kernel oops or panic) via a crafted ntfs filesystem.
CVE-2018-12930	ntfs_end_buffer_async_read in the ntfs.ko filesystem driver in the Linux kernel 4.15.0 allows attackers to trigger a stack-based out-of-bounds write and cause a denial of service (kernel oops or panic) or possibly have unspecified other impact via a crafted ntfs filesystem.
CVE-2018-12931	ntfs_attr_find in the ntfs.ko filesystem driver in the Linux kernel 4.15.0 allows attackers to trigger a stack-based out-of-bounds write and cause a denial of service (kernel oops or panic) or possibly have unspecified other impact via a crafted ntfs filesystem.
CVE-2018-12934	remember_Ktype in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.30, allows attackers to trigger excessive memory consumption (aka OOM). This can occur during execution of cxxfilt.
CVE-2018-15594	arch/x86/kernel/paravirt.c in the Linux kernel before 4.18.1 mishandles certain indirect calls, which makes it easier for attackers to conduct Spectre-v2 attacks against paravirtual guests.
CVE-2018-15919	Remotely observable behaviour in auth-gss2.c in OpenSSH through 7.8 could be used by remote attackers to detect existence of users on a target system when GSS2 is in use. NOTE: the discoverer states 'We understand that the OpenSSH developers do not want to treat such a username enumeration (or "oracle") as a vulnerability.'
CVE-2018-16335	newoffsets handling in ChopUpSingleUncompressedStrip in tif_dirread.c in LibTIFF 4.0.9 allows remote attackers to cause a denial of service (heap-based buffer overflow and application crash) or possibly have unspecified other impact via a crafted TIFF file, as demonstrated by tiff2pdf. This is a different vulnerability than CVE-2018-15209.
CVE-2018-16862	A security flaw was found in the Linux kernel in a way that the cleancache subsystem clears an inode after the final file truncation (removal). The new file created with the same inode may contain leftover pages from cleancache and the old file data instead of the new one.
CVE-2018-17101	An issue was discovered in LibTIFF 4.0.9. There are two out-of-bounds writes in cpTags in tools/tiff2bw.c and tools/pal2rgb.c, which can cause a denial of service (application crash) or possibly have unspecified other impact via a crafted image file.
CVE-2018-17360	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.31. a heap-based buffer over-read in bfd_getl32 in libbfd.c allows an attacker to cause a denial of service through a crafted PE file. This vulnerability can be triggered by the executable objdump.
CVE-2018-17794	An issue was discovered in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a NULL pointer dereference in work_stuff_copy_to_from when called from iterate_demangle_function.
CVE-2018-17977	The Linux kernel 4.14.67 mishandles certain interaction among XFRM Netlink messages, IPPROTO_AH packets, and IPPROTO_IP packets, which allows local users to cause a denial of service (memory consumption and system hang) by leveraging root access to execute crafted applications, as demonstrated on CentOS 7.

CVE ID	Description
CVE-2018-17985	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a stack consumption problem caused by the cplus_demangle_type function making recursive calls to itself in certain scenarios involving many 'P' characters.
CVE-2018-18064	cairo through 1.15.14 has an out-of-bounds stack-memory write during processing of a crafted document by WebKitGTK+ because of the interaction between cairo-rectangular-scan-converter.c (the generate and render_rows functions) and cairo-image-compositor.c (the _cairo_image_spans_and_zero function).
CVE-2018-18309	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.31. An invalid memory address dereference was discovered in read_reloc in reloc.c. The vulnerability causes a segmentation fault and application crash, which leads to denial of service, as demonstrated by objdump, because of missing _bfd_clear_contents bounds checking.
CVE-2018-18483	The get_count function in cplus-dem.c in GNU libiberty, as distributed in GNU Binutils 2.31, allows remote attackers to cause a denial of service (malloc called with the result of an integer-overflowing calculation) or possibly have unspecified other impact via a crafted string, as demonstrated by c++filt.
CVE-2018-18484	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. Stack Exhaustion occurs in the C++ demangling functions provided by libiberty, and there is a stack consumption problem caused by recursive stack frames: cplus_demangle_type, d_bare_function_type, d_function_type.
CVE-2018-18605	A heap-based buffer over-read issue was discovered in the function sec_merge_hash_lookup in merge.c in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.31, because _bfd_add_merge_section mishandles section merges when size is not a multiple of entsize. A specially crafted ELF allows remote attackers to cause a denial of service, as demonstrated by ld.
CVE-2018-18606	An issue was discovered in the merge_strings function in merge.c in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.31. There is a NULL pointer dereference in _bfd_add_merge_section when attempting to merge sections with large alignments. A specially crafted ELF allows remote attackers to cause a denial of service, as demonstrated by ld.
CVE-2018-18607	An issue was discovered in elf_link_input_bfd in elflink.c in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.31. There is a NULL pointer dereference in elf_link_input_bfd when used for finding STT_TLS symbols without any TLS section. A specially crafted ELF allows remote attackers to cause a denial of service, as demonstrated by ld.
CVE-2018-18700	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a stack consumption vulnerability resulting from infinite recursion in the functions d_name(), d_encoding(), and d_local_name() in cp-demangle.c. Remote attackers could leverage this vulnerability to cause a denial-of-service via an ELF file, as demonstrated by nm.
CVE-2018-18701	An issue was discovered in cp-demangle.c in GNU libiberty, as distributed in GNU Binutils 2.31. There is a stack consumption vulnerability resulting from infinite recursion in the functions next_is_type_qual() and cplus_demangle_type() in cp-demangle.c. Remote attackers could leverage this vulnerability to cause a denial-of-service via an ELF file, as demonstrated by nm.
CVE-2018-18873	An issue was discovered in JasPer 2.0.14. There is a NULL pointer dereference in the function ras_putdatastd in ras/ras_enc.c.
CVE-2018-19139	An issue has been found in JasPer 2.0.14. There is a memory leak in jas_malloc.c when called from jpc_unk_getparms in jpc_cs.c.
CVE-2018-19210	In LibTIFF 4.0.9, there is a NULL pointer dereference in the TIFFWriteDirectorySec function in tif_dirwrite.c that will lead to a denial of service attack, as demonstrated by tiffset.
CVE-2018-19211	In ncurses 6.1, there is a NULL pointer dereference at function _nc_parse_entry in parse_entry.c that will lead to a denial of service attack. The product proceeds to the dereference code path even after a "dubious character '*' in name or alias field" detection.
CVE-2018-19217	In ncurses, possibly a 6.x version, there is a NULL pointer dereference at the function _nc_name_match that will lead to a denial of service attack. NOTE: the original report stated version 6.1, but the issue did not reproduce for that version according to the maintainer or a reliable third-party
CVE-2018-19539	An issue was discovered in JasPer 2.0.14. There is an access violation in the function jas_image_readcmpt in libjasper/base/jas_image.c, leading to a denial of service.
CVE-2018-19540	An issue was discovered in JasPer 1.900.8, 1.900.9, 1.900.10, 1.900.11, 1.900.12, 1.900.13, 1.900.14, 1.900.15, 1.900.16, 1.900.17, 1.900.18, 1.900.19, 1.900.20, 1.900.21, 1.900.22, 1.900.23, 1.900.24, 1.900.25, 1.900.26, 1.900.27, 1.900.28, 1.900.29, 1.900.30, 1.900.31, 2.0.0, 2.0.1, 2.0.2, 2.0.3, 2.0.4, 2.0.5, 2.0.6, 2.0.7, 2.0.8, 2.0.9, 2.0.10, 2.0.11, 2.0.12, 2.0.13, 2.0.14, 2.0.15, 2.0.16. There is a heap-based buffer overflow of size 1 in the function jas_icctxtdesc_input in libjasper/base/jas_icc.c.

CVE ID	Description
CVE-2018-19541	An issue was discovered in JasPer 1.900.8, 1.900.9, 1.900.10, 1.900.11, 1.900.12, 1.900.13, 1.900.14, 1.900.15, 1.900.16, 1.900.17, 1.900.18, 1.900.19, 1.900.20, 1.900.21, 1.900.22, 1.900.23, 1.900.24, 1.900.25, 1.900.26, 1.900.27, 1.900.28, 1.900.29, 1.900.30, 1.900.31, 2.0.0, 2.0.1, 2.0.2, 2.0.3, 2.0.4, 2.0.5, 2.0.6, 2.0.7, 2.0.8, 2.0.9, 2.0.10, 2.0.11, 2.0.12, 2.0.13, 2.0.14, 2.0.15, 2.0.16. There is a heap-based buffer over-read of size 8 in the function <code>jas_image_depalettize</code> in <code>libjasper/base/jas_image.c</code> .
CVE-2018-19542	An issue was discovered in JasPer 2.0.14. There is a NULL pointer dereference in the function <code>jp2_decode</code> in <code>libjasper/jp2/jp2_dec.c</code> , leading to a denial of service.
CVE-2018-19543	An issue was discovered in JasPer 2.0.14. There is a heap-based buffer over-read of size 8 in the function <code>jp2_decode</code> in <code>libjasper/jp2/jp2_dec.c</code> .
CVE-2018-19876	cairo 1.16.0, in <code>cairo_ft_apply_variations()</code> in <code>cairo-ft-font.c</code> , would free memory using a free function incompatible with WebKit's <code>fastMalloc</code> , leading to an application crash with a "free(): invalid pointer" error.
CVE-2018-19932	An issue was discovered in the Binary File Descriptor (BFD) library (aka <code>libbfd</code>), as distributed in GNU Binutils through 2.31. There is an integer overflow and infinite loop caused by the <code>IS_CONTAINED_BY_LMA</code> macro in <code>elf.c</code> .
CVE-2018-20002	The <code>_bfd_generic_read_minisymbols</code> function in <code>syms.c</code> in the Binary File Descriptor (BFD) library (aka <code>libbfd</code>), as distributed in GNU Binutils 2.31, has a memory leak via a crafted ELF file, leading to a denial of service (memory consumption), as demonstrated by <code>nm</code> .
CVE-2018-20225	An issue was discovered in <code>pip</code> (all versions) because it installs the version with the highest version number, even if the user had intended to obtain a private package from a private index. This only affects use of the <code>--extra-index-url</code> option, and exploitation requires that the package does not already exist in the public index (and thus the attacker can put the package there with an arbitrary version number). NOTE: it has been reported that this is intended functionality and the user is responsible for using <code>--extra-index-url</code> securely
CVE-2018-20570	<code>jp2_encode</code> in <code>jp2/jp2_enc.c</code> in JasPer 2.0.14 has a heap-based buffer over-read.
CVE-2018-20622	JasPer 2.0.14 has a memory leak in <code>base/jas_malloc.c</code> in <code>libjasper.a</code> when <code>"--output-format jp2"</code> is used.
CVE-2018-20623	In GNU Binutils 2.31.1, there is a use-after-free in the error function in <code>elfcomm.c</code> when called from the <code>process_archive</code> function in <code>readelf.c</code> via a crafted ELF file.
CVE-2018-20651	A NULL pointer dereference was discovered in <code>elf_link_add_object_symbols</code> in <code>elflink.c</code> in the Binary File Descriptor (BFD) library (aka <code>libbfd</code>), as distributed in GNU Binutils 2.31.1. This occurs for a crafted ET_DYN with no program headers. A specially crafted ELF file allows remote attackers to cause a denial of service, as demonstrated by <code>ld</code> .
CVE-2018-20657	The <code>demangle_template</code> function in <code>cplus-dem.c</code> in GNU <code>libiberty</code> , as distributed in GNU Binutils 2.31.1, has a memory leak via a crafted string, leading to a denial of service (memory consumption), as demonstrated by <code>cxxfilt</code> , a related issue to CVE-2018-12698.
CVE-2018-20671	<code>load_specific_debug_section</code> in <code>objdump.c</code> in GNU Binutils through 2.31.1 contains an integer overflow vulnerability that can trigger a heap-based buffer overflow via a crafted section size.
CVE-2018-20673	The <code>demangle_template</code> function in <code>cplus-dem.c</code> in GNU <code>libiberty</code> , as distributed in GNU Binutils 2.31.1, contains an integer overflow vulnerability (for "Create an array for saving the template argument values") that can trigger a heap-based buffer overflow, as demonstrated by <code>nm</code> .
CVE-2018-20786	<code>libvterm</code> through 0+bzr726, as used in Vim and other products, mishandles certain out-of-memory conditions, leading to a denial of service (application crash), related to <code>screen.c</code> , <code>state.c</code> , and <code>vterm.c</code> .
CVE-2018-20796	In the GNU C Library (aka <code>glibc</code> or <code>libc6</code>) through 2.29, <code>check_dst_limits_calc_pos_1</code> in <code>posix/regexec.c</code> has Uncontrolled Recursion, as demonstrated by <code>'(\\227)(\\1\\1 t1\\2537)+'</code> in <code>grep</code> .
CVE-2018-20839	<code>systemd</code> 242 changes the VT1 mode upon a logout, which allows attackers to read cleartext passwords in certain circumstances, such as watching a shutdown, or using Ctrl-Alt-F1 and Ctrl-Alt-F2. This occurs because the KDGKBMODE (aka current keyboard mode) check is mishandled.
CVE-2018-5360	LibTIFF before 4.0.6 mishandles the reading of TIFF files, as demonstrated by a heap-based buffer over-read in the <code>ReadTIFFImage</code> function in <code>coders/tiff.c</code> in <code>GraphicsMagick</code> 1.3.27.
CVE-2018-5407	Simultaneous Multi-threading (SMT) in processors can enable local users to exploit software vulnerable to timing attacks via a side-channel timing attack on 'port contention'.

CVE ID	Description
CVE-2018-5709	An issue was discovered in MIT Kerberos 5 (aka krb5) through 1.16. There is a variable "dbentry->n_key_data" in kadmin/dbutil/dump.c that can store 16-bit data but unknowingly the developer has assigned a "u4" variable to it, which is for 32-bit data. An attacker can use this vulnerability to affect other artifacts of the database as we know that a Kerberos database dump file contains trusted data.
CVE-2018-5711	gd_gif_in.c in the GD Graphics Library (aka libgd), as used in PHP before 5.6.33, 7.0.x before 7.0.27, 7.1.x before 7.1.13, and 7.2.x before 7.2.1, has an integer signedness error that leads to an infinite loop via a crafted GIF file, as demonstrated by a call to the imagecreatefromgif or imagecreatefromstring PHP function. This is related to GetCode_ and gdImageCreateFromGifCtx.
CVE-2018-6829	cipher/elgamal.c in Libgcrypt through 1.8.2, when used to encrypt messages directly, improperly encodes plaintexts, which allows attackers to obtain sensitive information by reading ciphertext data (i.e., it does not have semantic security in face of a ciphertext-only attack). The Decisional Diffie-Hellman (DDH) assumption does not hold for Libgcrypt's ElGamal implementation.
CVE-2018-6872	The elf_parse_notes function in elf.c in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.30, allows remote attackers to cause a denial of service (out-of-bounds read and segmentation violation) via a note with a large alignment.
CVE-2018-6952	A double free exists in the another_hunk function in pch.c in GNU patch through 2.7.6.
CVE-2018-7273	In the Linux kernel through 4.15.4, the floppy driver reveals the addresses of kernel functions and global variables using printk calls within the function show_floppy in drivers/block/floppy.c. An attacker can read this information from dmesg and use the addresses to find the locations of kernel code and data and bypass kernel security protections such as KASLR.
CVE-2018-8043	The unimac_mdio_probe function in drivers/net/phy/mdio-bcm-unimac.c in the Linux kernel through 4.15.8 does not validate certain resource availability, which allows local users to cause a denial of service (NULL pointer dereference).
CVE-2018-9055	JasPer 2.0.14 allows denial of service via a reachable assertion in the function jpc_firstone in libjasper/jpc/jpc_math.c.
CVE-2018-9252	JasPer 2.0.14 allows denial of service via a reachable assertion in the function jpc_abstorelstepsize in libjasper/jpc/jpc_enc.c.
CVE-2019-1010022	GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass stack guard protection. The component is: nptl. The attack vector is: Exploit stack buffer overflow vulnerability and use this bypass vulnerability to bypass stack guard. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."
CVE-2019-1010023	GNU Libc current is affected by: Re-mapping current loaded library with malicious ELF file. The impact is: In worst case attacker may evaluate privileges. The component is: libld. The attack vector is: Attacker sends 2 ELF files to victim and asks to run ldd on it. ldd execute code. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."
CVE-2019-1010024	GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may bypass ASLR using cache of thread stack and heap. The component is: glibc. NOTE: Upstream comments indicate "this is being treated as a non-security bug and no real threat."
CVE-2019-1010025	GNU Libc current is affected by: Mitigation bypass. The impact is: Attacker may guess the heap addresses of pthread_created thread. The component is: glibc. NOTE: the vendor's position is "ASLR bypass itself is not a vulnerability."
CVE-2019-11038	When using the gdImageCreateFromXbm() function in the GD Graphics Library (aka LibGD) 2.2.5, as used in the PHP GD extension in PHP versions 7.1.x below 7.1.30, 7.2.x below 7.2.19 and 7.3.x below 7.3.6, it is possible to supply data that will cause the function to use the value of uninitialized variable. This may lead to disclosing contents of the stack that has been left there by previous code.
CVE-2019-11358	jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.
CVE-2019-12380	**DISPUTED** An issue was discovered in the efi subsystem in the Linux kernel through 5.1.5. phys_efi_set_virtual_address_map in arch/x86/platform/efi/efi.c and efi_call_phys_prolog in arch/x86/platform/efi/efi_64.c mishandle memory allocation failures. NOTE: This id is disputed as not being an issue because "All the code touched by the referenced commit runs only at boot, before any user processes are started. Therefore, there is no possibility for an unprivileged user to control it."

CVE ID	Description
CVE-2019-12904	In Libgcrypt 1.8.4, the C implementation of AES is vulnerable to a flush-and-reload side-channel attack because physical addresses are available to other processes. (The C implementation is used on platforms where an assembly-language implementation is unavailable.) NOTE: the vendor's position is that the issue report cannot be validated because there is no description of an attack
CVE-2019-12972	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.32. There is a heap-based buffer over-read in <code>_bfd_doprnt</code> in <code>bfd.c</code> because <code>elf_object_p</code> in <code>elfcode.h</code> mishandles an <code>e_shstrndx</code> section of type <code>SHT_GROUP</code> by omitting a trailing <code>'\0'</code> character.
CVE-2019-13117	In <code>numbers.c</code> in <code>libxslt 1.1.33</code> , an <code>xsl:number</code> with certain format strings could lead to a uninitialized read in <code>xsltNumberFormatInsertNumbers</code> . This could allow an attacker to discern whether a byte on the stack contains the characters A, a, I, i, or 0, or any other character.
CVE-2019-13118	In <code>numbers.c</code> in <code>libxslt 1.1.33</code> , a type holding grouping characters of an <code>xsl:number</code> instruction was too narrow and an invalid character/length combination could be passed to <code>xsltNumberFormatDecimal</code> , leading to a read of uninitialized stack data.
CVE-2019-14250	An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.32. <code>simple_object_elf_match</code> in <code>simple-object-elf.c</code> does not check for a zero <code>shstrndx</code> value, leading to an integer overflow and resultant heap-based buffer overflow.
CVE-2019-14899	A vulnerability was discovered in Linux, FreeBSD, OpenBSD, MacOS, iOS, and Android that allows a malicious access point, or an adjacent user, to determine if a connected user is using a VPN, make positive inferences about the websites they are visiting, and determine the correct sequence and acknowledgement numbers in use, allowing the bad actor to inject data into the TCP stream. This provides everything that is needed for an attacker to hijack active connections inside the VPN tunnel.
CVE-2019-15213	An issue was discovered in the Linux kernel before 5.2.3. There is a use-after-free caused by a malicious USB device in the <code>drivers/media/usb/dvb-usb/dvb-usb-init.c</code> driver.
CVE-2019-15217	An issue was discovered in the Linux kernel before 5.2.3. There is a NULL pointer dereference caused by a malicious USB device in the <code>drivers/media/usb/zr364xx/zr364xx.c</code> driver.
CVE-2019-15218	An issue was discovered in the Linux kernel before 5.1.8. There is a NULL pointer dereference caused by a malicious USB device in the <code>drivers/media/usb/siano/smsusb.c</code> driver.
CVE-2019-15219	An issue was discovered in the Linux kernel before 5.1.8. There is a NULL pointer dereference caused by a malicious USB device in the <code>drivers/usb/misc/sisusbvga/sisusb.c</code> driver.
CVE-2019-15291	An issue was discovered in the Linux kernel through 5.2.9. There is a NULL pointer dereference caused by a malicious USB device in the <code>flexcop_usb_probe</code> function in the <code>drivers/media/usb/b2c2/flexcop-usb.c</code> driver.
CVE-2019-1547	Normally in OpenSSL EC groups always have a co-factor present and this is used in side channel resistant code paths. However, in some cases, it is possible to construct a group using explicit parameters (instead of using a named curve). In those cases it is possible that such a group does not have the cofactor present. This can occur even where all the parameters match a known named curve. If such a curve is used then OpenSSL falls back to non-side channel resistant code paths which may result in full key recovery during an ECDSA signature operation. In order to be vulnerable an attacker would have to have the ability to time the creation of a large number of signatures where explicit parameters with no co-factor present are in use by an application using libcrypto. For the avoidance of doubt libssl is not vulnerable because explicit parameters are never used. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s).
CVE-2019-1551	There is an overflow bug in the x64_64 Montgomery squaring procedure used in exponentiation with 512-bit moduli. No EC algorithms are affected. Analysis suggests that attacks against 2-prime RSA1024, 3-prime RSA1536, and DSA1024 as a result of this defect would be very difficult to perform and are not believed likely. Attacks against DH512 are considered just feasible. However, for an attack the target would have to re-use the DH512 private key, which is not recommended anyway. Also applications directly using the low level API <code>BN_mod_exp</code> may be affected if they use <code>BN_FLG_CONSTTIME</code> . Fixed in OpenSSL 1.1.1e (Affected 1.1.1-1.1.1d). Fixed in OpenSSL 1.0.2u (Affected 1.0.2-1.0.2t).

CVE ID	Description
CVE-2019-1552	OpenSSL has internal defaults for a directory tree where it can find a configuration file as well as certificates used for verification in TLS. This directory is most commonly referred to as OPENSSLDIR, and is configurable with the --prefix / --openssldir configuration options. For OpenSSL versions 1.1.0 and 1.1.1, the mingw configuration targets assume that resulting programs and libraries are installed in a Unix-like environment and the default prefix for program installation as well as for OPENSSLDIR should be '/usr/local'. However, mingw programs are Windows programs, and as such, find themselves looking at sub-directories of 'C:/usr/local', which may be world writable, which enables untrusted users to modify OpenSSL's default configuration, insert CA certificates, modify (or even replace) existing engine modules, etc. For OpenSSL 1.0.2, '/usr/local/ssl' is used as default for OPENSSLDIR on all Unix and Windows targets, including Visual C builds. However, some build instructions for the diverse Windows targets on 1.0.2 encourage you to specify your own --prefix. OpenSSL versions 1.1.1, 1.1.0 and 1.0.2 are affected by this issue. Due to the limited scope of affected deployments this has been assessed as low severity and therefore we are not creating new releases at this time. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s).
CVE-2019-1559	If an application encounters a fatal protocol error and then calls SSL_shutdown() twice (once to send a close_notify, and once to receive one) then OpenSSL can respond differently to the calling application if a 0 byte record is received with invalid padding compared to if a 0 byte record is received with an invalid MAC. If the application then behaves differently based on that in a way that is detectable to the remote peer, then this amounts to a padding oracle that could be used to decrypt data. In order for this to be exploitable "non-stitched" ciphersuites must be in use. Stitched ciphersuites are optimised implementations of certain commonly used ciphersuites. Also the application must call SSL_shutdown() twice even if a protocol error has occurred (applications should not do this but some do anyway). Fixed in OpenSSL 1.0.2r (Affected 1.0.2-1.0.2q).
CVE-2019-1563	In situations where an attacker receives automated notification of the success or failure of a decryption attempt an attacker, after sending a very large number of messages to be decrypted, can recover a CMS/ PKCS7 transported encryption key or decrypt any RSA encrypted message that was encrypted with the public RSA key, using a Bleichenbacher padding oracle attack. Applications are not affected if they use a certificate together with the private RSA key to the CMS_decrypt or PKCS7_decrypt functions to select the correct recipient info to decrypt. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s).
CVE-2019-15794	Overlayfs in the Linux kernel and shifts, a non-upstream patch to the Linux kernel included in the Ubuntu 5.0 and 5.3 kernel series, both replace vma->vm_file in their mmap handlers. On error the original value is not restored, and the reference is put for the file to which vm_file points. On upstream kernels this is not an issue, as no callers dereference vm_file following after call_mmap() returns an error. However, the aufs patches change mmap_region() to replace the fput() using a local variable with vma_fput(), which will fput() vm_file, leading to a refcount underflow.
CVE-2019-15807	In the Linux kernel before 5.1.13, there is a memory leak in drivers/scsi/libsas/sas_expander.c when SAS expander discovery fails. This will cause a BUG and denial of service.
CVE-2019-16089	An issue was discovered in the Linux kernel through 5.2.13. nbd_genl_status in drivers/block/nbd.c does not check the nla_nest_start_noflag return value.
CVE-2019-16229	drivers/gpu/drm/amd/amdkfd/kfd_interrupt.c in the Linux kernel 5.2.14 does not check the alloc_workqueue return value, leading to a NULL pointer dereference. NOTE: The security community disputes this issues as not being serious enough to be deserving a CVE id
CVE-2019-16230	drivers/gpu/drm/radeon/radeon_display.c in the Linux kernel 5.2.14 does not check the alloc_workqueue return value, leading to a NULL pointer dereference. NOTE: A third-party software maintainer states that the work queue allocation is happening during device initialization, which for a graphics card occurs during boot. It is not attacker controllable and OOM at that time is highly unlikely
CVE-2019-17113	In libopenmpt before 0.3.19 and 0.4.x before 0.4.9, ModPlug_InstrumentName and ModPlug_SampleName in libopenmpt_modplug.c do not restrict the lengths of libmodplug output-buffer strings in the C API, leading to a buffer overflow.
CVE-2019-17567	Apache HTTP Server versions 2.4.6 to 2.4.46 mod_proxy_wstunnel configured on an URL that is not necessarily Upgraded by the origin server was tunneling the whole connection regardless, thus allowing for subsequent requests on the same connection to pass through with no HTTP validation, authentication or authorization possibly configured.
CVE-2019-18806	A memory leak in the ql_alloc_large_buffers() function in drivers/net/ethernet/qlogic/qla3xxx.c in the Linux kernel before 5.3.5 allows local users to cause a denial of service (memory consumption) by triggering pci_dma_mapping_error() failures, aka CID-1acb8f2a7a9f.

CVE ID	Description
CVE-2019-19039	<code>__btrfs_free_extent</code> in <code>fs/btrfs/extent-tree.c</code> in the Linux kernel through 5.3.12 calls <code>btrfs_print_leaf</code> in a certain <code>ENOENT</code> case, which allows local users to obtain potentially sensitive information about register values via the <code>dmesg</code> program. NOTE: The BTRFS development team disputes this issues as not being a vulnerability because “1) The kernel provide facilities to restrict access to <code>dmesg</code> - <code>dmesg_restrict=1</code> <code>sysctl</code> option. So it's really up to the system administrator to judge whether <code>dmesg</code> access shall be disallowed or not. 2) <code>WARN/WARN_ON</code> are widely used macros in the linux kernel. If this CVE is considered valid this would mean there are literally thousands CVE lurking in the kernel - something which clearly is not the case.
CVE-2019-19054	A memory leak in the <code>cx23888_ir_probe()</code> function in <code>drivers/media/pci/cx23885/cx23888-ir.c</code> in the Linux kernel through 5.3.11 allows attackers to cause a denial of service (memory consumption) by triggering <code>kfifo_alloc()</code> failures, aka CID-a7b2df76b42b.
CVE-2019-19066	A memory leak in the <code>bfa_dm_get_stats()</code> function in <code>drivers/scsi/bfa/bfa_attr.c</code> in the Linux kernel through 5.3.11 allows attackers to cause a denial of service (memory consumption) by triggering <code>bfa_port_get_stats()</code> failures, aka CID-0e62395da2bd.
CVE-2019-19076	A memory leak in the <code>nfp_abm_u32_knode_replace()</code> function in <code>drivers/net/ethernet/netronome/nfp/abm/cls.c</code> in the Linux kernel before 5.3.6 allows attackers to cause a denial of service (memory consumption), aka CID-78beef629fd9. NOTE: This has been argued as not a valid vulnerability. The upstream commit 78beef629fd9 was reverted
CVE-2019-19080	Four memory leaks in the <code>nfp_flower_spawn_phy_reprs()</code> function in <code>drivers/net/ethernet/netronome/nfp/flower/main.c</code> in the Linux kernel before 5.3.4 allow attackers to cause a denial of service (memory consumption), aka CID-8572cea1461a.
CVE-2019-19081	A memory leak in the <code>nfp_flower_spawn_vnic_reprs()</code> function in <code>drivers/net/ethernet/netronome/nfp/flower/main.c</code> in the Linux kernel before 5.3.4 allows attackers to cause a denial of service (memory consumption), aka CID-8ce39eb5a67a.
CVE-2019-19082	Memory leaks in <code>*create_resource_pool()</code> functions under <code>drivers/gpu/drm/amd/display/dc</code> in the Linux kernel through 5.3.11 allow attackers to cause a denial of service (memory consumption). This affects the <code>dce120_create_resource_pool()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce120/dce120_resource.c</code> , the <code>dce110_create_resource_pool()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce110/dce110_resource.c</code> , the <code>dce100_create_resource_pool()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce100/dce100_resource.c</code> , the <code>dcn10_create_resource_pool()</code> function in <code>drivers/gpu/drm/amd/display/dc/dcn10/dcn10_resource.c</code> , and the <code>dce112_create_resource_pool()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce112/dce112_resource.c</code> , aka CID-104c307147ad.
CVE-2019-19083	Memory leaks in <code>*clock_source_create()</code> functions under <code>drivers/gpu/drm/amd/display/dc</code> in the Linux kernel before 5.3.8 allow attackers to cause a denial of service (memory consumption). This affects the <code>dce112_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce112/dce112_resource.c</code> , the <code>dce100_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce100/dce100_resource.c</code> , the <code>dcn10_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dcn10/dcn10_resource.c</code> , the <code>dcn20_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dcn20/dcn20_resource.c</code> , the <code>dce120_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce120/dce120_resource.c</code> , the <code>dce110_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce110/dce110_resource.c</code> , and the <code>dce80_clock_source_create()</code> function in <code>drivers/gpu/drm/amd/display/dc/dce80/dce80_resource.c</code> , aka CID-055e547478a1.
CVE-2019-19244	<code>sqlite3Select</code> in <code>select.c</code> in SQLite 3.30.1 allows a crash if a sub-select uses both <code>DISTINCT</code> and window functions, and also has certain <code>ORDER BY</code> usage.
CVE-2019-19378	In the Linux kernel 5.0.21, mounting a crafted <code>btrfs</code> filesystem image can lead to slab-out-of-bounds write access in <code>index_rbio_pages</code> in <code>fs/btrfs/raid56.c</code> .
CVE-2019-19462	<code>relay_open</code> in <code>kernel/relay.c</code> in the Linux kernel through 5.4.1 allows local users to cause a denial of service (such as relay blockage) by triggering a <code>NULL alloc_percpu</code> result.
CVE-2019-19530	In the Linux kernel before 5.2.10, there is a use-after-free bug that can be caused by a malicious USB device in the <code>drivers/usb/class/cdc-acm.c</code> driver, aka CID-c52873e5a1ef.
CVE-2019-19814	In the Linux kernel 5.0.21, mounting a crafted <code>f2fs</code> filesystem image can cause <code>__remove_dirty_segment</code> slab-out-of-bounds write access because an array is bounded by the number of dirty types (8) but the array index can exceed this.
CVE-2019-19965	In the Linux kernel through 5.4.6, there is a <code>NULL</code> pointer dereference in <code>drivers/scsi/libsas/sas_discover.c</code> because of mishandling of port disconnection during discovery, related to a PHY down race condition, aka CID-f70267f379b5.

CVE ID	Description
CVE-2019-20095	mwifiex_tm_cmd in drivers/net/wireless/marvell/mwifiex/cfg80211.c in the Linux kernel before 5.1.6 has some error-handling cases that did not free allocated hostcmd memory, aka CID-003b686ace82. This will cause a memory leak and denial of service.
CVE-2019-20426	In the Lustre file system before 2.12.3, the ptlrpc module has an out-of-bounds access and panic due to the lack of validation for specific fields of packets sent by a client. In the function ldlm_cancel_hpreq_check, there is no lock_count bounds check.
CVE-2019-20633	GNU patch through 2.7.6 contains a free(p_line[p_end]) Double Free vulnerability in the function another_hunk in pch.c that can cause a denial of service via a crafted patch file. NOTE: this issue exists because of an incomplete fix for CVE-2018-6952.
CVE-2019-20794	An issue was discovered in the Linux kernel 4.18 through 5.6.11 when unprivileged user namespaces are allowed. A user can create their own PID namespace, and mount a FUSE filesystem. Upon interaction with this FUSE filesystem, if the userspace component is terminated via a kill of the PID namespace's pid 1, it will result in a hung task, and resources being permanently locked up until system reboot. This can result in resource exhaustion.
CVE-2019-20838	libpcre in PCRE before 8.43 allows a subject buffer over-read in JIT when UTF is disabled, and \X or \R has more than one fixed quantifier, a related issue to CVE-2019-20454.
CVE-2019-2201	In generate_jsimd_ycc_rgb_convert_neon of jsimd_arm64_neon.S, there is a possible out of bounds write due to a missing bounds check. This could lead to remote code execution in an unprivileged process with no additional execution privileges needed. User interaction is needed for exploitation. Product: Android Versions: Android-8.0 Android-8.1 Android-9 Android-10 Android ID: A-120551338
CVE-2019-25160	In the Linux kernel, the following vulnerability has been resolved: netlabel: fix out-of-bounds memory accesses There are two array out-of-bounds memory accesses, one in cipso_v4_map_lvl_valid(), the other in netlbl_bitmap_walk(). Both errors are embarrassingly simple, and the fixes are straightforward. As a FYI for anyone backporting this patch to kernels prior to v4.8, you'll want to apply the netlbl_bitmap_walk() patch to cipso_v4_bitmap_walk() as netlbl_bitmap_walk() doesn't exist before Linux v4.8.
CVE-2019-6110	In OpenSSH 7.9, due to accepting and displaying arbitrary stderr output from the server, a malicious server (or Man-in-The-Middle attacker) can manipulate the client output, for example to use ANSI control codes to hide additional files being transferred.
CVE-2019-6128	The TIFFFdOpen function in tif_unix.c in LibTIFF 4.0.10 has a memory leak, as demonstrated by pal2rgb.
CVE-2019-6461	An issue was discovered in cairo 1.16.0. There is an assertion problem in the function _cairo_arc_in_direction in the file cairo-arc.c.
CVE-2019-6462	An issue was discovered in cairo 1.16.0. There is an infinite loop in the function _arc_error_normalized in the file cairo-arc.c, related to _arc_max_angle_for_tolerance_normalized.
CVE-2019-6988	An issue was discovered in OpenJPEG 2.3.0. It allows remote attackers to cause a denial of service (attempted excessive memory allocation) in opj_calloc in openjp2/opj_malloc.c, when called from opj_tcd_init_tile in openjp2/tcd.c, as demonstrated by the 64-bit opj_decompress.
CVE-2019-7317	png_image_free in png.c in libpng 1.6.x before 1.6.37 has a use-after-free because png_image_free_function is called under png_safe_execute.
CVE-2019-8457	SQLite3 from 3.6.0 to and including 3.27.2 is vulnerable to heap out-of-bound read in the rtreenode() function when handling invalid rtree tables.
CVE-2019-8905	do_core_note in readelf.c in libmagic.a in file 5.35 has a stack-based buffer over-read, related to file_printable, a different vulnerability than CVE-2018-10360.
CVE-2019-8906	do_core_note in readelf.c in libmagic.a in file 5.35 has an out-of-bounds read because memcpy is misused.
CVE-2019-9071	An issue was discovered in GNU libiberty, as distributed in GNU Binutils 2.32. It is a stack consumption issue in d_count_templates_scopes in cp-demangle.c after many recursive calls.
CVE-2019-9075	An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils 2.32. It is a heap-based buffer overflow in _bfd_archive_64_bit_slurp_armap in archive64.c.
CVE-2019-9077	An issue was discovered in GNU Binutils 2.32. It is a heap-based buffer overflow in process_mips_specific in readelf.c via a malformed MIPS option section.

CVE ID	Description
CVE-2019-9192	In the GNU C Library (aka glibc or libc6) through 2.29, <code>check_dst_limits_calc_pos_1</code> in <code>posix/regexec.c</code> has Uncontrolled Recursion, as demonstrated by <code>'(\)(\\1\\1)*'</code> in <code>grep</code> , a different issue than CVE-2018-20796. NOTE: the software maintainer disputes that this is a vulnerability because the behavior occurs only with a crafted pattern
CVE-2019-9674	<code>Lib/zipfile.py</code> in Python through 3.7.2 allows remote attackers to cause a denial of service (resource consumption) via a ZIP bomb.
CVE-2019-9923	<code>pax_decode_header</code> in <code>sparse.c</code> in GNU Tar before 1.32 had a NULL pointer dereference when parsing certain archives that have malformed extended headers.
CVE-2019-9936	In SQLite 3.27.2, running <code>fts5</code> prefix queries inside a transaction could trigger a heap-based buffer over-read in <code>fts5HashEntrySort</code> in <code>sqlite3.c</code> , which may lead to an information leak. This is related to <code>ext/fts5/fts5_hash.c</code> .
CVE-2019-9937	In SQLite 3.27.2, interleaving reads and writes in a single transaction with an <code>fts5</code> virtual table will lead to a NULL Pointer Dereference in <code>fts5ChunkIterate</code> in <code>sqlite3.c</code> . This is related to <code>ext/fts5/fts5_hash.c</code> and <code>ext/fts5/fts5_index.c</code> .
CVE-2020-0499	In <code>FLAC__bitreader_read_rice_signed_block</code> of <code>bitreader.c</code> , there is a possible out of bounds read due to a heap buffer overflow. This could lead to remote information disclosure with no additional execution privileges needed. User interaction is needed for exploitation. Product: Android Versions: Android-11 Android ID: A-156076070
CVE-2020-10135	Legacy pairing and secure-connections pairing authentication in Bluetooth BR/EDR Core Specification v5.2 and earlier may allow an unauthenticated user to complete authentication without pairing credentials via adjacent access. An unauthenticated, adjacent attacker could impersonate a Bluetooth BR/EDR master or slave to pair with a previously paired remote device to successfully complete the authentication procedure without knowing the link key.
CVE-2020-10663	The JSON gem through 2.2.0 for Ruby, as used in Ruby 2.4 through 2.4.9, 2.5 through 2.5.7, and 2.6 through 2.6.5, has an Unsafe Object Creation Vulnerability. This is quite similar to CVE-2013-0269, but does not rely on poor garbage-collection behavior within Ruby. Specifically, use of JSON parsing methods can lead to creation of a malicious object within the interpreter, with adverse effects that are application-dependent.
CVE-2020-10693	A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.
CVE-2020-10781	A flaw was found in the Linux Kernel before 5.8-rc6 in the ZRAM kernel module, where a user with a local account and the ability to read the <code>/sys/class/zram-control/hot_add</code> file can create ZRAM device nodes in the <code>/dev/</code> directory. This read allocates kernel memory and is not accounted for a user that triggers the creation of that ZRAM device. With this vulnerability, continually reading the device may consume a large amount of system memory and cause the Out-of-Memory (OOM) killer to activate and terminate random userspace processes, possibly making the system inoperable.
CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. <code>.html()</code> , <code>.append()</code> , and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.
CVE-2020-11023	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <code><option></code> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. <code>.html()</code> , <code>.append()</code> , and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.
CVE-2020-11494	An issue was discovered in <code>slc_bump</code> in <code>drivers/net/can/slcanc.c</code> in the Linux kernel 3.16 through 5.6.2. It allows attackers to read uninitialized <code>can_frame</code> data, potentially containing sensitive information from kernel stack memory, if the configuration lacks <code>CONFIG_INIT_STACK_ALL</code> , aka CID-b9258a2cece4.
CVE-2020-11609	An issue was discovered in the <code>stv06xx</code> subsystem in the Linux kernel before 5.6.1. <code>drivers/media/usb/gspca/stv06xx/stv06xx.c</code> and <code>drivers/media/usb/gspca/stv06xx/stv06xx_pb0100.c</code> mishandle invalid descriptors, as demonstrated by a NULL pointer dereference, aka CID-485b06aadb93.
CVE-2020-12413	The Raccoon attack is a timing attack on DHE ciphersuites inherit in the TLS specification. To mitigate this vulnerability, Firefox disabled support for DHE ciphersuites.
CVE-2020-12480	In Play Framework 2.6.0 through 2.8.1, the CSRF filter can be bypassed by making CORS simple requests with content types that contain parameters that can't be parsed.

CVE ID	Description
CVE-2020-12656	gss_mech_free in net/sunrpc/auth_gss/gss_mech_switch.c in the rpcsec_gss_krb5 implementation in the Linux kernel through 5.6.10 lacks certain domain_release calls, leading to a memory leak. Note: This was disputed with the assertion that the issue does not grant any access not already available. It is a problem that on unloading a specific kernel module some memory is leaked, but loading kernel modules is a privileged operation. A user could also write a kernel module to consume any amount of memory they like and load that replicating the effect of this bug
CVE-2020-12768	An issue was discovered in the Linux kernel before 5.6. svm_cpu_uninit in arch/x86/kvm/svm.c has a memory leak, aka CID-d80b64ff297e. NOTE: third parties dispute this issue because it's a one-time leak at the boot, the size is negligible, and it can't be triggered at will
CVE-2020-12769	An issue was discovered in the Linux kernel before 5.4.17. drivers/spi/spi-dw.c allows attackers to cause a panic via concurrent calls to dw_spi_irq and dw_spi_transfer_one, aka CID-19b61392c5a8.
CVE-2020-13529	An exploitable denial-of-service vulnerability exists in Systemd 245. A specially crafted DHCP FORCERENEW packet can cause a server running the DHCP client to be vulnerable to a DHCP ACK spoofing attack. An attacker can forge a pair of FORCERENEW and DCHP ACK packets to reconfigure the server.
CVE-2020-13844	Arm Armv8-A core implementations utilizing speculative execution past unconditional changes in control flow may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis, aka "straight-line speculation."
CVE-2020-13956	Apache HttpClient versions prior to version 4.5.13 and 5.0.3 can misinterpret malformed authority component in request URIs passed to the library as java.net.URI object and pick the wrong target host for request execution.
CVE-2020-14304	A memory disclosure flaw was found in the Linux kernel's ethernet drivers, in the way it read data from the EEPROM of the device. This flaw allows a local user to read uninitialized values from the kernel memory. The highest threat from this vulnerability is to confidentiality.
CVE-2020-14356	A flaw null pointer dereference in the Linux kernel cgroupv2 subsystem in versions before 5.7.10 was found in the way when reboot the system. A local user could use this flaw to crash the system or escalate their privileges on the system.
CVE-2020-14390	A flaw was found in the Linux kernel in versions before 5.9-rc6. When changing screen size, an out-of-bounds memory write can occur leading to memory corruption or a denial of service. Due to the nature of the flaw, privilege escalation cannot be fully ruled out.
CVE-2020-14416	In the Linux kernel before 5.4.16, a race condition in tty->disc_data handling in the slip and slcan line discipline could lead to a use-after-free, aka CID-0ace17d56824. This affects drivers/net/slip/slip.c and drivers/net/can/slcan.c.
CVE-2020-15522	Bouncy Castle BC Java before 1.66, BC C# .NET before 1.8.7, BC-FJA before 1.0.1.2, 1.0.2.1, and BC-FNA before 1.0.1.1 have a timing issue within the EC math library that can expose information about the private key when an attacker is able to observe timing information for the generation of multiple deterministic ECDSA signatures.
CVE-2020-17049	A security feature bypass vulnerability exists in the way Key Distribution Center (KDC) determines if a service ticket can be used for delegation via Kerberos Constrained Delegation (KCD). To exploit the vulnerability, a compromised service that is configured to use KCD could tamper with a service ticket that is not valid for delegation to force the KDC to accept it. The update addresses this vulnerability by changing how the KDC validates service tickets used with KCD.
CVE-2020-18768	There exists one heap buffer overflow in _TIFFmemcpy in tif_unix.c in libtiff 4.0.10, which allows an attacker to cause a denial-of-service through a crafted tiff file.
CVE-2020-19185	Buffer Overflow vulnerability in one_one_mapping function in progs/dump_entry.c:1373 in ncurses 6.1 allows remote attackers to cause a denial of service via crafted command.
CVE-2020-19186	Buffer Overflow vulnerability in _nc_find_entry function in tinfo/comp_hash.c:66 in ncurses 6.1 allows remote attackers to cause a denial of service via crafted command.
CVE-2020-19187	Buffer Overflow vulnerability in fmt_entry function in progs/dump_entry.c:1100 in ncurses 6.1 allows remote attackers to cause a denial of service via crafted command.
CVE-2020-19188	Buffer Overflow vulnerability in fmt_entry function in progs/dump_entry.c:1116 in ncurses 6.1 allows remote attackers to cause a denial of service via crafted command.
CVE-2020-19189	Buffer Overflow vulnerability in postprocess_terminfo function in tinfo/parse_entry.c:997 in ncurses 6.1 allows remote attackers to cause a denial of service via crafted command.

CVE ID	Description
CVE-2020-19190	Buffer Overflow vulnerability in <code>_nc_find_entry</code> in <code>tinfo/comp_hash.c:70</code> in <code>ncurses 6.1</code> allows remote attackers to cause a denial of service via crafted command.
CVE-2020-1968	The Raccoon attack exploits a flaw in the TLS specification which can lead to an attacker being able to compute the pre-master secret in connections which have used a Diffie-Hellman (DH) based ciphersuite. In such a case this would result in the attacker being able to eavesdrop on all encrypted communications sent over that TLS connection. The attack can only be exploited if an implementation re-uses a DH secret across multiple TLS connections. Note that this issue only impacts DH ciphersuites and not ECDH ciphersuites. This issue affects OpenSSL 1.0.2 which is out of support and no longer receiving public updates. OpenSSL 1.1.1 is not vulnerable to this issue. Fixed in OpenSSL 1.0.2w (Affected 1.0.2-1.0.2v).
CVE-2020-1971	The X.509 GeneralName type is a generic type for representing different types of names. One of those name types is known as EDIPartyName. OpenSSL provides a function <code>GENERAL_NAME_cmp</code> which compares different instances of a <code>GENERAL_NAME</code> to see if they are equal or not. This function behaves incorrectly when both <code>GENERAL_NAME</code> s contain an <code>EDIPARTYNAME</code> . A NULL pointer dereference and a crash may occur leading to a possible denial of service attack. OpenSSL itself uses the <code>GENERAL_NAME_cmp</code> function for two purposes: 1) Comparing CRL distribution point names between an available CRL and a CRL distribution point embedded in an X509 certificate 2) When verifying that a timestamp response token signer matches the timestamp authority name (exposed via the API functions <code>TS_RESP_verify_response</code> and <code>TS_RESP_verify_token</code>) If an attacker can control both items being compared then that attacker could trigger a crash. For example if the attacker can trick a client or server into checking a malicious certificate against a malicious CRL then this may occur. Note that some applications automatically download CRLs based on a URL embedded in a certificate. This checking happens prior to the signatures on the certificate and CRL being verified. OpenSSL's <code>s_server</code> , <code>s_client</code> and <code>verify</code> tools have support for the <code>"-crl_download"</code> option which implements automatic CRL downloading and this attack has been demonstrated to work against those tools. Note that an unrelated bug means that affected versions of OpenSSL cannot parse or construct correct encodings of <code>EDIPARTYNAME</code> . However it is possible to construct a malformed <code>EDIPARTYNAME</code> that OpenSSL's parser will accept and hence trigger this attack. All OpenSSL 1.1.1 and 1.0.2 versions are affected by this issue. Other OpenSSL releases are out of support and have not been checked. Fixed in OpenSSL 1.1.1i (Affected 1.1.1-1.1.1h). Fixed in OpenSSL 1.0.2x (Affected 1.0.2-1.0.2w).
CVE-2020-20703	Buffer Overflow vulnerability in <code>VIM v.8.1.2135</code> allows a remote attacker to execute arbitrary code via the operand parameter.
CVE-2020-21469	An issue was discovered in PostgreSQL 12.2 allows attackers to cause a denial of service via repeatedly sending <code>SIGHUP</code> signals. NOTE: this is disputed by the vendor because untrusted users cannot send <code>SIGHUP</code> signals; they can only be sent by a PostgreSQL superuser, a user with <code>pg_reload_conf</code> access, or a user with sufficient privileges at the OS level (the postgres account or the root account).
CVE-2020-21514	An issue was discovered in Fluent Fluentd v.1.8.0 and Fluent-ui v.1.2.2 allows attackers to gain escalated privileges and execute arbitrary code due to a default password.
CVE-2020-21674	Heap-based buffer overflow in <code>archive_string_append_from_wcs()</code> (<code>archive_string.c</code>) in <code>libarchive-3.4.1dev</code> allows remote attackers to cause a denial of service (out-of-bounds write in heap memory resulting into a crash) via a crafted archive file. NOTE: this only affects users who downloaded the development code from GitHub. Users of the product's official releases are unaffected.
CVE-2020-25639	A NULL pointer dereference flaw was found in the Linux kernel's GPU Nouveau driver functionality in versions prior to 5.12-rc1 in the way the user calls <code>ioctl DRM_IOCTL_NOUVEAU_CHANNEL_ALLOC</code> . This flaw allows a local user to crash the system.
CVE-2020-25645	A flaw was found in the Linux kernel in versions before 5.9-rc7. Traffic between two Geneve endpoints may be unencrypted when IPsec is configured to encrypt traffic for the specific UDP port used by the GENEVE tunnel allowing anyone between the two endpoints to read the traffic unencrypted. The main threat from this vulnerability is to data confidentiality.
CVE-2020-25658	It was found that <code>python-rsa</code> is vulnerable to Bleichenbacher timing attacks. An attacker can use this flaw via the RSA decryption API to decrypt parts of the cipher text encrypted with RSA.
CVE-2020-26244	Python oic is a Python OpenID Connect implementation. In Python oic before version 1.2.1, there are several related cryptographic issues affecting client implementations that use the library. The issues are: 1) The <code>IdToken</code> signature algorithm was not checked automatically, but only if the expected algorithm was passed in as a kwarg. 2) JWA <code>'none'</code> algorithm was allowed in all flows. 3) <code>oic.consumer.Consumer.parse_authz</code> returns an unverified <code>IdToken</code> . The verification of the token was left to the discretion of the implementator. 4) <code>iat</code> claim was not checked for sanity (i.e. it could be in the future). These issues are patched in version 1.2.1.

CVE ID	Description
CVE-2020-26558	Bluetooth LE and BR/EDR secure pairing in Bluetooth Core Specification 2.1 through 5.2 may permit a nearby man-in-the-middle attacker to identify the Passkey used during pairing (in the Passkey authentication procedure) by reflection of the public key and the authentication evidence of the initiating device, potentially permitting this attacker to complete authenticated pairing with the responding device using the correct Passkey for the pairing session. The attack methodology determines the Passkey value one bit at a time.
CVE-2020-26939	In Legion of the Bouncy Castle BC before 1.61 and BC-FJA before 1.0.1.2, attackers can obtain sensitive information about a private exponent because of Observable Differences in Behavior to Error Inputs. This occurs in org.bouncycastle.crypto.encodings.OAEP_Encoding. Sending invalid ciphertext that decrypts to a short payload in the OAEP Decoder could result in the throwing of an early exception, potentially leaking some information about the private exponent of the RSA private key performing the encryption.
CVE-2020-27170	An issue was discovered in the Linux kernel before 5.11.8. kernel/bpf/verifier.c performs undesirable out-of-bounds speculation on pointer arithmetic, leading to side-channel attacks that defeat Spectre mitigations and obtain sensitive information from kernel memory, aka CID-f232326f6966. This affects pointer types that do not define a ptr_limit.
CVE-2020-27171	An issue was discovered in the Linux kernel before 5.11.8. kernel/bpf/verifier.c has an off-by-one error (with a resultant integer underflow) affecting out-of-bounds speculation on pointer arithmetic, leading to side-channel attacks that defeat Spectre mitigations and obtain sensitive information from kernel memory, aka CID-10d2bb2e6b1d.
CVE-2020-28097	The vgacon subsystem in the Linux kernel before 5.8.10 mishandles software scrollbar. There is a vgacon_scrolldelta out-of-bounds read, aka CID-973c096f6a85.
CVE-2020-29374	An issue was discovered in the Linux kernel before 5.7.3, related to mm/gup.c and mm/huge_memory.c. The get_user_pages (aka gup) implementation, when used for a copy-on-write page, does not properly consider the semantics of read operations and therefore can grant unintended write access, aka CID-17839856fd58.
CVE-2020-29582	In JetBrains Kotlin before 1.4.21, a vulnerable Java API was used for temporary file and folder creation. An attacker was able to read data from such files and list directories due to insecure permissions.
CVE-2020-35493	A flaw exists in binutils in bfd/pef.c. An attacker who is able to submit a crafted PEF file to be parsed by objdump could cause a heap buffer overflow -> out-of-bounds read that could lead to an impact to application availability. This flaw affects binutils versions prior to 2.34.
CVE-2020-35494	There's a flaw in binutils /opcodes/tic4x-dis.c. An attacker who is able to submit a crafted input file to be processed by binutils could cause usage of uninitialized memory. The highest threat is to application availability with a lower threat to data confidentiality. This flaw affects binutils versions prior to 2.34.
CVE-2020-35495	There's a flaw in binutils /bfd/pef.c. An attacker who is able to submit a crafted input file to be processed by the objdump program could cause a null pointer dereference. The greatest threat from this flaw is to application availability. This flaw affects binutils versions prior to 2.34.
CVE-2020-35496	There's a flaw in bfd_pef_scan_start_address() of bfd/pef.c in binutils which could allow an attacker who is able to submit a crafted file to be processed by objdump to cause a NULL pointer dereference. The greatest threat of this flaw is to application availability. This flaw affects binutils versions prior to 2.34.
CVE-2020-35501	A flaw was found in the Linux kernels implementation of audit rules, where a syscall can unexpectedly not be correctly not be logged by the audit subsystem
CVE-2020-35507	There's a flaw in bfd_pef_parse_function_stubs of bfd/pef.c in binutils in versions prior to 2.34 which could allow an attacker who is able to submit a crafted file to be processed by objdump to cause a NULL pointer dereference. The greatest threat of this flaw is to application availability.
CVE-2020-35512	A use-after-free flaw was found in D-Bus Development branch <= 1.13.16, dbus-1.12.x stable branch <= 1.12.18, and dbus-1.10.x and older branches <= 1.10.30 when a system has multiple usernames sharing the same UID. When a set of policy rules references these usernames, D-Bus may free some memory in the heap, which is still used by data structures necessary for the other usernames sharing the UID, possibly leading to a crash or other undefined behaviors
CVE-2020-35538	A crafted input file could cause a null pointer dereference in jcopy_sample_rows() when processed by libjpeg-turbo.
CVE-2020-36311	An issue was discovered in the Linux kernel before 5.9. arch/x86/kvm/svm/sev.c allows attackers to cause a denial of service (soft lockup) by triggering destruction of a large SEV VM (which requires unregistering many encrypted regions), aka CID-7be74942f184.

CVE ID	Description
CVE-2020-36518	jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.
CVE-2020-36789	In the Linux kernel, the following vulnerability has been resolved: can: dev: can_get_echo_skb(): prevent call to kfree_skb() in hard IRQ context If a driver calls can_get_echo_skb() during a hardware IRQ (which is often, but not always, the case), the 'WARN_ON(in_irq)' in net/core/skbuff.c#skb_release_head_state() might be triggered, under network congestion circumstances, together with the potential risk of a NULL pointer dereference. The root cause of this issue is the call to kfree_skb() instead of dev_kfree_skb_irq() in net/core/dev.c#enqueue_to_backlog(). This patch prevents the skb to be freed within the call to netif_rx() by incrementing its reference count with skb_get(). The skb is finally freed by one of the in-irq-context safe functions: dev_consume_skb_any() or dev_kfree_skb_any(). The "any" version is used because some drivers might call can_get_echo_skb() in a normal context. The reason for this issue to occur is that initially, in the core network stack, loopback skb were not supposed to be received in hardware IRQ context. The CAN stack is an exception. This bug was previously reported back in 2017 in [1] but the proposed patch never got accepted. While [1] directly modifies net/core/dev.c, we try to propose here a smoother modification local to CAN network stack (the assumption behind is that only CAN devices are affected by this issue). [1] http://lore.kernel.org/r/57a3ffb6-3309-3ad5-5a34-e93c3fe3614d@cetitec.com
CVE-2020-36790	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix a memory leak We forgot to free new_model_number
CVE-2020-36791	In the Linux kernel, the following vulnerability has been resolved: net_sched: keep alloc_hash updated after hash allocation In commit 599be01ee567 ("net_sched: fix an OOB access in cls_tcindex") I moved cp->hash calculation before the first tcindex_alloc_perfect_hash(), but cp->alloc_hash is left untouched. This difference could lead to another out of bound access. cp->alloc_hash should always be the size allocated, we should update it after this tcindex_alloc_perfect_hash().
CVE-2020-3702	u'Specifically timed and handcrafted traffic can cause internal errors in a WLAN device that lead to improper layer 2 Wi-Fi encryption with a consequent possibility of information disclosure over the air for a discrete set of traffic' in Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music, Snapdragon Wearables, Snapdragon Wired Infrastructure and Networking in APQ8053, IPQ4019, IPQ8064, MSM8909W, MSM8996AU, QCA9531, QCN5502, QCS405, SDX20, SM6150, SM7150
CVE-2020-7656	jquery prior to 1.9.0 allows Cross-site Scripting attacks via the load method. The load method fails to recognize and remove "<script>" HTML tags that contain a whitespace character, i.e: "</script >", which results in the enclosed script logic to be executed.
CVE-2020-8130	There is an OS command injection vulnerability in Ruby Rake < 12.3.3 in Rake::FileList when supplying a filename that begins with the pipe character ` `.
CVE-2020-8559	The Kubernetes kube-apiserver in versions v1.6-v1.15, and versions prior to v1.16.13, v1.17.9 and v1.18.6 are vulnerable to an unvalidated redirect on proxied upgrade requests that could allow an attacker to escalate privileges from a node compromise to a full cluster compromise.
CVE-2020-8694	Insufficient access control in the Linux kernel driver for some Intel(R) Processors may allow an authenticated user to potentially enable information disclosure via local access.
CVE-2020-8908	A temp directory creation vulnerability exists in all versions of Guava, allowing an attacker with access to the machine to potentially access data in a temporary directory created by the Guava API com.google.common.io.Files.createTempDir(). By default, on unix-like systems, the created directory is world-readable (readable by an attacker with access to the system). The method in question has been marked @Deprecated in versions 30.0 and later and should not be used. For Android developers, we recommend choosing a temporary directory API provided by Android, such as context.getCacheDir(). For other Java developers, we recommend migrating to the Java 7 API java.nio.file.Files.createTempDirectory() which explicitly configures permissions of 700, or configuring the Java runtime's java.io.tmpdir system property to point to a location whose permissions are appropriately configured.
CVE-2020-9492	In Apache Hadoop 3.2.0 to 3.2.1, 3.0.0-alpha1 to 3.1.3, and 2.0.0-alpha to 2.10.0, WebHDFS client might send SPNEGO authorization header to remote URL without proper verification.
CVE-2021-0561	In append_to_verify_fifo_interleaved_ of stream_encoder.c, there is a possible out of bounds write due to a missing bounds check. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.Product: AndroidVersions: Android-11Android ID: A-174302683

CVE ID	Description
CVE-2021-20193	A flaw was found in the src/list.c of tar 1.33 and earlier. This flaw allows an attacker who can submit a crafted input file to tar to cause uncontrolled consumption of memory. The highest threat from this vulnerability is to system availability.
CVE-2021-20320	A flaw was found in s390 eBPF JIT in bpf_jit_insn in arch/s390/net/bpf_jit_comp.c in the Linux kernel. In this flaw, a local attacker with special user privilege can circumvent the verifier and may lead to a confidentiality problem.
CVE-2021-22569	An issue in protobuf-java allowed the interleaving of com.google.protobuf.UnknownFieldSet fields in such a way that would be processed out of order. A small malicious payload can occupy the parser for several minutes by creating large numbers of short-lived objects that cause frequent, repeated pauses. We recommend upgrading libraries beyond the vulnerable versions.
CVE-2021-22570	Nullptr dereference when a null char is present in a proto symbol. The symbol is parsed incorrectly, leading to an unchecked call into the proto file's name during generation of the resulting error message. Since the symbol is incorrectly parsed, the file is nullptr. We recommend upgrading to version 3.15.0 or greater.
CVE-2021-23840	Calls to EVP_CipherUpdate, EVP_EncryptUpdate and EVP_DecryptUpdate may overflow the output length argument in some cases where the input length is close to the maximum permissible length for an integer on the platform. In such cases the return value from the function call will be 1 (indicating success), but the output length value will be negative. This could cause applications to behave incorrectly or crash. OpenSSL versions 1.1.1i and below are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1j. OpenSSL versions 1.0.2x and below are affected by this issue. However OpenSSL 1.0.2 is out of support and no longer receiving public updates. Premium support customers of OpenSSL 1.0.2 should upgrade to 1.0.2y. Other users should upgrade to 1.1.1j. Fixed in OpenSSL 1.1.1j (Affected 1.1.1-1.1.1i). Fixed in OpenSSL 1.0.2y (Affected 1.0.2-1.0.2x).
CVE-2021-23841	The OpenSSL public API function X509_issuer_and_serial_hash() attempts to create a unique hash value based on the issuer and serial number data contained within an X509 certificate. However it fails to correctly handle any errors that may occur while parsing the issuer field (which might occur if the issuer field is maliciously constructed). This may subsequently result in a NULL pointer deref and a crash leading to a potential denial of service attack. The function X509_issuer_and_serial_hash() is never directly called by OpenSSL itself so applications are only vulnerable if they use this function directly and they use it on certificates that may have been obtained from untrusted sources. OpenSSL versions 1.1.1i and below are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1j. OpenSSL versions 1.0.2x and below are affected by this issue. However OpenSSL 1.0.2 is out of support and no longer receiving public updates. Premium support customers of OpenSSL 1.0.2 should upgrade to 1.0.2y. Other users should upgrade to 1.1.1j. Fixed in OpenSSL 1.1.1j (Affected 1.1.1-1.1.1i). Fixed in OpenSSL 1.0.2y (Affected 1.0.2-1.0.2x).
CVE-2021-24032	Beginning in v1.4.1 and prior to v1.4.9, due to an incomplete fix for CVE-2021-24031, the Zstandard command-line utility created output files with default permissions and restricted those permissions immediately afterwards. Output files could therefore momentarily be readable or writable to unintended parties.
CVE-2021-25317	A Incorrect Default Permissions vulnerability in the packaging of cups of SUSE Linux Enterprise Server 11-SP4-LTSS, SUSE Manager Server 4.0, SUSE OpenStack Cloud Crowbar 9; openSUSE Leap 15.2, Factory allows local attackers with control of the lp users to create files as root with 0644 permissions without the ability to set the content. This issue affects: SUSE Linux Enterprise Server 11-SP4-LTSS cups versions prior to 1.3.9. SUSE Manager Server 4.0 cups versions prior to 2.2.7. SUSE OpenStack Cloud Crowbar 9 cups versions prior to 1.7.5. openSUSE Leap 15.2 cups versions prior to 2.2.7. openSUSE Factory cups version 2.3.3op2-2.1 and prior versions.
CVE-2021-26934	An issue was discovered in the Linux kernel 4.18 through 5.10.16, as used by Xen. The backend allocation (aka be-alloc) mode of the drm_xen_front drivers was not meant to be a supported configuration, but this wasn't stated accordingly in its support status entry.
CVE-2021-28168	Eclipse Jersey 2.28 to 2.33 and Eclipse Jersey 3.0.0 to 3.0.1 contains a local information disclosure vulnerability. This is due to the use of the File.createTempFile which creates a file inside of the system temporary directory with the permissions: -rw-r--r--. Thus the contents of this file are viewable by all other users locally on the system. As such, if the contents written is security sensitive, it can be disclosed to other local users.
CVE-2021-28170	In the Jakarta Expression Language implementation 3.0.3 and earlier, a bug in the ELParserTokenManager enables invalid EL expressions to be evaluated as if they were valid.

CVE ID	Description
CVE-2021-28711	Rogue backends can cause DoS of guests via high frequency events T[his CNA information record relates to multiple CVEs; the text explains which aspects/vulnerabilities correspond to which CVE.] Xen offers the ability to run PV backends in regular unprivileged guests, typically referred to as "driver domains". Running PV backends in driver domains has one primary security advantage: if a driver domain gets compromised, it doesn't have the privileges to take over the system. However, a malicious driver domain could try to attack other guests via sending events at a high frequency leading to a Denial of Service in the guest due to trying to service interrupts for elongated amounts of time. There are three affected backends: * blkfront patch 1, CVE-2021-28711 * netfront patch 2, CVE-2021-28712 * hvc_xen (console) patch 3, CVE-2021-28713
CVE-2021-28712	Rogue backends can cause DoS of guests via high frequency events T[his CNA information record relates to multiple CVEs; the text explains which aspects/vulnerabilities correspond to which CVE.] Xen offers the ability to run PV backends in regular unprivileged guests, typically referred to as "driver domains". Running PV backends in driver domains has one primary security advantage: if a driver domain gets compromised, it doesn't have the privileges to take over the system. However, a malicious driver domain could try to attack other guests via sending events at a high frequency leading to a Denial of Service in the guest due to trying to service interrupts for elongated amounts of time. There are three affected backends: * blkfront patch 1, CVE-2021-28711 * netfront patch 2, CVE-2021-28712 * hvc_xen (console) patch 3, CVE-2021-28713
CVE-2021-28713	Rogue backends can cause DoS of guests via high frequency events T[his CNA information record relates to multiple CVEs; the text explains which aspects/vulnerabilities correspond to which CVE.] Xen offers the ability to run PV backends in regular unprivileged guests, typically referred to as "driver domains". Running PV backends in driver domains has one primary security advantage: if a driver domain gets compromised, it doesn't have the privileges to take over the system. However, a malicious driver domain could try to attack other guests via sending events at a high frequency leading to a Denial of Service in the guest due to trying to service interrupts for elongated amounts of time. There are three affected backends: * blkfront patch 1, CVE-2021-28711 * netfront patch 2, CVE-2021-28712 * hvc_xen (console) patch 3, CVE-2021-28713
CVE-2021-28972	In drivers/pci/hotplug/rpadlp_sysfs.c in the Linux kernel through 5.11.8, the RPA PCI Hotplug driver has a user-tolerable buffer overflow when writing a new device name to the driver from userspace, allowing userspace to write data to the kernel stack frame directly. This occurs because add_slot_store and remove_slot_store mishandle drc_name '\0' termination, aka CID-cc7a0bb058b8.
CVE-2021-29657	arch/x86/kvm/svm/nested.c in the Linux kernel before 5.11.12 has a use-after-free in which an AMD KVM guest can bypass access control on host OS MSRs when there are nested guests, aka CID-a58d9166a756. This occurs because of a TOCTOU race condition associated with a VMCB12 double fetch in nested_svm_vmrn.
CVE-2021-3178	fs/nfsd/nfs3xdr.c in the Linux kernel through 5.10.8, when there is an NFS export of a subdirectory of a filesystem, allows remote attackers to traverse to other parts of the filesystem via REaddirPLUS. NOTE: some parties argue that such a subdirectory export is not intended to prevent this attack; see also the exports(5) no_subtree_check default behavior
CVE-2021-31879	GNU Wget through 1.21.1 does not omit the Authorization header upon a redirect to a different origin, a related issue to CVE-2018-1000007.
CVE-2021-3236	vim 8.2.2348 is affected by null pointer dereference, allows local attackers to cause a denial of service (DoS) via the ex_buffer_all method.
CVE-2021-33036	In Apache Hadoop 2.2.0 to 2.10.1, 3.0.0-alpha1 to 3.1.4, 3.2.0 to 3.2.2, and 3.3.0 to 3.3.1, a user who can escalate to yarn user can possibly run arbitrary commands as root user. Users should upgrade to Apache Hadoop 2.10.2, 3.2.3, 3.3.2 or higher.
CVE-2021-33294	In elfutils 0.183, an infinite loop was found in the function handle_syntab in readelf.c .Which allows attackers to cause a denial of service (infinite loop) via crafted file.
CVE-2021-33560	Libgcrypt before 1.8.8 and 1.9.x before 1.9.3 mishandles ElGamal encryption because it lacks exponent blinding to address a side-channel attack against mpi_powm, and the window size is not chosen appropriately. This, for example, affects use of ElGamal in OpenPGP.
CVE-2021-33624	In kernel/bpf/verifier.c in the Linux kernel before 5.12.13, a branch can be mispredicted (e.g., because of type confusion) and consequently an unprivileged BPF program can read arbitrary memory locations via a side-channel attack, aka CID-9183671af6db.
CVE-2021-3411	A flaw was found in the Linux kernel in versions prior to 5.10. A violation of memory access was found while detecting a padding of int3 in the linking state. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability.

CVE ID	Description
CVE-2021-34428	For Eclipse Jetty versions <= 9.4.40, <= 10.0.2, <= 11.0.2, if an exception is thrown from the SessionListener#sessionDestroyed() method, then the session ID is not invalidated in the session ID manager. On deployments with clustered sessions and multiple contexts this can result in a session not being invalidated. This can result in an application used on a shared computer being left logged in.
CVE-2021-3443	A NULL pointer dereference flaw was found in the way Jasper versions before 2.0.27 handled component references in the JP2 image format decoder. A specially crafted JP2 image file could cause an application using the Jasper library to crash when opened.
CVE-2021-3444	The bpf verifier in the Linux kernel did not properly handle mod32 destination register truncation when the source register was known to be 0. A local attacker with the ability to load bpf programs could use this gain out-of-bounds reads in kernel memory leading to information disclosure (kernel memory), and possibly out-of-bounds writes that could potentially lead to code execution. This issue was addressed in the upstream kernel in commit 9b00f1b78809 ("bpf: Fix truncation handling for mod32 dst reg wrt zero") and in Linux stable kernels 5.11.2, 5.10.19, and 5.4.101.
CVE-2021-34556	In the Linux kernel through 5.13.7, an unprivileged BPF program can obtain sensitive information from kernel memory via a Speculative Store Bypass side-channel attack because the protection mechanism neglects the possibility of uninitialized memory locations on the BPF stack.
CVE-2021-3467	A NULL pointer dereference flaw was found in the way Jasper versions before 2.0.26 handled component references in CDEF box in the JP2 image format decoder. A specially crafted JP2 image file could cause an application using the Jasper library to crash when opened.
CVE-2021-34693	net/can/bcm.c in the Linux kernel through 5.12.10 allows local users to obtain sensitive information from kernel stack memory because parts of a data structure are uninitialized.
CVE-2021-34981	Linux Kernel Bluetooth CMTP Module Double Free Privilege Escalation Vulnerability. This vulnerability allows local attackers to escalate privileges on affected installations of Linux Kernel. An attacker must first obtain the ability to execute high-privileged code on the target system in order to exploit this vulnerability. The specific flaw exists within the CMTP module. The issue results from the lack of validating the existence of an object prior to performing further free operations on the object. An attacker can leverage this vulnerability to escalate privileges and execute code in the context of the kernel. Was ZDI-CAN-11977.
CVE-2021-35477	In the Linux kernel through 5.13.7, an unprivileged BPF program can obtain sensitive information from kernel memory via a Speculative Store Bypass side-channel attack because a certain preempting store operation does not necessarily occur before a store operation that has an attacker-controlled value.
CVE-2021-3655	A vulnerability was found in the Linux kernel in versions prior to v5.14-rc1. Missing size validations on inbound SCTP packets may allow the kernel to read uninitialized memory.
CVE-2021-3714	A flaw was found in the Linux kernels memory deduplication mechanism. Previous work has shown that memory deduplication can be attacked via a local exploitation mechanism. The same technique can be used if an attacker can upload page sized files and detect the change in access time from a networked service to determine if the page has been merged.
CVE-2021-37404	There is a potential heap buffer overflow in Apache Hadoop libhdfs native code. Opening a file path provided by user without validation may result in a denial of service or arbitrary code execution. Users should upgrade to Apache Hadoop 2.10.2, 3.2.3, 3.3.2 or higher.
CVE-2021-38166	In kernel/bpf/hashtab.c in the Linux kernel through 5.13.8, there is an integer overflow and out-of-bounds write when many elements are placed in a single bucket. NOTE: exploitation might be impractical without the CAP_SYS_ADMIN capability.
CVE-2021-38198	arch/x86/kvm/mmu/paging_tmpl.h in the Linux kernel before 5.12.11 incorrectly computes the access permissions of a shadow page, leading to a missing guest protection page fault.
CVE-2021-38199	fs/nfs/nfs4client.c in the Linux kernel before 5.13.4 has incorrect connection-setup ordering, which allows operators of remote NFSv4 servers to cause a denial of service (hanging of mounts) by arranging for those servers to be unreachable during trunking detection.
CVE-2021-38206	The mac80211 subsystem in the Linux kernel before 5.12.13, when a device supporting only 5 GHz is used, allows attackers to cause a denial of service (NULL pointer dereference in the radiotap parser) by injecting a frame with 802.11a rates.
CVE-2021-3826	Heap/stack buffer overflow in the clang_lname function in d-demangle.c in libiberty allows attackers to potentially cause a denial of service (segmentation fault and crash) via a crafted mangled symbol.
CVE-2021-38593	Qt 5.x before 5.15.6 and 6.x through 6.1.2 has an out-of-bounds write in QOutlineMapper::convertPath (called from QRasterPaintEngine::fill and QPaintEngineEx::stroke).

CVE ID	Description
CVE-2021-3864	A flaw was found in the way the dumpable flag setting was handled when certain SUID binaries executed its descendants. The prerequisite is a SUID binary that sets real UID equal to effective UID, and real GID equal to effective GID. The descendant will then have a dumpable value set to 1. As a result, if the descendant process crashes and core_pattern is set to a relative value, its core dump is stored in the current directory with uid:gid permissions. An unprivileged local user with eligible root SUID binary could use this flaw to place core dumps into root-owned directories, potentially resulting in escalation of privileges.
CVE-2021-3927	vim is vulnerable to Heap-based Buffer Overflow
CVE-2021-39537	An issue was discovered in ncurses through v6.2-1. _nc_captaininfo in captaininfo.c has a heap-based buffer overflow.
CVE-2021-3974	vim is vulnerable to Use After Free
CVE-2021-3997	A flaw was found in systemd. An uncontrolled recursion in systemd-tmpfiles may lead to a denial of service at boot time when too many nested directories are created in /tmp.
CVE-2021-4001	A race condition was found in the Linux kernel's ebpf verifier between bpf_map_update_elem and bpf_map_freeze due to a missing lock in kernel/bpf/syscall.c. In this flaw, a local user with a special privilege (cap_sys_admin or cap_bpf) can modify the frozen mapped address space. This flaw affects kernel versions prior to 5.16 rc2.
CVE-2021-40490	A race condition was discovered in ext4_write_inline_data_end in fs/ext4/inline.c in the ext4 subsystem in the Linux kernel through 5.13.13.
CVE-2021-41182	jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `altField` option of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `altField` option is now treated as a CSS selector. A workaround is to not accept the value of the `altField` option from untrusted sources.
CVE-2021-41183	jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of various `*Text` options of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. The values passed to various `*Text` options are now always treated as pure text, not HTML. A workaround is to not accept the value of the `*Text` options from untrusted sources.
CVE-2021-41184	jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `of` option of the `.position()` util from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `of` option is now treated as a CSS selector. A workaround is to not accept the value of the `of` option from untrusted sources.
CVE-2021-4135	A memory leak vulnerability was found in the Linux kernel's eBPF for the Simulated networking device driver in the way user uses BPF for the device such that function nsim_map_alloc_elem being called. A local user could use this flaw to get unauthorized access to some data.
CVE-2021-4159	A vulnerability was found in the Linux kernel's EBPF verifier when handling internal data structures. Internal memory locations could be returned to userspace. A local attacker with the permissions to insert eBPF code to the kernel can use this to leak internal kernel memory details defeating some of the exploit mitigations in place for the kernel.
CVE-2021-4166	vim is vulnerable to Out-of-bounds Read
CVE-2021-4209	A NULL pointer dereference flaw was found in GnuTLS. As Nettle's hash update functions internally call memcpy, providing zero-length input may cause undefined behavior. This flaw leads to a denial of service after authentication in rare circumstances.
CVE-2021-4217	A flaw was found in unzip. The vulnerability occurs due to improper handling of Unicode strings, which can lead to a null pointer dereference. This flaw allows an attacker to input a specially crafted zip file, leading to a crash or code execution.
CVE-2021-42357	When using Apache Knox SSO prior to 1.6.1, a request could be crafted to redirect a user to a malicious page due to improper URL parsing. A request that included a specially crafted request parameter could be used to redirect the user to a page controlled by an attacker. This URL would need to be presented to the user outside the normal request flow through a XSS or phishing campaign.

CVE ID	Description
CVE-2021-4439	In the Linux kernel, the following vulnerability has been resolved: isdn: cpai: check ctr->cnr to avoid array index out of bound The cmtplib_add_connection() would add a cmtplib session to a controller and run a kernel thread to process cmtplib. __module_get(THIS_MODULE); session->task = kthread_run(cmtplib_session, session, "kcmtplib_ctr_%d", session->num); During this process, the kernel thread would call detach_capi_ctr() to detach a register controller. if the controller was not attached yet, detach_capi_ctr() would trigger an array-index-out-of-bounds bug. [46.866069][T6479] UBSAN: array-index-out-of-bounds in drivers/isdn/capi/kcapi.c:483:21 [46.867196][T6479] index -1 is out of range for type 'capi_ctr *[32]' [46.867982][T6479] CPU: 1 PID: 6479 Comm: kcmtplib_ctr_0 Not tainted 5.15.0-rc2+ #8 [46.869002][T6479] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-2 04/01/2014 [46.870107][T6479] Call Trace: [46.870473][T6479] dump_stack_lvl+0x57/0x7d [46.870974][T6479] ubsan_epilogue+0x5/0x40 [46.871458][T6479] __ubsan_handle_out_of_bounds.cold+0x43/0x48 [46.872135][T6479] detach_capi_ctr+0x64/0xc0 [46.872639][T6479] cmtplib_session+0x5c8/0x5d0 [46.873131][T6479] ? __init_waitqueue_head+0x60/0x60 [46.873712][T6479] ? cmtplib_add_msgpart+0x120/0x120 [46.874256][T6479] kthread+0x147/0x170 [46.874709][T6479] ? set_kthread_struct+0x40/0x40 [46.875248][T6479] ret_from_fork+0x1f/0x30 [46.875773][T6479]
CVE-2021-4440	In the Linux kernel, the following vulnerability has been resolved: x86/xen: Drop USERGS_SYCRET64 paravirt call commit afd30525a659ac0ae0904f0cb4a2ca75522c3123 upstream. USERGS_SYCRET64 is used to return from a syscall via SYSRET, but a Xen PV guest will nevertheless use the IRET hypercall, as there is no sysret PV hypercall defined. So instead of testing all the prerequisites for doing a sysret and then mangling the stack for Xen PV again for doing an iret just use the iret exit from the beginning. This can easily be done via an ALTERNATIVE like it is done for the sysenter compat case already. It should be noted that this drops the optimization in Xen for not restoring a few registers when returning to user mode, but it seems as if the saved instructions in the kernel mode than compensate for this drop (a kernel build in a Xen PV guest was slightly faster with this patch applied). While at it remove the stale sysret32 remnants. [pawan: Brad Spengler and Salvatore Bonaccorso <carnil@debian.org> reported a problem with the 5.10 backport commit edc702b4a820 ("x86/entry_64: Add VERW just before userspace transition"). When CONFIG_PARAVIRT_XXL=y, CLEAR_CPU_BUFFERS is not executed in syscall_return_via_sysret path as USERGS_SYCRET64 is runtime patched to: .cpu_usergs_sysret64 = { 0x0f, 0x01, 0xf8, 0x48, 0x0f, 0x07 }, // swapgs; sysretq which is missing CLEAR_CPU_BUFFERS. It turns out dropping USERGS_SYCRET64 simplifies the code, allowing CLEAR_CPU_BUFFERS to be explicitly added to syscall_return_via_sysret path. Below is with CONFIG_PARAVIRT_XXL=y and this patch applied: syscall_return_via_sysret: ... <+342>: swapgs <+345>: xchg %ax,%ax <+347>: verw -0x1a2(%rip) <----- <+354>: sysretq]
CVE-2021-4442	In the Linux kernel, the following vulnerability has been resolved: tcp: add sanity tests to TCP_QUEUE_SEQ Qingyu Li reported a syzkaller bug where the repro changes RCV_SEQ after restoring data in the receive queue. mprotect(0x4aa000, 12288, PROT_READ) = 0 mmap(0x1ffff000, 4096, PROT_NONE, MAP_PRIVATE MAP_FIXED MAP_ANONYMOUS, -1, 0) = 0x1ffff000 mmap(0x20000000, 16777216, PROT_READ PROT_WRITE PROT_EXEC, MAP_PRIVATE MAP_FIXED MAP_ANONYMOUS, -1, 0) = 0x20000000 mmap(0x21000000, 4096, PROT_NONE, MAP_PRIVATE MAP_FIXED MAP_ANONYMOUS, -1, 0) = 0x21000000 socket(AF_INET6, SOCK_STREAM, IPPROTO_IP) = 3 setsockopt(3, SOL_TCP, TCP_REPAIR, [1], 4) = 0 connect(3, {sa_family=AF_INET6, sin6_port=htons(0), sin6_flowinfo=htonl(0), inet_pton(AF_INET6, "::1", &sin6_addr), sin6_scope_id=0}, 28) = 0 setsockopt(3, SOL_TCP, TCP_REPAIR_QUEUE, [1], 4) = 0 sendmsg(3, {msg_name=NULL, msg_namelen=0, msg_iov=[{iov_base="0x0000000000000003\0\0", iov_len=20}], msg_iovlen=1, msg_controllen=0, msg_flags=0}, 0) = 20 setsockopt(3, SOL_TCP, TCP_REPAIR, [0], 4) = 0 setsockopt(3, SOL_TCP, TCP_QUEUE_SEQ, [128], 4) = 0 recvfrom(3, NULL, 20, 0, NULL, NULL) = -1 ECONNRESET (Connection reset by peer) syslog shows: [111.205099] TCP rcvmsg seq # bug 2: copied 80, seq 0, rcvnext 80, fl 0 [111.207894] WARNING: CPU: 1 PID: 356 at net/ipv4/tcp.c:2343 tcp_rcvmsg_locked+0x90e/0x29a0 This should not be allowed. TCP_QUEUE_SEQ should only be used when queues are empty. This patch fixes this case, and the tx path as well.
CVE-2021-4460	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix UBSAN shift-out-of-bounds warning If get_num_sdma_queues or get_num_xgmi_sdma_queues is 0, we end up doing a shift operation where the number of bits shifted equals number of bits in the operand. This behaviour is undefined. Set num_sdma_queues or num_xgmi_sdma_queues to ULLONG_MAX, if the count is >= number of bits in the operand. Bug: https://gitlab.freedesktop.org/drm/amd/-/issues/1472
CVE-2021-44732	Mbed TLS before 3.0.1 has a double free in certain out-of-memory conditions, as demonstrated by an mbedtls_ssl_set_session() failure.
CVE-2021-45078	stab_xcoff_builtin_type in stabs.c in GNU Binutils through 2.37 allows attackers to cause a denial of service (heap-based buffer overflow) or possibly have unspecified other impact, as demonstrated by an out-of-bounds write. NOTE: this issue exists because of an incorrect fix for CVE-2018-12699.

CVE ID	Description
CVE-2021-45261	An Invalid Pointer vulnerability exists in GNU patch 2.7 via the another_hunk function, which causes a Denial of Service.
CVE-2021-45402	The check_alu_op() function in kernel/bpf/verifier.c in the Linux kernel through v5.16-rc5 did not properly update bounds while handling the mov32 instruction, which allows local users to obtain potentially sensitive address information, aka a "pointer leak."
CVE-2021-46877	jackson-databind 2.10.x through 2.12.x before 2.12.6 and 2.13.x before 2.13.1 allows attackers to cause a denial of service (2 GB transient heap usage per read) in uncommon situations involving JsonNode JDK serialization.
CVE-2021-46904	In the Linux kernel, the following vulnerability has been resolved: net: hso: fix null-ptr-deref during tty device unregistration Multiple ttys try to claim the same the minor number causing a double unregistration of the same device. The first unregistration succeeds but the next one results in a null-ptr-deref. The get_free_serial_index() function returns an available minor number but doesn't assign it immediately. The assignment is done by the caller later. But before this assignment, calls to get_free_serial_index() would return the same minor number. Fix this by modifying get_free_serial_index to assign the minor number immediately after one is found to be and rename it to obtain_minor() to better reflect what it does. Similarly, rename set_serial_by_index() to release_minor() and modify it to free up the minor number of the given hso_serial. Every obtain_minor() should have corresponding release_minor() call.
CVE-2021-46908	In the Linux kernel, the following vulnerability has been resolved: bpf: Use correct permission flag for mixed signed bounds arithmetic We forbid adding unknown scalars with mixed signed bounds due to the spectre v1 masking mitigation. Hence this also needs bypass_spec_v1 flag instead of allow_ptr_leaks.
CVE-2021-46910	In the Linux kernel, the following vulnerability has been resolved: ARM: 9063/1: mm: reduce maximum number of CPUs if DEBUG_KMAP_LOCAL is enabled The debugging code for kmap_local() doubles the number of per-CPU fixmap slots allocated for kmap_local(), in order to use half of them as guard regions. This causes the fixmap region to grow downwards beyond the start of its reserved window if the supported number of CPUs is large, and collide with the newly added virtual DT mapping right below it, which is obviously not good. One manifestation of this is EFI boot on a kernel built with NR_CPUS=32 and CONFIG_DEBUG_KMAP_LOCAL=y, which may pass the FDT in highmem, resulting in block entries below the fixmap region that the fixmap code misidentifies as fixmap table entries, and subsequently tries to dereference using a phys-to-virt translation that is only valid for lowmem. This results in a cryptic splat such as the one below. ftrace: allocating 45548 entries in 89 pages 8<--- cut here --- Unable to handle kernel paging request at virtual address fc6006f0 pgd = (ptrval) [fc6006f0] *pgd=80000040207003, *pmd=00000000 Internal error: Oops: a06 [#1] SMP ARM Modules linked in: CPU: 0 PID: 0 Comm: swapper Not tainted 5.11.0+ #382 Hardware name: Generic DT based system PC is at cpu_ca15_set_pte_ext+0x24/0x30 LR is at __set_fixmap+0xe4/0x118 pc : [<c041ac9c>] lr : [<c04189d8>] psr: 400000d3 sp : c1601ed8 ip : 00400000 fp : 00800000 r10: 0000071f r9 : 00421000 r8 : 00c00000 r7 : 00c00000 r6 : 0000071f r5 : ffade000 r4 : 4040171f r3 : 00c00000 r2 : 4040171f r1 : c041ac78 r0 : fc6006f0 Flags: nZcv IRQs off FIQs off Mode SVC_32 ISA ARM Segment none Control: 30c5387d Table: 40203000 DAC: 00000001 Process swapper (pid: 0, stack limit = 0x(ptrval)) So let's limit CONFIG_NR_CPUS to 16 when CONFIG_DEBUG_KMAP_LOCAL=y. Also, fix the BUILD_BUG_ON() check that was supposed to catch this, by checking whether the region grows below the start address rather than above the end address.

CVE ID	Description
CVE-2021-46915	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_limit: avoid possible divide error in nft_limit_init div_u64() divides u64 by u32. nft_limit_init() wants to divide u64 by u64, use the appropriate math function (div64_u64) divide error: 0000 [#1] PREEMPT SMP KASAN CPU: 1 PID: 8390 Comm: syz-executor188 Not tainted 5.12.0-rc4-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:div_u64_rem include/linux/math64.h:28 [inline] RIP: 0010:div_u64 include/linux/math64.h:127 [inline] RIP: 0010:nft_limit_init+0x2a2/0x5e0 net/netfilter/nft_limit.c:85 Code: ef 4c 01 eb 41 0f 92 c7 48 89 de e8 38 a5 22 fa 4d 85 ff 0f 85 97 02 00 00 e8 ea 9e 22 fa 4c 0f af f3 45 89 ed 31 d2 4c 89 f0 <49> f7 f5 49 89 c6 e8 d3 9e 22 fa 48 8d 7d 48 48 b8 00 00 00 00 00 RSP: 0018:ffff90009447198 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 0000200000000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffffffff875152e6 RDI: 0000000000000003 RBP: ffff888020f80908 R08: 0000200000000000 R09: 0000000000000000 R10: ffffffff875152d8 R11: 0000000000000000 R12: ffff90009447270 R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 FS: 000000000097a300(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000200001c4 CR3: 0000000026a52000 CR4: 00000000001506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: nf_tables_newexpr net/netfilter/nf_tables_api.c:2675 [inline] nft_expr_init+0x145/0x2d0 net/netfilter/nf_tables_api.c:2713 nft_set_elem_expr_alloc+0x27/0x280 net/netfilter/nf_tables_api.c:5160 nf_tables_newset +0x1997/0x3150 net/netfilter/nf_tables_api.c:4321 nfnetlink_rcv_batch+0x85a/0x21b0 net/netfilter/nfnetlink.c:456 nfnetlink_rcv_skb_batch net/netfilter/nfnetlink.c:580 [inline] nfnetlink_rcv +0x3af/0x420 net/netfilter/nfnetlink.c:598 netlink_unicast_kernel net/netlink/af_netlink.c:1312 [inline] netlink_unicast+0x533/0x7d0 net/netlink/af_netlink.c:1338 netlink_sendmsg+0x856/0xd90 net/netlink/af_netlink.c:1927 sock_sendmsg_nosec net/socket.c:654 [inline] sock_sendmsg+0xcf/0x120 net/socket.c:674 __sys_sendmsg+0x6e8/0x810 net/socket.c:2350 __sys_sendmsg+0xf3/0x170 net/socket.c:2404 __sys_sendmsg+0xe5/0x1b0 net/socket.c:2433 do_syscall_64+0x2d/0x70 arch/x86/entry/common.c:46 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2021-46917	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: fix wq cleanup of WQCFG registers A pre-release silicon erratum workaround where wq reset does not clear WQCFG registers was leaked into upstream code. Use wq reset command instead of blasting the MMIO region. This also address an issue where we clobber registers in future devices.
CVE-2021-46918	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: clear MSIX permission entry on shutdown Add disabling/clearing of MSIX permission entries on device shutdown to mirror the enabling of the MSIX entries on probe. Current code left the MSIX enabled and the pasid entries still programmed at device shutdown.
CVE-2021-46920	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix clobbering of SWERR overflow bit on writeback Current code blindly writes over the SWERR and the OVERFLOW bits. Write back the bits actually read instead so the driver avoids clobbering the OVERFLOW bit that comes after the register is read.
CVE-2021-46921	In the Linux kernel, the following vulnerability has been resolved: locking/qrwlock: Fix ordering in queued_write_lock_slowpath() While this code is executed with the wait_lock held, a reader can acquire the lock without holding wait_lock. The writer side loops checking the value with the atomic_cond_read_acquire(), but only truly acquires the lock when the compare-and-exchange is completed successfully which isn't ordered. This exposes the window between the acquire and the cmpxchg to an A-B-A problem which allows reads following the lock acquisition to observe values speculatively before the write lock is truly acquired. We've seen a problem in epoll where the reader does a xchg while holding the read lock, but the writer can see a value change out from under it. Writer Reader ----- ep_scan_ready_list() - write_lock_irq() - queued_write_lock_slowpath() - atomic_cond_read_acquire() read_lock_irqsave(&ep->lock, flags); --> (observes value before unlock) chain_epi_lockless() epi->next = xchg(&ep->ovflist, epi); read_unlock_irqrestore(&ep->lock, flags); atomic_cmpxchg_relaxed() -- READ_ONCE(ep->ovflist); A core can order the read of the ovflist ahead of the atomic_cmpxchg_relaxed(). Switching the cmpxchg to use acquire semantics addresses this issue at which point the atomic_cond_read can be switched to use relaxed semantics. [peterz: use try_cmpxchg()]

CVE ID	Description
CVE-2021-46925	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix kernel panic caused by race of smc_sock A crash occurs when smc_cdc_tx_handler() tries to access smc_sock but smc_release() has already freed it. [4570.695099] BUG: unable to handle page fault for address: 000000002eae9e88 [4570.696048] #PF: supervisor write access in kernel mode [4570.696728] #PF: error_code(0x0002) - not-present page [4570.697401] PGD 0 P4D 0 [4570.697716] Oops: 0002 [#1] PREEMPT SMP NOPTI [4570.698228] CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.16.0-rc4+ #111 [4570.699013] Hardware name: Alibaba Cloud Alibaba Cloud ECS, BIOS 8c24b4c 04/0 [4570.699933] RIP: 0010: _raw_spin_lock+0x1a/0x30 <...> [4570.711446] Call Trace: [4570.711746] <IRQ> [4570.711992] smc_cdc_tx_handler+0x41/0xc0 [4570.712470] smc_wr_tx_tasklet_fn+0x213/0x560 [4570.712981] ? smc_cdc_tx_dismiss+0x10/0x10 [4570.713489] tasklet_action_common.isra.17+0x66/0x140 [4570.714083] __do_softirq+0x123/0x2f4 [4570.714521] irq_exit_rcu+0xc4/0xf0 [4570.714934] common_interrupt+0xba/0xe0 Though smc_cdc_tx_handler() checked the existence of smc connection, smc_release() may have already dismissed and released the smc socket before smc_cdc_tx_handler() further visits it. smc_cdc_tx_handler() smc_release() if (!conn) smc_cdc_tx_dismiss_slots() smc_cdc_tx_dismiss() sock_put(&smc->sk) <- last sock_put, smc_sock freed bh_lock_sock(&smc->sk) (panic) To make sure we won't receive any CDC messages after we free the smc_sock, add a refcount on the smc_connection for inflight CDC message(posted to the QP but haven't received related CQE), and don't release the smc_connection until all the inflight CDC messages haven been done, for both success or failed ones. Using refcount on CDC messages brings another problem: when the link is going to be destroyed, smcr_link_clear() will reset the QP, which then remove all the pending CQEs related to the QP in the CQ. To make sure all the CQEs will always come back so the refcount on the smc_connection can always reach 0, smc_ib_modify_qp_reset() was replaced by smc_ib_modify_qp_error(). And remove the timeout in smc_wr_tx_wait_no_pending_sends() since we need to wait for all pending WQEs done, or we may encounter use-after- free when handling CQEs. For IB device removal routine, we need to wait for all the QPs on that device been destroyed before we can destroy CQs on the device, or the refcount on smc_connection won't reach 0 and smc_sock cannot be released.
CVE-2021-46926	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: intel-sdw-acpi: harden detection of controller The existing code currently sets a pointer to an ACPI handle before checking that it's actually a SoundWire controller. This can lead to issues where the graph walk continues and eventually fails, but the pointer was set already. This patch changes the logic so that the information provided to the caller is set when a controller is found.
CVE-2021-46929	In the Linux kernel, the following vulnerability has been resolved: sctp: use call_rcu to free endpoint This patch is to delay the endpoint free by calling call_rcu() to fix another use-after-free issue in sctp_sock_dump(): BUG: KASAN: use-after-free in __lock_acquire+0x36d9/0x4c20 Call Trace: __lock_acquire+0x36d9/0x4c20 kernel/locking/lockdep.c:3218 lock_acquire+0x1ed/0x520 kernel/locking/lockdep.c:3844 __raw_spin_lock_bh include/linux/spinlock_api_smp.h:135 [inline] _raw_spin_lock_bh+0x31/0x40 kernel/locking/spinlock.c:168 spin_lock_bh include/linux/spinlock.h:334 [inline] __lock_sock+0x203/0x350 net/core/sock.c:2253 lock_sock_nested+0xfe/0x120 net/core/sock.c:2774 lock_sock include/net/sock.h:1492 [inline] sctp_sock_dump+0x122/0xb20 net/sctp/diag.c:324 sctp_for_each_transport+0x2b5/0x370 net/sctp/socket.c:5091 sctp_diag_dump+0x3ac/0x660 net/sctp/diag.c:527 __inet_diag_dump+0xa8/0x140 net/ipv4/inet_diag.c:1049 inet_diag_dump+0x9b/0x110 net/ipv4/inet_diag.c:1065 netlink_dump+0x606/0x1080 net/netlink/af_netlink.c:2244 __netlink_dump_start+0x59a/0x7c0 net/netlink/af_netlink.c:2352 netlink_dump_start include/linux/netlink.h:216 [inline] inet_diag_handler_cmd+0x2ce/0x3f0 net/ipv4/inet_diag.c:1170 __sock_diag_cmd net/core/sock_diag.c:232 [inline] sock_diag_rcv_msg+0x31d/0x410 net/core/sock_diag.c:263 netlink_rcv_skb+0x172/0x440 net/netlink/af_netlink.c:2477 sock_diag_rcv+0x2a/0x40 net/core/sock_diag.c:274 This issue occurs when asoc is peeled off and the old sk is freed after getting it by asoc->base.sk and before calling lock_sock(sk). To prevent the sk free, as a holder of the sk, ep should be alive when calling lock_sock(). This patch uses call_rcu() and moves sock_put and ep free into sctp_endpoint_destroy_rcu(), so that it's safe to try to hold the ep under rcu_read_lock in sctp_transport_traverse_process(). If sctp_endpoint_hold() returns true, it means this ep is still alive and we have held it and can continue to dump it; If it returns false, it means this ep is dead and can be freed after rcu_read_unlock, and we should skip it. In sctp_sock_dump(), after locking the sk, if this ep is different from tsp->asoc->ep, it means during this dumping, this asoc was peeled off before calling lock_sock(), and the sk should be skipped; If this ep is the same with tsp->asoc->ep, it means no peeloff happens on this asoc, and due to lock_sock, no peeloff will happen either until release_sock. Note that delaying endpoint free won't delay the port release, as the port release happens in sctp_endpoint_destroy() before calling call_rcu(). Also, freeing endpoint by call_rcu() makes it safe to access the sk by asoc->base.sk in sctp_assocs_seq_show() and sctp_rcv(). Thanks Jones to bring this issue up. v1->v2: - improve the changelog. - add kfree(ep) into sctp_endpoint_destroy_rcu(), as Jakub noticed.

CVE ID	Description
CVE-2021-46932	In the Linux kernel, the following vulnerability has been resolved: Input: appletouch - initialize work before device registration Syzbot has reported warning in __flush_work(). This warning is caused by work->func == NULL, which means missing work initialization. This may happen, since input_dev->close() calls cancel_work_sync(&dev->work), but dev->work initialization happens _after_ input_register_device() call. So this patch moves dev->work initialization before registering input device
CVE-2021-46937	In the Linux kernel, the following vulnerability has been resolved: mm/damon/dbgfs: fix 'struct pid' leaks in 'dbgfs_target_ids_write()' DAMON debugfs interface increases the reference counts of 'struct pid's for targets from the 'target_ids' file write callback ('dbgfs_target_ids_write()'), but decreases the counts only in DAMON monitoring termination callback ('dbgfs_before_terminate()'). Therefore, when 'target_ids' file is repeatedly written without DAMON monitoring start/termination, the reference count is not decreased and therefore memory for the 'struct pid' cannot be freed. This commit fixes this issue by decreasing the reference counts when 'target_ids' is written.
CVE-2021-46960	In the Linux kernel, the following vulnerability has been resolved: cifs: Return correct error code from smb2_get_enc_key Avoid a warning if the error percolates back up: [440700.376476] CIFS VFS: \\\otters.example.com crypt_message: Could not get encryption key [440700.386947] ----- [cut here]----- [440700.386948] err = 1 [440700.386977] WARNING: CPU: 11 PID: 2733 at /build/linux-hwe-5.4-p6lk6L/linux-hwe-5.4-5.4.0/lib/errseq.c:74 errseq_set+0x5c/0x70 ... [440700.397304] CPU: 11 PID: 2733 Comm: tar Tainted: G OE 5.4.0-70-generic #78~18.04.1-Ubuntu ... [440700.397334] Call Trace: [440700.397346] __filemap_set_wb_err+0x1a/0x70 [440700.397419] cifs_writewrites+0x9c7/0xb30 [cifs] [440700.397426] do_writewrites+0x4b/0xe0 [440700.397444] __filemap_fdatwrite_range+0xcb/0x100 [440700.397455] filemap_write_and_wait+0x42/0xa0 [440700.397486] cifs_setattr+0x68b/0xf30 [cifs] [440700.397493] notify_change+0x358/0x4a0 [440700.397500] utimes_common+0xe9/0x1c0 [440700.397510] do_utimes+0xc5/0x150 [440700.397520] __x64_sys_utimensat+0x88/0xd0
CVE-2021-46961	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Do not enable irqs when handling spurious interrupts We triggered the following error while running our 4.19 kernel with the pseudo-NMI patches backported to it: [14.816231] -----[cut here]----- [14.816231] kernel BUG at irq.c:99! [14.816232] Internal error: Oops - BUG: 0 [#1] SMP [14.816232] Process swapper/0 (pid: 0, stack limit = 0x(____ptrval____)) [14.816233] CPU: 0 PID: 0 Comm: swapper/0 Tainted: G O 4.19.95.aarch64 #14 [14.816233] Hardware name: evb (DT) [14.816234] pstate: 80400085 (Nzcv daIf +PAN -UAO) [14.816234] pc : asm_nmi_enter+0x94/0x98 [14.816235] lr : asm_nmi_enter+0x18/0x98 [14.816235] sp : ffff000008003c50 [14.816235] pmr_save: 00000070 [14.816237] x29: ffff000008003c50 x28: ffff0000095f56c0 [14.816238] x27: 0000000000000000 x26: ffff000008004000 [14.816239] x25: 0000000015e0000 x24: ffff8008fb916000 [14.816240] x23: 0000000020400005 x22: ffff0000080817cc [14.816241] x21: ffff000008003da0 x20: 0000000000000060 [14.816242] x19: 00000000000003ff x18: ffffffff00000000 [14.816243] x17: 0000000000000008 x16: 003d090000000000 [14.816244] x15: ffff0000095ea6c8 x14: ffff8008fff5ab40 [14.816244] x13: ffff8008fff58b9d x12: 0000000000000000 [14.816245] x11: ffff000008c8a200 x10: 000000008e31fca5 [14.816246] x9 : ffff000008c8a208 x8 : 000000000000000f [14.816247] x7 : 0000000000000004 x6 : ffff8008fff58b9e [14.816248] x5 : 0000000000000000 x4 : 0000000080000000 [14.816249] x3 : 0000000000000000 x2 : 0000000080000000 [14.816250] x1 : 0000000000120000 x0 : ffff0000095f56c0 [14.816251] Call trace: [14.816251] asm_nmi_enter+0x94/0x98 [14.816251] el1_irq+0x8c/0x180 (IRQ C) [14.816252] gic_handle_irq+0xbc/0x2e4 [14.816252] el1_irq+0xcc/0x180 (IRQ B) [14.816253] arch_timer_handler_virt+0x38/0x58 [14.816253] handle_percpu_devid_irq+0x90/0x240 [14.816253] generic_handle_irq+0x34/0x50 [14.816254] __handle_domain_irq+0x68/0xc0 [14.816254] gic_handle_irq+0xf8/0x2e4 [14.816255] el1_irq+0xcc/0x180 (IRQ A) [14.816255] arch_cpu_idle+0x34/0x1c8 [14.816255] default_idle_call+0x24/0x44 [14.816256] do_idle+0x1d0/0x2c8 [14.816256] cpu_startup_entry+0x28/0x30 [14.816256] rest_init+0xb8/0xc8 [14.816257] start_kernel+0x4c8/0x4f4 [14.816257] Code: 940587f1 d5384100 b9401001 36a7fd01 (d4210000) [14.816258] Modules linked in: start_dp(O) smeth(O) [15.103092] ---[end trace 701753956cb14aa8]--- [15.103093] Kernel panic - not syncing: Fatal exception in interrupt [15.103099] SMP: stopping secondary CPUs [15.103100] Kernel Offset: disabled [15.103100] CPU features: 0x36,a2400218 [15.103100] Memory Limit: none which is cause by a 'BUG_ON(in_nmi())' in nmi_enter(). From the call trace, we can find three interrupts (noted A, B, C above): interrupt (A) is preempted by (B), which is further interrupted by (C). Subsequent investigations show that (B) results in nmi_enter() being called, but that it actually is a spurious interrupt. Furthermore, interrupts are reenabled in the context of (B), and (C) fires with NMI priority. We end-up with a nested NMI situation, something we definitely do not want to (and cannot) handle. The bug here is that spurious interrupts should never result in any state change, and we should just return to the interrupted context. Moving the handling of spurious interrupts as early as possible in the GICv3 handler fixes this issue. [maz: rewrote commit message, corrected Fixes: tag]

CVE ID	Description
CVE-2021-46963	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix crash in qla2xxx_mqueuecommand() RIP: 0010:kmem_cache_free+0xfa/0x1b0 Call Trace: qla2xxx_mqueuecommand+0x2b5/0x2c0 [qla2xxx] scsi_queue_rq+0x5e2/0xa40 __blk_mq_try_issue_directly+0x128/0x1d0 blk_mq_request_issue_directly+0x4e/0xb0 Fix incorrect call to free srb in qla2xxx_mqueuecommand(), as srb is now allocated by upper layers. This fixes smatch warning of srb unintended free.
CVE-2021-46973	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: Avoid potential use after free in MHI send It is possible that the MHI ul_callback will be invoked immediately following the queueing of the skb for transmission, leading to the callback decrementing the refcount of the associated sk and freeing the skb. As such the dereference of skb and the increment of the sk refcount must happen before the skb is queued, to avoid the skb to be used after free and potentially the sk to drop its last refcount..
CVE-2021-46976	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix crash in auto_retire The retire logic uses the 2 lower bits of the pointer to the retire function to store flags. However, the auto_retire function is not guaranteed to be aligned to a multiple of 4, which causes crashes as we jump to the wrong address, for example like this: 2021-04-24T18:03:53.804300Z WARNING kernel: [516.876901] invalid opcode: 0000 [#1] PREEMPT SMP NOPTI 2021-04-24T18:03:53.804310Z WARNING kernel: [516.876906] CPU: 7 PID: 146 Comm: kworker/u16:6 Tainted: G U 5.4.105-13595-g3cd84167b2df #1 2021-04-24T18:03:53.804311Z WARNING kernel: [516.876907] Hardware name: Google Volteer2/Volteer2, BIOS Google_Volteer2.13672.76.0 02/22/2021 2021-04-24T18:03:53.804312Z WARNING kernel: [516.876911] Workqueue: events_unbound active_work 2021-04-24T18:03:53.804313Z WARNING kernel: [516.876914] RIP: 0010:auto_retire+0x1/0x20 2021-04-24T18:03:53.804314Z WARNING kernel: [516.876916] Code: e8 01 f2 ff ff eb 02 31 db 48 89 d8 5b 5d c3 0f 1f 44 00 00 55 48 89 e5 f0 ff 87 c8 00 00 00 0f 8b ab 47 4a 00 31 c0 5d c3 0f <1f> 44 00 00 55 48 89 e5 f0 ff 8f c8 00 00 00 0f 88 9a 47 4a 00 74 2021-04-24T18:03:53.804319Z WARNING kernel: [516.876918] RSP: 0018:ffff9b4d809fbc38 EFLAGS: 00010286 2021-04-24T18:03:53.804320Z WARNING kernel: [516.876919] RAX: 0000000000000007 RBX: ffff927915079600 RCX: 0000000000000007 2021-04-24T18:03:53.804320Z WARNING kernel: [516.876921] RDX: ffff9b4d809fbc40 RSI: 0000000000000286 RDI: ffff927915079600 2021-04-24T18:03:53.804321Z WARNING kernel: [516.876922] RBP: ffff9b4d809fbc68 R08: 8080808080808080 R09: fefefefefefeff 2021-04-24T18:03:53.804321Z WARNING kernel: [516.876924] R10: 0000000000000010 R11: ffffffff92e44bd8 R12: ffff9279150796a0 2021-04-24T18:03:53.804322Z WARNING kernel: [516.876925] R13: ffff92791c368180 R14: ffff927915079640 R15: 000000001c867605 2021-04-24T18:03:53.804323Z WARNING kernel: [516.876926] FS: 0000000000000000(0000) GS:ffff92791ffc0000(0000) knlGS:0000000000000000 2021-04-24T18:03:53.804323Z WARNING kernel: [516.876928] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 2021-04-24T18:03:53.804324Z WARNING kernel: [516.876929] CR2: 0000239514955000 CR3: 000000007f82da001 CR4: 0000000000760ee0 2021-04-24T18:03:53.804325Z WARNING kernel: [516.876930] PKRU: 55555554 2021-04-24T18:03:53.804325Z WARNING kernel: [516.876931] Call Trace: 2021-04-24T18:03:53.804326Z WARNING kernel: [516.876935] __active_retire+0x77/0xcf 2021-04-24T18:03:53.804326Z WARNING kernel: [516.876939] process_one_work+0x1da/0x394 2021-04-24T18:03:53.804327Z WARNING kernel: [516.876941] worker_thread+0x216/0x375 2021-04-24T18:03:53.804327Z WARNING kernel: [516.876944] kthread+0x147/0x156 2021-04-24T18:03:53.804335Z WARNING kernel: [516.876946] ? pr_cont_work+0x58/0x58 2021-04-24T18:03:53.804335Z WARNING kernel: [516.876948] ? kthread_blkcg+0x2e/0x2e 2021-04-24T18:03:53.804336Z WARNING kernel: [516.876950] ret_from_fork+0x1f/0x40 2021-04-24T18:03:53.804336Z WARNING kernel: [516.876952] Modules linked in: cdc_mbim cdc_ncm cdc_wdm xt_cgroup rfcomm cmac algif_hash algif_skcipher af_alg xt_MASQUERADE uinput snd_soc_rt5682_sdw snd_soc_rt5682 snd_soc_max98373_sdw snd_soc_max98373 snd_soc_rl6231 regmap_sdw snd_soc_sof_sdw snd_soc_hdac_hdmi snd_soc_dmic snd_hda_codec_hdmi snd_sof_pci snd_sof_intel_hda_common intel_ipu6_psys snd_sof_xtensa_dsp soundwire_intel soundwire_generic_allocation soundwire_cadence snd_sof_intel_hda snd_sof snd_soc_hdac_hda snd_soc_acpi_intel_match snd_soc_acpi snd_hda_ext_core soundwire_bus snd_hda_intel snd_intel_dspcfg snd_hda_codec snd_hwdep snd_hda_core intel_ipu6_isys videobuf2_dma_contig videobuf2_v4l2 videobuf2_common videobuf2_memops mei_hdcp intel_ipu6 ov2740 ov8856 at24 sx9310 dw9768 v4l2_fwnode cros_ec_typec intel_pmc_mux roles acpi_als typec fuse iio_trig_sysfs cros_ec_light_prox cros_ec_lid_angle cros_ec_sensors cros ---truncated---

CVE ID	Description
CVE-2021-46978	In the Linux kernel, the following vulnerability has been resolved: KVM: nVMX: Always make an attempt to map eVMCS after migration When enlightened VMCS is in use and nested state is migrated with vmx_get_nested_state()/vmx_set_nested_state() KVM can't map evmcs page right away: evmcs gpa is not 'struct kvm_vmx_nested_state_hdr' and we can't read it from VP assist page because userspace may decide to restore HV_X64_MSR_VP_ASSIST_PAGE after restoring nested state (and QEMU, for example, does exactly that). To make sure eVMCS is mapped / vmx_set_nested_state() raises KVM_REQ_GET_NESTED_STATE_PAGES request. Commit f2c7ef3ba955 ("KVM: nSVM: cancel KVM_REQ_GET_NESTED_STATE_PAGES on nested vmexit") added KVM_REQ_GET_NESTED_STATE_PAGES clearing to nested_vmx_vmxexit() to make sure MSR permission bitmap is not switched when an immediate exit from L2 to L1 happens right after migration (caused by a pending event, for example). Unfortunately, in the exact same situation we still need to have eVMCS mapped so nested_sync_vmcs12_to_shadow() reflects changes in VMCS12 to eVMCS. As a band-aid, restore nested_get_evmcs_page() when clearing KVM_REQ_GET_NESTED_STATE_PAGES in nested_vmx_vmxexit(). The 'fix' is far from being ideal as we can't easily propagate possible failures and even if we could, this is most likely already too late to do so. The whole 'KVM_REQ_GET_NESTED_STATE_PAGES' idea for mapping eVMCS after migration seems to be fragile as we diverge too much from the 'native' path when vmptr loading happens on vmx_set_nested_state().
CVE-2021-46988	In the Linux kernel, the following vulnerability has been resolved: userfaultfd: release page in error path to avoid BUG_ON Consider the following sequence of events: 1. Userspace issues a UFFD ioctl, which ends up calling into shmem_mfill_atomic_pte(). We successfully account the blocks, we shmem_alloc_page(), but then the copy_from_user() fails. We return -ENOENT. We don't release the page we allocated. 2. Our caller detects this error code, tries the copy_from_user() after dropping the mmap_lock, and retries, calling back into shmem_mfill_atomic_pte(). 3. Meanwhile, let's say another process filled up the tmpfs being used. 4. So shmem_mfill_atomic_pte() fails to account blocks this time, and immediately returns - without releasing the page. This triggers a BUG_ON in our caller, which asserts that the page should always be consumed, unless -ENOENT is returned. To fix this, detect if we have such a "dangling" page when accounting fails, and if so, release it before returning.
CVE-2021-46997	In the Linux kernel, the following vulnerability has been resolved: arm64: entry: always set GIC_PRIO_PSR_I_SET during entry Zenghui reports that booting a kernel with "irqchip.gicv3_pseudo_nmi=1" on the command line hits a warning during kernel entry, due to the way we manipulate the PMR. Early in the entry sequence, we call lockdep_hardirqs_off() to inform lockdep that interrupts have been masked (as the HW sets DAIF when entering an exception). Architecturally PMR_EL1 is not affected by exception entry, and we don't set GIC_PRIO_PSR_I_SET in the PMR early in the exception entry sequence, so early in exception entry the PMR can indicate that interrupts are unmasked even though they are masked by DAIF. If DEBUG_LOCKDEP is selected, lockdep_hardirqs_off() will check that interrupts are masked, before we set GIC_PRIO_PSR_I_SET in any of the exception entry paths, and hence lockdep_hardirqs_off() will WARN() that something is amiss. We can avoid this by consistently setting GIC_PRIO_PSR_I_SET during exception entry so that kernel code sees a consistent environment. We must also update local_daif_inherit() to undo this, as currently only touches DAIF. For other paths, local_daif_restore() will update both DAIF and the PMR. With this done, we can remove the existing special cases which set this later in the entry code. We always use (GIC_PRIO_IRQON GIC_PRIO_PSR_I_SET) for consistency with local_daif_save(), as this will warn if it ever encounters (GIC_PRIO_IRQOFF GIC_PRIO_PSR_I_SET), and never sets this itself. This matches the gic_prio_kentry_setup that we have to retain for ret_to_user. The original splat from Zenghui's report was: DEBUG_LOCKS_WARN_ON(!irqs_disabled()) WARNING: CPU: 3 PID: 125 at kernel/locking/lockdep.c:4258 lockdep_hardirqs_off+0xd4/0xe8 Modules linked in: CPU: 3 PID: 125 Comm: modprobe Tainted: G W 5.12.0-rc8+ #463 Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 pstate: 604003c5 (nZCv DAIF +PAN -UAO -TCO BTYPE=--) pc : lockdep_hardirqs_off+0xd4/0xe8 lr : lockdep_hardirqs_off+0xd4/0xe8 sp : ffff80002a39bad0 pmr_save: 000000e0 x29: ffff80002a39bad0 x28: ffff0000de214bc0 x27: ffff0000de1c0400 x26: 000000000049b328 x25: 0000000000406f30 x24: ffff0000de1c00a0 x23: 0000000020400005 x22: ffff8000105f747c x21: 0000000096000044 x20: 0000000000498ef9 x19: ffff80002a39bc88 x18: ffffffff7fffffff x17: 0000000000000000 x16: ffff800011c61eb0 x15: ffff800011700a88 x14: 0720072007200720 x13: 0720072007200720 x12: 0720072007200720 x11: 0720072007200720 x10: 0720072007200720 x9: ffff80002a39bad0 x8: ffff80002a39bad0 x7: ffff8000119f0800 x6: c0000000ffff7fff x5: ffff8000119f07a8 x4: 0000000000000001 x3: 9bcdab23f2432800 x2: ffff800011730538 x1: 9bcdab23f2432800 x0: 0000000000000000 Call trace: lockdep_hardirqs_off+0xd4/0xe8 enter_from_kernel_mode.isra.5+0x7c/0xa8 el1_abort+0x24/0x100 el1_sync_handler+0x80/0xd0 el1_sync+0x6c/0x100 __arch_clear_user+0xc/0x90 load_elf_binary+0x9fc/0x1450 bprm_execve+0x404/0x880 kernel_execve+0x180/0x188 call_usermodehelper_exec_async+0xdc/0x158 ret_from_fork+0x10/0x18

CVE ID	Description
CVE-2021-46999	In the Linux kernel, the following vulnerability has been resolved: sctp: do asoc update earlier in sctp_sf_do_dupcook_a There's a panic that occurs in a few of envs, the call trace is as below: [] general protection fault, ... 0x29acd70f1000a: 0000 [#1] SMP PTI [] RIP: 0010:sctp_ulpevent_notify_peer_addr_change+0x4b/0x1fa [sctp] [] sctp_assoc_control_transport+0x1b9/0x210 [sctp] [] sctp_do_8_2_transport_strike.isra.16+0x15c/0x220 [sctp] [] sctp_cmd_interpreter.isra.21+0x1231/0x1a10 [sctp] [] sctp_do_sm+0xc3/0x2a0 [sctp] [] sctp_generate_timeout_event+0x81/0xf0 [sctp] This is caused by a transport use-after-free issue. When processing a duplicate COOKIE-ECHO chunk in sctp_sf_do_dupcook_a(), both COOKIE-ACK and SHUTDOWN chunks are allocated with the transport from the new asoc. However, later in the sideeffect machine, the old asoc is used to send them out and old asoc's shutdown_last_sent_to is set to the transport that SHUTDOWN chunk attached to in sctp_cmd_setup_t2(), which actually belongs to the new asoc. After the new_asoc is freed and the old asoc T2 timeout, the old asoc's shutdown_last_sent_to that is already freed would be accessed in sctp_sf_t2_timer_expire(). Thanks Alexander and Jere for helping dig into this issue. To fix it, this patch is to do the asoc update first, then allocate the COOKIE-ACK and SHUTDOWN chunks with the 'updated' old asoc. This would make more sense, as a chunk from an asoc shouldn't be sent out with another asoc. We had fixed quite a few issues caused by this.
CVE-2021-47001	In the Linux kernel, the following vulnerability has been resolved: xprtrdma: Fix cwnd update ordering After a reconnect, the reply handler is opening the cwnd (and thus enabling more RPC Calls to be sent) / before/ rprdma_post_recv() can post enough Receive WRs to receive their replies. This causes an RNR and the new connection is lost immediately. The race is most clearly exposed when KASAN and disconnect injection are enabled. This slows down rprdma_rep_create() enough to allow the send side to post a bunch of RPC Calls before the Receive completion handler can invoke ib_post_recv().
CVE-2021-47003	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix potential null dereference on pointer status There are calls to idxd_cmd_exec that pass a null status pointer however a recent commit has added an assignment to *status that can end up with a null pointer dereference. The function expects a null status pointer sometimes as there is a later assignment to *status where status is first null checked. Fix the issue by null checking status before making the assignment. Addresses-Coverity: ("Explicit null dereferenced")
CVE-2021-47010	In the Linux kernel, the following vulnerability has been resolved: net: Only allow init netns to set default tcp cong to a restricted algo tcp_set_default_congestion_control() is netns-safe in that it writes to &net->ipv4.tcp_congestion_control, but it also sets ca->flags = TCP_CONG_NON_RESTRICTED which is not namespaced. This has the unintended side-effect of changing the global net.ipv4.tcp_allowed_congestion_control sysctl, despite the fact that it is read-only: 97684f0970f6 ("net: Make tcp_allowed_congestion_control readonly in non-init netns") Resolve this netns "leak" by only allowing the init netns to set the default algorithm to one that is restricted. This restriction could be removed if tcp_allowed_congestion_control were namespace-ified in the future. This bug was uncovered with https://github.com/JonathonReinhart/linux-netns-sysctl-verify
CVE-2021-47011	In the Linux kernel, the following vulnerability has been resolved: mm: memcontrol: slab: fix obtain a reference to a freeing memcg Patch series "Use obj_cgroup APIs to charge kmem pages", v5. Since Roman's series "The new cgroup slab memory controller" applied. All slab objects are charged with the new APIs of obj_cgroup. The new APIs introduce a struct obj_cgroup to charge slab objects. It prevents long-living objects from pinning the original memory cgroup in the memory. But there are still some corner objects (e.g. allocations larger than order-1 page on SLUB) which are not charged with the new APIs. Those objects (include the pages which are allocated from buddy allocator directly) are charged as kmem pages which still hold a reference to the memory cgroup. E.g. We know that the kernel stack is charged as kmem pages because the size of the kernel stack can be greater than 2 pages (e.g. 16KB on x86_64 or arm64). If we create a thread (suppose the thread stack is charged to memory cgroup A) and then move it from memory cgroup A to memory cgroup B. Because the kernel stack of the thread hold a reference to the memory cgroup A. The thread can pin the memory cgroup A in the memory even if we remove the cgroup A. If we want to see this scenario by using the following script. We can see that the system has added 500 dying cgroups (This is not a real world issue, just a script to show that the large kmmallocs are charged as kmem pages which can pin the memory cgroup in the memory). #!/bin/bash cat /proc/cgroups grep memory cd /sys/fs/cgroup/memory echo 1 > memory.move_charge_at_immigrate for i in range{1..500} do mkdir kmem_test echo \$\$ > kmem_test/cgroup.procs sleep 3600 & echo \$\$ > cgroup.procs echo `cat kmem_test/cgroup.procs` > cgroup.procs rmdir kmem_test done cat /proc/cgroups grep memory This patchset aims to make those kmem pages to drop the reference to memory cgroup by using the APIs of obj_cgroup. Finally, we can see that the number of the dying cgroups will not increase if we run the above test script. This patch (of 7): The rcu_read_lock/unlock only can guarantee that the memcg will not be freed, but it cannot guarantee the success of css_get (which is in the refill_stock when cached memcg changed) to memcg. rcu_read_lock() memcg = obj_cgroup_memcg(old) __memcg_kmem_uncharge(memcg) refill_stock(memcg) if (stock->cached != memcg) // css_get can change the ref counter from 0 back to 1. css_get(&memcg->css) rcu_read_unlock() This fix is very like the commit: eefbfa7fd678 ("mm: memcg/slab: fix use after free in obj_cgroup_charge") Fix this by holding a reference to the memcg which is passed to the __memcg_kmem_uncharge() before calling __memcg_kmem_uncharge().

CVE ID	Description
CVE-2021-47012	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Fix a use after free in siw_alloc_mr Our code analyzer reported a UAF. In siw_alloc_mr(), it calls siw_mr_add_mem(mr,...). In the implementation of siw_mr_add_mem(), mem is assigned to mr->mem and then mem is freed via kfree(mem) if xa_alloc_cyclic() failed. Here, mr->mem still point to a freed object. After, the execution continue up to the err_out branch of siw_alloc_mr, and the freed mr->mem is used in siw_mr_drop_mem(mr). My patch moves "mr->mem = mem" behind the if (xa_alloc_cyclic(<0>) {} section, to avoid the uaf.
CVE-2021-47014	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_ct: fix wild memory access when clearing fragments while testing re-assembly/re-fragmentation using act_ct, it's possible to observe a crash like the following one: KASAN: maybe wild-memory-access in range [0x0001000000000448-0x000100000000044f] CPU: 50 PID: 0 Comm: swapper/50 Tainted: G S 5.12.0-rc7+ #424 Hardware name: Dell Inc. PowerEdge R730/072T6D, BIOS 2.4.3 01/17/2017 RIP: 0010:inet_frag_rbtrees_purge+0x50/0xc0 Code: 00 fc ff df 48 89 c3 31 ed 48 89 df e8 a9 7a 38 ff 4c 89 fe 48 89 df 49 89 c6 e8 5b 3a 38 ff 48 8d 7b 40 48 89 f8 48 c1 e8 03 <42> 80 3c 20 00 75 59 48 8d bd d0 00 00 00 4c 8b 6b 40 48 89 f8 48 RSP: 0018:ffff888c31449db8 EFLAGS: 00010203 RAX: 0000200000000089 RBX: 000100000000040e RCX: ffffffff989eb960 RDX: 0000000000000140 RSI: ffffffff97cfb977 RDI: 000100000000044e RBP: 0000000000000900 R08: 0000000000000000 R09: fffffd1186289350 R10: 0000000000000003 R11: fffffd1186289350 R12: dffffc0000000000 R13: 000100000000040e R14: 0000000000000000 R15: ffff888155e02160 FS: 0000000000000000(0000) GS:ffff888c31440000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005600cb70a5b8 CR3: 0000000a2c014005 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <IRQ> inet_frag_destroy+0xa9/0x150 call_timer_fn+0x2d/0x180 run_timer_softirq+0x4fe/0xe70 __do_softirq+0x197/0x5a0 irq_exit_rcu+0x1de/0x200 sysvec_apic_timer_interrupt+0x6b/0x80 </IRQ> when act_ct temporarily stores an IP fragment, restoring the skb qdisc cb results in putting random data in FRAG_CB(), and this causes those "wild" memory accesses later, when the rbtree is purged. Never overwrite the skb cb in case tcf_ct_handle_fragments() returns -EINPROGRESS.
CVE-2021-47017	In the Linux kernel, the following vulnerability has been resolved: ath10k: Fix a use after free in ath10k_htc_send_bundle In ath10k_htc_send_bundle, the bundle_skb could be freed by dev_kfree_skb_any(bundle_skb). But the bundle_skb is used later by bundle_skb->len. As skb_len = bundle_skb->len, my patch replaces bundle_skb->len to skb_len after the bundle_skb was freed.
CVE-2021-47020	In the Linux kernel, the following vulnerability has been resolved: soundwire: stream: fix memory leak in stream config error path When stream config is failed, master runtime will release all slave runtime in the slave_rt_list, but slave runtime is not added to the list at this time. This patch frees slave runtime in the config error path to fix the memory leak.
CVE-2021-47024	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: free queued packets when closing socket As reported by syzbot [1], there is a memory leak while closing the socket. We partially solved this issue with commit ac03046ece2b ("vsock/virtio: free packets during the socket release"), but we forgot to drain the RX queue when the socket is definitely closed by the scheduled work. To avoid future issues, let's use the new virtio_transport_remove_sock() to drain the RX queue before removing the socket from the af_vsock lists calling vsock_remove_sock(). [1] https://syzkaller.appspot.com/bug?extid=24452624fc4c571eadd9
CVE-2021-47036	In the Linux kernel, the following vulnerability has been resolved: udp: skip L4 aggregation for UDP tunnel packets If NETIF_F_GRO_FRAGLIST or NETIF_F_GRO_UDP_FWD are enabled, and there are UDP tunnels available in the system, udp_gro_receive() could end-up doing L4 aggregation (either SKB_GSO_UDP_L4 or SKB_GSO_FRAGLIST) at the outer UDP tunnel level for packets effectively carrying and UDP tunnel header. That could cause inner protocol corruption. If e.g. the relevant packets carry a vxlan header, different vxlan ids will be ignored/ aggregated to the same GSO packet. Inner headers will be ignored, too, so that e.g. TCP over vxlan push packets will be held in the GRO engine till the next flush, etc. Just skip the SKB_GSO_UDP_L4 and SKB_GSO_FRAGLIST code path if the current packet could land in a UDP tunnel, and let udp_gro_receive() do GRO via udp_sk(sk)->gro_receive. The check implemented in this patch is broader than what is strictly needed, as the existing UDP tunnel could be e.g. configured on top of a different device: we could end-up skipping GRO at all for some packets. Anyhow, that is a very thin corner case and covering it will add quite a bit of complexity. v1 -> v2: - hopefully clarify the commit message

CVE ID	Description
CVE-2021-47038	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: avoid deadlock between hci_dev->lock and socket lock Commit eab2404ba798 ("Bluetooth: Add BT_PHY socket option") added a dependency between socket lock and hci_dev->lock that could lead to deadlock. It turns out that hci_conn_get_phy() is not in any way relying on hdev being immutable during the runtime of this function, neither does it even look at any of the members of hdev, and as such there is no need to hold that lock. This fixes the lockdep splat below: <pre>===== WARNING: possible circular locking dependency detected 5.12.0-rc1-00026-g73d464503354 #10 Not tainted ----- bluetoothd/1118 is trying to acquire lock: ffff8f078383c078 (&hdev->lock){+..}-{3:3}, at: hci_conn_get_phy+0x1c/0x150 [bluetooth] but task is already holding lock: ffff8f07e831d920 (sk_lock-AF_BLUETOOTH-BTPROTO_L2CAP){+..}-{0:0}, at: l2cap_sock_getsockopt+0x8b/0x610 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #3 (sk_lock-AF_BLUETOOTH-BTPROTO_L2CAP){+..}- {0:0}: lock_sock_nested+0x72/0xa0 l2cap_sock_ready_cb+0x18/0x70 [bluetooth] l2cap_config_rsp +0x27a/0x520 [bluetooth] l2cap_sig_channel+0x658/0x1330 [bluetooth] l2cap_recv_frame +0x1ba/0x310 [bluetooth] hci_rx_work+0x1cc/0x640 [bluetooth] process_one_work+0x244/0x5f0 worker_thread+0x3c/0x380 kthread+0x13e/0x160 ret_from_fork+0x22/0x30 -> #2 (&chan- >lock#2/1){+..}-{3:3}: __mutex_lock+0xa3/0xa10 l2cap_chan_connect+0x33a/0x940 [bluetooth] l2cap_sock_connect+0x141/0x2a0 [bluetooth] __sys_connect+0x9b/0xc0 __x64_sys_connect +0x16/0x20 do_syscall_64+0x33/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae -> #1 (&conn- >chan_lock){+..}-{3:3}: __mutex_lock+0xa3/0xa10 l2cap_chan_connect+0x322/0x940 [bluetooth] l2cap_sock_connect+0x141/0x2a0 [bluetooth] __sys_connect+0x9b/0xc0 __x64_sys_connect +0x16/0x20 do_syscall_64+0x33/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae -> #0 (&hdev- >lock){+..}-{3:3}: __lock_acquire+0x147a/0x1a50 lock_acquire+0x277/0x3d0 __mutex_lock +0xa3/0xa10 hci_conn_get_phy+0x1c/0x150 [bluetooth] l2cap_sock_getsockopt+0x5a9/0x610 [bluetooth] __sys_getsockopt+0xcc/0x200 __x64_sys_getsockopt+0x20/0x30 do_syscall_64+0x33/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae other info that might help us debug this: Chain exists of: &hdev->lock --> &chan->lock#2/1 --> sk_lock-AF_BLUETOOTH-BTPROTO_L2CAP Possible unsafe locking scenario: CPU0 CPU1 ---- ---- lock(sk_lock-AF_BLUETOOTH-BTPROTO_L2CAP); lock(&chan->lock#2/1); lock(sk_lock-AF_BLUETOOTH-BTPROTO_L2CAP); lock(&hdev- >lock); *** DEADLOCK *** 1 lock held by bluetoothd/1118: #0: ffff8f07e831d920 (sk_lock- AF_BLUETOOTH-BTPROTO_L2CAP){+..}-{0:0}, at: l2cap_sock_getsockopt+0x8b/0x610 [bluetooth] stack backtrace: CPU: 3 PID: 1118 Comm: bluetoothd Not tainted 5.12.0-rc1-00026- g73d464503354 #10 Hardware name: LENOVO 20K5S22R00/20K5S22R00, BIOS R0IET38W (1.16) 05/31/2017 Call Trace: dump_stack+0x7f/0xa1 check_noncircular+0x105/0x120 ? __lock_acquire +0x147a/0x1a50 __lock_acquire+0x147a/0x1a50 lock_acquire+0x277/0x3d0 ? hci_conn_get_phy +0x1c/0x150 [bluetooth] ? __lock_acquire+0x2e1/0x1a50 ? lock_is_held_type+0xb4/0x120 ? hci_conn_get_phy+0x1c/0x150 [bluetooth] __mutex_lock+0xa3/0xa10 ? hci_conn_get_phy +0x1c/0x150 [bluetooth] ? lock_acquire+0x277/0x3d0 ? mark_held_locks+0x49/0x70 ? mark_held_locks+0x49/0x70 ? hci_conn_get_phy+0x1c/0x150 [bluetooth] hci_conn_get_phy+0x truncated---</pre>

CVE ID	Description
CVE-2021-47041	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: fix incorrect locking in state_change sk callback We are not changing anything in the TCP connection state so we should not take a write_lock but rather a read lock. This caused a deadlock when running nvmet-tcp and nvme-tcp on the same system, where state_change callbacks on the host and on the controller side have causal relationship and made lockdep report on this with blktests: ===== WARNING: inconsistent lock state 5.12.0-rc3 #1 Tainted: G I ----- inconsistent {IN-SOFTIRQ-W} -> {SOFTIRQ-ON-R} usage. nvme/1324 [HC0[0]:SC0[0]:HE1:SE1] takes: ffff888363151000 (clock-AF_INET){++-?}-{2:2}, at: nvme_tcp_state_change+0x21/0x150 [nvme_tcp] {IN-SOFTIRQ-W} state was registered at: __lock_acquire+0x79b/0x18d0 lock_acquire+0x1ca/0x480 __raw_write_lock_bh+0x39/0x80 nvmet_tcp_state_change+0x21/0x170 [nvmet_tcp] tcp_fin+0x2a8/0x780 tcp_data_queue+0xf94/0x1f20 tcp_rcv_established+0x6ba/0x1f00 tcp_v4_do_rcv+0x502/0x760 tcp_v4_rcv+0x257e/0x3430 ip_protocol_deliver_rcu+0x69/0x6a0 ip_local_deliver_finish+0x1e2/0x2f0 ip_local_deliver+0x1a2/0x420 ip_rcv+0x4fb/0x6b0 __netif_receive_skb_one_core+0x162/0x1b0 process_backlog+0x1ff/0x770 __napi_poll.constprop.0+0xa9/0x5c0 net_rx_action+0x7b3/0xb30 __do_softirq+0x1f0/0x940 do_softirq+0xa1/0xd0 __local_bh_enable_ip+0xd8/0x100 ip_finish_output2+0x6b7/0x18a0 __ip_queue_xmit+0x706/0x1aa0 __tcp_transmit_skb+0x2068/0x2e20 tcp_write_xmit+0xc9e/0x2bb0 __tcp_push_pending_frames+0x92/0x310 inet_shutdown+0x158/0x300 __nvme_tcp_stop_queue+0x36/0x270 [nvme_tcp] nvme_tcp_stop_queue+0x87/0xb0 [nvme_tcp] nvme_tcp_teardown_admin_queue+0x69/0xe0 [nvme_tcp] nvme_do_delete_ctrl+0x100/0x10c [nvme_core] nvme_sysfs_delete.cold+0x8/0xd [nvme_core] kernfs_fop_write_iter+0x2c7/0x460 new_sync_write+0x36c/0x610 vfs_write+0x5c0/0x870 ksys_write+0xf9/0x1d0 do_syscall_64+0x33/0x40 entry_SYSCALL_64_after_hwframe+0x44/0xae irq event stamp: 10687 hardirqs last enabled at (10687): [<fffffff9ec376bd>] _raw_spin_unlock_irqrestore+0x2d/0x40 hardirqs last disabled at (10686): [<fffffff9ec374d8>] _raw_spin_lock_irqsave+0x68/0x90 softirqs last enabled at (10684): [<fffffff9f000608>] __do_softirq+0x608/0x940 softirqs last disabled at (10649): [<fffffff9cdedd31>] do_softirq+0xa1/0xd0 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(clock-AF_INET); <Interrupt> lock(clock-AF_INET); *** DEADLOCK *** 5 locks held by nvme/1324: #0: ffff8884a01fe470 (sb_writers#4) {+.+}-{0:0}, at: ksys_write+0xf9/0x1d0 #1: ffff8886e435c090 (&of->mutex){+.+}-{3:3}, at: kernfs_fop_write_iter+0x216/0x460 #2: ffff888104d90c38 (kn->active#255){++++}-{0:0}, at: kernfs_remove_self+0x22d/0x330 #3: ffff8884634538d0 (&queue->queue_lock){+.+}-{3:3}, at: nvme_tcp_stop_queue+0x52/0xb0 [nvme_tcp] #4: ffff888363150d30 (sk_lock-AF_INET) {+.+}-{0:0}, at: inet_shutdown+0x59/0x300 stack backtrace: CPU: 26 PID: 1324 Comm: nvme Tainted: G I 5.12.0-rc3 #1 Hardware name: Dell Inc. PowerEdge R640/06NR82, BIOS 2.10.0 11/12/2020 Call Trace: dump_stack+0x93/0xc2 mark_lock_irq.cold+0x2c/0xb3 ? verify_lock_unused+0x390/0x390 ? stack_trace_consume_entry+0x160/0x160 ? lock_downgrade+0x100/0x100 ? save_trace+0x88/0x5e0 ? _raw_spin_unlock_irqrestore+0x2d/0x40 mark_lock+0x530/0x1470 ? mark_lock_irq+0x1d10/0x1d10 ? enqueue_timer+0x660/0x660 mark_usage+0x215/0x2a0 __lock_acquire+0x79b/0x18d0 ? tcp_schedule_loss_probe.part.0+0x38c/0x520 lock_acquire+0x1ca/0x480 ? nvme_tcp_state_change+0x21/0x150 [nvme_tcp] ? rcu_read_unlock+0x40/0x40 ? tcp_mtu_probe+0x1ae0/0x1ae0 ? kmalloc_reserve+0xa0/0xa0 ? sysfs_file_ops+0x170/0x170 _raw_read_lock+0x3d/0xa0 ? nvme_tcp_state_change+0x21/0x150 [nvme_tcp] nvme_tcp_state_change+0x21/0x150 [nvme_tcp] ? sysfs_file_ops ---truncated---
CVE-2021-47045	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix null pointer dereference in lpfc_prep_els_iocb() It is possible to call lpfc_issue_els_plogi() passing a did for which no matching ndlp is found. A call is then made to lpfc_prep_els_iocb() with a null pointer to a lpfc_nodelist structure resulting in a null pointer dereference. Fix by returning an error status if no valid ndlp is found. Fix up comments regarding ndlp reference counting.
CVE-2021-47070	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Fix another memory leak in error handling paths Memory allocated by 'vmbus_alloc_ring()' at the beginning of the probe function is never freed in the error handling path. Add the missing 'vmbus_free_ring()' call. Note that it is already freed in the .remove function.
CVE-2021-47071	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Fix a memory leak in error handling paths If 'vmbus_establish_gpadl()' fails, the (recv/send)_gpadl will not be updated and 'hv_uio_cleanup()' in the error handling path will not be able to free the corresponding buffer. In such a case, we need to free the buffer explicitly.
CVE-2021-47074	In the Linux kernel, the following vulnerability has been resolved: nvme-loop: fix memory leak in nvme_loop_create_ctrl() When creating loop ctrl in nvme_loop_create_ctrl(), if nvme_init_ctrl() fails, the loop ctrl should be freed before jumping to the "out" label.
CVE-2021-47075	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix memory leak in nvmet_alloc_ctrl() When creating ctrl in nvmet_alloc_ctrl(), if the cntlid_min is larger than cntlid_max of the subsystem, and jumps to the "out_free_changed_ns_list" label, but the ctrl->sqs lack of be freed. Fix this by jumping to the "out_free_sqs" label.

CVE ID	Description
CVE-2021-47077	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Add pointer checks in qedf_update_link_speed() The following trace was observed: [14.042059] Call Trace: [14.042061] <IRQ> [14.042068] qedf_link_update+0x144/0x1f0 [qedf] [14.042117] qed_link_update+0x5c/0x80 [qed] [14.042135] qed_mcp_handle_link_change+0x2d2/0x410 [qed] [14.042155] ? qed_set_ptt+0x70/0x80 [qed] [14.042170] ? qed_set_ptt+0x70/0x80 [qed] [14.042186] ? qed_rd+0x13/0x40 [qed] [14.042205] qed_mcp_handle_events+0x437/0x690 [qed] [14.042221] ? qed_set_ptt+0x70/0x80 [qed] [14.042239] qed_int_sp_dpc+0x3a6/0x3e0 [qed] [14.042245] tasklet_action_common.isra.14+0x5a/0x100 [14.042250] __do_softirq+0xe4/0x2f8 [14.042253] irq_exit+0xf7/0x100 [14.042255] do_IRQ+0x7f/0xd0 [14.042257] common_interrupt+0xf/0xf [14.042259] </IRQ> API qedf_link_update() is getting called from QED but by that time shost_data is not initialised. This results in a NULL pointer dereference when we try to dereference shost_data while updating supported_speeds. Add a NULL pointer check before dereferencing shost_data.
CVE-2021-47078	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Clear all QP fields if creation failed rxe_qp_do_cleanup() relies on valid pointer values in QP for the properly created ones, but in case rxe_qp_from_init() failed it was filled with garbage and caused tot the following error. refcount_t: underflow; use-after-free. WARNING: CPU: 1 PID: 12560 at lib/refcount.c:28 refcount_warn_saturate+0x1d1/0x1e0 lib/refcount.c:28 Modules linked in: CPU: 1 PID: 12560 Comm: syz-executor.4 Not tainted 5.12.0-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:refcount_warn_saturate+0x1d1/0x1e0 lib/refcount.c:28 Code: e9 db fe ff ff 48 89 df e8 2c e2 ea fd e9 8a fe ff ff e8 72 6a a7 fd 48 c7 c7 e0 b2 c1 89 c6 05 dc 3a e6 09 01 e8 ee 74 fb 04 <of> 0b e9 af fe ff ff 0f 1f 84 00 00 00 00 00 41 56 41 55 41 54 55 RSP: 0018:ffff900097ceba8 EFLAGS: 00010286 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 00000000000040000 RSI: ffffffff815bb075 RDI: fffff520012f9d67 RBP: 0000000000000003 R08: 0000000000000000 R09: 0000000000000000 R10: ffffffff815b4eae R11: 0000000000000000 R12: ffff8880322a4800 R13: ffff8880322a4940 R14: ffff888033044e00 R15: 0000000000000000 FS: 00007f6eb2be3700(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f6e5d41000 CR3: 000000001d181000 CR4: 00000000001506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: __refcount_sub_and_test include/linux/refcount.h:283 [inline] __refcount_dec_and_test include/linux/refcount.h:315 [inline] refcount_dec_and_test include/linux/refcount.h:333 [inline] kref_put include/linux/kref.h:64 [inline] rxe_qp_do_cleanup+0x96f/0xaf0 drivers/infiniband/sw/rxe/rxe_qp.c:805 execute_in_process_context+0x37/0x150 kernel/workqueue.c:3327 rxe_elem_release+0x9f/0x180 drivers/infiniband/sw/rxe/rxe_pool.c:391 kref_put include/linux/kref.h:65 [inline] rxe_create_qp+0x2cd/0x310 drivers/infiniband/sw/rxe/rxe_verbs.c:425 _ib_create_qp drivers/infiniband/core/core_priv.h:331 [inline] ib_create_named_qp+0x2ad/0x1370 drivers/infiniband/core/verbs.c:1231 ib_create_qp include/rdma/ib_verbs.h:3644 [inline] create_mad_qp+0x177/0x2d0 drivers/infiniband/core/mad.c:2920 ib_mad_port_open drivers/infiniband/core/mad.c:3001 [inline] ib_mad_init_device+0xd6f/0x1400 drivers/infiniband/core/mad.c:3092 add_client_context+0x405/0x5e0 drivers/infiniband/core/device.c:717 enable_device_and_get+0x1cd/0x3b0 drivers/infiniband/core/device.c:1331 ib_register_device drivers/infiniband/core/device.c:1413 [inline] ib_register_device+0x7c7/0xa50 drivers/infiniband/core/device.c:1365 rxe_register_device+0x3d5/0x4a0 drivers/infiniband/sw/rxe/rxe_verbs.c:1147 rxe_add+0x12fe/0x16d0 drivers/infiniband/sw/rxe/rxe.c:247 rxe_net_add+0x8c/0xe0 drivers/infiniband/sw/rxe/rxe_net.c:503 rxe_newlink drivers/infiniband/sw/rxe/rxe.c:269 [inline] rxe_newlink+0xb7/0xe0 drivers/infiniband/sw/rxe/rxe.c:250 nldev_newlink+0x30e/0x550 drivers/infiniband/core/nldev.c:1555 rdma_nl_rcv_msg+0x36d/0x690 drivers/infiniband/core/netlink.c:195 rdma_nl_rcv_skb drivers/infiniband/core/netlink.c:239 [inline] rdma_nl_rcv+0x2ee/0x430 drivers/infiniband/core/netlink.c:259 netlink_unicast_kernel net/netlink/af_netlink.c:1312 [inline] netlink_unicast+0x533/0x7d0 net/netlink/af_netlink.c:1338 netlink_sendmsg+0x856/0xd90 net/netlink/af_netlink.c:1927 sock_sendmsg_nosec net/socket.c:654 [inline] sock_sendmsg+0xcf/0x120 net/socket.c:674 __sys_sendmsg+0x6e8/0x810 net/socket.c:2350 __sys_sendmsg+0xf3/0x170 net/socket.c:2404 __sys_sendmsg+0xe5/0x1b0 net/socket.c:2433 do_syscall_64+0x3a/0xb0 arch/x86/entry/common.c:47 entry_SYSCALL_64_after_hwframe+0 ---truncated---
CVE-2021-47079	In the Linux kernel, the following vulnerability has been resolved: platform/x86: ideapad-laptop: fix a NULL pointer dereference The third parameter of dytc_cql_command should not be NULL since it will be dereferenced immediately.

CVE ID	Description
CVE-2021-47092	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Always clear vmx->fail on emulation_required Revert a relatively recent change that set vmx->fail if the vCPU is in L2 and emulation_required is true, as that behavior is completely bogus. Setting vmx->fail and synthesizing a VM-Exit is contradictory and wrong: (a) it's impossible to have both a VM-Fail and VM-Exit (b) vmcs.EXIT_REASON is not modified on VM-Fail (c) emulation_required refers to guest state and guest state checks are always VM-Exits, not VM-Fails. For KVM specifically, emulation_required is handled before nested exits in __vmx_handle_exit(), thus setting vmx->fail has no immediate effect, i.e. KVM calls into handle_invalid_guest_state() and vmx->fail is ignored. Setting vmx->fail can ultimately result in a WARN in nested_vmx_vmexit() firing when tearing down the VM as KVM never expects vmx->fail to be set when L2 is active, KVM always reflects those errors into L1. <pre> -----[cut here]----- WARNING: CPU: 0 PID: 21158 at arch/x86/kvm/vmx/nested.c:4548 nested_vmx_vmexit+0x16bd/0x17e0 arch/x86/kvm/vmx/nested.c:4547 Modules linked in: CPU: 0 PID: 21158 Comm: syz-executor.1 Not tainted 5.16.0-rc3-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:nested_vmx_vmexit +0x16bd/0x17e0 arch/x86/kvm/vmx/nested.c:4547 Code: <0f> 0b e9 2e f8 ff ff e8 57 b3 5d 00 0f 0b e9 00 f1 ff ff 89 e9 80 Call Trace: vmx_leave_nested arch/x86/kvm/vmx/nested.c:6220 [inline] nested_vmx_free_vcpu+0x83/0xc0 arch/x86/kvm/vmx/nested.c:330 vmx_free_vcpu +0x11f/0x2a0 arch/x86/kvm/vmx/vmx.c:6799 kvm_arch_vcpu_destroy+0x6b/0x240 arch/x86/ kvm/x86.c:10989 kvm_vcpu_destroy+0x29/0x90 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:441 kvm_free_vcpus arch/x86/kvm/x86.c:11426 [inline] kvm_arch_destroy_vm+0x3ef/0x6b0 arch/ x86/kvm/x86.c:11545 kvm_destroy_vm arch/x86/kvm/../../../../virt/kvm/kvm_main.c:1189 [inline] kvm_put_kvm+0x751/0xe40 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:1220 kvm_vcpu_release +0x53/0x60 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:3489 __fput+0x3fc/0x870 fs/file_table.c:280 task_work_run+0x146/0x1c0 kernel/task_work.c:164 exit_task_work include/linux/task_work.h:32 [inline] do_exit+0x705/0x24f0 kernel/exit.c:832 do_group_exit+0x168/0x2d0 kernel/exit.c:929 get_signal+0x1740/0x2120 kernel/signal.c:2852 arch_do_signal_or_restart+0x9c/0x730 arch/x86/kernel/ signal.c:868 handle_signal_work kernel/entry/common.c:148 [inline] exit_to_user_mode_loop kernel/ entry/common.c:172 [inline] exit_to_user_mode_prepare+0x191/0x220 kernel/entry/common.c:207 __syscall_exit_to_user_mode_work kernel/entry/common.c:289 [inline] syscall_exit_to_user_mode +0x2e/0x70 kernel/entry/common.c:300 do_syscall_64+0x53/0xd0 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x44/0xae </pre>

CVE ID	Description
CVE-2021-47094	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Don't advance iterator after restart due to yielding After dropping mmu_lock in the TDP MMU, restart the iterator during tdp_iter_next() and do not advance the iterator. Advancing the iterator results in skipping the top-level SPTE and all its children, which is fatal if any of the skipped SPTEs were not visited before yielding. When zapping all SPTEs, i.e. when min_level == root_level, restarting the iter and then invoking tdp_iter_next() is always fatal if the current gfn has as a valid SPTE, as advancing the iterator results in try_step_side() skipping the current gfn, which wasn't visited before yielding. Sprinkle WARNs on iter->yielded being true in various helpers that are often used in conjunction with yielding, and tag the helper with __must_check to reduce the probability of improper usage. Failing to zap a top-level SPTE manifests in one of two ways. If a valid SPTE is skipped by both kvm_tdp_mmu_zap_all() and kvm_tdp_mmu_put_root(), the shadow page will be leaked and KVM will WARN accordingly. WARNING: CPU: 1 PID: 3509 at arch/x86/kvm/mmu/tdp_mmu.c:46 [kvm] RIP: 0010:kvm_mmu_uninit_tdp_mmu+0x3e/0x50 [kvm] Call Trace: <TASK> kvm_arch_destroy_vm+0x130/0x1b0 [kvm] kvm_destroy_vm+0x162/0x2a0 [kvm] kvm_vcpu_release+0x34/0x60 [kvm] __fput+0x82/0x240 task_work_run+0x5c/0x90 do_exit+0x364/0xa10 ? futex_unqueue+0x38/0x60 do_group_exit+0x33/0xa0 get_signal+0x155/0x850 arch_do_signal_or_restart+0xed/0x750 exit_to_user_mode_prepare+0xc5/0x120 syscall_exit_to_user_mode+0x1d/0x40 do_syscall_64+0x48/0xc0 entry_SYSCALL_64_after_hwframe+0x44/0xae If kvm_tdp_mmu_zap_all() skips a gfn/SPTE but that SPTE is then zapped by kvm_tdp_mmu_put_root(), KVM triggers a use-after-free in the form of marking a struct page as dirty/ accessed after it has been put back on the free list. This directly triggers a WARN due to encountering a page with page_count() == 0, but it can also lead to data corruption and additional errors in the kernel. WARNING: CPU: 7 PID: 1995658 at arch/x86/kvm/../../../virt/kvm/kvm_main.c:171 RIP: 0010:kvm_is_zone_device_pfn.part.0+0x9e/0xd0 [kvm] Call Trace: <TASK> kvm_set_pfn_dirty+0x120/0x1d0 [kvm] __handle_changed_spte+0x92e/0xca0 [kvm] __handle_changed_spte+0x63c/0xca0 [kvm] __handle_changed_spte+0x63c/0xca0 [kvm] __handle_changed_spte+0x63c/0xca0 [kvm] zap_gfn_range+0x549/0x620 [kvm] kvm_tdp_mmu_put_root+0x1b6/0x270 [kvm] mmu_free_root_page+0x219/0x2c0 [kvm] kvm_mmu_free_roots+0x1b4/0x4e0 [kvm] kvm_mmu_unload+0x1c/0xa0 [kvm] kvm_arch_destroy_vm+0x1f2/0x5c0 [kvm] kvm_put_kvm+0x3b1/0x8b0 [kvm] kvm_vcpu_release+0x4e/0x70 [kvm] __fput+0x1f7/0x8c0 task_work_run+0xf8/0x1a0 do_exit+0x97b/0x2230 do_group_exit+0xda/0x2a0 get_signal+0x3be/0x1e50 arch_do_signal_or_restart+0x244/0x17f0 exit_to_user_mode_prepare+0xcb/0x120 syscall_exit_to_user_mode+0x1d/0x40 do_syscall_64+0x4d/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae Note, the underlying bug existed even before commit 1af4a96025b3 ("KVM: x86/mmu: Yield in TDU MMU iter even if no SPTEs changed") moved calls to tdp_mmu_iter_cond_resched() to the beginning of loops, as KVM could still incorrectly advance past a top-level entry when yielding on a lower-level entry. But with respect to leaking shadow pages, the bug was introduced by yielding before processing the current gfn. Alternatively, tdp_mmu_iter_cond_resched() could simply fall through, or callers could jump to their "retry" label. The downside of that approach is that tdp_mmu_iter_cond_resched() __must__ be called before anything else in the loop, and there's no easy way to enforce that requirement. Ideally, KVM would handling the cond_resched() fully within the iterator macro (the code is actually quite clean) and avoid this entire class of bugs, but that is extremely difficult do wh ---truncated---

CVE ID	Description
CVE-2021-47103	In the Linux kernel, the following vulnerability has been resolved: inet: fully convert sk->sk_rx_dst to RCU rules syzbot reported various issues around early demux, one being included in this changelog [1] sk->sk_rx_dst is using RCU protection without clearly documenting it. And following sequences in tcp_v4_do_rcv()/tcp_v6_do_rcv() are not following standard RCU rules. [a] dst_release(dst); [b] sk->sk_rx_dst = NULL; They look wrong because a delete operation of RCU protected pointer is supposed to clear the pointer before the call_rcu()/synchronize_rcu() guarding actual memory freeing. In some cases indeed, dst could be freed before [b] is done. We could cheat by clearing sk_rx_dst before calling dst_release(), but this seems the right time to stick to standard RCU annotations and debugging facilities. [1] BUG: KASAN: use-after-free in dst_check include/net/dst.h:470 [inline] BUG: KASAN: use-after-free in tcp_v4_early_demux +0x95b/0x960 net/ipv4/tcp_ipv4.c:1792 Read of size 2 at addr ffff88807f1cb73a by task syz-executor.5/9204 CPU: 0 PID: 9204 Comm: syz-executor.5 Not tainted 5.16.0-rc5-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl +0xcd/0x134 lib/dump_stack.c:106 print_address_description.constprop.0.cold+0x8d/0x320 mm/kasan/report.c:247 __kasan_report mm/kasan/report.c:433 [inline] kasan_report.cold+0x83/0xdf mm/kasan/report.c:450 dst_check include/net/dst.h:470 [inline] tcp_v4_early_demux+0x95b/0x960 net/ipv4/tcp_ipv4.c:1792 ip_rcv_finish_core.constprop.0+0x15de/0x1e80 net/ipv4/ip_input.c:340 ip_list_rcv_finish.constprop.0+0x1b2/0x6e0 net/ipv4/ip_input.c:583 ip_sublist_rcv net/ipv4/ip_input.c:609 [inline] ip_list_rcv+0x34e/0x490 net/ipv4/ip_input.c:644 __netif_receive_skb_list_ptype net/core/dev.c:5508 [inline] __netif_receive_skb_list_core+0x549/0x8e0 net/core/dev.c:5556 __netif_receive_skb_list net/core/dev.c:5608 [inline] netif_receive_skb_list_internal+0x75e/0xd80 net/core/dev.c:5699 gro_normal_list net/core/dev.c:5853 [inline] gro_normal_list net/core/dev.c:5849 [inline] napi_complete_done+0x1f1/0x880 net/core/dev.c:6590 virtqueue_napi_complete drivers/net/virtio_net.c:339 [inline] virtnet_poll+0xca2/0x11b0 drivers/net/virtio_net.c:1557 __napi_poll +0xaf/0x440 net/core/dev.c:7023 napi_poll net/core/dev.c:7090 [inline] net_rx_action+0x801/0xb40 net/core/dev.c:7177 __do_softirq+0x29b/0x9c2 kernel/softirq.c:558 invoke_softirq kernel/softirq.c:432 [inline] __irq_exit_rcu+0x123/0x180 kernel/softirq.c:637 irq_exit_rcu+0x5/0x20 kernel/softirq.c:649 common_interrupt+0x52/0xc0 arch/x86/kernel/irq.c:240 asm_common_interrupt+0x1e/0x40 arch/x86/include/asm/idtentry.h:629 RIP: 0033:0x7f5e972bfd57 Code: 39 d1 73 14 0f 1f 80 00 00 00 00 48 8b 50 f8 48 83 e8 08 48 39 ca 77 f3 48 39 c3 73 3e 48 89 13 48 8b 50 f8 48 89 38 49 8b 0e <48> 8b 3e 48 83 c3 08 48 83 c6 08 eb bc 48 39 d1 72 9e 48 39 d0 73 RSP: 002b:00007fff8a413210 EFLAGS: 00000283 RAX: 00007f5e97108990 RBX: 00007f5e97108338 RCX: ffffffff81d3aa45 RDX: ffffffff81d3aa45 RSI: 00007f5e97108340 RDI: ffffffff81d3aa45 RBP: 00007f5e97107eb8 R08: 00007f5e97108d88 R09: 0000000093c2e8d9 R10: 0000000000000000 R11: 0000000000000000 R12: 00007f5e97107eb0 R13: 00007f5e97108338 R14: 00007f5e97107ea8 R15: 0000000000000019 </TASK> Allocated by task 13: kasan_save_stack+0x1e/0x50 mm/kasan/common.c:38 kasan_set_track mm/kasan/common.c:46 [inline] set_alloc_info mm/kasan/common.c:434 [inline] __kasan_slab_alloc+0x90/0xc0 mm/kasan/common.c:467 kasan_slab_alloc include/linux/kasan.h:259 [inline] slab_post_alloc_hook mm/slab.h:519 [inline] slab_alloc_node mm/slub.c:3234 [inline] slab_alloc mm/slub.c:3242 [inline] kmem_cache_alloc +0x202/0x3a0 mm/slub.c:3247 dst_alloc+0x146/0x1f0 net/core/dst.c:92 rt_dst_alloc+0x73/0x430 net/ipv4/route.c:1613 ip_route_input_slow+0x1817/0x3a20 net/ipv4/route.c:234 ---truncated---
CVE-2021-47116	In the Linux kernel, the following vulnerability has been resolved: ext4: fix memory leak in ext4_mb_init_backend on error path. Fix a memory leak discovered by syzbot when a file system is corrupted with an illegally large s_log_groups_per_flex.
CVE-2021-47120	In the Linux kernel, the following vulnerability has been resolved: HID: magicmouse: fix NULL-deref on disconnect Commit 9d7b18668956 ("HID: magicmouse: add support for Apple Magic Trackpad 2") added a sanity check for an Apple trackpad but returned success instead of -ENODEV when the check failed. This means that the remove callback will dereference the never-initialised driver data pointer when the driver is later unbound (e.g. on USB disconnect).

CVE ID	Description
CVE-2021-47133	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: Fix memory leak in amd_sfh_work Kmemleak tool detected a memory leak in the amd_sfh driver. ===== unreferenced object 0xffff88810228ada0 (size 32): comm "insmod", pid 3968, jiffies 4295056001 (age 775.792s) hex dump (first 32 bytes): 00 20 73 1f 81 88 ff ff 00 01 00 00 00 00 ad de . s..... 22 01 00 00 00 00 ad de 01 00 02 00 00 00 00 00 "..... backtrace: [<000000007b4c8799>] kmem_cache_alloc_trace+0x163/0x4f0 [<0000000005326893>] amd_sfh_get_report+0xa4/0x1d0 [amd_sfh] [<000000002a9e5ec4>] amdtp_hid_request+0x62/0x80 [amd_sfh] [<00000000b8a95807>] sensor_hub_get_feature+0x145/0x270 [hid_sensor_hub] [<00000000fda054ee>] hid_sensor_parse_common_attributes+0x215/0x460 [hid_sensor_iio_common] [<0000000021279ecf>] hid_accel_3d_probe+0xff/0x4a0 [hid_sensor_accel_3d] [<00000000915760ce>] platform_probe+0x6a/0xd0 [<0000000060258a1f>] really_probe+0x192/0x620 [<00000000fa812f2d>] driver_probe_device+0x14a/0x1d0 [<000000005e79f7fd>] __device_attach_driver+0xbd/0x110 [<0000000070d15018>] bus_for_each_drv+0xfd/0x160 [<0000000013a3c312>] __device_attach+0x18b/0x220 [<000000008c7b4afc>] device_initial_probe+0x13/0x20 [<00000000e6e99665>] bus_probe_device+0xfe/0x120 [<00000000833fa90b>] device_add+0x6a6/0xe00 [<00000000fa901078>] platform_device_add+0x180/0x380 ===== The fix is to freeing request_list entry once the processed entry is removed from the request_list.
CVE-2021-47134	In the Linux kernel, the following vulnerability has been resolved: efi/fdt: fix panic when no valid fdt found setup_arch() would invoke efi_init()->efi_get_fdt_params(). If no valid fdt found then initial_boot_params will be null. So we should stop further fdt processing here. I encountered this issue on risc-v.
CVE-2021-47136	In the Linux kernel, the following vulnerability has been resolved: net: zero-initialize tc skb extension on allocation Function skb_ext_add() doesn't initialize created skb extension with any value and leaves it up to the user. However, since extension of type TC_SKB_EXT originally contained only single value tc_skb_ext->chain its users used to just assign the chain value without setting whole extension memory to zero first. This assumption changed when TC_SKB_EXT extension was extended with additional fields but not all users were updated to initialize the new fields which leads to use of uninitialized memory afterwards. UBSAN log: [778.299821] UBSAN: invalid-load in net/openvswitch/flow.c:899:28 [778.301495] load of value 107 is not a valid value for type '_Bool' [778.303215] CPU: 0 PID: 0 Comm: swapper/0 Not tainted 5.12.0-rc7+ #2 [778.304933] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 [778.307901] Call Trace: [778.308680] <IRQ> [778.309358] dump_stack+0xbb/0x107 [778.310307] ubsan_epilogue+0x5/0x40 [778.311167] __ubsan_handle_load_invalid_value.cold+0x43/0x48 [778.312454] ? memset+0x20/0x40 [778.313230] ovs_flow_key_extract.cold+0xf/0x14 [openvswitch] [778.314532] ovs_vport_receive+0x19e/0x2e0 [openvswitch] [778.315749] ? ovs_vport_find_upcall_portid+0x330/0x330 [openvswitch] [778.317188] ? create_prof_cpu_mask+0x20/0x20 [778.318220] ? arch_stack_walk+0x82/0xf0 [778.319153] ? secondary_startup_64_no_verify+0xb0/0xbb [778.320399] ? stack_trace_save+0x91/0xc0 [778.321362] ? stack_trace_consume_entry+0x160/0x160 [778.322517] ? lock_release+0x52e/0x760 [778.323444] netdev_frame_hook+0x323/0x610 [openvswitch] [778.324668] ? ovs_netdev_get_vport+0xe0/0xe0 [openvswitch] [778.325950] __netif_receive_skb_core+0x771/0x2db0 [778.327067] ? lock_downgrade+0x6e0/0x6f0 [778.328021] ? lock_acquire+0x565/0x720 [778.328940] ? generic_xdp_tx+0x4f0/0x4f0 [778.329902] ? inet_gro_receive+0x2a7/0x10a0 [778.330914] ? lock_downgrade+0x6f0/0x6f0 [778.331867] ? udp4_gro_receive+0x4c4/0x13e0 [778.332876] ? lock_release+0x52e/0x760 [778.333808] ? dev_gro_receive+0xcc8/0x2380 [778.334810] ? lock_downgrade+0x6f0/0x6f0 [778.335769] __netif_receive_skb_list_core+0x295/0x820 [778.336955] ? process_backlog+0x780/0x780 [778.337941] ? mlx5e_rep_tc_netdevice_event_unregister+0x20/0x20 [mlx5_core] [778.339613] ? seqcount_lockdep_reader_access.constprop.0+0xa7/0xc0 [778.341033] ? kvm_clock_get_cycles+0x14/0x20 [778.342072] netif_receive_skb_list_internal+0x5f5/0xcb0 [778.343288] ? __kasan_kmalloc+0x7a/0x90 [778.344234] ? mlx5e_handle_rx_cqe_mpwqrq+0x9e0/0x9e0 [mlx5_core] [778.345676] ? mlx5e_xmit_xdp_frame_mpwqe+0x14d0/0x14d0 [mlx5_core] [778.347140] ? __netif_receive_skb_list_core+0x820/0x820 [778.348351] ? mlx5e_post_rx_mpwqes+0xa6/0x25d0 [mlx5_core] [778.349688] ? napi_gro_flush+0x26c/0x3c0 [778.350641] napi_complete_done+0x188/0x6b0 [778.351627] mlx5e_napi_poll+0x373/0x1b80 [mlx5_core] [778.352853] __napi_poll+0x9f/0x510 [778.353704] ? mlx5_flow_namespace_set_mode+0x260/0x260 [mlx5_core] [778.355158] net_rx_action+0x34c/0xa40 [778.356060] ? napi_threaded_poll+0x3d0/0x3d0 [778.357083] ? sched_clock_cpu+0x18/0x190 [778.358041] ? __common_interrupt+0x8e/0x1a0 [778.359045] __do_softirq+0x1ce/0x984 [778.359938] __irq_exit_rcu+0x137/0x1d0 [778.360865] irq_exit_rcu+0xa/0x20 [778.361708] common_interrupt+0x80/0xa0 [778.362640] <IRQ> [778.363212] asm_common_interrupt+0x1e/0x40 [778.364204] RIP: 0010:native_safe_halt+0xe/0x10 [778.365273] Code: 4f ff ff ff 4c 89 e7 e8 50 3f 40 fe e9 dc fe ff ff 48 89 df e8 43 3f 40 fe eb 90 cc e9 07 00 00 0f 0d 2d 74 05 62 00 fb f4 <c3> 90 e9 07 00 00 0f 0d 2d 64 05 62 00 f4 c3 cc cc 0f 1f 44 00 [778.369355] RSP: 0018:ffffffffff84407e48 EFLAGS: 00000246 [778.370570] RAX ---truncated---

CVE ID	Description
CVE-2021-47138	In the Linux kernel, the following vulnerability has been resolved: cxgb4: avoid accessing registers when clearing filters Hardware register having the server TID base can contain invalid values when adapter is in bad state (for example, due to AER fatal error). Reading these invalid values in the register can lead to out-of-bound memory access. So, fix by using the saved server TID base when clearing filters.
CVE-2021-47140	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Clear DMA ops when switching domain Since commit 08a27c1c3ecf ("iommu: Add support to change default domain of an iommu group") a user can switch a device between IOMMU and direct DMA through sysfs. This doesn't work for AMD IOMMU at the moment because dev->dma_ops is not cleared when switching from a DMA to an identity IOMMU domain. The DMA layer thus attempts to use the dma-iommu ops on an identity domain, causing an oops: # echo 0000:00:05.0 > /sys/sys/bus/pci/drivers/e1000e/unbind # echo identity > /sys/bus/pci/devices/0000:00:05.0/iommu_group/type # echo 0000:00:05.0 > /sys/sys/bus/pci/drivers/e1000e/bind ... BUG: kernel NULL pointer dereference, address: 0000000000000028 ... Call Trace: iommu_dma_alloc e1000e_setup_tx_resources e1000e_open Since iommu_change_dev_def_domain() calls probe_finalize() again, clear the dma_ops there like Vt-d does.
CVE-2021-47141	In the Linux kernel, the following vulnerability has been resolved: gve: Add NULL pointer checks when freeing irqs. When freeing notification blocks, we index priv->msix_vectors. If we failed to allocate priv->msix_vectors (see abort_with_msix_vectors) this could lead to a NULL pointer dereference if the driver is unloaded.
CVE-2021-47143	In the Linux kernel, the following vulnerability has been resolved: net/smc: remove device from smcd_dev_list after failed device_add() If the device_add() for a smcd_dev fails, there's no cleanup step that rolls back the earlier list_add(). The device subsequently gets freed, and we end up with a corrupted list. Add some error handling that removes the device from the list.
CVE-2021-47146	In the Linux kernel, the following vulnerability has been resolved: mld: fix panic in mld_newpack() mld_newpack() doesn't allow to allocate high order page, only order-0 allocation is allowed. If headroom size is too large, a kernel panic could occur in skb_put(). Test commands: ip netns del A ip netns del B ip netns add A ip netns add B ip link add veth0 type veth peer name veth1 ip link set veth0 netns A ip link set veth1 netns B ip netns exec A ip link set lo up ip netns exec A ip link set veth0 up ip netns exec A ip -6 a a 2001:db8:0::1/64 dev veth0 ip netns exec B ip link set lo up ip netns exec B ip link set veth1 up ip netns exec B ip -6 a a 2001:db8:0::2/64 dev veth1 for i in {1..99} do let A=\$i-1 ip netns exec A ip link add ip6gre\$i type ip6gre \ local 2001:db8:\$A::1 remote 2001:db8:\$A::2 encaps limit 100 ip netns exec A ip -6 a a 2001:db8:\$i::1/64 dev ip6gre\$i ip netns exec A ip link set ip6gre\$i up ip netns exec B ip link add ip6gre\$i type ip6gre \ local 2001:db8:\$A::2 remote 2001:db8:\$A::1 encaps limit 100 ip netns exec B ip -6 a a 2001:db8:\$i::2/64 dev ip6gre\$i ip netns exec B ip link set ip6gre\$i up done Splat looks like: kernel BUG at net/core/skbuff.c:110! invalid opcode: 0000 [0] SMP DEBUG_PAGEALLOC KASAN PTI CPU: 0 PID: 7 Comm: kworker/0:1 Not tainted 5.12.0+ #891 Workqueue: ipv6_addrconf addrconf_dad_work RIP: 0010:skb_panic+0x15d/0x15f Code: 92 fe 4c 8b 4c 24 10 53 8b 4d 70 45 89 e0 48 c7 c7 00 ae 79 83 41 57 41 56 41 55 48 8b 54 24 a6 26 f9 ff <0f> 0b 48 8b 6c 24 20 89 34 24 e8 4a 4e 92 fe 8b 34 24 48 c7 c1 20 RSP: 0018:ffff88810091f820 EFLAGS: 00010282 RAX: 0000000000000089 RBX: ffff8881086e9000 RCX: 0000000000000000 RDX: 0000000000000089 RSI: 0000000000000008 RDI: ffffd1020123efb RBP: ffff888005f6eac0 R08: ffffd1022fc0031 R09: ffffd1022fc0031 R10: ffff888117e00187 R11: ffffd1022fc0030 R12: 0000000000000028 R13: ffff888008284eb0 R14: 00000000000000ed8 R15: 00000000000000ec0 FS: 0000000000000000(0000) GS:ffff888117c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f8b801c5640 CR3: 0000000033c2c006 CR4: 00000000003706f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: ? ip6_mc_hdr.isra.26.constprop.46+0x12a/0x600 ? ip6_mc_hdr.isra.26.constprop.46+0x12a/0x600 skb_put.cold.104+0x22/0x22 ip6_mc_hdr.isra.26.constprop.46+0x12a/0x600 ? rcu_read_lock_sched_held+0x91/0xc0 mld_newpack+0x398/0x8f0 ? ip6_mc_hdr.isra.26.constprop.46+0x600/0x600 ? lock_contended+0xc40/0xc40 add_grhead.isra.33+0x280/0x380 add_grec+0x5ca/0xff0 ? mld_sendpack+0xf40/0xf40 ? lock_downgrade+0x690/0x690 mld_send_initial_cr.part.34+0xb9/0x180 ipv6_mc_dad_complete+0x15d/0x1b0 addrconf_dad_completed+0x8d2/0xbb0 ? lock_downgrade+0x690/0x690 ? addrconf_rs_timer+0x660/0x660 ? addrconf_dad_work+0x73c/0x10e0 addrconf_dad_work+0x73c/0x10e0 Allowing high order page allocation could fix this problem.

CVE ID	Description
CVE-2021-47162	In the Linux kernel, the following vulnerability has been resolved: tipc: skb_linearize the head skb when reassembling msg's. It's not a good idea to append the frag skb to a skb's frag_list if the frag_list already has skbs from elsewhere, such as this skb was created by pskb_copy() where the frag_list was cloned (all the skbs in it were skb_get'ed) and shared by multiple skbs. However, the new appended frag skb should have been only seen by the current skb. Otherwise, it will cause use after free crashes as this appended frag skb are seen by multiple skbs but it only got skb_get called once. The same thing happens with a skb updated by pskb_may_pull() with a skb_cloned skb. Li Shuang has reported quite a few crashes caused by this when doing testing over macvlan devices: [] kernel BUG at net/core/skbuff.c:1970! [] Call Trace: [] skb_clone+0x4d/0xb0 [] macvlan_broadcast+0xd8/0x160 [macvlan] [] macvlan_process_broadcast+0x148/0x150 [macvlan] [] process_one_work+0x1a7/0x360 [] worker_thread+0x30/0x390 [] kernel BUG at mm/usercopy.c:102! [] Call Trace: [] __check_heap_object+0xd3/0x100 [] __check_object_size+0xff/0x16b [] simple_copy_to_iter+0x1c/0x30 [] __skb_datagram_iter+0x7d/0x310 [] __skb_datagram_iter+0x2a5/0x310 [] skb_copy_datagram_iter+0x3b/0x90 [] tipc_rcvmsg+0x14a/0x3a0 [tipc] [] ____sys_rcvmsg+0x91/0x150 [] ____sys_rcvmsg+0x7b/0xc0 [] kernel BUG at mm/slub.c:305! [] Call Trace: [] <IRQ> [] kmem_cache_free+0x3ff/0x400 [] __netif_receive_skb_core+0x12c/0xc40 [] ? kmem_cache_alloc+0x12e/0x270 [] netif_receive_skb_internal+0x3d/0xb0 [] ? get_rx_page_info+0x8e/0xa0 [be2net] [] be_poll+0x6ef/0xd00 [be2net] [] ? irq_exit+0x4f/0x100 [] net_rx_action+0x149/0x3b0 ... This patch is to fix it by linearizing the head skb if it has frag_list set in tipc_buf_append(). Note that we choose to do this before calling skb_unshare(), as __skb_linearize() will avoid skb_copy(). Also, we can not just drop the frag_list either as the early time.
CVE-2021-47163	In the Linux kernel, the following vulnerability has been resolved: tipc: wait and exit until all work queues are done. On some host, a crash could be triggered simply by repeating these commands several times: # modprobe tipc # tipc bearer enable media udp name UDP1 localip 127.0.0.1 # rmmmod tipc [] BUG: unable to handle kernel paging request at ffffffff096bb00 [] Workqueue: events 0xffffffff096bb00 [] Call Trace: [] ? process_one_work+0x1a7/0x360 [] ? worker_thread+0x30/0x390 [] ? create_worker+0x1a0/0x1a0 [] ? kthread+0x116/0x130 [] ? kthread_flush_work_fn+0x10/0x10 [] ? ret_from_fork+0x35/0x40 When removing the TIPC module, the UDP tunnel sock will be delayed to release in a work queue as sock_release() can't be done in rtnl_lock(). If the work queue is scheduled to run after the TIPC module is removed, kernel will crash as the work queue function cleanup_bearer() code no longer exists when trying to invoke it. To fix it, this patch introduce a member wq_count in tipc_net to track the numbers of work queues in schedule, and wait and exit until all work queues are done in tipc_exit_net().
CVE-2021-47167	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix an Oopsable condition in __nfs_pageio_add_request(). Ensure that nfs_pageio_error_cleanup() resets the mirror array contents, so that the structure reflects the fact that it is now empty. Also change the test in nfs_pageio_do_add_request() to be more robust by checking whether or not the list is empty rather than relying on the value of pg_count.
CVE-2021-47168	In the Linux kernel, the following vulnerability has been resolved: NFS: fix an incorrect limit in filelayout_decode_layout(). The "sizeof(struct nfs_fh)" is two bytes too large and could lead to memory corruption. It should be NFS_MAXFHSIZE because that's the size of the ->data[] buffer. I reversed the size of the arguments to put the variable on the left.
CVE-2021-47170	In the Linux kernel, the following vulnerability has been resolved: USB: usbfs: Don't WARN about excessively large memory allocations. Syzbot found that the kernel generates a WARNING if the user tries to submit a bulk transfer through usbfs with a buffer that is way too large. This isn't a bug in the kernel; it's merely an invalid request from the user and the usbfs code does handle it correctly. In theory the same thing can happen with async transfers, or with the packet descriptor table for isochronous transfers. To prevent the MM subsystem from complaining about these bad allocation requests, add the __GFP_NOWARN flag to the kmalloc calls for these buffers.
CVE-2021-47173	In the Linux kernel, the following vulnerability has been resolved: misc/uss720: fix memory leak in uss720_probe. uss720_probe forgets to decrease the refcount of usbdev in uss720_probe. Fix this by decreasing the refcount of usbdev by usb_put_dev. BUG: memory leak unreferenced object 0xffff888101113800 (size 2048): comm "kworker/0:1", pid 7, jiffies 4294956777 (age 28.870s) hex dump (first 32 bytes): ff ff ff 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<ffffffff82b8e822>] kmalloc include/linux/slab.h:554 [inline] [<ffffffff82b8e822>] kcalloc include/linux/slab.h:684 [inline] [<ffffffff82b8e822>] usb_alloc_dev+0x32/0x450 drivers/usb/core/usb.c:582 [<ffffffff82b98441>] hub_port_connect drivers/usb/core/hub.c:5129 [inline] [<ffffffff82b98441>] hub_port_connect_change drivers/usb/core/hub.c:5363 [inline] [<ffffffff82b98441>] port_event drivers/usb/core/hub.c:5509 [inline] [<ffffffff82b98441>] hub_event+0x1171/0x20c0 drivers/usb/core/hub.c:5591 [<ffffffff81259229>] process_one_work+0x2c9/0x600 kernel/workqueue.c:2275 [<ffffffff81259b19>] worker_thread+0x59/0x5d0 kernel/workqueue.c:2421 [<ffffffff81261228>] kthread+0x178/0x1b0 kernel/kthread.c:292 [<ffffffff8100227f>] ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:294

CVE ID	Description
CVE-2021-47176	In the Linux kernel, the following vulnerability has been resolved: s390/dasd: add missing discipline function Fix crash with illegal operation exception in dasd_device_tasklet. Commit b72949328869 ("s390/dasd: Prepare for additional path event handling") renamed the verify_path function for ECKD but not for FBA and DIAG. This leads to a panic when the path verification function is called for a FBA or DIAG device. Fix by defining a wrapper function for dasd_generic_verify_path().
CVE-2021-47177	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix sysfs leak in alloc_iommu() iommu_device_sysfs_add() is called before, so is has to be cleaned on subsequent errors.
CVE-2021-47178	In the Linux kernel, the following vulnerability has been resolved: scsi: target: core: Avoid smp_processor_id() in preemptible code The BUG message "BUG: using smp_processor_id() in preemptible [00000000] code" was observed for TCMU devices with kernel config DEBUG_PREEMPT. The message was observed when blktests block/005 was run on TCMU devices with fileio backend or user:zbc backend [1]. The commit 1130b499b4a7 ("scsi: target: tcm_loop: Use LIO wq cmd submission helper") triggered the symptom. The commit modified work queue to handle commands and changed 'current->nr_cpu_allowed' at smp_processor_id() call. The message was also observed at system shutdown when TCMU devices were not cleaned up [2]. The function smp_processor_id() was called in SCSI host work queue for abort handling, and triggered the BUG message. This symptom was observed regardless of the commit 1130b499b4a7 ("scsi: target: tcm_loop: Use LIO wq cmd submission helper"). To avoid the preemptible code check at smp_processor_id(), get CPU ID with raw_smp_processor_id() instead. The CPU ID is used for performance improvement then thread move to other CPU will not affect the code. [1] [56.468103] run blktests block/005 at 2021-05-12 14:16:38 [57.369473] check_preemption_disabled: 85 callbacks suppressed [57.369480] BUG: using smp_processor_id() in preemptible [00000000] code: fio/1511 [57.369506] BUG: using smp_processor_id() in preemptible [00000000] code: fio/1510 [57.369512] BUG: using smp_processor_id() in preemptible [00000000] code: fio/1506 [57.369552] caller is __target_init_cmd+0x157/0x170 [target_core_mod] [57.369606] CPU: 4 PID: 1506 Comm: fio Not tainted 5.13.0-rc1+ #34 [57.369613] Hardware name: System manufacturer System Product Name/PRIME Z270-A, BIOS 1302 03/15/2018 [57.369617] Call Trace: [57.369621] BUG: using smp_processor_id() in preemptible [00000000] code: fio/1507 [57.369628] dump_stack+0x6d/0x89 [57.369642] check_preemption_disabled+0xc8/0xd0 [57.369628] caller is __target_init_cmd+0x157/0x170 [target_core_mod] [57.369655] __target_init_cmd+0x157/0x170 [target_core_mod] [57.369695] target_init_cmd+0x76/0x90 [target_core_mod] [57.369732] tcm_loop_queuecommand+0x109/0x210 [tcm_loop] [57.369744] scsi_queue_rq+0x38e/0xc40 [57.369761] __blk_mq_try_issue_directly+0x109/0x1c0 [57.369779] blk_mq_try_issue_directly+0x43/0x90 [57.369790] blk_mq_submit_bio+0x4e5/0x5d0 [57.369812] submit_bio_noacct+0x46e/0x4e0 [57.369830] __blkdev_direct_IO_simple+0x1a3/0x2d0 [57.369859] ? set_init_blocksize.isra.0+0x60/0x60 [57.369880] generic_file_read_iter+0x89/0x160 [57.369898] blkdev_read_iter+0x44/0x60 [57.369906] new_sync_read+0x102/0x170 [57.369929] vfs_read+0xd4/0x160 [57.369941] __x64_sys_pread64+0x6e/0xa0 [57.369946] ? lockdep_hardirqs_on+0x79/0x100 [57.369958] do_syscall_64+0x3a/0x70 [57.369965] entry_SYSCALL_64_after_hwframe+0x44/0xae [57.369973] RIP: 0033:0x7f7ed4c1399f [57.369979] Code: 08 89 3c 24 48 89 4c 24 18 e8 7d f3 ff ff 4c 8b 54 24 18 48 8b 54 24 10 41 89 c0 48 8b 74 24 08 8b 3c 24 b8 11 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 31 44 89 c7 48 89 04 24 e8 cd f3 ff ff 48 8b [57.369983] RSP: 002b:00007ffd7918c580 EFLAGS: 00000293 ORIG_RAX: 0000000000000011 [57.369990] RAX: ffffffff7ffda RBX: 00000000015b4540 RCX: 00007f7ed4c1399f [57.369993] RDX: 0000000000001000 RSI: 00000000015de000 RDI: 0000000000000009 [57.369996] RBP: 00000000015b4540 R08: 0000000000000000 R09: 0000000000000001 [57.369999] R10: 0000000000e5c000 R11: 0000000000000293 R12: 00007f7eb5269a70 [57.370002] R13: 0000000000000000 R14: 0000000000001000 R15: 00000000015b4568 [57.370031] CPU: 7 PID: 1507 Comm: fio Not tainted 5.13.0-rc1+ #34 [57.370036] Hardware name: System manufacturer System Product Name/PRIME Z270-A, BIOS 1302 03/15/2018 [57.370039] Call Trace: [57.370045] dump_stack+0x6d/0x89 [57.370056] ch ---truncated---

[illegible]

CVE ID	Description
CVE-2021-47196	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Set send and receive CQ before forwarding to the driver Preset both receive and send CQ pointers prior to call to the drivers and overwrite it later again till the mlx4 is going to be changed do not overwrite ibqp properties. This change is needed for mlx5, because in case of QP creation failure, it will go to the path of QP destroy which relies on proper CQ pointers. BUG: KASAN: use-after-free in create_qp.cold+0x164/0x16e [mlx5_ib] Write of size 8 at addr ffff8880064c55c0 by task a.out/246 CPU: 0 PID: 246 Comm: a.out Not tainted 5.15.0+ #291 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 Call Trace: dump_stack_lvl+0x45/0x59 print_address_description.constprop.0+0x1f/0x140 kasan_report.cold+0x83/0xdf create_qp.cold+0x164/0x16e [mlx5_ib] mlx5_ib_create_qp+0x358/0x28a0 [mlx5_ib] create_qp.part.0+0x45b/0x6a0 [ib_core] ib_create_qp_user+0x97/0x150 [ib_core] ib_uverbs_handler_UVERBS_METHOD_QP_CREATE+0x92c/0x1250 [ib_uverbs] ib_uverbs_cmd_verbs+0x1c38/0x3150 [ib_uverbs] ib_uverbs_ioctl+0x169/0x260 [ib_uverbs] __x64_sys_ioctl+0x866/0x14d0 do_syscall_64+0x3d/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae Allocated by task 246: kasan_save_stack+0x1b/0x40 __kasan_kmalloc+0xa4/0xd0 create_qp.part.0+0x92/0x6a0 [ib_core] ib_create_qp_user+0x97/0x150 [ib_core] ib_uverbs_handler_UVERBS_METHOD_QP_CREATE+0x92c/0x1250 [ib_uverbs] ib_uverbs_cmd_verbs+0x1c38/0x3150 [ib_uverbs] ib_uverbs_ioctl+0x169/0x260 [ib_uverbs] __x64_sys_ioctl+0x866/0x14d0 do_syscall_64+0x3d/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae Freed by task 246: kasan_save_stack+0x1b/0x40 kasan_set_track+0x1c/0x30 kasan_set_free_info+0x20/0x30 __kasan_slab_free+0x10c/0x150 slab_free_freelist_hook+0xb4/0x1b0 kfree+0xe7/0x2a0 create_qp.part.0+0x52b/0x6a0 [ib_core] ib_create_qp_user+0x97/0x150 [ib_core] ib_uverbs_handler_UVERBS_METHOD_QP_CREATE+0x92c/0x1250 [ib_uverbs] ib_uverbs_cmd_verbs+0x1c38/0x3150 [ib_uverbs] ib_uverbs_ioctl+0x169/0x260 [ib_uverbs] __x64_sys_ioctl+0x866/0x14d0 do_syscall_64+0x3d/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2021-47197	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: nullify cq->dbg pointer in mlx5_debug_cq_remove() Prior to this patch in case mlx5_core_destroy_cq() failed it proceeds to rest of destroy operations. mlx5_core_destroy_cq() could be called again by user and cause additional call of mlx5_debug_cq_remove(). cq->dbg was not nullify in previous call and cause the crash. Fix it by nullify cq->dbg pointer after removal. Also proceed to destroy operations only if FW return 0 for MLX5_CMD_OP_DESTROY_CQ command. general protection fault, probably for non-canonical address 0x2000300004058: 0000 [#1] SMP PTI CPU: 5 PID: 1228 Comm: python Not tainted 5.15.0-rc5_for_upstream_min_debug_2021_10_14_11_06 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:lockref_get+0x1/0x60 Code: 5d e9 53 ff ff ff 48 8d 7f 70 e8 0a 2e 48 00 c7 85 d0 00 00 00 02 00 00 00 c6 45 70 00 fb 5d c3 cc cc cc cc cc cc cc cc 53 <48> 8b 17 48 89 fb 85 d2 75 3d 48 89 d0 bf 64 00 00 00 48 89 c1 48 RSP: 0018:ffff888137dd7a38 EFLAGS: 00010206 RAX: 0000000000000000 RBX: ffff888107d5f458 RCX: 00000000ffffffffff RDX: 0000000000002c2b0 RSI: ffffffff8155e2e0 RDI: 0002000300004058 RBP: ffff888137dd7a88 R08: 0002000300004058 R09: ffff8881144a9f88 R10: 0000000000000000 R11: 0000000000000000 R12: ffff8881141d4000 R13: ffff888137dd7c68 R14: ffff888137dd7d58 R15: ffff888137dd7cc0 FS: 00007f4644f2a4c0(0000) GS: ffff8887a2d40000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005b4500f4380 CR3: 00000000114f7a003 CR4: 0000000000170ea0 Call Trace: simple_recursive_removal+0x33/0x2e0 ? debugfs_remove+0x60/0x60 debugfs_remove+0x40/0x60 mlx5_debug_cq_remove+0x32/0x70 [mlx5_core] mlx5_core_destroy_cq+0x41/0x1d0 [mlx5_core] devx_obj_cleanup+0x151/0x330 [mlx5_ib] ? __pollwait+0xd0/0xd0 ? xas_load+0x5/0x70 ? xa_load+0x62/0xa0 destroy_hw_idr_uobject+0x20/0x80 [ib_uverbs] uverbs_destroy_uobject+0x3b/0x360 [ib_uverbs] uobj_destroy+0x54/0xa0 [ib_uverbs] ib_uverbs_cmd_verbs+0xaf2/0x1160 [ib_uverbs] ? uverbs_finalize_object+0xd0/0xd0 [ib_uverbs] ib_uverbs_ioctl+0xc4/0x1b0 [ib_uverbs] __x64_sys_ioctl+0x3e4/0x8e0
CVE-2021-47198	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix use-after-free in lpfc_unreg_rpi() routine An error is detected with the following report when unloading the driver: "KASAN: use-after-free in lpfc_unreg_rpi+0x1b1b" The NLP_REG_LOGIN_SEND nlp_flag is set in lpfc_reg_fab_ctrl_node(), but the flag is not cleared upon completion of the login. This allows a second call to lpfc_unreg_rpi() to proceed with nlp_rpi set to LPFC_RPI_ALLOW_ERROR. This results in a use after free access when used as an rpi_ids array index. Fix by clearing the NLP_REG_LOGIN_SEND nlp_flag in lpfc_mbx_cmpl_fc_reg_login().

CVE ID	Description
CVE-2021-47199	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: CT, Fix multiple allocations and memleak of mod acts CT clear action offload adds additional mod_hdr actions to the flow's original mod actions in order to clear the registers which hold ct_state. When such flow also includes encap action, a neigh update event can cause the driver to unoffload the flow and then reoffload it. Each time this happens, the ct clear handling adds that same set of mod_hdr actions to reset ct_state until the max of mod_hdr actions is reached. Also the driver never releases the allocated mod_hdr actions and causing a memleak. Fix above two issues by moving CT clear mod acts allocation into the parsing actions phase and only use it when offloading the rule. The release of mod acts will be done in the normal flow_put(). backtrace: [<code><000000007316e2f3></code>] krealloc+0x83/0xd0 [<code><00000000ef157de1></code>] mlx5e_mod_hdr_alloc+0x147/0x300 [mlx5_core] [<code><00000000970ce4ae></code>] mlx5e_tc_match_to_reg_set_and_get_id+0xd7/0x240 [mlx5_core] [<code><0000000067c5fa17></code>] mlx5e_tc_match_to_reg_set+0xa/0x20 [mlx5_core] [<code><00000000d032eb98></code>] mlx5_tc_ct_entry_set_registers.isra.0+0x36/0xc0 [mlx5_core] [<code><00000000fd23b869></code>] mlx5_tc_ct_flow_offload+0x272/0x1f10 [mlx5_core] [<code><000000004fc24acc></code>] mlx5e_tc_offload_fdb_rules.part.0+0x150/0x620 [mlx5_core] [<code><00000000dc741c17></code>] mlx5e_tc_encap_flows_add+0x489/0x690 [mlx5_core] [<code><00000000e92e49d7></code>] mlx5e_rep_update_flows+0x6e4/0x9b0 [mlx5_core] [<code><00000000f60f5602></code>] mlx5e_rep_neigh_update+0x39a/0x5d0 [mlx5_core]
CVE-2021-47201	In the Linux kernel, the following vulnerability has been resolved: iavf: free q_vectors before queues in iavf_disable_vf iavf_free_queues() clears adapter->num_active_queues, which iavf_free_q_vectors() relies on, so swap the order of these two function calls in iavf_disable_vf(). This resolves a panic encountered when the interface is disabled and then later brought up again after PF communication is restored.
CVE-2021-47207	In the Linux kernel, the following vulnerability has been resolved: ALSA: gus: fix null pointer dereference on pointer block The pointer block return from snd_gf1_dma_next_block could be null, so there is a potential null pointer dereference issue. Fix this by adding a null check before dereference.
CVE-2021-47210	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tipd: Remove WARN_ON in tps6598x_block_read Calling tps6598x_block_read with a higher than allowed len can be handled by just returning an error. There's no need to crash systems with panic-on-warn enabled.
CVE-2021-47211	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: fix null pointer dereference on pointer cs_desc The pointer cs_desc return from snd_usb_find_clock_source could be null, so there is a potential null pointer dereference issue. Fix this by adding a null check before dereference.
CVE-2021-47215	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: kTLS, Fix crash in RX resync flow For the TLS RX resync flow, we maintain a list of TLS contexts that require some attention, to communicate their resync information to the HW. Here we fix list corruptions, by protecting the entries against movements coming from resync_handle_seq_match(), until their resync handling in napi is fully completed.
CVE-2021-47217	In the Linux kernel, the following vulnerability has been resolved: x86/hyperv: Fix NULL deref in set_hv_tscchange_cb() if Hyper-V setup fails Check for a valid hv_vp_index array prior to dereferencing hv_vp_index when setting Hyper-V's TSC change callback. If Hyper-V setup failed in hyperv_init(), the kernel will still report that it's running under Hyper-V, but will have silently disabled nearly all functionality. BUG: kernel NULL pointer dereference, address: 0000000000000010 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] SMP CPU: 4 PID: 1 Comm: swapper/0 Not tainted 5.15.0-rc2+ #75 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 0.0.0 02/06/2015 RIP: 0010:set_hv_tscchange_cb+0x15/0xa0 Code: <8b> 04 82 8b 15 12 17 85 01 48 c1 e0 20 48 0d ee 00 01 00 f6 c6 08 ... Call Trace: kvm_arch_init+0x17c/0x280 kvm_init+0x31/0x330 vmx_init+0xba/0x13a do_one_initcall+0x41/0x1c0 kernel_init_freeable+0x1f2/0x23b kernel_init+0x16/0x120 ret_from_fork+0x22/0x30
CVE-2021-47218	In the Linux kernel, the following vulnerability has been resolved: selinux: fix NULL-pointer dereference when hashtable allocation fails When the hash table slot array allocation fails in hashtable_init(), h->size is left initialized with a non-zero value, but the h->htable pointer is NULL. This may then cause a NULL pointer dereference, since the policydb code relies on the assumption that even after a failed hashtable_init(), hashtable_map() and hashtable_destroy() can be safely called on it. Yet, these detect an empty hashtable only by looking at the size. Fix this by making sure that hashtable_init() always leaves behind a valid empty hashtable when the allocation fails.

CVE ID	Description
CVE-2021-47219	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Fix out-of-bound read in resp_report_tgtpgs() The following issue was observed running syzkaller: BUG: KASAN: slab-out-of-bounds in memcpy include/linux/string.h:377 [inline] BUG: KASAN: slab-out-of-bounds in sg_copy_buffer+0x150/0x1c0 lib/scatterlist.c:831 Read of size 2132 at addr ffff8880aea95dc8 by task syz-executor.0/9815 CPU: 0 PID: 9815 Comm: syz-executor.0 Not tainted 4.19.202-00874-gfc0fe04215a9 #2 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014 Call Trace: __dump_stack lib/dump_stack.c:77 [inline] dump_stack+0xe4/0x14a lib/dump_stack.c:118 print_address_description+0x73/0x280 mm/kasan/report.c:253 kasan_report_error mm/kasan/report.c:352 [inline] kasan_report+0x272/0x370 mm/kasan/report.c:410 memcpy+0x1f/0x50 mm/kasan/kasan.c:302 memcpy include/linux/string.h:377 [inline] sg_copy_buffer+0x150/0x1c0 lib/scatterlist.c:831 fill_from_dev_buffer+0x14f/0x340 drivers/scsi/scsi_debug.c:1021 resp_report_tgtpgs+0x5aa/0x770 drivers/scsi/scsi_debug.c:1772 schedule_resp+0x464/0x12f0 drivers/scsi/scsi_debug.c:4429 scsi_debug_queuecommand+0x467/0x1390 drivers/scsi/scsi_debug.c:5835 scsi_dispatch_cmd+0x3fc/0x9b0 drivers/scsi/scsi_lib.c:1896 scsi_request_fn+0x1042/0x1810 drivers/scsi/scsi_lib.c:2034 __blk_run_queue_uncond block/blk-core.c:464 [inline] __blk_run_queue+0x1a4/0x380 block/blk-core.c:484 blk_execute_rq_nowait+0x1c2/0x2d0 block/blk-exec.c:78 sg_common_write.isra.19+0xd74/0x1dc0 drivers/scsi/sg.c:847 sg_write.part.23+0x6e0/0xd00 drivers/scsi/sg.c:716 sg_write+0x64/0xa0 drivers/scsi/sg.c:622 __vfs_write+0xed/0x690 fs/read_write.c:485 kill_bdev: block_device:00000000e138492c vfs_write+0x184/0x4c0 fs/read_write.c:549 ksys_write+0x107/0x240 fs/read_write.c:599 do_syscall_64+0xc2/0x560 arch/x86/entry/common.c:293 entry_SYSCALL_64_after_hwframe+0x49/0xbe We get 'alen' from command its type is int. If userspace passes a large length we will get a negative 'alen'. Switch n, alen, and rlen to u32.
CVE-2021-47222	In the Linux kernel, the following vulnerability has been resolved: net: bridge: fix vlan tunnel dst refcnt when egressing The egress tunnel code uses dst_clone() and directly sets the result which is wrong because the entry might have 0 refcnt or be already deleted, causing number of problems. It also triggers the WARN_ON() in dst_hold()[1] when a refcnt couldn't be taken. Fix it by using dst_hold_safe() and checking if a reference was actually taken before setting the dst. [1] dmesg WARN_ON log and following refcnt errors WARNING: CPU: 5 PID: 38 at include/net/dst.h:230 br_handle_egress_vlan_tunnel+0x10b/0x134 [bridge] Modules linked in: 8021q garp mrp bridge stp llc bonding ipv6 virtio_net CPU: 5 PID: 38 Comm: ksoftirqd/5 Kdump: loaded Tainted: G W 5.13.0-rc3+ #360 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-1.fc33 04/01/2014 RIP: 0010:br_handle_egress_vlan_tunnel+0x10b/0x134 [bridge] Code: e8 85 bc 01 e1 45 84 f6 74 90 45 31 f6 85 db 48 c7 c7 a0 02 19 a0 41 0f 94 c6 31 c9 31 d2 44 89 f6 e8 64 bc 01 e1 85 db 75 02 <0f> 0b 31 c9 31 d2 44 89 f6 48 c7 c7 70 02 19 a0 e8 4b bc 01 e1 49 RSP: 0018:ffff8881003d39e8 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000001 RDI: ffffffff01902a0 RBP: ffff8881040c6700 R08: 0000000000000000 R09: 0000000000000001 R10: 2ce93d0054fe0d00 R11: 54fe0d000000e000 R12: ffff888109515000 R13: 0000000000000000 R14: 0000000000000001 R15: 0000000000000401 FS: 0000000000000000(0000) GS:ffff88822bf40000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f42ba70f030 CR3: 0000000109926000 CR4: 0000000000000060 Call Trace: br_handle_vlan+0xbc/0xca [bridge] __br_forward+0x23/0x164 [bridge] deliver_clone+0x41/0x48 [bridge] br_handle_frame_finish+0x36f/0x3aa [bridge] ? skb_dst+0x2e/0x38 [bridge] ? br_handle_ingress_vlan_tunnel+0x3e/0x1c8 [bridge] ? br_handle_frame_finish+0x3aa/0x3aa [bridge] br_handle_frame+0x2c3/0x377 [bridge] ? __skb_pull+0x33/0x51 ? vlan_do_receive+0x4f/0x36a ? br_handle_frame_finish+0x3aa/0x3aa [bridge] __netif_receive_skb_core+0x539/0x7c6 ? __list_del_entry_valid+0x16e/0x1c2 __netif_receive_skb_list_core+0x6d/0xd6 netif_receive_skb_list_internal+0x1d9/0x1fa gro_normal_list+0x22/0x3e dev_gro_receive+0x55b/0x600 ? detach_buf_split+0x58/0x140 napi_gro_receive+0x94/0x12e virtnet_poll+0x15d/0x315 [virtio_net] __napi_poll+0x2c/0x1c9 net_rx_action+0xe6/0x1fb __do_softirq+0x115/0x2d8 run_ksoftirqd+0x18/0x20 smpboot_thread_fn+0x183/0x19c ? smpboot_unregister_percpu_thread+0x66/0x66 kthread+0x10a/0x10f ? kthread_mod_delayed_work+0xb6/0xb6 ret_from_fork+0x22/0x30 ---[end trace 49f61b07f775fd2b]--- dst_release: dst:00000000c02d677a refcnt:-1 dst_release underflow
CVE-2021-47223	In the Linux kernel, the following vulnerability has been resolved: net: bridge: fix vlan tunnel dst null pointer dereference This patch fixes a tunnel_dst null pointer dereference due to lockless access in the tunnel egress path. When deleting a vlan tunnel the tunnel_dst pointer is set to NULL without waiting a grace period (i.e. while it's still usable) and packets egressing are dereferencing it without checking. Use READ/WRITE_ONCE to annotate the lockless use of tunnel_id, use RCU for accessing tunnel_dst and make sure it is read only once and checked in the egress path. The dst is already properly RCU protected so we don't need to do anything fancy than to make sure tunnel_id and tunnel_dst are read only once and checked in the egress path.

CVE ID	Description
CVE-2021-47225	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix deadlock in AP/VLAN handling Syzbot reports that when you have AP_VLAN interfaces that are up and close the AP interface they belong to, we get a deadlock. No surprise - since we dev_close() them with the wiphy mutex held, which goes back into the netdev notifier in cfg80211 and tries to acquire the wiphy mutex there. To fix this, we need to do two things: 1) prevent changing iftype while AP_VLANs are up, we can't easily fix this case since cfg80211 already calls us with the wiphy mutex held, but change_interface() is relatively rare in drivers anyway, so changing iftype isn't used much (and userspace has to fall back to down/change/up anyway) 2) pull the dev_close() loop over VLANs out of the wiphy mutex section in the normal stop case
CVE-2021-47226	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Invalidate FPU state after a failed XRSTOR from a user buffer Both Intel and AMD consider it to be architecturally valid for XRSTOR to fail with #PF but nonetheless change the register state. The actual conditions under which this might occur are unclear [1], but it seems plausible that this might be triggered if one sibling thread unmaps a page and invalidates the shared TLB while another sibling thread is executing XRSTOR on the page in question. __fpu__restore_sig() can execute XRSTOR while the hardware registers are preserved on behalf of a different victim task (using the fpu_fpregs_owner_ctx mechanism), and, in theory, XRSTOR could fail but modify the registers. If this happens, then there is a window in which __fpu__restore_sig() could schedule out and the victim task could schedule back in without reloading its own FPU registers. This would result in part of the FPU state that __fpu__restore_sig() was attempting to load leaking into the victim task's user-visible state. Invalidate preserved FPU registers on XRSTOR failure to prevent this situation from corrupting any state. [1] Frequent readers of the errata lists might imagine "complex microarchitectural conditions".
CVE-2021-47227	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Prevent state corruption in __fpu__restore_sig() The non-compacted slowpath uses __copy_from_user() and copies the entire user buffer into the kernel buffer, verbatim. This means that the kernel buffer may now contain entirely invalid state on which XRSTOR will #GP. validate_user_xstate_header() can detect some of that corruption, but that leaves the onus on callers to clear the buffer. Prior to XSAVES support, it was possible just to reinitialize the buffer, completely, but with supervisor states that is not longer possible as the buffer clearing code split got it backwards. Fixing that is possible but not corrupting the state in the first place is more robust. Avoid corruption of the kernel XSAVE buffer by using copy_user_to_xstate() which validates the XSAVE header contents before copying the actual states to the kernel. copy_user_to_xstate() was previously only called for compacted-format kernel buffers, but it works for both compacted and non-compacted forms. Using it for the non-compacted form is slower because of multiple __copy_from_user() operations, but that cost is less important than robust code in an already slow path. [Changelog polished by Dave Hansen]

CVE ID	Description
CVE-2021-47230	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Immediately reset the MMU context when the SMM flag is cleared Immediately reset the MMU context when the vCPU's SMM flag is cleared so that the SMM flag in the MMU role is always synchronized with the vCPU's flag. If RSM fails (which isn't correctly emulated), KVM will bail without calling <code>post_leave_smm()</code> and leave the MMU in a bad state. The bad MMU role can lead to a NULL pointer dereference when grabbing a shadow page's rmap for a page fault as the initial lookups for the gfn will happen with the vCPU's SMM flag (=0), whereas the rmap lookup will use the shadow page's SMM flag, which comes from the MMU (=1). SMM has an entirely different set of memslots, and so the initial lookup can find a memslot (SMM=0) and then explode on the rmap memslot lookup (SMM=1). general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] CPU: 1 PID: 8410 Comm: syz-executor382 Not tainted 5.13.0-rc5-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010: __gfn_to_rmap arch/x86/kvm/mmu/mmu.c:935 [inline] RIP: 0010:gfn_to_rmap+0x2b0/0x4d0 arch/x86/kvm/mmu/mmu.c:947 Code: <42> 80 3c 20 00 74 08 4c 89 ff e8 f1 79 a9 00 4c 89 fb 4d 8b 37 44 RSP: 0018:ffffc90000ffef98 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff888015b9f414 RCX: ffff888019669c40 RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000001 RBP: 0000000000000001 R08: ffffffff811d9cdb R09: ffffed10065a6002 R10: ffffed10065a6002 R11: 0000000000000000 R12: dffffc0000000000 R13: 0000000000000003 R14: 0000000000000001 R15: 0000000000000000 FS: 000000000124b300(0000) GS:ffff8880b9b00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000000028e31000 CR4: 00000000001526e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: rmap_add arch/x86/kvm/mmu/mmu.c:965 [inline] mmu_set_spte+0x862/0xe60 arch/x86/kvm/mmu/mmu.c:2604 __direct_map arch/x86/kvm/mmu/mmu.c:2862 [inline] direct_page_fault+0x1f74/0x2b70 arch/x86/kvm/mmu/mmu.c:3769 kvm_mmu_do_page_fault arch/x86/kvm/mmu.h:124 [inline] kvm_mmu_page_fault+0x199/0x1440 arch/x86/kvm/mmu/mmu.c:5065 vmx_handle_exit+0x26/0x160 arch/x86/kvm/vmx/vmx.c:6122 vcpu_enter_guest+0x3bdd/0x9630 arch/x86/kvm/x86.c:9428 vcpu_run+0x416/0xc20 arch/x86/kvm/x86.c:9494 kvm_arch_vcpu_ioctl_run+0x4e8/0xa40 arch/x86/kvm/x86.c:9722 kvm_vcpu_ioctl+0x70f/0xbb0 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:3460 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:1069 [inline] __se_sys_ioctl+0xfb/0x170 fs/ioctl.c:1055 do_syscall_64+0x3f/0xb0 arch/x86/entry/common.c:47 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x440ce9
CVE-2021-47238	In the Linux kernel, the following vulnerability has been resolved: net: ipv4: fix memory leak in <code>ip_mc_add1_src</code> BUG: memory leak unreferenced object 0xffff888101bc4c00 (size 32): comm "syz-executor527", pid 360, jiffies 4294807421 (age 19.329s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 ac 14 14 bb 00 00 02 00 backtrace: [<code><00000000f17c5244></code>] kcalloc include/linux/slab.h:558 [inline] [<code><00000000f17c5244></code>] kzalloc include/linux/slab.h:688 [inline] [<code><00000000f17c5244></code>] <code>ip_mc_add1_src</code> net/ipv4/igmp.c:1971 [inline] [<code><00000000f17c5244></code>] <code>ip_mc_add_src</code>+0x95f/0xdb0 net/ipv4/igmp.c:2095 [<code><000000001cb99709></code>] <code>ip_mc_source</code>+0x84c/0xea0 net/ipv4/igmp.c:2416 [<code><0000000052cf19ed></code>] <code>do_ip_setsockopt</code> net/ipv4/ip_sockglue.c:1294 [inline] [<code><0000000052cf19ed></code>] <code>ip_setsockopt</code>+0x114b/0x30c0 net/ipv4/ip_sockglue.c:1423 [<code><00000000477edfbc></code>] <code>raw_setsockopt</code>+0x13d/0x170 net/ipv4/raw.c:857 [<code><00000000e75ca9bb></code>] <code>__sys_setsockopt</code>+0x158/0x270 net/socket.c:2117 [<code><00000000bdb993a8></code>] <code>__do_sys_setsockopt</code> net/socket.c:2128 [inline] [<code><00000000bdb993a8></code>] <code>__se_sys_setsockopt</code> net/socket.c:2125 [inline] [<code><00000000bdb993a8></code>] <code>__x64_sys_setsockopt</code>+0xba/0x150 net/socket.c:2125 [<code><000000006a1ffdbd></code>] <code>do_syscall_64</code>+0x40/0x80 arch/x86/entry/common.c:47 [<code><00000000b11467c4></code>] <code>entry_SYSCALL_64_after_hwframe</code>+0x44/0xae In commit 24803f38a5c0 ("igmp: do not remove igmp source list info when set link down"), the <code>ip_mc_clear_src()</code> in <code>ip_mc_destroy_dev()</code> was removed, because it was also called in <code>igmpv3_clear_delrec()</code>. Rough callgraph: <code>inetdev_destroy -> ip_mc_destroy_dev -> igmpv3_clear_delrec -> ip_mc_clear_src -> RCU_INIT_POINTER(dev->ip_ptr, NULL)</code> However, <code>ip_mc_clear_src()</code> called in <code>igmpv3_clear_delrec()</code> doesn't release <code>in_dev->mc_list->sources</code>. And <code>RCU_INIT_POINTER()</code> assigns the NULL to <code>dev->ip_ptr</code>. As a result, <code>in_dev</code> cannot be obtained through <code>inetdev_by_index()</code> and then <code>in_dev->mc_list->sources</code> cannot be released by <code>ip_mc_del1_src()</code> in the <code>sock_close</code>. Rough call sequence goes like: <code>sock_close -> __sock_release -> inet_release -> ip_mc_drop_socket -> inetdev_by_index -> ip_mc_leave_src -> ip_mc_del_src -> ip_mc_del1_src</code> So we still need to call <code>ip_mc_clear_src()</code> in <code>ip_mc_destroy_dev()</code> to free <code>in_dev->mc_list->sources</code>.
CVE-2021-47239	In the Linux kernel, the following vulnerability has been resolved: net: usb: fix possible use-after-free in <code>sm5c75xx_bind</code> The commit 46a8b29c6306 ("net: usb: fix memory leak in <code>sm5c75xx_bind</code>") fails to clean up the work scheduled in <code>sm5c75xx_reset->sm5c75xx_set_multicast</code>, which leads to use-after-free if the work is scheduled to start after the deallocation. In addition, this patch also removes a dangling pointer - <code>dev->data[0]</code>. This patch calls <code>cancel_work_sync</code> to cancel the scheduled work and set the dangling pointer to NULL.

CVE ID	Description
CVE-2021-47241	In the Linux kernel, the following vulnerability has been resolved: ethtool: strset: fix message length calculation Outer nest for ETHTOOL_A_STRSET_STRINGSETS is not accounted for. This may result in ETHTOOL_MSG_STRSET_GET producing a warning like: calculated message payload length (684) not sufficient WARNING: CPU: 0 PID: 30967 at net/ethtool/netlink.c:369 ethnl_default_doit +0x87a/0xa20 and a splat. As usually with such warnings three conditions must be met for the warning to trigger: - there must be no skb size rounding up (e.g. reply_size of 684); - string set must be per-device (so that the header gets populated); - the device name must be at least 12 characters long, all in all with current user space it looks like reading priv flags is the only place this could potentially happen. Or with syzbot :)
CVE-2021-47242	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix soft lookup in subflow_error_report() Maxim reported a soft lookup in subflow_error_report(): watchdog: BUG: soft lockup - CPU#0 stuck for 22s! [swapper/0:0] RIP: 0010:native_queued_spin_lock_slowpath RSP: 0018:ffffa859c0003bc0 EFLAGS: 00000202 RAX: 0000000000000101 RBX: 0000000000000001 RCX: 0000000000000000 RDX: ffff9195c2772d88 RSI: 0000000000000000 RDI: ffff9195c2772d88 RBP: ffff9195c2772d00 R08: 00000000000067b0 R09: c6e31da9eb1e44f4 R10: ffff9195ef379700 R11: ffff9195edb50710 R12: ffff9195c2772d88 R13: ffff9195f500e3d0 R14: ffff9195ef379700 R15: ffff9195ef379700 FS: 0000000000000000(0000) GS:ffff91961f400000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000c000407000 CR3: 0000000002988000 CR4: 00000000000006f0 Call Trace: <IRQ> _raw_spin_lock_bh subflow_error_report mptcp_subflow_data_available __mptcp_move_skbs_from_subflow mptcp_data_ready tcp_data_queue tcp_rcv_established tcp_v4_do_rcv tcp_v4_rcv ip_protocol_deliver_rcu ip_local_deliver_finish __netif_receive_skb_one_core netif_receive_skb rtl8139_poll 8139too __napi_poll net_rx_action __do_softirq __irq_exit_rcu common_interrupt </IRQ> The calling function - mptcp_subflow_data_available() - can be invoked from different contexts: - plain ssk socket lock - ssk socket lock + mptcp_data_lock - ssk socket lock + mptcp_data_lock + msk socket lock. Since subflow_error_report() tries to acquire the mptcp_data_lock, the latter two call chains will cause soft lookup. This change addresses the issue moving the error reporting call to outer functions, where the held locks list is known and the we can acquire only the needed one.

CVE ID	Description
CVE-2021-47246	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix page reclaim for dead peer hairpin When adding a hairpin flow, a firmware-side send queue is created for the peer net device, which claims some host memory pages for its internal ring buffer. If the peer net device is removed/unbound before the hairpin flow is deleted, then the send queue is not destroyed which leads to a stack trace on pci device remove: [748.005230] mlx5_core 0000:08:00.2: wait_func:1094:(pid 12985): MANAGE_PAGES(0x108) timeout. Will cause a leak of a command resource [748.005231] mlx5_core 0000:08:00.2: reclaim_pages:514:(pid 12985): failed reclaiming pages: err -110 [748.001835] mlx5_core 0000:08:00.2: mlx5_reclaim_root_pages:653:(pid 12985): failed reclaiming pages (-110) for func id 0x0 [748.002171] -----[cut here]----- [748.001177] FW pages counter is 4 after reclaiming all pages [748.001186] WARNING: CPU: 1 PID: 12985 at drivers/net/ethernet/mellanox/mlx5/core/pagealloc.c:685 mlx5_reclaim_startup_pages+0x34b/0x460 [mlx5_core] [+0.002771] Modules linked in: cls_flower mlx5_ib mlx5_core ptp pps_core act_mirred sch_ingress openvswitch nsh xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 br_netfilter rpcrdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi rdma_cm ib_umad ib_ipoib iw_cm ib_cm ib_uverbs ib_core overlay fuse [last unloaded: pps_core] [748.007225] CPU: 1 PID: 12985 Comm: tee Not tainted 5.12.0+ #1 [748.001376] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 [748.002315] RIP: 0010:mlx5_reclaim_startup_pages+0x34b/0x460 [mlx5_core] [748.001679] Code: 28 00 00 00 0f 85 22 01 00 00 48 81 c4 b0 00 00 00 31 c0 5b 5d 41 5c 41 5d 41 5e 41 5f c3 48 c7 c7 40 cc 19 a1 e8 9f 71 0e e2 <0f> 0b e9 30 ff ff ff 48 c7 c7 a0 cc 19 a1 e8 8c 71 0e e2 0f 0b e9 [748.003781] RSP: 0018:ffff88815220faf8 EFLAGS: 00010286 [748.001149] RAX: 0000000000000000 RBX: ffff8881b4900280 RCX: 0000000000000000 [748.001445] RDX: 0000000000000002 RSI: 0000000000000004 RDI: ffffd102a441f51 [748.001614] RBP: 00000000000032b9 R08: 0000000000000001 R09: ffffd1054a15ee8 [748.001446] R10: ffff8882a50af73b R11: ffffd1054a15ee7 R12: ffff8bfff07c1e30 [748.001447] R13: dffffc0000000000 R14: ffff8881b492cba8 R15: 0000000000000000 [748.001429] FS: 00007f58bd08b580(0000) GS:ffff8882a5080000(0000) knlGS:0000000000000000 [748.001695] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [748.001309] CR2: 000055a026351740 CR3: 00000001d3b48006 CR4: 0000000000370ea0 [748.001506] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [748.001483] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [748.001654] Call Trace: [748.000576] ? mlx5_satisfy_startup_pages+0x290/0x290 [mlx5_core] [748.001416] ? mlx5_cmd_teardown_hca+0xa2/0xd0 [mlx5_core] [748.001354] ? mlx5_cmd_init_hca+0x280/0x280 [mlx5_core] [748.001203] mlx5_function_teardown+0x30/0x60 [mlx5_core] [748.001275] mlx5_uninit_one+0xa7/0xc0 [mlx5_core] [748.001200] remove_one+0x5f/0xc0 [mlx5_core] [748.001075] pci_device_remove+0x9f/0x1d0 [748.000833] device_release_driver_internal+0x1e0/0x490 [748.001207] unbind_store+0x19f/0x200 [748.000942] ? sysfs_file_ops+0x170/0x170 [748.001000] kernfs_fop_write_iter+0x2bc/0x450 [748.000970] new_sync_write+0x373/0x610 [748.001124] ? new_sync_read+0x600/0x600 [748.001057] ? lock_acquire+0x4d6/0x700 [748.000908] ? lockdep_hardirqs_on_prepare+0x400/0x400 [748.001126] ? fd_install+0x1c9/0x4d0 [748.000951] vfs_write+0x4d0/0x800 [748.000804] ksys_write+0xf9/0x1d0 [748.000868] ? __x64_sys_read+0xb0/0xb0 [748.000811] ? filp_open+0x50/0x50 [748.000919] ? syscall_enter_from_user_mode+0x1d/0x50 [748.001223] do_syscall_64+0x3f/0x80 [748.000892] entry_SYSCALL_64_after_hwframe+0x44/0xae [748.00 --- truncated---

CVE ID	Description
CVE-2021-47247	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix use-after-free of encap entry in neigh update handler Function <code>mlx5e_rep_neigh_update()</code> wasn't updated to accommodate <code>rtln</code> lock removal from TC filter update path and properly handle concurrent encap entry insertion/deletion which can lead to following use-after-free: [23827.464923] ===== [23827.469446] BUG: KASAN: use-after-free in <code>mlx5e_encap_take+0x72/0x140</code> [<code>mlx5_core</code>] [23827.470971] Read of size 4 at addr <code>ffff8881d132228c</code> by task <code>kworker/u20:6/21635</code> [23827.472251] [23827.472615] CPU: 9 PID: 21635 Comm: <code>kworker/u20:6</code> Not tainted 5.13.0-rc3+ #5 [23827.473788] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 [23827.475639] Workqueue: <code>mlx5e mlx5e_rep_neigh_update</code> [<code>mlx5_core</code>] [23827.476731] Call Trace: [23827.477260] <code>dump_stack+0xbb/0x107</code> [23827.477906] <code>print_address_description.constprop.0+0x18/0x140</code> [23827.478896] ? <code>mlx5e_encap_take+0x72/0x140</code> [<code>mlx5_core</code>] [23827.479879] ? <code>mlx5e_encap_take+0x72/0x140</code> [<code>mlx5_core</code>] [23827.480905] <code>kasan_report.cold+0x7c/0xd8</code> [23827.481701] ? <code>mlx5e_encap_take+0x72/0x140</code> [<code>mlx5_core</code>] [23827.482744] <code>kasan_check_range+0x145/0x1a0</code> [23827.493112] <code>mlx5e_encap_take+0x72/0x140</code> [<code>mlx5_core</code>] [23827.494054] ? <code>mlx5e_tc_tun_encap_info_equal_generic+0x140/0x140</code> [<code>mlx5_core</code>] [23827.495296] <code>mlx5e_rep_neigh_update+0x41e/0x5e0</code> [<code>mlx5_core</code>] [23827.496338] ? <code>mlx5e_rep_neigh_entry_release+0xb80/0xb80</code> [<code>mlx5_core</code>] [23827.497486] ? <code>read_word_at_a_time+0xe/0x20</code> [23827.498250] ? <code>strncpy+0xa0/0x2a0</code> [23827.498889] <code>process_one_work+0x8ac/0x14e0</code> [23827.499638] ? <code>lockdep_hardirqs_on_prepare+0x400/0x400</code> [23827.500537] ? <code>pwq_dec_nr_in_flight+0x2c0/0x2c0</code> [23827.501359] ? <code>rwlock_bug.part.0+0x90/0x90</code> [23827.502116] <code>worker_thread+0x53b/0x1220</code> [23827.502831] ? <code>process_one_work+0x14e0/0x14e0</code> [23827.503627] <code>kthread+0x328/0x3f0</code> [23827.504254] ? <code>_raw_spin_unlock_irq+0x24/0x40</code> [23827.505065] ? <code>_kthread_bind_mask+0x90/0x90</code> [23827.505912] <code>ret_from_fork+0x1f/0x30</code> [23827.506621] [23827.506987] Allocated by task 28248: [23827.507694] <code>kasan_save_stack+0x1b/0x40</code> [23827.508476] ? <code>__kasan_kmalloc+0x7c/0x90</code> [23827.509197] <code>mlx5e_attach_encap+0xde1/0x1d40</code> [<code>mlx5_core</code>] [23827.510194] <code>mlx5e_tc_add_fdb_flow+0x397/0xc40</code> [<code>mlx5_core</code>] [23827.511218] <code>__mlx5e_add_fdb_flow+0x519/0xb30</code> [<code>mlx5_core</code>] [23827.512234] <code>mlx5e_configure_flow+0x191c/0x4870</code> [<code>mlx5_core</code>] [23827.513298] <code>tc_setup_cb_add+0x1d5/0x420</code> [23827.514023] <code>fl_hw_replace_filter+0x382/0x6a0</code> [<code>cls_flow</code>] [23827.514975] <code>fl_change+0x2ceb/0x4a51</code> [<code>cls_flow</code>] [23827.515821] <code>tc_new_tfilter+0x89a/0x2070</code> [23827.516548] <code>rtnetlink_rcv_msg+0x644/0x8c0</code> [23827.517300] <code>netlink_rcv_skb+0x11d/0x340</code> [23827.518021] <code>netlink_unicast+0x42b/0x700</code> [23827.518742] <code>netlink_sendmsg+0x743/0xc20</code> [23827.519467] <code>sock_sendmsg+0xb2/0xe0</code> [23827.520131] <code>__sys_sendmsg+0x590/0x770</code> [23827.520851] <code>__sys_sendmsg+0xd8/0x160</code> [23827.521552] <code>__sys_sendmsg+0xb7/0x140</code> [23827.522238] <code>do_syscall_64+0x3a/0x70</code> [23827.522907] <code>entry_SYSCALL_64_after_hwframe+0x44/0xae</code> [23827.523797] [23827.524163] Freed by task 25948: [23827.524780] <code>kasan_save_stack+0x1b/0x40</code> [23827.525488] <code>kasan_set_track+0x1c/0x30</code> [23827.526187] <code>kasan_set_free_info+0x20/0x30</code> [23827.526968] <code>__kasan_slab_free+0xed/0x130</code> [23827.527709] <code>slab_free_freelist_hook+0xcf/0x1d0</code> [23827.528528] <code>kmem_cache_free_bulk+0x33a/0x6e0</code> [23827.529317] <code>kfree_rcu_work+0x55f/0xb70</code> [23827.530024] <code>process_one_work+0x8ac/0x14e0</code> [23827.530770] <code>worker_thread+0x53b/0x1220</code> [23827.531480] <code>kthread+0x328/0x3f0</code> [23827.532114] <code>ret_from_fork+0x1f/0x30</code> [23827.532785] [23827.533147] Last potentially related work creation: [23827.534007] <code>kasan_save_stack+0x1b/0x40</code> [23827.534710] <code>kasan_record_aux_stack+0xab/0xc0</code> [23827.535492] <code>kfree_call_rcu+0x31/0x7b0</code> [23827.536206] <code>mlx5e_tc_del</code> ---truncated---
CVE-2021-47248	In the Linux kernel, the following vulnerability has been resolved: udp: fix race between <code>close()</code> and <code>udp_abort()</code> Kaustubh reported and diagnosed a panic in <code>udp_lib_lookup()</code>. The root cause is <code>udp_abort()</code> racing with <code>close()</code>. Both racing functions acquire the socket lock, but <code>udp{v6}_destroy_sock()</code> release it before performing destructive actions. We can't easily extend the socket lock scope to avoid the race, instead use the <code>SOCK_DEAD</code> flag to prevent <code>udp_abort</code> from doing any action when the critical race happens. Diagnosed-and-tested-by: Kaustubh Pandey <kapandey@codeaurora.org>

CVE ID	Description
CVE-2021-47250	In the Linux kernel, the following vulnerability has been resolved: net: ipv4: fix memory leak in netlbl_cipsov4_add_std Reported by syzkaller: BUG: memory leak unreferenced object 0xffff888105df7000 (size 64): comm "syz-executor842", pid 360, jiffies 4294824824 (age 22.546s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000e67ed558>] kcalloc include/linux/slab.h:590 [inline] [<00000000e67ed558>] kzalloc include/linux/slab.h:720 [inline] [<00000000e67ed558>] netlbl_cipsov4_add_std net/netlabel/netlabel_cipso_v4.c:145 [inline] [<00000000e67ed558>] netlbl_cipsov4_add+0x390/0x2340 net/netlabel/netlabel_cipso_v4.c:416 [<0000000006040154>] genl_family_rcv_msg_doit.isra.0+0x20e/0x320 net/netlink/genetlink.c:739 [<00000000204d7a1c>] genl_family_rcv_msg net/netlink/genetlink.c:783 [inline] [<00000000204d7a1c>] genl_rcv_msg+0x2bf/0x4f0 net/netlink/genetlink.c:800 [<00000000c0d6a995>] netlink_rcv_skb+0x134/0x3d0 net/netlink/af_netlink.c:2504 [<00000000d78b9d2c>] genl_rcv+0x24/0x40 net/netlink/genetlink.c:811 [<000000009733081b>] netlink_unicast_kernel net/netlink/af_netlink.c:1314 [inline] [<000000009733081b>] netlink_unicast+0x4a0/0x6a0 net/netlink/af_netlink.c:1340 [<00000000d5fd43b8>] netlink_sendmsg+0x789/0xc70 net/netlink/af_netlink.c:1929 [<000000000a2d1e40>] sock_sendmsg_nosec net/socket.c:654 [inline] [<000000000a2d1e40>] sock_sendmsg+0x139/0x170 net/socket.c:674 [<00000000321d1969>] ____sys_sendmsg+0x658/0x7d0 net/socket.c:2350 [<00000000964e16bc>] __sys_sendmsg+0xf8/0x170 net/socket.c:2404 [<000000001615e288>] __sys_sendmsg+0xd3/0x190 net/socket.c:2433 [<000000004ee8b6a5>] do_syscall_64+0x37/0x90 arch/x86/entry/common.c:47 [<00000000171c7cee>] entry_SYSCALL_64_after_hwframe+0x44/0xae The memory of doi_def->map.std pointing is allocated in netlbl_cipsov4_add_std, but no place has freed it. It should be freed in cipso_v4_doi_free which frees the cipso DOI resource.
CVE-2021-47251	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix skb length check in ieee80211_scan_rx() Replace hard-coded compile-time constants for header length check with dynamic determination based on the frame type. Otherwise, we hit a validation WARN_ON in cfg80211 later. [style fixes, reword commit message]
CVE-2021-47254	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix use-after-free in gfs2_glock_shrink_scan The GLF_LRU flag is checked under lru_lock in gfs2_glock_remove_from_lru() to remove the glock from the lru list in __gfs2_glock_put(). On the shrink scan path, the same flag is cleared under lru_lock but because of cond_resched_lock(&lru_lock) in gfs2_dispose_glock_lru(), progress on the put side can be made without deleting the glock from the lru list. Keep GLF_LRU across the race window opened by cond_resched_lock(&lru_lock) to ensure correct behavior on both sides - clear GLF_LRU after list_del under lru_lock.
CVE-2021-47255	In the Linux kernel, the following vulnerability has been resolved: kvm: LAPIC: Restore guard to prevent illegal APIC register access Per the SDM, "any access that touches bytes 4 through 15 of an APIC register may cause undefined behavior and must not be executed." Worse, such an access in kvm_lapic_reg_read can result in a leak of kernel stack contents. Prior to commit 01402cf81051 ("kvm: LAPIC: write down valid APIC registers"), such an access was explicitly disallowed. Restore the guard that was removed in that commit.
CVE-2021-47258	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix error handling of scsi_host_alloc() After device is initialized via device_initialize(), or its name is set via dev_set_name(), the device has to be freed via put_device(). Otherwise device name will be leaked because it is allocated dynamically in dev_set_name(). Fix the leak by replacing kfree() with put_device(). Since scsi_host_dev_release() properly handles IDA and kthread removal, remove special-casing these from the error handling as well.
CVE-2021-47259	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix use-after-free in nfs4_init_client() KASAN reports a use-after-free when attempting to mount two different exports through two different NICs that belong to the same server. Olga was able to hit this with kernels starting somewhere between 5.7 and 5.10, but I traced the patch that introduced the clear_bit() call to 4.13. So something must have changed in the refcounting of the clp pointer to make this call to nfs_put_client() the very last one.
CVE-2021-47260	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix a potential NULL dereference in nfs_get_client() None of the callers are expecting NULL returns from nfs_get_client() so this code will lead to an Oops. It's better to return an error pointer. I expect that this is dead code so hopefully no one is affected.

CVE ID	Description
CVE-2021-47261	In the Linux kernel, the following vulnerability has been resolved: IB/mlx5: Fix initializing CQ fragments buffer The function init_cq_frag_buf() can be called to initialize the current CQ fragments buffer cq->buf, or the temporary cq->resize_buf that is filled during CQ resize operation. However, the offending commit started to use function get_cqe() for getting the CQEs, the issue with this change is that get_cqe() always returns CQEs from cq->buf, which leads us to initialize the wrong buffer, and in case of enlarging the CQ we try to access elements beyond the size of the current cq->buf and eventually hit a kernel panic. [exception RIP: init_cq_frag_buf+103] [ffff9f799ddcbcd8] mlx5_ib_resize_cq at ffffffff0835d60 [mlx5_ib] [ffff9f799ddcbdb0] ib_resize_cq at ffffffff05270df [ib_core] [ffff9f799ddcbdc0] llc_rdma_setup_qp at ffffffff0a6a712 [llt] [ffff9f799ddcbe10] llc_rdma_cc_event_action at ffffffff0a6b411 [llt] [ffff9f799ddcbe98] llc_rdma_client_conn_thread at ffffffff0a6bb75 [llt] [ffff9f799ddcbec8] kthread at ffffffff0a66c5da1 [ffff9f799ddcbf50] ret_from_fork_nospec_begin at ffffffff0a6d95ddd Fix it by getting the needed CQE by calling mlx5_frag_buf_get_wqe() that takes the correct source buffer as a parameter.
CVE-2021-47262	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Ensure liveness of nested VM-Enter fail tracepoint message Use the __string() machinery provided by the tracing subsystem to make a copy of the string literals consumed by the "nested VM-Enter failed" tracepoint. A complete copy is necessary to ensure that the tracepoint can't outlive the data/memory it consumes and dereference stale memory. Because the tracepoint itself is defined by kvm, if kvm-intel and/or kvm-amd are built as modules, the memory holding the string literals defined by the vendor modules will be freed when the module is unloaded, whereas the tracepoint and its data in the ring buffer will live until kvm is unloaded (or "indefinitely" if kvm is built-in). This bug has existed since the tracepoint was added, but was recently exposed by a new check in tracing to detect exactly this type of bug. fmt: '%s%s' current_buffer: 'vmx_dirty_log_t-140127 [003] kvm_nested_vmenter_failed: ' WARNING: CPU: 3 PID: 140134 at kernel/trace/trace.c:3759 trace_check_vprintf+0x3be/0x3e0 CPU: 3 PID: 140134 Comm: less Not tainted 5.13.0-rc1-ce2e73ce600a-req #184 Hardware name: ASUS Q87M-E/Q87M-E, BIOS 1102 03/03/2014 RIP: 0010:trace_check_vprintf+0x3be/0x3e0 Code: <0f> 0b 44 8b 4c 24 1c e9 a9 fe ff ff c6 44 02 ff 00 49 8b 97 b0 20 RSP: 0018:ffffa895cc37bcb0 EFLAGS: 00010282 RAX: 0000000000000000 RBX: fffffa895cc37bd08 RCX: 0000000000000027 RDX: 00000000000000027 RSI: 00000000ffffdfff RDI: fffff9766cfad74f8 RBP: ffffffc0a041d4 R08: fffff9766cfad74f0 R09: fffffa895cc37bad8 R10: 0000000000000001 R11: 0000000000000001 R12: ffffffc0a041d4 R13: ffffffc0f4dba8 R14: 0000000000000000 R15: fffff976409f2c000 FS: 00007f92fa200740(0000) GS:ffff9766cfac0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000559bd11b0000 CR3: 000000019fbaa002 CR4: 00000000001726e0 Call Trace: trace_event_vprintf+0x5e/0x80 trace_raw_output_kvm_nested_vmenter_failed+0x3a/0x60 [kvm] print_trace_line+0x1dd/0x4e0 s_show+0x45/0x150 seq_read_iter+0x2d5/0x4c0 seq_read+0x106/0x150 vfs_read+0x98/0x180 ksys_read+0x5f/0xe0 do_syscall_64+0x40/0xb0 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2021-47264	In the Linux kernel, the following vulnerability has been resolved: ASoC: core: Fix Null-point-dereference in fmt_single_name() Check the return value of devm_kstrdup() in case of Null-point-dereference.
CVE-2021-47265	In the Linux kernel, the following vulnerability has been resolved: RDMA: Verify port when creating flow rule Validate port value provided by the user and with that remove no longer needed validation by the driver. The missing check in the mlx5_ib driver could cause to the below oops. Call trace: _create_flow_rule+0x2d4/0xf28 [mlx5_ib] mlx5_ib_create_flow+0x2d0/0x5b0 [mlx5_ib] ib_uverbs_ex_create_flow+0x4cc/0x624 [ib_uverbs] ib_uverbs_handler_UVERBS_METHOD_INVOKE_WRITE+0xd4/0x150 [ib_uverbs] ib_uverbs_cmd_verbs.isra.7+0xb28/0xc50 [ib_uverbs] ib_uverbs_ioctl+0x158/0x1d0 [ib_uverbs] do_vfs_ioctl+0xd0/0xaf0 ksys_ioctl+0x84/0xb4 __arm64_sys_ioctl+0x28/0xc4 el0_svc_common.constprop.3+0xa4/0x254 el0_svc_handler+0x84/0xa0 el0_svc+0x10/0x26c Code: b9401260 f9615681 51000400 8b001c20 (f9403c1a)

CVE ID	Description
CVE-2021-47266	In the Linux kernel, the following vulnerability has been resolved: RDMA/ipoib: Fix warning caused by destroying non-initial netns After the commit 5ce2dced8e95 ("RDMA/ipoib: Set rtnl_link_ops for ipoib interfaces"), if the IPoIB device is moved to non-initial netns, destroying that netns lets the device vanish instead of moving it back to the initial netns, This is happening because default_device_exit() skips the interfaces due to having rtnl_link_ops set. Steps to reproduce: ip netns add foo ip link set mlx5_ib0 netns foo ip netns delete foo WARNING: CPU: 1 PID: 704 at net/core/dev.c:11435 netdev_exit+0x3f/0x50 Modules linked in: xt_CHECKSUM xt_MASQUERADE xt_contrack ipt_REJECT nf_reject_ipv4 nft_compat nft_counter nft_chain_nat nf_nat nf_contrack nf_defrag_ipv6 nf_defrag_ipv4 nf_tables nfnetlink tun d fuse CPU: 1 PID: 704 Comm: kworker/u64:3 Tainted: G S W 5.13.0-rc1+ #1 Hardware name: Dell Inc. PowerEdge R630/02C2CP, BIOS 2.1.5 04/11/2016 Workqueue: netns cleanup_net RIP: 0010:netdev_exit+0x3f/0x50 Code: 48 8b bb 30 01 00 00 e8 ef 81 b1 ff 48 81 fb c0 3a 54 a1 74 13 48 8b 83 90 00 00 00 48 81 c3 90 00 00 00 48 39 d8 75 02 5b c3 <0f> 0b 5b c3 66 66 2e 0f 1f 84 00 00 00 00 66 90 0f 1f 44 00 RSP: 0018:ffffb297079d7e08 EFLAGS: 00010206 RAX: ffff8eb542c00040 RBX: ffff8eb541333150 RCX: 000000008010000d RDX: 000000008010000e RSI: 000000008010000d RDI: ffff8eb440042c00 RBP: ffff8b297079d7e48 R08: 0000000000000001 R09: ffffffff9fdeac00 R10: ffff8eb5003be000 R11: 0000000000000001 R12: ffffffff1545620 R13: ffffffff1545628 R14: 0000000000000000 R15: ffffffff1543b20 FS: 0000000000000000(0000) GS:ffff8ed37fa00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005601b5f4c2e8 CR3: 0000001fc8c10002 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: ops_exit_list.isra.9+0x36/0x70 cleanup_net+0x234/0x390 process_one_work+0x1cb/0x360 ? process_one_work+0x360/0x360 worker_thread+0x30/0x370 ? process_one_work+0x360/0x360 kthread+0x116/0x130 ? kthread_park+0x80/0x80 ret_from_fork+0x22/0x30 To avoid the above warning and later on the kernel panic that could happen on shutdown due to a NULL pointer dereference, make sure to set the netns_refund flag that was introduced by commit 3a5ca857079e ("can: dev: Move device back to init netns on owning netns delete") to properly restore the IPoIB interfaces to the initial netns.
CVE-2021-47276	In the Linux kernel, the following vulnerability has been resolved: frtrace: Do not blindly read the ip address in frtrace_bug() It was reported that a bug on arm64 caused a bad ip address to be used for updating into a nop in frtrace_init(), but the error path (rightfully) returned -EINVAL and not -EFAULT, as the bug caused more than one error to occur. But because -EINVAL was returned, the frtrace_bug() tried to report what was at the location of the ip address, and read it directly. This caused the machine to panic, as the ip was not pointing to a valid memory address. Instead, read the ip address with copy_from_kernel_nofault() to safely access the memory, and if it faults, report that the address faulted, otherwise report what was in that location.
CVE-2021-47277	In the Linux kernel, the following vulnerability has been resolved: kvm: avoid speculation-based attacks from out-of-range memslot accesses KVM's mechanism for accessing guest memory translates a guest physical address (gpa) to a host virtual address using the right-shifted gpa (also known as gfn) and a struct kvm_memory_slot. The translation is performed in __gfn_to_hva_memslot using the following formula: hva = slot->userspace_addr + (gfn - slot->base_gfn) * PAGE_SIZE It is expected that gfn falls within the boundaries of the guest's physical memory. However, a guest can access invalid physical addresses in such a way that the gfn is invalid. __gfn_to_hva_memslot is called from kvm_vcpu_gfn_to_hva_prot, which first retrieves a memslot through __gfn_to_memslot. While __gfn_to_memslot does check that the gfn falls within the boundaries of the guest's physical memory or not, a CPU can speculate the result of the check and continue execution speculatively using an illegal gfn. The speculation can result in calculating an out-of-bounds hva. If the resulting host virtual address is used to load another guest physical address, this is effectively a Spectre gadget consisting of two consecutive reads, the second of which is data dependent on the first. Right now it's not clear if there are any cases in which this is exploitable. One interesting case was reported by the original author of this patch, and involves visiting guest page tables on x86. Right now these are not vulnerable because the hva read goes through get_user(), which contains an LFENCE speculation barrier. However, there are patches in progress for x86 uaccess.h to mask kernel addresses instead of using LFENCE; once these land, a guest could use speculation to read from the VMM's ring 3 address space. Other architectures such as ARM already use the address masking method, and would be susceptible to this same kind of data-dependent access gadgets. Therefore, this patch proactively protects from these attacks by masking out-of-bounds gfn's in __gfn_to_hva_memslot, which blocks speculation of invalid hvas. Sean Christopherson noted that this patch does not cover kvm_read_guest_offset_cached. This however is limited to a few bytes past the end of the cache, and therefore it is unlikely to be useful in the context of building a chain of data dependent accesses.

CVE ID	Description
CVE-2021-47280	In the Linux kernel, the following vulnerability has been resolved: drm: Fix use-after-free read in drm_getunique() There is a time-of-check-to-time-of-use error in drm_getunique() due to retrieving file_priv->master prior to locking the device's master mutex. An example can be seen in the crash report of the use-after-free error found by Syzbot: https://syzkaller.appspot.com/bug?id=148d2f1dfac64af52ffd27b661981a540724f803 In the report, the master pointer was used after being freed. This is because another process had acquired the device's master mutex in drm_setmaster_ioctl(), then overwrote fpriv->master in drm_new_set_master(). The old value of fpriv->master was subsequently freed before the mutex was unlocked. To fix this, we lock the device's master mutex before retrieving the pointer from from fpriv->master. This patch passes the Syzbot reproducer test.
CVE-2021-47290	In the Linux kernel, the following vulnerability has been resolved: scsi: target: Fix NULL dereference on XCOPY completion CPU affinity control added with commit 39ae3edda325 ("scsi: target: core: Make completion affinity configurable") makes target_complete_cmd() queue work on a CPU based on se_tpg->se_tpg_wwn->cmd_compl_affinity state. LIO's EXTENDED COPY worker is a special case in that read/write cmds are dispatched using the global xcopy_pt_tpg, which carries a NULL se_tpg_wwn pointer following initialization in target_xcopy_setup_pt(). The NULL xcopy_pt_tpg->se_tpg_wwn pointer is dereferenced on completion of any EXTENDED COPY initiated read/write cmds. E.g using the libiscsi SCSI.ExtendedCopy.Simple test: BUG: kernel NULL pointer dereference, address: 00000000000001a8 RIP: 0010:target_complete_cmd+0x9d/0x130 [target_core_mod] Call Trace: fd_execute_rw+0x148/0x42a [target_core_file] ? __dynamic_pr_debug+0xa7/0xe0 ? target_check_reservation+0x5b/0x940 [target_core_mod] __target_execute_cmd+0x1e/0x90 [target_core_mod] transport_generic_new_cmd+0x17c/0x330 [target_core_mod] target_xcopy_issue_pt_cmd+0x9/0x60 [target_core_mod] target_xcopy_read_source.isra.7+0x10b/0x1b0 [target_core_mod] ? target_check_fua+0x40/0x40 [target_core_mod] ? transport_complete_task_attr+0x130/0x130 [target_core_mod] target_xcopy_do_work+0x61f/0xc00 [target_core_mod] This fix makes target_complete_cmd() queue work on se_cmd->cpuid if se_tpg_wwn is NULL.
CVE-2021-47293	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_skbmod: Skip non-Ethernet packets Currently tcf_skbmod_act() assumes that packets use Ethernet as their L2 protocol, which is not always the case. As an example, for CAN devices: \$ ip link add dev vcan0 type vcan \$ ip link set up vcan0 \$ tc qdisc add dev vcan0 root handle 1: htb \$ tc filter add dev vcan0 parent 1: protocol ip prio 10 \ matchall action skbmod swap mac Doing the above silently corrupts all the packets. Do not perform skbmod actions for non-Ethernet packets.
CVE-2021-47295	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix memory leak in tcindex_partial_destroy_work Syzbot reported memory leak in tcindex_set_parms(). The problem was in non-freed perfect hash in tcindex_partial_destroy_work(). In tcindex_set_parms() new tcindex_data is allocated and some fields from old one are copied to new one, but not the perfect hash. Since tcindex_partial_destroy_work() is the destroy function for old tcindex_data, we need to free perfect hash to avoid memory leak.
CVE-2021-47298	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix potential memory leak on unlikely error case If skb_linearize is needed and fails we could leak a msg on the error handling. To fix ensure we kfree the msg block before returning error. Found during code review.

CVE ID	Description
CVE-2021-47303	In the Linux kernel, the following vulnerability has been resolved: bpf: Track subprog poke descriptors correctly and fix use-after-free Subprograms are calling map_poke_track(), but on program release there is no hook to call map_poke_untrack(). However, on program release, the aux memory (and poke descriptor table) is freed even though we still have a reference to it in the element list of the map aux data. When we run map_poke_run(), we then end up accessing free'd memory, triggering KASAN in prog_array_map_poke_run(): [...] [402.824689] BUG: KASAN: use-after-free in prog_array_map_poke_run+0xc2/0x34e [402.824698] Read of size 4 at addr ffff8881905a7940 by task hubble-fgs/4337 [402.824705] CPU: 1 PID: 4337 Comm: hubble-fgs Tainted: G I 5.12.0+ #399 [402.824715] Call Trace: [402.824719] dump_stack+0x93/0xc2 [402.824727] print_address_description.constprop.0+0x1a/0x140 [402.824736] ? prog_array_map_poke_run+0xc2/0x34e [402.824740] ? prog_array_map_poke_run+0xc2/0x34e [402.824744] kasan_report.cold+0x7c/0xd8 [402.824752] ? prog_array_map_poke_run+0xc2/0x34e [402.824757] prog_array_map_poke_run+0xc2/0x34e [402.824765] bpf_fd_array_map_update_elem+0x124/0x1a0 [...] The elements concerned are walked as follows: for (i = 0; i < elem->aux->size_poke_tab; i++) { poke = &elem->aux->poke_tab[i]; [...] The access to size_poke_tab is a 4 byte read, verified by checking offsets in the KASAN dump: [402.825004] The buggy address belongs to the object at ffff8881905a7800 which belongs to the cache kmalloc-1k of size 1024 [402.825008] The buggy address is located 320 bytes inside of 1024-byte region [ffff8881905a7800, ffff8881905a7c00) The pahole output of bpf_prog_aux: struct bpf_prog_aux { [...] /* --- cacheline 5 boundary (320 bytes) --- */ u32 size_poke_tab; /* 320 4 */ [...] In general, subprograms do not necessarily manage their own data structures. For example, BTF func_info and linfo are just pointers to the main program structure. This allows reference counting and cleanup to be done on the latter which simplifies their management a bit. The aux->poke_tab struct, however, did not follow this logic. The initial proposed fix for this use-after-free bug further embedded poke data tracking into the subprogram with proper reference counting. However, Daniel and Alexei questioned why we were treating these objects special; I agree, its unnecessary. The fix here removes the per subprogram poke table allocation and map tracking and instead simply points the aux->poke_tab pointer at the main programs poke table. This way, map tracking is simplified to the main program and we do not need to manage them per subprogram. This also means, bpf_prog_free_deferred(), which unwinds the program reference counting and kfree's objects, needs to ensure that we don't try to double free the poke_tab when free'ing the subprog structures. This is easily solved by NULL'ing the poke_tab pointer. The second detail is to ensure that per subprogram JIT logic only does fixups on poke_tab[] entries it owns. To do this, we add a pointer in the poke structure to point at the subprogram value so JITs can easily check while walking the poke_tab structure if the current entry belongs to the current program. The aux pointer is stable and therefore suitable for such comparison. On the jit_subprogs() error path, we omit cleaning up the poke->aux field because these are only ever referenced from the JIT side, but on error we will never make it to the JIT, so its fine to leave them dangling. Removing these pointers would complicate the error path for no reason. However, we do need to untrack all poke descriptors from the main program as otherwise they could race with the freeing of JIT memory from the subprograms. Lastly, a748c6975dea3 ("bpf: propagate poke des ---truncated---
CVE-2021-47305	In the Linux kernel, the following vulnerability has been resolved: dma-buf/sync_file: Don't leak fences on merge failure Each add_fence() call does a dma_fence_get() on the relevant fence. In the error path, we weren't calling dma_fence_put() so all those fences got leaked. Also, in the krealloc_array failure case, we weren't freeing the fences array. Instead, ensure that i and fences are always zero-initialized and dma_fence_put() all the fences and kfree(fences) on every error path.
CVE-2021-47307	In the Linux kernel, the following vulnerability has been resolved: cifs: prevent NULL deref in cifs_compose_mount_options() The optional @ref parameter might contain a NULL node_name, so prevent dereferencing it in cifs_compose_mount_options(). Addresses-Coverity: 1476408 ("Explicit null dereferenced")
CVE-2021-47308	In the Linux kernel, the following vulnerability has been resolved: scsi: libfc: Fix array index out of bound exception Fix array index out of bound exception in fc_rport_prli_resp().
CVE-2021-47313	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Fix potential memleak in cppc_cpufreq_cpu_init It's a classic example of memleak, we allocate something, we fail and never free the resources. Make sure we free all resources on policy ->init() failures.
CVE-2021-47316	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix NULL dereference in nfs3svc_encode_getaclres In error cases the dentry may be NULL. Before 20798dfe249a, the encoder also checked dentry and d_really_is_positive(dentry), but that looks like overkill to me--zero status should be enough to guarantee a positive dentry. This isn't the first time we've seen an error-case NULL dereference hidden in the initialization of a local variable in an xdr encoder. But I went back through the other recent rewrites and didn't spot any similar bugs.

CVE ID	Description
CVE-2021-47317	In the Linux kernel, the following vulnerability has been resolved: powerpc/bpf: Fix detecting BPF atomic instructions Commit 91c960b0056672 ("bpf: Rename BPF_XADD and prepare to encode other atomics in .imm") converted BPF_XADD to BPF_ATOMIC and added a way to distinguish instructions based on the immediate field. Existing JIT implementations were updated to check for the immediate field and to reject programs utilizing anything more than BPF_ADD (such as BPF_FETCH) in the immediate field. However, the check added to powerpc64 JIT did not look at the correct BPF instruction. Due to this, such programs would be accepted and incorrectly JIT'ed resulting in soft lockups, as seen with the atomic bounds test. Fix this by looking at the correct immediate value.
CVE-2021-47319	In the Linux kernel, the following vulnerability has been resolved: virtio-blk: Fix memory leak among suspend/resume procedure The vblk->vqs should be freed before we call init_vqs() in virtblk_restore().
CVE-2021-47320	In the Linux kernel, the following vulnerability has been resolved: nfs: fix acl memory leak of posix_acl_create() When looking into another nfs xfstests report, I found acl and default_acl in nfs3_proc_create() and nfs3_proc_mknod() error paths are possibly leaked. Fix them in advance.
CVE-2021-47325	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Fix arm_smmu_device refcount leak in address translation The reference counting issue happens in several exception handling paths of arm_smmu_iova_to_phys_hard(). When those error scenarios occur, the function forgets to decrease the refcount of "smmu" increased by arm_smmu_rpm_get(), causing a refcount leak. Fix this issue by jumping to "out" label when those error scenarios occur.
CVE-2021-47327	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Fix arm_smmu_device refcount leak when arm_smmu_rpm_get fails arm_smmu_rpm_get() invokes pm_runtime_get_sync(), which increases the refcount of the "smmu" even though the return value is less than 0. The reference counting issue happens in some error handling paths of arm_smmu_rpm_get() in its caller functions. When arm_smmu_rpm_get() fails, the caller functions forget to decrease the refcount of "smmu" increased by arm_smmu_rpm_get(), causing a refcount leak. Fix this issue by calling pm_runtime_resume_and_get() instead of pm_runtime_get_sync() in arm_smmu_rpm_get(), which can keep the refcount balanced in case of failure.
CVE-2021-47332	In the Linux kernel, the following vulnerability has been resolved: ALSA: usx2y: Don't call free_pages_exact() with NULL address Unlike some other functions, we can't pass NULL pointer to free_pages_exact(). Add a proper NULL check for avoiding possible Oops.
CVE-2021-47333	In the Linux kernel, the following vulnerability has been resolved: misc: alcor_pci: fix null-ptr-deref when there is no PCI bridge There is an issue with the ASPM(optional) capability checking function. A device might be attached to root complex directly, in this case, bus->self(bridge) will be NULL, thus priv->parent_pdev is NULL. Since alcor_pci_init_check_aspm(priv->parent_pdev) checks the PCI link's ASPM capability and populate parent_cap_off, which will be used later by alcor_pci_aspm_ctrl() to dynamically turn on/off device, what we can do here is to avoid checking the capability if we are on the root complex. This will make pdev_cap_off 0 and alcor_pci_aspm_ctrl() will simply return when bring called, effectively disable ASPM for the device. [1.246492] BUG: kernel NULL pointer dereference, address: 00000000000000c0 [1.248731] RIP: 0010:pci_read_config_byte+0x5/0x40 [1.253998] Call Trace: [1.254131] ? alcor_pci_find_cap_offset.isra.0+0x3a/0x100 [alcor_pci] [1.254476] alcor_pci_probe+0x169/0x2d5 [alcor_pci]
CVE-2021-47339	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-core: explicitly clear ioctl input data As seen from a recent syzbot bug report, mistakes in the compat ioctl implementation can lead to uninitialized kernel stack data getting used as input for driver ioctl handlers. The reported bug is now fixed, but it's possible that other related bugs are still present or get added in the future. As the drivers need to check user input already, the possible impact is fairly low, but it might still cause an information leak. To be on the safe side, always clear the entire ioctl buffer before calling the conversion handler functions that are meant to initialize them.
CVE-2021-47342	In the Linux kernel, the following vulnerability has been resolved: ext4: fix possible UAF when remounting r/o a mmp-protected file system After commit 618f003199c6 ("ext4: fix memory leak in ext4_fill_super"), after the file system is remounted read-only, there is a race where the kmmpd thread can exit, causing sbi->s_mmp_tsk to point at freed memory, which the call to ext4_stop_mmpd() can trip over. Fix this by only allowing kmmpd() to exit when it is stopped via ext4_stop_mmpd(). Bug-Report-Link: <20210629143603.2166962-1-yebin10@huawei.com>

CVE ID	Description
CVE-2021-47343	In the Linux kernel, the following vulnerability has been resolved: dm btree remove: assign new_root only when removal succeeds remove_raw() in dm_btree_remove() may fail due to IO read error (e.g. read the content of origin block fails during shadowing), and the value of shadow_spine::root is uninitialized, but the uninitialized value is still assign to new_root in the end of dm_btree_remove(). For dm-thin, the value of pmd->details_root or pmd->root will become an uninitialized value, so if trying to read details_info tree again out-of-bound memory may occur as showed below: general protection fault, probably for non-canonical address 0x3fdb14c8d7520 CPU: 4 PID: 515 Comm: dmsetup Not tainted 5.13.0-rc6 Hardware name: QEMU Standard PC RIP: 0010:metadata_ll_load_ie+0x14/0x30 Call Trace: sm_metadata_count_is_more_than_one+0xb9/0xe0 dm_tm_shadow_block+0x52/0x1c0 shadow_step+0x59/0xf0 remove_raw+0xb2/0x170 dm_btree_remove+0xf4/0x1c0 dm_pool_delete_thin_device+0xc3/0x140 pool_message+0x218/0x2b0 target_message+0x251/0x290 ctl_ioctl+0x1c4/0x4d0 dm_ctl_ioctl+0xe/0x20 __x64_sys_ioctl+0x7b/0xb0 do_syscall_64+0x40/0xb0 entry_SYSCALL_64_after_hwframe+0x44/0xae Fixing it by only assign new_root when removal succeeds
CVE-2021-47349	In the Linux kernel, the following vulnerability has been resolved: mwifiex: bring down link before deleting interface We can deadlock when rmmod'ing the driver or going through firmware reset, because the cfg80211_unregister_wdev() has to bring down the link for us, ... which then grab the same wiphy lock. nl80211_del_interface() already handles a very similar case, with a nice description: / * * We hold RTNL, so this is safe, without RTNL opencount cannot * reach 0, and thus the rdev cannot be deleted. * * We need to do it for the dev_close(), since that will call * the netdev notifiers, and we need to acquire the mutex there * but don't know if we get there from here or from some other * place (e.g. "ip link set ... down"). */ mutex_unlock(&rdev->wiphy.mtx); ... Do similarly for mwifiex teardown, by ensuring we bring the link down first. Sample deadlock trace: [247.103516] INFO: task rmmod:2119 blocked for more than 123 seconds. [247.110630] Not tainted 5.12.4 #5 [247.115796] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [247.124557] task:rmmod state:D stack: 0 pid: 2119 ppid: 2114 flags:0x00400208 [247.133905] Call trace: [247.136644] __switch_to+0x130/0x170 [247.140643] __schedule+0x714/0xa0c [247.144548] schedule_preempt_disabled+0x88/0xf4 [247.149714] __mutex_lock_common+0x43c/0x750 [247.154496] mutex_lock_nested+0x5c/0x68 [247.158884] cfg80211_netdev_notifier_call+0x280/0x4e0 [cfg80211] [247.165769] raw_notifier_call_chain+0x4c/0x78 [247.170742] call_netdevice_notifiers_info+0x68/0xa4 [247.176305] __dev_close_many+0x7c/0x138 [247.180693] dev_close_many+0x7c/0x10c [247.184893] unregister_netdevice_many+0xfc/0x654 [247.190158] unregister_netdevice_queue+0xb4/0xe0 [247.195424] _cfg80211_unregister_wdev+0xa4/0x204 [cfg80211] [247.201816] cfg80211_unregister_wdev+0x20/0x2c [cfg80211] [247.208016] mwifiex_del_virtual_intf+0xc8/0x188 [mwifiex] [247.214174] mwifiex_uninit_sw+0x158/0x1b0 [mwifiex] [247.219747] mwifiex_remove_card+0x38/0xa0 [mwifiex] [247.225316] mwifiex_pcie_remove+0xd0/0xe0 [mwifiex_pcie] [247.231451] pci_device_remove+0x50/0xe0 [247.235849] device_release_driver_internal+0x110/0x1b0 [247.241701] driver_detach+0x5c/0x9c [247.245704] bus_remove_driver+0x84/0xb8 [247.250095] driver_unregister+0x3c/0x60 [247.254486] pci_unregister_driver+0x2c/0x90 [247.259267] cleanup_module+0x18/0xcdc [mwifiex_pcie]
CVE-2021-47350	In the Linux kernel, the following vulnerability has been resolved: powerpc/mm: Fix lockup on kernel exec fault The powerpc kernel is not prepared to handle exec faults from kernel. Especially, the function is_exec_fault() will return 'false' when an exec fault is taken by kernel, because the check is based on reading current->thread.regs->trap which contains the trap from user. For instance, when provoking a LKDTM EXEC_USERSPACE test, current->thread.regs->trap is set to SYSCALL trap (0xc00), and the fault taken by the kernel is not seen as an exec fault by set_access_flags_filter(). Commit d7df2443cd5f ("powerpc/mm: Fix spurious segfaults on radix with autonuma") made it clear and handled it properly. But later on commit d3ca587404b3 ("powerpc/mm: Fix reporting of kernel execute faults") removed that handling, introducing test based on error_code. And here is the problem, because on the 603 all upper bits of SRR1 get cleared when the TLB instruction miss handler bails out to ISI. Until commit cbd7e6ca0210 ("powerpc/fault: Avoid heavy search_exception_tables() verification"), an exec fault from kernel at a userspace address was indirectly caught by the lack of entry for that address in the exception tables. But after that commit the kernel mainly relies on KUAP or on core mm handling to catch wrong user accesses. Here the access is not wrong, so mm handles it. It is a minor fault because PAGE_EXEC is not set, set_access_flags_filter() should set PAGE_EXEC and voila. But as is_exec_fault() returns false as explained in the beginning, set_access_flags_filter() bails out without setting PAGE_EXEC flag, which leads to a forever minor exec fault. As the kernel is not prepared to handle such exec faults, the thing to do is to fire in bad_kernel_fault() for any exec fault taken by the kernel, as it was prior to commit d3ca587404b3.
CVE-2021-47359	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix soft lockup during fsstress Below traces are observed during fsstress and system got hung. [130.698396] watchdog: BUG: soft lockup - CPU#6 stuck for 26s!

CVE ID	Description
CVE-2021-47360	In the Linux kernel, the following vulnerability has been resolved: binder: make sure fd closes complete During BC_FREE_BUFFER processing, the BINDER_TYPE_FDA object cleanup may close 1 or more fds. The close operations are completed using the task work mechanism -- which means the thread needs to return to userspace or the file object may never be dereferenced -- which can lead to hung processes. Force the binder thread back to userspace if an fd is closed during BC_FREE_BUFFER handling.
CVE-2021-47367	In the Linux kernel, the following vulnerability has been resolved: virtio-net: fix pages leaking when building skb in big mode We try to use build_skb() if we had sufficient tailroom. But we forget to release the unused pages chained via private in big mode which will leak pages. Fixing this by release the pages after building the skb in big mode.
CVE-2021-47370	In the Linux kernel, the following vulnerability has been resolved: mptcp: ensure tx skbs always have the MPTCP ext Due to signed/unsigned comparison, the expression: info->size_goal - skb->len > 0 evaluates to true when the size goal is smaller than the skb size. That results in lack of tx cache refill, so that the skb allocated by the core TCP code lacks the required MPTCP skb extensions. Due to the above, syzbot is able to trigger the following WARN_ON(): WARNING: CPU: 1 PID: 810 at net/mptcp/protocol.c:1366 mptcp_sendmsg_frag+0x1362/0x1bc0 net/mptcp/protocol.c:1366 Modules linked in: CPU: 1 PID: 810 Comm: syz-executor.4 Not tainted 5.14.0-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:mptcp_sendmsg_frag+0x1362/0x1bc0 net/mptcp/protocol.c:1366 Code: ff 4c 8b 74 24 50 48 8b 5c 24 58 e9 0f fb ff ff e8 13 44 8b f8 4c 89 e7 45 31 ed e8 98 57 2e fe e9 81 f4 ff ff e8 fe 43 8b f8 <0f> 0b 41 bd ea ff ff e9 6f f4 ff ff 4c 89 e7 e8 b9 8e d2 f8 e9 RSP: 0018:ffffc9000531f6a0 EFLAGS: 00010216 RAX: 0000000000000697f RBX: 0000000000000000 RCX: ffff90012107000 RDX: 0000000000040000 RSI: ffffffff88eac9e2 RDI: 0000000000000003 RBP: ffff888078b15780 R08: 0000000000000000 R09: 0000000000000000 R10: ffffffff88eac017 R11: 0000000000000000 R12: ffff88801de0a280 R13: 00000000000006b58 R14: ffff888066278280 R15: ffff88803c2fe9c0 FS: 00007fd9f866e700(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007faebcb2f718 CR3: 00000000267cb000 CR4: 0000000001506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: __mptcp_push_pending+0x1fb/0x6b0 net/mptcp/protocol.c:1547 mptcp_release_cb+0xfe/0x210 net/mptcp/protocol.c:3003 release_sock+0xb4/0x1b0 net/core/sock.c:3206 sk_stream_wait_memory+0x604/0xed0 net/core/stream.c:145 mptcp_sendmsg+0xc39/0x1bc0 net/mptcp/protocol.c:1749 inet6_sendmsg+0x99/0xe0 net/ipv6/af_inet6.c:643 sock_sendmsg_nosec net/socket.c:704 [inline] sock_sendmsg+0xcfc/0x120 net/socket.c:724 sock_write_iter+0x2a0/0x3e0 net/socket.c:1057 call_write_iter include/linux/fs.h:2163 [inline] new_sync_write+0x40b/0x640 fs/read_write.c:507 vfs_write+0x7cf/0xae0 fs/read_write.c:594 ksys_write+0x1ee/0x250 fs/read_write.c:647 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x4665f9 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fd9f866e188 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 RAX: ffffffff88eac9e2 RBX: 000000000056c038 RCX: 00000000004665f9 RDX: 00000000000e7b78 RSI: 0000000020000000 RDI: 0000000000000003 RBP: 000000000004bfcc4 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 000000000056c038 R13: 0000000000a9fb1f R14: 00007fd9f866e300 R15: 0000000000022000 Fix the issue rewriting the relevant expression to avoid sign-related problems - note: size_goal is always >= 0. Additionally, ensure that the skb in the tx cache always carries the relevant extension.
CVE-2021-47374	In the Linux kernel, the following vulnerability has been resolved: dma-debug: prevent an error message from causing runtime problems For some drivers, that use the DMA API. This error message can be reached several millions of times per second, causing spam to the kernel's printk buffer and bringing the CPU usage up to 100% (so, it should be rate limited). However, since there is at least one driver that is in the mainline and suffers from the error condition, it is more useful to err_printk() here instead of just rate limiting the error message (in hopes that it will make it easier for other drivers that suffer from this issue to be spotted).

CVE ID	Description
CVE-2021-47376	In the Linux kernel, the following vulnerability has been resolved: bpf: Add oversize check before call kvccalloc() Commit 7661809d493b ("mm: don't allow oversized kvmalloc() calls") add the oversize check. When the allocation is larger than what kmalloc() supports, the following warning triggered: WARNING: CPU: 0 PID: 8408 at mm/util.c:597 kvmalloc_node+0x108/0x110 mm/util.c:597 Modules linked in: CPU: 0 PID: 8408 Comm: syz-executor221 Not tainted 5.14.0-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:kvmalloc_node+0x108/0x110 mm/util.c:597 Call Trace: kvmalloc include/linux/mm.h:806 [inline] kvmalloc_array include/linux/mm.h:824 [inline] kvccalloc include/linux/mm.h:829 [inline] check_btf_line kernel/bpf/verifier.c:9925 [inline] check_btf_info kernel/bpf/verifier.c:10049 [inline] bpf_check+0xd634/0x150d0 kernel/bpf/verifier.c:13759 bpf_prog_load kernel/bpf/syscall.c:2301 [inline] __sys_bpf+0x11181/0x126e0 kernel/bpf/syscall.c:4587 __do_sys_bpf kernel/bpf/syscall.c:4691 [inline] __se_sys_bpf kernel/bpf/syscall.c:4689 [inline] __x64_sys_bpf+0x78/0x90 kernel/bpf/syscall.c:4689 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2021-47380	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: Fix potential NULL pointer dereference devm_add_action_or_reset() can suddenly invoke amd_mp2_pci_remove() at registration that will cause NULL pointer dereference since corresponding data is not initialized yet. The patch moves initialization of data before devm_add_action_or_reset(). Found by Linux Driver Verification project (linuxtesting.org). [jkosina@suse.cz: rebase]
CVE-2021-47381	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Fix DSP oops stack dump output contents Fix @buf arg given to hex_dump_to_buffer() and stack address used in dump error output.
CVE-2021-47387	In the Linux kernel, the following vulnerability has been resolved: cpufreq: schedutil: Use kobject release() method to free sugov_tunables The struct sugov_tunables is protected by the kobject, so we can't free it directly. Otherwise we would get a call trace like this: ODEBUG: free active (active state 0) object type: timer_list hint: delayed_work_timer_fn+0x0/0x30 WARNING: CPU: 3 PID: 720 at lib/debugobjects.c:505 debug_print_object+0xb8/0x100 Modules linked in: CPU: 3 PID: 720 Comm: a.sh Tainted: G W 5.14.0-rc1-next-20210715-yocto-standard+ #507 Hardware name: Marvell OcteonTX CN96XX board (DT) pstate: 40400009 (nZcv daif +PAN -UAO -TCO BTYPE=) pc : debug_print_object+0xb8/0x100 lr : debug_print_object+0xb8/0x100 sp : ffff80001ecaf910 x29: ffff80001ecaf910 x28: ffff00011b10b8d0 x27: ffff800011043d80 x26: ffff00011a8f0000 x25: ffff800013cb3ff0 x24: 0000000000000000 x23: ffff80001142aa68 x22: ffff800011043d80 x21: ffff00010de46f20 x20: ffff800013c0c520 x19: ffff800011d8f5b0 x18: 0000000000000010 x17: 6e6968207473696c x16: 5f72656d6974203a x15: 6570797420746365 x14: 6a626f2029302065 x13: 303378302f307830 x12: 2b6e665f72656d69 x11: ffff8000124b1560 x10: ffff800012331520 x9: ffff8000100ca6b0 x8: 000000000017ffe8 x7: c0000000ffffffffff x6: 0000000000000001 x5: ffff800011d8c000 x4: ffff800011d8c740 x3: 0000000000000000 x2: ffff0001108301c0 x1: ab3c90eedf9c0f00 x0: 0000000000000000 Call trace: debug_print_object+0xb8/0x100 __debug_check_no_obj_freed+0x1c0/0x230 debug_check_no_obj_freed+0x20/0x88 slab_free_freelist_hook+0x154/0x1c8 kfree+0x114/0x5d0 sugov_exit+0xbc/0xc0 cpufreq_exit_governor+0x44/0x90 cpufreq_set_policy+0x268/0x4a8 store_scaling_governor+0xe0/0x128 store+0xc0/0xf0 sysfs_kf_write+0x54/0x80 kernfs_fop_write_iter+0x128/0x1c0 new_sync_write+0xf0/0x190 vfs_write+0x2d4/0x478 ksys_write+0x74/0x100 __arm64_sys_write+0x24/0x30 invoke_syscall.constprop.0+0x54/0xe0 do_el0_svc+0x64/0x158 el0_svc+0x2c/0xb0 el0t_64_sync_handler+0xb0/0xb8 el0t_64_sync+0x198/0x19c irq event stamp: 5518 hardirqs last enabled at (5517): [<ffff8000100cbd7c>] console_unlock+0x554/0x6c8 hardirqs last disabled at (5518): [<ffff800010fc0638>] el1_dbg+0x28/0xa0 softirqs last enabled at (5504): [<ffff8000100106e0>] __do_softirq+0x4d0/0x6c0 softirqs last disabled at (5483): [<ffff800010049548>] irq_exit+0x1b0/0x1b8 So split the original sugov_tunables_free() into two functions, sugov_clear_global_tunables() is just used to clear the global_tunables and the new sugov_tunables_free() is used as kobj_type::release to release the sugov_tunables safely.
CVE-2021-47389	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: fix missing sev_decommission in sev_receive_start DECOMMISSION the current SEV context if binding an ASID fails after RECEIVE_START. Per AMD's SEV API, RECEIVE_START generates a new guest context and thus needs to be paired with DECOMMISSION: The RECEIVE_START command is the only command other than the LAUNCH_START command that generates a new guest context and guest handle. The missing DECOMMISSION can result in subsequent SEV launch failures, as the firmware leaks memory and might not be able to allocate more SEV guest contexts in the future. Note, LAUNCH_START suffered the same bug, but was previously fixed by commit 934002cd660b ("KVM: SVM: Call SEV Guest Decommission if ASID binding fails").

CVE ID	Description
CVE-2021-47395	In the Linux kernel, the following vulnerability has been resolved: mac80211: limit injected vht mcs/nss in ieee80211_parse_tx_radiotap Limit max values for vht mcs and nss in ieee80211_parse_tx_radiotap routine in order to fix the following warning reported by syzbot: WARNING: CPU: 0 PID: 10717 at include/net/mac80211.h:989 ieee80211_rate_set_vht include/net/mac80211.h:989 [inline] WARNING: CPU: 0 PID: 10717 at include/net/mac80211.h:989 ieee80211_parse_tx_radiotap +0x101e/0x12d0 net/mac80211/tx.c:2244 Modules linked in: CPU: 0 PID: 10717 Comm: syz-executor.5 Not tainted 5.14.0-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:ieee80211_rate_set_vht include/net/mac80211.h:989 [inline] RIP: 0010:ieee80211_parse_tx_radiotap+0x101e/0x12d0 net/mac80211/tx.c:2244 RSP: 0018:ffffc9000186f3e8 EFLAGS: 00010216 RAX: 0000000000000618 RBX: ffff88804ef76500 RCX: ffff900143a5000 RDX: 0000000000040000 RSI: ffffffff888f478e RDI: 0000000000000003 RBP: 00000000ffffff R08: 0000000000000000 R09: 0000000000000100 R10: ffffffff888f46f9 R11: 0000000000000000 R12: 00000000ffffff R13: ffff88804ef7653c R14: 0000000000000001 R15: 0000000000000004 FS: 00007fbf5718f700(0000) GS:ffff8880b9c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000001b2de23000 CR3: 000000006a671000 CR4: 00000000001506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000600 Call Trace: ieee80211_monitor_select_queue +0xa6/0x250 net/mac80211/iface.c:740 netdev_core_pick_tx+0x169/0x2e0 net/core/dev.c:4089 __dev_queue_xmit+0x6f9/0x3710 net/core/dev.c:4165 __bpf_tx_skb net/core/filter.c:2114 [inline] __bpf_redirect_no_mac net/core/filter.c:2139 [inline] __bpf_redirect+0x5ba/0xd20 net/core/filter.c:2162 __bpf_clone_redirect net/core/filter.c:2429 [inline] bpf_clone_redirect+0x2ae/0x420 net/core/filter.c:2401 bpf_prog_eeb6f53a69e5c6a2+0x59/0x234 bpf_dispatcher_nop_func include/linux/bpf.h:717 [inline] __bpf_prog_run include/linux/filter.h:624 [inline] bpf_prog_run include/linux/filter.h:631 [inline] bpf_test_run+0x381/0xa30 net/bpf/test_run.c:119 bpf_prog_test_run_skb +0xb84/0x1ee0 net/bpf/test_run.c:663 bpf_prog_test_run kernel/bpf/syscall.c:3307 [inline] __sys_bpf +0x2137/0x5df0 kernel/bpf/syscall.c:4605 __do_sys_bpf kernel/bpf/syscall.c:4691 [inline] __se_sys_bpf kernel/bpf/syscall.c:4689 [inline] __x64_sys_bpf+0x75/0xb0 kernel/bpf/syscall.c:4689 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x4665f9
CVE-2021-47397	In the Linux kernel, the following vulnerability has been resolved: sctp: break out if skb_header_pointer returns NULL in sctp_rcv_oob We should always check if skb_header_pointer's return is NULL before using it, otherwise it may cause null-ptr-deref, as syzbot reported: KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] RIP: 0010:sctp_rcv_oob net/sctp/input.c:705 [inline] RIP: 0010:sctp_rcv+0x1d84/0x3220 net/sctp/input.c:196 Call Trace: <IRQ> sctp6_rcv+0x38/0x60 net/sctp/ipv6.c:1109 ip6_protocol_deliver_rcu+0x2e9/0x1ca0 net/ipv6/ipv6_input.c:422 ip6_input_finish +0x62/0x170 net/ipv6/ipv6_input.c:463 NF_HOOK include/linux/netfilter.h:307 [inline] NF_HOOK include/linux/netfilter.h:301 [inline] ip6_input+0x9c/0xd0 net/ipv6/ipv6_input.c:472 dst_input include/net/dst.h:460 [inline] ip6_rcv_finish net/ipv6/ipv6_input.c:76 [inline] NF_HOOK include/linux/netfilter.h:307 [inline] NF_HOOK include/linux/netfilter.h:301 [inline] ipv6_rcv+0x28c/0x3c0 net/ipv6/ipv6_input.c:297
CVE-2021-47398	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Fix kernel pointer leak Pointers should be printed with %p or %px rather than cast to 'unsigned long long' and printed with %llx. Change %llx to %p to print the secured pointer.
CVE-2021-47404	In the Linux kernel, the following vulnerability has been resolved: HID: betop: fix slab-out-of-bounds Write in betop_probe Syzbot reported slab-out-of-bounds Write bug in hid-betopff driver. The problem is the driver assumes the device must have an input report but some malicious devices violate this assumption. So this patch checks hid_device's input is non empty before it's been used.
CVE-2021-47407	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Handle SRCU initialization failure during page track init Check the return of init_srcu_struct(), which can fail due to OOM, when initializing the page track mechanism. Lack of checking leads to a NULL pointer deref found by a modified syzkaller. [Move the call towards the beginning of kvm_arch_init_vm. - Paolo]

CVE ID	Description
CVE-2021-47413	In the Linux kernel, the following vulnerability has been resolved: usb: chipidea: ci_hdrc_imx: Also search for 'phys' phandle When passing 'phys' in the devicetree to describe the USB PHY phandle (which is the recommended way according to Documentation/devicetree/bindings/usb/ci-hdrc-usb2.txt) the following NULL pointer dereference is observed on i.MX7 and i.MX8MM: [1.489344] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000098 [1.498170] Mem abort info: [1.500966] ESR = 0x96000044 [1.504030] EC = 0x25: DABT (current EL), IL = 32 bits [1.509356] SET = 0, FnV = 0 [1.512416] EA = 0, S1PTW = 0 [1.515569] FSC = 0x04: level 0 translation fault [1.520458] Data abort info: [1.523349] ISV = 0, ISS = 0x00000044 [1.527196] CM = 0, WnR = 1 [1.530176] [0000000000000098] user address but active_mm is swapper [1.536544] Internal error: Oops: 96000044 [#1] PREEMPT SMP [1.542125] Modules linked in: [1.545190] CPU: 3 PID: 7 Comm: kworker/u8:0 Not tainted 5.14.0-dirty #3 [1.551901] Hardware name: Kontron i.MX8MM N801X S (DT) [1.557133] Workqueue: events_unbound deferred_probe_work_func [1.562984] pstate: 80000005 (Nzcv daif -PAN -UAO -TCO BTYP=---) [1.568998] pc : imx7d_charger_detection+0x3f0/0x510 [1.573973] lr : imx7d_charger_detection+0x22c/0x510 This happens because the charger functions check for the phy presence inside the imx_usbmisc_data structure (data->usb_phy), but the chipidea core populates the usb_phy passed via 'phys' inside 'struct ci_hdrc' (ci->usb_phy) instead. This causes the NULL pointer dereference inside imx7d_charger_detection(). Fix it by also searching for 'phys' in case 'fsl,usbphy' is not found. Tested on a imx7s-warp board.
CVE-2021-47415	In the Linux kernel, the following vulnerability has been resolved: iwlwifi: mvm: Fix possible NULL dereference In __iwl_mvm_remove_time_event() check that 'te_data->vif' is NULL before dereferencing it.
CVE-2021-47416	In the Linux kernel, the following vulnerability has been resolved: phy: mdio: fix memory leak Syzbot reported memory leak in MDIO bus interface, the problem was in wrong state logic. MDIOBUS_ALLOCATED indicates 2 states: 1. Bus is only allocated 2. Bus allocated and __mdibus_register() fails, but device_register() was called In case of device_register() has been called we should call put_device() to correctly free the memory allocated for this device, but mdibus_free() calls just kfree(dev) in case of MDIOBUS_ALLOCATED state To avoid this behaviour we need to set bus->state to MDIOBUS_UNREGISTERED_before_ calling device_register(), because put_device() should be called even in case of device_register() failure.
CVE-2021-47419	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_taprio: properly cancel timer from taprio_destroy() There is a comment in qdisc_create() about us not calling ops->reset() in some cases. err_out4: /* * Any broken qdiscs that would require a ops->reset() here? * The qdisc was never in action so it shouldn't be necessary. */ As taprio sets a timer before actually receiving a packet, we need to cancel it from ops->destroy, just in case ops->reset has not been called. syzbot reported: ODEBAG: free active (active state 0) object type: hrtimer hint: advance_sched+0x0/0x9a0 arch/x86/include/asm/atomic64_64.h:22 WARNING: CPU: 0 PID: 8441 at lib/debugobjects.c:505 debug_print_object+0x16e/0x250 lib/debugobjects.c:505 Modules linked in: CPU: 0 PID: 8441 Comm: syz-executor813 Not tainted 5.14.0-rc6-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:debug_print_object+0x16e/0x250 lib/debugobjects.c:505 Code: ff df 48 89 fa 48 c1 ea 03 80 3c 02 00 0f 85 af 00 00 00 48 8b 14 dd e0 d3 e3 89 4c 89 ee 48 c7 c7 e0 c7 e3 89 e8 5b 86 11 05 <0f> 0b 83 05 85 03 92 09 01 48 83 c4 18 5b 5d 41 5c 41 5d 41 5e c3 RSP: 0018:ffff9000130f330 EFLAGS: 00010282 RAX: 0000000000000000 RBX: 0000000000000003 RCX: 0000000000000000 RDX: ffff88802baeb880 RSI: ffffffff815d87b5 RDI: fffff52000261e58 RBP: 0000000000000001 R08: 0000000000000000 R09: 0000000000000000 R10: ffffffff815d25ee R11: 0000000000000000 R12: ffffffff898dd020 R13: ffffffff89e3ce20 R14: ffffffff81653630 R15: dffffc0000000000 FS: 0000000000f0d300(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffb64b3e000 CR3: 0000000036557000 CR4: 00000000001506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: __debug_check_no_obj_freed lib/debugobjects.c:987 [inline] debug_check_no_obj_freed+0x301/0x420 lib/debugobjects.c:1018 slab_free_hook mm/slab.c:1603 [inline] slab_free_freelist_hook+0x171/0x240 mm/slab.c:1653 slab_free mm/slab.c:3213 [inline] kfree+0xe4/0x540 mm/slab.c:4267 qdisc_create+0xbcf/0x1320 net/sched/sch_api.c:1299 tc_modify_qdisc+0x4c8/0x1a60 net/sched/sch_api.c:1663 rtnetlink_rcv_msg+0x413/0xb80 net/core/rtnetlink.c:5571 netlink_rcv_skb+0x153/0x420 net/netlink/af_netlink.c:2504 netlink_unicast_kernel net/netlink/af_netlink.c:1314 [inline] netlink_unicast+0x533/0x7d0 net/netlink/af_netlink.c:1340 netlink_sendmsg+0x86d/0xdb0 net/netlink/af_netlink.c:1929 sock_sendmsg_nosec net/socket.c:704 [inline] sock_sendmsg+0xcf/0x120 net/socket.c:724 __sys_sendmsg+0x6e8/0x810 net/socket.c:2403 __sys_sendmsg+0xf3/0x170 net/socket.c:2457 __sys_sendmsg+0xe5/0x1b0 net/socket.c:2486 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80
CVE-2021-47420	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: fix a potential ttm->sg memory leak Memory is allocated for ttm->sg by kmalloc in kfd_mem_dmamap_userptr, but isn't freed by kfree in kfd_mem_dmaunmap_userptr. Free it!

CVE ID	Description
CVE-2021-47421	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: handle the case of pci_channel_io_frozen only in amdgpu_pci_resume In current code, when a PCI error state pci_channel_io_normal is detected, it will report PCI_ERS_RESULT_CAN_RECOVER status to PCI driver, and PCI driver will continue the execution of PCI resume callback report_resume by pci_walk_bridge, and the callback will go into amdgpu_pci_resume finally, where write lock is released unconditionally without acquiring such lock first. In this case, a deadlock will happen when other threads start to acquire the read lock. To fix this, add a member in amdgpu_device struct to cache pci_channel_state, and only continue the execution in amdgpu_pci_resume when it's pci_channel_io_frozen.
CVE-2021-47424	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix freeing of uninitialized misc IRQ vector When VSI set up failed in i40e_probe() as part of PF switch set up driver was trying to free misc IRQ vectors in i40e_clear_interrupt_scheme and produced a kernel Oops: Trying to free already-free IRQ 266 WARNING: CPU: 0 PID: 5 at kernel/irq/manage.c:1731 __free_irq+0x9a/0x300 Workqueue: events work_for_cpu_fn RIP: 0010: __free_irq+0x9a/0x300 Call Trace: ? synchronize_irq+0x3a/0xa0 free_irq+0x2e/0x60 i40e_clear_interrupt_scheme+0x53/0x190 [i40e] i40e_probe.part.108+0x134b/0x1a40 [i40e] ? kmem_cache_alloc+0x158/0x1c0 ? acpi_ut_update_ref_count.part.1+0x8e/0x345 ? acpi_ut_update_object_reference+0x15e/0x1e2 ? strstr+0x21/0x70 ? irq_get_irq_data+0xa/0x20 ? mp_check_pin_attr+0x13/0xc0 ? irq_get_irq_data+0xa/0x20 ? mp_map_pin_to_irq+0xd3/0x2f0 ? acpi_register_gsi_ioapic+0x93/0x170 ? pci_conf1_read+0xa4/0x100 ? pci_bus_read_config_word+0x49/0x70 ? do_pci_enable_device+0xcc/0x100 local_pci_probe+0x41/0x90 work_for_cpu_fn+0x16/0x20 process_one_work+0x1a7/0x360 worker_thread+0x1cf/0x390 ? create_worker+0x1a0/0x1a0 kthread+0x112/0x130 ? kthread_flush_work_fn+0x10/0x10 ret_from_fork+0x1f/0x40 The problem is that at that point misc IRQ vectors were not allocated yet and we get a call trace that driver is trying to free already free IRQ vectors. Add a check in i40e_clear_interrupt_scheme for __I40E_MISC_IRQ_REQUESTED PF state before calling i40e_free_misc_vector. This state is set only if misc IRQ vectors were properly initialized.
CVE-2021-47426	In the Linux kernel, the following vulnerability has been resolved: bpf, s390: Fix potential memory leak about jit_data Make sure to free jit_data through kfree() in the error path.
CVE-2021-47427	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi: Fix iscsi_task use after free Commit d39df158518c ("scsi: iscsi: Have abort handler get ref to conn") added iscsi_get_conn()/iscsi_put_conn() calls during abort handling but then also changed the handling of the case where we detect an already completed task where we now end up doing a goto to the common put/cleanup code. This results in a iscsi_task use after free, because the common cleanup code will do a put on the iscsi_task. This reverts the goto and moves the iscsi_get_conn() to after we've checked if the iscsi_task is valid.
CVE-2021-47431	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix gart.bo pin_count leak gmc_v{9,10}_0_gart_disable() isn't called matched with corresponding gart_enable function in SRIOV case. This will lead to gart.bo pin_count leak on driver unload.
CVE-2021-47438	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix memory leak in mlx5_core_destroy_cq() error path Prior to this patch in case mlx5_core_destroy_cq() failed it returns without completing all destroy operations and that leads to memory leak. Instead, complete the destroy flow before return error. Also move mlx5_debug_cq_remove() to the beginning of mlx5_core_destroy_cq() to be symmetrical with mlx5_core_create_cq(). kmemleak complains on: unreferenced object 0xc000000038625100 (size 64): comm "ethtool", pid 28301, jiffies 4298062946 (age 785.380s) hex dump (first 32 bytes): 60 01 48 94 00 00 00 c0 b8 05 34 c3 00 00 00 c0 .H.....4..... 02 00 00 00 00 00 00 00 db 7d c1 00 00 00 c0}..... backtrace: [<000000009e8643cb>] add_res_tree+0xd0/0x270 [mlx5_core] [<00000000e7cb8e6c>] mlx5_debug_cq_add+0x5c/0xc0 [mlx5_core] [<000000002a12918f>] mlx5_core_create_cq+0x1d0/0x2d0 [mlx5_core] [<00000000cef0a696>] mlx5e_create_cq+0x210/0x3f0 [mlx5_core] [<000000009c642c26>] mlx5e_open_cq+0xb4/0x130 [mlx5_core] [<0000000058dfa578>] mlx5e_ptp_open+0x7f4/0xe10 [mlx5_core] [<0000000081839561>] mlx5e_open_channels+0x9cc/0x13e0 [mlx5_core] [<000000009cf05d4>] mlx5e_switch_priv_channels+0xa4/0x230 [mlx5_core] [<0000000042bedd8>] mlx5e_safe_switch_params+0x14c/0x300 [mlx5_core] [<0000000004bc9db8>] set_pflag_tx_port_ts+0x9c/0x160 [mlx5_core] [<00000000a0553443>] mlx5e_set_priv_flags+0xd0/0x1b0 [mlx5_core] [<00000000a8f3d84b>] ethnl_set_privflags+0x234/0x2d0 [<00000000fd27f27c>] genl_family_rcv_msg_doit+0x108/0x1d0 [<00000000f495e2bb>] genl_family_rcv_msg+0xe4/0x1f0 [<00000000646c5c2c>] genl_rcv_msg+0x78/0x120 [<00000000d53e384e>] netlink_rcv_skb+0x74/0x1a0

CVE ID	Description
CVE-2021-47449	In the Linux kernel, the following vulnerability has been resolved: ice: fix locking for Tx timestamp tracking flush Commit 4dd0d5c33c3e ("ice: add lock around Tx timestamp tracker flush") added a lock around the Tx timestamp tracker flow which is used to cleanup any left over SKBs and prepare for device removal. This lock is problematic because it is being held around a call to ice_clear_phy_tstamp. The clear function takes a mutex to send a PHY write command to firmware. This could lead to a deadlock if the mutex actually sleeps, and causes the following warning on a kernel with preemption debugging enabled: [715.419426] BUG: sleeping function called from invalid context at kernel/locking/mutex.c:573 [715.427900] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 3100, name: rmmmod [715.435652] INFO: lockdep is turned off. [715.439591] Preemption disabled at: [715.439594] [<0000000000000000>] 0x0 [715.446678] CPU: 52 PID: 3100 Comm: rmmmod Tainted: G W OE 5.15.0-rc4+ #42 bdd7ec3018e725f159ca0d372ce8c2c0e784891c [715.458058] Hardware name: Intel Corporation S2600STQ/S2600STQ, BIOS SE5C620.86B.02.01.0010.010620200716 01/06/2020 [715.468483] Call Trace: [715.470940] dump_stack_lvl+0x6a/0x9a [715.474613] __might_sleep.cold+0x224/0x26a [715.478895] __mutex_lock+0xb3/0x1440 [715.482569] ? stack_depot_save+0x378/0x500 [715.486763] ? ice_sq_send_cmd+0x78/0x14c0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.494979] ? kfree+0xc1/0x520 [715.498128] ? mutex_lock_io_nested+0x12a0/0x12a0 [715.502837] ? kasan_set_free_info+0x20/0x30 [715.507110] ? __kasan_slab_free+0x10b/0x140 [715.511385] ? slab_free_freelist_hook+0xc7/0x220 [715.516092] ? kfree+0xc1/0x520 [715.519235] ? ice_deinit_lag+0x16c/0x220 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.527359] ? ice_remove+0x1cf/0x6a0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.535133] ? pci_device_remove+0xab/0x1d0 [715.539318] ? __device_release_driver+0x35b/0x690 [715.544110] ? driver_detach+0x214/0x2f0 [715.548035] ? bus_remove_driver+0x11d/0x2f0 [715.552309] ? pci_unregister_driver+0x26/0x250 [715.556840] ? ice_module_exit+0xc/0x2f [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.564799] ? __do_sys_delete_module.constprop.0+0x2d8/0x4e0 [715.570554] ? do_syscall_64+0x3b/0x90 [715.574303] ? entry_SYSCALL_64_after_hwframe+0x44/0xae [715.579529] ? start_flush_work+0x542/0x8f0 [715.583719] ? ice_sq_send_cmd+0x78/0x14c0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.591923] ice_sq_send_cmd+0x78/0x14c0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.599960] ? wait_for_completion_io+0x250/0x250 [715.604662] ? lock_acquire+0x196/0x200 [715.608504] ? do_raw_spin_trylock+0xa5/0x160 [715.612864] ice_sbq_rw_reg+0x1e6/0x2f0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.620813] ? ice_reset+0x130/0x130 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.628497] ? __debug_check_no_obj_freed+0x1e8/0x3c0 [715.633550] ? trace_hardirqs_on+0x1c/0x130 [715.637748] ice_write_phy_reg_e810+0x70/0xf0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.646220] ? do_raw_spin_trylock+0xa5/0x160 [715.650581] ? ice_ptp_release+0x910/0x910 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.658797] ? ice_ptp_release+0x255/0x910 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.667013] ice_clear_phy_tstamp+0x2c/0x110 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.675403] ice_ptp_release+0x408/0x910 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.683440] ice_remove+0x560/0x6a0 [ice 9a7e1ec00971c89ecd3fe0d4dc7da2b3786a421d] [715.691037] ? _raw_spin_unlock_irqrestore+0x46/0x73 [715.696005] pci_device_remove+0xab/0x1d0 [715.700018] __device_release_driver+0x35b/0x690 [715.704637] driver_detach+0x214/0x2f0 [715.708389] bus_remove_driver+0x11d/0x2f0 [715.712489] pci_unregister_driver+0x26/0x250 [71 ---truncated---
CVE-2021-47452	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: skip netdev events generated on netns removal syzbot reported following (harmless) WARN: WARNING: CPU: 1 PID: 2648 at net/netfilter/core.c:468 nft_netdev_unregister_hooks net/netfilter/nf_tables_api.c:230 [inline] nf_tables_unregister_hook include/net/netfilter/nf_tables.h:1090 [inline] __nft_release_basechain+0x138/0x640 net/netfilter/nf_tables_api.c:9524 nft_netdev_event net/netfilter/nft_chain_filter.c:351 [inline] nf_tables_netdev_event+0x521/0x8a0 net/netfilter/nft_chain_filter.c:382 reproducer: unshare -n bash -c 'ip link add br0 type bridge; nft add table netdev t ; \nft add chain netdev t ingress \{ type filter hook ingress device "br0" \ priority 0\; policy drop\; \}' Problem is that when netns device exit hooks create the UNREGISTER event, the .pre_exit hook for nf_tables core has already removed the base hook. Notifier attempts to do this again. The need to do base hook unregister unconditionally was needed in the past, because notifier was last stage where reg->dev dereference was safe. Now that nf_tables does the hook removal in .pre_exit, this isn't needed anymore.
CVE-2021-47453	In the Linux kernel, the following vulnerability has been resolved: ice: Avoid crash from unnecessary IDA free In the remove path, there is an attempt to free the aux_idx IDA whether it was allocated or not. This can potentially cause a crash when unloading the driver on systems that do not initialize support for RDMA. But, this free cannot be gated by the status bit for RDMA, since it is allocated if the driver detects support for RDMA at probe time, but the driver can enter into a state where RDMA is not supported after the IDA has been allocated at probe time and this would lead to a memory leak. Initialize aux_idx to an invalid value and check for a valid value when unloading to determine if an IDA free is necessary.

CVE ID	Description
CVE-2021-47462	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: do not allow illegal MPOL_F_NUMA_BALANCING MPOL_LOCAL in mbind() syzbot reported access to uninitialized memory in mbind() [1] Issue came with commit bda420b98505 ("numa balancing: migrate on fault among multiple bound nodes") This commit added a new bit in MPOL_MODE_FLAGS, but only checked valid combination (MPOL_F_NUMA_BALANCING can only be used with MPOL_BIND) in do_set_mempolicy() This patch moves the check in sanitize_mpol_flags() so that it is also used by mbind() [1] BUG: KMSAN: uninit-value in __mpol_equal+0x567/0x590 mm/mempolicy.c:2260 __mpol_equal+0x567/0x590 mm/mempolicy.c:2260 mpol_equal include/linux/mempolicy.h:105 [inline] vma_merge+0x4a1/0x1e60 mm/mmap.c:1190 mbind_range+0xcc8/0x1e80 mm/mempolicy.c:811 do_mbind+0xf42/0x15f0 mm/mempolicy.c:1333 kernel_mbind mm/mempolicy.c:1483 [inline] __do_sys_mbind mm/mempolicy.c:1490 [inline] __se_sys_mbind+0x437/0xb80 mm/mempolicy.c:1486 __x64_sys_mbind+0x19d/0x200 mm/mempolicy.c:1486 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae Uninit was created at: slab_alloc_node mm/slub.c:3221 [inline] slab_alloc mm/slub.c:3230 [inline] kmem_cache_alloc+0x751/0xff0 mm/slub.c:3235 mpol_new mm/mempolicy.c:293 [inline] do_mbind+0x912/0x15f0 mm/mempolicy.c:1289 kernel_mbind mm/mempolicy.c:1483 [inline] __do_sys_mbind mm/mempolicy.c:1490 [inline] __se_sys_mbind+0x437/0xb80 mm/mempolicy.c:1486 __x64_sys_mbind+0x19d/0x200 mm/mempolicy.c:1486 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae ===== Kernel panic - not syncing: panic_on_kmsan set ... CPU: 0 PID: 15049 Comm: syz-executor.0 Tainted: G B 5.15.0-rc2-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 Call Trace: __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x1ff/0x28e lib/dump_stack.c:106 dump_stack+0x25/0x28 lib/dump_stack.c:113 panic+0x44f/0xdeb kernel/panic.c:232 kmsan_report+0x2ee/0x300 mm/kmsan/report.c:186 __msan_warning+0xd7/0x150 mm/kmsan/instrumentation.c:208 __mpol_equal+0x567/0x590 mm/mempolicy.c:2260 mpol_equal include/linux/mempolicy.h:105 [inline] vma_merge+0x4a1/0x1e60 mm/mmap.c:1190 mbind_range+0xcc8/0x1e80 mm/mempolicy.c:811 do_mbind+0xf42/0x15f0 mm/mempolicy.c:1333 kernel_mbind mm/mempolicy.c:1483 [inline] __do_sys_mbind mm/mempolicy.c:1490 [inline] __se_sys_mbind+0x437/0xb80 mm/mempolicy.c:1486 __x64_sys_mbind+0x19d/0x200 mm/mempolicy.c:1486 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2021-47464	In the Linux kernel, the following vulnerability has been resolved: audit: fix possible null-pointer dereference in audit_filter_rules Fix possible null-pointer dereference in audit_filter_rules. audit_filter_rules() error: we previously assumed 'ctx' could be null
CVE-2021-47465	In the Linux kernel, the following vulnerability has been resolved: KVM: PPC: Book3S HV: Fix stack handling in idle_kvm_start_guest() In commit 10d91611f426 ("powerpc/64s: Reimplement book3s idle code in C") kvm_start_guest() became idle_kvm_start_guest(). The old code allocated a stack frame on the emergency stack, but didn't use the frame to store anything, and also didn't store anything in its caller's frame. idle_kvm_start_guest() on the other hand is written more like a normal C function, it creates a frame on entry, and also stores CR/LR into its callers frame (per the ABI). The problem is that there is no caller frame on the emergency stack. The emergency stack for a given CPU is allocated with: paca_ptrs[i]->emergency_sp = alloc_stack(limit, i) + THREAD_SIZE; So emergency_sp actually points to the first address above the emergency stack allocation for a given CPU, we must not store above it without first decrementing it to create a frame. This is different to the regular kernel stack, paca->kstack, which is initialised to point at an initial frame that is ready to use. idle_kvm_start_guest() stores the backchain, CR and LR all of which write outside the allocation for the emergency stack. It then creates a stack frame and saves the non-volatile registers. Unfortunately the frame it creates is not large enough to fit the non-volatiles, and so the saving of the non-volatile registers also writes outside the emergency stack allocation. The end result is that we corrupt whatever is at 0-24 bytes, and 112-248 bytes above the emergency stack allocation. In practice this has gone unnoticed because the memory immediately above the emergency stack happens to be used for other stack allocations, either another CPUs mc_emergency_sp or an IRQ stack. See the order of calls to irqstack_early_init() and emergency_stack_init(). The low addresses of another stack are the top of that stack, and so are only used if that stack is under extreme pressure, which essentially never happens in practice - and if it did there's a high likelihood we'd crash due to that stack overflowing. Still, we shouldn't be corrupting someone else's stack, and it is purely luck that we aren't corrupting something else. To fix it we save CR/LR into the caller's frame using the existing r1 on entry, we then create a SWITCH_FRAME_SIZE frame (which has space for pt_regs) on the emergency stack with the backchain pointing to the existing stack, and then finally we switch to the new frame on the emergency stack.

CVE ID	Description
CVE-2021-47481	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Initialize the ODP xarray when creating an ODP MR Normally the zero fill would hide the missing initialization, but an errant set to desc_size in reg_create() causes a crash: BUG: unable to handle page fault for address: 0000000800000000 PGD 0 P4D 0 Oops: 0000 [#1] SMP PTI CPU: 5 PID: 890 Comm: ib_write_bw Not tainted 5.15.0-rc4+ #47 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:mlx5_ib_dereg_mr+0x14/0x3b0 [mlx5_ib] Code: 48 63 cd 4c 89 f7 48 89 0c 24 e8 37 30 03 e1 48 8b 0c 24 eb a0 90 0f 1f 44 00 00 41 56 41 55 41 54 55 53 48 89 fb 48 83 ec 30 <48> 8b 2f 65 48 8b 04 25 28 00 00 00 48 89 44 24 28 31 c0 8b 87 c8 RSP: 0018:ffff88811afa3a60 EFLAGS: 00010286 RAX: 000000000000001c RBX: 0000000800000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000800000000 RBP: 0000000800000000 R08: 0000000000000000 R09: c0000000fffff7ff R10: ffff88811afa38f8 R11: ffff88811afa38f0 R12: ffffffff02c7ac0 R13: 0000000000000000 R14: ffff88811afa3cd8 R15: ffff88810772fa00 FS: 00007f47b9080740(0000) GS:ffff88852cd40000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000800000000 CR3: 000000010761e003 CR4: 0000000000370ea0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: mlx5_ib_free_odp_mr+0x95/0xc0 [mlx5_ib] mlx5_ib_dereg_mr+0x128/0x3b0 [mlx5_ib] ib_dereg_mr_user+0x45/0xb0 [ib_core] ? xas_load+0x8/0x80 destroy_hw_idr_uobject+0x1a/0x50 [ib_uverbs] uverbs_destroy_uobject+0x2f/0x150 [ib_uverbs] uobj_destroy+0x3c/0x70 [ib_uverbs] ib_uverbs_cmd_verbs+0x467/0xb00 [ib_uverbs] ? uverbs_finalize_object+0x60/0x60 [ib_uverbs] ? ttwu_queue_wakelist+0xa9/0xe0 ? pty_write+0x85/0x90 ? file_tty_write.isra.33+0x214/0x330 ? process_echoes+0x60/0x60 ib_uverbs_ioctl+0xa7/0x110 [ib_uverbs] __x64_sys_ioctl+0x10d/0x8e0 ? vfs_write+0x17f/0x260 do_syscall_64+0x3c/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae Add the missing xarray initialization and remove the desc_size set.
CVE-2021-47483	In the Linux kernel, the following vulnerability has been resolved: regmap: Fix possible double-free in regcache_rbtrees_exit() In regcache_rbtrees_insert_to_block(), when 'present' realloc failed, the 'blk' which is supposed to assign to 'rbnode->block' will be freed, so 'rbnode->block' points a freed memory, in the error handling path of regcache_rbtrees_init(), 'rbnode->block' will be freed again in regcache_rbtrees_exit(), KASAN will report double-free as follows: BUG: KASAN: double-free or invalid-free in kfree+0xce/0x390 Call Trace: slab_free_freelist_hook+0x10d/0x240 kfree+0xce/0x390 regcache_rbtrees_exit+0x15d/0x1a0 regcache_rbtrees_init+0x224/0x2c0 regcache_init+0x88d/0x1310 __regmap_init+0x3151/0x4a80 __devm_regmap_init+0x7d/0x100 madera_spi_probe+0x10f/0x333 [madera_spi] spi_probe+0x183/0x210 really_probe+0x285/0xc30 To fix this, moving up the assignment of rbnode->block to immediately after the reallocation has succeeded so that the data structure stays valid even if the second reallocation fails.
CVE-2021-47490	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: fix memleak in ttm_transferred_destroy We need to cleanup the fences for ghost objects as well. Bug: https://bugzilla.kernel.org/show_bug.cgi?id=214029 Bug: https://bugzilla.kernel.org/show_bug.cgi?id=214447
CVE-2021-47492	In the Linux kernel, the following vulnerability has been resolved: mm, thp: bail out early in collapse_file for writeback page Currently collapse_file does not explicitly check PG_writeback, instead, page_has_private and try_to_release_page are used to filter writeback pages. This does not work for xfs with blocksize equal to or larger than pagesize, because in such case xfs has no page->private. This makes collapse_file bail out early for writeback page. Otherwise, xfs_end_page_writeback will panic as follows. page:ffffe00201bcc80 refcount:0 mapcount:0 mapping:ffff0003f88c86a8 index:0x0 pfn:0x84ef32 aops:xfs_address_space_operations [xfs] ino:30000b7 dentry name:"libtest.so" flags: 0x57ffff0000008027(locked referenced uptodate active writeback) raw: 57ffff0000008027 ffff80001b48bc28 ffff80001b48bc28 ffff0003f88c86a8 raw: 0000000000000000 0000000000000000 00000000ffffffffff ffff0000c3e9a000 page dumped because: VM_BUG_ON_PAGE(((unsigned int) page_ref_count(page) + 127u <= 127u)) page->mem_cgroup:ffff0000c3e9a000 ----- [cut here]----- kernel BUG at include/linux/mm.h:1212! Internal error: Oops - BUG: 0 [#1] SMP Modules linked in: BUG: Bad page state in process khugepaged pfn:84ef32 xfs(E) page:ffffe00201bcc80 refcount:0 mapcount:0 mapping:0 index:0x0 pfn:0x84ef32 librc32c(E) rkill(E) aes_ce_blk(E) crypto_simd(E) ... CPU: 25 PID: 0 Comm: swapper/25 Kdump: loaded Tainted: ... pstate: 60400005 (nZCv daif +PAN -UAO -TCO BTYPPE=) Call trace: end_page_writeback+0x1c0/0x214 iomap_finish_page_writeback+0x13c/0x204 iomap_finish_ioend+0xe8/0x19c iomap_writepage_end_bio+0x38/0x50 bio_endio+0x168/0x1ec blk_update_request+0x278/0x3f0 blk_mq_end_request+0x34/0x15c virtblk_request_done+0x38/0x74 [virtio_blk] blk_done_softirq+0xc4/0x110 __do_softirq+0x128/0x38c __irq_exit_rcu+0x118/0x150 irq_exit+0x1c/0x30 __handle_domain_irq+0x8c/0xf0 gic_handle_irq+0x84/0x108 el1_irq+0xcc/0x180 arch_cpu_idle+0x18/0x40 default_idle_call+0x4c/0x1a0 cpuidle_idle_call+0x168/0x1e0 do_idle+0xb4/0x104 cpu_startup_entry+0x30/0x9c secondary_start_kernel+0x104/0x180 Code: d4210000 b0006161 910c8021 94013f4d (d4210000) ---[end trace 4a88c6a074082f8c]--- Kernel panic - not syncing: Oops - BUG: Fatal exception in interrupt

CVE ID	Description
CVE-2021-47494	In the Linux kernel, the following vulnerability has been resolved: cfg80211: fix management registrations locking The management registrations locking was broken, the list was locked for each wdev, but cfg80211_mgmt_registrations_update() iterated it without holding all the correct spinlocks, causing list corruption. Rather than trying to fix it with fine-grained locking, just move the lock to the wiphy/rdev (still need the list on each wdev), we already need to hold the wdev lock to change it, so there's no contention on the lock in any case. This trivially fixes the bug since we hold one wdev's lock already, and now will hold the lock that protects all lists.
CVE-2021-47496	In the Linux kernel, the following vulnerability has been resolved: net/tls: Fix flipped sign in tls_err_abort() calls sk->sk_err appears to expect a positive value, a convention that ktls doesn't always follow and that leads to memory corruption in other code. For instance, [kworker] tls_encrypt_done(..., err=<negative error from crypto request>) tls_err_abort(..., err) sk->sk_err = err; [task] splice_from_pipe_feed ... tls_sw_do_sendpage if (sk->sk_err) { ret = -sk->sk_err; // ret is positive splice_from_pipe_feed (continued) ret = actor(...) // ret is still positive and interpreted as bytes // written, resulting in underflow of buf->len and // sd->len, leading to huge buf->offset and bogus // addresses computed in later calls to actor() Fix all tls_err_abort() callers to pass a negative error code consistently and centralize the error-prone sign flip there, throwing in a warning to catch future misuse and unlining the function so it really does only warn once.
CVE-2021-47506	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix use-after-free due to delegation race A delegation break could arrive as soon as we've called vfs_setlease. A delegation break runs a callback which immediately (in nfsd4_cb_recall_prepare) adds the delegation to del_recall_lru. If we then exit nfs4_set_delegation without hashing the delegation, it will be freed as soon as the callback is done with it, without ever being removed from del_recall_lru. Symptoms show up later as use-after-free or list corruption warnings, usually in the laundromat thread. I suspect aba2072f4523 "nfsd: grant read delegations to clients holding writes" made this bug easier to hit, but I looked as far back as v3.0 and it looks to me it already had the same problem. So I'm not sure where the bug was introduced; it may have been there from the beginning.
CVE-2021-47507	In the Linux kernel, the following vulnerability has been resolved: nfsd: Fix nsfd startup race (again) Commit bd5ae9288d64 ("nfsd: register pernet ops last, unregister first") has re-opened rpc_pipefs_event() race against nfsd_net_id registration (register_pernet_subsys()) which has been fixed by commit bb7ffbf29e76 ("nfsd: fix nsfd startup race triggering BUG_ON"). Restore the order of register_pernet_subsys() vs register_cld_notifier(). Add WARN_ON() to prevent a future regression. Crash info: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000012 CPU: 8 PID: 345 Comm: mount Not tainted 5.4.144-... #1 pc : rpc_pipefs_event+0x54/0x120 [nfsd] lr : rpc_pipefs_event+0x48/0x120 [nfsd] Call trace: rpc_pipefs_event+0x54/0x120 [nfsd] blocking_notifier_call_chain rpc_fill_super get_tree_keyed rpc_fs_get_tree vfs_get_tree do_mount ksys_mount __arm64_sys_mount el0_svc_handler el0_svc
CVE-2021-47514	In the Linux kernel, the following vulnerability has been resolved: devlink: fix netns refcount leak in devlink_nl_cmd_reload() While preparing my patch series adding netns refcount tracking, I spotted bugs in devlink_nl_cmd_reload() Some error paths forgot to release a refcount on a netns. To fix this, we can reduce the scope of get_net()/put_net() section around the call to devlink_reload().
CVE-2021-47517	In the Linux kernel, the following vulnerability has been resolved: ethtool: do not perform operations on net devices being unregistered There is a short period between a net device starts to be unregistered and when it is actually gone. In that time frame ethtool operations could still be performed, which might end up in unwanted or undefined behaviours[1]. Do not allow ethtool operations after a net device starts its unregistration. This patch targets the netlink part as the ioctl one isn't affected: the reference to the net device is taken and the operation is executed within an rtnl lock section and the net device won't be found after unregister. [1] For example adding Tx queues after unregister ends up in NULL pointer exceptions and UaFs, such as: BUG: KASAN: use-after-free in kobject_get+0x14/0x90 Read of size 1 at addr ffff88801961248c by task ethtool/755 CPU: 0 PID: 755 Comm: ethtool Not tainted 5.15.0-rc6+ #778 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-4.fc34 04/014 Call Trace: dump_stack_lvl+0x57/0x72 print_address_description.constprop.0+0x1f/0x140 kasan_report.cold+0x7f/0x11b kobject_get+0x14/0x90 kobject_add_internal+0x3d1/0x450 kobject_init_and_add+0xba/0xf0 netdev_queue_update_kobjects+0xcf/0x200 netif_set_real_num_tx_queues+0xb4/0x310 veth_set_channels+0x1c3/0x550 ethtnl_set_channels+0x524/0x610
CVE-2021-47529	In the Linux kernel, the following vulnerability has been resolved: iwlmwifi: Fix memory leaks in error handling path Should an error occur (invalid TLV len or memory allocation failure), the memory already allocated in 'reduce_power_data' should be freed before returning, otherwise it is leaking.

CVE ID	Description
CVE-2021-47536	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix wrong list_del in smc_lgr_cleanup_early smc_lgr_cleanup_early() meant to delete the link group from the link group list, but it deleted the list head by mistake. This may cause memory corruption since we didn't remove the real link group from the list and later memseted the link group structure. We got a list corruption panic when testing: [231.277259] list_del corruption. prev->next should be ffff8881398a8000, but was 0000000000000000 [231.278222] -----[cut here]----- [231.278726] kernel BUG at lib/list_debug.c:53! [231.279326] invalid opcode: 0000 [#1] SMP NOPTI [231.279803] CPU: 0 PID: 5 Comm: kworker/0:0 Not tainted 5.10.46+ #435 [231.280466] Hardware name: Alibaba Cloud ECS, BIOS 8c24b4c 04/01/2014 [231.281248] Workqueue: events smc_link_down_work [231.281732] RIP: 0010: __list_del_entry_valid+0x70/0x90 [231.282258] Code: 4c 60 82 e8 7d cc 6a 00 0f 0b 48 89 fe 48 c7 c7 88 4c 60 82 e8 6c cc 6a 00 0f 0b 48 89 fe 48 c7 c7 c0 4c 60 82 e8 5b cc 6a 00 <0f> 0b 48 89 fe 48 c7 c7 00 4d 60 82 e8 4a cc 6a 00 0f 0b cc cc cc [231.284146] RSP: 0018:ffffc90000033d58 EFLAGS: 00010292 [231.284685] RAX: 0000000000000054 RBX: ffff8881398a8000 RCX: 0000000000000000 [231.285415] RDX: 0000000000000001 RSI: ffff88813bc18040 RDI: ffff88813bc18040 [231.286141] RBP: ffffffff8305ad40 R08: 0000000000000003 R09: 0000000000000001 [231.286873] R10: ffffffff82803da0 R11: ffffc90000033b90 R12: 0000000000000001 [231.287606] R13: 0000000000000000 R14: ffff8881398a8000 R15: 0000000000000003 [231.288337] FS: 0000000000000000(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000 [231.289160] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [231.289754] CR2: 000000000e72058 CR3: 000000010fa96006 CR4: 0000000003706f0 [231.290485] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [231.291211] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 [231.291940] Call Trace: [231.292211] smc_lgr_terminate_sched+0x53/0xa0 [231.292677] smc_switch_conns+0x75/0x6b0 [231.293085] ? update_load_avg+0x1a6/0x590 [231.293517] ? ttwu_do_wakeup+0x17/0x150 [231.293907] ? update_load_avg+0x1a6/0x590 [231.294317] ? newidle_balance+0xca/0x3d0 [231.294716] smcr_link_down+0x50/0x1a0 [231.295090] ? __wake_up_common_lock+0x77/0x90 [231.295534] smc_link_down_work+0x46/0x60 [231.295933] process_one_work+0x18b/0x350
CVE-2021-47550	In the Linux kernel, the following vulnerability has been resolved: drm/amd/amdgpu: fix potential memleak In function amdgpu_get_xgmi_hive, when kobject_init_and_add failed There is a potential memleak if not call kobject_put.

CVE ID	Description
CVE-2021-47552	In the Linux kernel, the following vulnerability has been resolved: blk-mq: cancel blk-mq dispatch work in both blk_cleanup_queue and disk_release() For avoiding to slow down queue destroy, we don't call blk_mq_quiesce_queue() in blk_cleanup_queue(), instead of delaying to cancel dispatch work in blk_release_queue(). However, this way has caused kernel oops[1], reported by Changhui. The log shows that scsi_device can be freed before running blk_release_queue(), which is expected too since scsi_device is released after the scsi disk is closed and the scsi_device is removed. Fixes the issue by canceling blk-mq dispatch work in both blk_cleanup_queue() and disk_release(): 1) when disk_release() is run, the disk has been closed, and any sync dispatch activities have been done, so canceling dispatch work is enough to quiesce filesystem I/O dispatch activity. 2) in blk_cleanup_queue(), we only focus on passthrough request, and passthrough request is always explicitly allocated & freed by its caller, so once queue is frozen, all sync dispatch activity for passthrough request has been done, then it is enough to just cancel dispatch work for avoiding any dispatch activity. [1] kernel panic log [12622.769416] BUG: kernel NULL pointer dereference, address: 0000000000000300 [12622.777186] #PF: supervisor read access in kernel mode [12622.782918] #PF: error_code(0x0000) - not-present page [12622.788649] PGD 0 P4D 0 [12622.791474] Oops: 0000 [#1] PREEMPT SMP PTI [12622.796138] CPU: 10 PID: 744 Comm: kworker/10:1H Kdump: loaded Not tainted 5.15.0+ #1 [12622.804877] Hardware name: Dell Inc. PowerEdge R730/0H21J3, BIOS 1.5.4 10/002/2015 [12622.813321] Workqueue: kblockd blk_mq_run_work_fn [12622.818572] RIP: 0010:sbitmap_get+0x75/0x190 [12622.823336] Code: 85 80 00 00 00 41 8b 57 08 85 d2 0f 84 b1 00 00 00 45 31 e4 48 63 cd 48 8d 1c 49 48 c1 e3 06 49 03 5f 10 4c 8d 6b 40 83 f0 01 <48> 8b 33 44 89 f2 4c 89 ef 0f b6 c8 e8 fa f3 ff ff 83 f8 ff 75 58 [12622.844290] RSP: 0018:ffffb00a446dbd40 EFLAGS: 00010202 [12622.850120] RAX: 0000000000000001 RBX: 0000000000000300 RCX: 0000000000000004 [12622.858082] RDX: 0000000000000006 RSI: 0000000000000082 RDI: ffffa0b7a2dfe030 [12622.866042] RBP: 0000000000000004 R08: 0000000000000001 R09: ffffa0b742721334 [12622.874003] R10: 0000000000000008 R11: 0000000000000008 R12: 0000000000000000 [12622.881964] R13: 0000000000000034 R14: 0000000000000000 R15: ffffa0b7a2dfe030 [12622.889926] FS: 0000000000000000(0000) GS:ffffa0baafb40000(0000) knlGS:0000000000000000 [12622.898956] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [12622.905367] CR2: 0000000000000300 CR3: 0000000641210001 CR4: 00000000001706e0 [12622.913328] Call Trace: [12622.916055] <TASK> [12622.918394] scsi_mq_get_budget+0x1a/0x110 [12622.922969] __blk_mq_do_dispatch_sched+0x1d4/0x320 [12622.928404] ? pick_next_task_fair+0x39/0x390 [12622.933268] __blk_mq_sched_dispatch_requests+0xf4/0x140 [12622.939194] blk_mq_sched_dispatch_requests+0x30/0x60 [12622.944829] __blk_mq_run_hw_queue+0x30/0xa0 [12622.949593] process_one_work+0x1e8/0x3c0 [12622.954059] worker_thread+0x50/0x3b0 [12622.958144] ? rescuer_thread+0x370/0x370 [12622.962616] kthread+0x158/0x180 [12622.966218] ? set_kthread_struct+0x40/0x40 [12622.970884] ret_from_fork+0x22/0x30 [12622.974875] </TASK> [12622.977309] Modules linked in: scsi_debug rpcsec_gss_krb5 auth_rpcgss nfsv4 dns_resolver nfs lockd grace fscache netfs sunrpc dm_multipath intel_rapl_msr intel_rapl_common dell_wmi_descriptor sb_edac rfkill video x86_pkg_temp thermal intel_powerclamp dcdbas coretemp kvm_intel kvm mgag200 irqbypass i2c_algo_bit rapl drm_kms_helper ipmi_ssif intel_cstate intel_uncore syscopyarea sysfillrect sysimgblt fb_sys_fops pcspkr cec mei_me lpc_ich mei ipmi_si ipmi_devintf ipmi_msghandler acpi_power_meter drm fuse xfs libcrc32c sr_mod cdrom sd_mod t10_pi sg ixgbe ahci libahci crct10dif_pclmul crc32_pclmul crc32c_intel libata megaraid_sas ghash_clmulni_intel tg3 wdat_w ---truncated---
CVE-2021-47554	In the Linux kernel, the following vulnerability has been resolved: vdpa_sim: avoid putting an uninitialized iova_domain The system will crash if we put an uninitialized iova_domain, this could happen when an error occurs before initializing the iova_domain in vdpasim_create(). BUG: kernel NULL pointer dereference, address: 0000000000000000 ... RIP: 0010:__cpuhp_state_remove_instance+0x96/0x1c0 ... Call Trace: <TASK> put_iova_domain+0x29/0x220 vdpasim_free+0xd1/0x120 [vdpa_sim] vdpasim_release_dev+0x21/0x40 [vdpa] device_release+0x33/0x90 kobject_release+0x63/0x160 vdpasim_create+0x127/0x2a0 [vdpa_sim] vdpasim_net_dev_add+0x7d/0xfe [vdpa_sim_net] vdpasim_nl_cmd_dev_add_set_doit+0xe1/0x1a0 [vdpa] genl_family_rcv_msg_doit+0x112/0x140 genl_rcv_msg+0xdf/0x1d0 ... So we must make sure the iova_domain is already initialized before put it. In addition, we may get the following warning in this case: WARNING: ... drivers/iommu/iovac.c:344 iova_cache_put+0x58/0x70 So we must make sure the iova_cache_get() is invoked only if the iova_cache_get() is already invoked. Let's fix it together.

CVE ID	Description
CVE-2021-47557	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_ets: don't peek at classes beyond 'nbands' when the number of DRR classes decreases, the round-robin active list can contain elements that have already been freed in ets_qdisc_change(). As a consequence, it's possible to see a NULL dereference crash, caused by the attempt to call cl->qdisc->ops->peek(cl->qdisc) when cl->qdisc is NULL: BUG: kernel NULL pointer dereference, address: 0000000000000018 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 1 PID: 910 Comm: mausezahn Not tainted 5.16.0-rc1+ #475 Hardware name: Red Hat KVM, BIOS 1.11.1-4.module+el8.1.0+4066+0f1aadab 04/01/2014 RIP: 0010:ets_qdisc_dequeue+0x129/0x2c0 [sch_ets] Code: c5 01 41 39 ad e4 02 00 00 0f 87 18 ff ff ff 49 8b 85 c0 02 00 00 49 39 c4 0f 84 ba 00 00 00 49 8b ad c0 02 00 00 48 8b 7d 10 <48> 8b 47 18 48 8b 40 38 0f ae e8 ff d0 48 89 c3 48 85 c0 0f 84 9d RSP: 0000:ffffb36c0b5fdd8 EFLAGS: 00010287 RAX: ffff956678efed30 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000000002 RSI: ffffffff9b938dc9 RDI: 0000000000000000 RBP: ffff956678efed30 R08: e2f3207fe360129c R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000001 R12: ffff956678efead0 R13: ffff956678efe800 R14: ffff956611545000 R15: ffff95667ac8f100 FS: 00007f2aa9120740(0000) GS: ffff95667b800000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000018 CR3: 000000011070c000 CR4: 0000000000350ee0 Call Trace: <TASK> qdisc_peek_dequeued+0x29/0x70 [sch_ets] tbf_dequeue+0x22/0x260 [sch_tbf] __qdisc_run+0x7f/0x630 net_tx_action+0x290/0x4c0 __do_softirq+0xee/0x4f8 irq_exit_rcu+0xf4/0x130 sysvec_apic_timer_interrupt+0x52/0xc0 asm_sysvec_apic_timer_interrupt+0x12/0x20 RIP: 0033:0x7f2aa7fc9ad4 Code: b9 ff ff 48 8b 54 24 18 48 83 c4 08 48 89 ee 48 89 df 5b 5d e9 ed fc ff ff 0f 1f 00 66 2e 0f 1f 84 00 00 00 00 00 f3 0f 1e fa <53> 48 83 ec 10 48 8b 05 10 64 33 00 48 8b 00 48 85 c0 0f 85 84 00 RSP: 002b:00007ffe5d33fab8 EFLAGS: 00000202 RAX: 0000000000000002 RBX: 0000561f72c31460 RCX: 0000561f72c31720 RDX: 0000000000000000 RSI: 0000561f72c31722 RDI: 0000561f72c31720 RBP: 000000000000002a R08: 00007ffe5d33fa40 R09: 0000000000000014 R10: 0000000000000000 R11: 0000000000000246 R12: 0000561f7187e380 R13: 0000000000000000 R14: 0000000000000000 R15: 0000561f72c31460 </TASK> Modules linked in: sch_ets sch_tbf dummy rfkill iTCO_wdt intel_rapl_msr iTCO_vendor_support intel_rapl_common joydev virtio_balloon lpc_ich i2c_i801 i2c_smbus pcspkr ip_tables xfs libcrc32c crct10dif_pclmul crc32_pclmul crc32c_intel ahci libahci ghash_clmulni_intel serio_raw libata virtio_blk virtio_console virtio_net net_failover failover sunrpc dm_mirror dm_region_hash dm_log dm_mod CR2: 0000000000000018 Ensuring that 'alist' was never zeroed [1] was not sufficient, we need to remove from the active list those elements that are no more SP nor DRR. [1] https://lore.kernel.org/netdev/60d274838bf09777f0371253416e8af71360bc08.1633609148.git.dcaratti@redhat.com/ v3: fix race between ets_qdisc_change() and ets_qdisc_dequeue() delisting DRR classes beyond 'nbands' in ets_qdisc_change() with the qdisc lock acquired, thanks to Cong Wang. v2: when a NULL qdisc is found in the DRR active list, try to dequeue skb from the next list item.
CVE-2021-47558	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: Disable Tx queues when reconfiguring the interface The Tx queues were not disabled in situations where the driver needed to stop the interface to apply a new configuration. This could result in a kernel panic when doing any of the 3 following actions: * reconfiguring the number of queues (ethtool -L) * reconfiguring the size of the ring buffers (ethtool -G) * installing/removing an XDP program (ip l set dev ethX xdp) Prevent the panic by making sure netif_tx_disable is called when stopping an interface. Without this patch, the following kernel panic can be observed when doing any of the actions above: Unable to handle kernel paging request at virtual address ffff80001238d040 [...] Call trace: dwmac4_set_addr+0x8/0x10 dev_hard_start_xmit+0xe4/0x1ac sch_direct_xmit+0xe8/0x39c __dev_queue_xmit+0x3ec/0xaf0 dev_queue_xmit+0x14/0x20 [...] [end trace 0000000000000002]---
CVE-2021-47559	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix NULL pointer dereferencing in smc_vlan_by_tcpsk() Coverity reports a possible NULL dereferencing problem: in smc_vlan_by_tcpsk(): 6. returned_null: netdev_lower_get_next returns NULL (checked 29 out of 30 times). 7. var_assigned: Assigning: ndev = NULL return value from netdev_lower_get_next. 1623 ndev = (struct net_device *)netdev_lower_get_next(ndev, &lower); CID 1468509 (#1 of 1): Dereference null return value (NULL_RETURNS) 8. dereference: Dereferencing a pointer that might be NULL ndev when calling is_vlan_dev. 1624 if (is_vlan_dev(ndev)) { Remove the manual implementation and use netdev_walk_all_lower_dev() to iterate over the lower devices. While on it remove an obsolete function parameter comment.

CVE ID	Description
CVE-2021-47562	In the Linux kernel, the following vulnerability has been resolved: ice: fix vsi->txq_map sizing The approach of having XDP queue per CPU regardless of user's setting exposed a hidden bug that could occur in case when Rx queue count differ from Tx queue count. Currently vsi->txq_map's size is equal to the doubled vsi->alloc_txq, which is not correct due to the fact that XDP rings were previously based on the Rx queue count. Below splat can be seen when ethtool -L is used and XDP rings are configured: [682.875339] BUG: kernel NULL pointer dereference, address: 000000000000000f [682.883403] #PF: supervisor read access in kernel mode [682.889345] #PF: error_code(0x0000) - not-present page [682.895289] PGD 0 P4D 0 [682.898218] Oops: 0000 [#1] PREEMPT SMP PTI [682.903055] CPU: 42 PID: 2878 Comm: ethtool Tainted: G OE 5.15.0-rc5+ #1 [682.912214] Hardware name: Intel Corp. GRANTLEY/GRANTLEY, BIOS GRRFCRB1.86B.0276.D07.1605190235 05/19/2016 [682.923380] RIP: 0010:devres_remove+0x44/0x130 [682.928527] Code: 49 89 f4 55 48 89 fd 4c 89 ff 53 48 83 ec 10 e8 92 b9 49 00 48 8b 9d a8 02 00 00 48 8d 8d a0 02 00 00 49 89 c2 48 39 cb 74 0f <4c> 3b 63 10 74 25 48 8b 5b 08 48 39 cb 75 f1 4c 89 ff 4c 89 d6 e8 [682.950237] RSP: 0018:ffffc90006a679f0 EFLAGS: 00010002 [682.956285] RAX: 0000000000000286 RBX: ffffffff81690d60 RCX: ffff88908343a370 [682.964538] RDX: 0000000000000001 RSI: ffffffff81690d60 RDI: 0000000000000000 [682.972789] RBP: ffff88908343a0d0 R08: 0000000000000000 R09: 0000000000000000 [682.981040] R10: 0000000000000286 R11: 3fffffffffffffff R12: ffffffff81690d60 [682.989282] R13: ffffffff81690a00 R14: ffff8890819807a8 R15: ffff88908343a36c [682.997535] FS: 00007f08c7bfa740(0000) GS:ffff88a03fd00000(0000) knlGS:0000000000000000 [683.006910] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [683.013557] CR2: 000000000000000f CR3: 0000001080a66003 CR4: 00000000003706e0 [683.021819] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [683.030075] DR3: 0000000000000000 DR6: 00000000ffff0fff DR7: 0000000000000400 [683.038336] Call Trace: [683.041167] devm_kfree+0x33/0x50 [683.045004] ice_vsi_free_arrays+0x5e/0xc0 [ice] [683.050380] ice_vsi_rebuild+0x4c8/0x750 [ice] [683.055543] ice_vsi_recfg_qs+0x9a/0x110 [ice] [683.060697] ice_set_channels+0x14f/0x290 [ice] [683.065962] ethnl_set_channels+0x333/0x3f0 [683.070807] genl_family_rcv_msg_doit+0xea/0x150 [683.076152] genl_rcv_msg+0xde/0x1d0 [683.080289] ? channels_prepare_data+0x60/0x60 [683.085432] ? genl_get_cmd+0xd0/0xd0 [683.089667] netlink_rcv_skb+0x50/0xf0 [683.094006] genl_rcv+0x24/0x40 [683.097638] netlink_unicast+0x239/0x340 [683.102177] netlink_sendmsg+0x22e/0x470 [683.106717] sock_sendmsg+0x5e/0x60 [683.110756] __sys_sendto+0xee/0x150 [683.114894] ? handle_mm_fault+0xd0/0x2a0 [683.119535] ? do_user_addr_fault+0x1f3/0x690 [683.134173] __x64_sys_sendto+0x25/0x30 [683.148231] do_syscall_64+0x3b/0xc0 [683.161992] entry_SYSCALL_64_after_hwframe+0x44/0xae Fix this by taking into account the value that num_possible_cpus() yields in addition to vsi->alloc_txq instead of doubling the latter.

CVE ID	Description
CVE-2021-47563	In the Linux kernel, the following vulnerability has been resolved: ice: avoid bpf_prog refcount underflow Ice driver has the routines for managing XDP resources that are shared between ndo_bpf op and VSI rebuild flow. The latter takes place for example when user changes queue count on an interface via ethtool's set_channels(). There is an issue around the bpf_prog refcounting when VSI is being rebuilt - since ice_prepare_xdp_rings() is called with vsi->xdp_prog as an argument that is used later on by ice_vsi_assign_bpf_prog(), same bpf_prog pointers are swapped with each other. Then it is also interpreted as an 'old_prog' which in turn causes us to call bpf_prog_put on it that will decrement its refcount. Below splat can be interpreted in a way that due to zero refcount of a bpf_prog it is wiped out from the system while kernel still tries to refer to it: [481.069429] BUG: unable to handle page fault for address: fffff9000640f038 [481.077390] #PF: supervisor read access in kernel mode [481.083335] #PF: error_code(0x0000) - not-present page [481.089276] PGD 100000067 P4D 100000067 PUD 1001cb067 PMD 106d2b067 PTE 0 [481.097141] Oops: 0000 [#1] PREEMPT SMP PTI [481.101980] CPU: 12 PID: 3339 Comm: sudo Tainted: G OE 5.15.0-rc5+ #1 [481.110840] Hardware name: Intel Corp. GRANTLEY/GRANTLEY, BIOS GRRFCRB1.86B.0276.D07.1605190235 05/19/2016 [481.122021] RIP: 0010:dev_xdp_prog_id +0x25/0x40 [481.127265] Code: 80 00 00 00 00 0f 1f 44 00 00 89 f6 48 c1 e6 04 48 01 fe 48 8b 86 98 08 00 00 48 85 c0 74 13 48 8b 50 18 31 c0 48 85 d2 74 07 <48> 8b 42 38 8b 40 20 c3 48 8b 96 90 08 00 00 eb e8 66 2e 0f 1f 84 [481.148991] RSP: 0018:ffffc90007b63868 EFLAGS: 00010286 [481.155034] RAX: 0000000000000000 RBX: ffff889080824000 RCX: 0000000000000000 [481.163278] RDX: fffff9000640f000 RSI: ffff889080824010 RDI: ffff889080824000 [481.171527] RBP: ffff888107af7d00 R08: 0000000000000000 R09: ffff88810db5f6e0 [481.179776] R10: 0000000000000000 R11: ffff8890885b9988 R12: ffff88810db5f4bc [481.188026] R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 [481.196276] FS: 00007f5466d5bec0(0000) GS:ffff88903fb00000(0000) knlGS:0000000000000000 [481.205633] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [481.212279] CR2: fffff9000640f038 CR3: 000000014429c006 CR4: 000000000003706e0 [481.220530] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [481.228771] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [481.237029] Call Trace: [481.239856] rtnl_fill_ifinfo+0x768/0x12e0 [481.244602] rtnl_dump_ifinfo+0x525/0x650 [481.249246] ? __alloc_skb+0xa5/0x280 [481.253484] netlink_dump+0x168/0x3c0 [481.257725] netlink_rcvmsg +0x21e/0x3e0 [481.262263] ____sys_rcvmsg+0x87/0x170 [481.266707] ? __might_fault +0x20/0x30 [481.271046] ? _copy_from_user+0x66/0xa0 [481.275591] ? iovector_from_user +0xf6/0x1c0 [481.280226] ____sys_rcvmsg+0x82/0x100 [481.284566] ? sock_sendmsg+0x5e/0x60 [481.288791] ? __sys_sendto+0xee/0x150 [481.293129] __sys_rcvmsg+0x56/0xa0 [481.297267] do_syscall_64+0x3b/0xc0 [481.301395] entry_SYSCALL_64_after_hwframe+0x44/0xae [481.307238] RIP: 0033:0x7f5466f39617 [481.311373] Code: 0c 00 f7 d8 64 89 02 48 c7 c0 ff ff ff eb bd 0f 1f 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 2f 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 51 c3 48 83 ec 28 89 54 24 1c 48 89 74 24 10 [481.342944] RSP: 002b:00007ffedc7f4308 EFLAGS: 00000246 ORIG_RAX: 000000000000002f [481.361783] RAX: ffffffffefda RBX: 00007ffedc7f5460 RCX: 00007f5466f39617 [481.380278] RDX: 0000000000000000 RSI: 00007ffedc7f5360 RDI: 0000000000000003 [481.398500] RBP: 00007ffedc7f53f0 R08: 0000000000000000 R09: 000055d556f04d50 [481.416463] R10: 0000000000000077 R11: 00000000000000246 R12: 00007ffedc7f5360 [481.434131] R13: 00007ffedc7f5350 R14: 00007ffedc7f5344 R15: 0000000000000e98 [481.451520] Modules linked in: ice ---truncated---
CVE-2021-47565	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix kernel panic during drive powercycle test While looping over shost's sdev list it is possible that one of the drives is getting removed and its sas_target object is freed but its sdev object remains intact. Consequently, a kernel panic can occur while the driver is trying to access the sas_address field of sas_target object without also checking the sas_target object for NULL.

CVE ID	Description
CVE-2021-47566	In the Linux kernel, the following vulnerability has been resolved: proc/vmcore: fix clearing user buffer by properly using clear_user() To clear a user buffer we cannot simply use memset, we have to use clear_user(). With a virtio-mem device that registers a vmcore_cb and has some logically unplugged memory inside an added Linux memory block, I can easily trigger a BUG by copying the vmcore via "cp": systemd[1]: Starting Kdump Vmcore Save Service... kdump[420]: Kdump is using the default log level(3). kdump[453]: saving to /sysroot/var/crash/127.0.0.1-2021-11-11-14:59:22/ kdump[458]: saving vmcore-dmesg.txt to /sysroot/var/crash/127.0.0.1-2021-11-11-14:59:22/ kdump[465]: saving vmcore-dmesg.txt complete kdump[467]: saving vmcore BUG: unable to handle page fault for address: 00007f2374e01000 #PF: supervisor write access in kernel mode #PF: error_code(0x0003) - permissions violation PGD 7a523067 P4D 7a523067 PUD 7a528067 PMD 7a525067 PTE 800000007048f867 Oops: 0003 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 468 Comm: cp Not tainted 5.15.0+ #6 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.14.0-27-g64f37cc530f1-prebuilt.qemu.org 04/01/2014 RIP: 0010:read_from_oldmem.part.0.cold+0x1d/0x86 Code: ff ff ff e8 05 ff fe ff e9 b9 e9 7f ff 48 89 de 48 c7 c7 38 3b 60 82 e8 f1 fe ff 83 fd 08 72 3c 49 8d 7d 08 4c 89 e9 89 e8 <49> c7 45 00 00 00 00 00 49 c7 44 05 f8 00 00 00 00 48 83 e7 f81 RSP: 0018:ffff9000073be08 EFLAGS: 00010212 RAX: 00000000000001000 RBX: 000000000002fd000 RCX: 00007f2374e01000 RDX: 0000000000000001 RSI: 00000000ffffdfff RDI: 00007f2374e01008 RBP: 00000000000001000 R08: 0000000000000000 R09: ffff9000073bc50 R10: ffff9000073bc48 R11: ffffffff829461a8 R12: 0000000000000f000 R13: 00007f2374e01000 R14: 0000000000000000 R15: ffff88807bd421e8 FS: 00007f2374e12140(0000) GS:ffff88807f000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f2374e01000 CR3: 000000007a4aa000 CR4: 0000000000350eb0 Call Trace: read_vmcore+0x236/0x2c0 proc_reg_read+0x55/0xa0 vfs_read+0x95/0x190 ksys_read+0x4f/0xc0 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae Some x86-64 CPUs have a CPU feature called "Supervisor Mode Access Prevention (SMAP)", which is used to detect wrong access from the kernel to user buffers like this: SMAP triggers a permissions violation on wrong access. In the x86-64 variant of clear_user(), SMAP is properly handled via clac()+stac(). To fix, properly use clear_user() when we're dealing with a user buffer.
CVE-2021-47576	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Sanity check block descriptor length in resp_mode_select() In resp_mode_select() sanity check the block descriptor len to avoid UAF. BUG: KASAN: use-after-free in resp_mode_select+0xa4c/0xb40 drivers/scsi/scsi_debug.c:2509 Read of size 1 at addr ffff888026670f50 by task scsicmd/15032 CPU: 1 PID: 15032 Comm: scsicmd Not tainted 5.15.0-01d0625 #15 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Call Trace: <TASK> dump_stack_lvl+0x89/0xb5 lib/dump_stack.c:107 print_address_description.constprop.9+0x28/0x160 mm/kasan/report.c:257 kasan_report.cold.14+0x7d/0x117 mm/kasan/report.c:443 __asan_report_load1_noabort+0x14/0x20 mm/kasan/report_generic.c:306 resp_mode_select+0xa4c/0xb40 drivers/scsi/scsi_debug.c:2509 schedule_resp+0x4af/0x1a10 drivers/scsi/scsi_debug.c:5483 scsi_debug_queuecommand+0x8c9/0x1e70 drivers/scsi/scsi_debug.c:7537 scsi_queue_rq+0x16b4/0x2d10 drivers/scsi/scsi_lib.c:1521 blk_mq_dispatch_rq_list+0xb9b/0x2700 block/blk-mq.c:1640 __blk_mq_sched_dispatch_requests+0x28f/0x590 block/blk-mq-sched.c:325 blk_mq_sched_dispatch_requests+0x105/0x190 block/blk-mq-sched.c:358 __blk_mq_run_hw_queue+0xe5/0x150 block/blk-mq.c:1762 __blk_mq_delay_run_hw_queue+0x4f8/0x5c0 block/blk-mq.c:1839 blk_mq_run_hw_queue+0x18d/0x350 block/blk-mq.c:1891 blk_mq_sched_insert_request+0x3db/0x4e0 block/blk-mq-sched.c:474 blk_execute_rq_nowait+0x16b/0x1c0 block/blk-exec.c:63 sg_common_write.isra.18+0xeb3/0x2000 drivers/scsi/sg.c:837 sg_new_write.isra.19+0x570/0x8c0 drivers/scsi/sg.c:775 sg_ioctl_common+0x14d6/0x2710 drivers/scsi/sg.c:941 sg_ioctl+0xa2/0x180 drivers/scsi/sg.c:1166 __x64_sys_ioctl+0x19d/0x220 fs/ioctl.c:52 do_syscall_64+0x3a/0x80 arch/x86/entry/common.c:50 entry_SYSCALL_64_after_hwframe+0x44/0xae arch/x86/entry/entry_64.S:113

CVE ID	Description
CVE-2021-47578	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_debug: Don't call kcalloc() if size arg is zero If the size arg to kcalloc() is zero, it returns ZERO_SIZE_PTR. Because of that, for a following NULL pointer check to work on the returned pointer, kcalloc() must not be called with the size arg equal to zero. Return early without error before the kcalloc() call if size arg is zero. BUG: KASAN: null-ptr-deref in memcpy include/linux/fortify-string.h:191 [inline] BUG: KASAN: null-ptr-deref in sg_copy_buffer+0x138/0x240 lib/scatterlist.c:974 Write of size 4 at addr 0000000000000010 by task syz-executor.1/22789 CPU: 1 PID: 22789 Comm: syz-executor.1 Not tainted 5.15.0-syzk #1 Hardware name: Red Hat KVM, BIOS 1.13.0-2 Call Trace: __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x89/0xb5 lib/dump_stack.c:106 __kasan_report mm/kasan/report.c:446 [inline] kasan_report.cold.14+0x112/0x117 mm/kasan/report.c:459 check_region_inline mm/kasan/generic.c:183 [inline] kasan_check_range+0x1a3/0x210 mm/kasan/generic.c:189 memcpy+0x3b/0x60 mm/kasan/shadow.c:66 memcpy include/linux/fortify-string.h:191 [inline] sg_copy_buffer+0x138/0x240 lib/scatterlist.c:974 do_dout_fetch drivers/scsi/scsi_debug.c:2954 [inline] do_dout_fetch drivers/scsi/scsi_debug.c:2946 [inline] resp_verify+0x49e/0x930 drivers/scsi/scsi_debug.c:4276 schedule_resp+0x4d8/0x1a70 drivers/scsi/scsi_debug.c:5478 scsi_debug_queuecommand+0x8c9/0x1ec0 drivers/scsi/scsi_debug.c:7533 scsi_dispatch_cmd drivers/scsi/scsi_lib.c:1520 [inline] scsi_queue_rq+0x16b0/0x2d40 drivers/scsi/scsi_lib.c:1699 blk_mq_dispatch_rq_list+0xb9b/0x2700 block/blk-mq.c:1639 __blk_mq_sched_dispatch_requests+0x28f/0x590 block/blk-mq-sched.c:325 blk_mq_sched_dispatch_requests+0x105/0x190 block/blk-mq-sched.c:358 __blk_mq_run_hw_queue+0xe5/0x150 block/blk-mq.c:1761 __blk_mq_delay_run_hw_queue+0x4f8/0x5c0 block/blk-mq.c:1838 blk_mq_run_hw_queue+0x18d/0x350 block/blk-mq.c:1891 blk_mq_sched_insert_request+0x3db/0x4e0 block/blk-mq-sched.c:474 blk_execute_rq_nowait+0x16b/0x1c0 block/blk-exec.c:62 blk_execute_rq+0xdb/0x360 block/blk-exec.c:102 sg_scsi_ioctl drivers/scsi/scsi_ioctl.c:621 [inline] scsi_ioctl+0x8bb/0x15c0 drivers/scsi/scsi_ioctl.c:930 sg_ioctl_common+0x172d/0x2710 drivers/scsi/sg.c:1112 sg_ioctl+0xa2/0x180 drivers/scsi/sg.c:1165 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:874 [inline] __se_sys_ioctl fs/ioctl.c:860 [inline] __x64_sys_ioctl+0x19d/0x220 fs/ioctl.c:860 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3a/0x80 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2021-47588	In the Linux kernel, the following vulnerability has been resolved: sit: do not call ipip6_dev_free() from sit_init_net() ipip6_dev_free is sit dev->priv_destructor, already called by register_netdevice() if something goes wrong. Alternative would be to make ipip6_dev_free() robust against multiple invocations, but other drivers do not implement this strategy. syzbot reported: dst_release underflow WARNING: CPU: 0 PID: 5059 at net/core/dst.c:173 dst_release+0xd8/0xe0 net/core/dst.c:173 Modules linked in: CPU: 1 PID: 5059 Comm: syz-executor.4 Not tainted 5.16.0-rc5-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:dst_release+0xd8/0xe0 net/core/dst.c:173 Code: 4c 89 f2 89 d9 31 c0 5b 41 5e 5d e9 da d5 44 f9 e8 1d 90 5f f9 c6 05 87 48 c6 05 01 48 c7 c7 80 44 99 8b 31 c0 e8 e8 67 29 f9 <0f> 0b eb 85 0f 1f 40 00 53 48 89 fb e8 f7 8f 5f f9 48 83 c3 a8 48 RSP: 0018:ffffc9000aa5faa0 EFLAGS: 00010246 RAX: d6894a925dd15a00 RBX: 0000000000000000 RCX: 0000000000004000 RDX: fffff90005e19000 RSI: 0000000000003fff RDI: 0000000000004000 RBP: 0000000000000000 R08: ffffffff816a1f42 R09: fffffd1017344f2c R10: fffffd1017344f2c R11: 0000000000000000 R12: 0000607f462b1358 R13: 1fffffff1bfd305 R14: fffff8ffffcb1358 R15: dffffc0000000000 FS: 00007f66c71a2700(0000) GS:ffff8880b9a00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f88aed5058 CR3: 0000000023e0f000 CR4: 0000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> dst_cache_destroy+0x107/0x1e0 net/core/dst_cache.c:160 ipip6_dev_free net/ipv6/sit.c:1414 [inline] sit_init_net+0x229/0x550 net/ipv6/sit.c:1936 ops_init+0x313/0x430 net/core/net_namespace.c:140 setup_net+0x35b/0x9d0 net/core/net_namespace.c:326 copy_net_ns+0x359/0x5c0 net/core/net_namespace.c:470 create_new_namespaces+0x4ce/0xa00 kernel/nsproxy.c:110 unshare_nsproxy_namespaces+0x11e/0x180 kernel/nsproxy.c:226 ksys_unshare+0x57d/0xb50 kernel/fork.c:3075 __do_sys_unshare kernel/fork.c:3146 [inline] __se_sys_unshare kernel/fork.c:3144 [inline] __x64_sys_unshare+0x34/0x40 kernel/fork.c:3144 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x44/0xd0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f66c882ce99 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff f7 d8 64 89 01 48 RSP: 002b:00007f66c71a2168 EFLAGS: 00000246 ORIG_RAX: 0000000000000110 RAX: ffffffff816a1f42 RBX: 00007f66c893ff60 RCX: 00007f66c882ce99 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000048040200 RBP: 00007f66c8886ff1 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007f66c71a2300 R14: 00007f66c71a2300 R15: 0000000000002200 </TASK>

CVE ID	Description
CVE-2021-47591	In the Linux kernel, the following vulnerability has been resolved: mptcp: remove tcp ulp setsockopt support TCP_ULP setsockopt cannot be used for mptcp because its already used internally to plumb subflow (tcp) sockets to the mptcp layer. syzbot managed to trigger a crash for mptcp connections that are in fallback mode: KASAN: null-ptr-deref in range [0x0000000000000020-0x0000000000000027] CPU: 1 PID: 1083 Comm: syz-executor.3 Not tainted 5.16.0-rc2-syzkaller #0 RIP: 0010:tls_build_proto net/tls/tls_main.c:776 [inline] [...] __tcp_set_ulp net/ipv4/tcp_ulp.c:139 [inline] tcp_set_ulp +0x428/0x4c0 net/ipv4/tcp_ulp.c:160 do_tcp_setsockopt+0x455/0x37c0 net/ipv4/tcp.c:3391 mptcp_setsockopt+0x1b47/0x2400 net/mptcp/socket.c:638 Remove support for TCP_ULP setsockopt.
CVE-2021-47594	In the Linux kernel, the following vulnerability has been resolved: mptcp: never allow the PM to close a listener subflow Currently, when deleting an endpoint the netlink PM traverses all the local MPTCP sockets, regardless of their status. If an MPTCP listener socket is bound to the IP matching the delete endpoint, the listener TCP socket will be closed. That is unexpected, the PM should only affect data subflows. Additionally, syzbot was able to trigger a NULL ptr dereference due to the above: general protection fault, probably for non-canonical address 0xdffffc0000000003: 0000 [#1] PREEMPT SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000018-0x000000000000001f] CPU: 1 PID: 6550 Comm: syz-executor122 Not tainted 5.16.0-rc4-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:__lock_acquire +0xd7d/0x54a0 kernel/locking/lockdep.c:4897 Code: 0f 0e 41 be 01 00 00 00 0f 86 c8 00 00 00 89 05 69 cc 0f 0e e9 bd 00 00 00 48 b8 00 00 00 00 00 fc ff df 48 89 da 48 c1 ea 03 <80> 3c 02 00 0f 85 f3 2f 00 00 48 81 3b 20 75 17 8f 0f 84 52 f3 ff RSP: 0018:ffff90001f2f818 EFLAGS: 00010016 RAX: dffffc0000000000 RBX: 0000000000000018 RCX: 0000000000000000 RDX: 0000000000000003 RSI: 0000000000000000 RDI: 0000000000000001 RBP: 0000000000000000 R08: 0000000000000001 R09: 0000000000000001 R10: 0000000000000000 R11: 000000000000000a R12: 0000000000000000 R13: ffff88801b98d700 R14: 0000000000000000 R15: 0000000000000001 FS: 00007f177cd3d700(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f177cd1b268 CR3: 000000001dd55000 CR4: 0000000000350ee0 Call Trace: <TASK> lock_acquire kernel/locking/lockdep.c:5637 [inline] lock_acquire+0x1ab/0x510 kernel/locking/lockdep.c:5602 __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:110 [inline] _raw_spin_lock_irqsave+0x39/0x50 kernel/locking/spinlock.c:162 finish_wait+0xc0/0x270 kernel/sched/wait.c:400 inet_csk_wait_for_connect net/ipv4/inet_connection_sock.c:464 [inline] inet_csk_accept+0x7de/0x9d0 net/ipv4/inet_connection_sock.c:497 mptcp_accept+0xe5/0x500 net/mptcp/protocol.c:2865 inet_accept+0xe4/0x7b0 net/ipv4/af_inet.c:739 mptcp_stream_accept+0x2e7/0x10e0 net/mptcp/protocol.c:3345 do_accept+0x382/0x510 net/socket.c:1773 __sys_accept4_file+0x7e/0xe0 net/socket.c:1816 __sys_accept4+0xb0/0x100 net/socket.c:1846 __do_sys_accept net/socket.c:1864 [inline] __se_sys_accept net/socket.c:1861 [inline] __x64_sys_accept+0x71/0xb0 net/socket.c:1861 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe +0x44/0xae RIP: 0033:0x7f177cd8b8e9 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 b1 14 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f177cd3d308 EFLAGS: 00000246 ORIG_RAX: 000000000000002b RAX: ffffffff8bda RBX: 00007f177ce13408 RCX: 00007f177cd8b8e9 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003 RBP: 00007f177ce13400 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 00007f177ce1340c R13: 00007f177cde1004 R14: 6d705f706374706d R15: 00000000000022000 </TASK> Fix the issue explicitly skipping MPTCP socket in TCP_LISTEN status.

CVE ID	Description
CVE-2021-47595	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_ets: don't remove idle classes from the round-robin list Shuang reported that the following script: 1) tc qdisc add dev ddd0 handle 10: parent 1: ets bands 8 strict 4 priomap 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 2) mausezahn ddd0 -A 10.10.10.1 -B 10.10.10.2 -c 0 -a own -b 00:c1:a0:c1:a0:00 -t udp & 3) tc qdisc change dev ddd0 handle 10: ets bands 4 strict 2 quanta 2500 2500 priomap 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 crashes systematically when line 2) is commented: list_del corruption, ffff8e028404bd30->next is LIST_POISON1 (dead000000000100) -----[cut here]----- kernel BUG at lib/list_debug.c:47! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 954 Comm: tc Not tainted 5.16.0-rc4+ #478 Hardware name: Red Hat KVM, BIOS 1.11.1-4.module+el8.1.0+4066+0f1aadab 04/01/2014 RIP: 0010: __list_del_entry_valid.cold.1+0x12/0x47 Code: fe ff 0f 0b 48 89 c1 4c 89 c6 48 c7 c7 08 42 1b 87 e8 1d c5 fe ff 0f 0b 48 89 fe 48 89 c2 48 c7 c7 98 42 1b 87 e8 09 c5 fe ff <0f> 0b 48 c7 c7 48 43 1b 87 e8 fb c4 fe ff 0f 0b 48 89 f2 48 89 fe RSP: 0018:ffffae46807a3888 EFLAGS: 00010246 RAX: 000000000000004e RBX: 0000000000000007 RCX: 0000000000000202 RDX: 0000000000000000 RSI: ffffffff871ac536 RDI: 00000000fffffffb RBP: fffffae46807a3a10 R08: 0000000000000000 R09: c0000000ffff7fff R10: 0000000000000001 R11: fffffae46807a36a8 R12: ffff8e028404b800 R13: ffff8e028404bd30 R14: dead000000000100 R15: ffff8e02fafa2400 FS: 00007efdc92e4480(0000) GS:ffff8e02fb600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000682f48 CR3: 00000001058be000 CR4: 0000000000350ef0 Call Trace: <TASK> ets_qdisc_change+0x58b/0xa70 [sch_ets] tc_modify_qdisc +0x323/0x880 rtnetlink_rcv_msg+0x169/0x4a0 netlink_rcv_skb+0x50/0x100 netlink_unicast +0x1a5/0x280 netlink_sendmsg+0x257/0x4d0 sock_sendmsg+0x5b/0x60 ____sys_sendmsg +0x1f2/0x260 __sys_sendmsg+0x7c/0xc0 __sys_sendmsg+0x57/0xa0 do_syscall_64+0x3a/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7efdc8031338 Code: 89 02 48 c7 c0 ff ff ff ff eb b5 0f 1f 80 00 00 00 00 f3 0f 1e fa 48 8d 05 25 43 2c 00 8b 00 85 c0 75 17 b8 2e 00 00 00 0f 05 <48> 3d 00 f0 ff 77 58 c3 0f 1f 80 00 00 00 00 41 54 41 89 d4 55 RSP: 002b:00007ffdf1ce9828 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffff871ac536 RBX: 0000000061b37a97 RCX: 00007efdc8031338 RDX: 0000000000000000 RSI: 00007ffdf1ce9890 RDI: 0000000000000003 RBP: 0000000000000000 R08: 0000000000000001 R09: 000000000078a940 R10: 000000000000000c R11: 0000000000000246 R12: 0000000000000001 R13: 0000000000688880 R14: 0000000000000000 R15: 0000000000000000 </TASK> Modules linked in: sch_ets sch_tbf dummy rkill iTCO_wdt iTCO_vendor_support intel_rapl_msr intel_rapl_common joydev pcspkr i2c_i801 virtio_balloon i2c_smbus lpc_ich ip_tables xfs libcrc32c crct10dif_pclmul crc32_pclmul crc32c_intel serio_raw ghash_clmulni_intel ahci libahci libata virtio_blk virtio_console virtio_net net_failover failover sunrpc dm_mirror dm_region_hash dm_log dm_mod [last unloaded: sch_ets] --- [end trace f35878d1912655c2]--- RIP: 0010: __list_del_entry_valid.cold.1+0x12/0x47 Code: fe ff 0f 0b 48 89 c1 4c 89 c6 48 c7 c7 08 42 1b 87 e8 1d c5 fe ff 0f 0b 48 89 fe 48 89 c2 48 c7 c7 98 42 1b 87 e8 09 c5 fe ff <0f> 0b 48 c7 c7 48 43 1b 87 e8 fb c4 fe ff 0f 0b 48 89 f2 48 89 fe RSP: 0018:ffffae46807a3888 EFLAGS: 00010246 RAX: 000000000000004e RBX: 0000000000000007 RCX: 0000000000000202 RDX: 0000000000000000 RSI: ffffffff871ac536 RDI: 00000000fffffffb RBP: fffffae46807a3a10 R08: 0000000000000000 R09: c0000000ffff7fff R10: 0000000000000001 R11: fffffae46807a36a8 R12: ffff8e028404b800 R13: ffff8e028404bd30 R14: dead000000000100 R15: ffff8e02fafa2400 FS: 00007efdc92e4480(0000) GS:ffff8e02fb600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000 ---truncated---

CVE ID	Description
CVE-2021-47597	In the Linux kernel, the following vulnerability has been resolved: inet_diag: fix kernel-infoleak for UDP sockets KMSAN reported a kernel-infoleak [1], that can exploited by unpriv users. After analysis it turned out UDP was not initializing r->idiag_expires. Other users of inet_sk_diag_fill() might make the same mistake in the future, so fix this in inet_sk_diag_fill(). [1] BUG: KMSAN: kernel-infoleak in instrument_copy_to_user include/linux/instrumented.h:121 [inline] BUG: KMSAN: kernel-infoleak in copyout lib/iov_iter.c:156 [inline] BUG: KMSAN: kernel-infoleak in _copy_to_iter+0x69d/0x25c0 lib/iov_iter.c:670 instrument_copy_to_user include/linux/instrumented.h:121 [inline] copyout lib/iov_iter.c:156 [inline] _copy_to_iter+0x69d/0x25c0 lib/iov_iter.c:670 copy_to_iter include/linux/uio.h:155 [inline] simple_copy_to_iter+0xf3/0x140 net/core/datagram.c:519 __skb_datagram_iter+0x2cb/0x1280 net/core/datagram.c:425 skb_copy_datagram_iter+0xdc/0x270 net/core/datagram.c:533 skb_copy_datagram_msg include/linux/skbuff.h:3657 [inline] netlink_rcvmsg+0x660/0x1c60 net/netlink/af_netlink.c:1974 sock_rcvmsg_nosec net/socket.c:944 [inline] sock_rcvmsg net/socket.c:962 [inline] sock_read_iter+0x5a9/0x630 net/socket.c:1035 call_read_iter include/linux/fs.h:2156 [inline] new_sync_read fs/read_write.c:400 [inline] vfs_read+0x1631/0x1980 fs/read_write.c:481 ksys_read+0x28c/0x520 fs/read_write.c:619 __do_sys_read fs/read_write.c:629 [inline] __se_sys_read fs/read_write.c:627 [inline] __x64_sys_read+0xdb/0x120 fs/read_write.c:627 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae Uninit was created at: slab_post_alloc_hook mm/slab.h:524 [inline] slab_alloc_node mm/slub.c:3251 [inline] __kmallocc_node_track_caller+0xe0c/0x1510 mm/slub.c:4974 kmallocc_reserve net/core/skbuff.c:354 [inline] __alloc_skb+0x545/0xf90 net/core/skbuff.c:426 alloc_skb include/linux/skbuff.h:1126 [inline] netlink_dump+0x3d5/0x16a0 net/netlink/af_netlink.c:2245 __netlink_dump_start+0xd1c/0xee0 net/netlink/af_netlink.c:2370 netlink_dump_start include/linux/netlink.h:254 [inline] inet_diag_handler_cmd+0x2e7/0x400 net/ipv4/inet_diag.c:1343 sock_diag_rcv_msg+0x24a/0x620 netlink_rcv_skb+0x447/0x800 net/netlink/af_netlink.c:2491 sock_diag_rcv+0x63/0x80 net/core/sock_diag.c:276 netlink_unicast_kernel net/netlink/af_netlink.c:1319 [inline] netlink_unicast+0x1095/0x1360 net/netlink/af_netlink.c:1345 netlink_sendmsg+0x16f3/0x1870 net/netlink/af_netlink.c:1916 sock_sendmsg_nosec net/socket.c:704 [inline] sock_sendmsg net/socket.c:724 [inline] sock_write_iter+0x594/0x690 net/socket.c:1057 do_iter_readv_writev+0xa7f/0xc70 do_iter_write+0x52c/0x1500 fs/read_write.c:851 vfs_writv fs/read_write.c:924 [inline] do_writv+0x63f/0xe30 fs/read_write.c:967 __do_sys_writv fs/read_write.c:1040 [inline] __se_sys_writv fs/read_write.c:1037 [inline] __x64_sys_writv+0xe5/0x120 fs/read_write.c:1037 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae Bytes 68-71 of 312 are uninitialized Memory access of size 312 starts at ffff88812ab54000 Data copied to user address 0000000020001440 CPU: 1 PID: 6365 Comm: syz-executor801 Not tainted 5.16.0-rc3-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011
CVE-2021-47600	In the Linux kernel, the following vulnerability has been resolved: dm btree remove: fix use after free in rebalance_children() Move dm_tm_unlock() after dm_tm_dec().
CVE-2021-47603	In the Linux kernel, the following vulnerability has been resolved: audit: improve robustness of the audit queue handling If the audit daemon were ever to get stuck in a stopped state the kernel's kauditd_thread() could get blocked attempting to send audit records to the userspace audit daemon. With the kernel thread blocked it is possible that the audit queue could grow unbounded as certain audit record generating events must be exempt from the queue limits else the system enter a deadlock state. This patch resolves this problem by lowering the kernel thread's socket sending timeout from MAX_SCHEDULE_TIMEOUT to HZ/10 and tweaks the kauditd_send_queue() function to better manage the various audit queues when connection problems occur between the kernel and the audit daemon. With this patch, the backlog may temporarily grow beyond the defined limits when the audit daemon is stopped and the system is under heavy audit pressure, but kauditd_thread() will continue to make progress and drain the queues as it would for other connection problems. For example, with the audit daemon put into a stopped state and the system configured to audit every syscall it was still possible to shutdown the system without a kernel panic, deadlock, etc.; granted, the system was slow to shutdown but that is to be expected given the extreme pressure of recording every syscall. The timeout value of HZ/10 was chosen primarily through experimentation and this developer's "gut feeling". There is likely no one perfect value, but as this scenario is limited in scope (root privileges would be needed to send SIGSTOP to the audit daemon), it is likely not worth exposing this as a tunable at present. This can always be done at a later date if it proves necessary.

CVE ID	Description
CVE-2021-47606	In the Linux kernel, the following vulnerability has been resolved: net: netlink: af_netlink: Prevent empty skb by adding a check on len. Adding a check on len parameter to avoid empty skb. This prevents a division error in netem_enqueue function which is caused when skb->len=0 and skb->data_len=0 in the randomized corruption step as shown below. skb->data[random_u32() % skb_headlen(skb)] ^= 1<<(prandom_u32() % 8); Crash Report: [343.170349] netdevsim netdevsim0 netdevsim3: set [1, 0] type 2 family 0 port 6081 - 0 [343.216110] netem: version 1.3 [343.235841] divide error: 0000 [#1] PREEMPT SMP KASAN NOPTI [343.236680] CPU: 3 PID: 4288 Comm: reproducer Not tainted 5.16.0-rc1+ [343.237569] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.11.0-2.el7 04/01/2014 [343.238707] RIP: 0010:netem_enqueue+0x1590/0x33c0 [sch_netem] [343.239499] Code: 89 85 58 ff ff e8 5f 5d e9 d3 48 8b b5 48 ff ff ff 8b 8d 50 ff ff ff 8b 85 58 ff ff ff 48 8b bd 70 ff ff ff 31 d2 2b 4f 74 <f7> f1 48 b8 00 00 00 00 00 fc ff df 49 01 d5 4c 89 e9 48 c1 e9 03 [343.241883] RSP: 0018:ffff88800bcd7368 EFLAGS: 00010246 [343.242589] RAX: 00000000ba7c0a9c RBX: 0000000000000001 RCX: 0000000000000000 [343.243542] RDX: 0000000000000000 RSI: ffff88800f8edb10 RDI: ffff88800f8eda40 [343.244474] RBP: ffff88800bcd7458 R08: 0000000000000000 R09: ffffffff94fb8445 [343.245403] R10: ffffffff94fb8336 R11: ffffffff94fb8445 R12: 0000000000000000 [343.246355] R13: ffff88800a5a7000 R14: ffff88800a5b5800 R15: 0000000000000002 [343.247291] FS: 00007fdde2bd7700(0000) GS:ffff888109780000(0000) knlGS:0000000000000000 [343.248350] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [343.249120] CR2: 00000000200000c0 CR3: 000000000ef4c000 CR4: 0000000000000060 [343.250076] Call Trace: [343.250423] <TASK> [343.250713] ? memcpy+0x4d/0x60 [343.251162] ? netem_init+0xa0/0xa0 [sch_netem] [343.251795] ? __sanitizer_cov_trace_pc+0x21/0x60 [343.252443] netem_enqueue+0xe28/0x33c0 [sch_netem] [343.253102] ? stack_trace_save+0x87/0xb0 [343.253655] ? filter_irq_stacks+0xb0/0xb0 [343.254220] ? netem_init+0xa0/0xa0 [sch_netem] [343.254837] ? __kasan_check_write+0x14/0x20 [343.255418] ? _raw_spin_lock+0x88/0xd6 [343.255953] dev_qdisc_enqueue+0x50/0x180 [343.256508] __dev_queue_xmit+0x1a7e/0x3090 [343.257083] ? netdev_core_pick_tx+0x300/0x300 [343.257690] ? check_kcov_mode+0x10/0x40 [343.258219] ? _raw_spin_unlock_irqrestore+0x29/0x40 [343.258899] ? __kasan_init_slab_obj+0x24/0x30 [343.259529] ? setup_object.isra.71+0x23/0x90 [343.260121] ? new_slab+0x26e/0x4b0 [343.260609] ? kasan_poison+0x3a/0x50 [343.261118] ? kasan_unpoison+0x28/0x50 [343.261637] ? __kasan_slab_alloc+0x71/0x90 [343.262214] ? memcpy+0x4d/0x60 [343.262674] ? write_comp_data+0x2f/0x90 [343.263209] ? __kasan_check_write+0x14/0x20 [343.263802] ? __skb_clone+0x5d6/0x840 [343.264329] ? __sanitizer_cov_trace_pc+0x21/0x60 [343.264958] dev_queue_xmit+0x1c/0x20 [343.265470] netlink_deliver_tap+0x652/0x9c0 [343.266067] netlink_unicast+0x5a0/0x7f0 [343.266608] ? netlink_attachskb+0x860/0x860 [343.267183] ? __sanitizer_cov_trace_pc+0x21/0x60 [343.267820] ? write_comp_data+0x2f/0x90 [343.268367] netlink_sendmsg+0x922/0xe80 [343.268899] ? netlink_unicast+0x7f0/0x7f0 [343.269472] ? __sanitizer_cov_trace_pc+0x21/0x60 [343.270099] ? write_comp_data+0x2f/0x90 [343.270644] ? netlink_unicast+0x7f0/0x7f0 [343.271210] sock_sendmsg+0x155/0x190 [343.271721] ____sys_sendmsg+0x75f/0x8f0 [343.272262] ? kernel_sendmsg+0x60/0x60 [343.272788] ? write_comp_data+0x2f/0x90 [343.273332] ? write_comp_data+0x2f/0x90 [343.273869] ____sys_sendmsg+0x10f/0x190 [343.274405] ? sendmsg_copy_msghdr+0x80/0x80 [343.274984] ? slab_post_alloc_hook+0x70/0x230 [343.275597] ? futex_wait_setup+0x240/0x240 [343.276175] ? security_file_alloc+0x3e/0x170 [343.276779] ? write_comp_d ---truncated---
CVE-2021-47607	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix kernel address leakage in atomic cmpxchg's r0 aux reg The implementation of BPF_CMPXCHG on a high level has the following parameters: .-[old-val] .-[new-val] BPF_R0 = cmpxchg{32,64}(DST_REG + insn->off, BPF_R0, SRC_REG) ~-[mem-loc] ~-[old-val] Given a BPF insn can only have two registers (dst, src), the R0 is fixed and used as an auxiliary register for input (old value) as well as output (returning old value from memory location). While the verifier performs a number of safety checks, it misses to reject unprivileged programs where R0 contains a pointer as old value. Through brute-forcing it takes about ~16sec on my machine to leak a kernel pointer with BPF_CMPXCHG. The PoC is basically probing for kernel addresses by storing the guessed address into the map slot as a scalar, and using the map value pointer as R0 while SRC_REG has a canary value to detect a matching address. Fix it by checking R0 for pointers, and reject if that's the case for unprivileged programs.

CVE ID	Description
CVE-2021-47608	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix kernel address leakage in atomic fetch The change in commit 37086bdc737 ("bpf: Propagate stack bounds to registers in atomics w/ BPF_FETCH") around check_mem_access() handling is buggy since this would allow for unprivileged users to leak kernel pointers. For example, an atomic fetch/and with -1 on a stack destination which holds a spilled pointer will migrate the spilled register type into a scalar, which can then be exported out of the program (since scalar != pointer) by dumping it into a map value. The original implementation of XADD was preventing this situation by using a double call to check_mem_access() one with BPF_READ and a subsequent one with BPF_WRITE, in both cases passing -1 as a placeholder value instead of register as per XADD semantics since it didn't contain a value fetch. The BPF_READ also included a check in check_stack_read_fixed_off() which rejects the program if the stack slot is of __is_pointer_value() if dst_regno < 0. The latter is to distinguish whether we're dealing with a regular stack spill/ fill or some arithmetical operation which is disallowed on non-scalars, see also 6e7e63cbb023 ("bpf: Forbid XADD on spilled pointers for unprivileged users") for more context on check_mem_access() and its handling of placeholder value -1. One minimally intrusive option to fix the leak is for the BPF_FETCH case to initially check the BPF_READ case via check_mem_access() with -1 as register, followed by the actual load case with non-negative load_reg to propagate stack bounds to registers.
CVE-2021-47616	In the Linux kernel, the following vulnerability has been resolved: RDMA: Fix use-after-free in rxe_queue_cleanup On error handling path in rxe_qp_from_init() qp->sq.queue is freed and then rxe_create_qp() will drop last reference to this object. qp clean up function will try to free this queue one time and it causes UAF bug. Fix it by zeroing queue pointer after freeing queue in rxe_qp_from_init().
CVE-2021-47617	In the Linux kernel, the following vulnerability has been resolved: PCI: pciehp: Fix infinite loop in IRQ handler upon power fault The Power Fault Detected bit in the Slot Status register differs from all other hotplug events in that it is sticky: It can only be cleared after turning off slot power. Per PCIe r5.0, sec. 6.7.1.8: If a power controller detects a main power fault on the hot-plug slot, it must automatically set its internal main power fault latch [...]. The main power fault latch is cleared when software turns off power to the hot-plug slot. The stickiness used to cause interrupt storms and infinite loops which were fixed in 2009 by commits 5651c48cfafe ("PCI pciehp: fix power fault interrupt storm problem") and 99f0169c17f3 ("PCI: pciehp: enable software notification on empty slots"). Unfortunately in 2020 the infinite loop issue was inadvertently reintroduced by commit 8edf5332c393 ("PCI: pciehp: Fix MSI interrupt race"): The hardirq handler pciehp_isr() clears the PFD bit until pciehp's power_fault_detected flag is set. That happens in the IRQ thread pciehp_ist(), which never learns of the event because the hardirq handler is stuck in an infinite loop. Fix by setting the power_fault_detected flag already in the hardirq handler.
CVE-2021-47619	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix queues reservation for XDP When XDP was configured on a system with large number of CPUs and X722 NIC there was a call trace with NULL pointer dereference. i40e 0000:87:00.0: failed to get tracking for 256 queues for VSI 0 err -12 i40e 0000:87:00.0: setup of MAIN VSI failed BUG: kernel NULL pointer dereference, address: 0000000000000000 RIP: 0010:i40e_xdp+0xea/0x1b0 [i40e] Call Trace: ? i40e_reconfig_rss_queues+0x130/0x130 [i40e] dev_xdp_install+0x61/0xe0 dev_xdp_attach+0x18a/0x4c0 dev_change_xdp_fd+0x1e6/0x220 do_setlink+0x616/0x1030 ? ahci_port_stop+0x80/0x80 ? ata_qc_issue+0x107/0x1e0 ? lock_timer_base+0x61/0x80 ? __mod_timer+0x202/0x380 rtnl_setlink+0xe5/0x170 ? bpf_lsm_binder_transaction+0x10/0x10 ? security_capable+0x36/0x50 rtnetlink_rcv_msg+0x121/0x350 ? rtnl_calcit.isra.0+0x100/0x100 netlink_rcv_skb+0x50/0xf0 netlink_unicast+0x1d3/0x2a0 netlink_sendmsg+0x22a/0x440 sock_sendmsg+0x5e/0x60 __sys_sendto+0xf0/0x160 ? __sys_getsockname+0x7e/0xc0 ? __copy_from_user+0x3c/0x80 ? __sys_setsockopt+0xc8/0x1a0 __x64_sys_sendto+0x20/0x30 do_syscall_64+0x33/0x40 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f83fa7a39e0 This was caused by PF queue pile fragmentation due to flow director VSI queue being placed right after main VSI. Because of this main VSI was not able to resize its queue allocation for XDP resulting in no queues allocated for main VSI when XDP was turned on. Fix this by always allocating last queue in PF queue pile for a flow director VSI.
CVE-2021-47620	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: refactor malicious adv data check Check for out-of-bound read was being performed at the end of while num_reports loop, and would fill journal with false positives. Added check to beginning of loop processing so that it doesn't get checked after ptr has been advanced.

CVE ID	Description
CVE-2021-47623	In the Linux kernel, the following vulnerability has been resolved: powerpc/fixmap: Fix VM debug warning on unmap Unmapping a fixmap entry is done by calling <code>__set_fixmap()</code> with <code>FIXMAP_PAGE_CLEAR</code> as flags. Today, powerpc <code>__set_fixmap()</code> calls <code>map_kernel_page()</code>. <code>map_kernel_page()</code> is not happy when called a second time for the same page. WARNING: CPU: 0 PID: 1 at arch/powerpc/mm/pgtable.c:194 set_pte_at+0xc/0x1e8 CPU: 0 PID: 1 Comm: swapper Not tainted 5.16.0-rc3-s3k-dev-01993-g350ff07feb7d-dirty #682 NIP: c0017cd4 LR: c00187f0 CTR: 00000010 REGS: e1011d50 TRAP: 0700 Not tainted (5.16.0-rc3-s3k-dev-01993-g350ff07feb7d-dirty) MSR: 00029032 <EE,ME,IR,DR,RI> CR: 42000208 XER: 00000000 GPR00: c0165fec e1011e10 c14c0000 c0ee2550 ff800000 c0f3d000 00000000 c001686c GPR08: 00001000 b00045a9 00000001 c0f58460 c0f50000 00000000 c0007e10 00000000 GPR16: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 GPR24: 00000000 00000000 c0ee2550 00000000 c0f57000 00000ff8 00000000 ff800000 NIP [c0017cd4] set_pte_at+0xc/0x1e8 LR [c00187f0] map_kernel_page+0x9c/0x100 Call Trace: [e1011e10] [c0736c68] vsnprintf+0x358/0x6c8 (unreliable) [e1011e30] [c0165fec] __set_fixmap+0x30/0x44 [e1011e40] [c0c13bdc] early_iounmap+0x11c/0x170 [e1011e70] [c0c06cb0] ioremap_legacy_serial_console+0x88/0xc0 [e1011e90] [c0c03634] do_one_initcall+0x80/0x178 [e1011ef0] [c0c0385c] kernel_init_freeable+0xb4/0x250 [e1011f20] [c0007e34] kernel_init+0x24/0x140 [e1011f30] [c0016268] ret_from_kernel_thread+0x5c/0x64 Instruction dump: 7fe3fb78 48019689 80010014 7c630034 83e1000c 5463d97e 7c0803a6 38210010 4e800020 81250000 712a0001 41820008 <0fe00000> 9421ffe0 93e1001c 48000030 Implement unmap_kernel_page() which clears an existing pte.
CVE-2021-47634	In the Linux kernel, the following vulnerability has been resolved: ubi: Fix race condition between <code>ctrl_cdev_ioctl</code> and <code>ubi_cdev_ioctl</code> Hulk Robot reported a KASAN report about use-after-free: ===== BUG: KASAN: use-after-free in <code>__list_del_entry_valid+0x13d/0x160</code> Read of size 8 at addr ffff888035e37d98 by task ubiattach/1385 [...] Call Trace: klist_dec_and_del+0xa7/0x4a0 klist_put+0xc7/0x1a0 device_del+0x4d4/0xed0 cdev_device_del+0x1a/0x80 ubi_attach_mtd_dev+0x2951/0x34b0 [ubi] ctrl_cdev_ioctl+0x286/0x2f0 [ubi] Allocated by task 1414: device_add+0x60a/0x18b0 cdev_device_add+0x103/0x170 ubi_create_volume+0x1118/0x1a10 [ubi] ubi_cdev_ioctl+0xb7f/0x1ba0 [ubi] Freed by task 1385: cdev_device_del+0x1a/0x80 ubi_remove_volume+0x438/0x6c0 [ubi] ubi_cdev_ioctl+0xbf4/0x1ba0 [ubi] [...] ===== The lock held by <code>ctrl_cdev_ioctl</code> is <code>ubi_devices_mutex</code>, but the lock held by <code>ubi_cdev_ioctl</code> is <code>ubi->device_mutex</code>. Therefore, the two locks can be concurrent. <code>ctrl_cdev_ioctl</code> contains two operations: <code>ubi_attach</code> and <code>ubi_detach</code>. <code>ubi_detach</code> is bug-free because it uses reference counting to prevent concurrency. However, <code>uif_init</code> and <code>uif_close</code> in <code>ubi_attach</code> may race with <code>ubi_cdev_ioctl</code>. <code>uif_init</code> will race with <code>ubi_cdev_ioctl</code> as in the following stack. cpu1 cpu2 cpu3 _____ _____ ctrl_cdev_ioctl ubi_attach_mtd_dev uif_init ubi_cdev_ioctl ubi_create_volume cdev_device_add ubi_add_volume // sysfs exist kill_volumes ubi_cdev_ioctl ubi_remove_volume cdev_device_del // first free ubi_free_volume cdev_del // double free cdev_device_del And uif_close will race with ubi_cdev_ioctl as in the following stack. cpu1 cpu2 cpu3 _____ _____ ctrl_cdev_ioctl ubi_attach_mtd_dev uif_init ubi_cdev_ioctl ubi_create_volume cdev_device_add ubi_debugfs_init_dev //error goto out_uif; uif_close kill_volumes ubi_cdev_ioctl ubi_remove_volume cdev_device_del // first free ubi_free_volume // double free The cause of this problem is that commit 714fb87e8bc0 make device "available" before it becomes accessible via sysfs. Therefore, we roll back the modification. We will fix the race condition between ubi device creation and udev by removing <code>ubi_get_device</code> in <code>vol_attribute_show</code> and <code>dev_attribute_show</code>. This avoids accessing uninitialized <code>ubi_devices[ubi_num]</code>. <code>ubi_get_device</code> is used to prevent devices from being deleted during sysfs execution. However, now kernfs ensures that devices will not be deleted before all reference counting are released. The key process is shown in the following stack. device_del device_remove_attrs device_remove_groups sysfs_remove_groups sysfs_remove_group remove_files kernfs_remove_by_name kernfs_remove_by_name_ns __kernfs_remove kernfs_drain

CVE ID	Description
CVE-2021-47639	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Zap _all_ roots when unmapping gfn range in TDP MMU Zap both valid and invalid roots when zapping/ unmapping a gfn range, as KVM must ensure it holds no references to the freed page after returning from the unmap operation. Most notably, the TDP MMU doesn't zap invalid roots in mmu_notifier callbacks. This leads to use-after-free and other issues if the mmu_notifier runs to completion while an invalid root zipper yields as KVM fails to honor the requirement that there must be _no_ references to the page after the mmu_notifier returns. The bug is most easily reproduced by hacking KVM to cause a collision between set_nx_huge_pages() and kvm_mmu_notifier_release(), but the bug exists between kvm_mmu_notifier_invalidate_range_start() and memslot updates as well. Invalidating a root ensures pages aren't accessible by the guest, and KVM won't read or write page data itself, but KVM will trigger e.g. kvm_set_pfn_dirty() when zapping SPTes, and thus completing a zap of an invalid root _after_ the mmu_notifier returns is fatal. WARNING: CPU: 24 PID: 1496 at arch/x86/kvm/./../virt/kvm/kvm_main.c:173 [kvm] RIP: 0010:kvm_is_zone_device_pfn+0x96/0xa0 [kvm] Call Trace: <TASK> kvm_set_pfn_dirty+0xa8/0xe0 [kvm] __handle_changed_spte+0x2ab/0x5e0 [kvm] __handle_changed_spte+0x2ab/0x5e0 [kvm] __handle_changed_spte+0x2ab/0x5e0 [kvm] zap_gfn_range+0x1f3/0x310 [kvm] kvm_tdp_mmu_zap_invalidated_roots+0x50/0x90 [kvm] kvm_mmu_zap_all_fast+0x177/0x1a0 [kvm] set_nx_huge_pages+0xb4/0x190 [kvm] param_attr_store+0x70/0x100 module_attr_store+0x19/0x30 kernfs_fop_write_iter+0x119/0x1b0 new_sync_write+0x11c/0x1b0 vfs_write+0x1cc/0x270 ksys_write+0x5f/0xe0 do_syscall_64+0x38/0xc0 entry_SYSCALL_64_after_hwframe+0x44/0xae </TASK>
CVE-2021-47646	In the Linux kernel, the following vulnerability has been resolved: Revert "Revert "block, bfq: honor already-setup queue merges"" A crash [1] happened to be triggered in conjunction with commit 2d52c58b9c9b ("block, bfq: honor already-setup queue merges"). The latter was then reverted by commit ebc69e897e17 ("Revert "block, bfq: honor already-setup queue merges""). Yet, the reverted commit was not the one introducing the bug. In fact, it actually triggered a UAF introduced by a different commit, and now fixed by commit d29bd41428cf ("block, bfq: reset last_bfqq_created on group change"). So, there is no point in keeping commit 2d52c58b9c9b ("block, bfq: honor already-setup queue merges") out. This commit restores it. [1] https://bugzilla.kernel.org/show_bug.cgi?id=214503
CVE-2021-47650	In the Linux kernel, the following vulnerability has been resolved: ASoC: soc-compress: prevent the potentially use of null pointer There is one call trace that snd_soc_register_card()->snd_soc_bind_card()->soc_init_pcm_runtime()->snd_soc_dai_compress_new()->snd_soc_new_compress(). In the trace the 'codec_dai' transfers from card->dai_link, and we can see from the snd_soc_add_pcm_runtime() in snd_soc_bind_card() that, if value of card->dai_link->num_codecs is 0, then 'codec_dai' could be null pointer caused by index out of bound in 'asoc_rtd_to_codec(rtd, 0)'. And snd_soc_register_card() is called by various platforms. Therefore, it is better to add the check in the case of misusing. And because 'cpu_dai' has already checked in soc_init_pcm_runtime(), there is no need to check again. Adding the check as follow, then if 'codec_dai' is null, snd_soc_new_compress() will not pass through the check 'if (playback + capture != 1)', avoiding the leftover use of 'codec_dai'.
CVE-2021-47657	In the Linux kernel, the following vulnerability has been resolved: drm/virtio: Ensure that objs is not NULL in virtio_gpu_array_put_free() If virtio_gpu_object_shmem_init() fails (e.g. due to fault injection, as it happened in the bug report by syzbot), virtio_gpu_array_put_free() could be called with objs equal to NULL. Ensure that objs is not NULL in virtio_gpu_array_put_free(), or otherwise return from the function.
CVE-2021-47659	In the Linux kernel, the following vulnerability has been resolved: drm/plane: Move range check for format_count earlier While the check for format_count > 64 in __drm_universal_plane_init() shouldn't be hit (it's a WARN_ON), in its current position it will then leak the plane->format_types array and fail to call drm_mode_object_unregister() leaking the modeset identifier. Move it to the start of the function to avoid allocating those resources in the first place.
CVE-2021-47668	In the Linux kernel, the following vulnerability has been resolved: can: dev: can_restart: fix use after free bug After calling netif_rx_ni(skb), dereferencing skb is unsafe. Especially, the can_frame cf which aliases skb memory is accessed after the netif_rx_ni() in: stats->rx_bytes += cf->len; Reordering the lines solves the issue.
CVE-2022-0171	A flaw was found in the Linux kernel. The existing KVM SEV API has a vulnerability that allows a non-root (host) user-level application to crash the host kernel by creating a confidential guest VM instance in AMD CPU that supports Secure Encrypted Virtualization (SEV).
CVE-2022-0235	node-fetch is vulnerable to Exposure of Sensitive Information to an Unauthorized Actor
CVE-2022-0264	A vulnerability was found in the Linux kernel's eBPF verifier when handling internal data structures. Internal memory locations could be returned to userspace. A local attacker with the permissions to insert eBPF code to the kernel can use this to leak internal kernel memory details defeating some of the exploit mitigations in place for the kernel. This flaws affects kernel versions < v5.16-rc6

CVE ID	Description
CVE-2022-0351	Access of Memory Location Before Start of Buffer in GitHub repository vim/vim prior to 8.2.
CVE-2022-0391	A flaw was found in Python, specifically within the urllib.parse module. This module helps break Uniform Resource Locator (URL) strings into components. The issue involves how the urlparse method does not sanitize input and allows characters like '\r' and '\n' in the URL path. This flaw allows an attacker to input a crafted URL, leading to injection attacks. This flaw affects Python versions prior to 3.10.0b1, 3.9.5, 3.8.11, 3.7.11 and 3.6.14.
CVE-2022-0400	An out-of-bounds read vulnerability was discovered in linux kernel in the smc protocol stack, causing remote dos.
CVE-2022-0480	A flaw was found in the filelock_init in fs/locks.c function in the Linux kernel. This issue can lead to host memory exhaustion due to memcg not limiting the number of Portable Operating System Interface (POSIX) file locks.
CVE-2022-0529	A flaw was found in Unzip. The vulnerability occurs during the conversion of a wide string to a local string that leads to a heap of out-of-bound write. This flaw allows an attacker to input a specially crafted zip file, leading to a crash or code execution.
CVE-2022-0530	A flaw was found in Unzip. The vulnerability occurs during the conversion of a wide string to a local string that leads to a heap of out-of-bound write. This flaw allows an attacker to input a specially crafted zip file, leading to a crash or code execution.
CVE-2022-0563	A flaw was found in the util-linux chfn and chsh utilities when compiled with Readline support. The Readline library uses an "INPUTRC" environment variable to get a path to the library config file. When the library cannot parse the specified file, it prints an error message containing data from the file. This flaw allows an unprivileged user to read root-owned files, potentially leading to privilege escalation. This flaw affects util-linux versions prior to 2.37.4.
CVE-2022-0995	An out-of-bounds (OOB) memory write flaw was found in the Linux kernel's watch_queue event notification subsystem. This flaw can overwrite parts of the kernel state, potentially allowing a local user to gain privileged access or cause a denial of service on the system.
CVE-2022-1056	Out-of-bounds Read error in tiffcrop in libtiff 4.3.0 allows attackers to cause a denial-of-service via a crafted tiff file. For users that compile libtiff from sources, the fix is available with commit 46dc8fcd.
CVE-2022-1247	An issue found in linux-kernel that leads to a race condition in rose_connect(). The rose driver uses rose_neigh->use to represent how many objects are using the rose_neigh. When a user wants to delete a rose_route via rose_ioctl(), the rose driver calls rose_del_node() and removes neighbours only if their "count" and "use" are zero.
CVE-2022-1354	A heap buffer overflow flaw was found in Libtiffs' tiffinfo.c in TIFFReadRawDataStriped() function. This flaw allows an attacker to pass a crafted TIFF file to the tiffinfo tool, triggering a heap buffer overflow issue and causing a crash that leads to a denial of service.
CVE-2022-1471	SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConstructor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond.
CVE-2022-1619	Heap-based Buffer Overflow in function cmdline_erase_chars in GitHub repository vim/vim prior to 8.2.4899. This vulnerabilities are capable of crashing software, modify memory, and possible remote execution
CVE-2022-1665	A set of pre-production kernel packages of Red Hat Enterprise Linux for IBM Power architecture can be booted by the grub in Secure Boot mode even though it shouldn't. These kernel builds don't have the secure boot lockdown patches applied to it and can bypass the secure boot validations, allowing the attacker to load another non-trusted code.
CVE-2022-1720	Buffer Over-read in function grab_file_name in GitHub repository vim/vim prior to 8.2.4956. This vulnerability is capable of crashing the software, memory modification, and possible remote execution.
CVE-2022-20154	In lock_sock_nested of sock.c, there is a possible use after free due to a race condition. This could lead to local escalation of privilege with System execution privileges needed. User interaction is not needed for exploitation.Product: AndroidVersions: Android kernelAndroid ID: A-174846563References: Upstream kernel
CVE-2022-2047	In Eclipse Jetty versions 9.4.0 thru 9.4.46, and 10.0.0 thru 10.0.9, and 11.0.0 thru 11.0.9 versions, the parsing of the authority segment of an http scheme URI, the Jetty HttpURI class improperly detects an invalid input as a hostname. This can lead to failures in a Proxy scenario.

CVE ID	Description
CVE-2022-20566	In l2cap_chan_put of l2cap_core, there is a possible use after free due to improper locking. This could lead to local escalation of privilege with no additional execution privileges needed. User interaction is not needed for exploitation.Product: AndroidVersions: Android kernelAndroid ID: A-165329981References: Upstream kernel
CVE-2022-2124	Buffer Over-read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2125	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 8.2.
CVE-2022-2126	Out-of-bounds Read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2129	Out-of-bounds Write in GitHub repository vim/vim prior to 8.2.
CVE-2022-2175	Buffer Over-read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2182	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 8.2.
CVE-2022-2183	Out-of-bounds Read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2206	Out-of-bounds Read in GitHub repository vim/vim prior to 8.2.
CVE-2022-2207	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 8.2.
CVE-2022-2208	NULL Pointer Dereference in GitHub repository vim/vim prior to 8.2.5163.
CVE-2022-2210	Out-of-bounds Write in GitHub repository vim/vim prior to 8.2.
CVE-2022-2284	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.
CVE-2022-2285	Integer Overflow or Wraparound in GitHub repository vim/vim prior to 9.0.
CVE-2022-2286	Out-of-bounds Read in GitHub repository vim/vim prior to 9.0.
CVE-2022-2287	Out-of-bounds Read in GitHub repository vim/vim prior to 9.0.
CVE-2022-2343	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0044.
CVE-2022-2344	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0045.
CVE-2022-2345	Use After Free in GitHub repository vim/vim prior to 9.0.0046.
CVE-2022-23990	Expat (aka libexpat) before 2.4.4 has an integer overflow in the doProlog function.
CVE-2022-24329	In JetBrains Kotlin before 1.6.0, it was not possible to lock dependencies for Multiplatform Gradle Projects.
CVE-2022-24795	yajl-ruby is a C binding to the YAJL JSON parsing and generation library. The 1.x branch and the 2.x branch of `yajl` contain an integer overflow which leads to subsequent heap memory corruption when dealing with large (~2GB) inputs. The reallocation logic at `yajl_buf.c#L64` may result in the `need` 32bit integer wrapping to 0 when `need` approaches a value of 0x80000000 (i.e. ~2GB of data), which results in a reallocation of buf->alloc into a small heap chunk. These integers are declared as `size_t` in the 2.x branch of `yajl`, which practically prevents the issue from triggering on 64bit platforms, however this does not preclude this issue triggering on 32bit builds on which `size_t` is a 32bit integer. Subsequent population of this under-allocated heap chunk is based on the original buffer size, leading to heap memory corruption. This vulnerability mostly impacts process availability. Maintainers believe exploitation for arbitrary code execution is unlikely. A patch is available and anticipated to be part of yajl-ruby version 1.4.2. As a workaround, avoid passing large inputs to YAJL.
CVE-2022-25168	Apache Hadoop's FileUtil.unTar(File, File) API does not escape the input file name before being passed to the shell. An attacker can inject arbitrary commands. This is only used in Hadoop 3.3 InMemoryAliasMap.completeBootstrapTransfer, which is only ever run by a local user. It has been used in Hadoop 2.x for yarn localization, which does enable remote code execution. It is used in Apache Spark, from the SQL command ADD ARCHIVE. As the ADD ARCHIVE command adds new binaries to the classpath, being able to execute shell scripts does not confer new permissions to the caller. SPARK-38305. "Check existence of file before untarring/zippping", which is included in 3.3.0, 3.1.4, 3.2.2, prevents shell commands being executed, regardless of which version of the hadoop libraries are in use. Users should upgrade to Apache Hadoop 2.10.2, 3.2.4, 3.3.3 or upper (including HADOOP-18136).
CVE-2022-2522	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0061.

CVE ID	Description
CVE-2022-25836	Bluetooth® Low Energy Pairing in Bluetooth Core Specification v4.0 through v5.3 may permit an unauthenticated MITM to acquire credentials with two pairing devices via adjacent access when the MITM negotiates Legacy Passkey Pairing with the pairing Initiator and Secure Connections Passkey Pairing with the pairing Responder and brute forces the Passkey entered by the user into the Initiator. The MITM attacker can use the identified Passkey value to complete authentication with the Responder via Bluetooth pairing method confusion.
CVE-2022-27664	In net/http in Go before 1.18.6 and 1.19.x before 1.19.1, attackers can cause a denial of service because an HTTP/2 connection can hang during closing if shutdown were preempted by a fatal error.
CVE-2022-27943	libiberty/rust-demangle.c in GNU GCC 11.2 allows stack consumption in demangle_const, as demonstrated by nm-new.
CVE-2022-2819	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0211.
CVE-2022-2845	Improper Validation of Specified Quantity in Input in GitHub repository vim/vim prior to 9.0.0218.
CVE-2022-2849	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0220.
CVE-2022-2879	Reader.Read does not set a limit on the maximum size of file headers. A maliciously crafted archive could cause Read to allocate unbounded amounts of memory, potentially causing resource exhaustion or panics. After fix, Reader.Read limits the maximum size of header blocks to 1 MiB.
CVE-2022-2880	Requests forwarded by ReverseProxy include the raw query parameters from the inbound request, including unparsable parameters rejected by net/http. This could permit query parameter smuggling when a Go proxy forwards a parameter with an unparsable value. After fix, ReverseProxy sanitizes the query parameters in the forwarded query when the outbound request's Form field is set after the ReverseProxy. Director function returns, indicating that the proxy has parsed the query parameters. Proxies which do not parse query parameters continue to forward the original query parameters unchanged.
CVE-2022-2905	An out-of-bounds memory read flaw was found in the Linux kernel's BPF subsystem in how a user calls the bpf_tail_call function with a key larger than the max_entries of the map. This flaw allows a local user to gain unauthorized access to data.
CVE-2022-2923	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.0240.
CVE-2022-2946	Use After Free in GitHub repository vim/vim prior to 9.0.0246.
CVE-2022-2961	A use-after-free flaw was found in the Linux kernel's PLP Rose functionality in the way a user triggers a race condition by calling bind while simultaneously triggering the rose_bind() function. This flaw allows a local user to crash or potentially escalate their privileges on the system.
CVE-2022-2963	A vulnerability found in jasper. This security vulnerability happens because of a memory leak bug in function cmdopts_parse that can cause a crash or segmentation fault.
CVE-2022-2980	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.0259.
CVE-2022-3037	Use After Free in GitHub repository vim/vim prior to 9.0.0322.
CVE-2022-31015	Waitress is a Web Server Gateway Interface server for Python 2 and 3. Waitress versions 2.1.0 and 2.1.1 may terminate early due to a thread closing a socket while the main thread is about to call select(). This will lead to the main thread raising an exception that is not handled and then causing the entire application to be killed. This issue has been fixed in Waitress 2.1.2 by no longer allowing the WSGI thread to close the socket. Instead, that is always delegated to the main thread. There is no work-around for this issue. However, users using waitress behind a reverse proxy server are less likely to have issues if the reverse proxy always reads the full response.

CVE ID	Description
CVE-2022-31023	Play Framework is a web framework for Java and Scala. Versions prior to 2.8.16 are vulnerable to generation of error messages containing sensitive information. Play Framework, when run in dev mode, shows verbose errors for easy debugging, including an exception stack trace. Play does this by configuring its <code>DefaultExceptionHandler</code> to do so based on the application mode. In its Scala API Play also provides a static object <code>DefaultExceptionHandler</code> that is configured to always show verbose errors. This is used as a default value in some Play APIs, so it is possible to inadvertently use this version in production. It is also possible to improperly configure the <code>DefaultExceptionHandler</code> object instance as the injected error handler. Both of these situations could result in verbose errors displaying to users in a production application, which could expose sensitive information from the application. In particular, the constructor for <code>CORSFilter</code> and <code>apply</code> method for <code>CORSAActionBuilder</code> use the static object <code>DefaultExceptionHandler</code> as a default value. This is patched in Play Framework 2.8.16. The <code>DefaultExceptionHandler</code> object has been changed to use the prod-mode behavior, and <code>DevExceptionHandler</code> has been introduced for the dev-mode behavior. A workaround is available. When constructing a <code>CORSFilter</code> or <code>CORSAActionBuilder</code> , ensure that a properly-configured error handler is passed. Generally this should be done by using the <code>ExceptionHandler</code> instance provided through dependency injection or through Play's <code>BuiltInComponents</code> . Ensure that the application is not using the <code>DefaultExceptionHandler</code> static object in any code that may be run in production.
CVE-2022-31160	jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of jQuery. Versions prior to 1.13.2 are potentially vulnerable to cross-site scripting. Initializing a checkboxradio widget on an input enclosed within a label makes that parent label contents considered as the input label. Calling <code>.checkboxradio("refresh")</code> on such a widget and the initial HTML contained encoded HTML entities will make them erroneously get decoded. This can lead to potentially executing JavaScript code. The bug has been patched in jQuery UI 1.13.2. To remediate the issue, someone who can change the initial HTML can wrap all the non-input contents of the <code>label</code> in a <code>span</code> .
CVE-2022-3153	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.0404.
CVE-2022-3171	A parsing issue with binary data in protobuf-java core and lite versions prior to 3.21.7, 3.20.3, 3.19.6 and 3.16.3 can lead to a denial of service attack. Inputs containing multiple instances of non-repeated embedded messages with repeated or unknown fields causes objects to be converted back-n-forth between mutable and immutable forms, resulting in potentially long garbage collection pauses. We recommend updating to the versions mentioned above.
CVE-2022-32149	An attacker may cause a denial of service by crafting an Accept-Language header which ParseAcceptLanguage will take significant time to parse.
CVE-2022-3219	GnuPG can be made to spin on a relatively small input by (for example) crafting a public key with thousands of signatures attached, compressed down to just a few KB.
CVE-2022-32190	JoinPath and URL.JoinPath do not remove <code>../</code> path elements appended to a relative path. For example, <code>JoinPath("https://go.dev", "../go")</code> returns the URL <code>"https://go.dev/./go"</code> , despite the <code>JoinPath</code> documentation stating that <code>../</code> path elements are removed from the result.
CVE-2022-3234	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0483.
CVE-2022-3235	Use After Free in GitHub repository vim/vim prior to 9.0.0490.
CVE-2022-3238	A double-free flaw was found in the Linux kernel's NTFS3 subsystem in how a user triggers remount and umount simultaneously. This flaw allows a local user to crash or potentially escalate their privileges on the system.
CVE-2022-3256	Use After Free in GitHub repository vim/vim prior to 9.0.0530.
CVE-2022-3296	Stack-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.0577.
CVE-2022-33064	An off-by-one error in function <code>wav_read_header</code> in <code>src/wav.c</code> in <code>Libsndfile 1.1.0</code> , results in a write out of bound, which allows an attacker to execute arbitrary code, Denial of Service or other unspecified impacts.
CVE-2022-33070	Protobuf-c v1.4.0 was discovered to contain an invalid arithmetic shift via the function <code>parse_tag_and_wiretype</code> in <code>protobuf-c/protobuf-c.c</code> . This vulnerability allows attackers to cause a Denial of Service (DoS) via unspecified vectors.
CVE-2022-3344	A flaw was found in the KVM's AMD nested virtualization (SVM). A malicious L1 guest could purposely fail to intercept the shutdown of a cooperative nested guest (L2), possibly leading to a page fault and kernel panic in the host (L0).
CVE-2022-3352	Use After Free in GitHub repository vim/vim prior to 9.0.0614.

CVE ID	Description
CVE-2022-3424	A use-after-free flaw was found in the Linux kernel's SGI GRU driver in the way the first <code>gru_file_unlocked_ioctl</code> function is called by the user, where a fail pass occurs in the <code>gru_check_chiplet_assignment</code> function. This flaw allows a local user to crash or potentially escalate their privileges on the system.
CVE-2022-3509	A parsing issue similar to CVE-2022-3171, but with <code>textformat</code> in <code>protobuf-java</code> core and lite versions prior to 3.21.7, 3.20.3, 3.19.6 and 3.16.3 can lead to a denial of service attack. Inputs containing multiple instances of non-repeated embedded messages with repeated or unknown fields causes objects to be converted back-n-forth between mutable and immutable forms, resulting in potentially long garbage collection pauses. We recommend updating to the versions mentioned above.
CVE-2022-3510	A parsing issue similar to CVE-2022-3171, but with <code>Message-Type Extensions</code> in <code>protobuf-java</code> core and lite versions prior to 3.21.7, 3.20.3, 3.19.6 and 3.16.3 can lead to a denial of service attack. Inputs containing multiple instances of non-repeated embedded messages with repeated or unknown fields causes objects to be converted back-n-forth between mutable and immutable forms, resulting in potentially long garbage collection pauses. We recommend updating to the versions mentioned above.
CVE-2022-3534	A vulnerability classified as critical has been found in Linux Kernel. Affected is the function <code>btf_dump_name_dups</code> of the file <code>tools/lib/bpf/btf_dump.c</code> of the component <code>libbpf</code> . The manipulation leads to use after free. It is recommended to apply a patch to fix this issue. The identifier of this vulnerability is VDB-211032.
CVE-2022-3570	Multiple heap buffer overflows in <code>tiffcrop.c</code> utility in <code>libtiff</code> library Version 4.4.0 allows attacker to trigger unsafe or out of bounds memory access via crafted TIFF image file which could result into application crash, potential information disclosure or any other context-dependent impact
CVE-2022-3598	LibTIFF 4.4.0 has an out-of-bounds write in <code>extractContigSamplesShifted24bits</code> in <code>tools/tiffcrop.c:3604</code> , allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile <code>libtiff</code> from sources, the fix is available with commit <code>cffb883b</code> .
CVE-2022-3599	LibTIFF 4.4.0 has an out-of-bounds read in <code>writeSingleSection</code> in <code>tools/tiffcrop.c:7345</code> , allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile <code>libtiff</code> from sources, the fix is available with commit <code>e8131125</code> .
CVE-2022-3606	A vulnerability was found in Linux Kernel. It has been classified as problematic. This affects the function <code>find_prog_by_sec_insn</code> of the file <code>tools/lib/bpf/libbpf.c</code> of the component <code>BPF</code> . The manipulation leads to null pointer dereference. It is recommended to apply a patch to fix this issue. The identifier VDB-211749 was assigned to this vulnerability.
CVE-2022-36280	An out-of-bounds(OOB) memory access vulnerability was found in <code>vmwgfx</code> driver in <code>drivers/gpu/vmxgfx/vmxgfx_kms.c</code> in GPU component in the Linux kernel with device file <code>"/dev/dri/renderD128 (or Dxxx)"</code> . This flaw allows a local attacker with a user account on the system to gain privilege, causing a denial of service(DoS).
CVE-2022-3705	A vulnerability was found in <code>vim</code> and classified as problematic. Affected by this issue is the function <code>qf_update_buffer</code> of the file <code>quickfix.c</code> of the component <code>autocmd Handler</code> . The manipulation leads to use after free. The attack may be launched remotely. Upgrading to version 9.0.0805 is able to address this issue. The name of the patch is <code>d0fab10ed2a86698937e3c3fed2f10bd9bb5e731</code> . It is recommended to upgrade the affected component. The identifier of this vulnerability is VDB-212324.
CVE-2022-38533	In GNU Binutils before 2.40, there is a heap-buffer-overflow in the error function <code>bfd_getl32</code> when called from the <code>strip_main</code> function in <code>strip-new</code> via a crafted file.
CVE-2022-3903	An incorrect read request flaw was found in the Infrared Transceiver USB driver in the Linux kernel. This issue occurs when a user attaches a malicious USB device. A local user could use this flaw to starve the resources, causing denial of service or potentially crashing the system.
CVE-2022-39327	Azure CLI is the command-line interface for Microsoft Azure. In versions previous to 2.40.0, Azure CLI contains a vulnerability for potential code injection. Critical scenarios are where a hosting machine runs an Azure CLI command where parameter values have been provided by an external source. The vulnerability is only applicable when the Azure CLI command is run on a Windows machine and with any version of PowerShell and when the parameter value contains the <code>'&'</code> or <code>' '</code> symbols. If any of these prerequisites are not met, this vulnerability is not applicable. Users should upgrade to version 2.40.0 or greater to receive a mitigation for the vulnerability.

CVE ID	Description
CVE-2022-39379	Fluentd collects events from various data sources and writes them to files, RDBMS, NoSQL, IaaS, SaaS, Hadoop and so on. A remote code execution (RCE) vulnerability in non-default configurations of Fluentd allows unauthenticated attackers to execute arbitrary code via specially crafted JSON payloads. Fluentd setups are only affected if the environment variable `FLUENT_OJ_OPTION_MODE` is explicitly set to `object`. Please note: The option FLUENT_OJ_OPTION_MODE was introduced in Fluentd version 1.13.2. Earlier versions of Fluentd are not affected by this vulnerability. This issue was patched in version 1.15.3. As a workaround do not use `FLUENT_OJ_OPTION_MODE=object`.
CVE-2022-3964	A vulnerability classified as problematic has been found in ffmpeg. This affects an unknown part of the file libavcodec/rpzaenc.c of the component QuickTime RPZA Video Encoder. The manipulation of the argument y_size leads to out-of-bounds read. It is possible to initiate the attack remotely. The name of the patch is 92f9b28ed84a77138105475beba16c146bdaf984. It is recommended to apply a patch to fix this issue. The associated identifier of this vulnerability is VDB-213543.
CVE-2022-3965	A vulnerability classified as problematic was found in ffmpeg. This vulnerability affects the function smc_encode_stream of the file libavcodec/smcenc.c of the component QuickTime Graphics Video Encoder. The manipulation of the argument y_size leads to out-of-bounds read. The attack can be initiated remotely. The name of the patch is 13c13109759090b7f7182480d075e13b36ed8edd. It is recommended to apply a patch to fix this issue. The identifier of this vulnerability is VDB-213544.
CVE-2022-40090	An issue was discovered in function TIFFReadDirectory libtiff before 4.4.0 allows attackers to cause a denial of service via crafted TIFF file.
CVE-2022-40755	JasPer 3.0.6 allows denial of service via a reachable assertion in the function inttobits in libjasper/base/jas_image.c.
CVE-2022-40897	Python Packaging Authority (PyPA) setuptools before 65.5.1 allows remote attackers to cause a denial of service via HTML in a crafted package or custom PackageIndex page. There is a Regular Expression Denial of Service (ReDoS) in package_index.py.
CVE-2022-41404	An issue in the fetch() method in the BasicProfile class of org.ini4j before v0.5.4 allows attackers to cause a Denial of Service (DoS) via unspecified vectors.
CVE-2022-41409	Integer overflow vulnerability in pcre2test before 10.41 allows attackers to cause a denial of service or other unspecified impacts via negative input.
CVE-2022-41715	Programs which compile regular expressions from untrusted sources may be vulnerable to memory exhaustion or denial of service. The parsed regexp representation is linear in the size of the input, but in some cases the constant factor can be as high as 40,000, making relatively small regexps consume much larger amounts of memory. After fix, each regexp being parsed is limited to a 256 MB memory footprint. Regular expressions whose representation would use more space than that are rejected. Normal use of regular expressions is unaffected.
CVE-2022-41716	Due to unsanitized NUL values, attackers may be able to maliciously set environment variables on Windows. In syscall.StartProcess and os/exec.Cmd, invalid environment variable values containing NUL values are not properly checked for. A malicious environment variable value can exploit this behavior to set a value for a different environment variable. For example, the environment variable string "A=B \x00C=D" sets the variables "A=B" and "C=D".
CVE-2022-41717	An attacker can cause excessive memory growth in a Go server accepting HTTP/2 requests. HTTP/2 server connections contain a cache of HTTP header keys sent by the client. While the total number of entries in this cache is capped, an attacker sending very large keys can cause the server to allocate approximately 64 MiB per open connection.
CVE-2022-41720	On Windows, restricted files can be accessed via os.DirFS and http.Dir. The os.DirFS function and http.Dir type provide access to a tree of files rooted at a given directory. These functions permit access to Windows device files under that root. For example, os.DirFS("C:/tmp").Open("COM1") opens the COM1 device. Both os.DirFS and http.Dir only provide read-only filesystem access. In addition, on Windows, an os.DirFS for the directory (the root of the current drive) can permit a maliciously crafted path to escape from the drive and access any path on the system. With fix applied, the behavior of os.DirFS("") has changed. Previously, an empty root was treated equivalently to "/", so os.DirFS("").Open("tmp") would open the path "/tmp". This now returns an error.
CVE-2022-41721	A request smuggling attack is possible when using MaxBytesHandler. When using MaxBytesHandler, the body of an HTTP request is not fully consumed. When the server attempts to read HTTP2 frames from the connection, it will instead be reading the body of the HTTP request, which could be attacker-manipulated to represent arbitrary HTTP2 requests.

CVE ID	Description
CVE-2022-41722	A path traversal vulnerability exists in filepath.Clean on Windows. On Windows, the filepath.Clean function could transform an invalid path such as "a/../c:/b" into the valid path "c:\b". This transformation of a relative (if invalid) path into an absolute path could enable a directory traversal attack. After fix, the filepath.Clean function transforms this path into the relative (but still invalid) path ".\c\b".
CVE-2022-41723	A maliciously crafted HTTP/2 stream could cause excessive CPU consumption in the HPACK decoder, sufficient to cause a denial of service from a small number of small requests.
CVE-2022-41724	Large handshake records may cause panics in crypto/tls. Both clients and servers may send large TLS handshake records which cause servers and clients, respectively, to panic when attempting to construct responses. This affects all TLS 1.3 clients, TLS 1.2 clients which explicitly enable session resumption (by setting Config.ClientSessionCache to a non-nil value), and TLS 1.3 servers which request client certificates (by setting Config.ClientAuth >= RequestClientCert).
CVE-2022-41725	A denial of service is possible from excessive resource consumption in net/http and mime/multipart. Multipart form parsing with mime/multipart.Reader.ReadForm can consume largely unlimited amounts of memory and disk files. This also affects form parsing in the net/http package with the Request methods FormFile, FormValue, ParseMultipartForm, and PostFormValue. ReadForm takes a maxMemory parameter, and is documented as storing "up to maxMemory bytes +10MB (reserved for non-file parts) in memory". File parts which cannot be stored in memory are stored on disk in temporary files. The unconfigurable 10MB reserved for non-file parts is excessively large and can potentially open a denial of service vector on its own. However, ReadForm did not properly account for all memory consumed by a parsed form, such as map entry overhead, part names, and MIME headers, permitting a maliciously crafted form to consume well over 10MB. In addition, ReadForm contained no limit on the number of disk files created, permitting a relatively small request body to create a large number of disk temporary files. With fix, ReadForm now properly accounts for various forms of memory overhead, and should now stay within its documented limit of 10MB + maxMemory bytes of memory consumption. Users should still be aware that this limit is high and may still be hazardous. In addition, ReadForm now creates at most one on-disk temporary file, combining multiple form parts into a single temporary file. The mime/multipart.File interface type's documentation states, "If stored on disk, the File's underlying concrete type will be an *os.File.". This is no longer the case when a form contains more than one file part, due to this coalescing of parts into a single file. The previous behavior of using distinct files for each form part may be reenabled with the environment variable GODEBUG=multipartfiles=distinct. Users should be aware that multipart.ReadForm and the http.Request methods that call it do not limit the amount of disk consumed by temporary files. Callers can limit the size of form data with http.MaxBytesReader.
CVE-2022-41848	drivers/char/pcmcia/synclink_cs.c in the Linux kernel through 5.19.12 has a race condition and resultant use-after-free if a physically proximate attacker removes a PCMCIA device while calling ioctl, aka a race condition between mgs_lpc_ioctl and mgs_lpc_detach.
CVE-2022-41850	roccat_report_event in drivers/hid/hid-roccat.c in the Linux kernel through 5.19.12 has a race condition and resultant use-after-free in certain situations where a report is received while copying a report->value is in progress.
CVE-2022-41862	In PostgreSQL, a modified, unauthenticated server can send an unterminated string during the establishment of Kerberos transport encryption. In certain conditions a server can cause a libpq client to over-read and report an error message containing uninitialized bytes.
CVE-2022-42003	In FasterXML jackson-databind before versions 2.13.4.1 and 2.12.17.1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled.
CVE-2022-42004	In FasterXML jackson-databind before 2.13.4, resource exhaustion can occur because of a lack of a check in BeanDeserializer._deserializeFromArray to prevent use of deeply nested arrays. An application is vulnerable only with certain customized choices for deserialization.
CVE-2022-4292	Use After Free in GitHub repository vim/vim prior to 9.0.0882.
CVE-2022-4293	Floating Point Comparison with Incorrect Operator in GitHub repository vim/vim prior to 9.0.0804.
CVE-2022-44034	An issue was discovered in the Linux kernel through 6.0.6. drivers/char/pcmcia/scr24x_cs.c has a race condition and resultant use-after-free if a physically proximate attacker removes a PCMCIA device while calling open(), aka a race condition between scr24x_open() and scr24x_remove().
CVE-2022-44840	Heap buffer overflow vulnerability in binutils readelf before 2.40 via function find_section_in_set in file readelf.c.
CVE-2022-4543	A flaw named "EntryBleed" was found in the Linux Kernel Page Table Isolation (KPTI). This issue could allow a local attacker to leak KASLR base via prefetch side-channels based on TLB timing for Intel systems.

CVE ID	Description
CVE-2022-45885	An issue was discovered in the Linux kernel through 6.0.9. drivers/media/dvb-core/dvb_frontend.c has a race condition that can cause a use-after-free when a device is disconnected.
CVE-2022-46337	A cleverly devised username might bypass LDAP authentication checks. In LDAP-authenticated Derby installations, this could let an attacker fill up the disk by creating junk Derby databases. In LDAP-authenticated Derby installations, this could also allow the attacker to execute malware which was visible to and executable by the account which booted the Derby server. In LDAP-protected databases which weren't also protected by SQL GRANT/REVOKE authorization, this vulnerability could also let an attacker view and corrupt sensitive data and run sensitive database functions and procedures. Mitigation: Users should upgrade to Java 21 and Derby 10.17.1.0. Alternatively, users who wish to remain on older Java versions should build their own Derby distribution from one of the release families to which the fix was backported: 10.16, 10.15, and 10.14. Those are the releases which correspond, respectively, with Java LTS versions 17, 11, and 8.
CVE-2022-47007	An issue was discovered function stab_demangle_v3_arg in stabs.c in Binutils 2.34 thru 2.38, allows attackers to cause a denial of service due to memory leaks.
CVE-2022-47008	An issue was discovered function make_tmpdir, and make_tmpname in bucomm.c in Binutils 2.34 thru 2.38, allows attackers to cause a denial of service due to memory leaks.
CVE-2022-47010	An issue was discovered function pr_function_type in prdbg.c in Binutils 2.34 thru 2.38, allows attackers to cause a denial of service due to memory leaks.
CVE-2022-47011	An issue was discovered function parse_stab_struct_fields in stabs.c in Binutils 2.34 thru 2.38, allows attackers to cause a denial of service due to memory leaks.
CVE-2022-48303	GNU Tar through 1.34 has a one-byte out-of-bounds read that results in use of uninitialized memory for a conditional jump. Exploitation to change the flow of control has not been demonstrated. The issue occurs in from_header in list.c via a V7 archive in which mtime has approximately 11 whitespace characters.
CVE-2022-48338	An issue was discovered in GNU Emacs through 28.2. In ruby-mode.el, the ruby-find-library-file function has a local command injection vulnerability. The ruby-find-library-file function is an interactive function, and bound to C-c C-f. Inside the function, the external command gem is called through shell-command-to-string, but the feature-name parameters are not escaped. Thus, malicious Ruby source files may cause commands to be executed.
CVE-2022-48628	In the Linux kernel, the following vulnerability has been resolved: ceph: drop messages from MDS when unmounting When unmounting all the dirty buffers will be flushed and after the last osd request is finished the last reference of the i_count will be released. Then it will flush the dirty cap/snap to MDSs, and the unmounting won't wait the possible acks, which will ihold the inodes when updating the metadata locally but makes no sense any more, of this. This will make the evict_inodes() to skip these inodes. If encrypt is enabled the kernel generate a warning when removing the encrypt keys when the skipped inodes still hold the keyring: WARNING: CPU: 4 PID: 168846 at fs/crypto/keyring.c:242 fscrypt_destroy_keyring+0x7e/0xd0 CPU: 4 PID: 168846 Comm: umount Tainted: G S 6.1.0-rc5-ceph-g72ead199864c #1 Hardware name: Supermicro SYS-5018R-WR/X10SRW-F, BIOS 2.0 12/17/2015 RIP: 0010:fscrypt_destroy_keyring+0x7e/0xd0 RSP: 0018:ffffc9000b277e28 EFLAGS: 00010202 RAX: 0000000000000002 RBX: ffff88810d52ac00 RCX: ffff88810b56aa00 RDX: 0000000080000000 RSI: ffffffff822f3a09 RDI: ffff888108f59000 RBP: ffff8881d394fb88 R08: 0000000000000028 R09: 0000000000000000 R10: 0000000000000001 R11: 11ff4fe6834fcd91 R12: ffff8881d394fc40 R13: ffff888108f59000 R14: ffff8881d394f800 R15: 0000000000000000 FS: 00007fd83f6f1080(0000) GS:ffff88885fd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f918d417000 CR3: 000000017f89a005 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: <TASK> generic_shutdown_super+0x47/0x120 kill_anon_super+0x14/0x30 ceph_kill_sb+0x36/0x90 [ceph] deactivate_locked_super+0x29/0x60 cleanup_mnt+0xb8/0x140 task_work_run+0x67/0xb0 exit_to_user_mode_prepare+0x23d/0x240 syscall_exit_to_user_mode+0x25/0x60 do_syscall_64+0x40/0x80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7fd83dc39e9b Later the kernel will crash when iput() the inodes and dereferencing the "sb->s_master_keys", which has been released by the generic_shutdown_super().

CVE ID	Description
CVE-2022-48631	In the Linux kernel, the following vulnerability has been resolved: ext4: fix bug in extents parsing when eh_entries == 0 and eh_depth > 0 When walking through an inode extents, the ext4_ext_binsearch_idx() function assumes that the extent header has been previously validated. However, there are no checks that verify that the number of entries (eh->eh_entries) is non-zero when depth is > 0. And this will lead to problems because the EXT_FIRST_INDEX() and EXT_LAST_INDEX() will return garbage and result in this: [135.245946] -----[cut here]----- [135.247579] kernel BUG at fs/ext4/extents.c:2258! [135.249045] invalid opcode: 0000 [#1] PREEMPT SMP [135.250320] CPU: 2 PID: 238 Comm: tmp118 Not tainted 5.19.0-rc8+ #4 [135.252067] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.15.0-0-g2dd4b9b-rebuilt.opensuse.org 04/01/2014 [135.255065] RIP: 0010:ext4_ext_map_blocks+0xc20/0xcb0 [135.256475] Code: [135.261433] RSP: 0018:ffff900005939f8 EFLAGS: 00010246 [135.262847] RAX: 0000000000000024 RBX: ffff90000593b70 RCX: 0000000000000023 [135.264765] RDX: ffff8880038e5f10 RSI: 0000000000000003 RDI: ffff8880046e922c [135.266670] RBP: ffff8880046e9348 R08: 0000000000000001 R09: ffff888002ca580c [135.268576] R10: 0000000000002602 R11: 0000000000000000 R12: 0000000000000024 [135.270477] R13: 0000000000000000 R14: 0000000000000024 R15: 0000000000000000 [135.272394] FS: 00007fdabdc56740(0000) GS:ffff88807d000000(0000) knlGS:0000000000000000 [135.274510] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [135.276075] CR2: 00007ffc26bd4f00 CR3: 0000000006261004 CR4: 0000000000170ea0 [135.277952] Call Trace: [135.278635] <TASK> [135.279247] ? preempt_count_add+0x6d/0xa0 [135.280358] ? percpu_counter_add_batch+0x55/0xb0 [135.281612] ? _raw_read_unlock+0x18/0x30 [135.282704] ext4_map_blocks+0x294/0x5a0 [135.283745] ? xa_load+0x6f/0xa0 [135.284562] ext4_mpage_readpages+0x3d6/0x770 [135.285646] read_pages+0x67/0x1d0 [135.286492] ? folio_add_lru+0x51/0x80 [135.287441] page_cache_ra_unbounded+0x124/0x170 [135.288510] filemap_get_pages+0x23d/0x5a0 [135.289457] ? path_openat+0xa72/0xdd0 [135.290332] filemap_read+0xbf/0x300 [135.291158] ? _raw_spin_lock_irqsave+0x17/0x40 [135.292192] new_sync_read+0x103/0x170 [135.293014] vfs_read+0x15d/0x180 [135.293745] ksys_read+0xa1/0xe0 [135.294461] do_syscall_64+0x3c/0x80 [135.295284] entry_SYSCALL_64_after_hwframe+0x46/0xb0 This patch simply adds an extra check in __ext4_ext_check(), verifying that eh_entries is not 0 when eh_depth is > 0.

CVE ID	Description
CVE-2022-48633	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: Fix WARN_ON(lock->magic != lock) error psb_gem_unpin() calls dma_resv_lock() but the underlying ww_mutex gets destroyed by drm_gem_object_release() move the drm_gem_object_release() call in psb_gem_free_object() to after the unpin to fix the below warning: [79.693962] -----[cut here]----- [79.693992] DEBUG_LOCKS_WARN_ON(lock->magic != lock) [79.694015] WARNING: CPU: 0 PID: 240 at kernel/locking/mutex.c:582 __ww_mutex_lock.constprop.0+0x569/0xfb0 [79.694052] Modules linked in: rfcomm snd_seq_dummy snd_hrtimer qrtr bnep ath9k ath9k_common ath9k_hw snd_hda_codec_realtek snd_hda_codec_generic ledtrig_audio snd_hda_codec_hdmi snd_hda_intel ath3k snd_intel_dspcfg mac80211 snd_intel_sdw_acpi btusb snd_hda_codec_btrtl btbcm btintel btmtk bluetooth at24 snd_hda_core snd_hwdep uvcvideo snd_seq libarc4 videobuf2_vmalloc ath videobuf2_memops videobuf2_v4l2 videobuf2_common snd_seq_device videodev acer_wmi intel_powerclamp coretemp mc snd_pcm joydev sparse_keymap ecdh_generic pcspkr wmi_bmof cfg80211 i2c_i801 i2c_smbus snd_timer snd_r8169 rfkill lpc_ich soundcore acpi_cpufreq zram rtc_xpci_sdmcc mmc_core serio_raw rtc_xpci gma500_gfx(E) video wmi ip6_tables ip_tables i2c_dev fuse [79.694436] CPU: 0 PID: 240 Comm: plymouthd Tainted: G W E 6.0.0-rc3+ #490 [79.694457] Hardware name: Packard Bell dot s/SJE01_CT, BIOS V1.10 07/23/2013 [79.694469] RIP: 0010: __ww_mutex_lock.constprop.0+0x569/0xfb0 [79.694496] Code: ff 85 c0 0f 84 15 fb ff ff 8b 05 ca 3c 11 01 85 c0 0f 85 07 fb ff ff 48 c7 c6 30 cb 84 aa 48 c7 c7 a3 e1 82 aa e8 ac 29 f8 ff <0f> 0b e9 ed fa ff ff e8 5b 83 8a ff 85 c0 74 10 44 8b 0d 98 3c 11 [79.694513] RSP: 0018:ffffad1dc048bbe0 EFLAGS: 00010282 [79.694623] RAX: 0000000000000028 RBX: 0000000000000000 RCX: 0000000000000000 [79.694636] RDX: 0000000000000001 RSI: ffffffff8b0ffc RDI: 00000000ffffffff [79.694650] RBP: fffffad1dc048bc8 R08: 0000000000000000 R09: fffffad1dc048ba90 [79.694662] R10: 0000000000000003 R11: ffffffffad62fe8 R12: ffff9ff302103138 [79.694675] R13: ffff9ff306ec8000 R14: ffff9ff307779078 R15: ffff9ff3014c0270 [79.694690] FS: 00007ff1cccf1740(0000) GS:ffff9ff3bc200000(0000) knlGS:0000000000000000 [79.694705] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [79.694719] CR2: 0000559ecbcb4420 CR3: 0000000013210000 CR4: 000000000000006f [79.694734] Call Trace: [79.694749] <TASK> [79.694761] ? __schedule+0x47f/0x1670 [79.694796] ? psb_gem_unpin+0x27/0x1a0 [gma500_gfx] [79.694830] ? lock_is_held_type+0xe3/0x140 [79.694864] ? ww_mutex_lock+0x38/0xa0 [79.694885] ? __cond_resched+0x1c/0x30 [79.694902] ww_mutex_lock+0x38/0xa0 [79.694925] psb_gem_unpin+0x27/0x1a0 [gma500_gfx] [79.694964] psb_gem_unpin+0x199/0x1a0 [gma500_gfx] [79.694996] drm_gem_object_release_handle+0x50/0x60 [79.695020] ? drm_gem_object_handle_put_unlocked+0xf0/0xf0 [79.695042] idr_for_each+0x4b/0xb0 [79.695066] ? _raw_spin_unlock_irqrestore+0x30/0x60 [79.695095] drm_gem_release+0x1c/0x30 [79.695118] drm_file_free.part.0+0x1ea/0x260 [79.695150] drm_release+0x6a/0x120 [79.695175] __fput+0x9f/0x260 [79.695203] task_work_run+0x59/0xa0 [79.695227] do_exit+0x387/0xbe0 [79.695250] ? seqcount_lockdep_reader_access.constprop.0+0x82/0x90 [79.695275] ? lockdep_hardirqs_on+0x7d/0x100 [79.695304] do_group_exit+0x33/0xb0 [79.695331] __x64_sys_exit_group+0x14/0x20 [79.695353] do_syscall_64+0x58/0x80 [79.695376] ? up_read+0x17/0x20 [79.695401] ? lock_is_held_type+0xe3/0x140 [79.695429] ? asm_exc_page_fault+0x22/0x30 [79.695450] ? lockdep_hardirqs_on+0x7d/0x100 [79.695473] entry_SYSCALL_64_after_hwframe+0x63/0xcd [79.695493] RIP: 0033:0x7ff1ccfe3c7 [79.695516] Code: Unable to access opcode bytes at RIP 0x7ff1ccfe3c7. [79.695607] RSP: 002b:00007ffed4413378 EFLAGS: ---truncated---

CVE ID	Description
CVE-2022-48634	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: Fix BUG: sleeping function called from invalid context errors gma_crtc_page_flip() was holding the event_lock spinlock while calling crtc_funcs->mode_set_base() which takes ww_mutex. The only reason to hold event_lock is to clear gma_crtc->page_flip_event on mode_set_base() errors. Instead unlock it after setting gma_crtc->page_flip_event and on errors re-take the lock and clear gma_crtc->page_flip_event it is still set. This fixes the following WARN/stacktrace: [512.122953] BUG: sleeping function called from invalid context at kernel/locking/mutex.c:870 [512.123004] in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 1253, name: gnome-shell [512.123031] preempt_count: 1, expected: 0 [512.123048] RCU nest depth: 0, expected: 0 [512.123066] INFO: lockdep is turned off. [512.123080] irq event stamp: 0 [512.123094] hardirqs last enabled at (0): [<ffffffff8d0ec28c>] copy_process+0x9fc/0x1de0 [512.123176] softirqs last enabled at (0): [<ffffffff8d0ec28c>] copy_process+0x9fc/0x1de0 [512.123207] softirqs last disabled at (0): [<0000000000000000>] 0x0 [512.123233] Preemption disabled at: [512.123241] [<0000000000000000>] 0x0 [512.123275] CPU: 3 PID: 1253 Comm: gnome-shell Tainted: G W 5.19.0+ #1 [512.123304] Hardware name: Packard Bell dot s/SJE01_CT, BIOS V1.10 07/23/2013 [512.123323] Call Trace: [512.123346] <TASK> [512.123370] dump_stack_lvl+0x5b/0x77 [512.123412] __might_resched.cold+0xff/0x13a [512.123458] ww_mutex_lock+0x1e/0xa0 [512.123495] psb_gem_pin+0x2c/0x150 [gma500_gfx] [512.123601] gma_pipe_set_base+0x76/0x240 [gma500_gfx] [512.123708] gma_crtc_page_flip+0x95/0x130 [gma500_gfx] [512.123808] drm_mode_page_flip_ioctl+0x57d/0x5d0 [512.123897] ? drm_mode_cursor2_ioctl+0x10/0x10 [512.123936] drm_ioctl_kernel+0xa1/0x150 [512.123984] drm_ioctl+0x21f/0x420 [512.124025] ? drm_mode_cursor2_ioctl+0x10/0x10 [512.124070] ? rcu_read_lock_bh_held+0xb/0x60 [512.124104] ? lock_release+0x1ef/0x2d0 [512.124161] __x64_sys_ioctl+0x8d/0xd0 [512.124203] do_syscall_64+0x58/0x80 [512.124239] ? do_syscall_64+0x67/0x80 [512.124267] ? trace_hardirqs_on_prepare+0x55/0xe0 [512.124300] ? do_syscall_64+0x67/0x80 [512.124340] ? rcu_read_lock_sched_held+0x10/0x80 [512.124377] entry_SYSCALL_64_after_hwframe+0x63/0xcd [512.124411] RIP: 0033:0x7fcc4a70740f [512.124442] Code: 00 48 89 44 24 18 31 c0 48 8d 44 24 60 c7 04 24 10 00 00 00 48 89 44 24 08 48 8d 44 24 20 48 89 44 24 10 b8 10 00 00 00 0f 05 <89> c2 3d 00 f0 ff 77 18 48 8b 44 24 18 64 48 2b 04 25 28 00 00 [512.124470] RSP: 002b:00007ffda73f5390 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 [512.124503] RAX: ffffffff8ffffda RBX: 000055cc9e474500 RCX: 00007fcc4a70740f [512.124524] RDX: 00007ffda73f5420 RSI: 00000000c01864b0 RDI: 0000000000000009 [512.124544] RBP: 00007ffda73f5420 R08: 000055cc9c0b0cb0 R09: 0000000000000034 [512.124564] R10: 0000000000000000 R11: 0000000000000246 R12: 00000000c01864b0 [512.124584] R13: 0000000000000009 R14: 000055cc9df484d0 R15: 000055cc9af5d0c0 [512.124647] </TASK>
CVE-2022-48635	In the Linux kernel, the following vulnerability has been resolved: fsdax: Fix infinite loop in dax_iomap_rw() I got an infinite loop and a WARNING report when executing a tail command in virtiofs. WARNING: CPU: 10 PID: 964 at fs/iomap/iter.c:34 iomap_iter+0x3a2/0x3d0 Modules linked in: CPU: 10 PID: 964 Comm: tail Not tainted 5.19.0-rc7 Call Trace: <TASK> dax_iomap_rw+0xea/0x620 ? __this_cpu_preempt_check+0x13/0x20 fuse_dax_read_iter+0x47/0x80 fuse_file_read_iter+0xae/0xd0 new_sync_read+0xfe/0x180 ? 0xffffffff81000000 vfs_read+0x14d/0x1a0 ksys_read+0x6d/0xf0 __x64_sys_read+0x1a/0x20 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd The tail command will call read() with a count of 0. In this case, iomap_iter() will report this WARNING, and always return 1 which causing the infinite loop in dax_iomap_rw(). Fixing by checking count whether is 0 in dax_iomap_rw().
CVE-2022-48637	In the Linux kernel, the following vulnerability has been resolved: bnxt: prevent skb UAF after handing over to PTP worker When reading the timestamp is required bnxt_tx_int() hands over the ownership of the completed skb to the PTP worker. The skb should not be used afterwards, as the worker may run before the rest of our code and free the skb, leading to a use-after-free. Since dev_kfree_skb_any() accepts NULL make the loss of ownership more obvious and set skb to NULL.
CVE-2022-48638	In the Linux kernel, the following vulnerability has been resolved: cgroup: cgroup_get_from_id() must check the looked-up kn is a directory cgroup has to be one kernfs dir, otherwise kernel panic is caused, especially cgroup id is provide from userspace.
CVE-2022-48639	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix possible refcount leak in tc_new_tfilter() tfilter_put need to be called to put the refcount got by tp->ops->get to avoid possible refcount leak when chain->tmplt_ops != NULL and chain->tmplt_ops != tp->ops.

CVE ID	Description
CVE-2022-48640	In the Linux kernel, the following vulnerability has been resolved: bonding: fix NULL deref in bond_rr_gen_slave_id Fix a NULL dereference of the struct bonding.rr_tx_counter member because if a bond is initially created with an initial mode != zero (Round Robin) the memory required for the counter is never created and when the mode is changed there is never any attempt to verify the memory is allocated upon switching modes. This causes the following Oops on an aarch64 machine: [334.686773] Unable to handle kernel paging request at virtual address ffff2c91ac905000 [334.694703] Mem abort info: [334.697486] ESR = 0x0000000096000004 [334.701234] EC = 0x25: DABT (current EL), IL = 32 bits [334.706536] SET = 0, FnV = 0 [334.709579] EA = 0, S1PTW = 0 [334.712719] FSC = 0x04: level 0 translation fault [334.717586] Data abort info: [334.720454] ISV = 0, ISS = 0x00000004 [334.724288] CM = 0, WnR = 0 [334.727244] swapper pgtable: 4k pages, 48-bit VAs, pgdp=000008044d662000 [334.733944] [ffff2c91ac905000] pgd=0000000000000000, p4d=0000000000000000 [334.740734] Internal error: Oops: 96000004 [#1] SMP [334.745602] Modules linked in: bonding tls veth rkill sunrpc arm_spe_pmu vfat fat acpi_ipmi ipmi_ssif ixgbe igb i40e mdio ipmi_devintf ipmi_msghandler arm_cm_n arm_dsu_pmu cppc_cpufreq acpi_tad fuse zram crct10dif_ce ast ghash_ce sbas_gwtd nvme drm_vram_helper drm_ttm_helper nvme_core ttm xgene_hwmon [334.772217] CPU: 7 PID: 2214 Comm: ping Not tainted 6.0.0-rc4-00133-g64ae13ed4784 #4 [334.779950] Hardware name: GIGABYTE R272-P31-00/MP32-AR1-00, BIOS F18v (SCP: 1.08.20211002) 12/01/2021 [334.789244] pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [334.796196] pc : bond_rr_gen_slave_id +0x40/0x124 [bonding] [334.801691] lr : bond_xmit_roundrobin_slave_get+0x38/0xdc [bonding] [334.807962] sp : ffff8000221733e0 [334.811265] x29: ffff8000221733e0 x28: ffffd9ac8572d198 x27: ffff80002217357c [334.818392] x26: 0000000000000002a x25: ffffd9ac8572d198 x24: ffff07ff980fa000 [334.825519] x23: ffffd9ac8572d198 x22: ffff07ff98102000 x21: ffff07ff981029c0 [334.832646] x20: 0000000000000001 x19: ffff07ff981029c0 x18: 0000000000000014 [334.839773] x17: 0000000000000000 x16: ffffd9ac8572d198 x15: 0000000000000000 [334.846899] x14: ffff07ff981029c0 x13: ffff07ff981029c0 x12: 0000000000000000 [334.854026] x11: ffffd9ac8572d198 x10: 0000000000000001 x9: ffffd9ac8572d198 [334.861153] x8: ffff07ff981029c0 x7: 0000000000000000 x6: 0000000000000000 [334.868279] x5: ffff2c91ac905000 x4: ffff2c91ac905000 x3: ffff07ff981029c0 [334.875406] x2: ffff2c91ac905000 x1: 0000000000000001 x0: ffff07ff981029c0 [334.882532] Call trace: [334.884967] bond_rr_gen_slave_id+0x40/0x124 [bonding] [334.890109] bond_xmit_roundrobin_slave_get+0x38/0xdc [bonding] [334.896033] __bond_start_xmit+0x128/0x3a0 [bonding] [334.901001] bond_start_xmit+0x54/0xb0 [bonding] [334.905622] dev_hard_start_xmit+0xb4/0x220 [334.909798] __dev_queue_xmit+0x1a0/0x720 [334.913799] arp_xmit+0x3c/0x8c [334.916932] arp_send_dst+0x98/0xd0 [334.920410] arp_sollicit +0xe8/0x230 [334.923888] neigh_probe+0x60/0xb0 [334.927279] __neigh_event_send+0x3b0/0x470 [334.931453] neigh_resolve_output+0x70/0x90 [334.935626] ip_finish_output2+0x158/0x514 [334.939714] __ip_finish_output+0xac/0x1a4 [334.943800] ip_finish_output+0x40/0xfc [334.947626] ip_output+0xf8/0x1a4 [334.950931] ip_send_skb+0x5c/0x100 [334.954410] ip_push_pending_frames +0x3c/0x60 [334.958758] raw_sendmsg+0x458/0x6d0 [334.962325] inet_sendmsg+0x50/0x80 [334.965805] sock_sendmsg+0x60/0x6c [334.969286] __sys_sendto+0xc8/0x134 [334.972853] __arm64_sys_sendto+0x34/0x4c ---truncated---
CVE-2022-48641	In the Linux kernel, the following vulnerability has been resolved: netfilter: ebttables: fix memory leak when blob is malformed The bug fix was incomplete, it "replaced" crash with a memory leak. The old code had an assignment to "ret" embedded into the conditional, restore this.
CVE-2022-48643	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fix nft_counters_enabled underflow at nf_tables_addchain() syzbot is reporting underflow of nft_counters_enabled counter at nf_tables_addchain() [1], for commit 43eb8949cfdffa76 ("netfilter: nf_tables: do not leave chain stats enabled on error") missed that nf_tables_chain_destroy() after nft_basechain_init() in the error path of nf_tables_addchain() decrements the counter because nft_basechain_init() makes nft_is_base_chain() return true by setting NFT_CHAIN_BASE flag. Increment the counter immediately after returning from nft_basechain_init().
CVE-2022-48646	In the Linux kernel, the following vulnerability has been resolved: sfc/siena: fix null pointer dereference in efx_hard_start_xmit Like in previous patch for sfc, prevent potential (but unlikely) NULL pointer dereference.
CVE-2022-48647	In the Linux kernel, the following vulnerability has been resolved: sfc: fix TX channel offset when using legacy interrupts In legacy interrupt mode the tx_channel_offset was hardcoded to 1, but that's not correct if efx_separate_tx_channels is false. In that case, the offset is 0 because the tx queues are in the single existing channel at index 0, together with the rx queue. Without this fix, as soon as you try to send any traffic, it tries to get the tx queues from an uninitialized channel getting these errors: WARNING: CPU: 1 PID: 0 at drivers/net/ethernet/sfc/tx.c:540 efx_hard_start_xmit+0x12e/0x170 [sfc] [...] RIP: 0010:efx_hard_start_xmit+0x12e/0x170 [sfc] [...] Call Trace: <IRQ> dev_hard_start_xmit+0xd7/0x230 sch_direct_xmit+0x9f/0x360 __dev_queue_xmit+0x890/0xa40 [...] BUG: unable to handle kernel NULL pointer dereference at 0000000000000020 [...] RIP: 0010:efx_hard_start_xmit+0x153/0x170 [sfc] [...] Call Trace: <IRQ> dev_hard_start_xmit+0xd7/0x230 sch_direct_xmit+0x9f/0x360 __dev_queue_xmit +0x890/0xa40 [...]

CVE ID	Description
CVE-2022-48648	In the Linux kernel, the following vulnerability has been resolved: sfc: fix null pointer dereference in efx_hard_start_xmit Trying to get the channel from the tx_queue variable here is wrong because we can only be here if tx_queue is NULL, so we shouldn't dereference it. As the above comment in the code says, this is very unlikely to happen, but it's wrong anyway so let's fix it. I hit this issue because of a different bug that caused tx_queue to be NULL. If that happens, this is the error message that we get here: BUG: unable to handle kernel NULL pointer dereference at 0000000000000020 [...] RIP: 0010:efx_hard_start_xmit+0x153/0x170 [sfc]
CVE-2022-48651	In the Linux kernel, the following vulnerability has been resolved: ipvlan: Fix out-of-bound bugs caused by unset skb->mac_header If an AF_PACKET socket is used to send packets through ipvlan and the default xmit function of the AF_PACKET socket is changed from dev_queue_xmit() to packet_direct_xmit() via setsockopt() with the option name of PACKET_QDISC_BYPASS, the skb->mac_header may not be reset and remains as the initial value of 65535, this may trigger slab-out-of-bounds bugs as following: ===== UG: KASAN: slab-out-of-bounds in ipvlan_xmit_mode_l2+0xdb/0x330 [ipvlan] PU: 2 PID: 1768 Comm: raw_send Kdump: loaded Not tainted 6.0.0-rc4+ #6 ardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-1.fc33 all Trace: print_address_description.constprop.0+0x1d/0x160 print_report.cold+0x4f/0x112 kasan_report+0xa3/0x130 ipvlan_xmit_mode_l2+0xdb/0x330 [ipvlan] ipvlan_start_xmit+0x29/0xa0 [ipvlan] __dev_direct_xmit+0x2e2/0x380 packet_direct_xmit+0x22/0x60 packet_snd+0x7c9/0xc40 sock_sendmsg+0x9a/0xa0 __sys_sendto+0x18a/0x230 __x64_sys_sendto+0x74/0x90 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd The root cause is: 1. packet_snd() only reset skb->mac_header when sock->type is SOCK_RAW and skb->protocol is not specified as in packet_parse_headers() 2. packet_direct_xmit() doesn't reset skb->mac_header as dev_queue_xmit() In this case, skb->mac_header is 65535 when ipvlan_xmit_mode_l2() is called. So when ipvlan_xmit_mode_l2() gets mac header with eth_hdr() which use "skb->head + skb->mac_header", out-of-bound access occurs. This patch replaces eth_hdr() with skb_eth_hdr() in ipvlan_xmit_mode_l2() and reset mac header in multicast to solve this out-of-bound bug.
CVE-2022-48652	In the Linux kernel, the following vulnerability has been resolved: ice: Fix crash by keep old cfg when update TCs more than queues There are problems if allocated queues less than Traffic Classes. Commit a632b2a4c920 ("ice: ethtool: Prohibit improper channel config for DCB") already disallow setting less queues than TCs. Another case is if we first set less queues, and later update more TCs config due to LLDP, ice_vsi_cfg_tc() will failed but left dirty num_tx/rx and tc_cfg in vsi, that will cause invalid pointer access. [95.968089] ice 0000:3b:00.1: More TCs defined than queues/rings allocated. [95.968092] ice 0000:3b:00.1: Trying to use more Rx queues (8), than were allocated (1)! [95.968093] ice 0000:3b:00.1: Failed to config TC for VSI index: 0 [95.969621] general protection fault: 0000 [#1] SMP NOPTI [95.969705] CPU: 1 PID: 58405 Comm: lldpad Kdump: loaded Tainted: G U W O ----- -t - 4.18.0 #1 [95.969867] Hardware name: O.E.M/BC11SPSCB10, BIOS 8.23 12/30/2021 [95.969992] RIP: 0010:devm_kmalloc+0xa/0x60 [95.970052] Code: 5c ff ff ff 31 c0 5b 5d 41 5c 3c b8 f4 ff ff ff eb f4 0f 1f 40 00 66 2e 0f 1f 84 00 00 00 00 0f 1f 44 00 00 48 89 f8 89 d1 <8b> 97 60 02 00 00 48 8d 7e 18 48 39 f7 72 3f 55 89 ce 53 48 8b 4c [95.970344] RSP: 0018:ffffc9003f553888 EFLAGS: 00010206 [95.970425] RAX: dead000000000200 RBX: fffffea003c425b00 RCX: 00000000006080c0 [95.970536] RDX: 00000000006080c0 RSI: 0000000000000200 RDI: dead000000000200 [95.970648] RBP: dead000000000200 R08: 00000000000463c0 R09: ffff888ffa900000 [95.970760] R10: 0000000000000000 R11: 0000000000000002 R12: ffff888ff6b40100 [95.970870] R13: ffff888ff6a55018 R14: 0000000000000000 R15: ffff888ff6a55460 [95.970981] FS: 00007f51b7d24700(0000) GS:ffff88903ee80000(0000) knlGS:0000000000000000 [95.971108] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [95.971197] CR2: 00007fac5410d710 CR3: 0000000f2c1de002 CR4: 00000000007606e0 [95.971309] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [95.971419] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [95.971530] PKRU: 55555554 [95.971573] Call Trace: [95.971622] ice_setup_rx_ring+0x39/0x110 [ice] [95.971695] ice_vsi_setup_rx_rings+0x54/0x90 [ice] [95.971774] ice_vsi_open+0x25/0x120 [ice] [95.971843] ice_open_internal+0xb8/0x1f0 [ice] [95.971919] ice_ena_vsi+0x4f/0xd0 [ice] [95.971987] ice_dcb_ena_dis_vsi.constprop.5+0x29/0x90 [ice] [95.972082] ice_pf_dcb_cfg+0x29a/0x380 [ice] [95.972154] ice_dcbnl_setets+0x174/0x1b0 [ice] [95.972220] dcbnl_ieee_set+0x89/0x230 [95.972279] ? dcbnl_ieee_del+0x150/0x150 [95.972341] dcb_doit+0x124/0x1b0 [95.972392] rtnetlink_rcv_msg+0x243/0x2f0 [95.972457] ? dcb_doit+0x14d/0x1b0 [95.972510] ? __kmalloc_node_track_caller+0x1d3/0x280 [95.972591] ? rtnl_calcit.isra.31+0x100/0x100 [95.972661] netlink_rcv_skb+0xc0/0xf0 [95.972720] netlink_unicast+0x16d/0x220 [95.972781] netlink_sendmsg+0x2ba/0x3a0 [95.975891] sock_sendmsg+0x4c/0x50 [95.979032] __sys_sendmsg+0x2e4/0x300 [95.982147] ? kmem_cache_alloc+0x13e/0x190 [95.985242] ? __wake_up_common_lock+0x79/0x90 [95.988338] ? __check_object_size+0xac/0x1b0 [95.991440] ? __copy_to_user+0x22/0x30 [95.994539] ? move_addr_to_user+0xb0/0xd0 [95.997619] ? __sys_sendmsg+0x53/0x80 [96.000664] __sys_sendmsg+0x53/0x80 [96.003747] do_syscall_64+0x5b/0x1d0 [96.006862] entry_SYSCALL_64_after_hwframe+0x65/0xca Only update num_tx/rx when passed check, and restore tc_cfg if setup queue map failed.

CVE ID	Description
CVE-2022-48653	In the Linux kernel, the following vulnerability has been resolved: ice: Don't double unplug aux on peer initiated reset In the IDC callback that is accessed when the aux drivers request a reset, the function to unplug the aux devices is called. This function is also called in the ice_prepare_for_reset function. This double call is causing a "scheduling while atomic" BUG. [662.676430] ice 0000:4c:00.0 rocep76s0: cqp opcode = 0x1 maj_err_code = 0xffff min_err_code = 0x8003 [662.676609] ice 0000:4c:00.0 rocep76s0: [Modify QP Cmd Error][op_code=8] status=-29 waiting=1 completion_err=1 maj=0xffff min=0x8003 [662.815006] ice 0000:4c:00.0 rocep76s0: ICE OICR event notification: oicr = 0x10000003 [662.815014] ice 0000:4c:00.0 rocep76s0: critical PE Error, GLPE_CRITERR=0x00011424 [662.815017] ice 0000:4c:00.0 rocep76s0: Requesting a reset [662.815475] BUG: scheduling while atomic: swapper/37/0/0x00010002 [662.815475] BUG: scheduling while atomic: swapper/37/0/0x00010002 [662.815477] Modules linked in: rpcsec_gss_krb5 auth_rpcgss nfsv4 dns_resolver nfs lockd grace fscache netfs rkill 8021q garp mrp stp llc vfat fat rprcdma intel_rapl_msr intel_rapl_common sunrpc i10nm_edac rdma_ucm nfit ib_srpt libnvdimd ib_isert iscsi_target_mod x86_pkg_temp_thermal intel_powerclamp coretemp target_core_mod snd_hda_intel ib_iser snd_intel_dspcfg libiscsi snd_intel_sdw_acpi scsi_transport_iscsi kvm_intel iTCO_wdt rdma_cm snd_hda_codec kvm iw_cm ipmi_ssif iTCO_vendor_support snd_hda_core irqbypass crct10dif_pclmul crc32_pclmul ghash_clmulni_intel snd_hwdep snd_seq snd_seq_device rapl snd_pcm snd_timer isst_if_mbox_pci pcspkr isst_if_mmio irdma intel_uncore idxd acpi_ipmi joydev isst_if_common snd mei_me idxd_bus ipmi_si soundcore i2c_i801 mei ipmi_devintf i2c_smbus i2c_ismt ipmi_msghandler acpi_power_meter acpi_pad rv(OE) ib_uverbs ib_cm ib_core xfs libcrc32c ast i2c_algo_bit drm_vram_helper drm_kms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops drm_ttm_helper rttm [662.815546] nvme nvme_core ice drm crc32c_intel i40e t10_pi wmi pinctrl_emmitsburg dm_mirror dm_region_hash dm_log dm_mod fuse [662.815557] Preemption disabled at: [662.815558] [<0000000000000000>] 0x0 [662.815563] CPU: 37 PID: 0 Comm: swapper/37 Kdump: loaded Tainted: G S OE 5.17.1 #2 [662.815566] Hardware name: Intel Corporation D50DNP/D50DNP, BIOS SE5C6301.86B.6624.D18.2111021741 11/02/2021 [662.815568] Call Trace: [662.815572] <IRQ> [662.815574] dump_stack_lvl+0x33/0x42 [662.815581] __schedule_bug.cold.147+0x7d/0x8a [662.815588] __schedule+0x798/0x990 [662.815595] schedule+0x44/0xc0 [662.815597] schedule_preempt_disabled+0x14/0x20 [662.815600] __mutex_lock.isra.11+0x46c/0x490 [662.815603] ? __ibdev_printk+0x76/0xc0 [ib_core] [662.815633] device_del+0x37/0x3d0 [662.815639] ice_unplug_aux_dev+0x1a/0x40 [ice] [662.815674] ice_schedule_reset+0x3c/0xd0 [ice] [662.815693] irdma_iidc_event_handler.cold.7+0xb6/0xd3 [irdma] [662.815712] ? bitmap_find_next_zero_area_off+0x45/0xa0 [662.815719] ice_send_event_to_aux+0x54/0x70 [ice] [662.815741] ice_misc_intr+0x21d/0x2d0 [ice] [662.815756] __handle_irq_event_percpu+0x4c/0x180 [662.815762] handle_irq_event_percpu+0xf/0x40 [662.815764] handle_irq_event+0x34/0x60 [662.815766] handle_edge_irq+0x9a/0x1c0 [662.815770] __common_interrupt+0x62/0x100 [662.815774] common_interrupt+0xb4/0xd0 [662.815779] </IRQ> [662.815780] <TASK> [662.815780] asm_common_interrupt+0x1e/0x40 [662.815785] RIP: 0010:cpuidle_enter_state+0xd6/0x380 [662.815789] Code: 49 89 c4 0f 1f 44 00 00 31 ff e8 65 d7 95 ff 45 84 ff 74 12 9c 58 f6 c4 02 0f 85 64 02 00 00 31 ff e8 ae c5 9c ff fb 45 85 f6 <0f> 88 12 01 00 00 49 63 d6 4c 2b 24 24 48 8d 04 52 48 8d 04 82 49 [662.815791] RSP: 0018:ff2c2c4f18edbe80 EFLAGS: 00000202 [662.815793] RAX: ff280805df140000 RBX: 0000000000000002 RCX: 000000000000001f [662.815795] RDX: 0000009a52da2d08 R ---truncated---
CVE-2022-48658	In the Linux kernel, the following vulnerability has been resolved: mm: slub: fix flush_cpu_slab()/__free_slab() invocations in task context. Commit 5a836bf6b09f ("mm: slub: move flush_cpu_slab() invocations __free_slab() invocations out of IRQ context") moved all flush_cpu_slab() invocations to the global workqueue to avoid a problem related with deactivate_slab()/__free_slab() being called from an IRQ context on PREEMPT_RT kernels. When the flush_all_cpu_locked() function is called from a task context it may happen that a workqueue with WQ_MEM_RECLAIM bit set ends up flushing the global workqueue, this will cause a dependency issue. workqueue: WQ_MEM_RECLAIM nvme-delete-wq:nvme_delete_ctrl_work [nvme_core] is flushing !WQ_MEM_RECLAIM events:flush_cpu_slab WARNING: CPU: 37 PID: 410 at kernel/workqueue.c:2637 check_flush_dependency+0x10a/0x120 Workqueue: nvme-delete-wq nvme_delete_ctrl_work [nvme_core] RIP: 0010:check_flush_dependency+0x10a/0x120[453.262125] Call Trace: __flush_work.isra.0+0xbf/0x220 ? __queue_work+0x1dc/0x420 flush_all_cpus_locked+0xfb/0x120 __kmem_cache_shutdown+0x2b/0x320 kmem_cache_destroy+0x49/0x100 bioset_exit+0x143/0x190 blk_release_queue+0xb9/0x100 kobject_cleanup+0x37/0x130 nvme_fc_ctrl_free+0xc6/0x150 [nvme_fc] nvme_free_ctrl+0x1ac/0x2b0 [nvme_core] Fix this bug by creating a workqueue for the flush operation with the WQ_MEM_RECLAIM bit set.

CVE ID	Description
CVE-2022-48659	In the Linux kernel, the following vulnerability has been resolved: mm/slab: fix to return errno if kcalloc() fails In create_unique_id(), kcalloc(, GFP_KERNEL) can fail due to out-of-memory, if it fails, return errno correctly rather than triggering panic via BUG_ON(); kernel BUG at mm/slab.c:5893! Internal error: Oops - BUG: 0 [#1] PREEMPT SMP Call trace: sysfs_slab_add +0x258/0x260 mm/slab.c:5973 __kmem_cache_create+0x60/0x118 mm/slab.c:4899 create_cache mm/slab_common.c:229 [inline] kmem_cache_create_usercopy+0x19c/0x31c mm/slab_common.c:335 kmem_cache_create+0x1c/0x28 mm/slab_common.c:390 f2fs_kmem_cache_create fs/f2fs/f2fs.h:2766 [inline] f2fs_init_xattr_caches+0x78/0xb4 fs/f2fs/xattr.c:808 f2fs_fill_super+0x1050/0x1e0c fs/f2fs/super.c:4149 mount_bdev+0x1b8/0x210 fs/super.c:1400 f2fs_mount+0x44/0x58 fs/f2fs/super.c:4512 legacy_get_tree+0x30/0x74 fs/fs_context.c:610 vfs_get_tree+0x40/0x140 fs/super.c:1530 do_new_mount+0x1dc/0x4e4 fs/namespace.c:3040 path_mount+0x358/0x914 fs/namespace.c:3370 do_mount fs/namespace.c:3383 [inline] __do_sys_mount fs/namespace.c:3591 [inline] __se_sys_mount fs/namespace.c:3568 [inline] __arm64_sys_mount+0x2f8/0x408 fs/namespace.c:3568
CVE-2022-48662	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gem: Really move i915_gem_context.link under ref protection i915_perf assumes that it can use the i915_gem_context reference to protect its i915->gem.contexts.list iteration. However, this requires that we do not remove the context from the list until after we drop the final reference and release the struct. If, as currently, we remove the context from the list during context_close(), the link.next pointer may be poisoned while we are holding the context reference and cause a GPF: [4070.573157] i915 0000:00:02.0: [drm:i915_perf_open_ioctl [i915]] filtering on ctx_id=0x1fffff ctx_id_mask=0x1fffff [4070.574881] general protection fault, probably for non-canonical address 0xdead000000000100: 0000 [#1] PREEMPT SMP [4070.574897] CPU: 1 PID: 284392 Comm: amd_performance Tainted: G E 5.17.9 #180 [4070.574903] Hardware name: Intel Corporation NUC7i5BNK/NUC7i5BNB, BIOS BNKBL357.86A.0052.2017.0918.1346 09/18/2017 [4070.574907] RIP: 0010:oa_configure_all_contexts.isra.0+0x222/0x350 [i915] [4070.574982] Code: 08 e8 32 6e 10 e1 4d 8b 6d 50 b8 ff ff ff ff 49 83 ed 50 f0 41 0f c1 04 24 83 f8 01 0f 84 e3 00 00 00 85 c0 0f 8e fa 00 00 00 <49> 8b 45 50 48 8d 70 b0 49 8d 45 50 48 39 44 24 10 0f 85 34 fe ff [4070.574990] RSP: 0018:ffff90002077b78 EFLAGS: 00010202 [4070.574995] RAX: 0000000000000002 RBX: 0000000000000002 RCX: 0000000000000000 [4070.575000] RDX: 0000000000000001 RSI: fffff90002077b20 RDI: fffff88810ddc7c68 [4070.575004] RBP: 0000000000000001 R08: fffff888103242648 R09: ffffffffefc [4070.575008] R10: ffffffff82c50bc0 R11: 0000000000025c80 R12: fffff888101bf1860 [4070.575012] R13: dead0000000000b0 R14: fffff90002077c04 R15: fffff88810be5cab [4070.575016] FS: 00007f1ed50c0780(0000) GS:ffff88885ec80000(0000) knlGS:0000000000000000 [4070.575021] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [4070.575025] CR2: 00007f1ed5590280 CR3: 000000010ef6f005 CR4: 00000000003706e0 [4070.575029] Call Trace: [4070.575033] <TASK> [4070.575037] lrc_configure_all_contexts+0x13e/0x150 [i915] [4070.575103] gen8_enable_metric_set +0x4d/0x90 [i915] [4070.575164] i915_perf_open_ioctl+0x4bc/0x1500 [i915] [4070.575224] ? asm_common_interrupt+0x1e/0x40 [4070.575232] ? i915_oa_init_reg_state+0x110/0x110 [i915] [4070.575290] drm_ioctl_kernel+0x85/0x110 [4070.575296] ? update_load_avg +0x5f/0x5e0 [4070.575302] drm_ioctl+0x1d3/0x370 [4070.575307] ? i915_oa_init_reg_state +0x110/0x110 [i915] [4070.575382] ? gen8_gt_irq_handler+0x46/0x130 [i915] [4070.575445] __x64_sys_ioctl+0x3c4/0x8d0 [4070.575451] ? __do_softirq+0xaa/0x1d2 [4070.575456] do_syscall_64+0x35/0x80 [4070.575461] entry_SYSCALL_64_after_hwframe+0x44/0xae [4070.575467] RIP: 0033:0x7f1ed5c10397 [4070.575471] Code: 3c 1c e8 1c ff ff 85 c0 79 87 49 c7 c4 ff ff ff ff 5b 5d 4c 89 e0 41 5c c3 66 0f 1f 84 00 00 00 00 00 b8 10 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d a9 da 0d 00 f7 d8 64 89 01 48 [4070.575478] RSP: 002b:00007ffd65c8d7a8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 [4070.575484] RAX: ffffffffefcda RBX: 0000000000000006 RCX: 00007f1ed5c10397 [4070.575488] RDX: 00007ffd65c8d7c0 RSI: 0000000040106476 RDI: 0000000000000006 [4070.575492] RBP: 00005620972f9c60 R08: 000000000000000a R09: 0000000000000005 [4070.575496] R10: 000000000000000d R11: 0000000000000246 R12: 000000000000000a [4070.575500] R13: 000000000000000d R14: 0000000000000000 R15: 00007ffd65c8d7c0 [4070.575505] </TASK> [4070.575507] Modules linked in: nls_asci(E) nls_cp437(E) vfat(E) fat(E) i915(E) x86_pkg_temp_thermal(E) intel_powerclamp(E) crc10dif_pclmul(E) crc32_pclmul(E) crc32c_intel(E) aesni_intel(E) crypto_simd(E) intel_gtt(E) cryptd(E) ttmm(E) rapl(E) intel_cstate(E) drm_kms_helper(E) cfbfillrect(E) syscopyarea(E) cfbimgblt(E) intel_uncore(E) sysfillrect(E) mei_me(E) sysimgblt(E) i2c_i801(E) fb_sys_fops(E) mei(E) intel_pch_thermal(E) i2c_smbus ---truncated---
CVE-2022-48672	In the Linux kernel, the following vulnerability has been resolved: of: fdt: fix off-by-one error in unflatten_dt_nodes() Commit 78c44d910d3e ("drivers/of: Fix depth when unflattening devicetree") forgot to fix up the depth check in the loop body in unflatten_dt_nodes() which makes it possible to overflow the nps[] buffer... Found by Linux Verification Center (linuxtesting.org) with the SVACE static analysis tool.

CVE ID	Description
CVE-2022-48673	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix possible access to freed memory in link clear After modifying the QP to the Error state, all RX WR would be completed with WC in IB_WC_WR_FLUSH_ERR status. Current implementation does not wait for it is done, but destroy the QP and free the link group directly. So there is a risk that accessing the freed memory in tasklet context. Here is a crash example: BUG: unable to handle page fault for address: ffffffff8f220860 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD f7300e067 P4D f7300e067 PUD f7300f063 PMD 8c4e45063 PTE 800ffff08c9df060 Oops: 0002 [#1] SMP PTI CPU: 1 PID: 0 Comm: swapper/1 Kdump: loaded Tainted: G S OE 5.10.0-0607+ #23 Hardware name: Inspur NF5280M4/YZMB-00689-101, BIOS 4.1.20 07/09/2018 RIP: 0010:native_queued_spin_lock_slowpath+0x176/0x1b0 Code: f3 90 48 8b 32 48 85 f6 74 f6 eb d5 c1 ee 12 83 e0 03 83 ee 01 48 c1 e0 05 48 63 f6 48 05 00 c8 02 00 48 03 04 f5 00 09 98 8e <48> 89 10 8b 42 08 85 c0 75 09 f3 90 8b 42 08 85 c0 74 f7 48 8b 32 RSP: 0018:ffffb3b6c001ebd8 EFLAGS: 00010086 RAX: ffffffff8f220860 RBX: 0000000000000246 RCX: 0000000000080000 RDX: ffff91db1f86c800 RSI: 000000000000173c RDI: ffff91db62bace00 RBP: ffff91db62bacc00 R08: 0000000000000000 R09: c0000010000028b R10: 000000000005198 R11: fffffb3b6c001ea58 R12: ffff91db80e05010 R13: 000000000000000a R14: 0000000000000006 R15: 0000000000000040 FS: 0000000000000000(0000) GS:ffff91db1f840000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffff8f220860 CR3: 00000001f9580004 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <IRQ> _raw_spin_lock_irqsave+0x30/0x40 mlx5_ib_poll_cq+0x4c/0xc50 [mlx5_ib] smc_wr_rx_tasklet_fn +0x56/0xa0 [smc] tasklet_action_common.isra.21+0x66/0x100 __do_softirq+0xd5/0x29c asm_call_irq_on_stack+0x12/0x20 </IRQ> do_softirq_own_stack+0x37/0x40 irq_exit_rcu+0x9d/0xa0 sysvec_call_function_single+0x34/0x80 asm_sysvec_call_function_single+0x12/0x20
CVE-2022-48686	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix UAF when detecting digest errors We should also bail from the io_work loop when we set rd_enabled to true, so we don't attempt to read data from the socket when the TCP stream is already out-of-sync or corrupted.
CVE-2022-48687	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: fix out-of-bounds read when setting HMAC data. The SRv6 layer allows defining HMAC data that can later be used to sign IPv6 Segment Routing Headers. This configuration is realised via netlink through four attributes: SEG6_ATTR_HMACKEYID, SEG6_ATTR_SECRET, SEG6_ATTR_SECRETLEN and SEG6_ATTR_ALGID. Because the SECRETLEN attribute is decoupled from the actual length of the SECRET attribute, it is possible to provide invalid combinations (e.g., secret = "", secretlen = 64). This case is not checked in the code and with an appropriately crafted netlink message, an out-of-bounds read of up to 64 bytes (max secret length) can occur past the skb end pointer and into skb_shared_info: Breakpoint 1, seg6_genl_sethmac (skb=<optimized out>, info=<optimized out>) at net/ipv6/seg6.c:208 208 memcpy(hinfo->secret, secret, slen); (gdb) bt #0 seg6_genl_sethmac (skb=<optimized out>, info=<optimized out>) at net/ipv6/seg6.c:208 #1 0xffffffff81e012e9 in genl_family_rcv_msg_doit (skb=skb@entry=0xfffff88800b1f9f00, nlh=nlh@entry=0xfffff88800b1b7600, extack=extack@entry=0xfffff90000ba7af0, ops=ops@entry=0xfffff90000ba7a80, hdrlen=4, net=0xffffffff84237580 <init_net>, family=<optimized out>, family=<optimized out>) at net/netlink/genetlink.c:731 #2 0xffffffff81e01435 in genl_family_rcv_msg (extack=0xfffff90000ba7af0, nlh=0xfffff88800b1b7600, skb=0xfffff88800b1f9f00, family=0xffffffff82fef6c0 <seg6_genl_family>) at net/netlink/genetlink.c:775 #3 genl_rcv_msg (skb=0xfffff88800b1f9f00, nlh=0xfffff88800b1b7600, extack=0xfffff90000ba7af0) at net/netlink/genetlink.c:792 #4 0xffffffff81dffc3 in netlink_rcv_skb (skb=skb@entry=0xfffff88800b1f9f00, cb=cb@entry=0xffffffff81e01350 <genl_rcv_msg>) at net/netlink/af_netlink.c:2501 #5 0xffffffff81e00919 in genl_rcv (skb=0xfffff88800b1f9f00) at net/netlink/genetlink.c:803 #6 0xffffffff81dff6ae in netlink_unicast_kernel (ssk=0xfffff888010eec800, skb=0xfffff88800b1f9f00, sk=0xfffff888004aed000) at net/netlink/af_netlink.c:1319 #7 netlink_unicast (ssk=ssk@entry=0xfffff888010eec800, skb=skb@entry=0xfffff88800b1f9f00, portid=portid@entry=0, nonblock=<optimized out>) at net/netlink/af_netlink.c:1345 #8 0xffffffff81dff9a4 in netlink_sendmsg (sock=<optimized out>, msg=0xfffff90000ba7e48, len=<optimized out>) at net/netlink/af_netlink.c:1921 ... (gdb) p/x ((struct sk_buff *)0xfffff88800b1f9f00)->head + ((struct sk_buff *)0xfffff88800b1f9f00)->end \$1 = 0xfffff88800b1b76c0 (gdb) p/x secret \$2 = 0xfffff88800b1b76c0 (gdb) p slen \$3 = 64 '@' The OOB data can then be read back from userspace by dumping HMAC state. This commit fixes this by ensuring SECRETLEN cannot exceed the actual length of SECRET.
CVE-2022-48690	In the Linux kernel, the following vulnerability has been resolved: ice: Fix DMA mappings leak Fix leak, when user changes ring parameters. During reallocation of RX buffers, new DMA mappings are created for those buffers. New buffers with different RX ring count should substitute older ones, but those buffers were freed in ice_vsi_cfg_rxq and reallocated again with ice_alloc_rx_buf. kfree on rx_buf caused leak of already mapped DMA. Reallocate ZC with xdp_buf struct, when BPF program loads. Reallocate back to rx_buf, when BPF program unloads. If BPF program is loaded/unloaded and XSK pools are created, reallocate RX queues accordingly in XDP_SETUP_XSK_POOL handler. Steps for reproduction: while : do for ((i=0; i<=8160; i=i+32)) do ethtool -G enp130s0f0 rx \$i tx \$i sleep 0.5 ethtool -g enp130s0f0 done done

CVE ID	Description
CVE-2022-48691	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: clean up hook list when offload flags check fails splice back the hook list so nft_chain_release_hook() has a chance to release the hooks. BUG: memory leak unreferenced object 0xfffff88810180b100 (size 96): comm "syz-executor133", pid 3619, jiffies 4294945714 (age 12.690s) hex dump (first 32 bytes): 28 64 23 02 81 88 ff ff 28 64 23 02 81 88 ff ff (d#.....(d#..... 90 a8 aa 83 ff ff ff ff 00 00 b5 0f 81 88 ff ff backtrace: [<fffff83a8c59b>] kmalloc include/linux/slab.h:600 [inline] [<fffff83a8c59b>] nft_netdev_hook_alloc+0x3b/0xc0 net/netfilter/nf_tables_api.c:1901 [<fffff83a9239a>] nft_chain_parse_netdev net/netfilter/nf_tables_api.c:1998 [inline] [<fffff83a9239a>] nft_chain_parse_hook+0x33a/0x530 net/netfilter/nf_tables_api.c:2073 [<fffff83a9b14b>] nf_tables_addchain.constprop.0+0x10b/0x950 net/netfilter/nf_tables_api.c:2218 [<fffff83a9c41b>] nf_tables_newchain+0xa8b/0xc60 net/netfilter/nf_tables_api.c:2593 [<fffff83a3d6a6>] nfnetlink_rcv_batch+0xa46/0xd20 net/netfilter/nfnetlink.c:517 [<fffff83a3db79>] nfnetlink_rcv_skb_batch net/netfilter/nfnetlink.c:638 [inline] [<fffff83a3db79>] nfnetlink_rcv+0x1f9/0x220 net/netfilter/nfnetlink.c:656 [<fffff83a13b17>] netlink_unicast_kernel net/netlink/af_netlink.c:1319 [inline] [<fffff83a13b17>] netlink_unicast+0x397/0x4c0 net/netlink/af_netlink.c:1345 [<fffff83a13fd6>] netlink_sendmsg+0x396/0x710 net/netlink/af_netlink.c:1921 [<fffff83865ab6>] sock_sendmsg_nosec net/socket.c:714 [inline] [<fffff83865ab6>] sock_sendmsg+0x56/0x80 net/socket.c:734 [<fffff8386601c>] __sys_sendmsg+0x36c/0x390 net/socket.c:2482 [<fffff8386a918>] __sys_sendmsg+0xa8/0x110 net/socket.c:2536 [<fffff8386aaa8>] __sys_sendmsg+0x88/0x100 net/socket.c:2565 [<fffff845e5955>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<fffff845e5955>] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 [<fffff84800087>] entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2022-48694	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix drain SQ hang with no completion SW generated completions for outstanding WRs posted on SQ after QP is in error target the wrong CQ. This causes the ib_drain_sq to hang with no completion. Fix this to generate completions on the right CQ. [863.969340] INFO: task kworker/u52:2:671 blocked for more than 122 seconds. [863.979224] Not tainted 5.14.0-130.el9.x86_64 #1 [863.986588] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [863.996997] task:kworker/u52:2 state:D stack:0 pid:671 ppid:2 flags:0x00004000 [864.007272] Workqueue: xprtiod xprt_autoclose [sunrpc] [864.014056] Call Trace: [864.017575] __schedule+0x206/0x580 [864.022296] schedule+0x43/0xa0 [864.026736] schedule_timeout+0x115/0x150 [864.032185] __wait_for_common+0x93/0x1d0 [864.037717] ? usleep_range_state+0x90/0x90 [864.043368] __ib_drain_sq+0xf6/0x170 [ib_core] [864.049371] ? __rdma_block_iter_next+0x80/0x80 [ib_core] [864.056240] ib_drain_sq+0x66/0x70 [ib_core] [864.062003] rprdma_xprt_disconnect+0x82/0x3b0 [rprdma] [864.069365] ? xprt_prepare_transmit+0x5d/0xc0 [sunrpc] [864.076386] xprt_rdma_close+0xe/0x30 [rprdma] [864.082593] xprt_autoclose+0x52/0x100 [sunrpc] [864.088718] process_one_work+0x1e8/0x3c0 [864.094170] worker_thread+0x50/0x3b0 [864.099109] ? rescuer_thread+0x370/0x370 [864.104473] kthread+0x149/0x170 [864.109022] ? set_kthread_struct+0x40/0x40 [864.114713] ret_from_fork+0x22/0x30
CVE-2022-48696	In the Linux kernel, the following vulnerability has been resolved: regmap: spi: Reserve space for register address/padding Currently the max_raw_read and max_raw_write limits in regmap_spi struct do not take into account the additional size of the transmitted register address and padding. This may result in exceeding the maximum permitted SPI message size, which could cause undefined behaviour, e.g. data corruption. Fix regmap_get_spi_bus() to properly adjust the above mentioned limits by reserving space for the register address/padding as set in the regmap configuration.
CVE-2022-48697	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix a use-after-free Fix the following use-after-free complaint triggered by blktests nvme/004: BUG: KASAN: user-memory-access in blk_mq_complete_request_remote+0xac/0x350 Read of size 4 at addr 0000607bd1835943 by task kworker/13:1/460 Workqueue: nvmet-wq nvme_loop_execute_work [nvme_loop] Call Trace: show_stack+0x52/0x58 dump_stack_lvl+0x49/0x5e print_report.cold+0x36/0x1e2 kasan_report+0xb9/0xf0 __asan_load4+0x6b/0x80 blk_mq_complete_request_remote+0xac/0x350 nvme_loop_queue_response+0x1df/0x275 [nvme_loop] __nvmet_req_complete+0x132/0x4f0 [nvmet] nvmet_req_complete+0x15/0x40 [nvmet] nvmet_execute_io_connect+0x18a/0x1f0 [nvmet] nvme_loop_execute_work+0x20/0x30 [nvme_loop] process_one_work+0x56e/0xa70 worker_thread+0x2d1/0x640 kthread+0x183/0x1c0 ret_from_fork+0x1f/0x30

CVE ID	Description
CVE-2022-48702	In the Linux kernel, the following vulnerability has been resolved: ALSA: emu10k1: Fix out of bounds access in snd_emu10k1_pcm_channel_alloc() The voice allocator sometimes begins allocating from near the end of the array and then wraps around, however snd_emu10k1_pcm_channel_alloc() accesses the newly allocated voices as if it never wrapped around. This results in out of bounds access if the first voice has a high enough index so that first_voice + requested_voice_count > NUM_G (64). The more voices are requested, the more likely it is for this to occur. This was initially discovered using PipeWire, however it can be reproduced by calling aplay multiple times with 16 channels: aplay -r 48000 -D plughw:CARD=Live,DEV=3 -c 16 /dev/zero UBSAN: array-index-out-of-bounds in sound/pci/emu10k1/emupcm.c:127:40 index 65 is out of range for type 'snd_emu10k1_voice [64]' CPU: 1 PID: 31977 Comm: aplay Tainted: G W IOE 6.0.0-rc2-emu10k1+ #7 Hardware name: ASUSTEK COMPUTER INC P5W DH Deluxe/P5W DH Deluxe, BIOS 3002 07/22/2010 Call Trace: <TASK> dump_stack_lvl+0x49/0x63 dump_stack+0x10/0x16 ubsan_epilogue+0x9/0x3f __ubsan_handle_out_of_bounds.cold+0x44/0x49 snd_emu10k1_playback_hw_params+0x3bc/0x420 [snd_emu10k1] snd_pcm_hw_params+0x29f/0x600 [snd_pcm] snd_pcm_common_ioctl+0x188/0x1410 [snd_pcm] ? exit_to_user_mode_prepare+0x35/0x170 ? do_syscall_64+0x69/0x90 ? syscall_exit_to_user_mode+0x26/0x50 ? do_syscall_64+0x69/0x90 ? exit_to_user_mode_prepare+0x35/0x170 snd_pcm_ioctl+0x27/0x40 [snd_pcm] __x64_sys_ioctl+0x95/0xd0 do_syscall_64+0x5c/0x90 ? do_syscall_64+0x69/0x90 ? do_syscall_64+0x69/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2022-48703	In the Linux kernel, the following vulnerability has been resolved: thermal/int340x_thermal: handle data_vault when the value is ZERO_SIZE_PTR In some case, the GDDV returns a package with a buffer which has zero length. It causes that kmempdup() returns ZERO_SIZE_PTR (0x10). Then the data_vault_read() got NULL point dereference problem when accessing the 0x10 value in data_vault. [71.024560] BUG: kernel NULL pointer dereference, address: 0000000000000010 This patch uses ZERO_OR_NULL_PTR() for checking ZERO_SIZE_PTR or NULL value in data_vault.
CVE-2022-48704	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: add a force flush to delay work when radeon Although radeon card fence and wait for gpu to finish processing current batch rings, there is still a corner case that radeon lockup work queue may not be fully flushed, and meanwhile the radeon_suspend_kms() function has called pci_set_power_state() to put device in D3hot state. Per PCI spec rev 4.0 on 5.3.1.4.1 D3hot State. > Configuration and Message requests are the only TLPs accepted by a Function in > the D3hot state. All other received Requests must be handled as Unsupported Requests, > and all received Completions may optionally be handled as Unexpected Completions. This issue will happen in following logs: Unable to handle kernel paging request at virtual address 00008800e0008010 CPU 0 kworker/0:3(131): Oops 0 pc = [<ffffffff811bea5c>] ra = [<ffffffff81240844>] ps = 0000 Tainted: G W pc is at si_gpu_check_soft_reset+0x3c/0x240 ra is at si_dma_is_lockup+0x34/0xd0 v0 = 0000000000000000 t0 = fff08800e0008010 t1 = 000000000010000 t2 = 0000000000008010 t3 = fff00007e3c00000 t4 = fff00007e3c00258 t5 = 000000000000ffff t6 = 0000000000000001 t7 = fff00007ef078000 s0 = fff00007e3c016e8 s1 = fff00007e3c00000 s2 = fff00007e3c00018 s3 = fff00007e3c00000 s4 = fff00007fff59d80 s5 = 0000000000000000 s6 = fff00007ef07bd98 a0 = fff00007e3c00000 a1 = fff00007e3c016e8 a2 = 0000000000000008 a3 = 0000000000000001 a4 = 8f5c28f5c28f5c29 a5 = ffffffff810f4338 t8 = 0000000000000275 t9 = ffffffff809b66f8 t10 = ff6769c5d964b800 t11= 000000000000b886 pv = ffffffff811bea20 at = 0000000000000000 gp = ffffffff81d89690 sp = 00000000aa814126 Disabling lock debugging due to kernel taint Trace: [<ffffffff81240844>] si_dma_is_lockup+0x34/0xd0 [<ffffffff8119610>] radeon_fence_check_lockup+0xd0/0x290 [<ffffffff80977010>] process_one_work+0x280/0x550 [<ffffffff80977350>] worker_thread+0x70/0x7c0 [<ffffffff80977410>] worker_thread+0x130/0x7c0 [<ffffffff80982040>] kthread+0x200/0x210 [<ffffffff809772e0>] worker_thread+0x0/0x7c0 [<ffffffff80981f8c>] kthread+0x14c/0x210 [<ffffffff80911658>] ret_from_kernel_thread+0x18/0x20 [<ffffffff80981e40>] kthread+0x0/0x210 Code: ad3e0008 43f0074a ad7e0018 ad9e0020 8c3001e8 40230101 <88210000> 4821ed21 So force lockup work queue flush to fix this problem.

CVE ID	Description
CVE-2022-48705	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921e: fix crash in chip reset fail In case of drv own fail in reset, we may need to run mac_reset several times. The sequence would trigger system crash as the log below. Because we do not re-enable/schedule "tx_napi" before disable it again, the process would keep waiting for state change in napi_disable(). To avoid the problem and keep status synchronize for each run, goto final resource handling if drv own failed. [5857.353423] mt7921e 0000:3b:00.0: driver own failed [5858.433427] mt7921e 0000:3b:00.0: Timeout for driver own [5859.633430] mt7921e 0000:3b:00.0: driver own failed [5859.633444] -----[cut here]----- [5859.633446] WARNING: CPU: 6 at kernel/kthread.c:659 kthread_park+0x11d [5859.633717] Workqueue: mt76 mt7921_mac_reset_work [mt7921_common] [5859.633728] RIP: 0010:kthread_park+0x11d/0x150 [5859.633736] RSP: 0018:ffff8881b676fc68 EFLAGS: 00010202 [5859.633766] Call Trace: [5859.633768] <TASK> [5859.633771] mt7921e_mac_reset+0x176/0x6f0 [mt7921e] [5859.633778] mt7921_mac_reset_work+0x184/0x3a0 [mt7921_common] [5859.633785] ? mt7921_mac_set_timing+0x520/0x520 [mt7921_common] [5859.633794] ? __kasan_check_read+0x11/0x20 [5859.633802] process_one_work+0x7ee/0x1320 [5859.633810] worker_thread+0x53c/0x1240 [5859.633818] kthread+0x2b8/0x370 [5859.633824] ? process_one_work+0x1320/0x1320 [5859.633828] ? kthread_complete_and_exit+0x30/0x30 [5859.633834] ret_from_fork+0x1f/0x30 [5859.633842] </TASK>
CVE-2022-48709	In the Linux kernel, the following vulnerability has been resolved: ice: switch: fix potential memleak in ice_add_adv_recipe() When ice_add_special_words() fails, the 'rm' is not released, which will lead to a memory leak. Fix this up by going to 'err_unroll' label. Compile tested only.
CVE-2022-48710	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix a possible null pointer dereference In radeon_fp_native_mode(), the return value of drm_mode_duplicate() is assigned to mode, which will lead to a NULL pointer dereference on failure of drm_mode_duplicate(). Add a check to avoid npd. The failure status of drm_cvt_mode() on the other path is checked too.
CVE-2022-48711	In the Linux kernel, the following vulnerability has been resolved: tipc: improve size validations for received domain records The function tipc_mon_rcv() allows a node to receive and process domain_record structs from peer nodes to track their views of the network topology. This patch verifies that the number of members in a received domain record does not exceed the limit defined by MAX_MON_DOMAIN, something that may otherwise lead to a stack overflow. tipc_mon_rcv() is called from the function tipc_link_proto_rcv(), where we are reading a 32 bit message data length field into a uint16. To avert any risk of bit overflow, we add an extra sanity check for this in that function. We cannot see that happen with the current code, but future designers being unaware of this risk, may introduce it by allowing delivery of very large (> 64k) sk buffers from the bearer layer. This potential problem was identified by Eric Dumazet. This fixes CVE-2022-0435
CVE-2022-48713	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/pt: Fix crash with stop filters in single-range mode Add a check for !buf->single before calling pt_buffer_region_size in a place where a missing check can cause a kernel crash. Fixes a bug introduced by commit 670638477aed ("perf/x86/intel/pt: Opportunistically use single range output mode"), which added a support for PT single-range output mode. Since that commit if a PT stop filter range is hit while tracing, the kernel will crash because of a null pointer dereference in pt_handle_status due to calling pt_buffer_region_size without a ToPA configured. The commit which introduced single-range mode guarded almost all uses of the ToPA buffer variables with checks of the buf->single variable, but missed the case where tracing was stopped by the PT hardware, which happens when execution hits a configured stop filter. Tested that hitting a stop filter while PT recording successfully records a trace with this patch but crashes without this patch.
CVE-2022-48714	In the Linux kernel, the following vulnerability has been resolved: bpf: Use VM_MAP instead of VM_ALLOC for ringbuf After commit 2fd3fb0be1d1 ("kasan, vmalloc: unpoison VM_ALLOC pages after mapping"), non-VM_ALLOC mappings will be marked as accessible in __get_vm_area_node() when KASAN is enabled. But now the flag for ringbuf area is VM_ALLOC, so KASAN will complain out-of-bound access after vmalloc() returns. Because the ringbuf area is created by mapping allocated pages, so use VM_MAP instead. After the change, info in /proc/vmallocinfo also changes from [start]-[end] 24576 ringbuf_map_alloc+0x171/0x290 vmalloc user to [start]-[end] 24576 ringbuf_map_alloc+0x171/0x290 vmalloc user

CVE ID	Description
CVE-2022-48715	In the Linux kernel, the following vulnerability has been resolved: scsi: bnx2fc: Make bnx2fc_recv_frame() mp safe Running tests with a debug kernel shows that bnx2fc_recv_frame() is modifying the per_cpu lport stats counters in a non-mpsafely way. Just boot a debug kernel and run the bnx2fc driver with the hardware enabled. [1391.699147] BUG: using smp_processor_id() in preemptible [00000000] code: bnx2fc_ [1391.699160] caller is bnx2fc_recv_frame+0xb9f9/0x1760 [bnx2fc] [1391.699174] CPU: 2 PID: 4355 Comm: bnx2fc_l2_threa Kdump: loaded Tainted: G B [1391.699180] Hardware name: HP ProLiant DL120 G7, BIOS J01 07/01/2013 [1391.699183] Call Trace: [1391.699188] dump_stack_lvl+0x57/0x7d [1391.699198] check_preemption_disabled+0xc8/0xd0 [1391.699205] bnx2fc_recv_frame+0xb9f9/0x1760 [bnx2fc] [1391.699215] ? do_raw_spin_trylock+0xb5/0x180 [1391.699221] ? bnx2fc_npiv_create_vports.isra.0+0x4e0/0x4e0 [bnx2fc] [1391.699229] ? bnx2fc_l2_rcv_thread+0xb7/0x3a0 [bnx2fc] [1391.699240] bnx2fc_l2_rcv_thread+0x1af/0x3a0 [bnx2fc] [1391.699250] ? bnx2fc_ulp_init+0xc0/0xc0 [bnx2fc] [1391.699258] kthread+0x364/0x420 [1391.699263] ? _raw_spin_unlock_irq+0x24/0x50 [1391.699268] ? set_kthread_struct+0x100/0x100 [1391.699273] ret_from_fork+0x22/0x30 Restore the old get_cpu/put_cpu code with some modifications to reduce the size of the critical section.
CVE-2022-48721	In the Linux kernel, the following vulnerability has been resolved: net/smc: Forward wakeup to smc socket waitqueue after fallback When we replace TCP with SMC and a fallback occurs, there may be some socket waitqueue entries remaining in smc socket->wq, such as epoll_entries inserted by userspace applications. After the fallback, data flows over TCP/IP and only clsock->wq will be woken up. Applications can't be notified by the entries which were inserted in smc socket->wq before fallback. So we need a mechanism to wake up smc socket->wq at the same time if some entries remaining in it. The current workaround is to transfer the entries from smc socket->wq to clsock->wq during the fallback. But this may cause a crash like this: general protection fault, probably for non-canonical address 0xdead000000000100: 0000 [#1] PREEMPT SMP PTI CPU: 3 PID: 0 Comm: swapper/3 Kdump: loaded Tainted: G E 5.16.0+ #107 RIP: 0010: __wake_up_common+0x65/0x170 Call Trace: <IRQ> __wake_up_common_lock+0x7a/0xc0 sock_def_readable+0x3c/0x70 tcp_data_queue+0x4a7/0xc40 tcp_rcv_established+0x32f/0x660 ? sk_filter_trim_cap+0xc9/0x2e0 tcp_v4_do_rcv+0x10b/0x260 tcp_v4_rcv+0xd2a/0xde0 ip_protocol_deliver_rcu+0x3b/0x1d0 ip_local_deliver_finish+0x54/0x60 ip_local_deliver+0x6a/0x110 ? tcp_v4_early_demux+0xa2/0x140 ? tcp_v4_early_demux+0x10d/0x140 ip_sublist_rcv_finish+0x49/0x60 ip_sublist_rcv+0x19d/0x230 ip_list_rcv+0x13e/0x170 __netif_receive_skb_list_core+0x1c2/0x240 netif_receive_skb_list_internal+0x1e6/0x320 napi_complete_done+0x11d/0x190 mlx5e_napi_poll+0x163/0x6b0 [mlx5_core] __napi_poll+0x3c/0x1b0 net_rx_action+0x27c/0x300 __do_softirq+0x114/0x2d2 irq_exit_rcu+0xb4/0xe0 common_interrupt+0xba/0xe0 <IRQ> <TASK> The crash is caused by privately transferring waitqueue entries from smc socket->wq to clsock->wq. The owners of these entries, such as epoll, have no idea that the entries have been transferred to a different socket wait queue and still use original waitqueue spinlock (smc socket->wq.wait.lock) to make the entries operation exclusive, but it doesn't work. The operations to the entries, such as removing from the waitqueue (now is clsock->wq after fallback), may cause a crash when clsock waitqueue is being iterated over at the moment. This patch tries to fix this by no longer transferring wait queue entries privately, but introducing own implementations of clsock's callback functions in fallback situation. The callback functions will forward the wakeup to smc socket->wq if clsock->wq is actually woken up and smc socket->wq has remaining entries.
CVE-2022-48724	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix potential memory leak in intel_setup_irq_remapping() After commit e3beca48a45b ("irqdomain/treewide: Keep firmware node unconditionally allocated"). For tear down scenario, fn is only freed after fail to allocate ir_domain, though it also should be freed in case dmar_enable_qi returns error. Besides free fn, irq_domain and ir_msi_domain need to be removed as well if intel_setup_irq_remapping fails to enable queued invalidation. Improve the rewinding path by add out_free_ir_domain and out_free_fwnode labels per Baolu's suggestion.
CVE-2022-48725	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Fix refcounting leak in siw_create_qp() The atomic_inc() needs to be paired with an atomic_dec() on the error path.

CVE ID	Description
CVE-2022-48728	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix AIP early init panic An early failure in hfi1_ipoib_setup_rm() can lead to the following panic: BUG: unable to handle kernel NULL pointer dereference at 00000000000001b0 PGD 0 P4D 0 Oops: 0002 [#1] SMP NOPTI Workqueue: events work_for_cpu_fn RIP: 0010:try_to_grab_pending+0x2b/0x140 Code: 1f 44 00 00 41 55 41 54 55 48 89 d5 53 48 89 fb 9c 58 0f 1f 44 00 00 48 89 c2 fa 66 0f 1f 44 00 00 48 89 55 00 40 84 f6 75 77 <f0> 48 0f ba 2b 00 72 09 31 c0 5b 5d 41 5c 41 5d c3 48 89 df e8 6c RSP: 0018:ffffb6b3cf7cfa48 EFLAGS: 00010046 RAX: 0000000000000246 RBX: 00000000000001b0 RCX: 0000000000000000 RDX: 0000000000000246 RSI: 0000000000000000 RDI: 00000000000001b0 RBP: ffff6b3cf7cfa70 R08: 0000000000000f09 R09: 0000000000000001 R10: 0000000000000000 R11: 0000000000000001 R12: 0000000000000000 R13: ffff6b3cf7cfa90 R14: ffffffff9b2fbfc0 R15: ffff8a4fdf244690 FS: 0000000000000000(0000) GS:ffff8a527f400000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000000001b0 CR3: 00000017e2410003 CR4: 00000000007706f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: __cancel_work_timer+0x42/0x190 ? dev_printk_emit+0x4e/0x70 iowait_cancel_work+0x15/0x30 [hfi1] hfi1_ipoib_txreq_deinit+0x5a/0x220 [hfi1] ? dev_err+0x6c/0x90 hfi1_ipoib_netdev_dtor+0x15/0x30 [hfi1] hfi1_ipoib_setup_rm+0x10e/0x150 [hfi1] rdma_init_netdev+0x5a/0x80 [ib_core] ? hfi1_ipoib_free_rdma_netdev+0x20/0x20 [hfi1] ipoib_intf_init+0x6c/0x350 [ib_ipoib] ipoib_intf_alloc+0x5c/0xc0 [ib_ipoib] ipoib_add_one+0xbe/0x300 [ib_ipoib] add_client_context+0x12c/0x1a0 [ib_core] enable_device_and_get+0xdc/0x1d0 [ib_core] ib_register_device+0x572/0x6b0 [ib_core] rvt_register_device+0x11b/0x220 [rdmavt] hfi1_register_ib_device+0x6b4/0x770 [hfi1] do_init_one.isra.20+0x3e3/0x680 [hfi1] local_pci_probe+0x41/0x90 work_for_cpu_fn+0x16/0x20 process_one_work+0x1a7/0x360 ? create_worker+0x1a0/0x1a0 worker_thread+0x1cf/0x390 ? create_worker+0x1a0/0x1a0 kthread+0x116/0x130 ? kthread_flush_work_fn+0x10/0x10 ret_from_fork+0x1f/0x40 The panic happens in hfi1_ipoib_txreq_deinit() because there is a NULL deref when hfi1_ipoib_netdev_dtor() is called in this error case. hfi1_ipoib_txreq_init() and hfi1_ipoib_rxq_init() are self unwinding so fix by adjusting the error paths accordingly. Other changes: - hfi1_ipoib_free_rdma_netdev() is deleted including the free_netdev() since the netdev core code deletes calls free_netdev() - The switch to the accelerated entrances is moved to the success path.
CVE-2022-48729	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix panic with larger ipoib send_queue_size When the ipoib send_queue_size is increased from the default the following panic happens: RIP: 0010:hfi1_ipoib_drain_tx_ring+0x45/0xf0 [hfi1] Code: 31 e4 eb 0f 8b 85 c8 02 00 00 41 83 c4 01 44 39 e0 76 60 8b 8d cc 02 00 00 44 89 e3 be 01 00 00 00 d3 e3 48 03 9d c0 02 00 00 <c7> 83 18 01 00 00 00 00 00 48 8b bb 30 01 00 00 e8 25 af a7 e0 RSP: 0018:ffff9000798f4a0 EFLAGS: 00010286 RAX: 0000000000008000 RBX: ffff9000aa0f000 RCX: 000000000000000f RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000000 RBP: ffff88810ff08000 R08: ffff88889476d900 R09: 0000000000000101 R10: 0000000000000000 R11: ffff90006590ff8 R12: 0000000000000200 R13: ffff9000798fba8 R14: 0000000000000000 R15: 0000000000000001 FS: 00007fd0f79cc3c0(0000) GS:ffff88885fb00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffff9000aa0f118 CR3: 0000000889c84001 CR4: 0000000001706e0 Call Trace: <TASK> hfi1_ipoib_napi_tx_disable+0x45/0x60 [hfi1] hfi1_ipoib_dev_stop+0x18/0x80 [hfi1] ipoib_ib_dev_stop+0x1d/0x40 [ib_ipoib] ipoib_stop+0x48/0xc0 [ib_ipoib] __dev_close_many+0x9e/0x110 __dev_change_flags+0xd9/0x210 dev_change_flags+0x21/0x60 do_setlink+0x31c/0x10f0 ? __nla_validate_parse+0x12d/0x1a0 ? __nla_parse+0x21/0x30 ? inet6_validate_link_af+0x5e/0xf0 ? cpumask_next+0x1f/0x20 ? __snmp6_fill_stats64.isra.53+0xbb/0x140 ? __nla_validate_parse+0x47/0x1a0 __rtnl_newlink+0x530/0x910 ? pskb_expand_head+0x73/0x300 ? __kmallocc_node_track_caller+0x109/0x280 ? __nla_put+0xc/0x20 ? cpumask_next_and+0x20/0x30 ? update_sd_lb_stats.constprop.144+0xd3/0x820 ? _raw_spin_unlock_irqrestore+0x25/0x37 ? __wake_up_common_lock+0x87/0xc0 ? kmem_cache_alloc_trace+0x3d/0x3d0 rtnl_newlink+0x43/0x60 The issue happens when the shift that should have been a function of the txq item size mistakenly used the ring size. Fix by using the item size.
CVE-2022-48730	In the Linux kernel, the following vulnerability has been resolved: dma-buf: heaps: Fix potential spectre v1 gadget It appears like nr could be a Spectre v1 gadget as it's supplied by a user and used as an array index. Prevent the contents of kernel memory from being leaked to userspace via speculative execution by using array_index_nospec. [sumits: added fixes and cc: stable tags]

CVE ID	Description
CVE-2022-48731	In the Linux kernel, the following vulnerability has been resolved: mm/kmemleak: avoid scanning potential huge holes When using devm_request_free_mem_region() and devm_memremap_pages() to add ZONE_DEVICE memory, if requested free mem region's end pfn were huge(e.g., 0x400000000), the node_end_pfn() will be also huge (see move_pfn_range_to_zone()). Thus it creates a huge hole between node_start_pfn() and node_end_pfn(). We found on some AMD APUs, amd_kfd requested such a free mem region and created a huge hole. In such a case, following code snippet was just doing busy test_bit() looping on the huge hole. for (pfn = start_pfn; pfn < end_pfn; pfn++) { struct page *page = pfn_to_online_page(pfn); if (!page) continue; ... } So we got a soft lockup: watchdog: BUG: soft lockup - CPU#6 stuck for 26s! [bash:1221] CPU: 6 PID: 1221 Comm: bash Not tainted 5.15.0-custom #1 RIP: 0010:pfn_to_online_page+0x5/0xd0 Call Trace: ? kmemleak_scan+0x16a/0x440 kmemleak_write+0x306/0x3a0 ? common_file_perm+0x72/0x170 full_proxy_write+0x5c/0x90 vfs_write+0xb9/0x260 ksys_write+0x67/0xe0 __x64_sys_write+0x1a/0x20 do_syscall_64+0x3b/0xc0 entry_SYSCALL_64_after_hwframe+0x44/0xae I did some tests with the patch. (1) amdgpu module unloaded before the patch: real 0m0.976s user 0m0.000s sys 0m0.968s after the patch: real 0m0.981s user 0m0.000s sys 0m0.973s (2) amdgpu module loaded before the patch: real 0m35.365s user 0m0.000s sys 0m35.354s after the patch: real 0m1.049s user 0m0.000s sys 0m1.042s
CVE-2022-48732	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: fix off by one in BIOS boundary checking Bounds checking when parsing init scripts embedded in the BIOS reject access to the last byte. This causes driver initialization to fail on Apple eMac's with GeForce 2 MX GPUs, leaving the system with no working console. This is probably only seen on OpenFirmware machines like PowerPC Macs because the BIOS image provided by OF is only the used parts of the ROM, not a power-of-two blocks read from PCI directly so PCs always have empty bytes at the end that are never accessed.
CVE-2022-48744	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Avoid field-overflowing memcpy() In preparation for FORTIFY_SOURCE performing compile-time and run-time field bounds checking for memcpy(), memmove(), and memset(), avoid intentionally writing across neighboring fields. Use flexible arrays instead of zero-element arrays (which look like they are always overflowing) and split the cross-field memcpy() into two halves that can be appropriately bounds-checked by the compiler. We were doing: #define ETH_HLEN 14 #define VLAN_HLEN 4 ... #define MLX5E_XDP_MIN_INLINE (ETH_HLEN + VLAN_HLEN) ... struct mlx5e_tx_wqe *wqe = mlx5_wq_cyc_get_wqe(wq, pi); ... struct mlx5_wqe_eth_seg *eseg = &wqe->eth; struct mlx5_wqe_data_seg *dseg = wqe->data; ... memcpy(eseg->inline_hdr.start, xdptxd->data, MLX5E_XDP_MIN_INLINE); target is wqe->eth.inline_hdr.start (which the compiler sees as being 2 bytes in size), but copying 18, intending to write across start (really vlan_tci, 2 bytes). The remaining 16 bytes get written into wqe->data[0], covering byte_count (4 bytes), lkey (4 bytes), and addr (8 bytes). struct mlx5e_tx_wqe { struct mlx5_wqe_ctrl_seg ctrl; /* 0 16 */ struct mlx5_wqe_eth_seg eth; /* 16 16 */ struct mlx5_wqe_data_seg data[]; /* 32 0 */ /* size: 32, cachelines: 1, members: 3 */ /* last cacheline: 32 bytes */ }; struct mlx5_wqe_eth_seg { u8 swp_outer_l4_offset; /* 0 1 */ u8 swp_outer_l3_offset; /* 1 1 */ u8 swp_inner_l4_offset; /* 2 1 */ u8 swp_inner_l3_offset; /* 3 1 */ u8 cs_flags; /* 4 1 */ u8 swp_flags; /* 5 1 */ __be16 mss; /* 6 2 */ __be32 flow_table_metadata; /* 8 4 */ union { struct { __be16 sz; /* 12 2 */ u8 start[2]; /* 14 2 */ } inline_hdr; /* 12 4 */ struct { __be16 type; /* 12 2 */ __be16 vlan_tci; /* 14 2 */ } insert; /* 12 4 */ __be32 trailer; /* 12 4 */ }; /* 12 4 */ /* size: 16, cachelines: 1, members: 9 */ /* last cacheline: 16 bytes */ }; struct mlx5_wqe_data_seg { __be32 byte_count; /* 0 4 */ __be32 lkey; /* 4 4 */ __be64 addr; /* 8 8 */ /* size: 16, cachelines: 1, members: 3 */ /* last cacheline: 16 bytes */ }; So, split the memcpy() so the compiler can reason about the buffer sizes. "pahole" shows no size nor member offset changes to struct mlx5e_tx_wqe nor struct mlx5e_umr_wqe. "objdump -d" shows no meaningful object code changes (i.e. only source line number induced differences and optimizations).
CVE-2022-48745	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Use del_timer_sync in fw reset flow of halting poll Substitute del_timer() with del_timer_sync() in fw reset polling deactivation flow, in order to prevent a race condition which occurs when del_timer() is called and timer is deactivated while another process is handling the timer interrupt. A situation that led to the following call trace: RIP: 0010:run_timer_softirq+0x137/0x420 <IRQ> recalibrate_cpu_khz+0x10/0x10 ktime_get+0x3e/0xa0 ? sched_clock_cpu+0xb/0xc0 __do_softirq+0xf5/0x2ea irq_exit_rcu+0xc1/0xf0 sysvec_apic_timer_interrupt+0x9e/0xc0 asm_sysvec_apic_timer_interrupt+0x12/0x20 </IRQ>

CVE ID	Description
CVE-2022-48746	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix handling of wrong devices during bond netevent Current implementation of bond netevent handler only check if the handled netdev is VF representor and it missing a check if the VF representor is on the same phys device of the bond handling the netevent. Fix by adding the missing check and optimizing the check if the netdev is VF representor so it will not access uninitialized private data and crashes. BUG: kernel NULL pointer dereference, address: 000000000000036c PGD 0 P4D 0 Oops: 0000 [#1] SMP NOPTI Workqueue: eth3bond0 bond_mii_monitor [bonding] RIP: 0010:mlx5e_is_uplink_rep+0xc/0x50 [mlx5_core] RSP: 0018:ffff88812d69fd60 EFLAGS: 00010282 RAX: 0000000000000000 RBX: ffff8881cf800000 RCX: 0000000000000000 RDX: ffff88812d69fe10 RSI: 000000000000001b RDI: ffff8881cf800880 RBP: ffff8881cf800000 R08: 00000445cabccf2b R09: 0000000000000008 R10: 0000000000000004 R11: 0000000000000008 R12: ffff88812d69fe10 R13: 00000000fffffefe R14: ffff88820cf9000 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff88846fb00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000000036c CR3: 0000000103d80006 CR4: 0000000000370ea0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff00ff DR7: 0000000000000400 Call Trace: mlx5e_eswitch_uplink_rep+0x31/0x40 [mlx5_core] mlx5e_rep_is_lag_netdev+0x94/0xc0 [mlx5_core] mlx5e_rep_esw_bond_netevent+0xeb/0x3d0 [mlx5_core] raw_notifier_call_chain+0x41/0x60 call_netdevice_notifiers_info+0x34/0x80 netdev_lower_state_changed+0x4e/0xa0 bond_mii_monitor+0x56b/0x640 [bonding] process_one_work+0x1b9/0x390 worker_thread+0x4d/0x3d0 ? rescuer_thread+0x350/0x350 kthread+0x124/0x150 ? set_kthread_struct+0x40/0x40 ret_from_fork+0x1f/0x30
CVE-2022-48748	In the Linux kernel, the following vulnerability has been resolved: net: bridge: vlan: fix memory leak in __allowed_ingress When using per-vlan state, if vlan snooping and stats are disabled, untagged or priority-tagged ingress frame will go to check pvid state. If the port state is forwarding and the pvid state is not learning/forwarding, untagged or priority-tagged frame will be dropped but skb memory is not freed. Should free skb when __allowed_ingress returns false.
CVE-2022-48751	In the Linux kernel, the following vulnerability has been resolved: net/smc: Transitional solution for clcsock race issue We encountered a crash in smc_setsockopt() and it is caused by accessing smc->clcsock after clcsock was released. BUG: kernel NULL pointer dereference, address: 0000000000000020 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 1 PID: 50309 Comm: nginx Kdump: loaded Tainted: G E 5.16.0-rc4+ #53 RIP: 0010:smc_setsockopt+0x59/0x280 [smc] Call Trace: <TASK> __sys_setsockopt+0xfc/0x190 __x64_sys_setsockopt+0x20/0x30 do_syscall_64+0x34/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f16ba83918e </TASK> This patch tries to fix it by holding clcsock_release_lock and checking whether clcsock has already been released before access. In case that a crash of the same reason happens in smc_getsockopt() or smc_switch_to_fallback(), this patch also checks smc->clcsock in them too. And the caller of smc_switch_to_fallback() will identify whether fallback succeeds according to the return value.

CVE ID	Description
CVE-2022-48752	In the Linux kernel, the following vulnerability has been resolved: powerpc/perf: Fix power_pmu_disable to call clear_pmi_irq_pending only if PMI is pending Running selftest with CONFIG_PPC_IRQ_SOFT_MASK_DEBUG enabled in kernel triggered below warning: [172.851380] -----[cut here]----- [172.851391] WARNING: CPU: 8 PID: 2901 at arch/powerpc/include/asm/hw_irq.h:246 power_pmu_disable+0x270/0x280 [172.851402] Modules linked in: dm_mod bonding nft_ct nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 ip_set nf_tables rkill nfnetlink sunrpc xfs libcrc32c pseries_rng xts vmx_crypto uio_pdrv_genirq uio sch_fq_codel ip_tables ext4 mbcache jbd2 sd_mod t10_pi sg ibmvscsi ibmveth scsi_transport_srp fuse [172.851442] CPU: 8 PID: 2901 Comm: lost_exception_ Not tainted 5.16.0-rc5-03218-g798527287598 #2 [172.851451] NIP: c00000000013d600 LR: c00000000013d5a4 CTR: c00000000013b180 [172.851458] REGS: c000000017687860 TRAP: 0700 Not tainted (5.16.0-rc5-03218-g798527287598) [172.851465] MSR: 800000000029033 <SF,EE,ME,IR,DR,RI,LE> CR: 48004884 XER: 20040000 [172.851482] CFAR: c00000000013d5b4 IRQMASK: 1 [172.851482] GPR00: c00000000013d5a4 c000000017687b00 c000000002a10600 0000000000000004 [172.851482] GPR04: 0000000082004000 c0000008ba08f0a8 0000000000000000 00000008b7ed0000 [172.851482] GPR08: 00000000446194f6 00000000000008000 c00000000013b118 c000000000d58e68 [172.851482] GPR12: c00000000013d390 c00000001ec54a80 0000000000000000 0000000000000000 [172.851482] GPR16: 0000000000000000 0000000000000000 c000000015d5c708 c0000000025396d0 [172.851482] GPR20: 0000000000000000 0000000000000000 c00000000a3bbf40 0000000000000003 [172.851482] GPR24: 0000000000000000 c0000008ba097400 c0000000161e0d00 c00000000a3bb600 [172.851482] GPR28: c000000015d5c700 00000000000000001 0000000082384090 c0000008ba0020d8 [172.851549] NIP [c00000000013d600] power_pmu_disable+0x270/0x280 [172.851557] LR [c00000000013d5a4] power_pmu_disable+0x214/0x280 [172.851565] Call Trace: [172.851568] [c000000017687b00] [c00000000013d5a4] power_pmu_disable+0x214/0x280 (unreliable) [172.851579] [c000000017687b40] [c0000000003403ac] perf_pmu_disable+0x4c/0x60 [172.851588] [c000000017687b60] [c0000000003445e4] __perf_event_task_sched_out+0x1d4/0x660 [172.851596] [c000000017687c50] [c000000000d1175c] __schedule+0xbcc/0x12a0 [172.851602] [c000000017687d60] [c000000000d11ea8] schedule+0x78/0x140 [172.851608] [c000000017687d90] [c0000000001a8080] sys_sched_yield+0x20/0x40 [172.851615] [c000000017687db0] [c000000000334dc] system_call_exception+0x18c/0x380 [172.851622] [c000000017687e10] [c0000000000c74c] system_call_common+0xec/0x268 The warning indicates that MSR_EE being set(interrupt enabled) when there was an overflown PMC detected. This could happen in power_pmu_disable since it runs under interrupt soft disable condition (local_irq_save) and not with interrupts hard disabled. commit 2c9ac51b850d ("powerpc/perf: Fix PMU callbacks to clear pending PMI before resetting an overflown PMC") intended to clear PMI pending bit in Paca when disabling the PMU. It could happen that PMC gets overflown while code is in power_pmu_disable callback function. Hence add a check to see if PMI pending bit is set in Paca before clearing it via clear_pmi_pending.
CVE-2022-48755	In the Linux kernel, the following vulnerability has been resolved: powerpc64/bpf: Limit 'ldbrx' to processors compliant with ISA v2.06 Johan reported the below crash with test_bpf on ppc64 e5500: test_bpf: #296 ALU_END_FROM_LE 64: 0x0123456789abcdef -> 0x67452301 jited:1 Oops: Exception in kernel mode, sig: 4 [#1] BE PAGE_SIZE=4K SMP NR_CPUS=24 QEMU e500 Modules linked in: test_bpf(+) CPU: 0 PID: 76 Comm: insmod Not tainted 5.14.0-03771-g98c2059e008a-dirty #1 NIP: 8000000000061c3c LR: 800000000006dea64 CTR: 8000000000061c18 REGS: c0000000032d3420 TRAP: 0700 Not tainted (5.14.0-03771-g98c2059e008a-dirty) MSR: 0000000080089000 <EE,ME> CR: 88002822 XER: 20000000 IRQMASK: 0 <...> NIP [8000000000061c3c] 0x8000000000061c3c LR [800000000006dea64] __run_one+0x104/0x17c [test_bpf] Call Trace: __run_one+0x60/0x17c [test_bpf] (unreliable) .test_bpf_init+0x6a8/0xdc8 [test_bpf] .do_one_initcall+0x6c/0x28c .do_init_module+0x68/0x28c .load_module+0x2460/0x2abc __do_sys_init_module+0x120/0x18c .system_call_exception+0x110/0x1b8 system_call_common+0xf0/0x210 --- interrupt: c00 at 0x101d0acc <...> ---[end trace 47b2bf19090bb3d0]--- Illegal instruction The illegal instruction turned out to be 'ldbrx' emitted for BPF_FROM_L B E, which was only introduced in ISA v2.06. Guard use of the same and implement an alternative approach for older processors.

CVE ID	Description
CVE-2022-48758	In the Linux kernel, the following vulnerability has been resolved: scsi: bnx2fc: Flush destroy_work queue before calling bnx2fc_interface_put() The bnx2fc_destroy() functions are removing the interface before calling destroy_work. This results multiple WARNings from sysfs_remove_group() as the controller rport device attributes are removed too early. Replace the fcoe_port's destroy_work queue. It's not needed. The problem is easily reproducible with the following steps. Example: <pre>\$ dmesg -w & \$ systemctl enable --now fcoe \$ fipvlan -s -c ens2f1 \$ fcoeadm -d ens2f1.802 [583.464488] host2: libfc: Link down on port (7500a1) [583.472651] bnx2fc: 7500a1 - rport not created Yet!! [583.490468] -----[cut here]----- [583.538725] sysfs group 'power' not found for kobject 'rport-2:0-0' [583.568814] WARNING: CPU: 3 PID: 192 at fs/sysfs/group.c:279 sysfs_remove_group+0x6f/0x80 [583.607130] Modules linked in: dm_service_time 8021q garp mrp stp llc bnx2fc cnic uio rpcsec_gss_krb5 auth_rpcgss nfsv4 ... [583.942994] CPU: 3 PID: 192 Comm: kworker/3:2 Kdump: loaded Not tainted 5.14.0-39.el9.x86_64 #1 [583.984105] Hardware name: HP ProLiant DL120 G7, BIOS J01 07/01/2013 [584.016535] Workqueue: fc_wq_2 fc_rport_final_delete [scsi_transport_fc] [584.050691] RIP: 0010:sysfs_remove_group+0x6f/0x80 [584.074725] Code: ff 5b 48 89 ef 5d 41 5c e9 ee c0 ff ff 48 89 ef e8 f6 b8 ff ff eb d1 49 8b 14 24 48 8b 33 48 c7 c7 ... [584.162586] RSP: 0018:ffffb567c15afdc0 EFLAGS: 00010282 [584.188225] RAX: 0000000000000000 RBX: ffffffff8e4220 RCX: 0000000000000000 [584.221053] RDX: ffff8c1586ce84c0 RSI: ffff8c1586cd7cc0 RDI: ffff8c1586cd7cc0 [584.255089] RBP: 0000000000000000 R08: 0000000000000000 R09: ffff567c15afc00 [584.287954] R10: ffff567c15afb8 R11: ffffffff8bfe7f28 R12: ffff8c1486326400 [584.322356] R13: ffff8c1486326480 R14: ffff8c1483a4a000 R15: 0000000000000004 [584.355379] FS: 0000000000000000(0000) GS:ffff8c1586cc0000(0000) knlGS:0000000000000000 [584.394419] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [584.421123] CR2: 00007fe95a6f7840 CR3: 0000000107674002 CR4: 00000000000606e0 [584.454888] Call Trace: [584.466108] device_del+0xb2/0x3e0 [584.481701] device_unregister+0x13/0x60 [584.501306] bsg_unregister_queue+0x5b/0x80 [584.522029] bsg_remove_queue+0x1c/0x40 [584.541884] fc_rport_final_delete+0xf3/0x1d0 [scsi_transport_fc] [584.573823] process_one_work+0x1e3/0x3b0 [584.592396] worker_thread+0x50/0x3b0 [584.609256] ? rescuer_thread+0x370/0x370 [584.628877] kthread+0x149/0x170 [584.643673] ? set_kthread_struct+0x40/0x40 [584.662909] ret_from_fork+0x22/0x30 [584.680002] ---[end trace 53575cefa942ece]--- </pre>
CVE-2022-48761	In the Linux kernel, the following vulnerability has been resolved: usb: xhci-plat: fix crash when suspend if remote wake enable Crashed at i.mx8qm platform when suspend if enable remote wakeup Internal error: synchronous external abort: 96000210 [#1] PREEMPT SMP Modules linked in: CPU: 2 PID: 244 Comm: kworker/u12:6 Not tainted 5.15.5-dirty #12 Hardware name: Freescale i.MX8QM MEK (DT) Workqueue: events_unbound async_run_entry_fn pstate: 600000c5 (nZCv daIF -PAN -UAO -TCO -DIT -SSBS BTYP=) pc : xhci_disable_hub_port_wake.isra.62+0x60/0xf8 lr : xhci_disable_hub_port_wake.isra.62+0x34/0xf8 sp : ffff80001394bbf0 x29: ffff80001394bbf0 x28: 0000000000000000 x27: ffff00081193b578 x26: ffff00081193b570 x25: 0000000000000000 x24: 0000000000000000 x23: ffff00081193a29c x22: 0000000000020001 x21: 0000000000000001 x20: 0000000000000000 x19: ffff800014e90490 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000002 x12: 0000000000000000 x11: 0000000000000000 x10: 0000000000000960 x9: ffff80001394baa0 x8: ffff0008145d1780 x7: ffff0008f95b8e80 x6: 000000001853b453 x5: 0000000000000496 x4: 0000000000000000 x3: ffff00081193a29c x2: 0000000000000001 x1: 0000000000000000 x0: ffff000814591620 Call trace: xhci_disable_hub_port_wake.isra.62+0x60/0xf8 xhci_suspend+0x58/0x510 xhci_plat_suspend+0x50/0x78 platform_pm_suspend+0x2c/0x78 dpm_run_callback.isra.25+0x50/0xe8 __device_suspend+0x108/0x3c0 The basic flow: 1. run time suspend call xhci_suspend, xhci parent devices gate the clock. 2. echo mem >/sys/power/state, system _device_suspend call xhci_suspend 3. xhci_suspend call xhci_disable_hub_port_wake, which access register, but clock already gated by run time suspend. This problem was hidden by power domain driver, which call run time resume before it. But the below commit remove it and make this issue happen. commit c1df456d0f06e ("PM: domains: Don't runtime resume devices at genpd_prepare()") This patch call run time resume before suspend to make sure clock is on before access register. Test-by: Abel Vesa <abel.vesa@nxp.com>

CVE ID	Description
CVE-2022-48763	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Forcibly leave nested virt when SMM state is toggled Forcibly leave nested virtualization operation if userspace toggles SMM state via KVM_SET_VCPU_EVENTS or KVM_SYNC_X86_EVENTS. If userspace forces the vCPU out of SMM while it's post-VMXON and then injects an SMI, vmx_enter_smm() will overwrite vmx->nested.smm.vmxon and end up with both vmxon=false and smm.vmxon=false, but all other nVMX state allocated. Don't attempt to gracefully handle the transition as (a) most transitions are nonsencial, e.g. forcing SMM while L2 is running, (b) there isn't sufficient information to handle all transitions, e.g. SVM wants access to the SMRAM save state, and (c) KVM_SET_VCPU_EVENTS must precede KVM_SET_NESTED_STATE during state restore as the latter disallows putting the vCPU into L2 if SMM is active, and disallows tagging the vCPU as being post-VMXON in SMM if SMM is not active. Abuse of KVM_SET_VCPU_EVENTS manifests as a WARN and memory leak in nVMX due to failure to free vmcs01's shadow VMCS, but the bug goes far beyond just a memory leak, e.g. toggling SMM on while L2 is active puts the vCPU in an architecturally impossible state. WARNING: CPU: 0 PID: 3606 at free_loaded_vmcs arch/x86/kvm/vmx/vmx.c:2665 [inline] WARNING: CPU: 0 PID: 3606 at free_loaded_vmcs+0x158/0x1a0 arch/x86/kvm/vmx/vmx.c:2656 Modules linked in: CPU: 1 PID: 3606 Comm: syz-executor725 Not tainted 5.17.0-rc1-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:free_loaded_vmcs arch/x86/kvm/vmx/vmx.c:2665 [inline] RIP: 0010:free_loaded_vmcs+0x158/0x1a0 arch/x86/kvm/vmx/vmx.c:2656 Code: <0f> 0b eb b3 e8 8f 4d 9f 00 e9 f7 fe ff 48 89 df e8 92 4d 9f 00 Call Trace: <TASK> kvm_arch_vcpu_destroy+0x72/0x2f0 arch/x86/kvm/x86.c:11123 kvm_vcpu_destroy arch/x86/kvm/../../../../virt/kvm/kvm_main.c:441 [inline] kvm_destroy_vcpus+0x11f/0x290 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:460 kvm_free_vcpus arch/x86/kvm/x86.c:11564 [inline] kvm_arch_destroy_vm+0x2e8/0x470 arch/x86/kvm/x86.c:11676 kvm_destroy_vm arch/x86/kvm/../../../../virt/kvm/kvm_main.c:1217 [inline] kvm_put_kvm+0x4fa/0xb00 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:1250 kvm_vm_release+0x3f/0x50 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:1273 __fput+0x286/0x9f0 fs/file_table.c:311 task_work_run+0xdd/0x1a0 kernel/task_work.c:164 exit_task_work include/linux/task_work.h:32 [inline] do_exit+0xb29/0x2a30 kernel/exit.c:806 do_group_exit+0xd2/0x2f0 kernel/exit.c:935 get_signal+0x4b0/0x28c0 kernel/signal.c:2862 arch_do_signal_or_restart+0x2a9/0x1c40 arch/x86/kernel/signal.c:868 handle_signal_work kernel/entry/common.c:148 [inline] exit_to_user_mode_loop kernel/entry/common.c:172 [inline] exit_to_user_mode_prepare+0x17d/0x290 kernel/entry/common.c:207 __syscall_exit_to_user_mode_work kernel/entry/common.c:289 [inline] syscall_exit_to_user_mode+0x19/0x60 kernel/entry/common.c:300 do_syscall_64+0x42/0xb0 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x44/0xae </TASK>
CVE-2022-48764	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Free kvm_cpuid_entry2 array on post-KVM_RUN KVM_SET_CPUID{,2} Free the "struct kvm_cpuid_entry2" array on successful post-KVM_RUN KVM_SET_CPUID{,2} to fix a memory leak, the callers of kvm_set_cpuid() free the array only on failure. BUG: memory leak unreferenced object 0xffff88810963a800 (size 2048): comm "syz-executor025", pid 3610, jiffies 4294944928 (age 8.080s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 0d 00 00 00 47 65 6e 75 6e 74 65 6c 69 6e 65 49 00 00 00 00 GenuntelineI.... backtrace: [<ffffff814948ee>] kmalloc_node include/linux/slab.h:604 [inline] [<ffffff814948ee>] kvmalloc_node+0x3e/0x100 mm/util.c:580 [<ffffff814950f2>] kvmalloc include/linux/slab.h:732 [inline] [<ffffff814950f2>] vmemdup_user+0x22/0x100 mm/util.c:199 [<ffffff8109f5ff>] kvm_vcpu_ioctl_set_cpuid2+0x8f/0xf0 arch/x86/kvm/cpuid.c:423 [<ffffff810711b9>] kvm_arch_vcpu_ioctl+0xb99/0x1e60 arch/x86/kvm/x86.c:5251 [<ffffff8103e92d>] kvm_vcpu_ioctl+0x4ad/0x950 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:4066 [<ffffff815afacc>] vfs_ioctl fs/ioctl.c:51 [inline] [<ffffff815afacc>] __do_sys_ioctl fs/ioctl.c:874 [inline] [<ffffff815afacc>] __se_sys_ioctl fs/ioctl.c:860 [inline] [<ffffff815afacc>] __x64_sys_ioctl+0xfc/0x140 fs/ioctl.c:860 [<ffffff844a3335>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<ffffff844a3335>] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 [<ffffff84600068>] entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2022-48767	In the Linux kernel, the following vulnerability has been resolved: ceph: properly put ceph_string reference after async create attempt The reference acquired by try_prep_async_create is currently leaked. Ensure we put it.

CVE ID	Description
CVE-2022-48769	In the Linux kernel, the following vulnerability has been resolved: efivar: runtime: avoid EFIv2 runtime services on Apple x86 machines Aditya reports [0] that his recent MacbookPro crashes in the firmware when using the variable services at runtime. The culprit appears to be a call to QueryVariableInfo(), which we did not use to call on Apple x86 machines in the past as they only upgraded from EFI v1.10 to EFI v2.40 firmware fairly recently, and QueryVariableInfo() (along with UpdateCapsule() et al) was added in EFI v2.00. The only runtime service introduced in EFI v2.00 that we actually use in Linux is QueryVariableInfo(), as the capsule based ones are optional, generally not used at runtime (all the LVFS/fwupd firmware update infrastructure uses helper EFI programs that invoke capsule update at boot time, not runtime), and not implemented by Apple machines in the first place. QueryVariableInfo() is used to 'safely' set variables, i.e., only when there is enough space. This prevents machines with buggy firmwares from corrupting their NVRAMs when they run out of space. Given that Apple machines have been using EFI v1.10 services only for the longest time (the EFI v2.0 spec was released in 2006, and Linux support for the newly introduced runtime services was added in 2011, but the MacbookPro12,1 released in 2015 still claims to be EFI v1.10 only), let's avoid the EFI v2.0 ones on all Apple x86 machines. [0] https://lore.kernel.org/all/6D757C75-65B1-468B-842D-10410081A8E4@live.com/
CVE-2022-48770	In the Linux kernel, the following vulnerability has been resolved: bpf: Guard against accessing NULL pt_regs in bpf_get_task_stack() task_pt_regs() can return NULL on powerpc for kernel threads. This is then used in __bpf_get_stack() to check for user mode, resulting in a kernel oops. Guard against this by checking return value of task_pt_regs() before trying to obtain the call chain.
CVE-2022-48772	In the Linux kernel, the following vulnerability has been resolved: media: lgdt3306a: Add a check against null-pointer-def The driver should check whether the client provides the platform_data. The following log reveals it: [29.610324] BUG: KASAN: null-ptr-deref in kmemdup+0x30/0x40 [29.610730] Read of size 40 at addr 0000000000000000 by task bash/414 [29.612820] Call Trace: [29.613030] <TASK> [29.613201] dump_stack_lvl+0x56/0x6f [29.613496] ? kmemdup+0x30/0x40 [29.613754] print_report.cold+0x494/0x6b7 [29.614082] ? kmemdup+0x30/0x40 [29.614340] kasan_report+0x8a/0x190 [29.614628] ? kmemdup+0x30/0x40 [29.614888] kasan_check_range+0x14d/0x1d0 [29.615213] memcpy+0x20/0x60 [29.615454] kmemdup+0x30/0x40 [29.615700] lgdt3306a_probe+0x52/0x310 [29.616339] i2c_device_probe+0x951/0xa90
CVE-2022-48778	In the Linux kernel, the following vulnerability has been resolved: mtd: rawnand: gpmi: don't leak PM reference in error path If gpmi_nfc_apply_timings() fails, the PM runtime usage counter must be dropped.
CVE-2022-48780	In the Linux kernel, the following vulnerability has been resolved: net/smc: Avoid overwriting the copies of clcsock callback functions The callback functions of clcsock will be saved and replaced during the fallback. But if the fallback happens more than once, then the copies of these callback functions will be overwritten incorrectly, resulting in a loop call issue: clcsock->sk_error_report - smc_fback_error_report() <----- - smc_fback_forward_wakeup() (loop) - clcsock_callback() (incorrectly overwritten) - smc->clcsock_error_report() ----- So this patch fixes the issue by saving these function pointers only once in the fallback and avoiding overwriting.
CVE-2022-48783	In the Linux kernel, the following vulnerability has been resolved: net: dsa: lantiq_gswip: fix use after free in gswip_remove() of_node_put(priv->ds->slave_mii_bus->dev.of_node) should be done before mdiobus_free(priv->ds->slave_mii_bus).
CVE-2022-48784	In the Linux kernel, the following vulnerability has been resolved: cfg80211: fix race in netlink owner interface destruction My previous fix here to fix the deadlock left a race where the exact same deadlock (see the original commit referenced below) can still happen if cfg80211_destroy_ifaces() already runs while nl80211_netlink_notify() is still marking some interfaces as nl_owner_dead. The race happens because we have two loops here - first we dev_close() all the netdevs, and then we destroy them. If we also have two netdevs (first one need only be a wdev though) then we can find one during the first iteration, close it, and go to the second iteration -- but then find two, and try to destroy also the one we didn't close yet. Fix this by only iterating once.
CVE-2022-48787	In the Linux kernel, the following vulnerability has been resolved: iwlwifi: fix use-after-free If no firmware was present at all (or, presumably, all of the firmware files failed to parse), we end up unbinding by calling device_release_driver(), which calls remove(), which then in iwlwifi calls iwl_drv_stop(), freeing the 'drv' struct. However the new code I added will still erroneously access it after it was freed. Set 'failure=false' in this case to avoid the access, all data was already freed anyway.
CVE-2022-48788	In the Linux kernel, the following vulnerability has been resolved: nvme-rdma: fix possible use-after-free in transport error_recovery work While nvme_rdma_submit_async_event_work is checking the ctrl and queue state before preparing the AER command and scheduling io_work, in order to fully prevent a race where this check is not reliable the error recovery work must flush async_event_work before continuing to destroy the admin queue after setting the ctrl state to RESETING such that there is no race .submit_async_event and the error recovery handler itself changing the ctrl state.

CVE ID	Description
CVE-2022-48789	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix possible use-after-free in transport error_recovery work While nvme_tcp_submit_async_event_work is checking the ctrl and queue state before preparing the AER command and scheduling io_work, in order to fully prevent a race where this check is not reliable the error recovery work must flush async_event_work before continuing to destroy the admin queue after setting the ctrl state to RESETING such that there is no race .submit_async_event and the error recovery handler itself changing the ctrl state.
CVE-2022-48790	In the Linux kernel, the following vulnerability has been resolved: nvme: fix a possible use-after-free in controller reset during load Unlike .queue_rq, in .submit_async_event drivers may not check the ctrl readiness for AER submission. This may lead to a use-after-free condition that was observed with nvme-tcp. The race condition may happen in the following scenario: 1. driver executes its reset_ctrl_work 2. -> nvme_stop_ctrl - flushes ctrl async_event_work 3. ctrl sends AEN which is received by the host, which in turn schedules AEN handling 4. teardown admin queue (which releases the queue socket) 5. AEN processed, submits another AER, calling the driver to submit 6. driver attempts to send the cmd ==> use-after-free In order to fix that, add ctrl state check to validate the ctrl is actually able to accept the AER submission. This addresses the above race in controller resets because the driver during teardown should: 1. change ctrl state to RESETING 2. flush async_event_work (as well as other async work elements) So after 1,2, any other AER command will find the ctrl state to be RESETING and bail out without submitting the AER.
CVE-2022-48793	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: nSVM: fix potential NULL dereference on nested migration Turns out that due to review feedback and/or rebases I accidentally moved the call to nested_svm_load_cr3 to be too early, before the NPT is enabled, which is very wrong to do. KVM can't even access guest memory at that point as nested NPT is needed for that, and of course it won't initialize the walk_mmu, which is main issue the patch was addressing. Fix this for real.
CVE-2022-48796	In the Linux kernel, the following vulnerability has been resolved: iommu: Fix potential use-after-free during probe Kasan has reported the following use after free on dev->iommu. when a device probe fails and it is in process of freeing dev->iommu in dev_iommu_free function, a deferred_probe_work_func runs in parallel and tries to access dev->iommu->fwspec in of_iommu_configure path thus causing use after free. BUG: KASAN: use-after-free in of_iommu_configure+0xb4/0x4a4 Read of size 8 at addr fffff87a2f1acb8 by task kworker/u16:2/153 Workqueue: events_unbound deferred_probe_work_func Call trace: dump_backtrace+0x0/0x33c show_stack+0x18/0x24 dump_stack_lvl+0x16c/0x1e0 print_address_description+0x84/0x39c __kasan_report+0x184/0x308 kasan_report+0x50/0x78 __asan_load8+0xc0/0xc4 of_iommu_configure+0xb4/0x4a4 of_dma_configure_id+0x2fc/0x4d4 platform_dma_configure+0x40/0x5c really_probe+0x1b4/0xb74 driver_probe_device+0x11c/0x228 __device_attach_driver+0x14c/0x304 bus_for_each_drv+0x124/0x1b0 __device_attach+0x25c/0x334 device_initial_probe+0x24/0x34 bus_probe_device+0x78/0x134 deferred_probe_work_func+0x130/0x1a8 process_one_work+0x4c8/0x970 worker_thread+0x5c8/0xaec kthread+0x1f8/0x220 ret_from_fork+0x10/0x18 Allocated by task 1: __kasan_kmalloc+0xd4/0x114 __kasan_kmalloc+0x10/0x1c kmem_cache_alloc_trace+0xe4/0x3d4 __iommu_probe_device+0x90/0x394 probe_iommu_group+0x70/0x9c bus_for_each_dev+0x11c/0x19c bus_iommu_probe+0xb8/0x7d4 bus_set_iommu+0xcc/0x13c arm_smmu_bus_init+0x44/0x130 [arm_smmu] arm_smmu_device_probe+0xb88/0xc54 [arm_smmu] platform_drv_probe+0xe4/0x13c really_probe+0x2c8/0xb74 driver_probe_device+0x11c/0x228 device_driver_attach+0xf0/0x16c __driver_attach+0x80/0x320 bus_for_each_dev+0x11c/0x19c driver_attach+0x38/0x48 bus_add_driver+0x1dc/0x3a4 driver_register+0x18c/0x244 __platform_driver_register+0x88/0x9c init_module+0x64/0xff4 [arm_smmu] do_one_initcall+0x17c/0x2f0 do_init_module+0xe8/0x378 load_module+0x3f80/0x4a40 __se_sys_finit_module+0x1a0/0x1e4 __arm64_sys_finit_module+0x44/0x58 el0_svc_common+0x100/0x264 do_el0_svc+0x38/0xa4 el0_svc+0x20/0x30 el0_sync_handler+0x68/0xac el0_sync+0x160/0x180 Freed by task 1: kasan_set_track+0x4c/0x84 kasan_set_free_info+0x28/0x4c __kasan_slab_free+0x120/0x15c __kasan_slab_free+0x18/0x28 slab_free_freelist_hook+0x204/0x2fc kfree+0xfc/0x3a4 __iommu_probe_device+0x284/0x394 probe_iommu_group+0x70/0x9c bus_for_each_dev+0x11c/0x19c bus_iommu_probe+0xb8/0x7d4 bus_set_iommu+0xcc/0x13c arm_smmu_bus_init+0x44/0x130 [arm_smmu] arm_smmu_device_probe+0xb88/0xc54 [arm_smmu] platform_drv_probe+0xe4/0x13c really_probe+0x2c8/0xb74 driver_probe_device+0x11c/0x228 device_driver_attach+0xf0/0x16c __driver_attach+0x80/0x320 bus_for_each_dev+0x11c/0x19c driver_attach+0x38/0x48 bus_add_driver+0x1dc/0x3a4 driver_register+0x18c/0x244 __platform_driver_register+0x88/0x9c init_module+0x64/0xff4 [arm_smmu] do_one_initcall+0x17c/0x2f0 do_init_module+0xe8/0x378 load_module+0x3f80/0x4a40 __se_sys_finit_module+0x1a0/0x1e4 __arm64_sys_finit_module+0x44/0x58 el0_svc_common+0x100/0x264 do_el0_svc+0x38/0xa4 el0_svc+0x20/0x30 el0_sync_handler+0x68/0xac el0_sync+0x160/0x180 Fix this by setting dev->iommu to NULL first and then freeing dev_iommu structure in dev_iommu_free function.

CVE ID	Description
CVE-2022-48799	In the Linux kernel, the following vulnerability has been resolved: perf: Fix list corruption in perf_cgroup_switch() There's list corruption on cgrp_cpuctx_list. This happens on the following path: perf_cgroup_switch: list_for_each_entry(cgrp_cpuctx_list) cpu_ctx_sched_in ctx_sched_in ctx_pinned_sched_in merge_sched_in perf_cgroup_event_disable: remove the event from the list Use list_for_each_entry_safe() to allow removing an entry during iteration.
CVE-2022-48800	In the Linux kernel, the following vulnerability has been resolved: mm: vmscan: remove deadlock due to throttling failing to make progress A soft lockup bug in kcompactd was reported in a private bugzilla with the following visible in dmesg; watchdog: BUG: soft lockup - CPU#33 stuck for 26s! [kcompactd0:479] watchdog: BUG: soft lockup - CPU#33 stuck for 52s! [kcompactd0:479] watchdog: BUG: soft lockup - CPU#33 stuck for 78s! [kcompactd0:479] watchdog: BUG: soft lockup - CPU#33 stuck for 104s! [kcompactd0:479] The machine had 256G of RAM with no swap and an earlier failed allocation indicated that node 0 where kcompactd was run was potentially unreclaimable; Node 0 active_anon:29355112kB inactive_anon:2913528kB active_file:0kB inactive_file:0kB unevictable:64kB isolated(anon):0kB isolated(file):0kB mapped:8kB dirty:0kB writeback:0kB shmem:26780kB shmem_thp: 0kB shmem_pmdmapped: 0kB anon_thp: 23480320kB writeback_tmp:0kB kernel_stack:2272kB pagetables:24500kB all_unreclaimable? yes Vlastimil Babka investigated a crash dump and found that a task migrating pages was trying to drain PCP lists; PID: 52922 TASK: ffff969f820e5000 CPU: 19 COMMAND: "kworker/u128:3" Call Trace: __schedule schedule schedule_timeout wait_for_completion __flush_work __drain_all_pages __alloc_pages_slowpath.constprop.114 __alloc_pages alloc_migration_target migrate_pages migrate_to_node do_migrate_pages cpuset_migrate_mm_workfn process_one_work worker_thread kthread ret_from_fork This failure is specific to CONFIG_PREEMPT=n builds. The root of the problem is that kcompact0 is not rescheduling on a CPU while a task that has isolated a large number of the pages from the LRU is waiting on kcompact0 to reschedule so the pages can be released. While shrink_inactive_list() only loops once around too_many_isolated, reclaim can continue without rescheduling if sc->skipped_deactivate == 1 which could happen if there was no file LRU and the inactive anon list was not low.
CVE-2022-48801	In the Linux kernel, the following vulnerability has been resolved: iio: buffer: Fix file related error handling in IIO_BUFFER_GET_FD_IOCTL If we fail to copy the just created file descriptor to userland, we try to clean up by putting back 'fd' and freeing 'ib'. The code uses put_unused_fd() for the former which is wrong, as the file descriptor was already published by fd_install() which gets called internally by anon_inode_getfd(). This makes the error handling code leaving a half cleaned up file descriptor table around and a partially destructed 'file' object, allowing userland to play use-after-free tricks on us, by abusing the still usable fd and making the code operate on a dangling 'file->private_data' pointer. Instead of leaving the kernel in a partially corrupted state, don't attempt to explicitly clean up and leave this to the process exit path that'll release any still valid fds, including the one created by the previous call to anon_inode_getfd(). Simply return -EFAULT to indicate the error.

CVE ID	Description
CVE-2022-48802	In the Linux kernel, the following vulnerability has been resolved: fs/proc: task_mmu.c: don't read mapcount for migration entry The syzbot reported the below BUG: kernel BUG at include/linux/page-flags.h:785! invalid opcode: 0000 [#1] PREEMPT SMP KASAN CPU: 1 PID: 4392 Comm: syz-executor560 Not tainted 5.16.0-rc6-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:PageDoubleMap include/linux/page-flags.h:785 [inline] RIP: 0010: __page_mapcount+0x2d2/0x350 mm/util.c:744 Call Trace: page_mapcount include/linux/mm.h:837 [inline] smaps_account+0x470/0xb10 fs/proc/task_mmu.c:466 smaps_pte_entry fs/proc/task_mmu.c:538 [inline] smaps_pte_range+0x611/0x1250 fs/proc/task_mmu.c:601 walk_pmd_range mm/pagewalk.c:128 [inline] walk_pud_range mm/pagewalk.c:205 [inline] walk_p4d_range mm/pagewalk.c:240 [inline] walk_pg_d_range mm/pagewalk.c:277 [inline] __walk_page_range+0xe23/0x1ea0 mm/pagewalk.c:379 walk_page_vma+0x277/0x350 mm/pagewalk.c:530 smap_gather_stats.part.0+0x148/0x260 fs/proc/task_mmu.c:768 smap_gather_stats fs/proc/task_mmu.c:741 [inline] show_smap+0xc6/0x440 fs/proc/task_mmu.c:822 seq_read_iter+0xbb0/0x1240 fs/seq_file.c:272 seq_read+0x3e0/0x5b0 fs/seq_file.c:162 vfs_read+0x1b5/0x600 fs/read_write.c:479 ksys_read+0x12d/0x250 fs/read_write.c:619 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae The reproducer was trying to read /proc/\$PID/smaps when calling MADV_FREE at the mean time. MADV_FREE may split THPs if it is called for partial THP. It may trigger the below race: CPU A CPU B ----- smaps walk: MADV_FREE: page_mapcount() PageCompound() split_huge_page() page = compound_head(page) PageDoubleMap(page) When calling PageDoubleMap() this page is not a tail page of THP anymore so the BUG is triggered. This could be fixed by elevated refcount of the page before calling mapcount, but that would prevent it from counting migration entries, and it seems overkilling because the race just could happen when PMD is split so all PTE entries of tail pages are actually migration entries, and smaps_account() does treat migration entries as mapcount == 1 as Kirill pointed out. Add a new parameter for smaps_account() to tell this entry is migration entry then skip calling page_mapcount(). Don't skip getting mapcount for device private entries since they do track references with mapcount. Pagemap also has the similar issue although it was not reported. Fixed it as well. [shy828301@gmail.com: v4] [nathan@kernel.org: avoid unused variable warning in pagemap_pmd_range()]
CVE-2022-48805	In the Linux kernel, the following vulnerability has been resolved: net: usb: ax88179_178a: Fix out-of-bounds accesses in RX fixup ax88179_rx_fixup() contains several out-of-bounds accesses that can be triggered by a malicious (or defective) USB device, in particular: - The metadata array (hdr_off..hdr_off+2*pkt_cnt) can be out of bounds, causing OOB reads and (on big-endian systems) OOB endianness flips. - A packet can overlap the metadata array, causing a later OOB endianness flip to corrupt data used by a cloned SKB that has already been handed off into the network stack. - A packet SKB can be constructed whose tail is far beyond its end, causing out-of-bounds heap data to be considered part of the SKB's data. I have tested that this can be used by a malicious USB device to send a bogus ICMPv6 Echo Request and receive an ICMPv6 Echo Reply in response that contains random kernel heap data. It's probably also possible to get OOB writes from this on a little-endian system somehow - maybe by triggering skb_cow() via IP options processing -, but I haven't tested that.
CVE-2022-48806	In the Linux kernel, the following vulnerability has been resolved: eeprom: ee1004: limit i2c reads to I2C_SMBUS_BLOCK_MAX Commit effa453168a7 ("i2c: i801: Don't silently correct invalid transfer size") revealed that ee1004_eeprom_read() did not properly limit how many bytes to read at once. In particular, i2c_smbus_read_i2c_block_data_or_emulated() takes the length to read as an u8. If count == 256 after taking into account the offset and page boundary, the cast to u8 overflows. And this is common when user space tries to read the entire EEPROM at once. To fix it, limit each read to I2C_SMBUS_BLOCK_MAX (32) bytes, already the maximum length i2c_smbus_read_i2c_block_data_or_emulated() allows.
CVE-2022-48807	In the Linux kernel, the following vulnerability has been resolved: ice: Fix KASAN error in LAG NETDEV_UNREGISTER handler Currently, the same handler is called for both a NETDEV_BONDING_INFO LAG unlink notification as for a NETDEV_UNREGISTER call. This is causing a problem though, since the netdev_notifier_info passed has a different structure depending on which event is passed. The problem manifests as a call trace from a BUG: KASAN stack-out-of-bounds error. Fix this by creating a handler specific to NETDEV_UNREGISTER that only is passed valid elements in the netdev_notifier_info struct for the NETDEV_UNREGISTER event. Also included is the removal of an unbalanced dev_put on the peer_netdev and related braces.
CVE-2022-48809	In the Linux kernel, the following vulnerability has been resolved: net: fix a memleak when uncloning an skb dst and its metadata When uncloning an skb dst and its associated metadata, a new dst+metadata is allocated and later replaces the old one in the skb. This is helpful to have a non-shared dst+metadata attached to a specific skb. The issue is the uncloned dst+metadata is initialized with a refcount of 1, which is increased to 2 before attaching it to the skb. When tun_dst_unclone returns, the dst+metadata is only referenced from a single place (the skb) while its refcount is 2. Its refcount will never drop to 0 (when the skb is consumed), leading to a memory leak. Fix this by removing the call to dst_hold in tun_dst_unclone, as the dst+metadata refcount is already 1.

CVE ID	Description
CVE-2022-48810	In the Linux kernel, the following vulnerability has been resolved: ipmr,ip6mr: acquire RTNL before calling ip[6]mr_free_table() on failure path ip[6]mr_free_table() can only be called under RTNL lock. RTNL: assertion failed at net/core/dev.c (10367) WARNING: CPU: 1 PID: 5890 at net/core/dev.c:10367 unregister_netdevice_many+0x1246/0x1850 net/core/dev.c:10367 Modules linked in: CPU: 1 PID: 5890 Comm: syz-executor.2 Not tainted 5.16.0-syzkaller-11627-g422ee58dc0ef #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:unregister_netdevice_many+0x1246/0x1850 net/core/dev.c:10367 Code: 0f 85 9b ee ff ff e8 69 07 4b fa ba 7f 28 00 00 48 c7 c6 00 90 ae 8a 48 c7 c7 40 90 ae 8a c6 05 6d b1 51 06 01 e8 8c 90 d8 01 <0f> 0b e9 70 ee ff ff e8 3e 07 4b fa 4c 89 e7 e8 86 2a 59 fa e9 ee RSP: 0018:ffff900046ff6e0 EFLAGS: 00010286 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 RDX: ffff888050f51d00 RSI: ffffffff815fa008 RDI: fffff520008dfece RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: ffffffff815f3d6e R11: 0000000000000000 R12: 00000000ffffffffff R13: dffffc0000000000 R14: ffff900046ff750 R15: ffff88807b7dc000 FS: 00007f4ab736e700(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fee0b4f8990 CR3: 000000001e7d2000 CR4: 00000000003506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> mroute_clean_tables+0x244/0xb40 net/ipv6/ip6mr.c:1509 ip6mr_free_table net/ipv6/ip6mr.c:389 [inline] ip6mr_rules_init net/ipv6/ip6mr.c:246 [inline] ip6mr_net_init net/ipv6/ip6mr.c:1306 [inline] ip6mr_net_init+0x3f0/0x4e0 net/ipv6/ip6mr.c:1298 ops_init+0xaf/0x470 net/core/net_namespace.c:140 setup_net+0x54f/0xbb0 net/core/net_namespace.c:331 copy_net_ns+0x318/0x760 net/core/net_namespace.c:475 create_new_namespaces+0x3f6/0xb20 kernel/nsproxy.c:110 copy_namespaces+0x391/0x450 kernel/nsproxy.c:178 copy_process+0x2e0c/0x7300 kernel/fork.c:2167 kernel_clone+0xe7/0xab0 kernel/fork.c:2555 __do_sys_clone+0xc8/0x110 kernel/fork.c:2672 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f4ab89f9059 Code: Unable to access opcode bytes at RIP 0x7f4ab89f902f. RSP: 002b:00007f4ab736e118 EFLAGS: 00000206 ORIG_RAX: 0000000000000038 RAX: ffffffff815fa008 RBX: 00007f4ab8b0bf60 RCX: 00007f4ab89f9059 RDX: 0000000020000280 RSI: 0000000020000270 RDI: 0000000040200000 RBP: 00007f4ab8a5308d R08: 0000000020000300 R09: 0000000020000300 R10: 00000000200002c0 R11: 0000000000000206 R12: 0000000000000000 R13: 00007ffc3977cc1f R14: 00007f4ab736e300 R15: 0000000000022000 </TASK>
CVE-2022-48811	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: don't release napi in __ibmvnic_open() If __ibmvnic_open() encounters an error such as when setting link state, it calls release_resources() which frees the napi structures needlessly. Instead, have __ibmvnic_open() only clean up the work it did so far (i.e. disable napi and irq) and leave the rest to the callers. If caller of __ibmvnic_open() is ibmvnic_open(), it should release the resources immediately. If the caller is do_reset() or do_hard_reset(), they will release the resources on the next reset. This fixes following crash that occurred when running the drmgr command several times to add/remove a vnic interface: [102056] ibmvnic 30000003 env3: Disabling rx_scrq[6] irq [102056] ibmvnic 30000003 env3: Disabling rx_scrq[7] irq [102056] ibmvnic 30000003 env3: Replenished 8 pools Kernel attempted to read user page (10) - exploit attempt? (uid: 0) BUG: Kernel NULL pointer dereference on read at 0x00000010 Faulting instruction address: 0xc00000000a3c840 Oops: Kernel access of bad area, sig: 11 [#1] LE PAGE_SIZE=64K MMU=Radix SMP NR_CPUS=2048 NUMA pSeries ... CPU: 9 PID: 102056 Comm: kworker/9:2 Kdump: loaded Not tainted 5.16.0-rc5-autotest-g6441998e2e37 #1 Workqueue: events_long __ibmvnic_reset [ibmvnic] NIP: c00000000a3c840 LR: c0080000029b5378 CTR: c00000000a3c820 REGS: c0000000548e37e0 TRAP: 0300 Not tainted (5.16.0-rc5-autotest-g6441998e2e37) MSR: 8000000000009033 <SF,EE,ME,IR,DR,RI,LE> CR: 28248484 XER: 00000004 CFAR: c0080000029bdd24 DAR: 0000000000000010 DSISR: 40000000 IRQMASK: 0 GPR00: c0080000029b55d0 c0000000548e3a80 c000000028f0200 0000000000000000 ... NIP [c00000000a3c840] napi_enable+0x20/0xc0 LR [c0080000029b5378] __ibmvnic_open+0xf0/0x430 [ibmvnic] Call Trace: [c0000000548e3a80] [0000000000000006] 0x6 (unreliable) [c0000000548e3ab0] [c0080000029b55d0] __ibmvnic_open+0x348/0x430 [ibmvnic] [c0000000548e3b40] [c0080000029bcc28] __ibmvnic_reset+0x500/0xd0 [ibmvnic] [c0000000548e3c60] [c000000000176228] process_one_work+0x288/0x570 [c0000000548e3d00] [c000000000176588] worker_thread+0x78/0x660 [c0000000548e3da0] [c0000000001822f0] kthread+0x1c0/0x1d0 [c0000000548e3e10] [c0000000000cf64] ret_from_kernel_thread+0x5c/0x64 Instruction dump: 7d2948f8 792307e0 4e800020 60000000 3c4c01eb 384239e0 f821ffd1 39430010 38a0ffff e92d1100 f9210028 39200000 <e9030010> f9010020 60420000 e9210020 ---[end trace 5f8033b08fd27706]---

CVE ID	Description
CVE-2022-48823	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Fix refcount issue when LOGO is received during TMF Hung task call trace was seen during LOGO processing. [974.309060] [0000:00:00.0]:[qedf_ah_device_reset:868]: 1:0:2:0: LUN RESET Issued... [974.309065] [0000:00:00.0]:[qedf_initiate_tmf:2422]: tm_flags 0x10 sc_cmd 00000000c16b930f op = 0x2a target_id = 0x2 lun=0 [974.309178] [0000:00:00.0]:[qedf_initiate_tmf:2431]: portid=016900 tm_flags = LUN RESET [974.309222] [0000:00:00.0]:[qedf_initiate_tmf:2438]: orig io_req = 00000000ec78df8f xid = 0x180 ref_cnt = 1. [974.309625] host1: rport 016900: Received LOGO request while in state Ready [974.309627] host1: rport 016900: Delete port [974.309642] host1: rport 016900: work event 3 [974.309644] host1: rport 016900: lld callback ev 3 [974.313243] [0000:61:00.2]: [qedf_execute_tmf:2383]:1: fcport is uploading, not executing flush. [974.313295] [0000:61:00.2]: [qedf_execute_tmf:2400]:1: task mgmt command success... [984.031088] INFO: task jbd2/dm-15-8:7645 blocked for more than 120 seconds. [984.031136] Not tainted 4.18.0-305.el8.x86_64 #1 [984.031166] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [984.031209] jbd2/dm-15-8 D 0 7645 2 0x80004080 [984.031212] Call Trace: [984.031222] __schedule+0x2c4/0x700 [984.031230] ? unfreeze_partials.isra.83+0x16e/0x1a0 [984.031233] ? bit_wait_timeout+0x90/0x90 [984.031235] schedule+0x38/0xa0 [984.031238] io_schedule+0x12/0x40 [984.031240] bit_wait_io+0xd/0x50 [984.031243] __wait_on_bit+0x6c/0x80 [984.031248] ? free_buffer_head+0x21/0x50 [984.031251] out_of_line_wait_on_bit+0x91/0xb0 [984.031257] ? init_wait_var_entry+0x50/0x50 [984.031268] jbd2_journal_commit_transaction+0x112e/0x19f0 [jbd2] [984.031280] kjournald2+0xbd/0x270 [jbd2] [984.031284] ? finish_wait+0x80/0x80 [984.031291] ? commit_timeout+0x10/0x10 [jbd2] [984.031294] kthread+0x116/0x130 [984.031300] ? kthread_flush_work_fn+0x10/0x10 [984.031305] ret_from_fork+0x1f/0x40 There was a ref count issue when LOGO is received during TMF. This leads to one of the I/Os hanging with the driver. Fix the ref count.
CVE-2022-48825	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Add stag_work to all the vports Call trace seen when creating NPIV ports, only 32 out of 64 show online. stag work was not initialized for vport, hence initialize the stag work. WARNING: CPU: 8 PID: 645 at kernel/workqueue.c:1635 __queue_delayed_work+0x68/0x80 CPU: 8 PID: 645 Comm: kworker/8:1 Kdump: loaded Tainted: G IOE ----- -- 4.18.0-348.el8.x86_64 #1 Hardware name: Dell Inc. PowerEdge MX740c/0177V9, BIOS 2.12.2 07/09/2021 Workqueue: events fc_lport_timeout [libfc] RIP: 0010: __queue_delayed_work+0x68/0x80 Code: 89 b2 88 00 00 00 44 89 82 90 00 00 00 48 01 c8 48 89 42 50 41 81 f8 00 20 00 00 75 1d e9 60 24 07 00 44 89 c7 e9 98 f6 ff ff <0> 0b eb c5 0f 0b eb a1 0f 0b eb a7 0f 0b eb ac 44 89 c6 e9 40 23 RSP: 0018:ffffae514bc3be40 EFLAGS: 00010006 RAX: ffff8d25d6143750 RBX: 0000000000000202 RCX: 0000000000000002 RDX: ffff8d2e31383748 RSI: ffff8d25c000d600 RDI: ffff8d2e31383788 RBP: ffff8d2e31380de0 R08: 0000000000000200 R09: ffff8d2e31383750 R10: ffffffff0c957e0 R11: ffff8d2624800000 R12: ffff8d2e31380a58 R13: ffff8d2d915eb000 R14: ffff8d25c499b5c0 R15: ffff8d2e31380e18 FS: 0000000000000000(0000) GS: ffff8d2d1fb00000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055fd0484b8b8 CR3: 00000000fffc10006 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: queue_delayed_work_on+0x36/0x40 qedf_elsct_send+0x57/0x60 [qedf] fc_lport_enter_flogi+0x90/0xc0 [libfc] fc_lport_timeout+0xb7/0x140 [libfc] process_one_work+0x1a7/0x360 ? create_worker+0x1a0/0x1a0 worker_thread+0x30/0x390 ? create_worker+0x1a0/0x1a0 kthread+0x116/0x130 ? kthread_flush_work_fn+0x10/0x10 ret_from_fork+0x35/0x40 ---[end trace 008f00f722f2c2ff]-- Initialize stag work for all the vports.
CVE-2022-48827	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix the behavior of READ near OFFSET_MAX Dan Aloni reports: > Due to commit 8cfb9015280d ("NFS: Always provide aligned buffers to > the RPC read layers") on the client, a read of 0xffff is aligned up > to server rszie of 0x1000. > > As a result, in a test where the server has a file of size > 0x7fffffffffffff, and the client tries to read from the offset > 0x7ffffffffffff000, the read causes loff_t overflow in the server > and it returns an NFS code of EINVAL to the client. The client as > a result indefinitely retries the request. The Linux NFS client does not handle NFS?ERR_INVAL, even though all NFS specifications permit servers to return that status code for a READ. Instead of NFS?ERR_INVAL, have out-of-range READ requests succeed and return a short result. Set the EOF flag in the result to prevent the client from retrying the READ request. This behavior appears to be consistent with Solaris NFS servers. Note that NFSv3 and NFSv4 use u64 offset values on the wire. These must be converted to loff_t internally before use -- an implicit type cast is not adequate for this purpose. Otherwise VFS checks against sb->s_maxbytes do not work properly.
CVE-2022-48828	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix ia_size underflow iattr::ia_size is a loff_t, which is a signed 64-bit type. NFSv3 and NFSv4 both define file size as an unsigned 64-bit type. Thus there is a range of valid file size values an NFS client can send that is already larger than Linux can handle. Currently decode_fattr4() dumps a full u64 value into ia_size. If that value happens to be larger than S64_MAX, then ia_size underflows. I'm about to fix up the NFSv3 behavior as well, so let's catch the underflow in the common code path: nfsd_setattr().

CVE ID	Description
CVE-2022-48829	In the Linux kernel, the following vulnerability has been resolved: NFSD: Fix NFSv3 SETATTR/CREATE's handling of large file sizes iattr::ia_size is a loff_t, so these NFSv3 procedures must be careful to deal with incoming client size values that are larger than s64_max without corrupting the value. Silently capping the value results in storing a different value than the client passed in which is unexpected behavior, so remove the min_t() check in decode_sattr3(). Note that RFC 1813 permits only the WRITE procedure to return NFS3ERR_FBIG. We believe that NFSv3 reference implementations also return NFS3ERR_FBIG when ia_size is too large.
CVE-2022-48834	In the Linux kernel, the following vulnerability has been resolved: usb: usbtmc: Fix bug in pipe direction for control transfers The syzbot fuzzer reported a minor bug in the usbtmc driver: usb 5-1: BOGUS control dir, pipe 80001e80 doesn't match bRequestType 0 WARNING: CPU: 0 PID: 3813 at drivers/usb/core/urb.c:412 usb_submit_urb+0x13a5/0x1970 drivers/usb/core/urb.c:410 Modules linked in: CPU: 0 PID: 3813 Comm: syz-executor122 Not tainted 5.17.0-rc5-syzkaller-00306-g2293be58d6a1 #0 ... Call Trace: <TASK> usb_start_wait_urb+0x113/0x530 drivers/usb/core/message.c:58 usb_internal_control_msg drivers/usb/core/message.c:102 [inline] usb_control_msg+0x2a5/0x4b0 drivers/usb/core/message.c:153 usbtmc_ioctl_request drivers/usb/class/usbtmc.c:1947 [inline] The problem is that usbtmc_ioctl_request() uses usb_rcvctrlpipe() for all of its transfers, whether they are in or out. It's easy to fix.
CVE-2022-48835	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Page fault in reply q processing A page fault was encountered in mpt3sas on a LUN reset error path: [145.763216] mpt3sas_cm1: Task abort tm failed: handle(0x0002),timeout(30) tr_method(0x0) smid(3) msix_index(0) [145.778932] scsi 1:0:0:0: task abort: FAILED scmd(0x0000000024ba29a2) [145.817307] scsi 1:0:0:0: attempting device reset! scmd(0x0000000024ba29a2) [145.827253] scsi 1:0:0:0: [sg1] tag#2 CDB: Receive Diagnostic 1c 01 01 ff fc 00 [145.837617] scsi target1:0:0: handle(0x0002), sas_address(0x500605b0000272b9), phy(0) [145.848598] scsi target1:0:0: enclosure logical id(0x500605b0000272b8), slot(0) [149.858378] mpt3sas_cm1: Poll ReplyDescriptor queues for completion of smid(0), task_type(0x05), handle(0x0002) [149.875202] BUG: unable to handle page fault for address: 00000007fffc445d [149.885617] #PF: supervisor read access in kernel mode [149.894346] #PF: error_code(0x0000) - not-present page [149.903123] PGD 0 P4D 0 [149.909387] Oops: 0000 [#1] PREEMPT SMP NOPTI [149.917417] CPU: 24 PID: 3512 Comm: scsi_ah_1 Kdump: loaded Tainted: G S O 5.10.89-altav-1 #1 [149.934327] Hardware name: DDN 200NVX2 /200NVX2-MB , BIOS ATHG2.2.02.01 09/10/2021 [149.951871] RIP: 0010:_base_process_reply_queue+0x4b/0x900 [mpt3sas] [149.961889] Code: 0f 84 22 02 00 00 8d 48 01 49 89 fd 48 8d 57 38 f0 0f b1 4f 38 0f 85 d8 01 00 00 49 8b 45 10 45 31 e4 41 8b 55 0c 48 8d 1c d0 <0f> b6 03 83 e0 0f 3c 0f 0f 85 a2 00 00 00 e9 e6 01 00 00 0f b7 ee [149.991952] RSP: 0018:ffffc9000f1ebcb8 EFLAGS: 00010246 [150.000937] RAX: 0000000000000055 RBX: 00000007fffc445d RCX: 0000000002548f071 [150.011841] RDX: 00000000ffff8881 RSI: 0000000000000001 RDI: ffff888125ed50d8 [150.022670] RBP: 0000000000000000 R08: 0000000000000000 R09: c0000000ffff7fff [150.033445] R10: ffff9000f1ebb68 R11: ffff9000f1ebb60 R12: 0000000000000000 [150.044204] R13: ffff888125ed50d8 R14: 0000000000000080 R15: 34cdc00034cdea80 [150.054963] FS: 0000000000000000(0000) GS:ffff88dfaf200000(0000) knlGS:0000000000000000 [150.066715] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [150.076078] CR2: 00000007fffc445d CR3: 000000012448a006 CR4: 0000000000770ee0 [150.086887] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [150.097670] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [150.108323] PKRU: 55555554 [150.114690] Call Trace: [150.120497] ? printk+0x48/0x4a [150.127049] mpt3sas_scsih_issue_tm.cold.114+0x2e/0x2b3 [mpt3sas] [150.136453] mpt3sas_scsih_issue_locked_tm+0x86/0xb0 [mpt3sas] [150.145759] scsih_dev_reset+0xea/0x300 [mpt3sas] [150.153891] scsi_ah_ready_devs+0x541/0x9e0 [scsi_mod] [150.162206] ? __scsi_host_match+0x20/0x20 [scsi_mod] [150.170406] ? scsi_try_target_reset+0x90/0x90 [scsi_mod] [150.178925] ? blk_mq_tagset_busy_iter+0x45/0x60 [150.186638] ? scsi_try_target_reset+0x90/0x90 [scsi_mod] [150.195087] scsi_error_handler+0x3a5/0x4a0 [scsi_mod] [150.203206] ? __schedule+0x1e9/0x610 [150.209783] ? scsi_ah_get_sense+0x210/0x210 [scsi_mod] [150.217924] kthread+0x12e/0x150 [150.224041] ? kthread_worker_fn+0x130/0x130 [150.231206] ret_from_fork+0x1f/0x30 This is caused by mpt3sas_base_sync_reply_irqs() using an invalid reply_q pointer outside of the list_for_each_entry() loop. At the end of the full list traversal the pointer is invalid. Move the _base_process_reply_queue() call inside of the loop.
CVE-2022-48837	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: rndis: prevent integer overflow in rndis_set_response() If "BufOffset" is very large the "BufOffset + 8" operation can have an integer overflow.

CVE ID	Description
CVE-2022-48839	In the Linux kernel, the following vulnerability has been resolved: net/packet: fix slab-out-of-bounds access in packet_recvmg() syzbot found that when an AF_PACKET socket is using PACKET_COPY_THRESH and mmap operations, tpacket_rcv() is queueing skbs with garbage in skb->cb[], triggering a too big copy [1] Presumably, users of af_packet using mmap() already gets correct metadata from the mapped buffer, we can simply make sure to clear 12 bytes that might be copied to user space later. BUG: KASAN: stack-out-of-bounds in memcpy include/linux/fortify-string.h:225 [inline] BUG: KASAN: stack-out-of-bounds in packet_recvmg+0x56c/0x1150 net/packet/af_packet.c:3489 Write of size 165 at addr ffffc9000385fb78 by task syz-executor233/3631 CPU: 0 PID: 3631 Comm: syz-executor233 Not tainted 5.17.0-rc7-syzkaller-02396-g0b3660695e80 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xcd/0x134 lib/dump_stack.c:106 print_address_description.constprop.0.cold+0xf/0x336 mm/kasan/report.c:255 __kasan_report mm/kasan/report.c:442 [inline] kasan_report.cold+0x83/0xdf mm/kasan/report.c:459 check_region_inline mm/kasan/generic.c:183 [inline] kasan_check_range+0x13d/0x180 mm/kasan/generic.c:189 memcpy+0x39/0x60 mm/kasan/shadow.c:66 memcpy include/linux/fortify-string.h:225 [inline] packet_recvmg+0x56c/0x1150 net/packet/af_packet.c:3489 sock_recvmg_nosec net/socket.c:948 [inline] sock_recvmg net/socket.c:966 [inline] sock_recvmg net/socket.c:962 [inline] ____sys_recvmg+0x2c4/0x600 net/socket.c:2632 __sys_recvmg+0x127/0x200 net/socket.c:2674 __sys_recvmg+0xe2/0x1a0 net/socket.c:2704 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7dfd5954c29 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 41 15 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 c0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007ffcf8e71e48 EFLAGS: 00000246 ORIG_RAX: 000000000000002f RAX: ffffffffdfda RBX: 0000000000000003 RCX: 00007dfd5954c29 RDX: 0000000000000000 RSI: 0000000020000500 RDI: 0000000000000005 RBP: 0000000000000000 R08: 000000000000000d R09: 000000000000000d R10: 0000000000000000 R11: 0000000000000246 R12: 00007ffcf8e71e60 R13: 0000000000f4240 R14: 000000000000c1ff R15: 00007ffcf8e71e54 </TASK> addr ffffc9000385fb78 is located in stack of task syz-executor233/3631 at offset 32 in frame: ____sys_recvmg+0x0/0x600 include/linux/uio.h:246 this frame has 1 object: [32, 160) 'addr' Memory state around the buggy address: ffffc9000385fa80: 00 04 f3 f3 f3 f3 00 00 00 00 00 00 00 00 ffffc9000385fb00: 00 00 00 00 00 00 00 00 00 00 00 00 f1 f1 f1 00 >fffc9000385fb80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 f3 ^ ffffc9000385fc00: f3 f3 f3 00 00 00 00 00 00 00 00 00 00 f1 ffffc9000385fc80: f1 f1 f1 00 f2 f2 00 f2 f2 00 00 00 00 00 =====
CVE-2022-48841	In the Linux kernel, the following vulnerability has been resolved: ice: fix NULL pointer dereference in ice_update_vsi_tx_ring_stats() It is possible to do NULL pointer dereference in routine that updates Tx ring stats. Currently only stats and bytes are updated when ring pointer is valid, but later on ring is accessed to propagate gathered Tx stats onto VSI stats. Change the existing logic to move to next ring when ring is NULL.

CVE ID	Description
CVE-2022-48842	In the Linux kernel, the following vulnerability has been resolved: ice: Fix race condition during interface enslave Commit 5dbbbd01cbba83 ("ice: Avoid RTNL lock when re-creating auxiliary device") changes a process of re-creation of aux device so ice_plug_aux_dev() is called from ice_service_task() context. This unfortunately opens a race window that can result in dead-lock when interface has left LAG and immediately enters LAG again. Reproducer: `` #!/bin/sh ip link add lag0 type bond mode 1 miimon 100 ip link set lag0 for n in {1..10}; do echo Cycle: \$n ip link set ens7f0 master lag0 sleep 1 ip link set ens7f0 nomaster done `` This results in: [20976.208697] Workqueue: ice ice_service_task [ice] [20976.213422] Call Trace: [20976.215871] __schedule+0x2d1/0x830 [20976.219364] schedule+0x35/0xa0 [20976.222510] schedule_preempt_disabled+0xa/0x10 [20976.227043] __mutex_lock.isra.7+0x310/0x420 [20976.235071] enum_all_gids_of_dev_cb +0x1c/0x100 [ib_core] [20976.251215] ib_enum_roce_netdev+0xa4/0xe0 [ib_core] [20976.256192] ib_cache_setup_one+0x33/0xa0 [ib_core] [20976.261079] ib_register_device+0x40d/0x580 [ib_core] [20976.266139] irdma_ib_register_device+0x129/0x250 [irdma] [20976.281409] irdma_probe +0x2c1/0x360 [irdma] [20976.285691] auxiliary_bus_probe+0x45/0x70 [20976.289790] really_probe +0x1f2/0x480 [20976.298509] driver_probe_device+0x49/0xc0 [20976.302609] bus_for_each_drv +0x79/0xc0 [20976.306448] __device_attach+0xdc/0x160 [20976.310286] bus_probe_device +0x9d/0xb0 [20976.314128] device_add+0x43c/0x890 [20976.321287] __auxiliary_device_add +0x43/0x60 [20976.325644] ice_plug_aux_dev+0xb2/0x100 [ice] [20976.330109] ice_service_task +0xd0c/0xed0 [ice] [20976.342591] process_one_work+0x1a7/0x360 [20976.350536] worker_thread +0x30/0x390 [20976.358128] kthread+0x10a/0x120 [20976.365547] ret_from_fork+0x1f/0x40 ... [20976.438030] task:ip state:D stack: 0 pid:213658 ppid:213627 flags:0x00004084 [20976.446469] Call Trace: [20976.448921] __schedule+0x2d1/0x830 [20976.452414] schedule+0x35/0xa0 [20976.455559] schedule_preempt_disabled+0xa/0x10 [20976.460090] __mutex_lock.isra.7+0x310/0x420 [20976.464364] device_del+0x36/0x3c0 [20976.467772] ice_unplug_aux_dev+0x1a/0x40 [ice] [20976.472313] ice_lag_event_handler+0x2a2/0x520 [ice] [20976.477288] notifier_call_chain +0x47/0x70 [20976.481386] __netdev_upper_dev_link+0x18b/0x280 [20976.489845] bond_enslave +0xe05/0x1790 [bonding] [20976.494475] do_setlink+0x336/0xf50 [20976.502517] __rtnl_newlink +0x529/0x8b0 [20976.543441] rtnl_newlink+0x43/0x60 [20976.546934] rtnetlink_rcv_msg +0x2b1/0x360 [20976.559238] netlink_rcv_skb+0x4c/0x120 [20976.563079] netlink_unicast +0x196/0x230 [20976.567005] netlink_sendmsg+0x204/0x3d0 [20976.570930] sock_sendmsg +0x4c/0x50 [20976.574423] __sys_sendmsg+0x1eb/0x250 [20976.586807] __sys_sendmsg +0x7c/0xc0 [20976.606353] __sys_sendmsg+0x57/0xa0 [20976.609930] do_syscall_64+0x5b/0x1a0 [20976.613598] entry_SYSCALL_64_after_hwframe+0x65/0xca 1. Command 'ip link ... set nomaster' causes that ice_plug_aux_dev() is called from ice_service_task() context, aux device is created and associated device->lock is taken. 2. Command 'ip link ... set master...' calls ice's notifier under RTNL lock and that notifier calls ice_unplug_aux_dev(). That function tries to take aux device->lock but this is already taken by ice_plug_aux_dev() in step 1 3. Later ice_plug_aux_dev() tries to take RTNL lock but this is already taken in step 2 4. Dead-lock The patch fixes this issue by following changes: - Bit ICE_FLAG_PLUG_AUX_DEV is kept to be set during ice_plug_aux_dev() call in ice_service_task() - The bit is checked in ice_clear_rdma_cap() and only if it is not set then ice_unplug_aux_dev() is called. If it is set (in other words plugging of aux device was requested and ice_plug_aux_dev() is potentially running) then the function only clears the ---truncated---
CVE-2022-48843	In the Linux kernel, the following vulnerability has been resolved: drm/vrr: Set VRR capable prop only if it is attached to connector VRR capable property is not attached by default to the connector It is attached only if VRR is supported. So if the driver tries to call drm core set prop function without it being attached that causes NULL dereference.
CVE-2022-48844	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Fix leaking sent_cmd skb sent_cmd memory is not freed before freeing hci_dev causing it to leak its contents.
CVE-2022-48846	In the Linux kernel, the following vulnerability has been resolved: block: release rq qos structures for queue without disk blkcg_init_queue() may add rq qos structures to request queue, previously blk_cleanup_queue() calls rq_qos_exit() to release them, but commit 8e141f9eb803 ("block: drain file system I/O on del_gendisk") moves rq_qos_exit() into del_gendisk(), so memory leak is caused because queues may not have disk, such as un-present scsi luns, nvme admin queue, ... Fixes the issue by adding rq_qos_exit() to blk_cleanup_queue() back. BTW, v5.18 won't need this patch any more since we move blkcg_init_queue()/blkcg_exit_queue() into disk allocation/release handler, and patches have been in for-5.18/block.

CVE ID	Description
CVE-2022-48848	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Do not unregister events twice Nicolas reported that using: # trace-cmd record -e all -M 10 -p osnoise --poll Resulted in the following kernel warning: -----[cut here]----- WARNING: CPU: 0 PID: 1217 at kernel/tracepoint.c:404 tracepoint_probe_unregister+0x280/0x370 [...] CPU: 0 PID: 1217 Comm: trace-cmd Not tainted 5.17.0-rc6-next-20220307-nico+ #19 RIP: 0010:tracepoint_probe_unregister+0x280/0x370 [...] CR2: 00007ff919b29497 CR3: 0000000109da4005 CR4: 000000000170ef0 Call Trace: <TASK> osnoise_workload_stop+0x36/0x90 tracing_set_tracer+0x108/0x260 tracing_set_trace_write+0x94/0xd0 ? __check_object_size.part.0+0x10a/0x150 ? selinux_file_permission+0x104/0x150 vfs_write+0xb5/0x290 ksys_write+0x5f/0xe0 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7ff919a18127 [...] ---[end trace 0000000000000000]--- The warning complains about an attempt to unregister an unregistered tracepoint. This happens on trace-cmd because it first stops tracing, and then switches the tracer to nop. Which is equivalent to: # cd /sys/kernel/tracing/ # echo osnoise > current_tracer # echo 0 > tracing_on # echo nop > current_tracer The osnoise tracer stops the workload when no trace instance is actually collecting data. This can be caused both by disabling tracing or disabling the tracer itself. To avoid unregistering events twice, use the existing trace_osnoise_callback_enabled variable to check if the events (and the workload) are actually active before trying to deactivate them.
CVE-2022-48849	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: bypass tiling flag check in virtual display case (v2) vkms leverages common amdgpu framebuffer creation, and also as it does not support FB modifier, there is no need to check tiling flags when initing framebuffer when virtual display is enabled. This can fix below calldata: amdgpu 0000:00:08.0: GFX9+ requires FB check based on format modifier WARNING: CPU: 0 PID: 1023 at drivers/gpu/drm/amd/amdgpu/amdgpu_display.c:1150 amdgpu_display_framebuffer_init+0x8e7/0xb40 [amdgpu] v2: check adev->enable_virtual_display instead as vkms can be enabled in bare metal as well.
CVE-2022-48850	In the Linux kernel, the following vulnerability has been resolved: net-sysfs: add check for netdevice being present to speed_show When bringing down the netdevice or system shutdown, a panic can be triggered while accessing the sysfs path because the device is already removed. [755.549084] mlx5_core 0000:12:00.1: Shutdown was called [756.404455] mlx5_core 0000:12:00.0: Shutdown was called ... [757.937260] BUG: unable to handle kernel NULL pointer dereference at (null) [758.031397] IP: [<ffffffffff8ee11acb>] "amsd"="" #10="" #11="" #12="" #13="" #14="" #15="" #16="" #17="" #18="" #19="" #20="" #21="" #22="" #23="" #24="" #25="" #26="" #9="" (__link_state_start)="" ...="" 0000000000000000="" 0000000000000246="" 00000000000008d0="" 0000000000001000="" 0000000000001f080="" 00010046="" 0010="" 0018="" 0x280="" 1="" 12649="" [exception="" [ffff89240e1a38b0]="" [ffff89240e1a39c8]="" [ffff89240e1a3a18]="" [ffff89240e1a3ab8]="" [ffff89240e1a3ae8]="" [ffff89240e1a3b40]="" [ffff89240e1a3bb0]="" [ffff89240e1a3c08]="" [ffff89240e1a3ce8]="" [ffff89240e1a3d48]="" [ffff89240e1a3dd8]="" [ffff89240e1a3df8]="" [ffff89240e1a3e18]="" [ffff89240e1a3e28]="" [ffff89240e1a3e98]="" [ffff89240e1a3ed8]="" [ffff89240e1a3f08]="" [ffff89240e1a3f50]="" [mlx5_core]="" __ethtool_get_link_ksettings="" __link_state_nocarrier="" also="" at="" bt="" cmd_exec="" command:="" cpu:="" crash>="" cs:="" dev_attr_show="" dma_pool_alloc+0x1ab="" dma_pool_alloc+0x1ab]="" ffff8905ffc03c00="" ffff89240e1a3968="" ffff89240e1a39c0="" ffff8924108f2100="" ffff89243d874080="" ffff89243d874090="" ffff89243d874100="" ffff89443b0c0000="" ffffffff04680d4="" ffffffff04680f3="" ffffffff046ad62="" ffffffff046b4fb="" ffffffff0475434="" ffffffff04992bf="" ffffffff049ab36="" ffffffff04a7348="" ffffffff8edde9fd="" ffffffff8ee11acb="" ffffffff8ee4e3ff="" ffffffff8ee4f27f="" ffffffff8ee76d10="" ffffffff8eeda596="" ffffffff8eedaef5="" ffffffff8eedbedf="" ffffffff8f0b70e3="" ffffffff8f25db46="" ffffffff8f277208="" ffffffff8f38c778="" ffffffff8f395f92="" get_fec_supported_advertised="" is="" kernfs_fop_read="" kernfs_seq_show="" make="" mlx5_alloc_cmd_msg="" mlx5_cmd_exec="" mlx5_core_access_reg="" mlx5e_get_fec_caps="" mlx5e_get_link_ksettings="" net_device.state="" netdevice="" orig_rax:="" page_fault="" pid:="" present.<="" prevent="" r10:="" r11:="" r12:="" r13:="" r14:="" r15:="" r8:="" r9:="" rax:="" rbp:="" rbx:="" rcx:="" rdi:="" rdx:="" rflags:="" rip:="" rsi:="" rsp:="" scenario,="" seq_read="" speed_show="" ss:="" state="0x5" sure="" sys_read="" sysfs_kf_seq_show="" system_call_fastpath="" task:="" td="" that="" the="" this="" to="" vfs_read="" we=""></ffffffffff8ee11acb>]>

CVE ID	Description
CVE-2022-48853	In the Linux kernel, the following vulnerability has been resolved: swiotlb: fix info leak with DMA_FROM_DEVICE The problem I'm addressing was discovered by the LTP test covering cve-2018-1000204. A short description of what happens follows: 1) The test case issues a command code 00 (TEST UNIT READY) via the SG_IO interface with: dxfer_len == 524288, dxfer_dir == SG_DXFER_FROM_DEV and a corresponding dxferp. The peculiar thing about this is that TUR is not reading from the device. 2) In sg_start_req() the invocation of blk_rq_map_user() effectively bounces the user-space buffer. As if the device was to transfer into it. Since commit a45b599ad808 ("scsi: sg: allocate with __GFP_ZERO in sg_build_indirect()") we make sure this first bounce buffer is allocated with GFP_ZERO. 3) For the rest of the story we keep ignoring that we have a TUR, so the device won't touch the buffer we prepare as if the we had a DMA_FROM_DEVICE type of situation. My setup uses a virtio-scsi device and the buffer allocated by SG is mapped by the function virtqueue_add_split() which uses DMA_FROM_DEVICE for the "in" sgs (here scatter-gather and not scsi generics). This mapping involves bouncing via the swiotlb (we need swiotlb to do virtio in protected guest like s390 Secure Execution, or AMD SEV). 4) When the SCSI TUR is done, we first copy back the content of the second (that is swiotlb) bounce buffer (which most likely contains some previous IO data), to the first bounce buffer, which contains all zeros. Then we copy back the content of the first bounce buffer to the user-space buffer. 5) The test case detects that the buffer, which it zero-initialized, ain't all zeros and fails. One can argue that this is an swiotlb problem, because without swiotlb we leak all zeros, and the swiotlb should be transparent in a sense that it does not affect the outcome (if all other participants are well behaved). Copying the content of the original buffer into the swiotlb buffer is the only way I can think of to make swiotlb transparent in such scenarios. So let's do just that if in doubt, but allow the driver to tell us that the whole mapped buffer is going to be overwritten, in which case we can preserve the old behavior and avoid the performance impact of the extra bounce.
CVE-2022-48858	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix a race on command flush flow Fix a refcount use after free warning due to a race on command entry. Such race occurs when one of the commands releases its last refcount and frees its index and entry while another process running command flush flow takes refcount to this command entry. The process which handles commands flush may see this command as needed to be flushed if the other process released its refcount but didn't release the index yet. Fix it by adding the needed spin lock. It fixes the following warning trace: refcount_t: addition on 0; use-after-free. WARNING: CPU: 11 PID: 540311 at lib/refcount.c:25 refcount_warn_saturate+0x80/0xe0 ... RIP: 0010:refcount_warn_saturate+0x80/0xe0 ... Call Trace: <TASK> mlx5_cmd_trigger_completions+0x293/0x340 [mlx5_core] mlx5_cmd_flush+0x3a/0xf0 [mlx5_core] enter_error_state+0x44/0x80 [mlx5_core] mlx5_fw_fatal_reporter_err_work+0x37/0xe0 [mlx5_core] process_one_work+0x1be/0x390 worker_thread+0x4d/0x3d0 ? rescuer_thread+0x350/0x350 kthread+0x141/0x160 ? set_kthread_struct+0x40/0x40 ret_from_fork+0x1f/0x30 </TASK>
CVE-2022-48861	In the Linux kernel, the following vulnerability has been resolved: vdpa: fix use-after-free on vp_vdpa_remove When vp_vdpa driver is unbind, vp_vdpa is freed in vdpa_unregister_device and then vp_vdpa->mdev.pci_dev is dereferenced in vp_modern_remove, triggering use-after-free. Call Trace of unbinding driver free vp_vdpa : do_syscall_64 vfs_write kernfs_fop_write_iter kobject_release_driver_internal pci_device_remove vp_vdpa_remove vdpa_unregister_device kobject_release_device_release kfree Call Trace of dereference vp_vdpa->mdev.pci_dev: vp_modern_remove pci_release_selected_regions pci_release_region pci_resource_len pci_resource_end (dev)->resource[(bar)].end
CVE-2022-48862	In the Linux kernel, the following vulnerability has been resolved: vhost: fix hung thread due to erroneous iotlb entries In vhost_iotlb_add_range_ctx(), range size can overflow to 0 when start is 0 and last is ULONG_MAX. One instance where it can happen is when userspace sends an IOTLB message with iova=size=uaddr=0 (vhost_process_iotlb_msg). So, an entry with size = 0, start = 0, last = ULONG_MAX ends up in the iotlb. Next time a packet is sent, iotlb_access_ok() loops indefinitely due to that erroneous entry. Call Trace: <TASK> iotlb_access_ok+0x21b/0x3e0 drivers/vhost/vhost.c:1340 vq_meta_prefetch+0xbc/0x280 drivers/vhost/vhost.c:1366 vhost_transport_do_send_pkt+0xe0/0xf0 drivers/vhost/vsock.c:104 vhost_worker+0x23d/0x3d0 drivers/vhost/vhost.c:372 kthread+0x2e9/0x3a0 kernel/kthread.c:377 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:295 </TASK> Reported by syzbot at: https://syzkaller.appspot.com/bug?extid=0abd373e2e50d704db87 To fix this, do two things: 1. Return -EINVAL in vhost_chr_write_iter() when userspace asks to map a range with size 0. 2. Fix vhost_iotlb_add_range_ctx() to handle the range [0, ULONG_MAX] by splitting it into two entries.
CVE-2022-48863	In the Linux kernel, the following vulnerability has been resolved: mISDN: Fix memory leak in dsp_pipeline_build() dsp_pipeline_build() allocates dup pointer by kstrdup(cfg), but then it updates dup variable by strsep(&dup, " "). As a result when it calls kfree(dup), the dup variable contains NULL. Found by Linux Driver Verification project (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2022-48864	In the Linux kernel, the following vulnerability has been resolved: vdpa/mlx5: add validation for VIRTIO_NET_CTRL_MQ_VQ_PAIRS_SET command When control vq receives a VIRTIO_NET_CTRL_MQ_VQ_PAIRS_SET command request from the driver, presently there is no validation against the number of queue pairs to configure, or even if multiqueue had been negotiated or not is unverified. This may lead to kernel panic due to uninitialized resource for the queues were there any bogus request sent down by untrusted driver. Tie up the loose ends there.
CVE-2022-48865	In the Linux kernel, the following vulnerability has been resolved: tipc: fix kernel panic when enabling bearer When enabling a bearer on a node, a kernel panic is observed: [4.498085] RIP: 0010:tipc_mon_prep+0x4e/0x130 [tipc] ... [4.520030] Call Trace: [4.520689] <IRQ> [4.521236] tipc_link_build_proto_msg+0x375/0x750 [tipc] [4.522654] tipc_link_build_state_msg+0x48/0xc0 [tipc] [4.524034] __tipc_node_link_up+0xd7/0x290 [tipc] [4.525292] tipc_rcv+0x5da/0x730 [tipc] [4.526346] ? __netif_receive_skb_core+0xb7/0xfc0 [4.527601] tipc_l2_rcv_msg+0x5e/0x90 [tipc] [4.528737] __netif_receive_skb_list_core+0x20b/0x260 [4.530068] netif_receive_skb_list_internal+0x1bf/0x2e0 [4.531450] ? dev_gro_receive+0x4c2/0x680 [4.532512] napi_complete_done+0x6f/0x180 [4.533570] virtnet_poll+0x29c/0x42e [virtio_net] ... The node in question is receiving activate messages in another thread after changing bearer status to allow message sending/ receiving in current thread: thread 1 thread 2 ----- ----- tipc_enable_bearer() test_and_set_bit_lock() tipc_bearer_xmit_skb() tipc_l2_rcv_msg() tipc_rcv() __tipc_node_link_up() tipc_link_build_state_msg() tipc_link_build_proto_msg() tipc_mon_prep() { ... // null-pointer dereference u16 gen = mon->dom_gen; ... } // Not being executed yet tipc_mon_create() { ... // allocate mon = kzalloc(); ... } Monitoring pointer in thread 2 is dereferenced before monitoring data is allocated in thread 1. This causes kernel panic. This commit fixes it by allocating the monitoring data before enabling the bearer to receive messages.
CVE-2022-48868	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Let probe fail when workqueue cannot be enabled The workqueue is enabled when the appropriate driver is loaded and disabled when the driver is removed. When the driver is removed it assumes that the workqueue was enabled successfully and proceeds to free allocations made during workqueue enabling. Failure during workqueue enabling does not prevent the driver from being loaded. This is because the error path within drv_enable_wq() returns success unless a second failure is encountered during the error path. By returning success it is possible to load the driver even if the workqueue cannot be enabled and allocations that do not exist are attempted to be freed during driver remove. Some examples of problematic flows: (a) idxd_dmaengine_drv_probe() -> drv_enable_wq() -> idxd_wq_request_irq(): In above flow, if idxd_wq_request_irq() fails then idxd_wq_unmap_portal() is called on error exit path, but drv_enable_wq() returns 0 because idxd_wq_disable() succeeds. The driver is thus loaded successfully. idxd_dmaengine_drv_remove()->drv_disable_wq()->idxd_wq_unmap_portal() Above flow on driver unload triggers the WARN in devm_iounmap() because the device resource has already been removed during error path of drv_enable_wq(). (b) idxd_dmaengine_drv_probe() -> drv_enable_wq() -> idxd_wq_request_irq(): In above flow, if idxd_wq_request_irq() fails then idxd_wq_init_percpu_ref() is never called to initialize the percpu counter, yet the driver loads successfully because drv_enable_wq() returns 0. idxd_dmaengine_drv_remove()->__idxd_wq_quiesce()->percpu_ref_kill(): Above flow on driver unload triggers a BUG when attempting to drop the initial ref of the uninitialized percpu ref: BUG: kernel NULL pointer dereference, address: 0000000000000010 Fix the drv_enable_wq() error path by returning the original error that indicates failure of workqueue enabling. This ensures that the probe fails when an error is encountered and the driver remove paths are only attempted when the workqueue was enabled successfully.

CVE ID	Description
CVE-2022-48875	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: sdata can be NULL during AMPDU start ieee80211_tx_ba_session_handle_start() may get NULL for sdata when a deauthentication is ongoing. Here a trace triggering the race with the hostapd test multi_ap_fronthaul_on_ap: (gdb) list *drv_ampdu_action+0x46 0x8b16 is in drv_ampdu_action (net/mac80211/driver-ops.c:396). 391 int ret = -EOPNOTSUPP; 392 393 might_sleep(); 394 395 sdata = get_bss_sdata(sdata); 396 if (!check_sdata_in_driver(sdata)) 397 return -EIO; 398 399 trace_drv_ampdu_action(local, sdata, params); 400 wlan0: moving STA 02:00:00:00:03:00 to state 3 wlan0: associated wlan0: deauthenticating from 02:00:00:00:03:00 by local choice (Reason: 3=DEAUTH_LEAVING) wlan3.sta1: Open BA session requested for 02:00:00:00:00:00 tid 0 wlan3.sta1: dropped frame to 02:00:00:00:00:00 (unauthorized port) wlan0: moving STA 02:00:00:00:03:00 to state 2 wlan0: moving STA 02:00:00:00:03:00 to state 1 wlan0: Removed STA 02:00:00:00:03:00 wlan0: Destroyed STA 02:00:00:00:03:00 BUG: unable to handle page fault for address: ffffffff48 PGD 11814067 P4D 11814067 PUD 11816067 PMD 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 2 PID: 133397 Comm: kworker/u16:1 Tainted: G W 6.1.0-rc8-wt+ #59 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.0-20220807_005459-localhost 04/01/2014 Workqueue: phy3 ieee80211_ba_session_work [mac80211] RIP: 0010:drv_ampdu_action+0x46/0x280 [mac80211] Code: 53 48 89 f3 be 89 01 00 00 e8 d6 43 bf ef e8 21 46 81 f0 83 bb a0 1b 00 00 04 75 0e 48 8b 9b 28 0d 00 00 48 81 eb 10 0e 00 00 <8b> 93 58 09 00 00 f6 c2 20 0f 84 3b 01 00 00 8b 05 dd 1c 0f 00 85 RSP: 0018:ffff900025ebd20 EFLAGS: 00010287 RAX: 0000000000000000 RBX: ffffffff1f0 RCX: ffff888102228240 RDX: 0000000080000000 RSI: ffffffff918c5de0 RDI: ffff888102228b40 RBP: ffff900025ebd40 R08: 0000000000000001 R09: 0000000000000001 R10: 0000000000000001 R11: 0000000000000000 R12: ffff888118c18ec0 R13: 0000000000000000 R14: ffff900025ebd60 R15: ffff888018b7efb8 FS: 0000000000000000(0000) GS:ffff88817a600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffff48 CR3: 0000000105228006 CR4: 000000000170ee0 Call Trace: <TASK> ieee80211_tx_ba_session_handle_start+0xd0/0x190 [mac80211] ieee80211_ba_session_work+0xff/0x2e0 [mac80211] process_one_work+0x29f/0x620 worker_thread+0x4d/0x3d0 ? process_one_work+0x620/0x620 kthread+0xfb/0x120 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-48876	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix initialization of rx->link and rx->link_sta There are some codepaths that do not initialize rx->link_sta properly. This causes a crash in places which assume that rx->link_sta is valid if rx->sta is valid. One known instance is triggered by __ieee80211_rx_h_amsdu being called from fast-rx. It results in a crash like this one: BUG: kernel NULL pointer dereference, address: 00000000000000a8 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 0 P4D 0 Oops: 0002 [#1] PREEMPT SMP PTI CPU: 1 PID: 506 Comm: mt76-usb-rx phy Tainted: G E 6.1.0-debian64x+1.7 #3 Hardware name: ZOTAC ZBOX-ID92/ZBOX-IQ01/ZBOX-ID92/ZBOX-IQ01, BIOS B220P007 05/21/2014 RIP: 0010:ieee80211_deliver_skb+0x62/0x1f0 [mac80211] Code: 00 48 89 04 24 e8 9e a7 c3 df 89 c0 48 03 1c c5 a0 ea 39 a1 4c 01 6b 08 48 ff 03 48 83 7d 28 00 74 11 48 8b 45 30 48 63 55 44 <48> 83 84 d0 a8 00 00 00 01 41 8b 86 c0 11 00 00 8d 50 fd 83 fa 01 RSP: 0018:ffff999040803b10 EFLAGS: 00010286 RAX: 0000000000000000 RBX: fffff9903f496480 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ffff999040803ce0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: ffff8d21828ac900 R13: 0000000000000004a R14: ffff8d2198ed89c0 R15: ffff8d2198ed8000 FS: 0000000000000000(0000) GS:ffff8d24afe80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000000000a8 CR3: 0000000429810002 CR4: 0000000001706e0 Call Trace: <TASK> __ieee80211_rx_h_amsdu+0x1b5/0x240 [mac80211] ? ieee80211_prepare_and_rx_handle+0xcdd/0x1320 [mac80211] ? __local_bh_enable_ip+0x3b/0xa0 ieee80211_prepare_and_rx_handle+0xcdd/0x1320 [mac80211] ? prepare_transfer+0x109/0x1a0 [xhci_hcd] ieee80211_rx_list+0xa80/0xda0 [mac80211] mt76_rx_complete+0x207/0x2e0 [mt76] mt76_rx_poll_complete+0x357/0x5a0 [mt76] mt76u_rx_worker+0x4f5/0x600 [mt76_usb] ? mt76_get_min_avg_rssi+0x140/0x140 [mt76] __mt76_worker_fn+0x50/0x80 [mt76] kthread+0xed/0x120 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30 Since the initialization of rx->link and rx->link_sta is rather convoluted and duplicated in many places, clean it up by using a helper function to set it. [remove unnecessary rx->sta->sta.mlo check]
CVE-2022-48889	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: sof-nau8825: fix module alias overflow The maximum name length for a platform_device_id entry is 20 characters including the trailing NUL byte. The sof_nau8825.c file exceeds that, which causes an obscure error message: sound/soc/intel/boards/snd-soc-sof_nau8825.mod.c:35:45: error: illegal character encoding in string literal [-Werror,-Winvalid-source-encoding] MODULE_ALIAS("platform:adl_max98373_nau8825<U+0018><AA>"); ~~~~~ include/linux/module.h:168:49: note: expanded from macro 'MODULE_ALIAS' ~~~~~ include/linux/module.h:165:56: note: expanded from macro 'MODULE_INFO' ~~~~~ include/linux/moduleparam.h:26:47: note: expanded from macro 'MODULE_INFO' = __MODULE_INFO_PREFIX __stringify(tag) "=" info I could not figure out how to make the module handling robust enough to handle this better, but as a quick fix, using slightly shorter names that are still unique avoids the build issue.

CVE ID	Description
CVE-2022-48890	In the Linux kernel, the following vulnerability has been resolved: scsi: storvsc: Fix swiotlb bounce buffer leak in confidential VM storvsc_queuecommand() maps the scatter/gather list using scsi_dma_map(), which in a confidential VM allocates swiotlb bounce buffers. If the I/O submission fails in storvsc_do_io(), the I/O is typically retried by higher level code, but the bounce buffer memory is never freed. The mostly like cause of I/O submission failure is a full VMBus channel ring buffer, which is not uncommon under high I/O loads. Eventually enough bounce buffer memory leaks that the confidential VM can't do any I/O. The same problem can arise in a non-confidential VM with kernel boot parameter swiotlb=force. Fix this by doing scsi_dma_unmap() in the case of an I/O submission error, which frees the bounce buffer memory.
CVE-2022-48896	In the Linux kernel, the following vulnerability has been resolved: ixgbe: fix pci device refcount leak As the comment of pci_get_domain_bus_and_slot() says, it returns a PCI device with refcount incremented, when finish using it, the caller must decrement the reference count by calling pci_dev_put(). In ixgbe_get_first_secondary_devfn() and ixgbe_x550em_a_has_mii(), pci_dev_put() is called to avoid leak.
CVE-2022-48899	In the Linux kernel, the following vulnerability has been resolved: drm/virtio: Fix GEM handle creation UAF Userspace can guess the handle value and try to race GEM object creation with handle close, resulting in a use-after-free if we dereference the object after dropping the handle's reference. For that reason, dropping the handle's reference must be done <i>after</i> we are done dereferencing the object.
CVE-2022-48904	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix I/O page table memory leak The current logic updates the I/O page table mode for the domain before calling the logic to free memory used for the page table. This results in IOMMU page table memory leak, and can be observed when launching VM w/ pass-through devices. Fix by freeing the memory used for page table before updating the mode.
CVE-2022-48905	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: free reset-work-item when flushing Fix a tiny memory leak when flushing the reset work queue.
CVE-2022-48906	In the Linux kernel, the following vulnerability has been resolved: mptcp: Correctly set DATA_FIN timeout when number of retransmits is large Syzkaller with UBSAN uncovered a scenario where a large number of DATA_FIN retransmits caused a shift-out-of-bounds in the DATA_FIN timeout calculation: <pre>===== UBSAN: shift-out-of-bounds in net/mptcp/protocol.c:470:29 shift exponent 32 is too large for 32-bit type 'unsigned int' CPU: 1 PID: 13059 Comm: kworker/1:0 Not tainted 5.17.0-rc2-00630-g5fbf21c90c60 #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 Workqueue: events mptcp_worker Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x1cd/0x134 lib/dump_stack.c:106 ubsan_epilogue+0xb/0x5a lib/ubsan.c:151 __ubsan_handle_shift_out_of_bounds.cold+0xb2/0x20e lib/ubsan.c:330 mptcp_set_datafin_timeout net/mptcp/protocol.c:470 [inline] __mptcp_retrans.cold+0x72/0x77 net/mptcp/protocol.c:2445 mptcp_worker+0x58a/0xa70 net/mptcp/protocol.c:2528 process_one_work+0x9df/0x16d0 kernel/ workqueue.c:2307 worker_thread+0x95/0xe10 kernel/workqueue.c:2454 kthread+0x2f4/0x3b0 kernel/kthread.c:377 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:295 </TASK> =====</pre> This change limits the maximum timeout by limiting the size of the shift, which keeps all intermediate values in-bounds.
CVE-2022-48909	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix connection leak There's a potential leak issue under following execution sequence : smc_release smc_connect_work if (sk->sk_state == SMC_INIT) send_clc_confirm tcp_abort(); ... sk.sk_state = SMC_ACTIVE smc_close_active switch(sk->sk_state) { ... case SMC_ACTIVE: smc_close_final() // then wait peer closed Unfortunately, tcp_abort() may discard CLC CONFIRM messages that are still in the tcp send buffer, in which case our connection token cannot be delivered to the server side, which means that we cannot get a passive close message at all. Therefore, it is impossible for the to be disconnected at all. This patch tries a very simple way to avoid this issue, once the state has changed to SMC_ACTIVE after tcp_abort(), we can actively abort the smc connection, considering that the state is SMC_INIT before tcp_abort(), abandoning the complete disconnection process should not cause too much problem. In fact, this problem may exist as long as the CLC CONFIRM message is not received by the server. Whether a timer should be added after smc_close_final() needs to be discussed in the future. But even so, this patch provides a faster release for connection in above case, it should also be valuable.

CVE ID	Description
CVE-2022-48910	In the Linux kernel, the following vulnerability has been resolved: net: ipv6: ensure we call <code>ipv6_mc_down()</code> at most once There are two reasons for <code>addrconf_notify()</code> to be called with <code>NETDEV_DOWN</code> : either the network device is actually going down, or IPv6 was disabled on the interface. If either of them stays down while the other is toggled, we repeatedly call the code for <code>NETDEV_DOWN</code> , including <code>ipv6_mc_down()</code> , while never calling the corresponding <code>ipv6_mc_up()</code> in between. This will cause a new entry in <code>idev->mc_tomb</code> to be allocated for each multicast group the interface is subscribed to, which in turn leaks one struct <code>ifmcastdr6</code> per nontrivial multicast group the interface is subscribed to. The following reproducer will leak at least \$n objects: ip addr add ff2e::4242/32 dev eth0 autojoin sysctl -w net.ipv6.conf.eth0.disable_ipv6=1 for i in \$(seq 1 \$n); do ip link set up eth0; ip link set down eth0 done Joining groups with <code>IPV6_ADD_MEMBERSHIP</code> (unprivileged) or setting the sysctl <code>net.ipv6.conf.eth0.forwarding</code> to 1 ($\Rightarrow$ subscribing to <code>ff02::2</code>) can also be used to create a nontrivial <code>idev->mc_list</code> , which will the leak objects with the right up-down-sequence. Based on both sources for <code>NETDEV_DOWN</code> events the interface IPv6 state should be considered: - not ready if the network interface is not ready OR IPv6 is disabled for it - ready if the network interface is ready AND IPv6 is enabled for it The functions <code>ipv6_mc_up()</code> and <code>ipv6_down()</code> should only be run when this state changes. Implement this by remembering when the IPv6 state is ready, and only run <code>ipv6_mc_down()</code> if it actually changed from ready to not ready. The other direction (not ready $\rightarrow$ ready) already works correctly, as: - the interface notification triggered codepath for <code>NETDEV_UP / NETDEV_CHANGE</code> returns early if ipv6 is disabled, and - the <code>disable_ipv6=0</code> triggered codepath skips fully initializing the interface as long as <code>addrconf_link_ready(dev)</code> returns false - calling <code>ipv6_mc_up()</code> repeatedly does not leak anything
CVE-2022-48911	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_queue: fix possible use-after-free Eric Dumazet says: The sock_hold() side seems suspect, because there is no guarantee that sk_refcnt is not already 0. On failure, we cannot queue the packet and need to indicate an error. The packet will be dropped by the caller. v2: split skb prefetch hunk into separate change
CVE-2022-48913	In the Linux kernel, the following vulnerability has been resolved: blktrace: fix use after free for struct blk_trace When tracing the whole disk, 'dropped' and 'msg' will be created under 'q->debugfs_dir' and 'bt->dir' is NULL, thus <code>blk_trace_free()</code> won't remove those files. What's worse, the following UAF can be triggered because of accessing stale 'dropped' and 'msg': ===== BUG: KASAN: use-after-free in blk_dropped_read+0x89/0x100 Read of size 4 at addr ffff88816912f3d8 by task blktrace/1188 CPU: 27 PID: 1188 Comm: blktrace Not tainted 5.17.0-rc4-next-20220217+ #469 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS ?-20190727_073836-4 Call Trace: <TASK> dump_stack_lvl+0x34/0x44 print_address_description.constprop.0.cold+0xab/0x381 ? blk_dropped_read+0x89/0x100 ? blk_dropped_read+0x89/0x100 kasan_report.cold+0x83/0xdf ? blk_dropped_read+0x89/0x100 kasan_check_range+0x140/0x1b0 blk_dropped_read+0x89/0x100 ? blk_create_buf_file_callback+0x20/0x20 ? kmem_cache_free+0xa1/0x500 ? do_sys_openat2+0x258/0x460 full_proxy_read+0x8f/0xc0 vfs_read+0xc6/0x260 ksys_read+0xb9/0x150 ? vfs_write+0x3d0/0x3d0 ? fpregs_assert_state_consistent+0x55/0x60 ? exit_to_user_mode_prepare+0x39/0x1e0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7fbc080d92fd Code: ce 20 00 00 75 10 b8 00 00 00 00 0f 05 48 3d 01 f0 ff ff 73 31 c3 48 83 1 RSP: 002b:00007fbb95ff9cb0 EFLAGS: 00000293 ORIG_RAX: 0000000000000000 RAX: ffffffff881da RBX: 00007fbb95ff9dc0 RCX: 00007fbc080d92fd RDX: 0000000000000010 RSI: 00007fbb95ff9cc0 RDI: 0000000000000045 RBP: 0000000000000045 R08: 0000000000406299 R09: 00000000fffffdd R10: 000000000153afa0 R11: 0000000000000293 R12: 00007fbb780008c R13: 00007fbb78000938 R14: 0000000000608b30 R15: 00007fbb780029c8 </TASK> Allocated by task 1050: kasan_save_stack+0x1e/0x40 __kasan_kmalloc+0x81/0xa0 do_blk_trace_setup+0xcb/0x410 __blk_trace_setup+0xac/0x130 blk_trace_ioctl+0xe9/0x1c0 blkdev_ioctl+0xf1/0x390 __x64_sys_ioctl+0xa5/0xe0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae Freed by task 1050: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 kasan_set_free_info+0x20/0x30 __kasan_slab_free+0x103/0x180 kfree+0x9a/0x4c0 __blk_trace_remove+0x53/0x70 blk_trace_ioctl+0x199/0x1c0 blkdev_common_ioctl+0x5e9/0xb30 blkdev_ioctl+0x1a5/0x390 __x64_sys_ioctl+0xa5/0xe0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae The buggy address belongs to the object at ffff88816912f380 which belongs to the cache kcalloc-96 of size 96 The buggy address is located 88 bytes inside of 96-byte region [ffff88816912f380, ffff88816912f3e0] The buggy address belongs to the page: page:000000009a1b4e7c refcount:1 mapcount:0 mapping:0000000000000000 index:0xf flags: 0x1ffffc00000(slab node=0 zone=2 lastcpuid=0x1ffff) raw: 0017ffffc0000200 ffffea00044f1100 dead000000000002 ffff88810004c780 raw: 0000000000000000 0000000000200020 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff88816912f280: fa fb fb fb fb fb fb fb fb fc fc fc ffff88816912f300: fa fb fb fb fb fb fb fb fb fc fc fc >ffff88816912f380: fa fb fb fb fb fb fb fb fb fc fc fc ^ ffff88816912f400: fa fb fb fb fb fb fb fb fb fc fc fc ffff88816912f480: fa fb fb fb fb fb fb fb fb fc fc fc fc =====

CVE ID	Description
CVE-2022-48914	In the Linux kernel, the following vulnerability has been resolved: xen/netfront: destroy queues before real_num_tx_queues is zeroed xennet_destroy_queues() relies on info->netdev->real_num_tx_queues to delete queues. Since d7dac083414eb5bb99a6d2ed53dc2c1b405224e5 ("net-sysfs: update the queue counts in the unregistration path"), unregister_netdev() indirectly sets real_num_tx_queues to 0. Those two facts together means, that xennet_destroy_queues() called from xennet_remove() cannot do its job, because it's called after unregister_netdev(). This results in kfree-ing queues that are still linked in napi, which ultimately crashes: BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 1 PID: 52 Comm: xenwatch Tainted: G W 5.16.10-1.32.fc32.qubes.x86_64+ #226 RIP: 0010:free_netdev+0xa3/0x1a0 Code: ff 48 89 df e8 2e e9 00 00 48 8b 43 50 48 8b 08 48 8d b8 a0 fe ff ff 48 8d a9 a0 fe ff ff 49 39 c4 75 26 eb 47 e8 ed c1 66 ff <48> 8b 85 60 01 00 00 48 8d 95 60 01 00 00 48 89 ef 48 2d 60 01 00 RSP: 0000:ffffc90000bafd00 EFLAGS: 00010286 RAX: 0000000000000000 RBX: ffff88800edad000 RCX: 0000000000000000 RDX: 0000000000000001 RSI: ffff88800bfc30 RDI: 00000000fffffffb RBP: ffffffffefea0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000001 R12: ffff88800edad050 R13: ffff8880065f8f88 R14: 0000000000000000 R15: ffff8880066c6680 FS: 0000000000000000(0000) GS:ffff8880f3300000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000000e998c006 CR4: 0000000003706e00 Call Trace: <TASK> xennet_remove+0x13d/0x300 [xen_netfront] xenbus_dev_remove+0x6d/0xf0 __device_release_driver+0x17a/0x240 device_release_driver+0x24/0x30 bus_remove_device+0xd8/0x140 device_del+0x18b/0x410 ? _raw_spin_unlock+0x16/0x30 ? klist_iter_exit+0x14/0x20 ? xenbus_dev_request_and_reply+0x80/0x80 device_unregister+0x13/0x60 xenbus_dev_changed+0x18e/0x1f0 xenwatch_thread+0xc0/0x1a0 ? do_wait_intr_irq+0xa0/0xa0 kthread+0x16b/0x190 ? set_kthread_struct+0x40/0x40 ret_from_fork+0x22/0x30 </TASK> Fix this by calling xennet_destroy_queues() from xennet_uninit(), when real_num_tx_queues is still available. This ensures that queues are destroyed when real_num_tx_queues is set to 0, regardless of how unregister_netdev() was called. Originally reported at https://github.com/QubesOS/qubes-issues/issues/7257
CVE-2022-48915	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Fix TZ_GET_TRIP NULL pointer dereference Do not call get_trip_hyst() from thermal_genl_cmd_tz_get_trip() if the thermal zone does not define one.

CVE ID	Description
CVE-2022-48916	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix double list_add when enabling VMD in scalable mode When enabling VMD and IOMMU scalable mode, the following kernel panic call trace/kernel log is shown in Eagle Stream platform (Sapphire Rapids CPU) during booting: pci 0000:59:00.5: Adding to iommu group 42 ... vmd 0000:59:00.5: PCI host bridge to bus 10000:80 pci 10000:80:01.0: [8086:352a] type 01 class 0x060400 pci 10000:80:01.0: reg 0x10: [mem 0x00000000-0x0001ffff 64bit] pci 10000:80:01.0: enabling Extended Tags pci 10000:80:01.0: PME# supported from D0 D3hot D3cold pci 10000:80:01.0: DMAR: Setup RID2PASID failed pci 10000:80:01.0: Failed to add to iommu group 42: -16 pci 10000:80:03.0: [8086:352b] type 01 class 0x060400 pci 10000:80:03.0: reg 0x10: [mem 0x00000000-0x0001ffff 64bit] pci 10000:80:03.0: enabling Extended Tags pci 10000:80:03.0: PME# supported from D0 D3hot D3cold ----- [cut here]----- kernel BUG at lib/list_debug.c:29! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 7 Comm: kworker/0:1 Not tainted 5.17.0-rc3+ #7 Hardware name: Lenovo ThinkSystem SR650V3/SB27A86647, BIOS ESE101Y-1.00 01/13/2022 Workqueue: events work_for_cpu_fn RIP: 0010: __list_add_valid.cold+0x26/0x3f Code: 9a 4a ab ff 4c 89 c1 48 c7 c7 40 0c d9 9e e8 b9 b1 fe ff 0f 0b 48 89 f2 4c 89 c1 48 89 fe 48 c7 c7 0f 0c d9 9e e8 a2 b1 fe ff <0f> 0b 48 89 d1 4c 89 c6 4c 89 ca 48 c7 c7 98 0c d9 9e e8 8b b1 fe RSP: 0000:ff5ad434865b3a40 EFLAGS: 00010246 RAX: 0000000000000058 RBX: ff4d61160b74b880 RCX: ff4d61255e1fffa8 RDX: 0000000000000000 RSI: 00000000ffffffffff RDI: ffffffff9fd34f20 RBP: ff4d611d8e245c00 R08: 0000000000000000 R09: ff5ad434865b3888 R10: ff5ad434865b3880 R11: ff4d61257fdc6fe8 R12: ff4d61160b74b8a0 R13: ff4d61160b74b8a0 R14: ff4d611d8e245c10 R15: ff4d611d8001ba70 FS: 0000000000000000(0000) GS:ff4d611d5ea00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ff4d611fa1401000 CR3: 00000000aa0210001 CR4: 0000000000771ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe07f0 DR7: 00000000000000400 PKRU: 55555554 Call Trace: <TASK> intel_pasid_alloc_table+0x9c/0x1d0 dmar_insert_one_dev_info+0x423/0x540 ? device_to_iommu+0x12d/0x2f0 intel_iommu_attach_device+0x116/0x290 __iommu_attach_device+0x1a/0x90 iommu_group_add_device+0x190/0x2c0 __iommu_probe_device+0x13e/0x250 iommu_probe_device+0x24/0x150 iommu_bus_notifier+0x69/0x90 blocking_notifier_call_chain+0x5a/0x80 device_add+0x3db/0x7b0 ? arch_memremap_can_ram_remap+0x19/0x50 ? memremap+0x75/0x140 pci_device_add+0x193/0x1d0 pci_scan_single_device+0xb9/0xf0 pci_scan_slot+0x4c/0x110 pci_scan_child_bus_extend+0x3a/0x290 vmd_enable_domain.constprop.0+0x63e/0x820 vmd_probe+0x163/0x190 local_pci_probe+0x42/0x80 work_for_cpu_fn+0x13/0x20 process_one_work+0x1e2/0x3b0 worker_thread+0x1c4/0x3a0 ? rescuer_thread+0x370/0x370 kthread+0xc7/0xf0 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x1f/0x30 </TASK> Modules linked in: ---[end trace 0000000000000000]--- ... Kernel panic - not syncing: Fatal exception Kernel Offset: 0x1ca00000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-0xfffffffffbffffff) ---[end Kernel panic - not syncing: Fatal exception]--- The following 'lspci' output shows devices '10000:80:*' are subdevices of the VMD device 0000:59:00.5: \$ lspci ... 0000:59:00.5 RAID bus controller: Intel Corporation Volume Management Device NVMe RAID Controller (rev 20) ... 10000:80:01.0 PCI bridge: Intel Corporation Device 352a (rev 03) 10000:80:03.0 PCI bridge: Intel Corporation Device 352b (rev 03) 10000:80:05.0 PCI bridge: Intel Corporation Device 352c (rev 03) 10000:80:07.0 PCI bridge: Intel Corporation Device 352d (rev 03) 10000:81:00.0 Non-Volatile memory controller: Intel Corporation NVMe Datacenter SSD [3DNAND, Beta Rock Controller] 10000:82:00 ---truncated---
CVE-2022-48929	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix crash due to out of bounds access into reg2btf_ids. When commit e6ac2450d6de ("bpf: Support bpf program calling kernel function") added kfunc support, it defined reg2btf_ids as a cheap way to translate the verifier reg type to the appropriate btf_vmlinux BTF ID, however commit c25b2ae13603 ("bpf: Replace PTR_TO_XXX_OR_NULL with PTR_TO_XXX PTR_MAYBE_NULL") moved the __BPF_REG_TYPE_MAX from the last member of bpf_reg_type enum to after the base register types, and defined other variants using type flag composition. However, now, the direct usage of reg->type to index into reg2btf_ids may no longer fall into __BPF_REG_TYPE_MAX range, and hence lead to out of bounds access and kernel crash on dereference of bad pointer.
CVE-2022-48931	In the Linux kernel, the following vulnerability has been resolved: configs: fix a race in configs_{,un}register_subsystem() When configs_register_subsystem() or configs_unregister_subsystem() is executing link_group() or unlink_group(), it is possible that two processes add or delete list concurrently. Some unfortunate interleavings of them can cause kernel panic. One of cases is: A --> B --> C --> D A <-- B <-- C <-- D delete list_head *B delete list_head *C ----- ----- configs_unregister_subsystem configs_unregister_subsystem unlink_group unlink_group unlink_obj unlink_obj list_del_init list_del_init __list_del_entry __list_del_entry __list_del __list_del // next == C next->prev = prev next->prev = prev prev->next = next // prev == B prev->next = next Fix this by adding mutex when calling link_group() or unlink_group(), but parent configs_subsystem is NULL when config_item is root. So I create a mutex configs_subsystem_mutex.

CVE ID	Description
CVE-2022-48933	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: fix memory leak during stateful obj update stateful objects can be updated from the control plane. The transaction logic allocates a temporary object for this purpose. The ->init function was called for this object, so plain kfree() leaks resources. We must call ->destroy function of the object. nft_obj_destroy does this, but it also decrements the module refcount, but the update path doesn't increment it. To avoid special-casing the update object release, do module_get for the update case too and release it via nft_obj_destroy().
CVE-2022-48938	In the Linux kernel, the following vulnerability has been resolved: CDC-NCM: avoid overflow in sanity checking A broken device may give an extreme offset like 0xFFFF0 and a reasonable length for a fragment. In the sanity check as formulated now, this will create an integer overflow, defeating the sanity check. Both offset and offset + len need to be checked in such a manner that no overflow can occur. And those quantities should be unsigned.
CVE-2022-48939	In the Linux kernel, the following vulnerability has been resolved: bpf: Add schedule points in batch ops syzbot reported various soft lockups caused by bpf batch operations. INFO: task kworker/1:1:27 blocked for more than 140 seconds. INFO: task hung in rcu_barrier Nothing prevents batch ops to process huge amount of data, we need to add schedule points in them. Note that maybe_wait_bpf_programs(map) calls from generic_map_delete_batch() can be factorized by moving the call after the loop. This will be done later in -next tree once we get this fix merged, unless there is strong opinion doing this optimization sooner.
CVE-2022-48943	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: make apf token non-zero to fix bug In current async pagefault logic, when a page is ready, KVM relies on kvm_arch_can_dequeue_async_page_present() to determine whether to deliver a READY event to the Guest. This function test token value of struct kvm_vcpu_pv_apf_data, which must be reset to zero by Guest kernel when a READY event is finished by Guest. If value is zero meaning that a READY event is done, so the KVM can deliver another. But the kvm_arch_setup_async_pf() may produce a valid token with zero value, which is confused with previous mention and may lead the loss of this READY event. This bug may cause task blocked forever in Guest: INFO: task stress:7532 blocked for more than 1254 seconds. Not tainted 5.10.0 #16 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:stress state:D stack: 0 pid: 7532 ppid: 1409 flags:0x00000080 Call Trace: __schedule+0x1e7/0x650 schedule+0x46/0xb0 kvm_async_pf_task_wait_schedule+0xad/0xe0 ? exit_to_user_mode_prepare+0x60/0x70 __kvm_handle_async_pf+0x4f/0xb0 ? asm_exc_page_fault+0x8/0x30 exc_page_fault+0x6f/0x110 ? asm_exc_page_fault+0x8/0x30 asm_exc_page_fault+0x1e/0x30 RIP: 0033:0x402d00 RSP: 002b:00007ffd31912500 EFLAGS: 00010206 RAX: 00000000000071000 RBX: ffffffff00000000 RCX: 000000000021a32b0 RDX: 0000000000007d011 RSI: 0000000000007d000 RDI: 000000000021262b0 RBP: 000000000021262b0 R08: 0000000000000003 R09: 0000000000000086 R10: 000000000000000eb R11: 00007fbbdf2baa0 R12: 0000000000000000 R13: 0000000000000002 R14: 0000000000007d000 R15: 0000000000001000
CVE-2022-48946	In the Linux kernel, the following vulnerability has been resolved: udf: Fix preallocation discarding at indirect extent boundary When preallocation extent is the first one in the extent block, the code would corrupt extent tree header instead. Fix the problem and use udf_delete_aext() for deleting extent to avoid some code duplication.
CVE-2022-48950	In the Linux kernel, the following vulnerability has been resolved: perf: Fix perf_pending_task() UaF Per syzbot it is possible for perf_pending_task() to run after the event is free()'d. There are two related but distinct cases: - the task_work was already queued before destroying the event; - destroying the event itself queues the task_work. The first cannot be solved using task_work_cancel() since perf_release() itself might be called from a task_work (____fput), which means the current->task_works list is already empty and task_work_cancel() won't be able to find the perf_pending_task() entry. The simplest alternative is extending the perf_event lifetime to cover the task_work. The second is just silly, queueing a task_work while you know the event is going away makes no sense and is easily avoided by re-arranging how the event is marked STATE_DEAD and ensuring it goes through STATE_OFF on the way down.
CVE-2022-48951	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Check bounds for second channel in snd_soc_put_volsw_sx() The bounds checks in snd_soc_put_volsw_sx() are only being applied to the first channel, meaning it is possible to write out of bounds values to the second channel in stereo controls. Add appropriate checks.

CVE ID	Description
CVE-2022-48953	In the Linux kernel, the following vulnerability has been resolved: rtc: cmos: Fix event handler registration ordering issue Because <code>acpi_install_fixed_event_handler()</code> enables the event automatically on success, it is incorrect to call it before the handler routine passed to it is ready to handle events. Unfortunately, the rtc-cmos driver does exactly the incorrect thing by calling <code>cmos_wake_setup()</code> , which passes <code>rtc_handler()</code> to <code>acpi_install_fixed_event_handler()</code> , before <code>cmos_do_probe()</code> , because <code>rtc_handler()</code> uses <code>dev_get_drvdata()</code> to get to the cmos object pointer and the driver data pointer is only populated in <code>cmos_do_probe()</code> . This leads to a NULL pointer dereference in <code>rtc_handler()</code> on boot if the RTC fixed event happens to be active at the init time. To address this issue, change the initialization ordering of the driver so that <code>cmos_wake_setup()</code> is always called after a successful <code>cmos_do_probe()</code> call. While at it, change <code>cmos_pnp_probe()</code> to call <code>cmos_do_probe()</code> after the initial <code>if ()</code> statement used for computing the IRQ argument to be passed to <code>cmos_do_probe()</code> which is cleaner than calling it in each branch of that <code>if ()</code> (local variable "irq" can be of type int, because it is passed to that function as an argument of type int). Note that commit 6492fed7d8c9 ("rtc: rtc-cmos: Do not check ACPI_FADT_LOW_POWER_S0") caused this issue to affect a larger number of systems, because previously it only affected systems with <code>ACPI_FADT_LOW_POWER_S0</code> set, but it is present regardless of that commit.
CVE-2022-48961	In the Linux kernel, the following vulnerability has been resolved: net: mdio: fix unbalanced fwnode reference count in <code>mdio_device_release()</code> There is warning report about of_node refcount leak while probing mdio device: OF: ERROR: memory leak, expected refcount 1 instead of 2, of_node_get()/of_node_put() unbalanced - destroy cset entry: attach overlay node /spi/soc@0/mdio@710700c0/ethernet@4 In <code>of_mdibus_register_device()</code> , we increase fwnode refcount by <code>fwnode_handle_get()</code> before associating the of_node with mdio device, but it has never been decreased in normal path. Since that, in <code>mdio_device_release()</code> , it needs to call <code>fwnode_handle_put()</code> in addition instead of calling <code>kfree()</code> directly. After above, just calling <code>mdio_device_free()</code> in the error handle path of <code>of_mdibus_register_device()</code> is enough to keep the refcount balanced.
CVE-2022-48969	In the Linux kernel, the following vulnerability has been resolved: xen-netfront: Fix NULL string after live migration A NAPI is setup for each network string to poll data to kernel The string with source host is destroyed before live migration and new string with target host is setup after live migration. The NAPI for the old string is not deleted until setup new string with target host after migration. With <code>busy_poll/</code> <code>busy_read</code> enabled, the NAPI can be polled before got deleted when resume VM. BUG: unable to handle kernel NULL pointer dereference at 0000000000000008 IP: <code>xennet_poll+0xae/0xd20 PGD 0 P4D 0</code> Oops: 0000 [#1] SMP PTI Call Trace: <code>finish_task_switch+0x71/0x230 timerqueue_del+0x1d/0x40 hrtimer_try_to_cancel+0xb5/0x110 xennet_alloc_rx_buffers+0x2a0/0x2a0 napi_busy_loop+0xdb/0x270 sock_poll+0x87/0x90 do_sys_poll+0x26f/0x580 tracing_map_insert+0x1d4/0x2f0 event_hist_trigger+0x14a/0x260 finish_task_switch+0x71/0x230 __schedule+0x256/0x890 recalc_sigpending+0x1b/0x50 xen_sched_clock+0x15/0x20 __rb_reserve_next+0x12d/0x140 ring_buffer_lock_reserve+0x123/0x3d0 event_triggers_call+0x87/0xb0 trace_event_buffer_commit+0x1c4/0x210 xen_clocksource_get_cycles+0x15/0x20 ktime_get_ts64+0x51/0xf0 SyS_ppoll+0x160/0x1a0 SyS_ppoll+0x160/0x1a0 do_syscall_64+0x73/0x130 entry_SYSCALL_64_after_hwframe+0x41/0xae ... RIP: <code>xennet_poll+0xae/0xd20 RSP: ffffb4f041933900 CR2: 0000000000000008</code> ---[end trace f8601785b354351c]--- xen frontend should remove the NAPIs for the old strings before live migration as the bond strings are destroyed There is a tiny window between the strings are set to NULL and the NAPIs are disabled, It is safe as the NAPI threads are still frozen at that time</code>
CVE-2022-48971	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix not cleanup led when <code>bt_init</code> fails <code>bt_init()</code> calls <code>bt_leds_init()</code> to register led, but if it fails later, <code>bt_leds_cleanup()</code> is not called to unregister it. This can cause panic if the argument "bluetooth-power" in text is freed and then another <code>led_trigger_register()</code> tries to access it: BUG: unable to handle page fault for address: <code>ffffffffffc06d3bc0 RIP: 0010:strcmp+0xc/0x30 Call Trace: <TASK> led_trigger_register+0x10d/0x4f0 led_trigger_register_simple+0x7d/0x100 bt_init+0x39/0xf7 [bluetooth] do_one_initcall+0xd0/0x4e0</code>
CVE-2022-48972	In the Linux kernel, the following vulnerability has been resolved: mac802154: fix missing INIT_LIST_HEAD in <code>ieee802154_if_add()</code> Kernel fault injection test reports null-ptr-deref as follows: BUG: kernel NULL pointer dereference, address: 0000000000000008 RIP: <code>0010:cfg802154_netdev_notifier_call+0x120/0x310 include/linux/list.h:114 Call Trace: <TASK> raw_notifier_call_chain+0x6d/0xa0 kernel/notifier.c:87 call_netdevice_notifiers_info+0x6e/0xc0 net/core/dev.c:1944 unregister_netdevice_many_notify+0x60d/0xcb0 net/core/dev.c:1982 unregister_netdevice_queue+0x154/0x1a0 net/core/dev.c:10879 register_netdevice+0x9a8/0xb90 net/core/dev.c:10083 ieee802154_if_add+0x6ed/0x7e0 net/mac802154/iface.c:659 ieee802154_register_hw+0x29c/0x330 net/mac802154/main.c:229 mcr20a_probe+0xaa/0xcb1 drivers/net/ieee802154/mcr20a.c:1316 ieee802154_if_add() allocates <code>wpan_dev</code> as netdev's private data, but not init the list in struct <code>wpan_dev</code>. <code>cfg802154_netdev_notifier_call()</code> manage the list when device register/unregister, and may lead to null-ptr-deref. Use <code>INIT_LIST_HEAD()</code> on it to initialize it correctly.</code>

CVE ID	Description
CVE-2022-48975	In the Linux kernel, the following vulnerability has been resolved: gpiolib: fix memory leak in gpiochip_setup_dev() Here is a backtrace report about memory leak detected in gpiochip_setup_dev(): unreferenced object 0xffff88810b406400 (size 512): comm "python3", pid 1682, jiffies 4295346908 (age 24.090s) backtrace: kmalloc_trace device_add device_private_init at drivers/base/core.c:3361 (inlined by) device_add at drivers/base/core.c:3411 cdev_device_add gpiolib_cdev_register gpiochip_setup_dev gpiochip_add_data_with_key gcdev_register() & gcdev_unregister() would call device_add() & device_del() (no matter CONFIG_GPIO_CDEV is enabled or not) to register/unregister device. However, if device_add() succeeds, some resource (like struct device_private allocated by device_private_init()) is not released by device_del(). Therefore, after device_add() succeeds by gcdev_register(), it needs to call put_device() to release resource in the error handle path. Here we move forward the register of release function, and let it release every piece of resource by put_device() instead of kfree(). While at it, fix another subtle issue, i.e. when gc->ngpio is equal to 0, we still call kcalloc() and, in case of further error, kfree() on the ZERO_PTR pointer, which is not NULL. It's not a bug per se, but rather waste of the resources and potentially wrong expectation about contents of the gdev->descs variable.
CVE-2022-48977	In the Linux kernel, the following vulnerability has been resolved: can: af_can: fix NULL pointer dereference in can_rcv_filter Analogue to commit 8aa59e355949 ("can: af_can: fix NULL pointer dereference in can_rx_register()") we need to check for a missing initialization of ml_priv in the receive path of CAN frames. Since commit 4e096a18867a ("net: introduce CAN specific pointer in the struct net_device") the check for dev->type to be ARPHRD_CAN is not sufficient anymore since bonding or tun netdevices claim to be CAN devices but do not initialize ml_priv accordingly.
CVE-2022-48978	In the Linux kernel, the following vulnerability has been resolved: HID: core: fix shift-out-of-bounds in hid_report_raw_event Syzbot reported shift-out-of-bounds in hid_report_raw_event. microsoft 0003:045E:07DA.0001: hid_field_extract() called with n (128) > 32! (swapper/0) ===== UBSAN: shift-out-of-bounds in drivers/hid/hid-core.c:1323:20 shift exponent 127 is too large for 32-bit type 'int' CPU: 0 PID: 0 Comm: swapper/0 Not tainted 6.1.0-rc4-syzkaller-00159-g4bbf3422df78 #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Call Trace: <IRQ> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x1e3/0x2cb lib/dump_stack.c:106 ubsan_epilogue lib/ubsan.c:151 [inline] __ubsan_handle_shift_out_of_bounds+0x3a6/0x420 lib/ubsan.c:322 snto32 drivers/hid/hid-core.c:1323 [inline] hid_input_fetch_field drivers/hid/hid-core.c:1572 [inline] hid_process_report drivers/hid/hid-core.c:1665 [inline] hid_report_raw_event+0xd56/0x18b0 drivers/hid/hid-core.c:1998 hid_input_report+0x408/0x4f0 drivers/hid/hid-core.c:2066 hid_irq_in+0x459/0x690 drivers/hid/usbhid/hid-core.c:284 __usb_hcd_giveback_urb+0x369/0x530 drivers/usb/core/hcd.c:1671 dummy_timer+0x86b/0x3110 drivers/usb/gadget/udc/dummy_hcd.c:1988 call_timer_fn+0xf5/0x210 kernel/time/timer.c:1474 expire_timers kernel/time/timer.c:1519 [inline] __run_timers+0x76a/0x980 kernel/time/timer.c:1790 run_timer_softirq+0x63/0xf0 kernel/time/timer.c:1803 __do_softirq+0x277/0x75b kernel/softirq.c:571 __irq_exit_rcu+0xec/0x170 kernel/softirq.c:650 irq_exit_rcu+0x5/0x20 kernel/softirq.c:662 sysvec_apic_timer_interrupt+0x91/0xb0 arch/x86/kernel/apic/apic.c:1107 ===== If the size of the integer (unsigned n) is bigger than 32 in snto32(), shift exponent will be too large for 32-bit type 'int', resulting in a shift-out-of-bounds bug. Fix this by adding a check on the size of the integer (unsigned n) in snto32(). To add support for n greater than 32 bits, set n to 32, if n is greater than 32.
CVE-2022-48979	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix array index out of bound error in DCN32 DML [Why&How] LinkCapacitySupport array is indexed with the number of voltage states and not the number of max DPPs. Fix the error by changing the array declaration to use the correct (larger) array size of total number of voltage states.

CVE ID	Description
CVE-2022-48982	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix crash when replugging CSR fake controllers It seems fake CSR 5.0 clones can cause the suspend notifier to be registered twice causing the following kernel panic: [71.986122] Call Trace: [71.986124] <TASK> [71.986125] blocking_notifier_chain_register+0x33/0x60 [71.986130] hci_register_dev +0x316/0x3d0 [bluetooth 99b5497ea3d09708fa1366c1dc03288bf3cca8da] [71.986154] btusb_probe+0x979/0xd85 [btusb e1e0605a4f4c01984a4b9c8ac58c3666ae287477] [71.986159] ? __pm_runtime_set_status+0x1a9/0x300 [71.986162] ? ktime_get_mono_fast_ns+0x3e/0x90 [71.986167] usb_probe_interface+0xe3/0x2b0 [71.986171] really_probe+0xdb/0x380 [71.986174] ? pm_runtime_barrier+0x54/0x90 [71.986177] __driver_probe_device+0x78/0x170 [71.986180] driver_probe_device+0x1f/0x90 [71.986183] __device_attach_driver+0x89/0x110 [71.986186] ? driver_allows_async_probing+0x70/0x70 [71.986189] bus_for_each_drv+0x8c/0xe0 [71.986192] __device_attach+0xb2/0x1e0 [71.986195] bus_probe_device+0x92/0xb0 [71.986198] device_add +0x422/0x9a0 [71.986201] ? sysfs_merge_group+0xd4/0x110 [71.986205] usb_set_configuration +0x57a/0x820 [71.986208] usb_generic_driver_probe+0x4f/0x70 [71.986211] usb_probe_device +0x3a/0x110 [71.986213] really_probe+0xdb/0x380 [71.986216] ? pm_runtime_barrier+0x54/0x90 [71.986219] __driver_probe_device+0x78/0x170 [71.986221] driver_probe_device+0x1f/0x90 [71.986224] __device_attach_driver+0x89/0x110 [71.986227] ? driver_allows_async_probing +0x70/0x70 [71.986230] bus_for_each_drv+0x8c/0xe0 [71.986232] __device_attach+0xb2/0x1e0 [71.986235] bus_probe_device+0x92/0xb0 [71.986237] device_add+0x422/0x9a0 [71.986239] ? _dev_info+0x7d/0x98 [71.986242] ? blake2s_update+0x4c/0xc0 [71.986246] usb_new_device.cold +0x148/0x36d [71.986250] hub_event+0xa8a/0x1910 [71.986255] process_one_work+0x1c4/0x380 [71.986259] worker_thread+0x51/0x390 [71.986262] ? rescuer_thread+0x3b0/0x3b0 [71.986264] kthread+0xdb/0x110 [71.986266] ? kthread_complete_and_exit+0x20/0x20 [71.986268] ret_from_fork +0x1f/0x30 [71.986273] </TASK> [71.986274] ---[end trace 0000000000000000]--- [71.986284] btusb: probe of 2-1.6:1.0 failed with error -17
CVE-2022-48985	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix race on per-CQ variable napi_work_done After calling napi_complete_done(), the NAPIF_STATE_SCHED bit may be cleared, and another CPU can start napi thread and access per-CQ variable, cq->work_done. If the other thread (for example, from busy_poll) sets it to a value >= budget, this thread will continue to run when it should stop, and cause memory corruption and panic. To fix this issue, save the per-CQ work_done variable in a local variable before napi_complete_done(), so it won't be corrupted by a possible concurrent thread after napi_complete_done(). Also, add a flag bit to advertise to the NIC firmware: the NAPI work_done variable race is fixed, so the driver is able to reliably support features like busy_poll.
CVE-2022-48986	In the Linux kernel, the following vulnerability has been resolved: mm/gup: fix gup_pud_range() for dax For dax pud, pud_huge() returns true on x86. So the function works as long as hugetlb is configured. However, dax doesn't depend on hugetlb. Commit 414fd080d125 ("mm/gup: fix gup_pmd_range() for dax") fixed devmap-backed huge PMDs, but missed devmap-backed huge PUDs. Fix this as well. This fixes the below kernel panic: general protection fault, probably for non-canonical address 0x69e7c000cc478: 0000 [#1] SMP < snip > Call Trace: <TASK> get_user_pages_fast +0x1f/0x40 iov_iter_get_pages+0xc6/0x3b0 ? mempool_alloc+0x5d/0x170 bio_iov_iter_get_pages +0x82/0x4e0 ? bvec_alloc+0x91/0xc0 ? bio_alloc_bioset+0x19a/0x2a0 blkdev_direct_IO +0x282/0x480 ? __io_complete_rw_common+0xc0/0xc0 ? filemap_range_has_page+0x82/0xc0 generic_file_direct_write+0x9d/0x1a0 ? inode_update_time+0x24/0x30 __generic_file_write_iter +0xbd/0x1e0 blkdev_write_iter+0xb4/0x150 ? io_import_iovec+0x8d/0x340 io_write+0xf9/0x300 io_issue_sqe+0x3c3/0x1d30 ? sysvec_reschedule_ipi+0x6c/0x80 __io_queue_sqe+0x33/0x240 ? fget +0x76/0xa0 io_submit_sqes+0xe6a/0x18d0 ? __fget_light+0xd1/0x100 __x64_sys_io_uring_enter +0x199/0x880 ? __context_tracking_enter+0x1f/0x70 ? irqentry_exit_to_user_mode+0x24/0x30 ? irqentry_exit+0x1d/0x30 ? __context_tracking_exit+0xe/0x70 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x61/0xcb RIP: 0033:0x7fc97c11a7be < snip > </TASK> ---[end trace 48b2e0e67debcaeb]--- RIP: 0010:internal_get_user_pages_fast+0x340/0x990 < snip > Kernel panic - not syncing: Fatal exception Kernel Offset: disabled
CVE-2022-48987	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-dv-timings.c: fix too strict blanking sanity checks Sanity checks were added to verify the v4l2_bt_timings blanking fields in order to avoid integer overflows when userspace passes weird values. But that assumed that userspace would correctly fill in the front porch, backporch and sync values, but sometimes all you know is the total blanking, which is then assigned to just one of these fields. And that can fail with these checks. So instead set a maximum for the total horizontal and vertical blanking and check that each field remains below that. That is still sufficient to avoid integer overflows, but it also allows for more flexibility in how userspace fills in these fields.

CVE ID	Description
CVE-2022-48988	In the Linux kernel, the following vulnerability has been resolved: memcg: fix possible use-after-free in memcg_write_event_control() memcg_write_event_control() accesses the dentry->d_name of the specified control fd to route the write call. As a cgroup interface file can't be renamed, it's safe to access d_name as long as the specified file is a regular cgroup file. Also, as these cgroup interface files can't be removed before the directory, it's safe to access the parent too. Prior to 347c4a874710 ("memcg: remove cgroup_event->cft"), there was a call to __file_cft() which verified that the specified file is a regular cgroupfs file before further accesses. The cftype pointer returned from __file_cft() was no longer necessary and the commit inadvertently dropped the file type check with it allowing any file to slip through. With the invariants broken, the d_name and parent accesses can now race against renames and removals of arbitrary files and cause use-after-free's. Fix the bug by resurrecting the file type check in __file_cft(). Now that cgroupfs is implemented through kernfs, checking the file operations needs to go through a layer of indirection. Instead, let's check the superblock and dentry type.
CVE-2022-4899	A vulnerability was found in zstd v1.4.10, where an attacker can supply empty string as an argument to the command line tool to cause buffer overrun.
CVE-2022-48990	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix use-after-free during gpu recovery [Why] [754.862560] refcount_t: underflow; use-after-free. [754.862898] Call Trace: [754.862903] <TASK> [754.862913] amdgpu_job_free_cb+0xc2/0xe1 [amdgpu] [754.863543] drm_sched_main.cold+0x34/0x39 [amd_sched] [How] The fw_fence may be not init, check whether dma_fence_init is performed before job free
CVE-2022-48991	In the Linux kernel, the following vulnerability has been resolved: mm/khugepaged: invoke MMU notifiers in shmem/file collapse paths Any codepath that zaps page table entries must invoke MMU notifiers to ensure that secondary MMUs (like KVM) don't keep accessing pages which aren't mapped anymore. Secondary MMUs don't hold their own references to pages that are mirrored over, so failing to notify them can lead to page use-after-free. I'm marking this as addressing an issue introduced in commit f3f0e1d2150b ("khugepaged: add support of collapse for tmpfs/shmem pages"), but most of the security impact of this only came in commit 27e1f8273113 ("khugepaged: enable collapse pmd for pte-mapped THP"), which actually omitted flushes for the removal of present PTEs, not just for the removal of empty page tables.
CVE-2022-48994	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: Fix function prototype mismatch in snd_seq_expand_var_event With clang's kernel control flow integrity (kCFI, CONFIG_CFI_CLANG), indirect call targets are validated against the expected function pointer prototype to make sure the call target is valid to help mitigate ROP attacks. If they are not identical, there is a failure at run time, which manifests as either a kernel panic or thread getting killed. seq_copy_in_user() and seq_copy_in_kernel() did not have prototypes matching snd_seq_dump_func_t. Adjust this and remove the casts. There are not resulting binary output differences. This was found as a result of Clang's new -Wcast-function-type-strict flag, which is more sensitive than the simpler -Wcast-function-type, which only checks for type width mismatches.

CVE ID	Description
CVE-2022-49003	In the Linux kernel, the following vulnerability has been resolved: nvme: fix SRCU protection of nvme_ns_head list Walking the nvme_ns_head siblings list is protected by the head's srcu in nvme_ns_head_submit_bio() but not nvme_mpath_revalidate_paths(). Removing namespaces from the list also fails to synchronize the srcu. Concurrent scan work can therefore cause use-after-frees. Hold the head's srcu lock in nvme_mpath_revalidate_paths() and synchronize with the srcu, not the global RCU, in nvme_ns_remove(). Observed the following panic when making NVMe/RDMA connections with native multipath on the Rocky Linux 8.6 kernel (it seems the upstream kernel has the same race condition). Disassembly shows the faulting instruction is cmp 0x50(%rdx),%rcx; computing capacity != get_capacity(ns->disk). Address 0x50 is dereferenced because ns->disk is NULL. The NULL disk appears to be the result of concurrent scan work freeing the namespace (note the log line in the middle of the panic). [37314.206036] BUG: unable to handle kernel NULL pointer dereference at 0000000000000050 [37314.206036] nvme0n3: detected capacity change from 0 to 11811160064 [37314.299753] PGD 0 P4D 0 [37314.299756] Oops: 0000 [#1] SMP PTI [37314.299759] CPU: 29 PID: 322046 Comm: kworker/u98:3 Kdump: loaded Tainted: G W X ----- - - 4.18.0-372.32.1.el8test86.x86_64 #1 [37314.299762] Hardware name: Dell Inc. PowerEdge R720/0JP31P, BIOS 2.7.0 05/23/2018 [37314.299763] Workqueue: nvme-wq nvme_scan_work [nvme_core] [37314.299783] RIP: 0010:nvme_mpath_revalidate_paths+0x26/0xb0 [nvme_core] [37314.299790] Code: 1f 44 00 00 66 66 66 90 55 53 48 8b 5f 50 48 8b 83 c8 c9 00 00 48 8b 13 48 8b 48 50 48 39 d3 74 20 48 8d 42 d0 48 8b 50 20 <48> 3b 4a 50 74 05 f0 80 60 70 ef 48 8b 50 30 48 8d 42 d0 48 39 d3 [37315.058803] RSP: 0018:ffffa28f913d10 EFLAGS: 00010202 [37315.121316] RAX: ffff927a077da800 RBX: ffff92991dd70000 RCX: 0000000001600000 [37315.206704] RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff92991b719800 [37315.292106] RBP: ffff929a6b70c000 R08: 000000010234cd4a R09: c0000000ffff7fff [37315.377501] R10: 0000000000000001 R11: ffff928f913a30 R12: 0000000000000000 [37315.462889] R13: ffff92992716600c R14: ffff929964e6e030 R15: ffff92991dd70000 [37315.548286] FS: 0000000000000000(0000) GS:ffff92b87fb80000(0000) knlGS:0000000000000000 [37315.645111] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [37315.713871] CR2: 0000000000000050 CR3: 0000002208810006 CR4: 000000000000606e [37315.799267] Call Trace: [37315.828515] nvme_update_ns_info+0x1ac/0x250 [nvme_core] [37315.892075] nvme_validate_or_alloc_ns+0x2ff/0xa00 [nvme_core] [37315.961871] ? __blk_mq_free_request+0x6b/0x90 [37316.015021] nvme_scan_work+0x151/0x240 [nvme_core] [37316.073371] process_one_work+0x1a7/0x360 [37316.121318] ? create_worker+0x1a0/0x1a0 [37316.168227] worker_thread+0x30/0x390 [37316.212024] ? create_worker+0x1a0/0x1a0 [37316.258939] kthread+0x10a/0x120 [37316.297557] ? set_kthread_struct+0x50/0x50 [37316.347590] ret_from_fork+0x35/0x40 [37316.390360] Modules linked in: nvme_rdma nvme_tcp(X) nvme_fabrics nvme_core netconsole iscsi_tcp libiscsi_tcp dm_queue_length dm_service_time nf_contrack_netlink br_netfilter bridge stp llc overlay nft_chain_nat ipt_MASQUERADE nf_nat xt_addrtype xt_CT nft_counter xt_state xt_conntrack nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 xt_comment xt_multiport nft_compat nf_tables libcrc32c nfnetlink dm_multipath tg3 rpcrdma sunrpc rdma_ucm ib_srpt ib_isert iscsi_target_mod target_core_mod ib_iser libiscsi scsi_transport_iscsi ib_umad rdma_cm ib_ipoib iw_cm ib_cm intel_rapl_msr iTCO_wdt iTCO_vendor_support dcdbas intel_rapl_common sb_edac x86_pkg_temp_thermal intel_powerclamp coretemp kvm_intel ipmi_ssif kvm irqbypass crct10dif_pclmul crc32_pclmul mlx5_ib ghash_clmulni_intel ib_uverbs rapl intel_cstate intel_uncore ib_core ipmi_si joydev mei_me pcspkr ipmi_devintf mei_lpc_ich wmi ipmi_msghandler acpi_power_meter ex ---truncated---
CVE-2022-49005	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Fix bounds check for _sx controls For _sx controls the semantics of the max field is not the usual one, max is the number of steps rather than the maximum value. This means that our check in snd_soc_put_volsw_sx() needs to just check against the maximum value.

CVE ID	Description
CVE-2022-49006	In the Linux kernel, the following vulnerability has been resolved: tracing: Free buffers when a used dynamic event is removed After 65536 dynamic events have been added and removed, the "type" field of the event then uses the first type number that is available (not currently used by other events). A type number is the identifier of the binary blobs in the tracing ring buffer (known as events) to map them to logic that can parse the binary blob. The issue is that if a dynamic event (like a kprobe event) is traced and is in the ring buffer, and then that event is removed (because it is dynamic, which means it can be created and destroyed), if another dynamic event is created that has the same number that new event's logic on parsing the binary blob will be used. To show how this can be an issue, the following can crash the kernel: # cd /sys/kernel/tracing # for i in `seq 65536`; do echo 'p:kprobes/foo do_sys_openat2 \$arg1:u32' > kprobe_events # done For every iteration of the above, the writing to the kprobe_events will remove the old event and create a new one (with the same format) and increase the type number to the next available on until the type number reaches over 65535 which is the max number for the 16 bit type. After it reaches that number, the logic to allocate a new number simply looks for the next available number. When an dynamic event is removed, that number is then available to be reused by the next dynamic event created. That is, once the above reaches the max number, the number assigned to the event in that loop will remain the same. Now that means deleting one dynamic event and created another will reuse the previous events type number. This is where bad things can happen. After the above loop finishes, the kprobes/foo event which reads the do_sys_openat2 function call's first parameter as an integer. # echo 1 > kprobes/foo/enable # cat /etc/passwd > /dev/null # cat trace cat-2211 [005] 2007.849603: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 cat-2211 [005] 2007.849620: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 cat-2211 [005] 2007.849838: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 cat-2211 [005] 2007.849880: foo: (do_sys_openat2+0x0/0x130) arg1=4294967196 # echo 0 > kprobes/foo/enable Now if we delete the kprobe and create a new one that reads a string: # echo 'p:kprobes/foo do_sys_openat2 +0(\$arg2):string' > kprobe_events And now we can the trace: # cat trace sendmail-1942 [002] 530.136320: foo: (do_sys_openat2+0x0/0x240) arg1= cat-2046 [004] 530.930817: foo: (do_sys_openat2+0x0/0x240) arg1= cat-2046 [004] 530.930961: foo: (do_sys_openat2+0x0/0x240) arg1= cat-2046 [004] 530.934278: foo: (do_sys_openat2+0x0/0x240) arg1= cat-2046 [004] 530.934563: foo: (do_sys_openat2+0x0/0x240) arg1= ---truncated---
CVE-2022-49013	In the Linux kernel, the following vulnerability has been resolved: sctp: fix memory leak in sctp_stream_outq_migrate() When sctp_stream_outq_migrate() is called to release stream out resources, the memory pointed to by prio_head in stream out is not released. The memory leak information is as follows: unreferenced object 0xffff88801fe79f80 (size 64): comm "sctp_repo", pid 7957, jiffies 4294951704 (age 36.480s) hex dump (first 32 bytes): 80 9f e7 1f 80 88 ff ff 80 9f e7 1f 80 88 ff ff 90 9f e7 1f 80 88 ff ff 90 9f e7 1f 80 88 ff ff backtrace: [<ffff8b1b215c6>] kmalloc_trace+0x26/0x60 [<ffff8b1b215c6>] sctp_sched_prio_set+0x4cc/0x770 [<ffff8b1b215c6>] sctp_stream_init_ext+0xd2/0x1b0 [<ffff8b1b215c6>] sctp_sendmsg_to_asoc+0x1614/0x1a30 [<ffff8b1b215c6>] sctp_sendmsg+0xda1/0x1ef0 [<ffff8b1b215c6>] inet_sendmsg+0x9d/0xe0 [<ffff8b1b215c6>] sock_sendmsg+0xd3/0x120 [<ffff8b1b215c6>] __sys_sendto+0x23a/0x340 [<ffff8b1b215c6>] __x64_sys_sendto+0xe1/0x1b0 [<ffff8b1b215c6>] do_syscall_64+0x39/0xb0 [<ffff8b1b215c6>] entry_SYSCALL_64_after_hwframe+0x63/0xcd

CVE ID	Description
CVE-2022-49014	In the Linux kernel, the following vulnerability has been resolved: net: tun: Fix use-after-free in tun_detach() syzbot reported use-after-free in tun_detach() [1]. This causes call trace like below: <pre>===== BUG: KASAN: use-after-free in notifier_call_chain+0x1ee/0x200 kernel/notifier.c:75 Read of size 8 at addr ffff88807324e2a8 by task syz-executor.0/3673 CPU: 0 PID: 3673 Comm: syz-executor.0 Not tainted 6.1.0-rc5-syzkaller-00044-gcc675d22e422 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xd1/0x138 lib/ dump_stack.c:106 print_address_description mm/kasan/report.c:284 [inline] print_report+0x15e/0x461 mm/kasan/report.c:395 kasan_report+0xbf/0x1f0 mm/kasan/report.c:495 notifier_call_chain +0x1ee/0x200 kernel/notifier.c:75 call_netdevice_notifiers_info+0x86/0x130 net/core/dev.c:1942 call_netdevice_notifiers_extack net/core/dev.c:1983 [inline] call_netdevice_notifiers net/core/dev.c:1997 [inline] netdev_wait_allrefs_any net/core/dev.c:10237 [inline] netdev_run_todo+0xabc6/0x1100 net/ core/dev.c:10351 tun_detach drivers/net/tun.c:704 [inline] tun_chr_close+0xe4/0x190 drivers/net/ tun.c:3467 __fput+0x27c/0xa90 fs/file_table.c:320 task_work_run+0x16f/0x270 kernel/task_work.c:179 exit_task_work include/linux/task_work.h:38 [inline] do_exit+0xb3d/0x2a30 kernel/exit.c:820 do_group_exit+0xd4/0x2a0 kernel/exit.c:950 get_signal+0x21b1/0x2440 kernel/signal.c:2858 arch_do_signal_or_restart+0x86/0x2300 arch/x86/kernel/signal.c:869 exit_to_user_mode_loop kernel/ entry/common.c:168 [inline] exit_to_user_mode_prepare+0x15f/0x250 kernel/entry/common.c:203 __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline] syscall_exit_to_user_mode +0x1d/0x50 kernel/entry/common.c:296 do_syscall_64+0x46/0xb0 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x63/0xcd The cause of the issue is that sock_put() from __tun_detach() drops last reference count for struct net, and then notifier_call_chain() from netdev_state_change() accesses that struct net. This patch fixes the issue by calling sock_put() from tun_detach() after all necessary accesses for the struct net has done.</pre>
CVE-2022-49017	In the Linux kernel, the following vulnerability has been resolved: tipc: re-fetch skb cb after tipc_msg_validate As the call trace shows, the original skb was freed in tipc_msg_validate(), and dereferencing the old skb cb would cause an use-after-free crash. BUG: KASAN: use-after-free in tipc_crypto_rcv_complete+0x1835/0x2240 [tipc] Call Trace: <IRQ> tipc_crypto_rcv_complete+0x1835/0x2240 [tipc] tipc_crypto_rcv+0xd32/0x1ec0 [tipc] tipc_rcv+0x744/0x1150 [tipc] ... Allocated by task 47078: kmem_cache_alloc_node+0x158/0x4d0 __alloc_skb+0x1c1/0x270 tipc_buf_acquire+0x1e/0xe0 [tipc] tipc_msg_create+0x33/0x1c0 [tipc] tipc_link_build_proto_msg+0x38a/0x2100 [tipc] tipc_link_timeout+0x8b8/0xef0 [tipc] tipc_node_timeout+0x2a1/0x960 [tipc] call_timer_fn+0x2d/0x1c0 ... Freed by task 47078: tipc_msg_validate+0x7b/0x440 [tipc] tipc_crypto_rcv_complete+0x4b5/0x2240 [tipc] tipc_crypto_rcv+0xd32/0x1ec0 [tipc] tipc_rcv+0x744/0x1150 [tipc] This patch fixes it by re-fetching the skb cb from the new allocated skb after calling tipc_msg_validate().
CVE-2022-49021	In the Linux kernel, the following vulnerability has been resolved: net: phy: fix null-ptr-deref while probe() failed I got a null-ptr-deref report as following when doing fault injection test: BUG: kernel NULL pointer dereference, address: 0000000000000058 Oops: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 1 PID: 253 Comm: 507-spi-dm9051 Tainted: G B N 6.1.0-rc3+ Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 RIP: 0010:klist_put+0x2d/0xd0 Call Trace: <TASK> klist_remove+0xf1/0x1c0 device_release_driver_internal+0x23e/0x2d0 bus_remove_device+0x1bd/0x240 device_del+0x357/0x770 phy_device_remove+0x11/0x30 mdiodbus_unregister+0xa5/0x140 release_nodes+0x6a/0xa0 devres_release_all+0xf8/0x150 device_unbind_cleanup+0x19/0xd0 //probe path: phy_device_register() device_add() phy_connect_phy_attach_direct() //set device driver probe() //it's failed, driver is not bound device_bind_driver() // probe failed, it's not called //remove path: phy_device_remove() device_del() device_release_driver_internal() __device_release_driver() //dev->drv is not NULL klist_remove() <- knode_driver is not added yet, cause null-ptr-deref In phy_attach_direct(), after setting the 'dev->driver', probe() fails, device_bind_driver() is not called, so the knode_driver->n_klist is not set, then it causes null-ptr-deref in __device_release_driver() while deleting device. Fix this by setting dev->driver to NULL in the error path in phy_attach_direct().
CVE-2022-49022	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix possible oob access in ieee80211_get_rate_duration Fix possible out-of-bound access in ieee80211_get_rate_duration routine as reported by the following UBSAN report: UBSAN: array-index-out-of-bounds in net/mac80211/airtime.c:455:47 index 15 is out of range for type 'u16 [12]' CPU: 2 PID: 217 Comm: kworker/u32:10 Not tainted 6.1.0-060100rc3-generic Hardware name: Acer Aspire TC-281/Aspire TC-281, BIOS R01-A2 07/18/2017 Workqueue: mt76 mt76u_tx_status_data [mt76_usb] Call Trace: <TASK> show_stack+0x4e/0x61 dump_stack_lvl+0x4a/0x6f dump_stack+0x10/0x18 ubsan_epilogue+0x9/0x43 __ubsan_handle_out_of_bounds.cold+0x42/0x47 ieee80211_get_rate_duration.constprop.0+0x22f/0x2a0 [mac80211] ? ieee80211_tx_status_ext+0x32e/0x640 [mac80211] ieee80211_calc_rx_airtime+0xda/0x120 [mac80211] ieee80211_calc_tx_airtime+0xb4/0x100 [mac80211] mt76x02_send_tx_status+0x266/0x480 [mt76x02_lib] mt76x02_tx_status_data+0x52/0x80 [mt76x02_lib] mt76u_tx_status_data+0x67/0xd0 [mt76_usb] process_one_work+0x225/0x400 worker_thread+0x50/0x3e0 ? process_one_work+0x400/0x400 kthread+0xe9/0x110 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30

CVE ID	Description
CVE-2022-49027	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix error handling in iavf_init_module() The iavf_init_module() won't destroy workqueue when pci_register_driver() failed. Call destroy_workqueue() when pci_register_driver() failed to prevent the resource leak. Similar to the handling of u132_hcd_init in commit f276e002793c ("usb: u132-hcd: fix resource leak")
CVE-2022-49028	In the Linux kernel, the following vulnerability has been resolved: ixgbev: Fix resource leak in ixgbev_init_module() ixgbev_init_module() won't destroy the workqueue created by create_singlethread_workqueue() when pci_register_driver() failed. Add destroy_workqueue() in fail path to prevent the resource leak. Similar to the handling of u132_hcd_init in commit f276e002793c ("usb: u132-hcd: fix resource leak")
CVE-2022-49029	In the Linux kernel, the following vulnerability has been resolved: hwmon: (ibmpex) Fix possible UAF when ibmpex_register_bmc() fails Smatch report warning as follows: drivers/hwmon/ibmpex.c:509 ibmpex_register_bmc() warn: '&data->list' not removed from list If ibmpex_find_sensors() fails in ibmpex_register_bmc(), data will be freed, but data->list will not be removed from driver_data.bmc_data, then list traversal may cause UAF. Fix by removing it from driver_data.bmc_data before free().
CVE-2022-49034	In the Linux kernel, the following vulnerability has been resolved: sh: cpuinfo: Fix a warning for CONFIG_CPUMASK_OFFSTACK When CONFIG_CPUMASK_OFFSTACK and CONFIG_DEBUG_PER_CPU_MAPS are selected, cpu_max_bits_warn() generates a runtime warning similar as below when showing /proc/cpuinfo. Fix this by using nr_cpu_ids (the runtime limit) instead of NR_CPUS to iterate CPUs. [3.052463] -----[cut here]----- [3.059679] WARNING: CPU: 3 PID: 1 at include/linux/cpumask.h:108 show_cpuinfo+0x5e8/0x5f0 [3.070072] Modules linked in: efivarfs autofs4 [3.076257] CPU: 0 PID: 1 Comm: systemd Not tainted 5.19-rc5+ #1052 [3.099465] Stack : 9000000100157b08 9000000000f18530 9000000000cf846c 9000000100154000 [3.109127] 9000000100157a50 0000000000000000 9000000100157a58 90000000000ef7430 [3.118774] 90000001001578e8 0000000000000040 0000000000000020 ffffffff [3.128412] 0000000000000000 1ab25f00ec96a37 900000010021de80 900000000101c890 [3.138056] 0000000000000000 0000000000000000 0000000000000000 0000000000000000 [3.147711] ffff8000339dc220 0000000000000001 0000000000000000 0000000000000000 [3.157364] 900000000101c998 0000000000000004 90000000000ef7430 0000000000000000 [3.167012] 0000000000000009 0000000000000006c 0000000000000000 0000000000000000 [3.176641] 9000000000d3de08 9000000001639390 90000000002086d8 00007ffff0080286 [3.186260] 00000000000000b0 0000000000000004 0000000000000000 0000000000071c1c [3.195868] ... [3.199917] Call Trace: [3.203941] [<90000000002086d8>] show_stack+0x38/0x14c [3.210666] [<9000000000cf846c>] dump_stack_lvl+0x60/0x88 [3.217625] [<900000000023d268>] __warn+0xd0/0x100 [3.223958] [<9000000000cf3c90>] warn_slowpath_fmt+0x7c/0xccc [3.231150] [<9000000000210220>] show_cpuinfo+0x5e8/0x5f0 [3.238080] [<90000000004f578c>] seq_read_iter+0x354/0x4b4 [3.245098] [<90000000004c2e90>] new_sync_read+0x17c/0x1c4 [3.252114] [<90000000004c5174>] vfs_read+0x138/0x1d0 [3.258694] [<90000000004c55f8>] ksys_read+0x70/0x100 [3.265265] [<9000000000cfde9c>] do_syscall+0x7c/0x94 [3.271820] [<9000000000202fe4>] handle_syscall+0xc4/0x160 [3.281824] ---[end trace 8b484262b4b8c24c]---
CVE-2022-49044	In the Linux kernel, the following vulnerability has been resolved: dm integrity: fix memory corruption when tag_size is less than digest size It is possible to set up dm-integrity in such a way that the "tag_size" parameter is less than the actual digest size. In this situation, a part of the digest beyond tag_size is ignored. In this case, dm-integrity would write beyond the end of the ic->recalc_tags array and corrupt memory. The corruption happened in integrity_recalc->integrity_sector_checksum->crypto_shash_final. Fix this corruption by increasing the tags array so that it has enough padding at the end to accommodate the loop in integrity_recalc() being able to write a full digest size for the last member of the tags array.
CVE-2022-49048	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix panic when forwarding a pkt with no in6 dev kongweibin reported a kernel panic in ip6_forward() when input interface has no in6 dev associated. The following tc commands were used to reproduce this panic: tc qdisc del dev vxlan100 root tc qdisc add dev vxlan100 root netem corrupt 5%

CVE ID	Description
CVE-2022-49052	In the Linux kernel, the following vulnerability has been resolved: mm: fix unexpected zeroed page mapping with zram swap Two processes under CLONE_VM cloning, user process can be corrupted by seeing zeroed page unexpectedly. CPU A CPU B do_swap_page do_swap_page SWP_SYNCHRONOUS_IO path SWP_SYNCHRONOUS_IO path swap_readpage valid data swap_slot_free_notify delete zram entry swap_readpage zeroed(invalid) data pte_lock map the *zero data* to userspace pte_unlock pte_lock if (!pte_same) goto out_nomap; pte_unlock return and next refault will read zeroed data The swap_slot_free_notify is bogus for CLONE_VM case since it doesn't increase the refcount of swap slot at copy_mm so it couldn't catch up whether it's safe or not to discard data from backing device. In the case, only the lock it could rely on to synchronize swap slot freeing is page table lock. Thus, this patch gets rid of the swap_slot_free_notify function. With this patch, CPU A will see correct data. CPU A CPU B do_swap_page do_swap_page SWP_SYNCHRONOUS_IO path SWP_SYNCHRONOUS_IO path swap_readpage original data pte_lock map the original data swap_free swap_range_free bd_disk->fops->swap_slot_free_notify swap_readpage read zeroed data pte_unlock pte_lock if (!pte_same) goto out_nomap; pte_unlock return on next refault will see mapped data by CPU B The concern of the patch would increase memory consumption since it could keep wasted memory with compressed form in zram as well as uncompressed form in address space. However, most of cases of zram uses no readahead and do_swap_page is followed by swap_free so it will free the compressed form from in zram quickly.
CVE-2022-49053	In the Linux kernel, the following vulnerability has been resolved: scsi: target: tcmu: Fix possible page UAF tcmu_try_get_data_page() looks up pages under cmdr_lock, but it does not take refcount properly and just returns page pointer. When tcmu_try_get_data_page() returns, the returned page may have been freed by tcmu_blocks_release(). We need to get_page() under cmdr_lock to avoid concurrent tcmu_blocks_release().
CVE-2022-49060	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix NULL pointer dereference in smc_pnet_find_ib() dev_name() was called with dev.parent as argument but without to NULL-check it before. Solve this by checking the pointer before the call to dev_name()).
CVE-2022-49063	In the Linux kernel, the following vulnerability has been resolved: ice: arfs: fix use-after-free when freeing @rx_cpu_rmap The CI testing bots triggered the following splat: [718.203054] BUG: KASAN: use-after-free in free_irq_cpu_rmap+0x53/0x80 [718.206349] Read of size 4 at addr ffff8881bd127e00 by task sh/20834 [718.212852] CPU: 28 PID: 20834 Comm: sh Kdump: loaded Tainted: G S W IOE 5.17.0-rc8_nextqueue-devqueue-02643-g23f3121aca93 #1 [718.219695] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0012.070720200218 07/07/2020 [718.223418] Call Trace: [718.227139] [718.230783] dump_stack_lvl+0x33/0x42 [718.234431] print_address_description.constprop.9+0x21/0x170 [718.238177] ? free_irq_cpu_rmap+0x53/0x80 [718.241885] ? free_irq_cpu_rmap+0x53/0x80 [718.245539] kasan_report.cold.18+0x7f/0x11b [718.249197] ? free_irq_cpu_rmap+0x53/0x80 [718.252852] free_irq_cpu_rmap+0x53/0x80 [718.256471] ice_free_rx_rmap.part.11+0x37/0x50 [ice] [718.260174] ice_remove_arfs +0x5f/0x70 [ice] [718.263810] ice_rebuild_arfs+0x3b/0x70 [ice] [718.267419] ice_rebuild +0x39c/0xb60 [ice] [718.270974] ? asm_sysvec_apic_timer_interrupt+0x12/0x20 [718.274472] ? ice_init_phy_user_cfg+0x360/0x360 [ice] [718.278033] ? delay_tsc+0x4a/0xb0 [718.281513] ? preempt_count_sub+0x14/0xc0 [718.284984] ? delay_tsc+0x8f/0xb0 [718.288463] ice_do_reset +0x92/0xf0 [ice] [718.292014] ice_pci_err_resume+0x91/0xf0 [ice] [718.295561] pci_reset_function +0x53/0x80 <...> [718.393035] Allocated by task 690: [718.433497] Freed by task 20834: [718.495688] Last potentially related work creation: [718.568966] The buggy address belongs to the object at ffff8881bd127e00 which belongs to the cache kmalloc-96 of size 96 [718.574085] The buggy address is located 0 bytes inside of 96-byte region [ffff8881bd127e00, ffff8881bd127e60] [718.579265] The buggy address belongs to the page: [718.598905] Memory state around the buggy address: [718.601809] ffff8881bd127d00: fa fb fb fb fb fb fb fb fb fc fc fc [718.604796] ffff8881bd127d80: 00 00 00 00 00 00 00 00 00 fc fc fc fc [718.607794] >ffff8881bd127e00: fa fb fb fb fb fb fb fb fb fc fc fc [718.610811] ^ [718.613819] ffff8881bd127e80: 00 00 00 00 00 00 00 00 00 fc fc fc [718.617107] ffff8881bd127f00: fa fb fb fb fb fb fb fb fb fc fc fc This is due to that free_irq_cpu_rmap() is always being called *after* (devm_)free_irq() and thus it tries to work with IRQ desc's already freed. For example, on device reset the driver frees the rmap right before allocating a new one (the splat above). Make rmap creation and freeing function symmetrical with {request,free}_irq() calls i.e. do that on ifup/ifdown instead of device probe/remove/resume. These operations can be performed independently from the actual device aRFS configuration. Also, make sure ice_vsi_free_irq() clears IRQ affinity notifiers only when aRFS is disabled -- otherwise, CPU rmap sets and clears its own and they must not be touched manually.

CVE ID	Description
CVE-2022-49065	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix the svc_deferred_event trace class Fix a NULL deref crash that occurs when an svc_rqst is deferred while the sunrpc tracing subsystem is enabled. svc_revisit() sets dr->xprt to NULL, so it can't be relied upon in the tracepoint to provide the remote's address. Unfortunately we can't revert the "svc_deferred_class" hunk in commit ece200ddd54b ("sunrpc: Save remote presentation address in svc_xprt for trace events") because there is now a specific check of event format specifiers for unsafe dereferences. The warning that check emits is: event svc_defer_recv has unsafe dereference of argument 1 A "%pISpc" format specifier with a "struct sockaddr *" is indeed flagged by this check. Instead, take the brute-force approach used by the svcrdma_qp_error tracepoint. Convert the dr::addr field into a presentation address in the TP_fast_assign() arm of the trace event, and store that as a string. This fix can be backported to - stable kernels. In the meantime, commit c6ced22997ad ("tracing: Update print fmt check to handle new __get_sockaddr() macro") is now in v5.18, so this wonky fix can be replaced with __sockaddr() and friends properly during the v5.19 merge window.
CVE-2022-4907	Uninitialized Use in FFmpeg in Google Chrome prior to 108.0.5359.71 allowed a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page. (Chromium security severity: Medium)
CVE-2022-49070	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix unregistering of framebuffer without device OF framebuffers do not have an underlying device in the Linux device hierarchy. Do a regular unregister call instead of hot unplugging such a non-existing device. Fixes a NULL dereference. An example error message on ppc64le is shown below. BUG: Kernel NULL pointer dereference on read at 0x000000060 Faulting instruction address: 0xc00000000080dfa4 Oops: Kernel access of bad area, sig: 11 [#1] LE PAGE_SIZE=64K MMU=Hash SMP NR_CPUS=2048 NUMA pSeries [...] CPU: 2 PID: 139 Comm: systemd-udevd Not tainted 5.17.0-ae085d7f9365 #1 NIP: c00000000080dfa4 LR: c00000000080df9c CTR: c000000000797430 REGS: c000000004132fe0 TRAP: 0300 Not tainted (5.17.0-ae085d7f9365) MSR: 8000000002009033 <SF,VEC,EE,ME,IR,DR,RI,LE> CR: 28228282 XER: 20000000 CFAR: c00000000000c80c DAR: 0000000000000060 DSISR: 40000000 IRQMASK: 0 GPR00: c00000000080df9c c000000004133280 c00000000169d200 0000000000000029 GPR04: 00000000ffffefff c000000004132f90 c000000004132f88 0000000000000000 GPR08: c0000000015658f8 c0000000015cd200 c0000000014f57d0 00000000048228283 GPR12: 0000000000000000 c00000003ffe300 0000000020000000 0000000000000000 GPR16: 0000000000000000 0000000113fc4a40 0000000000000005 0000000113fcfb80 GPR20: 000001000f7283b0 0000000000000000 c000000000e4a588 c000000000e4a5b0 GPR24: 0000000000000001 000000000000a000 c008000000db0168 c0000000021f6ec0 GPR28: c0000000016d65a8 c000000004b36460 0000000000000000 c0000000016d64b0 NIP [c00000000080dfa4] do_remove_conflicting_framebuffers+0x184/0x1d0 [c000000004133280] [c00000000080df9c] do_remove_conflicting_framebuffers+0x17c/0x1d0 (unreliable) [c000000004133350] [c00000000080e4d0] remove_conflicting_framebuffers+0x60/0x150 [c0000000041333a0] [c00000000080e6f4] remove_conflicting_pci_framebuffers+0x134/0x1b0 [c000000004133450] [c008000000e70438] drm_aperture_remove_conflicting_pci_framebuffers+0x90/0x100 [drm] [c000000004133490] [c008000000da0ce4] bochs_pci_probe+0x6c/0xa64 [bochs] [...] [c000000004133db0] [c00000000002aaa0] system_call_exception+0x170/0x2d0 [c000000004133e10] [c00000000000c3cc] system_call_common+0xec/0x250 The bug [1] was introduced by commit 27599aacbaef ("fbdev: Hot-unplug firmware fb devices on forced removal"). Most firmware framebuffers have an underlying platform device, which can be hot-unplugged before loading the native graphics driver. OF framebuffers do not (yet) have that device. Fix the code by unregistering the framebuffer as before without a hot unplug. Tested with 5.17 on qemu ppc64le emulation.
CVE-2022-49072	In the Linux kernel, the following vulnerability has been resolved: gpio: Restrict usage of GPIO chip irq members before initialization GPIO chip irq members are exposed before they could be completely initialized and this leads to race conditions. One such issue was observed for the gc->irq.domain variable which was accessed through the I2C interface in gpiochip_to_irq() before it could be initialized by gpiochip_add_irqchip(). This resulted in Kernel NULL pointer dereference. Following are the logs for reference :- kernel: Call Trace: kernel: gpiod_to_irq+0x53/0x70 kernel: acpi_dev_gpio_irq_get_by+0x113/0x1f0 kernel: i2c_acpi_get_irq+0xc0/0xd0 kernel: i2c_device_probe+0x28a/0x2a0 kernel: really_probe+0xf2/0x460 kernel: RIP: 0010:gpiochip_to_irq+0x47/0xc0 To avoid such scenarios, restrict usage of GPIO chip irq members before they are completely initialized.
CVE-2022-49074	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Fix GICR_CTLR.RWP polling It turns out that our polling of RWP is totally wrong when checking for it in the redistributors, as we test the *distributor* bit index, whereas it is a different bit number in the RDs... Oopsie boo. This is embarrassing. Not only because it is wrong, but also because it took *8 years* to notice the blunder... Just fix the damn thing.

CVE ID	Description
CVE-2022-49076	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Fix use-after-free bug for mm struct Under certain conditions, such as MPI_Abort, the hfi1 cleanup code may represent the last reference held on the task mm. hfi1_mmu_rb_unregister() then drops the last reference and the mm is freed before the final use in hfi1_release_user_pages(). A new task may allocate the mm structure while it is still being used, resulting in problems. One manifestation is corruption of the mmap_sem counter leading to a hang in down_write(). Another is corruption of an mm struct that is in use by another task.
CVE-2022-49077	In the Linux kernel, the following vulnerability has been resolved: mremap.c: avoid pointless invalidate_range_start/end on mremap(old_size=0) If an mremap() syscall with old_size=0 ends up in move_page_tables(), it will call invalidate_range_start()/invalidate_range_end() unnecessarily, i.e. with an empty range. This causes a WARN in KVM's mmu_notifier. In the past, empty ranges have been diagnosed to be off-by-one bugs, hence the WARNING. Given the low (so far) number of unique reports, the benefits of detecting more buggy callers seem to outweigh the cost of having to fix cases such as this one, where userspace is doing something silly. In this particular case, an early return from move_page_tables() is enough to fix the issue.
CVE-2022-49078	In the Linux kernel, the following vulnerability has been resolved: lz4: fix LZ4_decompress_safe_partial read out of bound When partialDecoding, it is EOF if we've either filled the output buffer or can't proceed with reading an offset for following match. In some extreme corner cases when compressed data is suitably corrupted, UAF will occur. As reported by KASAN [1], LZ4_decompress_safe_partial may lead to read out of bound problem during decoding. lz4 upstream has fixed it [2] and this issue has been discussed here [3] before. current decompression routine was ported from lz4 v1.8.3, bumping lib/lz4 to v1.9.+ is certainly a huge work to be done later, so, we'd better fix it first. [1] https://lore.kernel.org/all/000000000000830d1205cf7f0477@google.com/ [2] https://github.com/lz4/lz4/commit/c5d6f8a8be3927c0bec91bcc58667a6cfad244ad# [3] https://lore.kernel.org/all/CC666AE8-4CA4-4951-B6FB-A2EFDE3AC03B@fb.com/
CVE-2022-49082	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix use after free in _scsih_expander_node_remove() The function mpt3sas_transport_port_remove() called in _scsih_expander_node_remove() frees the port field of the sas_expander structure, leading to the following use-after-free splat from KASAN when the ioc_info() call following that function is executed (e.g. when doing rmmmod of the driver module): [3479.371167] <pre> ===== [3479.378496] BUG: KASAN: use-after-free in _scsih_expander_node_remove+0x710/0x750 [mpt3sas] [3479.386936] Read of size 1 at addr ffff8881c037691c by task rmmmod/1531 [3479.393524] [3479.395035] CPU: 18 PID: 1531 Comm: rmmmod Not tainted 5.17.0-rc8+ #1436 [3479.401712] Hardware name: Supermicro Super Server/H12SSL-NT, BIOS 2.1 06/02/2021 [3479.409263] Call Trace: [3479.411743] <TASK> [3479.413875] dump_stack_lvl+0x45/0x59 [3479.417582] print_address_description.constprop.0+0x1f/0x120 [3479.423389] ? _scsih_expander_node_remove +0x710/0x750 [mpt3sas] [3479.429469] kasan_report.cold+0x83/0xdf [3479.433438] ? _scsi_h_expander_node_remove+0x710/0x750 [mpt3sas] [3479.439514] _scsi_h_expander_node_remove +0x710/0x750 [mpt3sas] [3479.445411] ? _raw_spin_unlock_irqrestore+0x2d/0x40 [3479.452032] scsih_remove+0x525/0xc90 [mpt3sas] [3479.458212] ? mpt3sas_expander_remove+0x1d0/0x1d0 [mpt3sas] [3479.465529] ? down_write+0xde/0x150 [3479.470746] ? up_write+0x14d/0x460 [3479.475840] ? kernfs_find_ns+0x137/0x310 [3479.481438] pci_device_remove+0x65/0x110 [3479.487013] __device_release_driver+0x316/0x680 [3479.493180] driver_detach+0x1ec/0x2d0 [3479.498499] bus_remove_driver+0xe7/0x2d0 [3479.504081] pci_unregister_driver+0x26/0x250 [3479.510033] _mpt3sas_exit+0x2b/0x6cf [mpt3sas] [3479.516144] __x64_sys_delete_module +0x2fd/0x510 [3479.522315] ? free_module+0xaa0/0xaa0 [3479.527593] ? __cond_resched +0x1c/0x90 [3479.532951] ? lockdep_hardirqs_on_prepare+0x273/0x3e0 [3479.539607] ? syscall_enter_from_user_mode+0x21/0x70 [3479.546161] ? trace_hardirqs_on+0x1c/0x110 [3479.551828] do_syscall_64+0x35/0x80 [3479.556884] entry_SYSCALL_64_after_hwframe +0x44/0xae [3479.563402] RIP: 0033:0x7f1fc482483b ... [3479.943087] ===== </pre> Fix this by introducing the local variable port_id to store the port ID value before executing mpt3sas_transport_port_remove(). This local variable is then used in the call to ioc_info() instead of dereferencing the freed port structure.
CVE-2022-49084	In the Linux kernel, the following vulnerability has been resolved: qede: confirm skb is allocated before using qede_build_skb() assumes build_skb() always works and goes straight to skb_reserve(). However, build_skb() can fail under memory pressure. This results in a kernel panic because the skb to reserve is NULL. Add a check in case build_skb() failed to allocate and return NULL. The NULL return is handled correctly in callers to qede_build_skb().

CVE ID	Description
CVE-2022-49089	In the Linux kernel, the following vulnerability has been resolved: IB/rdmavt: add lock to call to rvt_error_qp to prevent a race condition The documentation of the function rvt_error_qp says both r_lock and s_lock need to be held when calling that function. It also asserts using lockdep that both of those locks are held. However, the commit I referenced in Fixes accidentally makes the call to rvt_error_qp in rvt_ruc_loopback no longer covered by r_lock. This results in the lockdep assertion failing and also possibly in a race condition.
CVE-2022-49094	In the Linux kernel, the following vulnerability has been resolved: net/tls: fix slab-out-of-bounds bug in decrypt_internal The memory size of tls_ctx->rx.iv for AES128-CCM is 12 setting in tls_set_sw_offload(). The return value of crypto_aead_ivsize() for "ccm(aes)" is 16. So memcpy() require 16 bytes from 12 bytes memory space will trigger slab-out-of-bounds bug as following: ===== BUG: KASAN: slab-out-of-bounds in decrypt_internal+0x385/0xc40 [tls] Read of size 16 at addr ffff888114e84e60 by task tls/10911 Call Trace: <TASK> dump_stack_lvl+0x34/0x44 print_report.cold+0x5e/0x5db ? decrypt_internal+0x385/0xc40 [tls] kasan_report+0xab/0x120 ? decrypt_internal+0x385/0xc40 [tls] kasan_check_range+0xf9/0x1e0 memcpy+0x20/0x60 decrypt_internal+0x385/0xc40 [tls] ? tls_get_rec+0x2e0/0x2e0 [tls] ? process_rx_list+0x1a5/0x420 [tls] ? tls_setup_from_iter.constprop.0+0x2e0/0x2e0 [tls] decrypt_skb_update+0x9d/0x400 [tls] tls_sw_recvmsg+0x3c8/0xb50 [tls] Allocated by task 10911: kasan_save_stack+0x1e/0x40 __kasan_kmalloc+0x81/0xa0 tls_set_sw_offload+0x2eb/0xa20 [tls] tls_setsockopt+0x68c/0x700 [tls] __sys_setsockopt+0xfe/0x1b0 Replace the crypto_aead_ivsize() with prot->iv_size + prot->salt_size when memcpy() iv value in TLS_1_3_VERSION scenario.
CVE-2022-49096	In the Linux kernel, the following vulnerability has been resolved: net: sfc: add missing xdp queue reinitialization After rx/tx ring buffer size is changed, kernel panic occurs when it acts XDP_TX or XDP_REDIRECT. When tx/rx ring buffer size is changed(ethtool -G), sfc driver reallocates and reinitializes rx and tx queues and their buffer (tx_queue->buffer). But it misses reinitializing xdp queues(efx->xdp_tx_queues). So, while it is acting XDP_TX or XDP_REDIRECT, it uses the uninitialized tx_queue->buffer. A new function efx_set_xdp_channels() is separated from efx_set_channels() to handle only xdp queues. Splat looks like: BUG: kernel NULL pointer dereference, address: 000000000000002a #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 0 P4D 0 Oops: 0002 [#4] PREEMPT SMP NOPTI RIP: 0010:efx_tx_map_chunk+0x54/0x90 [sfc] CPU: 2 PID: 0 Comm: swapper/2 Tainted: G D 5.17.0+ #55 e8beeee8289528f11357029357cf Code: 48 8b 8d a8 01 00 48 8d 14 52 4c 8d 2c d0 44 89 e0 48 85 c9 74 0e 44 89 e2 4c 89 f6 48 80 RSP: 0018:ffff92f121e45c60 EFLAGS: 00010297 RIP: 0010:efx_tx_map_chunk+0x54/0x90 [sfc] RAX: 0000000000000040 RBX: ffff92ea506895c0 RCX: ffffffff0330870 RDX: 0000000000000001 RSI: 00000001139b10ce RDI: ffff92ea506895c0 RBP: ffffffff0358a80 R08: 00000001139b110d R09: 0000000000000000 R10: 0000000000000001 R11: ffff92ea414c0088 R12: 0000000000000040 R13: 0000000000000018 R14: 00000001139b10ce R15: ffff92ea506895c0 FS: 0000000000000000(0000) GS:ffff92f121e00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 Code: 48 8b 8d a8 01 00 00 48 8d 14 52 4c 8d 2c d0 44 89 e0 48 85 c9 74 0e 44 89 e2 4c 89 f6 48 80 CR2: 000000000000002a CR3: 00000003e6810004 CR4: 00000000007706e0 RSP: 0018:ffff92f121e85c60 EFLAGS: 00010297 PKRU: 55555554 RAX: 0000000000000040 RBX: ffff92ea50689700 RCX: ffffffff0330870 RDX: 0000000000000001 RSI: 00000001145a90ce RDI: ffff92ea50689700 RBP: ffffffff0358a80 R08: 00000001145a910d R09: 0000000000000000 R10: 0000000000000001 R11: ffff92ea414c0088 R12: 0000000000000040 R13: 0000000000000018 R14: 00000001145a90ce R15: ffff92ea50689700 FS: 0000000000000000(0000) GS:ffff92f121e80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000000002a CR3: 00000003e6810005 CR4: 00000000007706e0 PKRU: 55555554 Call Trace: <IRQ> efx_xdp_tx_buffers+0x12b/0x3d0 [sfc 84c94b8e32d44d296c17e10a634d3ad454de4ba5] __efx_rx_packet+0x5c3/0x930 [sfc 84c94b8e32d44d296c17e10a634d3ad454de4ba5] efx_rx_packet+0x28c/0x2e0 [sfc 84c94b8e32d44d296c17e10a634d3ad454de4ba5] efx_ef10_ev_process+0x5f8/0xf40 [sfc 84c94b8e32d44d296c17e10a634d3ad454de4ba5] ? enqueue_task_fair+0x95/0x550 efx_poll+0xc4/0x360 [sfc 84c94b8e32d44d296c17e10a634d3ad454de4ba5]
CVE-2022-49097	In the Linux kernel, the following vulnerability has been resolved: NFS: Avoid writeback threads getting stuck in mempool_alloc() In a low memory situation, allow the NFS writeback code to fail without getting stuck in infinite loops in mempool_alloc().
CVE-2022-49100	In the Linux kernel, the following vulnerability has been resolved: virtio_console: eliminate anonymous module_init & module_exit Eliminate anonymous module_init() and module_exit(), which can lead to confusion or ambiguity when reading System.map, crashes/oops/bugs, or an initcall_debug log. Give each of these init and exit functions unique driver-specific names to eliminate the anonymous names. Example 1: (System.map) ffffffff832fc78c t init ffffffff832fc79e t init ffffffff832fc8f8 t init Example 2: (initcall_debug log) calling init+0x0/0x12 @ 1 initcall init+0x0/0x12 returned 0 after 15 usecs calling init+0x0/0x60 @ 1 initcall init+0x0/0x60 returned 0 after 2 usecs calling init+0x0/0x9a @ 1 initcall init+0x0/0x9a returned 0 after 74 usecs

CVE ID	Description
CVE-2022-49103	In the Linux kernel, the following vulnerability has been resolved: NFSv4.2: fix reference count leaks in <code>_nfs42_proc_copy_notify()</code> [You don't often get email from xiongx18@fudan.edu.cn. Learn why this is important at http://aka.ms/LearnAboutSenderIdIdentification .] The reference counting issue happens in two error paths in the function <code>_nfs42_proc_copy_notify()</code> . In both error paths, the function simply returns the error code and forgets to balance the refcount of object <code>`ctx`</code> , bumped by <code>get_nfs_open_context()</code> earlier, which may cause refcount leaks. Fix it by balancing refcount of the <code>`ctx`</code> object before the function returns in both error paths.
CVE-2022-49107	In the Linux kernel, the following vulnerability has been resolved: ceph: fix memory leak in <code>ceph_readdir</code> when <code>note_last_dentry</code> returns error Reset the <code>last_readdir</code> at the same time, and add a comment explaining why we don't free <code>last_readdir</code> when <code>dir_emit</code> returns false.
CVE-2022-49109	In the Linux kernel, the following vulnerability has been resolved: ceph: fix inode reference leakage in <code>ceph_get_snapdir()</code> The <code>ceph_get_inode()</code> will search for or insert a new inode into the hash for the given vino, and return a reference to it. If new is non-NULL, its reference is consumed. We should release the reference when in error handling cases.
CVE-2022-49110	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: revisit gc autotuning as of commit 4608dfc07e1 ("netfilter: conntrack: collect all entries in one cycle") conntrack gc was changed to run every 2 minutes. On systems where conntrack hash table is set to large value, most evictions happen from gc worker rather than the packet path due to hash table distribution. This causes netlink event overflows when events are collected. This change collects average expiry of scanned entries and reschedules to the average remaining value, within 1 to 60 second interval. To avoid event overflows, reschedule after each bucket and add a limit for both run time and number of evictions per run. If more entries have to be evicted, reschedule and restart 1 jiffy into the future.
CVE-2022-49113	In the Linux kernel, the following vulnerability has been resolved: powerpc/secvar: fix refcount leak in <code>format_show()</code> Refcount leak will happen when <code>format_show</code> returns failure in multiple cases. Unified management of <code>of_node_put</code> can fix this problem.
CVE-2022-49116	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: use memset avoid memory leaks Use <code>memset</code> to initialize structs to prevent memory leaks in <code>l2cap_ecred_connect</code>
CVE-2022-49118	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Free irq vectors in order for v3 HW If the driver probe fails to request the channel IRQ or fatal IRQ, the driver will free the IRQ vectors before freeing the IRQs in <code>free_irq()</code> , and this will cause a kernel BUG like this: -----[cut here]----- kernel BUG at drivers/pci/msi.c:369! Internal error: Oops - BUG: 0 [#1] PREEMPT SMP Call trace: free_msi_irqs+0x118/0x13c pci_disable_msi+0x9c/0x120 pci_free_irq_vectors+0x24/0x3c hisi_sas_v3_probe+0x360/0x9d0 [hisi_sas_v3_hw] local_pci_probe+0x44/0xb0 work_for_cpu_fn+0x20/0x34 process_one_work+0x1d0/0x340 worker_thread+0x2e0/0x460 kthread+0x180/0x190 ret_from_fork+0x10/0x20 ---[end trace b88990335b610c11]--- So we use <code>devm_add_action()</code> to control the order in which we free the vectors.

CVE ID	Description
CVE-2022-49123	In the Linux kernel, the following vulnerability has been resolved: ath11k: Fix frames flush failure caused by deadlock We are seeing below warnings: kernel: [25393.301506] ath11k_pci 0000:01:00.0: failed to flush mgmt transmit queue 0 kernel: [25398.421509] ath11k_pci 0000:01:00.0: failed to flush mgmt transmit queue 0 kernel: [25398.421831] ath11k_pci 0000:01:00.0: dropping mgmt frame for vdev 0, is_started 0 this means ath11k fails to flush mgmt. frames because wmi_mgmt_tx_work has no chance to run in 5 seconds. By setting /proc/sys/kernel/hung_task_timeout_secs to 20 and increasing ATH11K_FLUSH_TIMEOUT to 50 we get below warnings: kernel: [120.763160] INFO: task wpa_supplicant:924 blocked for more than 20 seconds. kernel: [120.763169] Not tainted 5.10.90 #12 kernel: [120.763177] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. kernel: [120.763186] task:wpa_supplicant state:D stack: 0 pid: 924 ppid: 1 flags:0x000043a0 kernel: [120.763201] Call Trace: kernel: [120.763214] __schedule+0x785/0x12fa kernel: [120.763224] ? lockdep_hardirqs_on_prepare+0xe2/0x1bb kernel: [120.763242] schedule+0x7e/0xa1 kernel: [120.763253] schedule_timeout+0x98/0xfe kernel: [120.763266] ? run_local_timers+0x4a/0x4a kernel: [120.763291] ath11k_mac_flush_tx_complete+0x197/0x2b1 [ath11k 13c3a9bf37790f4ac8103b3decf7ab4008ac314a] kernel: [120.763306] ? init_wait_entry+0x2e/0x2e kernel: [120.763343] _ieee80211_flush_queues+0x167/0x21f [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763378] _ieee80211_recalc_idle+0x105/0x125 [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763411] _ieee80211_recalc_idle+0x14/0x27 [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763441] _ieee80211_free_chanctx+0x77/0xa2 [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763473] _ieee80211_vif_release_channel+0x100/0x131 [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763540] _ieee80211_vif_release_channel+0x66/0x81 [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763572] _ieee80211_destroy_auth_data+0xa3/0xe6 [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763612] _ieee80211_mgd_deauth+0x178/0x29b [mac80211 335da900954f1c5ea7f1613d92088ce83342042c] kernel: [120.763654] cfg80211_mlme_deauth+0x1a8/0x22c [cfg80211 8945aa5bc2af5f6972336665d8ad6f9c191ad5be] kernel: [120.763697] nl80211_deauthenticate+0xfa/0x123 [cfg80211 8945aa5bc2af5f6972336665d8ad6f9c191ad5be] kernel: [120.763715] genl_rcv_msg+0x392/0x3c2 kernel: [120.763750] ? nl80211_associate+0x432/0x432 [cfg80211 8945aa5bc2af5f6972336665d8ad6f9c191ad5be] kernel: [120.763782] ? nl80211_associate+0x432/0x432 [cfg80211 8945aa5bc2af5f6972336665d8ad6f9c191ad5be] kernel: [120.763802] ? genl_rcv+0x36/0x36 kernel: [120.763814] netlink_rcv_skb+0x89/0xf7 kernel: [120.763829] genl_rcv+0x28/0x36 kernel: [120.763840] netlink_unicast+0x179/0x24b kernel: [120.763854] netlink_sendmsg+0x393/0x401 kernel: [120.763872] sock_sendmsg+0x72/0x76 kernel: [120.763886] __sys_sendmsg+0x170/0x1e6 kernel: [120.763897] ? copy_msghdr_from_user+0x7a/0xa2 kernel: [120.763914] __sys_sendmsg+0x95/0xd1 kernel: [120.763940] __sys_sendmsg+0x85/0xbf kernel: [120.763956] do_syscall_64+0x43/0x55 kernel: [120.763966] entry_SYSCALL_64_after_hwframe+0x44/0xa9 kernel: [120.763977] RIP: 0033:0x79089f3fcc83 kernel: [120.763986] RSP: 002b:00007ffe604f0508 EFLAGS: 00000246 ORIG_RAX: 000000000000002e kernel: [120.763997] RAX: ffffffffefda RBX: 000059b40e987690 RCX: 000079089f3fcc83 kernel: [120.764006] RDX: 0000000000000000 RSI: 00007ffe604f0558 RDI: 0000000000000009 kernel: [120.764014] RBP: 00007ffe604f0540 R08: 0000000000000004 R09: 0000000000400000 kernel: [120.764023] R10: 00007ffe604f0638 R11: 0000000000000246 R12: 000059b40ea04980 kernel: [120.764032] R13: 00007ffe604 ---truncated---
CVE-2022-49124	In the Linux kernel, the following vulnerability has been resolved: x86/mce: Work around an erratum on fast string copy instructions A rare kernel panic scenario can happen when the following conditions are met due to an erratum on fast string copy instructions: 1) An uncorrected error. 2) That error must be in first cache line of a page. 3) Kernel must execute page_copy from the page immediately before that page. The fast string copy instructions ("REP; MOVS*") could consume an uncorrectable memory error in the cache line _right after_ the desired region to copy and raise an MCE. Bit 0 of MSR_IA32_MISC_ENABLE can be cleared to disable fast string copy and will avoid such spurious machine checks. However, that is less preferable due to the permanent performance impact. Considering memory poison is rare, it's desirable to keep fast string copy enabled until an MCE is seen. Intel has confirmed the following: 1. The CPU erratum of fast string copy only applies to Skylake, Cascade Lake and Cooper Lake generations. Directly return from the MCE handler: 2. Will result in complete execution of the "REP; MOVS*" with no data loss or corruption. 3. Will not result in another MCE firing on the next poisoned cache line due to "REP; MOVS*". 4. Will resume execution from a correct point in code. 5. Will result in the same instruction that triggered the MCE firing a second MCE immediately for any other software recoverable data fetch errors. 6. Is not safe without disabling the fast string copy, as the next fast string copy of the same buffer on the same CPU would result in a PANIC MCE. This should mitigate the erratum completely with the only caveat that the fast string copy is disabled on the affected hyper thread thus performance degradation. This is still better than the OS crashing on MCEs raised on an irrelevant process due to "REP; MOVS*" accesses in a kernel context, e.g., copy_page. Injected errors on 1st cache line of 8 anonymous pages of process 'proc1' and observed MCE consumption from 'proc2' with no panic (directly returned). Without the fix, the host panicked within a few minutes on a random 'proc2' process due to kernel access from copy_page. [bp: Fix comment style + touch ups, zap an unlikely(), improve the quirk function's readability.]

CVE ID	Description
CVE-2022-49126	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix memory leaks Fix memory leaks related to operational reply queue's memory segments which are not getting freed while unloading the driver.
CVE-2022-49129	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921: fix crash when startup fails. If the nic fails to start, it is possible that the reset_work has already been scheduled. Ensure the work item is canceled so we do not have use-after-free crash in case cleanup is called before the work item is executed. This fixes crash on my x86_64 apu2 when mt7921k radio fails to work. Radio still fails, but OS does not crash.
CVE-2022-49130	In the Linux kernel, the following vulnerability has been resolved: ath11k: mhi: use mhi_sync_power_up() If amss.bin was missing ath11k would crash during 'rmmod ath11k_pci'. The reason for that was that we were using mhi_async_power_up() which does not check any errors. But mhi_sync_power_up() on the other hand does check for errors so let's use that to fix the crash. I was not able to find a reason why an async version was used. ath11k_mhi_start() (which enables state ATH11K_MHI_POWER_ON) is called from ath11k_hif_power_up(), which can sleep. So sync version should be safe to use here. [145.569731] general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP DEBUG_PAGEALLOC KASAN PTI [145.569789] KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] [145.569843] CPU: 2 PID: 1628 Comm: rmmod Kdump: loaded Tainted: G W 5.16.0-wt-ath+ #567 [145.569898] Hardware name: Intel(R) Client Systems NUC8i7HVK/NUC8i7HVB, BIOS HNKBLi70.86A.0067.2021.0528.1339 05/28/2021 [145.569956] RIP: 0010:ath11k_hal_srng_access_begin+0xb5/0x2b0 [ath11k] [145.570028] Code: df 48 89 fa 48 c1 ea 03 80 3c 02 00 0f 85 ec 01 00 00 48 8b ab a8 00 00 00 48 b8 00 00 00 00 00 00 fc ff df 48 89 ea 48 c1 ea 03 <0f> b6 14 02 48 89 e8 83 e0 07 83 c0 03 45 85 ed 75 48 38 d0 7c 08 [145.570089] RSP: 0018:ffff9000025d7ac0 EFLAGS: 00010246 [145.570144] RAX: dffffc0000000000 RBX: ffff88814fca2dd8 RCX: 1fffffff50cb455 [145.570196] RDX: 0000000000000000 RSI: ffff88814fca2dd8 RDI: ffff88814fca2e80 [145.570252] RBP: 0000000000000000 R08: 0000000000000000 R09: ffffffff8659497 [145.570329] R10: fffffbfff50cb292 R11: 0000000000000001 R12: ffff88814fca0000 [145.570410] R13: 0000000000000000 R14: ffff88814fca2798 R15: ffff88814fca2dd8 [145.570465] FS: 00007fa399988540(0000) GS:ffff888233e00000(0000) knlGS:0000000000000000 [145.570519] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [145.570571] CR2: 00007fa399b51421 CR3: 0000000137898002 CR4: 00000000003706e0 [145.570623] Call Trace: [145.570675] <TASK> [145.570727] ? ath11k_ce_tx_process_cb+0x34b/0x860 [ath11k] [145.570797] ath11k_ce_tx_process_cb+0x356/0x860 [ath11k] [145.570864] ? tasklet_init+0x150/0x150 [145.570919] ? ath11k_ce_alloc_pipes+0x280/0x280 [ath11k] [145.570986] ? tasklet_clear_sched+0x42/0xe0 [145.571042] ? tasklet_kill+0xe9/0x1b0 [145.571095] ? tasklet_clear_sched+0xe0/0xe0 [145.571148] ? irq_has_action+0x120/0x120 [145.571202] ath11k_ce_cleanup_pipes+0x45a/0x580 [ath11k] [145.571270] ? ath11k_pci_stop+0x10e/0x170 [ath11k_pci] [145.571345] ath11k_core_stop+0x8a/0xc0 [ath11k] [145.571434] ath11k_core_deinit+0x9e/0x150 [ath11k] [145.571499] ath11k_pci_remove+0xd2/0x260 [ath11k_pci] [145.571553] pci_device_remove+0x9a/0x1c0 [145.571605] __device_release_driver+0x332/0x660 [145.571659] driver_detach+0x1e7/0x2c0 [145.571712] bus_remove_driver+0xe2/0x2d0 [145.571772] pci_unregister_driver+0x21/0x250 [145.571826] __do_sys_delete_module+0x30a/0x4b0 [145.571879] ? free_module+0xac0/0xac0 [145.571933] ? lockdep_hardirqs_on_prepare.part.0+0x18c/0x370 [145.571986] ? syscall_enter_from_user_mode+0x1d/0x50 [145.572039] ? lockdep_hardirqs_on+0x79/0x100 [145.572097] do_syscall_64+0x3b/0x90 [145.572153] entry_SYSCALL_64_after_hwframe+0x44/0xae Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03003-QCAHSPSWPL_V1_V2_SILICONZ_LITE-2

CVE ID	Description
CVE-2022-49132	In the Linux kernel, the following vulnerability has been resolved: ath11k: pci: fix crash on suspend if board file is not found Mario reported that the kernel was crashing on suspend if ath11k was not able to find a board file: [473.693286] PM: Suspending system (s2idle) [473.693291] printk: Suspending console(s) (use no_console_suspend to debug) [474.407787] BUG: unable to handle page fault for address: 0000000000002070 [474.407791] #PF: supervisor read access in kernel mode [474.407794] #PF: error_code(0x0000) - not-present page [474.407798] PGD 0 P4D 0 [474.407801] Oops: 0000 [#1] PREEMPT SMP NOPTI [474.407805] CPU: 2 PID: 2350 Comm: kworker/u32:14 Tainted: G W 5.16.0 #248 [...] [474.407868] Call Trace: [474.407870] <TASK> [474.407874] ? _raw_spin_lock_irqsave+0x2a/0x60 [474.407882] ? lock_timer_base+0x72/0xa0 [474.407889] ? _raw_spin_unlock_irqrestore+0x29/0x3d [474.407892] ? try_to_del_timer_sync+0x54/0x80 [474.407896] ath11k_dp_rx_pktlog_stop+0x49/0xc0 [ath11k] [474.407912] ath11k_core_suspend+0x34/0x130 [ath11k] [474.407923] ath11k_pci_pm_suspend+0x1b/0x50 [ath11k_pci] [474.407928] pci_pm_suspend+0x7e/0x170 [474.407935] ? pci_pm_freeze+0xc0/0xc0 [474.407939] dpm_run_callback+0x4e/0x150 [474.407947] __device_suspend+0x148/0x4c0 [474.407951] async_suspend+0x20/0x90 dmesg-efi-164255130401001: Oops#1 Part1 [474.407955] async_run_entry_fn+0x33/0x120 [474.407959] process_one_work+0x220/0x3f0 [474.407966] worker_thread+0x4a/0x3d0 [474.407971] kthread+0x17a/0x1a0 [474.407975] ? process_one_work+0x3f0/0x3f0 [474.407979] ? set_kthread_struct+0x40/0x40 [474.407983] ret_from_fork+0x22/0x30 [474.407991] </TASK> The issue here is that board file loading happens after ath11k_pci_probe() successfully returns (ath11k initialisation happens asynchronously) and the suspend handler is still enabled, of course failing as ath11k is not properly initialised. Fix this by checking ATH11K_FLAG_QMI_FAIL during both suspend and resume. Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03003-QCAHSPSWPL_V1_V2_SILICONZ_LITE-2
CVE-2022-49134	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum: Guard against invalid local ports When processing events generated by the device's firmware, the driver protects itself from events reported for non-existent local ports, but not for the CPU port (local port 0), which exists, but does not have all the fields as any local port. This can result in a NULL pointer dereference when trying access 'struct mlxsw_sp_port' fields which are not initialized for CPU port. Commit 63b08b1f6834 ("mlxsw: spectrum: Protect driver from buggy firmware") already handled such issue by bailing early when processing a PUDE event reported for the CPU port. Generalize the approach by moving the check to a common function and making use of it in all relevant places.
CVE-2022-49138	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: Ignore multiple conn complete events When one of the three connection complete events is received multiple times for the same handle, the device is registered multiple times which leads to memory corruptions. Therefore, consequent events for a single connection are ignored. The conn->state can hold different values, therefore HCI_CONN_HANDLE_UNSET is introduced to identify new connections. To make sure the events do not contain this or another invalid handle HCI_CONN_HANDLE_MAX and checks are introduced. Buglink: https://bugzilla.kernel.org/show_bug.cgi?id=215497
CVE-2022-49139	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix null ptr deref on hci_sync_conn_complete_evt This event is just specified for SCO and eSCO link types. On the reception of a HCI_Synchronous_Connection_Complete for a BDADDR of an existing LE connection, LE link type and a status that triggers the second case of the packet processing a NULL pointer dereference happens, as conn->link is NULL.
CVE-2022-49154	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: fix panic on out-of-bounds guest IRQ As guest_irq is coming from KVM_IRQFD API call, it may trigger crash in svm_update_pi_irt(e) due to out-of-bounds: crash> bt PID: 22218 TASK: ffff951a6ad74980 CPU: 73 COMMAND: "vcpu8" #0 [ffffb1ba6707fa40] machine_kexec at ffffffff8565b397 #1 [ffffb1ba6707fa90] __crash_kexec at ffffffff85788a6d #2 [ffffb1ba6707fb58] crash_kexec at ffffffff8578995d #3 [ffffb1ba6707fb70] oops_end at ffffffff85623c0d #4 [ffffb1ba6707fb90] no_context at ffffffff856692c9 #5 [ffffb1ba6707fbf8] exc_page_fault at ffffffff85f95b51 #6 [ffffb1ba6707fc50] asm_exc_page_fault at ffffffff86000ace [exception RIP: svm_update_pi_irt+227] RIP: ffffffff8c0761b53 RSP: fffffb1ba6707fd08 RFLAGS: 00010086 RAX: fffffb1ba6707fd78 RBX: fffffb1ba66d91000 RCX: 0000000000000001 RDX: 00003c803f63f1c0 RSI: 0000000000000019a RDI: fffffb1ba66db2ab8 RBP: 0000000000000019a R8: 0000000000000040 R9: fffff94ca41b82200 R10: ffffffff8c0761b53 R11: 0000000000000001 R12: 0000000000000001 R13: 0000000000000001 R14: ffffffff8c0761b53 R15: 000000000000000f ORIG_RAX: ffffffff8c0761b53 CS: 0010 SS: 0018 #7 [ffffb1ba6707fdb8] kvm_irq_routing_update at ffffffff8c09f19a1 [kvm] #8 [ffffb1ba6707fde0] kvm_set_irq_routing at ffffffff8c09f2133 [kvm] #9 [ffffb1ba6707fe18] kvm_vm_ioctl at ffffffff8c09ef544 [kvm] RIP: 00007f143c36488b RSP: 00007f143a4e04b8 RFLAGS: 00000246 RAX: ffffffff8c0761b53 RBX: 00007f05780041d0 RCX: 00007f143c36488b RDX: 00007f05780041d0 RSI: 0000000004008ae6a RDI: 0000000000000020 RBP: 000000000000004e R8: 0000000000000008 R9: 00007f05780041e0 R10: 00007f0578004560 R11: 0000000000000246 R12: 000000000000004e R13: 0000000000000001a R14: 00007f1424001c60 R15: 00007f0578003bc0 ORIG_RAX: 0000000000000010 CS: 0033 SS: 002b Vmx have been fix this in commit 3a8b0677fc61 (KVM: VMX: Do not BUG() on out-of-bounds guest IRQ), so we can just copy source from that to fix this.

CVE ID	Description
CVE-2022-49155	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Suppress a kernel complaint in qla_create_qpair() [12.323788] BUG: using smp_processor_id() in preemptible [00000000] code: systemd-udevd/1020 [12.332297] caller is qla2xxx_create_qpair+0x32a/0x5d0 [qla2xxx] [12.338417] CPU: 7 PID: 1020 Comm: systemd-udevd Tainted: G I ----- 5.14.0-29.el9.x86_64 #1 [12.348827] Hardware name: Dell Inc. PowerEdge R610/0F0XJ6, BIOS 6.6.0 05/22/2018 [12.356356] Call Trace: [12.358821] dump_stack_lvl+0x34/0x44 [12.362514] check_preemption_disabled+0xd9/0xe0 [12.367164] qla2xxx_create_qpair+0x32a/0x5d0 [qla2xxx] [12.372481] qla2x00_probe_one+0xa3a/0x1b80 [qla2xxx] [12.377617] ? _raw_spin_lock_irqsave+0x19/0x40 [12.384284] local_pci_probe+0x42/0x80 [12.390162] ? pci_match_device+0xd7/0x110 [12.396366] pci_device_probe+0xfd/0x1b0 [12.402372] really_probe+0x1e7/0x3e0 [12.408114] __driver_probe_device+0xfe/0x180 [12.414544] driver_probe_device+0x1e/0x90 [12.420685] __driver_attach+0xc0/0x1c0 [12.426536] ? __device_attach_driver+0xe0/0xe0 [12.433061] ? __device_attach_driver+0xe0/0xe0 [12.439538] bus_for_each_dev+0x78/0xc0 [12.445294] bus_add_driver+0x12b/0x1e0 [12.451021] driver_register+0x8f/0xe0 [12.456631] ? 0xfffffff0c07bc000 [12.461773] qla2x00_module_init+0x1be/0x229 [qla2xxx] [12.468776] do_one_initcall+0x44/0x200 [12.474401] ? load_module+0xad3/0xba0 [12.479908] ? kmem_cache_alloc_trace+0x45/0x410 [12.486268] do_init_module+0x5c/0x280 [12.491730] __do_sys_init_module+0x12e/0x1b0 [12.497785] do_syscall_64+0x3b/0x90 [12.503029] entry_SYSCALL_64_after_hwframe+0x44/0xae [12.509764] RIP: 0033:0x7f554f73ab2e
CVE-2022-49157	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix premature hw access after PCI error After a recoverable PCI error has been detected and recovered, qla driver needs to check to see if the error condition still persist and/or wait for the OS to give the resume signal. Sep 8 22:26:03 localhost kernel: WARNING: CPU: 9 PID: 124606 at qla_tmpl.c:440 qla27xx_fwdt_entry_t266+0x55/0x60 [qla2xxx] Sep 8 22:26:03 localhost kernel: RIP: 0010:qla27xx_fwdt_entry_t266+0x55/0x60 [qla2xxx] Sep 8 22:26:03 localhost kernel: Call Trace: Sep 8 22:26:03 localhost kernel: ? qla27xx_walk_template+0xb1/0x1b0 [qla2xxx] Sep 8 22:26:03 localhost kernel: ? qla27xx_execute_fwdt_template+0x12a/0x160 [qla2xxx] Sep 8 22:26:03 localhost kernel: ? qla27xx_fwddump+0xa0/0x1c0 [qla2xxx] Sep 8 22:26:03 localhost kernel: ? qla2xxx_pci_mmio_enabled+0xfb/0x120 [qla2xxx] Sep 8 22:26:03 localhost kernel: ? report_mmio_enabled+0x44/0x80 Sep 8 22:26:03 localhost kernel: ? report_slot_reset+0x80/0x80 Sep 8 22:26:03 localhost kernel: ? pci_walk_bus+0x70/0x90 Sep 8 22:26:03 localhost kernel: ? aer_dev_correctable_show+0xc0/0xc0 Sep 8 22:26:03 localhost kernel: ? pcie_do_recovery+0x1bb/0x240 Sep 8 22:26:03 localhost kernel: ? aer_recover_work_func+0xaa/0xd0 Sep 8 22:26:03 localhost kernel: ? process_one_work+0x1a7/0x360 .. Sep 8 22:26:03 localhost kernel: qla2xxx [0000:42:00.2]-8041:22: detected PCI disconnect. Sep 8 22:26:03 localhost kernel: qla2xxx [0000:42:00.2]-107ff:22: qla27xx_fwdt_entry_t262: dump ram MB failed. Area 5h start 198013h end 198013h Sep 8 22:26:03 localhost kernel: qla2xxx [0000:42:00.2]-107ff:22: Unable to capture FW dump Sep 8 22:26:03 localhost kernel: qla2xxx [0000:42:00.2]-1015:22: cmd=0x0, waited 5221 msecs Sep 8 22:26:03 localhost kernel: qla2xxx [0000:42:00.2]-680d:22: mmio enabled returning. Sep 8 22:26:03 localhost kernel: qla2xxx [0000:42:00.2]-d04c:22: MBX Command timeout for cmd 0, iocntol=ffffff jiffies=10140f2e5 mb[0-3]=[0xffff 0xffff 0xffff 0xffff]
CVE-2022-49159	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Implement ref count for SRB The timeout handler and the done function are racing. When qla2x00_async_iocb_timeout() starts to run it can be preempted by the normal response path (via the firmware?). qla24xx_async_gpisc_sp_done() releases the SRB unconditionally. When scheduling back to qla2x00_async_iocb_timeout() qla24xx_async_abort_cmd() will access a freed sp->qpair pointer: qla2xxx [0000:83:00.0]-2871:0: Async-gpisc timeout - hdl=63d portid=234500 50:06:0e:80:08:77:b6:21. qla2xxx [0000:83:00.0]-2853:0: Async done-gpisc res 0, WWPN 50:06:0e:80:08:77:b6:21 qla2xxx [0000:83:00.0]-2854:0: Async-gpisc OUT WWPN 20:45:00:27:f8:75:33:00 speeds=2c00 speed=0400. qla2xxx [0000:83:00.0]-28d8:0: qla24xx_handle_gpisc_event 50:06:0e:80:08:77:b6:21 DS 7 LS 6 rc 0 login 1 1 rscn 1 0 lid 5 BUG: unable to handle kernel NULL pointer dereference at 0000000000000004 IP: qla24xx_async_abort_cmd+0x1b/0x1c0 [qla2xxx] Obvious solution to this is to introduce a reference counter. One reference is taken for the normal code path (the 'good' case) and one for the timeout path. As we always race between the normal good case and the timeout/abort handler we need to serialize it. Also we cannot assume any order between the handlers. Since this is slow path we can use proper synchronization via locks. When we are able to cancel a timer (del_timer returns 1) we know there can't be any error handling in progress because the timeout handler hasn't expired yet, thus we can safely decrement the refcounter by one. If we are not able to cancel the timer, we know an abort handler is running. We have to make sure we call sp->done() in the abort handlers before calling kref_put().

CVE ID	Description
CVE-2022-49171	In the Linux kernel, the following vulnerability has been resolved: ext4: don't BUG if someone dirty pages without asking ext4 first [un]pin_user_pages_remote is dirtying pages without properly warning the file system in advance. A related race was noted by Jan Kara in 2018[1]; however, more recently instead of it being a very hard-to-hit race, it could be reliably triggered by process_vm_writev(2) which was discovered by Syzbot[2]. This is technically a bug in mm/gup.c, but arguably ext4 is fragile in that if some other kernel subsystem dirty pages without properly notifying the file system using page_mkwrite(), ext4 will BUG, while other file systems will not BUG (although data will still be lost). So instead of crashing with a BUG, issue a warning (since there may be potential data loss) and just mark the page as clean to avoid unprivileged denial of service attacks until the problem can be properly fixed. More discussion and background can be found in the thread starting at [2]. [1] https://lore.kernel.org/linux-mm/20180103100430.GE4911@quack2.suse.cz [2] https://lore.kernel.org/Yg0m6ljcNmfaSokM@google.com
CVE-2022-49174	In the Linux kernel, the following vulnerability has been resolved: ext4: fix ext4_mb_mark_bb() with flex_bg with fast_commit In case of flex_bg feature (which is by default enabled), extents for any given inode might span across blocks from two different block group. ext4_mb_mark_bb() only reads the buffer_head of block bitmap once for the starting block group, but it fails to read it again when the extent length boundary overflows to another block group. Then in this below loop it accesses memory beyond the block group bitmap buffer_head and results into a data abort. for (i = 0; i < clen; i++) if (!mb_test_bit(blkoff + i, bitmap_bh->b_data) == !state) already++; This patch adds this functionality for checking block group boundary in ext4_mb_mark_bb() and update the buffer_head(bitmap_bh) for every different block group. w/o this patch, I was easily able to hit a data access abort using Power platform. <...> [74.327662] EXT4-fs error (device loop3): ext4_mb_generate_buddy:1141: group 11, block bitmap and bg descriptor inconsistent: 21248 vs 23294 free clusters [74.533214] EXT4-fs (loop3): shut down requested (2) [74.536705] Aborting journal on device loop3-8. [74.702705] BUG: Unable to handle kernel data access on read at 0xc00000005e980000 [74.703727] Faulting instruction address: 0xc000000007bffb8 cpu 0xd: Vector: 300 (Data Access) at [c000000015db7060] pc: c000000007bffb8: ext4_mb_mark_bb +0x198/0x5a0 lr: c000000007bfeec: ext4_mb_mark_bb+0xcc/0x5a0 sp: c000000015db7300 msr: 800000000280b033 dar: c00000005e980000 dsisr: 40000000 current = 0xc000000027af6880 paca = 0xc00000003ffd5200 irqmask: 0x03 irq_happened: 0x01 pid = 5167, comm = mount <...> enter ? for help [c000000015db7380] c000000000782708 ext4_ext_clear_bb+0x378/0x410 [c000000015db7400] c000000000813f14 ext4_fc_replay+0x1794/0x2000 [c000000015db7580] c000000000833f7c do_one_pass+0xe9c/0x12a0 [c000000015db7710] c000000000834504 jbd2_journal_recover +0x184/0x2d0 [c000000015db77c0] c000000000841398 jbd2_journal_load+0x188/0x4a0 [c000000015db7880] c000000000804de8 ext4_fill_super+0x2638/0x3e10 [c000000015db7a40] c0000000005f8404 get_tree_bdev+0x2b4/0x350 [c000000015db7ae0] c000000000ef058 ext4_get_tree +0x28/0x40 [c000000015db7b00] c0000000005f6344 vfs_get_tree+0x44/0x100 [c000000015db7b70] c00000000063c408 path_mount+0xdd8/0xe70 [c000000015db7c40] c00000000063c8f0 sys_mount +0x450/0x550 [c000000015db7d50] c000000000035770 system_call_exception+0x4a0/0x4e0 [c000000015db7e10] c0000000000c74c system_call_common+0xec/0x250

CVE ID	Description
CVE-2022-49176	In the Linux kernel, the following vulnerability has been resolved: bfq: fix use-after-free in bfq_dispatch_request KASAN reports a use-after-free report when doing normal scsi-mq test [69832.239032] <pre> ===== [69832.241810] BUG: KASAN: use-after-free in bfq_dispatch_request+0x1045/0x44b0 [69832.243267] Read of size 8 at addr ffff88802622ba88 by task kworker/3:1H/155 [69832.244656] [69832.245007] CPU: 3 PID: 155 Comm: kworker/3:1H Not tainted 5.10.0-10295-g576c6382529e #8 [69832.246626] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014 [69832.249069] Workqueue: kblockd blk_mq_run_work_fn [69832.250022] Call Trace: [69832.250541] dump_stack +0x9b/0xce [69832.251232] ? bfq_dispatch_request+0x1045/0x44b0 [69832.252243] print_address_description.constprop.6+0x3e/0x60 [69832.253381] ? __cpuidle_text_end +0x5/0x5 [69832.254211] ? vprintk_func+0x6b/0x120 [69832.254994] ? bfq_dispatch_request +0x1045/0x44b0 [69832.255952] ? bfq_dispatch_request+0x1045/0x44b0 [69832.256914] kasan_report.cold.9+0x22/0x3a [69832.257753] ? bfq_dispatch_request+0x1045/0x44b0 [69832.258755] check_memory_region+0x1c1/0x1e0 [69832.260248] bfq_dispatch_request +0x1045/0x44b0 [69832.261181] ? bfq_bfq_expire+0x2440/0x2440 [69832.262032] ? blk_mq_delay_run_hw_queues+0xf9/0x170 [69832.263022] __blk_mq_do_dispatch_sched +0x52f/0x830 [69832.264011] ? blk_mq_sched_request_inserted+0x100/0x100 [69832.265101] __blk_mq_sched_dispatch_requests+0x398/0x4f0 [69832.266206] ? blk_mq_do_dispatch_ctx+0x570/0x570 [69832.267147] ? __switch_to+0x5f4/0xee0 [69832.267898] blk_mq_sched_dispatch_requests+0xdf/0x140 [69832.268946] __blk_mq_run_hw_queue +0xc0/0x270 [69832.269840] blk_mq_run_work_fn+0x51/0x60 [69832.278170] process_one_work +0x6d4/0xfe0 [69832.278984] worker_thread+0x91/0xc80 [69832.279726] ? __kthread_parkme +0xb0/0x110 [69832.280554] ? process_one_work+0xfe0/0xfe0 [69832.281414] kthread +0x32d/0x3f0 [69832.282082] ? kthread_park+0x170/0x170 [69832.282849] ret_from_fork +0x1f/0x30 [69832.283573] [69832.283886] Allocated by task 7725: [69832.284599] kasan_save_stack +0x19/0x40 [69832.285385] __kasan_kmalloc.constprop.2+0xc1/0xd0 [69832.286350] kmem_cache_alloc_node+0x13f/0x460 [69832.287237] bfq_get_queue+0x3d4/0x1140 [69832.287993] bfq_get_bfq_handle_split+0x103/0x510 [69832.289015] bfq_init_rq+0x337/0x2d50 [69832.289749] bfq_insert_requests+0x304/0x4e10 [69832.290634] blk_mq_sched_insert_requests+0x13e/0x390 [69832.291629] blk_mq_flush_plug_list+0x4b4/0x760 [69832.292538] blk_flush_plug_list +0x2c5/0x480 [69832.293392] io_schedule_prepare+0xb2/0xd0 [69832.294209] io_schedule_timeout +0x13/0x80 [69832.295014] wait_for_common_io.constprop.1+0x13c/0x270 [69832.296137] submit_bio_wait+0x103/0x1a0 [69832.296932] blkdev_issue_discard+0xe6/0x160 [69832.297794] blk_ioctl_discard+0x219/0x290 [69832.298614] blkdev_common_ioctl+0x50a/0x1750 [69832.304715] blkdev_ioctl+0x470/0x600 [69832.305474] block_ioctl+0xde/0x120 [69832.306232] vfs_ioctl +0x6c/0xc0 [69832.306877] __se_sys_ioctl+0x90/0xa0 [69832.307629] do_syscall_64+0x2d/0x40 [69832.308362] entry_SYSCALL_64_after_hwframe+0x44/0xa9 [69832.309382] [69832.309701] Freed by task 155: [69832.310328] kasan_save_stack+0x19/0x40 [69832.311121] kasan_set_track +0x1c/0x30 [69832.311868] kasan_set_free_info+0x1b/0x30 [69832.312699] __kasan_slab_free +0x111/0x160 [69832.313524] kmem_cache_free+0x94/0x460 [69832.314367] bfq_put_queue +0x582/0x940 [69832.315112] __bfq_bfqd_reset_in_service+0x166/0x1d0 [69832.317275] bfq_bfq_expire+0xb27/0x2440 [69832.318084] bfq_dispatch_request+0x697/0x44b0 [69832.318991] __blk_mq_do_dispatch_sched+0x52f/0x830 [69832.319984] __blk_mq_sched_dispatch_requests +0x398/0x4f0 [69832.321087] blk_mq_sched_dispatch_requests+0xdf/0x140 [69832.322225] __blk_mq_run_hw_queue+0xc0/0x270 [69832.323114] blk_mq_run_work_fn+0x51/0x6 ---truncated---</pre>
CVE-2022-49177	In the Linux kernel, the following vulnerability has been resolved: hwrng: cavium - fix NULL but dereferenced coccicheck error Fix following coccicheck warning: ./drivers/char/hw_random/cavium-rng-vf.c:182:17-20: ERROR: pdev is NULL but dereferenced.
CVE-2022-49178	In the Linux kernel, the following vulnerability has been resolved: memstick/mspro_block: fix handling of read-only devices Use set_disk_ro to propagate the read-only state to the block layer instead of checking for it in ->open and leaking a reference in case of a read-only device.

CVE ID	Description
CVE-2022-49179	In the Linux kernel, the following vulnerability has been resolved: block, bfq: don't move oom_bfq Our test report a UAF: [2073.019181] ===== [2073.019188] BUG: KASAN: use-after-free in __bfq_put_async_bfqq+0xa0/0x168 [2073.019191] Write of size 8 at addr ffff8000ccf64128 by task rmmmod/72584 [2073.019192] [2073.019196] CPU: 0 PID: 72584 Comm: rmmmod Kdump: loaded Not tainted 4.19.90-yk #5 [2073.019198] Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 [2073.019200] Call trace: [2073.019203] dump_backtrace+0x0/0x310 [2073.019206] show_stack+0x28/0x38 [2073.019210] dump_stack +0xec/0x15c [2073.019216] print_address_description+0x68/0x2d0 [2073.019220] kasan_report +0x238/0x2f0 [2073.019224] __asan_store8+0x88/0xb0 [2073.019229] __bfq_put_async_bfqq +0xa0/0x168 [2073.019233] bfq_put_async_queues+0xbc/0x208 [2073.019236] bfq_pd_offline +0x178/0x238 [2073.019240] blkcg_deactivate_policy+0x1f0/0x420 [2073.019244] bfq_exit_queue +0x128/0x178 [2073.019249] blk_mq_exit_sched+0x12c/0x160 [2073.019252] elevator_exit +0xc8/0xd0 [2073.019256] blk_exit_queue+0x50/0x88 [2073.019259] blk_cleanup_queue +0x228/0x3d8 [2073.019267] null_del_dev+0xfc/0x1e0 [null_blk] [2073.019274] null_exit +0x90/0x114 [null_blk] [2073.019278] __arm64_sys_delete_module+0x358/0x5a0 [2073.019282] el0_svc_common+0xc8/0x320 [2073.019287] el0_svc_handler+0xf8/0x160 [2073.019290] el0_svc+0x10/0x218 [2073.019291] [2073.019294] Allocated by task 14163: [2073.019301] kasan_kmalloc+0xe0/0x190 [2073.019305] kmem_cache_alloc_node_trace+0x1cc/0x418 [2073.019308] bfq_pd_alloc+0x54/0x118 [2073.019313] blkcg_activate_policy+0x250/0x460 [2073.019317] bfq_create_group_hierarchy+0x38/0x110 [2073.019321] bfq_init_queue+0x6d0/0x948 [2073.019325] blk_mq_init_sched+0x1d8/0x390 [2073.019330] elevator_switch_mq+0x88/0x170 [2073.019334] elevator_switch+0x140/0x270 [2073.019338] elv_iosched_store+0x1a4/0x2a0 [2073.019342] queue_attr_store+0x90/0xe0 [2073.019348] sysfs_kf_write+0xa8/0xe8 [2073.019351] kernfs_fop_write+0x1f8/0x378 [2073.019359] __vfs_write+0xe0/0x360 [2073.019363] vfs_write +0xf0/0x270 [2073.019367] ksys_write+0xdc/0x1b8 [2073.019371] __arm64_sys_write +0x50/0x60 [2073.019375] el0_svc_common+0xc8/0x320 [2073.019380] el0_svc_handler +0xf8/0x160 [2073.019383] el0_svc+0x10/0x218 [2073.019385] [2073.019387] Freed by task 72584: [2073.019391] __kasan_slab_free+0x120/0x228 [2073.019394] kasan_slab_free +0x10/0x18 [2073.019397] kfree+0x94/0x368 [2073.019400] bfqg_put+0x64/0xb0 [2073.019404] bfqg_and_blkcg_put+0x90/0xb0 [2073.019408] bfq_put_queue+0x220/0x228 [2073.019413] __bfq_put_async_bfqq+0x98/0x168 [2073.019416] bfq_put_async_queues+0xbc/0x208 [2073.019420] bfq_pd_offline+0x178/0x238 [2073.019424] blkcg_deactivate_policy+0x1f0/0x420 [2073.019429] bfq_exit_queue+0x128/0x178 [2073.019433] blk_mq_exit_sched+0x12c/0x160 [2073.019437] elevator_exit+0xc8/0xd0 [2073.019440] blk_exit_queue+0x50/0x88 [2073.019443] blk_cleanup_queue +0x228/0x3d8 [2073.019451] null_del_dev+0xfc/0x1e0 [null_blk] [2073.019459] null_exit +0x90/0x114 [null_blk] [2073.019462] __arm64_sys_delete_module+0x358/0x5a0 [2073.019467] el0_svc_common+0xc8/0x320 [2073.019471] el0_svc_handler+0xf8/0x160 [2073.019474] el0_svc+0x10/0x218 [2073.019475] [2073.019479] The buggy address belongs to the object at ffff8000ccf63f00 which belongs to the cache kmalloc-1024 of size 1024 [2073.019484] The buggy address is located 552 bytes inside of 1024-byte region [ffff8000ccf63f00, ffff8000ccf64300) [2073.019486] The buggy address belongs to the page: [2073.019492] page:ffffe000333d800 count:1 mapcount:0 mapping:ffff8000c0003a00 index:0x0 compound_mapcount: 0 [2073.020123] flags: 0x7ffff0000008100(slab head) [2073.020403] raw: 07ffff00000008100 ffff7e0003334c08 ffff7e00001f5a08 ffff8000c0003a00 [2073.020409] ra ---truncated---
CVE-2022-49183	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_ct: fix ref leak when switching zones When switching zones or network namespaces without doing a ct clear in between, it is now leaking a reference to the old ct entry. That's because tcf_ct_skb_nfct_cached() returns false and tcf_ct_flow_table_lookup() may simply overwrite it. The fix is to, as the ct entry is not reusable, free it already at tcf_ct_skb_nfct_cached().
CVE-2022-49190	In the Linux kernel, the following vulnerability has been resolved: kernel/resource: fix kfree() of bootmem memory again Since commit ebf7d8f270d ("mem hotunplug: fix kfree() of bootmem memory"), we could get a resource allocated during boot via alloc_resource(). And it's required to release the resource using free_resource(). However, many people use kfree directly which will result in kernel BUG. In order to fix this without fixing every call site, just leak a couple of bytes in such corner case.

CVE ID	Description
CVE-2022-49193	In the Linux kernel, the following vulnerability has been resolved: ice: fix 'scheduling while atomic' on aux critical err interrupt There's a kernel BUG splat on processing aux critical error interrupts in ice_misc_intr(): [2100.917085] BUG: scheduling while atomic: swapper/15/0/0x00010000 ... [2101.060770] Call Trace: [2101.063229] <IRQ> [2101.065252] dump_stack+0x41/0x60 [2101.068587] __schedule_bug.cold.100+0x4c/0x58 [2101.073060] __schedule+0x6a4/0x830 [2101.076570] schedule+0x35/0xa0 [2101.079727] schedule_preempt_disabled+0xa/0x10 [2101.084284] __mutex_lock.isra.7+0x310/0x420 [2101.088580] ? ice_misc_intr+0x201/0x2e0 [ice] [2101.093078] ice_send_event_to_aux+0x25/0x70 [ice] [2101.097921] ice_misc_intr+0x220/0x2e0 [ice] [2101.102232] __handle_irq_event_percpu+0x40/0x180 [2101.106965] handle_irq_event_percpu+0x30/0x80 [2101.111434] handle_irq_event+0x36/0x53 [2101.115292] handle_edge_irq+0x82/0x190 [2101.119148] handle_irq+0x1c/0x30 [2101.122480] do_IRQ+0x49/0xd0 [2101.125465] common_interrupt+0xf/0xf [2101.129146] </IRQ> ... As Andrew correctly mentioned previously[0], the following call ladder happens: ice_misc_intr() <- hardirq ice_send_event_to_aux() device_lock() mutex_lock() might_sleep() might_resched() <- oops Add a new PF state bit which indicates that an aux critical error occurred and serve it in ice_service_task() in process context. The new ice_pf::oicr_err_reg is read-write in both hardirq and process contexts, but only 3 bits of non-critical data probably aren't worth explicit synchronizing (and they're even in the same byte [31:24]). [0] https://lore.kernel.org/all/YeSRUVmrdmLUXHDn@lunn.ch
CVE-2022-49196	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Fix use after free in remove_phb_dynamic() In remove_phb_dynamic() we use &phb->io_resource, after we've called device_unregister(&host_bridge->dev). But the unregister may have freed phb, because pcibios_free_controller_deferred() is the release function for the host_bridge. If there are no outstanding references when we call device_unregister() then phb will be freed out from under us. This has gone mainly unnoticed, but with slub_debug and page_poison enabled it can lead to a crash: PID: 7574 TASK: c0000000d492cb80 CPU: 13 COMMAND: "drmgr" #0 [c0000000e4f075a0] crash_kexec at c00000000027d7dc #1 [c0000000e4f075d0] oops_end at c00000000029608 #2 [c0000000e4f07650] __bad_page_fault at c000000000904b4 #3 [c0000000e4f076c0] do_bad_slb_fault at c00000000009a5a8 #4 [c0000000e4f076f0] data_access_slb_common_virt at c000000000008b30 Data SLB Access [380] exception frame: R0: c000000000167250 R1: c0000000e4f07a00 R2: c000000002a46100 R3: c000000002b39ce8 R4: 00000000000000c0 R5: 00000000000000a9 R6: 3894674d000000c0 R7: 0000000000000000 R8: 00000000000000ff R9: 0000000000000100 R10: 6b6b6b6b6b6b6b6b R11: 0000000000008000 R12: c0000000023da80 R13: c0000009ffd38b00 R14: 0000000000000000 R15: 000000011c87f0f0 R16: 0000000000000006 R17: 0000000000000003 R18: 0000000000000002 R19: 0000000000000004 R20: 0000000000000005 R21: 000000011c87ede8 R22: 000000011c87c5a8 R23: 000000011c87d3a0 R24: 0000000000000000 R25: 0000000000000001 R26: c0000000e4f07cc8 R27: c00000004d1cc400 R28: c0080000031d00e8 R29: c00000004d23d800 R30: c00000004d1d2400 R31: c00000004d1d2540 NIP: c00000000167258 MSR: 800000000009033 OR3: c0000000000e9f474 CTR: 0000000000000000 LR: c000000000167250 XER: 0000000020040003 CCR: 0000000024088420 MQ: 0000000000000000 DAR: 6b6b6b6b6b6b6b63 DSISR: c0000000e4f07920 Syscall Result: ffffffff2 [NIP : release_resource+56] [LR : release_resource+48] #5 [c0000000e4f07a00] release_resource at c000000000167258 (unreliable) #6 [c0000000e4f07a30] remove_phb_dynamic at c000000000105648 #7 [c0000000e4f07ab0] dlpar_remove_slot at c0080000031a09e8 [rpadlpar_io] #8 [c0000000e4f07b50] remove_slot_store at c0080000031a0b9c [rpadlpar_io] #9 [c0000000e4f07be0] kobj_attr_store at c000000000817d8c #10 [c0000000e4f07c00] sysfs_kf_write at c00000000063e504 #11 [c0000000e4f07c20] kernfs_fop_write_iter at c00000000063d868 #12 [c0000000e4f07c70] new_sync_write at c00000000054339c #13 [c0000000e4f07d10] vfs_write at c000000000546624 #14 [c0000000e4f07d60] ksys_write at c0000000005469f4 #15 [c0000000e4f07db0] system_call_exception at c00000000030840 #16 [c0000000e4f07e10] system_call_vectored_common at c00000000000c168 To avoid it, we can take a reference to the host_bridge->dev until we're done using phb. Then when we drop the reference the phb will be freed.
CVE-2022-49197	In the Linux kernel, the following vulnerability has been resolved: af_netlink: Fix shift out of bounds in group mask calculation When a netlink message is received, netlink_recvmsg() fills in the address of the sender. One of the fields is the 32-bit bitfield nl_groups, which carries the multicast group on which the message was received. The least significant bit corresponds to group 1, and therefore the highest group that the field can represent is 32. Above that, the UB sanitizer flags the out-of-bounds shift attempts. Which bits end up being set in such case is implementation defined, but it's either going to be a wrong non-zero value, or zero, which is at least not misleading. Make the latter choice deterministic by always setting to 0 for higher-numbered multicast groups. To get information about membership in groups >= 32, userspace is expected to use nl_pktinfo control messages[0], which are enabled by NETLINK_PKTINFO socket option. [0] https://lwn.net/Articles/147608/ The way to trigger this issue is e.g. through monitoring the BRVLAN group: # bridge monitor vlan & # ip link add name br type bridge Which produces the following citation: UBSAN: shift-out-of-bounds in net/netlink/af_netlink.c:162:19 shift exponent 32 is too large for 32-bit type 'int'

CVE ID	Description
CVE-2022-49198	In the Linux kernel, the following vulnerability has been resolved: mptcp: Fix crash due to tcp_tsorted_anchor was initialized before release skb Got crash when doing pressure test of mptcp: <pre>===== dst_release: dst:ffffa06ce6e5c058 refcnt:-1 kernel tried to execute NX-protected page - exploit attempt? (uid: 0) BUG: unable to handle kernel paging request at fffffa06ce6e5c058 PGD 190a01067 P4D 190a01067 PUD 43fffb067 PMD 22e403063 PTE 8000000226e5c063 Oops: 0011 [#1] SMP PTI CPU: 7 PID: 7823 Comm: kworker/7:0 Kdump: loaded Tainted: G E Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.2.1 04/01/2014 Call Trace: ? skb_release_head_state +0x68/0x100 ? skb_release_all+0xe/0x30 ? kfree_skb+0x32/0xa0 ? mptcp_sendmsg_frag +0x57e/0x750 ? __mptcp_retrans+0x21b/0x3c0 ? __switch_to_asm+0x35/0x70 ? mptcp_worker +0x25e/0x320 ? process_one_work+0x1a7/0x360 ? worker_thread+0x30/0x390 ? create_worker +0x1a0/0x1a0 ? kthread+0x112/0x130 ? kthread_flush_work_fn+0x10/0x10 ? ret_from_fork+0x35/0x40 =====</pre> In __mptcp_alloc_tx_skb skb was allocated and skb->tcp_tsorted_anchor will be initialized, in under memory pressure situation sk_wmem_schedule will return false and then kfree_skb. In this case skb->skb_refdst is not null because skb_refdst and tcp_tsorted_anchor are stored in the same mem, and kfree_skb will try to release dst and cause crash.
CVE-2022-49201	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: fix race between xmit and reset There is a race between reset and the transmit paths that can lead to ibmvnic_xmit() accessing an scrq after it has been freed in the reset path. It can result in a crash like: Kernel attempted to read user page (0) - exploit attempt? (uid: 0) BUG: Kernel NULL pointer dereference on read at 0x00000000 Faulting instruction address: 0xc0080000016189f8 Oops: Kernel access of bad area, sig: 11 [#1] ... NIP [c0080000016189f8] ibmvnic_xmit+0x60/0xb60 [ibmvnic] LR [c000000000c0046c] dev_hard_start_xmit+0x11c/0x280 Call Trace: [c008000001618f08] ibmvnic_xmit+0x570/0xb60 [ibmvnic] (unreliable) [c000000000c0046c] dev_hard_start_xmit+0x11c/0x280 [c000000000c9cfcc] sch_direct_xmit+0xec/0x330 [c000000000bfe640] __dev_xmit_skb+0x3a0/0x9d0 [c000000000c00ad4] __dev_queue_xmit+0x394/0x730 [c008000002db813c] __bond_start_xmit+0x254/0x450 [bonding] [c008000002db8378] bond_start_xmit+0x40/0xc0 [bonding] [c000000000c0046c] dev_hard_start_xmit+0x11c/0x280 [c000000000c00ca4] __dev_queue_xmit+0x564/0x730 [c000000000cf97e0] neigh_hh_output+0xd0/0x180 [c000000000cfa69c] ip_finish_output2+0x31c/0x5c0 [c000000000cfd244] __ip_queue_xmit+0x194/0x4f0 [c000000000d2a3c4] __tcp_transmit_skb+0x434/0x9b0 [c000000000d2d1e0] __tcp_retransmit_skb+0x1d0/0x6a0 [c000000000d2d984] tcp_retransmit_skb+0x34/0x130 [c000000000d310e8] tcp_retransmit_timer+0x388/0x6d0 [c000000000d315ec] tcp_write_timer_handler+0x1bc/0x330 [c000000000d317bc] tcp_write_timer+0x5c/0x200 [c000000000243270] call_timer_fn+0x50/0x1c0 [c000000000243704] __run_timers.part.0+0x324/0x460 [c000000000243894] run_timer_softirq+0x54/0xa0 [c0000000000ea713c] __do_softirq+0x15c/0x3e0 [c000000000166258] __irq_exit_rcu+0x158/0x190 [c000000000166420] irq_exit+0x20/0x40 [c0000000002853c] timer_interrupt+0x14c/0x2b0 [c000000000009a00] decrementer_common_virt+0x210/0x220 --- interrupt: 900 at plpar_hcall_norets_notrace+0x18/0x2c The immediate cause of the crash is the access of tx_scrq in the following snippet during a reset, where the tx_scrq can be either NULL or an address that will soon be invalid: ibmvnic_xmit() { ... tx_scrq = adapter->tx_scrq[queue_num]; txq = netdev_get_tx_queue(netdev, queue_num); ind_bufp = &tx_scrq->ind_buf; if (test_bit(0, &adapter->resetting)) { ... } But beyond that, the call to ibmvnic_xmit() itself is not safe during a reset and the reset path attempts to avoid this by stopping the queue in ibmvnic_cleanup(). However just after the queue was stopped, an in-flight ibmvnic_complete_tx() could have restarted the queue even as the reset is progressing. Since the queue was restarted we could get a call to ibmvnic_xmit() which can then access the bad tx_scrq (or other fields). We cannot however simply have ibmvnic_complete_tx() check the ->resetting bit and skip starting the queue. This can race at the "back-end" of a good reset which just restarted the queue but has not cleared the ->resetting bit yet. If we skip restarting the queue due to ->resetting being true, the queue would remain stopped indefinitely potentially leading to transmit timeouts. IOW ->resetting is too broad for this purpose. Instead use a new flag that indicates whether or not the queues are active. Only the open/ reset paths control when the queues are active. ibmvnic_complete_tx() and others wake up the queue only if the queue is marked active. So we will have: A. reset/open thread in ibmvnic_cleanup() and __ibmvnic_open() ->resetting = true ->tx_queues_active = false disable tx queues ... ->tx_queues_active = true start tx queues B. Tx interrupt in ibmvnic_complete_tx(): if (->tx_queues_active) netif_wake_subqueue(); To ensure that ->tx_queues_active and state of the queues are consistent, we need a lock which: - must also be taken in the interrupt path (ibmvnic_complete_tx()) - shared across the multiple ---truncated---

CVE ID	Description
CVE-2022-49204	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix more uncharged while msg has more_data In tcp_bpf_send_verdict(), if msg has more data after tcp_bpf_sendmsg_redir(): tcp_bpf_send_verdict() tosend = msg->sg.size //msg->sg.size = 22220 case __SK_REDIRECT: sk_msg_return() //uncharged msg->sg.size(22220) sk->sk_forward_alloc tcp_bpf_sendmsg_redir() //after tcp_bpf_sendmsg_redir, msg->sg.size=11000 goto more_data; tosend = msg->sg.size //msg->sg.size = 11000 case __SK_REDIRECT: sk_msg_return() // uncharged msg->sg.size(11000) to sk->sk_forward_alloc The msg->sg.size(11000) has been uncharged twice, to fix we can charge the remaining msg->sg.size before goto more data. This issue can cause the following info: WARNING: CPU: 0 PID: 9860 at net/core/stream.c:208 sk_stream_kill_queues+0xd4/0x1a0 Call Trace: <TASK> inet_csk_destroy_sock+0x55/0x110 __tcp_close+0x279/0x470 tcp_close+0x1f/0x60 inet_release+0x3f/0x80 __sock_release+0x3d/0xb0 sock_close+0x11/0x20 __fput+0x92/0x250 task_work_run+0x6a/0xa0 do_exit+0x33b/0xb60 do_group_exit+0x2f/0xa0 get_signal+0xb6/0x950 arch_do_signal_or_restart+0xac/0x2a0 ? vfs_write+0x237/0x290 exit_to_user_mode_prepare+0xa9/0x200 syscall_exit_to_user_mode +0x12/0x30 do_syscall_64+0x46/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae </TASK> WARNING: CPU: 0 PID: 2136 at net/ipv4/af_inet.c:155 inet_sock_destruct+0x13c/0x260 Call Trace: <TASK> __sk_destruct+0x24/0x1f0 sk_psock_destroy+0x19b/0x1c0 process_one_work +0x1b3/0x3c0 worker_thread+0x30/0x350 ? process_one_work+0x3c0/0x3c0 kthread+0xe6/0x110 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-49205	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix double uncharge the mem of sk_msg If tcp_bpf_sendmsg is running during a tear down operation, psock may be freed. tcp_bpf_sendmsg() tcp_bpf_send_verdict() sk_msg_return() tcp_bpf_sendmsg_redir() unlikely(!psock)) sk_msg_free() The mem of msg has been uncharged in tcp_bpf_send_verdict() by sk_msg_return(), and would be uncharged by sk_msg_free() again. When psock is null, we can simply returning an error code, this would then trigger the sk_msg_free_nocharge in the error path of __SK_REDIRECT and would have the side effect of throwing an error up to user space. This would be a slight change in behavior from user side but would look the same as an error if the redirect on the socket threw an error. This issue can cause the following info: WARNING: CPU: 0 PID: 2136 at net/ipv4/af_inet.c:155 inet_sock_destruct+0x13c/0x260 Call Trace: <TASK> __sk_destruct+0x24/0x1f0 sk_psock_destroy +0x19b/0x1c0 process_one_work+0x1b3/0x3c0 worker_thread+0x30/0x350 ? process_one_work +0x3c0/0x3c0 kthread+0xe6/0x110 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-49206	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix memory leak in error flow for subscribe event routine In case the second xa_insert() fails, the obj_event is not released. Fix the error unwind flow to free that memory to avoid a memory leak.
CVE-2022-49207	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix memleak in sk_psock_queue_msg If tcp_bpf_sendmsg is running during a tear down operation we may enqueue data on the ingress msg queue while tear down is trying to free it. sk1 (redirect sk2) sk2 ----- tcp_bpf_sendmsg() tcp_bpf_send_verdict() tcp_bpf_sendmsg_redir() bpf_tcp_ingress() sock_map_close() lock_sock() lock_sock() ... blocking sk_psock_stop sk_psock_clear_state(psock, SK_PSOCK_TX_ENABLED); release_sock(sk); lock_sock() sk_mem_charge() get_page() sk_psock_queue_msg() sk_psock_test_state(psock, SK_PSOCK_TX_ENABLED); drop_sk_msg() release_sock() While drop_sk_msg(), the msg has charged memory from sk by sk_mem_charge and has sg pages need to put. To fix we use sk_msg_free() and then kfree() msg. This issue can cause the following info: WARNING: CPU: 0 PID: 9202 at net/core/stream.c:205 sk_stream_kill_queues +0xc8/0xe0 Call Trace: <IRQ> inet_csk_destroy_sock+0x55/0x110 tcp_rcv_state_process +0xe5f/0xe90 ? sk_filter_trim_cap+0x10d/0x230 ? tcp_v4_do_rcv+0x161/0x250 tcp_v4_do_rcv +0x161/0x250 tcp_v4_rcv+0xc3a/0xce0 ip_protocol_deliver_rcu+0x3d/0x230 ip_local_deliver_finish +0x54/0x60 ip_local_deliver+0xfd/0x110 ? ip_protocol_deliver_rcu+0x230/0x230 ip_rcv+0xd6/0x100 ? ip_local_deliver+0x110/0x110 __netif_receive_skb_one_core+0x85/0xa0 process_backlog+0xa4/0x160 __napi_poll+0x29/0x1b0 net_rx_action+0x287/0x300 __do_softirq+0xff/0x2fc do_softirq+0x79/0x90 </IRQ> WARNING: CPU: 0 PID: 531 at net/ipv4/af_inet.c:154 inet_sock_destruct+0x175/0x1b0 Call Trace: <TASK> __sk_destruct+0x24/0x1f0 sk_psock_destroy+0x19b/0x1c0 process_one_work +0x1b3/0x3c0 ? process_one_work+0x3c0/0x3c0 worker_thread+0x30/0x350 ? process_one_work +0x3c0/0x3c0 kthread+0xe6/0x110 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-49208	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Prevent some integer underflows My static checker complains that: drivers/infiniband/hw/irdma/ctrl.c:3605 irdma_sc_ceq_init() warn: can subtract underflow 'info->dev->hmc_fpm_misc.max_ceqs'? It appears that "info->dev->hmc_fpm_misc.max_ceqs" comes from the firmware in irdma_sc_parse_fpm_query_buf() so, yes, there is a chance that it could be zero. Even if we trust the firmware, it's easy enough to change the condition just as a hardenning measure.

CVE ID	Description
CVE-2022-49209	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix memleak in tcp_bpf_sendmsg while sk msg is full If tcp_bpf_sendmsg() is running while sk msg is full. When sk_msg_alloc() returns -ENOMEM error, tcp_bpf_sendmsg() goes to wait_for_memory. If partial memory has been allocated by sk_msg_alloc(), that is, msg_tx->sg.size is greater than osize after sk_msg_alloc(), memleak occurs. To fix we use sk_msg_trim() to release the allocated memory, then goto wait for memory. Other call paths of sk_msg_alloc() have the similar issue, such as tls_sw_sendmsg(), so handle sk_msg_trim logic inside sk_msg_alloc(), as Cong Wang suggested. This issue can cause the following info: WARNING: CPU: 3 PID: 7950 at net/core/stream.c:208 sk_stream_kill_queues+0xd4/0x1a0 Call Trace: <TASK> inet_csk_destroy_sock+0x55/0x110 __tcp_close+0x279/0x470 tcp_close+0x1f/0x60 inet_release+0x3f/0x80 __sock_release+0x3d/0xb0 sock_close+0x11/0x20 __fput+0x92/0x250 task_work_run+0x6a/0xa0 do_exit+0x33b/0xb60 do_group_exit+0x2f/0xa0 get_signal+0xb6/0x950 arch_do_signal_or_restart+0xac/0x2a0 exit_to_user_mode_prepare+0xa9/0x200 syscall_exit_to_user_mode+0x12/0x30 do_syscall_64+0x46/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae </TASK> WARNING: CPU: 3 PID: 2094 at net/ipv4/af_inet.c:155 inet_sock_destruct+0x13c/0x260 Call Trace: <TASK> __sk_destruct+0x24/0x1f0 sk_psock_destroy+0x19b/0x1c0 process_one_work+0x1b3/0x3c0 kthread+0xe6/0x110 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-49215	In the Linux kernel, the following vulnerability has been resolved: xsk: Fix race at socket teardown Fix a race in the xsk socket teardown code that can lead to a NULL pointer dereference splat. The current xsk unbind code in xsk_unbind_dev() starts by setting xs->state to XSK_UNBOUND, sets xs->dev to NULL and then waits for any NAPI processing to terminate using synchronize_net(). After that, the release code starts to tear down the socket state and free allocated memory. BUG: kernel NULL pointer dereference, address: 000000000000000c0 PGD 80000000932469067 P4D 80000000932469067 PUD 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 25 PID: 69132 Comm: grcppp_sync_ser Tainted: G I 5.16.0+ #2 Hardware name: Dell Inc. PowerEdge R730/0599V5, BIOS 1.2.10 03/09/2015 RIP: 0010:__xsk_sendmsg+0x2c/0x690 [...] RSP: 0018:ffffa2348bd13d50 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 0000000000000040 RCX: ffff8d5fc632d258 RDX: 0000000000400000 RSI: ffffa2348bd13e10 RDI: ffff8d5fc5489800 RBP: ffffa2348bd13db0 R08: 0000000000000000 R09: 00007fffff000 R10: 0000000000000000 R11: 0000000000000000 R12: ffff8d5fc5489800 R13: ffff8d5fcb0f5140 R14: ffff8d5fcb0f5140 R15: 0000000000000000 FS: 00007f991cff9400(0000) GS: ffff8d6f1f700000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000000000c0 CR3: 0000000114888005 CR4: 00000000001706e0 Call Trace: <TASK> ? aa_sk_perm+0x43/0x1b0 xsk_sendmsg+0xf0/0x110 sock_sendmsg+0x65/0x70 __sys_sendto+0x113/0x190 ? debug_smp_processor_id+0x17/0x20 ? fpregs_assert_state_consistent+0x23/0x50 ? exit_to_user_mode_prepare+0xa5/0x1d0 __x64_sys_sendto+0x29/0x30 do_syscall_64+0x3b/0xc0 entry_SYSCALL_64_after_hwframe+0x44/0xae There are two problems with the current code. First, setting xs->dev to NULL before waiting for all users to stop using the socket is not correct. The entry to the data plane functions xsk_poll(), xsk_sendmsg(), and xsk_recvmmsg() are all guarded by a test that xs->state is in the state XSK_BOUND and if not, it returns right away. But one process might have passed this test but still have not gotten to the point in which it uses xs->dev in the code. In this interim, a second process executing xsk_unbind_dev() might have set xs->dev to NULL which will lead to a crash for the first process. The solution here is just to get rid of this NULL assignment since it is not used anymore. Before commit 42fddcc7c64b ("xsk: use state member for socket synchronization"), xs->dev was the gatekeeper to admit processes into the data plane functions, but it was replaced with the state variable xs->state in the aforementioned commit. The second problem is that synchronize_net() does not wait for any process in xsk_poll(), xsk_sendmsg(), or xsk_recvmmsg() to complete, which means that the state they rely on might be cleaned up prematurely. This can happen when the notifier gets called (at driver unload for example) as it uses xsk_unbind_dev(). Solve this by extending the RCU critical region from just the ndo_xsk_wakeup to the whole functions mentioned above, so that both the test of xs->state == XSK_BOUND and the last use of any member of xs is covered by the RCU critical section. This will guarantee that when synchronize_net() completes, there will be no processes left executing xsk_poll(), xsk_sendmsg(), or xsk_recvmmsg() and state can be cleaned up safely. Note that we need to drop the RCU lock for the skb xmit path as it uses functions that might sleep. Due to this, we have to retest the xs->state after we grab the mutex that protects the skb xmit code from, among a number of things, an xsk_unbind_dev() being executed from the notifier at the same time.
CVE-2022-49220	In the Linux kernel, the following vulnerability has been resolved: dax: make sure inodes are flushed before destroy cache A bug can be triggered by following command \$ modprobe nd_pmem && modprobe -r nd_pmem [10.060014] BUG dax_cache (Not tainted): Objects remaining in dax_cache on __kmem_cache_shutdown() [10.060938] Slab 0x0000000085b729ac objects=9 used=1 fp=0x000000004f5ae469 flags=0x200000000010200(slab head node) [10.062433] Call Trace: [10.062673] dump_stack_lvl+0x34/0x44 [10.062865] slab_err+0x90/0xd0 [10.063619] __kmem_cache_shutdown+0x13b/0x2f0 [10.063848] kmem_cache_destroy+0x4a/0x110 [10.064058] __x64_sys_delete_module+0x265/0x300 This is caused by dax_fs_exit() not flushing inodes before destroy cache. To fix this issue, call rcu_barrier() before destroy cache.

CVE ID	Description
CVE-2022-49225	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921s: fix a possible memory leak in mt7921_load_patch Always release fw data at the end of mt7921_load_patch routine.
CVE-2022-49231	In the Linux kernel, the following vulnerability has been resolved: rtw88: fix memory overrun and memory leak during hw_scan Previously we allocated less memory than actual required, overwrite to the buffer causes the mm module to complaint and raise access violation faults. Along with potential memory leaks when returned early. Fix these by passing the correct size and proper deinit flow.
CVE-2022-49235	In the Linux kernel, the following vulnerability has been resolved: ath9k_htc: fix uninit value bugs Syzbot reported 2 KMSAN bugs in ath9k. All of them are caused by missing field initialization. In htc_connect_service() svc_meta_len and pad are not initialized. Based on code it looks like in current skb there is no service data, so simply initialize svc_meta_len to 0. htc_issue_send() does not initialize htc_frame_hdr::control array. Based on firmware code, it will initialize it by itself, so simply zero whole array to make KMSAN happy Fail logs: BUG: KMSAN: kernel-usb-infoleak in usb_submit_urb+0x6c1/0x2aa0 drivers/usb/core/urb.c:430 usb_submit_urb+0x6c1/0x2aa0 drivers/usb/core/urb.c:430 hif_usb_send_regout drivers/net/wireless/ath/ath9k/hif_usb.c:127 [inline] hif_usb_send+0x5f0/0x16f0 drivers/net/wireless/ath/ath9k/hif_usb.c:479 htc_issue_send drivers/net/wireless/ath/ath9k/htc_hst.c:34 [inline] htc_connect_service+0x143e/0x1960 drivers/net/wireless/ath/ath9k/htc_hst.c:275 ... Uninit was created at: slab_post_alloc_hook mm/slab.h:524 [inline] slab_alloc_node mm/slab.c:3251 [inline] __kmalloc_node_track_caller+0xe0c/0x1510 mm/slab.c:4974 kmalloc_reserve net/core/skbuff.c:354 [inline] __alloc_skb+0x545/0xf90 net/core/skbuff.c:426 alloc_skb include/linux/skbuff.h:1126 [inline] htc_connect_service+0x1029/0x1960 drivers/net/wireless/ath/ath9k/htc_hst.c:258 ... Bytes 4-7 of 18 are uninitialized Memory access of size 18 starts at ffff88027377e00 BUG: KMSAN: kernel-usb-infoleak in usb_submit_urb+0x6c1/0x2aa0 drivers/usb/core/urb.c:430 usb_submit_urb+0x6c1/0x2aa0 drivers/usb/core/urb.c:430 hif_usb_send_regout drivers/net/wireless/ath/ath9k/hif_usb.c:127 [inline] hif_usb_send+0x5f0/0x16f0 drivers/net/wireless/ath/ath9k/hif_usb.c:479 htc_issue_send drivers/net/wireless/ath/ath9k/htc_hst.c:34 [inline] htc_connect_service+0x143e/0x1960 drivers/net/wireless/ath/ath9k/htc_hst.c:275 ... Uninit was created at: slab_post_alloc_hook mm/slab.h:524 [inline] slab_alloc_node mm/slab.c:3251 [inline] __kmalloc_node_track_caller+0xe0c/0x1510 mm/slab.c:4974 kmalloc_reserve net/core/skbuff.c:354 [inline] __alloc_skb+0x545/0xf90 net/core/skbuff.c:426 alloc_skb include/linux/skbuff.h:1126 [inline] htc_connect_service+0x1029/0x1960 drivers/net/wireless/ath/ath9k/htc_hst.c:258 ... Bytes 16-17 of 18 are uninitialized Memory access of size 18 starts at ffff88027377e00
CVE-2022-49236	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix UAF due to race between btf_try_get_module and load_module While working on code to populate kfunc BTF ID sets for module BTF from its initcall, I noticed that by the time the initcall is invoked, the module BTF can already be seen by userspace (and the BPF verifier). The existing btf_try_get_module calls try_module_get which only fails if mod->state == MODULE_STATE_GOING, i.e. it can increment module reference when module initcall is happening in parallel. Currently, BTF parsing happens from MODULE_STATE_COMING notifier callback. At this point, the module initcalls have not been invoked. The notifier callback parses and prepares the module BTF, allocates an ID, which publishes it to userspace, and then adds it to the btf_modules list allowing the kernel to invoke btf_try_get_module for the BTF. However, at this point, the module has not been fully initialized (i.e. its initcalls have not finished). The code in module.c can still fail and free the module, without caring for other users. However, nothing stops btf_try_get_module from succeeding between the state transition from MODULE_STATE_COMING to MODULE_STATE_LIVE. This leads to a use-after-free issue when BPF program loads successfully in the state transition, load_module's do_init_module call fails and frees the module, and BPF program fd on close calls module_put for the freed module. Future patch has test case to verify we don't regress in this area in future. There are multiple points after prepare_coming_module (in load_module) where failure can occur and module loading can return error. We illustrate and test for the race using the last point where it can practically occur (in module __init function). An illustration of the race: CPU 0 CPU 1 load_module notifier_call(MODULE_STATE_COMING) btf_parse_module btf_alloc_id // Published to userspace list_add(&btf_mod->list, btf_modules) mod->init(...) ... ^ bpf_check check_pseudo_btf_id btf_try_get_module returns true ... module __init in progress return prog_fd ... V if (ret < 0) free_module(mod) ... close(prog_fd) ... bpf_prog_free_deferred module_put(used_btf.mod) // use-after-free We fix this issue by setting a flag BTF_MODULE_F_LIVE, from the notifier callback when MODULE_STATE_LIVE state is reached for the module, so that we return NULL from btf_try_get_module for modules that are not fully formed. Since try_module_get already checks that module is not in MODULE_STATE_GOING state, and that is the only transition a live module can make before being removed from btf_modules list, this is enough to close the race and prevent the bug. A later selftest patch crafts the race condition artificially to verify that it has been fixed, and that verifier fails to load program (with ENXIO). Lastly, a couple of comments: 1. Even if this race didn't exist, it seems more appropriate to only access resources (ksyms and kfuncs) of a fully formed module which has been initialized completely. 2. This patch was born out of need for synchronization against module initcall for the next patch, so it is needed for correctness even without the aforementioned race condition. The BTF resources initialized by module initcall are set up once and then only looked up, so just waiting until the initcall has finished ensures correct behavior.

CVE ID	Description
CVE-2022-49237	In the Linux kernel, the following vulnerability has been resolved: ath11k: add missing of_node_put() to avoid leak The node pointer is returned by of_find_node_by_type() or of_parse_phandle() with refcount incremented. Calling of_node_put() to avoid the refcount leak.
CVE-2022-49248	In the Linux kernel, the following vulnerability has been resolved: ALSA: firewire-lib: fix uninitialized flag for AV/C deferred transaction AV/C deferred transaction was supported at a commit 00a7bb81c20f ("ALSA: firewire-lib: Add support for deferred transaction") while 'deferrable' flag can be uninitialized for non-control/notify AV/C transactions. UBSAN reports it: kernel: <pre>===== kernel: UBSAN: invalid-load in /build/linux-aa0B4d/linux-5.15.0/sound/firewire/fcp.c:363:9 kernel: load of value 158 is not a valid value for type '_Bool' kernel: CPU: 3 PID: 182227 Comm: irq/35-firewire Tainted: P OE 5.15.0-18-generic #18-Ubuntu kernel: Hardware name: Gigabyte Technology Co., Ltd. AX370-Gaming 5/AX370-Gaming 5, BIOS F42b 08/01/2019 kernel: Call Trace: kernel: <IRQ> kernel: show_stack+0x52/0x58 kernel: dump_stack_lvl +0x4a/0x5f kernel: dump_stack+0x10/0x12 kernel: ubsan_epilogue+0x9/0x45 kernel: __ubsan_handle_load_invalid_value.cold+0x44/0x49 kernel: fcp_response.part.0.cold+0x1a/0x2b [snd_firewire_lib] kernel: fcp_response+0x28/0x30 [snd_firewire_lib] kernel: fw_core_handle_request +0x230/0x3d0 [firewire_core] kernel: handle_ar_packet+0x1d9/0x200 [firewire_ohci] kernel: ? handle_ar_packet+0x1d9/0x200 [firewire_ohci] kernel: ? transmit_complete_callback +0x9f/0x120 [firewire_core] kernel: ar_context_tasklet+0xa8/0x2e0 [firewire_ohci] kernel: tasklet_action_common.constprop.0+0xea/0xf0 kernel: tasklet_action+0x22/0x30 kernel: __do_softirq+0xd9/0x2e3 kernel: ? irq_finalize_oneshot.part.0+0xf0/0xf0 kernel: do_softirq +0x75/0xa0 kernel: </IRQ> kernel: <TASK> kernel: __local_bh_enable_ip+0x50/0x60 kernel: irq_forced_thread_fn+0x7e/0x90 kernel: irq_thread+0xba/0x190 kernel: ? irq_thread_fn +0x60/0x60 kernel: kthread+0x11e/0x140 kernel: ? irq_thread_check_affinity+0xf0/0xf0 kernel: ? set_kthread_struct+0x50/0x50 kernel: ret_from_fork+0x22/0x30 kernel: </TASK> kernel: =====</pre> This commit fixes the bug. The bug has no disadvantage for the non- control/notify AV/C transactions since the flag has an effect for AV/C response with INTERIM (0x0f) status which is not used for the transactions in AV/C general specification.
CVE-2022-49264	In the Linux kernel, the following vulnerability has been resolved: exec: Force single empty string when argv is empty Quoting[1] Ariadne Conill: "In several other operating systems, it is a hard requirement that the second argument to execve(2) be the name of a program, thus prohibiting a scenario where argc < 1. POSIX 2017 also recommends this behaviour, but it is not an explicit requirement[2]: The argument arg0 should point to a filename string that is associated with the process being started by one of the exec functions. ... Interestingly, Michael Kerrisk opened an issue about this in 2008[3], but there was no consensus to support fixing this issue then. Hopefully now that CVE-2021-4034 shows practical exploitative use[4] of this bug in a shellcode, we can reconsider. This issue is being tracked in the KSPP issue tracker[5]." While the initial code searches[6][7] turned up what appeared to be mostly corner case tests, trying to that just reject argv == NULL (or an immediately terminated pointer list) quickly started tripping[8] existing userspace programs. The next best approach is forcing a single empty string into argv and adjusting argc to match. The number of programs depending on argc == 0 seems a smaller set than those calling execve with a NULL argv. Account for the additional stack space in bprm_stack_limits(). Inject an empty string when argc == 0 (and set argc = 1). Warn about the case so userspace has some notice about the change: process './argc0' launched './argc0' with NULL argv: empty string added Additionally WARN() and reject NULL argv usage for kernel threads. [1] https://lore.kernel.org/lkml/20220127000724.15106-1-ariadne@dereferenced.org/ [2] https://pubs.opengroup.org/onlinepubs/9699919799/functions/exec.html [3] https://bugzilla.kernel.org/show_bug.cgi?id=8408 [4] https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt [5] https://github.com/KSPP/linux/issues/176 [6] https://codesearch.debian.net/search?q=execve%5C+%5C%28%5B%5E%2C%5D%2B%2C+*NULL&literal=0 [7] https://codesearch.debian.net/search?q=execclp%3F%5Cs+%5C%28%5B%5E%2C%5D%2B%2C%5Cs*NULL&literal=0 [8] https://lore.kernel.org/lkml/20220131144352.GE16385@xsang-OptiPlex-9020/

CVE ID	Description
CVE-2022-49265	In the Linux kernel, the following vulnerability has been resolved: PM: domains: Fix sleep-in-atomic bug caused by genpd_debug_remove() When a genpd with GENPD_FLAG_IRQ_SAFE gets removed, the following sleep-in-atomic bug will be seen, as genpd_debug_remove() will be called with a spinlock being held. [0.029183] BUG: sleeping function called from invalid context at kernel/locking/rwsem.c:1460 [0.029204] in_atomic(): 1, irqs_disabled(): 128, non_block: 0, pid: 1, name: swapper/0 [0.029219] preempt_count: 1, expected: 0 [0.029230] CPU: 1 PID: 1 Comm: swapper/0 Not tainted 5.17.0-rc4+ #489 [0.029245] Hardware name: Thundercomm TurboX CM2290 (DT) [0.029256] Call trace: [0.029265] dump_backtrace.part.0+0x9c/0xd0 [0.029285] show_stack+0x3c/0xa0 [0.029298] dump_stack_lvl+0x7c/0xa0 [0.029311] dump_stack+0x18/0x34 [0.029323] __might_resched+0x10c/0x13c [0.029338] __might_sleep+0x4c/0x80 [0.029351] down_read+0x24/0xd0 [0.029363] lookup_one_len_unlocked+0x9c/0xcc [0.029379] lookup_positive_unlocked+0x10/0x50 [0.029392] debugfs_lookup+0x68/0xac [0.029406] genpd_remove.part.0+0x12c/0x1b4 [0.029419] of_genpd_remove_last+0xa8/0xd4 [0.029434] psci_cpuidle_domain_probe+0x174/0x53c [0.029449] platform_probe+0x68/0xe0 [0.029462] really_probe+0x190/0x430 [0.029473] __driver_probe_device+0x90/0x18c [0.029485] driver_probe_device+0x40/0xe0 [0.029497] __driver_attach+0xf4/0x1d0 [0.029508] bus_for_each_dev+0x70/0xd0 [0.029523] driver_attach+0x24/0x30 [0.029534] bus_add_driver+0x164/0x22c [0.029545] driver_register+0x78/0x130 [0.029556] __platform_driver_register+0x28/0x34 [0.029569] psci_idle_init_domains+0x1c/0x28 [0.029583] do_one_initcall+0x50/0x1b0 [0.029595] kernel_init_freeable+0x214/0x280 [0.029609] kernel_init+0x2c/0x13c [0.029622] ret_from_fork+0x10/0x20 It doesn't seem necessary to call genpd_debug_remove() with the lock, so move it out from locking to fix the problem.
CVE-2022-49266	In the Linux kernel, the following vulnerability has been resolved: block: fix rq-qos breakage from skipping rq_qos_done_bio() a647a524a467 ("block: don't call rq_qos_ops->done_bio if the bio isn't tracked") made bio_endio() skip rq_qos_done_bio() if BIO_TRACKED is not set. While this fixed a potential oops, it also broke blk-iocost by skipping the done_bio callback for merged bios. Before, whether a bio goes through rq_qos_throttle() or rq_qos_merge(), rq_qos_done_bio() would be called on the bio on completion with BIO_TRACKED distinguishing the former from the latter. rq_qos_done_bio() is not called for bios which went through rq_qos_merge(). This royally confuses blk-iocost as the merged bios never finish and are considered perpetually in-flight. One reliably reproducible failure mode is an intermediate cgroup getting stuck active preventing its children from being activated due to the leaf-only rule, leading to loss of control. The following is from resctl-bench protection scenario which emulates isolating a web server like workload from a memory bomb run on an iocost configuration which should yield a reasonable level of protection. # cat /sys/block/nvme2n1/device/model Samsung SSD 970 PRO 512GB # cat /sys/fs/cgroup/io.cost.model 259:0 ctrl=user model=linear rbps=834913556 rseqiops=93622 rrandiops=102913 wbps=618985353 wseqiops=72325 wrandiops=71025 # cat /sys/fs/cgroup/io.cost.qos 259:0 enable=1 ctrl=user rpct=95.00 rlat=18776 wpct=95.00 wlat=8897 min=60.00 max=100.00 # resctl-bench -m 29.6G -r out.json run protection::scenario=mem-hog,loops=1 ... Memory Hog Summary ===== IO Latency: R p50=242u:336u/2.5m p90=794u:1.4m/7.5m p99=2.7m:8.0m/62.5m max=8.0m:36.4m/350m W p50=221u:323u/1.5m p90=709u:1.2m/5.5m p99=1.5m:2.5m/9.5m max=6.9m:35.9m/350m Isolation and Request Latency Impact Distributions: min p01 p05 p10 p25 p50 p75 p90 p95 p99 max mean stdev isol% 15.90 15.90 15.90 40.05 57.24 59.07 60.01 74.63 74.63 90.35 90.35 58.12 15.82 lat_imp% 0 0 0 0 4.55 14.68 15.54 233.5 548.1 548.1 53.88 143.6 Result: isol=58.12:15.82% lat_imp=53.88%:143.6 work_csv=100.0% missing=3.96% The isolation result of 58.12% is close to what this device would show without any IO control. Fix it by introducing a new flag BIO_QOS_MERGED to mark merged bios and calling rq_qos_done_bio() on them too. For consistency and clarity, rename BIO_TRACKED to BIO_QOS_THROTTLED. The flag checks are moved into rq_qos_done_bio() so that it's next to the code paths that set the flags. With the patch applied, the above same benchmark shows: # resctl-bench -m 29.6G -r out.json run protection::scenario=mem-hog,loops=1 ... Memory Hog Summary ===== IO Latency: R p50=123u:84.4u/985u p90=322u:256u/2.5m p99=1.6m:1.4m/9.5m max=11.1m:36.0m/350m W p50=429u:274u/995u p90=1.7m:1.3m/4.5m p99=3.4m:2.7m/11.5m max=7.9m:5.9m/26.5m Isolation and Request Latency Impact Distributions: min p01 p05 p10 p25 p50 p75 p90 p95 p99 max mean stdev isol% 84.91 84.91 89.51 90.73 92.31 94.49 96.36 98.04 98.71 100.0 100.0 94.42 2.81 5.73 11.11 13.92 17.53 22.61 4.10 4.68 Result: isol=94.42:2.81% lat_imp=4.10%:4.68 work_csv=58.34% missing=0%
CVE-2022-49267	In the Linux kernel, the following vulnerability has been resolved: mmc: core: use sysfs_emit() instead of sprintf() sprintf() (still used in the MMC core for the sysfs output) is vulnerable to the buffer overflow. Use the new-fangled sysfs_emit() instead. Found by Linux Verification Center (linuxtesting.org) with the SVACE static analysis tool.

CVE ID	Description
CVE-2022-49271	In the Linux kernel, the following vulnerability has been resolved: cifs: prevent bad output lengths in smb2_ioctl_query_info() When calling smb2_ioctl_query_info() with smb_query_info::flags=PASSTHRU_FSCTL and smb_query_info::output_buffer_length=0, the following would return 0x10 buffer = memdup_user(arg + sizeof(struct smb_query_info), qi.output_buffer_length); if (IS_ERR(buffer)) { kfree(vars); return PTR_ERR(buffer); } rather than a valid pointer thus making IS_ERR() check fail. This would then cause a NULL ptr dereference in @buffer when accessing it later in smb2_ioctl_query_ioctl(). While at it, prevent having a @buffer smaller than 8 bytes to correctly handle SMB2_SET_INFO FileEndOfFileInformation requests when smb_query_info::flags=PASSTHRU_SET_INFO. Here is a small C reproducer which triggers a NULL ptr in @buffer when passing an invalid smb_query_info::flags <pre>#include <stdio.h> #include <stdlib.h> #include <stdint.h> #include <unistd.h> #include <fcntl.h> #include <sys/ioctl.h> #define die(s) perror(s), exit(1) #define QUERY_INFO 0xc018cf07 int main(int argc, char *argv[]) { int fd; if (argc < 2) exit(1); fd = open(argv[1], O_RDONLY); if (fd == -1) die("open"); if (ioctl(fd, QUERY_INFO, (uint32_t)) { 0, 0, 0, 4, 0, 0}) == -1) die("ioctl"); close(fd); return 0; } mount.cifs //srv/share /mnt -o ... gcc repro.c && ./a.out /mnt/f0 [114.138620] general protection fault, probably for non-canonical address 0xdfdfc00000000000: 0000 [#1] PREEMPT SMP KASAN NOPTI [114.139310] KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] [114.139775] CPU: 2 PID: 995 Comm: a.out Not tainted 5.17.0-rc8 #1 [114.140148] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.15.0-0-g2dd4b9b-rebuilt.opensuse.org 04/01/2014 [114.140818] RIP: 0010:smb2_ioctl_query_info +0x206/0x410 [cifs] [114.141221] Code: 00 00 00 00 fc ff df 48 c1 ea 03 80 3c 02 00 0f 85 c8 01 00 00 48 b8 00 00 00 00 fc ff df 4c 8b 7b 28 4c 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 9c 01 00 00 49 8b 3f e8 58 02 fb ff 48 8b 14 24 [114.142348] RSP: 0018:ffff90000b47b00 EFLAGS: 00010256 [114.142692] RAX: dffffc0000000000 RBX: ffff888115503200 RCX: ffffffff020580d [114.143119] RDX: 0000000000000000 RSI: 0000000000000004 RDI: ffffffff043a380 [114.143544] RBP: ffff888115503278 R08: 0000000000000001 R09: 0000000000000000 [114.143983] R10: ffffbf4087470 R11: 0000000000000001 R12: ffff888115503288 [114.144424] R13: 00000000fffff0 R14: ffff888115503228 R15: 0000000000000000 [114.144852] FS: 00007f7aeabdf740(0000) GS:ffff888151600000(0000) knlGS:0000000000000000 [114.145338] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [114.145692] CR2: 00007f7aeacdf5e CR3: 000000012000e000 CR4: 0000000000350ee0 [114.146131] Call Trace: [114.146291] <TASK> [114.146432] ? smb2_query_reparse_tag+0x890/0x890 [cifs] [114.146800] ? cifs_mapchar +0x460/0x460 [cifs] [114.147121] ? rcu_read_lock_sched_held+0x3f/0x70 [114.147412] ? cifs_strndup_to_utf16+0x15b/0x250 [cifs] [114.147775] ? dentry_path_raw+0xa6/0xf0 [114.148024] ? cifs_convert_path_to_utf16+0x198/0x220 [cifs] [114.148413] ? smb2_check_message+0x1080/0x1080 [cifs] [114.148766] ? rcu_read_lock_sched_held+0x3f/0x70 [114.149065] cifs_ioctl+0x1577/0x3320 [cifs] [114.149371] ? lock_downgrade+0x6f0/0x6f0 [114.149631] ? cifs_readdir+0x2e60/0x2e60 [cifs] [114.149956] ? rcu_read_lock_sched_held+0x3f/0x70 [114.150250] ? __rseq_handle_notify_resume +0x80b/0xbe0 [114.150562] ? __up_read+0x192/0x710 [114.150791] ? __ia32_sys_rseq+0xf0/0xf0 [114.151025] ? __x64_sys_openat+0x11f/0x1d0 [114.151296] __x64_sys_ioctl+0x127/0x190 [114.151549] do_syscall_64+0x3b/0x90 [114.151768] entry_SYSCALL_64_after_hwframe +0x44/0xae [114.152079] RIP: 0033:0x7f7aead043df [114.152306] Code: 00 48 89 44 24 18 31 c0 48 8d 44 24 60 c7 04 24 ---truncated---</pre>
CVE-2022-49280	In the Linux kernel, the following vulnerability has been resolved: NFSD: prevent underflow in nfssvc_decode_writeargs() Smatch complains: fs/nfsd/nfsxdr.c:341 nfssvc_decode_writeargs() warn: no lower bound on 'args->len' Change the type to unsigned to prevent this issue.
CVE-2022-49281	In the Linux kernel, the following vulnerability has been resolved: cifs: fix handlecache and multiuser In multiuser each individual user has their own tcon structure for the share and thus their own handle for a cached directory. When we umount such a share we much make sure to release the pinned down dentry for each such tcon and not just the master tcon. Otherwise we will get nasty warnings on amount that dentries are still in use: [3459.590047] BUG: Dentry 00000000115c6f41 {i=12000000019d95,n=/} still in use! (2) [unmount of cifs cifs] ... [3459.590492] Call Trace: [3459.590500] d_walk+0x61/0x2a0 [3459.590518] ? shrink_lock_dentry.part.0+0xe0/0xe0 [3459.590526] shrink_dcache_for_umount+0x49/0x110 [3459.590535] generic_shutdown_super+0x1a/0x110 [3459.590542] kill_anon_super+0x14/0x30 [3459.590549] cifs_kill_sb+0xf5/0x104 [cifs] [3459.590773] deactivate_locked_super+0x36/0xa0 [3459.590782] cleanup_mnt+0x131/0x190 [3459.590789] task_work_run+0x5c/0x90 [3459.590798] exit_to_user_mode_loop+0x151/0x160 [3459.590809] exit_to_user_mode_prepare+0x83/0xd0 [3459.590818] syscall_exit_to_user_mode+0x12/0x30 [3459.590828] do_syscall_64+0x48/0x90 [3459.590833] entry_SYSCALL_64_after_hwframe+0x44/0xae

CVE ID	Description
CVE-2022-49286	In the Linux kernel, the following vulnerability has been resolved: tpm: use try_get_ops() in tpm-space.c As part of the series conversion to remove nested TPM operations: https://lore.kernel.org/all/20190205224723.19671-1-jarkko.sakkinen@linux.intel.com/ exposure of the chip->tpm_mutex was removed from much of the upper level code. In this conversion, tpm2_del_space() was missed. This didn't matter much because it's usually called closely after a converted operation, so there's only a very tiny race window where the chip can be removed before the space flushing is done which causes a NULL deref on the mutex. However, there are reports of this window being hit in practice, so fix this by converting tpm2_del_space() to use tpm_try_get_ops(), which performs all the teardown checks before acquiring the mutex.
CVE-2022-49287	In the Linux kernel, the following vulnerability has been resolved: tpm: fix reference counting for struct tpm_chip The following sequence of operations results in a refcount warning: 1. Open device / dev/tpmrm. 2. Remove module tpm_tis_spi. 3. Write a TPM command to the file descriptor opened at step 1. -----[cut here]----- WARNING: CPU: 3 PID: 1161 at lib/refcount.c:25 kobject_get +0xa0/0xa4 refcount_t: addition on 0; use-after-free. Modules linked in: tpm_tis_spi tpm_tis_core tpm mdio_bcm_unimac brcmfmac sha256_generic libsha256 sha256_arm hci_uart btbcm bluetooth cfg80211 vc4 brcmutil ecdh_generic ecc snd_soc_core crc32_arm_ce libaes raspberrypi_hwmon ac97_bus snd_pcm_dmaengine bcm2711_thermal snd_pcm snd_timer genet_snd phy_generic soundcore [last unloaded: spi_bcm2835] CPU: 3 PID: 1161 Comm: hold_open Not tainted 5.10.0ls-main-dirty #2 Hardware name: BCM2711 [<c0410c3c>] (unwind_backtrace) from [<c040b580>] (show_stack +0x10/0x14) [<c040b580>] (show_stack) from [<c1092174>] (dump_stack+0xc4/0xd8) [<c1092174>] (dump_stack) from [<c0445a30>] (__warn+0x104/0x108) [<c0445a30>] (__warn) from [<c0445aa8>] (warn_slowpath_fmt+0x74/0xb8) [<c0445aa8>] (warn_slowpath_fmt) from [<c08435d0>] (kobject_get +0xa0/0xa4) [<c08435d0>] (kobject_get) from [<bf0a715c>] (tpm_try_get_ops+0x14/0x54 [tpm]) [<bf0a715c>] (tpm_try_get_ops [tpm]) from [<bf0a7d6c>] (tpm_common_write+0x38/0x60 [tpm]) [<bf0a7d6c>] (tpm_common_write [tpm]) from [<c05a7ac0>] (vfs_write+0xc4/0x3c0) [<c05a7ac0>] (vfs_write) from [<c05a7ee4>] (ksys_write+0x58/0xcc) [<c05a7ee4>] (ksys_write) from [<c04001a0>] (ret_fast_syscall+0x0/0x4c) Exception stack(0xc226bfa8 to 0xc226bff0) bfa0: 00000000 000105b4 00000003 beafe664 00000014 00000000 bfc0: 00000000 000105b4 000103f8 00000004 00000000 00000000 b6f9c000 beafe684 bfe0: 0000006c beafe648 0001056c b6eb6944 ---[end trace d4b8409def9b8b1f]--- The reason for this warning is the attempt to get the chip->dev reference in tpm_common_write() although the reference counter is already zero. Since commit 8979b02aaf1d ("tpm: Fix reference count to main device") the extra reference used to prevent a premature zero counter is never taken, because the required TPM_CHIP_FLAG_TPM2 flag is never set. Fix this by moving the TPM 2 character device handling from tpm_chip_alloc() to tpm_add_char_device() which is called at a later point in time when the flag has been set in case of TPM2. Commit fdc915f7f719 ("tpm: expose spaces via a device link /dev/tpmrm<n>") already introduced function tpm_devs_release() to release the extra reference but did not implement the required put on chip->devs that results in the call of this function. Fix this by putting chip->devs in tpm_chip_unregister(). Finally move the new implementation for the TPM 2 handling into a new function to avoid multiple checks for the TPM_CHIP_FLAG_TPM2 flag in the good case and error cases.
CVE-2022-49288	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix races among concurrent prealloc proc writes We have no protection against concurrent PCM buffer preallocation changes via proc files, and it may potentially lead to UAF or some weird problem. This patch applies the PCM open_mutex to the proc write operation for avoiding the racy proc writes and the PCM stream open (and further operations).
CVE-2022-49290	In the Linux kernel, the following vulnerability has been resolved: mac80211: fix potential double free on mesh join While commit 6a01afc8468 ("mac80211: mesh: Free ie data when leaving mesh") fixed a memory leak on mesh leave / teardown it introduced a potential memory corruption caused by a double free when rejoining the mesh: ieee80211_leave_mesh() -> kfree(sdata->u.mesh.ie); ... ieee80211_join_mesh() -> copy_mesh_setup() -> old_ie = ifmsh->ie; -> kfree(old_ie); This double free / kernel panics can be reproduced by using wpa_supplicant with an encrypted mesh (if set up without encryption via "iw" then ifmsh->ie is always NULL, which avoids this issue). And then calling: \$ iw dev mesh0 mesh leave \$ iw dev mesh0 mesh join my-mesh Note that typically these commands are not used / working when using wpa_supplicant. And it seems that wpa_supplicant or wpa_cli are going through a NETDEV_DOWN/NETDEV_UP cycle between a mesh leave and mesh join where the NETDEV_UP resets the mesh.ie to NULL via a memcpy of default_mesh_setup in cfg80211_netdev_notifier_call, which then avoids the memory corruption, too. The issue was first observed in an application which was not using wpa_supplicant but "Senf" instead, which implements its own calls to nl80211. Fixing the issue by removing the kfree()ing of the mesh IE in the mesh join function and leaving it solely up to the mesh leave to free the mesh IE.

CVE ID	Description
CVE-2022-49291	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix races among concurrent hw_params and hw_free calls Currently we have neither proper check nor protection against the concurrent calls of PCM hw_params and hw_free ioctls, which may result in a UAF. Since the existing PCM stream lock can't be used for protecting the whole ioctl operations, we need a new mutex to protect those racy calls. This patch introduced a new mutex, runtime->buffer_mutex, and applies it to both hw_params and hw_free ioctl code paths. Along with it, the both functions are slightly modified (the mmap_count check is moved into the state-check block) for code simplicity.
CVE-2022-49293	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: initialize registers in nft_do_chain() Initialize registers to avoid stack leak into userspace.
CVE-2022-49294	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check if modulo is 0 before dividing. [How & Why] If a value of 0 is read, then this will cause a divide-by-0 panic.
CVE-2022-49295	In the Linux kernel, the following vulnerability has been resolved: nbd: call genl_unregister_family() first in nbd_cleanup() Otherwise there may be race between module removal and the handling of netlink command, which can lead to the oops as shown below: BUG: kernel NULL pointer dereference, address: 0000000000000098 Oops: 0002 [#1] SMP PTI CPU: 1 PID: 31299 Comm: nbd-client Tainted: G E 5.14.0-rc4 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) RIP: 0010:down_write+0x1a/0x50 Call Trace: start_creating+0x89/0x130 debugfs_create_dir+0x1b/0x130 nbd_start_device+0x13d/0x390 [nbd] nbd_genl_connect+0x42f/0x748 [nbd] genl_family_rcv_msg_doit.isra.0+0xec/0x150 genl_rcv_msg+0xe5/0x1e0 netlink_rcv_skb+0x55/0x100 genl_rcv+0x29/0x40 netlink_unicast+0x1a8/0x250 netlink_sendmsg+0x21b/0x430 ____sys_sendmsg+0x2a4/0x2d0 ____sys_sendmsg+0x81/0xc0 __sys_sendmsg+0x62/0xb0 __x64_sys_sendmsg+0x1f/0x30 do_syscall_64+0x3b/0xc0 entry_SYSCALL_64_after_hwframe+0x44/0xae Modules linked in: nbd(E-)
CVE-2022-49296	In the Linux kernel, the following vulnerability has been resolved: ceph: fix possible deadlock when holding Fwb to get inline_data 1, mount with wsync. 2, create a file with O_RDWR, and the request was sent to mds.0: ceph_atomic_open()-> ceph_mdsc_do_request(openc) finish_open(file, dentry, ceph_open)-> ceph_open()-> ceph_init_file()-> ceph_init_file_info()-> ceph_uninline_data()-> { ... if (inline_version == 1 /* initial version, no data */ inline_version == CEPH_INLINE_NONE) goto out_unlock; ... } The inline_version will be 1, which is the initial version for the new create file. And here the ci->i_inline_version will keep with 1, it's buggy. 3, buffer write to the file immediately: ceph_write_iter()-> ceph_get_caps(file, need=Fw, want=Fb, ...); generic_perform_write()-> a_ops->write_begin()-> ceph_write_begin()-> netfs_write_begin()-> netfs_begin_read()-> netfs_rreq_submit_slice()-> netfs_read_from_server()-> rreq->netfs_ops->issue_read()-> ceph_netfs_issue_read()-> { ... if (ci->i_inline_version != CEPH_INLINE_NONE && ceph_netfs_issue_op_inline(subreq)) return; ... } ceph_put_cap_refs(ci, Fwb); The ceph_netfs_issue_op_inline() will send a getattr(Fsr) request to mds.1. 4, then the mds.1 will request the rd lock for CInode::filelock from the auth mds.0, the mds.0 will do the CInode::filelock state transition from excl -> sync, but it need to revoke the Fxwb caps back from the clients. While the kernel client has already held the Fwb caps and waiting for the getattr(Fsr). It's deadlock! URL: https://tracker.ceph.com/issues/55377
CVE-2022-49297	In the Linux kernel, the following vulnerability has been resolved: nbd: fix io hung while disconnecting device In our tests, "qemu-nbd" triggers a io hung: INFO: task qemu-nbd:11445 blocked for more than 368 seconds. Not tainted 5.18.0-rc3-next-20220422-00003-g2176915513ca #884 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:qemu-nbd state:D stack: 0 pid:11445 ppid: 1 flags:0x00000000 Call Trace: <TASK> __schedule+0x480/0x1050 ? _raw_spin_lock_irqsave+0x3e/0xb0 schedule+0x9c/0x1b0 blk_mq_freeze_queue_wait+0x9d/0xf0 ? ipi_rseq+0x70/0x70 blk_mq_freeze_queue+0x2b/0x40 nbd_add_socket+0x6b/0x270 [nbd] nbd_ioctl+0x383/0x510 [nbd] blkdev_ioctl+0x18e/0x3e0 __x64_sys_ioctl+0xac/0x120 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7fd8ff706577 RSP: 002b:00007fd8fcdfeb8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffffda RBX: 0000000040000000 RCX: 00007fd8ff706577 RDX: 000000000000000d RSI: 000000000000ab00 RDI: 000000000000000f RBP: 000000000000000f R08: 0000000000000be8 R09: 000055fe497c62b0 R10: 00000002aff20000 R11: 0000000000000246 R12: 000000000000006d R13: 0000000000000000 R14: 00007ffe82dc5e70 R15: 00007fd8fcdff9c0 "qemu-nbd -d" will call ioctl 'NBD_DISCONNECT' first, however, following message was found: block nbd0: Send disconnect failed -32 Which indicate that something is wrong with the server. Then, "qemu-nbd -d" will call ioctl 'NBD_CLEAR_SOCK', however ioctl can't clear requests after commit 2516ab1543fd("nbd: only clear the queue on device teardown"). And in the meantime, request can't complete through timeout because nbd_xmit_timeout() will always return 'BLK_EH_RESET_TIMER', which means such request will never be completed in this situation. Now that the flag 'NBD_CMD_INFLIGHT' can make sure requests won't complete multiple times, switch back to call nbd_clear_sock() in nbd_clear_sock_ioctl(), so that inflight requests can be cleared.

CVE ID	Description
CVE-2022-49300	In the Linux kernel, the following vulnerability has been resolved: nbd: fix race between nbd_alloc_config() and module removal When nbd module is being removing, nbd_alloc_config() may be called concurrently by nbd_genl_connect(), although try_module_get() will return false, but nbd_alloc_config() doesn't handle it. The race may lead to the leak of nbd_config and its related resources (e.g. recv_workq) and oops in nbd_read_stat() due to the unload of nbd module as shown below: BUG: kernel NULL pointer dereference, address: 0000000000000040 Oops: 0000 [#1] SMP PTI CPU: 5 PID: 13840 Comm: kworker/u17:33 Not tainted 5.14.0+ #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) Workqueue: knbd16-recv recv_work [nbd] RIP: 0010:nbd_read_stat.cold+0x130/0x1a4 [nbd] Call Trace: recv_work+0x3b/0xb0 [nbd] process_one_work+0x1ed/0x390 worker_thread+0x4a/0x3d0 kthread+0x12a/0x150 ret_from_fork+0x22/0x30 Fixing it by checking the return value of try_module_get() in nbd_alloc_config(). As nbd_alloc_config() may return ERR_PTR(-ENODEV), assign nbd->config only when nbd_alloc_config() succeeds to ensure the value of nbd->config is binary (valid or NULL). Also adding a debug message to check the reference counter of nbd_config during module removal.
CVE-2022-49307	In the Linux kernel, the following vulnerability has been resolved: tty: synclink_gt: Fix null-pointer-dereference in slgt_clean() When the driver fails at alloc_hdlcdev(), and then we remove the driver module, we will get the following splat: [25.065966] general protection fault, probably for non-canonical address 0xdffffc0000000182: 0000 [#1] PREEMPT SMP KASAN PTI [25.066914] KASAN: null-ptr-deref in range [0x00000000000000c10-0x00000000000000c17] [25.069262] RIP: 0010:detach_hdlc_protocol+0x2a/0x3e0 [25.077709] Call Trace: [25.077924] <TASK> [25.078108] unregister_hdlc_device+0x16/0x30 [25.078481] slgt_cleanup+0x157/0x9f0 [synclink_gt] Fix this by checking whether the 'info->netdev' is a null pointer first.
CVE-2022-49308	In the Linux kernel, the following vulnerability has been resolved: extcon: Modify extcon device to be created after driver data is set Currently, someone can invoke the sysfs such as state_show() intermittently before dev_set_drvdata() is done. And it can be a cause of kernel Oops because of edev is Null at that time. So modified the driver registration to after setting driver data. - Oops's backtrace. Backtrace: [<c067865c>] (state_show) from [<c05222e8>] (dev_attr_show) [<c05222c0>] (dev_attr_show) from [<c02c66e0>] (sysfs_kf_seq_show) [<c02c6648>] (sysfs_kf_seq_show) from [<c02c496c>] (kernfs_seq_show) [<c02c4938>] (kernfs_seq_show) from [<c025e2a0>] (seq_read) [<c025e11c>] (seq_read) from [<c02c50a0>] (kernfs_fop_read) [<c02c5064>] (kernfs_fop_read) from [<c0231cac>] (__vfs_read) [<c0231c5c>] (__vfs_read) from [<c0231ee0>] (vfs_read) [<c0231e34>] (vfs_read) from [<c0232464>] (ksys_read) [<c02323f0>] (ksys_read) from [<c02324fc>] (sys_read) [<c02324e4>] (sys_read) from [<c00091d0>] (__sys_trace_return)
CVE-2022-49314	In the Linux kernel, the following vulnerability has been resolved: tty: Fix a possible resource leak in icom_probe When pci_read_config_dword failed, call pci_release_regions() and pci_disable_device() to recycle the resource previously allocated.
CVE-2022-49321	In the Linux kernel, the following vulnerability has been resolved: xprtrdma: treat all calls not a bcall when bc_serv is NULL When a rdma server returns a fault format reply, nfs v3 client may treats it as a bcall when bc service is not exist. The debug message at rprcdma_bc_receive_call are, [56579.837169] RPC: rprcdma_bc_receive_call: callback XID 00000001, length=20 [56579.837174] RPC: rprcdma_bc_receive_call: 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 04 After that, rprcdma_bc_receive_call will meets NULL pointer as, [226.057890] BUG: unable to handle kernel NULL pointer dereference at 00000000000000c8 ... [226.058704] RIP: 0010:_raw_spin_lock+0xc/0x20 ... [226.059732] Call Trace: [226.059878] rprcdma_bc_receive_call+0x138/0x327 [rprcdma] [226.060011] __ib_process_cq+0x89/0x170 [ib_core] [226.060092] ib_cq_poll_work+0x26/0x80 [ib_core] [226.060257] process_one_work+0x1a7/0x360 [226.060367] ? create_worker+0x1a0/0x1a0 [226.060440] worker_thread+0x30/0x390 [226.060500] ? create_worker+0x1a0/0x1a0 [226.060574] kthread+0x116/0x130 [226.060661] ? kthread_flush_work_fn+0x10/0x10 [226.060724] ret_from_fork+0x35/0x40 ...

CVE ID	Description
CVE-2022-49322	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix sleeping function called from invalid context on RT kernel When setting bootparams="trace_event=initcall:initcall_start tp_printk=1" in the cmdline, the output_printk() was called, and the spin_lock_irqsave() was called in the atomic and irq disable interrupt context situation. On the PREEMPT_RT kernel, these locks are replaced with sleepable rt-spinlock, so the stack calltrace will be triggered. Fix it by raw_spin_lock_irqsave when PREEMPT_RT and "trace_event=initcall:initcall_start tp_printk=1" enabled. BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:46 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1, name: swapper/0 preempt_count: 2, expected: 0 RCU nest depth: 0, expected: 0 Preemption disabled at: [<ffffffff8992303e>] try_to_wake_up+0x7e/0xba0 CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.17.1-rt17+ #19 34c5812404187a875f32bee7977f7367f9679ea7 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-2 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x60/0x8c dump_stack+0x10/0x12 __might_resched.cold+0x11d/0x155 rt_spin_lock+0x40/0x70 trace_event_buffer_commit+0x2fa/0x4c0 ? map_vsyscall+0x93/0x93 trace_event_raw_event_initcall_start+0xbe/0x110 ? perf_trace_initcall_finish+0x210/0x210 ? probe_sched_wakeup+0x34/0x40 ? ttwu_do_wakeup+0xda/0x310 ? trace_hardirqs_on+0x35/0x170 ? map_vsyscall+0x93/0x93 do_one_initcall+0x217/0x3c0 ? trace_event_raw_event_initcall_level+0x170/0x170 ? push_cpu_stop+0x400/0x400 ? cblst_init_generic+0x241/0x290 kernel_init_freeable+0x1ac/0x347 ? _raw_spin_unlock_irq+0x65/0x80 ? rest_init+0xf0/0xf0 kernel_init+0x1e/0x150 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-49323	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: fix possible null-ptr-deref in arm_smmu_device_probe() It will cause null-ptr-deref when using 'res', if platform_get_resource() returns NULL, so move using 'res' after devm_ioremap_resource() that will check it to avoid null-ptr-deref. And use devm_platform_get_and_ioremap_resource() to simplify code.
CVE-2022-49325	In the Linux kernel, the following vulnerability has been resolved: tcp: add accessors to read/set tp->snd_cwnd We had various bugs over the years with code breaking the assumption that tp->snd_cwnd is greater than zero. Lately, syzbot reported the WARN_ON_ONCE(!tp->prior_cwnd) added in commit 8b8a321ff72c ("tcp: fix zero cwnd in tcp_cwnd_reduction") can trigger, and without a repro we would have to spend considerable time finding the bug. Instead of complaining too late, we want to catch where and when tp->snd_cwnd is set to an illegal value.
CVE-2022-49341	In the Linux kernel, the following vulnerability has been resolved: bpf, arm64: Clear prog->jited_len along prog->jited syzbot reported an illegal copy_to_user() attempt from bpf_prog_get_info_by_fd() [1] There was no repro yet on this bug, but I think that commit 0aef499f3172 ("mm/usercopy: Detect vmalloc overruns") is exposing a prior bug in bpf arm64. bpf_prog_get_info_by_fd() looks at prog->jited_len to determine if the JIT image can be copied out to user space. My theory is that syzbot managed to get a prog where prog->jited_len has been set to 43, while prog->bpf_func has been cleared. It is not clear why copy_to_user(uinsns, NULL, ulen) is triggering this particular warning. I thought find_vma_area(NULL) would not find a vm_struct. As we do not hold vmap_area_lock spinlock, it might be possible that the found vm_struct was garbage. [1] usercopy: Kernel memory exposure attempt detected from vmalloc (offset 792633534417210172, size 43)! kernel BUG at mm/usercopy.c:101! Internal error: Oops - BUG: 0 [#1] PREEMPT SMP Modules linked in: CPU: 0 PID: 25002 Comm: syz-executor.1 Not tainted 5.18.0-syzkaller-10139-g8291eaafed36 #0 Hardware name: linux,dummy-virt (DT) pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : usercopy_abort+0x90/0x94 mm/usercopy.c:101 lr : usercopy_abort+0x90/0x94 mm/usercopy.c:89 sp : ffff80000b773a20 x29: ffff80000b773a30 x28: ffff80000b745000 x27: ffff80000b773b48 x26: 0000000000000000 x25: 000000000000002b x24: 0000000000000000 x23: 00000000000000e0 x22: ffff80000b75db67 x21: 0000000000000001 x20: 000000000000002b x19: ffff80000b75db3c x18: 00000000ffffffffff x17: 2820636f6c6c616d x16: 76206d6f72662064 x15: 6574636574656420 x14: 74706d6574746120 x13: 2129333420657a69 x12: 73202c3237313031 x11: 3237313434333533 x10: 3336323937207465 x9: 657275736f707865 x8: ffff80000a30c550 x7: ffff80000b773830 x6: ffff80000b773830 x5: 0000000000000000 x4: ffff00007fbaa10 x3: 0000000000000000 x2: 0000000000000000 x1: f7ff000028fc0000 x0: 0000000000000064 Call trace: usercopy_abort+0x90/0x94 mm/usercopy.c:89 check_heap_object mm/usercopy.c:186 [inline] __check_object_size mm/usercopy.c:252 [inline] __check_object_size+0x198/0x36c mm/usercopy.c:214 check_object_size include/linux/thread_info.h:199 [inline] check_copy_size include/linux/thread_info.h:235 [inline] copy_to_user include/linux/uaccess.h:159 [inline] bpf_prog_get_info_by_fd.isra.0+0xf14/0xfdc kernel/bpf/syscall.c:3993 bpf_obj_get_info_by_fd+0x12c/0x510 kernel/bpf/syscall.c:4253 __sys_bpf+0x900/0x2150 kernel/bpf/syscall.c:4956 __do_sys_bpf kernel/bpf/syscall.c:5021 [inline] __se_sys_bpf kernel/bpf/syscall.c:5019 [inline] __arm64_sys_bpf+0x28/0x40 kernel/bpf/syscall.c:5019 __invoke_syscall arch/arm64/kernel/syscall.c:38 [inline] invoke_syscall+0x48/0x114 arch/arm64/kernel/syscall.c:52 el0_svc common.constprop.0+0x44/0xec arch/arm64/kernel/syscall.c:142 do_el0_svc+0xa0/0xc0 arch/arm64/kernel/syscall.c:206 el0_svc+0x44/0xb0 arch/arm64/kernel/entry-common.c:624 el0t_64_sync_handler+0x1ac/0x1b0 arch/arm64/kernel/entry-common.c:642 el0t_64_sync+0x198/0x19c arch/arm64/kernel/entry.S:581 Code: aa003e3d 000038c0 91248000 97fff65f (d4010000)

CVE ID	Description
CVE-2022-49344	In the Linux kernel, the following vulnerability has been resolved: af_unix: Fix a data-race in unix_dgram_peer_wake_me(). unix_dgram_poll() calls unix_dgram_peer_wake_me() without `other`'s lock held and check if its receive queue is full. Here we need to use unix_recvq_full_lockless() instead of unix_recvq_full(), otherwise KCSAN will report a data-race.
CVE-2022-49345	In the Linux kernel, the following vulnerability has been resolved: net: xfrm: unexport __init-annotated xfrm4_protocol_init() EXPORT_SYMBOL and __init is a bad combination because the .init.text section is freed up after the initialization. Hence, modules cannot use symbols annotated __init. The access to a freed symbol may end up with kernel panic. modpost used to detect it, but it has been broken for a decade. Recently, I fixed modpost so it started to warn it again, then this showed up in linux-next builds. There are two ways to fix it: - Remove __init - Remove EXPORT_SYMBOL I chose the latter for this case because the only in-tree call-site, net/ipv4/xfrm4_policy.c is never compiled as modular. (CONFIG_XFRM is boolean)
CVE-2022-49348	In the Linux kernel, the following vulnerability has been resolved: ext4: filter out EXT4_FC_REPLAY from on-disk superblock field s_state The EXT4_FC_REPLAY bit in sbi->s_mount_state is used to indicate that we are in the middle of replay the fast commit journal. This was actually a mistake, since the sbi->s_mount_info is initialized from es->s_state. Arguably s_mount_state is misleadingly named, but the name is historical --- s_mount_state and s_state dates back to ext2. What should have been used is the ext4_{set,clear,test}_mount_flag() inline functions, which sets EXT4_MF_* bits in sbi->s_mount_flags. The problem with using EXT4_FC_REPLAY is that a maliciously corrupted superblock could result in EXT4_FC_REPLAY getting set in s_mount_state. This bypasses some sanity checks, and this can trigger a BUG() in ext4_es_cache_extent(). As a easy-to-backport-fix, filter out the EXT4_FC_REPLAY bit for now. We should eventually transition away from EXT4_FC_REPLAY to something like EXT4_MF_REPLAY.
CVE-2022-49349	In the Linux kernel, the following vulnerability has been resolved: ext4: fix use-after-free in ext4_rename_dir_prepare We got issue as follows: EXT4-fs (loop0): mounted filesystem without journal. Opts: ,errors=continue ext4_get_first_dir_block: bh->b_data=0xfffff88810bee6000 len=34478 ext4_get_first_dir_block: *parent_de=0xfffff88810bee6ae bh->b_data=0xfffff88810bee6000 ext4_rename_dir_prepare: [1] parent_de=0xfffff88810bee6ae ===== BUG: KASAN: use-after-free in ext4_rename_dir_prepare+0x152/0x220 Read of size 4 at addr ffff88810bee6gae by task rep/1895 CPU: 13 PID: 1895 Comm: rep Not tainted 5.10.0+ #241 Call Trace: dump_stack+0xbe/0xf9 print_address_description.constprop.0+0x1e/0x220 kasan_report.cold+0x37/0x7f ext4_rename_dir_prepare+0x152/0x220 ext4_rename+0xf44/0x1ad0 ext4_rename2+0x11c/0x170 vfs_rename+0xa84/0x1440 do_renameat2+0x683/0x8f0 ___x64_sys_renameat+0x53/0x60 do_syscall_64+0x33/0x40 entry_SYSCALL_64_after_hwframe+0x44/0xa9 RIP: 0033:0x7f45a6fc41c9 RSP: 002b:00007ffc5a470218 EFLAGS: 00000246 ORIG_RAX: 0000000000000108 RAX: ffffffff8fffd8 RBX: 0000000000000000 RCX: 00007f45a6fc41c9 RDX: 0000000000000005 RSI: 0000000020000180 RDI: 0000000000000005 RBP: 00007ffc5a470240 R08: 00007ffc5a470160 R09: 0000000020000080 R10: 00000000200001c0 R11: 0000000000000246 R12: 000000000400bb0 R13: 00007ffc5a470320 R14: 0000000000000000 R15: 0000000000000000 The buggy address belongs to the page: page:00000000440015ce refcount:0 mapcount:0 mapping:0000000000000000 index:0x1 pfn:0x10bee flags: 0x2000000000000000 raw: 0200000000000000 ffffea00043ff4c8 fffffea0004325608 0000000000000000 raw: 0000000000000000 0000000000000000 00000000ffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff88810bee580: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff >ffff88810bee680: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ^ ffff88810bee700: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ffffffff Disabling lock debugging due to kernel taint ext4_rename_dir_prepare: [2] parent_de->ino=3537895424 ext4_rename_dir_prepare: [3] dir=0xfffff888124170140 ext4_rename_dir_prepare: [4] ino=2 ext4_rename_dir_prepare: ent->dir->i_ino=2 parent=-757071872 Reason is first directory entry which 'rec_len' is 34478, then will get illegal parent entry. Now, we do not check directory entry after read directory block in 'ext4_get_first_dir_block'. To solve this issue, check directory entry in 'ext4_get_first_dir_block'. [Trigger an ext4_error() instead of just warning if the directory is missing a '.' or '..' entry. Also make sure we return an error code if the file system is corrupted. -TYT]
CVE-2022-49350	In the Linux kernel, the following vulnerability has been resolved: net: mdio: unexport __init-annotated mdio_bus_init() EXPORT_SYMBOL and __init is a bad combination because the .init.text section is freed up after the initialization. Hence, modules cannot use symbols annotated __init. The access to a freed symbol may end up with kernel panic. modpost used to detect it, but it has been broken for a decade. Recently, I fixed modpost so it started to warn it again, then this showed up in linux-next builds. There are two ways to fix it: - Remove __init - Remove EXPORT_SYMBOL I chose the latter for this case because the only in-tree call-site, drivers/net/phy/phy_device.c is never compiled as modular. (CONFIG_PHYLIB is boolean)

CVE ID	Description
CVE-2022-49352	In the Linux kernel, the following vulnerability has been resolved: ext4: fix warning in ext4_handle_inode_extension We got issue as follows: EXT4-fs error (device loop0) in ext4_reserve_inode_write:5741: Out of memory EXT4-fs error (device loop0): ext4_setattr:5462: inode #13: comm syz-executor.0: mark_inode_dirty error EXT4-fs error (device loop0) in ext4_setattr:5519: Out of memory EXT4-fs error (device loop0): ext4_ind_map_blocks:595: inode #13: comm syz-executor.0: Can't allocate blocks for non-extent mapped inodes with bigalloc ----- [cut here]----- WARNING: CPU: 1 PID: 4361 at fs/ext4/file.c:301 ext4_file_write_iter+0x11c9/0x1220 Modules linked in: CPU: 1 PID: 4361 Comm: syz-executor.0 Not tainted 5.10.0+ #1 RIP: 0010:ext4_file_write_iter+0x11c9/0x1220 RSP: 0018:ffff924d80b27c00 EFLAGS: 00010282 RAX: ffffffff815a3379 RBX: 0000000000000000 RCX: 000000003b000000 RDX: ffff924d81601000 RSI: 000000000000009c RDI: 000000000000009c RBP: 000000000000000d R08: ffffffffbc5a2c6b R09: 0000902e0e52a96f R10: ffff902e2b7c1b40 R11: ffff902e2b7c1b40 R12: 000000000000000a R13: 0000000000000001 R14: ffff902e0e52aa10 R15: ffffffff8bfb8 FS: 00007f81a7f65700(0000) GS:ffff902e3bc80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffff600400 CR3: 000000012db88001 CR4: 0000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: do_iter_readv_writev+0x2e5/0x360 do_iter_write+0x112/0x4c0 do_pwritev+0x1e5/0x390 __x64_sys_pwritev2+0x7e/0xa0 do_syscall_64+0x37/0x50 entry_SYSCALL_64_after_hwframe+0x44/0xa9 Above issue may happen as follows: Assume inode.i_size=4096 EXT4_I(inode)->i_disksize=4096 step 1: set inode->i_size = 8192 ext4_setattr if (attr->ia_size != inode->i_size) EXT4_I(inode)->i_disksize = attr->ia_size; rc = ext4_mark_inode_dirty ext4_reserve_inode_write ext4_get_inode_loc __ext4_get_inode_loc sb_getblk --> return -ENOMEM ... if (!error) ->will not update i_size i_size_write(inode, attr->ia_size); Now: inode.i_size=4096 EXT4_I(inode)->i_disksize=8192 step 2: Direct write 4096 bytes ext4_file_write_iter ext4_dio_write_iter iomap_dio_rw ->return error if (extend) ext4_handle_inode_extension WARN_ON_ONCE(i_size_read(inode) < EXT4_I(inode)->i_disksize); ->Then trigger warning. To solve above issue, if mark inode dirty failed in ext4_setattr just set 'EXT4_I(inode)->i_disksize' with old value.
CVE-2022-49353	In the Linux kernel, the following vulnerability has been resolved: powerpc/papr_scm: don't requests stats with '0' sized stats buffer Sachin reported [1] that on a POWER-10 lpar he is seeing a kernel panic being reported with vPMEM when papr_scm probe is being called. The panic is of the form below and is observed only with following option disabled(profile) for the said LPAR 'Enable Performance Information Collection' in the HMC: Kernel attempted to write user page (1c) - exploit attempt? (uid: 0) BUG: Kernel NULL pointer dereference on write at 0x0000001c Faulting instruction address: 0xc008000001b90844 Oops: Kernel access of bad area, sig: 11 [#1] <snip> NIP [c008000001b90844] drc_pmem_query_stats+0x5c/0x270 [papr_scm] LR [c008000001b92794] papr_scm_probe+0x2ac/0x6ec [papr_scm] Call Trace: 0xc00000000941bca0 (unreliable) papr_scm_probe+0x2ac/0x6ec [papr_scm] platform_probe+0x98/0x150 really_probe+0xfc/0x510 __driver_probe_device+0x17c/0x230 <snip> ---[end trace 0000000000000000]--- Kernel panic - not syncing: Fatal exception On investigation looks like this panic was caused due to a 'stat_buffer' of size==0 being provided to drc_pmem_query_stats() to fetch all performance stats-ids of an NVDIMM. However drc_pmem_query_stats() shouldn't have been called since the vPMEM NVDIMM doesn't support and performance stat-id's. This was caused due to missing check for 'p->stat_buffer_len' at the beginning of papr_scm_pmu_check_events() which indicates that the NVDIMM doesn't support performance-stats. Fix this by introducing the check for 'p->stat_buffer_len' at the beginning of papr_scm_pmu_check_events(). [1] https://lore.kernel.org/all/6B3A522A-6A5F-4CC9-B268-0C63AA6E07D3@linux.ibm.com
CVE-2022-49357	In the Linux kernel, the following vulnerability has been resolved: efi: Do not import certificates from UEFI Secure Boot for T2 Macs On Apple T2 Macs, when Linux attempts to read the db and dbx efi variables at early boot to load UEFI Secure Boot certificates, a page fault occurs in Apple firmware code and EFI runtime services are disabled with the following logs: [Firmware Bug]: Page fault caused by firmware at PA: 0xffffb1edc0068000 WARNING: CPU: 3 PID: 104 at arch/x86/platform/efi/quirks.c:735 efi_crash_gracefully_on_page_fault+0x50/0xf0 (Removed some logs from here) Call Trace: <TASK> page_fault_oops+0x4f/0x2c0 ? search_bpf_extables+0x6b/0x80 ? search_module_extables+0x50/0x80 ? search_exception_tables+0x5b/0x60 kernelmode_fixup_or_oops+0x9e/0x110 __bad_area_nosemaphore+0x155/0x190 bad_area_nosemaphore+0x16/0x20 do_kern_addr_fault+0x8c/0xa0 exc_page_fault+0xd8/0x180 asm_exc_page_fault+0x1e/0x30 (Removed some logs from here) ? __efi_call+0x28/0x30 ? switch_mm+0x20/0x30 ? efi_call_rts+0x19a/0x8e0 ? process_one_work+0x222/0x3f0 ? worker_thread+0x4a/0x3d0 ? kthread+0x17a/0x1a0 ? process_one_work+0x3f0/0x3f0 ? set_kthread_struct+0x40/0x40 ? ret_from_fork+0x22/0x30 </TASK> ---[end trace 1f82023595a5927f]--- efi: Froze efi_rts_wq and disabled EFI Runtime Services integrity: Couldn't get size: 0x8000000000000015 integrity: MODSIGN: Couldn't get UEFI db list efi: EFI Runtime Services are disabled! integrity: Couldn't get size: 0x8000000000000015 integrity: Couldn't get UEFI dbx list integrity: Couldn't get size: 0x8000000000000015 integrity: Couldn't get mokx list integrity: Couldn't get size: 0x80000000 So we avoid reading these UEFI variables and thus prevent the crash.

CVE ID	Description
CVE-2022-49358	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: memleak flow rule from commit path Abort path release flow rule object, however, commit path does not. Update code to destroy these objects before releasing the transaction.
CVE-2022-49370	In the Linux kernel, the following vulnerability has been resolved: firmware: dmi-sysfs: Fix memory leak in dmi_sysfs_register_handle kobject_init_and_add() takes reference even when it fails. According to the doc of kobject_init_and_add() If this function returns an error, kobject_put() must be called to properly clean up the memory associated with the object. Fix this issue by calling kobject_put().
CVE-2022-49371	In the Linux kernel, the following vulnerability has been resolved: driver core: fix deadlock in __device_attach In __device_attach function, The lock holding logic is as follows: ... __device_attach device_lock(dev) // get lock dev async_schedule_dev(__device_attach_async_helper, dev); // func async_schedule_node async_schedule_node_domain(func) entry = kzalloc(sizeof(struct async_entry), GFP_ATOMIC); /* when fail or work limit, sync to execute func, but __device_attach_async_helper will get lock dev as well, which will lead to A-A deadlock. */ if (!entry atomic_read(&entry_count) > MAX_WORK) { func; else queue_work_node(node, system_unbound_wq, &entry->work) device_unlock(dev) As shown above, when it is allowed to do async probes, because of out of memory or work limit, async work is not allowed, to do sync execute instead. it will lead to A-A deadlock because of __device_attach_async_helper getting lock dev. To fix the deadlock, move the async_schedule_dev outside device_lock, as we can see, in async_schedule_node_domain, the parameter of queue_work_node is system_unbound_wq, so it can accept concurrent operations. which will also not change the code logic, and will not lead to deadlock.
CVE-2022-49374	In the Linux kernel, the following vulnerability has been resolved: tipc: check attribute length for bearer name syzbot reported uninit-value: ===== BUG: KMSAN: uninit-value in string_nocheck lib/vsprintf.c:644 [inline] BUG: KMSAN: uninit-value in string+0x4f9/0x6f0 lib/vsprintf.c:725 string_nocheck lib/vsprintf.c:644 [inline] string+0x4f9/0x6f0 lib/vsprintf.c:725 vsnprintf+0x2222/0x3650 lib/vsprintf.c:2806 vprintk_store+0x537/0x2150 kernel/printk/printk.c:2158 vprintk_emit+0x28b/0xab0 kernel/printk/printk.c:2256 vprintk_default+0x86/0xa0 kernel/printk/printk.c:2283 vprintk+0x15f/0x180 kernel/printk/printk_safe.c:50 _printk+0x18d/0x1cf kernel/printk/printk.c:2293 tipc_enable_bearer net/tipc/bearer.c:371 [inline] __tipc_nl_bearer_enable+0x2022/0x22a0 net/tipc/bearer.c:1033 tipc_nl_bearer_enable+0x6c/0xb0 net/tipc/bearer.c:1042 genl_family_rcv_msg_doit net/netlink/genetlink.c:731 [inline] - Do sanity check the attribute length for TIPC_NLA_BEARER_NAME. - Do not use 'illegal name' in printing message.
CVE-2022-49377	In the Linux kernel, the following vulnerability has been resolved: blk-mq: don't touch ->tagset in blk_mq_get_sq_hctx blk_mq_run_hw_queues() could be run when there isn't queued request and after queue is cleaned up, at that time tagset is freed, because tagset lifetime is covered by driver, and often freed after blk_cleanup_queue() returns. So don't touch ->tagset for figuring out current default hctx by the mapping built in request queue, so use-after-free on tagset can be avoided. Meantime this way should be fast than retrieving mapping from tagset.
CVE-2022-49384	In the Linux kernel, the following vulnerability has been resolved: md: fix double free of io_acct_set bioset Now io_acct_set is alloc and free in personality. Remove the codes that free io_acct_set in md_free and md_stop.
CVE-2022-49385	In the Linux kernel, the following vulnerability has been resolved: driver: base: fix UAF when driver_attach failed When driver_attach(drv); failed, the driver_private will be freed. But it has been added to the bus, which caused a UAF. To fix it, we need to delete it from the bus when failed.
CVE-2022-49388	In the Linux kernel, the following vulnerability has been resolved: ubi: ubi_create_volume: Fix use-after-free when volume creation failed There is an use-after-free problem for 'eba_tbl' in ubi_create_volume()'s error handling path: ubi_eba_replace_table(vol, eba_tbl) vol->eba_tbl = tbl out_mapping: ubi_eba_destroy_table(eba_tbl) // Free 'eba_tbl' out_unlock: put_device(&vol->dev) vol_release kfree(tbl->entries) // UAF Fix it by removing redundant 'eba_tbl' releasing. Fetch a reproducer in [Link].

CVE ID	Description
CVE-2022-49390	In the Linux kernel, the following vulnerability has been resolved: macsec: fix UAF bug for real_dev Create a new macsec device but not get reference to real_dev. That can not ensure that real_dev is freed after macsec. That will trigger the UAF bug for real_dev as following: <pre>===== BUG: KASAN: use-after-free in macsec_get_iflink+0x5f/0x70 drivers/net/macsec.c:3662 Call Trace: ... macsec_get_iflink+0x5f/0x70 drivers/net/macsec.c:3662 dev_get_iflink+0x73/0xe0 net/core/ dev.c:637 default_operstate net/core/link_watch.c:42 [inline] rfc2863_policy+0x233/0x2d0 net/ core/link_watch.c:54 linkwatch_do_dev+0x2a/0x150 net/core/link_watch.c:161 Allocated by task 22209: ... alloc_netdev_mqs+0x98/0x1100 net/core/dev.c:10549 rtnl_create_link+0x9d7/0xc00 net/ core/rtnetlink.c:3235 veth_newlink+0x20e/0xa90 drivers/net/veth.c:1748 Freed by task 8: ... kfree +0xd6/0x4d0 mm/slab.c:4552 kvfree+0x42/0x50 mm/util.c:615 device_release+0x9f/0x240 drivers/ base/core.c:2229 kobject_cleanup lib/kobject.c:673 [inline] kobject_release lib/kobject.c:704 [inline] kref_put include/linux/kref.h:65 [inline] kobject_put+0x1c8/0x540 lib/kobject.c:721 netdev_run_todo +0x72e/0x10b0 net/core/dev.c:10327 After commit faab39f63c1f ("net: allow out-of-order netdev unregistration") and commit e5f80fcf869a ("ipv6: give an IPv6 dev to blackhole_netdev"), we can add dev_hold_track() in macsec_dev_init() and dev_put_track() in macsec_free_netdev() to fix the problem.</pre>
CVE-2022-49394	In the Linux kernel, the following vulnerability has been resolved: blk-iolatency: Fix inflight count imbalances and IO hangs on offline iolateness needs to track the number of inflight IOs per cgroup. As this tracking can be expensive, it is disabled when no cgroup has iolateness configured for the device. To ensure that the inflight counters stay balanced, iolateness_set_limit() freezes the request_queue while manipulating the enabled counter, which ensures that no IO is in flight and thus all counters are zero. Unfortunately, iolateness_set_limit() isn't the only place where the enabled counter is manipulated. iolateness_pd_offline() can also dec the counter and trigger disabling. As this disabling happens without freezing the q, this can easily happen while some IOs are in flight and thus leak the counts. This can be easily demonstrated by turning on iolateness on an one empty cgroup while IOs are in flight in other cgroups and then removing the cgroup. Note that iolateness shouldn't have been enabled elsewhere in the system to ensure that removing the cgroup disables iolateness for the whole device. The following keeps flipping on and off iolateness on sda: echo +io > /sys/fs/cgroup/cgroup.subtree_control while true; do mkdir -p /sys/fs/cgroup/test echo '8:0 target=100000' > /sys/fs/cgroup/test/io.latency sleep 1 rmdir /sys/fs/cgroup/test sleep 1 done and there's concurrent fio generating direct rand reads: fio --name test --filename=/dev/sda --direct=1 --rw=randread \ --runtime=600 --time_based --iodepth=256 --numjobs=4 --bs=4k while monitoring with the following drgn script: while True: for css in css_for_each_descendant_pre(prog['blkcg_root'].css.address_of_()): for pos in hlist_for_each(container_of(css, 'struct blkcg', 'css').blkcg_list): blkcg = container_of(pos, 'struct blkcg_gg', 'blkcg_node') pd = blkcg.pd[prog['blkcg_policy_iolateness'].plid] if pd.value_() == 0: continue iolat = container_of(pd, 'struct iolateness_grp', 'pd') inflight = iolat.rq_wait.inflight.counter.value_() if inflight: print(f'inflight={inflight} {disk_name(blkcg.q.disk).decode("utf-8")}') f'{cgroup_path(css.cgroup).decode("utf-8")}') time.sleep(1) The monitoring output looks like the following: inflight=1 sda /user.slice inflight=1 sda /user.slice ... inflight=14 sda /user.slice inflight=13 sda /user.slice inflight=17 sda /user.slice inflight=15 sda /user.slice inflight=18 sda /user.slice inflight=17 sda /user.slice inflight=20 sda /user.slice inflight=19 sda /user.slice <- fio stopped, inflight stuck at 19 inflight=19 sda /user.slice inflight=19 sda /user.slice If a cgroup with stuck inflight ends up getting throttled, the throttled IOs will never get issued as there's no completion event to wake it up leading to an indefinite hang. This patch fixes the bug by unifying enable handling into a work item which is automatically kicked off from iolateness_set_min_lat_nsec() which is called from both iolateness_set_limit() and iolateness_pd_offline() paths. Punting to a work item is necessary as iolateness_pd_offline() is called under spinlocks while freezing a request_queue requires a sleepable context. This also simplifies the code reducing LOC sans the comments and avoids the unnecessary freezes which were happening whenever a cgroup's latency target is newly set or cleared.

CVE ID	Description
CVE-2022-49400	In the Linux kernel, the following vulnerability has been resolved: md: Don't set mddev private to NULL in raid0 pers->free In normal stop process, it does like this: do_md_stop __md_stop (pers->free()); mddev->private=NULL) md_free (free mddev) __md_stop sets mddev->private to NULL after pers->free. The raid device will be stopped and mddev memory is free. But in reshape, it doesn't free the mddev and mddev will still be used in new raid. In reshape, it first sets mddev->private to new_pers and then runs old_pers->free(). Now raid0 sets mddev->private to NULL in raid0_free. The new raid can't work anymore. It will panic when dereference mddev->private because of NULL pointer dereference. It can panic like this: [63010.814972] kernel BUG at drivers/md/raid10.c:928! [63010.819778] invalid opcode: 0000 [#1] PREEMPT SMP NOPTI [63010.825011] CPU: 3 PID: 44437 Comm: md0_resync Kdump: loaded Not tainted 5.14.0-86.el9.x86_64 #1 [63010.833789] Hardware name: Dell Inc. PowerEdge R6415/07YXFK, BIOS 1.15.0 09/11/2020 [63010.841440] RIP: 0010:raise_barrier+0x161/0x170 [raid10] [63010.865508] RSP: 0018:ffffc312408bbc10 EFLAGS: 00010246 [63010.870734] RAX: 0000000000000000 RBX: ffffa00bf7d39800 RCX: 0000000000000000 [63010.877866] RDY: 0000000000000000 RSI: 0000000000000001 RDI: ffffa00bf7d39800 [63010.884999] RBP: 0000000000000000 R08: ffffa4945e74400 R09: 0000000000000000 [63010.892132] R10: ffffa00eed02f798 R11: 0000000000000000 R12: ffffa00bbc435200 [63010.899266] R13: ffffa00bf7d39800 R14: 0000000000000400 R15: 0000000000000003 [63010.906399] FS: 0000000000000000(0000) GS:ffffa00eed000000(0000) knlGS:0000000000000000 [63010.914485] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [63010.920229] CR2: 00007f5cfbe99828 CR3: 0000000105efe000 CR4: 0000000003506e0 [63010.927363] Call Trace: [63010.929822] ? bio_reset+0xe/0x40 [63010.933144] ? raid10_alloc_init_r10buf+0x60/0xa0 [raid10] [63010.938629] raid10_sync_request+0x756/0x1610 [raid10] [63010.943770] md_do_sync.cold+0x3e4/0x94c [63010.947698] md_thread+0xab/0x160 [63010.951024] ? md_write_inc+0x50/0x50 [63010.954688] kthread+0x149/0x170 [63010.957923] ? set_kthread_struct+0x40/0x40 [63010.962107] ret_from_fork+0x22/0x30 Removing the code that sets mddev->private to NULL in raid0 can fix problem.
CVE-2022-49402	In the Linux kernel, the following vulnerability has been resolved: ftrace: Clean up hash direct_functions on register failures We see the following GPF when register_ftrace_direct fails: [] general protection fault, probably for non-canonical address \ 0x2000000000000010: 0000 [#1] PREEMPT SMP DEBUG_PAGEALLOC PTI [...] [] RIP: 0010:ftrace_find_rec_direct+0x53/0x70 [] Code: 48 c1 e0 03 48 03 42 08 48 8b 10 31 c0 48 85 d2 74 [...] [] RSP: 0018:ffff9000138bc10 EFLAGS: 00010206 [] RAX: 0000000000000000 RBX: ffffffff813e0df0 RCX: 000000000000003b [] RDY: 0200000000000000 RSI: 000000000000000c RDI: ffffffff813e0df0 [] RBP: ffffffff813e0df0 R08: ffffffff81180ce0 R09: 0000000000000001 [] R10: ffffc9000138bc18 R11: 0000000000000001 R12: ffffffff813e0df0 [] R13: ffffffff813e0df0 R14: ffff888171b56400 R15: 0000000000000000 [] FS: 00007fa9420c7780(0000) GS:ffff888ff6a00000(0000) knlGS:000000000 [] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [] CR2: 000000000770d000 CR3: 0000000107d50003 CR4: 000000000370ee0 [] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [] Call Trace: [] <TASK> [] register_ftrace_direct+0x54/0x290 [] ? render_sigset_t+0xa0/0xa0 [] bpf_trampoline_update+0x3f5/0x4a0 [] ? 0xffffffa00a3000 [] bpf_trampoline_link_prog+0xa9/0x140 [] bpf_tracing_prog_attach+0x1dc/0x450 [] bpf_raw_tracepoint_open+0x9a/0x1e0 [] ? find_held_lock+0x2d/0x90 [] ? lock_release+0x150/0x430 [] __sys_bpf+0xbd6/0x2700 [] ? lock_is_held_type+0xd8/0x130 [] __x64_sys_bpf+0x1c/0x20 [] do_syscall_64+0x3a/0x80 [] entry_SYSCALL_64_after_hwframe+0x44/0xae [] RIP: 0033:0x7fa9421defa9 [] Code: 00 c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 44 00 00 9 f8 [...] [] RSP: 002b:00007ffed743bd78 EFLAGS: 00000246 ORIG_RAX: 0000000000000141 [] RAX: ffffffff813e0df0 RBX: 00000000069d2480 RCX: 00007fa9421defa9 [] RDY: 0000000000000078 RSI: 00007ffed743bd80 RDI: 0000000000000011 [] RBP: 00007ffed743be00 R08: 0000000000bb7270 R09: 0000000000000000 [] R10: 00000000069da210 R11: 0000000000000246 R12: 0000000000000001 [] R13: 00007ffed743c4b0 R14: 00000000069d2480 R15: 0000000000000001 [] </TASK> [] Modules linked in: klp_vm(OK) [] ---[end trace 0000000000000000]--- One way to trigger this is: 1. load a livepatch that patches kernel function xxx; 2. run bpftrace -e 'kfunc:xxx {}', this will fail (expected for now); 3. repeat #2 => gpf. This is because the entry is added to direct_functions, but not removed. Fix this by remove the entry from direct_functions when register_ftrace_direct fails. Also remove the last trailing space from ftrace.c, so we don't have to worry about it anymore.
CVE-2022-49404	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: Fix potential integer multiplication overflow errors When multiplying of different types, an overflow is possible even when storing the result in a larger type. This is because the conversion is done after the multiplication. So arithmetic overflow and thus incorrect value is possible. Correct an instance of this in the inter packet delay calculation. Fix by ensuring one of the operands is u64 which will promote the other to u64 as well ensuring no overflow.

CVE ID	Description
CVE-2022-49407	In the Linux kernel, the following vulnerability has been resolved: dlm: fix plock invalid read This patch fixes an invalid read showed by KASAN. A unlock will allocate a "struct plock_op" and a followed send_op() will append it to a global send_list data structure. In some cases a followed dev_read() moves it to recv_list and dev_write() will cast it to "struct plock_xop" and access fields which are only available in those structures. At this point an invalid read happens by accessing those fields. To fix this issue the "callback" field is moved to "struct plock_op" to indicate that a cast to "plock_xop" is allowed and does the additional "plock_xop" handling if set. Example of the KASAN output which showed the invalid read: [2064.296453] ===== <pre>[2064.304852] BUG: KASAN: slab-out-of-bounds in dev_write+0x52b/0x5a0 [dlm] [2064.306491] Read of size 8 at addr ffff88800ef227d8 by task dlm_control/7484 [2064.308168] [2064.308575] CPU: 0 PID: 7484 Comm: dlm_control/ Kdump: loaded Not tainted 5.14.0+ #9 [2064.310292] Hardware name: Red Hat KVM, BIOS 0.5.1 01/01/2011 [2064.311618] Call Trace: [2064.312218] dump_stack_lvl+0x56/0x7b [2064.313150] print_address_description.constprop.8+0x21/0x150 [2064.314578] ? dev_write+0x52b/0x5a0 [dlm] [2064.315610] ? dev_write+0x52b/0x5a0 [dlm] [2064.316595] kasan_report.cold.14+0x7f/0x11b [2064.317674] ? dev_write+0x52b/0x5a0 [dlm] [2064.318687] dev_write+0x52b/0x5a0 [dlm] [2064.319629] ? dev_read+0x4a0/0x4a0 [dlm] [2064.320713] ? bpf_lsm_kernfs_init_security+0x10/0x10 [2064.321926] vfs_write+0x17e/0x930 [2064.322769] ? __fget_light+0x1aa/0x220 [2064.323753] ksys_write+0xf1/0x1c0 [2064.324548] ? __ia32_sys_read+0xb0/0xb0 [2064.325464] do_syscall_64+0x3a/0x80 [2064.326387] entry_SYSCALL_64_after_hwframe+0x44/0xae [2064.327606] RIP: 0033:0x7f807e4ba96f [2064.328470] Code: 89 54 24 18 48 89 74 24 10 89 7c 24 08 e8 39 87 f8 ff 48 8b 54 24 18 48 8b 74 24 10 41 89 c0 8b 7c 24 08 b8 01 00 00 0f 05 <48> 3d 00 0f ff 77 31 44 89 c7 48 89 44 24 08 e8 7c 87 f8 ff 48 [2064.332902] RSP: 002b:00007ffd50cfe6e0 EFLAGS: 00000293 ORIG_RAX: 0000000000000001 [2064.334658] RAX: ffffffff8daa0000 RBX: 000055cc3886eb30 RCX: 00007f807e4ba96f [2064.336275] RDX: 0000000000000004 RSI: 00007ffd50cfe7e0 RDI: 0000000000000010 [2064.337980] RBP: 00007ffd50cfe7e0 R08: 0000000000000000 R09: 0000000000000001 [2064.339560] R10: 000055cc3886eb30 R11: 0000000000000293 R12: 000055cc3886eb80 [2064.341237] R13: 000055cc3886eb00 R14: 000055cc3886f590 R15: 0000000000000001 [2064.342857] [2064.343226] Allocated by task 12438: [2064.344057] kasan_save_stack+0x1c/0x40 [2064.345079] __kasan_kmalloc+0x84/0xa0 [2064.345933] kmem_cache_alloc_trace+0x13b/0x220 [2064.346953] dlm_posix_unlock+0xec/0x720 [dlm] [2064.348811] do_lock_file_wait.part.32+0xca/0x1d0 [2064.351070] fcntl_setlk+0x281/0xabc0 [2064.352879] do_fcntl+0x5e4/0xfe0 [2064.354657] __x64_sys_fcntl+0x11f/0x170 [2064.356550] do_syscall_64+0x3a/0x80 [2064.358259] entry_SYSCALL_64_after_hwframe+0x44/0xae [2064.360745] [2064.361511] Last potentially related work creation: [2064.363957] kasan_save_stack +0x1c/0x40 [2064.365811] __kasan_record_aux_stack+0xaf/0xc0 [2064.368100] call_rcu +0x11b/0xf70 [2064.369785] dlm_process_incoming_buffer+0x47d/0xfd0 [dlm] [2064.372404] receive_from_sock+0x290/0x770 [dlm] [2064.374607] process_rcv_sockets+0x32/0x40 [dlm] [2064.377290] process_one_work+0x9a8/0x16e0 [2064.379357] worker_thread+0x87/0xbf0 [2064.381188] kthread+0x3ac/0x490 [2064.383460] ret_from_fork+0x22/0x30 [2064.385588] [2064.386518] Second to last potentially related work creation: [2064.389219] kasan_save_stack +0x1c/0x40 [2064.391043] __kasan_record_aux_stack+0xaf/0xc0 [2064.393303] call_rcu +0x11b/0xf70 [2064.394885] dlm_process_incoming_buffer+0x47d/0xfd0 [dlm] [2064.397694] receive_from_sock+0x290/0x770 ---truncated---</pre>
CVE-2022-49410	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix potential double free in create_var_ref() In create_var_ref(), init_var_ref() is called to initialize the fields of variable ref_field, which is allocated in the previous function call to create_hist_field(). Function init_var_ref() allocates the corresponding fields such as ref_field->system, but frees these fields when the function encounters an error. The caller later calls destroy_hist_field() to conduct error handling, which frees the fields and the variable itself. This results in double free of the fields which are already freed in the previous function. Fix this by storing NULL to the corresponding fields when they are freed in init_var_ref().
CVE-2022-49411	In the Linux kernel, the following vulnerability has been resolved: bfq: Make sure bfqg for which we are queueing requests is online Bios queued into BFQ IO scheduler can be associated with a cgroup that was already offlined. This may then cause insertion of this bfq_group into a service tree. But this bfq_group will get freed as soon as last bio associated with it is completed leading to use after free issues for service tree users. Fix the problem by making sure we always operate on online bfq_group. If the bfq_group associated with the bio is not online, we pick the first online parent.

CVE ID	Description
CVE-2022-49412	In the Linux kernel, the following vulnerability has been resolved: bfq: Avoid merging queues with different parents It can happen that the parent of a bfqq changes between the moment we decide two queues are worth to merge (and set bic->stable_merge_bfqq) and the moment bfq_setup_merge() is called. This can happen e.g. because the process submitted IO for a different cgroup and thus bfqq got reparented. It can even happen that the bfqq we are merging with has parent cgroup that is already offline and going to be destroyed in which case the merge can lead to use-after-free issues such as: BUG: KASAN: use-after-free in __bfq_deactivate_entity+0x9cb/0xa50 Read of size 8 at addr ffff88800693c0c0 by task runc:[2:INIT]/10544 CPU: 0 PID: 10544 Comm: runc:[2:INIT] Tainted: G E 5.15.2-0.g5fb85fd-default #1 openSUSE Tumbleweed (unreleased) f1f3b891c72369aebcd2e43e4641a6358867c70 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a-rebuilt.opensuse.org 04/01/2014 Call Trace: <IRQ> dump_stack_lvl+0x46/0x5a print_address_description.constprop.0+0x1f/0x140 ? __bfq_deactivate_entity+0x9cb/0xa50 kasan_report.cold+0x7f/0x11b ? __bfq_deactivate_entity+0x9cb/0xa50 __bfq_deactivate_entity+0x9cb/0xa50 ? update_curr+0x32f/0x5d0 bfq_deactivate_entity+0xa0/0x1d0 bfq_del_bfqq_busy+0x28a/0x420 ? resched_curr+0x116/0x1d0 ? bfq_requeue_bfqq+0x70/0x70 ? check_preempt_wakeup+0x52b/0xabc0 __bfq_bfqq_expire+0x1a2/0x270 bfq_bfqq_expire+0xd16/0x2160 ? try_to_wake_up+0x4ee/0x1260 ? bfq_end_wr_async_queues+0xe0/0xe0 ? _raw_write_unlock_bh+0x60/0x60 ? _raw_spin_lock_irq+0x81/0xe0 bfq_idle_slice_timer+0x109/0x280 ? bfq_dispatch_request+0x4870/0x4870 __hrtimer_run_queues+0x37d/0x700 ? enqueue_hrtimer+0x1b0/0x1b0 ? kvm_clock_get_cycles+0xd/0x10 ? ktime_get_update_offsets_now+0x6f/0x280 hrtimer_interrupt+0x2c8/0x740 Fix the problem by checking that the parent of the two bfqq's we are merging in bfq_setup_merge() is the same.
CVE-2022-49413	In the Linux kernel, the following vulnerability has been resolved: bfq: Update cgroup information before merging bio When the process is migrated to a different cgroup (or in case of writeback just starts submitting bios associated with a different cgroup) bfq_merge_bio() can operate with stale cgroup information in bic. Thus the bio can be merged to a request from a different cgroup or it can result in merging of bfqq's for different cgroups or bfqq's of already dead cgroups and causing possible use-after-free issues. Fix the problem by updating cgroup information in bfq_merge_bio().
CVE-2022-49414	In the Linux kernel, the following vulnerability has been resolved: ext4: fix race condition between ext4_write and ext4_convert_inline_data Hulk Robot reported a BUG_ON: <pre> ===== EXT4- fs error (device loop3): ext4_mb_generate_buddy:805: group 0, block bitmap and bg descriptor inconsistent: 25 vs 31513 free clusters kernel BUG at fs/ext4/ext4_jbd2.c:53! invalid opcode: 0000 [#1] SMP KASAN PTI CPU: 0 PID: 25371 Comm: syz-executor.3 Not tainted 5.10.0+ #1 RIP: 0010:ext4_put_nojournal fs/ext4/ext4_jbd2.c:53 [inline] RIP: 0010:__ext4_journal_stop +0x10e/0x110 fs/ext4/ext4_jbd2.c:116 [...] Call Trace: ext4_write_inline_data_end+0x59a/0x730 fs/ ext4/inline.c:795 generic_perform_write+0x279/0x3c0 mm/filemap.c:3344 ext4_buffered_write_iter +0x2e3/0x3d0 fs/ext4/file.c:270 ext4_file_write_iter+0x30a/0x11c0 fs/ext4/file.c:520 do_iter_readv_writev+0x339/0x3c0 fs/read_write.c:732 do_iter_write+0x107/0x430 fs/read_write.c:861 vfs_writev fs/read_write.c:934 [inline] do_pwritev+0x1e5/0x380 fs/read_write.c:1031 [...] ===== Above issue may happen as follows: cpu1 cpu2 _____ _____ do_pwritev vfs_writev do_iter_write ext4_file_write_iter ext4_buffered_write_iter generic_perform_write ext4_da_write_begin vfs_fallocate ext4_fallocate ext4_convert_inline_data ext4_convert_inline_data_nolock ext4_destroy_inline_data_nolock clear EXT4_STATE_MAY_INLINE_DATA ext4_map_blocks ext4_ext_map_blocks ext4_mb_new_blocks ext4_mb_regular_allocator ext4_mb_good_group_nolock ext4_mb_init_group ext4_mb_init_cache ext4_mb_generate_buddy --> error ext4_test_inode_state(inode, EXT4_STATE_MAY_INLINE_DATA) ext4_restore_inline_data set EXT4_STATE_MAY_INLINE_DATA ext4_block_write_begin ext4_da_write_end ext4_test_inode_state(inode, EXT4_STATE_MAY_INLINE_DATA) ext4_write_inline_data_end handle=NULL ext4_journal_stop(handle) __ext4_journal_stop ext4_put_nojournal(handle) ref_cnt = (unsigned long)handle BUG_ON(ref_cnt == 0) ---> BUG_ON The lock held by ext4_convert_inline_data is xattr_sem, but the lock held by generic_perform_write is i_rwsem. Therefore, the two locks can be concurrent. To solve above issue, we add inode_lock() for ext4_convert_inline_data(). At the same time, move ext4_convert_inline_data() in front of ext4_punch_hole(), remove similar handling from ext4_punch_hole(). </pre>
CVE-2022-49416	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix use-after-free in chanctx code In ieee80211_vif_use_reserved_context(), when we have an old context and the new context's replace_state is set to IEEE80211_CHANCTX_REPLACE_NONE, we free the old context in ieee80211_vif_use_reserved_reassign(). Therefore, we cannot check the old_ctx anymore, so we should set it to NULL after this point. However, since the new_ctx replace state is clearly not IEEE80211_CHANCTX_REPLACES_OTHER, we're not going to do anything else in this function and can just return to avoid accessing the freed old_ctx.

CVE ID	Description
CVE-2022-49418	In the Linux kernel, the following vulnerability has been resolved: NFSv4: Fix free of uninitialized nfs4_label on referral lookup. Send along the already-allocated fatttr along with nfs4_fs_locations, and drop the memcopy of fatttr. We end up growing two more allocations, but this fixes up a crash as: PID: 790 TASK: ffff88811b43c000 CPU: 0 COMMAND: "ls" #0 [ffffc90000857920] panic at ffffffff81b9bfde #1 [ffffc900008579c0] do_trap at ffffffff81023a9b #2 [ffffc90000857a10] do_error_trap at ffffffff81023b78 #3 [ffffc90000857a58] exc_stack_segment at ffffffff81be1f45 #4 [ffffc90000857a80] asm_exc_stack_segment at ffffffff81c009de #5 [ffffc90000857b08] nfs_lookup at ffffffff810302322 [nfs] #6 [ffffc90000857b70] __lookup_slow at ffffffff813a4a5f #7 [ffffc90000857c60] walk_component at ffffffff813a86c4 #8 [ffffc90000857cb8] path_lookupat at ffffffff813a9553 #9 [ffffc90000857cf0] filename_lookup at ffffffff813ab86b
CVE-2022-49420	In the Linux kernel, the following vulnerability has been resolved: net: annotate races around sk->sk_bound_dev_if UDP sendmsg() is lockless, and reads sk->sk_bound_dev_if while this field can be changed by another thread. Adds minimal annotations to avoid KCSAN splats for UDP. Following patches will add more annotations to potential lockless readers. BUG: KCSAN: data-race in __ip6_datagram_connect / udpv6_sendmsg write to 0xffff888136d47a94 of 4 bytes by task 7681 on cpu 0: __ip6_datagram_connect+0x6e2/0x930 net/ipv6/datagram.c:221 ip6_datagram_connect+0x2a/0x40 net/ipv6/datagram.c:272 inet_dgram_connect+0x107/0x190 net/ipv4/af_inet.c:576 __sys_connect_file net/socket.c:1900 [inline] __sys_connect+0x197/0x1b0 net/socket.c:1917 __do_sys_connect net/socket.c:1927 [inline] __se_sys_connect net/socket.c:1924 [inline] __x64_sys_connect+0x3d/0x50 net/socket.c:1924 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x2b/0x50 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae read to 0xffff888136d47a94 of 4 bytes by task 7670 on cpu 1: udpv6_sendmsg+0xc60/0x16e0 net/ipv6/udp.c:1436 inet6_sendmsg+0x5f/0x80 net/ipv6/af_inet6.c:652 sock_sendmsg_nosec net/socket.c:705 [inline] sock_sendmsg net/socket.c:725 [inline] __sys_sendmsg+0x39a/0x510 net/socket.c:2413 __sys_sendmsg net/socket.c:2467 [inline] __sys_sendmmsg+0x267/0x4c0 net/socket.c:2553 __do_sys_sendmmsg net/socket.c:2582 [inline] __se_sys_sendmmsg net/socket.c:2579 [inline] __x64_sys_sendmmsg+0x53/0x60 net/socket.c:2579 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x2b/0x50 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae value changed: 0x00000000 -> 0xffffffffb Reported by Kernel Concurrency Sanitizer on: CPU: 1 PID: 7670 Comm: syz-executor.3 Tainted: G W 5.18.0-rc1-syzkaller-dirty #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 I chose to not add Fixes: tag because race has minor consequences and stable teams busy enough.
CVE-2022-49422	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix the error handling path in idxd_cdev_register() If a call to alloc_chrdev_region() fails, the already allocated resources are leaking. Add the needed error handling path to fix the leak.
CVE-2022-49432	In the Linux kernel, the following vulnerability has been resolved: powerpc/xics: fix refcount leak in icp_opal_init() The of_find_compatible_node() function returns a node pointer with refcount incremented, use of_node_put() on it when done.
CVE-2022-49434	In the Linux kernel, the following vulnerability has been resolved: PCI: Avoid pci_dev_lock() AB/BA deadlock with sriov_numvfs_store() The sysfs sriov_numvfs_store() path acquires the device lock before the config space access lock: sriov_numvfs_store device_lock # A (1) acquire device lock sriov_configure vfio_pci_sriov_configure # (for example) vfio_pci_core_sriov_configure pci_disable_sriov sriov_disable pci_cfg_access_lock pci_wait_cfg # B (4) wait for dev->block_cfg_access == 0 Previously, pci_dev_lock() acquired the config space access lock before the device lock: pci_dev_lock pci_cfg_access_lock dev->block_cfg_access = 1 # B (2) set dev->block_cfg_access = 1 device_lock # A (3) wait for device lock Any path that uses pci_dev_lock(), e.g., pci_reset_function(), may deadlock with sriov_numvfs_store() if the operations occur in the sequence (1) (2) (3) (4). Avoid the deadlock by reversing the order in pci_dev_lock() so it acquires the device lock before the config space access lock, the same as the sriov_numvfs_store() path. [bhelgaas: combined and adapted commit log from Jay Zhou's independent subsequent posting: https://lore.kernel.org/r/20220404062539.1710-1-jianjay.zhou@huawei.com]
CVE-2022-49436	In the Linux kernel, the following vulnerability has been resolved: powerpc/papr_scm: Fix leaking nvdimmm_events_map elements Right now 'char *' elements allocated for individual 'stat_id' in 'papr_scm_priv.nvdimmm_events_map[]' during papr_scm_pmu_check_events(), get leaked in papr_scm_remove() and papr_scm_pmu_register(), papr_scm_pmu_check_events() error paths. Also individual 'stat_id' arent NULL terminated 'char *' instead they are fixed 8-byte sized identifiers. However papr_scm_pmu_register() assumes it to be a NULL terminated 'char *' and at other places it assumes it to be a 'papr_scm_perf_stat.stat_id' sized string which is 8-bytes in size. Fix this by allocating the memory for papr_scm_priv.nvdimmm_events_map to also include space for 'stat_id' entries. This is possible since number of available events/stat_ids are known upfront. This saves some memory and one extra level of indirection from 'nvdimmm_events_map' to 'stat_id'. Also rest of the code can continue to call 'kfree(papr_scm_priv.nvdimmm_events_map)' without needing to iterate over the array and free up individual elements.

345

CVE ID	Description
CVE-2022-49441	In the Linux kernel, the following vulnerability has been resolved: tty: fix deadlock caused by calling printk() under tty_port->lock pty_write() invokes kmalloc() which may invoke a normal printk() to print failure message. This can cause a deadlock in the scenario reported by syz-bot below: CPU0 CPU1 CPU2 ----- lock(console_owner); lock(&port_lock_key); lock(&port->lock); lock(&port_lock_key); lock(&port->lock); lock(console_owner); As commit dbdda842fe96 ("printk: Add console owner and waiter logic to load balance console writes") said, such deadlock can be prevented by using printk_deferred() in kmalloc() (which is invoked in the section guarded by the port->lock). But there are too many printk() on the kmalloc() path, and kmalloc() can be called from anywhere, so changing printk() to printk_deferred() is too complicated and inelegant. Therefore, this patch chooses to specify __GFP_NOWARN to kmalloc(), so that printk() will not be called, and this deadlock problem can be avoided. Syzbot reported the following lockdep error: ===== WARNING: possible circular locking dependency detected 5.4.143-00237-g08ccc19a-dirty #10 Not tainted ----- syz-executor.4/29420 is trying to acquire lock: ffffffff8aedb2a0 (console_owner){....}-{0:0}, at: console_trylock_spinning kernel/printk/printk.c:1752 [inline] ffffffff8aedb2a0 (console_owner){....}-{0:0}, at: vprintk_emit+0x2ca/0x470 kernel/printk/printk.c:2023 but task is already holding lock: ffff8880119c9158 (&port->lock){-.-}-{2:2}, at: pty_write+0xf4/0x1f0 drivers/tty/pty.c:120 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #2 (&port->lock){-.-}-{2:2}: __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:110 [inline] __raw_spin_lock_irqsave+0x35/0x50 kernel/locking/spinlock.c:159 tty_port_tty_get drivers/tty/tty_port.c:288 [inline] <-- lock(&port->lock); tty_port_default_wakeup+0x1d/0xb0 drivers/tty/tty_port.c:47 serial8250_tx_chars+0x530/0xa80 drivers/tty/serial/8250/8250_port.c:1767 serial8250_handle_irq.part.0+0x31f/0x3d0 drivers/tty/serial/8250/8250_port.c:1854 serial8250_handle_irq drivers/tty/serial/8250/8250_port.c:1827 [inline] <-- lock(&port_lock_key); serial8250_default_handle_irq+0xb2/0x220 drivers/tty/serial/8250/8250_port.c:1870 serial8250_interrupt+0xfd/0x200 drivers/tty/serial/8250/8250_core.c:126 __handle_irq_event_percpu+0x109/0xa50 kernel/irq/handle.c:156 [...] -> #1 (&port_lock_key){-.-}-{2:2}: __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:110 [inline] __raw_spin_lock_irqsave+0x35/0x50 kernel/locking/spinlock.c:159 serial8250_console_write+0x184/0xa40 drivers/tty/serial/8250/8250_port.c:3198 <-- lock(&port_lock_key); call_console_drivers kernel/printk/printk.c:1819 [inline] console_unlock+0x8cb/0xd00 kernel/printk/printk.c:2504 vprintk_emit+0x1b5/0x470 kernel/printk/printk.c:2024 <-- lock(console_owner); vprintk_func+0x8d/0x250 kernel/printk/printk_safe.c:394 printk+0xba/0xed kernel/printk/printk.c:2084 register_console+0x8b3/0xc10 kernel/printk/printk.c:2829 univ8250_console_init+0x3a/0x46 drivers/tty/serial/8250/8250_core.c:681 console_init+0x49d/0x6d3 kernel/printk/printk.c:2915 start_kernel+0x5e9/0x879 init/main.c:713 secondary_startup_64+0xa4/0xb0 arch/x86/kernel/head_64.S:241 -> #0 (console_owner){....}-{0:0}: [...] lock_acquire+0x127/0x340 kernel/locking/lockdep.c:4734 console_trylock_spinning kernel/printk/printk.c:1773 ---truncated---
CVE-2022-49442	In the Linux kernel, the following vulnerability has been resolved: drivers/base/node.c: fix compaction sysfs file leak Compaction sysfs file is created via compaction_register_node in register_node. But we forgot to remove it in unregister_node. Thus compaction sysfs file is leaked. Using compaction_unregister_node to fix this issue.

CVE ID	Description
CVE-2022-49443	In the Linux kernel, the following vulnerability has been resolved: list: fix a data-race around ep->rdllist ep_poll() first calls ep_events_available() with no lock held and checks if ep->rdllist is empty by list_empty_careful(), which reads rdllist->prev. Thus all accesses to it need some protection to avoid store/load-tearing. Note INIT_LIST_HEAD_RCU() already has the annotation for both prev and next. Commit bf3b9f6372c4 ("epoll: Add busy poll support to epoll with socket fds.") added the first lockless ep_events_available(), and commit c5a282e9635e ("fs/epoll: reduce the scope of wq lock in epoll_wait()") made some ep_events_available() calls lockless and added single call under a lock, finally commit e59d3c64cba6 ("epoll: eliminate unnecessary lock for zero timeout") made the last ep_events_available() lockless. BUG: KCSAN: data-race in do_epoll_wait / do_epoll_wait write to 0xffff88810480c7d8 of 8 bytes by task 1802 on cpu 0: INIT_LIST_HEAD include/linux/list.h:38 [inline] list_splice_init include/linux/list.h:492 [inline] ep_start_scan fs/eventpoll.c:622 [inline] ep_send_events fs/eventpoll.c:1656 [inline] ep_poll fs/eventpoll.c:1806 [inline] do_epoll_wait+0x4eb/0xf40 fs/eventpoll.c:2234 do_epoll_pwait fs/eventpoll.c:2268 [inline] __do_sys_epoll_pwait fs/eventpoll.c:2281 [inline] __se_sys_epoll_pwait+0x12b/0x240 fs/eventpoll.c:2275 __x64_sys_epoll_pwait+0x74/0x80 fs/eventpoll.c:2275 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x44/0xd0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae read to 0xffff88810480c7d8 of 8 bytes by task 1799 on cpu 1: list_empty_careful include/linux/list.h:329 [inline] ep_events_available fs/eventpoll.c:381 [inline] ep_poll fs/eventpoll.c:1797 [inline] do_epoll_wait+0x279/0xf40 fs/eventpoll.c:2234 do_epoll_pwait fs/eventpoll.c:2268 [inline] __do_sys_epoll_pwait fs/eventpoll.c:2281 [inline] __se_sys_epoll_pwait+0x12b/0x240 fs/eventpoll.c:2275 __x64_sys_epoll_pwait+0x74/0x80 fs/eventpoll.c:2275 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x44/0xd0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae value changed: 0xffff88810480c7d0 -> 0xffff888103c15098 Reported by Kernel Concurrency Sanitizer on: CPU: 1 PID: 1799 Comm: syz-fuzzer Tainted: G W 5.17.0-rc7-syzkaller-dirty #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011
CVE-2022-49447	In the Linux kernel, the following vulnerability has been resolved: ARM: hisi: Add missing of_node_put after of_find_compatible_node of_find_compatible_node will increment the refcount of the returned device_node. Calling of_node_put() to avoid the refcount leak
CVE-2022-49455	In the Linux kernel, the following vulnerability has been resolved: misc: ocxl: fix possible double free in ocxl_file_register_afu info_release() will be called in device_unregister() when info->dev's reference count is 0. So there is no need to call ocxl_afu_put() and kfree() again. Fix this by adding free_minor() and return to err_unregister error path.
CVE-2022-49456	In the Linux kernel, the following vulnerability has been resolved: bonding: fix missed rcu protection When removing the rcu_read_lock in bond_ethtool_get_ts_info() as discussed [1], I didn't notice it could be called via setsockopt, which doesn't hold rcu lock, as syzbot pointed: stack backtrace: CPU: 0 PID: 3599 Comm: syz-executor317 Not tainted 5.18.0-rc5-syzkaller-01392-g01f4685797a5 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xcd/0x134 lib/dump_stack.c:106 bond_option_active_slave_get_rcu include/net/bonding.h:353 [inline] bond_ethtool_get_ts_info+0x32c/0x3a0 drivers/net/bonding/bond_main.c:5595 __ethtool_get_ts_info+0x173/0x240 net/ethtool/common.c:554 ethtool_get_phc_vclocks +0x99/0x110 net/ethtool/common.c:568 sock_timestamping_bind_phc net/core/sock.c:869 [inline] sock_set_timestamping+0x3a3/0x7e0 net/core/sock.c:916 sock_setsockopt+0x543/0x2ec0 net/core/sock.c:1221 __sys_setsockopt+0x55e/0x6a0 net/socket.c:2223 __do_sys_setsockopt net/socket.c:2238 [inline] __se_sys_setsockopt net/socket.c:2235 [inline] __x64_sys_setsockopt+0xba/0x150 net/socket.c:2235 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f8902c8eb39 Fix it by adding rcu_read_lock and take a ref on the real_dev. Since dev_hold() and dev_put() can take NULL these days, we can skip checking if real_dev exist. [1] https://lore.kernel.org/netdev/27565.1642742439@famine/
CVE-2022-49459	In the Linux kernel, the following vulnerability has been resolved: thermal/drivers/broadcom: Fix potential NULL dereference in sr_thermal_probe platform_get_resource() may return NULL, add proper check to avoid potential NULL dereferencing.

CVE ID	Description
CVE-2022-49465	In the Linux kernel, the following vulnerability has been resolved: blk-throttle: Set BIO_THROTTLED when bio has been throttled 1.In current process, all bio will set the BIO_THROTTLED flag after __blk_throtl_bio(). 2.If bio needs to be throttled, it will start the timer and stop submit bio directly. Bio will submit in blk_throtl_dispatch_work_fn() when the timer expires.But in the current process, if bio is throttled. The BIO_THROTTLED will be set to bio after timer start. If the bio has been completed, it may cause use-after-free blow. BUG: KASAN: use-after-free in blk_throtl_bio+0x12f0/0x2c70 Read of size 2 at addr ffff88801b8902d4 by task fio/26380 dump_stack+0x9b/0xce print_address_description.constprop.6+0x3e/0x60 kasan_report.cold.9+0x22/0x3a blk_throtl_bio+0x12f0/0x2c70 submit_bio_checks+0x701/0x1550 submit_bio_noacct+0x83/0xc80 submit_bio+0xa7/0x330 mpage_readahead+0x380/0x500 read_pages+0x1c1/0xbf0 page_cache_ra_unbounded+0x471/0x6f0 do_page_cache_ra+0xda/0x110 ondemand_readahead+0x442/0xae0 page_cache_async_ra+0x210/0x300 generic_file_buffered_read+0x4d9/0x2130 generic_file_read_iter+0x315/0x490 blkdev_read_iter+0x113/0x1b0 aio_read+0x2ad/0x450 io_submit_one+0xc8e/0x1d60 __se_sys_io_submit+0x125/0x350 do_syscall_64+0x2d/0x40 entry_SYSCALL_64_after_hwframe+0x44/0xa9 Allocated by task 26380: kasan_save_stack+0x19/0x40 __kasan_kmalloc.constprop.2+0xc1/0xd0 kmem_cache_alloc+0x146/0x440 mempool_alloc+0x125/0x2f0 bio_alloc_bioset+0x353/0x590 mpage_alloc+0x3b/0x240 do_mpage_readpage+0xddf/0x1ef0 mpage_readahead+0x264/0x500 read_pages+0x1c1/0xbf0 page_cache_ra_unbounded+0x471/0x6f0 do_page_cache_ra+0xda/0x110 ondemand_readahead+0x442/0xae0 page_cache_async_ra+0x210/0x300 generic_file_buffered_read+0x4d9/0x2130 generic_file_read_iter+0x315/0x490 blkdev_read_iter+0x113/0x1b0 aio_read+0x2ad/0x450 io_submit_one+0xc8e/0x1d60 __se_sys_io_submit+0x125/0x350 do_syscall_64+0x2d/0x40 entry_SYSCALL_64_after_hwframe+0x44/0xa9 Freed by task 0: kasan_save_stack+0x19/0x40 kasan_set_track+0x1c/0x30 kasan_set_free_info+0x1b/0x30 __kasan_slab_free+0x111/0x160 kmem_cache_free+0x94/0x460 mempool_free+0xd6/0x320 bio_free+0xe0/0x130 bio_put+0xab/0xe0 bio_endio+0x3a6/0x5d0 blk_update_request+0x590/0x1370 scsi_end_request+0x7d/0x400 scsi_io_completion+0x1aa/0xe50 scsi_softirq_done+0x11b/0x240 blk_mq_complete_request+0xd4/0x120 scsi_mq_done+0xf0/0x200 virtscsi_vq_done+0xbc/0x150 vring_interrupt+0x179/0x390 __handle_irq_event_percpu+0xf7/0x490 handle_irq_event_percpu+0x7b/0x160 handle_irq_event+0xcc/0x170 handle_edge_irq+0x215/0xb20 common_interrupt+0x60/0x120 asm_common_interrupt+0x1e/0x40 Fix this by move BIO_THROTTLED set into the queue_lock.
CVE-2022-49468	In the Linux kernel, the following vulnerability has been resolved: thermal/core: Fix memory leak in __thermal_cooling_device_register() I got memory leak as follows when doing fault injection test: unreferenced object 0xffff888010080000 (size 264312): comm "182", pid 102533, jiffies 4296434960 (age 10.100s) hex dump (first 32 bytes): 00 00 00 00 ad 4e ad de ff ff ff 00 00 00 00 N..... ff ff ff ff ff ff ff 40 7f 1f b9 ff ff ff ff@..... backtrace: [<0000000038b2f4fc>] kmalloc_order_trace+0x1d/0x110 mm/slab_common.c:969 [<00000000ebcb8da5>] __kmalloc+0x373/0x420 include/linux/slab.h:510 [<0000000084137f13>] thermal_cooling_device_setup_sysfs+0x15d/0x2d0 include/linux/slab.h:586 [<00000000352b8755>] __thermal_cooling_device_register+0x332/0xa60 drivers/thermal/thermal_core.c:927 [<00000000fb9f331b>] devm_thermal_of_cooling_device_register+0x6b/0xf0 drivers/thermal/thermal_core.c:1041 [<000000009b8012d2>] max6650_probe.cold+0x557/0x6aa drivers/hwmon/max6650.c:211 [<00000000da0b7e04>] i2c_device_probe+0x472/0xac0 drivers/i2c/i2c-core-base.c:561 If device_register() fails, thermal_cooling_device_destroy_sysfs() need be called to free the memory allocated in thermal_cooling_device_setup_sysfs().
CVE-2022-49471	In the Linux kernel, the following vulnerability has been resolved: rtw89: cfo: check mac_id to avoid out-of-bounds Somehow, hardware reports incorrect mac_id and pollute memory. Check index before we access the array. UBSAN: array-index-out-of-bounds in rtw89/phy.c:2517:23 index 188 is out of range for type 's32 [64]' CPU: 1 PID: 51550 Comm: irq/35-rtw89_pc Tainted: G OE Call Trace: <IRQ> show_stack+0x52/0x58 dump_stack_lvl+0x4c/0x63 dump_stack+0x10/0x12 ubsan_epilogue+0x9/0x45 __ubsan_handle_out_of_bounds.cold+0x44/0x49 ? __alloc_skb+0x92/0x1d0 rtw89_phy_cfo_parse+0x44/0x7f [rtw89_core] rtw89_core_rx+0x261/0x871 [rtw89_core] ? __alloc_skb+0xee/0x1d0 rtw89_pci_napi_poll+0x3fa/0x4ea [rtw89_pci] __napi_poll+0x33/0x1a0 net_rx_action+0x126/0x260 ? __queue_work+0x217/0x4c0 __do_softirq+0xd9/0x315 ? disable_irq_nosync+0x10/0x10 do_softirq.part.0+0x6d/0x90 </IRQ> <TASK> __local_bh_enable_ip+0x62/0x70 rtw89_pci_interrupt_threadfn+0x182/0x1a6 [rtw89_pci] irq_thread_fn+0x28/0x60 irq_thread+0xc8/0x190 ? irq_thread_fn+0x60/0x60 kthread+0x16b/0x190 ? irq_thread_check_affinity+0xe0/0xe0 ? set_kthread_struct+0x50/0x50 ret_from_fork+0x22/0x30 </TASK>
CVE-2022-49472	In the Linux kernel, the following vulnerability has been resolved: net: phy: micrel: Allow probing without .driver_data Currently, if the .probe element is present in the phy_driver structure and the .driver_data is not, a NULL pointer dereference happens. Allow passing .probe without .driver_data by inserting NULL checks for priv->type.

CVE ID	Description
CVE-2022-49474	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix dangling sco_conn and use-after-free in sco_sock_timeout Connecting the same socket twice consecutively in sco_sock_connect() could lead to a race condition where two sco_conn objects are created but only one is associated with the socket. If the socket is closed before the SCO connection is established, the timer associated with the dangling sco_conn object won't be canceled. As the sock object is being freed, the use-after-free problem happens when the timer callback function sco_sock_timeout() accesses the socket. Here's the call trace: dump_stack+0x107/0x163 ? refcount_inc+0x1c/print_address_description.constprop.0+0x1c/0x47e ? refcount_inc+0x1c/0x7b kasan_report+0x13a/0x173 ? refcount_inc+0x1c/0x7b check_memory_region+0x132/0x139 refcount_inc+0x1c/0x7b sco_sock_timeout+0xb2/0x1ba process_one_work+0x739/0xbd1 ? cancel_delayed_work+0x13f/0x13f ? __raw_spin_lock_init+0xf0/0xf0 ? to_kthread+0x59/0x85 worker_thread+0x593/0x70e kthread+0x346/0x35a ? drain_workqueue+0x31a/0x31a ? kthread_bind+0x4b/0x4b ret_from_fork+0x1f/0x30
CVE-2022-49476	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921: fix kernel crash at mt7921_pci_remove The crash log shown it is possible that mt7921_irq_handler is called while devm_free_irq is being handled so mt76_free_device need to be postponed until devm_free_irq is completed to solve the crash we free the mt76 device too early. [9299.339705] #PF: supervisor read access in kernel mode [9299.339735] #PF: error_code(0x0000) - not-present page [9299.339768] PGD 0 P4D 0 [9299.339786] Oops: 0000 [#1] SMP PTI [9299.339812] CPU: 1 PID: 1624 Comm: prepare-suspend Not tainted 5.15.14-1.fc32.qubes.x86_64 #1 [9299.339863] Hardware name: Xen HVM domU, BIOS 4.14.3 01/20/2022 [9299.339901] RIP: 0010:mt7921_irq_handler+0x1e/0x70 [mt7921e] [9299.340048] RSP: 0018:ffff8a1b80c27cb0 EFLAGS: 00010082 [9299.340081] RAX: 0000000000000000 RBX: ffff98a4cb752020 RCX: ffffffff96211c5 [9299.340123] RDX: 0000000000000000 RSI: 00000000000d4204 RDI: ffff98a4cb752020 [9299.340165] RBP: ffff98a4c28a62a4 R08: ffff98a4c37a96c0 R09: 0000000080150011 [9299.340207] R10: 0000000040000000 R11: 0000000000000000 R12: ffff98a4c4eaa080 [9299.340249] R13: ffff98a4c28a6360 R14: ffff98a4cb752020 R15: ffff98a4c28a6228 [9299.340297] FS: 00007260840d3740(0000) GS:ffff98a4ef700000(0000) knlGS:0000000000000000 [9299.340345] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [9299.340383] CR2: 0000000000000008 CR3: 0000000004c56001 CR4: 0000000000770ee0 [9299.340432] PKRU: 55555554 [9299.340449] Call Trace: [9299.340467] <TASK> [9299.340485] __free_irq+0x221/0x350 [9299.340527] free_irq+0x30/0x70 [9299.340553] devm_free_irq+0x55/0x80 [9299.340579] mt7921_pci_remove+0x2f/0x40 [mt7921e] [9299.340616] pci_device_remove+0x3b/0xa0 [9299.340651] __device_release_driver+0x17a/0x240 [9299.340686] device_driver_detach+0x3c/0xa0 [9299.340714] unbind_store+0x113/0x130 [9299.340740] kernfs_fop_write_iter+0x124/0x1b0 [9299.340775] new_sync_write+0x15c/0x1f0 [9299.340806] vfs_write+0x1d2/0x270 [9299.340831] ksys_write+0x67/0xe0 [9299.340857] do_syscall_64+0x3b/0x90 [9299.340887] entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2022-49478	In the Linux kernel, the following vulnerability has been resolved: media: pvrusb2: fix array-index-out-of-bounds in pvr2_i2c_core_init Syzbot reported that -1 is used as array index. The problem was in missing validation check. hdw->unit_number is initialized with -1 and then if init table walk fails this value remains unchanged. Since code blindly uses this member for array indexing adding sanity check is the easiest fix for that. hdw->workpoll initialization moved upper to prevent warning in __flush_work.
CVE-2022-49479	In the Linux kernel, the following vulnerability has been resolved: mt76: fix tx status related use-after-free race on station removal There is a small race window where ongoing tx activity can lead to a skb getting added to the status tracking idr after that idr has already been cleaned up, which will keep the wcid linked in the status poll list. Fix this by only adding status skbs if the wcid pointer is still assigned in dev->wcid, which gets cleared early by mt76_sta_pre_rcu_remove
CVE-2022-49492	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: fix a NULL pointer dereference in nvme_alloc_admin_tags In nvme_alloc_admin_tags, the admin_q can be set to an error (typically -ENOMEM) if the blk_mq_init_queue call fails to set up the queue, which is checked immediately after the call. However, when we return the error message up the stack, to nvme_reset_work the error takes us to nvme_remove_dead_ctrl() nvme_dev_disable() nvme_suspend_queue(&dev->queues[0]). Here, we only check that the admin_q is non-NULL, rather than not an error or NULL, and begin quiescing a queue that never existed, leading to bad / NULL pointer dereference.
CVE-2022-49493	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt5645: Fix erroneous cleanup order There is a logic error when removing rt5645 device as the function rt5645_i2c_remove() first cancel the &rt5645->jack_detect_work and delete the &rt5645->btn_check_timer later. However, since the timer handler rt5645_btn_check_callback() will re-queue the jack_detect_work, this cleanup order is buggy. That is, once the del_timer_sync in rt5645_i2c_remove is concurrently run with the rt5645_btn_check_callback, the canceled jack_detect_work will be rescheduled again, leading to possible use-after-free. This patch fix the issue by placing the del_timer_sync function before the cancel_delayed_work_sync.

CVE ID	Description
CVE-2022-49497	In the Linux kernel, the following vulnerability has been resolved: net: remove two BUG() from skb_checksum_help() I have a syzbot report that managed to get a crash in skb_checksum_help() If syzbot can trigger these BUG(), it makes sense to replace them with more friendly WARN_ON_ONCE() since skb_checksum_help() can instead return an error code. Note that syzbot will still crash there, until real bug is fixed.
CVE-2022-49498	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Check for null pointer of pointer substream before dereferencing it Pointer substream is being dereferenced on the assignment of pointer card before substream is being null checked with the macro PCM_RUNTIME_CHECK. Although PCM_RUNTIME_CHECK calls BUG_ON, it still is useful to perform the the pointer check before card is assigned.
CVE-2022-49503	In the Linux kernel, the following vulnerability has been resolved: ath9k_htc: fix potential out of bounds access with invalid rxstatus->rs_keyix The "rxstatus->rs_keyix" eventually gets passed to test_bit() so we need to ensure that it is within the bitmap. drivers/net/wireless/ath/ath9k/common.c:46 ath9k_cm_n_rx_accept() error: passing untrusted data 'rx_stats->rs_keyix' to 'test_bit()'
CVE-2022-49504	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Inhibit aborts if external loopback plug is inserted After running a short external loopback test, when the external loopback is removed and a normal cable inserted that is directly connected to a target device, the system oops in the lpfc_set_rrq_active() routine. When the loopback was inserted an FLOGI was transmit. As we're looped back, we receive the FLOGI request. The FLOGI is ABTS'd as we recognize the same wppn thus understand it's a loopback. However, as the ABTS sends address information the port is not set to (fffffe), the ABTS is dropped on the wire. A short 1 frame loopback test is run and completes before the ABTS times out. The loopback is unplugged and the new cable plugged in, and the an FLOGI to the new device occurs and completes. Due to a mixup in ref counting the completion of the new FLOGI releases the fabric ndlp. Then the original ABTS completes and references the released ndlp generating the oops. Correct by no-op'ing the ABTS when in loopback mode (it will be dropped anyway). Added a flag to track the mode to recognize when it should be no-op'd.
CVE-2022-49508	In the Linux kernel, the following vulnerability has been resolved: HID: elan: Fix potential double free in elan_input_configured 'input' is a managed resource allocated with devm_input_allocate_device(), so there is no need to call input_free_device() explicitly or there will be a double free. According to the doc of devm_input_allocate_device(): * Managed input devices do not need to be explicitly unregistered or * freed as it will be done automatically when owner device unbinds from * its driver (or binding fails).
CVE-2022-49516	In the Linux kernel, the following vulnerability has been resolved: ice: always check VF VSI pointer values The ice_get_vf_vsi function can return NULL in some cases, such as if handling messages during a reset where the VSI is being removed and recreated. Several places throughout the driver do not bother to check whether this VSI pointer is valid. Static analysis tools maybe report issues because they detect paths where a potentially NULL pointer could be dereferenced. Fix this by checking the return value of ice_get_vf_vsi everywhere.
CVE-2022-49518	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc3-topology: Correct get_control_data for non bytes payload It is possible to craft a topology where sof_get_control_data() would do out of bounds access because it expects that it is only called when the payload is bytes type. Confusingly it also handles other types of controls, but the payload parsing implementation is only valid for bytes. Fix the code to count the non bytes controls and instead of storing a pointer to sof_abi_hdr in sof_widget_data (which is only valid for bytes), store the pointer to the data itself and add a new member to save the size of the data. In case of non bytes controls we store the pointer to the chanv itself, which is just an array of values at the end. In case of bytes control, drop the wrong cdata->data (wdata[i].pdata) check against NULL since it is incorrect and invalid in this context. The data is pointing to the end of cdata struct, so it should never be null.

CVE ID	Description
CVE-2022-49520	In the Linux kernel, the following vulnerability has been resolved: arm64: compat: Do not treat syscall number as ESR_ELx for a bad syscall If a compat process tries to execute an unknown system call above the <code>__ARM_NR_COMPAT_END</code> number, the kernel sends a SIGILL signal to the offending process. Information about the error is printed to dmesg in <code>compat_arm_syscall()</code> -> <code>arm64_notify_die()</code> -> <code>arm64_force_sig_fault()</code> -> <code>arm64_show_signal()</code>. <code>arm64_show_signal()</code> interprets a non-zero value for <code>current->thread.fault_code</code> as an exception syndrome and displays the message associated with the <code>ESR_ELx.EC</code> field (bits 31:26). <code>current->thread.fault_code</code> is set in <code>compat_arm_syscall()</code> -> <code>arm64_notify_die()</code> with the bad syscall number instead of a valid <code>ESR_ELx</code> value. This means that the <code>ESR_ELx.EC</code> field has the value that the user set for the syscall number and the kernel can end up printing bogus exception messages*. For example, for the syscall number <code>0x68000000</code>, which evaluates to <code>ESR_ELx.EC</code> value of <code>0x1A</code> (<code>ESR_ELx_EC_FPAC</code>) the kernel prints this error: [18.349161] syscall[300]: unhandled exception: ERET/ERETAA/ERETAB, ESR 0x68000000, Oops - bad compat syscall(2) in syscall[10000+50000] [18.350639] CPU: 2 PID: 300 Comm: syscall Not tainted 5.18.0-rc1 #79 [18.351249] Hardware name: Pine64 RockPro64 v2.0 (DT) [...] which is misleading, as the bad compat syscall has nothing to do with pointer authentication. Stop <code>arm64_show_signal()</code> from printing exception syndrome information by having <code>compat_arm_syscall()</code> set the <code>ESR_ELx</code> value to 0, as it has no meaning for an invalid system call number. The example above now becomes: [19.935275] syscall[301]: unhandled exception: Oops - bad compat syscall(2) in syscall[10000+50000] [19.936124] CPU: 1 PID: 301 Comm: syscall Not tainted 5.18.0-rc1-00005-g7e08006d4102 #80 [19.936894] Hardware name: Pine64 RockPro64 v2.0 (DT) [...] which although shows less information because the syscall number, wrongfully advertised as the ESR value, is missing, it is better than showing plainly wrong information. The syscall number can be easily obtained with <code>strace</code>. *A 32-bit value above or equal to <code>0x8000_0000</code> is interpreted as a negative integer in <code>compat_arm_syscall()</code> and the condition <code>scno < __ARM_NR_COMPAT_END</code> evaluates to true; the syscall will exit to userspace in this case with the <code>ENOSYS</code> error code instead of <code>arm64_notify_die()</code> being called.
CVE-2022-49521	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix resource leak in <code>lpfc_sli4_send_seq_to_ulp()</code> If no handler is found in <code>lpfc_complete_unsol_iocb()</code> to match the <code>rcld</code> of a received frame, the frame is dropped and resources are leaked. Fix by returning resources when discarding an unhandled frame type. Update <code>lpfc_fc_frame_check()</code> handling of NOP basic link service.
CVE-2022-49524	In the Linux kernel, the following vulnerability has been resolved: media: pci: cx23885: Fix the error handling in <code>cx23885_initdev()</code> When the driver fails to call the <code>dma_set_mask()</code>, the driver will get the following splat: [55.853884] BUG: KASAN: use-after-free in <code>__process_removed_driver+0x3c/0x240</code> [55.854486] Read of size 8 at addr <code>ffff88810de60408</code> by task <code>modprobe/590</code> [55.856822] Call Trace: [55.860327] <code>__process_removed_driver+0x3c/0x240</code> [55.861347] <code>bus_for_each_dev+0x102/0x160</code> [55.861681] <code>i2c_del_driver+0x2f/0x50</code> This is because the driver has initialized the i2c related resources in <code>cx23885_dev_setup()</code> but not released them in error handling, fix this bug by modifying the error path that jumps after failing to call the <code>dma_set_mask()</code>.

CVE ID	Description
CVE-2022-49526	In the Linux kernel, the following vulnerability has been resolved: md/bitmap: don't set sb values if can't pass sanity check If bitmap area contains invalid data, kernel will crash then mdadm triggers "Segmentation fault". This is cluster-md speical bug. In non-clustered env, mdadm will handle broken metadata case. In clustered array, only kernel space handles bitmap slot info. But even this bug only happened in clustered env, current sanity check is wrong, the code should be changed. How to trigger: (faulty injection) dd if=/dev/zero bs=1M count=1 oflag=direct of=/dev/sda dd if=/dev/zero bs=1M count=1 oflag=direct of=/dev/sdb mdadm -C /dev/md0 -b clustered -e 1.2 -n 2 -l mirror /dev/sda /dev/sdb mdadm -Ss echo aaa > magic.txt == below modifying slot 2 bitmap data == dd if=magic.txt of=/dev/sda seek=16384 bs=1 count=3 <== destroy magic dd if=/dev/zero of=/dev/sda seek=16436 bs=1 count=4 <== ZERO chunksize mdadm -A /dev/md0 /dev/sda /dev/sdb == kernel crashes. mdadm outputs "Segmentation fault" == Reason of kernel crash: In md_bitmap_read_sb (called by md_bitmap_create), bad bitmap magic didn't block chunksize assignment, and zero value made DIV_ROUND_UP_SECTOR_T() trigger "divide error". Crash log: kernel: md: md0 stopped. kernel: md/raid1:md0: not clean -- starting background reconstruction kernel: md/raid1:md0: active with 2 out of 2 mirrors kernel: dlm: kernel: md-cluster: Joined cluster 44810aba-38bb-e6b8-daca-bc97a0b254aa slot 1 kernel: md0: invalid bitmap file superblock: bad magic kernel: md_bitmap_copy_from_slot can't get bitmap from slot 2 kernel: md-cluster: Could not gather bitmaps from slot 2 kernel: divide error: 0000 [#1] SMP NOPTI kernel: CPU: 0 PID: 1603 Comm: mdadm Not tainted 5.14.6-1-default kernel: Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) kernel: RIP: 0010:md_bitmap_create+0x1d1/0x850 [md_mod] kernel: RSP: 0018:ffffc22ac0843ba0 EFLAGS: 00010246 kernel: kernel: Call Trace: kernel: ? dlm_lock_sync+0xd0/0xd0 [md_cluster 77fe..7a0] kernel: md_bitmap_copy_from_slot+0x2c/0x290 [md_mod 24ea..d3a] kernel: load_bitmaps+0xec/0x210 [md_cluster 77fe..7a0] kernel: md_bitmap_load+0x81/0x1e0 [md_mod 24ea..d3a] kernel: do_md_run+0x30/0x100 [md_mod 24ea..d3a] kernel: md_ioctl+0x1290/0x15a0 [md_mod 24ea....d3a] kernel: ? mddev_unlock+0xaa/0x130 [md_mod 24ea..d3a] kernel: ? blkdev_ioctl+0xb1/0x2b0 kernel: block_ioctl+0x3b/0x40 kernel: __x64_sys_ioctl+0x7f/0xb0 kernel: do_syscall_64+0x59/0x80 kernel: ? exit_to_user_mode_prepare+0x1ab/0x230 kernel: ? syscall_exit_to_user_mode+0x18/0x40 kernel: ? do_syscall_64+0x69/0x80 kernel: entry_SYSCALL_64_after_hwframe+0x44/0xae kernel: RIP: 0033:0x7f4a15fa722b kernel: kernel: ---[end trace 8afa7612f559c868]--- kernel: RIP: 0010:md_bitmap_create+0x1d1/0x850 [md_mod]
CVE-2022-49531	In the Linux kernel, the following vulnerability has been resolved: loop: implement ->free_disk Ensure that the lo_device which is stored in the gendisk private data is valid until the gendisk is freed. Currently the loop driver uses a lot of effort to make sure a device is not freed when it is still in use, but to to fix a potential deadlock this will be relaxed a bit soon.
CVE-2022-49532	In the Linux kernel, the following vulnerability has been resolved: drm/virtio: fix NULL pointer dereference in virtio_gpu_conn_get_modes drm_cvt_mode may return NULL and we should check it. This bug is found by syzkaller: FAULT_INJECTION stacktrace: [168.567394] FAULT_INJECTION: forcing a failure. name failslab, interval 1, probability 0, space 0, times 1 [168.567403] CPU: 1 PID: 6425 Comm: syz Kdump: loaded Not tainted 4.19.90-vhulk2201.1.0.h1035.kasan.eulerosv2r10.aarch64 #1 [168.567406] Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 [168.567408] Call trace: [168.567414] dump_backtrace+0x0/0x310 [168.567418] show_stack+0x28/0x38 [168.567423] dump_stack+0xec/0x15c [168.567427] should_fail+0x3ac/0x3d0 [168.567437] __should_failslab+0xb8/0x120 [168.567441] should_failslab+0x28/0xc0 [168.567445] kmem_cache_alloc_trace+0x50/0x640 [168.567454] drm_mode_create+0x40/0x90 [168.567458] drm_cvt_mode+0x48/0xc78 [168.567477] virtio_gpu_conn_get_modes+0xa8/0x140 [virtio_gpu] [168.567485] drm_helper_probe_single_connector_modes+0x3a4/0xd80 [168.567492] drm_mode_getconnector+0x2e0/0xa70 [168.567496] drm_ioctl_kernel+0x11c/0x1d8 [168.567514] drm_ioctl+0x558/0x6d0 [168.567522] do_vfs_ioctl+0x160/0xf30 [168.567525] ksys_ioctl+0x98/0xd8 [168.567530] __arm64_sys_ioctl+0x50/0xc8 [168.567536] el0_svc_common+0xc8/0x320 [168.567540] el0_svc_handler+0xf8/0x160 [168.567544] el0_svc+0x10/0x218 KASAN stacktrace: [168.567561] BUG: KASAN: null-ptr-deref in virtio_gpu_conn_get_modes+0xb4/0x140 [virtio_gpu] [168.567565] Read of size 4 at addr 0000000000000054 by task syz/6425 [168.567566] [168.567571] CPU: 1 PID: 6425 Comm: syz Kdump: loaded Not tainted 4.19.90-vhulk2201.1.0.h1035.kasan.eulerosv2r10.aarch64 #1 [168.567573] Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 [168.567575] Call trace: [168.567578] dump_backtrace+0x0/0x310 [168.567582] show_stack+0x28/0x38 [168.567586] dump_stack+0xec/0x15c [168.567591] kasan_report+0x244/0x2f0 [168.567594] __asan_load4+0x58/0xb0 [168.567607] virtio_gpu_conn_get_modes+0xb4/0x140 [virtio_gpu] [168.567612] drm_helper_probe_single_connector_modes+0x3a4/0xd80 [168.567617] drm_mode_getconnector+0x2e0/0xa70 [168.567621] drm_ioctl_kernel+0x11c/0x1d8 [168.567624] drm_ioctl+0x558/0x6d0 [168.567628] do_vfs_ioctl+0x160/0xf30 [168.567632] ksys_ioctl+0x98/0xd8 [168.567636] __arm64_sys_ioctl+0x50/0xc8 [168.567641] el0_svc_common+0xc8/0x320 [168.567645] el0_svc_handler+0xf8/0x160 [168.567649] el0_svc+0x10/0x218

CVE ID	Description
CVE-2022-49534	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Protect memory leak for NPIV ports sending PLOGI_RJT There is a potential memory leak in <code>lpfc_ignore_els_cmpl()</code> and <code>lpfc_els_rsp_reject()</code> that was allocated from NPIV PLOGI_RJT (<code>lpfc_rcv_plogi()</code> 's <code>login_mbox</code>). Check if <code>cmdioch->context_un.mbox</code> was allocated in <code>lpfc_ignore_els_cmpl()</code> , and then free it back to <code>phba->mbox_mem_pool</code> along with <code>mbox->ctx_buf</code> for service parameters. For <code>lpfc_els_rsp_reject()</code> failure, free both the <code>ctx_buf</code> for service parameters and the <code>login_mbox</code> .
CVE-2022-49535	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix null pointer dereference after failing to issue FLOGI and PLOGI If <code>lpfc_issue_els_flogi()</code> fails and returns non-zero status, the node reference count is decremented to trigger the release of the <code>ndlist</code> structure. However, if there is a prior registration or <code>dev-loss-evt</code> work pending, the node may be released prematurely. When <code>dev-loss-evt</code> completes, the released node is referenced causing a use-after-free null pointer dereference. Similarly, when processing non-zero ELS PLOGI completion status in <code>lpfc_cmpl_els_plogi()</code> , the <code>ndlp</code> flags are checked for a transport registration before triggering node removal. If <code>dev-loss-evt</code> work is pending, the node may be released prematurely and a subsequent call to <code>lpfc_dev_loss_tmo_handler()</code> results in a use after free <code>ndlp</code> dereference. Add test for pending <code>dev-loss</code> before decrementing the node reference count for FLOGI, PLOGI, PRLI, and ADISC handling.
CVE-2022-49536	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix SCSI I/O completion and abort handler deadlock During stress I/O tests with 500+ vports, hard LOCKUP call traces are observed. CPU A: <code>native_queued_spin_lock_slowpath+0x192 _raw_spin_lock_irqsave+0x32 lpfc_handle_fcp_err+0x4c6 lpfc_fcp_io_cmd_wqe_cmpl+0x964 lpfc_sli4_fp_handle_cqe+0x266 __lpfc_sli4_process_cq+0x105 __lpfc_sli4_hba_process_cq+0x3c lpfc_cq_poll_hdlr+0x16 irq_poll_softirq+0x76 __softirqentry_text_start+0xe4 irq_exit+0xf7 do_IRQ+0x7f CPU B: <code>native_queued_spin_lock_slowpath+0x5b _raw_spin_lock+0x1c lpfc_abort_handler+0x13e scmd_eh_abort_handler+0x85 process_one_work+0x1a7 worker_thread+0x30 kthread+0x112 ret_from_fork+0x1f</code> Diagram of lockup: CPUA CPUB ---- ---- <code>lpfc_cmd->buf_lock</code> <code>phba->hbalock</code> <code>lpfc_cmd->buf_lock</code> <code>phba->hbalock</code> Fix by reordering the taking of the <code>lpfc_cmd->buf_lock</code> and <code>phba->hbalock</code> in <code>lpfc_abort_handler</code> routine so that it tries to take the <code>lpfc_cmd->buf_lock</code> first before <code>phba->hbalock</code>.</code>
CVE-2022-49538	In the Linux kernel, the following vulnerability has been resolved: ALSA: jack: Access input_dev under mutex It is possible when using ASoC that <code>input_dev</code> is unregistered while calling <code>snd_jack_report</code> , which causes NULL pointer dereference. In order to prevent this serialize access to <code>input_dev</code> using mutex lock.
CVE-2022-49542	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Move cfg_log_verbose check before calling lpfc_dmp_dbg() In an attempt to log message 0126 with <code>LOG_TRACE_EVENT</code> , the following hard lockup call trace hangs the system. Call Trace: <code>_raw_spin_lock_irqsave+0x32/0x40 lpfc_dmp_dbg.part.32+0x28/0x220 [lpfc] lpfc_cmpl_els_fdisc+0x145/0x460 [lpfc] lpfc_sli_cancel_jobs+0x92/0xd0 [lpfc] lpfc_els_flush_cmd+0x43c/0x670 [lpfc] lpfc_els_flush_all_cmd+0x37/0x60 [lpfc] lpfc_sli4_async_event_proc+0x956/0x1720 [lpfc] lpfc_do_work+0x1485/0x1d70 [lpfc] kthread+0x112/0x130 ret_from_fork+0x1f/0x40 Kernel panic - not syncing: Hard LOCKUP The same CPU tries to claim the <code>phba->port_list_lock</code> twice. Move the <code>cfg_log_verbose</code> checks as part of the <code>lpfc_printf_vlog()</code> and <code>lpfc_printf_log()</code> macros before calling <code>lpfc_dmp_dbg()</code>. There is no need to take the <code>phba->port_list_lock</code> within <code>lpfc_dmp_dbg()</code>.</code>
CVE-2022-49548	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix potential array overflow in bpf_trampoline_get_progs() The <code>cnt</code> value in the ' <code>cnt >= BPF_MAX_TRAMP_PROGS</code> ' check does not include <code>BPF_TRAMP_MODIFY_RETURN</code> bpf programs, so the number of the attached <code>BPF_TRAMP_MODIFY_RETURN</code> bpf programs in a trampoline can exceed <code>BPF_MAX_TRAMP_PROGS</code> . When this happens, the assignment ' <code>*progs++ = aux->prog</code> ' in <code>bpf_trampoline_get_progs()</code> will cause <code>progs</code> array overflow as the <code>progs</code> field in the <code>bpf_trampoline</code> struct can only hold at most <code>BPF_MAX_TRAMP_PROGS</code> bpf programs.
CVE-2022-49549	In the Linux kernel, the following vulnerability has been resolved: x86/MCE/AMD: Fix memory leak when threshold_create_bank() fails In <code>mce_threshold_create_device()</code> , if <code>threshold_create_bank()</code> fails, the previously allocated <code>threshold_banks</code> array <code>@bp</code> will be leaked because the call to <code>mce_threshold_remove_device()</code> will not free it. This happens because <code>mce_threshold_remove_device()</code> fetches the pointer through the <code>threshold_banks</code> per-CPU variable but <code>bp</code> is written there only after the bank creation is successful, and not before, when <code>threshold_create_bank()</code> fails. Add a helper which unwinds all the bank creation work previously done and pass into it the previously allocated <code>threshold_banks</code> array for freeing. [bp: Massage.]

CVE ID	Description
CVE-2022-49552	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix combination of jit blinding and pointers to bpf subprogs. The combination of jit blinding and pointers to bpf subprogs causes: [36.989548] BUG: unable to handle page fault for address: 0000000100000001 [36.990342] #PF: supervisor instruction fetch in kernel mode [36.990968] #PF: error_code(0x0010) - not-present page [36.994859] RIP: 0010:0x100000001 [36.995209] Code: Unable to access opcode bytes at RIP 0xffffffff7. [37.004091] Call Trace: [37.004351] <TASK> [37.004576] ? bpf_loop+0x4d/0x70 [37.004932] ? bpf_prog_3899083f75e4c5de_F+0xe3/0x13b The jit blinding logic didn't recognize that ld_imm64 with an address of bpf subprogram is a special instruction and proceeded to randomize it. By itself it wouldn't have been an issue, but jit_subprogs() logic relies on two step process to JIT all subprogs and then JIT them again when addresses of all subprogs are known. Blinding process in the first JIT phase caused second JIT to miss adjustment of special ld_imm64. Fix this issue by ignoring special ld_imm64 instructions that don't have user controlled constants and shouldn't be blinded.
CVE-2022-49554	In the Linux kernel, the following vulnerability has been resolved: zsmalloc: fix races between asynchronous zspage free and page migration The asynchronous zspage free worker tries to lock a zspage's entire page list without defending against page migration. Since pages which haven't yet been locked can concurrently migrate off the zspage page list while lock_zspage() churns away, lock_zspage() can suffer from a few different lethal races. It can lock a page which no longer belongs to the zspage and unsafely dereference page_private(), it can unsafely dereference a torn pointer to the next page (since there's a data race), and it can observe a spurious NULL pointer to the next page and thus not lock all of the zspage's pages (since a single page migration will reconstruct the entire page list, and create_page_chain() unconditionally zeroes out each list pointer in the process). Fix the races by using migrate_read_lock() in lock_zspage() to synchronize with page migration.
CVE-2022-49556	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Use kzalloc for sev ioct interfaces to prevent kernel data leak For some sev ioct interfaces, the length parameter that is passed maybe less than or equal to SEV_FW_BLOB_MAX_SIZE, but larger than the data that PSP firmware returns. In this case, kmalloc will allocate memory that is the size of the input rather than the size of the data. Since PSP firmware doesn't fully overwrite the allocated buffer, these sev ioct interface may return uninitialized kernel slab memory.
CVE-2022-49558	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: double hook unregistration in netns path __nft_release_hooks() is called from pre_netns exit path which unregisters the hooks, then the NETDEV_UNREGISTER event is triggered which unregisters the hooks again. [565.221461] WARNING: CPU: 18 PID: 193 at net/netfilter/core.c:495 __nf_unregister_net_hook+0x247/0x270 [...] [565.246890] CPU: 18 PID: 193 Comm: kworker/u64:1 Tainted: G E 5.18.0-rc7+ #27 [565.253682] Workqueue: netns cleanup_net [565.257059] RIP: 0010: __nf_unregister_net_hook+0x247/0x270 [...] [565.297120] Call Trace: [565.300900] <TASK> [565.304683] nf_tables_flowtable_event+0x16a/0x220 [nf_tables] [565.308518] raw_notifier_call_chain+0x63/0x80 [565.312386] unregister_netdevice_many+0x54f/0xb50 Unregister and destroy netdev hook from netns pre_exit via kfree_rcu so the NETDEV_UNREGISTER path see unregistered hooks.
CVE-2022-49561	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: re-fetch conntrack after insertion In case the conntrack is clashing, insertion can free skb->nfct and set skb->nfct to the already-confirmed entry. This wasn't found before because the conntrack entry and the extension space used to free'd after an rcu grace period, plus the race needs events enabled to trigger.
CVE-2022-49562	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Use __try_cmpxchg_user() to update guest PTE A/D bits Use the recently introduced __try_cmpxchg_user() to update guest PTE A/D bits instead of mapping the PTE into kernel address space. The VM_PFNMAP path is broken as it assumes that vm_pgoff is the base pfn of the mapped VMA range, which is conceptually wrong as vm_pgoff is the offset relative to the file and has nothing to do with the pfn. The horrific hack worked for the original use case (backing guest memory with /dev/mem), but leads to accessing "random" pfns for pretty much any other VM_PFNMAP case.
CVE-2022-49563	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - add param check for RSA Reject requests with a source buffer that is bigger than the size of the key. This is to prevent a possible integer underflow that might happen when copying the source scatterlist into a linear buffer.
CVE-2022-49564	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - add param check for DH Reject requests with a source buffer that is bigger than the size of the key. This is to prevent a possible integer underflow that might happen when copying the source scatterlist into a linear buffer.

CVE ID	Description
CVE-2022-49565	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/lbr: Fix unchecked MSR access error on HSW The fuzzer triggers the below trace. [7763.384369] unchecked MSR access error: WRMSR to 0x689 (tried to write 0x1fffffff8101349e) at rIP: 0xffffffff810704a4 (native_write_msr+0x4/0x20) [7763.397420] Call Trace: [7763.399881] <TASK> [7763.401994] intel_pmu_lbr_restore+0x9a/0x1f0 [7763.406363] intel_pmu_lbr_sched_task+0x91/0x1c0 [7763.410992] __perf_event_task_sched_in+0x1cd/0x240 On a machine with the LBR format LBR_FORMAT_EIP_FLAGS2, when the TSX is disabled, a TSX quirk is required to access LBR from registers. The lbr_from_signext_quirk_needed() is introduced to determine whether the TSX quirk should be applied. However, the lbr_from_signext_quirk_needed() is invoked before the intel_pmu_lbr_init(), which parses the LBR format information. Without the correct LBR format information, the TSX quirk never be applied. Move the lbr_from_signext_quirk_needed() into the intel_pmu_lbr_init(). Checking x86_pmu.lbr_has_tsx in the lbr_from_signext_quirk_needed() is not required anymore. Both LBR_FORMAT_EIP_FLAGS2 and LBR_FORMAT_INFO have LBR_TSX flag, but only the LBR_FORMAT_EIP_FLAGS2 requires the quirk. Update the comments accordingly.
CVE-2022-49566	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - fix memory leak in RSA When an RSA key represented in form 2 (as defined in PKCS #1 V2.1) is used, some components of the private key persist even after the TFM is released. Replace the explicit calls to free the buffers in qat_rsa_exit_tfm() with a call to qat_rsa_clear_ctx() which frees all buffers referenced in the TFM context.
CVE-2022-49567	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: fix uninit-value in mpol_rebind_policy() mpol_set_nodemask()(mm/mempolicy.c) does not set up nodemask when pol->mode is MPOL_LOCAL. Check pol->mode before access pol->w.cpuset_mems_allowed in mpol_rebind_policy()(mm/mempolicy.c). BUG: KMSAN: uninit-value in mpol_rebind_policy mm/mempolicy.c:352 [inline] BUG: KMSAN: uninit-value in mpol_rebind_task+0x2ac/0x2c0 mm/mempolicy.c:368 mpol_rebind_policy mm/mempolicy.c:352 [inline] mpol_rebind_task+0x2ac/0x2c0 mm/mempolicy.c:368 cpuset_change_task_nodemask kernel/cgroup/cpuset.c:1711 [inline] cpuset_attach+0x787/0x15e0 kernel/cgroup/cpuset.c:2278 cgroup_migrate_execute+0x1023/0x1d20 kernel/cgroup/cgroup.c:2515 cgroup_migrate kernel/cgroup/cgroup.c:2771 [inline] cgroup_attach_task+0x540/0x8b0 kernel/cgroup/cgroup.c:2804 __cgroup1_procs_write+0x5cc/0x7a0 kernel/cgroup/cgroup-v1.c:520 cgroup1_tasks_write+0x94/0xb0 kernel/cgroup/cgroup-v1.c:539 cgroup_file_write+0x4c2/0x9e0 kernel/cgroup/cgroup.c:3852 kernfs_fop_write_iter+0x66a/0x9f0 fs/kernfs/file.c:296 call_write_iter include/linux/fs.h:2162 [inline] new_sync_write fs/read_write.c:503 [inline] vfs_write+0x1318/0x2030 fs/read_write.c:590 ksys_write+0x28b/0x510 fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __x64_sys_write+0xdb/0x120 fs/read_write.c:652 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae Uninit was created at: slab_post_alloc_hook mm/slab.h:524 [inline] slab_alloc_node mm/slub.c:3251 [inline] slab_alloc mm/slub.c:3259 [inline] kmem_cache_alloc+0x902/0x11c0 mm/slub.c:3264 mpol_new mm/mempolicy.c:293 [inline] do_set_mempolicy+0x421/0xb70 mm/mempolicy.c:853 kernel_set_mempolicy mm/mempolicy.c:1504 [inline] __do_sys_set_mempolicy mm/mempolicy.c:1510 [inline] __se_sys_set_mempolicy+0x44c/0xb60 mm/mempolicy.c:1507 __x64_sys_set_mempolicy+0xd8/0x110 mm/mempolicy.c:1507 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x54/0xd0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x44/0xae KMSAN: uninit-value in mpol_rebind_task (2) https://syzkaller.appspot.com/bug?id=d6eb90f952c2a5de9ea718a1b873c55cb13b59dc This patch seems to fix below bug too. KMSAN: uninit-value in mpol_rebind_mm (2) https://syzkaller.appspot.com/bug?id=f2fec0d7013f54ec4162f60743a2b28df40926b The uninit-value is pol->w.cpuset_mems_allowed in mpol_rebind_policy(). When syzkaller reproducer runs to the beginning of mpol_new(), mpol_new() mm/mempolicy.c do_mbind() mm/mempolicy.c kernel_mbind() mm/mempolicy.c `mode` is 1(MPOL_PREFERRED), nodes_empty(*nodes) is `true` and `flags` is 0. Then mode = MPOL_LOCAL; ... policy->mode = mode; policy->flags = flags; will be executed. So in mpol_set_nodemask(), mpol_set_nodemask() mm/mempolicy.c do_mbind() kernel_mbind() pol->mode is 4 (MPOL_LOCAL), that `nodemask` in `pol` is not initialized, which will be accessed in mpol_rebind_policy().
CVE-2022-49571	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_max_reordering. While reading sysctl_tcp_max_reordering, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49572	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_slow_start_after_idle. While reading sysctl_tcp_slow_start_after_idle, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49573	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_early_retrans. While reading sysctl_tcp_early_retrans, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.

CVE ID	Description
CVE-2022-49574	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_recovery. While reading sysctl_tcp_recovery, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49575	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_thin_linear_timeouts. While reading sysctl_tcp_thin_linear_timeouts, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49576	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix data-races around sysctl_fib_multipath_hash_fields. While reading sysctl_fib_multipath_hash_fields, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49577	In the Linux kernel, the following vulnerability has been resolved: udp: Fix a data-race around sysctl_udp_l3mdev_accept. While reading sysctl_udp_l3mdev_accept, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49578	In the Linux kernel, the following vulnerability has been resolved: ip: Fix data-races around sysctl_ip_prot_sock. sysctl_ip_prot_sock is accessed concurrently, and there is always a chance of data-race. So, all readers and writers need some basic protection to avoid load/store-tearing.
CVE-2022-49579	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix data-races around sysctl_fib_multipath_hash_policy. While reading sysctl_fib_multipath_hash_policy, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49580	In the Linux kernel, the following vulnerability has been resolved: ipv4: Fix a data-race around sysctl_fib_multipath_use_neigh. While reading sysctl_fib_multipath_use_neigh, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49581	In the Linux kernel, the following vulnerability has been resolved: be2net: Fix buffer overflow in be_get_module_eeprom be_cmd_read_port_transceiver_data assumes that it is given a buffer that is at least PAGE_DATA_LEN long, or twice that if the module supports SFF 8472. However, this is not always the case. Fix this by passing the desired offset and length to be_cmd_read_port_transceiver_data so that we only copy the bytes once.
CVE-2022-49583	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix handling of dummy receive descriptors Fix memory leak caused by not handling dummy receive descriptor properly. iavf_get_rx_buffer now sets the rx_buffer return value for dummy receive descriptors. Without this patch, when the hardware writes a dummy descriptor, iavf would not free the page allocated for the previous receive buffer. This is an unlikely event but can still happen. [Jesse: massaged commit message]
CVE-2022-49585	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_fastopen_blackhole_timeout. While reading sysctl_tcp_fastopen_blackhole_timeout, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49586	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_fastopen. While reading sysctl_tcp_fastopen, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49587	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_notsent_lowat. While reading sysctl_tcp_notsent_lowat, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49589	In the Linux kernel, the following vulnerability has been resolved: igmp: Fix data-races around sysctl_igmp_qrv. While reading sysctl_igmp_qrv, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers. This test can be packed into a helper, so such changes will be in the follow-up series after net is merged into net-next. qrv ?: READ_ONCE(net->ipv4.sysctl_igmp_qrv);
CVE-2022-49590	In the Linux kernel, the following vulnerability has been resolved: igmp: Fix data-races around sysctl_igmp_llm_reports. While reading sysctl_igmp_llm_reports, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers. This test can be packed into a helper, so such changes will be in the follow-up series after net is merged into net-next. if (ipv4_is_local_multicast(pmc->multiaddr) && !READ_ONCE(net->ipv4.sysctl_igmp_llm_reports))
CVE-2022-49593	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_probe_interval. While reading sysctl_tcp_probe_interval, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49595	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_probe_threshold. While reading sysctl_tcp_probe_threshold, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.

CVE ID	Description
CVE-2022-49596	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_min_snd_mss. While reading sysctl_tcp_min_snd_mss, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49597	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_base_mss. While reading sysctl_tcp_base_mss, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49598	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_mtu_probing. While reading sysctl_tcp_mtu_probing, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49599	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix data-races around sysctl_tcp_l3mdev_accept. While reading sysctl_tcp_l3mdev_accept, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49601	In the Linux kernel, the following vulnerability has been resolved: tcp/dccp: Fix a data-race around sysctl_tcp_fwmark_accept. While reading sysctl_tcp_fwmark_accept, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49602	In the Linux kernel, the following vulnerability has been resolved: ip: Fix a data-race around sysctl_fwmark_reflect. While reading sysctl_fwmark_reflect, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49603	In the Linux kernel, the following vulnerability has been resolved: ip: Fix data-races around sysctl_ip_fwd_update_priority. While reading sysctl_ip_fwd_update_priority, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49604	In the Linux kernel, the following vulnerability has been resolved: ip: Fix data-races around sysctl_ip_fwd_use_pmtu. While reading sysctl_ip_fwd_use_pmtu, it can be changed concurrently. Thus, we need to add READ_ONCE() to its readers.
CVE-2022-49605	In the Linux kernel, the following vulnerability has been resolved: igc: Reinstate IGC_REMOVED logic and implement it properly The initially merged version of the igc driver code (via commit 146740f9abc4, "igc: Add support for PF") contained the following IGC_REMOVED checks in the igc_rd32/wr32() MMIO accessors: u32 igc_rd32(struct igc_hw *hw, u32 reg) { u8 __iomem *hw_addr = READ_ONCE(hw->hw_addr); u32 value = 0; if (IGC_REMOVED(hw_addr)) return ~value; value = readl(&hw_addr[reg]); /* reads should not return all F's */ if (!~value) && (!reg ! (~readl(hw_addr))) hw->hw_addr = NULL; return value; } And: #define wr32(reg, val) \ do { \ u8 __iomem *hw_addr = READ_ONCE((hw->hw_addr); \ if (!IGC_REMOVED(hw_addr)) \ writel((val), &hw_addr[reg]); \ } while (0) E.g. igb has similar checks in its MMIO accessors, and has a similar macro E1000_REMOVED, which is implemented as follows: #define E1000_REMOVED(h) unlikely(! (h)) These checks serve to detect and take note of an 0xffffffff MMIO read return from the device, which can be caused by a PCIe link flap or some other kind of PCI bus error, and to avoid performing MMIO reads and writes from that point onwards. However, the IGC_REMOVED macro was not originally implemented: #ifndef IGC_REMOVED #define IGC_REMOVED(a) (0) #endif /* IGC_REMOVED */ This led to the IGC_REMOVED logic to be removed entirely in a subsequent commit (commit 3c215fb18e70, "igc: remove IGC_REMOVED function"), with the rationale that such checks matter only for virtualization and that igc does not support virtualization -- but a PCIe device can become detached even without virtualization being in use, and without proper checks, a PCIe bus error affecting an igc adapter will lead to various NULL pointer dereferences, as the first access after the error will set hw->hw_addr to NULL, and subsequent accesses will blindly dereference this now-NULL pointer. This patch reinstates the IGC_REMOVED checks in igc_rd32/wr32(), and implements IGC_REMOVED the way it is done for igb, by checking for the unlikely() case of hw_addr being NULL. This change prevents the oopses seen when a PCIe link flap occurs on an igc adapter.
CVE-2022-49606	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix sleep from invalid context BUG Taking the qos_mutex to process RoCEv2 QP's on netdev events causes a kernel splat. Fix this by removing the handling for RoCEv2 in irdma_cm_tearardown_connections that uses the mutex. This handling is only needed for iWARP to avoid having connections established while the link is down or having connections remain functional after the IP address is removed. BUG: sleeping function called from invalid context at kernel/locking/mutex. Call Trace: kernel: dump_stack+0x66/0x90 kernel: __might_sleep.cold.92+0x8d/0x9a kernel: mutex_lock+0x1c/0x40 kernel: irdma_cm_tearardown_connections+0x28e/0x4d0 [irdma] kernel: ? check_preempt_curr+0x7a/0x90 kernel: ? select_idle_sibling+0x22/0x3c0 kernel: ? select_task_rq_fair+0x94c/0xc90 kernel: ? irdma_exec_cqp_cmd+0xc27/0x17c0 [irdma] kernel: ? wake_up_common+0x7a/0x190 kernel: irdma_if_notify+0x3cc/0x450 [irdma] kernel: ? sched_clock_cpu+0xc/0xb0 kernel: irdma_inet6addr_event+0xc6/0x150 [irdma]

CVE ID	Description
CVE-2022-49607	In the Linux kernel, the following vulnerability has been resolved: perf/core: Fix data race between perf_event_set_output() and perf_mmap_close() Yang Jihing reported a race between perf_event_set_output() and perf_mmap_close(): CPU1 CPU2 perf_mmap_close(e2) if (atomic_dec_and_test(&e2->rb->mmap_count)) // 1 - > 0 detach_rest = true ioctl(e1, IOC_SET_OUTPUT, e2) perf_event_set_output(e1, e2) ... list_for_each_entry_rcu(e, &e2->rb->event_list, rb_entry) ring_buffer_attach(e, NULL); // e1 isn't yet added and // therefore not detached ring_buffer_attach(e1, e2->rb) list_add_rcu(&e1->rb_entry, &e2->rb->event_list) After this; e1 is attached to an unmapped rb and a subsequent perf_mmap() will loop forever more: again: mutex_lock(&e->mmap_mutex); if (event->rb) { ... if (!atomic_inc_not_zero(&e->rb->mmap_count)) { ... mutex_unlock(&e->mmap_mutex); goto again; } } The loop in perf_mmap_close() holds e2->mmap_mutex, while the attach in perf_event_set_output() holds e1->mmap_mutex. As such there is no serialization to avoid this race. Change perf_event_set_output() to take both e1->mmap_mutex and e2->mmap_mutex to alleviate that problem. Additionally, have the loop in perf_mmap() detach the rb directly, this avoids having to wait for the concurrent perf_mmap_close() to get around to doing it to make progress.
CVE-2022-49610	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Prevent RSB underflow before vmxenter On VMX, there are some balanced returns between the time the guest's SPEC_CTRL value is written, and the vmxenter. Balanced returns (matched by a preceding call) are usually ok, but it's at least theoretically possible an NMI with a deep call stack could empty the RSB before one of the returns. For maximum paranoia, don't allow *any* returns (balanced or otherwise) between the SPEC_CTRL write and the vmxenter. [bp: Fix 32-bit build.]
CVE-2022-49611	In the Linux kernel, the following vulnerability has been resolved: x86/speculation: Fill RSB on vmxexit for IBRS Prevent RSB underflow/poisoning attacks with RSB. While at it, add a bunch of comments to attempt to document the current state of tribal knowledge about RSB attacks and what exactly is being mitigated.
CVE-2022-49612	In the Linux kernel, the following vulnerability has been resolved: power: supply: core: Fix boundary conditions in interpolation The functions power_supply_temp2resist_simple and power_supply_ocv2cap_simple handle boundary conditions incorrectly. The change was introduced in a4585ba2050f460f749bbaf2b67bd56c41e30283 ("power: supply: core: Use library interpolation"). There are two issues: First, the lines "high = i - 1" and "high = i" in ocv2cap have the wrong order compared to temp2resist. As a consequence, ocv2cap sets high=-1 if ocv>table[0].ocv, which causes an out-of-bounds read. Second, the logic of temp2resist is also not correct. Consider the case table[] = {{20, 100}, {10, 80}, {0, 60}}. For temp=5, we expect a resistance of 70% by interpolation. However, temp2resist sets high=low=2 and returns 60.
CVE-2022-49613	In the Linux kernel, the following vulnerability has been resolved: serial: 8250: Fix PM usage_count for console handover When console is enabled, univ8250_console_setup() calls serial8250_console_setup() before .dev is set to uart_port. Therefore, it will not call pm_runtime_get_sync(). Later, when the actual driver is going to take over univ8250_console_exit() is called. As .dev is already set, serial8250_console_exit() makes pm_runtime_put_sync() call with usage count being zero triggering PM usage count warning (extra debug for univ8250_console_setup(), univ8250_console_exit(), and serial8250_register_ports()): [0.068987] univ8250_console_setup ttyS0 nodev [0.499670] printk: console [ttyS0] enabled [0.717955] printk: console [ttyS0] printing thread started [1.960163] serial8250_register_ports assigned dev for ttyS0 [1.976830] printk: console [ttyS0] disabled [1.976888] printk: console [ttyS0] printing thread stopped [1.977073] univ8250_console_exit ttyS0 usage:0 [1.977075] serial8250 serial8250: Runtime PM usage count underflow! [1.977429] dw-apb-uart.6: ttyS0 at MMIO 0x4010006000 (irq = 33, base_baud = 115200) is a 16550A [1.977812] univ8250_console_setup ttyS0 usage:2 [1.978167] printk: console [ttyS0] printing thread started [1.978203] printk: console [ttyS0] enabled To fix the issue, call pm_runtime_get_sync() in serial8250_register_ports() as soon as .dev is set for an uart_port if it has console enabled. This problem became apparent only recently because 82586a721595 ("PM: runtime: Avoid device usage count underflows") added the warning printout. I confirmed this problem also occurs with v5.18 (w/o the warning printout, obviously).
CVE-2022-49615	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt711-sdca: fix kernel NULL pointer dereference when IO error The initial settings will be written before the codec probe function. But, the rt711->component doesn't be assigned yet. If IO error happened during initial settings operations, it will cause the kernel panic. This patch changed component->dev to slave->dev to fix this issue.
CVE-2022-49616	In the Linux kernel, the following vulnerability has been resolved: ASoC: rt7*-sdw: harden jack_detect_handler Realtek headset codec drivers typically check if the card is instantiated before proceeding with the jack detection. The rt700, rt711 and rt711-sdca are however missing a check on the card pointer, which can lead to NULL dereferences encountered in driver bind/unbind tests.

CVE ID	Description
CVE-2022-49617	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: sof_sdw: handle errors on card registration If the card registration fails, typically because of deferred probes, the device properties added for headset codecs are not removed, which leads to kernel oopses in driver bind/unbind tests. We already clean-up the device properties when the card is removed, this code can be moved as a helper and called upon card registration errors.
CVE-2022-49620	In the Linux kernel, the following vulnerability has been resolved: net: tipc: fix possible refcount leak in tipc_sk_create() Free sk in case tipc_sk_insert() fails.
CVE-2022-49622	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: avoid skb access on nf_stolen When verdict is NF_STOLEN, the skb might have been freed. When tracing is enabled, this can result in a use-after-free: 1. access to skb->nf_trace 2. access to skb->mark 3. computation of trace id 4. dump of packet payload To avoid 1, keep a cached copy of skb->nf_trace in the trace state struct. Refresh this copy whenever verdict is != STOLEN. Avoid 2 by skipping skb->mark access if verdict is STOLEN. 3 is avoided by precomputing the trace id. Only dump the packet when verdict is not "STOLEN".
CVE-2022-49623	In the Linux kernel, the following vulnerability has been resolved: powerpc/xive/spapr: correct bitmap allocation size kasan detects access beyond the end of the xibm->bitmap allocation: BUG: KASAN: slab-out-of-bounds in _find_first_zero_bit+0x40/0x140 Read of size 8 at addr c00000001d1d0118 by task swapper/0/1 CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.19.0-rc2-00001-g90df023b36dd #28 Call Trace: [c00000001d98f770] [c0000000012baab8] dump_stack_lvl+0xac/0x108 (unreliable) [c00000001d98f7b0] [c00000000068faac] print_report+0x37c/0x710 [c00000001d98f880] [c0000000006902c0] kasan_report+0x110/0x354 [c00000001d98f950] [c000000000692324] __asan_load8+0xa4/0xe0 [c00000001d98f970] [c0000000011c6ed0] _find_first_zero_bit+0x40/0x140 [c00000001d98f9b0] [c0000000000dbfbc] xive_spapr_get_ipi+0xcc/0x260 [c00000001d98fa70] [c0000000000d6d28] xive_setup_cpu_ipi+0x1e8/0x450 [c00000001d98fb30] [c000000004032a20] pSeries_smp_probe+0x5c/0x118 [c00000001d98fb60] [c000000004018b44] smp_prepare_cpus+0x944/0x9ac [c00000001d98fc90] [c000000004009f9c] kernel_init_freeable+0x2d4/0x640 [c00000001d98fd90] [c0000000000131e8] kernel_init+0x28/0x1d0 [c00000001d98fe10] [c0000000000cd54] ret_from_kernel_thread+0x5c/0x64 Allocated by task 0: kasan_save_stack+0x34/0x70 __kasan_kmalloc+0xb4/0xf0 __kmalloc+0x268/0x540 xive_spapr_init+0x4d0/0x77c pseries_init_irq+0x40/0x27c init_IRQ+0x44/0x84 start_kernel+0x2a4/0x538 start_here_common+0x1c/0x20 The buggy address belongs to the object at c00000001d1d0118 which belongs to the cache kmalloc-8 of size 8 The buggy address is located 0 bytes inside of 8-byte region [c00000001d1d0118, c00000001d1d0120) The buggy address belongs to the physical page: page:c00c000000074740 refcount:1 mapcount:0 mapping:0000000000000000 index:0xc00000001d1d0558 pfn:0x1d1d flags: 0x7ffff000000200(slab node=0 zone=0 lastcpupid=0x7ffff) raw: 007ffff000000200 c00000001d0003c8 c00000001d0003c8 c00000001d010480 raw: c00000001d1d0558 000000001e1000a 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: c00000001d1d0000: fc 00 fc fc fc fc fc fc fc fc fc fc fc fc fc c00000001d1d0080: fc fc 00 fc fc fc fc fc fc fc fc fc fc fc fc fc >c00000001d1d0100: fc fc fc 02 fc fc fc fc fc fc fc fc fc fc fc fc fc ^ c00000001d1d0180: fc fc fc fc 04 fc fc fc fc fc fc fc fc fc fc c00000001d1d0200: fc fc fc fc 04 fc fc fc fc fc fc fc fc fc fc This happens because the allocation uses the wrong unit (bits) when it should pass (BITS_TO_LONGS(count) * sizeof(long)) or equivalent. With small numbers of bits, the allocated object can be smaller than sizeof(long), which results in invalid accesses. Use bitmap_zalloc() to allocate and initialize the irq bitmap, paired with bitmap_free() for consistency.

CVE ID	Description
CVE-2022-49624	In the Linux kernel, the following vulnerability has been resolved: net: atlantic: remove aq_nic_deinit() when resume aq_nic_deinit() has been called while suspending, so we don't have to call it again on resume. Actually, call it again leads to another hang issue when resuming from S3. Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992345] Call Trace: Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992346] <TASK> Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992348] aq_nic_deinit+0xb4/0xd0 [atlantic] Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992356] aq_pm_thaw+0x7f/0x100 [atlantic] Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992362] pci_pm_resume+0x5c/0x90 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992366] ? pci_pm_thaw+0x80/0x80 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992368] dpm_run_callback+0x4e/0x120 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992371] device_resume+0xad/0x200 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992373] async_resume+0x1e/0x40 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992374] async_run_entry_fn+0x33/0x120 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992377] process_one_work+0x220/0x3c0 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992380] worker_thread+0x4d/0x3f0 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992382] ? process_one_work+0x3c0/0x3c0 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992384] kthread+0x12a/0x150 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992386] ? set_kthread_struct+0x40/0x40 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992387] ret_from_fork+0x22/0x30 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992391] </TASK> Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992392] ---[end trace 1ec8c79604ed5e0d]--- Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992394] PM: dpm_run_callback(): pci_pm_resume+0x0/0x90 returns -110 Jul 8 03:09:44 u-Precision-7865-Tower kernel: [5910.992397] atlantic 0000:02:00.0: PM: failed to resume async: error -110
CVE-2022-49625	In the Linux kernel, the following vulnerability has been resolved: sfc: fix kernel panic when creating VF When creating VFs a kernel panic can happen when calling to efx_ef10_try_update_nic_stats_vf. When releasing a DMA coherent buffer, sometimes, I don't know in what specific circumstances, it has to unmap memory with vunmap. It is disallowed to do that in IRQ context or with BH disabled. Otherwise, we hit this line in vunmap, causing the crash: BUG_ON(in_interrupt()); This patch reenables BH to release the buffer. Log messages when the bug is hit: kernel BUG at mm/vmalloc.c:2727! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 6 PID: 1462 Comm: NetworkManager Kdump: loaded Tainted: G I ----- 5.14.0-119.el9.x86_64 #1 Hardware name: Dell Inc. PowerEdge R740/06WXJT, BIOS 2.8.2 08/27/2020 RIP: 0010:vunmap+0x2e/0x30 ...skip... Call Trace: __iommu_dma_free+0x96/0x100 efx_nic_free_buffer+0x2b/0x40 [sfc] efx_ef10_try_update_nic_stats_vf+0x14a/0x1c0 [sfc] efx_ef10_update_stats_vf+0x18/0x40 [sfc] efx_start_all+0x15e/0x1d0 [sfc] efx_net_open+0x5a/0xe0 [sfc] __dev_open+0xe7/0x1a0 __dev_change_flags+0x1d7/0x240 dev_change_flags+0x21/0x60 ...skip...
CVE-2022-49626	In the Linux kernel, the following vulnerability has been resolved: sfc: fix use after free when disabling sriov Use after free is detected by kfence when disabling sriov. What was read after being freed was vf->pci_dev: it was freed from pci_disable_sriov and later read in efx_ef10_sriov_free_vf_vports, called from efx_ef10_sriov_free_vf_vswitching. Set the pointer to NULL at release time to not trying to read it later. Reproducer and dmesg log (note that kfence doesn't detect it every time): \$ echo 1 > /sys/class/net/enp65s0f0np0/device/sriov_numvfs \$ echo 0 > /sys/class/net/enp65s0f0np0/device/sriov_numvfs BUG: KFENCE: use-after-free read in efx_ef10_sriov_free_vf_vswitching+0x82/0x170 [sfc] Use-after-free read at 0x00000000ff3c1ba5 (in kfence-#224): efx_ef10_sriov_free_vf_vswitching+0x82/0x170 [sfc] efx_ef10_pci_sriov_disable+0x38/0x70 [sfc] efx_pci_sriov_configure+0x24/0x40 [sfc] sriov_numvfs_store+0xfe/0x140 kernfs_fop_write_iter+0x11c/0x1b0 new_sync_write+0x11f/0x1b0 vfs_write+0x1eb/0x280 ksys_write+0x5f/0xe0 do_syscall_64+0x5c/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae kfence-#224: 0x00000000edb8ef95-0x00000000671f5ce1, size=2792, cache=kmalloc-4k allocated by task 6771 on cpu 10 at 3137.860196s: pci_alloc_dev+0x21/0x60 pci_iiov_add_virtfn+0x2a2/0x320 sriov_enable+0x212/0x3e0 efx_ef10_sriov_configure+0x67/0x80 [sfc] efx_pci_sriov_configure+0x24/0x40 [sfc] sriov_numvfs_store+0xba/0x140 kernfs_fop_write_iter+0x11c/0x1b0 new_sync_write+0x11f/0x1b0 vfs_write+0x1eb/0x280 ksys_write+0x5f/0xe0 do_syscall_64+0x5c/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae freed by task 6771 on cpu 12 at 3170.991309s: device_release+0x34/0x90 kobject_cleanup+0x3a/0x130 pci_iiov_remove_virtfn+0xd9/0x120 sriov_disable+0x30/0xe0 efx_ef10_pci_sriov_disable+0x57/0x70 [sfc] efx_pci_sriov_configure+0x24/0x40 [sfc] sriov_numvfs_store+0xfe/0x140 kernfs_fop_write_iter+0x11c/0x1b0 new_sync_write+0x11f/0x1b0 vfs_write+0x1eb/0x280 ksys_write+0x5f/0xe0 do_syscall_64+0x5c/0x80 entry_SYSCALL_64_after_hwframe+0x44/0xae
CVE-2022-49630	In the Linux kernel, the following vulnerability has been resolved: tcp: Fix a data-race around sysctl_tcp_ecn_fallback. While reading sysctl_tcp_ecn_fallback, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.
CVE-2022-49632	In the Linux kernel, the following vulnerability has been resolved: icmp: Fix a data-race around sysctl_icmp_errors_use_inbound_ifaddr. While reading sysctl_icmp_errors_use_inbound_ifaddr, it can be changed concurrently. Thus, we need to add READ_ONCE() to its reader.

CVE ID	Description
CVE-2022-49636	In the Linux kernel, the following vulnerability has been resolved: vlan: fix memory leak in vlan_newlink() Blamed commit added back a bug I fixed in commit 9bbd917e0bec ("vlan: fix memory leak in vlan_dev_set_egress_priority") If a memory allocation fails in vlan_changelink() after other allocations succeeded, we need to call vlan_dev_free_egress_priority() to free all allocated memory because after a failed ->newlink() we do not call any methods like ndo_uninit() or dev->priv_destructor(). In following example, if the allocation for last element 2000:2001 fails, we need to free eight prior allocations: ip link add link dummy0 dummy0.100 type vlan id 100 \egress-qos-map 1:2 2:3 3:4 4:5 5:6 6:7 7:8 8:9 2000:2001 syzbot report was: BUG: memory leak unreferenced object 0xffff888117bd1060 (size 32): comm "syz-executor408", pid 3759, jiffies 4294956555 (age 34.090s) hex dump (first 32 bytes): 09 00 00 00 00 a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<ffffff83fc60ad>] kmalloc include/linux/slab.h:600 [inline] [<ffffff83fc60ad>] vlan_dev_set_egress_priority+0xed/0x170 net/8021q/vlan_dev.c:193 [<ffffff83fc6628>] vlan_changelink+0x178/0x1d0 net/8021q/vlan_netlink.c:128 [<ffffff83fc67c8>] vlan_newlink+0x148/0x260 net/8021q/vlan_netlink.c:185 [<ffffff838b1278>] rtnl_newlink_create net/core/rtnetlink.c:3363 [inline] [<ffffff838b1278>] __rtnl_newlink+0xa58/0xdc0 net/core/rtnetlink.c:3580 [<ffffff838b1629>] rtnl_newlink+0x49/0x70 net/core/rtnetlink.c:3593 [<ffffff838ac66c>] rtnetlink_rcv_msg+0x21c/0x5c0 net/core/rtnetlink.c:6089 [<ffffff839f9c37>] netlink_rcv_skb+0x87/0x1d0 net/netlink/af_netlink.c:2501 [<ffffff839f8da7>] netlink_unicast_kernel net/netlink/af_netlink.c:1319 [inline] [<ffffff839f8da7>] netlink_unicast+0x397/0x4c0 net/netlink/af_netlink.c:1345 [<ffffff839f9266>] netlink_sendmsg+0x396/0x710 net/netlink/af_netlink.c:1921 [<ffffff8384dbf6>] sock_sendmsg_nosec net/socket.c:714 [inline] [<ffffff8384dbf6>] sock_sendmsg+0x56/0x80 net/socket.c:734 [<ffffff8384e15c>] ____sys_sendmsg+0x36c/0x390 net/socket.c:2488 [<ffffff838523cb>] ____sys_sendmsg+0x8b/0xd0 net/socket.c:2542 [<ffffff838525b8>] __sys_sendmsg net/socket.c:2571 [inline] [<ffffff838525b8>] __do_sys_sendmsg net/socket.c:2580 [inline] [<ffffff838525b8>] __se_sys_sendmsg net/socket.c:2578 [inline] [<ffffff838525b8>] __x64_sys_sendmsg+0x78/0xf0 net/socket.c:2578 [<ffffff845ad8d5>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<ffffff845ad8d5>] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 [<ffffff8460006a>] entry_SYSCALL_64_after_hwframe+0x46/0xb0
CVE-2022-49638	In the Linux kernel, the following vulnerability has been resolved: icmp: Fix data-races around sysctl. While reading icmp sysctl variables, they can be changed concurrently. So, we need to add READ_ONCE() to avoid data-races.
CVE-2022-49639	In the Linux kernel, the following vulnerability has been resolved: cipso: Fix data-races around sysctl. While reading cipso sysctl variables, they can be changed concurrently. So, we need to add READ_ONCE() to avoid data-races.
CVE-2022-49640	In the Linux kernel, the following vulnerability has been resolved: sysctl: Fix data races in proc_douintvec_minmax(). A sysctl variable is accessed concurrently, and there is always a chance of data-race. So, all readers and writers need some basic protection to avoid load/store-tearing. This patch changes proc_douintvec_minmax() to use READ_ONCE() and WRITE_ONCE() internally to fix data-races on the sysctl side. For now, proc_douintvec_minmax() itself is tolerant to a data-race, but we still need to add annotations on the other subsystem's side.
CVE-2022-49641	In the Linux kernel, the following vulnerability has been resolved: sysctl: Fix data races in proc_douintvec(). A sysctl variable is accessed concurrently, and there is always a chance of data-race. So, all readers and writers need some basic protection to avoid load/store-tearing. This patch changes proc_douintvec() to use READ_ONCE() and WRITE_ONCE() internally to fix data-races on the sysctl side. For now, proc_douintvec() itself is tolerant to a data-race, but we still need to add annotations on the other subsystem's side.
CVE-2022-49642	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: dwc-qos: Disable split header for Tegra194 There is a long-standing issue with the Synopsys DWC Ethernet driver for Tegra194 where random system crashes have been observed [0]. The problem occurs when the split header feature is enabled in the stmmac driver. In the bad case, a larger than expected buffer length is received and causes the calculation of the total buffer length to overflow. This results in a very large buffer length that causes the kernel to crash. Why this larger buffer length is received is not clear, however, the feedback from the NVIDIA design team is that the split header feature is not supported for Tegra194. Therefore, disable split header support for Tegra194 to prevent these random crashes from occurring. [0] https://lore.kernel.org/linux-tegra/b0b17697-f23e-8fa5-3757-604a86f3a095@nvidia.com/
CVE-2022-49643	In the Linux kernel, the following vulnerability has been resolved: ima: Fix a potential integer overflow in ima_appraise_measurement When the ima-modsig is enabled, the rc passed to evm_verifyxattr() may be negative, which may cause the integer overflow problem.
CVE-2022-49644	In the Linux kernel, the following vulnerability has been resolved: drm/i915: fix a possible refcount leak in intel_dp_add_mst_connector() If drm_connector_init fails, intel_connector_free will be called to take care of proper free. So it is necessary to drop the refcount of port before intel_connector_free. (cherry picked from commit cea9ed611e85d36a05db52b6457bf584b7d969e2)

CVE ID	Description
CVE-2022-49646	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix queue selection for mesh/OCB interfaces When using iTXQ, the code assumes that there is only one vif queue for broadcast packets, using the BE queue. Allowing non-BE queue marking violates that assumption and txq->ac == skb_queue_mapping is no longer guaranteed. This can cause issues with queue handling in the driver and also causes issues with the recent ATF change, resulting in an AQL underflow warning.
CVE-2022-49647	In the Linux kernel, the following vulnerability has been resolved: cgroup: Use separate src/dst nodes when preloading css_sets for migration Each cset (css_set) is pinned by its tasks. When we're moving tasks around across csets for a migration, we need to hold the source and destination csets to ensure that they don't go away while we're moving tasks about. This is done by linking cset->mg_preload_node on either the mgctx->preloaded_src_csets or mgctx->preloaded_dst_csets list. Using the same cset->mg_preload_node for both the src and dst lists was deemed okay as a cset can't be both the source and destination at the same time. Unfortunately, this overloading becomes problematic when multiple tasks are involved in a migration and some of them are identity noop migrations while others are actually moving across cgroups. For example, this can happen with the following sequence on cgroup1: #1> mkdir -p /sys/fs/cgroup/misc/a/b #2> echo \$\$ > /sys/fs/cgroup/misc/a/cgroup.procs #3> RUN_A_COMMAND_WHICH_CREATES_MULTIPLE_THREADS & #4> PID=\$! #5> echo \$PID > /sys/fs/cgroup/misc/a/b/tasks #6> echo \$PID > /sys/fs/cgroup/misc/a/cgroup.procs the process including the group leader back into a. In this final migration, non-leader threads would be doing identity migration while the group leader is doing an actual one. After #3, let's say the whole process was in cset A, and that after #4, the leader moves to cset B. Then, during #6, the following happens: 1. cgroup_migrate_add_src() is called on B for the leader. 2. cgroup_migrate_add_src() is called on A for the other threads. 3. cgroup_migrate_prepare_dst() is called. It scans the src list. 4. It notices that B wants to migrate to A, so it tries to A to the dst list but realizes that its ->mg_preload_node is already busy. 5. and then it notices A wants to migrate to A as it's an identity migration, it culls it by list_del_init()'ing its ->mg_preload_node and putting references accordingly. 6. The rest of migration takes place with B on the src list but nothing on the dst list. This means that A isn't held while migration is in progress. If all tasks leave A before the migration finishes and the incoming task pins it, the cset will be destroyed leading to use-after-free. This is caused by overloading cset->mg_preload_node for both src and dst preload lists. We wanted to exclude the cset from the src list but ended up inadvertently excluding it from the dst list too. This patch fixes the issue by separating out cset->mg_preload_node into ->mg_src_preload_node and ->mg_dst_preload_node, so that the src and dst preloadings don't interfere with each other.
CVE-2022-49651	In the Linux kernel, the following vulnerability has been resolved: srcu: Tighten cleanup_srcu_struct() GP checks Currently, cleanup_srcu_struct() checks for a grace period in progress, but it does not check for a grace period that has not yet started but which might start at any time. Such a situation could result in a use-after-free bug, so this commit adds a check for a grace period that is needed but not yet started to cleanup_srcu_struct().
CVE-2022-49653	In the Linux kernel, the following vulnerability has been resolved: i2c: piix4: Fix a memory leak in the EFCH MMIO support The recently added support for EFCH MMIO regions introduced a memory leak in that code path. The leak is caused by the fact that release_resource() merely removes the resource from the tree but does not free its memory. We need to call release_mem_region() instead, which does free the memory. As a nice side effect, this brings back some symmetry between the legacy and MMIO paths.
CVE-2022-49656	In the Linux kernel, the following vulnerability has been resolved: ARM: meson: Fix refcount leak in meson_smp_prepare_cpus of_find_compatible_node() returns a node pointer with refcount incremented, we should use of_node_put() on it when done. Add missing of_node_put() to avoid refcount leak.
CVE-2022-49657	In the Linux kernel, the following vulnerability has been resolved: usbnet: fix memory leak in error case usbnet_write_cmd_async() mixed up which buffers need to be freed in which error case. v2: add Fixes tag v3: fix uninitialized buf pointer

CVE ID	Description
CVE-2022-49658	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix insufficient bounds propagation from adjust_scalar_min_max_vals Kuee reported a corner case where the tnum becomes constant after the call to __reg_bound_offset(), but the register's bounds are not, that is, its min bounds are still not equal to the register's max bounds. This in turn allows to leak pointers through turning a pointer register as is into an unknown scalar via adjust_ptr_min_max_vals(). Before: func#0 @0 0: R1=ctx(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) R10=fp(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) 0: (b7) r0 = 1 ; R0_w=scalar(imm=1,umin=1,umax=1,var_off=(0x1; 0x0)) 1: (b7) r3 = 0 ; R3_w=scalar(imm=0,umax=0,var_off=(0x0; 0x0)) 2: (87) r3 = -r3 ; R3_w=scalar() 3: (87) r3 = -r3 ; R3_w=scalar() 4: (47) r3 = 32767 ; R3_w=scalar(smin=-9223372036854743041,umin=32767,var_off=(0x7fff; 0xfffffffffff8000),s32_min=-2147450881) 5: (75) if r3 >= 0x0 goto pc+1 ; R3_w=scalar(umin=9223372036854808575,var_off=(0x80000000000007fff; 0x7fffffff8000),s32_min=-2147450881,u32_min=32767) 6: (95) exit from 5 to 7: R0=scalar(imm=1,umin=1,umax=1,var_off=(0x1; 0x0)) R1=ctx(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) R3=scalar(umin=32767,umax=9223372036854775807,var_off=(0x7fff; 0x7fffffff8000),s32_min=-2147450881) R10=fp(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) 7: (d5) if r3 <= 0x8000 goto pc+1 ; R3=scalar(umin=32769,umax=9223372036854775807,var_off=(0x7fff; 0x7fffffff8000),s32_min=-2147450881,u32_min=32767) 8: (95) exit from 7 to 9: R0=scalar(imm=1,umin=1,umax=1,var_off=(0x1; 0x0)) R1=ctx(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) R3=scalar(umin=32767,umax=32768,var_off=(0x7fff; 0x8000)) R10=fp(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) 9: (07) r3 += -32767 ; R3_w=scalar(imm=0,umax=1,var_off=(0x0; 0x0)) <--- [*] 10: (95) exit What can be seen here is that R3=scalar(umin=32767,umax=32768,var_off=(0x7fff; 0x8000)) after the operation R3 += -32767 results in a 'malformed' constant, that is, R3_w=scalar(imm=0,umax=1,var_off=(0x0; 0x0)). Intersecting with var_off has not been done at that point via __update_reg_bounds(), which would have improved the umax to be equal to umin. Refactor the tnum <> min/max bounds information flow into a reg_bounds_sync() helper and use it consistently everywhere. After the fix, bounds have been corrected to R3_w=scalar(imm=0,umax=0,var_off=(0x0; 0x0)) and thus the register is regarded as a 'proper' constant scalar of 0. After: func#0 @0 0: R1=ctx(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) R10=fp(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) 0: (b7) r0 = 1 ; R0_w=scalar(imm=1,umin=1,umax=1,var_off=(0x1; 0x0)) 1: (b7) r3 = 0 ; R3_w=scalar(imm=0,umax=0,var_off=(0x0; 0x0)) 2: (87) r3 = -r3 ; R3_w=scalar() 3: (87) r3 = -r3 ; R3_w=scalar() 4: (47) r3 = 32767 ; R3_w=scalar(smin=-9223372036854743041,umin=32767,var_off=(0x7fff; 0xfffffffffff8000),s32_min=-2147450881) 5: (75) if r3 >= 0x0 goto pc+1 ; R3_w=scalar(umin=9223372036854808575,var_off=(0x80000000000007fff; 0x7fffffff8000),s32_min=-2147450881,u32_min=32767) 6: (95) exit from 5 to 7: R0=scalar(imm=1,umin=1,umax=1,var_off=(0x1; 0x0)) R1=ctx(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) R3=scalar(umin=32767,umax=9223372036854775807,var_off=(0x7fff; 0x7fffffff8000),s32_min=-2147450881) R10=fp(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) 7: (d5) if r3 <= 0x8000 goto pc+1 ; R3=scalar(umin=32769,umax=9223372036854775807,var_off=(0x7fff; 0x7fffffff8000),s32_min=-2147450881,u32_min=32767) 8: (95) exit from 7 to 9: R0=scalar(imm=1,umin=1,umax=1,var_off=(0x1; 0x0)) R1=ctx(off=0,imm=0,umax=0,var_off=(0x0; 0x0)) R3=scalar(umin=32767,umax=32768,var_off=(0x7fff; 0x8000)) R10=fp(off=0 ---truncated---

CVE ID	Description
CVE-2022-49663	In the Linux kernel, the following vulnerability has been resolved: tunnels: do not assume mac header is set in skb_tunnel_check_pmtu() Recently added debug in commit f9aefd6b2aa3 ("net: warn if mac header was not set") caught a bug in skb_tunnel_check_pmtu(), as shown in this syzbot report [1]. In ndo_start_xmit() paths, there is really no need to use skb->mac_header, because skb->data is supposed to point at it. [1] WARNING: CPU: 1 PID: 8604 at include/linux/skbuff.h:2784 skb_mac_header_len include/linux/skbuff.h:2784 [inline] WARNING: CPU: 1 PID: 8604 at include/linux/skbuff.h:2784 skb_tunnel_check_pmtu+0x5de/0x2f90 net/ipv4/ip_tunnel_core.c:413 Modules linked in: CPU: 1 PID: 8604 Comm: syz-executor.3 Not tainted 5.19.0-rc2-syzkaller-00443-g8720bd951b8e #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 RIP: 0010:skb_mac_header_len include/linux/skbuff.h:2784 [inline] RIP: 0010:skb_tunnel_check_pmtu+0x5de/0x2f90 net/ipv4/ip_tunnel_core.c:413 Code: 00 00 00 00 fc ff df 4c 89 fa 48 c1 ea 03 80 3c 02 00 0f 84 b9 fe ff ff 4c 89 ff e8 7c 0f d7 f9 e9 ac fe ff ff e8 c2 13 8a f9 <0f> 0b e9 28 fc ff ff e8 b6 13 8a f9 48 8b 54 24 70 48 b8 00 00 00 RSP: 0018:ffff90002e4f520 EFLAGS: 00010212 RAX: 0000000000000324 RBX: ffff88804d5fd500 RCX: ffff90005b52000 RDX: 0000000000040000 RSI: ffffffff87f05e3e RDI: 0000000000000003 RBP: ffff90002e4f650 R08: 0000000000000003 R09: 0000000000000fff R10: 0000000000000fff R11: 0000000000000000 R12: 0000000000000fff R13: 0000000000000000 R14: 000000000000ffcd R15: 000000000000001f FS: 00007f3babb9700(0000) GS:ffff8880b9b00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000080 CR3: 0000000075319000 CR4: 0000000003506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> geneve_xmit_skb drivers/net/geneve.c:927 [inline] geneve_xmit+0xcfb/0x35d0 drivers/net/geneve.c:1107 __netdev_start_xmit include/linux/netdevice.h:4805 [inline] netdev_start_xmit include/linux/netdevice.h:4819 [inline] __dev_direct_xmit+0x500/0x730 net/core/dev.c:4309 dev_direct_xmit include/linux/netdevice.h:3007 [inline] packet_direct_xmit+0x1b8/0x2c0 net/packet/af_packet.c:282 packet_snd net/packet/af_packet.c:3073 [inline] packet_sendmsg+0x21f4/0x55d0 net/packet/af_packet.c:3104 sock_sendmsg_nosec net/socket.c:714 [inline] sock_sendmsg+0xcf/0x120 net/socket.c:734 __sys_sendmsg+0x6eb/0x810 net/socket.c:2489 __sys_sendmsg+0xf3/0x170 net/socket.c:2543 __sys_sendmsg net/socket.c:2572 [inline] __do_sys_sendmsg net/socket.c:2581 [inline] __se_sys_sendmsg net/socket.c:2579 [inline] __x64_sys_sendmsg+0x132/0x220 net/socket.c:2579 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7f3baaa89109 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f3babb9168 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffff87f05e3e RBX: 00007f3baab9bf60 RCX: 00007f3baaa89109 RDX: 0000000000000000 RSI: 0000000020000a00 RDI: 0000000000000003 RBP: 00007f3baaa89109 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007ffe74f2543f R14: 00007f3babb9300 R15: 0000000000002200 </TASK>
CVE-2022-49664	In the Linux kernel, the following vulnerability has been resolved: tipc: move bc link creation back to tipc_node_create Shuang Li reported a NULL pointer dereference crash: [] BUG: kernel NULL pointer dereference, address: 0000000000000068 [] RIP: 0010:tipc_link_is_up+0x5/0x10 [tipc] [] Call Trace: [] <IRQ> [] tipc_bcast_rcv+0xa2/0x190 [tipc] [] tipc_node_bc_rcv+0x8b/0x200 [tipc] [] tipc_rcv+0x3af/0x5b0 [tipc] [] tipc_udp_rcv+0xc7/0x1e0 [tipc] It was caused by the 'l' passed into tipc_bcast_rcv() is NULL. When it creates a node in tipc_node_check_dest(), after inserting the new node into hashtable in tipc_node_create(), it creates the bc link. However, there is a gap between this insert and bc link creation, a bc packet may come in and get the node from the hashtable then try to dereference its bc link, which is NULL. This patch is to fix it by moving the bc link creation before inserting into the hashtable. Note that for a preliminary node becoming "real", the bc link creation should also be called before it's reshaped, as we don't create it for preliminary nodes.

CVE ID	Description
CVE-2022-49667	In the Linux kernel, the following vulnerability has been resolved: net: bonding: fix use-after-free after 802.3ad slave unbind commit 0622cab0341c ("bonding: fix 802.3ad aggregator reselection"), resolve case, when there is several aggregation groups in the same bond. bond_3ad_unbind_slave will invalidate (clear) aggregator when __agg_active_ports return zero. So, ad_clear_agg can be executed even, when num_of_ports!=0. Than bond_3ad_unbind_slave can be executed again for, previously cleared aggregator. NOTE: at this time bond_3ad_unbind_slave will not update slave ports list, because lag_ports==NULL. So, here we got slave ports, pointing to freed aggregator memory. Fix with checking actual number of ports in group (as was before commit 0622cab0341c ("bonding: fix 802.3ad aggregator reselection")), before ad_clear_agg(). The KASAN logs are as follows: [767.617392] ===== <pre>[767.630776] BUG: KASAN: use-after-free in bond_3ad_state_machine_handler +0x13dc/0x1470 [767.638764] Read of size 2 at addr ffff00011ba9d430 by task kworker/u8:7/767 [767.647361] CPU: 3 PID: 767 Comm: kworker/u8:7 Tainted: G O 5.15.11 #15 [767.655329] Hardware name: DNI AmazonGo1 A7040 board (DT) [767.660760] Workqueue: lacp_1 bond_3ad_state_machine_handler [767.666468] Call trace: [767.668930] dump_backtrace +0x0/0x2d0 [767.672625] show_stack+0x24/0x30 [767.675965] dump_stack_lvl+0x68/0x84 [767.679659] print_address_description.constprop.0+0x74/0x2b8 [767.685451] kasan_report +0x1f0/0x260 [767.689148] __asan_load2+0x94/0xd0 [767.692667] bond_3ad_state_machine_handler +0x13dc/0x1470</pre>
CVE-2022-49670	In the Linux kernel, the following vulnerability has been resolved: linux/dim: Fix divide by 0 in RDMA DIM Fix a divide 0 error in rdma_dim_stats_compare() when prev->cpe_ratio == 0. CallTrace: Hardware name: H3C R4900 G3/RS33M2C9S, BIOS 2.00.37P21 03/12/2020 task: ffff880194b78000 task.stack: ffff90006714000 RIP: 0010:backport_rdma_dim+0x10e/0x240 [mlx_compat] RSP: 0018:ffff880c10e83ec0 EFLAGS: 00010202 RAX: 0000000000002710 RBX: ffff88096cd7f780 RCX: 0000000000000064 RDX: 0000000000000000 RSI: 0000000000000002 RDI: 0000000000000001 RBP: 0000000000000001 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: 000000001d7c6c09 R13: ffff88096cd7f780 R14: ffff880b174fe800 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff880c10e80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000a0965b00 CR3: 000000000200a003 CR4: 0000000007606e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <IRQ> ib_poll_handler+0x43/0x80 [ib_core] irq_poll_softirq+0xae/0x110 __do_softirq+0xd1/0x28c irq_exit+0xde/0xf0 do_IRQ+0x54/0xe0 common_interrupt+0x8f/0x8f </IRQ> ? cpuidle_enter_state+0xd9/0x2a0 ? cpuidle_enter_state+0xc7/0x2a0 ? do_idle+0x170/0x1d0 ? cpu_startup_entry+0x6f/0x80 ? start_secondary+0x1b9/0x210 ? secondary_startup_64+0xa5/0xb0 Code: 0f 87 e1 00 00 00 8b 4c 24 14 44 8b 43 14 89 c8 4d 63 c8 44 29 c0 99 31 d0 29 d0 31 d2 48 98 48 8d 04 80 48 8d 04 80 48 c1 e0 02 <49> f7 f1 48 83 f8 0a 0f 86 c1 00 00 00 44 39 c1 7f 10 48 89 df RIP: backport_rdma_dim+0x10e/0x240 [mlx_compat] RSP: ffff880c10e83ec0
CVE-2022-49671	In the Linux kernel, the following vulnerability has been resolved: RDMA/cm: Fix memory leak in ib_cm_insert_listen cm_alloc_id_priv() allocates resource for the cm_id_priv. When cm_init_listen() fails it doesn't free it, leading to memory leak. Add the missing error unwind.
CVE-2022-49672	In the Linux kernel, the following vulnerability has been resolved: net: tun: unlink NAPI from device on destruction Syzbot found a race between tun file and device destruction. NAPIs live in struct tun_file which can get destroyed before the netdev so we have to del them explicitly. The current code is missing deleting the NAPI if the queue was detached first.
CVE-2022-49673	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix KASAN warning in raid5_add_disks There's a KASAN warning in raid5_add_disk when running the LVM testsuite. The warning happens in the test lvconvert-raid-reshape-linear_to_raid6-single-type.sh. We fix the warning by verifying that rdev->saved_raid_disk is within limits.
CVE-2022-49674	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix accesses beyond end of raid member array On dm-raid table load (using raid_ctr), dm-raid allocates an array rs->devs[rs->raid_disks] for the raid device members. rs->raid_disks is defined by the number of raid metadata and image tuples passed into the target's constructor. In the case of RAID layout changes being requested, that number can be different from the current number of members for existing raid sets as defined in their superblocks. Example RAID layout changes include: - raid1 legs being added/removed - raid4/5/6/10 number of stripes changed (stripe reshaping) - takeover to higher raid level (e.g. raid5 -> raid6) When accessing array members, rs->raid_disks must be used in control loops instead of the potentially larger value in rs->md.raid_disks. Otherwise it will cause memory access beyond the end of the rs->devs array. Fix this by changing code that is prone to out-of-bounds access. Also fix validate_raid_redundancy() to validate all devices that are added. Also, use braces to help clean up raid_iterate_devices(). The out-of-bounds memory accesses was discovered using KASAN. This commit was verified to pass all LVM2 RAID tests (with KASAN enabled).

CVE ID	Description
CVE-2022-49675	In the Linux kernel, the following vulnerability has been resolved: tick/nohz: unexport __init-annotated tick_nohz_full_setup() EXPORT_SYMBOL and __init is a bad combination because the .init.text section is freed up after the initialization. Hence, modules cannot use symbols annotated __init. The access to a freed symbol may end up with kernel panic. modpost used to detect it, but it had been broken for a decade. Commit 28438794aba4 ("modpost: fix section mismatch check for exported init/exit sections") fixed it so modpost started to warn it again, then this showed up: MODPOST vmlinux.symvers WARNING: modpost: vmlinux.o(___ksymtab_gpl+tick_nohz_full_setup+0x0): Section mismatch in reference from the variable __ksymtab_tick_nohz_full_setup to the function .init.text:tick_nohz_full_setup() The symbol tick_nohz_full_setup is exported and annotated __init Fix this by removing the __init annotation of tick_nohz_full_setup or drop the export. Drop the export because tick_nohz_full_setup() is only called from the built-in code in kernel/sched/isolation.c.
CVE-2022-49679	In the Linux kernel, the following vulnerability has been resolved: ARM: Fix refcount leak in axxia_boot_secondary of_find_compatible_node() returns a node pointer with refcount incremented, we should use of_node_put() on it when done. Add missing of_node_put() to avoid refcount leak.
CVE-2022-49687	In the Linux kernel, the following vulnerability has been resolved: virtio_net: fix xdp_rxq_info bug after suspend/resume The following sequence currently causes a driver bug warning when using virtio_net: # ip link set eth0 up # echo mem > /sys/power/state (or e.g. # rtcwake -s 10 -m mem) <resume> # ip link set eth0 down Missing register, driver bug WARNING: CPU: 0 PID: 375 at net/core/xdp.c:138 xdp_rxq_info_unreg+0x58/0x60 Call trace: xdp_rxq_info_unreg+0x58/0x60 virtnet_close+0x58/0xac __dev_close_many+0xac/0x140 __dev_change_flags+0xd8/0x210 dev_change_flags+0x24/0x64 do_setlink+0x230/0xdd0 ... This happens because virtnet_freeze() frees the receive_queue completely (including struct xdp_rxq_info) but does not call xdp_rxq_info_unreg(). Similarly, virtnet_restore() sets up the receive_queue again but does not call xdp_rxq_info_reg(). Actually, parts of virtnet_freeze_down() and virtnet_restore_up() are almost identical to virtnet_close() and virtnet_open(): only the calls to xdp_rxq_info_(un)reg() are missing. This means that we can fix this easily and avoid such problems in the future by just calling virtnet_close()/open() from the freeze/restore handlers. Aside from adding the missing xdp_rxq_info calls the only difference is that the refill work is only cancelled if netif_running(). However, this should not make any functional difference since the refill work should only be active if the network interface is actually up.
CVE-2022-49696	In the Linux kernel, the following vulnerability has been resolved: tipc: fix use-after-free Read in tipc_named_reinit syzbot found the following issue on: ===== BUG: KASAN: use-after-free in tipc_named_reinit+0x94f/0x9b0 net/tipc/name_distr.c:413 Read of size 8 at addr ffff88805299a000 by task kworker/1:9/23764 CPU: 1 PID: 23764 Comm: kworker/1:9 Not tainted 5.18.0-rc4-syzkaller-00878-g17d49e6e8012 #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 Workqueue: events tipc_net_finalize_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xcd/0x134 lib/dump_stack.c:106 print_address_description.constprop.0.cold+0xeb/0x495 mm/kasan/report.c:313 print_report mm/kasan/report.c:429 [inline] kasan_report.cold+0xf4/0x1c6 mm/kasan/report.c:491 tipc_named_reinit+0x94f/0x9b0 net/tipc/name_distr.c:413 tipc_net_finalize+0x234/0x3d0 net/tipc/net.c:138 process_one_work+0x996/0x1610 kernel/workqueue.c:2289 worker_thread+0x665/0x1080 kernel/workqueue.c:2436 kthread+0x2e9/0x3a0 kernel/kthread.c:376 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:298 </TASK> [...] ===== In the commit d966ddcc3821 ("tipc: fix a deadlock when flushing scheduled work"), the cancel_work_sync() function just to make sure ONLY the work tipc_net_finalize_work() is executing/pending on any CPU completed before tipc namespace is destroyed through tipc_exit_net(). But this function is not guaranteed the work is the last queued. So, the destroyed instance may be accessed in the work which will try to enqueue later. In order to completely fix, we re-order the calling of cancel_work_sync() to make sure the work tipc_net_finalize_work() was last queued and it must be completed by calling cancel_work_sync().
CVE-2022-49697	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix request_sock leak in sk lookup helpers A customer reported a request_socket leak in a Calico cloud environment. We found that a BPF program was doing a socket lookup with takes a refcnt on the socket and that it was finding the request_socket but returning the parent LISTEN socket via sk_to_full_sk() without decrementing the child request socket 1st, resulting in request_sock slab object leak. This patch retains the existing behaviour of returning full socks to the caller but it also decrements the child request_socket if one is present before doing so to prevent the leak. Thanks to Curtis Taylor for all the help in diagnosing and testing this. And thanks to Antoine Tenart for the reproducer and patch input. v2 of this patch contains, refactor as per Daniel Borkmann's suggestions to validate RCU flags on the listen socket so that it balances with bpf_sk_release() and update comments as per Martin KaFai Lau's suggestion. One small change to Daniels suggestion, put "sk = sk2" under "if (sk2 != sk)" to avoid an extra instruction.

CVE ID	Description
CVE-2022-49698	In the Linux kernel, the following vulnerability has been resolved: netfilter: use get_random_u32 instead of prandom bh might occur while updating per-cpu rnd_state from user context, ie. local_out path. BUG: using smp_processor_id() in preemptible [00000000] code: nginx/2725 caller is nft_ng_random_eval +0x24/0x54 [nft_numgen] Call Trace: check_preemption_disabled+0xde/0xe0 nft_ng_random_eval +0x24/0x54 [nft_numgen] Use the random driver instead, this also avoids need for local prandom state. Moreover, prandom now uses the random driver since d4150779e60f ("random32: use real rng for non-deterministic randomness"). Based on earlier patch from Pablo Neira.
CVE-2022-49700	In the Linux kernel, the following vulnerability has been resolved: mm/slab: add missing TID updates on slab deactivation The fastpath in slab_alloc_node() assumes that c->slab is stable as long as the TID stays the same. However, two places in __slab_alloc() currently don't update the TID when deactivating the CPU slab. If multiple operations race the right way, this could lead to an object getting lost; or, in an even more unlikely situation, it could even lead to an object being freed onto the wrong slab's freelist, messing up the `inuse` counter and eventually causing a page to be freed to the page allocator while it still contains slab objects. (I haven't actually tested these cases though, this is just based on looking at the code. Writing testcases for this stuff seems like it'd be a pain...) The race leading to state inconsistency is (all operations on the same CPU and kmem_cache): - task A: begin do_slab_free(): - read TID - read pcpu freelist (==NULL) - check `slab == c->slab` (true) - [PREEMPT A->B] - task B: begin slab_alloc_node(): - fastpath fails (`c->freelist` is NULL) - enter __slab_alloc() - slub_get_cpu_ptr() (disables preemption) - enter __slab_alloc() - take local_lock_irqsave() - read c->freelist as NULL - get_freelist() returns NULL - write `c->slab = NULL` - drop local_unlock_irqrestore() - goto new_slab - slub_percpu_partial() is NULL - get_partial() returns NULL - slub_put_cpu_ptr() (enables preemption) - [PREEMPT B->A] - task A: finish do_slab_free(): - this_cpu_cmpxchg_double() succeeds() - [CORRUPT STATE: c->slab==NULL, c->freelist!=NULL] From there, the object on c->freelist will get lost if task B is allowed to continue from here: It will proceed to the retry_load_slab label, set c->slab, then jump to load_freelist, which clobbers c->freelist. But if we instead continue as follows, we get worse corruption: - task A: run __slab_free() on object from other struct slab: - CPU_PARTIAL_FREE case (slab was on no list, is now on pcpu partial) - task A: run slab_alloc_node() with NUMA node constraint: - fastpath fails (c->slab is NULL) - call __slab_alloc() - slub_get_cpu_ptr() (disables preemption) - enter __slab_alloc() - c->slab is NULL: goto new_slab - slub_percpu_partial() is non-NULL - set c->slab to slub_percpu_partial(c) - [CORRUPT STATE: c->slab points to slab-1, c->freelist has objects from slab-2] - goto redo - node_match() fails - goto deactivate_slab - existing c->freelist is passed into deactivate_slab() - inuse count of slab-1 is decremented to account for object from slab-2 At this point, the inuse count of slab-1 is 1 lower than it should be. This means that if we free all allocated objects in slab-1 except for one, SLUB will think that slab-1 is completely unused, and may free its page, leading to use-after-free.

CVE ID	Description
CVE-2022-49701	In the Linux kernel, the following vulnerability has been resolved: scsi: ibmvfc: Allocate/free queue resource only during probe/remove Currently, the sub-queues and event pool resources are allocated/freed for every CRQ connection event such as reset and LPM. This exposes the driver to a couple issues. First the inefficiency of freeing and reallocating memory that can simply be reused after being sanitized. Further, a system under memory pressure runs the risk of allocation failures that could result in a crippled driver. Finally, there is a race window where command submission/completion can try to pull/return elements from/to an event pool that is being deleted or already has been deleted due to the lack of host state around freeing/allocating resources. The following is an example of list corruption following a live partition migration (LPM): Oops: Exception in kernel mode, sig: 5 [#1] LE PAGE_SIZE=64K MMU=Hash SMP NR_CPUS=2048 NUMA pSeries Modules linked in: vfat fat isofs cdrom ext4 mbcache jbd2 nft_counter nft_compat nf_tables nfnetlink rpadlpar_io rpaphp xsk_diag nfsv3 nfs_acl nfs lockd grace fscache netfs rkill bonding tls sunrpc pseries_rng drm drm_panel_orientation_quirks xfs libcrc32c dm_service_time sd_mod t10_pi sg ibmvfc scsi_transport_fc ibmveth vmx_crypto dm_multipath dm_mirror dm_region_hash dm_log dm_mod ipmi_devintf ipmi_msghandler fuse CPU: 0 PID: 2108 Comm: ibmvfc_0 Kdump: loaded Not tainted 5.14.0-70.9.1.el9_0.ppc64le #1 NIP: c0000000007c4bb0 LR: c0000000007c4bac CTR: 00000000005b9a10 REGS: c00000025c10b760 TRAP: 0700 Not tainted (5.14.0-70.9.1.el9_0.ppc64le) MSR: 800000000282b033 <SF,VEC,VSX,EE,FP,ME,IR,DR,RI,LE> CR: 2800028f XER: 0000000f CFAR: c000000001f55bc IRQMASK: 0 GPR0: c0000000007c4bac c00000025c10ba00 c000000002a47c00 000000000000004e GPR04: c0000031e3006f88 c0000031e308bd00 c00000025c10b768 0000000000000027 GPR08: 0000000000000000 c0000031e3009dc0 00000031e0eb0000 0000000000000000 GPR12: c0000031e2ffffa8 c000000002dd0000 c000000000187108 c00000020fcee2c0 GPR16: 0000000000000000 0000000000000000 0000000000000000 0000000000000000 GPR20: 0000000000000000 0000000000000000 0000000000000000 c008000002f81300 GPR24: 5deadbeef0000100 5deadbeef0000122 c000000263ba6910 c00000024cc88000 GPR28: 000000000000003c c0000002430a0000 c0000002430ac300 000000000000c300 NIP [c0000000007c4bb0] __list_del_entry_valid+0x90/0x100 LR [c0000000007c4bac] __list_del_entry_valid+0x8c/0x100 Call Trace: [c00000025c10ba00] [c0000000007c4bac] __list_del_entry_valid+0x8c/0x100 (unreliable) [c00000025c10ba60] [c008000002f42284] ibmvfc_free_queue+0xec/0x210 [ibmvfc] [c00000025c10bb10] [c008000002f4246c] ibmvfc_deregister_scsi_channel+0xc4/0x160 [ibmvfc] [c00000025c10bba0] [c008000002f42580] ibmvfc_release_sub_crqs+0x78/0x130 [ibmvfc] [c00000025c10bc20] [c008000002f4f6cc] ibmvfc_do_work+0x5c4/0xc70 [ibmvfc] [c00000025c10bce0] [c008000002f4fdec] ibmvfc_work+0x74/0x1e8 [ibmvfc] [c00000025c10bda0] [c0000000001872b8] kthread+0x1b8/0x1c0 [c00000025c10be10] [c00000000000cd64] ret_from_kernel_thread+0x5c/0x64 Instruction dump: 40820034 38600001 38210060 4e800020 7c0802a6 7c641b78 3c62fe7a 7d254b78 3863b590 f8010070 4ba309cd 60000000 <0fe00000> 7c0802a6 3c62fe7a 3863b640 ---[end trace 11a2b65a92f8b66c]--- ibmvfc 30000003: Send warning. Receive queue closed, will retry. Add registration/deregistration helpers that are called instead during connection resets to sanitize and reconfigure the queues.
CVE-2022-49703	In the Linux kernel, the following vulnerability has been resolved: scsi: ibmvfc: Store vhost pointer during subcrq allocation Currently the back pointer from a queue to the vhost adapter isn't set until after subcrq interrupt registration. The value is available when a queue is first allocated and can/should be also set for primary and async queues as well as subcrqs. This fixes a crash observed during kexec/kdump on Power 9 with legacy XICS interrupt controller where a pending subcrq interrupt from the previous kernel can be replayed immediately upon IRQ registration resulting in dereference of a garbage backpointer in ibmvfc_interrupt_scsi(). Kernel attempted to read user page (58) - exploit attempt? (uid: 0) BUG: Kernel NULL pointer dereference on read at 0x00000058 Faulting instruction address: 0xc008000003216a08 Oops: Kernel access of bad area, sig: 11 [#1] ... NIP [c008000003216a08] ibmvfc_interrupt_scsi+0x40/0xb0 [ibmvfc] LR [c00000000082079e8] __handle_irq_event_percpu+0x98/0x270 Call Trace: [c0000000047fa3d80] [c00000000123e6180] 0xc00000000123e6180 (unreliable) [c0000000047fa3df0] [c0000000082079e8] __handle_irq_event_percpu+0x98/0x270 [c0000000047fa3ea0] [c000000008207d18] handle_irq_event+0x98/0x188 [c0000000047fa3ef0] [c00000000820f564] handle_fasteoi_irq+0xc4/0x310 [c0000000047fa3f40] [c000000008205c60] generic_handle_irq+0x50/0x80 [c0000000047fa3f60] [c000000008015c40] __do_irq+0x70/0x1a0 [c0000000047fa3f90] [c000000008016d7c] __do_IRQ+0x9c/0x130 [c000000014622f60] [0000000020000000] 0x20000000 [c000000014622ff0] [c000000008016e50] do_IRQ+0x40/0xa0 [c000000014623020] [c000000008017044] replay_soft_interrupts+0x194/0x2f0 [c000000014623210] [c0000000080172a8] arch_local_irq_restore+0x108/0x170 [c000000014623240] [c0000000080eb1008] _raw_spin_unlock_irqrestore+0x58/0xb0 [c000000014623270] [c00000000820b12c] __setup_irq+0x49c/0x9f0 [c000000014623310] [c00000000820b7c0] request_threaded_irq+0x140/0x230 [c000000014623380] [c008000003212a50] ibmvfc_register_scsi_channel+0x1e8/0x2f0 [ibmvfc] [c000000014623450] [c008000003213d1c] ibmvfc_init_sub_crqs+0xc4/0x1f0 [ibmvfc] [c0000000146234d0] [c0080000032145a8] ibmvfc_reset_crq+0x150/0x210 [ibmvfc] [c000000014623550] [c0080000032147c8] ibmvfc_init_crq+0x160/0x280 [ibmvfc] [c0000000146235f0] [c00800000321a9cc] ibmvfc_probe+0x2a4/0x530 [ibmvfc]

CVE ID	Description
CVE-2022-49708	In the Linux kernel, the following vulnerability has been resolved: ext4: fix bug_on ext4_mb_use_inode_pa Hulk Robot reported a BUG_ON: <pre>===== kernel BUG at fs/ext4/mballo.c:3211! [...] RIP: 0010:ext4_mb_mark_disk_space_used.cold +0x85/0x136f [...] Call Trace: ext4_mb_new_blocks+0x9df/0x5d30 ext4_ext_map_blocks +0x1803/0x4d80 ext4_map_blocks+0x3a4/0x1a10 ext4_writepages+0x126d/0x2c30 do_writepages+0x7f/0x1b0 __filemap_fdatawrite_range+0x285/0x3b0 file_write_and_wait_range +0xb1/0x140 ext4_sync_file+0x1aa/0xca0 vfs_fsync_range+0xfb/0x260 do_fsync+0x48/0xa0 [...] =====</pre> Above issue may happen as follows: ----- do_fsync vfs_fsync_range ext4_sync_file file_write_and_wait_range __filemap_fdatawrite_range do_writepages ext4_writepages mpage_map_and_submit_extent mpage_map_one_extent ext4_map_blocks ext4_mb_new_blocks ext4_mb_normalize_request >>> start + size <= ac->ac_o_ex.fe_logical ext4_mb_regular_allocator ext4_mb_simple_scan_group ext4_mb_use_best_found ext4_mb_new_preallocation ext4_mb_new_inode_pa ext4_mb_use_inode_pa >>> set ac- >ac_b_ex.fe_len <= 0 ext4_mb_mark_disk_space_used >>> BUG_ON(ac->ac_b_ex.fe_len <= 0); we can easily reproduce this problem with the following commands: `fallocate -l100M disk` `mkfs.ext4 -b  1024 -g 256 disk` `mount disk /mnt` `fstress -d /mnt -l 0 -n 1000 -p 1` The size must be smaller than or equal to EXT4_BLOCKS_PER_GROUP. Therefore, "start + size <= ac->ac_o_ex.fe_logical" may occur when the size is truncated. So start should be the start position of the group where ac_o_ex.fe_logical is located after alignment. In addition, when the value of fe_logical or EXT4_BLOCKS_PER_GROUP is very large, the value calculated by start_off is more accurate.
CVE-2022-49710	In the Linux kernel, the following vulnerability has been resolved: dm mirror log: round up region bitmap size to BITS_PER_LONG The code in dm-log rounds up bitset_size to 32 bits. It then uses find_next_zero_bit_le on the allocated region. find_next_zero_bit_le accesses the bitmap using unsigned long pointers. So, on 64-bit architectures, it may access 4 bytes beyond the allocated size. Fix this bug by rounding up bitset_size to BITS_PER_LONG. This bug was found by running the lvm2 testsuite with kasan.
CVE-2022-49715	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Fix refcount leak in gic_populate_ppi_partitions of _find_node_by_phandle() returns a node pointer with refcount incremented, we should use of_node_put() on it when not need anymore. Add missing of_node_put() to avoid refcount leak.
CVE-2022-49716	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3: Fix error handling in gic_populate_ppi_partitions of _get_child_by_name() returns a node pointer with refcount incremented, we should use of_node_put() on it when not need anymore. When kcalloc fails, it missing of_node_put() and results in refcount leak. Fix this by goto out_put_node label.
CVE-2022-49720	In the Linux kernel, the following vulnerability has been resolved: block: Fix handling of offline queues in blk_mq_alloc_request_hctx() This patch prevents that test nvme/004 triggers the following: UBSAN: array-index-out-of-bounds in block/blk-mq.h:135:9 index 512 is out of range for type 'long unsigned int [512]' Call Trace: show_stack+0x52/0x58 dump_stack_lvl+0x49/0x5e dump_stack+0x10/0x12 ubsan_epilogue+0x9/0x3b __ubsan_handle_out_of_bounds.cold+0x44/0x49 blk_mq_alloc_request_hctx +0x304/0x310 __nvme_submit_sync_cmd+0x70/0x200 [nvme_core] nvmmf_connect_io_queue +0x23e/0x2a0 [nvme_fabrics] nvme_loop_connect_io_queues+0x8d/0xb0 [nvme_loop] nvme_loop_create_ctrl+0x58e/0x7d0 [nvme_loop] nvmmf_create_ctrl+0x1d7/0x4d0 [nvme_fabrics] nvmmf_dev_write+0xae/0x111 [nvme_fabrics] vfs_write+0x144/0x560 ksys_write+0xb7/0x140 __x64_sys_write+0x42/0x50 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe +0x44/0xae

CVE ID	Description
CVE-2022-49722	In the Linux kernel, the following vulnerability has been resolved: ice: Fix memory corruption in VF driver Disable VF's RX/TX queues, when it's disabled. VF can have queues enabled, when it requests a reset. If PF driver assumes that VF is disabled, while VF still has queues configured, VF may unmap DMA resources. In such scenario device still can map packets to memory, which ends up silently corrupting it. Previously, VF driver could experience memory corruption, which lead to crash: [5119.170157] BUG: unable to handle kernel paging request at 00001b9780003237 [5119.170166] PGD 0 P4D 0 [5119.170173] Oops: 0002 [#1] PREEMPT_RT SMP PTI [5119.170181] CPU: 30 PID: 427592 Comm: kworker/u96:2 Kdump: loaded Tainted: G W I ----- 4.18.0-372.9.1.rt7.166.el8.x86_64 #1 [5119.170189] Hardware name: Dell Inc. PowerEdge R740/014X06, BIOS 2.3.10 08/15/2019 [5119.170193] Workqueue: iavf iavf_adminq_task [iavf] [5119.170219] RIP: 0010: __page_frag_cache_drain+0x5/0x30 [5119.170238] Code: 0f 0f b6 77 51 85 f6 74 07 31 d2 e9 05 df ff ff e9 90 fe ff ff 48 8b 05 49 db 33 01 eb b4 0f 1f 80 00 00 00 00 0f 1f 44 00 00 <f0> 29 77 34 74 01 c3 48 8b 07 f6 c4 80 74 0f 0f b6 77 51 85 f6 74 [5119.170244] RSP: 0018:ffffa43b0bdcfd78 EFLAGS: 00010282 [5119.170250] RAX: ffffffff896b3e40 RBX: ffff8fb282524000 RCX: 0000000000000002 [5119.170254] RDX: 0000000049000000 RSI: 0000000000000000 RDI: 00001b9780003203 [5119.170259] RBP: ffff8fb248217b00 R08: 0000000000000022 R09: 0000000000000009 [5119.170262] R10: 2b849d6300000000 R11: 0000000000000020 R12: 0000000000000000 [5119.170265] R13: 0000000000001000 R14: 0000000000000009 R15: 0000000000000000 [5119.170269] FS: 0000000000000000(0000) GS:ffff8fb1201c0000(0000) knlGS:0000000000000000 [5119.170274] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [5119.170279] CR2: 00001b9780003237 CR3: 00000008f3e1a003 CR4: 00000000007726e0 [5119.170283] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [5119.170286] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [5119.170290] PKRU: 55555554 [5119.170292] Call Trace: [5119.170298] iavf_clean_rx_ring+0xad/0x110 [iavf] [5119.170324] iavf_free_rx_resources+0xe/0x50 [iavf] [5119.170342] iavf_free_all_rx_resources.part.51+0x30/0x40 [iavf] [5119.170358] iavf_virtchnl_completion+0xd8a/0x15b0 [iavf] [5119.170377] ? iavf_clean_arq_element+0x210/0x280 [iavf] [5119.170397] iavf_adminq_task+0x126/0x2e0 [iavf] [5119.170416] process_one_work+0x18f/0x420 [5119.170429] worker_thread+0x30/0x370 [5119.170437] ? process_one_work+0x420/0x420 [5119.170445] kthread+0x151/0x170 [5119.170452] ? set_kthread_struct+0x40/0x40 [5119.170460] ret_from_fork+0x35/0x40 [5119.170477] Modules linked in: iavf sctp ip6_udp_tunnel udp_tunnel mlx4_en mlx4_core nfp tls vhost_net vhost vhost_iotlb tap tun xt_CHECKSUM ipt_MASQUERADE xt_conntrack ipt_REJECT nf_reject_ipv4 nft_compat nft_counter nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 nf_tables nfnetlink bridge stp llc rpcsec_gss_krb5 auth_rpcgss nfsv4 dns_resolver nfs lockd grace fscache sunrpc intel_rapl_msr iTCO_wdt iTCO_vendor_support dell_smbios wmi_bmof dell_wmi_descriptor dcdbas kvm_intel kvm irqbypass intel_rapl_common isst_if_common skx_edac irdma nfit libnvdimm x86_pkg_temp_thermal i40e intel_powerclamp coretemp crct10dif_pclmul crc32_pclmul ghash_clmulni_intel ib_uverbs rapl ipmi_ssif intel_cstate intel_uncore mei_me pcspkr acpi_ipmi ib_core mei_lpc_ich i2c_i801 ipmi_si ipmi_devintf wmi ipmi_msghandler acpi_power_meter xfs libcrc32c sd_mod t10_pi sg mgag200 drm_kms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops ice ahci drm_libahci crc32c intel libata tg3 megaraid_sas [5119.170613] i2c_algo_bit dm_mirror dm_region_hash dm_log dm_mod fuse [last unloaded: iavf] [5119.170627] CR2: 00001b9780003237
CVE-2022-49725	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix call trace in setup_tx_descriptors After PF reset and ethtool -t there was call trace in dmesg sometimes leading to panic. When there was some time, around 5 seconds, between reset and test there were no errors. Problem was that pf reset calls i40e_vsi_close in prep_for_reset and ethtool -t calls i40e_vsi_close in diag_test. If there was not enough time between those commands the second i40e_vsi_close starts before previous i40e_vsi_close was done which leads to crash. Add check to diag_test if pf is in reset and don't start offline tests if it is true. Add netif_info("testing failed") into unhappy path of i40e_diag_test()
CVE-2022-49728	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix signed integer overflow in __ip6_append_data Resurrect ubsan overflow checks and ubsan report this warning, fix it by change the variable [length] type to size_t. UBSAN: signed-integer-overflow in net/ipv6/ip6_output.c:1489:19 2147479552 + 8567 cannot be represented in type 'int' CPU: 0 PID: 253 Comm: err Not tainted 5.16.0+ #1 Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0x214/0x230 show_stack+0x30/0x78 dump_stack_lvl+0xf8/0x118 dump_stack+0x18/0x30 ubsan_epilogue+0x18/0x60 handle_overflow+0xd0/0xf0 __ubsan_handle_add_overflow+0x34/0x44 __ip6_append_data.isra.48+0x1598/0x1688 ip6_append_data+0x128/0x260 udpv6_sendmsg+0x680/0xdd0 inet6_sendmsg+0x54/0x90 sock_sendmsg+0x70/0x88 __sys_sendmsg+0xe8/0x368 __sys_sendmsg+0x98/0xe0 __sys_sendmmsg+0xf4/0x3b8 __arm64_sys_sendmmsg+0x34/0x48 invoke_syscall+0x64/0x160 el0_svc_common.constprop.4+0x124/0x300 do_el0_svc+0x44/0xc8 el0_svc+0x3c/0x1e8 el0t_64_sync_handler+0x88/0xb0 el0t_64_sync+0x16c/0x170 Changes since v1: -Change the variable [length] type to unsigned, as Eric Dumazet suggested. Changes since v2: -Don't change exthdrln type in ip6_make_skb, as Paolo Abeni suggested. Changes since v3: -Don't change ulen type in udpv6_sendmsg and l2tp_ip6_sendmsg, as Jakub Kicinski suggested.

CVE ID	Description
CVE-2022-49730	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Resolve NULL ptr dereference after an ELS LOGO is aborted A use-after-free crash can occur after an ELS LOGO is aborted. Specifically, a nodelist structure is freed and then ndlp->vport->cfg_log_verbose is dereferenced in lpfc_nlp_get() when the discovery state machine is mistakenly called a second time with NLP_EVT_DEVICE_RM argument. Rework lpfc_cmpl_els_logo() to prevent the duplicate calls to release a nodelist structure.
CVE-2022-49731	In the Linux kernel, the following vulnerability has been resolved: ata: libata-core: fix NULL pointer deref in ata_host_alloc_pinfo() In an unlikely (and probably wrong?) case that the 'ppi' parameter of ata_host_alloc_pinfo() points to an array starting with a NULL pointer, there's going to be a kernel oops, as the 'pi' local variable won't get reassigned from the initial value of NULL. Initialize 'pi' instead to '&ata_dummy_port_info' to fix the possible kernel oops for good... Found by Linux Verification Center (linuxtesting.org) with the SVACE static analysis tool.
CVE-2022-49739	In the Linux kernel, the following vulnerability has been resolved: gfs2: Always check inode size of inline inodes Check if the inode size of stuffed (inline) inodes is within the allowed range when reading inodes from disk (gfs2_dinode_in()). This prevents us from on-disk corruption. The two checks in stuffed_readpage() and gfs2_unstuffer_page() that just truncate inline data to the maximum allowed size don't actually make sense, and they can be removed now as well.
CVE-2022-49740	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Check the count value of channel spec to prevent out-of-bounds reads This patch fixes slab-out-of-bounds reads in brcmfmac that occur in brcmf_construct_chaninfo() and brcmf_enable_bw40_2g() when the count value of channel specifications provided by the device is greater than the length of 'list->element[]', decided by the size of the 'list' allocated with kzalloc(). The patch adds checks that make the functions free the buffer and return -EINVAL if that is the case. Note that the negative return is handled by the caller, brcmf_setup_wiphybands() or brcmf_cfg80211_attach(). Found by a modified version of syzkaller. Crash Report from brcmf_construct_chaninfo(): ===== BUG: KASAN: slab-out-of-bounds in brcmf_setup_wiphybands+0x1238/0x1430 Read of size 4 at addr ffff888115f24600 by task kworker/0:2/1896 CPU: 0 PID: 1896 Comm: kworker/0:2 Tainted: G W O 5.14.0+ #132 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org 04/01/2014 Workqueue: usb_hub_wq hub_event Call Trace: dump_stack_lvl+0x57/0x7d print_address_description.constprop.0.cold+0x93/0x334 kasan_report.cold+0x83/0xdf brcmf_setup_wiphybands+0x1238/0x1430 brcmf_cfg80211_attach+0x2118/0x3fd0 brcmf_attach+0x389/0xd40 brcmf_usb_probe+0x12de/0x1690 usb_probe_interface+0x25f/0x710 really_probe+0x1be/0xa90 __driver_probe_device+0x2ab/0x460 driver_probe_device+0x49/0x120 __device_attach_driver+0x18a/0x250 bus_for_each_drv+0x123/0x1a0 __device_attach+0x207/0x330 bus_probe_device+0x1a2/0x260 device_add+0xa61/0x1ce0 usb_set_configuration+0x984/0x1770 usb_generic_driver_probe+0x69/0x90 usb_probe_device+0x9c/0x220 really_probe+0x1be/0xa90 __driver_probe_device+0x2ab/0x460 driver_probe_device+0x49/0x120 __device_attach_driver+0x18a/0x250 bus_for_each_drv+0x123/0x1a0 __device_attach+0x207/0x330 bus_probe_device+0x1a2/0x260 device_add+0xa61/0x1ce0 usb_new_device.cold+0x463/0xf66 hub_event+0x10d5/0x3330 process_one_work+0x873/0x13e0 worker_thread+0x8b/0xd10 kthread+0x379/0x450 ret_from_fork+0x1f/0x30 Allocated by task 1896: kasan_save_stack+0x1b/0x40 __kasan_kmalloc+0x7c/0x90 kmem_cache_alloc_trace+0x19e/0x330 brcmf_setup_wiphybands+0x290/0x1430 brcmf_cfg80211_attach+0x2118/0x3fd0 brcmf_attach+0x389/0xd40 brcmf_usb_probe+0x12de/0x1690 usb_probe_interface+0x25f/0x710 really_probe+0x1be/0xa90 __driver_probe_device+0x2ab/0x460 driver_probe_device+0x49/0x120 __device_attach_driver+0x18a/0x250 bus_for_each_drv+0x123/0x1a0 __device_attach+0x207/0x330 bus_probe_device+0x1a2/0x260 device_add+0xa61/0x1ce0 usb_new_device.cold+0x463/0xf66 hub_event+0x10d5/0x3330 process_one_work+0x873/0x13e0 worker_thread+0x8b/0xd10 kthread+0x379/0x450 ret_from_fork+0x1f/0x30 The buggy address belongs to the object at ffff888115f24000 which belongs to the cache kcalloc-2k of size 2048 The buggy address is located 1536 bytes inside of 2048-byte region [ffff888115f24000, ffff888115f24800) Memory state around the buggy address: ffff888115f24500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ffff888115f24580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >ffff888115f24600: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ^ ffff888115f24680: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ffff888115f24700: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ===== Crash Report from brcmf_enable_bw40_2g(): ===== ---truncated---
CVE-2022-49743	In the Linux kernel, the following vulnerability has been resolved: ovl: Use "buf" flexible array for memcpy() destination The "buf" flexible array needs to be the memcpy() destination to avoid false positive run-time warning from the recent FORTIFY_SOURCE hardening: memcpy: detected field-spanning write (size 93) of single field "&fh->fb" at fs/overlayfs/export.c:799 (size 21)

CVE ID	Description
CVE-2022-49748	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd: fix potential integer overflow on shift of a int The left shift of int 32 bit integer constant 1 is evaluated using 32 bit arithmetic and then passed as a 64 bit function argument. In the case where i is 32 or more this can lead to an overflow. Avoid this by shifting using the BIT_ULL macro instead.
CVE-2022-49749	In the Linux kernel, the following vulnerability has been resolved: i2c: designware: use casting of u64 in clock multiplication to avoid overflow In functions i2c_dw_scl_lcnt() and i2c_dw_scl_hcnt() may have overflow by depending on the values of the given parameters including the ic_clk. For example in our use case where ic_clk is larger than one million, multiplication of ic_clk * 4700 will result in 32 bit overflow. Add cast of u64 to the calculation to avoid multiplication overflow, and use the corresponding define for divide.
CVE-2022-49750	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Add u64 casts to avoid overflowing The fields of the _CPC object are unsigned 32-bits values. To avoid overflows while using _CPC's values, add 'u64' casts.
CVE-2022-49753	In the Linux kernel, the following vulnerability has been resolved: dmaengine: Fix double increment of client_count in dma_chan_get() The first time dma_chan_get() is called for a channel the channel client_count is incorrectly incremented twice for public channels, first in balance_ref_count(), and again prior to returning. This results in an incorrect client count which will lead to the channel resources not being freed when they should be. A simple test of repeated module load and unload of async_tx on a Dell Power Edge R7425 also shows this resulting in a kref underflow warning. [124.329662] async_tx: api initialized (async) [129.000627] async_tx: api initialized (async) [130.047839] -----[cut here]----- [130.052472] refcount_t: underflow; use-after-free. [130.057279] WARNING: CPU: 3 PID: 19364 at lib/refcount.c:28 refcount_warn_saturate+0xba/0x110 [130.065811] Modules linked in: async_tx(-) rkill intel_rapl_msr intel_rapl_common amd64_edac edac_mce_amd ipmi_ssif kvm_amd dcdbas kvm mgag200 drm_shmem_helper acpi_ipmi irqbypass drm_kms_helper ipmi_si syscopyarea sysfillrect rapl pcspkr ipmi_devintf sysimgblt fb_sys_fops k10temp i2c_piix4 ipmi_msghandler acpi_power_meter acpi_cpufreq vfat fat drm fuse xfs libcrc32c sd_mod t10_pi sg ahci crct10dif_pclmul libahci crc32_pclmul crc32c_intel ghash_clmulni_intel igb megaraid_sas i40e libata i2c_algo_bit ccp sp5100_tco dca dm_mirror dm_region_hash dm_log dm_mod [last unloaded: async_tx] [130.117361] CPU: 3 PID: 19364 Comm: modprobe Kdump: loaded Not tainted 5.14.0-185.el9.x86_64 #1 [130.126091] Hardware name: Dell Inc. PowerEdge R7425/02MJ3T, BIOS 1.18.0 01/17/2022 [130.133806] RIP: 0010:refcount_warn_saturate+0xba/0x110 [130.139041] Code: 01 01 e8 6d bd 55 0f 0b e9 72 9d 8a 00 80 3d 26 18 9c 01 00 75 85 48 c7 c7 f8 a3 03 9d c6 05 16 18 9c 01 01 e8 4a bd 55 00 <0f> 0b e9 4f 9d 8a 00 80 3d 01 18 9c 01 00 0f 85 5e ff ff ff 48 c7 [130.157807] RSP: 0018:ffff98898afe68 EFLAGS: 00010286 [130.163036] RAX: 0000000000000000 RBX: ffff9da06028e598 RCX: 0000000000000000 [130.170172] RDX: ffff9daf9de26480 RSI: ffff9daf9de198a0 RDI: ffff9daf9de198a0 [130.177316] RBP: ffff9da7cddf3970 R08: 0000000000000000 R09: 00000000ffff7fff [130.184459] R10: ffff98898afd00 R11: ffffffff9d9e8c28 R12: ffff9da7cddf1970 [130.191596] R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 [130.198739] FS: 00007f646435c740(0000) GS:ffff9daf9de00000(0000) knlGS:0000000000000000 [130.206832] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [130.212586] CR2: 00007f6463b214f0 CR3: 000000008ab98c00 CR4: 0000000003506e0 [130.219729] Call Trace: [130.222192] <TASK> [130.224305] dma_chan_put+0x10d/0x110 [130.227988] dmaengine_put+0x7a/0xa0 [130.231575] __do_sys_delete_module.constprop.0+0x178/0x280 [130.237157] ? syscall_trace_enter.constprop.0+0x145/0x1d0 [130.242652] do_syscall_64+0x5c/0x90 [130.246240] ? exc_page_fault+0x62/0x150 [130.250178] entry_SYSCALL_64_after_hwframe+0x63/0xcd [130.255243] RIP: 0033:0x7f6463a3f5ab [130.258830] Code: 73 01 c3 48 8b 0d 75 a8 1b 00 f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 90 f3 0f 1e fa b8 b0 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 45 a8 1b 00 f7 d8 64 89 01 48 [130.277591] RSP: 002b:00007fff22f972c8 EFLAGS: 00000206 ORIG_RAX: 00000000000000b0 [130.285164] RAX: ffffffff9d9da RBX: 000055b6786edd40 RCX: 00007f6463a3f5ab [130.292303] RDX: 0000000000000000 RSI: 0000000000000800 RDI: 000055b6786edda8 [130.299443] RBP: 000055b6786edd40 R08: 0000000000000000 R09: 0000000000000000 [130.306584] R10: 00007f6463b9eac0 R11: 00000000000000206 R12: 000055b6786edda8 [130.313731] R13: 0000000000000000 R14: 000055b6786edda8 R15: 00007fff22f995f8 [130.320875] </TASK> [130.323081] ---[end trace eff7156d56b5cf25]--- cat /sys/class/dma/dma0chan*/in_use would get the wrong result. 2 2 2 Test-by: Jie Hai <haijie1@huawei.com>

CVE ID	Description
CVE-2022-49759	In the Linux kernel, the following vulnerability has been resolved: VMCI: Use threaded irq's instead of tasklets The vmci_dispatch_dgs() tasklet function calls vmci_read_data() which uses wait_event() resulting in invalid sleep in an atomic context (and therefore potentially in a deadlock). Use threaded irq's to fix this issue and completely remove usage of tasklets. [20.264639] BUG: sleeping function called from invalid context at drivers/misc/vmw_vmci/vmci_guest.c:145 [20.264643] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 762, name: vmtoolsd [20.264645] preempt_count: 101, expected: 0 [20.264646] RCU nest depth: 0, expected: 0 [20.264647] 1 lock held by vmtoolsd/762: [20.264648] #0: ffff0000874ae440 (sk_lock-AF_VSOCK){+..+.-} {0:0}, at: vsock_connect+0x60/0x330 [vsock] [20.264658] Preemption disabled at: [20.264659] [<ffff80000151d7d8>] vmci_send_datagram+0x44/0xa0 [vmw_vmci] [20.264665] CPU: 0 PID: 762 Comm: vmtoolsd Not tainted 5.19.0-0.rc8.20220727.git39c3c396f813.60.fc37.aarch64 #1 [20.264667] Hardware name: VMware, Inc. VBSA/VBSA, BIOS VEFI 12/31/2020 [20.264668] Call trace: [20.264669] dump_backtrace+0xc4/0x130 [20.264672] show_stack+0x24/0x80 [20.264673] dump_stack_lvl+0x88/0xb4 [20.264676] dump_stack+0x18/0x34 [20.264677] __might_resched+0x1a0/0x280 [20.264679] __might_sleep+0x58/0x90 [20.264681] vmci_read_data+0x74/0x120 [vmw_vmci] [20.264683] vmci_dispatch_dgs+0x64/0x204 [vmw_vmci] [20.264686] tasklet_action_common.constprop.0+0x13c/0x150 [20.264688] tasklet_action+0x40/0x50 [20.264689] __do_softirq+0x23c/0x6b4 [20.264690] __irq_exit_rcu+0x104/0x214 [20.264691] irq_exit_rcu+0x1c/0x50 [20.264693] el1h_interrupt+0x38/0x6c [20.264695] el1h_64_irq_handler+0x18/0x24 [20.264696] el1h_64_irq+0x68/0x6c [20.264697] preempt_count_sub+0xa4/0xe0 [20.264698] _raw_spin_unlock_irqrestore+0x64/0xb0 [20.264701] vmci_send_datagram+0x7c/0xa0 [vmw_vmci] [20.264703] vmci_datagram_dispatch+0x84/0x100 [vmw_vmci] [20.264706] vmci_datagram_send+0x2c/0x40 [vmw_vmci] [20.264709] vmci_transport_send_control_pkt+0xb8/0x120 [vmw_vsock_vmci_transport] [20.264711] vmci_transport_connect+0x40/0x7c [vmw_vsock_vmci_transport] [20.264713] vsock_connect+0x278/0x330 [vsock] [20.264715] __sys_connect_file+0x8c/0xc0 [20.264718] __sys_connect+0x84/0xb4 [20.264720] __arm64_sys_connect+0x2c/0x3c [20.264721] invoke_syscall+0x78/0x100 [20.264723] el0_svc_common.constprop.0+0x68/0x124 [20.264724] do_el0_svc+0x38/0x4c [20.264725] el0_svc+0x60/0x180 [20.264726] el0t_64_sync_handler+0x11c/0x150 [20.264728] el0t_64_sync+0x190/0x194
CVE-2022-49764	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent bpf program recursion for raw tracepoint probes We got report from sysbot [1] about warnings that were caused by bpf program attached to contention_begin raw tracepoint triggering the same tracepoint by using bpf_trace_printk helper that takes trace_printk_lock lock. Call Trace: <TASK> ? trace_event_raw_event_bpf_trace_printk+0x5f/0x90 bpf_trace_printk+0x2b/0xe0 bpf_prog_a9aec6167c091eef_prog+0x1f/0x24 bpf_trace_run2+0x26/0x90 native_queued_spin_lock_slowpath+0x1c6/0x2b0 _raw_spin_lock_irqsave+0x44/0x50 bpf_trace_printk+0x3f/0xe0 bpf_prog_a9aec6167c091eef_prog+0x1f/0x24 bpf_trace_run2+0x26/0x90 native_queued_spin_lock_slowpath+0x1c6/0x2b0 _raw_spin_lock_irqsave+0x44/0x50 bpf_trace_printk+0x3f/0xe0 bpf_prog_a9aec6167c091eef_prog+0x1f/0x24 bpf_trace_run2+0x26/0x90 native_queued_spin_lock_slowpath+0x1c6/0x2b0 _raw_spin_lock_irqsave+0x44/0x50 bpf_trace_printk+0x3f/0xe0 bpf_prog_a9aec6167c091eef_prog+0x1f/0x24 bpf_trace_run2+0x26/0x90 native_queued_spin_lock_slowpath+0x1c6/0x2b0 _raw_spin_lock_irqsave+0x44/0x50 __unfreeze_partials+0x5b/0x160 ... The can be reproduced by attaching bpf program as raw tracepoint on contention_begin tracepoint. The bpf prog calls bpf_trace_printk helper. Then by running perf bench the spin lock code is forced to take slow path and call contention_begin tracepoint. Fixing this by skipping execution of the bpf program if it's already running, Using bpf prog 'active' field, which is being currently used by trampoline programs for the same reason. Moving bpf_prog_inc_misses_counter to syscall.c because trampoline.c is compiled in just for CONFIG_BPF_JIT option. [1] https://lore.kernel.org/bpf/YxhFe3EwqchC%2FfYf@krava/T/#t
CVE-2022-49766	In the Linux kernel, the following vulnerability has been resolved: netlink: Bounds-check struct nlmsgerr creation In preparation for FORTIFY_SOURCE doing bounds-check on memcpy(), switch from __nlmsg_put to nlmsg_put(), and explain the bounds check for dealing with the memcpy() across a composite flexible array struct. Avoids this future run-time warning: memcpy: detected field-spanning write (size 32) of single field "&errmsg->msg" at net/netlink/af_netlink.c:2447 (size 16)
CVE-2022-49769	In the Linux kernel, the following vulnerability has been resolved: gfs2: Check sb_bsize_shift after reading superblock Fuzzers like to scribble over sb_bsize_shift but in reality it's very unlikely that this field would be corrupted on its own. Nevertheless it should be checked to avoid the possibility of messy mount errors due to bad calculations. It's always a fixed value based on the block size so we can just check that it's the expected value. Tested with: mkfs.gfs2 -O -p lock_nolock /dev/vdb for i in 0 -1 64 65 32 33; do gfs2_edit -p sb field sb_bsize_shift \$i /dev/vdb mount /dev/vdb /mnt/test && umount /mnt/test done Before this patch we get a withdraw after [76.413681] gfs2: fsid=loop0.0: fatal: invalid metadata block [76.413681] bh = 19 (type: exp=5, found=4) [76.413681] function = gfs2_meta_buffer, file = fs/gfs2/meta_io.c, line = 492 and with UBSAN configured we also get complaints like [76.373395] UBSAN: shift-out-of-bounds in fs/gfs2/ops_fstype.c:295:19 [76.373815] shift exponent 4294967287 is too large for 64-bit type 'long unsigned int' After the patch, these complaints don't appear, mount fails immediately and we get an explanation in dmesg.

CVE ID	Description
CVE-2022-49771	In the Linux kernel, the following vulnerability has been resolved: dm ioctl: fix misbehavior if list_versions races with module loading __list_versions will first estimate the required space using the "dm_target_iterate(list_version_get_needed, &needed)" call and then will fill the space using the "dm_target_iterate(list_version_get_info, &iter_info)" call. Each of these calls locks the targets using the "down_read(&_lock)" and "up_read(&_lock)" calls, however between the first and second "dm_target_iterate" there is no lock held and the target modules can be loaded at this point, so the second "dm_target_iterate" call may need more space than what was the first "dm_target_iterate" returned. The code tries to handle this overflow (see the beginning of list_version_get_info), however this handling is incorrect. The code sets "param->data_size = param->data_start + needed" and "iter_info.end = (char *)vers+len" - "needed" is the size returned by the first dm_target_iterate call; "len" is the size of the buffer allocated by userspace. "len" may be greater than "needed"; in this case, the code will write up to "len" bytes into the buffer, however param->data_size is set to "needed", so it may write data past the param->data_size value. The ioctl interface copies only up to param->data_size into userspace, thus part of the result will be truncated. Fix this bug by setting "iter_info.end = (char *)vers + needed;" - this guarantees that the second "dm_target_iterate" call will write only up to the "needed" buffer and it will exit with "DM_BUFFER_FULL_FLAG" if it overflows the "needed" space - in this case, userspace will allocate a larger buffer and retry. Note that there is also a bug in list_version_get_needed - we need to add "strlen(tt->name) + 1" to the needed size, not "strlen(tt->name)".
CVE-2022-49772	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Drop snd_BUG_ON() from snd_usbmidi_output_open() snd_usbmidi_output_open() has a check of the NULL port with snd_BUG_ON(). snd_BUG_ON() was used as this shouldn't have happened, but in reality, the NULL port may be seen when the device gives an invalid endpoint setup at the descriptor, hence the driver skips the allocation. That is, the check itself is valid and snd_BUG_ON() should be dropped from there. Otherwise it's confusing as if it were a real bug, as recently syzbot stumbled on it.
CVE-2022-49773	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix optc2_configure warning on dcn314 [Why] dcn314 uses optc2_configure_crc() that wraps optc1_configure_crc() + set additional registers not applicable to dcn314. It's not critical but when used leads to warning like: WARNING: drivers/gpu/drm/amd/amdgpu/./display/dc/dc_helper.c Call Trace: <TASK> generic_reg_set_ex+0x6d/0xe0 [amdgpu] optc2_configure_crc+0x60/0x80 [amdgpu] dc_stream_configure_crc+0x129/0x150 [amdgpu] amdgpu_dm_crc_configure_crc_source+0x5d/0xe0 [amdgpu] [How] Use optc1_configure_crc() directly
CVE-2022-49776	In the Linux kernel, the following vulnerability has been resolved: macvlan: enforce a consistent minimal mtu macvlan should enforce a minimal mtu of 68, even at link creation. This patch avoids the current behavior (which could lead to crashes in ipv6 stack if the link is brought up) \$ ip link add macvlan1 link eno1 mtu 8 type macvlan # This should fail ! \$ ip link sh dev macvlan1 5: macvlan1@eno1: <BROADCAST,MULTICAST> mtu 8 qdisc noop state DOWN mode DEFAULT group default qlen 1000 link/ether 02:47:6c:24:74:82 brd ff:ff:ff:ff:ff:ff \$ ip link set macvlan1 mtu 67 Error: mtu less than device minimum. \$ ip link set macvlan1 mtu 68 \$ ip link set macvlan1 mtu 8 Error: mtu less than device minimum.
CVE-2022-49779	In the Linux kernel, the following vulnerability has been resolved: kprobes: Skip clearing aggrprobe's post_handler in kprobe-on-ftrace case In __unregister_kprobe_top(), if the currently unregistered probe has post_handler but other child probes of the aggrprobe do not have post_handler, the post_handler of the aggrprobe is cleared. If this is a ftrace-based probe, there is a problem. In later calls to disarm_kprobe(), we will use kprobe_ftrace_ops because post_handler is NULL. But we're armed with kprobe_ipmodify_ops. This triggers a WARN in __disarm_kprobe_ftrace() and may even cause use-after-free: Failed to disarm kprobe-ftrace at kernel_clone+0x0/0x3c0 (error -2) WARNING: CPU: 5 PID: 137 at kernel/kprobes.c:1135 __disarm_kprobe_ftrace.isra.21+0xcf/0xe0 Modules linked in: testKprobe_007(-) CPU: 5 PID: 137 Comm: rmmmod Not tainted 6.1.0-rc4-dirty #18 [...] Call Trace: <TASK> __disable_kprobe+0xcd/0xe0 __unregister_kprobe_top+0x12/0x150 ? mutex_lock+0xe/0x30 unregister_kprobes.part.23+0x31/0xa0 unregister_kprobe+0x32/0x40 __x64_sys_delete_module+0x15e/0x260 ? do_user_addr_fault+0x2cd/0x6b0 do_syscall_64+0x3a/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd [...] For the kprobe-on-ftrace case, we keep the post_handler setting to identify this aggrprobe armed with kprobe_ipmodify_ops. This way we can disarm it correctly.
CVE-2022-49780	In the Linux kernel, the following vulnerability has been resolved: scsi: target: tcm_loop: Fix possible name leak in tcm_loop_setup_hba_bus() If device_register() fails in tcm_loop_setup_hba_bus(), the name allocated by dev_set_name() need be freed. As comment of device_register() says, it should use put_device() to give up the reference in the error path. So fix this by calling put_device(), then the name can be freed in kobject_cleanup(). The 'tl_hba' will be freed in tcm_loop_release_adapter(), so it don't need goto error label in this case.

CVE ID	Description
CVE-2022-49781	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd: Fix crash due to race between amd_pmu_enable_all, perf NMI and throttling amd_pmu_enable_all() does: if (!test_bit(idx, cpuc->active_mask)) continue; amd_pmu_enable_event(cpuc->events[idx]); A perf NMI of another event can come between these two steps. Perf NMI handler internally disables and enables _all_ events, including the one which nmi-intercepted amd_pmu_enable_all() was in process of enabling. If that unintentionally enabled event has very low sampling period and causes immediate successive NMI, causing the event to be throttled, cpuc->events[idx] and cpuc->active_mask gets cleared by x86_pmu_stop(). This will result in amd_pmu_enable_event() getting called with event=NULL when amd_pmu_enable_all() resumes after handling the NMIs. This causes a kernel crash: BUG: kernel NULL pointer dereference, address: 0000000000000198 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page [...] Call Trace: <TASK> amd_pmu_enable_all+0x68/0xb0 ctx_resched+0xd9/0x150 event_function+0xb8/0x130 ? hrtimer_start_range_ns+0x141/0x4a0 ? perf_duration_warn+0x30/0x30 remote_function+0x4d/0x60 __flush_smp_call_function_queue+0xc4/0x500 flush_smp_call_function_queue+0x11d/0x1b0 do_idle+0x18f/0x2d0 cpu_startup_entry+0x19/0x20 start_secondary+0x121/0x160 secondary_startup_64_no_verify+0xe5/0xeb </TASK> amd_pmu_disable_all()/amd_pmu_enable_all() calls inside perf NMI handler were recently added as part of BRS enablement but I'm not sure whether we really need them. We can just disable BRS in the beginning and enable it back while returning from NMI. This will solve the issue by not enabling those events whose active_masks are set but are not yet enabled in hw pmu.
CVE-2022-49782	In the Linux kernel, the following vulnerability has been resolved: perf: Improve missing SIGTRAP checking To catch missing SIGTRAP we employ a WARN in __perf_event_overflow(), which fires if pending_sigtrap was already set: returning to user space without consuming pending_sigtrap, and then having the event fire again would re-enter the kernel and trigger the WARN. This, however, seemed to miss the case where some events not associated with progress in the user space task can fire and the interrupt handler runs before the IRQ work meant to consume pending_sigtrap (and generate the SIGTRAP). syzbot gifted us this stack trace: WARNING: CPU: 0 PID: 3607 at kernel/events/core.c:9313 __perf_event_overflow Modules linked in: CPU: 0 PID: 3607 Comm: syz-executor100 Not tainted 6.1.0-rc2-syzkaller-00073-g88619e77b33d #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/11/2022 RIP: 0010: __perf_event_overflow+0x498/0x540 kernel/events/core.c:9313 <...> Call Trace: <TASK> perf_swevent_hrtimer+0x34f/0x3c0 kernel/events/core.c:10729 __run_hrtimer kernel/time/hrtimer.c:1685 [inline] __hrtimer_run_queues+0x1c6/0xfb0 kernel/time/hrtimer.c:1749 hrtimer_interrupt+0x31c/0x790 kernel/time/hrtimer.c:1811 local_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1096 [inline] __sysvec_apic_timer_interrupt+0x17c/0x640 arch/x86/kernel/apic/apic.c:1113 sysvec_apic_timer_interrupt+0x40/0xc0 arch/x86/kernel/apic/apic.c:1107 asm_sysvec_apic_timer_interrupt+0x16/0x20 arch/x86/include/asm/identrty.h:649 <...> </TASK> In this case, syzbot produced a program with event type PERF_TYPE_SOFTWARE and config PERF_COUNT_SW_CPU_CLOCK. The hrtimer manages to fire again before the IRQ work got a chance to run, all while never having returned to user space. Improve the WARN to check for real progress in user space: approximate this by storing a 32-bit hash of the current IP into pending_sigtrap, and if an event fires while pending_sigtrap still matches the previous IP, we assume no progress (false negatives are possible given we could return to user space and trigger again on the same IP).
CVE-2022-49783	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Drop fpregs lock before inheriting FPU permissions Mike Galbraith reported the following against an old fork of preempt-rt but the same issue also applies to the current preempt-rt tree. BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:46 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1, name: systemd preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 Preemption disabled at: fpu_clone CPU: 6 PID: 1 Comm: systemd Tainted: G E (unreleased) Call Trace: <TASK> dump_stack_lvl ? fpu_clone __might_resched rt_spin_lock fpu_clone ? copy_thread ? copy_process ? shmem_alloc_inode ? kmem_cache_alloc ? kernel_clone ? __do_sys_clone ? do_syscall_64 ? __x64_sys_rt_sigprocmask ? syscall_exit_to_user_mode ? do_syscall_64 ? syscall_exit_to_user_mode ? do_syscall_64 ? syscall_exit_to_user_mode ? do_syscall_64 ? syscall_exit_to_user_mode ? do_syscall_64 ? exc_page_fault ? entry_SYSCALL_64_after_hwframe </TASK> Mike says: The splat comes from fpu_inherit_perms() being called under fpregs_lock(), and us reaching the spin_lock_irq() therein due to fpu_state_size_dynamic() returning true despite static key __fpu_state_size_dynamic having never been enabled. Mike's assessment looks correct. fpregs_lock on a PREEMPT_RT kernel disables preemption so calling spin_lock_irq() in fpu_inherit_perms() is unsafe. This problem exists since commit 9e798e9aa14c ("x86/fpu: Prepare fpu_clone() for dynamically enabled features"). Even though the original bug report should not have enabled the paths at all, the bug still exists. fpregs_lock is necessary when editing the FPU registers or a task's FP state but it is not necessary for fpu_inherit_perms(). The only write of any FP state in fpu_inherit_perms() is for the new child which is not running yet and cannot context switch or be borrowed by a kernel thread yet. Hence, fpregs_lock is not protecting anything in the new child until clone() completes and can be dropped earlier. The siglock still needs to be acquired by fpu_inherit_perms() as the read of the parent's permissions has to be serialised. [bp: Cleanup splat.]

CVE ID	Description
CVE-2022-49784	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd/uncore: Fix memory leak for events array When a CPU comes online, the per-CPU NB and LLC uncore contexts are freed but not the events array within the context structure. This causes a memory leak as identified by the kmemleak detector. [...] unreferenced object 0xffff8c5944b8e320 (size 32): comm "swapper/0", pid 1, jiffies 4294670387 (age 151.072s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000759fb79>] amd_uncore_cpu_up_prepare+0xaf/0x230 [<00000000ddc9e126>] cpuhp_invoke_callback+0x2cf/0x470 [<0000000093e727d4>] cpuhp_issue_call+0x14d/0x170 [<0000000045464d54>] __cpuhp_setup_state_cpuslocked+0x11e/0x330 [<0000000069f67cbd>] __cpuhp_setup_state+0x6b/0x110 [<0000000015365e0f>] amd_uncore_init+0x260/0x321 [<00000000089152d2>] do_one_initcall+0x3f/0x1f0 [<000000002d0bd18d>] kernel_init_freeable+0x1ca/0x212 [<0000000030be8dde>] kernel_init+0x11/0x120 [<0000000059709e59>] ret_from_fork+0x22/0x30 unreferenced object 0xffff8c5944b8dd40 (size 64): comm "swapper/0", pid 1, jiffies 4294670387 (age 151.072s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000306efe8b>] amd_uncore_cpu_up_prepare+0x183/0x230 [<00000000ddc9e126>] cpuhp_invoke_callback+0x2cf/0x470 [<0000000093e727d4>] cpuhp_issue_call+0x14d/0x170 [<0000000045464d54>] __cpuhp_setup_state_cpuslocked+0x11e/0x330 [<0000000069f67cbd>] __cpuhp_setup_state+0x6b/0x110 [<0000000015365e0f>] amd_uncore_init+0x260/0x321 [<00000000089152d2>] do_one_initcall+0x3f/0x1f0 [<000000002d0bd18d>] kernel_init_freeable+0x1ca/0x212 [<0000000030be8dde>] kernel_init+0x11/0x120 [<0000000059709e59>] ret_from_fork+0x22/0x30 [...] Fix the problem by freeing the events array before freeing the uncore context.
CVE-2022-49785	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Add overflow check in sgx_validate_offset_length() sgx_validate_offset_length() function verifies "offset" and "length" arguments provided by userspace, but was missing an overflow check on their addition. Add it.
CVE-2022-49789	In the Linux kernel, the following vulnerability has been resolved: scsi: zfcp: Fix double free of FSF request when qdio send fails We used to use the wrong type of integer in 'zfcp_fsf_req_send()' to cache the FSF request ID when sending a new FSF request. This is used in case the sending fails and we need to remove the request from our internal hash table again (so we don't keep an invalid reference and use it when we free the request again). In 'zfcp_fsf_req_send()' we used to cache the ID as 'int' (signed and 32 bit wide), but the rest of the zfcp code (and the firmware specification) handles the ID as 'unsigned long'/'u64' (unsigned and 64 bit wide [s390x ELF ABI]). For one this has the obvious problem that when the ID grows past 32 bit (this can happen reasonably fast) it is truncated to 32 bit when storing it in the cache variable and so doesn't match the original ID anymore. The second less obvious problem is that even when the original ID has not yet grown past 32 bit, as soon as the 32nd bit is set in the original ID (0x80000000 = 2^147'483'648) we will have a mismatch when we cast it back to 'unsigned long'. As the cached variable is of a signed type, the compiler will choose a sign-extending instruction to load the 32 bit variable into a 64 bit register (e.g.: 'lgr %r11,188(%r15)'). So once we pass the cached variable into 'zfcp_reqlist_find_rm()' to remove the request again all the leading zeros will be flipped to ones to extend the sign and won't match the original ID anymore (this has been observed in practice). If we can't successfully remove the request from the hash table again after 'zfcp_qdio_send()' fails (this happens regularly when zfcp cannot notify the adapter about new work because the adapter is already gone during e.g. a ChpID toggle) we will end up with a double free. We unconditionally free the request in the calling function when 'zfcp_fsf_req_send()' fails, but because the request is still in the hash table we end up with a stale memory reference, and once the zfcp adapter is either reset during recovery or shutdown we end up freeing the same memory twice. The resulting stack traces vary depending on the kernel and have no direct correlation to the place where the bug occurs. Here are three examples that have been seen in practice: list_del corruption. next->prev should be 00000001b9d13800, but was 00000000dead4ead. (next=00000001bd131a00) -----[cut here]----- kernel BUG at lib/list_debug.c:62! monitor event: 0040 ilc:2 [#1] PREEMPT SMP Modules linked in: ... CPU: 9 PID: 1617 Comm: zfcp0.0.1740 Kdump: loaded Hardware name: ... Krlnl PSW : 0704d00180000000 000000003cbeea1f8 (__list_del_entry_valid +0x98/0x140) R:0 T:1 IO:1 EX:1 Key:0 M:1 W:0 P:0 AS:3 CC:1 PM:0 RI:0 EA:3 Krlnl GPRS: 00000000916d12f1 0000000080000000 000000000000006d 000000003cb665cd6 0000000000000001 0000000000000000 0000000000000000 00000000d28d21e8 00000000d3844000 00000380099efd28 000000001bd131a0 000000001b9d13800 000000000d3290100 0000000000000000 000000003cbeea1f4 00000380099efc70 Krlnl Code: 000000003cbeea1e8: c020004f68a7 larl %r2,000000003cc8d7336 000000003cbeea1ee: c0e50027fd65 brasl %r14,000000003cc3e9cb8 #000000003cbeea1f4: af000000 mc 0,0 >000000003cbeea1f8: c02000920440 larl %r2,000000003cd12aa78 000000003cbeea1fe: c0e500289c25 brasl %r14,000000003cc3fda48 000000003cbeea204: b9040043 lgr %r4,%r3 000000003cbeea208: b9040051 lgr %r5,%r1 000000003cbeea20c: b9040032 lgr %r3,%r2 Call Trace: [<000000003cbeea1f8>] __list_del_entry_valid+0x98/0x140 ([<000000003cbeea1f4>] __list_del_entry_valid+0x94/0x140) [<000003ff7ff502fe>] zfcp_fsf_req_dismiss_all+0xde/0x150 [zfcp] [<000003ff7ff49cd0>] zfcp_erp_strategy_do_action+0x160/0x280 [zfcp] ---truncated---

CVE ID	Description
CVE-2022-49802	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix null pointer dereference in ftrace_add_mod() The @ftrace_mod is allocated by kzalloc(), so both the members {prev,next} of @ftrace_mode->list are NULL, it's not a valid state to call list_del(). If kstrdup() for @ftrace_mod->{func module} fails, it goes to @out_free tag and calls free_ftrace_mod() to destroy @ftrace_mod, then list_del() will write prev->next and next->prev, where null pointer dereference happens. BUG: kernel NULL pointer dereference, address: 0000000000000008 Oops: 0002 [#1] PREEMPT SMP NOPTI Call Trace: <TASK> ftrace_mod_callback+0x20d/0x220 ? do_filp_open+0xd9/0x140 ftrace_process_regex.isra.51+0xbf/0x130 ftrace_regex_write.isra.52.part.53+0x6e/0x90 vfs_write+0xee/0x3a0 ? __audit_filter_op+0xb1/0x100 ? auditd_test_task+0x38/0x50 ksys_write+0xa5/0xe0 do_syscall_64+0x3a/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd Kernel panic - not syncing: Fatal exception So call INIT_LIST_HEAD() to initialize the list member to fix this issue.
CVE-2022-49803	In the Linux kernel, the following vulnerability has been resolved: netdevsim: Fix memory leak of nsim_dev->fa_cookie kmemleak reports this issue: unreferenced object 0xffff8881bac872d0 (size 8): comm "sh", pid 58603, jiffies 4481524462 (age 68.065s) hex dump (first 8 bytes): 04 00 00 00 de ad be ef backtrace: [<00000000c80b8577>] __kmalloc+0x49/0x150 [<000000005292b8c6>] nsim_dev_trap_fa_cookie_write+0xc1/0x210 [netdevsim] [<0000000093d78e77>] full_proxy_write+0xf3/0x180 [<000000005a662c16>] vfs_write+0x1c5/0xaf0 [<000000007aabf84a>] ksys_write+0xed/0x1c0 [<000000005f1d2e47>] do_syscall_64+0x3b/0x90 [<000000006001c6ec>] entry_SYSCALL_64_after_hwframe+0x63/0xcd The issue occurs in the following scenarios: nsim_dev_trap_fa_cookie_write() kmalloc() fa_cookie nsim_dev->fa_cookie = fa_cookie .. nsim_drv_remove() The fa_cookie allocated in nsim_dev_trap_fa_cookie_write() is not freed. To fix, add kfree(nsim_dev->fa_cookie) to nsim_drv_remove().
CVE-2022-49812	In the Linux kernel, the following vulnerability has been resolved: bridge: switchdev: Fix memory leaks when changing VLAN protocol The bridge driver can offload VLANs to the underlying hardware either via switchdev or the 8021q driver. When the former is used, the VLAN is marked in the bridge driver with the 'BR_VLFLAG_ADDED_BY_SWITCHDEV' private flag. To avoid the memory leaks mentioned in the cited commit, the bridge driver will try to delete a VLAN via the 8021q driver if the VLAN is not marked with the previously mentioned flag. When the VLAN protocol of the bridge changes, switchdev drivers are notified via the 'SWITCHDEV_ATTR_ID_BRIDGE_VLAN_PROTOCOL' attribute, but the 8021q driver is also called to add the existing VLANs with the new protocol and delete them with the old protocol. In case the VLANs were offloaded via switchdev, the above behavior is both redundant and buggy. Redundant because the VLANs are already programmed in hardware and drivers that support VLAN protocol change (currently only mlx5) change the protocol upon the switchdev attribute notification. Buggy because the 8021q driver is called despite these VLANs being marked with 'BR_VLFLAG_ADDED_BY_SWITCHDEV'. This leads to memory leaks [1] when the VLANs are deleted. Fix by not calling the 8021q driver for VLANs that were already programmed via switchdev. [1] unreferenced object 0xffff8881f6771200 (size 256): comm "ip", pid 446855, jiffies 4298238841 (age 55.240s) hex dump (first 32 bytes): 00 00 7f 0e 83 88 ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000012819ac>] vlan_vid_add+0x437/0x750 [<00000000f2281fad>] __br_vlan_set_proto+0x289/0x920 [<000000000632b56f>] br_changelink+0x3d6/0x13f0 [<0000000089d25f04>] __rtnl_newlink+0x8ae/0x14c0 [<00000000f6276baf>] rtnl_newlink+0x5f/0x90 [<00000000746dc902>] rtnetlink_rcv_msg+0x336/0xa00 [<000000001c2241c0>] netlink_rcv_skb+0x11d/0x340 [<0000000010588814>] netlink_unicast+0x438/0x710 [<00000000e1a4cd5c>] netlink_sendmsg+0x788/0xc40 [<00000000e8992d4e>] sock_sendmsg+0xb0/0xe0 [<000000000621b8f91>] ____sys_sendmsg+0x4ff/0x6d0 [<000000000ea26996>] __sys_sendmsg+0x12e/0x1b0 [<00000000684f7e25>] __sys_sendmsg+0xab/0x130 [<000000004538b104>] do_syscall_64+0x3d/0x90 [<0000000091ed9678>] entry_SYSCALL_64_after_hwframe+0x46/0xb0
CVE-2022-49813	In the Linux kernel, the following vulnerability has been resolved: net: ena: Fix error handling in ena_init() The ena_init() won't destroy workqueue created by create_singlethread_workqueue() when pci_register_driver() failed. Call destroy_workqueue() when pci_register_driver() failed to prevent the resource leak.
CVE-2022-49817	In the Linux kernel, the following vulnerability has been resolved: net: mhi: Fix memory leak in mhi_net_dellink() MHI driver registers network device without setting the needs_free_netdev flag, and does NOT call free_netdev() when unregisters network device, which causes a memory leak. This patch calls free_netdev() to fix it since netdev_priv is used after unregister.
CVE-2022-49821	In the Linux kernel, the following vulnerability has been resolved: mISDN: fix possible memory leak in mISDN_dsp_element_register() After commit 1fa5ae857bb1 ("driver core: get rid of struct device's bus_id string array"), the name of device is allocated dynamically, use put_device() to give up the reference, so that the name can be freed in kobject_cleanup() when the refcount is 0. The 'entry' is going to be freed in mISDN_dsp_dev_release(), so the kfree() is removed. list_del() is called in mISDN_dsp_dev_release(), so it need be initialized.

CVE ID	Description
CVE-2022-49822	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix connections leak when tlink setup failed If the tlink setup failed, lost to put the connections, then the module refcnt leak since the cifsd kthread not exit. Also leak the fscache info, and for next mount with fsc, it will print the follow errors: CIFS: Cache volume key already in use (cifs,127.0.0.1:445,TEST) Let's check the result of tlink setup, and do some cleanup.
CVE-2022-49823	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix error handling in ata_tdev_add() In ata_tdev_add(), the return value of transport_add_device() is not checked. As a result, it causes null-ptr-deref while removing the module, because transport_remove_device() is called to remove the device that was not added. Unable to handle kernel NULL pointer dereference at virtual address 00000000000000d0 CPU: 13 PID: 13603 Comm: rmmmod Kdump: loaded Tainted: G W 6.1.0-rc3+ #36 pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPEDev=) pc : device_del+0x48/0x3a0 lr : device_del+0x44/0x3a0 Call trace: device_del+0x48/0x3a0 attribute_container_class_device_del+0x28/0x40 transport_remove_classdev+0x60/0x7c attribute_container_device_trigger+0x118/0x120 transport_remove_device+0x20/0x30 ata_tdev_delete+0x24/0x50 [libata] ata_tlink_delete+0x40/0xa0 [libata] ata_tport_delete+0x2c/0x60 [libata] ata_port_detach+0x148/0x1b0 [libata] ata_pci_remove_one+0x50/0x80 [libata] ahci_remove_one+0x4c/0x8c [ahci] Fix this by checking and handling return value of transport_add_device() in ata_tdev_add(). In the error path, device_del() is called to delete the device which was added earlier in this function, and ata_tdev_free() is called to free ata_dev.
CVE-2022-49824	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix error handling in ata_tlink_add() In ata_tlink_add(), the return value of transport_add_device() is not checked. As a result, it causes null-ptr-deref while removing the module, because transport_remove_device() is called to remove the device that was not added. Unable to handle kernel NULL pointer dereference at virtual address 00000000000000d0 CPU: 33 PID: 13850 Comm: rmmmod Kdump: loaded Tainted: G W 6.1.0-rc3+ #12 pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPEDev=) pc : device_del+0x48/0x39c lr : device_del+0x44/0x39c Call trace: device_del+0x48/0x39c attribute_container_class_device_del+0x28/0x40 transport_remove_classdev+0x60/0x7c attribute_container_device_trigger+0x118/0x120 transport_remove_device+0x20/0x30 ata_tlink_delete+0x88/0xb0 [libata] ata_tport_delete+0x2c/0x60 [libata] ata_port_detach+0x148/0x1b0 [libata] ata_pci_remove_one+0x50/0x80 [libata] ahci_remove_one+0x4c/0x8c [ahci] Fix this by checking and handling return value of transport_add_device() in ata_tlink_add().
CVE-2022-49825	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix error handling in ata_tport_add() In ata_tport_add(), the return value of transport_add_device() is not checked. As a result, it causes null-ptr-deref while removing the module, because transport_remove_device() is called to remove the device that was not added. Unable to handle kernel NULL pointer dereference at virtual address 00000000000000d0 CPU: 12 PID: 13605 Comm: rmmmod Kdump: loaded Tainted: G W 6.1.0-rc3+ #8 pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPEDev=) pc : device_del+0x48/0x39c lr : device_del+0x44/0x39c Call trace: device_del+0x48/0x39c attribute_container_class_device_del+0x28/0x40 transport_remove_classdev+0x60/0x7c attribute_container_device_trigger+0x118/0x120 transport_remove_device+0x20/0x30 ata_tport_delete+0x34/0x60 [libata] ata_port_detach+0x148/0x1b0 [libata] ata_pci_remove_one+0x50/0x80 [libata] ahci_remove_one+0x4c/0x8c [ahci] Fix this by checking and handling return value of transport_add_device() in ata_tport_add().
CVE-2022-49826	In the Linux kernel, the following vulnerability has been resolved: ata: libata-transport: fix double ata_host_put() in ata_tport_add() In the error path in ata_tport_add(), when calling put_device(), ata_tport_release() is called, it will put the refcount of 'ap->host'. And then ata_host_put() is called again, the refcount is decreased to 0, ata_host_release() is called, all ports are freed and set to null. When unbinding the device after failure, ata_host_stop() is called to release the resources, it leads a null-ptr-deref(), because all the ports all freed and null. Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 CPU: 7 PID: 18671 Comm: modprobe Kdump: loaded Tainted: G E 6.1.0-rc3+ #8 pstate: 80400009 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPEDev=) pc : ata_host_stop+0x3c/0x84 [libata] lr : release_nodes+0x64/0xd0 Call trace: ata_host_stop+0x3c/0x84 [libata] release_nodes+0x64/0xd0 devres_release_all+0xbc/0x1b0 device_unbind_cleanup+0x20/0x70 really_probe+0x158/0x320 __driver_probe_device+0x84/0x120 driver_probe_device+0x44/0x120 __driver_attach+0xb4/0x220 bus_for_each_dev+0x78/0xdc driver_attach+0x2c/0x40 bus_add_driver+0x184/0x240 driver_register+0x80/0x13c __pci_register_driver+0x4c/0x60 ahci_pci_driver_init+0x30/0x1000 [ahci] Fix this by removing redundant ata_host_put() in the error path.

CVE ID	Description
CVE-2022-49828	In the Linux kernel, the following vulnerability has been resolved: hugetlbfs: don't delete error page from pagecache This change is very similar to the change that was made for shmем [1], and it solves the same problem but for HugeTLBFS instead. Currently, when poison is found in a HugeTLB page, the page is removed from the page cache. That means that attempting to map or read that hugepage in the future will result in a new hugepage being allocated instead of notifying the user that the page was poisoned. As [1] states, this is effectively memory corruption. The fix is to leave the page in the page cache. If the user attempts to use a poisoned HugeTLB page with a syscall, the syscall will fail with EIO, the same error code that shmем uses. For attempts to map the page, the thread will get a BUS_MCEERR_AR SIGBUS. [1]: commit a76054266661 ("mm: shmем: don't truncate page if memory failure happens")
CVE-2022-49829	In the Linux kernel, the following vulnerability has been resolved: drm/scheduler: fix fence ref counting We leaked dependency fences when processes were beeing killed. Additional to that grab a reference to the last scheduled fence.
CVE-2022-49832	In the Linux kernel, the following vulnerability has been resolved: pinctrl: devicetree: fix null pointer dereferencing in pinctrl_dt_to_map Here is the BUG report by KASAN about null pointer dereference: BUG: KASAN: null-ptr-deref in strcmp+0x2e/0x50 Read of size 1 at addr 0000000000000000 by task python3/2640 Call Trace: strcmp __of_find_property of_find_property pinctrl_dt_to_map kasprintf() would return NULL pointer when kmalloc() fail to allocate. So directly return ENOMEM, if kasprintf() return NULL pointer.
CVE-2022-49835	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: fix potential memleak in 'add_widget_node' As 'kobject_add' may allocated memory for 'kobject->name' when return error. And in this function, if call 'kobject_add' failed didn't free kobject. So call 'kobject_put' to recycling resources.
CVE-2022-49837	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix memory leaks in __check_func_call kmemleak reports this issue: unreferenced object 0xffff88817139d000 (size 2048): comm "test_progs", pid 33246, jiffies 4307381979 (age 45851.820s) hex dump (first 32 bytes): 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000045f075f0>] kmalloc_trace+0x27/0xa0 [<00000000098b7c90a>] __check_func_call+0x316/0x1230 [<000000000b4c3c403>] check_helper_call+0x172e/0x4700 [<000000000aa3875b7>] do_check+0x21d8/0x45e0 [<0000000001147357b>] do_check_common+0x767/0xaf0 [<000000000b5a595b4>] bpf_check+0x43e3/0x5bc0 [<00000000011e391b1>] bpf_prog_load+0xf26/0x1940 [<00000000007f765c0>] __sys_bpf+0xd2c/0x3650 [<000000000839815d6>] __x64_sys_bpf+0x75/0xc0 [<000000000946ee250>] do_syscall_64+0x3b/0x90 [<0000000000506b7f>] entry_SYSCALL_64_after_hwframe+0x63/0xcd The root case here is: In function prepare_func_exit(), the callee is not released in the abnormal scenario after "state->curframe--;". To fix, move "state->curframe--;" to the very bottom of the function, right when we free callee and reset frame[] pointer to NULL, as Andrii suggested. In addition, function __check_func_call() has a similar problem. In the abnormal scenario before "state->curframe++;", the callee also should be released by free_func_state().
CVE-2022-49838	In the Linux kernel, the following vulnerability has been resolved: sctp: clear out_curr if all frag chunks of current msg are pruned A crash was reported by Zhen Chen: list_del corruption, ffffa035ddf01c18->next is NULL WARNING: CPU: 1 PID: 250682 at lib/list_debug.c:49 __list_del_entry_valid+0x59/0xe0 RIP: 0010: __list_del_entry_valid+0x59/0xe0 Call Trace: sctp_sched_dequeue_common+0x17/0x70 [sctp] sctp_sched_fcfs_dequeue+0x37/0x50 [sctp] sctp_outq_flush_data+0x85/0x360 [sctp] sctp_outq_uncork+0x77/0xa0 [sctp] sctp_cmd_interpreter.constprop.0+0x164/0x1450 [sctp] sctp_side_effects+0x37/0xe0 [sctp] sctp_do_sm+0xd0/0x230 [sctp] sctp_primitive_SEND+0x2f/0x40 [sctp] sctp_sendmsg_to_asoc+0x3fa/0x5c0 [sctp] sctp_sendmsg+0x3d5/0x440 [sctp] sock_sendmsg+0x5b/0x70 and in sctp_sched_fcfs_dequeue() it dequeued a chunk from stream out_curr outq while this outq was empty. Normally stream->out_curr must be set to NULL once all frag chunks of current msg are dequeued, as we can see in sctp_sched_dequeue_done(). However, in sctp_prsctp_prune_unsent() as it is not a proper dequeue, sctp_sched_dequeue_done() is not called to do this. This patch is to fix it by simply setting out_curr to NULL when the last frag chunk of current msg is dequeued from out_curr stream in sctp_prsctp_prune_unsent().

CVE ID	Description
CVE-2022-49839	In the Linux kernel, the following vulnerability has been resolved: scsi: scsi_transport_sas: Fix error handling in sas_phy_add() If transport_add_device() fails in sas_phy_add(), the kernel will crash trying to delete the device in transport_remove_device() called from sas_remove_host(). Unable to handle kernel NULL pointer dereference at virtual address 000000000000108 CPU: 61 PID: 42829 Comm: rmmmod Kdump: loaded Tainted: G W 6.1.0-rc1+ #173 pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYP=--) pc : device_del+0x54/0x3d0 lr : device_del+0x37c/0x3d0 Call trace: device_del+0x54/0x3d0 attribute_container_class_device_del+0x28/0x38 transport_remove_classdev+0x6c/0x80 attribute_container_device_trigger+0x108/0x110 transport_remove_device+0x28/0x38 sas_phy_delete+0x30/0x60 [scsi_transport_sas] do_sas_phy_delete+0x6c/0x80 [scsi_transport_sas] device_for_each_child+0x68/0xb0 sas_remove_children+0x40/0x50 [scsi_transport_sas] sas_remove_host+0x20/0x38 [scsi_transport_sas] hisi_sas_remove+0x40/0x68 [hisi_sas_main] hisi_sas_v2_remove+0x20/0x30 [hisi_sas_v2_hw] platform_remove+0x2c/0x60 Fix this by checking and handling return value of transport_add_device() in sas_phy_add().
CVE-2022-49840	In the Linux kernel, the following vulnerability has been resolved: bpf, test_run: Fix alignment problem in bpf_prog_test_run_skb() We got a syzkaller problem because of aarch64 alignment fault if KFENCE enabled. When the size from user bpf program is an odd number, like 399, 407, etc, it will cause the struct skb_shared_info's unaligned access. As seen below: BUG: KFENCE: use-after-free read in __skb_clone+0x23c/0x2a0 net/core/skbuff.c:1032 Use-after-free read at 0xffff6254ffffac077 (in kfence-#213): __lse_atomic_add arch/arm64/include/asm/atomic_lse.h:26 [inline] arch_atomic_add arch/arm64/include/asm/atomic.h:28 [inline] arch_atomic_inc include/linux/atomic-arch-fallback.h:270 [inline] atomic_inc include/asm-generic/atomic-instrumented.h:241 [inline] __skb_clone+0x23c/0x2a0 net/core/skbuff.c:1032 skb_clone+0xf4/0x214 net/core/skbuff.c:1481 ____bpf_clone_redirect net/core/filter.c:2433 [inline] bpf_clone_redirect+0x78/0x1c0 net/core/filter.c:2420 bpf_prog_d3839dd9068ceb51+0x80/0x330 bpf_dispatcher_nop_func include/linux/bpf.h:728 [inline] bpf_test_run+0x3c0/0x6c0 net/bpf/test_run.c:53 bpf_prog_test_run_skb+0x638/0xa7c net/bpf/test_run.c:594 bpf_prog_test_run kernel/bpf/syscall.c:3148 [inline] __do_sys_bpf kernel/bpf/syscall.c:4441 [inline] __se_sys_bpf+0xad0/0x1634 kernel/bpf/syscall.c:4381 kfence-#213: 0xffff6254ffffac000-0xffff6254ffffac196, size=407, cache=kmalloc-512 allocated by task 15074 on cpu 0 at 1342.585390s: kmalloc include/linux/slab.h:568 [inline] kzalloc include/linux/slab.h:675 [inline] bpf_test_init.isra.0+0xac/0x290 net/bpf/test_run.c:191 bpf_prog_test_run_skb+0x11c/0xa7c net/bpf/test_run.c:512 bpf_prog_test_run kernel/bpf/syscall.c:3148 [inline] __do_sys_bpf kernel/bpf/syscall.c:4441 [inline] __se_sys_bpf+0xad0/0x1634 kernel/bpf/syscall.c:4381 __arm64_sys_bpf+0x50/0x60 kernel/bpf/syscall.c:4381 To fix the problem, we adjust @size so that (@size + @hearoom) is a multiple of SMP_CACHE_BYTES. So we make sure the struct skb_shared_info is aligned to a cache line.
CVE-2022-49842	In the Linux kernel, the following vulnerability has been resolved: ASoC: core: Fix use-after-free in snd_soc_exit() KASAN reports a use-after-free: BUG: KASAN: use-after-free in device_del+0xb5b/0xc60 Read of size 8 at addr fffff88800865050 by task rmmmod/387 CPU: 2 PID: 387 Comm: rmmmod Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) Call Trace: <TASK> dump_stack_lvl+0x79/0x9a print_report+0x17f/0x47b kasan_report+0xbb/0xf0 device_del+0xb5b/0xc60 platform_device_del.part.0+0x24/0x200 platform_device_unregister+0x2e/0x40 snd_soc_exit+0xa/0x22 [snd_soc_core] __do_sys_delete_module.constprop.0+0x34f/0x5b0 do_syscall_64+0x3a/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd ... </TASK> It's because in snd_soc_init(), snd_soc_util_init() is possible to fail, but its ret is ignored, which makes soc_dummy_dev unregistered twice. snd_soc_init() snd_soc_util_init() platform_device_register_simple(soc_dummy_dev) platform_driver_register() # fail platform_device_unregister(soc_dummy_dev) platform_driver_register() # success ... snd_soc_exit() snd_soc_util_exit() # soc_dummy_dev will be unregistered for second time To fix it, handle error and stop snd_soc_init() when util_init() fail. Also clean debugfs when util_init() or driver_register() fail.
CVE-2022-49853	In the Linux kernel, the following vulnerability has been resolved: net: macvlan: fix memory leaks of macvlan_common_newlink kmemleak reports memory leaks in macvlan_common_newlink, as follows: ip link add link eth0 name .. type macvlan mode source macaddr add <MAC-ADDR> kmemleak reports: unreferenced object 0xfffff8880109bb140 (size 64): comm "ip", pid 284, jiffies 4294986150 (age 430.108s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 b8 aa 5a 12 80 88 ff ff Z..... 80 1b fa 0d 80 88 ff ff 1e ff ac af c7 c1 6b 6b kk backtrace: [<ffffffffff813e06a7>] kmem_cache_alloc_trace+0x1c7/0x300 [<ffffffffff81b66025>] macvlan_hash_add_source+0x45/0xd0 [<ffffffffff81b66a67>] macvlan_changelink_sources+0xd7/0x170 [<ffffffffff81b6775c>] macvlan_common_newlink+0x38c/0x5a0 [<ffffffffff81b6797e>] macvlan_newlink+0xe/0x20 [<ffffffffff81d97f8f>] __rtnl_newlink+0x7af/0xa50 [<ffffffffff81d98278>] rtnl_newlink+0x48/0x70 ... In the scenario where the macvlan mode is configured as 'source', macvlan_changelink_sources() will be executed to reconfigure list of remote source mac addresses, at the same time, if register_netdevice() return an error, the resource generated by macvlan_changelink_sources() is not cleaned up. Using this patch, in the case of an error, it will execute macvlan_flush_sources() to ensure that the resource is cleaned up.

CVE ID	Description
CVE-2022-49863	In the Linux kernel, the following vulnerability has been resolved: can: af_can: fix NULL pointer dereference in can_rx_register() It causes NULL pointer dereference when testing as following: (a) use syscall(__NR_socket, 0x10ul, 3ul, 0) to create netlink socket. (b) use syscall(__NR_sendmsg, ...) to create bond link device and vxcan link device, and bind vxcan device to bond device (can also use ifenslave command to bind vxcan device to bond device). (c) use syscall(__NR_socket, 0x1dul, 3ul, 1) to create CAN socket. (d) use syscall(__NR_bind, ...) to bind the bond device to CAN socket. The bond device invokes the can-raw protocol registration interface to receive CAN packets. However, ml_priv is not allocated to the dev, dev_rcv_lists is assigned to NULL in can_rx_register(). In this case, it will occur the NULL pointer dereference issue. The following is the stack information: BUG: kernel NULL pointer dereference, address: 0000000000000008 PGD 122a4067 P4D 122a4067 PUD 1223c067 PMD 0 Oops: 0000 [#1] PREEMPT SMP RIP: 0010:can_rx_register+0x12d/0x1e0 Call Trace: <TASK> raw_enable_filters+0x8d/0x120 raw_enable_allfilters+0x3b/0x130 raw_bind+0x118/0x4f0 __sys_bind+0x163/0x1a0 __x64_sys_bind+0x1e/0x30 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x63/0xcd </TASK>
CVE-2022-49865	In the Linux kernel, the following vulnerability has been resolved: ipv6: addrlabel: fix infoleak when sending struct ifaddrllbmsg to network When copying a 'struct ifaddrllbmsg' to the network, __ifal_reserved remained uninitialized, resulting in a 1-byte infoleak: BUG: KMSAN: kernel-network-infoleak in __netdev_start_xmit ./include/linux/netdevice.h:4841 __netdev_start_xmit ./include/linux/netdevice.h:4841 netdev_start_xmit ./include/linux/netdevice.h:4857 xmit_one net/core/dev.c:3590 dev_hard_start_xmit+0x1dc/0x800 net/core/dev.c:3606 __dev_queue_xmit+0x17e8/0x4350 net/core/dev.c:4256 dev_queue_xmit ./include/linux/netdevice.h:3009 __netlink_deliver_tap_skb net/netlink/af_netlink.c:307 __netlink_deliver_tap+0x728/0xad0 net/netlink/af_netlink.c:325 netlink_deliver_tap net/netlink/af_netlink.c:338 __netlink_sendskb net/netlink/af_netlink.c:1263 netlink_sendskb+0x1d9/0x200 net/netlink/af_netlink.c:1272 netlink_unicast+0x56d/0xf50 net/netlink/af_netlink.c:1360 nlmsg_unicast ./include/net/netlink.h:1061 rtnl_unicast+0x5a/0x80 net/core/rtnetlink.c:758 ip6addrllb_get+0xfad/0x10f0 net/ipv6/addrlabel.c:628 rtnetlink_rcv_msg+0xb33/0x1570 net/core/rtnetlink.c:6082 ... Uninit was created at: slab_post_alloc_hook+0x118/0xb00 mm/slab.h:742 slab_alloc_node mm/slab.c:3398 __kmem_cache_alloc_node+0x4f2/0x930 mm/slab.c:3437 __do_kmalloc_node mm/slab_common.c:954 __kmalloc_node_track_caller+0x117/0x3d0 mm/slab_common.c:975 kmalloc_reserve net/core/skbuff.c:437 __alloc_skb+0x27a/0xab0 net/core/skbuff.c:509 alloc_skb ./include/linux/skbuff.h:1267 nlmsg_new ./include/net/netlink.h:964 ip6addrllb_get+0x490/0x10f0 net/ipv6/addrlabel.c:608 rtnetlink_rcv_msg+0xb33/0x1570 net/core/rtnetlink.c:6082 netlink_rcv_skb+0x299/0x550 net/netlink/af_netlink.c:2540 rtnetlink_rcv+0x26/0x30 net/core/rtnetlink.c:6109 netlink_unicast_kernel net/netlink/af_netlink.c:1319 netlink_unicast+0x9ab/0xf50 net/netlink/af_netlink.c:1345 netlink_sendmsg+0xebc/0x10f0 net/netlink/af_netlink.c:1921 ... This patch ensures that the reserved field is always initialized.
CVE-2022-49869	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix possible crash in bnxt_hwrng_set_coal() During the error recovery sequence, the rtnl_lock is not held for the entire duration and some datastructures may be freed during the sequence. Check for the BNXT_STATE_OPEN flag instead of netif_running() to ensure that the device is fully operational before proceeding to reconfigure the coalescing settings. This will fix a possible crash like this: BUG: unable to handle kernel NULL pointer dereference at 0000000000000000 PGD 0 P4D 0 Oops: 0000 [#1] SMP NOPTI CPU: 10 PID: 181276 Comm: ethtool Kdump: loaded Tainted: G IOE ----- - - 4.18.0-348.el8.x86_64 #1 Hardware name: Dell Inc. PowerEdge R740/0F9N89, BIOS 2.3.10 08/15/2019 RIP: 0010:bnxt_hwrng_set_coal+0x1fb/0x2a0 [bnxt_en] Code: c2 66 83 4e 22 08 66 89 46 1c e8 10 cb 00 00 41 83 c6 01 44 39 b3 68 01 00 00 0f 8e a3 00 00 00 48 8b 93 c8 00 00 00 49 63 c6 <48> 8b 2c c2 48 8b 85 b8 02 00 00 48 85 c0 74 2e 48 8b 74 24 08 f6 RSP: 0018:ffffb11c8dcaba50 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff8d168a8b0ac0 RCX: 00000000000000c5 RDX: 0000000000000000 RSI: ffff8d162f72c000 RDI: ffff8d168a8b0b28 RBP: 0000000000000000 R08: b6e1f68a12e9a7eb R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000037 R12: ffff8d168a8b109c R13: ffff8d168a8b10aa R14: 0000000000000000 R15: ffffffff01ac4e0 FS: 00007f3852e4c740(0000) GS:ffff8d24c0080000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 000000041b3ee003 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: ethnl_set_coalesce+0x3ce/0x4c0 genl_family_rcv_msg_doit.isra.15+0x10f/0x150 genl_family_rcv_msg+0xb3/0x160 ? coalesce_fill_reply+0x480/0x480 genl_rcv_msg+0x47/0x90 ? genl_family_rcv_msg+0x160/0x160 netlink_rcv_skb+0x4c/0x120 genl_rcv+0x24/0x40 netlink_unicast+0x196/0x230 netlink_sendmsg+0x204/0x3d0 sock_sendmsg+0x4c/0x50 __sys_sendto+0xee/0x160 ? syscall_trace_enter+0x1d3/0x2c0 ? __audit_syscall_exit+0x249/0x2a0 __x64_sys_sendto+0x24/0x30 do_syscall_64+0x5b/0x1a0 entry_SYSCALL_64_after_hwframe+0x65/0xca RIP: 0033:0x7f38524163bb

CVE ID	Description
CVE-2022-49870	In the Linux kernel, the following vulnerability has been resolved: capabilities: fix undefined behavior in bit shift for CAP_TO_MASK Shifting signed 32-bit value by 31 bits is undefined, so changing significant bit to unsigned. The UBSAN warning calltrace like below: UBSAN: shift-out-of-bounds in security/commoncap.c:1252:2 left shift of 1 by 31 places cannot be represented in type 'int' Call Trace: <TASK> dump_stack_lvl+0x7d/0xa5 dump_stack+0x15/0x1b ubsan_epilogue+0xe/0x4e __ubsan_handle_shift_out_of_bounds+0x1e7/0x20c cap_task_prctl+0x561/0x6f0 security_task_prctl+0x5a/0xb0 __x64_sys_prctl+0x61/0x8f0 do_syscall_64+0x58/0x80 entry_SYSCALL_64_after_hwframe+0x63/0xcd </TASK>
CVE-2022-49871	In the Linux kernel, the following vulnerability has been resolved: net: tun: Fix memory leaks of napi_get_frags kmemleak reports after running test_progs: unreferenced object 0xffff8881b1672dc0 (size 232): comm "test_progs", pid 394388, jiffies 4354712116 (age 841.975s) hex dump (first 32 bytes): e0 84 d7 a8 81 88 ff ff 80 2c 67 b1 81 88 ff ff g..... 00 40 c5 9b 81 88 ff ff 00 00 00 00 00 00 00 00 .@..... backtrace: [<00000000c8f01748>] napi_skb_cache_get+0xd4/0x150 [<0000000041c7fc09>] __napi_build_skb+0x15/0x50 [<00000000431c7079>] __napi_alloc_skb+0x26e/0x540 [<000000003ecfa30e>] napi_get_frags+0x59/0x140 [<00000000099b2199e>] tun_get_user+0x183d/0x3bb0 [tun] [<0000000008a5adef0>] tun_chr_write_iter+0xc0/0x1b1 [tun] [<0000000049993ff4>] do_iter_readv_writev+0x19f/0x320 [<0000000008f338ea2>] do_iter_write+0x135/0x630 [<0000000008a3377a4>] vfs_writev+0x12e/0x440 [<000000000a6b5639a>] do_writev+0x104/0x280 [<000000000ccf065d8>] do_syscall_64+0x3b/0x90 [<000000000d776e329>] entry_SYSCALL_64_after_hwframe+0x63/0xcd The issue occurs in the following scenarios: tun_get_user() napi_gro_frags() napi_frags_finish() case GRO_NORMAL: gro_normal_one() list_add_tail(&skb->list, &napi->rx_list); <-- While napi->rx_count < READ_ONCE(gro_normal_batch), <-- gro_normal_list() is not called, napi->rx_list is not empty <-- not ask to complete the gro work, will cause memory leaks in <-- following tun_napi_del() ... tun_napi_del() netif_napi_del() __netif_napi_del() <-- &napi->rx_list is not empty, which caused memory leaks To fix, add napi_complete() after napi_gro_frags().
CVE-2022-49872	In the Linux kernel, the following vulnerability has been resolved: net: gso: fix panic on frag_list with mixed head alloc types Since commit 3dcdbd134f32 ("net: gso: Fix skb_segment splat when splitting gso_size mangled skb having linear-headed frag_list"), it is allowed to change gso_size of a GRO packet. However, that commit assumes that "checking the first list_skb member suffices; i.e if either of the list_skb members have non head_frag head, then the first one has too". It turns out this assumption does not hold. We've seen BUG_ON being hit in skb_segment when skbs on the frag_list had differing head_frag with the vmxnet3 driver. This happens because __netdev_alloc_skb and __napi_alloc_skb can return a skb that is page backed or kmalloced depending on the requested size. As the result, the last small skb in the GRO packet can be kmalloced. There are three different locations where this can be fixed: (1) We could check head_frag in GRO and not allow GROing skbs with different head_frag. However, that would lead to performance regression on normal forward paths with unmodified gso_size, where !head_frag in the last packet is not a problem. (2) Set a flag in bpf_skb_net_grow and bpf_skb_net_shrink indicating that NETIF_F_SG is undesirable. That would need to eat a bit in sk_buff. Furthermore, that flag can be unset when all skbs on the frag_list are page backed. To retain good performance, bpf_skb_net_grow/shrink would have to walk the frag_list. (3) Walk the frag_list in skb_segment when determining whether NETIF_F_SG should be cleared. This of course slows things down. This patch implements (3). To limit the performance impact in skb_segment, the list is walked only for skbs with SKB_GSO_DODGY set that have gso_size changed. Normal paths thus will not hit it. We could check only the last skb but since we need to walk the whole list anyway, let's stay on the safe side.
CVE-2022-49873	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix wrong reg type conversion in release_reference() Some helper functions will allocate memory. To avoid memory leaks, the verifier requires the eBPF program to release these memories by calling the corresponding helper functions. When a resource is released, all pointer registers corresponding to the resource should be invalidated. The verifier use release_references() to do this job, by apply __mark_reg_unknown() to each relevant register. It will give these registers the type of SCALAR_VALUE. A register that will contain a pointer value at runtime, but of type SCALAR_VALUE, which may allow the unprivileged user to get a kernel pointer by storing this register into a map. Using __mark_reg_not_init() while NOT allow_ptr_leaks can mitigate this problem.
CVE-2022-49874	In the Linux kernel, the following vulnerability has been resolved: HID: hyperv: fix possible memory leak in mousevsc_probe() If hid_add_device() returns error, it should call hid_destroy_device() to free hid_dev which is allocated in hid_allocate_device().

CVE ID	Description
CVE-2022-49876	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix general-protection-fault in ieee80211_subif_start_xmit() When device is running and the interface status is changed, the gpf issue is triggered. The problem triggering process is as follows: Thread A: Thread B ieee80211_runtime_change_iftype() process_one_work() ieee80211_do_stop() sdata->bss = NULL ieee80211_subif_start_xmit() ieee80211_multicast_to_unicast //!sdata->bss->multicast_to_unicast cause gpf issue When the interface status is changed, the sending queue continues to send packets. After the bss is set to NULL, the bss is accessed. As a result, this causes a general-protection-fault issue. The following is the stack information: general protection fault, probably for non-canonical address 0xdffffc000000002f: 0000 [#1] PREEMPT SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000178-0x000000000000017f] Workqueue: mld mld_ifc_work RIP: 0010:ieee80211_subif_start_xmit+0x25b/0x1310 Call Trace: <TASK> dev_hard_start_xmit+0x1be/0x990 __dev_queue_xmit+0x2c9a/0x3b60 ip6_finish_output2+0xf92/0x1520 ip6_finish_output+0x6af/0x11e0 ip6_output+0x1ed/0x540 mld_sendpack+0xa09/0xe70 mld_ifc_work+0x71c/0xbd0 process_one_work+0x9bf/0x1710 worker_thread+0x665/0x1080 kthread+0x2e4/0x3a0 ret_from_fork+0x1f/0x30 </TASK>
CVE-2022-49878	In the Linux kernel, the following vulnerability has been resolved: bpf, verifier: Fix memory leak in array reallocation for stack state If an error (NULL) is returned by krealloc(), callers of realloc_array() were setting their allocation pointers to NULL, but on error krealloc() does not touch the original allocation. This would result in a memory resource leak. Instead, free the old allocation on the error handling path. The memory leak information is as follows as also reported by Zhengchao: unreferenced object 0xffff888019801800 (size 256): comm "bpf_repo", pid 6490, jiffies 4294959200 (age 17.170s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000b211474b>] __kmalloc_node_track_caller+0x45/0xc0 [<0000000086712a0b>] krealloc+0x83/0xd0 [<00000000139aab02>] realloc_array+0x82/0xe2 [<00000000b1ca41d1>] grow_stack_state+0xfb/0x186 [<00000000cd6f36d2>] check_mem_access.cold+0x141/0x1341 [<0000000081780455>] do_check_common+0x5358/0xb350 [<0000000015f6b091>] bpf_check.cold+0xc3/0x29d [<000000002973c690>] bpf_prog_load+0x13db/0x2240 [<00000000028d1644>] __sys_bpf+0x1605/0x4ce0 [<00000000053f29bd>] __x64_sys_bpf+0x75/0xb0 [<0000000056fedaf5>] do_syscall_64+0x35/0x80 [<000000002bd58261>] entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2022-49879	In the Linux kernel, the following vulnerability has been resolved: ext4: fix BUG_ON() when directory entry has invalid rec_len The rec_len field in the directory entry has to be a multiple of 4. A corrupted filesystem image can be used to hit a BUG() in ext4_rec_len_to_disk(), called from make_indexed_dir(). -----[cut here]----- kernel BUG at fs/ext4/ext4.h:2413! ... RIP: 0010:make_indexed_dir+0x53f/0x5f0 ... Call Trace: <TASK> ? add_dirent_to_buf+0x1b2/0x200 ext4_add_entry+0x36e/0x480 ext4_add_nondir+0x2b/0xc0 ext4_create+0x163/0x200 path_openat+0x635/0xe90 do_filp_open+0xb4/0x160 ? __create_object.isra.0+0x1de/0x3b0 ? _raw_spin_unlock+0x12/0x30 do_sys_openat2+0x91/0x150 __x64_sys_open+0x6c/0xa0 do_syscall_64+0x3c/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 The fix simply adds a call to ext4_check_dir_entry() to validate the directory entry, returning -EFSCORRUPTED if the entry is invalid.

CVE ID	Description
CVE-2022-49880	In the Linux kernel, the following vulnerability has been resolved: ext4: fix warning in 'ext4_da_release_space' Syzkaller report issue as follows: EXT4-fs (loop0): Free/Dirty block details EXT4-fs (loop0): free_blocks=0 EXT4-fs (loop0): dirty_blocks=0 EXT4-fs (loop0): Block reservation details EXT4-fs (loop0): i_reserved_data_blocks=0 EXT4-fs warning (device loop0): ext4_da_release_space:1527: ext4_da_release_space: ino 18, to_free 1 with only 0 reserved data blocks -----[cut here]----- WARNING: CPU: 0 PID: 92 at fs/ext4/inode.c:1528 ext4_da_release_space+0x25e/0x370 fs/ext4/inode.c:1524 Modules linked in: CPU: 0 PID: 92 Comm: kworker/u4:4 Not tainted 6.0.0-syzkaller-09423-g493ffd6605b2 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/22/2022 Workqueue: writeback wb_workfn (flush-7:0) RIP: 0010:ext4_da_release_space+0x25e/0x370 fs/ext4/inode.c:1528 RSP: 0018:ffffc900015f6c90 EFLAGS: 00010296 RAX: 42215896cd52ea00 RBX: 0000000000000000 RCX: 42215896cd52ea00 RDX: 0000000000000000 RSI: 0000000080000001 RDI: 0000000000000000 RBP: 1ffff1100e907d96 R08: ffffffff816aa79d R09: fffff520002bece5 R10: fffff520002bece5 R11: 1ffff920002bece4 R12: ffff888021fd2000 R13: ffff88807483ecb0 R14: 0000000000000001 R15: ffff88807483e740 FS: 0000000000000000(0000) GS:ffff8880b9a00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005555569ba628 CR3: 000000000c88e000 CR4: 0000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ext4_es_remove_extent+0x1ab/0x260 fs/ext4/extents_status.c:1461 mpage_release_unused_pages+0x24d/0xef0 fs/ext4/inode.c:1589 ext4_writepages+0x12eb/0x3be0 fs/ext4/inode.c:2852 do_writepages+0x3c3/0x680 mm/page-writeback.c:2469 __writeback_single_inode+0xd1/0x670 fs/fs-writeback.c:1587 writeback_sb_inodes+0xb3b/0x18f0 fs/fs-writeback.c:1870 wb_writeback+0x41f/0x7b0 fs/fs-writeback.c:2044 wb_do_writeback fs/fs-writeback.c:2187 [inline] wb_workfn+0x3cb/0xef0 fs/fs-writeback.c:2227 process_one_work+0x877/0xdb0 kernel/workqueue.c:2289 worker_thread+0xb14/0x1330 kernel/workqueue.c:2436 kthread+0x266/0x300 kernel/kthread.c:376 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:306 </TASK> Above issue may happens as follows: ext4_da_write_begin ext4_create_inline_data ext4_clear_inode_flag(inode, EXT4_INODE_EXTENTS); ext4_set_inode_flag(inode, EXT4_INODE_INLINE_DATA); __ext4_ioct ext4_ext_migrate -> will lead to eh->eh_entries not zero, and set extent flag ext4_da_write_begin ext4_da_convert_inline_data_to_extent ext4_da_write_inline_data_begin ext4_da_map_blocks ext4_insert_delayed_block if (!ext4_es_scan_clu(inode, &ext4_es_is_delonly, lblk)) if (!ext4_es_scan_clu(inode, &ext4_es_is_mapped, lblk)) ext4_clu_mapped(inode, EXT4_B2C(sbi, lblk)); -> will return 1 allocated = true; ext4_es_insert_delayed_block(inode, lblk, allocated); ext4_writepages mpage_map_and_submit_extent(handle, &mpd, &give_up_on_write); -> return -ENOSPC mpage_release_unused_pages(&mpd, give_up_on_write); -> give_up_on_write == 1 ext4_es_remove_extent ext4_da_release_space(inode, reserved); if (unlikely(to_free > ei->i_reserved_data_blocks)) -> to_free == 1 but ei->i_reserved_data_blocks == 0 -> then trigger warning as above To solve above issue, forbid inode do migrate which has inline data.
CVE-2022-49881	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: fix memory leak in query_regdb_file() In the function query_regdb_file() the alpha2 parameter is duplicated using kmemdup() and subsequently freed in regdb_fw_cb(). However, request_firmware_nowait() can fail without calling regdb_fw_cb() and thus leak memory.
CVE-2022-49883	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: smm: number of GPRs in the SMRAM image depends on the image format On 64 bit host, if the guest doesn't have X86_FEATURE_LM, KVM will access 16 gprs to 32-bit smram image, causing out-of-bound ram access. On 32 bit host, the rsm_load_state_64/enter_smm_save_state_64 is compiled out, thus access overflow can't happen.
CVE-2022-49884	In the Linux kernel, the following vulnerability has been resolved: KVM: Initialize gfn_to_pfn_cache locks in dedicated helper Move the gfn_to_pfn_cache lock initialization to another helper and call the new helper during VM/vCPU creation. There are race conditions possible due to kvm_gfn_to_pfn_cache_init()'s ability to re-initialize the cache's locks. For example: a race between ioctl(KVM_XEN_HVM_EVTCHN_SEND) and kvm_gfn_to_pfn_cache_init() leads to a corrupted shinfo gpc lock. (thread 1) (thread 2) kvm_xen_set_evtchn_fast read_lock_irqsave(&gpc->lock, ...) kvm_gfn_to_pfn_cache_init rwlock_init(&gpc->lock) read_unlock_irqrestore(&gpc->lock, ...) Rename "cache_init" and "cache_destroy" to activate+deactivate to avoid implying that the cache really is destroyed/freed. Note, there more races in the newly named kvm_gpc_activate() that will be addressed separately. [sean: call out that this is a bug fix]

CVE ID	Description
CVE-2022-49885	In the Linux kernel, the following vulnerability has been resolved: ACPI: APEI: Fix integer overflow in ghes_estatus_pool_init() Change num_ghes from int to unsigned int, preventing an overflow and causing subsequent vmalloc() to fail. The overflow happens in ghes_estatus_pool_init() when calculating len during execution of the statement below as both multiplication operands here are signed int: len += (num_ghes * GHES_ESOURCE_PREALLOC_MAX_SIZE); The following call trace is observed because of this bug: [9.317108] swapper/0: vmalloc error: size 18446744071562596352, exceeds total pages, mode:0xcc0(GFP_KERNEL), nodemask=(null),cpuset=/,mems_allowed=0-1 [9.317131] Call Trace: [9.317134] <TASK> [9.317137] dump_stack_lvl+0x49/0x5f [9.317145] dump_stack+0x10/0x12 [9.317146] warn_alloc.cold+0x7b/0xdf [9.317150] ? __device_attach+0x16a/0x1b0 [9.317155] __vmalloc_node_range+0x702/0x740 [9.317160] ? device_add+0x17f/0x920 [9.317164] ? dev_set_name+0x53/0x70 [9.317166] ? platform_device_add+0xf9/0x240 [9.317168] __vmalloc_node+0x49/0x50 [9.317170] ? ghes_estatus_pool_init+0x43/0xa0 [9.317176] vmalloc+0x21/0x30 [9.317177] ghes_estatus_pool_init+0x43/0xa0 [9.317179] acpi_hest_init+0x129/0x19c [9.317185] acpi_init+0x434/0x4a4 [9.317188] ? acpi_sleep_proc_init+0x2a/0x2a [9.317190] do_one_initcall+0x48/0x200 [9.317195] kernel_init_freeable+0x221/0x284 [9.317200] ? rest_init+0xe0/0xe0 [9.317204] kernel_init+0x1a/0x130 [9.317205] ret_from_fork+0x22/0x30 [9.317208] </TASK> [rjw: Subject and changelog edits]
CVE-2022-49886	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Panic on bad configs that #VE on "private" memory access All normal kernel memory is "TDX private memory". This includes everything from kernel stacks to kernel text. Handling exceptions on arbitrary accesses to kernel memory is essentially impossible because they can happen in horribly nasty places like kernel entry/exit. But, TDX hardware can theoretically _deliver_ a virtualization exception (#VE) on any access to private memory. But, it's not as bad as it sounds. TDX can be configured to never deliver these exceptions on private memory with a "TD attribute" called ATTR_SEPT_VE_DISABLE. The guest has no way to *set* this attribute, but it can check it. Ensure ATTR_SEPT_VE_DISABLE is set in early boot. panic() if it is unset. There is no sane way for Linux to run with this attribute clear so a panic() is appropriate. There's small window during boot before the check where kernel has an early #VE handler. But the handler is only for port I/O and will also panic() as soon as it sees any other #VE, such as a one generated by a private memory access. [dhansen: Rewrite changelog and rebase on new tdx_parse_tinfo(). Add Kirill's tested-by because I made changes since he wrote this.]

CVE ID	Description
CVE-2022-49888	In the Linux kernel, the following vulnerability has been resolved: arm64: entry: avoid kprobe recursion The cortex_a76_erratum_1463225_debug_handler() function is called when handling debug exceptions (and synchronous exceptions from BRK instructions), and so is called when a probed function executes. If the compiler does not inline cortex_a76_erratum_1463225_debug_handler(), it can be probed. If cortex_a76_erratum_1463225_debug_handler() is probed, any debug exception or software breakpoint exception will result in recursive exceptions leading to a stack overflow. This can be triggered with the ftrace multiple_probes selftest, and as per the example splat below. This is a regression caused by commit: 6459b8469753e9fe ("arm64: entry: consolidate Cortex-A76 erratum 1463225 workaround") ... which removed the NOKPROBE_SYMBOL() annotation associated with the function. My intent was that cortex_a76_erratum_1463225_debug_handler() would be inlined into its caller, el1_dbg(), which is marked noinstr and cannot be probed. Mark cortex_a76_erratum_1463225_debug_handler() as __always_inline to ensure this. Example splat prior to this patch (with recursive entries elided): # echo p cortex_a76_erratum_1463225_debug_handler > /sys/kernel/debug/tracing/kprobe_events # echo p do_el0_svc >> /sys/kernel/debug/tracing/kprobe_events # echo 1 > /sys/kernel/debug/tracing/events/kprobes/enable Insufficient stack space to handle exception! ESR: 0x0000000096000047 -- DABT (current EL) FAR: 0xffff800009cefff0 Task stack: [0xffff800009cf0000..0xffff800009cf4000] IRQ stack: [0xffff800008000000..0xffff800008004000] Overflow stack: [0xffff00007fbc00f0..0xffff00007fbc10f0] CPU: 0 PID: 145 Comm: sh Not tainted 6.0.0 #2 Hardware name: linux,dummy-virt (DT) pstate: 604003c5 (nZCv DAIF +PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : arm64_enter_el1_dbg+0x4/0x20 lr : el1_dbg+0x24/0x5c sp : ffff800009cf0000 x29: ffff800009cf0000 x28: ffff000002c74740 x27: 0000000000000000 x26: 0000000000000000 x25: 0000000000000000 x24: 0000000000000000 x23: 00000000604003c5 x22: ffff80000801745c x21: 0000aaaa95ac068 x20: 00000000f2000004 x19: ffff800009cf0040 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000010 x10: ffff800008c87190 x9 : ffff800008ca00d0 x8 : 000000000000003c x7 : 0000000000000000 x6 : 0000000000000000 x5 : 0000000000000000 x4 : 0000000000000000 x3 : 00000000000043a4 x2 : 00000000f2000004 x1 : 00000000f2000004 x0 : ffff800009cf0040 Kernel panic - not syncing: kernel stack overflow CPU: 0 PID: 145 Comm: sh Not tainted 6.0.0 #2 Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0xe4/0x104 show_stack+0x18/0x4c dump_stack_lvl+0x64/0x7c dump_stack+0x18/0x38 panic+0x14c/0x338 test_taint+0x0/0x2c panic_bad_stack+0x104/0x118 handle_bad_stack+0x34/0x48 __bad_stack+0x78/0x7c arm64_enter_el1_dbg+0x4/0x20 el1h_64_sync_handler+0x40/0x98 el1h_64_sync+0x64/0x68 cortex_a76_erratum_1463225_debug_handler+0x0/0x34 ... el1h_64_sync_handler+0x40/0x98 el1h_64_sync+0x64/0x68 cortex_a76_erratum_1463225_debug_handler+0x0/0x34 ... el1h_64_sync_handler+0x40/0x98 el1h_64_sync+0x64/0x68 do_el0_svc+0x0/0x28 el0t_64_sync_handler+0x84/0xf0 el0t_64_sync+0x18c/0x190 Kernel Offset: disabled CPU features: 0x0080,00005021,19001080 Memory Limit: none ---[end Kernel panic - not syncing: kernel stack overflow]--- With this patch, cortex_a76_erratum_1463225_debug_handler() is inlined into el1_dbg(), and el1_dbg() cannot be probed: # echo p cortex_a76_erratum_1463225_debug_handler > /sys/kernel/debug/tracing/kprobe_events sh: write error: No such file or directory # grep -w cortex_a76_errat ---truncated---
CVE-2022-49889	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Check for NULL cpu_buffer in ring_buffer_wake_waiters() On some machines the number of listed CPUs may be bigger than the actual CPUs that exist. The tracing subsystem allocates a per_cpu directory with access to the per CPU ring buffer via a cpuX file. But to save space, the ring buffer will only allocate buffers for online CPUs, even though the CPU array will be as big as the nr_cpu_ids. With the addition of waking waiters on the ring buffer when closing the file, the ring_buffer_wake_waiters() now needs to make sure that the buffer is allocated (with the irq_work allocated with it) before trying to wake waiters, as it will cause a NULL pointer dereference. While debugging this, I added a NULL check for the buffer itself (which is OK to do), and also NULL pointer checks against buffer->buffers (which is not fine, and will WARN) as well as making sure the CPU number passed in is within the nr_cpu_ids (which is also not fine if it isn't). Bugzilla: https://bugzilla.opensuse.org/show_bug.cgi?id=1204705

CVE ID	Description
CVE-2022-49892	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix use-after-free for dynamic ftrace_ops KASAN reported a use-after-free with ftrace ops [1]. It was found from vmcore that perf had registered two ops with the same content successively, both dynamic. After unregistering the second ops, a use-after-free occurred. In ftrace_shutdown(), when the second ops is unregistered, the FTRACE_UPDATE_CALLS command is not set because there is another enabled ops with the same content. Also, both ops are dynamic and the ftrace callback function is ftrace_ops_list_func, so the FTRACE_UPDATE_TRACE_FUNC command will not be set. Eventually the value of 'command' will be 0 and ftrace_shutdown() will skip the rcu synchronization. However, ftrace may be activated. When the ops is released, another CPU may be accessing the ops. Add the missing synchronization to fix this problem. [1] BUG: KASAN: use-after-free in __ftrace_ops_list_func kernel/trace/ftrace.c:7020 [inline] BUG: KASAN: use-after-free in ftrace_ops_list_func+0x2b0/0x31c kernel/trace/ftrace.c:7049 Read of size 8 at addr ffff56551965bbc8 by task syz-executor.2/14468 CPU: 1 PID: 14468 Comm: syz-executor.2 Not tainted 5.10.0 #7 Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0x0/0x40c arch/arm64/kernel/stacktrace.c:132 show_stack+0x30/0x40 arch/arm64/kernel/stacktrace.c:196 __dump_stack lib/dump_stack.c:77 [inline] dump_stack+0x1b4/0x248 lib/dump_stack.c:118 print_address_description.constprop.0+0x28/0x48c mm/kasan/report.c:387 __kasan_report mm/kasan/report.c:547 [inline] kasan_report+0x118/0x210 mm/kasan/report.c:564 check_memory_region_inline mm/kasan/generic.c:187 [inline] __asan_load8+0x98/0xc0 mm/kasan/generic.c:253 __ftrace_ops_list_func kernel/trace/ftrace.c:7020 [inline] ftrace_ops_list_func+0x2b0/0x31c kernel/trace/ftrace.c:7049 ftrace_graph_call+0x0/0x4 __might_sleep+0x8/0x100 include/linux/perf_event.h:1170 __might_fault mm/memory.c:5183 [inline] __might_fault+0x58/0x70 mm/memory.c:5171 do_strncpy_from_user lib/strncpy_from_user.c:41 [inline] strncpy_from_user+0x1f4/0x4b0 lib/strncpy_from_user.c:139 getname_flags+0xb0/0x31c fs/namei.c:149 getname+0x2c/0x40 fs/namei.c:209 [...] Allocated by task 14445: kasan_save_stack+0x24/0x50 mm/kasan/common.c:48 kasan_set_track mm/kasan/common.c:56 [inline] __kasan_kmalloc mm/kasan/common.c:479 [inline] __kasan_kmalloc.constprop.0+0x110/0x13c mm/kasan/common.c:449 kasan_kmalloc+0xc/0x14 mm/kasan/common.c:493 kmem_cache_alloc_trace+0x440/0x924 mm/slub.c:2950 kmalloc include/linux/slab.h:563 [inline] kzalloc include/linux/slab.h:675 [inline] perf_event_alloc.part.0+0xb4/0x1350 kernel/events/core.c:11230 perf_event_alloc kernel/events/core.c:11733 [inline] __do_sys_perf_event_open kernel/events/core.c:11831 [inline] __se_sys_perf_event_open+0x550/0x15f4 kernel/events/core.c:11723 __arm64_sys_perf_event_open+0x6c/0x80 kernel/events/core.c:11723 [...] Freed by task 14445: kasan_save_stack+0x24/0x50 mm/kasan/common.c:48 kasan_set_track+0x24/0x34 mm/kasan/common.c:56 kasan_set_free_info+0x20/0x40 mm/kasan/generic.c:358 __kasan_slab_free.part.0+0x11c/0x1b0 mm/kasan/common.c:437 __kasan_slab_free mm/kasan/common.c:445 [inline] kasan_slab_free+0x2c/0x40 mm/kasan/common.c:446 slab_free_hook mm/slub.c:1569 [inline] slab_free_freelist_hook mm/slub.c:1608 [inline] slab_free mm/slub.c:3179 [inline] kfree+0x12c/0xc10 mm/slub.c:4176 perf_event_alloc.part.0+0xa0c/0x1350 kernel/events/core.c:11434 perf_event_alloc kernel/events/core.c:11733 [inline] __do_sys_perf_event_open kernel/events/core.c:11831 [inline] __se_sys_perf_event_open+0x550/0x15f4 kernel/events/core.c:11723 [...]
CVE-2022-49900	In the Linux kernel, the following vulnerability has been resolved: i2c: piix4: Fix adapter not be removed in piix4_remove() In piix4_probe(), the piix4 adapter will be registered in: piix4_probe() piix4_add_adapters_sb800() / piix4_add_adapter() i2c_add_adapter() Based on the probed device type, piix4_add_adapters_sb800() or single piix4_add_adapter() will be called. For the former case, piix4_adapter_count is set as the number of adapters, while for another case it is not set and kept default *zero*. When piix4 is removed, piix4_remove() removes the adapters added in piix4_probe(), basing on the piix4_adapter_count value. Because the count is zero for the single adapter case, the adapter won't be removed and makes the sources allocated for adapter leaked, such as the i2c client and device. These sources can still be accessed by i2c or bus and cause problems. An easily reproduced case is that if a new adapter is registered, i2c will get the leaked adapter and try to call smbus_algorithm, which was already freed: Triggered by: rmmmod i2c_piix4 && modprobe max31730 BUG: unable to handle page fault for address: ffffffff053d860 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page Oops: 0000 [#1] PREEMPT SMP KASAN CPU: 0 PID: 3752 Comm: modprobe Tainted: G Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) RIP: 0010:i2c_default_probe (drivers/i2c/i2c-core-base.c:2259) i2c_core RSP: 0018:ffff888107477710 EFLAGS: 00000246 ... <TASK> i2c_detect (drivers/i2c/i2c-core-base.c:2302) i2c_core __process_new_driver (drivers/i2c/i2c-core-base.c:1336) i2c_core bus_for_each_dev (drivers/base/bus.c:301) i2c_for_each_dev (drivers/i2c/i2c-core-base.c:1823) i2c_core i2c_register_driver (drivers/i2c/i2c-core-base.c:1861) i2c_core do_one_initcall (init/main.c:1296) do_init_module (kernel/module/main.c:2455) ... </TASK> ---[end trace 0000000000000000]--- Fix this problem by correctly set piix4_adapter_count as 1 for the single adapter so it can be normally removed.

CVE ID	Description
CVE-2022-49901	In the Linux kernel, the following vulnerability has been resolved: blk-mq: Fix kmemleak in blk_mq_init_allocated_queue There is a kmemleak caused by modprobe null_blk.ko unreferenced object 0xffff8881acb1f000 (size 1024): comm "modprobe", pid 836, jiffies 4294971190 (age 27.068s) hex dump (first 32 bytes): 00 00 00 00 ad 4e ad de ff ff ff ff 00 00 00 00 N..... ff ff ff ff ff ff ff 00 53 99 9e ff ff ff ff S..... backtrace: [<000000004a10c249>] kcalloc_node_trace+0x22/0x60 [<00000000648f7950>] blk_mq_alloc_and_init_hctx+0x289/0x350 [<00000000af06de0e>] blk_mq_realloc_hw_ctxs+0x2fe/0x3d0 [<00000000e00c1872>] blk_mq_init_allocated_queue+0x48c/0x1440 [<00000000d16b4e68>] __blk_mq_alloc_disk+0xc8/0x1c0 [<00000000d10c98c3>] 0xffffffffc450d69d [<00000000b9299f48>] 0xffffffffc4538392 [<0000000061c39ed6>] do_one_initcall+0xd0/0x4f0 [<00000000b389383b>] do_init_module+0x1a4/0x680 [<0000000087cf3542>] load_module+0x6249/0x7110 [<00000000beba61b8>] __do_sys_finit_module+0x140/0x200 [<00000000fdcff51>] do_syscall_64+0x35/0x80 [<000000003c0f1f71>] entry_SYSCALL_64_after_hwframe+0x46/0xb0 That is because q->ma_ops is set to NULL before blk_release_queue is called. blk_mq_init_queue_data blk_mq_init_allocated_queue blk_mq_realloc_hw_ctxs for (i = 0; i < set->nr_hw_queues; i++) { old_hctx = xa_load(&q->hctx_table, i); if (!blk_mq_alloc_and_init_hctx(..., i, ...)) [1] if (!old_hctx) break; xa_for_each_start(&q->hctx_table, j, hctx, j) blk_mq_exit_hctx(q, set, hctx, j); [2] if (!q->nr_hw_queues) [3] goto err_hctxs; err_exit: q->mq_ops = NULL; [4] blk_put_queue blk_release_queue if (queue_is_mq(q)) [5] blk_mq_release(q); [1]: blk_mq_alloc_and_init_hctx failed at i != 0. [2]: The hctxs allocated by [1] are moved to q->unused_hctx_list and will be cleaned up in blk_mq_release. [3]: q->nr_hw_queues is 0. [4]: Set q->mq_ops to NULL. [5]: queue_is_mq returns false due to [4]. And blk_mq_release will not be called. The hctxs in q->unused_hctx_list are leaked. To fix it, call blk_release_queue in exception path.
CVE-2022-49905	In the Linux kernel, the following vulnerability has been resolved: net/smc: Fix possible leaked pernet namespace in smc_init() In smc_init(), register_pernet_subsys(&smc_net_stat_ops) is called without any error handling. If it fails, registering of &smc_net_ops won't be reverted. And if smc_nl_init() fails, &smc_net_stat_ops itself won't be reverted. This leaves wild ops in subsystem linkedlist and when another module tries to call register_pernet_operations() it triggers page fault: BUG: unable to handle page fault for address: ffffffbfff81b964c RIP: 0010:register_pernet_operations+0x1b9/0x5f0 Call Trace: <TASK> register_pernet_subsys+0x29/0x40 ebtables_init+0x58/0x1000 [ebtables] ...
CVE-2022-49906	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Free rwi on reset success Free the rwi structure in the event that the last rwi in the list processed successfully. The logic in commit 4f408e1fa6e1 ("ibmvnic: retry reset if there are no other resets") introduces an issue that results in a 32 byte memory leak whenever the last rwi in the list gets processed.
CVE-2022-49907	In the Linux kernel, the following vulnerability has been resolved: net: mdio: fix undefined behavior in bit shift for __mdiobus_register Shifting signed 32-bit value by 31 bits is undefined, so changing significant bit to unsigned. The UBSAN warning calltrace like below: UBSAN: shift-out-of-bounds in drivers/net/phy/mdio_bus.c:586:27 left shift of 1 by 31 places cannot be represented in type 'int' Call Trace: <TASK> dump_stack_lvl+0x7d/0xa5 dump_stack+0x15/0x1b ubsan_epilogue+0xe/0x4e __ubsan_handle_shift_out_of_bounds+0x1e7/0x20c __mdiobus_register+0x49d/0x4e0 fixed_mdio_bus_init+0xd8/0x12d do_one_initcall+0x76/0x430 kernel_init_freeable+0x3b3/0x422 kernel_init+0x24/0x1e0 ret_from_fork+0x1f/0x30 </TASK>
CVE-2022-49908	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix memory leak in vhci_write Syzkaller reports a memory leak as follows: ===== BUG: memory leak unreferenced object 0xffff88810d81ac00 (size 240): [...] hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<ffffffffff838733d9>] __alloc_skb+0x1f9/0x270 net/core/skbuff.c:418 [<ffffffffff833f742f>] alloc_skb include/linux/skbuff.h:1257 [inline] [<ffffffffff833f742f>] bt_skb_alloc include/net/bluetooth/bluetooth.h:469 [inline] [<ffffffffff833f742f>] vhci_get_user drivers/bluetooth/hci_vhci.c:391 [inline] [<ffffffffff833f742f>] vhci_write+0x5f/0x230 drivers/bluetooth/hci_vhci.c:511 [<ffffffffff815e398d>] call_write_iter include/linux/fs.h:2192 [inline] [<ffffffffff815e398d>] new_sync_write fs/read_write.c:491 [inline] [<ffffffffff815e398d>] vfs_write+0x42d/0x540 fs/read_write.c:578 [<ffffffffff815e3cdd>] ksys_write+0x9d/0x160 fs/read_write.c:631 [<ffffffffff845e0645>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<ffffffffff845e0645>] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 [<ffffffffff84600087>] entry_SYSCALL_64_after_hwframe+0x63/0xcd ===== HCI core will use hci_rx_work() to process frame, which is queued to the hdev->rx_q tail in hci_recv_frame() by HCI driver. Yet the problem is that, HCI core may not free the skb after handling ACL data packets. To be more specific, when start fragment does not contain the L2CAP length, HCI core just copies skb into conn->rx_skb and finishes frame process in l2cap_recv_acldata(), without freeing the skb, which triggers the above memory leak. This patch solves it by releasing the relative skb, after processing the above case in l2cap_recv_acldata().

CVE ID	Description
CVE-2022-49910	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix use-after-free caused by l2cap_reassemble_sdu Fix the race condition between the following two flows that run in parallel: 1. l2cap_reassemble_sdu -> chan->ops->recv (l2cap_sock_recv_cb) -> __sock_queue_rcv_skb. 2. bt_sock_recvmsg -> skb_recv_datagram, skb_free_datagram. An SKB can be queued by the first flow and immediately dequeued and freed by the second flow, therefore the callers of l2cap_reassemble_sdu can't use the SKB after that function returns. However, some places continue accessing struct l2cap_ctrl that resides in the SKB's CB for a short time after l2cap_reassemble_sdu returns, leading to a use-after-free condition (the stack trace is below, line numbers for kernel 5.19.8). Fix it by keeping a local copy of struct l2cap_ctrl. BUG: KASAN: use-after-free in l2cap_rx_state_recv (net/bluetooth/l2cap_core.c:6906) bluetooth Read of size 1 at addr ffff88812025f2f0 by task kworker/u17:3/43169 Workqueue: hci0 hci_rx_work [bluetooth] Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:107 (discriminator 4)) print_report.cold (mm/kasan/report.c:314 mm/kasan/report.c:429) ? l2cap_rx_state_recv (net/bluetooth/l2cap_core.c:6906) bluetooth kasan_report (mm/kasan/report.c:162 mm/kasan/report.c:493) ? l2cap_rx_state_recv (net/bluetooth/l2cap_core.c:6906) bluetooth l2cap_rx_state_recv (net/bluetooth/l2cap_core.c:6906) bluetooth l2cap_rx (net/bluetooth/l2cap_core.c:7236 net/bluetooth/l2cap_core.c:7271) bluetooth ret_from_fork (arch/x86/entry/entry_64.S:306) </TASK> Allocated by task 43169: kasan_save_stack (mm/kasan/common.c:39) __kasan_slab_alloc (mm/kasan/common.c:45 mm/kasan/common.c:436 mm/kasan/common.c:469) kmem_cache_alloc_node (mm/slab.h:750 mm/slub.c:3243 mm/slub.c:3293) __alloc_skb (net/core/skbuff.c:414) l2cap_recv_frag (/include/net/bluetooth/bluetooth.h:425 net/bluetooth/l2cap_core.c:8329) bluetooth l2cap_recv_acldata (net/bluetooth/l2cap_core.c:8442) bluetooth hci_rx_work (net/bluetooth/hci_core.c:3642 net/bluetooth/hci_core.c:3832) bluetooth process_one_work (kernel/workqueue.c:2289) worker_thread (/include/linux/list.h:292 kernel/workqueue.c:2437) kthread (kernel/kthread.c:376) ret_from_fork (arch/x86/entry/entry_64.S:306) Freed by task 27920: kasan_save_stack (mm/kasan/common.c:39) kasan_set_track (mm/kasan/common.c:45) kasan_set_free_info (mm/kasan/generic.c:372) ____kasan_slab_free (mm/kasan/common.c:368 mm/kasan/common.c:328) slab_free_freelist_hook (mm/slub.c:1780) kmem_cache_free (mm/slub.c:3536 mm/slub.c:3553) skb_free_datagram (/include/net/sock.h:1578 ./include/net/sock.h:1639 net/core/datagram.c:323) bt_sock_recvmsg (net/bluetooth/af_bluetooth.c:295) bluetooth l2cap_sock_recvmsg (net/bluetooth/l2cap_sock.c:1212) bluetooth sock_read_iter (net/socket.c:1087) new_sync_read (/include/linux/fs.h:2052 fs/read_write.c:401) vfs_read (fs/read_write.c:482) ksys_read (fs/read_write.c:620) do_syscall_64 (arch/x86/entry/common.c:50 arch/x86/entry/common.c:80) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120)
CVE-2022-49915	In the Linux kernel, the following vulnerability has been resolved: mISDN: fix possible memory leak in mISDN_register_device() After commit 1fa5ae857bb1 ("driver core: get rid of struct device's bus_id string array"), the name of device is allocated dynamically, add put_device() to give up the reference, so that the name can be freed in kobject_cleanup() when the refcount is 0. Set device class before put_device() to avoid null release() function WARN message in device_release().
CVE-2022-49917	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix WARNING in ip_vs_app_net_cleanup() During the initialization of ip_vs_app_net_init(), if file ip_vs_app fails to be created, the initialization is successful by default. Therefore, the ip_vs_app file doesn't be found during the remove in ip_vs_app_net_cleanup(). It will cause WRNING. The following is the stack information: name 'ip_vs_app' WARNING: CPU: 1 PID: 9 at fs/proc/generic.c:712 remove_proc_entry+0x389/0x460 Modules linked in: Workqueue: netns cleanup_net RIP: 0010:remove_proc_entry+0x389/0x460 Call Trace: <TASK> ops_exit_list+0x125/0x170 cleanup_net+0x4ea/0xb00 process_one_work+0x9bf/0x1710 worker_thread+0x665/0x1080 kthread+0x2e4/0x3a0 ret_from_fork+0x1f/0x30 </TASK>
CVE-2022-49918	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix WARNING in __ip_vs_cleanup_batch() During the initialization of ip_vs_conn_net_init(), if file ip_vs_conn or ip_vs_conn_sync fails to be created, the initialization is successful by default. Therefore, the ip_vs_conn or ip_vs_conn_sync file doesn't be found during the remove. The following is the stack information: name 'ip_vs_conn_sync' WARNING: CPU: 3 PID: 9 at fs/proc/generic.c:712 remove_proc_entry+0x389/0x460 Modules linked in: Workqueue: netns cleanup_net RIP: 0010:remove_proc_entry+0x389/0x460 Call Trace: <TASK> __ip_vs_cleanup_batch+0x7d/0x120 ops_exit_list+0x125/0x170 cleanup_net+0x4ea/0xb00 process_one_work+0x9bf/0x1710 worker_thread+0x665/0x1080 kthread+0x2e4/0x3a0 ret_from_fork+0x1f/0x30 </TASK>
CVE-2022-49919	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: release flow rule object from commit path No need to postpone this to the commit release path, since no packets are walking over this object, this is accessed from control plane only. This helped uncovered UAF triggered by races with the netlink notifier.
CVE-2022-49921	In the Linux kernel, the following vulnerability has been resolved: net: sched: Fix use after free in red_enqueue() We can't use "skb" again after passing it to qdisc_enqueue(). This is basically identical to commit 2f09707d0c97 ("sch_sfb: Also store skb len before calling child enqueue").

CVE ID	Description
CVE-2022-49925	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix null-ptr-deref in <code>ib_core_cleanup()</code> KASAN reported a null-ptr-deref error: KASAN: null-ptr-deref in range <code>[0x000000000000118-0x00000000000011f]</code> CPU: 1 PID: 379 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) RIP: 0010:destroy_workqueue+0x2f/0x740 RSP: 0018:ffff888016137df8 EFLAGS: 00000202 ... Call Trace: <code>ib_core_cleanup+0xa/0xa1 [ib_core] __do_sys_delete_module.constprop.0+0x34f/0x5b0 do_syscall_64+0x3a/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd</code> RIP: 0033:0x7fa1a0d221b7 ... It is because the fail of <code>roce_gid_mgmt_init()</code> is ignored: <code>ib_core_init()</code> <code>roce_gid_mgmt_init()</code> <code>gid_cache_wq = alloc_ordered_workqueue # fail ... ib_core_cleanup()</code> <code>roce_gid_mgmt_cleanup()</code> <code>destroy_workqueue(gid_cache_wq) # destroy an unallocated wq</code> Fix this by catching the fail of <code>roce_gid_mgmt_init()</code> in <code>ib_core_init()</code> .
CVE-2022-49927	In the Linux kernel, the following vulnerability has been resolved: nfs4: Fix kmemleak when allocate slot failed If one of the slot allocate failed, should cleanup all the other allocated slots, otherwise, the allocated slots will leak: unreferenced object 0xffff8881115aa100 (size 64): comm ""mount.nfs"", pid 679, jiffies 4294744957 (age 115.037s) hex dump (first 32 bytes): 00 cc 19 73 81 88 ff 00 a0 5a 11 81 88 ff ff ...Z..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<code>000000007a4c434a</code>] <code>nfs4_find_or_create_slot+0x8e/0x130</code> [<code>000000005472a39c</code>] <code>nfs4_realloc_slot_table+0x23f/0x270</code> [<code>00000000cd8ca0eb</code>] <code>nfs40_init_client+0x4a/0x90</code> [<code>00000000128486db</code>] <code>nfs4_init_client+0xce/0x270</code> [<code>000000008d2cacad</code>] <code>nfs4_set_client+0x1a2/0x2b0</code> [<code>000000000e593b52</code>] <code>nfs4_create_server+0x300/0x5f0</code> [<code>000000000e4425dd2</code>] <code>nfs4_try_get_tree+0x65/0x110</code> [<code>00000000d3a6176f</code>] <code>vfs_get_tree+0x41/0xf0</code> [<code>0000000016b5ad4c</code>] <code>path_mount+0x9b3/0xdd0</code> [<code>00000000494cae71</code>] <code>__x64_sys_mount+0x190/0x1d0</code> [<code>0000000005d56bdec</code>] <code>do_syscall_64+0x35/0x80</code> [<code>000000000687c9ae4</code>] <code>entry_SYSCALL_64_after_hwframe+0x46/0xb0</code>
CVE-2022-49928	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix null-ptr-deref when xps sysfs alloc failed There is a null-ptr-deref when xps sysfs alloc failed: BUG: KASAN: null-ptr-deref in <code>sysfs_do_create_link_sd+0x40/0xd0</code> Read of size 8 at addr 0000000000000030 by task gssproxy/457 CPU: 5 PID: 457 Comm: gssproxy Not tainted 6.0.0-09040-g02357b27ee03 #9 Call Trace: <TASK> <code>dump_stack_lvl+0x34/0x44 kasan_report+0xa3/0x120 sysfs_do_create_link_sd+0x40/0xd0 rpc_sysfs_client_setup+0x161/0x1b0 rpc_new_client+0x3fc/0x6e0 rpc_create_xprt+0x71/0x220 rpc_create+0x1d4/0x350 gssp_rpc_create+0xc3/0x160 set_gssp_clnt+0xbc/0x140 write_gssp+0x116/0x1a0 proc_reg_write+0xd6/0x130 vfs_write+0x177/0x690 ksys_write+0xb9/0x150 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0</code> When the <code>xprt_switch</code> sysfs alloc failed, should not add <code>xprt</code> and <code>switch</code> sysfs to it, otherwise, maybe null-ptr-deref; also initialize the 'xps_sysfs' to NULL to avoid oops when destroy it.
CVE-2022-49929	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix mr leak in <code>RESPST_ERR_RNR rxe_recheck_mr()</code> will increase mr's <code>ref_cnt</code> , so we should call <code>rxe_put(mr)</code> to drop mr's <code>ref_cnt</code> in <code>RESPST_ERR_RNR</code> to avoid below warning: WARNING: CPU: 0 PID: 4156 at <code>drivers/infiniband/sw/rxe/rxe_pool.c:259 __rxe_cleanup+0x1df/0x240 [rdma_rxe]</code> ... Call Trace: <code>rxe_dereg_mr+0x4c/0x60 [rdma_rxe] ib_dereg_mr_user+0xa8/0x200 [ib_core] ib_mr_pool_destroy+0x77/0xb0 [ib_core] nvme_rdma_destroy_queue_ib+0x89/0x240 [nvme_rdma] nvme_rdma_free_queue+0x40/0x50 [nvme_rdma] nvme_rdma_teardown_io_queues.part.0+0xc3/0x120 [nvme_rdma] nvme_rdma_error_recovery_work+0x4d/0xf0 [nvme_rdma] process_one_work+0x582/0xa40 ? <code>pwq_dec_nr_in_flight+0x100/0x100 ? rwlock_bug.part.0+0x60/0x60 worker_thread+0x2a9/0x700 ? process_one_work+0xa40/0xa40 kthread+0x168/0x1a0 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30</code></code>
CVE-2022-49931	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Correctly move list in <code>sc_disable()</code> Commit 13bac861952a ("IB/hfi1: Fix abba locking issue with <code>sc_disable()</code> ") incorrectly tries to move a list from one list head to another. The result is a kernel crash. The crash is triggered when a link goes down and there are waiters for a send to complete. The following signature is seen: BUG: kernel NULL pointer dereference, address: 0000000000000030 [...] Call Trace: <code>sc_disable+0x1ba/0x240 [hfi1] pio_freeze+0x3d/0x60 [hfi1] handle_freeze+0x27/0x1b0 [hfi1] process_one_work+0x1b0/0x380 ? process_one_work+0x380/0x380 worker_thread+0x30/0x360 ? process_one_work+0x380/0x380 kthread+0xd7/0x100 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x1f/0x30</code> The fix is to use the correct call to move the list.

CVE ID	Description
CVE-2022-49932	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Do _all_ initialization before exposing /dev/kvm to userspace Call kvm_init() only after _all_ setup is complete, as kvm_init() exposes /dev/kvm to userspace and thus allows userspace to create VMs (and call other ioctls). E.g. KVM will encounter a NULL pointer when attempting to add a vCPU to the per-CPU loaded_vmcss_on_cpu list if userspace is able to create a VM before vmx_init() configures said list. BUG: kernel NULL pointer dereference, address: 0000000000000008 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 0 P4D 0 Oops: 0002 [#1] SMP CPU: 6 PID: 1143 Comm: stable Not tainted 6.0.0-rc7+ #988 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 0.0.0 02/06/2015 RIP: 0010:vmx_vcpu_load_vmcs+0x68/0x230 [kvm_intel] <TASK> vmx_vcpu_load+0x16/0x60 [kvm_intel] kvm_arch_vcpu_load+0x32/0x1f0 [kvm] vcpu_load+0x2f/0x40 [kvm] kvm_arch_vcpu_create+0x231/0x310 [kvm] kvm_ym_ioctl+0x79f/0xe10 [kvm] ? handle_mm_fault+0xb1/0x220 __x64_sys_ioctl+0x80/0xb0 do_syscall_64+0x2b/0x50 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7f5a6b05743b </TASK> Modules linked in: vhost_net vhost vhost_iotlb tap kvm_intel(+) kvm irqbypass
CVE-2022-49935	In the Linux kernel, the following vulnerability has been resolved: dma-buf/dma-resv: check if the new fence is really later Previously when we added a fence to a dma_resv object we always assumed the the newer than all the existing fences. With Jason's work to add a UAPI to explicit export/import that's not necessary the case any more. So without this check we would allow userspace to force the kernel into an use after free error. Since the change is very small and defensive it's probably a good idea to backport this to stable kernels as well just in case others are using the dma_resv object in the same way.
CVE-2022-49936	In the Linux kernel, the following vulnerability has been resolved: USB: core: Prevent nested device-reset calls Automatic kernel fuzzing revealed a recursive locking violation in usb-storage: ===== WARNING: possible recursive locking detected 5.18.0 #3 Not tainted ----- kworker/1:3/1205 is trying to acquire lock: ffff888018638db8 (&us_interface_key[i]){+.+.}-{3:3}, at: usb_stor_pre_reset +0x35/0x40 drivers/usb/storage/usb.c:230 but task is already holding lock: ffff888018638db8 (&us_interface_key[i]){+.+.}-{3:3}, at: usb_stor_pre_reset+0x35/0x40 drivers/usb/storage/usb.c:230 ... stack backtrace: CPU: 1 PID: 1205 Comm: kworker/1:3 Not tainted 5.18.0 #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 Workqueue: usb_hub_wq hub_event Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl +0xcd/0x134 lib/dump_stack.c:106 print_deadlock_bug kernel/locking/lockdep.c:2988 [inline] check_deadlock kernel/locking/lockdep.c:3031 [inline] validate_chain kernel/locking/lockdep.c:3816 [inline] __lock_acquire.cold+0x152/0x3ca kernel/locking/lockdep.c:5053 lock_acquire kernel/locking/lockdep.c:5665 [inline] lock_acquire+0x1ab/0x520 kernel/locking/lockdep.c:5630 __mutex_lock_common kernel/locking/mutex.c:603 [inline] __mutex_lock+0x14f/0x1610 kernel/locking/mutex.c:747 usb_stor_pre_reset+0x35/0x40 drivers/usb/storage/usb.c:230 usb_reset_device +0x37d/0x9a0 drivers/usb/core/hub.c:6109 rtl871xu_dev_remove+0x21a/0x270 drivers/staging/rtl8712/usb_intf.c:622 usb_unbind_interface+0x1bd/0x890 drivers/usb/core/driver.c:458 device_remove drivers/base/dd.c:545 [inline] device_remove+0x11f/0x170 drivers/base/dd.c:537 __device_release_driver drivers/base/dd.c:1222 [inline] device_release_driver_internal+0x1a7/0x2f0 drivers/base/dd.c:1248 usb_driver_release_interface+0x102/0x180 drivers/usb/core/driver.c:627 usb_forced_unbind_intf +0x4d/0xa0 drivers/usb/core/driver.c:1118 usb_reset_device+0x39b/0x9a0 drivers/usb/core/hub.c:6114 This turned out not to be an error in usb-storage but rather a nested device reset attempt. That is, as the rtl8712 driver was being unbound from a composite device in preparation for an unrelated USB reset (that driver does not have pre_reset or post_reset callbacks), its ->remove routine called usb_reset_device() -- thus nesting one reset call within another. Performing a reset as part of disconnect processing is a questionable practice at best. However, the bug report points out that the USB core does not have any protection against nested resets. Adding a reset_in_progress flag and testing it will prevent such errors in the future.

CVE ID	Description
CVE-2022-49937	In the Linux kernel, the following vulnerability has been resolved: media: mceusb: Use new usb_control_msg_*() routines Automatic kernel fuzzing led to a WARN about invalid pipe direction in the mceusb driver: -----[cut here]----- usb 6-1: BOGUS control dir, pipe 80000380 doesn't match bRequestType 40 WARNING: CPU: 0 PID: 2465 at drivers/usb/core/urb.c:410 usb_submit_urb+0x1326/0x1820 drivers/usb/core/urb.c:410 Modules linked in: CPU: 0 PID: 2465 Comm: kworker/0:2 Not tainted 5.19.0-rc4-00208-g69cb6c6556ad #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 Workqueue: usb_hub_wq hub_event RIP: 0010:usb_submit_urb+0x1326/0x1820 drivers/usb/core/urb.c:410 Code: 7c 24 40 e8 ac 23 91 fd 48 8b 7c 24 40 e8 b2 70 1b ff 45 89 e8 44 89 f1 4c 89 e2 48 89 c6 48 c7 c7 a0 30 a9 86 e8 48 07 11 02 <0f> 0b e9 1c f0 ff ff e8 7e 23 91 fd 0f b6 1d 63 22 83 05 31 ff 41 RSP: 0018:ffff900032becf0 EFLAGS: 00010282 RAX: 0000000000000000 RBX: ffff8881100f3058 RCX: 0000000000000000 RDX: ffff90004961000 RSI: ffff888114c6d580 RDI: ffff52000657d90 RBP: ffff888105ad90f0 R08: ffffffff812c3638 R09: 0000000000000000 R10: 0000000000000005 R11: ffffd1023504ef1 R12: ffff888105ad9000 R13: 0000000000000040 R14: 0000000080000380 R15: ffff88810ba96500 FS: 0000000000000000(0000) GS:ffff88811a800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffe810bda58 CR3: 000000010b720000 CR4: 000000000350ef0 Call Trace: <TASK> usb_start_wait_urb+0x101/0x4c0 drivers/usb/core/message.c:58 usb_internal_control_msg drivers/usb/core/message.c:102 [inline] usb_control_msg+0x31c/0x4a0 drivers/usb/core/message.c:153 mceusb_gen1_init drivers/media/rc/mceusb.c:1431 [inline] mceusb_dev_probe+0x258e/0x33f0 drivers/media/rc/mceusb.c:1807 The reason for the warning is clear enough; the driver sends an unusual read request on endpoint 0 but does not set the USB_DIR_IN bit in the bRequestType field. More importantly, the whole situation can be avoided and the driver simplified by converting it over to the relatively new usb_control_msg_recv() and usb_control_msg_send() routines. That's what this fix does.
CVE-2022-49938	In the Linux kernel, the following vulnerability has been resolved: cifs: fix small mempool leak in SMB2_negotiate() In some cases of failure (dialect mismatches) in SMB2_negotiate(), after the request is sent, the checks would return -EIO when they should be rather setting rc = -EIO and jumping to neg_exit to free the response buffer from mempool.
CVE-2022-49940	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: add sanity check for gsm->receive in gsm_receive_buf() A null pointer dereference can happen when attempting to access the "gsm->receive()" function in gsmlld_receive_buf(). Currently, the code assumes that gsm->receive is only called after MUX activation. Since the gsmlld_receive_buf() function can be accessed without the need to initialize the MUX, the gsm->receive() function will not be set and a NULL pointer dereference will occur. Fix this by avoiding the call to "gsm->receive()" in case the function is not initialized by adding a sanity check. Call Trace: <TASK> gsmlld_receive_buf+0x1c2/0x2f0 drivers/tty/n_gsm.c:2861 tiocsti drivers/tty/tty_io.c:2293 [inline] tty_ioctl+0xa75/0x15d0 drivers/tty/tty_io.c:2692 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:870 [inline] __se_sys_ioctl fs/ioctl.c:856 [inline] __x64_sys_ioctl+0x193/0x200 fs/ioctl.c:856 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2022-49948	In the Linux kernel, the following vulnerability has been resolved: vt: Clear selection before changing the font When changing the console font with ioctl(KDFONTOP) the new font size can be bigger than the previous font. A previous selection may thus now be outside of the new screen size and thus trigger out-of-bounds accesses to graphics memory if the selection is removed in vc_do_resize(). Prevent such out-of-memory accesses by dropping the selection before the various con_font_set() console handlers are called.
CVE-2022-49949	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: Fix memory leak in firmware upload In the case of firmware-upload, an instance of struct fw_upload is allocated in firmware_upload_register(). This data needs to be freed in fw_dev_release(). Create a new fw_upload_free() function in sysfs_upload.c to handle the firmware-upload specific memory frees and incorporate the missing kfree call for the fw_upload structure.

CVE ID	Description
CVE-2022-49961	In the Linux kernel, the following vulnerability has been resolved: bpf: Do mark_chain_precision for ARG_CONST_ALLOC_SIZE_OR_ZERO Precision markers need to be propagated whenever we have an ARG_CONST_* style argument, as the verifier cannot consider imprecise scalars to be equivalent for the purposes of states_equal check when such arguments refine the return value (in this case, set mem_size for PTR_TO_MEM). The resultant mem_size for the R0 is derived from the constant value, and if the verifier incorrectly prunes states considering them equivalent where such arguments exist (by seeing that both registers have reg->precise as false in regsafe), we can end up with invalid programs passing the verifier which can do access beyond what should have been the correct mem_size in that explored state. To show a concrete example of the problem: 0000000000000000 <prog>: 0: r2 = *(u32*)(r1 + 80) 1: r1 = *(u32*)(r1 + 76) 2: r3 = r1 3: r3 += 4 4: if r3 > r2 goto +18 <LBB5_5> 5: w2 = 0 6: *(u32*)(r1 + 0) = r2 7: r1 = *(u32*)(r1 + 0) 8: r2 = 1 9: if w1 == 0 goto +1 <LBB5_3> 10: r2 = -1 00000000000000058 <LBB5_3>: 11: r1 = 0 ll 13: r3 = 0 14: call bpf_ringbuf_reserve 15: if r0 == 0 goto +7 <LBB5_5> 16: r1 = r0 17: r1 += 16777215 18: w2 = 0 19: *(u8*)(r1 + 0) = r2 20: r1 = r0 21: r2 = 0 22: call bpf_ringbuf_submit 00000000000000b8 <LBB5_5>: 23: w0 = 0 24: exit For the first case, the single line execution's exploration will prune the search at insn 14 for the branch insn 9's second leg as it will be verified first using r2 = -1 (UINT_MAX), while as w1 at insn 9 will always be 0 so at runtime we don't get error for being greater than UINT_MAX/4 from bpf_ringbuf_reserve. The verifier during regsafe just sees reg->precise as false for both r2 registers in both states, hence considers them equal for purposes of states_equal. If we propagated precise markers using the backtracking support, we would use the precise marking to then ensure that old r2 (UINT_MAX) was within the new r2 (1) and this would never be true, so the verification would rightfully fail. The end result is that the out of bounds access at instruction 19 would be permitted without this fix. Note that reg->precise is always set to true when user does not have CAP_BPF (or when subprog count is greater than 1 (i.e. use of any static or global functions)), hence this is only a problem when precision marks need to be explicitly propagated (i.e. privileged users with CAP_BPF). A simplified test case has been included in the next patch to prevent future regressions.
CVE-2022-49963	In the Linux kernel, the following vulnerability has been resolved: drm/i915/ttm: fix CCS handling Crucible + recent Mesa seems to sometimes hit: GEM_BUG_ON(num_ccs_blks > NUM_CCS_BLKS_PER_XFER) And it looks like we can also trigger this with gem_lmem_swapping, if we modify the test to use slightly larger object sizes. Looking closer it looks like we have the following issues in migrate_copy(): - We are using plain integer in various places, which we can easily overflow with a large object. - We pass the entire object size (when the src is lmem) into emit_pte() and then try to copy it, which doesn't work, since we only have a few fixed sized windows in which to map the pages and perform the copy. With an object > 8M we therefore aren't properly copying the pages. And then with an object > 64M we trigger the GEM_BUG_ON(num_ccs_blks > NUM_CCS_BLKS_PER_XFER). So it looks like our copy handling for any object > 8M (which is our CHUNK_SZ) is currently broken on DG2. Testcase: igt@gem_lmem_swapping (cherry picked from commit 8676145eb2f53a9940ff70910caf0125bd8a4bc2)
CVE-2022-49964	In the Linux kernel, the following vulnerability has been resolved: arm64: cacheinfo: Fix incorrect assignment of signed error value to unsigned fw_level Though acpi_find_last_cache_level() always returned signed value and the document states it will return any errors caused by lack of a PPTT table, it never returned negative values before. Commit 0c80f9e165f8 ("ACPI: PPTT: Leave the table mapped for the runtime usage") however changed it by returning -ENOENT if no PPTT was found. The value returned from acpi_find_last_cache_level() is then assigned to unsigned fw_level. It will result in the number of cache leaves calculated incorrectly as a huge value which will then cause the following warning from __alloc_pages as the order would be greater than MAX_ORDER because of incorrect and huge cache leaves value. WARNING: CPU: 0 PID: 1 at mm/page_alloc.c:5407 __alloc_pages+0x74/0x314 Modules linked in: CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.19.0-10393-g7c2a8d3ac4c0 #73 pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : __alloc_pages+0x74/0x314 lr : alloc_pages+0xe8/0x318 Call trace: __alloc_pages+0x74/0x314 alloc_pages+0xe8/0x318 kmalloc_order_trace+0x68/0x1dc __kmalloc+0x240/0x338 detect_cache_attributes+0xe0/0x56c update_siblings_masks+0x38/0x284 store_cpu_topology+0x78/0x84 smp_prepare_cpus+0x48/0x134 kernel_init_freeable+0xc4/0x14c kernel_init+0x2c/0x1b4 ret_from_fork+0x10/0x20 Fix the same by changing fw_level to be signed integer and return the error from init_cache_level() early in case of error.
CVE-2022-49967	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix a data-race around bpf_jit_limit. While reading bpf_jit_limit, it can be changed concurrently via sysctl, WRITE_ONCE() in __do_proc_doulongvec_minmax(). The size of bpf_jit_limit is long, so we need to add a paired READ_ONCE() to avoid load-tearing.

CVE ID	Description
CVE-2022-49970	In the Linux kernel, the following vulnerability has been resolved: bpf, cgroup: Fix kernel BUG in purge_effective_progs Syzkaller reported a triggered kernel BUG as follows: -----[cut here]----- kernel BUG at kernel/bpf/cgroup.c:925! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 1 PID: 194 Comm: detach Not tainted 5.19.0-14184-g69dac8e431af #8 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 RIP: 0010: __cgroup_bpf_detach+0x1f2/0x2a0 Code: 00 e8 92 60 30 00 84 c0 75 d8 4c 89 e0 31 f6 85 f6 74 19 42 f6 84 28 48 05 00 00 02 75 0e 48 8b 80 c0 00 00 00 48 85 c0 75 e5 <0f> 0b 48 8b 0c5 RSP: 0018:ffff9000055bdb0 EFLAGS: 00000246 RAX: 0000000000000000 RBX: ffff888100ec0800 RCX: ffff900000f1000 RDX: 0000000000000000 RSI: 0000000000000001 RDI: ffff888100ec4578 RBP: 0000000000000000 R08: ffff888100ec0800 R09: 0000000000000040 R10: 0000000000000000 R11: 0000000000000000 R12: ffff888100ec4000 R13: 000000000000000d R14: ffff90000199000 R15: ffff888100effb00 FS: 00007f68213d2b80(0000) GS:ffff88813bc80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055f74a0e5850 CR3: 0000000102836000 CR4: 000000000000006e0 Call Trace: <TASK> cgroup_bpf_prog_detach+0xcc/0x100 __sys_bpf+0x2273/0x2a00 __x64_sys_bpf+0x17/0x20 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7f68214dbcb9 Code: 08 44 89 e0 5b 41 5c c3 66 0f 1f 84 00 00 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff8 RSP: 002b:00007ffeb487db68 EFLAGS: 00000246 ORIG_RAX: 0000000000000141 RAX: ffffffffda RBX: 000000000000000b RCX: 00007f68214dbcb9 RDX: 0000000000000090 RSI: 00007ffeb487db70 RDI: 0000000000000009 RBP: 0000000000000003 R08: 0000000000000012 R09: 0000000b00000003 R10: 00007ffeb487db70 R11: 0000000000000246 R12: 00007ffeb487dc20 R13: 0000000000000004 R14: 0000000000000001 R15: 000055f74a1011b0 </TASK> Modules linked in: ---[end trace 0000000000000000]--- Repetition steps: For the following cgroup tree, root cg1 cg2 1. attach prog2 to cg2, and then attach prog1 to cg1, both bpf progs attach type is NONE or OVERRIDE. 2. write 1 to /proc/thread-self/fail-nth for failslab. 3. detach prog1 for cg1, and then kernel BUG occur. Failslab injection will cause kmallocc fail and fall back to purge_effective_progs. The problem is that cg2 have attached another prog, so when go through cg2 layer, iteration will add pos to 1, and subsequent operations will be skipped by the following condition, and cg will meet NULL in the end. `if (pos && !(cg->bpf.flags[atype] & BPF_F_ALLOW_MULTI))` The NULL cg means no link or prog match, this is as expected, and it's not a bug. So here just skip the no match situation.
CVE-2022-49972	In the Linux kernel, the following vulnerability has been resolved: xsk: Fix corrupted packets for XDP_SHARED_UMEM Fix an issue in XDP_SHARED_UMEM mode together with aligned mode where packets are corrupted for the second and any further sockets bound to the same umem. In other words, this does not affect the first socket bound to the umem. The culprit for this bug is that the initialization of the DMA addresses for the pre-populated xsk buffer pool entries was not performed for any socket but the first one bound to the umem. Only the linear array of DMA addresses was populated. Fix this by populating the DMA addresses in the xsk buffer pool for every socket bound to the same umem.
CVE-2022-49973	In the Linux kernel, the following vulnerability has been resolved: skmsg: Fix wrong last sg check in sk_msg_recvmg() Fix one kernel NULL pointer dereference as below: [224.462334] Call Trace: [224.462394] __tcp_bpf_recvmg+0xd3/0x380 [224.462441] ? sock_has_perm+0x78/0xa0 [224.462463] tcp_bpf_recvmg+0x12e/0x220 [224.462494] inet_recvmg+0x5b/0xd0 [224.462534] __sys_recvfrom+0xc8/0x130 [224.462574] ? syscall_trace_enter+0x1df/0x2e0 [224.462606] ? __do_page_fault+0x2de/0x500 [224.462635] __x64_sys_recvfrom+0x24/0x30 [224.462660] do_syscall_64+0x5d/0x1d0 [224.462709] entry_SYSCALL_64_after_hwframe+0x65/0xca In commit 9974d37ea75f ("skmsg: Fix invalid last sg check in sk_msg_recvmg()"), we change last sg check to sg_is_last(), but in sockmap redirection case (without stream_parser/stream_verdict/skb_verdict), we did not mark the end of the scatterlist. Check the sk_msg_alloc, sk_msg_page_add, and bpf_msg_push_data functions, they all do not mark the end of sg. They are expected to use sg.end for end judgment. So the judgment of '(i != msg_rx->sg.end)' is added back here.
CVE-2022-49975	In the Linux kernel, the following vulnerability has been resolved: bpf: Don't redirect packets with invalid pkt_len Syzbot found an issue [1]: fq_codel_drop() try to drop a flow whitout any skbs, that is, the flow->head is null. The root cause, as the [2] says, is because that bpf_prog_test_run_skb() run a bpf prog which redirects empty skbs. So we should determine whether the length of the packet modified by bpf prog or others like bpf_prog_test is valid before forwarding it directly.

CVE ID	Description
CVE-2022-49979	In the Linux kernel, the following vulnerability has been resolved: net: fix refcount bug in sk_psock_get (2) Syzkaller reports refcount bug as follows: -----[cut here]----- refcount_t: saturated; leaking memory. WARNING: CPU: 1 PID: 3605 at lib/refcount.c:19 refcount_warn_saturate +0xf4/0x1e0 lib/refcount.c:19 Modules linked in: CPU: 1 PID: 3605 Comm: syz-executor208 Not tainted 5.18.0-syzkaller-03023-g7e062cda7d90 #0 <TASK> __refcount_add_not_zero include/linux/refcount.h:163 [inline] __refcount_inc_not_zero include/linux/refcount.h:227 [inline] refcount_inc_not_zero include/linux/refcount.h:245 [inline] sk_psock_get+0x3bc/0x410 include/linux/skmsg.h:439 tls_data_ready+0x6d/0x1b0 net/tls/tls_sw.c:2091 tcp_data_ready+0x106/0x520 net/ipv4/tcp_input.c:4983 tcp_data_queue+0x25f2/0x4c90 net/ipv4/tcp_input.c:5057 tcp_rcv_state_process +0x1774/0x4e80 net/ipv4/tcp_input.c:6659 tcp_v4_do_rcv+0x339/0x980 net/ipv4/tcp_ipv4.c:1682 sk_backlog_rcv include/net/sock.h:1061 [inline] __release_sock+0x134/0x3b0 net/core/sock.c:2849 release_sock+0x54/0x1b0 net/core/sock.c:3404 inet_shutdown+0x1e0/0x430 net/ipv4/af_inet.c:909 __sys_shutdown_sock net/socket.c:2331 [inline] __sys_shutdown_sock net/socket.c:2325 [inline] __sys_shutdown+0xf1/0x1b0 net/socket.c:2343 __do_sys_shutdown net/socket.c:2351 [inline] __se_sys_shutdown net/socket.c:2349 [inline] __x64_sys_shutdown+0x50/0x70 net/socket.c:2349 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 <TASK> During SMC fallback process in connect syscall, kernel will replaces TCP with SMC. In order to forward wakeup smc socket waitqueue after fallback, kernel will sets clsk->sk_user_data to origin smc socket in smc_fback_replace_callbacks(). Later, in shutdown syscall, kernel will calls sk_psock_get(), which treats the clsk->sk_user_data as psock type, triggering the refcnt warning. So, the root cause is that smc and psock, both will use sk_user_data field. So they will mismatch this field easily. This patch solves it by using another bit(defined as SK_USER_DATA_PSOCK) in PTRMASK, to mark whether sk_user_data points to a psock object or not. This patch depends on a PTRMASK introduced in commit f1ff5ce2cd5e ("net, sk_msg: Clear sk_user_data pointer on clone if tagged"). For there will possibly be more flags in the sk_user_data field, this patch also refactor sk_user_data flags code to be more generic to improve its maintainability.
CVE-2022-49981	In the Linux kernel, the following vulnerability has been resolved: HID: hidraw: fix memory leak in hidraw_release() Free the buffered reports before deleting the list entry. BUG: memory leak unreferenced object 0xfffff88810e72f180 (size 32): comm "softirq", pid 0, jiffies 4294945143 (age 16.080s) hex dump (first 32 bytes): 64 f3 c6 6a d1 88 07 04 00 00 00 00 00 00 00 00 d.j..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<ffffffffff814ac6c3>] kmemdup +0x23/0x50 mm/util.c:128 [<ffffffffff8357c1d2>] kmemdup include/linux/fortify-string.h:440 [inline] [<ffffffffff8357c1d2>] hidraw_report_event+0xa2/0x150 drivers/hid/hidraw.c:521 [<ffffffffff8356ddad>] hid_report_raw_event+0x27d/0x740 drivers/hid/hid-core.c:1992 [<ffffffffff8356e41e>] hid_input_report +0x1ae/0x270 drivers/hid/hid-core.c:2065 [<ffffffffff835f0d3f>] hid_irq_in+0x1ff/0x250 drivers/hid/usbhid/hid-core.c:284 [<ffffffffff82d3c7f9>] __usb_hcd_giveback_urb+0xf9/0x230 drivers/usb/core/hcd.c:1670 [<ffffffffff82d3cc26>] usb_hcd_giveback_urb+0x1b6/0x1d0 drivers/usb/core/hcd.c:1747 [<ffffffffff82ef1e14>] dummy_timer+0x8e4/0x14c0 drivers/usb/gadget/udc/dummy_hcd.c:1988 [<ffffffffff812f50a8>] call_timer_fn+0x38/0x200 kernel/time/timer.c:1474 [<ffffffffff812f5586>] expire_timers kernel/time/timer.c:1519 [inline] [<ffffffffff812f5586>] __run_timers.part.0+0x316/0x430 kernel/time/timer.c:1790 [<ffffffffff812f56e4>] __run_timers kernel/time/timer.c:1768 [inline] [<ffffffffff812f56e4>] run_timer_softirq+0x44/0x90 kernel/time/timer.c:1803 [<ffffffffff848000e6>] __do_softirq+0xe6/0x2ea kernel/softirq.c:571 [<ffffffffff81246db0>] invoke_softirq kernel/softirq.c:445 [inline] [<ffffffffff81246db0>] __irq_exit_rcu kernel/softirq.c:650 [inline] [<ffffffffff81246db0>] irq_exit_rcu+0xc0/0x110 kernel/softirq.c:662 [<ffffffffff84574f02>] sysvec_apic_timer_interrupt +0xa2/0xd0 arch/x86/kernel/apic/apic.c:1106 [<ffffffffff84600c8b>] asm_sysvec_apic_timer_interrupt +0x1b/0x20 arch/x86/include/asm/idtentry.h:649 [<ffffffffff8458a070>] native_safe_halt arch/x86/include/asm/irqflags.h:51 [inline] [<ffffffffff8458a070>] arch_safe_halt arch/x86/include/asm/irqflags.h:89 [inline] [<ffffffffff8458a070>] acpi_safe_halt drivers/acpi/processor_idle.c:111 [inline] [<ffffffffff8458a070>] acpi_idle_do_entry+0xc0/0xd0 drivers/acpi/processor_idle.c:554
CVE-2022-49982	In the Linux kernel, the following vulnerability has been resolved: media: pvrusb2: fix memory leak in pvr_probe The error handling code in pvr2_hdw_create forgets to unregister the v4l2 device. When pvr2_hdw_create returns back to pvr2_context_create, it calls pvr2_context_destroy to destroy context, but mp->hdw is NULL, which leads to that pvr2_hdw_destroy directly returns. Fix this by adding v4l2_device_unregister to decrease the refcount of usb interface.
CVE-2022-49984	In the Linux kernel, the following vulnerability has been resolved: HID: steam: Prevent NULL pointer dereference in steam_{recv,send}_report It is possible for a malicious device to forgo submitting a Feature Report. The HID Steam driver presently makes no provision for this and de-references the 'struct hid_report' pointer obtained from the HID devices without first checking its validity. Let's change that.

CVE ID	Description
CVE-2022-49987	In the Linux kernel, the following vulnerability has been resolved: md: call __md_stop_writes in md_stop From the link [1], we can see raid1d was running even after the path raid_dtr -> md_stop -> __md_stop. Let's stop write first in destructor to align with normal md-raid to fix the KASAN issue. [1]. https://lore.kernel.org/linux-raid/CAPHsuW5gc4AakdGNdF8ubpezAuDLFOYUO_sfMZcec6hQFm8nhg@mail.gmail.com/T/#m7f12bf90481c02c6d2da68c64aedd4779b7df74a
CVE-2022-49989	In the Linux kernel, the following vulnerability has been resolved: xen/privcmd: fix error exit of privcmd_ioctl_dm_op() The error exit of privcmd_ioctl_dm_op() is calling unlock_pages() potentially with pages being NULL, leading to a NULL dereference. Additionally lock_pages() doesn't check for pin_user_pages_fast() having been completely successful, resulting in potentially not locking all pages into memory. This could result in sporadic failures when using the related memory in user mode. Fix all of that by calling unlock_pages() always with the real number of pinned pages, which will be zero in case pages being NULL, and by checking the number of pages pinned by pin_user_pages_fast() matching the expected number of pages.
CVE-2022-49990	In the Linux kernel, the following vulnerability has been resolved: s390: fix double free of GS and RI CBs on fork() failure The pointers for guarded storage and runtime instrumentation control blocks are stored in the thread_struct of the associated task. These pointers are initially copied on fork() via arch_dup_task_struct() and then cleared via copy_thread() before fork() returns. If fork() happens to fail after the initial task dup and before copy_thread(), the newly allocated task and associated thread_struct memory are freed via free_task() -> arch_release_task_struct(). This results in a double free of the guarded storage and runtime info structs because the fields in the failed task still refer to memory associated with the source task. This problem can manifest as a BUG_ON() in set_freepointer() (with CONFIG_SLAB_FREELIST_HARDENED enabled) or KASAN splat (if enabled) when running trinity syscall fuzz tests on s390x. To avoid this problem, clear the associated pointer fields in arch_dup_task_struct() immediately after the new task is copied. Note that the RI flag is still cleared in copy_thread() because it resides in thread stack memory and that is where stack info is copied.
CVE-2022-49993	In the Linux kernel, the following vulnerability has been resolved: loop: Check for overflow while configuring loop The userspace can configure a loop using an ioctl call, wherein a configuration of type loop_config is passed (see lo_ioctl()'s case on line 1550 of drivers/block/loop.c). This proceeds to call loop_configure() which in turn calls loop_set_status_from_info() (see line 1050 of loop.c), passing &config->info which is of type loop_info64*. This function then sets the appropriate values, like the offset. loop_device has lo_offset of type loff_t (see line 52 of loop.c), which is typedef-chained to long long, whereas loop_info64 has lo_offset of type __u64 (see line 56 of include/uapi/linux/loop.h). The function directly copies offset from info to the device as follows (See line 980 of loop.c): lo->lo_offset = info->lo_offset; This results in an overflow, which triggers a warning in iomap_iter() due to a call to iomap_iter_done() which has: WARN_ON_ONCE(iter->iomap.offset > iter->pos); Thus, check for negative value during loop_set_status_from_info(). Bug report: https://syzkaller.appspot.com/bug?id=c620fe14aac810396d3c3edc9ad73848bf69a29e

CVE ID	Description
CVE-2022-50000	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable: fix stuck flows on cleanup due to pending work To clear the flow table on flow table free, the following sequence normally happens in order: 1) gc_step work is stopped to disable any further stats/del requests. 2) All flow table entries are set to teardown state. 3) Run gc_step which will queue HW del work for each flow table entry. 4) Waiting for the above del work to finish (flush). 5) Run gc_step again, deleting all entries from the flow table. 6) Flow table is freed. But if a flow table entry already has pending HW stats or HW add work step 3 will not queue HW del work (it will be skipped), step 4 will wait for the pending add/stats to finish, and step 5 will queue HW del work which might execute after freeing of the flow table. To fix the above, this patch flushes the pending work, then it sets the teardown flag to all flows in the flowtable and it forces a garbage collector run to queue work to remove the flows from hardware, then it flushes this new pending work and (finally) it forces another garbage collector run to remove the entry from the software flowtable. Stack trace: [47773.882335] BUG: KASAN: use-after-free in down_read+0x99/0x460 [47773.883634] Write of size 8 at addr ffff888103b45aa8 by task kworker/u20:6/543704 [47773.885634] CPU: 3 PID: 543704 Comm: kworker/u20:6 Not tainted 5.12.0-rc7+ #2 [47773.886745] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009) [47773.888438] Workqueue: nf_flow_offload_del flow_offload_work_handler [nf_flow_table] [47773.889727] Call Trace: [47773.890214] dump_stack+0xbb/0x107 [47773.890818] print_address_description.constprop.0+0x18/0x140 [47773.892990] kasan_report.cold+0x7c/0xd8 [47773.894459] kasan_check_range+0x145/0x1a0 [47773.895174] down_read+0x99/0x460 [47773.899706] nf_flow_offload_tuple+0x24f/0x3c0 [nf_flow_table] [47773.907137] flow_offload_work_handler+0x72d/0xbe0 [nf_flow_table] [47773.913372] process_one_work+0x8ac/0x14e0 [47773.921325] [47773.921325] Allocated by task 592159: [47773.922031] kasan_save_stack+0x1b/0x40 [47773.922730] __kasan_kmalloc+0x7a/0x90 [47773.923411] tcf_ct_flow_table_get+0x3cb/0x1230 [act_ct] [47773.924363] tcf_ct_init+0x71c/0x1156 [act_ct] [47773.925207] tcf_action_init_1+0x45b/0x700 [47773.925987] tcf_action_init+0x453/0x6b0 [47773.926692] tcf_exts_validate+0x3d0/0x600 [47773.927419] fl_change+0x757/0x4a51 [cls_flowler] [47773.928227] tc_new_tfilter+0x89a/0x2070 [47773.936652] [47773.936652] Freed by task 543704: [47773.937303] kasan_save_stack+0x1b/0x40 [47773.938039] kasan_set_track+0x1c/0x30 [47773.938731] kasan_set_free_info+0x20/0x30 [47773.939467] __kasan_slab_free+0xe7/0x120 [47773.940194] slab_free_freelist_hook+0x86/0x190 [47773.941038] kfree+0xce/0x3a0 [47773.941644] tcf_ct_flow_table_cleanup_work Original patch description and stack trace by Paul Blakey.
CVE-2022-50004	In the Linux kernel, the following vulnerability has been resolved: xfrm: policy: fix metadata dst->dev xmit null pointer dereference When we try to transmit an skb with metadata_dst attached (i.e. dst->dev == NULL) through xfrm interface we can hit a null pointer dereference[1] in xfrmi_xmit2() -> xfrm_lookup_with_ifid() due to the check for a loopback skb device when there's no policy which dereferences dst->dev unconditionally. Not having dst->dev can be interpreted as it not being a loopback device, so just add a check for a null dst_orig->dev. With this fix xfrm interface's Tx error counters go up as usual. [1] net-next calltrace captured via netconsole: BUG: kernel NULL pointer dereference, address: 00000000000000c0 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP CPU: 1 PID: 7231 Comm: ping Kdump: loaded Not tainted 5.19.0+ #24 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.0-1.fc36 04/01/2014 RIP: 0010:xfrm_lookup_with_ifid+0x5eb/0xa60 Code: 8d 74 24 38 e8 26 a4 37 00 48 89 c1 e9 12 fc ff ff 49 63 ed 41 83 fd be 0f 85 be 01 00 00 41 be ff ff ff ff 45 31 ed 48 8b 03 <f6> 80 c0 00 00 00 08 75 0f 41 80 bc 24 19 0d 00 00 01 0f 84 1e 02 RSP: 0018:ffffb0db82c679f0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: fffffb0db7fcd430 RCX: fffffb0db82c67a10 RDX: 0000000000000000 RSI: 0000000000000000 RDI: fffffb0db82c67a80 RBP: fffffb0db82c67a80 R08: fffffb0db82c67a14 R09: 0000000000000000 R10: 0000000000000000 R11: fffff8fa449667dc8 R12: ffffffff966db880 R13: 0000000000000000 R14: 00000000ffffff R15: 0000000000000000 FS: 00007ff35c83f000(0000) GS:ffff8fa478480000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000000000c0 CR3: 000000001ebb7000 CR4: 0000000000350ee0 Call Trace: <TASK> xfrmi_xmit+0xde/0x460 ? tcf_bpf_act+0x13d/0x2a0 dev_hard_start_xmit+0x72/0x1e0 __dev_queue_xmit+0x251/0xd30 ip_finish_output2+0x140/0x550 ip_push_pending_frames+0x56/0x80 raw_sendmsg+0x663/0x10a0 ? try_charge_memcg+0x3fd/0x7a0 ? __mod_memcg_lruvec_state+0x93/0x110 ? sock_sendmsg+0x30/0x40 sock_sendmsg+0x30/0x40 __sys_sendto+0xeb/0x130 ? handle_mm_fault+0xae/0x280 ? do_user_addr_fault+0x1e7/0x680 ? kvm_read_and_reset_apf_flags+0x3b/0x50 __x64_sys_sendto+0x20/0x30 do_syscall_64+0x34/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7ff35cac1366 Code: eb 0b 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b8 0f 1f 00 41 89 ca 64 8b 04 25 18 00 00 00 85 c0 75 11 b8 2c 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 72 c3 90 55 48 83 ec 30 44 89 4c 24 2c 4c 89 RSP: 002b:00007fff738e4028 EFLAGS: 00000246 ORIG_RAX: 000000000000002c RAX: ffffffff7ffda RBX: 00007fff738e57b0 RCX: 00007ff35cac1366 RDX: 0000000000000040 RSI: 0000557164e4b450 RDI: 0000000000000003 RBP: 0000557164e4b450 R08: 00007fff738e7a2c R09: 0000000000000010 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000040 R13: 00007fff738e5770 R14: 00007fff738e4030 R15: 00000001d0000001 </TASK> Modules linked in: netconsole veth br_netfilter bridge bonding virtio_net [last unloaded: netconsole] CR2: 00000000000000c0

CVE ID	Description
CVE-2022-50008	In the Linux kernel, the following vulnerability has been resolved: kprobes: don't call disarm_kprobe() for disabled kprobes The assumption in __disable_kprobe() is wrong, and it could try to disarm an already disarmed kprobe and fire the WARN_ONCE() below. [0] We can easily reproduce this issue. 1. Write 0 to /sys/kernel/debug/kprobes/enabled. # echo 0 > /sys/kernel/debug/kprobes/enabled 2. Run execsnoop. At this time, one kprobe is disabled. # /usr/share/bcc/tools/execsnoop & [1] 2460 PCOMM PID PPID RET ARGS # cat /sys/kernel/debug/kprobes/list ffffffff91345650 r __x64_sys_execve+0x0 [FTRACE] ffffffff91345650 k __x64_sys_execve+0x0 [DISABLED] [FTRACE] 3. Write 1 to /sys/kernel/debug/kprobes/enabled, which changes kprobes_all_disarmed to false but does not arm the disabled kprobe. # echo 1 > /sys/kernel/debug/kprobes/enabled # cat /sys/kernel/debug/kprobes/list ffffffff91345650 r __x64_sys_execve+0x0 [FTRACE] ffffffff91345650 k __x64_sys_execve+0x0 [DISABLED][FTRACE] 4. Kill execsnoop, when __disable_kprobe() calls disarm_kprobe() for the disabled kprobe and hits the WARN_ONCE() in __disarm_kprobe_ftrace(). # fg /usr/share/bcc/tools/execsnoop ^C Actually, WARN_ONCE() is fired twice, and __unregister_kprobe_top() misses some cleanups and leaves the aggregated kprobe in the hash table. Then, __unregister_trace_kprobe() initialises tk->rp.kp.list and creates an infinite loop like this. aggregated kprobe.list -> kprobe.list -. ^ '._. In this situation, these commands fall into the infinite loop and result in RCU stall or soft lockup. cat /sys/kernel/debug/kprobes/list : show_kprobe_addr() enters into the infinite loop with RCU. /usr/share/bcc/tools/execsnoop : warn_kprobe_rereg() holds kprobe_mutex, and __get_valid_kprobe() is stuck in the loop. To avoid the issue, make sure we don't call disarm_kprobe() for disabled kprobes. [0] Failed to disarm kprobe-ftrace at __x64_sys_execve+0x0/0x40 (error -2) WARNING: CPU: 6 PID: 2460 at kernel/kprobes.c:1130 __disarm_kprobe_ftrace.isra.19 (kernel/kprobes.c:1129) Modules linked in: ena CPU: 6 PID: 2460 Comm: execsnoop Not tainted 5.19.0+ #28 Hardware name: Amazon EC2 c5.2xlarge/, BIOS 1.0 10/16/2017 RIP: 0010: __disarm_kprobe_ftrace.isra.19 (kernel/kprobes.c:1129) Code: 24 8b 02 eb c1 80 3d c4 83 f2 01 00 75 d4 48 8b 75 00 89 c2 48 c7 c7 90 fa 0f 92 89 04 24 c6 05 ab 83 01 e8 e4 94 f0 ff <0f> 0b 8b 04 24 eb b1 89 c6 48 c7 c7 60 fa 0f 92 89 04 24 e8 cc 94 RSP: 0018:ffff9e6c154bd98 EFLAGS: 00010282 RAX: 0000000000000000 RBX: ffffffff930f7b00 RCX: 0000000000000001 RDX: 0000000080000001 RSI: ffffffff921461c5 RDI: 00000000ffffffff RBP: ffff89c504286da8 R08: 0000000000000000 R09: c0000000ffffffffff R10: 0000000000000000 R11: ffff9e6ec154bc28 R12: ffff89c502394e40 R13: ffff89c502394c00 R14: ffff9e6ec154bc00 R15: 0000000000000000 FS: 00007fe800398740(0000) GS: ffff89c812d80000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000c00057f010 CR3: 0000000103b54006 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> __disable_kprobe (kernel/kprobes.c:1716) disable_kprobe (kernel/kprobes.c:2392) __disable_trace_kprobe (kernel/trace/trace_kprobe.c:340) disable_trace_kprobe (kernel/trace/trace_kprobe.c:429) perf_trace_event_unreg.isra.2 (/include/linux/tracepoint.h:93 kernel/trace/trace_event_perf.c:168) perf_kprobe_destroy (kernel/trace/trace_event_perf.c:295) _free_event (kernel/events/core.c:4971) perf_event_release_kernel (kernel/events/core.c:5176) perf_release (kernel/events/core.c:5186) __fput (fs/file_table.c:321) task_work_run (/include/linux/---truncated---
CVE-2022-50012	In the Linux kernel, the following vulnerability has been resolved: powerpc/64: Init jump labels before parse_early_param() On 64-bit, calling jump_label_init() in setup_feature_keys() is too late because static keys may be used in subroutines of parse_early_param() which is again subroutine of early_init_devtree(). For example booting with "threadirqs": static_key_enable_cpuslocked(): static key '0xc00000002953260' used before call to jump_label_init() WARNING: CPU: 0 PID: 0 at kernel/jump_label.c:166 static_key_enable_cpuslocked+0xfc/0x120 ... NIP static_key_enable_cpuslocked+0xfc/0x120 LR static_key_enable_cpuslocked+0xf8/0x120 Call Trace: static_key_enable_cpuslocked+0xf8/0x120 (unreliable) static_key_enable+0x30/0x50 setup_forced_irqthreads+0x28/0x40 do_early_param+0xa0/0x108 parse_args+0x290/0x4e0 parse_early_options+0x48/0x5c parse_early_param+0x58/0x84 early_init_devtree+0xd4/0x518 early_setup+0xb4/0x214 So call jump_label_init() just before parse_early_param() in early_init_devtree(). [mpe: Add call trace to change log and minor wording edits.]
CVE-2022-50016	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: cnl: Do not process IPC reply before firmware boot It is not yet clear, but it is possible to create a firmware so broken that it will send a reply message before a FW_READY message (it is not yet clear if FW_READY will arrive later). Since the reply_data is allocated only after the FW_READY message, this will lead to a NULL pointer dereference if not filtered out. The issue was reported with IPC4 firmware but the same condition is present for IPC3.

CVE ID	Description
CVE-2022-50021	In the Linux kernel, the following vulnerability has been resolved: ext4: block range must be validated before use in ext4_mb_clear_bb() Block range to free is validated in ext4_free_blocks() using ext4_inode_block_valid() and then it's passed to ext4_mb_clear_bb(). However in some situations on bigalloc file system the range might be adjusted after the validation in ext4_free_blocks() which can lead to troubles on corrupted file systems such as one found by syzkaller that resulted in the following BUG kernel BUG at fs/ext4/ext4.h:3319! PREEMPT SMP NOPTI CPU: 28 PID: 4243 Comm: repro Kdump: loaded Not tainted 5.19.0-rc6+ #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1.fc35 04/01/2014 RIP: 0010:ext4_free_blocks+0x95e/0xa90 Call Trace: <TASK> ? lock_timer_base+0x61/0x80 ? __es_remove_extent+0x5a/0x760 ? __mod_timer+0x256/0x380 ? ext4_ind_truncate_ensure_credits+0x90/0x220 ext4_clear_blocks+0x107/0x1b0 ext4_free_data+0x15b/0x170 ext4_ind_truncate+0x214/0x2c0 ? __raw_spin_unlock+0x15/0x30 ? ext4_discard_preallocations+0x15a/0x410 ? ext4_journal_check_start+0xe/0x90 ? __ext4_journal_start_sb+0x2f/0x110 ext4_truncate+0x1b5/0x460 ? __ext4_journal_start_sb+0x2f/0x110 ext4_evict_inode+0x2b4/0x6f0 evict+0xd0/0x1d0 ext4_enable_quotas+0x11f/0x1f0 ext4_orphan_cleanup+0x3de/0x430 ? proc_create_seq_private+0x43/0x50 ext4_fill_super+0x295f/0x3ae0 ? snprintf+0x39/0x40 ? sget_fc+0x19c/0x330 ? ext4_reconfigure+0x850/0x850 get_tree_bdev+0x16d/0x260 vfs_get_tree+0x25/0xb0 path_mount+0x431/0xa70 __x64_sys_mount+0xe2/0x120 do_syscall_64+0x5b/0x80 ? do_user_addr_fault+0x1e2/0x670 ? exc_page_fault+0x70/0x170 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7fd4e512ace Fix it by making sure that the block range is properly validated before used every time it changes in ext4_free_blocks() or ext4_mb_clear_bb().
CVE-2022-50025	In the Linux kernel, the following vulnerability has been resolved: cxl: Fix a memory leak in an error handling path A bitmap_zalloc() must be balanced by a corresponding bitmap_free() in the error handling path of afu_allocate_irqs().
CVE-2022-50045	In the Linux kernel, the following vulnerability has been resolved: powerpc/pci: Fix get_phb_number() locking The recent change to get_phb_number() causes a DEBUG_ATOMIC_SLEEP warning on some systems: BUG: sleeping function called from invalid context at kernel/locking/mutex.c:580 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1, name: swapper preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 1 lock held by swapper/1: #0: c157efb0 (hose_spinlock){+..-}-{2:2}, at: pcibios_alloc_controller+0x64/0x220 Preemption disabled at: [<00000000>] 0x0 CPU: 0 PID: 1 Comm: swapper Not tainted 5.19.0-yocto-standard+ #1 Call Trace: [d101dc90] [c073b264] dump_stack_lvl+0x50/0x8c (unreliable) [d101dcb0] [c0093b70] __might_resched+0x258/0x2a8 [d101dcd0] [c0d3e634] __mutex_lock+0x6c/0x6ec [d101dd50] [c0a84174] of_alias_get_id+0x50/0xf4 [d101dd80] [c002ec78] pcibios_alloc_controller+0x1b8/0x220 [d101ddd0] [c140c9dc] pmac_pci_init+0x198/0x784 [d101de50] [c140852c] discover_phbs+0x30/0x4c [d101de60] [c0007fd4] do_one_initcall+0x94/0x344 [d101ded0] [c1403b40] kernel_init_freeable+0x1a8/0x22c [d101df10] [c00086e0] kernel_init+0x34/0x160 [d101df30] [c001b334] ret_from_kernel_thread+0x5c/0x64 This is because pcibios_alloc_controller() holds hose_spinlock but of_alias_get_id() takes of_mutex which can sleep. The hose_spinlock protects the phb_bitmap, and also the hose_list, but it doesn't need to be held while get_phb_number() calls the OF routines, because those are only looking up information in the device tree. So fix it by having get_phb_number() take the hose_spinlock itself, only where required, and then dropping the lock before returning. pcibios_alloc_controller() then needs to take the lock again before the list_add() but that's safe, the order of the list is not important.
CVE-2022-50048	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: possible module reference underflow in error path dst->ops is set on when nft_expr_clone() fails, but module refcount has not been bumped yet, therefore nft_expr_destroy() leads to module reference underflow.
CVE-2022-50050	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda: Fix potential buffer overflow by snprintf() snprintf() returns the would-be-filled size when the string overflows the given buffer size, hence using this value may result in the buffer overflow (although it's unrealistic). This patch replaces with a safer version, scnprintf() for papering over such a potential issue.
CVE-2022-50051	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: debug: Fix potential buffer overflow by snprintf() snprintf() returns the would-be-filled size when the string overflows the given buffer size, hence using this value may result in the buffer overflow (although it's unrealistic). This patch replaces with a safer version, scnprintf() for papering over such a potential issue.
CVE-2022-50055	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix adminq error handling iavf_alloc_asq_bufs/iavf_alloc_arq_bufs allocates with dma_alloc_coherent memory for VF mailbox. Free DMA regions for both ASQ and ARQ in case error happens during configuration of ASQ/ARQ registers. Without this change it is possible to see when unloading interface: 74626.583369: dma_debug_device_change: device driver has pending DMA allocations while released from device [count=32] One of leaked entries details: [device address=0x0000000b27ff9000] [size=4096 bytes] [mapped with DMA_BIDIRECTIONAL] [mapped as coherent]

CVE ID	Description
CVE-2022-50059	In the Linux kernel, the following vulnerability has been resolved: ceph: don't leak snap_rwsem in handle_cap_grant When handle_cap_grant is called on an IMPORT op, then the snap_rwsem is held and the function is expected to release it before returning. It currently fails to do that in all cases which could lead to a deadlock.
CVE-2022-50065	In the Linux kernel, the following vulnerability has been resolved: virtio_net: fix memory leak inside XPD_TX with mergeable When we call xdp_convert_buff_to_frame() to get xdpf, if it returns NULL, we should check if xdp_page was allocated by xdp_linearize_page(). If it is newly allocated, it should be freed here alone. Just like any other "goto err_xdp".
CVE-2022-50071	In the Linux kernel, the following vulnerability has been resolved: mptcp: move subflow cleanup in mptcp_destroy_common() If the mptcp socket creation fails due to a CGROUP_INET_SOCKET_CREATE eBPF program, the MPTCP protocol ends-up leaking all the subflows: the related cleanup happens in __mptcp_destroy_sock() that is not invoked in such code path. Address the issue moving the subflow sockets cleanup in the mptcp_destroy_common() helper, which is invoked in every msk cleanup path. Additionally get rid of the intermediate list_splice_init step, which is an unneeded relic from the past. The issue is present since before the reported root cause commit, but any attempt to backport the fix before that hash will require a complete rewrite.
CVE-2022-50079	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check correct bounds for stream encoder instances for DCN303 [Why & How] eng_id for DCN303 cannot be more than 1, since we have only two instances of stream encoders. Check the correct boundary condition for engine ID for DCN303 prevent the potential out of bounds access.
CVE-2022-50080	In the Linux kernel, the following vulnerability has been resolved: tee: add overflow check in register_shm_helper() With special lengths supplied by user space, register_shm_helper() has an integer overflow when calculating the number of pages covered by a supplied user space memory region. This causes internal_get_user_pages_fast() a helper function of pin_user_pages_fast() to do a NULL pointer dereference: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000010 Modules linked in: CPU: 1 PID: 173 Comm: optee_example_a Not tainted 5.19.0 #11 Hardware name: QEMU QEMU Virtual Machine, BIOS 0.0.0 02/06/2015 pc : internal_get_user_pages_fast+0x474/0xa80 Call trace: internal_get_user_pages_fast+0x474/0xa80 pin_user_pages_fast+0x24/0x4c register_shm_helper+0x194/0x330 tee_shm_register_user_buf+0x78/0x120 tee_ioctl+0xd0/0x11a0 __arm64_sys_ioctl+0xa8/0xec invoke_syscall+0x48/0x114 Fix this by adding an explicit call to access_ok() in tee_shm_register_user_buf() to catch an invalid user space address early.
CVE-2022-50082	In the Linux kernel, the following vulnerability has been resolved: ext4: fix warning in ext4_iomap_begin as race between bmap and write We got issue as follows: -----[cut here]----- WARNING: CPU: 3 PID: 9310 at fs/ext4/inode.c:3441 ext4_iomap_begin+0x182/0x5d0 RIP: 0010:ext4_iomap_begin+0x182/0x5d0 RSP: 0018:ffff88812460fa08 EFLAGS: 00010293 RAX: ffff88811f168000 RBX: 0000000000000000 RCX: ffffffff97793c12 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003 RBP: ffff88812c669160 R08: ffff88811f168000 R09: ffffd10258cd20f R10: ffff88812c669077 R11: ffffd10258cd20e R12: 0000000000000001 R13: 00000000000000a4 R14: 000000000000000c R15: ffff88812c6691ee FS: 00007fd0d6ff3740(0000) GS:ffff8883af180000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fd0d6dda290 CR3: 0000000104a62000 CR4: 0000000000000060 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 Call Trace: iomap_apply+0x119/0x570 iomap_bmap+0x124/0x150 ext4_bmap+0x14f/0x250 bmap+0x55/0x80 do_vfs_ioctl+0x952/0xbd0 __x64_sys_ioctl+0xc6/0x170 do_syscall_64+0x33/0x40 entry_SYSCALL_64_after_hwframe+0x44/0xa9 Above issue may happen as follows: bmap write bmap ext4_bmap iomap_bmap ext4_iomap_begin ext4_file_write_iter ext4_buffered_write_iter generic_perform_write ext4_da_write_begin ext4_da_write_inline_data_begin ext4_prepare_inline_data ext4_create_inline_data ext4_set_inode_flag(inode, EXT4_INODE_INLINE_DATA); if (WARN_ON_ONCE(ext4_has_inline_data(inode))) ->trigger bug_on To solved above issue hold inode lock in ext4_bmap.

CVE ID	Description
CVE-2022-50084	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix address sanitizer warning in raid_status There is this warning when using a kernel with the address sanitizer and running this testsuite: https://gitlab.com/cki-project/kernel-tests/-/tree/main/storage/swraid/scsi_raid <pre>===== BUG: KASAN: slab-out-of-bounds in raid_status+0x1747/0x2820 [dm_raid] Read of size 4 at addr ffff888079d2c7e8 by task lvcreate/13319 CPU: 0 PID: 13319 Comm: lvcreate Not tainted 5.18.0-0.rc3.<snip> #1 Hardware name: Red Hat KVM, BIOS 0.5.1 01/01/2011 Call Trace: <TASK> dump_stack_lvl+0x6a/0x9c print_address_description.constprop.0+0x1f/0x1e0 print_report.cold+0x55/0x244 kasan_report+0xc9/0x100 raid_status+0x1747/0x2820 [dm_raid] dm_ima_measure_on_table_load+0x4b8/0xca0 [dm_mod] table_load+0x35c/0x630 [dm_mod] ctl_ioctl+0x411/0x630 [dm_mod] dm_ctl_ioctl+0xa/0x10 [dm_mod] __x64_sys_ioctl+0x12a/0x1a0 do_syscall_64+0x5b/0x80 The warning is caused by reading conf->max_nr_stripes in raid_status. The code in raid_status reads mddev->private, casts it to struct r5conf and reads the entry max_nr_stripes. However, if we have different raid type than 4/5/6, mddev->private doesn't point to struct r5conf; it may point to struct r0conf, struct r1conf, struct r10conf or struct mpconf. If we cast a pointer to one of these structs to struct r5conf, we will be reading invalid memory and KASAN warns about it. Fix this bug by reading struct r5conf only if raid type is 4, 5 or 6.</pre>
CVE-2022-50085	In the Linux kernel, the following vulnerability has been resolved: dm raid: fix address sanitizer warning in raid_resume There is a KASAN warning in raid_resume when running the lvm test lvconvert-raid.sh. The reason for the warning is that mddev->raid_disks is greater than rs->raid_disks, so the loop touches one entry beyond the allocated length.
CVE-2022-50086	In the Linux kernel, the following vulnerability has been resolved: block: don't allow the same type rq_qos add more than once In our test of iocost, we encountered some list add/del corruptions of inner_walk list in ioc_timer_fn. The reason can be described as follows: cpu 0 cpu 1 ioc_qos_write ioc_qos_write ioc = q_to_ioc(queue); if (!ioc) { ioc = kzalloc(); ioc = q_to_ioc(queue); if (!ioc) { ioc = kzalloc(); ... rq_qos_add(q, rqos); } ... rq_qos_add(q, rqos); ... } When the io.cost.qos file is written by two cpus concurrently, rq_qos may be added to one disk twice. In that case, there will be two iocs enabled and running on one disk. They own different iocgs on their active list. In the ioc_timer_fn function, because of the iocgs from two iocs have the same root iocg, the root iocg's walk_list may be overwritten by each other and this leads to list add/del corruptions in building or destroying the inner_walk list. And so far, the blk-rq-qos framework works in case that one instance for one type rq_qos per queue by default. This patch make this explicit and also fix the crash above.
CVE-2022-50088	In the Linux kernel, the following vulnerability has been resolved: mm/damon/reclaim: fix potential memory leak in damon_reclaim_init() damon_reclaim_init() allocates a memory chunk for ctx with damon_new_ctx(). When damon_select_ops() fails, ctx is not released, which will lead to a memory leak. We should release the ctx with damon_destroy_ctx() when damon_select_ops() fails to fix the memory leak.

CVE ID	Description
CVE-2022-50090	In the Linux kernel, the following vulnerability has been resolved: btrfs: replace BTRFS_MAX_EXTENT_SIZE with fs_info->max_extent_size On zoned filesystem, data write out is limited by max_zone_append_size, and a large ordered extent is split according the size of a bio. OTOH, the number of extents to be written is calculated using BTRFS_MAX_EXTENT_SIZE, and that estimated number is used to reserve the metadata bytes to update and/or create the metadata items. The metadata reservation is done at e.g. btrfs_buffered_write() and then released according to the estimation changes. Thus, if the number of extent increases massively, the reserved metadata can run out. The increase of the number of extents easily occurs on zoned filesystem if BTRFS_MAX_EXTENT_SIZE > max_zone_append_size. And, it causes the following warning on a small RAM environment with disabling metadata over-commit (in the following patch). <pre>[75721.498492] -----[cut here]----- [75721.505624] BTRFS: block rsv 1 returned -28 [75721.512230] WARNING: CPU: 24 PID: 2327559 at fs/btrfs/block-rsv.c:537 btrfs_use_block_rsv +0x560/0x760 [btrfs] [75721.581854] CPU: 24 PID: 2327559 Comm: kworker/u64:10 Kdump: loaded Tainted: G W 5.18.0-rc2-BTRFS-ZNS+ #109 [75721.597200] Hardware name: Supermicro Super Server/H12SSL-NT, BIOS 2.0 02/22/2021 [75721.607310] Workqueue: btrfs-endio- write btrfs_work_helper [btrfs] [75721.616209] RIP: 0010:btrfs_use_block_rsv+0x560/0x760 [btrfs] [75721.646649] RSP: 0018:ffffc9000fbdf3e0 EFLAGS: 00010286 [75721.654126] RAX: 0000000000000000 RBX: 0000000000000400 RCX: 0000000000000000 [75721.663524] RDX: 0000000000000004 RSI: 0000000000000008 RDI: fffff52001f7be6e [75721.672921] RBP: fffff9000fbdf420 R08: 0000000000000001 R09: ffff889f8d1fc6c7 [75721.682493] R10: ffffd13f1a3f8d8 R11: 0000000000000001 R12: ffff88980a3c0e28 [75721.692284] R13: ffff889b66590000 R14: ffff88980a3c0e40 R15: ffff88980a3c0e8a [75721.701878] FS: 0000000000000000(0000) GS:ffff889f8d000000(0000) knlGS:0000000000000000 [75721.712601] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [75721.720726] CR2: 000055d12e05c018 CR3: 0000800193594000 CR4: 0000000000350ee0 [75721.730499] Call Trace: [75721.735166] <TASK> [75721.739886] btrfs_alloc_tree_block+0x1e1/0x1100 [btrfs] [75721.747545] ? btrfs_alloc_logged_file_extent+0x550/0x550 [btrfs] [75721.756145] ? btrfs_get_32+0xea/0x2d0 [btrfs] [75721.762852] ? btrfs_get_32+0xea/0x2d0 [btrfs] [75721.769520] ? push_leaf_left +0x420/0x620 [btrfs] [75721.776431] ? memcpy+0x4e/0x60 [75721.781931] split_leaf +0x433/0x12d0 [btrfs] [75721.788392] ? btrfs_get_token_32+0x580/0x580 [btrfs] [75721.795636] ? push_for_double_split.isra.0+0x420/0x420 [btrfs] [75721.803759] ? leaf_space_used+0x15d/0x1a0 [btrfs] [75721.811156] btrfs_search_slot+0x1bc3/0x2790 [btrfs] [75721.818300] ? lock_downgrade +0x7c0/0x7c0 [75721.824411] ? free_extent_buffer.part.0+0x107/0x200 [btrfs] [75721.832456] ? split_leaf+0x12d0/0x12d0 [btrfs] [75721.839149] ? free_extent_buffer.part.0+0x14f/0x200 [btrfs] [75721.846945] ? free_extent_buffer+0x13/0x20 [btrfs] [75721.853960] ? btrfs_release_path +0x4b/0x190 [btrfs] [75721.861429] btrfs_csum_file_blocks+0x85c/0x1500 [btrfs] [75721.869313] ? rcu_read_lock_sched_held+0x16/0x80 [75721.876085] ? lock_release+0x552/0xf80 [75721.881957] ? btrfs_del_csums+0x8c0/0x8c0 [btrfs] [75721.888886] ? __kasan_check_write+0x14/0x20 [75721.895152] ? do_raw_read_unlock+0x44/0x80 [75721.901323] ? _raw_write_lock_irq +0x60/0x80 [75721.907983] ? btrfs_global_root+0xb9/0xe0 [btrfs] [75721.915166] ? btrfs_csum_root +0x12b/0x180 [btrfs] [75721.921918] ? btrfs_get_global_root+0x820/0x820 [btrfs] [75721.929166] ? _raw_write_unlock+0x23/0x40 [75721.935116] ? unpin_extent_cache+0x1e3/0x390 [btrfs] [75721.942041] btrfs_finish_ordered_io.isra.0+0xa0c/0x1dc0 [btrfs] [75721.949906] ? try_to_wake_up +0x30/0x14a0 [75721.955700] ? btrfs_unlink_subvol+0xda0/0xda0 [btrfs] [75721.962661] ? rcu --- truncated--- </pre>
CVE-2022-50098	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix crash due to stale SRB access around I/O timeouts Ensure SRB is returned during I/O timeout error escalation. If that is not possible fail the escalation path. Following crash stack was seen: BUG: unable to handle kernel paging request at 0000002f56aa90f8 IP: qla_chk_edif_rx_sa_delete_pending+0x14/0x30 [qla2xxx] Call Trace: ? qla2x00_status_entry+0x19f/0x1c50 [qla2xxx] ? qla2x00_start_sp+0x116/0x1170 [qla2xxx] ? dma_pool_alloc+0x1d6/0x210 ? mempool_alloc+0x54/0x130 ? qla2xxx_process_response_queue+0x548/0x12b0 [qla2xxx] ? qla_do_work+0x2d/0x40 [qla2xxx] ? process_one_work+0x14c/0x390

CVE ID	Description
CVE-2022-50100	In the Linux kernel, the following vulnerability has been resolved: sched/core: Do not requeue task on CPU excluded from cpus_mask The following warning was triggered on a large machine early in boot on a distribution kernel but the same problem should also affect mainline. WARNING: CPU: 439 PID: 10 at ./kernel/workqueue.c:2231 process_one_work+0x4d/0x440 Call Trace: <TASK> rescuer_thread+0x1f6/0x360 kthread+0x156/0x180 ret_from_fork+0x22/0x30 </TASK> Commit c6e7bd7afaeb ("sched/core: Optimize ttwu() spinning on p->on_cpu") optimises ttwu by queueing a task that is descheduling on the wakelist, but does not check if the task descheduling is still allowed to run on that CPU. In this warning, the problematic task is a workqueue rescue thread which checks if the rescue is for a per-cpu workqueue and running on the wrong CPU. While this is early in boot and it should be possible to create workers, the rescue thread may still used if the MAYDAY_INITIAL_TIMEOUT is reached or MAYDAY_INTERVAL and on a sufficiently large machine, the rescue thread is being used frequently. Tracing confirmed that the task should have migrated properly using the stopper thread to handle the migration. However, a parallel wakeup from udev running on another CPU that does not share CPU cache observes p->on_cpu and uses task_cpu(p), queues the task on the old CPU and triggers the warning. Check that the wakee task that is descheduling is still allowed to run on its current CPU and if not, wait for the descheduling to complete and select an allowed CPU.
CVE-2022-50103	In the Linux kernel, the following vulnerability has been resolved: sched, cpuset: Fix dl_cpu_busy() panic due to empty cs->cpus_allowed With cgroup v2, the cpuset's cpus_allowed mask can be empty indicating that the cpuset will just use the effective CPUs of its parent. So cpuset_can_attach() can call task_can_attach() with an empty mask. This can lead to cpumask_any_and() returns nr_cpu_ids causing the call to dl_bw_of() to crash due to percpu value access of an out of bound CPU value. For example: [80468.182258] BUG: unable to handle page fault for address: ffffffff8b6648b0 : [80468.191019] RIP: 0010:dl_cpu_busy+0x30/0x2b0 : [80468.207946] Call Trace: [80468.208947] cpuset_can_attach+0xa0/0x140 [80468.209953] cgroup_migrate_execute+0x8c/0x490 [80468.210931] cgroup_update_dfl_csses+0x254/0x270 [80468.211898] cgroup_subtree_control_write+0x322/0x400 [80468.212854] kernfs_fop_write_iter+0x11c/0x1b0 [80468.213777] new_sync_write+0x11f/0x1b0 [80468.214689] vfs_write+0x1eb/0x280 [80468.215592] ksys_write+0x5f/0xe0 [80468.216463] do_syscall_64+0x5c/0x80 [80468.224287] entry_SYSCALL_64_after_hwframe+0x44/0xae Fix that by using effective_cpus instead. For cgroup v1, effective_cpus is the same as cpus_allowed. For v2, effective_cpus is the real cpumask to be used by tasks within the cpuset anyway. Also update task_can_attach()'s 2nd argument name to cs_effective_cpus to reflect the change. In addition, a check is added to task_can_attach() to guard against the possibility that cpumask_any_and() may return a value >= nr_cpu_ids.
CVE-2022-50104	In the Linux kernel, the following vulnerability has been resolved: powerpc/xive: Fix refcount leak in xive_get_max_prio of_find_node_by_path() returns a node pointer with refcount incremented, we should use of_node_put() on it when done. Add missing of_node_put() to avoid refcount leak.
CVE-2022-50107	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix memory leak when using fscache If we hit the 'index == next_cached' case, we leak a refcount on the struct page. Fix this by using readahead_folio() which takes care of the refcount for you.
CVE-2022-50113	In the Linux kernel, the following vulnerability has been resolved: ASoc: audio-graph-card2: Fix refcount leak bug in __graph_get_type() We should call of_node_put() for the reference before its replacement as it returned by of_get_parent() which has increased the refcount. Besides, we should also call of_node_put() before return.

CVE ID	Description
CVE-2022-50116	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: fix deadlock and link starvation in outgoing data path The current implementation queues up new control and user packets as needed and processes this queue down to the ldisc in the same code path. That means that the upper and the lower layer are hard coupled in the code. Due to this deadlocks can happen as seen below while transmitting data, especially during ldisc congestion. Furthermore, the data channels starve the control channel on high transmission load on the ldisc. Introduce an additional control channel data queue to prevent timeouts and link hangups during ldisc congestion. This is being processed before the user channel data queue in gsm_data_kick(), i.e. with the highest priority. Put the queue to ldisc data path into a workqueue and trigger it whenever new data has been put into the transmission queue. Change gsm_dlci_data_sweep() accordingly to fill up the transmission queue until TX_THRESH_HI. This solves the locking issue, keeps latency low and provides good performance on high data load. Note that now all packets from a DLCI are removed from the internal queue if the associated DLCI was closed. This ensures that no data is sent by the introduced write task to an already closed DLCI. BUG: spinlock recursion on CPU#0, test_v24_loop/124 lock: serial8250_ports+0x3a8/0x7500, .magic: dead4ead, .owner: test_v24_loop/124, .owner_cpu: 0 CPU: 0 PID: 124 Comm: test_v24_loop Tainted: G O 5.18.0-rc2 #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Call Trace: <IRQ> dump_stack_lvl+0x34/0x44 do_raw_spin_lock+0x76/0xa0 _raw_spin_lock_irqsave+0x72/0x80 uart_write_room+0x3b/0xc0 gsm_data_kick+0x14b/0x240 [n_gsm] gsmld_write_wakeup+0x35/0x70 [n_gsm] tty_wakeup+0x53/0x60 tty_port_default_wakeup+0x1b/0x30 serial8250_tx_chars+0x12f/0x220 serial8250_handle_irq.part.0+0xfe/0x150 serial8250_default_handle_irq+0x48/0x80 serial8250_interrupt+0x56/0xa0 __handle_irq_event_percpu+0x78/0x1f0 handle_irq_event+0x34/0x70 handle_fasteoi_irq+0x90/0x1e0 __common_interrupt+0x69/0x100 common_interrupt+0x48/0xc0 asm_common_interrupt+0x1e/0x40 RIP: 0010: __do_softirq+0x83/0x34e Code: 2a 0a ff 0f b7 ed c7 44 24 10 0a 00 00 00 48 c7 c7 51 2a 64 82 e8 2d e2 d5 ff 65 66 c7 05 83 af 1e 7e 00 00 fb b8 ff ff ff <49> c7 c2 40 61 80 82 0f bc c5 41 89 c4 41 83 c4 01 0f 84 e6 00 00 RSP: 0018:ffffc90000003f98 EFLAGS: 00000286 RAX: 00000000ffffffff RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffffffff82642a51 RDI: ffffffff825bb5e7 RBP: 0000000000000200 R08: 00000008de3271a8 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000000 R13: 0000000000000030 R14: 0000000000000000 R15: 0000000000000000 ? __do_softirq+0x73/0x34e irq_exit_rcu+0xb5/0x100 common_interrupt+0xa4/0xc0 </IRQ> <TASK> asm_common_interrupt+0x1e/0x40 RIP: 0010: _raw_spin_unlock_irqrestore+0x2e/0x50 Code: 00 55 48 89 fd 48 83 c7 18 53 48 89 f3 48 8b 74 24 10 e8 85 28 36 ff 48 89 ef e8 cd 58 36 ff 80 e7 02 74 01 fb bf 01 00 00 00 <e8> 3d 97 33 ff 65 8b 05 96 23 2b 7e 85 c0 74 03 5b 5d c3 0f 1f 44 RSP: 0018:ffffc9000020fd08 EFLAGS: 00000202 RAX: 0000000000000000 RBX: 0000000000000246 RCX: 0000000000000000 RDX: 0000000000000004 RSI: ffffffff8257fd74 RDI: 0000000000000001 RBP: ffff8880057de3a0 R08: 00000008de233000 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000000 R13: 0000000000000100 R14: 0000000000000202 R15: ffff8880057df0b8 ? _raw_spin_unlock_irqrestore+0x23/0x50 gsmtty_write+0x65/0x80 [n_gsm] n_tty_write+0x33f/0x530 ? swake_up_all+0xe0/0xe0 file_tty_write.constprop.0+0x1b1/0x320 ? n_tty_flush_buffer+0xb0/0xb0 new_sync_write+0x10c/0x190 vfs_write+0x282/0x310 ksys_write+0x68/0xe0 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f3e5e35c15c Code: 8b 7c 24 08 89 c5 e8 c5 ff ff 89 ef 89 44 24 ---truncated---

CVE ID	Description
CVE-2022-50118	In the Linux kernel, the following vulnerability has been resolved: powerpc/perf: Optimize clearing the pending PMI and remove WARN_ON for PMI check in power_pmu_disable commit 2c9ac51b850d ("powerpc/perf: Fix PMU callbacks to clear pending PMI before resetting an overflown PMC") added a new function "pmi_irq_pending" in hw_irq.h. This function is to check if there is a PMI marked as pending in Paca (PACA_IRQ_PMI). This is used in power_pmu_disable in a WARN_ON. The intention here is to provide a warning if there is PMI pending, but no counter is found overflown. During some of the perf runs, below warning is hit: WARNING: CPU: 36 PID: 0 at arch/powerpc/perf/core-book3s.c:1332 power_pmu_disable+0x25c/0x2c0 Modules linked in: ----- NIP [c000000000141c3c] power_pmu_disable+0x25c/0x2c0 LR [c000000000141c8c] power_pmu_disable+0x2ac/0x2c0 Call Trace: [c000000baffcfb90] [c000000000141c8c] power_pmu_disable+0x2ac/0x2c0 (unreliable) [c000000baffcf10] [c0000000003e2f8c] perf_pmu_disable+0x4c/0x60 [c000000baffcf30] [c0000000003e3344] group_sched_out.part.124+0x44/0x100 [c000000baffcf80] [c0000000003e353c] __perf_event_disable+0x13c/0x240 [c000000baffcfcd0] [c0000000003dd334] event_function+0xc4/0x140 [c000000baffcf20] [c0000000003d855c] remote_function+0x7c/0xa0 [c000000baffcf50] [c00000000026c394] flush_smp_call_function_queue+0xd4/0x300 [c000000baffcfde0] [c000000000065b24] smp_ipi_demux_relaxed+0xa4/0x100 [c000000baffcf20] [c000000000cb2b0] xive_muxed_ipi_action+0x20/0x40 [c000000baffcf40] [c000000000207c3c] __handle_irq_event_percpu+0x8c/0x250 [c000000baffcfce0] [c000000000207e2c] handle_irq_event_percpu+0x2c/0xa0 [c000000baffcf10] [c000000000210a04] handle_percpu_irq+0x84/0xc0 [c000000baffcf40] [c000000000205f14] generic_handle_irq+0x54/0x80 [c000000baffcf60] [c000000000015740] __do_irq+0x90/0x1d0 [c000000baffcf90] [c00000000016990] __do_IRQ+0xc0/0x140 [c0000009732f3940] [c000000bafceaca8] 0xc000000bafceaca8 [c0000009732f39d0] [c000000000016b78] do_IRQ+0x168/0x1c0 [c0000009732f3a00] [c0000000000090c8] hardware_interrupt_common_virt+0x218/0x220 This means that there is no PMC overflown among the active events in the PMU, but there is a PMI pending in Paca. The function "any_pmc_overflown" checks the PMCs on active events in cpuhw->n_events. Code snippet: <<>> if (any_pmc_overflown(cpuhw)) clear_pmi_irq_pending(); else WARN_ON(pmi_irq_pending()); <<>> Here the PMC overflown is not from active event. Example: When we do perf record, default cycles and instructions will be running on PMC6 and PMC5 respectively. It could happen that overflowed event is currently not active and pending PMI is for the inactive event. Debug logs from trace_printk: <<>> any_pmc_overflown: idx is 5: pmc value is 0xd9a power_pmu_disable: PMC1: 0x0, PMC2: 0x0, PMC3: 0x0, PMC4: 0x0, PMC5: 0xd9a, PMC6: 0x80002011 <<>> Here active PMC (from idx) is PMC5, but overflown PMC is PMC6(0x80002011). When we handle PMI interrupt for such cases, if the PMC overflown is from inactive event, it will be ignored. Reference commit: commit bc09c219b2e6 ("powerpc/perf: Fix finding overflown PMC in interrupt") Patch addresses two changes: 1) Fix 1: Removal of warning (WARN_ON(pmi_irq_pending());) We were printing warning if no PMC is found overflown among active PMU events, but PMI pending in PACA. But this could happen in cases where PMC overflown is not in active PMC. An inactive event could have caused the overflow. Hence the warning is not needed. To know pending PMI is from an inactive event, we need to loop through all PMC's which will cause more SPR reads via mfspr and increase in context switch. Also in existing function: perf_event_interrupt, already we ignore PMI's overflown when it is from an inactive PMC. 2) Fix 2: optimization in clearing pending PMI. Currently we check for any active PMC overflown before clearing PMI pending in Paca. This is causing additional SP ---truncated---
CVE-2022-50123	In the Linux kernel, the following vulnerability has been resolved: ASoC: mediatek: mt8173: Fix refcount leak in mt8173_rt5650_rt5676_dev_probe of_parse_phandle() returns a node pointer with refcount incremented, we should use of_node_put() on it when not need anymore. Fix missing of_node_put() in error paths.
CVE-2022-50129	In the Linux kernel, the following vulnerability has been resolved: RDMA/srpt: Fix a use-after-free Change the LIO port members inside struct srpt_port from regular members into pointers. Allocate the LIO port data structures from inside srpt_make_tport() and free these from inside srpt_make_tport(). Keep struct srpt_device as long as either an RDMA port or a LIO target port is associated with it. This patch decouples the lifetime of struct srpt_port (controlled by the RDMA core) and struct srpt_port_id (controlled by LIO). This patch fixes the following KASAN complaint: BUG: KASAN: use-after-free in srpt_enable_tpg+0x31/0x70 [ib_srpt] Read of size 8 at addr ffff888141cc34b8 by task check/5093 Call Trace: <TASK> show_stack+0x4e/0x53 dump_stack_lvl+0x51/0x66 print_address_description.constprop.0.cold+0xea/0x41e print_report.cold+0x90/0x205 kasan_report+0xb9/0xf0 __asan_load8+0x69/0x90 srpt_enable_tpg+0x31/0x70 [ib_srpt] target_fabric_tpg_base_enable_store+0xe2/0x140 [target_core_mod] configfs_write_iter+0x18b/0x210 new_sync_write+0x1f2/0x2f0 vfs_write+0x3e3/0x540 ksys_write+0xbb/0x140 __x64_sys_write+0x42/0x50 do_syscall_64+0x34/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 </TASK>
CVE-2022-50134	In the Linux kernel, the following vulnerability has been resolved: RDMA/hfi1: fix potential memory leak in setup_base_ctxt() setup_base_ctxt() allocates a memory chunk for uctxt->groups with hfi1_alloc_ctxt_rcv_groups(). When init_user_ctxt() fails, uctxt->groups is not released, which will lead to a memory leak. We should release the uctxt->groups with hfi1_free_ctxt_rcv_groups() when init_user_ctxt() fails.

CVE ID	Description
CVE-2022-50137	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix a window for use-after-free During a destroy CQ an interrupt may cause processing of a CQE after CQ resources are freed by irdma_cq_free_rsrc(). Fix this by moving the call to irdma_cq_free_rsrc() after the irdma_sc_cleanup_ceqes(), which is called under the cq_lock.
CVE-2022-50144	In the Linux kernel, the following vulnerability has been resolved: soundwire: revisit driver bind/unbind and callbacks In the SoundWire probe, we store a pointer from the driver ops into the 'slave' structure. This can lead to kernel oopses when unbinding codec drivers, e.g. with the following sequence to remove machine driver and codec driver. /sbin/modprobe -r snd_soc_sof_sdw /sbin/modprobe -r snd_soc_rt711 The full details can be found in the BugLink below, for reference the two following examples show different cases of driver ops/callbacks being invoked after the driver .remove(). kernel: BUG: kernel NULL pointer dereference, address: 0000000000000150 kernel: Workqueue: events cdns_update_slave_status_work [soundwire_cadence] kernel: RIP: 0010:mux_lock+0x19/0x30 kernel: Call Trace: kernel: ? sdw_handle_slave_status+0x426/0xe00 [soundwire_bus 94ff184bf398570c3f8ff7efe9e32529f532e4ae] kernel: ? newidle_balance+0x26a/0x400 kernel: ? cdns_update_slave_status_work+0x1e9/0x200 [soundwire_cadence 1bcf98eebe5ba9833cd433323769ac923c9c6f82] kernel: BUG: unable to handle page fault for address: ffffffff07654c8 kernel: Workqueue: pm pm_runtime_work kernel: RIP: 0010:sdw_bus_prep_clk_stop+0x6f/0x160 [soundwire_bus] kernel: Call Trace: kernel: <TASK> kernel: sdw_cdns_clock_stop+0xb5/0x1b0 [soundwire_cadence 1bcf98eebe5ba9833cd433323769ac923c9c6f82] kernel: intel_suspend_runtime+0x5f/0x120 [soundwire_intel aca858f7c87048d3152a4a41bb68abb9b663a1dd] kernel: ? dpm_sysfs_remove+0x60/0x60 This was not detected earlier in Intel tests since the tests first remove the parent PCI device and shut down the bus. The sequence above is a corner case which keeps the bus operational but without a driver bound. While trying to solve this kernel oopses, it became clear that the existing SoundWire bus does not deal well with the unbind case. Commit 528be501b7d4a ("soundwire: sdw_slave: add probe_complete structure and new fields") added a 'probed' status variable and a 'probe_complete' struct completion. This status is however not reset on remove and likewise the 'probe complete' is not re-initialized, so the bind/unbind/bind test cases would fail. The timeout used before the 'update_status' callback was also a bad idea in hindsight, there should really be no timing assumption as to if and when a driver is bound to a device. An initial draft was based on device_lock() and device_unlock() was tested. This proved too complicated, with deadlocks created during the suspend-resume sequences, which also use the same device_lock/unlock() as the bind/unbind sequences. On a CometLake device, a bad DSDT/BIOS caused spurious resumes and the use of device_lock() caused hangs during suspend. After multiple weeks or testing and painful reverse-engineering of deadlocks on different devices, we looked for alternatives that did not interfere with the device core. A bus notifier was used successfully to keep track of DRIVER_BOUND and DRIVER_UNBIND events. This solved the bind-unbind-bind case in tests, but it can still be defeated with a theoretical corner case where the memory is freed by a .remove while the callback is in use. The notifier only helps make sure the driver callbacks are valid, but not that the memory allocated in probe remains valid while the callbacks are invoked. This patch suggests the introduction of a new 'sdw_dev_lock' mutex protecting probe/remove and all driver callbacks. Since this mutex is 'local' to SoundWire only, it does not interfere with existing locks and does not create deadlocks. In addition, this patch removes the 'probe_complete' completion, instead we directly invoke the 'update_status' from the probe routine. That removes any sort of timing dependency and a much better support for the device/driver model, the driver could be bound before the bus started, or eons after the bus started and the hardware would be properly initialized in all cases. BugLink: https://github.com/thesofproject/linux/is ---truncated---
CVE-2022-50159	In the Linux kernel, the following vulnerability has been resolved: of: check previous kernel's ima-kexec-buffer against memory bounds Presently ima_get_kexec_buffer() doesn't check if the previous kernel's ima-kexec-buffer lies outside the addressable memory range. This can result in a kernel panic if the new kernel is booted with 'mem=X' arg and the ima-kexec-buffer was allocated beyond that range by the previous kernel. The panic is usually of the form below: \$ sudo kexec --initrd initrd vmlinux --append='mem=16G' <snip> BUG: Unable to handle kernel data access on read at 0xc000c01fff7f0000 Faulting instruction address: 0xc000000000837974 Oops: Kernel access of bad area, sig: 11 [#1] <snip> NIP [c000000000837974] ima_restore_measurement_list+0x94/0x6c0 LR [c00000000083b55c] ima_load_kexec_buffer+0xac/0x160 Call Trace: [c00000000371fa80] [c00000000083b55c] ima_load_kexec_buffer+0xac/0x160 [c00000000371fb00] [c0000000020512c4] ima_init+0x80/0x108 [c00000000371fb70] [c0000000020514dc] init_ima+0x4c/0x120 [c00000000371fbf0] [c000000000012240] do_one_initcall+0x60/0x2c0 [c00000000371fcc0] [c000000002004ad0] kernel_init_freeable+0x344/0x3ec [c00000000371fda0] [c0000000000128a4] kernel_init+0x34/0x1b0 [c00000000371fe10] [c00000000000ce64] ret_from_kernel_thread+0x5c/0x64 Instruction dump: f92100b8 f92100c0 90e10090 910100a0 4182050c 282a0017 3bc00000 40810330 7c0802a6 fb610198 7c9b2378 f80101d0 <a1240000> 2c090001 40820614 e9240010 ---[end trace 0000000000000000]--- Fix this issue by checking returned PFN range of previous kernel's ima-kexec-buffer with page_is_ram() to ensure correct memory bounds.

CVE ID	Description
CVE-2022-50166	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: When HCI work queue is drained, only queue chained work The HCI command, event, and data packet processing workqueue is drained to avoid deadlock in commit 76727c02c1e1 ("Bluetooth: Call drain_workqueue() before resetting state"). There is another delayed work, which will queue command to this drained workqueue. Which results in the following error report: Bluetooth: hci2: command 0x040f tx timeout WARNING: CPU: 1 PID: 18374 at kernel/workqueue.c:1438 __queue_work+0xdad/0x1140 Workqueue: events hci_cmd_timeout RIP: 0010:__queue_work+0xdad/0x1140 RSP: 0000:ffffc90002cffc60 EFLAGS: 00010093 RAX: 0000000000000000 RBX: ffff8880b9d3ec00 RCX: 0000000000000000 RDX: ffff888024ba0000 RSI: ffffffff814e048d RDI: ffff8880b9d3ec08 RBP: 0000000000000008 R08: 0000000000000000 R09: 00000000b9d39700 R10: ffffffff814f73c6 R11: 0000000000000000 R12: ffff88807cce4c60 R13: 0000000000000000 R14: ffff8880796d8800 R15: ffff8880796d8800 FS: 0000000000000000(0000) GS:ffff8880b9d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000c0174b4000 CR3: 000000007cae9000 CR4: 00000000003506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? queue_work_on+0xcb/0x110 ? lockdep_hardirqs_off+0x90/0xd0 queue_work_on+0xee/0x110 process_one_work+0x996/0x1610 ? pwq_dec_nr_in_flight+0x2a0/0x2a0 ? rwlock_bug.part.0+0x90/0x90 ? _raw_spin_lock_irq+0x41/0x50 worker_thread+0x665/0x1080 ? process_one_work+0x1610/0x1610 kthread+0x2e9/0x3a0 ? kthread_complete_and_exit+0x40/0x40 ret_from_fork+0x1f/0x30 </TASK> To fix this, we can add a new HCI_DRAIN_WQ flag, and don't queue the timeout workqueue while command workqueue is draining.
CVE-2022-50167	In the Linux kernel, the following vulnerability has been resolved: bpf: fix potential 32-bit overflow when accessing ARRAY map element If BPF array map is bigger than 4GB, element pointer calculation can overflow because both index and elem_size are u32. Fix this everywhere by forcing 64-bit multiplication. Extract this formula into separate small helper and use it consistently in various places. Speculative-preventing formula utilizing index_mask trick is left as is, but explicit u64 casts are added in both places.
CVE-2022-50168	In the Linux kernel, the following vulnerability has been resolved: bpf, x86: fix freeing of not-finalized bpf_prog_pack syzbot reported a few issues with bpf_prog_pack [1], [2]. This only happens with multiple subprogs. In jit_subprogs(), we first call bpf_int_jit_compile() on each sub program. And then, we call it on each sub program again. jit_data is not freed in the first call of bpf_int_jit_compile(). Similarly we don't call bpf_jit_binary_pack_finalize() in the first call of bpf_int_jit_compile(). If bpf_int_jit_compile() failed for one sub program, we will call bpf_jit_binary_pack_finalize() for this sub program. However, we don't have a chance to call it for other sub programs. Then we will hit "goto out_free" in jit_subprogs(), and call bpf_jit_free on some subprograms that haven't got bpf_jit_binary_pack_finalize() yet. At this point, bpf_jit_binary_pack_free() is called and the whole 2MB page is freed erroneously. Fix this with a custom bpf_jit_free() for x86_64, which calls bpf_jit_binary_pack_finalize() if necessary. Also, with custom bpf_jit_free(), bpf_prog_aux->use_bpf_prog_pack is not needed any more, remove it. [1] https://syzkaller.appspot.com/bug?extid=2f649ec6d2eea1495a8f [2] https://syzkaller.appspot.com/bug?extid=87f65c75f4a72db05445
CVE-2022-50170	In the Linux kernel, the following vulnerability has been resolved: kunit: executor: Fix a memory leak on failure in kunit_filter_tests It's possible that memory allocation for 'filtered' will fail, but for the copy of the suite to succeed. In this case, the copy could be leaked. Properly free 'copy' in the error case for the allocation of 'filtered' failing. Note that there may also have been a similar issue in kunit_filter_subsuites, before it was removed in "kunit: flatten kunit_suite*** to kunit_suite** in .kunit_test_suites". This was reported by clang-analyzer via the kernel test robot, here: https://lore.kernel.org/all/c8073b8e-7b9e-0830-4177-87c12f16349c@intel.com/ And by smatch via Dan Carpenter and the kernel test robot: https://lore.kernel.org/all/202207101328.ASjx88yj-lkp@intel.com/

CVE ID	Description
CVE-2022-50174	In the Linux kernel, the following vulnerability has been resolved: net: hinic: avoid kernel hung in hinic_get_stats64() When using hinic device as a bond slave device, and reading device stats of master bond device, the kernel may hung. The kernel panic calltrace as follows: Kernel panic - not syncing: softlockup: hung tasks Call trace: native_queued_spin_lock_slowpath+0x1ec/0x31c dev_get_stats+0x60/0xcc dev_seq_printf_stats+0x40/0x120 dev_seq_show+0x1c/0x40 seq_read_iter+0x3c8/0x4dc seq_read+0xe0/0x130 proc_reg_read+0xa8/0xe0 vfs_read+0xb0/0x1d4 ksys_read+0x70/0xfc __arm64_sys_read+0x20/0x30 el0_svc_common+0x88/0x234 do_el0_svc+0x2c/0x90 el0_svc+0x1c/0x30 el0_sync_handler+0xa8/0xb0 el0_sync+0x148/0x180 And the calltrace of task that actually caused kernel hungs as follows: __switch_to+124 __schedule+548 schedule+72 schedule_timeout+348 __down_common+188 __down+24 down+104 hinic_get_stats64+44 [hinic] dev_get_stats+92 bond_get_stats+172 [bonding] dev_get_stats+92 dev_seq_printf_stats+60 dev_seq_show+24 seq_read_iter+964 seq_read+220 proc_reg_read+164 vfs_read+172 ksys_read+108 __arm64_sys_read+28 el0_svc_common+132 do_el0_svc+40 el0_svc+24 el0_sync_handler+164 el0_sync+324 When getting device stats from bond, kernel will call bond_get_stats(). It first holds the spinlock bond->stats_lock, and then call hinic_get_stats64() to collect hinic device's stats. However, hinic_get_stats64() calls `down(&nic_dev->mgmt_lock)` to protect its critical section, which may schedule current task out. And if system is under high pressure, the task cannot be woken up immediately, which eventually triggers kernel hung panic. Since previous patch has replaced hinic_dev.tx_stats/rx_stats with local variable in hinic_get_stats64(), there is nothing need to be protected by lock, so just removing down()/up() is ok.
CVE-2022-50176	In the Linux kernel, the following vulnerability has been resolved: drm/mcde: Fix refcount leak in mcde_dsi_bind Every iteration of for_each_available_child_of_node() decrements the reference counter of the previous node. There is no decrement when break out from the loop and results in refcount leak. Add missing of_node_put() to fix this.
CVE-2022-50179	In the Linux kernel, the following vulnerability has been resolved: ath9k: fix use-after-free in ath9k_hif_usb_rx_cb Syzbot reported use-after-free Read in ath9k_hif_usb_rx_cb() [0]. The problem was in incorrect htc_handle->drv_priv initialization. Probable call trace which can trigger use-after-free: ath9k_htc_probe_device() /* htc_handle->drv_priv = priv; */ ath9k_htc_wait_for_target() <--- Failed ieee80211_free_hw() <--- priv pointer is freed <IRQ> ... ath9k_hif_usb_rx_cb() ath9k_hif_usb_rx_stream() RX_STAT_INC() <--- htc_handle->drv_priv access In order to not add fancy protection for drv_priv we can move htc_handle->drv_priv initialization at the end of the ath9k_htc_probe_device() and add helper macro to make all *_STAT_* macros NULL safe, since syzbot has reported related NULL deref in that macros [1]
CVE-2022-50185	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix potential buffer overflow in ni_set_mc_special_registers() The last case label can write two buffers 'mc_reg_address[j]' and 'mc_data[j]' with 'j' offset equal to SMC_NISLANDS_MC_REGISTER_ARRAY_SIZE since there are no checks for this value in both case labels after the last 'j++'. Instead of changing '>' to '>=' there, add the bounds check at the start of the second 'case' (the first one already has it). Also, remove redundant last checks for 'j' index bigger than array size. The expression is always false. Moreover, before or after the patch 'table->last' can be equal to SMC_NISLANDS_MC_REGISTER_ARRAY_SIZE and it seems it can be a valid value. Detected using the static analysis tool - Svace.
CVE-2022-50186	In the Linux kernel, the following vulnerability has been resolved: ath11k: fix missing skb drop on htc_tx_completion error On htc_tx_completion error the skb is not dropped. This is wrong since the completion_handler logic expect the skb to be consumed anyway even when an error is triggered. Not freeing the skb on error is a memory leak since the skb won't be freed anywhere else. Correctly free the packet on eid >= ATH11K_HTC_EP_COUNT before returning. Tested-on: IPQ8074 hw2.0 AHB WLAN.HK.2.5.0.1-01208-QCAHKSUPL_SILICONZ-1
CVE-2022-50197	In the Linux kernel, the following vulnerability has been resolved: cpufreq: zynq: Fix refcount leak in zynq_get_revision_of_find_compatible_node() returns a node pointer with refcount incremented, we should use of_node_put() on it when done. Add missing of_node_put() to avoid refcount leak.
CVE-2022-50201	In the Linux kernel, the following vulnerability has been resolved: selinux: fix memleak in security_read_kernel() In this function, it directly returns the result of __security_read_policy without freeing the allocated memory in *data, cause memory leak issue, so free the memory if __security_read_policy failed. [PM: subject line tweak]

CVE ID	Description
CVE-2022-50202	In the Linux kernel, the following vulnerability has been resolved: PM: hibernate: defer device probing when resuming from hibernation syzbot is reporting hung task at misc_open() [1], for there is a race window of AB-BA deadlock which involves probe_count variable. Currently wait_for_device_probe() from snapshot_open() from misc_open() can sleep forever with misc_mtx held if probe_count cannot become 0. When a device is probed by hub_event() work function, probe_count is incremented before the probe function starts, and probe_count is decremented after the probe function completed. There are three cases that can prevent probe_count from dropping to 0. (a) A device being probed stopped responding (i.e. broken/malicious hardware). (b) A process emulating a USB device using /dev/raw-gadget interface stopped responding for some reason. (c) New device probe requests keeps coming in before existing device probe requests complete. The phenomenon syzbot is reporting is (b). A process which is holding system_transition_mutex and misc_mtx is waiting for probe_count to become 0 inside wait_for_device_probe(), but the probe function which is called from hub_event() work function is waiting for the processes which are blocked at mutex_lock(&misc_mtx) to respond via /dev/raw-gadget interface. This patch mitigates (b) by deferring wait_for_device_probe() from snapshot_open() to snapshot_write() and snapshot_ioctl(). Please note that the possibility of (b) remains as long as any thread which is emulating a USB device via /dev/raw-gadget interface can be blocked by uninterruptible blocking operations (e.g. mutex_lock()). Please also note that (a) and (c) are not addressed. Regarding (c), we should change the code to wait for only one device which contains the image for resuming from hibernation. I don't know how to address (a), for use of timeout for wait_for_device_probe() might result in loss of user data in the image. Maybe we should require the userland to wait for the image device before opening /dev/snapshot interface.
CVE-2022-50203	In the Linux kernel, the following vulnerability has been resolved: ARM: OMAP2+: display: Fix refcount leak bug In omapdss_init_fbdev(), of_find_node_by_name() will return a node pointer with refcount incremented. We should use of_node_put() when it is not used anymore.
CVE-2022-50213	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: do not allow SET_ID to refer to another table When doing lookups for sets on the same batch by using its ID, a set from a different table can be used. Then, when the table is removed, a reference to the set may be kept after the set is freed, leading to a potential use-after-free. When looking for sets by ID, use the table that was used for the lookup by name, and only return sets belonging to that same table. This fixes CVE-2022-2586, also reported as ZDI-CAN-17470.
CVE-2022-50215	In the Linux kernel, the following vulnerability has been resolved: scsi: sg: Allow waiting for commands to complete on removed device When a SCSI device is removed while in active use, currently sg will immediately return -ENODEV on any attempt to wait for active commands that were sent before the removal. This is problematic for commands that use SG_FLAG_DIRECT_IO since the data buffer may still be in use by the kernel when userspace frees or reuses it after getting ENODEV, leading to corrupted userspace memory (in the case of READ-type commands) or corrupted data being sent to the device (in the case of WRITE-type commands). This has been seen in practice when logging out of a iscsi_tcp session, where the iSCSI driver may still be processing commands after the device has been marked for removal. Change the policy to allow userspace to wait for active sg commands even when the device is being removed. Return -ENODEV only when there are no more responses to read.
CVE-2022-50218	In the Linux kernel, the following vulnerability has been resolved: iio: light: isl29028: Fix the warning in isl29028_remove() The driver use the non-managed form of the register function in isl29028_remove(). To keep the release order as mirroring the ordering in probe, the driver should use non-managed form in probe, too. The following log reveals it: [32.374955] isl29028 0-0010: remove [32.376861] general protection fault, probably for non-canonical address 0xdffffc0000000006: 0000 [#1] PREEMPT SMP KASAN PTI [32.377676] KASAN: null-ptr-deref in range [0x0000000000000030-0x0000000000000037] [32.379432] RIP: 0010:kernfs_find_and_get_ns +0x28/0xe0 [32.385461] Call Trace: [32.385807] sysfs_unmerge_group+0x59/0x110 [32.386110] dpm_sysfs_remove+0x58/0xc0 [32.386391] device_del+0x296/0xe50 [32.386959] cdev_device_del +0x1d/0xd0 [32.387231] devm_iio_device_unreg+0x27/0xb0 [32.387542] devres_release_group +0x319/0x3d0 [32.388162] i2c_device_remove+0x93/0x1f0
CVE-2022-50219	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix KASAN use-after-free Read in compute_effective_progs Syzbot found a Use After Free bug in compute_effective_progs(). The reproducer creates a number of BPF links, and causes a fault injected alloc to fail, while calling bpf_link_detach on them. Link detach triggers the link to be freed by bpf_link_free(), which calls __cgroup_bpf_detach() and update_effective_progs(). If the memory allocation in this function fails, the function restores the pointer to the bpf_cgroup_link on the cgroup list, but the memory gets freed just after it returns. After this, every subsequent call to update_effective_progs() causes this already deallocated pointer to be dereferenced in prog_list_length(), and triggers KASAN UAF error. To fix this issue don't preserve the pointer to the prog or link in the list, but remove it and replace it with a dummy prog without shrinking the table. The subsequent call to __cgroup_bpf_detach() or __cgroup_bpf_detach() will correct it.

CVE ID	Description
CVE-2022-50220	In the Linux kernel, the following vulnerability has been resolved: usbnet: Fix linkwatch use-after-free on disconnect usbnet uses the work usbnet_deferred_kevent() to perform tasks which may sleep. On disconnect, completion of the work was originally awaited in ->ndo_stop(). But in 2003, that was moved to ->disconnect() by historic commit "[PATCH] USB: usbnet, prevent exotic rtnl deadlock": https://git.kernel.org/tglx/history/c/0f138bbfd83c The change was made because back then, the kernel's workqueue implementation did not allow waiting for a single work. One had to wait for completion of *all* work by calling flush_scheduled_work(), and that could deadlock when waiting for usbnet_deferred_kevent() with rtnl_mutex held in ->ndo_stop(). The commit solved one problem but created another: It causes a use-after-free in USB Ethernet drivers aqc111.c, asix_devices.c, ax88179_178a.c, ch9200.c and smsc75xx.c: * If the drivers receive a link change interrupt immediately before disconnect, they raise EVENT_LINK_RESET in their (non-sleepable) ->status() callback and schedule usbnet_deferred_kevent(). * usbnet_deferred_kevent() invokes the driver's ->link_reset() callback, which calls netif_carrier_{on,off}(). * That in turn schedules the work linkwatch_event(). Because usbnet_deferred_kevent() is awaited after unregister_netdev(), netif_carrier_{on,off}() may operate on an unregistered netdev and linkwatch_event() may run after free_netdev(), causing a use-after-free. In 2010, usbnet was changed to only wait for a single instance of usbnet_deferred_kevent() instead of *all* work by commit 23f333a2bfaf ("drivers/net: don't use flush_scheduled_work()"). Unfortunately the commit neglected to move the wait back to ->ndo_stop(). Rectify that omission at long last.
CVE-2022-50222	In the Linux kernel, the following vulnerability has been resolved: tty: vt: initialize unicode screen buffer syzbot reports kernel infoleak at vcs_read() [1], for buffer can be read immediately after resize operation. Initialize buffer using kzalloc(). ----- #include <fcntl.h> #include <unistd.h> #include <sys/ioctl.h> #include <linux/fb.h> int main(int argc, char *argv[]) { struct fb_var_screeninfo var = { }; const int fb_fd = open("/dev/fb0", 3); ioctl(fb_fd, FBIOPUT_VSCREENINFO, &var); var.yres = 0x21; ioctl(fb_fd, FBIOPUT_VSCREENINFO, &var); return read(open("/dev/vcsu", O_RDONLY), &var, sizeof(var)) == -1; } -----
CVE-2022-50224	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/mmu: Treat NX as a valid SPTE bit for NPT Treat the NX bit as valid when using NPT, as KVM will set the NX bit when the NX huge page mitigation is enabled (mindblowing) and trigger the WARN that fires on reserved SPTE bits being set. KVM has required NX support for SVM since commit b26a71a1a5b9 ("KVM: SVM: Refuse to load kvm_amd if NX support is not available") for exactly this reason, but apparently it never occurred to anyone to actually test NPT with the mitigation enabled. ----- [cut here]----- spte = 0x800000018a600ee7, level = 2, rsvd bits = 0x800f0000001fe000 WARNING: CPU: 152 PID: 15966 at arch/x86/kvm/mmu/spte.c:215 make_spte+0x327/0x340 [kvm] Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 10.48.0 01/27/2022 RIP: 0010:make_spte+0x327/0x340 [kvm] Call Trace: <TASK> tdp_mmu_map_handle_target_level +0xc3/0x230 [kvm] kvm_tdp_mmu_map+0x343/0x3b0 [kvm] direct_page_fault+0x1ae/0x2a0 [kvm] kvm_tdp_page_fault+0x7d/0x90 [kvm] kvm_mmu_page_fault+0xfb/0x2e0 [kvm] npf_interception +0x55/0x90 [kvm_amd] svm_invoke_exit_handler+0x31/0xf0 [kvm_amd] svm_handle_exit +0xf6/0x1d0 [kvm_amd] vcpu_enter_guest+0xb6d/0xee0 [kvm] ? kvm_pmu_trigger_event+0x6d/0x230 [kvm] vcpu_run+0x65/0x2c0 [kvm] kvm_arch_vcpu_ioctl_run+0x355/0x610 [kvm] kvm_vcpu_ioctl +0x551/0x610 [kvm] __se_sys_ioctl+0x77/0xc0 __x64_sys_ioctl+0x1d/0x20 do_syscall_64+0xa4/0xa0 entry_SYSCALL_64_after_hwframe+0x46/0xb0 </TASK> ---[end trace 0000000000000000]---
CVE-2022-50226	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp - Use kzalloc for sev ioctl interfaces to prevent kernel memory leak For some sev ioctl interfaces, input may be passed that is less than or equal to SEV_FW_BLOB_MAX_SIZE, but larger than the data that PSP firmware returns. In this case, kmalloc will allocate memory that is the size of the input rather than the size of the data. Since PSP firmware doesn't fully overwrite the buffer, the sev ioctl interfaces with the issue may return uninitialized slab memory. Currently, all of the ioctl interfaces in the ccp driver are safe, but to prevent future problems, change all ioctl interfaces that allocate memory with kmalloc to use kzalloc and memset the data buffer to zero in sev_ioctl_do_platform_status.
CVE-2022-50229	In the Linux kernel, the following vulnerability has been resolved: ALSA: bcd2000: Fix a UAF bug on the error path of probing When the driver fails in snd_card_register() at probe time, it will free the 'bcd2k->midi_out_urb' before killing it, which may cause a UAF bug. The following log can reveal it: [50.727020] BUG: KASAN: use-after-free in bcd2000_input_complete+0x1f1/0x2e0 [snd_bcd2000] [50.727623] Read of size 8 at addr ffff88810fab0e88 by task swapper/4/0 [50.729530] Call Trace: [50.732899] bcd2000_input_complete+0x1f1/0x2e0 [snd_bcd2000] Fix this by adding usb_kill_urb() before usb_free_urb().

CVE ID	Description
CVE-2022-50230	In the Linux kernel, the following vulnerability has been resolved: arm64: set UXN on swapper page tables [This issue was fixed upstream by accident in c3cee924bd85 ("arm64: head: cover entire kernel image in initial ID map") as part of a large refactoring of the arm64 boot flow. This simple fix is therefore preferred for -stable backporting] On a system that implements FEAT_EPAN, read/write access to the idmap is denied because UXN is not set on the swapper PTEs. As a result, idmap_kpti_install_ng_mappings panics the kernel when accessing __idmap_kpti_flag. Fix it by setting UXN on these PTEs.
CVE-2022-50232	In the Linux kernel, the following vulnerability has been resolved: arm64: set UXN on swapper page tables [This issue was fixed upstream by accident in c3cee924bd85 ("arm64: head: cover entire kernel image in initial ID map") as part of a large refactoring of the arm64 boot flow. This simple fix is therefore preferred for -stable backporting] On a system that implements FEAT_EPAN, read/write access to the idmap is denied because UXN is not set on the swapper PTEs. As a result, idmap_kpti_install_ng_mappings panics the kernel when accessing __idmap_kpti_flag. Fix it by setting UXN on these PTEs.
CVE-2022-50233	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: eir: Fix using strlen with hdev->{dev_name,short_name} Both dev_name and short_name are not guaranteed to be NULL terminated so this instead use strlen and then attempt to determine if the resulting string needs to be truncated or not.
CVE-2022-50234	In the Linux kernel, the following vulnerability has been resolved: io_uring/af_unix: defer registered files gc to io_uring release Instead of putting io_uring's registered files in unix_gc() we want it to be done by io_uring itself. The trick here is to consider io_uring registered files for cycle detection but not actually putting them down. Because io_uring can't register other ring instances, this will remove all refs to the ring file triggering the ->release path and clean up with io_ring_ctx_free(). [axboe: add kerneldoc comment to skb, fold in skb leak fix]
CVE-2022-50240	In the Linux kernel, the following vulnerability has been resolved: binder: fix UAF of alloc->vma in race with munmap() In commit 720c24192404 ("ANDROID: binder: change down_write to down_read") binder assumed the mmap read lock is sufficient to protect alloc->vma inside binder_update_page_range(). This used to be accurate until commit dd2283f2605e ("mm: mmap: zap pages with read mmap_sem in munmap"), which now downgrades the mmap_lock after detaching the vma from the rbtree in munmap(). Then it proceeds to teardown and free the vma with only the read lock held. This means that accesses to alloc->vma in binder_update_page_range() now will race with vm_area_free() in munmap() and can cause a UAF as shown in the following KASAN trace: <pre> ===== BUG: KASAN: use-after-free in vm_insert_page+0x7c/0x1f0 Read of size 8 at addr ffff16204ad00600 by task server/558 CPU: 3 PID: 558 Comm: server Not tainted 5.10.150-00001-gdc8dcf942daa #1 Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0x0/0x2a0 show_stack +0x18/0x2c dump_stack+0xf8/0x164 print_address_description.constprop.0+0x9c/0x538 kasan_report +0x120/0x200 __asan_load8+0xa0/0xc4 vm_insert_page+0x7c/0x1f0 binder_update_page_range +0x278/0x50c binder_alloc_new_buf+0x3f0/0xba0 binder_transaction+0x64c/0x3040 binder_thread_write+0x924/0x2020 binder_ioctl+0x1610/0x2e5c __arm64_sys_ioctl+0xd4/0x120 el0_svc_common.constprop.0+0xac/0x270 do_el0_svc+0x38/0xa0 el0_svc+0x1c/0x2c el0_sync_handler +0xe8/0x114 el0_sync+0x180/0x1c0 Allocated by task 559: kasan_save_stack+0x38/0x6c __kasan_kmalloc.constprop.0+0xe4/0xf0 kasan_slab_alloc+0x18/0x2c kmem_cache_alloc +0x1b0/0x2d0 vm_area_alloc+0x28/0x94 mmap_region+0x378/0x920 do_mmap+0x3f0/0x600 vm_mmap_pgoff+0x150/0x17c ksys_mmap_pgoff+0x284/0x2dc __arm64_sys_mmap+0x84/0xa4 el0_svc_common.constprop.0+0xac/0x270 do_el0_svc+0x38/0xa0 el0_svc+0x1c/0x2c el0_sync_handler+0xe8/0x114 el0_sync+0x180/0x1c0 Freed by task 560: kasan_save_stack +0x38/0x6c kasan_set_track+0x28/0x40 kasan_set_free_info+0x24/0x4c __kasan_slab_free +0x100/0x164 kasan_slab_free+0x14/0x20 kmem_cache_free+0xc4/0x34c vm_area_free +0x1c/0x2c remove_vma+0x7c/0x94 __do_munmap+0x358/0x710 __vm_munmap+0xbc/0x130 __arm64_sys_munmap+0x4c/0x64 el0_svc_common.constprop.0+0xac/0x270 do_el0_svc +0x38/0xa0 el0_svc+0x1c/0x2c el0_sync_handler+0xe8/0x114 el0_sync+0x180/0x1c0 [...] ===== To prevent </pre> the race above, revert back to taking the mmap write lock inside binder_update_page_range(). One might expect an increase of mmap lock contention. However, binder already serializes these calls via top level alloc->mutex. Also, there was no performance impact shown when running the binder benchmark tests. Note this patch is specific to stable branches 5.4 and 5.10. Since in newer kernel releases binder no longer caches a pointer to the vma. Instead, it has been refactored to use vma_lookup() which avoids the issue described here. This switch was introduced in commit a43cfc87caaf ("android: binder: stop saving a pointer to the VMA").

CVE ID	Description
CVE-2022-50241	In the Linux kernel, the following vulnerability has been resolved: NFSD: fix use-after-free on source server when doing inter-server copy Use-after-free occurred when the laundromat tried to free expired cpntf_state entry on the s2s_cp_stateids list after inter-server copy completed. The sc_cp_list that the expired copy state was inserted on was already freed. When COPY completes, the Linux client normally sends LOCKU(lock_state x), FREE_STATEID(lock_state x) and CLOSE(open_state y) to the source server. The nfs4_put_stid call from nfsd4_free_stateid cleans up the copy state from the s2s_cp_stateids list before freeing the lock state's stid. However, sometimes the CLOSE was sent before the FREE_STATEID request. When this happens, the nfsd4_close_open_stateid call from nfsd4_close frees all lock states on its st_locks list without cleaning up the copy state on the sc_cp_list list. When the time the FREE_STATEID arrives the server returns BAD_STATEID since the lock state was freed. This causes the use-after-free error to occur when the laundromat tries to free the expired cpntf_state. This patch adds a call to nfs4_free_cpntf_statelist in nfsd4_close_open_stateid to clean up the copy state before calling free_ol_stateid_reaplist to free the lock state's stid on the reaplist.
CVE-2022-50244	In the Linux kernel, the following vulnerability has been resolved: cxl: fix possible null-ptr-deref in cxl_pci_init_afu[adapter]() If device_register() fails in cxl_pci_afu[adapter>(), the device is not added, device_unregister() can not be called in the error path, otherwise it will cause a null-ptr-deref because of removing not added device. As comment of device_register() says, it should use put_device() to give up the reference in the error path. So split device_unregister() into device_del() and put_device(), then goes to put dev when register fails.
CVE-2022-50247	In the Linux kernel, the following vulnerability has been resolved: usb: xhci-mtk: fix leakage of shared hcd when fail to set wakeup irq Can not set the @shared_hcd to NULL before decrease the usage count by usb_put_hcd(), this will cause the shared hcd not released.
CVE-2022-50252	In the Linux kernel, the following vulnerability has been resolved: igb: Do not free q_vector unless new one was allocated Avoid potential use-after-free condition under memory pressure. If the kzalloc() fails, q_vector will be freed but left in the original adapter->q_vector[v_idx] array position.
CVE-2022-50253	In the Linux kernel, the following vulnerability has been resolved: bpf: make sure skb->len != 0 when redirecting to a tunneling device syzkaller managed to trigger another case where skb->len == 0 when we enter __dev_queue_xmit: WARNING: CPU: 0 PID: 2470 at include/linux/skbuff.h:2576 skb_assert_len include/linux/skbuff.h:2576 [inline] WARNING: CPU: 0 PID: 2470 at include/linux/skbuff.h:2576 __dev_queue_xmit+0x2069/0x35e0 net/core/dev.c:4295 Call Trace: dev_queue_xmit+0x17/0x20 net/core/dev.c:4406 __bpf_tx_skb net/core/filter.c:2115 [inline] __bpf_redirect_no_mac net/core/filter.c:2140 [inline] __bpf_redirect+0x5fb/0xda0 net/core/filter.c:2163 ____bpf_clone_redirect net/core/filter.c:2447 [inline] bpf_clone_redirect+0x247/0x390 net/core/filter.c:2419 bpf_prog_48159a89cb4a9a16+0x59/0x5e bpf_dispatcher_nop_func include/linux/bpf.h:897 [inline] __bpf_prog_run include/linux/filter.h:596 [inline] bpf_prog_run include/linux/filter.h:603 [inline] bpf_test_run+0x46c/0x890 net/bpf/test_run.c:402 bpf_prog_test_run_skb+0xbdc/0x14c0 net/bpf/test_run.c:1170 bpf_prog_test_run+0x345/0x3c0 kernel/bpf/syscall.c:3648 __sys_bpf+0x43a/0x6c0 kernel/bpf/syscall.c:5005 __do_sys_bpf kernel/bpf/syscall.c:5091 [inline] __se_sys_bpf kernel/bpf/syscall.c:5089 [inline] __x64_sys_bpf+0x7c/0x90 kernel/bpf/syscall.c:5089 do_syscall_64+0x54/0x70 arch/x86/entry/common.c:48 entry_SYSCALL_64_after_hwframe+0x61/0xc6 The reproducer doesn't really reproduce outside of syzkaller environment, so I'm taking a guess here. It looks like we do generate correct ETH_HLEN-sized packet, but we redirect the packet to the tunneling device. Before we do so, we __skb_pull l2 header and arrive again at skb->len == 0. Doesn't seem like we can do anything better than having an explicit check after __skb_pull?

CVE ID	Description
CVE-2022-50258	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Fix potential stack-out-of-bounds in brcmf_c_preinit_dcmds() This patch fixes a stack-out-of-bounds read in brcmfmac that occurs when 'buf' that is not null-terminated is passed as an argument of strsep() in brcmf_c_preinit_dcmds(). This buffer is filled with a firmware version string by memcpy() in brcmf_fil_iovar_data_get(). The patch ensures buf is null-terminated. Found by a modified version of syzkaller. [47.569679][T1897] brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac43236b for chip BCM43236/3 [47.582839][T1897] brcmfmac: brcmf_c_process_clm_blob: no clm_blob available (err=-2), device may have limited channels available [47.601565][T1897] ===== [47.602574][T1897] BUG: KASAN: stack-out-of-bounds in strsep+0x1b2/0x1f0 [47.603447][T1897] Read of size 1 at addr fffff90001f6f000 by task kworker/0:2/1897 [47.604336][T1897] [47.604621][T1897] CPU: 0 PID: 1897 Comm: kworker/0:2 Tainted: G O 5.14.0+ #131 [47.605617][T1897] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org 04/01/2014 [47.606907][T1897] Workqueue: usb_hub_wq hub_event [47.607453][T1897] Call Trace: [47.607801][T1897] dump_stack_lvl+0x8e/0xd1 [47.608295][T1897] print_address_description.constprop.0.cold+0xf/0x334 [47.609009][T1897] ? strsep+0x1b2/0x1f0 [47.609434][T1897] ? strsep+0x1b2/0x1f0 [47.609863][T1897] kasan_report.cold+0x83/0xdf [47.610366][T1897] ? strsep+0x1b2/0x1f0 [47.610882][T1897] strsep+0x1b2/0x1f0 [47.611300][T1897] ? brcmf_fil_iovar_data_get+0x3a/0xf0 [47.611883][T1897] brcmf_c_preinit_dcmds+0x995/0xc40 [47.612434][T1897] ? brcmf_c_set_joinpref_default+0x100/0x100 [47.613078][T1897] ? rcu_read_lock_sched_held+0xa1/0xd0 [47.613662][T1897] ? rcu_read_lock_bh_held+0xb0/0xb0 [47.614208][T1897] ? lock_acquire+0x19d/0x4e0 [47.614704][T1897] ? find_held_lock+0x2d/0x110 [47.615236][T1897] ? brcmf_usb_deq+0x1a7/0x260 [47.615741][T1897] ? brcmf_usb_rx_fill_all+0x5a/0xf0 [47.616288][T1897] brcmf_attach+0x246/0xd40 [47.616758][T1897] ? wiphy_new_nm+0x1703/0x1dd0 [47.617280][T1897] ? kmemdup+0x43/0x50 [47.617720][T1897] brcmf_usb_probe+0x12de/0x1690 [47.618244][T1897] ? brcmf_usbdev_qinit.constprop.0+0x470/0x470 [47.618901][T1897] usb_probe_interface+0x2aa/0x760 [47.619429][T1897] ? usb_probe_device+0x250/0x250 [47.619950][T1897] really_probe+0x205/0xb70 [47.620435][T1897] ? driver_allows_async_probing+0x130/0x130 [47.621048][T1897] __driver_probe_device+0x311/0x4b0 [47.621595][T1897] ? driver_allows_async_probing+0x130/0x130 [47.622209][T1897] driver_probe_device+0x4e/0x150 [47.622739][T1897] __device_attach_driver+0x1cc/0x2a0 [47.623287][T1897] bus_for_each_drv+0x156/0x1d0 [47.623796][T1897] ? bus_rescan_devices+0x30/0x30 [47.624309][T1897] ? lockdep_hardirqs_on_prepare+0x273/0x3e0 [47.624907][T1897] ? trace_hardirqs_on+0x46/0x160 [47.625437][T1897] __device_attach+0x23f/0x3a0 [47.625924][T1897] ? device_bind_driver+0xd0/0xd0 [47.626433][T1897] ? kobject_uevent_env+0x287/0x14b0 [47.627057][T1897] bus_probe_device+0x1da/0x290 [47.627557][T1897] device_add+0xb7b/0x1eb0 [47.628027][T1897] ? wait_for_completion+0x290/0x290 [47.628593][T1897] ? __fw_devlink_link_to_suppliers+0x5a0/0x5a0 [47.629249][T1897] usb_set_configuration+0xf59/0x16f0 [47.629829][T1897] usb_generic_driver_probe+0x82/0xa0 [47.630385][T1897] usb_probe_device+0xbb/0x250 [47.630927][T1897] ? usb_suspend+0x590/0x590 [47.631397][T1897] really_probe+0x205/0xb70 [47.631855][T1897] ? driver_allows_async_probing+0x130/0x130 [47.632469][T1897] __driver_probe_device+0x311/0x4b0 [47.633002][T1897] ---truncated---

CVE ID	Description
CVE-2022-50259	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: fix race in sock_map_free() sock_map_free() calls release_sock(sk) without owning a reference on the socket. This can cause use-after-free as syzbot found [1] Jakub Sitnicki already took care of a similar issue in sock_hash_free() in commit 75e68e5bf2c7 ("bpf, sockhash: Synchronize delete from bucket list on map free") [1] refcount_t: decrement hit 0; leaking memory. WARNING: CPU: 0 PID: 3785 at lib/refcount.c:31 refcount_warn_saturate+0x17c/0x1a0 lib/refcount.c:31 Modules linked in: CPU: 0 PID: 3785 Comm: kworker/u4:6 Not tainted 6.1.0-rc7-syzkaller-00103-gef4d3ea40565 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Workqueue: events_unbound bpf_map_free_deferred RIP: 0010:refcount_warn_saturate+0x17c/0x1a0 lib/refcount.c:31 Code: 68 8b 31 c0 e8 75 71 15 fd 0f 0b e9 64 ff ff e8 d9 6e 4e fd c6 05 62 9c 3d 0a 01 48 c7 c7 80 bb 68 8b 31 c0 e8 54 71 15 fd <0f> 0b e9 43 ff ff ff 89 d9 80 e1 07 80 c1 03 38 c1 0f 8c a2 fe ff RSP: 0018:ffff9000456fb60 EFLAGS: 00010246 RAX: eae59bab72dcd700 RBX: 0000000000000004 RCX: ffff8880207057c0 RDX: 0000000000000000 RSI: 0000000000000201 RDI: 0000000000000000 RBP: 0000000000000004 R08: ffffffff816fdbad R09: fffff520008adee5 R10: fffff520008adee5 R11: 1ffff920008adee4 R12: 0000000000000004 R13: dffffc0000000000 R14: ffff88807b1c6c00 R15: 1ffff1100f638dcf FS: 0000000000000000(0000) GS:ffff8880b9800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000001b30c30000 CR3: 000000000d08e000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __refcount_dec include/linux/refcount.h:344 [inline] refcount_dec include/linux/refcount.h:359 [inline] __sock_put include/net/sock.h:779 [inline] tcp_release_cb+0x2d0/0x360 net/ipv4/tcp_output.c:1092 release_sock+0xaf/0x1c0 net/core/sock.c:3468 sock_map_free+0x219/0x2c0 net/core/sock_map.c:356 process_one_work+0x81c/0xd10 kernel/workqueue.c:2289 worker_thread+0xb14/0x1330 kernel/workqueue.c:2436 kthread+0x266/0x300 kernel/kthread.c:376 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:306 </TASK>
CVE-2022-50271	In the Linux kernel, the following vulnerability has been resolved: vhost/vsock: Use kvmalloc/kvfree for larger packets. When copying a large file over sftp over vsock, data size is usually 32kB, and kmalloc seems to fail to try to allocate 32 32kB regions. vhost-5837: page allocation failure: order:4, mode:0x24040c0 Call Trace: [<ffffffffffb6a0df64>] dump_stack+0x97/0xdb [<ffffffffffb68d6aed>] warn_alloc_failed+0x10f/0x138 [<ffffffffffb68d868a>] ? __alloc_pages_direct_compact+0x38/0xc8 [<ffffffffffb664619f>] __alloc_pages_nodemask+0x84c/0x90d [<ffffffffffb6646e56>] alloc_kmem_pages+0x17/0x19 [<ffffffffffb6653a26>] kmalloc_order_trace+0x2b/0xdb [<ffffffffffb66682f3>] __kmalloc+0x177/0x1f7 [<ffffffffffb66e0d94>] ? copy_from_iter+0x8d/0x31d [<ffffffffffc0689ab7>] vhost_vsock_handle_tx_kick+0x1fa/0x301 [vhost_vsock] [<ffffffffffc06828d9>] vhost_worker+0xf7/0x157 [vhost] [<ffffffffffb683ddce>] kthread+0xfd/0x105 [<ffffffffffc06827e2>] ? vhost_dev_set_owner+0x22e/0x22e [vhost] [<ffffffffffb683dcd1>] ? flush_kthread_worker+0xf3/0xf3 [<ffffffffffb6eb332e>] ret_from_fork+0x4e/0x80 [<ffffffffffb683dcd1>] ? flush_kthread_worker+0xf3/0xf3 Work around by doing kvmalloc instead.
CVE-2022-50272	In the Linux kernel, the following vulnerability has been resolved: media: dvb-usb: az6027: fix null-ptr-deref in az6027_i2c_xfer() Wei Chen reports a kernel bug as blew: general protection fault, probably for non-canonical address KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017] ... Call Trace: <TASK> __i2c_transfer+0x77e/0x1930 drivers/i2c/i2c-core-base.c:2109 i2c_transfer+0x1d5/0x3d0 drivers/i2c/i2c-core-base.c:2170 i2cdev_ioctl_rdwr+0x393/0x660 drivers/i2c/i2c-dev.c:297 i2cdev_ioctl+0x75d/0x9f0 drivers/i2c/i2c-dev.c:458 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:870 [inline] __se_sys_ioctl+0xfb/0x170 fs/ioctl.c:856 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0x90 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7fd834a8bde In az6027_i2c_xfer(), if msg[i].addr is 0x99, a null-ptr-deref will caused when accessing msg[i].buf. For msg[i].len is 0 and msg[i].buf is null. Fix this by checking msg[i].len in az6027_i2c_xfer().
CVE-2022-50275	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: Add the missed acpi_put_table() to fix memory leak When the radeon driver reads the bios information from ACPI table in radeon_acpi_vfct_bios(), it misses to call acpi_put_table() to release the ACPI memory after the init, so add acpi_put_table() properly to fix the memory leak. v2: fix text formatting (Alex)

CVE ID	Description
CVE-2022-50279	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: Fix global-out-of-bounds bug in _rtl8812ae_phy_set_txpower_limit() There is a global-out-of-bounds reported by KASAN: BUG: KASAN: global-out-of-bounds in _rtl8812ae_eq_n_byte.part.0+0x3d/0x84 [rtl8821ae] Read of size 1 at addr ffffffff0773c43 by task NetworkManager/411 CPU: 6 PID: 411 Comm: NetworkManager Tainted: G D 6.1.0-rc8+ #144 e15588508517267d37 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), Call Trace: <TASK> ... kasan_report+0xbb/0x1c0 _rtl8812ae_eq_n_byte.part.0+0x3d/0x84 [rtl8821ae] rtl8821ae_phy_bb_config.cold+0x346/0x641 [rtl8821ae] rtl8821ae_hw_init+0x1f5e/0x79b0 [rtl8821ae] ... </TASK> The root cause of the problem is that the comparison order of "prate_section" in _rtl8812ae_phy_set_txpower_limit() is wrong. The _rtl8812ae_eq_n_byte() is used to compare the first n bytes of the two strings from tail to head, which causes the problem. In the _rtl8812ae_phy_set_txpower_limit(), it was originally intended to meet this requirement by carefully designing the comparison order. For example, "pregulation" and "pbandwidth" are compared in order of length from small to large, first is 3 and last is 4. However, the comparison order of "prate_section" dose not obey such order requirement, therefore when "prate_section" is "HT", when comparing from tail to head, it will lead to access out of bounds in _rtl8812ae_eq_n_byte(). As mentioned above, the _rtl8812ae_eq_n_byte() has the same function as strcmp(), so just strcmp() is enough. Fix it by removing _rtl8812ae_eq_n_byte() and use strcmp() barely. Although it can be fixed by adjusting the comparison order of "prate_section", this may cause the value of "rate_section" to not be from 0 to 5. In addition, commit "21e4b0726dc6" not only moved driver from staging to regular tree, but also added setting txpower limit function during the driver config phase, so the problem was introduced by this commit.
CVE-2022-50280	In the Linux kernel, the following vulnerability has been resolved: pnode: terminate at peers of source The propagate_mnt() function handles mount propagation when creating mounts and propagates the source mount tree @source_mnt to all applicable nodes of the destination propagation mount tree headed by @dest_mnt. Unfortunately it contains a bug where it fails to terminate at peers of @source_mnt when looking up copies of the source mount that become masters for copies of the source mount tree mounted on top of slaves in the destination propagation tree causing a NULL dereference. Once the mechanics of the bug are understood it's easy to trigger. Because of unprivileged user namespaces it is available to unprivileged users. While fixing this bug we've gotten confused multiple times due to unclear terminology or missing concepts. So let's start this with some clarifications: * The terms "master" or "peer" denote a shared mount. A shared mount belongs to a peer group. * A peer group is a set of shared mounts that propagate to each other. They are identified by a peer group id. The peer group id is available in @shared_mnt->mnt_group_id. Shared mounts within the same peer group have the same peer group id. The peers in a peer group can be reached via @shared_mnt->mnt_share. * The terms "slave mount" or "dependent mount" denote a mount that receives propagation from a peer in a peer group. IOW, shared mounts may have slave mounts and slave mounts have shared mounts as their master. Slave mounts of a given peer in a peer group are listed on that peers slave list available at @shared_mnt->mnt_slave_list. * The term "master mount" denotes a mount in a peer group. IOW, it denotes a shared mount or a peer mount in a peer group. The term "master mount" - or "master" for short - is mostly used when talking in the context of slave mounts that receive propagation from a master mount. A master mount of a slave identifies the closest peer group a slave mount receives propagation from. The master mount of a slave can be identified via @slave_mount->mnt_master. Different slaves may point to different masters in the same peer group. * Multiple peers in a peer group can have non-empty ->mnt_slave_lists. Non-empty ->mnt_slave_lists of peers don't intersect. Consequently, to ensure all slave mounts of a peer group are visited the ->mnt_slave_lists of all peers in a peer group have to be walked. * Slave mounts point to a peer in the closest peer group they receive propagation from via @slave_mnt->mnt_master (see above). Together with these peers they form a propagation group (see below). The closest peer group can thus be identified through the peer group id @slave_mnt->mnt_master->mnt_group_id of the peer/master that a slave mount receives propagation from. * A shared-slave mount is a slave mount to a peer group pg1 while also a peer in another peer group pg2. IOW, a peer group may receive propagation from another peer group. If a peer group pg1 is a slave to another peer group pg2 then all peers in peer group pg1 point to the same peer in peer group pg2 via ->mnt_master. IOW, all peers in peer group pg1 appear on the same ->mnt_slave_list. IOW, they cannot be slaves to different peer groups. * A pure slave mount is a slave mount that is a slave to a peer group but is not a peer in another peer group. * A propagation group denotes the set of mounts consisting of a single peer group pg1 and all slave mounts and shared-slave mounts that point to a peer in that peer group via ->mnt_master. IOW, all slave mounts such that @slave_mnt->mnt_master->mnt_group_id is equal to @shared_mnt->mnt_group_id. The concept of a propagation group makes it easier to talk about a single propagation level in a propagation tree. For example, in propagate_mnt() the immediate peers of @dest_mnt and all slaves of @dest_mnt's peer group form a propagation group pr ---truncated---

CVE ID	Description
CVE-2022-50282	In the Linux kernel, the following vulnerability has been resolved: chardev: fix error handling in cdev_device_add() While doing fault injection test, I got the following report: -----[cut here]----- kobject: '(null)' (0000000039956980): is not initialized, yet kobject_put() is being called. WARNING: CPU: 3 PID: 6306 at kobject_put+0x23d/0x4e0 CPU: 3 PID: 6306 Comm: 283 Tainted: G W 6.1.0-rc2-00005-g307c1086d7c9 #1253 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 RIP: 0010:kobject_put+0x23d/0x4e0 Call Trace: <TASK> cdev_device_add+0x15e/0x1b0 __iio_device_register+0x13b4/0x1af0 [industrialio] __devm_iio_device_register+0x22/0x90 [industrialio] max517_probe+0x3d8/0x6b4 [max517] i2c_device_probe+0xa81/0xc00 When device_add() is injected fault and returns error, if dev->devt is not set, cdev_add() is not called, cdev_del() is not needed. Fix this by checking dev->devt in error path.
CVE-2022-50285	In the Linux kernel, the following vulnerability has been resolved: mm,hugetlb: take hugetlb_lock before decrementing h->resv_huge_pages The h->* _huge_pages counters are protected by the hugetlb_lock, but alloc_huge_page has a corner case where it can decrement the counter outside of the lock. This could lead to a corrupted value of h->resv_huge_pages, which we have observed on our systems. Take the hugetlb_lock before decrementing h->resv_huge_pages to avoid a potential race.
CVE-2022-50286	In the Linux kernel, the following vulnerability has been resolved: ext4: fix delayed allocation bug in ext4_clu_mapped for bigalloc + inline When converting files with inline data to extents, delayed allocations made on a file system created with both the bigalloc and inline options can result in invalid extent status cache content, incorrect reserved cluster counts, kernel memory leaks, and potential kernel panics. With bigalloc, the code that determines whether a block must be delayed allocated searches the extent tree to see if that block maps to a previously allocated cluster. If not, the block is delayed allocated, and otherwise, it isn't. However, if the inline option is also used, and if the file containing the block is marked as able to store data inline, there isn't a valid extent tree associated with the file. The current code in ext4_clu_mapped() calls ext4_find_extent() to search the non-existent tree for a previously allocated cluster anyway, which typically finds nothing, as desired. However, a side effect of the search can be to cache invalid content from the non-existent tree (garbage) in the extent status tree, including bogus entries in the pending reservation tree. To fix this, avoid searching the extent tree when allocating blocks for bigalloc + inline files that are being converted from inline to extent mapped.
CVE-2022-50287	In the Linux kernel, the following vulnerability has been resolved: drm/i915/bios: fix a memory leak in generate_lfp_data_ptrs When (size != 0 ptrs->lvsd_entries != 3), the program tries to free() the ptrs. However, the ptrs is not created by calling kzmalloc(), but is obtained by pointer offset operation. This may lead to memory leaks or undefined behavior. Fix this by replacing the arguments of kfree() with ptrs_block. (cherry picked from commit 7674cd0b7d28b952151c3df26bffa7e07eb2b4ec)
CVE-2022-50297	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k: verify the expected usb_endpoints are present The bug arises when a USB device claims to be an ATH9K but doesn't have the expected endpoints. (In this case there was an interrupt endpoint where the driver expected a bulk endpoint.) The kernel needs to be able to handle such devices without getting an internal error. usb 1-1: BOGUS urb xfer, pipe 3 != type 1 WARNING: CPU: 3 PID: 500 at drivers/usb/core/urb.c:493 usb_submit_urb+0xce2/0x1430 drivers/usb/core/urb.c:493 Modules linked in: CPU: 3 PID: 500 Comm: kworker/3:2 Not tainted 5.10.135-syzkaller #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 Workqueue: events request_firmware_work_func RIP: 0010:usb_submit_urb+0xce2/0x1430 drivers/usb/core/urb.c:493 Call Trace: ath9k_hif_usb_alloc_rx_urbs drivers/net/wireless/ath/ath9k/hif_usb.c:908 [inline] ath9k_hif_usb_alloc_urbs+0x75e/0x1010 drivers/net/wireless/ath/ath9k/hif_usb.c:1019 ath9k_hif_usb_dev_init drivers/net/wireless/ath/ath9k/hif_usb.c:1109 [inline] ath9k_hif_usb_firmware_cb+0x142/0x530 drivers/net/wireless/ath/ath9k/hif_usb.c:1242 request_firmware_work_func+0x12e/0x240 drivers/base/firmware_loader/main.c:1097 process_one_work+0x9af/0x1600 kernel/workqueue.c:2279 worker_thread+0x61d/0x12f0 kernel/workqueue.c:2425 kthread+0x3b4/0x4a0 kernel/kthread.c:313 ret_from_fork+0x22/0x30 arch/x86/entry/entry_64.S:299 Found by Linux Verification Center (linuxtesting.org) with Syzkaller.

CVE ID	Description
CVE-2022-50299	In the Linux kernel, the following vulnerability has been resolved: md: Replace snprintf with scnprintf Current code produces a warning as shown below when total characters in the constituent block device names plus the slashes exceeds 200. snprintf() returns the number of characters generated from the given input, which could cause the expression “200 – len” to wrap around to a large positive number. Fix this by using scnprintf() instead, which returns the actual number of characters written into the buffer. [1513.267938] -----[cut here]----- [1513.267943] WARNING: CPU: 15 PID: 37247 at <snip>/lib/vsprintf.c:2509 vsnprintf+0x2c8/0x510 [1513.267944] Modules linked in: <snip> [1513.267969] CPU: 15 PID: 37247 Comm: mdadm Not tainted 5.4.0-1085-azure #90~18.04.1-Ubuntu [1513.267969] Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS Hyper-V UEFI Release v4.1 05/09/2022 [1513.267971] RIP: 0010:vsnprintf+0x2c8/0x510 <-snip-> [1513.267982] Call Trace: [1513.267986] snprintf+0x45/0x70 [1513.267990] ? disk_name+0x71/0xa0 [1513.267993] dump_zones+0x114/0x240 [raid0] [1513.267996] ? _cond_resched+0x19/0x40 [1513.267998] raid0_run+0x19e/0x270 [raid0] [1513.268000] md_run+0x5e0/0xc50 [1513.268003] ? security_capable+0x3f/0x60 [1513.268005] do_md_run+0x19/0x110 [1513.268006] md_ioctl+0x195e/0x1f90 [1513.268007] blkdev_ioctl+0x91f/0x9f0 [1513.268010] block_ioctl+0x3d/0x50 [1513.268012] do_vfs_ioctl+0xa9/0x640 [1513.268014] ? _fput+0x162/0x260 [1513.268016] ksys_ioctl+0x75/0x80 [1513.268017] __x64_sys_ioctl+0x1a/0x20 [1513.268019] do_syscall_64+0x5e/0x200 [1513.268021] entry_SYSCALL_64_after_hwframe+0x44/0xa9
CVE-2022-50303	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix double release compute pasid If kfd_process_device_init_vm returns failure after vm is converted to compute vm and vm->pasid set to compute pasid, KFD will not take pdd->drm_file reference. As a result, drm close file handler maybe called to release the compute pasid before KFD process destroy worker to release the same pasid and set vm->pasid to zero, this generates below WARNING backtrace and NULL pointer access. Add helper amdgpu_amdkfd_gpuvm_set_vm_pasid and call it at the last step of kfd_process_device_init_vm, to ensure vm pasid is the original pasid if acquiring vm failed or is the compute pasid with pdd->drm_file reference taken to avoid double release same pasid. amdgpu: Failed to create process VM object ida_free called for id=32770 which is not allocated. WARNING: CPU: 57 PID: 72542 at ./lib/ida.c:522 ida_free+0x96/0x140 RIP: 0010:ida_free+0x96/0x140 Call Trace: amdgpu_pasid_free_delayed+0xe1/0x2a0 [amdgpu] amdgpu_driver_postclose_kms+0x2d8/0x340 [amdgpu] drm_file_free.part.13+0x216/0x270 [drm] drm_close_helper.isra.14+0x60/0x70 [drm] drm_release+0x6e/0xf0 [drm] __fput+0xcc/0x280 ____fput+0xe/0x20 task_work_run+0x96/0xc0 do_exit+0x3d0/0xc10 BUG: kernel NULL pointer dereference, address: 0000000000000000 RIP: 0010:ida_free+0x76/0x140 Call Trace: amdgpu_pasid_free_delayed+0xe1/0x2a0 [amdgpu] amdgpu_driver_postclose_kms+0x2d8/0x340 [amdgpu] drm_file_free.part.13+0x216/0x270 [drm] drm_close_helper.isra.14+0x60/0x70 [drm] drm_release+0x6e/0xf0 [drm] __fput+0xcc/0x280 ____fput+0xe/0x20 task_work_run+0x96/0xc0 do_exit+0x3d0/0xc10
CVE-2022-50304	In the Linux kernel, the following vulnerability has been resolved: mtd: core: fix possible resource leak in init_mtd() I got the error report while inject fault in init_mtd(): sysfs: cannot create duplicate filename '/devices/virtual/bdi/mtd-0' Call Trace: <TASK> dump_stack_lvl+0x67/0x83 sysfs_warn_dup+0x60/0x70 sysfs_create_dir_ns+0x109/0x120 kobject_add_internal+0xce/0x2f0 kobject_add+0x98/0x110 device_add+0x179/0xc00 device_create_groups_vargs+0xf4/0x100 device_create+0x7b/0xb0 bdi_register_va.part.13+0x58/0x2d0 bdi_register+0x9b/0xb0 init_mtd+0x62/0x171 [mtd] do_one_initcall+0x6c/0x3c0 do_init_module+0x58/0x222 load_module+0x268e/0x27d0 __do_sys_finit_module+0xd5/0x140 do_syscall_64+0x37/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd </TASK> kobject_add_internal failed for mtd-0 with -EEXIST, don't try to register things with the same name in the same directory. Error registering mtd class or bdi: -17 If init_mtdchar() fails in init_mtd(), mtd_bdi will not be unregistered, as a result, we can't load the mtd module again, to fix this by calling bdi_unregister(mtd_bdi) after out_procfs label.
CVE-2022-50305	In the Linux kernel, the following vulnerability has been resolved: ASoC: sof_es8336: fix possible use-after-free in sof_es8336_remove() sof_es8336_remove() calls cancel_delayed_work(). However, that function does not wait until the work function finishes. This means that the callback function may still be running after the driver's remove function has finished, which would result in a use-after-free. Fix by calling cancel_delayed_work_sync(), which ensures that the work is properly cancelled, no longer running, and unable to re-schedule itself.
CVE-2022-50308	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: Add checks for devm_kcalloc As the devm_kcalloc may return NULL, the return value needs to be checked to avoid NULL pointer dereference.
CVE-2022-50311	In the Linux kernel, the following vulnerability has been resolved: cxl: Fix refcount leak in cxl_calc_capp_routing_of_get_next_parent() returns a node pointer with refcount incremented, we should use of_node_put() on it when not need anymore. This function only calls of_node_put() in normal path, missing it in the error path. Add missing of_node_put() to avoid refcount leak.
CVE-2022-50312	In the Linux kernel, the following vulnerability has been resolved: drivers: serial: jsm: fix some leaks in probe This error path needs to unwind instead of just returning directly.

CVE ID	Description
CVE-2022-50314	In the Linux kernel, the following vulnerability has been resolved: nbd: Fix hung when signal interrupts nbd_start_device_ioctl() syzbot reported hung task [1]. The following program is a simplified version of the reproducer: int main(void) { int sv[2], fd; if (socketpair(AF_UNIX, SOCK_STREAM, 0, sv) < 0) return 1; if ((fd = open("/dev/nbd0", 0)) < 0) return 1; if (ioctl(fd, NBD_SET_SIZE_BLOCKS, 0x81) < 0) return 1; if (ioctl(fd, NBD_SET_SOCKET, sv[0]) < 0) return 1; if (ioctl(fd, NBD_DO_IT) < 0) return 1; return 0; } When signal interrupt nbd_start_device_ioctl() waiting the condition atomic_read(&config->recv_threads) == 0, the task can hung because it waits the completion of the inflight IOs. This patch fixes the issue by clearing queue, not just shutdown, when signal interrupt nbd_start_device_ioctl().
CVE-2022-50317	In the Linux kernel, the following vulnerability has been resolved: drm/bridge: megachips: Fix a null pointer dereference bug When removing the module we will get the following warning: [31.911505] i2c-core: driver [stdp2690-ge-b850v3-fw] unregistered [31.912484] general protection fault, probably for non-canonical address 0xdffffc0000000001: 0000 [#1] PREEMPT SMP KASAN PTI [31.913338] KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] [31.915280] RIP: 0010:drm_bridge_remove+0x97/0x130 [31.921825] Call Trace: [31.922533] stdp4028_ge_b850v3_fw_remove+0x34/0x60 [megachips_stdpxxxx_ge_b850v3_fw] [31.923139] i2c_device_remove+0x181/0x1f0 The two bridges (stdp2690, stdp4028) do not probe at the same time, so the driver does not call ge_b850v3_resgiter() when probing, causing the driver to try to remove the object that has not been initialized. Fix this by checking whether both the bridges are probed.
CVE-2022-50318	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/uncore: Fix reference count leak in hswep_has_limit_sbox() pci_get_device() will increase the reference count for the returned 'dev'. We need to call pci_dev_put() to decrease the reference count. Since 'dev' is only used in pci_read_config_dword(), let's add pci_dev_put() right after it.
CVE-2022-50320	In the Linux kernel, the following vulnerability has been resolved: ACPI: tables: FPDT: Don't call acpi_os_map_memory() on invalid phys address On a Packard Bell Dot SC (Intel Atom N2600 model) there is a FPDT table which contains invalid physical addresses, with high bits set which fall outside the range of the CPU-s supported physical address range. Calling acpi_os_map_memory() on such an invalid phys address leads to the below WARN_ON in ioremap triggering resulting in an oops/stacktrace. Add code to verify the physical address before calling acpi_os_map_memory() to fix / avoid the oops. [1.226900] ioremap: invalid physical address 3001000000000000 [1.226949] -----[cut here]----- [1.226962] WARNING: CPU: 1 PID: 1 at arch/x86/mm/ioremap.c:200 __ioremap_caller.cold+0x43/0x5f [1.226996] Modules linked in: [1.227016] CPU: 1 PID: 1 Comm: swapper/0 Not tainted 6.0.0-rc3+ #490 [1.227029] Hardware name: Packard Bell dot s/SJE01_CT, BIOS V1.10 07/23/2013 [1.227038] RIP: 0010: __ioremap_caller.cold+0x43/0x5f [1.227054] Code: 96 00 00 e9 f8 af 24 ff 89 c6 48 c7 c7 d8 0c 84 99 e8 6a 96 00 00 e9 76 af 24 ff 48 89 fe 48 c7 c7 a8 0c 84 99 e8 56 96 00 00 <0f> 0b e9 60 af 24 ff 48 8b 34 24 48 c7 c7 40 0d 84 99 e8 3f 96 00 [1.227067] RSP: 0000:ffffb18c40033d60 EFLAGS: 00010286 [1.227084] RAX: 0000000000000032 RBX: 3001000000000000 RCX: 0000000000000000 [1.227095] RDX: 0000000000000001 RSI: 00000000ffffdfff RDI: 00000000ffffff [1.227105] RBP: 3001000000000000 R08: 0000000000000000 R09: fffffb18c40033c18 [1.227115] R10: 0000000000000003 R11: ffffffff99d62fe8 R12: 0000000000000008 [1.227124] R13: 0003001000000000 R14: 0000000000001000 R15: 3001000000000000 [1.227135] FS: 0000000000000000(0000) GS:ffff913a3c080000(0000) knlGS:0000000000000000 [1.227146] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [1.227156] CR2: 0000000000000000 CR3: 0000000018c26000 CR4: 00000000000000e0 [1.227167] Call Trace: [1.227176] <TASK> [1.227185] ? acpi_os_map_iomem+0x1c9/0x1e0 [1.227215] ? kmem_cache_alloc_trace+0x187/0x370 [1.227254] acpi_os_map_iomem+0x1c9/0x1e0 [1.227288] acpi_init_fpdt+0xa8/0x253 [1.227308] ? acpi_debugfs_init+0x1f/0x1f [1.227339] do_one_initcall+0x5a/0x300 [1.227406] ? rcu_read_lock_sched_held+0x3f/0x80 [1.227442] kernel_init_freeable+0x28b/0x2cc [1.227512] ? rest_init+0x170/0x170 [1.227538] kernel_init+0x16/0x140 [1.227552] ret_from_fork+0x1f/0x30 [1.227639] </TASK> [1.227647] irq event stamp: 186819 [1.227656] hardirqs last enabled at (186825): [<fffffff98184a6e>] __up_console_sem+0x5e/0x70 [1.227672] hardirqs last disabled at (186830): [<fffffff98184a53>] __up_console_sem+0x43/0x70 [1.227686] softirqs last enabled at (186576): [<fffffff980fbc9d>] __irq_exit_rcu+0xed/0x160 [1.227701] softirqs last disabled at (186569): [<fffffff980fbc9d>] __irq_exit_rcu+0xed/0x160 [1.227715] ---[end trace 0000000000000000]---
CVE-2022-50321	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: fix potential memory leak in brcmf_netdev_start_xmit() The brcmf_netdev_start_xmit() returns NETDEV_TX_OK without freeing skb in case of pskb_expand_head() fails, add dev_kfree_skb() to fix it. Compile tested only.
CVE-2022-50328	In the Linux kernel, the following vulnerability has been resolved: jbd2: fix potential use-after-free in jbd2_fc_wait_bufs In 'jbd2_fc_wait_bufs' use 'bh' after put buffer head reference count which may lead to use-after-free. So judge buffer if uptodate before put buffer head reference count.

CVE ID	Description
CVE-2022-50329	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix uaf for bfqq in bfq_exit_icq_bfq Commit 64dc8c732f5c ("block, bfq: fix possible uaf for 'bfqq->bic'") will access 'bic->bfqq' in bic_set_bfq(), however, bfq_exit_icq_bfq() can free bfqq first, and then call bic_set_bfq(), which will cause uaf. Fix the problem by moving bfq_exit_bfq() behind bic_set_bfq().
CVE-2022-50332	In the Linux kernel, the following vulnerability has been resolved: video/aperture: Call sysfb_disable() before removing PCI devices Call sysfb_disable() from aperture_remove_conflicting_pci_devices() before removing PCI devices. Without, simpledrm can still bind to simple-framebuffer devices after the hardware driver has taken over the hardware. Both drivers interfere with each other and results are undefined. Reported modesetting errors [1] are shown below. ---- snap ---- rcu: INFO: rcu_sched detected expedited stalls on CPUs/tasks: { 13-.... } 7 jiffies s: 165 root: 0x2000/. rcu: blocking rcu_node structures (internal RCU debug): Task dump for CPU 13: task:X state:R running task stack: 0 pid: 4242 ppid: 4228 flags:0x00000008 Call Trace: <TASK> ? commit_tail+0xd7/0x130 ? drm_atomic_helper_commit+0x126/0x150 ? drm_atomic_commit+0xa4/0xe0 ? drm_plane_get_damage_clips.cold+0x1c/0x1c ? drm_atomic_helper_dirtyfb+0x19e/0x280 ? drm_mode_dirtyfb_ioctl+0x10f/0x1e0 ? drm_mode_getfb2_ioctl+0x2d0/0x2d0 ? drm_ioctl_kernel+0xc4/0x150 ? drm_ioctl+0x246/0x3f0 ? drm_mode_getfb2_ioctl+0x2d0/0x2d0 ? __x64_sys_ioctl+0x91/0xd0 ? do_syscall_64+0x60/0xd0 ? entry_SYSCALL_64_after_hwframe+0x4b/0xb5 </TASK> ... rcu: INFO: rcu_sched detected expedited stalls on CPUs/tasks: { 13-.... } 30 jiffies s: 169 root: 0x2000/. rcu: blocking rcu_node structures (internal RCU debug): Task dump for CPU 13: task:X state:R running task stack: 0 pid: 4242 ppid: 4228 flags:0x0000400e Call Trace: <TASK> ? memcpy_toio+0x76/0xc0 ? memcpy_toio+0x1b/0xc0 ? drm_fb_memcpy_toio+0x76/0xb0 ? drm_fb_blit_toio+0x75/0x2b0 ? simpledrm_simple_display_pipe_update+0x132/0x150 ? drm_atomic_helper_commit_planes+0xb6/0x230 ? drm_atomic_helper_commit_tail+0x44/0x80 ? commit_tail+0xd7/0x130 ? drm_atomic_helper_commit+0x126/0x150 ? drm_atomic_commit+0xa4/0xe0 ? drm_plane_get_damage_clips.cold+0x1c/0x1c ? drm_atomic_helper_dirtyfb+0x19e/0x280 ? drm_mode_dirtyfb_ioctl+0x10f/0x1e0 ? drm_mode_getfb2_ioctl+0x2d0/0x2d0 ? drm_ioctl_kernel+0xc4/0x150 ? drm_ioctl+0x246/0x3f0 ? drm_mode_getfb2_ioctl+0x2d0/0x2d0 ? __x64_sys_ioctl+0x91/0xd0 ? do_syscall_64+0x60/0xd0 ? entry_SYSCALL_64_after_hwframe+0x4b/0xb5 </TASK> The problem was added by commit 5e0137612430 ("video/aperture: Disable and unregister sysfb devices via aperture helpers") to v6.0.3 and does not exist in the mainline branch. The mainline commit 5e0137612430 ("video/aperture: Disable and unregister sysfb devices via aperture helpers") has been backported from v6.0-rc1 to stable v6.0.3 from a larger patch series [2] that reworks fbdev framebuffer ownership. The backport misses a change to aperture_remove_conflicting_pci_devices(). Mainline itself is fine, because the function does not exist there as a result of the patch series. Instead of backporting the whole series, fix the additional function.
CVE-2022-50334	In the Linux kernel, the following vulnerability has been resolved: hugetlbfs: fix null- ptr-deref in hugetlbfs_parse_param() Syzkaller reports a null- ptr-deref bug as follows: ===== KASAN: null- ptr-deref in range [0x0000000000000000-0x0000000000000007] RIP: 0010:hugetlbfs_parse_param+0x1dd/0x8e0 fs/hugetlbfs/inode.c:1380 [...] Call Trace: <TASK> vfs_parse_fs_param fs/fs_context.c:148 [inline] vfs_parse_fs_param+0x1f9/0x3c0 fs/fs_context.c:129 vfs_parse_fs_string+0xdb/0x170 fs/fs_context.c:191 generic_parse_monolithic+0x16f/0x1f0 fs/fs_context.c:231 do_new_mount fs/namespace.c:3036 [inline] path_mount+0x12de/0x1e20 fs/namespace.c:3370 do_mount fs/namespace.c:3383 [inline] __do_sys_mount fs/namespace.c:3591 [inline] __se_sys_mount fs/namespace.c:3568 [inline] __x64_sys_mount+0x27f/0x300 fs/namespace.c:3568 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd [...] </TASK> ===== According to commit "vfs: parse: deal with zero length string value", kernel will set the param->string to null pointer in vfs_parse_fs_string() if fs string has zero length. Yet the problem is that, hugetlbfs_parse_param() will dereference the param->string, without checking whether it is a null pointer. To be more specific, if hugetlbfs_parse_param() parses an illegal mount parameter, such as "size=," , kernel will construct struct fs_parameter with null pointer in vfs_parse_fs_string(), then passes this struct fs_parameter to hugetlbfs_parse_param(), which triggers the above null- ptr-deref bug. This patch solves it by adding sanity check on param->string in hugetlbfs_parse_param().

CVE ID	Description
CVE-2022-50341	In the Linux kernel, the following vulnerability has been resolved: cifs: fix oops during encryption When running xfstests against Azure the following oops occurred on an arm64 system Unable to handle kernel write to read-only memory at virtual address ffff0001221cf000 Mem abort info: ESR = 0x9600004f EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x0f: level 3 permission fault Data abort info: ISV = 0, ISS = 0x0000004f CM = 0, WnR = 1 swapper pgtable: 4k pages, 48-bit VAs, pgdp=00000000294f3000 [ffff0001221cf000] pgd=18000001ffff8003, p4d=18000001ffff8003, pud=18000001ff82e003, pmd=18000001ff71d003, pte=00600001221cf787 Internal error: Oops: 9600004f [#1] PREEMPT SMP ... pstate: 80000005 (Nzcv daif -PAN -UAO -TCO BTYP=) pc : __memcpy+0x40/0x230 lr : scatterwalk_copypchunks+0xe0/0x200 sp : ffff800014e92de0 x29: ffff800014e92de0 x28: ffff000114f9de80 x27: 0000000000000008 x26: 0000000000000008 x25: ffff800014e92e78 x24: 0000000000000008 x23: 0000000000000001 x22: 0000040000000000 x21: ffff000000000000 x20: 0000000000000001 x19: ffff0001037c4488 x18: 0000000000000014 x17: 235e1c0d6efa9661 x16: a435f9576b6edd6c x15: 0000000000000058 x14: 0000000000000001 x13: 0000000000000008 x12: ffff000114f2e590 x11: ffffffff ffff x10: 0000040000000000 x9: ffff8000105c3580 x8: 2e9413b10000001a x7: 534b4410fb86b005 x6: 534b4410fb86b005 x5: ffff0001221cf008 x4: ffff0001037c4490 x3: 0000000000000001 x2: 0000000000000008 x1: ffff0001037c4488 x0: ffff0001221cf000 Call trace: __memcpy+0x40/0x230 scatterwalk_map_and_copy+0x98/0x100 crypto_ccm_encrypt+0x150/0x180 crypto_aead_encrypt+0x2c/0x40 crypt_message+0x750/0x880 smb3_init_transform_rq+0x298/0x340 smb_send_rqst.part.11+0xd8/0x180 smb_send_rqst+0x3c/0x100 compound_send_recv+0x534/0x5c0 smb2_query_info_compound+0x32c/0x440 smb2_set_ea+0x438/0x4c0 cifs_xattr_set+0x5d4/0x7c0 This is because in scatterwalk_copypchunks(), we attempted to write to a buffer (@sign) that was allocated in the stack (vmalloc area) by crypt_message() and thus accessing its remaining 8 (x2) bytes ended up crossing a page boundary. To simply fix it, we could just pass @sign kmalloc'd from crypt_message() and then we're done. Luckily, we don't seem to pass any other vmalloc'd buffers in smb_rqst::rq_iiov... Instead, let's map the correct pages and offsets from vmalloc buffers as well in cifs_sg_set_buf() and then avoiding such oopses.
CVE-2022-50344	In the Linux kernel, the following vulnerability has been resolved: ext4: fix null-ptr-deref in ext4_write_info I caught a null-ptr-deref bug as follows: <pre>===== KASAN: null-ptr-deref in range [0x0000000000000068-0x000000000000006f] CPU: 1 PID: 1589 Comm: umount Not tainted 5.10.0-02219-dirty #339 RIP: 0010:ext4_write_info +0x53/0x1b0 [...] Call Trace: dquot_writeback_dquots+0x341/0x9a0 ext4_sync_fs+0x19e/0x800 __sync_filesystem+0x83/0x100 sync_filesystem+0x89/0xf0 generic_shutdown_super +0x79/0x3e0 kill_block_super+0xa1/0x110 deactivate_locked_super+0xac/0x130 deactivate_super +0xb6/0xd0 cleanup_mnt+0x289/0x400 __cleanup_mnt+0x16/0x20 task_work_run +0x11c/0x1c0 exit_to_user_mode_prepare+0x203/0x210 syscall_exit_to_user_mode +0x5b/0x3a0 do_syscall_64+0x59/0x70 entry_SYSCALL_64_after_hwframe+0x44/0xa9 =====</pre> Above issue may happen as follows: ----- exit_to_user_mode_prepare task_work_run __cleanup_mnt cleanup_mnt deactivate_super deactivate_locked_super kill_block_super generic_shutdown_super shrink_dcache_for_umount dentry = sb->s_root sb->s_root = NULL <--- Here set NULL sync_filesystem __sync_filesystem sb->s_op->sync_fs > ext4_sync_fs dquot_writeback_dquots sb->dq_op->write_info > ext4_write_info ext4_journal_start(d_inode(sb->s_root), EXT4_HT_QUOTA, 2) d_inode(sb->s_root) s_root->d_inode <--- Null pointer dereference To solve this problem, we use ext4_journal_start_sb directly to avoid s_root being used.
CVE-2022-50348	In the Linux kernel, the following vulnerability has been resolved: nfsd: Fix a memory leak in an error handling path If this memdup_user() call fails, the memory allocated in a previous call a few lines above should be freed. Otherwise it leaks.
CVE-2022-50349	In the Linux kernel, the following vulnerability has been resolved: misc: tifm: fix possible memory leak in tifm_7xx1_switch_media() If device_register() returns error in tifm_7xx1_switch_media(), name of kobject which is allocated in dev_set_name() called in device_add() is leaked. Never directly free @dev after calling device_register(), even if it returned an error! Always use put_device() to give up the reference initialized.

CVE ID	Description
CVE-2022-50352	In the Linux kernel, the following vulnerability has been resolved: net: hns: fix possible memory leak in hnae_ae_register() Inject fault while probing module, if device_register() fails, but the refcount of kobject is not decreased to 0, the name allocated in dev_set_name() is leaked. Fix this by calling put_device(), so that name can be freed in callback function kobject_cleanup(). unreferenced object 0xffff00c01aba2100 (size 128): comm "systemd-udevd", pid 1259, jiffies 4294903284 (age 294.152s) hex dump (first 32 bytes): 68 6e 61 65 30 00 00 00 18 21 ba 1a c0 00 ff ff hnae0..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<0000000034783f26>] slab_post_alloc_hook+0xa0/0x3e0 [<00000000748188f2>] __kmem_cache_alloc_node+0x164/0x2b0 [<00000000ab0743e8>] __kmalloc_node_track_caller+0x6c/0x390 [<0000000006c0ffb13>] kvasprintf+0x8c/0x118 [<00000000fa27bfe1>] kvasprintf_const+0x60/0xc8 [<00000000083e10ed7>] kobject_set_name_vargs+0x3c/0xc0 [<000000000b87affc>] dev_set_name+0x7c/0xa0 [<000000003fd8fe26>] hnae_ae_register+0xcc/0x190 [hnae] [<00000000fe97edc9>] hns_dsaf_ae_init+0x9c/0x108 [hns_dsaf] [<00000000c36ff1eb>] hns_dsaf_probe+0x548/0x748 [hns_dsaf]
CVE-2022-50354	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix kfd_process_device_init_vm error handling Should only destroy the ib_mem and let process cleanup worker to free the outstanding BOs. Reset the pointer in pdd->qp structure, to avoid NULL pointer access in process destroy worker. BUG: kernel NULL pointer dereference, address: 0000000000000010 Call Trace: amdgpu_amdkfd_gpuvm_unmap_gtt_bo_from_kernel+0x46/0xb0 [amdgpu] kfd_process_device_destroy_cwsr_dgpu+0x40/0x70 [amdgpu] kfd_process_destroy_pdds+0x71/0x190 [amdgpu] kfd_process_wq_release+0x2a2/0x3b0 [amdgpu] process_one_work+0x2a1/0x600 worker_thread+0x39/0x3d0
CVE-2022-50358	In the Linux kernel, the following vulnerability has been resolved: brcmfmac: return error when getting invalid max_flowrings from dongle When firmware hit trap at initialization, host will read abnormal max_flowrings number from dongle, and it will cause kernel panic when doing iowrite to initialize dongle ring. To detect this error at early stage, we directly return error when getting invalid max_flowrings(>256).
CVE-2022-50359	In the Linux kernel, the following vulnerability has been resolved: media: cx88: Fix a null-ptr-deref bug in buffer_prepare() When the driver calls cx88_risc_buffer() to prepare the buffer, the function call may fail, resulting in a empty buffer and null-ptr-deref later in buffer_queue(). The following log can reveal it: [41.822762] general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN PTI [41.824488] KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] [41.828027] RIP: 0010:buffer_queue+0xc2/0x500 [41.836311] Call Trace: [41.836945] __enqueue_in_driver+0x141/0x360 [41.837262] vb2_start_streaming+0x62/0x4a0 [41.838216] vb2_core_streamon+0x1da/0x2c0 [41.838516] __vb2_init_fileio+0x981/0xbc0 [41.839141] __vb2_perform_fileio+0xb9f/0x1120 [41.840072] vb2_fop_read+0x20e/0x400 [41.840346] v4l2_read+0x215/0x290 [41.840603] vfs_read+0x162/0x4c0 Fix this by checking the return value of cx88_risc_buffer() [hverkuil: fix coding style issues]
CVE-2022-50365	In the Linux kernel, the following vulnerability has been resolved: skbuff: Account for tail adjustment during pull operations Extending the tail can have some unexpected side effects if a program uses a helper like BPF_FUNC_skb_pull_data to read partial content beyond the head skb headlen when all the skbs in the gso frag_list are linear with no head_frag - kernel BUG at net/core/skbuff.c:4219! pc : skb_segment+0xc4/0xd2c lr : skb_segment+0x63c/0xd2c Call trace: skb_segment+0xc4/0xd2c __udp_gso_segment+0xa4/0x544 udp4_ufo_fragment+0x184/0x1c0 inet_gso_segment+0x16c/0x3a4 skb_mac_gso_segment+0xd4/0x1b0 __skb_gso_segment+0xcc/0x12c udp_rcv_segment+0x54/0x16c udp_queue_rcv_skb+0x78/0x144 udp_unicast_rcv_skb+0x8c/0xa4 __udp4_lib_rcv+0x490/0x68c udp_rcv+0x20/0x30 ip_protocol_deliver_rcu+0x1b0/0x33c ip_local_deliver+0xd8/0x1f0 ip_rcv+0x98/0x1a4 deliver_ptype_list_skb+0x98/0x1ec __netif_receive_skb_core+0x978/0xc60 Fix this by marking these skbs as GSO_DODGY so segmentation can handle the tail updates accordingly.
CVE-2022-50366	In the Linux kernel, the following vulnerability has been resolved: powercap: intel_rapl: fix UBSAN shift-out-of-bounds issue When value < time_unit, the parameter of ilog2() will be zero and the return value is -1. u64(-1) is too large for shift exponent and then will trigger shift-out-of-bounds: shift exponent 18446744073709551615 is too large for 32-bit type 'int' Call Trace: rapl_compute_time_window_core rapl_write_data_raw set_time_window store_constraint_time_window_us
CVE-2022-50367	In the Linux kernel, the following vulnerability has been resolved: fs: fix UAF/GPF bug in nilfs_mdt_destroy In alloc_inode, inode_init_always() could return -ENOMEM if security_inode_alloc() fails, which causes inode->i_private uninitialized. Then nilfs_is_metadata_file_inode() returns true and nilfs_free_inode() wrongly calls nilfs_mdt_destroy(), which frees the uninitialized inode->i_private and leads to crashes(e.g., UAF/GPF). Fix this by moving security_inode_alloc just prior to this_cpu_inc(nr_inodes)

CVE ID	Description
CVE-2022-50370	In the Linux kernel, the following vulnerability has been resolved: i2c: designware: Fix handling of real but unexpected device interrupts Commit c7b79a752871 ("mfd: intel-lpss: Add Intel Alder Lake PCH-S PCI IDs") caused a regression on certain Gigabyte motherboards for Intel Alder Lake-S where system crashes to NULL pointer dereference in i2c_dw_xfer_msg() when system resumes from S3 sleep state ("deep"). I was able to debug the issue on Gigabyte Z690 AORUS ELITE and made following notes: - Issue happens when resuming from S3 but not when resuming from "s2idle" - PCI device 00:15.0 == i2c_designware.0 is already in D0 state when system enters into pci_pm_resume_noirq() while all other i2c_designware PCI devices are in D3. Devices were runtime suspended and in D3 prior entering into suspend - Interrupt comes after pci_pm_resume_noirq() when device interrupts are re-enabled - According to register dump the interrupt really comes from the i2c_designware.0. Controller is enabled, I2C target address register points to a one detectable I2C device address 0x60 and the DW_IC_RAW_INTR_STAT register START_DET, STOP_DET, ACTIVITY and TX_EMPTY bits are set indicating completed I2C transaction. My guess is that the firmware uses this controller to communicate with an on-board I2C device during resume but does not disable the controller before giving control to an operating system. I was told the UEFI update fixes this but never the less it revealed the driver is not ready to handle TX_EMPTY (or RX_FULL) interrupt when device is supposed to be idle and state variables are not set (especially the dev->msgs pointer which may point to NULL or stale old data). Introduce a new software status flag STATUS_ACTIVE indicating when the controller is active in driver point of view. Now treat all interrupts that occur when is not set as unexpected and mask all interrupts from the controller.
CVE-2022-50372	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix memory leak when build ntlmssp negotiate blob failed There is a memory leak when mount cifs: unreferenced object 0xffff888166059600 (size 448): comm "mount.cifs", pid 51391, jiffies 4295596373 (age 330.596s) hex dump (first 32 bytes): fe 53 4d 42 40 00 00 00 00 00 00 00 00 01 00 82 00 .SMB@..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<0000000060609a61>] mempool_alloc+0xe1/0x260 [<00000000adfa6c63>] cifs_small_buf_get+0x24/0x60 [<00000000ebb404c7>] __smb2_plain_req_init+0x32/0x460 [<00000000bcf875b4>] SMB2_sess_alloc_buffer+0xa4/0x3f0 [<00000000753a2987>] SMB2_sess_auth_rawntlmssp_negotiate+0xf5/0x480 [<00000000f0c1f4f9>] SMB2_sess_setup+0x253/0x410 [<00000000a8b83303>] cifs_setup_session+0x18f/0x4c0 [<00000000854bd16d>] cifs_get_smb_ses+0xae7/0x13c0 [<000000006cbc43d9>] mount_get_conns+0x7a/0x730 [<000000005922d816>] cifs_mount+0x103/0xd10 [<00000000e33def3b>] cifs_smb3_do_mount+0x1dd/0xc90 [<0000000078034979>] smb3_get_tree+0x1d5/0x300 [<000000004371f980>] vfs_get_tree+0x41/0xf0 [<00000000b670d8a7>] path_mount+0x9b3/0xdd0 [<000000005e839a7d>] __x64_sys_mount+0x190/0x1d0 [<000000009404c3b9>] do_syscall_64+0x35/0x80 When build ntlmssp negotiate blob failed, the session setup request should be freed.

CVE ID	Description
CVE-2022-50373	In the Linux kernel, the following vulnerability has been resolved: fs: dlm: fix race in lowcomms This patch fixes a race between queue_work() in _dlm_lowcomms_commit_msg() and srcu_read_unlock(). The queue_work() can take the final reference of a dlm_msg and so msg->idx can contain garbage which is signaled by the following warning: [676.237050] -----[cut here]----- [676.237052] WARNING: CPU: 0 PID: 1060 at include/linux/srcu.h:189 dlm_lowcomms_commit_msg+0x41/0x50 [676.238945] Modules linked in: dlm_locktorture torture rpcsec_gss_krb5 intel_rapl_msr intel_rapl_common iTCO_wdt iTCO_vendor_support qxl kvm_intel drm_ttm_helper vmw_vsock_virtio_transport kvm vmw_vsock_virtio_transport_common ttm irqbypass crc32_pclmul joydev crc32_intel serio_raw drm_kms_helper vsock virtio_scsi virtio_console virtio_balloon snd_pcm drm syscopyarea sysfillrect sysimgblt snd_timer fb_sys_fops i2c_i801 lpc_ich snd_i2c_smbus soundcore pcspkr [676.244227] CPU: 0 PID: 1060 Comm: lock_torture_wr Not tainted 5.19.0-rc3+ #1546 [676.245216] Hardware name: Red Hat KVM/RHEL-AV, BIOS 1.16.0-2.module+el8.7.0+15506+033991b0 04/01/2014 [676.246460] RIP: 0010:dlm_lowcomms_commit_msg+0x41/0x50 [676.247132] Code: fe ff ff ff 75 24 48 c7 c6 bd 0f 49 bb 48 c7 c7 38 7c 01 bd e8 00 e7 ca ff 89 de 48 c7 c7 60 78 01 bd e8 42 3d cd ff 5b 5d c3 <0f> 0b eb d8 66 66 2e 0f 1f 84 00 00 00 00 0f 1f 44 00 00 55 48 [676.249253] RSP: 0018:ffffa401c18ffc68 EFLAGS: 00010282 [676.249855] RAX: 0000000000000001 RBX: 00000000ffff8b76 RCX: 0000000000000006 [676.250713] RDX: 0000000000000000 RSI: ffffffffbcf3a10 RDI: ffffffffbc7b62e [676.251610] RBP: fffffa401c18ffc70 R08: 0000000000000001 R09: 0000000000000001 [676.252481] R10: 0000000000000001 R11: 0000000000000001 R12: 0000000000000005 [676.253421] R13: ffff8b76786ec370 R14: ffff8b76786ec370 R15: ffff8b76786ec480 [676.254257] FS: 0000000000000000(0000) GS:ffff8b7777800000(0000) knlGS:0000000000000000 [676.255239] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [676.255897] CR2: 00005590205d88b8 CR3: 000000017656c003 CR4: 0000000000770ee0 [676.256734] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [676.257567] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [676.258397] PKRU: 55555554 [676.258729] Call Trace: [676.259063] <TASK> [676.259354] dlm_midcomms_commit_mhandle+0x0cc/0x110 [676.259964] queue_bast+0x8b/0xb0 [676.260423] grant_pending_locks+0x166/0x1b0 [676.261007] _unlock_lock+0x75/0x90 [676.261469] unlock_lock.isra.57+0x62/0xa0 [676.262009] dlm_unlock+0x21e/0x330 [676.262457] ? lock_torture_stats+0x80/0x80 [dlm_locktorture] [676.263183] torture_unlock+0x5a/0x90 [dlm_locktorture] [676.263815] ? preempt_count_sub+0xba/0x100 [676.264361] ? complete+0x1d/0x60 [676.264777] lock_torture_writer+0xb8/0x150 [dlm_locktorture] [676.265555] kthread+0x10a/0x130 [676.266007] ? kthread_complete_and_exit+0x20/0x20 [676.266616] ret_from_fork+0x22/0x30 [676.267097] </TASK> [676.267381] irq event stamp: 9579855 [676.267824] hardirqs last enabled at (9579863): [<ffffffffffb14e6f8>] __up_console_sem+0x58/0x60 [676.268896] hardirqs last disabled at (9579872): [<ffffffffffb14e6dd>] __up_console_sem+0x3d/0x60 [676.270008] softirqs last enabled at (9579798): [<ffffffffffbc200349>] __do_softirq+0x349/0x4c7 [676.271438] softirqs last disabled at (9579897): [<ffffffffffb0d54c0>] irq_exit_rcu+0xb0/0xf0 [676.272796] ---[end trace 0000000000000000]--- I reproduced this warning with dlm_locktorture test which is currently not upstream. However this patch fix the issue by make a additional refcount between dlm_lowcomms_new_msg() and dlm_lowcomms_commit_msg(). In case of the race the kref_put() in dlm_lowcomms_commit_msg() will be the final put.
CVE-2022-50374	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_{ldisc,serdev}: check percpu_init_rwlock() failure syzbot is reporting NULL pointer dereference at hci_uart_tty_close() [1], for rcu_sync_enter() is called without rcu_sync_init() due to hci_uart_tty_open() ignoring percpu_init_rwlock() failure. While we are at it, fix that hci_uart_register_device() ignores percpu_init_rwlock() failure and hci_uart_unregister_device() does not call percpu_free_rwlock().
CVE-2022-50380	In the Linux kernel, the following vulnerability has been resolved: mm: /proc/pid/smmaps_rollup: fix no vma's null-deref Commit 258f669e7e88 ("mm: /proc/pid/smmaps_rollup: convert to single value seq_file") introduced a null-deref if there are no vma's in the task in show_smmaps_rollup.

CVE ID	Description
CVE-2022-50381	In the Linux kernel, the following vulnerability has been resolved: md: fix a crash in mempool_free There's a crash in mempool_free when running the lvm test shell/lvchange-rebuild-raid.sh. The reason for the crash is this: * super_written calls atomic_dec_and_test(&mddev->pending_writes) and wake_up(&mddev->sb_wait). Then it calls rdev_dec_pending(rdev, mddev) and bio_put(bio). * so, the process that waited on sb_wait and that is woken up is racing with bio_put(bio). * if the process wins the race, it calls bioset_exit before bio_put(bio) is executed. * bio_put(bio) attempts to free a bio into a destroyed bio set - causing a crash in mempool_free. We fix this bug by moving bio_put before atomic_dec_and_test. We also move rdev_dec_pending before atomic_dec_and_test as suggested by Neil Brown. The function md_end_flush has a similar bug - we must call bio_put before we decrement the number of in-progress bios. BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 11557f0067 P4D 11557f0067 PUD 0 Oops: 0002 [#1] PREEMPT SMP CPU: 0 PID: 73 Comm: kworker/0:1 Not tainted 6.1.0-rc3 #5 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-2 04/01/2014 Workqueue: kdelayd flush_expired_bios [dm_delay] RIP: 0010:mempool_free+0x47/0x80 Code: 48 89 ef 5b 5d ff e0 f3 c3 48 89 f7 e8 32 45 3f 00 48 63 53 08 48 89 c6 3b 53 04 7d 2d 48 8b 43 10 8d 4a 01 48 89 df 89 4b 08 <48> 89 2c d0 e8 b0 45 3f 00 48 8d 7b 30 5b 5d 31 c9 ba 01 00 00 00 RSP: 0018:ffff88910036bda8 EFLAGS: 00010093 RAX: 0000000000000000 RBX: ffff8891037b65d8 RCX: 0000000000000001 RDX: 0000000000000000 RSI: 0000000000000202 RDI: ffff8891037b65d8 RBP: ffff8891447ba240 R08: 0000000000012908 R09: 00000000003d0900 R10: 0000000000000000 R11: 000000000173544 R12: ffff889101a14000 R13: ffff8891562ac300 R14: ffff889102b41440 R15: ffff889100000005 FS: 0000000000000000(0000) GS:ffff88942fa00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000001102e9900 CR4: 000000000000006b Call Trace: <TASK> clone_endio+0xf4/0x1c0 [dm_mod] clone_endio+0xf4/0x1c0 [dm_mod] __submit_bio+0x76/0x120 submit_bio_noacct_noccheck+0xb6/0x2a0 flush_expired_bios+0x28/0x2f [dm_delay] process_one_work+0x1b4/0x300 worker_thread+0x45/0x3e0 ? rescuer_thread+0x380/0x380 kthread+0xc2/0x100 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x1f/0x30 </TASK> Modules linked in: brd dm_delay dm_raid dm_mod af_packet uvesafb cfbfillrect cfbimgblt cn cfbcopyarea fb font fbdev tun autofs4 binfmt_misc configs ipv6 virtio_rng virtio_balloon rng_core virtio_net pcspkr net_failover failover qemu_fw_cfg button mousedev raid10 raid456 libcrc32c async_raid6_recov async_memcpy async_pq raid6_pq async_xor xor async_tx raid1 raid0 md_mod sd_mod t10_pi crc64_rocksoft crc64 virtio_scsi scsi_mod evdev psmouse bsg scsi_common [last unloaded: brd] CR2: 0000000000000000 ---[end trace 0000000000000000]---
CVE-2022-50382	In the Linux kernel, the following vulnerability has been resolved: padata: Always leave BHs disabled when running ->parallel() A deadlock can happen when an overloaded system runs ->parallel() in the context of the current task: padata_do_parallel ->parallel() pccrypt_aead_enc/dec padata_do_serial spin_lock(&reorder->lock) // BHs still enabled <interrupt> ... __do_softirq ... padata_do_serial spin_lock(&reorder->lock) It's a bug for BHs to be on in _do_serial as Steffen points out, so ensure they're off in the "current task" case like they are in padata_parallel_worker to avoid this situation.
CVE-2022-50385	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix an Oops in nfs_d_automount() When mounting from a NFSv4 referral, path->dentry can end up being a negative dentry, so derive the struct nfs_server from the dentry itself instead.
CVE-2022-50386	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix user-after-free This uses l2cap_chan_hold_unless_zero() after calling __l2cap_get_chan_blah() to prevent the following trace: Bluetooth: l2cap_core.c:static void l2cap_chan_destroy(struct kref *kref) Bluetooth: chan 0000000023c4974d Bluetooth: parent 00000000ae861c08 ===== BUG: KASAN: use-after-free in __mutex_waiter_is_first kernel/locking/mutex.c:191 [inline] BUG: KASAN: use-after-free in __mutex_lock_common kernel/locking/mutex.c:671 [inline] BUG: KASAN: use-after-free in __mutex_lock+0x278/0x400 kernel/locking/mutex.c:729 Read of size 8 at addr ffff888006a49b08 by task kworker/u3:2/389

CVE ID	Description
CVE-2022-50388	In the Linux kernel, the following vulnerability has been resolved: nvme: fix multipath crash caused by flush request when blktrace is enabled The flush request initialized by blk_kick_flush has NULL bio, and it may be dealt with nvme_end_req during io completion. When blktrace is enabled, nvme_trace_bio_complete with multipath activated trying to access NULL pointer bio from flush request results in the following crash: [2517.831677] BUG: kernel NULL pointer dereference, address: 000000000000001a [2517.835213] #PF: supervisor read access in kernel mode [2517.838724] #PF: error_code(0x0000) - not-present page [2517.842222] PGD 7b2d51067 P4D 0 [2517.845684] Oops: 0000 [#1] SMP NOPTI [2517.849125] CPU: 2 PID: 732 Comm: kworker/2:1H Kdump: loaded Tainted: G S 5.15.67-0.cl9.x86_64 #1 [2517.852723] Hardware name: XFUSION 2288H V6/BC13MBSBC, BIOS 1.13 07/27/2022 [2517.856358] Workqueue: nvme_tcp_wq nvme_tcp_io_work [nvme_tcp] [2517.859993] RIP: 0010:blk_add_trace_bio_complete+0x6/0x30 [2517.863628] Code: 1f 44 00 00 48 8b 46 08 31 c9 ba 04 00 10 00 48 8b 80 50 03 00 00 48 8b 78 50 e9 e5 fe ff ff 0f 1f 44 00 00 41 54 49 89 f4 55 <0f> b6 7a 1a 48 89 d5 e8 3e 1c 2b 00 48 89 ee 4c 89 e7 5d 89 c1 ba [2517.871269] RSP: 0018:ff7f6a008d9dbcd0 EFLAGS: 00010286 [2517.875081] RAX: ff3d5b4be00b1d50 RBX: 000000002040002 RCX: ff3d5b0a270f2000 [2517.878966] RDX: 0000000000000000 RSI: ff3d5b0b021fb9f8 RDI: 0000000000000000 [2517.882849] RBP: ff3d5b0b96a6fa00 R08: 0000000000000001 R09: 0000000000000000 [2517.886718] R10: 0000000000000000 R11: 0000000000000000 R12: ff3d5b0b021fb9f8 [2517.890575] R13: 0000000002000000 R14: ff3d5b0b021fb1b0 R15: 0000000000000018 [2517.894434] FS: 0000000000000000(0000) GS:ff3d5b42bfc80000(0000) knlGS:0000000000000000 [2517.898299] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [2517.902157] CR2: 000000000000001a CR3: 00000004f023e005 CR4: 0000000000771ee0 [2517.906053] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [2517.909930] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [2517.913761] PKRU: 55555554 [2517.917558] Call Trace: [2517.921294] <TASK> [2517.924982] nvme_complete_rq+0x1c3/0x1e0 [nvme_core] [2517.928715] nvme_tcp_recv_pdu+0x4d7/0x540 [nvme_tcp] [2517.932442] nvme_tcp_recv_skb+0x4f/0x240 [nvme_tcp] [2517.936137] ? nvme_tcp_recv_pdu+0x540/0x540 [nvme_tcp] [2517.939830] tcp_read_sock+0x9c/0x260 [2517.943486] nvme_tcp_try_recv+0x65/0xa0 [nvme_tcp] [2517.947173] nvme_tcp_io_work+0x64/0x90 [nvme_tcp] [2517.950834] process_one_work+0x1e8/0x390 [2517.954473] worker_thread+0x53/0x3c0 [2517.958069] ? process_one_work+0x390/0x390 [2517.961655] kthread+0x10c/0x130 [2517.965211] ? set_kthread_struct+0x40/0x40 [2517.968760] ret_from_fork+0x1f/0x30 [2517.972285] </TASK> To avoid this situation, add a NULL check for req->bio before calling trace_block_bio_complete.
CVE-2022-50389	In the Linux kernel, the following vulnerability has been resolved: tpm: tpm_crb: Add the missed acpi_put_table() to fix memory leak In crb_acpi_add(), we get the TPM2 table to retrieve information like start method, and then assign them to the priv data, so the TPM2 table is not used after the init, should be freed, call acpi_put_table() to fix the memory leak.

CVE ID	Description
CVE-2022-50390	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: fix undefined behavior in bit shift for TTM_TT_FLAG_PRIV_POPULATED Shifting signed 32-bit value by 31 bits is undefined, so changing significant bit to unsigned. The UBSAN warning calltrace like below: UBSAN: shift-out-of-bounds in ./include/drm/ttm/ttm_tt.h:122:26 left shift of 1 by 31 places cannot be represented in type 'int' Call Trace: <TASK> dump_stack_lvl+0x7d/0xa5 dump_stack+0x15/0x1b ubsan_epilogue+0xe/0x4e __ubsan_handle_shift_out_of_bounds+0x1e7/0x20c ttm_bo_move_memcpy+0x3b4/0x460 [ttm] bo_driver_move+0x32/0x40 [drm_vram_helper] ttm_bo_handle_move_mem+0x118/0x200 [ttm] ttm_bo_validate+0xfa/0x220 [ttm] drm_gem_vram_pin_locked+0x70/0x1b0 [drm_vram_helper] drm_gem_vram_pin+0x48/0xb0 [drm_vram_helper] drm_gem_vram_plane_helper_prepare_fb+0x53/0xe0 [drm_vram_helper] drm_gem_vram_simple_display_pipe_prepare_fb+0x26/0x30 [drm_vram_helper] drm_simple_kms_plane_prepare_fb+0x4d/0xe0 [drm_kms_helper] drm_atomic_helper_prepare_planes+0xda/0x210 [drm_kms_helper] drm_atomic_helper_commit+0xc3/0x1e0 [drm_kms_helper] drm_atomic_commit+0x9c/0x160 [drm] drm_client_modeset_commit_atomic+0x33a/0x380 [drm] drm_client_modeset_commit_locked+0x77/0x220 [drm] drm_client_modeset_commit+0x31/0x60 [drm] __drm_fb_helper_restore_fbdev_mode_unlocked+0xa7/0x170 [drm_kms_helper] drm_fb_helper_set_par+0x51/0x90 [drm_kms_helper] fbcon_init+0x316/0x790 visual_init+0x113/0x1d0 do_bind_con_driver+0x2a3/0x5c0 do_take_over_console+0xa9/0x270 do_fbcon_takeover+0xa1/0x170 do_fb_registered+0x2a8/0x340 fbcon_fb_registered+0x47/0xe0 register_framebuffer+0x294/0x4a0 __drm_fb_helper_initial_config_and_unlock+0x43c/0x880 [drm_kms_helper] drm_fb_helper_initial_config+0x52/0x80 [drm_kms_helper] drm_fbdev_client_hotplug+0x156/0x1b0 [drm_kms_helper] drm_fbdev_generic_setup+0xfc/0x290 [drm_kms_helper] bochs_pci_probe+0x6ca/0x772 [bochs] local_pci_probe+0x4d/0xb0 pci_device_probe+0x119/0x320 really_probe+0x181/0x550 __driver_probe_device+0xc6/0x220 driver_probe_device+0x32/0x100 __driver_attach+0x195/0x200 bus_for_each_dev+0xbb/0x120 driver_attach+0x27/0x30 bus_add_driver+0x22e/0x2f0 driver_register+0xa9/0x190 __pci_register_driver+0x90/0xa0 bochs_pci_driver_init+0x52/0x1000 [bochs] do_one_initcall+0x76/0x430 do_init_module+0x61/0x28a load_module+0x1f82/0x2e50 __do_sys_finit_module+0xf8/0x190 __x64_sys_finit_module+0x23/0x30 do_syscall_64+0x58/0x80 entry_SYSCALL_64_after_hwframe+0x63/0xcd </TASK>
CVE-2022-50394	In the Linux kernel, the following vulnerability has been resolved: i2c: ismt: Fix an out-of-bounds bug in ismt_access() When the driver does not check the data from the user, the variable 'data->block[0]' may be very large to cause an out-of-bounds bug. The following log can reveal it: [33.995542] i2c i2c-1: ioctl, cmd=0x720, arg=0x7ffcb3dc3a20 [33.995978] ismt_smbus 0000:00:05.0: I2C_SMBUS_BLOCK_DATA: WRITE [33.996475] ===== [33.996995] BUG: KASAN: out-of-bounds in ismt_access.cold+0x374/0x214b [33.997473] Read of size 18446744073709551615 at addr ffff88810efcfdb1 by task ismt_poc/485 [33.999450] Call Trace: [34.001849] memcpy+0x20/0x60 [34.002077] ismt_access.cold+0x374/0x214b [34.003382] __i2c_smbus_xfer+0x44f/0xfb0 [34.004007] i2c_smbus_xfer+0x10a/0x390 [34.004291] i2cdev_ioctl_smbus+0x2c8/0x710 [34.005196] i2cdev_ioctl+0x5ec/0x74c Fix this bug by checking the size of 'data->block[0]' first.
CVE-2022-50395	In the Linux kernel, the following vulnerability has been resolved: integrity: Fix memory leakage in keyring allocation error path Key restriction is allocated in integrity_init_keyring(). However, if keyring allocation failed, it is not freed, causing memory leaks.

CVE ID	Description
CVE-2022-50401	In the Linux kernel, the following vulnerability has been resolved: nfsd: under NFSv4.1, fix double svc_xprt_put on rpc_create failure On error situation `clp->cl_cb_conn.cb_xprt` should not be given a reference to the xprt otherwise both client cleanup and the error handling path of the caller call to put it. Better to delay handing over the reference to a later branch. [72.530665] refcount_t: underflow; use-after-free. [72.531933] WARNING: CPU: 0 PID: 173 at lib/refcount.c:28 refcount_warn_saturate+0xc0/0x120 [72.533075] Modules linked in: nfsd(OE) nfsv4(OE) nfsv3(OE) nfs(OE) lockd(OE) compat_nfs_ssc(OE) nfs_acl(OE) rpcsec_gss_krb5(OE) auth_rpcgss(OE) rpcrdma(OE) dns_resolver fscache netfs grace rdma_cm iw_cm ib_cm sunrpc(OE) mlx5_ib mlx5_core mlxfw pci_hyperv_intf ib_uverbs ib_core xt_MASQUERADE nf_conntrack_netlink nft_counter xt_addrtype nft_compat br_netfilter bridge stp llc nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 ip_set overlay nf_tables nfnetlink crct10dif_pclmul crc32_pclmul ghash_clmulni_intel xfs serio_raw virtio_net virtio_blk net_failover failover fuse [last unloaded: sunrpc] [72.540389] CPU: 0 PID: 173 Comm: kworker/u16:5 Tainted: G OE 5.15.82-dan #1 [72.541511] Hardware name: Red Hat KVM/RHEL-AV, BIOS 1.16.0-3.module+el8.7.0+1084+97b81f61 04/01/2014 [72.542717] Workqueue: nfsd4_callbacks nfsd4_run_cb_work [nfsd] [72.543575] RIP: 0010:refcount_warn_saturate+0xc0/0x120 [72.544299] Code: 55 00 0f 0b 5d e9 01 50 98 00 80 3d 75 9e 39 08 00 0f 85 74 ff ff ff 48 c7 c7 e8 d1 60 8e c6 05 61 9e 39 08 01 e8 f6 51 55 00 <0f> 0b 5d e9 d9 4f 98 00 80 3d 4b 9e 39 08 00 0f 85 4c ff ff ff 48 [72.546666] RSP: 0018:ffffb3f841157cf0 EFLAGS: 00010286 [72.547393] RAX: 0000000000000026 RBX: ffff89ac6231d478 RCX: 0000000000000000 [72.548324] RDX: ffff89adb7c2c2c0 RSI: ffff89adb7c205c0 RDI: ffff89adb7c205c0 [72.549271] RBP: ffff89adb7c205c0 R08: 0000000000000000 R09: c0000000ffffffffff [72.550209] R10: 0000000000000001 R11: ffff89adb7c2c2c0 R12: ffff89ac6231d180 [72.551142] R13: ffff89ac6231d478 R14: ffff89ac40c06180 R15: ffff89ac6231d4b0 [72.552089] FS: 0000000000000000(0000) GS: ffff89adb7c00000(0000) knlGS:0000000000000000 [72.553175] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [72.553934] CR2: 0000563a310506a8 CR3: 0000000109a66000 CR4: 00000000000350ef0 [72.554874] Call Trace: [72.555278] <TASK> [72.555614] svc_xprt_put+0xaf/0xe0 [sunrpc] [72.556276] nfsd4_process_cb_update.isra.11+0xb7/0x410 [nfsd] [72.557087] ? update_load_avg+0x82/0x610 [72.557652] ? cpuacct_charge+0x60/0x70 [72.558212] ? dequeue_entity+0xdb/0x3e0 [72.558765] ? queued_spin_unlock+0x9/0x20 [72.559358] nfsd4_run_cb_work+0xfc/0x270 [nfsd] [72.560031] process_one_work+0x1df/0x390 [72.560600] worker_thread+0x37/0x3b0 [72.561644] ? process_one_work+0x390/0x390 [72.562247] kthread+0x12f/0x150 [72.562710] ? set_kthread_struct+0x50/0x50 [72.563309] ret_from_fork+0x22/0x30 [72.563818] </TASK> [72.564189] ---[end trace 031117b1c72ec616]--- [72.566019] list_add corruption. next->prev should be prev (ffff89ac4977e538), but was ffff89ac4763e018. (next=ffff89ac4763e018). [72.567647] -----[cut here]-----
CVE-2022-50402	In the Linux kernel, the following vulnerability has been resolved: drivers/md/md-bitmap: check the return value of md_bitmap_get_counter() Check the return value of md_bitmap_get_counter() in case it returns NULL pointer, which will result in a null pointer dereference. v2: update the check to include other dereference
CVE-2022-50409	In the Linux kernel, the following vulnerability has been resolved: net: If sock is dead don't access sock's sk_wq in sk_stream_wait_memory Fixes the below NULL pointer dereference: [...] [14.471200] Call Trace: [14.471562] <TASK> [14.471882] lock_acquire+0x245/0x2e0 [14.472416] ? remove_wait_queue+0x12/0x50 [14.473014] ? _raw_spin_lock_irqsave+0x17/0x50 [14.473681] _raw_spin_lock_irqsave+0x3d/0x50 [14.474318] ? remove_wait_queue+0x12/0x50 [14.474907] remove_wait_queue+0x12/0x50 [14.475480] sk_stream_wait_memory+0x20d/0x340 [14.476127] ? do_wait_intr_irq+0x80/0x80 [14.476704] do_tcp_sendpages+0x287/0x600 [14.477283] tcp_bpf_push+0xab/0x260 [14.477817] tcp_bpf_sendmsg_redir+0x297/0x500 [14.478461] ? __local_bh_enable_ip+0x77/0xe0 [14.479096] tcp_bpf_send_verdict+0x105/0x470 [14.479729] tcp_bpf_sendmsg+0x318/0x4f0 [14.480311] sock_sendmsg+0x2d/0x40 [14.480822] __sys_sendmsg+0x1b4/0x1c0 [14.481390] ? copy_msghdr_from_user+0x62/0x80 [14.482048] __sys_sendmsg+0x78/0xb0 [14.482580] ? vmf_insert_pfn_prot+0x91/0x150 [14.483215] ? __do_fault+0x2a/0x1a0 [14.483738] ? do_fault+0x15e/0x5d0 [14.484246] ? __handle_mm_fault+0x56b/0x1040 [14.484874] ? lock_is_held_type+0xdf/0x130 [14.485474] ? find_held_lock+0x2d/0x90 [14.486046] ? __sys_sendmsg+0x41/0x70 [14.486587] __sys_sendmsg+0x41/0x70 [14.487105] ? intel_pmu_drain_pebs_core+0x350/0x350 [14.487822] do_syscall_64+0x34/0x80 [14.488345] entry_SYSCALL_64_after_hwframe+0x63/0xcd [...] The test scenario has the following flow: thread1 thread2 ----- tcp_bpf_sendmsg tcp_bpf_send_verdict tcp_bpf_sendmsg_redir sock_close tcp_bpf_push_locked __sock_release tcp_bpf_push //inet_release do_tcp_sendpages sock->ops->release sk_stream_wait_memory // tcp_close sk_wait_event sk->sk_prot->close release_sock(__sk); *** lock_sock(sk); __tcp_close sock_orphan(sk) sk->sk_wq = NULL release_sock ***** lock_sock(__sk); remove_wait_queue(sk_sleep(sk), &wait); sk_sleep(sk) //NULL pointer dereference &rcu_dereference_raw(sk->sk_wq)->wait While waiting for memory in thread1, the socket is released with its wait queue because thread2 has closed it. This caused by tcp_bpf_send_verdict didn't increase the f_count of psock->sk_redir->sk_socket->file in thread1. We should check if SOCK_DEAD flag is set on wakeup in sk_stream_wait_memory before accessing the wait queue.

CVE ID	Description
CVE-2022-50411	In the Linux kernel, the following vulnerability has been resolved: ACPICA: Fix error code path in <code>acpi_ds_call_control_method()</code> A use-after-free in <code>acpi_ps_parse_aml()</code> after a failing invocation of <code>acpi_ds_call_control_method()</code> is reported by KASAN [1] and code inspection reveals that <code>next_walk_state</code> pushed to the thread by <code>acpi_ds_create_walk_state()</code> is freed on errors, but it is not popped from the thread beforehand. Thus <code>acpi_ds_get_current_walk_state()</code> called by <code>acpi_ps_parse_aml()</code> subsequently returns it as the new walk state which is incorrect. To address this, make <code>acpi_ds_call_control_method()</code> call <code>acpi_ds_pop_walk_state()</code> to pop <code>next_walk_state</code> from the thread before returning an error.
CVE-2022-50412	In the Linux kernel, the following vulnerability has been resolved: drm: bridge: adv7511: unregister cec i2c device after cec adapter <code>cec_unregister_adapter()</code> assumes that the underlying adapter ops are callable. For example, if the CEC adapter currently has a valid physical address, then the unregistration procedure will invalidate the physical address by setting it to <code>f.f.f.f</code> . Whence the following kernel oops observed after removing the <code>adv7511</code> module: Unable to handle kernel execution of user memory at virtual address 0000000000000000 Internal error: Oops: 86000004 [#1] PREEMPT_RT SMP Call trace: 0x0 adv7511_cec_adap_log_addr+0x1ac/0x1c8 [adv7511] cec_adap_unconfigure+0x44/0x90 [cec] __cec_s_phys_addr.part.0+0x68/0x230 [cec] __cec_s_phys_addr+0x40/0x50 [cec] cec_unregister_adapter+0xb4/0x118 [cec] adv7511_remove+0x60/0x90 [adv7511] i2c_device_remove+0x34/0xe0 device_release_driver_internal+0x114/0x1f0 driver_detach+0x54/0xe0 bus_remove_driver+0x60/0xd8 driver_unregister+0x34/0x60 i2c_del_driver+0x2c/0x68 adv7511_exit+0x1c/0x67c [adv7511] __arm64_sys_delete_module+0x154/0x288 invoke_syscall+0x48/0x100 el0_svc_common.constprop.0+0x48/0xe8 do_el0_svc+0x28/0x88 el0_svc+0x1c/0x50 el0t_64_sync_handler+0xa8/0xb0 el0t_64_sync+0x15c/0x160 Code: bad PC value ---[end trace 0000000000000000]--- Protect against this scenario by unregistering <code>i2c_cec</code> after unregistering the CEC adapter. Duly disable the CEC clock afterwards too.
CVE-2022-50419	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sysfs: Fix attempting to call <code>device_add</code> multiple times <code>device_add</code> shall not be called multiple times as stated in its documentation: 'Do not call this routine or <code>device_register()</code> more than once for any device structure' Syzkaller reports a bug as follows [1]: -----[cut here]----- kernel BUG at lib/list_debug.c:33! invalid opcode: 0000 [#1] PREEMPT SMP KASAN [...] Call Trace: <TASK> __list_add include/linux/list.h:69 [inline] list_add_tail include/linux/list.h:102 [inline] kobj_kset_join lib/kobject.c:164 [inline] kobject_add_internal+0x18f/0x8f0 lib/kobject.c:214 kobject_add_varg lib/kobject.c:358 [inline] kobject_add+0x150/0x1c0 lib/kobject.c:410 device_add+0x368/0x1e90 drivers/base/core.c:3452 hci_conn_add_sysfs+0x9b/0x1b0 net/bluetooth/hci_sysfs.c:53 hci_le_cis_established_evt+0x57c/0xae0 net/bluetooth/hci_event.c:6799 hci_le_meta_evt+0x2b8/0x510 net/bluetooth/hci_event.c:7110 hci_event_func net/bluetooth/hci_event.c:7440 [inline] hci_event_packet+0x63d/0xf0 net/bluetooth/hci_event.c:7495 hci_rx_work+0xae7/0x1230 net/bluetooth/hci_core.c:4007 process_one_work+0x991/0x1610 kernel/workqueue.c:2289 worker_thread+0x665/0x1080 kernel/workqueue.c:2436 kthread+0x2e4/0x3a0 kernel/kthread.c:376 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:306 </TASK>
CVE-2022-50422	In the Linux kernel, the following vulnerability has been resolved: scsi: libsas: Fix use-after-free bug in <code>smp_execute_task_sg()</code> When executing SMP task failed, the <code>smp_execute_task_sg()</code> calls <code>del_timer()</code> to delete " <code>slow_task->timer</code> ". However, if the timer handler <code>sas_task_internal_timedout()</code> is running, the <code>del_timer()</code> in <code>smp_execute_task_sg()</code> will not stop it and a UAF will happen. The process is shown below: (thread 1) (thread 2) <code>smp_execute_task_sg()</code> <code>sas_task_internal_timedout()</code> ... <code>del_timer()</code> <code>sas_free_task(task)</code> <code>kfree(task->slow_task)</code> //FREE <code>task->slow_task->...</code> //USE Fix by calling <code>del_timer_sync()</code> in <code>smp_execute_task_sg()</code> , which makes sure the timer handler have finished before the " <code>task->slow_task</code> " is deallocated.
CVE-2022-50423	In the Linux kernel, the following vulnerability has been resolved: ACPICA: Fix use-after-free in <code>acpi_ut_copy_ipackage_to_ipackage()</code> There is an use-after-free reported by KASAN: BUG: KASAN: use-after-free in <code>acpi_ut_remove_reference+0x3b/0x82</code> Read of size 1 at addr ffff888112afc460 by task modprobe/2111 CPU: 0 PID: 2111 Comm: modprobe Not tainted 6.1.0-rc7-dirty Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), Call Trace: <TASK> kasan_report+0xae/0xe0 acpi_ut_remove_reference+0x3b/0x82 acpi_ut_copy_iobject_to_iobject+0x3be/0x3d5 acpi_ds_store_object_to_local+0x15d/0x3a0 acpi_ex_store+0x78d/0x7fd acpi_ex_opcode_1A_1T_1R+0xbe4/0xf9b acpi_ps_parse_aml+0x217/0x8d5 ... </TASK> The root cause of the problem is that the <code>acpi_operand_object</code> is freed when <code>acpi_ut_walk_package_tree()</code> fails in <code>acpi_ut_copy_ipackage_to_ipackage()</code> , lead to repeated release in <code>acpi_ut_copy_iobject_to_iobject()</code> . The problem was introduced by "8aa5e56eeb61" commit, this commit is to fix memory leak in <code>acpi_ut_copy_iobject_to_iobject()</code> , repeatedly adding remove operation, lead to " <code>acpi_operand_object</code> " used after free. Fix it by removing <code>acpi_ut_remove_reference()</code> in <code>acpi_ut_copy_ipackage_to_ipackage()</code> . <code>acpi_ut_copy_ipackage_to_ipackage()</code> is called to copy an internal package object into another internal package object, when it fails, the memory of <code>acpi_operand_object</code> should be freed by the caller.

CVE ID	Description
CVE-2022-50424	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921: resource leaks at mt7921_check_offload_capability() Fixed coverity issue with resource leaks at variable "fw" going out of scope leaks the storage it points to mt7921_check_offload_capability(). Addresses-Coverity-ID: 1527806 ("Resource leaks")
CVE-2022-50425	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Fix copy_xstate_to_uabi() to copy init states correctly When an extended state component is not present in fpstate, but in init state, the function copies from init_fpstate via copy_feature(). But, dynamic states are not present in init_fpstate because of all-zeros init states. Then retrieving them from init_fpstate will explode like this: BUG: kernel NULL pointer dereference, address: 0000000000000000 ... RIP: 0010:memcpy_erms+0x6/0x10 ? __copy_xstate_to_uabi_buf+0x381/0x870 fpu_copy_guest_fpstate_to_uabi+0x28/0x80 kvm_arch_vcpu_ioctl+0x14c/0x1460 [kvm] ? __this_cpu_preempt_check+0x13/0x20 ? vmx_vcpu_put+0x2e/0x260 [kvm_intel] kvm_vcpu_ioctl+0xea/0x6b0 [kvm] ? kvm_vcpu_ioctl+0xea/0x6b0 [kvm] ? __fget_light+0xd4/0x130 __x64_sys_ioctl+0xe3/0x910 ? debug_smp_processor_id+0x17/0x20 ? fpregs_assert_state_consistent+0x27/0x50 do_syscall_64+0x3f/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd Adjust the 'mask' to zero out the userspace buffer for the features that are not available both from fpstate and from init_fpstate. The dynamic features depend on the compacted XSAVE format. Ensure it is enabled before reading XCOMP_BV in init_fpstate.
CVE-2022-50427	In the Linux kernel, the following vulnerability has been resolved: ALSA: ac97: fix possible memory leak in snd_ac97_dev_register() If device_register() fails in snd_ac97_dev_register(), it should call put_device() to give up reference, or the name allocated in dev_set_name() is leaked.
CVE-2022-50431	In the Linux kernel, the following vulnerability has been resolved: ALSA: aoa: i2sbus: fix possible memory leak in i2sbus_add_dev() dev_set_name() in soundbus_add_one() allocates memory for name, it need be freed when of_device_register() fails, call soundbus_dev_put() to give up the reference that hold in device_initialize(), so that it can be freed in kobject_cleanup() when the refcount hit to 0. And other resources are also freed in i2sbus_release_dev(), so it can return 0 directly.

CVE ID	Description
CVE-2022-50432	In the Linux kernel, the following vulnerability has been resolved: kernfs: fix use-after-free in <code>__kernfs_remove</code> Syzkaller managed to trigger concurrent calls to <code>kernfs_remove_by_name_ns()</code> for the same file resulting in a KASAN detected use-after-free. The race occurs when the root node is freed during <code>kernfs_drain()</code>. To prevent this acquire an additional reference for the root of the tree that is removed before calling <code>__kernfs_remove()</code>. Found by syzkaller with the following reproducer (slab_nomerge is required): <code>syz_mount_image\$ext4(0x0, &(0x7f0000000100)='./file0\x00', 0x100000, 0x0, 0x0, 0x0, 0x0) r0 = openat(0xffffffffffff9c, &(0x7f0000000080)='./proc/self/exe\x00', 0x0, 0x0) close(r0) pipe2(&(0x7f0000000140)={0xfffffffffffff, <r1=>0xfffffffffffff}, 0x800) mount\$9p_fd(0x0, &(0x7f0000000040)='./file0\x00', &(0x7f00000000c0), 0x408, &(0x7f0000000280)={'trans=fd', {'rfdno', 0x3d, r0}, 0x2c, {'wfdno', 0x3d, r1}, 0x2c, [{'@cache_loose'}, {'@mmap'}, {'@loose'}, {'@loose'}, {'@mmap'}], [{'@mask={'mask', 0x3d, '^MAY_EXEC'}}, {'@fsmagic={'fsmagic', 0x3d, 0x10001}}, {'@dont_hash'}]}) Sample report: ===== BUG: KASAN: use-after-free in kernfs_type include/linux/kernfs.h:335 [inline] BUG: KASAN: use-after-free in kernfs_leftmost_descendant fs/kernfs/dir.c:1261 [inline] BUG: KASAN: use-after-free in __kernfs_remove.part.0+0x843/0x960 fs/kernfs/dir.c:1369 Read of size 2 at addr ffff8880088807f0 by task syz-executor.2/857 CPU: 0 PID: 857 Comm: syz-executor.2 Not tainted 6.0.0-rc3-00363-g7726d4c3e60b #5 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl +0x6e/0x91 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:317 [inline] print_report.cold+0x5e/0x5e5 mm/kasan/report.c:433 kasan_report+0xa3/0x130 mm/kasan/report.c:495 kernfs_type include/linux/kernfs.h:335 [inline] kernfs_leftmost_descendant fs/kernfs/dir.c:1261 [inline] __kernfs_remove.part.0+0x843/0x960 fs/kernfs/dir.c:1369 __kernfs_remove fs/kernfs/dir.c:1356 [inline] kernfs_remove_by_name_ns+0x108/0x190 fs/kernfs/dir.c:1589 sysfs_slab_add+0x133/0x1e0 mm/slub.c:5943 __kmem_cache_create+0x3e0/0x550 mm/slub.c:4899 create_cache mm/slub_common.c:229 [inline] kmem_cache_create_usercopy+0x167/0x2a0 mm/slub_common.c:335 p9_client_create +0xd4d/0x1190 net/9p/client.c:993 v9fs_session_init+0x1e6/0x13c0 fs/9p/v9fs.c:408 v9fs_mount +0xb9/0xbd0 fs/9p/vfs_super.c:126 legacy_get_tree+0xf1/0x200 fs/fs_context.c:610 vfs_get_tree +0x85/0x2e0 fs/super.c:1530 do_new_mount fs/namespace.c:3040 [inline] path_mount+0x675/0x1d00 fs/namespace.c:3370 do_mount fs/namespace.c:3383 [inline] __do_sys_mount fs/namespace.c:3591 [inline] __se_sys_mount fs/namespace.c:3568 [inline] __x64_sys_mount+0x282/0x300 fs/namespace.c:3568 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x38/0x90 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7f725f983aed Code: 02 b8 ff ff ff c3 66 0f 1f 44 00 00 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f725f0f7028 EFLAGS: 00000246 ORIG_RAX: 00000000000000a5 RAX: ffffffffda RBX: 00007f725faa3f80 RCX: 00007f725f983aed RDX: 00000000200000c0 RSI: 0000000020000040 RDI: 0000000000000000 RBP: 00007f725f9f419c R08: 0000000020000280 R09: 0000000000000000 R10: 0000000000000408 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000006 R14: 00007f725faa3f80 R15: 00007f725f0d7000 </TASK> Allocated by task 855: kasan_save_stack+0x1e/0x40 mm/kasan/common.c:38 kasan_set_track mm/kasan/common.c:45 [inline] set_alloc_info mm/kasan/common.c:437 [inline] __kasan_slab_alloc+0x66/0x80 mm/kasan/common.c:470 kasan_slab_alloc include/linux/kasan.h:224 [inline] slab_post_alloc_hook mm/slub.h:7 ---truncated---</code>
CVE-2022-50434	In the Linux kernel, the following vulnerability has been resolved: blk-mq: fix possible memleak when register 'hctx' failed There's issue as follows when do fault injection test: unreferenced object 0xffff888132a9f400 (size 512): comm "insmod", pid 308021, jiffies 4324277909 (age 509.733s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 08 f4 a9 32 81 88 ff ff 2.... 08 f4 a9 32 81 88 ff ff 00 00 00 00 00 00 00 00 ...2..... backtrace: [<00000000e8952bb4>] kmalloc_node_trace+0x22/0xa0 [<00000000f9980e0f>] blk_mq_alloc_and_init_hctx+0x3f1/0x7e0 [<000000002e719efa>] blk_mq_realloc_hw_ctxs+0x1e6/0x230 [<000000004f1fda40>] blk_mq_init_allocated_queue+0x27e/0x910 [<00000000287123ec>] __blk_mq_alloc_disk +0x67/0xf0 [<00000000a2a34657>] 0xfffffffffa2ad310f [<00000000b173f718>] 0xfffffffffa2af824a [<00000000095a1dabb>] do_one_initcall+0x87/0x2a0 [<00000000f32fd93>] do_init_module +0xdf/0x320 [<00000000cbe8541e>] load_module+0x3006/0x3390 [<0000000069ed1bdb>] __do_sys_finit_module+0x113/0x1b0 [<00000000a1a29ae8>] do_syscall_64+0x35/0x80 [<000000009cd878b0>] entry_SYSCALL_64_after_hwframe+0x46/0xb0 Fault injection context as follows: kobject_add blk_mq_register_hctx blk_mq_sysfs_register blk_register_queue device_add_disk null_add_dev.part.0 [null_blk] As 'blk_mq_register_hctx' may already add some objects when failed halfway, but there isn't do fallback, caller don't know which objects add failed. To solve above issue just do fallback when add objects failed halfway in 'blk_mq_register_hctx'.

CVE ID	Description
CVE-2022-50435	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid crash when inline data creation follows DIO write When inode is created and written to using direct IO, there is nothing to clear the EXT4_STATE_MAY_INLINE_DATA flag. Thus when inode gets truncated later to say 1 byte and written using normal write, we will try to store the data as inline data. This confuses the code later because the inode now has both normal block and inline data allocated and the confusion manifests for example as: kernel BUG at fs/ext4/inode.c:2721! invalid opcode: 0000 [#1] PREEMPT SMP KASAN CPU: 0 PID: 359 Comm: repro Not tainted 5.19.0-rc8-00001-g31ba1e3b8305-dirty #15 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.0-1.fc36 04/01/2014 RIP: 0010:ext4_writepages+0x363d/0x3660 RSP: 0018:ffff90000ccf260 EFLAGS: 00010293 RAX: ffffffff81e1abcd RBX: 0000080000000000 RCX: ffff88810842a180 RDX: 0000000000000000 RSI: 0000080000000000 RDI: 0000000000000000 RBP: ffff90000ccf650 R08: ffffffff81e17d58 R09: ffffd10222c680b R10: dfffe910222c680c R11: 1ffff110222c680a R12: ffff888111634128 R13: ffff90000ccf880 R14: 0000084100000000 R15: 0000000000000001 FS: 0000f72635d2640(0000) GS:ffff88811b000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000565243379180 CR3: 000000010aa74000 CR4: 000000000150eb0 Call Trace: <TASK> do_writepages+0x397/0x640 filemap_fdatawrite_wbc+0x151/0x1b0 file_write_and_wait_range+0x1c9/0x2b0 ext4_sync_file+0x19e/0xa00 vfs_fsync_range+0x17b/0x190 ext4_buffered_write_iter+0x488/0x530 ext4_file_write_iter+0x449/0x1b90 vfs_write+0xbcd/0xf40 ksys_write+0x198/0x2c0 __x64_sys_write+0x7b/0x90 do_syscall_64+0x3d/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd </TASK> Fix the problem by clearing EXT4_STATE_MAY_INLINE_DATA when we are doing direct IO write to a file.
CVE-2022-50445	In the Linux kernel, the following vulnerability has been resolved: xfrm: Reinject transport-mode packets through workqueue The following warning is displayed when the tcp6-multi-diffip11 stress test case of the LTP test suite is tested: watchdog: BUG: soft lockup - CPU#0 stuck for 22s! [ns-tcpserver:48198] CPU: 0 PID: 48198 Comm: ns-tcpserver Kdump: loaded Not tainted 6.0.0-rc6+ #39 Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : des3_edc_encrypt+0x27c/0x460 [libdes] lr : 0x3f sp : ffff80000ceaa1b0 x29: ffff80000ceaa1b0 x28: ffff0000df056100 x27: ffff0000e51e5280 x26: ffff80004df75030 x25: ffff0000e51e4600 x24: 000000000000003b x23: 00000000000002080 x22: 000000000000003d x21: 0000000000000038 x20: 0000000080000020 x19: 000000000000000a x18: 0000000000000033 x17: ffff0000e51e4780 x16: ffff80004e2d1448 x15: ffff80004e2d1248 x14: ffff0000e51e4680 x13: ffff80004e2d1348 x12: ffff80004e2d1548 x11: ffff80004e2d1848 x10: ffff80004e2d1648 x9: ffff80004e2d1748 x8: ffff80004e2d1948 x7: 000000000bcaf83d x6: 000000000000001b x5: ffff80004e2d1048 x4: 00000000761bf3bf x3: 000000007f1dd0a3 x2: ffff0000e51e4780 x1: ffff0000e3b9a2f8 x0: 00000000db44e872 Call trace: des3_edc_encrypt+0x27c/0x460 [libdes] crypto_des3_edc_encrypt+0x1c/0x30 [des_generic] crypto_cbc_encrypt+0x148/0x190 crypto_skcipher_encrypt+0x2c/0x40 crypto_authenc_encrypt+0xc8/0xfc [authenc] crypto_aead_encrypt+0x2c/0x40 echainiv_encrypt+0x144/0x1a0 [echainiv] crypto_aead_encrypt+0x2c/0x40 esp6_output_tail+0x1c8/0x5d0 [esp6] esp6_output+0x120/0x278 [esp6] xfrm_output_one+0x458/0x4ec xfrm_output_resume+0x6c/0x1f0 xfrm_output+0xac/0x4ac __xfrm6_output+0x130/0x270 xfrm6_output+0x60/0xec ip6_xmit+0x2ec/0x5bc inet6_csk_xmit+0xbc/0x10c __tcp_transmit_skb+0x460/0x8c0 tcp_write_xmit+0x348/0x890 __tcp_push_pending_frames+0x44/0x110 tcp_rcv_established+0x3c8/0x720 tcp_v6_do_rcv+0xdc/0x4a0 tcp_v6_rcv+0xc24/0xcbb ip6_protocol_deliver_rcu+0xf0/0x574 ip6_input_finish+0x48/0x7c ip6_input+0x48/0xc0 ip6_rcv_finish+0x80/0x9c xfrm_trans_reinject+0xb0/0xf4 tasklet_action_common.constprop.0+0xf8/0x134 tasklet_action+0x30/0x3c __do_softirq+0x128/0x368 do_softirq+0xb4/0xc0 __local_bh_enable_ip+0xb0/0xb4 put_cpu_fpsimd_context+0x40/0x70 kernel_neon_end+0x20/0x40 sha1_base_do_update.constprop.0.isra.0+0x11c/0x140 [sha1_ce] sha1_ce_finup+0x94/0x110 [sha1_ce] crypto_shash_finup+0x34/0xc0 hmac_finup+0x48/0xe0 crypto_shash_finup+0x34/0xc0 shash_digest_unaligned+0x74/0x90 crypto_shash_digest+0x4c/0x9c shash_ahash_digest+0xc8/0xf0 shash_async_digest+0x28/0x34 crypto_ahash_digest+0x48/0xcc crypto_authenc_genivc+0x88/0xcc [authenc] crypto_authenc_encrypt+0xd8/0xfc [authenc] crypto_aead_encrypt+0x2c/0x40 echainiv_encrypt+0x144/0x1a0 [echainiv] crypto_aead_encrypt+0x2c/0x40 esp6_output_tail+0x1c8/0x5d0 [esp6] esp6_output+0x120/0x278 [esp6] xfrm_output_one+0x458/0x4ec xfrm_output_resume+0x6c/0x1f0 xfrm_output+0xac/0x4ac __xfrm6_output+0x130/0x270 xfrm6_output+0x60/0xec ip6_xmit+0x2ec/0x5bc inet6_csk_xmit+0xbc/0x10c __tcp_transmit_skb+0x460/0x8c0 tcp_write_xmit+0x348/0x890 __tcp_push_pending_frames+0x44/0x110 tcp_push+0xb4/0x14c tcp_sendmsg_locked+0x71c/0xb64 tcp_sendmsg+0x40/0x6c inet6_sendmsg+0x4c/0x80 sock_sendmsg+0x5c/0x6c __sys_sendto+0x128/0x15c __arm64_sys_sendto+0x30/0x40 invoke_syscall+0x50/0x120 el0_svc_common.constprop.0+0x170/0x194 do_el0_svc+0x38/0x4c el0_svc+0x28/0xe0 el0t_64_sync_handler+0xbc/0x13c el0t_64_sync+0x180/0x184 Get softirq info by bcc tool: ./softirqs -NT 10 Tracing soft irq event time... Hit Ctrl-C to end. 15:34:34 SOFTIRQ TOTAL_nsecs block 158990 timer 20030920 sched 46577080 net_rx 676746820 tasklet 9906067650 15:34:45 SOFTIRQ TOTAL_nsecs block 86100 sched 38849790 net_rx ---truncated---

CVE ID	Description
CVE-2022-50459	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi: iscsi_tcp: Fix null- ptr-deref while calling getpeername() Fix a NULL pointer crash that occurs when we are freeing the socket at the same time we access it via sysfs. The problem is that: 1. iscsi_sw_tcp_conn_get_param() and iscsi_sw_tcp_host_get_param() take the frwd_lock and do sock_hold() then drop the frwd_lock. sock_hold() does a get on the "struct sock". 2. iscsi_sw_tcp_release_conn() does sockfd_put() which does the last put on the "struct socket" and that does __sock_release() which sets the sock- >ops to NULL. 3. iscsi_sw_tcp_conn_get_param() and iscsi_sw_tcp_host_get_param() then call kernel_getpeername() which accesses the NULL sock->ops. Above we do a get on the "struct sock", but we needed a get on the "struct socket". Originally, we just held the frwd_lock the entire time but in commit bcf3a2953d36 ("scsi: iscsi: iscsi_tcp: Avoid holding spinlock while calling getpeername()") we switched to refcount based because the network layer changed and started taking a mutex in that path, so we could no longer hold the frwd_lock. Instead of trying to maintain multiple refcounts, this just has us use a mutex for accessing the socket in the interface code paths.
CVE-2022-50460	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix xid leak in cifs_flock() If not flock, before return -ENOLCK, should free the xid, otherwise, the xid will be leaked.
CVE-2022-50464	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7915: Fix PCI device refcount leak in mt7915_pci_init_hif2() As comment of pci_get_device() says, it returns a pci_device with its refcount increased. We need to call pci_dev_put() to decrease the refcount. Save the return value of pci_get_device() and call pci_dev_put() to decrease the refcount.
CVE-2022-50467	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix null ndlp ptr dereference in abnormal exit path for GFT_ID An error case exit from lpfc_cmpl_ct_cmd_gft_id() results in a call to lpfc_nlp_put() with a null pointer to a nodelist structure. Changed lpfc_cmpl_ct_cmd_gft_id() to initialize nodelist pointer upon entry.
CVE-2022-50470	In the Linux kernel, the following vulnerability has been resolved: xhci: Remove device endpoints from bandwidth list when freeing the device Endpoints are normally deleted from the bandwidth list when they are dropped, before the virt device is freed. If xHC host is dying or being removed then the endpoints aren't dropped cleanly due to functions returning early to avoid interacting with a non- accessible host controller. So check and delete endpoints that are still on the bandwidth list when freeing the virt device. Solves a list_del corruption kernel crash when unbinding xhci-pci, caused by xhci_mem_cleanup() when it later tried to delete already freed endpoints from the bandwidth list. This only affects hosts that use software bandwidth checking, which currently is only the xHC in intel Panther Point PCH (Ivy Bridge)
CVE-2022-50472	In the Linux kernel, the following vulnerability has been resolved: IB/mad: Don't call to function that might sleep while in atomic context Tracepoints are not allowed to sleep, as such the following splat is generated due to call to ib_query_pkey() in atomic context. WARNING: CPU: 0 PID: 1888000 at kernel/ trace/ring_buffer.c:2492 rb_commit+0xc1/0x220 CPU: 0 PID: 1888000 Comm: kworker/u9:0 Kdump: loaded Tainted: G OE ----- - - 4.18.0-305.3.1.el8.x86_64 #1 Hardware name: Red Hat KVM, BIOS 1.13.0-2.module_el8.3.0+555+a55c8938 04/01/2014 Workqueue: ib-comp-unb-wq ib_cq_poll_work [ib_core] RIP: 0010:rb_commit+0xc1/0x220 RSP: 0000:ffff8a8c80f9bca0 EFLAGS: 00010202 RAX: ffff8951c7c01300 RBX: ffff8951c7c14a00 RCX: 0000000000000246 RDX: ffff8951c707c000 RSI: ffff8951c707c57c RDI: ffff8951c7c14a00 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: ffff8951c7c01300 R11: 0000000000000001 R12: 0000000000000246 R13: 0000000000000000 R14: ffffffff964c70c0 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff8951fbc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000000005003 CR2: 00007f20e8f39010 CR3: 000000002ca10005 CR4: 0000000000170ef0 Call Trace: ring_buffer_unlock_commit+0x1d/0xa0 trace_buffer_unlock_commit_regs+0x3b/0x1b0 trace_event_buffer_commit+0x67/0x1d0 trace_event_raw_event_ib_mad_recv_done_handler +0x11c/0x160 [ib_core] ib_mad_recv_done+0x48b/0xc10 [ib_core] ? trace_event_raw_event_cq_poll +0x6f/0xb0 [ib_core] __ib_process_cq+0x91/0x1c0 [ib_core] ib_cq_poll_work+0x26/0x80 [ib_core] process_one_work+0x1a7/0x360 ? create_worker+0x1a0/0x1a0 worker_thread+0x30/0x390 ? create_worker+0x1a0/0x1a0 kthread+0x116/0x130 ? kthread_flush_work_fn+0x10/0x10 ret_from_fork +0x35/0x40 ---[end trace 78ba8509d3830a16]---
CVE-2022-50473	In the Linux kernel, the following vulnerability has been resolved: cpufreq: Init completion before kobject_init_and_add() In cpufreq_policy_alloc(), it will call uninitialed completion in cpufreq_sysfs_release() when kobject_init_and_add() fails. And that will cause a crash such as the following page fault in complete: BUG: unable to handle page fault for address: ffffffff88888888 [...] RIP: 0010:complete+0x98/0x1f0 [...] Call Trace: kobject_put+0x1be/0x4c0 cpufreq_online.cold+0xee/0x1fd cpufreq_add_dev+0x183/0x1e0 subsys_interface_register+0x3f5/0x4e0 cpufreq_register_driver +0x3b7/0x670 acpi_cpufreq_init+0x56c/0x1000 [acpi_cpufreq] do_one_initcall+0x13d/0x780 do_init_module+0x1c3/0x630 load_module+0x6e67/0x73b0 __do_sys_finit_module+0x181/0x240 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x63/0xcd

CVE ID	Description
CVE-2022-50475	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Make sure "ib_port" is valid when access sysfs node The "ib_port" structure must be set before adding the sysfs kobject, and reset after removing it, otherwise it may crash when accessing the sysfs node: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000050 Mem abort info: ESR = 0x96000006 Exception class = DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 Data abort info: ISV = 0, ISS = 0x00000006 CM = 0, WnR = 0 user pgtable: 4k pages, 48-bit VAs, pgdp = 00000000e85f5ba5 [0000000000000050] pgd=0000000848fd9003, pud=000000085b387003, pmd=0000000000000000 Internal error: Oops: 96000006 [#2] PREEMPT SMP Modules linked in: ib_umad(O) mlx5_ib(O) nfnetlink_cttimeout(E) nfnetlink(E) act_gact(E) cls_flower(E) sch_ingress(E) openvswitch(E) nsh(E) nf_nat_ipv6(E) nf_nat_ipv4(E) nf_conntrack(E) nf_nat(E) nf_conntrack(E) nf_defrag_ipv6(E) nf_defrag_ipv4(E) mst_pciconf(O) ipmi_devintf(E) ipmi_msghandler(E) ipmb_dev_int(OE) mlx5_core(O) mlxfw(O) mlxdevm(O) auxiliary(O) ib_uverbs(O) ib_core(O) mlx_compat(O) psample(E) sbas_gwtd(E) uio_pdrv_genirq(E) uio(E) mlxbf_pmc(OE) mlxbf_gige(OE) mlxbf_tmfifo(OE) gpio_mlxbf2(OE) pwr_mlxbf(OE) mlx_trio(OE) i2c_mlxbf(OE) mlx_bootctl(OE) bluefield_edac(OE) knem(O) ip_tables(E) ipv6(E) crc_ccitt(E) [last unloaded: mst_pci] Process grep (pid: 3372, stack limit = 0x0000000022055c92) CPU: 5 PID: 3372 Comm: grep Tainted: G D OE 4.19.161-mlnx.47.gadcd9e3 #1 Hardware name: https://www.mellanox.com BlueField SoC/BlueField SoC, BIOS BlueField:3.9.2-15-ga2403ab Sep 8 2022 pstate: 40000005 (nZcv daif -PAN -UAO) pc : hw_stat_port_show+0x4c/0x80 [ib_core] lr : port_attr_show+0x40/0x58 [ib_core] sp : ffff000029f43b50 x29: ffff000029f43b50 x28: 0000000019375000 x27: ffff8007b821a540 x26: ffff000029f43e30 x25: 0000000000008000 x24: ffff000000eaa958 x23: 0000000000000100 x22: ffff8007a4ce3000 x21: ffff8007baff8000 x20: ffff8007b9066ac0 x19: ffff8007bae97578 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000000 x10: 0000000000000000 x9 : 0000000000000000 x8 : ffff8007a4ce4000 x7 : 0000000000000000 x6 : 000000000000003f x5 : ffff000000e6a280 x4 : ffff8007a4ce3000 x3 : 0000000000000000 x2 : aaaaaaaaaaaaaab x1 : ffff8007b9066a10 x0 : ffff8007baff8000 Call trace: hw_stat_port_show+0x4c/0x80 [ib_core] port_attr_show+0x40/0x58 [ib_core] sysfs_kf_seq_show+0x8c/0x150 kernfs_seq_show+0x44/0x50 seq_read+0x1b4/0x45c kernfs_fop_read+0x148/0x1d8 __vfs_read+0x58/0x180 vfs_read+0x94/0x154 ksys_read+0x68/0xd8 __arm64_sys_read+0x28/0x34 el0_svc_common+0x88/0x18c el0_svc_handler+0x78/0x94 el0_svc+0x8/0xe8 Code: f2955562 aa1603e4 aa1503e0 f9405683 (f9402861)
CVE-2022-50476	In the Linux kernel, the following vulnerability has been resolved: ntb_netdev: Use dev_kfree_skb_any() in interrupt context TX/RX callback handlers (ntb_netdev_tx_handler(), ntb_netdev_rx_handler()) can be called in interrupt context via the DMA framework when the respective DMA operations have completed. As such, any calls by these routines to free skb's, should use the interrupt context safe dev_kfree_skb_any() function. Previously, these callback handlers would call the interrupt unsafe version of dev_kfree_skb(). This has not presented an issue on Intel IOAT DMA engines as that driver utilizes tasklets rather than a hard interrupt handler, like the AMD PTDMA DMA driver. On AMD systems, a kernel WARNING message is encountered, which is being issued from skb_release_head_state() due to in_hardirq() being true. Besides the user visible WARNING from the kernel, the other symptom of this bug was that TCP/IP performance across the ntb_netdev interface was very poor, i.e. approximately an order of magnitude below what was expected. With the repair to use dev_kfree_skb_any(), kernel WARNINGS from skb_release_head_state() ceased and TCP/IP performance, as measured by iperf, was on par with expected results, approximately 20 Gb/s on AMD Milan based server. Note that this performance is comparable with Intel based servers.
CVE-2022-50479	In the Linux kernel, the following vulnerability has been resolved: drm/amd: fix potential memory leak This patch fix potential memory leak (clk_src) when function run into last return NULL. s/free/kfree/ - Alex
CVE-2022-50481	In the Linux kernel, the following vulnerability has been resolved: cxl: fix possible null-ptr-deref in cxl_guest_init_afu(adapter) If device_register() fails in cxl_register_afu(adapter), the device is not added, device_unregister() can not be called in the error path, otherwise it will cause a null-ptr-deref because of removing not added device. As comment of device_register() says, it should use put_device() to give up the reference in the error path. So split device_unregister() into device_del() and put_device(), then goes to put dev when register fails.
CVE-2022-50482	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Clean up si_domain in the init_dmars() error path A splat from kmem_cache_destroy() was seen with a kernel prior to commit ee2653bbe89d ("iommu/vt-d: Remove domain and devinfo mempool") when there was a failure in init_dmars(), because the iommu_domain cache still had objects. While the mempool code is now gone, there still is a leak of the si_domain memory if init_dmars() fails. So clean up si_domain in the init_dmars() error path.

CVE ID	Description
CVE-2022-50484	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix potential memory leaks When the driver hits -ENOMEM at allocating a URB or a buffer, it aborts and goes to the error path that releases the all previously allocated resources. However, when -ENOMEM hits at the middle of the sync EP URB allocation loop, the partially allocated URBs might be left without released, because ep->nurbs is still zero at that point. Fix it by setting ep->nurbs at first, so that the error handler loops over the full URB list.
CVE-2022-50485	In the Linux kernel, the following vulnerability has been resolved: ext4: add EXT4_IGET_BAD flag to prevent unexpected bad inode There are many places that will get unhappy (and crash) when ext4_iget() returns a bad inode. However, if iget the boot loader inode, allows a bad inode to be returned, because the inode may not be initialized. This mechanism can be used to bypass some checks and cause panic. To solve this problem, we add a special iget flag EXT4_IGET_BAD. Only with this flag we'd be returning bad inode from ext4_iget(), otherwise we always return the error code if the inode is bad inode. (suggested by Jan Kara)
CVE-2022-50489	In the Linux kernel, the following vulnerability has been resolved: drm/mipi-dsi: Detach devices when removing the host Whenever the MIPI-DSI host is unregistered, the code of mipi_dsi_host_unregister() loops over every device currently found on that bus and will unregister it. However, it doesn't detach it from the bus first, which leads to all kind of resource leaks if the host wants to perform some clean up whenever a device is detached.
CVE-2022-50490	In the Linux kernel, the following vulnerability has been resolved: bpf: Propagate error from htab_lock_bucket() to userspace In __htab_map_lookup_and_delete_batch() if htab_lock_bucket() returns -EBUSY, it will go to next bucket. Going to next bucket may not only skip the elements in current bucket silently, but also incur out-of-bound memory access or expose kernel memory to userspace if current bucket_cnt is greater than bucket_size or zero. Fixing it by stopping batch operation and returning -EBUSY when htab_lock_bucket() fails, and the application can retry or skip the busy batch as needed.
CVE-2022-50493	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix crash when I/O abort times out While performing CPU hotplug, a crash with the following stack was seen: Call Trace: qla24xx_process_response_queue+0x42a/0x970 [qla2xxx] qla2x00_start_nvme_mq+0x3a2/0x4b0 [qla2xxx] qla_nvme_post_cmd+0x166/0x240 [qla2xxx] nvme_fc_start_fcp_op.part.0+0x119/0x2e0 [nvme_fc] blk_mq_dispatch_rq_list+0x17b/0x610 __blk_mq_sched_dispatch_requests+0xb0/0x140 blk_mq_sched_dispatch_requests+0x30/0x60 __blk_mq_run_hw_queue+0x35/0x90 __blk_mq_delay_run_hw_queue+0x161/0x180 blk_execute_rq+0xbe/0x160 __nvme_submit_sync_cmd+0x16f/0x220 [nvme_core] nvme_fc_connect_admin_queue+0x11a/0x170 [nvme_fabrics] nvme_fc_create_association.cold+0x50/0x3dc [nvme_fc] nvme_fc_connect_ctrl_work+0x19/0x30 [nvme_fc] process_one_work+0x1e8/0x3c0 On abort timeout, completion was called without checking if the I/O was already completed. Verify that I/O and abort request are indeed outstanding before attempting completion.
CVE-2022-50494	In the Linux kernel, the following vulnerability has been resolved: thermal: intel_powerclamp: Use get_cpu() instead of smp_processor_id() to avoid crash When CPU 0 is offline and intel_powerclamp is used to inject idle, it generates kernel BUG: BUG: using smp_processor_id() in preemptible [00000000] code: bash/15687 caller is debug_smp_processor_id+0x17/0x20 CPU: 4 PID: 15687 Comm: bash Not tainted 5.19.0-rc7+ #57 Call Trace: <TASK> dump_stack_lvl+0x49/0x63 dump_stack+0x10/0x16 check_preemption_disabled+0xdd/0xe0 debug_smp_processor_id+0x17/0x20 powerclamp_set_cur_state+0x7f/0xf9 [intel_powerclamp] ... Here CPU 0 is the control CPU by default and changed to the current CPU, if CPU 0 is offline. This check has to be performed under cpus_read_lock(), hence the above warning. Use get_cpu() instead of smp_processor_id() to avoid this BUG. [rjw: Subject edits]
CVE-2022-50496	In the Linux kernel, the following vulnerability has been resolved: dm cache: Fix UAF in destroy() Dm_cache also has the same UAF problem when dm_resume() and dm_destroy() are concurrent. Therefore, cancelling timer again in destroy().
CVE-2022-50497	In the Linux kernel, the following vulnerability has been resolved: binfmt_misc: fix shift-out-of-bounds in check_special_flags UBSAN reported a shift-out-of-bounds warning: left shift of 1 by 31 places cannot be represented in type 'int' Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x8d/0xcf lib/dump_stack.c:106 ubsan_epilogue+0xa/0x44 lib/ubsan.c:151 __ubsan_handle_shift_out_of_bounds+0x1e7/0x208 lib/ubsan.c:322 check_special_flags fs/binfmt_misc.c:241 [inline] create_entry fs/binfmt_misc.c:456 [inline] bm_register_write+0x9d3/0xa20 fs/binfmt_misc.c:654 vfs_write+0x11e/0x580 fs/read_write.c:582 ksys_write+0xcf/0x120 fs/read_write.c:637 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x34/0x80 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x4194e1 Since the type of Node's flags is unsigned long, we should define these macros with same type too.

CVE ID	Description
CVE-2022-50500	In the Linux kernel, the following vulnerability has been resolved: netdevsim: fix memory leak in nsim_drv_probe() when nsim_dev_resources_register() failed If some items in nsim_dev_resources_register() fail, memory leak will occur. The following is the memory leak information. unreferenced object 0xffff888074c02600 (size 128): comm "echo", pid 8159, jiffies 4294945184 (age 493.530s) hex dump (first 32 bytes): 40 47 ea 89 ff ff ff 01 00 00 00 00 00 00 @G..... ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff backtrace: [<0000000011a31c98>] kmalloc_trace+0x22/0x60 [<0000000027384c69>] devl_resource_register+0x144/0x4e0 [<00000000a16db248>] nsim_drv_probe+0x37a/0x1260 [<000000007d1f448c>] really_probe+0x20b/0xb10 [<00000000c416848a>] __driver_probe_device+0x1b3/0x4a0 [<00000000077e0351>] driver_probe_device+0x49/0x140 [<0000000054f2465a>] __device_attach_driver+0x18c/0x2a0 [<0000000008538f359>] bus_for_each_drv+0x151/0x1d0 [<00000000038e09747>] __device_attach+0x1c9/0x4e0 [<00000000dd86e533>] bus_probe_device+0x1d5/0x280 [<00000000839bea35>] device_add+0xae0/0x1cb0 [<000000009c2abf46>] new_device_store+0x3b6/0x5f0 [<00000000fb823df>] bus_attr_store+0x72/0xa0 [<000000007acc4295>] sysfs_kf_write+0x106/0x160 [<000000005f50cb4d>] kernfs_fop_write_iter+0x3a8/0x5a0 [<0000000075eb41bf>] vfs_write+0x8f0/0xc80
CVE-2022-50504	In the Linux kernel, the following vulnerability has been resolved: powerpc/rtas: avoid scheduling in rtas_os_term() It's unsafe to use rtas_busy_delay() to handle a busy status from the ibm,os-term RTAS function in rtas_os_term(): Kernel panic - not syncing: Attempted to kill init! exitcode=0x0000000b BUG: sleeping function called from invalid context at arch/powerpc/kernel/rtas.c:618 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 1, name: swapper/0 preempt_count: 2, expected: 0 CPU: 7 PID: 1 Comm: swapper/0 Tainted: G D 6.0.0-rc5-02182-gf8553a572277-dirty #9 Call Trace: [c000000007b8f000] [c000000001337110] dump_stack_lvl+0xb4/0x110 (unreliable) [c0000000007b8f040] [c0000000002440e4] __might_resched+0x394/0x3c0 [c000000007b8f0e0] [c00000000004f680] rtas_busy_delay+0x120/0x1b0 [c000000007b8f100] [c0000000000052d04] rtas_os_term+0xb8/0xf4 [c000000007b8f180] [c0000000001150fc] pseries_panic+0x50/0x68 [c000000007b8f1f0] [c000000000036354] ppc_panic_platform_handler+0x34/0x50 [c000000007b8f210] [c0000000002303c4] notifier_call_chain+0xd4/0x1c0 [c000000007b8f2b0] [c00000000002306cc] atomic_notifier_call_chain+0xac/0x1c0 [c0000000007b8f2f0] [c0000000001d62b8] panic+0x228/0x4d0 [c000000007b8f390] [c0000000001e573c] do_exit+0x140c/0x1420 [c000000007b8f480] [c0000000001e586c] make_task_dead+0xdc/0x200 Use rtas_busy_delay_time() instead, which signals without side effects whether to attempt the ibm,os-term RTAS call again.
CVE-2022-50505	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix pci device refcount leak in ppr_notifier() As comment of pci_get_domain_bus_and_slot() says, it returns a pci device with refcount increment, when finish using it, the caller must decrement the reference count by calling pci_dev_put(). So call it before returning from ppr_notifier() to avoid refcount leak.
CVE-2022-50511	In the Linux kernel, the following vulnerability has been resolved: lib/fonts: fix undefined behavior in bit shift for get_default_font Shifting signed 32-bit value by 31 bits is undefined, so changing significant bit to unsigned. The UBSAN warning calltrace like below: UBSAN: shift-out-of-bounds in lib/fonts/fonts.c:139:20 left shift of 1 by 31 places cannot be represented in type 'int' <TASK> dump_stack_lvl+0x7d/0xa5 dump_stack+0x15/0x1b ubsan_epilogue+0xe/0x4e __ubsan_handle_shift_out_of_bounds+0x1e7/0x20c get_default_font+0x1c7/0x1f0 fbcon_startup+0x347/0x3a0 do_take_over_console+0xce/0x270 do_fbcon_takeover+0xa1/0x170 do_fb_registered+0x2a8/0x340 fbcon_fb_registered+0x47/0xe0 register_framebuffer+0x294/0x4a0 __drm_fb_helper_initial_config_and_unlock+0x43c/0x880 [drm_kms_helper] drm_fb_helper_initial_config+0x52/0x80 [drm_kms_helper] drm_fbdev_client_hotplug+0x156/0x1b0 [drm_kms_helper] drm_fbdev_generic_setup+0xfc/0x290 [drm_kms_helper] bochs_pci_probe+0x6ca/0x772 [bochs] local_pci_probe+0x4d/0xb0 pci_device_probe+0x119/0x320 really_probe+0x181/0x550 __driver_probe_device+0xc6/0x220 driver_probe_device+0x32/0x100 __driver_attach+0x195/0x200 bus_for_each_dev+0xbb/0x120 driver_attach+0x27/0x30 bus_add_driver+0x22e/0x2f0 driver_register+0xa9/0x190 __pci_register_driver+0x90/0xa0 bochs_pci_driver_init+0x52/0x1000 [bochs] do_one_initcall+0x76/0x430 do_init_module+0x61/0x28a load_module+0x1f82/0x2e50 __do_sys_finit_module+0xf8/0x190 __x64_sys_finit_module+0x23/0x30 do_syscall_64+0x58/0x80 entry_SYSCALL_64_after_hwframe+0x63/0xcd </TASK>

CVE ID	Description
CVE-2022-50516	In the Linux kernel, the following vulnerability has been resolved: fs: dlm: fix invalid dereference of sb_lvbptr I experience issues when putting a lkbsb on the stack and have sb_lvbptr field to a dangled pointer while not using DLM_LKF_VALBLK. It will crash with the following kernel message, the dangled pointer is here 0xdeadbeef as example: [102.749317] BUG: unable to handle page fault for address: 00000000deadbeef [102.749320] #PF: supervisor read access in kernel mode [102.749323] #PF: error_code(0x0000) - not-present page [102.749325] PGD 0 P4D 0 [102.749332] Oops: 0000 [#1] PREEMPT SMP PTI [102.749336] CPU: 0 PID: 1567 Comm: lock_torture_wr Tainted: G W 5.19.0-rc3+ #1565 [102.749343] Hardware name: Red Hat KVM/RHEL-AV, BIOS 1.16.0-2.module+el8.7.0+15506+033991b0 04/01/2014 [102.749344] RIP: 0010:memcpy_erms+0x6/0x10 [102.749353] Code: cc cc cc cc eb 1e 0f 1f 00 48 89 f8 48 89 d1 48 c1 e9 03 83 e2 07 f3 48 a5 89 d1 f3 a4 c3 66 0f 1f 44 00 00 48 89 f8 48 89 d1 <f3> a4 c3 0f 1f 80 00 00 00 00 48 89 f8 48 83 fa 20 72 7e 40 38 fe [102.749355] RSP: 0018:ffff97a58145fd08 EFLAGS: 00010202 [102.749358] RAX: ffff901778b77070 RBX: 0000000000000000 RCX: 0000000000000040 [102.749360] RDX: 0000000000000040 RSI: 00000000deadbeef RDI: ffff901778b77070 [102.749362] RBP: ffff97a58145fd10 R08: ffff901760b67a70 R09: 0000000000000001 [102.749364] R10: ffff9017008e2cb8 R11: 0000000000000001 R12: ffff901760b67a70 [102.749366] R13: ffff901760b78f00 R14: 0000000000000003 R15: 0000000000000001 [102.749368] FS: 0000000000000000(0000) GS:ffff901876e00000(0000) knlGS:0000000000000000 [102.749372] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [102.749374] CR2: 00000000deadbeef CR3: 000000017c49a004 CR4: 0000000000770ef0 [102.749376] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [102.749378] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 [102.749379] PKRU: 55555554 [102.749381] Call Trace: [102.749382] <TASK> [102.749383] ? send_args+0xb2/0xd0 [102.749389] send_common+0xb7/0xd0 [102.749395] _unlock_lock +0x2c/0x90 [102.749400] unlock_lock.isra.56+0x62/0xa0 [102.749405] dlm_unlock+0x21e/0x330 [102.749411] ? lock_torture_stats+0x80/0x80 [dlm_locktorture] [102.749416] torture_unlock +0x5a/0x90 [dlm_locktorture] [102.749419] ? preempt_count_sub+0xba/0x100 [102.749427] lock_torture_writer+0xbd/0x150 [dlm_locktorture] [102.786186] kthread+0x10a/0x130 [102.786581] ? kthread_complete_and_exit+0x20/0x20 [102.787156] ret_from_fork+0x22/0x30 [102.787588] </TASK> [102.787855] Modules linked in: dlm_locktorture torture rpcsec_gss_krb5 intel_rapl_msr intel_rapl_common kvm_intel iTCO_wdt iTCO_vendor_support kvm vmw_vsock virtio_transport qxl irqbypass vmw_vsock_virtio_transport_common drm_ttm_helper crc32_pclmul joydev crc32_intel ttm vsock virtio_scsi virtio_balloon snd_pcm drm_kms_helper virtio_console snd_timer snd drm soundcore syscopyarea i2c_i801 sysfillrect sysimgblt i2c_smbus pcspkr fb_sys_fops lpc_ich serio_raw [102.792536] CR2: 00000000deadbeef [102.792930] ---[end trace 0000000000000000]--- This patch fixes the issue by checking also on DLM_LKF_VALBLK on exflags is set when copying the lvbptr array instead of if it's just null which fixes for me the issue. I think this patch can fix other dlm users as well, depending how they handle the init, freeing memory handling of sb_lvbptr and don't set DLM_LKF_VALBLK for some dlm_lock() calls. It might a there could be a hidden issue all the time. However with checking on DLM_LKF_VALBLK the user always need to provide a sb_lvbptr non-null value. There might be more intelligent handling between per ls lvblen, DLM_LKF_VALBLK and non-null to report the user the way how DLM API is used is wrong but can be added for later, this will only fix the current behaviour.
CVE-2022-50520	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: Fix PCI device refcount leak in radeon_atrm_get_bios() As comment of pci_get_class() says, it returns a pci_device with its refcount increased and decreased the refcount for the input parameter @from if it is not NULL. If we break the loop in radeon_atrm_get_bios() with 'pdev' not NULL, we need to call pci_dev_put() to decrease the refcount. Add the missing pci_dev_put() to avoid refcount leak.
CVE-2022-50521	In the Linux kernel, the following vulnerability has been resolved: platform/x86: mxm-wmi: fix memleak in mxm_wmi_call_mx[ds][mx]() The ACPI buffer memory (out.pointer) returned by wmi_evaluate_method() is not freed after the call, so it leads to memory leak. The method results in ACPI buffer is not used, so just pass NULL to wmi_evaluate_method() which fixes the memory leak.
CVE-2022-50527	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix size validation for non-exclusive domains (v4) Fix amdgpu_bo_validate_size() to check whether the TTM domain manager for the requested memory exists, else we get a kernel oops when dereferencing "man". v2: Make the patch standalone, i.e. not dependent on local patches. v3: Preserve old behaviour and just check that the manager pointer is not NULL. v4: Complain if GTT domain requested and it is uninitialized--most likely a bug.
CVE-2022-50528	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix memory leakage This patch fixes potential memory leakage and seg fault in _gpuvmm_import_dmabuf() function

CVE ID	Description
CVE-2022-50531	In the Linux kernel, the following vulnerability has been resolved: tipc: fix an information leak in tipc_topsrv_kern_subscr Use a 8-byte write to initialize sub_usr_handle in tipc_topsrv_kern_subscr(), otherwise four bytes remain uninitialized when issuing setsockopt(..., SOL_TIPC, ...). This resulted in an infoleak reported by KMSAN when the packet was received: ===== BUG: KMSAN: kernel-infoleak in copyout+0xbc/0x100 lib/iov_iter.c:169 instrument_copy_to_user ./include/linux/instrumented.h:121 copyout+0xbc/0x100 lib/iov_iter.c:169 _copy_to_iter +0x5c0/0x20a0 lib/iov_iter.c:527 copy_to_iter ./include/linux/uio.h:176 simple_copy_to_iter +0x64/0xa0 net/core/datagram.c:513 __skb_datagram_iter+0x123/0xdc0 net/core/datagram.c:419 skb_copy_datagram_iter+0x58/0x200 net/core/datagram.c:527 skb_copy_datagram_msg ./include/linux/skbuff.h:3903 packet_recvmmsg+0x521/0x1e70 net/packet/af_packet.c:3469 __sys_recvmmsg +0x2c4/0x810 net/socket.c:? __sys_recvmmsg+0x217/0x840 net/socket.c:2743 __sys_recvmmsg net/socket.c:2773 __do_sys_recvmmsg net/socket.c:2783 __se_sys_recvmmsg net/socket.c:2780 __x64_sys_recvmmsg+0x364/0x540 net/socket.c:2780 do_syscall_x64 arch/x86/entry/common.c:50 do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe +0x63/0xcd arch/x86/entry/entry_64.S:120 ... Uninit was stored to memory at: tipc_sub_subscribe +0x42d/0xb50 net/tipc/subscr.c:156 tipc_conn_rcv_sub+0x246/0x620 net/tipc/topsrv.c:375 tipc_topsrv_kern_subscr+0x2e8/0x400 net/tipc/topsrv.c:579 tipc_group_create+0x4e7/0x7d0 net/tipc/group.c:190 tipc_sk_join+0x2a8/0x770 net/tipc/socket.c:3084 tipc_setsockopt+0xae5/0xe40 net/tipc/socket.c:3201 __sys_setsockopt+0x87f/0xdc0 net/socket.c:2252 __do_sys_setsockopt net/socket.c:2263 __se_sys_setsockopt net/socket.c:2260 __x64_sys_setsockopt+0xe0/0x160 net/socket.c:2260 do_syscall_x64 arch/x86/entry/common.c:50 do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd arch/x86/entry/entry_64.S:120 Local variable sub created at: tipc_topsrv_kern_subscr+0x57/0x400 net/tipc/topsrv.c:562 tipc_group_create+0x4e7/0x7d0 net/tipc/group.c:190 Bytes 84-87 of 88 are uninitialized Memory access of size 88 starts at ffff88801ed57cd0 Data copied to user address 0000000020000400 ... =====
CVE-2022-50532	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix possible resource leaks in mpt3sas_transport_port_add() In mpt3sas_transport_port_add(), if sas_rphy_add() returns error, sas_rphy_free() needs be called to free the resource allocated in sas_end_device_alloc(). Otherwise a kernel crash will happen: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000108 CPU: 45 PID: 37020 Comm: bash Kdump: loaded Tainted: G W 6.1.0-rc1+ #189 pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : device_del+0x54/0x3d0 lr : device_del+0x37c/0x3d0 Call trace: device_del+0x54/0x3d0 attribute_container_class_device_del +0x28/0x38 transport_remove_classdev+0x6c/0x80 attribute_container_device_trigger+0x108/0x110 transport_remove_device+0x28/0x38 sas_rphy_remove+0x50/0x78 [scsi_transport_sas] sas_port_delete+0x30/0x148 [scsi_transport_sas] do_sas_phy_delete+0x78/0x80 [scsi_transport_sas] device_for_each_child+0x68/0xb0 sas_remove_children+0x30/0x50 [scsi_transport_sas] sas_rphy_remove+0x38/0x78 [scsi_transport_sas] sas_port_delete+0x30/0x148 [scsi_transport_sas] do_sas_phy_delete+0x78/0x80 [scsi_transport_sas] device_for_each_child+0x68/0xb0 sas_remove_children+0x30/0x50 [scsi_transport_sas] sas_remove_host+0x20/0x38 [scsi_transport_sas] scsih_remove+0xd8/0x420 [mpt3sas] Because transport_add_device() is not called when sas_rphy_add() fails, the device is not added. When sas_rphy_remove() is subsequently called to remove the device in the remove() path, a NULL pointer dereference happens.
CVE-2022-50534	In the Linux kernel, the following vulnerability has been resolved: dm thin: Use last transaction's pmd->root when commit failed Recently we found a softlock up problem in dm thin pool btree lookup code due to corrupted metadata: Kernel panic - not syncing: softlockup: hung tasks CPU: 7 PID: 2669225 Comm: kworker/u16:3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) Workqueue: dm-thin do_worker [dm_thin_pool] Call Trace: <IRQ> dump_stack +0x9c/0xd3 panic+0x35d/0x6b9 watchdog_timer_fn.cold+0x16/0x25 __run_hrtimer+0xa2/0x2d0 </IRQ> RIP: 0010: __relink_lru+0x102/0x220 [dm_bufio] __bufio_new+0x11f/0x4f0 [dm_bufio] new_read+0xa3/0x1e0 [dm_bufio] dm_bm_read_lock+0x33/0xd0 [dm_persistent_data] ro_step +0x63/0x100 [dm_persistent_data] btree_lookup_raw.constprop.0+0x44/0x220 [dm_persistent_data] dm_btree_lookup+0x16f/0x210 [dm_persistent_data] dm_thin_find_block+0x12c/0x210 [dm_thin_pool] __process_bio_read_only+0xc5/0x400 [dm_thin_pool] process_thin_deferred_bios+0x1a4/0x4a0 [dm_thin_pool] process_one_work+0x3c5/0x730 Following process may generate a broken btree mixed with fresh and stale btree nodes, which could get dm thin trapped in an infinite loop while looking up data block: Transaction 1: pmd->root = A, A->B->C // One path in btree pmd->root = X, X->Y->Z // Copy-up Transaction 2: X,Z is updated on disk, Y write failed. // Commit failed, dm thin becomes read-only. process_bio_read_only dm_thin_find_block __find_block dm_btree_lookup(pmd->root) The pmd->root points to a broken btree, Y may contain stale node pointing to any block, for example X, which gets dm thin trapped into a dead loop while looking up Z. Fix this by setting pmd->root in __open_metadata(), so that dm thin will use the last transaction's pmd->root if commit failed. Fetch a reproducer in [Link]. Linke: https://bugzilla.kernel.org/show_bug.cgi?id=216790

CVE ID	Description
CVE-2022-50535	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix potential null-deref in dm_resume [Why] Fixing smatch error: dm_resume() error: we previously assumed 'aconnector->dc_link' could be null [How] Check if dc_link null at the beginning of the loop, so further checks can be dropped.
CVE-2022-50536	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix repeated calls to sock_put() when msg has more_data In tcp_bpf_send_verdict() redirection, the eval variable is assigned to __SK_REDIRECT after the apply_bytes data is sent, if msg has more_data, sock_put() will be called multiple times. We should reset the eval variable to __SK_NONE every time more_data starts. This causes: IPv4: Attempt to release TCP socket in state 1 00000000b4c925d7 ----- [cut here]----- refcount_t: addition on 0; use-after-free. WARNING: CPU: 5 PID: 4482 at lib/refcount.c:25 refcount_warn_saturate+0x7d/0x110 Modules linked in: CPU: 5 PID: 4482 Comm: sockhash_bypass Kdump: loaded Not tainted 6.0.0 #1 Hardware name: Red Hat KVM, BIOS 1.11.0-2.el7 04/01/2014 Call Trace: <TASK> __tcp_transmit_skb+0xa1b/0xb90 ? __alloc_skb+0x8c/0x1a0 ? __kmalloc_node_track_caller+0x184/0x320 tcp_write_xmit+0x22a/0x1110 __tcp_push_pending_frames+0x32/0xf0 do_tcp_sendpages+0x62d/0x640 tcp_bpf_push+0xae/0x2c0 tcp_bpf_sendmsg_redir+0x260/0x410 ? preempt_count_add+0x70/0xa0 tcp_bpf_send_verdict+0x386/0x4b0 tcp_bpf_sendmsg+0x21b/0x3b0 sock_sendmsg+0x58/0x70 __sys_sendto+0xfa/0x170 ? xfd_validate_state+0x1d/0x80 ? switch_fpu_return+0x59/0xe0 __x64_sys_sendto+0x24/0x30 do_syscall_64+0x37/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2022-50543	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix mr->map double free rxe_mr_cleanup() which tries to free mr->map again will be called when rxe_mr_init_user() fails: CPU: 0 PID: 4917 Comm: rdma_flush_serv Kdump: loaded Not tainted 6.1.0-rc1-roce-flush+ #25 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x45/0x5d panic+0x19e/0x349 end_report.part.0+0x54/0x7c kasan_report.cold+0xa/0xf rxe_mr_cleanup+0x9d/0xf0 [rdma_rxe] __rxe_cleanup+0x10a/0x1e0 [rdma_rxe] rxe_reg_user_mr+0xb7/0xd0 [rdma_rxe] ib_uverbs_reg_mr+0x26a/0x480 [ib_uverbs] ib_uverbs_handler_UVERBS_METHOD_INVOKE_WRITE+0x1a2/0x250 [ib_uverbs] ib_uverbs_cmd_verbs+0x1397/0x15a0 [ib_uverbs] This issue was firstly exposed since commit b18c7da63fcb ("RDMA/rxe: Fix memory leak in error path code") and then we fixed it in commit 8ff5f5d9d8cf ("RDMA/rxe: Prevent double freeing rxe_map_set()") but this fix was reverted together at last by commit 1e75550648da (Revert "RDMA/rxe: Create duplicate mapping tables for FMRs") Simply let rxe_mr_cleanup() always handle freeing the mr->map once it is successfully allocated.
CVE-2022-50544	In the Linux kernel, the following vulnerability has been resolved: usb: host: xhci: Fix potential memory leak in xhci_alloc_stream_info() xhci_alloc_stream_info() allocates stream context array for stream_info ->stream_ctx_array with xhci_alloc_stream_ctx(). When some error occurs, stream_info->stream_ctx_array is not released, which will lead to a memory leak. We can fix it by releasing the stream_info->stream_ctx_array with xhci_free_stream_ctx() on the error path to avoid the potential memory leak.

CVE ID	Description
CVE-2022-50546	In the Linux kernel, the following vulnerability has been resolved: ext4: fix uninitialized value in 'ext4_evict_inode' Syzbot found the following issue: <pre> ===== BUG: KMSAN: uninit- value in ext4_evict_inode+0xdd/0x26b0 fs/ext4/inode.c:180 ext4_evict_inode+0xdd/0x26b0 fs/ext4/inode.c:180 evict+0x365/0x9a0 fs/inode.c:664 iput_final fs/inode.c:1747 [inline] iput +0x985/0xdd0 fs/inode.c:1773 __ext4_new_inode+0xe54/0x7ec0 fs/ext4/ialloc.c:1361 ext4_mknod +0x376/0x840 fs/ext4/namei.c:2844 vfs_mknod+0x79d/0x830 fs/namei.c:3914 do_mknodat +0x47d/0xaa0 __do_sys_mknodat fs/namei.c:3992 [inline] __se_sys_mknodat fs/namei.c:3989 [inline] __ia32_sys_mknodat+0xeb/0x150 fs/namei.c:3989 do_syscall_32_irqs_on arch/x86/ entry/common.c:112 [inline] __do_fast_syscall_32+0xa2/0x100 arch/x86/entry/common.c:178 do_fast_syscall_32+0x33/0x70 arch/x86/entry/common.c:203 do_SYSENTER_32+0x1b/0x20 arch/x86/entry/common.c:246 entry_SYSENTER_compat_after_hwframe+0x70/0x82 Uninit was created at: __alloc_pages+0x9f1/0xe80 mm/page_alloc.c:5578 alloc_pages+0xaae/0xd80 mm/ mempolicy.c:2285 alloc_slab_page mm/slub.c:1794 [inline] allocate_slab+0x1b5/0x1010 mm/ slub.c:1939 new_slab mm/slub.c:1992 [inline] __slab_alloc+0x10c3/0x2d60 mm/slub.c:3180 __slab_alloc mm/slub.c:3279 [inline] slab_alloc_node mm/slub.c:3364 [inline] slab_alloc mm/ slub.c:3406 [inline] __kmem_cache_alloc_lru mm/slub.c:3413 [inline] kmem_cache_alloc_lru +0x6f3/0xb30 mm/slub.c:3429 alloc_inode_sb include/linux/fs.h:3117 [inline] ext4_alloc_inode +0x5f/0x860 fs/ext4/super.c:1321 alloc_inode+0x83/0x440 fs/inode.c:259 new_inode_pseudo fs/ inode.c:1018 [inline] new_inode+0x3b/0x430 fs/inode.c:1046 __ext4_new_inode+0x2a7/0x7ec0 fs/ext4/ ialloc.c:959 ext4_mkdir+0x4d5/0x1560 fs/ext4/namei.c:2992 vfs_mkdir+0x62a/0x870 fs/namei.c:4035 do_mkdirat+0x466/0x7b0 fs/namei.c:4060 __do_sys_mkdirat fs/namei.c:4075 [inline] __se_sys_mkdirat fs/namei.c:4073 [inline] __ia32_sys_mkdirat+0xc4/0x120 fs/namei.c:4073 do_syscall_32_irqs_on arch/ x86/entry/common.c:112 [inline] __do_fast_syscall_32+0xa2/0x100 arch/x86/entry/common.c:178 do_fast_syscall_32+0x33/0x70 arch/x86/entry/common.c:203 do_SYSENTER_32+0x1b/0x20 arch/x86/entry/common.c:246 entry_SYSENTER_compat_after_hwframe+0x70/0x82 CPU: 1 PID: 4625 Comm: syz-executor.2 Not tainted 6.1.0-rc4-syzkaller-62821-gcb231e2f67ec #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 ===== Now, 'ext4_alloc_inode()' didn't init 'ei->i_flags'. If new inode failed before set 'ei->i_flags' in '__ext4_new_inode()', then do 'iput()'. As after 6bc0d63dad7f commit will access 'ei->i_flags' in 'ext4_evict_inode()' which will lead to access uninit-value. To solve above issue just init 'ei->i_flags' in 'ext4_alloc_inode()'.</pre>

CVE ID	Description
CVE-2022-50549	In the Linux kernel, the following vulnerability has been resolved: dm thin: Fix ABBA deadlock between shrink_slab and dm_pool_abort_metadata Following concurrent processes: P1(drop cache) P2(kworker) drop_caches_sysctl_handler drop_slab shrink_slab down_read(&shrinker_rwsem) - LOCK A do_shrink_slab super_cache_scan prune_icache_sb dispose_list evict ext4_evict_inode ext4_clear_inode ext4_discard_preallocations ext4_mb_load_buddy_gfp ext4_mb_init_cache ext4_read_block_bitmap_nowait ext4_read_bh_nowait submit_bh dm_submit_bio do_worker process_deferred_bios commit metadata_operation_failed dm_pool_abort_metadata down_write(&pmd->root_lock) - LOCK B __destroy_persistent_data_objects dm_block_manager_destroy dm_bufio_client_destroy unregister_shrinker down_write(&shrinker_rwsem) thin_map dm_thin_find_block # down_read(&pmd->root_lock) --> ABBA deadlock , which triggers hung task: [76.974820] INFO: task kworker/u4:3:63 blocked for more than 15 seconds. [76.976019] Not tainted 6.1.0-rc4-00011-g8f17dd350364-dirty #910 [76.978521] task:kworker/u4:3 state:D stack:0 pid:63 ppid:2 [76.978534] Workqueue: dm-thin do_worker [76.978552] Call Trace: [76.978564] __schedule+0x6ba/0x10f0 [76.978582] schedule+0x9d/0x1e0 [76.978588] rwsem_down_write_slowpath+0x587/0xdf0 [76.978600] down_write+0xec/0x110 [76.978607] unregister_shrinker+0x2c/0xf0 [76.978616] dm_bufio_client_destroy+0x116/0x3d0 [76.978625] dm_block_manager_destroy+0x19/0x40 [76.978629] __destroy_persistent_data_objects+0x5e/0x70 [76.978636] dm_pool_abort_metadata+0x8e/0x100 [76.978643] metadata_operation_failed+0x86/0x110 [76.978649] commit+0x6a/0x230 [76.978655] do_worker+0xc6e/0xd90 [76.978702] process_one_work+0x269/0x630 [76.978714] worker_thread+0x266/0x630 [76.978730] kthread+0x151/0x1b0 [76.978772] INFO: task test.sh:2646 blocked for more than 15 seconds. [76.979756] Not tainted 6.1.0-rc4-00011-g8f17dd350364-dirty #910 [76.982111] task:test.sh state:D stack:0 pid:2646 ppid:2459 [76.982128] Call Trace: [76.982139] __schedule+0x6ba/0x10f0 [76.982155] schedule+0x9d/0x1e0 [76.982159] rwsem_down_read_slowpath+0x4f4/0x910 [76.982173] down_read+0x84/0x170 [76.982177] dm_thin_find_block+0x4c/0xd0 [76.982183] thin_map+0x201/0x3d0 [76.982188] __map_bio+0x5b/0x350 [76.982195] dm_submit_bio+0x2b6/0x930 [76.982202] __submit_bio+0x123/0x2d0 [76.982209] submit_bio_noacct_nocheck+0x101/0x3e0 [76.982222] submit_bio_noacct+0x389/0x770 [76.982227] submit_bio+0x50/0xc0 [76.982232] submit_bh_wbc+0x15e/0x230 [76.982238] submit_bh+0x14/0x20 [76.982241] ext4_read_bh_nowait+0xc5/0x130 [76.982247] ext4_read_block_bitmap_nowait+0x340/0xc60 [76.982254] ext4_mb_init_cache+0x1ce/0xdc0 [76.982259] ext4_mb_load_buddy_gfp+0x987/0xfa0 [76.982263] ext4_discard_preallocations+0x45d/0x830 [76.982274] ext4_clear_inode+0x48/0xf0 [76.982280] ext4_evict_inode+0xcf/0xc70 [76.982285] evict+0x119/0x2b0 [76.982290] dispose_list+0x43/0xa0 [76.982294] prune_icache_sb+0x64/0x90 [76.982298] super_cache_scan+0x155/0x210 [76.982303] do_shrink_slab+0x19e/0x4e0 [76.982310] shrink_slab+0x2bd/0x450 [76.982317] drop_slab+0xcc/0x1a0 [76.982323] drop_caches_sysctl_handler+0xb7/0xe0 [76.982327] proc_sys_call_handler+0x1bc/0x300 [76.982331] proc_sys_write+0x17/0x20 [76.982334] vfs_write+0x3d3/0x570 [76.982342] ksys_write+0x73/0x160 [76.982347] __x64_sys_write+0x1e/0x30 [76.982352] do_syscall_64+0x35/0x80 [76.982357] entry_SYSCALL_64_after_hwframe+0x63/0xcd Funct ---truncated---
CVE-2022-50550	In the Linux kernel, the following vulnerability has been resolved: blk-iolatency: Fix memory leak on add_disk() failures When a gendisk is successfully initialized but add_disk() fails such as when a loop device has invalid number of minor device numbers specified, blkcg_init_disk() is called during init and then blkcg_exit_disk() during error handling. Unfortunately, iolatency gets initialized in the former but doesn't get cleaned up in the latter. This is because, in non-error cases, the cleanup is performed by del_gendisk() calling rq_qos_exit(), the assumption being that rq_qos policies, iolatency being one of them, can only be activated once the disk is fully registered and visible. That assumption is true for wbt and iocost, but not so for iolatency as it gets initialized before add_disk() is called. It is desirable to lazy-init rq_qos policies because they are optional features and add to hot path overhead once initialized - each IO has to walk all the registered rq_qos policies. So, we want to switch iolatency to lazy init too. However, that's a bigger change. As a fix for the immediate problem, let's just add an extra call to rq_qos_exit() in blkcg_exit_disk(). This is safe because duplicate calls to rq_qos_exit() become noop's.

CVE ID	Description
CVE-2022-50551	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Fix potential shift-out-of-bounds in brcmf_fw_alloc_request() This patch fixes a shift-out-of-bounds in brcmfmac that occurs in BIT(chiprev) when a 'chiprev' provided by the device is too large. It should also not be equal to or greater than BITS_PER_TYPE(u32) as we do bitwise AND with a u32 variable and BIT(chiprev). The patch adds a check that makes the function return NULL if that is the case. Note that the NULL case is later handled by the bus-specific caller, brcmf_usb_probe_cb() or brcmf_usb_reset_resume(), for example. Found by a modified version of syzkaller. UBSAN: shift-out-of-bounds in drivers/net/wireless/broadcom/brcm80211/brcmfmac/firmware.c shift exponent 151055786 is too large for 64-bit type 'long unsigned int' CPU: 0 PID: 1885 Comm: kworker/0:2 Tainted: G O 5.14.0+ #132 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org 04/01/2014 Workqueue: usb_hub_wq hub_event Call Trace: dump_stack_lvl+0x57/0x7d ubsan_epilogue+0x5/0x40 __ubsan_handle_shift_out_of_bounds.cold+0x53/0xdb ? lock_chain_count+0x20/0x20 brcmf_fw_alloc_request.cold+0x19/0x3ea ? brcmf_fw_get_firmwares+0x250/0x250 ? brcmf_usb_ioctl_resp_wait+0x1a7/0x1f0 brcmf_usb_get_fwname+0x114/0x1a0 ? brcmf_usb_reset_resume+0x120/0x120 ? number+0x6c4/0x9a0 brcmf_c_process_clm_blob+0x168/0x590 ? put_dec+0x90/0x90 ? enable_ptr_key_workfn+0x20/0x20 ? brcmf_common_pd_remove+0x50/0x50 ? rcu_read_lock_sched_held+0xa1/0xd0 brcmf_c_preinit_dcmds+0x673/0xc40 ? brcmf_c_set_joinpref_default+0x100/0x100 ? rcu_read_lock_sched_held+0xa1/0xd0 ? rcu_read_lock_bh_held+0xb0/0xb0 ? lock_acquire+0x19d/0x4e0 ? find_held_lock+0x2d/0x110 ? brcmf_usb_deq+0x1cc/0x260 ? mark_held_locks+0x9f/0xe0 ? lockdep_hardirqs_on_prepare+0x273/0x3e0 ? _raw_spin_unlock_irqrestore+0x47/0x50 ? trace_hardirqs_on+0x1c/0x120 ? brcmf_usb_deq+0x1a7/0x260 ? brcmf_usb_rx_fill_all+0x5a/0xf0 brcmf_attach+0x246/0xd40 ? wiphy_new_nm+0x1476/0x1d50 ? kmemdup+0x30/0x40 brcmf_usb_probe+0x12de/0x1690 ? brcmf_usbdev_qinit.constprop.0+0x470/0x470 usb_probe_interface+0x25f/0x710 really_probe+0x1be/0xa90 __driver_probe_device+0x2ab/0x460 ? usb_match_id.part.0+0x88/0xc0 driver_probe_device+0x49/0x120 __device_attach_driver+0x18a/0x250 ? driver_allows_async_probing+0x120/0x120 bus_for_each_drv+0x123/0x1a0 ? bus_rescan_devices+0x20/0x20 ? lockdep_hardirqs_on_prepare+0x273/0x3e0 ? trace_hardirqs_on+0x1c/0x120 __device_attach+0x207/0x330 ? device_bind_driver+0xb0/0xb0 ? kobject_uevent_env+0x230/0x12c0 bus_probe_device+0x1a2/0x260 device_add+0xa61/0x1ce0 ? __mutex_unlock_slowpath+0xe7/0x660 ? __fw_devlink_link_to_suppliers+0x550/0x550 usb_set_configuration+0x984/0x1770 ? kernfs_create_link+0x175/0x230 usb_generic_driver_probe+0x69/0x90 usb_probe_device+0x9c/0x220 really_probe+0x1be/0xa90 __driver_probe_device+0x2ab/0x460 driver_probe_device+0x49/0x120 __device_attach_driver+0x18a/0x250 ? driver_allows_async_probing+0x120/0x120 bus_for_each_drv+0x123/0x1a0 ? bus_rescan_devices+0x20/0x20 ? lockdep_hardirqs_on_prepare+0x273/0x3e0 ? trace_hardirqs_on+0x1c/0x120 __device_attach+0x207/0x330 ? device_bind_driver+0xb0/0xb0 ? kobject_uevent_env+0x230/0x12c0 bus_probe_device+0x1a2/0x260 device_add+0xa61/0x1ce0 ? __fw_devlink_link_to_suppliers+0x550/0x550 usb_new_device.cold+0x463/0xf66 ? hub_disconnect+0x400/0x400 ? _raw_spin_unlock_irq+0x24/0x30 hub_event+0x10d5/0x3330 ? hub_port_debounce+0x280/0x280 ? __lock_acquire+0x1671/0x5790 ? wq_calc_node_cpumask+0x170/0x2a0 ? lock_release+0x640/0x640 ? rcu_read_lock_sched_held+0xa1/0xd0 ? rcu_read_lock_bh_held+0xb0/0xb0 ? lockdep_hardirqs_on_prepare+0x273/0x3e0 process_one_work+0x873/0x13e0 ? lock_release+0x640/0x640 ? pwq_dec_nr_in_flight+0x320/0x320 ? rwlock_bug.part.0+0x90/0x90 worker_thread+0x8b/0xd10 ? __kthread_parkme+0xd9/0x1d0 ? pr ---truncated---
CVE-2022-50552	In the Linux kernel, the following vulnerability has been resolved: blk-mq: use quiesced elevator switch when reinitializing queues The hctx's run_work may be racing with the elevator switch when reinitializing hardware queues. The queue is merely frozen in this context, but that only prevents requests from allocating and doesn't stop the hctx work from running. The work may get an elevator pointer that's being torn down, and can result in use-after-free errors and kernel panics (example below). Use the quiesced elevator switch instead, and make the previous one static since it is now only used locally. nvme nvme0: resetting controller nvme nvme0: 32/0/0 default/read/poll queues BUG: kernel NULL pointer dereference, address: 0000000000000008 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 80000020c8861067 P4D 80000020c8861067 PUD 250f8c8067 PMD 0 Oops: 0000 [#1] SMP PTI Workqueue: kblockd blk_mq_run_work_fn RIP: 0010:kyber_has_work+0x29/0x70 ... Call Trace: __blk_mq_do_dispatch_sched+0x83/0x2b0 __blk_mq_sched_dispatch_requests+0x12e/0x170 blk_mq_sched_dispatch_requests+0x30/0x60 __blk_mq_run_hw_queue+0x2b/0x50 process_one_work+0x1ef/0x380 worker_thread+0x2d/0x3e0

CVE ID	Description
CVE-2022-50554	In the Linux kernel, the following vulnerability has been resolved: blk-mq: avoid double ->queue_rq() because of early timeout David Jeffery found one double ->queue_rq() issue, so far it can be triggered in VM use case because of long vmexit latency or preempt latency of vCPU pthread or long page fault in vCPU pthread, then block IO req could be timed out before queuing the request to hardware but after calling blk_mq_start_request() during ->queue_rq(), then timeout handler may handle it by requeue, then double ->queue_rq() is caused, and kernel panic. So far, it is driver's responsibility to cover the race between timeout and completion, so it seems supposed to be solved in driver in theory, given driver has enough knowledge. But it is really one common problem, lots of driver could have similar issue, and could be hard to fix all affected drivers, even it isn't easy for driver to handle the race. So David suggests this patch by draining in-progress ->queue_rq() for solving this issue.
CVE-2022-50556	In the Linux kernel, the following vulnerability has been resolved: drm: Fix potential null-ptr-deref due to drmm_mode_config_init() drmm_mode_config_init() will call drm_mode_create_standard_properties() and won't check the ret value. When drm_mode_create_standard_properties() failed due to alloc, property will be a NULL pointer and may causes the null-ptr-deref. Fix the null-ptr-deref by adding the ret value check. Found null-ptr-deref while testing insert module bochs: general protection fault, probably for non-canonical address 0xdffffc000000000c: 0000 [0000] SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000060-0x0000000000000067] CPU: 3 PID: 249 Comm: modprobe Not tainted 6.1.0-rc1+ #364 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.15.0-0-g2dd4b9b3f840-prebuilt.qemu.org 04/01/2014 RIP: 0010:drm_object_attach_property+0x73/0x3c0 [drm] Call Trace: <TASK> __drm_connector_init+0xb6c/0x1100 [drm] bochs_pci_probe.cold.11+0x4cb/0x7fe [bochs] pci_device_probe+0x17d/0x340 really_probe+0x1db/0x5d0 __driver_probe_device+0x1e7/0x250 driver_probe_device+0x4a/0x120 __driver_attach+0xcd/0x2c0 bus_for_each_dev+0x11a/0x1b0 bus_add_driver+0x3d7/0x500 driver_register+0x18e/0x320 do_one_initcall+0xc4/0x3e0 do_init_module+0x1b4/0x630 load_module+0x5dca/0x7230 __do_sys_finit_module+0x100/0x170 do_syscall_64+0x3f/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7ff65af9f839
CVE-2022-50558	In the Linux kernel, the following vulnerability has been resolved: regmap-irq: Use the new num_config_regs property in regmap_add_irq_chip_fwnode Commit faa87ce9196d ("regmap-irq: Introduce config registers for irq types") added the num_config_regs, then commit 9edd4f5aee84 ("regmap-irq: Deprecate type registers and virtual registers") suggested to replace num_type_reg with it. However, regmap_add_irq_chip_fwnode wasn't modified to use the new property. Later on, commit 255a03bb1bb3 ("ASoC: wcd9335: Convert irq chip to config regs") removed the old num_type_reg property from the WCD9335 driver's struct regmap_irq_chip, causing a null pointer dereference in regmap_irq_set_type when it tried to index d->type_buf as it was never allocated in regmap_add_irq_chip_fwnode: [39.199374] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 [39.200006] Call trace: [39.200014] regmap_irq_set_type+0x84/0x1c0 [39.200026] __irq_set_trigger+0x60/0x1c0 [39.200040] __setup_irq+0x2f4/0x78c [39.200051] request_threaded_irq+0xe8/0x1a0 Use num_config_regs in regmap_add_irq_chip_fwnode instead of num_type_reg, and fall back to it if num_config_regs isn't defined to maintain backward compatibility.
CVE-2022-50561	In the Linux kernel, the following vulnerability has been resolved: iio: fix memory leak in iio_device_register_eventset() When iio_device_register_sysfs_group() returns failed, iio_device_register_eventset() needs to free attrs array. Otherwise, kmemleak would scan & report memory leak as below: unreferenced object 0xffff88810a1cc3c0 (size 32): comm "100-i2c-vcnl302", pid 728, jiffies 4295052307 (age 156.027s) backtrace: __kmalloc+0x46/0x1b0 iio_device_register_eventset at drivers/iio/industrialio-event.c:541 __iio_device_register at drivers/iio/industrialio-core.c:1959 __devm_iio_device_register at drivers/iio/industrialio-core.c:2040
CVE-2022-50562	In the Linux kernel, the following vulnerability has been resolved: tpm: acpi: Call acpi_put_table() to fix memory leak The start and length of the event log area are obtained from TPM2 or TCGA table, so we call acpi_get_table() to get the ACPI information, but the acpi_get_table() should be coupled with acpi_put_table() to release the ACPI memory, add the acpi_put_table() properly to fix the memory leak. While we are at it, remove the redundant empty line at the end of the tpm_read_log_acpi().

CVE ID	Description
CVE-2022-50563	In the Linux kernel, the following vulnerability has been resolved: dm thin: Fix UAF in run_timer_softirq() When dm_resume() and dm_destroy() are concurrent, it will lead to UAF, as follows: BUG: KASAN: use-after-free in __run_timers+0x173/0x710 Write of size 8 at addr ffff88816d9490f0 by task swapper/0/0 <snip> Call Trace: <IRQ> dump_stack_lvl+0x73/0x9f print_report.cold+0x132/0xaa2 __raw_spin_lock_irqsave+0xcd/0x160 __run_timers+0x173/0x710 kasan_report+0xad/0x110 __run_timers+0x173/0x710 __asan_store8+0x9c/0x140 __run_timers+0x173/0x710 call_timer_fn+0x310/0x310 pvclock_clocksource_read+0xfa/0x250 kvm_clock_read+0x2c/0x70 kvm_clock_get_cycles+0xd/0x20 ktime_get+0x5c/0x110 lapic_next_event+0x38/0x50 clockevents_program_event+0xf1/0x1e0 run_timer_softirq+0x49/0x90 __do_softirq+0x16e/0x62c __irq_exit_rcu+0x1fa/0x270 irq_exit_rcu+0x12/0x20 sysvec_apic_timer_interrupt+0x8e/0xc0 One of the concurrency UAF can be shown as below: use free do_resume __find_device_hash_cell dm_get atomic_inc(&md->holders) dm_destroy __dm_destroy if (!dm_suspended_md(md)) atomic_read(&md->holders) msleep(1) dm_resume __dm_resume dm_table_resume_targets pool_resume do_waker #add delay work dm_put atomic_dec(&md->holders) dm_table_destroy pool_dtr __pool_dec __pool_destroy destroy_workqueue kfree(pool) # free pool time out __do_softirq run_timer_softirq # pool has already been freed This can be easily reproduced using: 1. create thin-pool 2. dmsetup suspend pool 3. dmsetup resume pool 4. dmsetup remove_all # Concurrent with 3 The root cause of this UAF bug is that dm_resume() adds timer after dm_destroy() skips cancelling the timer because of suspend status. After timeout, it will call run_timer_softirq(), however pool has already been freed. The concurrency UAF bug will happen. Therefore, cancelling timer again in __pool_destroy().
CVE-2022-50566	In the Linux kernel, the following vulnerability has been resolved: mtd: Fix device name leak when register device failed in add_mtd_device() There is a kmemleak when register device failed: unreferenced object 0xffff888101aab550 (size 8): comm "insmod", pid 3922, jiffies 4295277753 (age 925.408s) hex dump (first 8 bytes): 6d 74 64 30 00 88 ff ff mtd0.... backtrace: [<00000000bde26724>] __kmalloct_node_track_caller+0x4e/0x150 [<000000003c32b416>] kvasprintf+0xb0/0x130 [<000000001f7a8f15>] kobject_set_name_vargs+0x2f/0xb0 [<000000006e781163>] dev_set_name+0xab/0xe0 [<00000000e30d0c78>] add_mtd_device+0x4bb/0x700 [<00000000f3d34de7>] mtd_device_parse_register+0x2ac/0x3f0 [<00000000c0d88488>] 0xffffffffa0238457 [<00000000b40d0922>] 0xffffffffa02a008f [<0000000023d17b9d>] do_one_initcall+0x87/0x2a0 [<00000000770f6ca6>] do_init_module+0xdf/0x320 [<000000007b6768fe>] load_module+0x2f98/0x3330 [<00000000346bed5a>] __do_sys_finit_module+0x113/0x1b0 [<00000000674c2290>] do_syscall_64+0x35/0x80 [<000000004c6a8d97>] entry_SYSCALL_64_after_hwframe+0x46/0xb0 If register device failed, should call put_device() to give up the reference.
CVE-2022-50569	In the Linux kernel, the following vulnerability has been resolved: xfrm: Update ipcomp_scratches with NULL when freed Currently if ipcomp_alloc_scratches() fails to allocate memory ipcomp_scratches holds obsolete address. So when we try to free the percpu scratches using ipcomp_free_scratches() it tries to vfree non existent vm area. Described below: static void *__percpu *ipcomp_alloc_scratches(void) { ... scratches = alloc_percpu(void *); if (!scratches) return NULL; ipcomp_scratches does not know about this allocation failure. Therefore holding the old obsolete address. ... } So when we free, static void ipcomp_free_scratches(void) { ... scratches = ipcomp_scratches; Assigning obsolete address from ipcomp_scratches if (!scratches) return; for_each_possible_cpu(i) vfree(*per_cpu_ptr(scratches, i)); Trying to free non existent page, causing warning: trying to vfree existent vm area. ... } Fix this breakage by updating ipcomp_scratches with NULL when scratches is freed
CVE-2022-50575	In the Linux kernel, the following vulnerability has been resolved: xen/privcmd: Fix a possible warning in privcmd_ioctl_mmap_resource() As 'kdata.num' is user-controlled data, if user tries to allocate memory larger than(>=) MAX_ORDER, then kcalloc() will fail, it creates a stack trace and messes up dmesg with a warning. Call trace: -> privcmd_ioctl --> privcmd_ioctl_mmap_resource Add __GFP_NOWARN in order to avoid too large allocation warning. This is detected by static analysis using smatch.
CVE-2022-50577	In the Linux kernel, the following vulnerability has been resolved: ima: Fix memory leak in __ima_inode_hash() Commit f3cc6b25dcc5 ("ima: always measure and audit files in policy") lets measurement or audit happen even if the file digest cannot be calculated. As a result, iint->ima_hash could have been allocated despite ima_collect_measurement() returning an error. Since ima_hash belongs to a temporary inode metadata structure, declared at the beginning of __ima_inode_hash(), just add a kfree() call if ima_collect_measurement() returns an error different from -ENOMEM (in that case, ima_hash should not have been allocated).

CVE ID	Description
CVE-2022-50578	In the Linux kernel, the following vulnerability has been resolved: class: fix possible memory leak in __class_register() If class_add_groups() returns error, the 'cp->subsys' need be unregister, and the 'cp' need be freed. We can not call kset_unregister() here, because the 'cls' will be freed in callback function class_release() and it's also freed in caller's error path, it will cause double free. So fix this by calling kobject_del() and kfree_const(name) to cleanup kobject. Besides, call kfree() to free the 'cp'. Fault injection test can trigger this: unreferenced object 0xffff888102fa8190 (size 8): comm "modprobe", pid 502, jiffies 4294906074 (age 49.296s) hex dump (first 8 bytes): 70 6b 74 63 64 76 64 00 pktcdvd. backtrace: [<00000000e7c7703d>] __kmalloc_track_caller+0x1ae/0x320 [<000000005e4d70bc>] kstrdup+0x3a/0x70 [<00000000c2e5e85a>] kstrdup_const+0x68/0x80 [<00000000049a8c7>] kvasprintf_const+0x10b/0x190 [<0000000029123163>] kobject_set_name_vars+0x56/0x150 [<00000000747219c9>] kobject_set_name+0xab/0xe0 [<0000000005f1ea4e>] __class_register+0x15c/0x49a unreferenced object 0xffff888037274000 (size 1024): comm "modprobe", pid 502, jiffies 4294906074 (age 49.296s) hex dump (first 32 bytes): 00 40 27 37 80 88 ff ff 00 40 27 37 80 88 ff ff .@'7.....@'7.... 00 00 00 00 ad 4e ad de ff ff ff 00 00 00 00 N..... backtrace: [<000000000151f9600>] kmem_cache_alloc_trace+0x17c/0x2f0 [<00000000ecf3dd95>] __class_register+0x86/0x49a
CVE-2022-50579	In the Linux kernel, the following vulnerability has been resolved: arm64: ftrace: fix module PLTs with mcount Li Huafei reports that mcount-based ftrace with module PLTs was broken by commit: a6253579977e4c6f ("arm64: ftrace: consistently handle PLTs.") When a module PLTs are used and a module is loaded sufficiently far away from the kernel, we'll create PLTs for any branches which are out-of-range. These are separate from the special ftrace trampoline PLTs, which the module PLT code doesn't directly manipulate. When mcount is in use this is a problem, as each mcount callsite in a module will be initialized to point to a module PLT, but since commit a6253579977e4c6f ftrace_make_nop() will assume that the callsite has been initialized to point to the special ftrace trampoline PLT, and ftrace_find_callable_addr() rejects other cases. This means that when ftrace tries to initialize a callsite via ftrace_make_nop(), the call to ftrace_find_callable_addr() will find that the `mcount` stub is out-of-range and is not handled by the ftrace PLT, resulting in a splat: ftrace_test: loading out-of-tree module taints kernel. ftrace: no module PLT for _mcount -----[ftrace bug]----- ftrace failed to modify [<ffff800029180014>] #22="" (0)="" (dt)="" (nzcv="")="" -----[="" -----test_init-----="" ---[="" -dit="" -pan="" -ssbs="" -tco="" -uao="" 0000000000000000="" 0000000000000001="" 0000000000000022="" 000000000000bfff4="" 0000000000017fe8="" 0000000000057fa8="" 0000000002000000="" 0a0d2d2d2d2d2d2d="" 0x100="" 0x118="" 0x160="" 0x1ce8="" 0x238="" 0x270="" 0x30="" 0x430="" 0x50="" 0x60="" 0xb8="" 0xd0="" 0xffff800029180014="" 157="" 2000000="" 203a706d61727420="" 3="" 3030303866666666="" 3362653230383030="" 44:00:00:94="" 6.0.0-rc6-00151-gcd722513a189-dirty="" 60000005="" 6333626532303830="" 6465746365707865="" :=""]---=""]-----="" __arm64_sys_init_module+0x24="" __do_sys_init_module+0x1ec="" a="" a6253579977e4c6f.<="" actual:="" ad2cb14bb5438900="" an="" at="" behaviour="" btype="--" by="" c0000000ffffefff="" call="" callsite="" comm:="" commit="" cpu:="" cut="" daif="" do_el0_svc+0x3c="" el0_svc+0x1c="" el0_svc_common.constprop.4+0x84="" el0t_64_sync+0x15c="" el0t_64_sync_handler+0x90="" end="" expected="" ffff0000c4d38000="" ffff0000c4d86e00="" ffff80000802eb3c="" ffff8000098e8ea8="" ffff800009aa9118="" ffff800009d7e000="" ffff80000a62b000="" ffff80000aa24158="" ffff80000b2bbaf0="" ffffffff00000000="" fix="" flags:="" ftrace="" ftrace.c:2120="" ftrace_bug+0x21c="" ftrace_bug+0x94="" ftrace_module_init+0x44="" ftrace_process_locs+0x308="" g="" hardware="" here="" ignoring="" in="" in:="" initialising="" initializing="" insmod="" instruction="" invoke_syscall+0x54="" kernel="" linked="" linux,dummy-virt="" load_module+0x15b4="" lr="" mcount="" module,="" modules="" name:="" o="" of="" old="" pc="" pid:="" prior="" pstate:="" record="" reverting="" sites="" sp="" tainted:="" td="" the="" this="" to="" trace="" trace:="" tramp:="" warning:="" was="" when="" which="" x0="" x1="" x10:="" x11:="" x12:="" x13:="" x14:="" x15:="" x16:="" x17:="" x18:="" x19:="" x2="" x20:="" x21:="" x22:="" x23:="" x24:="" x25:="" x26:="" x27:="" x28:="" x29:="" x3="" x4="" x5="" x6="" x7="" x8="" x9="" =""></ffff800029180014>]>
CVE-2022-50580	In the Linux kernel, the following vulnerability has been resolved: blk-throttle: prevent overflow while calculating wait time There is a problem found by code review in tg_with_in_bps_limit() that 'bps_limit * jiffy_elapsed_rnd' might overflow. Fix the problem by calling mul_u64_u64_div_u64() instead.
CVE-2023-0030	A use-after-free flaw was found in the Linux kernel's nouveau driver in how a user triggers a memory overflow that causes the nvkm_vma_tail function to fail. This flaw allows a local user to crash or potentially escalate their privileges on the system.
CVE-2023-0045	The current implementation of the prctl syscall does not issue an IBPB immediately during the syscall. The ib_prctl_set function updates the Thread Information Flags (TIFs) for the task and updates the SPEC_CTRL MSR on the function __speculation_ctrl_update, but the IBPB is only issued on the next schedule, when the TIF bits are checked. This leaves the victim vulnerable to values already injected on the BTB, prior to the prctl syscall. The patch that added the support for the conditional mitigation via prctl (ib_prctl_set) dates back to the kernel 4.9.176. We recommend upgrading past commit a664ec9158eadd75121d39c9a0758016097fa96
CVE-2023-0049	Out-of-bounds Read in GitHub repository vim/vim prior to 9.0.1143.

CVE ID	Description
CVE-2023-0054	Out-of-bounds Write in GitHub repository vim/vim prior to 9.0.1145.
CVE-2023-0160	A deadlock flaw was found in the Linux kernel's BPF subsystem. This flaw allows a local user to potentially crash the system.
CVE-2023-0288	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1189.
CVE-2023-0433	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1225.
CVE-2023-0464	A security vulnerability has been identified in all supported versions of OpenSSL related to the verification of X.509 certificate chains that include policy constraints. Attackers may be able to exploit this vulnerability by creating a malicious certificate chain that triggers exponential use of computational resources, leading to a denial-of-service (DoS) attack on affected systems. Policy processing is disabled by default but can be enabled by passing the <code>'-policy'</code> argument to the command line utilities or by calling the <code>X509_VERIFY_PARAM_set1_policies()</code> function.
CVE-2023-0465	Applications that use a non-default option when verifying certificates may be vulnerable to an attack from a malicious CA to circumvent certain checks. Invalid certificate policies in leaf certificates are silently ignored by OpenSSL and other certificate policy checks are skipped for that certificate. A malicious CA could use this to deliberately assert invalid certificate policies in order to circumvent policy checking on the certificate altogether. Policy processing is disabled by default but can be enabled by passing the <code>'-policy'</code> argument to the command line utilities or by calling the <code>X509_VERIFY_PARAM_set1_policies()</code> function.
CVE-2023-0466	The function <code>X509_VERIFY_PARAM_add0_policy()</code> is documented to implicitly enable the certificate policy check when doing certificate verification. However the implementation of the function does not enable the check which allows certificates with invalid or incorrect policies to pass the certificate verification. As suddenly enabling the policy check could break existing deployments it was decided to keep the existing behavior of the <code>X509_VERIFY_PARAM_add0_policy()</code> function. Instead the applications that require OpenSSL to perform certificate policy check need to use <code>X509_VERIFY_PARAM_set1_policies()</code> or explicitly enable the policy check by calling <code>X509_VERIFY_PARAM_set_flags()</code> with the <code>X509_V_FLAG_POLICY_CHECK</code> flag argument. Certificate policy checks are disabled by default in OpenSSL and are not commonly used by applications.
CVE-2023-0512	Divide By Zero in GitHub repository vim/vim prior to 9.0.1247.
CVE-2023-0645	An out of bounds read exists in libjxl. An attacker using a specifically crafted file could cause an out of bounds read in the exif handler. We recommend upgrading to version 0.8.1 or past commit https://github.com/libjxl/libjxl/pull/2101/commits/d95b050c1822a5b1ede9e0dc937e43fca1b10159 https://github.com/libjxl/libjxl/pull/2101/commits/d95b050c1822a5b1ede9e0dc937e43fca1b10159
CVE-2023-0795	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3488, allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile libtiff from sources, the fix is available with commit afaabc3e.
CVE-2023-0796	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3592, allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile libtiff from sources, the fix is available with commit afaabc3e.
CVE-2023-0797	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in libtiff/tif_unix.c:368, invoked by tools/tiffcrop.c:2903 and tools/tiffcrop.c:6921, allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile libtiff from sources, the fix is available with commit afaabc3e.
CVE-2023-0798	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3400, allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile libtiff from sources, the fix is available with commit afaabc3e.
CVE-2023-0799	LibTIFF 4.4.0 has an out-of-bounds read in tiffcrop in tools/tiffcrop.c:3701, allowing attackers to cause a denial-of-service via a crafted tiff file. For users that compile libtiff from sources, the fix is available with commit afaabc3e.
CVE-2023-1076	A flaw was found in the Linux Kernel. The tun/tap sockets have their socket UID hardcoded to 0 due to a type confusion in their initialization function. While it will be often correct, as tuntap devices require CAP_NET_ADMIN, it may not always be the case, e.g., a non-root user only having that capability. This would make tun/tap sockets being incorrectly treated in filtering/routing decisions, possibly bypassing network filters.
CVE-2023-1077	In the Linux kernel, <code>pick_next_rt_entity()</code> may return a type confused entry, not detected by the <code>BUG_ON</code> condition, as the confused entry will not be NULL, but <code>list_head</code> . The buggy error condition would lead to a type confused entry with the list head, which would then be used as a type confused <code>sched_rt_entity</code> , causing memory corruption.

CVE ID	Description
CVE-2023-1127	Divide By Zero in GitHub repository vim/vim prior to 9.0.1367.
CVE-2023-1170	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1376.
CVE-2023-1175	Incorrect Calculation of Buffer Size in GitHub repository vim/vim prior to 9.0.1378.
CVE-2023-1193	A use-after-free flaw was found in <code>setup_async_work</code> in the KSMBD implementation of the in-kernel samba server and CIFS in the Linux kernel. This issue could allow an attacker to crash the system by accessing freed work.
CVE-2023-1264	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.1392.
CVE-2023-1370	[Json-smart](https://netplex.github.io/json-smart/) is a performance focused, JSON processor lib. When reaching a '[' or '{' character in the JSON input, the code parses an array or an object respectively. It was discovered that the code does not have any limit to the nesting of such arrays or objects. Since the parsing of nested arrays and objects is done recursively, nesting too many of them can cause a stack exhaustion (stack overflow) and crash the software.
CVE-2023-1380	A slab-out-of-bound read problem was found in <code>brcmf_get_assoc_ies</code> in <code>drivers/net/wireless/broadcom/brcm80211/brcmfmac/cfg80211.c</code> in the Linux Kernel. This issue could occur when <code>assoc_info->req_len</code> data is bigger than the size of the buffer, defined as <code>WL_EXTRA_BUF_MAX</code> , leading to a denial of service.
CVE-2023-1916	A flaw was found in <code>tiffcrop</code> , a program distributed by the <code>libtiff</code> package. A specially crafted tiff file can lead to an out-of-bounds read in the <code>extractImageSection</code> function in <code>tools/tiffcrop.c</code> , resulting in a denial of service and limited information disclosure. This issue affects <code>libtiff</code> versions 4.x.
CVE-2023-1932	A flaw was found in <code>hibernate-validator</code> 's <code>isValid</code> method in the <code>org.hibernate.validator.internal.constraintvalidators.hv.SafeHtmlValidator</code> class, which can be bypassed by omitting the tag ending in a less-than character. Browsers may render an invalid html, allowing HTML injection or Cross-Site-Scripting (XSS) attacks.
CVE-2023-1972	A potential heap based buffer overflow was found in <code>_bfd_elf_slurp_version_tables()</code> in <code>bfd/elf.c</code> . This may lead to loss of availability.
CVE-2023-20859	In Spring Vault, versions 3.0.x prior to 3.0.2 and versions 2.3.x prior to 2.3.3 and older versions, an application is vulnerable to insertion of sensitive information into a log file when it attempts to revoke a Vault batch token.
CVE-2023-20863	In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
CVE-2023-2253	A flaw was found in the <code>`/v2/_catalog`</code> endpoint in <code>distribution/distribution</code> , which accepts a parameter to control the maximum number of records returned (query string: <code>`n`</code>). This vulnerability allows a malicious user to submit an unreasonably large value for <code>`n`</code> , causing the allocation of a massive string array, possibly causing a denial of service through excessive use of memory.
CVE-2023-22656	Out-of-bounds read in Intel(R) Media SDK and some Intel(R) oneVPL software before version 23.3.5 may allow an authenticated user to potentially enable escalation of privilege via local access.
CVE-2023-24056	In <code>pkgconf</code> through 1.9.3, variable duplication can cause unbounded string expansion due to incorrect checks in <code>libpkgconf/tuple.c:pkgconf_tuple_parse</code> . For example, a <code>.pc</code> file containing a few hundred bytes can expand to one billion bytes.
CVE-2023-24532	The <code>ScalarMult</code> and <code>ScalarBaseMult</code> methods of the P256 Curve may return an incorrect result if called with some specific unreduced scalars (a scalar larger than the order of the curve). This does not impact usages of <code>crypto/ecdsa</code> or <code>crypto/ecdh</code> .
CVE-2023-24534	HTTP and MIME header parsing can allocate large amounts of memory, even when parsing small inputs, potentially leading to a denial of service. Certain unusual patterns of input data can cause the common function used to parse HTTP and MIME headers to allocate substantially more memory than required to hold the parsed headers. An attacker can exploit this behavior to cause an HTTP server to allocate large amounts of memory from a small request, potentially leading to memory exhaustion and a denial of service. With fix, header parsing now correctly allocates only the memory required to hold parsed headers.

CVE ID	Description
CVE-2023-24536	Multipart form parsing can consume large amounts of CPU and memory when processing form inputs containing very large numbers of parts. This stems from several causes: 1. mime/multipart.Reader.ReadForm limits the total memory a parsed multipart form can consume. ReadForm can undercount the amount of memory consumed, leading it to accept larger inputs than intended. 2. Limiting total memory does not account for increased pressure on the garbage collector from large numbers of small allocations in forms with many parts. 3. ReadForm can allocate a large number of short-lived buffers, further increasing pressure on the garbage collector. The combination of these factors can permit an attacker to cause an program that parses multipart forms to consume large amounts of CPU and memory, potentially resulting in a denial of service. This affects programs that use mime/multipart.Reader.ReadForm, as well as form parsing in the net/http package with the Request methods FormFile, FormValue, ParseMultipartForm, and PostFormValue. With fix, ReadForm now does a better job of estimating the memory consumption of parsed forms, and performs many fewer short-lived allocations. In addition, the fixed mime/multipart.Reader imposes the following limits on the size of parsed forms: 1. Forms parsed with ReadForm may contain no more than 1000 parts. This limit may be adjusted with the environment variable GODEBUG=multipartmaxparts=. 2. Form parts parsed with NextPart and NextRawPart may contain no more than 10,000 header fields. In addition, forms parsed with ReadForm may contain no more than 10,000 header fields across all parts. This limit may be adjusted with the environment variable GODEBUG=multipartmaxheaders=.
CVE-2023-24537	Calling any of the Parse functions on Go source code which contains //line directives with very large line numbers can cause an infinite loop due to integer overflow.
CVE-2023-24538	Templates do not properly consider backticks (``) as Javascript string delimiters, and do not escape them as expected. Backticks are used, since ES6, for JS template literals. If a template contains a Go template action within a Javascript template literal, the contents of the action can be used to terminate the literal, injecting arbitrary Javascript code into the Go template. As ES6 template literals are rather complex, and themselves can do string interpolation, the decision was made to simply disallow Go template actions from being used inside of them (e.g. "var a = {{.}}"), since there is no obviously safe way to allow this behavior. This takes the same approach as github.com/google/safehtml . With fix, Template.Parse returns an Error when it encounters templates like this, with an ErrorCode of value 12. This ErrorCode is currently unexported, but will be exported in the release of Go 1.21. Users who rely on the previous behavior can re-enable it using the GODEBUG flag jstmpllitinterp=1, with the caveat that backticks will now be escaped. This should be used with caution.
CVE-2023-24539	Angle brackets (<>) are not considered dangerous characters when inserted into CSS contexts. Templates containing multiple actions separated by a '/' character can result in unexpectedly closing the CSS context and allowing for injection of unexpected HTML, if executed with untrusted input.
CVE-2023-24540	Not all valid JavaScript whitespace characters are considered to be whitespace. Templates containing whitespace characters outside of the character set "\t\n\r\u0020\u2028\u2029" in JavaScript contexts that also contain actions may not be properly sanitized during execution.
CVE-2023-24607	Qt before 6.4.3 allows a denial of service via a crafted string when the SQL ODBC driver plugin is used and the size of SQLTCHAR is 4. The affected versions are 5.x before 5.15.13, 6.x before 6.2.8, and 6.3.x before 6.4.3.
CVE-2023-25434	libtiff 4.5.0 is vulnerable to Buffer Overflow via extractContigSamplesBytes() at /libtiff/tools/tiffcrop.c:3215.
CVE-2023-25435	libtiff 4.5.0 is vulnerable to Buffer Overflow via extractContigSamplesShifted8bits() at /libtiff/tools/tiffcrop.c:3753.
CVE-2023-25584	An out-of-bounds read flaw was found in the parse_module function in bfd/vms-alpha.c in Binutils.
CVE-2023-25585	A flaw was found in Binutils. The use of an uninitialized field in the struct module *module may lead to application crash and local denial of service.
CVE-2023-25588	A flaw was found in Binutils. The field `the_bfd` of `asymbol` struct is uninitialized in the `bfd_mach_o_get_synthetic_symtab` function, which may lead to an application crash and local denial of service.

CVE ID	Description
CVE-2023-26048	Jetty is a java based web server and servlet engine. In affected versions servlets with multipart support (e.g. annotated with <code>@MultipartConfig</code>) that call <code>HttpServletRequest.getParameter()</code> or <code>HttpServletRequest.getParts()</code> may cause <code>OutOfMemoryError</code> when the client sends a multipart request with a part that has a name but no filename and very large content. This happens even with the default settings of <code>fileSizeThreshold=0</code> which should stream the whole part content to disk. An attacker client may send a large multipart request and cause the server to throw <code>OutOfMemoryError</code> . However, the server may be able to recover after the <code>OutOfMemoryError</code> and continue its service -- although it may take some time. This issue has been patched in versions 9.4.51, 10.0.14, and 11.0.14. Users are advised to upgrade. Users unable to upgrade may set the multipart parameter <code>maxRequestSize</code> which must be set to a non-negative value, so the whole multipart content is limited (although still read into memory).
CVE-2023-26049	Jetty is a java based web server and servlet engine. Nonstandard cookie parsing in Jetty may allow an attacker to smuggle cookies within other cookies, or otherwise perform unintended behavior by tampering with the cookie parsing mechanism. If Jetty sees a cookie VALUE that starts with <code>""</code> (double quote), it will continue to read the cookie string until it sees a closing quote -- even if a semicolon is encountered. So, a cookie header such as: <code>DISPLAY_LANGUAGE="b; JSESSIONID=1337; c=d"</code> will be parsed as one cookie, with the name <code>DISPLAY_LANGUAGE</code> and a value of <code>b; JSESSIONID=1337; c=d</code> instead of 3 separate cookies. This has security implications because if, say, <code>JSESSIONID</code> is an <code>HttpOnly</code> cookie, and the <code>DISPLAY_LANGUAGE</code> cookie value is rendered on the page, an attacker can smuggle the <code>JSESSIONID</code> cookie into the <code>DISPLAY_LANGUAGE</code> cookie and thereby exfiltrate it. This is significant when an intermediary is enacting some policy based on cookies, so a smuggled cookie can bypass that policy yet still be seen by the Jetty server or its logging system. This issue has been addressed in versions 9.4.51, 10.0.14, 11.0.14, and 12.0.0.beta0 and users are advised to upgrade. There are no known workarounds for this issue.
CVE-2023-2609	NULL Pointer Dereference in GitHub repository vim/vim prior to 9.0.1531.
CVE-2023-2610	Integer Overflow or Wraparound in GitHub repository vim/vim prior to 9.0.1532.
CVE-2023-26242	<code>afu_mmio_region_get_by_offset</code> in <code>drivers/fpga/dfl-afu-region.c</code> in the Linux kernel through 6.1.12 has an integer overflow.
CVE-2023-2650	Issue summary: Processing some specially crafted ASN.1 object identifiers or data containing them may be very slow. Impact summary: Applications that use <code>OBJ_obj2txt()</code> directly, or use any of the OpenSSL subsystems OCSP, PKCS7/SMIME, CMS, CMP/CRMF or TS with no message size limit may experience notable to very long delays when processing those messages, which may lead to a Denial of Service. An OBJECT IDENTIFIER is composed of a series of numbers - sub-identifiers - most of which have no size limit. <code>OBJ_obj2txt()</code> may be used to translate an ASN.1 OBJECT IDENTIFIER given in DER encoding form (using the OpenSSL type <code>ASN1_OBJECT</code>) to its canonical numeric text form, which are the sub-identifiers of the OBJECT IDENTIFIER in decimal form, separated by periods. When one of the sub-identifiers in the OBJECT IDENTIFIER is very large (these are sizes that are seen as absurdly large, taking up tens or hundreds of KiBs), the translation to a decimal number in text may take a very long time. The time complexity is $O(n^2)$ with 'n' being the size of the sub-identifiers in bytes (*). With OpenSSL 3.0, support to fetch cryptographic algorithms using names / identifiers in string form was introduced. This includes using OBJECT IDENTIFIERS in canonical numeric text form as identifiers for fetching algorithms. Such OBJECT IDENTIFIERS may be received through the ASN.1 structure <code>AlgorithmIdentifier</code> , which is commonly used in multiple protocols to specify what cryptographic algorithm should be used to sign or verify, encrypt or decrypt, or digest passed data. Applications that call <code>OBJ_obj2txt()</code> directly with untrusted data are affected, with any version of OpenSSL. If the use is for the mere purpose of display, the severity is considered low. In OpenSSL 3.0 and newer, this affects the subsystems OCSP, PKCS7/SMIME, CMS, CMP/CRMF or TS. It also impacts anything that processes X.509 certificates, including simple things like verifying its signature. The impact on TLS is relatively low, because all versions of OpenSSL have a 100KiB limit on the peer's certificate chain. Additionally, this only impacts clients, or servers that have explicitly enabled client authentication. In OpenSSL 1.1.1 and 1.0.2, this only affects displaying diverse objects, such as X.509 certificates. This is assumed to not happen in such a way that it would cause a Denial of Service, so these versions are considered not affected by this issue in such a way that it would be cause for concern, and the severity is therefore considered low.
CVE-2023-26965	<code>loadImage()</code> in <code>tools/tiffcrop.c</code> in <code>LibTIFF</code> through 4.5.0 has a heap-based use after free via a crafted TIFF image.
CVE-2023-26966	<code>libtiff 4.5.0</code> is vulnerable to Buffer Overflow in <code>uv_encode()</code> when <code>libtiff</code> reads a corrupted little-endian TIFF file and specifies the output to be big-endian.

CVE ID	Description
CVE-2023-27534	A path traversal vulnerability exists in curl <8.0.0 SFTP implementation causes the tilde (~) character to be wrongly replaced when used as a prefix in the first path element, in addition to its intended use as the first element to indicate a path relative to the user's home directory. Attackers can exploit this flaw to bypass filtering or execute arbitrary code by crafting a path like ~/2/foo while accessing a server with a specific user.
CVE-2023-29400	Templates containing actions in unquoted HTML attributes (e.g. "attr={{.}}") executed with empty input can result in output with unexpected results when parsed due to HTML normalization rules. This may allow injection of arbitrary attributes into tags.
CVE-2023-29402	The go command may generate unexpected code at build time when using cgo. This may result in unexpected behavior when running a go program which uses cgo. This may occur when running an untrusted module which contains directories with newline characters in their names. Modules which are retrieved using the go command, i.e. via "go get", are not affected (modules retrieved using GOPATH-mode, i.e. GO111MODULE=off, may be affected).
CVE-2023-29403	On Unix platforms, the Go runtime does not behave differently when a binary is run with the setuid/setgid bits. This can be dangerous in certain cases, such as when dumping memory state, or assuming the status of standard i/o file descriptors. If a setuid/setgid binary is executed with standard I/O file descriptors closed, opening any files can result in unexpected content being read or written with elevated privileges. Similarly, if a setuid/setgid program is terminated, either via panic or signal, it may leak the contents of its registers.
CVE-2023-29404	The go command may execute arbitrary code at build time when using cgo. This may occur when running "go get" on a malicious module, or when running any other command which builds untrusted code. This is can be triggered by linker flags, specified via a "#cgo LDFLAGS" directive. The arguments for a number of flags which are non-optional are incorrectly considered optional, allowing disallowed flags to be smuggled through the LDFLAGS sanitization. This affects usage of both the gc and gccgo compilers.
CVE-2023-29405	The go command may execute arbitrary code at build time when using cgo. This may occur when running "go get" on a malicious module, or when running any other command which builds untrusted code. This is can be triggered by linker flags, specified via a "#cgo LDFLAGS" directive. Flags containing embedded spaces are mishandled, allowing disallowed flags to be smuggled through the LDFLAGS sanitization by including them in the argument of another flag. This only affects usage of the gccgo compiler.
CVE-2023-29406	The HTTP/1 client does not fully validate the contents of the Host header. A maliciously crafted Host header can inject additional headers or entire requests. With fix, the HTTP/1 client now refuses to send requests containing an invalid Request.Host or Request.URL.Host value.
CVE-2023-29409	Extremely large RSA keys in certificate chains can cause a client/server to expend significant CPU time verifying signatures. With fix, the size of RSA keys transmitted during handshakes is restricted to <= 8192 bits. Based on a survey of publicly trusted RSA keys, there are currently only three certificates in circulation with keys larger than this, and all three appear to be test certificates that are not actively deployed. It is possible there are larger keys in use in private PKIs, but we target the web PKI, so causing breakage here in the interests of increasing the default safety of users of crypto/tls seems reasonable.
CVE-2023-29483	eventlet before 0.35.2, as used in dnspython before 2.6.0, allows remote attackers to interfere with DNS name resolution by quickly sending an invalid packet from the expected IP address and source port, aka a "TuDoor" attack. In other words, dnspython does not have the preferred behavior in which the DNS name resolution algorithm would proceed, within the full time window, in order to wait for a valid packet. NOTE: dnspython 2.6.0 is unusable for a different reason that was addressed in 2.6.1.
CVE-2023-29499	A flaw was found in GLib. GVariant deserialization fails to validate that the input conforms to the expected format, leading to denial of service.
CVE-2023-2976	Use of Java's default temporary directory for file creation in `FileBackedOutputStream` in Google Guava versions 1.0 to 31.1 on Unix systems and Android Ice Cream Sandwich allows other users and apps on the machine with access to the default Java temporary directory to be able to access the files created by the class. Even though the security vulnerability is fixed in version 32.0.0, we recommend using version 32.0.1 as version 32.0.0 breaks some functionality under Windows.
CVE-2023-3006	A known cache speculation vulnerability, known as Branch History Injection (BHI) or Spectre-BHB, becomes actual again for the new hw AmpereOne. Spectre-BHB is similar to Spectre v2, except that malicious code uses the shared branch history (stored in the CPU Branch History Buffer, or BHB) to influence mispredicted branches within the victim's hardware context. Once that occurs, speculation caused by the mispredicted branches can cause cache allocation. This issue leads to obtaining information that should not be accessible.

CVE ID	Description
CVE-2023-30086	Buffer Overflow vulnerability found in Libtiff V.4.0.7 allows a local attacker to cause a denial of service via the tiffcp function in tiffcp.c.
CVE-2023-3022	A flaw was found in the IPv6 module of the Linux kernel. The arg.result was not used consistently in fib6_rule_lookup, sometimes holding rt6_info and other times fib6_info. This was not accounted for in other parts of the code where rt6_info was expected unconditionally, potentially leading to a kernel panic in fib6_rule_suppress.
CVE-2023-30774	A vulnerability was found in the libtiff library. This flaw causes a heap buffer overflow issue via the TIFFTAG_INKNAMES and TIFFTAG_NUMBEROFINKS values.
CVE-2023-30775	A vulnerability was found in the libtiff library. This security flaw causes a heap buffer overflow in extractContigSamples32bits, tiffcrop.c.
CVE-2023-31082	An issue was discovered in drivers/tty/n_gsm.c in the Linux kernel 6.2. There is a sleeping function called from an invalid context in gsmld_write, which will block the kernel. Note: This has been disputed by 3rd parties as not a valid vulnerability.
CVE-2023-31085	An issue was discovered in drivers/mtd/ubi/cdev.c in the Linux kernel 6.2. There is a divide-by-zero error in do_div(sz,mtd->erasesize), used indirectly by ctrl_cdev_ioctl, when mtd->erasesize is 0.
CVE-2023-31437	An issue was discovered in systemd 253. An attacker can modify a sealed log file such that, in some views, not all existing and sealed log messages are displayed. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."
CVE-2023-31438	An issue was discovered in systemd 253. An attacker can truncate a sealed log file and then resume log sealing such that checking the integrity shows no error, despite modifications. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."
CVE-2023-31439	An issue was discovered in systemd 253. An attacker can modify the contents of past events in a sealed log file and then adjust the file such that checking the integrity shows no error, despite modifications. NOTE: the vendor reportedly sent "a reply denying that any of the finding was a security vulnerability."
CVE-2023-31486	HTTP::Tiny before 0.083, a Perl core module since 5.13.9 and available standalone on CPAN, has an insecure default TLS configuration where users must opt in to verify certificates.
CVE-2023-3159	A use after free issue was discovered in driver/firewire in outbound_phy_packet_callback in the Linux Kernel. In this flaw a local attacker with special privilege may cause a use after free problem when queue_event() fails.
CVE-2023-3164	A heap-buffer-overflow vulnerability was found in LibTIFF, in extractImageSection() at tools/tiffcrop.c:7916 and tools/tiffcrop.c:7801. This flaw allows attackers to cause a denial of service via a crafted tiff file.
CVE-2023-32611	A flaw was found in GLib. GVariant deserialization is vulnerable to a slowdown issue where a crafted GVariant can cause excessive processing, leading to denial of service.
CVE-2023-32636	A flaw was found in glib, where the gvariant deserialization code is vulnerable to a denial of service introduced by additional input validation added to resolve CVE-2023-29499. The offset table validation may be very slow. This bug does not affect any released version of glib but does affect glib distributors who followed the guidance of glib developers to backport the initial fix for CVE-2023-29499.
CVE-2023-32665	A flaw was found in GLib. GVariant deserialization is vulnerable to an exponential blowup issue where a crafted GVariant can cause excessive processing, leading to denial of service.
CVE-2023-32681	Requests is a HTTP library. Since Requests 2.3.0, Requests has been leaking Proxy-Authorization headers to destination servers when redirected to an HTTPS endpoint. This is a product of how we use `rebuild_proxies` to reattach the `Proxy-Authorization` header to requests. For HTTP connections sent through the tunnel, the proxy will identify the header in the request itself and remove it prior to forwarding to the destination server. However when sent over HTTPS, the `Proxy-Authorization` header must be sent in the CONNECT request as the proxy has no visibility into the tunneled request. This results in Requests forwarding proxy credentials to the destination server unintentionally, allowing a malicious actor to potentially exfiltrate sensitive information. This issue has been patched in version 2.31.0.
CVE-2023-33053	Memory corruption in Kernel while parsing metadata.
CVE-2023-3316	A NULL pointer dereference in TIFFClose() is caused by a failure to open an output file (non-existent path or a path that requires permissions like /dev/null) while specifying zones.

CVE ID	Description
CVE-2023-33201	Bouncy Castle For Java before 1.74 is affected by an LDAP injection vulnerability. The vulnerability only affects applications that use an LDAP CertStore from Bouncy Castle to validate X.509 certificates. During the certificate validation process, Bouncy Castle inserts the certificate's Subject Name into an LDAP search filter without any escaping, which leads to an LDAP injection vulnerability.
CVE-2023-33202	Bouncy Castle for Java before 1.73 contains a potential Denial of Service (DoS) issue within the Bouncy Castle org.bouncycastle.openssl.PEMParser class. This class parses OpenSSL PEM encoded streams containing X.509 certificates, PKCS8 encoded keys, and PKCS7 objects. Parsing a file that has crafted ASN.1 data through the PEMParser causes an OutOfMemoryError, which can enable a denial of service attack. (For users of the FIPS Java API: BC-FJA 1.0.2.3 and earlier are affected; BC-FJA 1.0.2.4 is fixed.)
CVE-2023-3358	A null pointer dereference was found in the Linux kernel's Integrated Sensor Hub (ISH) driver. This issue could allow a local user to crash the system.
CVE-2023-3446	Issue summary: Checking excessively long DH keys or parameters may be very slow. Impact summary: Applications that use the functions DH_check(), DH_check_ex() or EVP_PKEY_param_check() to check a DH key or DH parameters may experience long delays. Where the key or parameters that are being checked have been obtained from an untrusted source this may lead to a Denial of Service. The function DH_check() performs various checks on DH parameters. One of those checks confirms that the modulus ('p' parameter) is not too large. Trying to use a very large modulus is slow and OpenSSL will not normally use a modulus which is over 10,000 bits in length. However the DH_check() function checks numerous aspects of the key or parameters that have been supplied. Some of those checks use the supplied modulus value even if it has already been found to be too large. An application that calls DH_check() and supplies a key or parameters obtained from an untrusted source could be vulnerable to a Denial of Service attack. The function DH_check() is itself called by a number of other OpenSSL functions. An application calling any of those other functions may similarly be affected. The other functions affected by this are DH_check_ex() and EVP_PKEY_param_check(). Also vulnerable are the OpenSSL dhparam and pkeyparam command line applications when using the '-check' option. The OpenSSL SSL/TLS implementation is not affected by this issue. The OpenSSL 3.0 and 3.1 FIPS providers are not affected by this issue.
CVE-2023-34462	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. The `SniHandler` can allocate up to 16MB of heap for each channel during the TLS handshake. When the handler or the channel does not have an idle timeout, it can be used to make a TCP server using the `SniHandler` to allocate 16MB of heap. The `SniHandler` class is a handler that waits for the TLS handshake to configure a `SslHandler` according to the indicated server name by the `ClientHello` record. For this matter it allocates a `ByteBuf` using the value defined in the `ClientHello` record. Normally the value of the packet should be smaller than the handshake packet but there are not checks done here and the way the code is written, it is possible to craft a packet that makes the `SslClientHelloHandler`. This vulnerability has been fixed in version 4.1.94.Final.
CVE-2023-3576	A memory leak flaw was found in Libtiff's tiffcrop utility. This issue occurs when tiffcrop operates on a TIFF image file, allowing an attacker to pass a crafted TIFF image file to tiffcrop utility, which causes this memory leak issue, resulting an application crash, eventually leading to a denial of service.
CVE-2023-35790	An issue was discovered in dec_patch_dictionary.cc in libjxl before 0.8.2. An integer underflow in patch decoding can lead to a denial of service, such as an infinite loop.
CVE-2023-36054	lib/kadm5/kadm_rpc_xdr.c in MIT Kerberos 5 (aka krb5) before 1.20.2 and 1.21.x before 1.21.1 frees an uninitialized pointer. A remote authenticated user can trigger a kadmind crash. This occurs because _xdr_kadm5_principal_ent_rec does not validate the relationship between n_key_data and the key_data array count.
CVE-2023-3618	A flaw was found in libtiff. A specially crafted tiff file can lead to a segmentation fault due to a buffer overflow in the Fax3Encode function in libtiff/tif_fax3.c, resulting in a denial of service.
CVE-2023-3635	GzipSource does not handle an exception that might be raised when parsing a malformed gzip buffer. This may lead to denial of service of the Okio client when handling a crafted GZIP archive, by using the GzipSource class.
CVE-2023-3640	A possible unauthorized memory access flaw was found in the Linux kernel's cpu_entry_area mapping of X86 CPU data to memory, where a user may guess the location of exception stacks or other important data. Based on the previous CVE-2023-0597, the 'Randomize per-cpu entry area' feature was implemented in /arch/x86/mm/cpu_entry_area.c, which works through the init_cea_offsets() function when KASLR is enabled. However, despite this feature, there is still a risk of per-cpu entry area leaks. This issue could allow a local user to gain access to some important data with memory in an expected location and potentially escalate their privileges on the system.

CVE ID	Description
CVE-2023-36478	Eclipse Jetty provides a web server and servlet container. In versions 11.0.0 through 11.0.15, 10.0.0 through 10.0.15, and 9.0.0 through 9.4.52, an integer overflow in <code>MetaDataBuilder.checkSize</code> allows for HTTP/2 HPACK header values to exceed their size limit. <code>MetaDataBuilder.java</code> determines if a header name or value exceeds the size limit, and throws an exception if the limit is exceeded. However, when length is very large and huffman is true, the multiplication by 4 in line 295 will overflow, and length will become negative. <code>(size+length)</code> will now be negative, and the check on line 296 will not be triggered. Furthermore, <code>MetaDataBuilder.checkSize</code> allows for user-entered HPACK header value sizes to be negative, potentially leading to a very large buffer allocation later on when the user-entered size is multiplied by 2. This means that if a user provides a negative length value (or, more precisely, a length value which, when multiplied by the 4/3 fudge factor, is negative), and this length value is a very large positive number when multiplied by 2, then the user can cause a very large buffer to be allocated on the server. Users of HTTP/2 can be impacted by a remote denial of service attack. The issue has been fixed in versions 11.0.16, 10.0.16, and 9.4.53. There are no known workarounds.
CVE-2023-36479	Eclipse Jetty Canonical Repository is the canonical repository for the Jetty project. Users of the <code>CgiServlet</code> with a very specific command structure may have the wrong command executed. If a user sends a request to a <code>org.eclipse.jetty.servlets.CGI Servlet</code> for a binary with a space in its name, the servlet will escape the command by wrapping it in quotation marks. This wrapped command, plus an optional command prefix, will then be executed through a call to <code>Runtime.exec</code> . If the original binary name provided by the user contains a quotation mark followed by a space, the resulting command line will contain multiple tokens instead of one. This issue was patched in version 9.4.52, 10.0.16, 11.0.16 and 12.0.0-beta2.
CVE-2023-37454	An issue was discovered in the Linux kernel through 6.4.2. A crafted UDF filesystem image causes a use-after-free write operation in the <code>udf_put_super</code> and <code>udf_close_lvid</code> functions in <code>fs/udf/super.c</code> . NOTE: the <code>suse.com</code> reference has a different perspective about this.
CVE-2023-37769	<code>stress-test master commit e4c878</code> was discovered to contain a FPE vulnerability via the component <code>combine_inner</code> at <code>/pixman-combine-float.c</code> .
CVE-2023-3817	Issue summary: Checking excessively long DH keys or parameters may be very slow. Impact summary: Applications that use the functions <code>DH_check()</code> , <code>DH_check_ex()</code> or <code>EVP_PKEY_param_check()</code> to check a DH key or DH parameters may experience long delays. Where the key or parameters that are being checked have been obtained from an untrusted source this may lead to a Denial of Service. The function <code>DH_check()</code> performs various checks on DH parameters. After fixing CVE-2023-3446 it was discovered that a large <code>q</code> parameter value can also trigger an overly long computation during some of these checks. A correct <code>q</code> value, if present, cannot be larger than the modulus <code>p</code> parameter, thus it is unnecessary to perform these checks if <code>q</code> is larger than <code>p</code> . An application that calls <code>DH_check()</code> and supplies a key or parameters obtained from an untrusted source could be vulnerable to a Denial of Service attack. The function <code>DH_check()</code> is itself called by a number of other OpenSSL functions. An application calling any of those other functions may similarly be affected. The other functions affected by this are <code>DH_check_ex()</code> and <code>EVP_PKEY_param_check()</code> . Also vulnerable are the OpenSSL <code>dhparam</code> and <code>pkeyparam</code> command line applications when using the <code>"-check"</code> option. The OpenSSL SSL/TLS implementation is not affected by this issue. The OpenSSL 3.0 and 3.1 FIPS providers are not affected by this issue.
CVE-2023-3894	Those using <code>jackson-dataformats-text</code> to parse TOML data may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow. This effect may support a denial of service attack.
CVE-2023-39318	The <code>html/template</code> package does not properly handle HTML-like <code>""</code> comment tokens, nor hashbang <code>"#!"</code> comment tokens, in <code><script></code> contexts. This may cause the template parser to improperly interpret the contents of <code><script></code> contexts, causing actions to be improperly escaped. This may be leveraged to perform an XSS attack.
CVE-2023-39319	The <code>html/template</code> package does not apply the proper rules for handling occurrences of <code>"<script"</code> , <code>"<!--"</code> , and <code>"</script"</code> within JS literals in <code><script></code> contexts. This may cause the template parser to improperly consider script contexts to be terminated early, causing actions to be improperly escaped. This could be leveraged to perform an XSS attack.
CVE-2023-39323	Line directives (<code>"//line"</code>) can be used to bypass the restrictions on <code>"//go:cgo_"</code> directives, allowing blocked linker and compiler flags to be passed during compilation. This can result in unexpected execution of arbitrary code when running <code>"go build"</code> . The line directive requires the absolute path of the file in which the directive lives, which makes exploiting this issue significantly more complex.

CVE ID	Description
CVE-2023-39325	A malicious HTTP/2 client which rapidly creates requests and immediately resets them can cause excessive server resource consumption. While the total number of requests is bounded by the <code>http2.Server.MaxConcurrentStreams</code> setting, resetting an in-progress request allows the attacker to create a new request while the existing one is still executing. With the fix applied, HTTP/2 servers now bound the number of simultaneously executing handler goroutines to the stream concurrency limit (<code>MaxConcurrentStreams</code>). New requests arriving when at the limit (which can only happen after the client has reset an existing, in-flight request) will be queued until a handler exits. If the request queue grows too large, the server will terminate the connection. This issue is also fixed in <code>golang.org/x/net/http2</code> for users manually configuring HTTP/2. The default stream concurrency limit is 250 streams (requests) per HTTP/2 connection. This value may be adjusted using the <code>golang.org/x/net/http2</code> package; see the <code>Server.MaxConcurrentStreams</code> setting and the <code>ConfigureServer</code> function.
CVE-2023-39326	A malicious HTTP sender can use chunk extensions to cause a receiver reading from a request or response body to read many more bytes from the network than are in the body. A malicious HTTP client can further exploit this to cause a server to automatically read a large amount of data (up to about 1GiB) when a handler fails to read the entire body of a request. Chunk extensions are a little-used HTTP feature which permit including additional metadata in a request or response body sent using the chunked encoding. The <code>net/http</code> chunked encoding reader discards this metadata. A sender can exploit this by inserting a large metadata segment with each byte transferred. The chunk reader now produces an error if the ratio of real body to encoded bytes grows too small.
CVE-2023-39328	A vulnerability was found in OpenJPEG similar to CVE-2019-6988. This flaw allows an attacker to bypass existing protections and cause an application crash through a maliciously crafted file.
CVE-2023-39329	A flaw was found in OpenJPEG. A resource exhaustion can occur in the <code>opj_t1_decode_cblks</code> function in <code>tcd.c</code> through a crafted image file, causing a denial of service.
CVE-2023-39410	When deserializing untrusted or corrupted data, it is possible for a reader to consume memory beyond the allowed constraints and thus lead to out of memory on the system. This issue affects Java applications using Apache Avro Java SDK up to and including 1.11.2. Users should update to <code>apache-avro</code> version 1.11.3 which addresses this issue.
CVE-2023-3978	Text nodes not in the HTML namespace are incorrectly literally rendered, causing text which should be escaped to not be. This could lead to an XSS attack.
CVE-2023-39804	In GNU tar before 1.35, mishandled extension attributes in a PAX archive can lead to an application crash in <code>xheader.c</code> .
CVE-2023-4010	A flaw was found in the USB Host Controller Driver framework in the Linux kernel. The <code>usb_giveback_urb</code> function has a logic loophole in its implementation. Due to the inappropriate judgment condition of the <code>goto</code> statement, the function cannot return under the input of a specific malformed descriptor file, so it falls into an endless loop, resulting in a denial of service.
CVE-2023-40167	Jetty is a Java based web server and servlet engine. Prior to versions 9.4.52, 10.0.16, 11.0.16, and 12.0.1, Jetty accepts the '+' character preceding the content-length value in a HTTP/1 header field. This is more permissive than allowed by the RFC and other servers routinely reject such requests with 400 responses. There is no known exploit scenario, but it is conceivable that request smuggling could result if jetty is used in combination with a server that does not close the connection after sending such a 400 response. Versions 9.4.52, 10.0.16, 11.0.16, and 12.0.1 contain a patch for this issue. There is no workaround as there is no known exploit scenario.
CVE-2023-4039	**DISPUTED** A failure in the <code>-fstack-protector</code> feature in GCC-based toolchains that target AArch64 allows an attacker to exploit an existing buffer overflow in dynamically-sized local variables in your application without this being detected. This stack-protector failure only applies to C99-style dynamically-sized local variables or those created using <code>alloca()</code> . The stack-protector operates as intended for statically-sized local variables. The default behavior when the stack-protector detects an overflow is to terminate your application, resulting in controlled loss of availability. An attacker who can exploit a buffer overflow without triggering the stack-protector might be able to change program flow control to cause an uncontrolled loss of availability or to go further and affect confidentiality or integrity. NOTE: The GCC project argues that this is a missed hardening bug and not a vulnerability by itself.
CVE-2023-40745	LibTIFF is vulnerable to an integer overflow. This flaw allows remote attackers to cause a denial of service (application crash) or possibly execute an arbitrary code via a crafted tiff image, which triggers a heap-based buffer overflow.
CVE-2023-41175	A vulnerability was found in <code>libtiff</code> due to multiple potential integer overflows in <code>raw2tiff.c</code> . This flaw allows remote attackers to cause a denial of service or possibly execute an arbitrary code via a crafted tiff image, which triggers a heap-based buffer overflow.
CVE-2023-4156	A heap out-of-bounds read flaw was found in <code>builtin.c</code> in the <code>gawk</code> package. This issue may lead to a crash and could be used to read sensitive information.

CVE ID	Description
CVE-2023-4194	A flaw was found in the Linux kernel's TUN/TAP functionality. This issue could allow a local user to bypass network filters and gain unauthorized access to some resources. The original patches fixing CVE-2023-1076 are incorrect or incomplete. The problem is that the following upstream commits - a096ccca6e50 ("tun: tun_chr_open(): correctly initialize socket uid"), - 66b2c338adce ("tap: tap_open(): correctly initialize socket uid"), pass "inode->i_uid" to sock_init_data_uid() as the last parameter and that turns out to not be accurate.
CVE-2023-42282	The ip package before 1.1.9 for Node.js might allow SSRF because some IP addresses (such as 0x7f.1) are improperly categorized as globally routable via isPublic.
CVE-2023-42503	Improper Input Validation, Uncontrolled Resource Consumption vulnerability in Apache Commons Compress in TAR parsing. This issue affects Apache Commons Compress: from 1.22 before 1.24.0. Users are recommended to upgrade to version 1.24.0, which fixes the issue. A third party can create a malformed TAR file by manipulating file modification times headers, which when parsed with Apache Commons Compress, will cause a denial of service issue via CPU consumption. In version 1.22 of Apache Commons Compress, support was added for file modification times with higher precision (issue # COMPRESS-612 [1]). The format for the PAX extended headers carrying this data consists of two numbers separated by a period [2], indicating seconds and subsecond precision (for example "1647221103.5998539"). The impacted fields are "atime", "ctime", "mtime" and "LIBARCHIVE.creationtime". No input validation is performed prior to the parsing of header values. Parsing of these numbers uses the BigDecimal [3] class from the JDK which has a publicly known algorithmic complexity issue when doing operations on large numbers, causing denial of service (see issue # JDK-6560193 [4]). A third party can manipulate file time headers in a TAR file by placing a number with a very long fraction (300,000 digits) or a number with exponent notation (such as "9e9999999") within a file modification time header, and the parsing of files with these headers will take hours instead of seconds, leading to a denial of service via exhaustion of CPU resources. This issue is similar to CVE-2012-2098 [5]. [1]: https://issues.apache.org/jira/browse/COMPRESS-612 [2]: https://pubs.opengroup.org/onlinepubs/9699919799/utilities/pax.html#tag_20_92_13_05 [3]: https://docs.oracle.com/javase/8/docs/api/java/math/BigDecimal.html [4]: https://bugs.openjdk.org/browse/JDK-6560193 [5]: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2098 Only applications using CompressorStreamFactory class (with auto-detection of file types), TarArchiveInputStream and TarFile classes to parse TAR files are impacted. Since this code was introduced in v1.22, only that version and later versions are impacted.
CVE-2023-44487	The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.
CVE-2023-44981	Authorization Bypass Through User-Controlled Key vulnerability in Apache ZooKeeper. If SASL Quorum Peer authentication is enabled in ZooKeeper (quorum.auth.enableSasl=true), the authorization is done by verifying that the instance part in SASL authentication ID is listed in zoo.cfg server list. The instance part in SASL auth ID is optional and if it's missing, like 'eve@EXAMPLE.COM', the authorization check will be skipped. As a result an arbitrary endpoint could join the cluster and begin propagating counterfeit changes to the leader, essentially giving it complete read-write access to the data tree. Quorum Peer authentication is not enabled by default. Users are recommended to upgrade to version 3.9.1, 3.8.3, 3.7.2, which fixes the issue. Alternately ensure the ensemble election/quorum communication is protected by a firewall as this will mitigate the issue. See the documentation for more details on correct cluster administration.
CVE-2023-4504	Due to failure in validating the length provided by an attacker-crafted PPD PostScript document, CUPS and libppd are susceptible to a heap-based buffer overflow and possibly code execution. This issue has been fixed in CUPS version 2.4.7, released in September of 2023.
CVE-2023-45142	OpenTelemetry-Go Contrib is a collection of third-party packages for OpenTelemetry-Go. A handler wrapper out of the box adds labels 'http.user_agent' and 'http.method' that have unbound cardinality. It leads to the server's potential memory exhaustion when many malicious requests are sent to it. HTTP header User-Agent or HTTP method for requests can be easily set by an attacker to be random and long. The library internally uses 'httpconv.ServerRequest' that records every value for HTTP 'method' and 'User-Agent'. In order to be affected, a program has to use the 'otelhttp.NewHandler' wrapper and not filter any unknown HTTP methods or User agents on the level of CDN, LB, previous middleware, etc. Version 0.44.0 fixed this issue when the values collected for attribute 'http.request.method' were changed to be restricted to a set of well-known values and other high cardinality attributes were removed. As a workaround to stop being affected, 'otelhttp.WithFilter()' can be used, but it requires manual careful configuration to not log certain requests entirely. For convenience and safe usage of this library, it should by default mark with the label 'unknown' non-standard HTTP methods and User agents to show that such requests were made but do not increase cardinality. In case someone wants to stay with the current behavior, library API should allow to enable it.

CVE ID	Description
CVE-2023-45283	The filepath package does not recognize paths with a \\?\\ prefix as special. On Windows, a path beginning with \\?\\ is a Root Local Device path equivalent to a path beginning with \\?. Paths with a \\?\\ prefix may be used to access arbitrary locations on the system. For example, the path \\?\\c:\\x is equivalent to the more common path c:\\x. Before fix, Clean could convert a rooted path such as \\a\\.\\?\\b into the root local device path \\?\\b. Clean will now convert this to .\\?\\b. Similarly, Join(\\, ??, b) could convert a seemingly innocent sequence of path elements into the root local device path \\?\\b. Join will now convert this to \\\\.\\?\\b. In addition, with fix, IsAbs now correctly reports paths beginning with \\?\\ as absolute, and VolumeName correctly reports the \\?\\ prefix as a volume name. UPDATE: Go 1.20.11 and Go 1.21.4 inadvertently changed the definition of the volume name in Windows paths starting with \\?, resulting in filepath.Clean(\\?\\c:) returning \\?\\c: rather than \\?\\c: (among other effects). The previous behavior has been restored.
CVE-2023-45284	On Windows, The IsLocal function does not correctly detect reserved device names in some cases. Reserved names followed by spaces, such as "COM1 ", and reserved names "COM" and "LPT" followed by superscript 1, 2, or 3, are incorrectly reported as local. With fix, IsLocal now correctly reports these names as non-local.
CVE-2023-45285	Using go get to fetch a module with the ".git" suffix may unexpectedly fallback to the insecure "git://" protocol if the module is unavailable via the secure "https://" and "git+ssh://" protocols, even if GOINSECURE is not set for said module. This only affects users who are not using the module proxy and are fetching modules directly (i.e. GOPROXY=off).
CVE-2023-45287	Before Go 1.20, the RSA based TLS key exchanges used the math/big library, which is not constant time. RSA blinding was applied to prevent timing attacks, but analysis shows this may not have been fully effective. In particular it appears as if the removal of PKCS#1 padding may leak timing information, which in turn could be used to recover session key bits. In Go 1.20, the crypto/tls library switched to a fully constant time RSA implementation, which we do not believe exhibits any timing side channels.
CVE-2023-45288	An attacker may cause an HTTP/2 endpoint to read arbitrary amounts of header data by sending an excessive number of CONTINUATION frames. Maintaining HPACK state requires parsing and processing all HEADERS and CONTINUATION frames on a connection. When a request's headers exceed MaxHeaderBytes, no memory is allocated to store the excess headers, but they are still parsed. This permits an attacker to cause an HTTP/2 endpoint to read arbitrary amounts of header data, all associated with a request which is going to be rejected. These headers can include Huffman-encoded data which is significantly more expensive for the receiver to decode than for an attacker to send. The fix sets a limit on the amount of excess header frames we will process before closing a connection.
CVE-2023-45289	When following an HTTP redirect to a domain which is not a subdomain match or exact match of the initial domain, an http.Client does not forward sensitive headers such as "Authorization" or "Cookie". For example, a redirect from foo.com to www.foo.com will forward the Authorization header, but a redirect to bar.com will not. A maliciously crafted HTTP redirect could cause sensitive headers to be unexpectedly forwarded.
CVE-2023-45290	When parsing a multipart form (either explicitly with Request.ParseMultipartForm or implicitly with Request.FormValue, Request.PostFormValue, or Request.FormFile), limits on the total size of the parsed form were not applied to the memory consumed while reading a single form line. This permits a maliciously crafted input containing very long lines to cause allocation of arbitrarily large amounts of memory, potentially leading to memory exhaustion. With fix, the ParseMultipartForm function now correctly limits the maximum size of form lines.
CVE-2023-45322	libxml2 through 2.11.5 has a use-after-free that can only occur after a certain memory allocation fails. This occurs in xmlUnlinkNode in tree.c. NOTE: the vendor's position is "I don't think these issues are critical enough to warrant a CVE ID ... because an attacker typically can't control when memory allocations fail."

CVE ID	Description
CVE-2023-45803	urllib3 is a user-friendly HTTP client library for Python. urllib3 previously wouldn't remove the HTTP request body when an HTTP redirect response using status 301, 302, or 303 after the request had its method changed from one that could accept a request body (like `POST`) to `GET` as is required by HTTP RFCs. Although this behavior is not specified in the section for redirects, it can be inferred by piecing together information from different sections and we have observed the behavior in other major HTTP client implementations like curl and web browsers. Because the vulnerability requires a previously trusted service to become compromised in order to have an impact on confidentiality we believe the exploitability of this vulnerability is low. Additionally, many users aren't putting sensitive data in HTTP request bodies, if this is the case then this vulnerability isn't exploitable. Both of the following conditions must be true to be affected by this vulnerability: 1. Using urllib3 and submitting sensitive information in the HTTP request body (such as form data or JSON) and 2. The origin service is compromised and starts redirecting using 301, 302, or 303 to a malicious peer or the redirected-to service becomes compromised. This issue has been addressed in versions 1.26.18 and 2.0.7 and users are advised to update to resolve this issue. Users unable to update should disable redirects for services that aren't expecting to respond with redirects with `redirects=False` and disable automatic redirects with `redirects=False` and handle 301, 302, and 303 redirects manually by stripping the HTTP request body.
CVE-2023-45853	MiniZip in zlib through 1.3 has an integer overflow and resultant heap-based buffer overflow in zipOpenNewFileInZip4_64 via a long filename, comment, or extra field. NOTE: MiniZip is not a supported part of the zlib product. NOTE: pyminizip through 0.2.6 is also vulnerable because it bundles an affected zlib version, and exposes the applicable MiniZip code through its compress API.
CVE-2023-46136	Werkzeug is a comprehensive WSGI web application library. If an upload of a file that starts with CR or LF and then is followed by megabytes of data without these characters: all of these bytes are appended chunk by chunk into internal bytearray and lookup for boundary is performed on growing buffer. This allows an attacker to cause a denial of service by sending crafted multipart data to an endpoint that will parse it. The amount of CPU time required can block worker processes from handling legitimate requests. This vulnerability has been patched in version 3.0.1.
CVE-2023-46246	Vim is an improved version of the good old UNIX editor Vi. Heap-use-after-free in memory allocated in the function `ga_grow_inner` in the file `src/alloc.c` at line 748, which is freed in the file `src/ex_docmd.c` in the function `do_cmdline` at line 1010 and then used again in `src/cmdhist.c` at line 759. When using the `:history` command, it's possible that the provided argument overflows the accepted value. Causing an Integer Overflow and potentially later an use-after-free. This vulnerability has been patched in version 9.0.2068.
CVE-2023-46749	Apache Shiro before 1.13.0 or 2.0.0-alpha-4, may be susceptible to a path traversal attack that results in an authentication bypass when used together with path rewriting Mitigation: Update to Apache Shiro 1.13.0+ or 2.0.0-alpha-4+, or ensure `blockSemicolon` is enabled (this is the default).
CVE-2023-46750	URL Redirection to Untrusted Site ('Open Redirect') vulnerability when "form" authentication is used in Apache Shiro. Mitigation: Update to Apache Shiro 1.13.0+ or 2.0.0-alpha-4+.
CVE-2023-47038	A vulnerability was found in perl 5.30.0 through 5.38.0. This issue occurs when a crafted regular expression is compiled by perl, which can allow an attacker controlled byte buffer overflow in a heap allocated buffer.
CVE-2023-47108	OpenTelemetry-Go Contrib is a collection of third-party packages for OpenTelemetry-Go. Starting in version 0.37.0 and prior to version 0.46.0, the grpc Unary Server Interceptor out of the box adds labels `net.peer.sock.addr` and `net.peer.sock.port` that have unbound cardinality. It leads to the server's potential memory exhaustion when many malicious requests are sent. An attacker can easily flood the peer address and port for requests. Version 0.46.0 contains a fix for this issue. As a workaround to stop being affected, a view removing the attributes can be used. The other possibility is to disable grpc metrics instrumentation by passing `otelgrpc.WithMeterProvider` option with `noop.NewMeterProvider`.
CVE-2023-47233	The brcm80211 component in the Linux kernel through 6.5.10 has a brcmf_cfg80211_detach use-after-free in the device unplugging (disconnect the USB by hotplug) code. For physically proximate attackers with local access, this "could be exploited in a real world scenario." This is related to brcmf_cfg80211_escan_timeout_worker in drivers/net/wireless/broadcom/brcm80211/brcmfmac/cfg80211.c.
CVE-2023-47282	Out-of-bounds write in Intel(R) Media SDK all versions and some Intel(R) oneVPL software before version 23.3.5 may allow an authenticated user to potentially enable escalation of privilege via local access.
CVE-2023-4733	Use After Free in GitHub repository vim/vim prior to 9.0.1840.
CVE-2023-4734	Integer Overflow or Wraparound in GitHub repository vim/vim prior to 9.0.1846.
CVE-2023-4735	Out-of-bounds Write in GitHub repository vim/vim prior to 9.0.1847.

CVE ID	Description
CVE-2023-4738	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1848.
CVE-2023-4750	Use After Free in GitHub repository vim/vim prior to 9.0.1857.
CVE-2023-4751	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1331.
CVE-2023-4752	Use After Free in GitHub repository vim/vim prior to 9.0.1858.
CVE-2023-4781	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1873.
CVE-2023-4806	A flaw has been identified in glibc. In an extremely rare situation, the getaddrinfo function may access memory that has been freed, resulting in an application crash. This issue is only exploitable when a NSS module implements only the <code>_nss_*_gethostbyname2_r</code> and <code>_nss_*_getcanonname_r</code> hooks without implementing the <code>_nss_*_gethostbyname3_r</code> hook. The resolved name should return a large number of IPv6 and IPv4, and the call to the getaddrinfo function should have the AF_INET6 address family with AI_CANONNAME, AI_ALL and AI_V4MAPPED as flags.
CVE-2023-4813	A flaw has been identified in glibc. In an uncommon situation, the gai_inet function may use memory that has been freed, resulting in an application crash. This issue is only exploitable when the getaddrinfo function is called and the hosts database in <code>/etc/nsswitch.conf</code> is configured with <code>SUCCESS=continue</code> or <code>SUCCESS=merge</code> .
CVE-2023-48231	Vim is an open source command line text editor. When closing a window, vim may try to access already freed window structure. Exploitation beyond crashing the application has not been shown to be viable. This issue has been addressed in commit <code>`25aabc2b`</code> which has been included in release version 9.0.2106. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48232	Vim is an open source command line text editor. A floating point exception may occur when calculating the line offset for overlong lines and smooth scrolling is enabled and the cpo-settings include the 'n' flag. This may happen when a window border is present and when the wrapped line continues on the next physical line directly in the window border because the 'cpo' setting includes the 'n' flag. Only users with non-default settings are affected and the exception should only result in a crash. This issue has been addressed in commit <code>`cb0b99f0`</code> which has been included in release version 9.0.2107. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48233	Vim is an open source command line text editor. If the count after the <code>:s</code> command is larger than what fits into a (signed) long variable, abort with <code>e_value_too_large</code> . Impact is low, user interaction is required and a crash may not even happen in all situations. This issue has been addressed in commit <code>`ac6378773`</code> which has been included in release version 9.0.2108. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48234	Vim is an open source command line text editor. When getting the count for a normal mode <code>z</code> command, it may overflow for large counts given. Impact is low, user interaction is required and a crash may not even happen in all situations. This issue has been addressed in commit <code>`58f9befca1`</code> which has been included in release version 9.0.2109. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48235	Vim is an open source command line text editor. When parsing relative ex addresses one may unintentionally cause an overflow. Ironically this happens in the existing overflow check, because the line number becomes negative and <code>LONG_MAX - lnum</code> will cause the overflow. Impact is low, user interaction is required and a crash may not even happen in all situations. This issue has been addressed in commit <code>`060623e`</code> which has been included in release version 9.0.2110. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48236	Vim is an open source command line text editor. When using the <code>z=</code> command, the user may overflow the count with values larger than <code>MAX_INT</code> . Impact is low, user interaction is required and a crash may not even happen in all situations. This vulnerability has been addressed in commit <code>`73b2d379`</code> which has been included in release version 9.0.2111. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48237	Vim is an open source command line text editor. In affected versions when shifting lines in operator pending mode and using a very large value, it may be possible to overflow the size of integer. Impact is low, user interaction is required and a crash may not even happen in all situations. This issue has been addressed in commit <code>`6bf131888`</code> which has been included in version 9.0.2112. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2023-48706	Vim is a UNIX editor that, prior to version 9.0.2121, has a heap-use-after-free vulnerability. When executing a <code>`:s`</code> command for the very first time and using a sub-replace-special atom inside the substitution part, it is possible that the recursive <code>`:s`</code> call causes free-ing of memory which may later then be accessed by the initial <code>`:s`</code> command. The user must intentionally execute the payload and the whole process is a bit tricky to do since it seems to work only reliably for the very first <code>:s</code> command. It may also cause a crash of Vim. Version 9.0.2121 contains a fix for this issue.

CVE ID	Description
CVE-2023-48713	Knative Serving builds on Kubernetes to support deploying and serving of applications and functions as serverless containers. An attacker who controls a pod to a degree where they can control the responses from the /metrics endpoint can cause Denial-of-Service of the autoscaler from an unbound memory allocation bug. This is a DoS vulnerability, where a non-privileged Knative user can cause a DoS for the cluster. This issue has been patched in version 0.39.0.
CVE-2023-48727	NULL pointer dereference in some Intel(R) oneVPL software before version 23.3.5 may allow an authenticated user to potentially enable information disclosure via local access.
CVE-2023-48795	The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGO before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the msdssh module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.
CVE-2023-49501	Buffer Overflow vulnerability in Ffmpeg v.n6.1-3-g466799d4f5 allows a local attacker to execute arbitrary code via the config_eq_output function in the libavfilter/asrc_afirsrc.c:495:30 component.
CVE-2023-49502	Buffer Overflow vulnerability in Ffmpeg v.n6.1-3-g466799d4f5 allows a local attacker to execute arbitrary code via the ff_bwdif_filter_intra_c function in the libavfilter/bwdifdsp.c:125:5 component.
CVE-2023-49528	Buffer Overflow vulnerability in FFmpeg version n6.1-3-g466799d4f5, allows a local attacker to execute arbitrary code and cause a denial of service (DoS) via the af_dialogueenhance.c:261:5 in the de_stereo component.
CVE-2023-50007	Buffer Overflow vulnerability in Ffmpeg v.n6.1-3-g466799d4f5 allows a local attacker to execute arbitrary code via theav_samples_set_silence function in thelibavutil/samplefmt.c:260:9 component.
CVE-2023-50008	Buffer Overflow vulnerability in Ffmpeg v.n6.1-3-g466799d4f5 allows a local attacker to execute arbitrary code via the av_malloc function in libavutil/mem.c:105:9 component.
CVE-2023-50495	Ncurses v6.4-20230418 was discovered to contain a segmentation fault via the component _nc_wrap_entry().
CVE-2023-51074	json-path v2.8.0 was discovered to contain a stack overflow via the Criteria.parse() method.
CVE-2023-51257	An invalid memory write issue in Jasper-Software Jasper v.4.1.1 and before allows a local attacker to execute arbitrary code.
CVE-2023-51767	OpenSSH through 10.0, when common types of DRAM are used, might allow row hammer attacks (for authentication bypass) because the integer value of authenticated in mm_answer_authpassword does not resist flips of a single bit. NOTE: this is applicable to a certain threat model of attacker-victim co-location in which the attacker has user privileges. NOTE: this is disputed by the Supplier, who states "we do not consider it to be the application's responsibility to defend against platform architectural weaknesses."
CVE-2023-52323	PyCryptodome and pycryptodomex before 3.19.1 allow side-channel leakage for OAEP decryption, exploitable for a Manger attack.
CVE-2023-52355	An out-of-memory flaw was found in libtiff that could be triggered by passing a crafted tiff file to the TIFFRasterScanlineSize64() API. This flaw allows a remote attacker to cause a denial of service via a crafted input with a size smaller than 379 KB.
CVE-2023-52428	In Connect2id Nimbus JOSE+JWT before 9.37.2, an attacker can cause a denial of service (resource consumption) via a large JWE p2c header value (aka iteration count) for the PasswordBasedDecrypter (PBKDF2) component.

CVE ID	Description
CVE-2023-52435	In the Linux kernel, the following vulnerability has been resolved: net: prevent mss overflow in skb_segment() Once again syzbot is able to crash the kernel in skb_segment() [1] GSO_BY_FRAGS is a forbidden value, but unfortunately the following computation in skb_segment() can reach it quite easily : mss = mss * partial_segs; 65535 = 3 * 5 * 17 * 257, so many initial values of mss can lead to a bad final result. Make sure to limit segmentation so that the new mss value is smaller than GSO_BY_FRAGS. [1] general protection fault, probably for non-canonical address 0xdffffc000000000e: 0000 [#1] PREEMPT SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000070-0x0000000000000077] CPU: 1 PID: 5079 Comm: syz-executor993 Not tainted 6.7.0-rc4-syzkaller-00141-g1ae4cd3cbdd0 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 11/10/2023 RIP: 0010:skb_segment+0x181d/0x3f30 net/core/skbuff.c:4551 Code: 83 e3 02 e9 fb ed ff ff e8 90 68 1c f9 48 8b 84 24 f8 00 00 00 48 8d 78 70 48 b8 00 00 00 00 00 00 fc ff df 48 89 fa 48 c1 ea 03 <0f> b6 04 02 84 c0 74 08 3c 03 0f 8e 8a 21 00 00 48 8b 84 24 f8 00 RSP: 0018:ffffc900043473d0 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: 0000000000010046 RCX: ffffffff886b1597 RDX: 000000000000000e RSI: ffffffff886b2520 RDI: 0000000000000070 RBP: ffff90004347578 R08: 0000000000000005 R09: 000000000000ffff R10: 000000000000ffff R11: 0000000000000002 R12: ffff888063202ac0 R13: 0000000000010000 R14: 0000000000000046 R15: 0000000000000000 FS: 0000555556e7e380(0000) GS:ffff880b99000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020010000 CR3: 0000000027ee2000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> udp6_ufo_fragment+0xa0e/0xd00 net/ipv6/udp_offload.c:109 ipv6_gso_segment+0x534/0x17e0 net/ipv6/ipv6_offload.c:120 skb_mac_gso_segment+0x290/0x610 net/core/gso.c:53 __skb_gso_segment+0x339/0x710 net/core/gso.c:124 skb_gso_segment include/net/gso.h:83 [inline] validate_xmit_skb+0x36c/0xeb0 net/core/dev.c:3626 __dev_queue_xmit+0x6f3/0x3d60 net/core/dev.c:4338 dev_queue_xmit include/linux/netdevice.h:3134 [inline] packet_xmit+0x257/0x380 net/packet/af_packet.c:276 packet_snd net/packet/af_packet.c:3087 [inline] packet_sendmsg+0x24c6/0x5220 net/packet/af_packet.c:3119 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0xd5/0x180 net/socket.c:745 __sys_sendto+0x255/0x340 net/socket.c:2190 __do_sys_sendto net/socket.c:2202 [inline] __se_sys_sendto net/socket.c:2198 [inline] __x64_sys_sendto+0xe0/0x1b0 net/socket.c:2198 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x40/0x110 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b RIP: 0033:0xf8692032aa9 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 d1 19 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fff8d685418 EFLAGS: 00000246 ORIG_RAX: 000000000000002c RAX: ffffffff886b1597 RBX: 0000000000000003 RCX: 00007f8692032aa9 RDX: 0000000000010048 RSI: 00000000020000c0 RDI: 0000000000000003 RBP: 000000000000f4240 R08: 00000000020000540 R09: 0000000000000014 R10: 0000000000000000 R11: 0000000000000246 R12: 00007fff8d685480 R13: 0000000000000001 R14: 00007fff8d685480 R15: 0000000000000003 <TASK> Modules linked in: ---[end trace 0000000000000000]--- RIP: 0010:skb_segment+0x181d/0x3f30 net/core/skbuff.c:4551 Code: 83 e3 02 e9 fb ed ff ff e8 90 68 1c f9 48 8b 84 24 f8 00 00 00 48 8d 78 70 48 b8 00 00 00 00 fc ff df 48 89 fa 48 c1 ea 03 <0f> b6 04 02 84 c0 74 08 3c 03 0f 8e 8a 21 00 00 48 8b 84 24 f8 00 RSP: 0018:ffffc900043473d0 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: 0000000000010046 RCX: ffffffff886b1597 RDX: 000000000000000e RSI: ffffffff886b2520 RDI: 0000000000000070 RBP: ffff90004347578 R0 ---truncated---
CVE-2023-52447	In the Linux kernel, the following vulnerability has been resolved: bpf: Defer the free of inner map when necessary When updating or deleting an inner map in map array or map htab, the map may still be accessed by non-sleepable program or sleepable program. However bpf_map_fd_put_ptr() decreases the ref-counter of the inner map directly through bpf_map_put(), if the ref-counter is the last one (which is true for most cases), the inner map will be freed by ops->map_free() in a kworker. But for now, most .map_free() callbacks don't use synchronize_rcu() or its variants to wait for the elapse of a RCU grace period, so after the invocation of ops->map_free completes, the bpf program which is accessing the inner map may incur use-after-free problem. Fix the free of inner map by invoking bpf_map_free_deferred() after both one RCU grace period and one tasks trace RCU grace period if the inner map has been removed from the outer map before. The deferment is accomplished by using call_rcu() or call_rcu_tasks_trace() when releasing the last ref-counter of bpf map. The newly-added rcu_head field in bpf_map shares the same storage space with work field to reduce the size of bpf_map.
CVE-2023-52450	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel/uncore: Fix NULL pointer dereference issue in upi_fill_topology() Get logical socket id instead of physical id in discover_upi_topology() to avoid out-of-bound access on 'upi = &type->topology[nid][idx];' line that leads to NULL pointer dereference in upi_fill_topology()

CVE ID	Description
CVE-2023-52452	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix accesses to uninit stack slots Privileged programs are supposed to be able to read uninitialized stack memory (ever since 6715df8d5) but, before this patch, these accesses were permitted inconsistently. In particular, accesses were permitted above state->allocated_stack, but not below it. In other words, if the stack was already "large enough", the access was permitted, but otherwise the access was rejected instead of being allowed to "grow the stack". This undesired rejection was happening in two places: - in check_stack_slot_within_bounds() - in check_stack_range_initialized() This patch arranges for these accesses to be permitted. A bunch of tests that were relying on the old rejection had to change; all of them were changed to add also run_unprivileged, in which case the old behavior persists. One tests couldn't be updated - global_func16 - because it can't run unprivileged for other reasons. This patch also fixes the tracking of the stack size for variable-offset reads. This second fix is bundled in the same commit as the first one because they're inter-related. Before this patch, writes to the stack using registers containing a variable offset (as opposed to registers with fixed, known values) were not properly contributing to the function's needed stack size. As a result, it was possible for a program to verify, but then to attempt to read out-of-bounds data at runtime because a too small stack had been allocated for it. Each function tracks the size of the stack it needs in bpf_subprog_info.stack_depth, which is maintained by update_stack_depth(). For regular memory accesses, check_mem_access() was calling update_state_depth() but it was passing in only the fixed part of the offset register, ignoring the variable offset. This was incorrect; the minimum possible value of that register should be used instead. This tracking is now fixed by centralizing the tracking of stack size in grow_stack_state(), and by lifting the calls to grow_stack_state() to check_stack_access_within_bounds() as suggested by Andrii. The code is now simpler and more convincingly tracks the correct maximum stack size. check_stack_range_initialized() can now rely on enough stack having been allocated for the access; this helps with the fix for the first issue. A few tests were changed to also check the stack depth computation. The one that fails without this patch is verifier_var_off:stack_write_priv_vs_unpriv.
CVE-2023-52454	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: Fix a kernel panic when host sends an invalid H2C PDU length If the host sends an H2CData command with an invalid DATAL, the kernel may crash in nvmet_tcp_build_pdu_iovec(). Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Ir : nvmet_tcp_io_work+0x6ac/0x718 [nvmet_tcp] Call trace: process_one_work+0x174/0x3c8 worker_thread+0x2d0/0x3e8 kthread+0x104/0x110 Fix the bug by raising a fatal error if DATAL isn't coherent with the packet size. Also, the PDU length should never exceed the MAXH2CDATA parameter which has been communicated to the host in nvmet_tcp_handle_icreq().
CVE-2023-52458	In the Linux kernel, the following vulnerability has been resolved: block: add check that partition length needs to be aligned with block size Before calling add partition or resize partition, there is no check on whether the length is aligned with the logical block size. If the logical block size of the disk is larger than 512 bytes, then the partition size maybe not the multiple of the logical block size, and when the last sector is read, bio_truncate() will adjust the bio size, resulting in an IO error if the size of the read command is smaller than the logical block size.If integrity data is supported, this will also result in a null pointer dereference when calling bio_integrity_free.
CVE-2023-52465	In the Linux kernel, the following vulnerability has been resolved: power: supply: Fix null pointer dereference in smb2_probe devm_kasprintf and devm_kzalloc return a pointer to dynamically allocated memory which can be NULL upon failure.
CVE-2023-52468	In the Linux kernel, the following vulnerability has been resolved: class: fix use-after-free in class_register() The lock_class_key is still registered and can be found in lock_keys_hash hlist after subsys_private is freed in error handler path.A task who iterate over the lock_keys_hash later may cause use-after-free.So fix that up and unregister the lock_class_key before kfree(cp). On our platform, a driver fails to kset_register because of creating duplicate filename '/class/xxx'.With Kasan enabled, it prints a invalid-access bug report. KASAN: invalid-access in lockdep_register_key +0x19c/0x1bc Write of size 8 at addr 15ffff808b8c0368 by task modprobe/252 Pointer tag: [15], memory tag: [fe] CPU: 7 PID: 252 Comm: modprobe Tainted: G W 6.6.0-mainline-maybe-dirty #1 Call trace: dump_backtrace+0x1b0/0x1e4 show_stack+0x2c/0x40 dump_stack_lvl+0xac/0xe0 print_report +0x18c/0x4d8 kasan_report+0xe8/0x148 __hwasan_store8_noabort+0x88/0x98 lockdep_register_key +0x19c/0x1bc class_register+0x94/0x1ec init_module+0xabc/0xf48 [rfkill] do_one_initcall+0x17c/0x72c do_init_module+0x19c/0x3f8 ... Memory state around the buggy address: ffffff808b8c0100: 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a ffffff808b8c0200: 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a 8a >fffff808b8c0300: fe fe fe fe fe fe fe fe fe fe fe fe fe fe fe fe ^ ffffff808b8c0400: 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 As CONFIG_KASAN_GENERIC is not set, Kasan reports invalid-access not use-after-free here.In this case, modprobe is manipulating the corrupted lock_keys_hash hlist where lock_class_key is already freed before. It's worth noting that this only can happen if lockdep is enabled, which is not true for normal system.

CVE ID	Description
CVE-2023-52474	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix bugs with non-PAGE_SIZE-end multi-iovec user SDMA requests hfi1 user SDMA request processing has two bugs that can cause data corruption for user SDMA requests that have multiple payload iovecs where an iovec other than the tail iovec does not run up to the page boundary for the buffer pointed to by that iovec. Here are the specific bugs: 1. user_sdma_txadd() does not use struct user_sdma_iovec->iov.iiov_len. Rather, user_sdma_txadd() will add up to PAGE_SIZE bytes from iovec to the packet, even if some of those bytes are past iovec->iov.iiov_len and are thus not intended to be in the packet. 2. user_sdma_txadd() and user_sdma_send_pkts() fail to advance to the next iovec in user_sdma_request->iovs when the current iovec is not PAGE_SIZE and does not contain enough data to complete the packet. The transmitted packet will contain the wrong data from the iovec pages. This has not been an issue with SDMA packets from hfi1 Verbs or PSM2 because they only produce iovecs that end short of PAGE_SIZE as the tail iovec of an SDMA request. Fixing these bugs exposes other bugs with the SDMA pin cache (struct mmu_rb_handler) that get in way of supporting user SDMA requests with multiple payload iovecs whose buffers do not end at PAGE_SIZE. So this commit fixes those issues as well. Here are the mmu_rb_handler bugs that non-PAGE_SIZE-end multi-iovec payload user SDMA requests can hit: 1. Overlapping memory ranges in mmu_rb_handler will result in duplicate pinnings. 2. When extending an existing mmu_rb_handler entry (struct mmu_rb_node), the mmu_rb code (1) removes the existing entry under a lock, (2) releases that lock, pins the new pages, (3) then reacquires the lock to insert the extended mmu_rb_node. If someone else comes in and inserts an overlapping entry between (2) and (3), insert in (3) will fail. The failure path code in this case unpins _all_ pages in either the original mmu_rb_node or the new mmu_rb_node that was inserted between (2) and (3). 3. In hfi1_mmu_rb_remove_unless_exact(), mmu_rb_node->refcount is incremented outside of mmu_rb_handler->lock. As a result, mmu_rb_node could be evicted by another thread that gets mmu_rb_handler->lock and checks mmu_rb_node->refcount before mmu_rb_node->refcount is incremented. 4. Related to #2 above, SDMA request submission failure path does not check mmu_rb_node->refcount before freeing mmu_rb_node object. If there are other SDMA requests in progress whose iovecs have pointers to the now-freed mmu_rb_node(s), those pointers to the now-freed mmu_rb nodes will be dereferenced when those SDMA requests complete.
CVE-2023-52475	In the Linux kernel, the following vulnerability has been resolved: Input: powermate - fix use-after-free in powermate_config_complete syzbot has found a use-after-free bug [1] in the powermate driver. This happens when the device is disconnected, which leads to a memory free from the powermate_device struct. When an asynchronous control message completes after the kfree and its callback is invoked, the lock does not exist anymore and hence the bug. Use usb_kill_urb() on pm->config to cancel any in-progress requests upon device disconnection. [1] https://syzkaller.appspot.com/bug?extid=0434ac83f907a1dbdd1e
CVE-2023-52481	In the Linux kernel, the following vulnerability has been resolved: arm64: errata: Add Cortex-A520 speculative unprivileged load workaround Implement the workaround for ARM Cortex-A520 erratum 2966298. On an affected Cortex-A520 core, a speculatively executed unprivileged load might leak data from a privileged load via a cache side channel. The issue only exists for loads within a translation regime with the same translation (e.g. same ASID and VMID). Therefore, the issue only affects the return to EL0. The workaround is to execute a TLBI before returning to EL0 after all loads of privileged data. A non-shareable TLBI to any address is sufficient. The workaround isn't necessary if page table isolation (KPTI) is enabled, but for simplicity it will be. Page table isolation should normally be disabled for Cortex-A520 as it supports the CSV3 feature and the E0PD feature (used when KASLR is enabled).
CVE-2023-52482	In the Linux kernel, the following vulnerability has been resolved: x86/srso: Add SRSO mitigation for Hygon processors Add mitigation for the speculative return stack overflow vulnerability which exists on Hygon processors too.
CVE-2023-52498	In the Linux kernel, the following vulnerability has been resolved: PM: sleep: Fix possible deadlocks in core system-wide PM code It is reported that in low-memory situations the system-wide resume core code deadlocks, because async_schedule_dev() executes its argument function synchronously if it cannot allocate memory (and not only in that case) and that function attempts to acquire a mutex that is already held. Executing the argument function synchronously from within dpm_async_fn() may also be problematic for ordering reasons (it may cause a consumer device's resume callback to be invoked before a requisite supplier device's one, for example). Address this by changing the code in question to use async_schedule_dev_nocall() for scheduling the asynchronous execution of device suspend and resume functions and to directly run them synchronously if async_schedule_dev_nocall() returns false.

CVE ID	Description
CVE-2023-52516	In the Linux kernel, the following vulnerability has been resolved: dma-debug: don't call <code>__dma_entry_alloc_check_leak()</code> under <code>free_entries_lock</code> <code>__dma_entry_alloc_check_leak()</code> calls into <code>printk -></code> serial console output (qcom geni) and grabs <code>port->lock</code> under <code>free_entries_lock</code> spin lock, which is a reverse locking dependency chain as <code>qcom_geni</code> IRQ handler can call into dma-debug code and grab <code>free_entries_lock</code> under <code>port->lock</code>. Move <code>__dma_entry_alloc_check_leak()</code> call out of <code>free_entries_lock</code> scope so that we don't acquire serial console's <code>port->lock</code> under it. Trimmed-down lockdep splat: The existing dependency chain (in reverse order) is: <code>-> #2 (free_entries_lock){-.-}- {2:2}: _raw_spin_lock_irqsave+0x60/0x80 dma_entry_alloc+0x38/0x110 debug_dma_map_page+0x60/0xf8 dma_map_page_attrs+0x1e0/0x230 dma_map_single_attrs.constprop.0+0x6c/0xc8 geni_se_rx_dma_prep+0x40/0xcc qcom_geni_serial_isr+0x310/0x510 __handle_irq_event_percpu+0x110/0x244 handle_irq_event_percpu+0x20/0x54 handle_irq_event+0x50/0x88 handle_fastevent_irq+0xa4/0xcc handle_irq_desc+0x28/0x40 generic_handle_domain_irq+0x24/0x30 gic_handle_irq+0xc4/0x148 do_interrupt_handler+0xa4/0xb0 el1_interrupt+0x34/0x64 el1h_64_irq_handler+0x18/0x24 el1h_64_irq+0x64/0x68 arch_local_irq_enable+0x4/0x8 ____do_softirq+0x18/0x24 ... -> #1 (&port_lock_key){-.-}- {2:2}: _raw_spin_lock_irqsave+0x60/0x80 qcom_geni_serial_console_write+0x184/0x1dc console_flush_all+0x344/0x454 console_unlock+0x94/0xf0 vprintk_emit+0x238/0x24c vprintk_default+0x3c/0x48 vprintk+0xb4/0xbc _printk+0x68/0x90 register_console+0x230/0x38c uart_add_one_port+0x338/0x494 qcom_geni_serial_probe+0x390/0x424 platform_probe+0x70/0xc0 really_probe+0x148/0x280 __driver_probe_device+0xfc/0x114 driver_probe_device+0x44/0x100 __device_attach_driver+0x64/0xdc bus_for_each_drv+0xb0/0xd8 __device_attach+0xe4/0x140 device_initial_probe+0x1c/0x28 bus_probe_device+0x44/0xb0 device_add+0x538/0x668 of_device_add+0x44/0x50 of_platform_device_create_pdata+0x94/0xc8 of_platform_bus_create+0x270/0x304 of_platform_populate+0xac/0xc4 devm_of_platform_populate+0x60/0xac geni_se_probe+0x154/0x160 platform_probe+0x70/0xc0 ... -> #0 (console_owner){-...}- {0:0}: __lock_acquire+0xdf8/0x109c lock_acquire+0x234/0x284 console_flush_all+0x330/0x454 console_unlock+0x94/0xf0 vprintk_emit+0x238/0x24c vprintk_default+0x3c/0x48 vprintk+0xb4/0xbc _printk+0x68/0x90 dma_entry_alloc+0xb4/0x110 debug_dma_map_sg+0xdc/0x2f8 __dma_map_sg_attrs+0xac/0xe4 dma_map_sgtable+0x30/0x4c get_pages+0x1d4/0x1e4 [msm] msm_gem_pin_pages_locked+0x38/0xac [msm] msm_gem_pin_vma_locked+0x58/0x88 [msm] msm_ioctl_gem_submit+0xde4/0x13ac [msm] drm_ioctl_kernel+0xe0/0x15c drm_ioctl+0x2e8/0x3f4 vfs_ioctl+0x30/0x50 ... Chain exists of: console_owner --> &port_lock_key --> free_entries_lock Possible unsafe locking scenario: CPU0 CPU1 ---- ---- lock(free_entries_lock); lock(&port_lock_key); lock(free_entries_lock); lock(console_owner); *** DEADLOCK *** Call trace: dump_backtrace+0xb4/0xf0 show_stack+0x20/0x30 dump_stack_lvl+0x60/0x84 dump_stack+0x18/0x24 print_circular_bug+0x1cc/0x234 check_noncircular+0x78/0xac __lock_acquire+0xdf8/0x109c lock_acquire+0x234/0x284 console_flush_all+0x330/0x454 consol --- truncated---</code>
CVE-2023-52519	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: ipc: Disable and reenables ACPI GPE bit The EHL (Elkhart Lake) based platforms provide a OOB (Out of band) service, which allows to wakeup device when the system is in S5 (Soft-Off state). This OOB service can be enabled/disabled from BIOS settings. When enabled, the ISH device gets PME wake capability. To enable PME wakeup, driver also needs to enable ACPI GPE bit. On resume, BIOS will clear the wakeup bit. So driver need to re-enable it in resume function to keep the next wakeup capability. But this BIOS clearing of wakeup bit doesn't decrement internal OS GPE reference count, so this reenabling on every resume will cause reference count to overflow. So first disable and reenables ACPI GPE bit using <code>acpi_disable_gpe()</code>.
CVE-2023-52532	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix TX CQE error handling For an unknown TX CQE error type (probably from a newer hardware), still free the SKB, update the queue tail, etc., otherwise the accounting will be wrong. Also, TX errors can be triggered by injecting corrupted packets, so replace the <code>WARN_ONCE</code> to <code>ratelimited</code> error logging.
CVE-2023-52568	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Resolves SECS reclaim vs. page fault for EAUG race The SGX EPC reclaimer (<code>ksgxd</code>) may reclaim the SECS EPC page for an enclave and set <code>secs.epc_page</code> to <code>NULL</code>. The SECS page is used for EAUG and ELDU in the SGX page fault handler. However, the <code>NULL</code> check for <code>secs.epc_page</code> is only done for ELDU, not EAUG before being used. Fix this by doing the same <code>NULL</code> check and reloading of the SECS page as needed for both EAUG and ELDU. The SECS page holds global enclave metadata. It can only be reclaimed when there are no other enclave pages remaining. At that point, virtually nothing can be done with the enclave until the SECS page is paged back in. An enclave can not run nor generate page faults without a resident SECS page. But it is still possible for a #PF for a non-SECS page to race with paging out the SECS page: when the last resident non-SECS page A triggers a #PF in a non-resident page B, and then page A and the SECS both are paged out before the #PF on B is handled. Hitting this bug requires that race triggered with a #PF for EAUG. Following is a trace when it happens. BUG: kernel <code>NULL</code> pointer dereference, address: 0000000000000000 RIP: 0010:sgx_encl_eaug_page+0xc7/0x210 Call Trace: ? __kmem_cache_alloc_node+0x16a/0x440 ? xa_load+0x6e/0xa0 sgx_vma_fault+0x119/0x230 __do_fault+0x36/0x140 do_fault+0x12f/0x400 __handle_mm_fault+0x728/0x1110 handle_mm_fault+0x105/0x310 do_user_addr_fault+0x1ee/0x750 ? __this_cpu_preempt_check+0x13/0x20 exc_page_fault+0x76/0x180 asm_exc_page_fault+0x27/0x30

CVE ID	Description
CVE-2023-52570	In the Linux kernel, the following vulnerability has been resolved: vfio/mdev: Fix a null-ptr-deref bug for mdev_unregister_parent() Inject fault while probing mdpv.ko, if kstrdup() of create_dir() fails in kobject_add_internal() in kobject_init_and_add() in mdev_type_add() in parent_create_sysfs_files(), it will return 0 and probe successfully. And when rmmod mdpv.ko, the mdpv_dev_exit() will call mdev_unregister_parent(), the mdev_type_remove() may traverse uninitialized parent->types[i] in parent_remove_sysfs_files(), and it will cause below null-ptr-deref. If mdev_type_add() fails, return the error code and kset_unregister() to fix the issue. general protection fault, probably for non-canonical address 0xdffffc0000000002: 0000 [0] PREEMPT SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017] CPU: 2 PID: 10215 Comm: rmmod Tainted: G W N 6.6.0-rc2+ #20 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 RIP: 0010:___kobject_del+0x62/0x1c0 Code: 48 89 fa 48 c1 ea 03 80 3c 02 00 0f 85 51 01 00 00 48 b8 00 00 00 00 00 fc ff df 48 8b 6b 28 48 8d 7d 10 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 24 01 00 00 48 8b 75 10 48 89 df 48 8d 6b 3c e8 RSP: 0018:ffff88810695fd30 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: ffffffff0270268 RCX: 0000000000000000 RDX: 0000000000000002 RSI: 0000000000000004 RDI: 0000000000000010 RBP: 0000000000000000 R08: 0000000000000001 R09: ffffd10233a4ef1 R10: ffff888119d2778b R11: 0000000063666572 R12: 0000000000000000 R13: fffffbfff404e2d4 R14: dffffc0000000000 R15: ffffffff0271660 FS: 00007fbc81981540(0000) GS:ffff888119d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fc14a142dc0 CR3: 0000000110a62003 CR4: 0000000000770ee0 DR0: ffffffff8fb0bce8 DR1: ffffffff8fb0bce9 DR2: ffffffff8fb0bcea DR3: ffffffff8fb0bceb DR6: 00000000fffe0ff0 DR7: 0000000000000060 PKRU: 55555554 Call Trace: <TASK> ? die_addr+0x3d/0xa0 ? exc_general_protection+0x144/0x220 ? asm_exc_general_protection+0x22/0x30 ? ___kobject_del+0x62/0x1c0 kobject_del+0x32/0x50 parent_remove_sysfs_files+0xd6/0x170 [mdev] mdev_unregister_parent+0xfb/0x190 [mdev] ? mdev_register_parent+0x270/0x270 [mdev] ? find_module_all+0x9d/0xe0 mdpv_dev_exit+0x17/0x63 [mdpv] __do_sys_delete_module.constprop.0+0x2fa/0x4b0 ? module_flags+0x300/0x300 ? __fput+0x4e7/0xa00 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7fbc813221b7 Code: 73 01 c3 48 8b 0d d1 8c 2c 00 f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 44 00 00 b8 b0 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d a1 8c 2c 00 f7 d8 64 89 01 48 RSP: 002b:00007ffe780e0648 EFLAGS: 00000206 ORIG_RAX: 00000000000000b0 RAX: ffffffff8b0bcefa RBX: 00007ffe780e06a8 RCX: 00007fbc813221b7 RDX: 000000000000000a RSI: 0000000000000800 RDI: 000055e214df9b58 RBP: 000055e214df9af0 R08: 00007ffe780df5c1 R09: 0000000000000000 R10: 00007fbc8139ecc0 R11: 0000000000000206 R12: 00007ffe780e0870 R13: 00007ffe780e0ed0 R14: 000055e214df9260 R15: 000055e214df9af0 </TASK> Modules linked in: mdpv(-) mdev vfio_iommu_type1 vfio [last unloaded: mdpv] Dumping ftrace buffer: (ftrace buffer empty) ---[end trace 0000000000000000]--- RIP: 0010:___kobject_del+0x62/0x1c0 Code: 48 89 fa 48 c1 ea 03 80 3c 02 00 0f 85 51 01 00 00 48 b8 00 00 00 00 00 fc ff df 48 8b 6b 28 48 8d 7d 10 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 24 01 00 00 48 8b 75 10 48 89 df 48 8d 6b 3c e8 RSP: 0018:ffff88810695fd30 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: ffffffff0270268 RCX: 0000000000000000 RDX: 0000000000000002 RSI: 0000000000000004 RDI: 0000000000000010 RBP: 0000000000000000 R08: 0000000000000001 R09: ffffd10233a4ef1 R10: ffff888119d2778b R11: 0000000063666572 R12: 0000000000000000 R13: fffffbfff404e2d4 R14: dffffc0000000000 R15: ffffffff0271660 FS: 00007fbc81981540(0000) GS:ffff888119d00000(000 ---truncated---
CVE-2023-52583	In the Linux kernel, the following vulnerability has been resolved: ceph: fix deadlock or deadcode of misusing dget() The lock order is incorrect between dentry and its parent, we should always make sure that the parent get the lock first. But since this deadcode is never used and the parent dir will always be set from the callers, let's just remove it.

CVE ID	Description
CVE-2023-52587	In the Linux kernel, the following vulnerability has been resolved: IB/ipoib: Fix mcast list locking Releasing the `priv->lock` while iterating the `priv->multicast_list` in `ipoib_mcast_join_task()` opens a window for `ipoib_mcast_dev_flush()` to remove the items while in the middle of iteration. If the mcast is removed while the lock was dropped, the for loop spins forever resulting in a hard lockup (as was reported on RHEL 4.18.0-372.75.1.el8_6 kernel): Task A (kworker/u72:2 below) Task B (kworker/u72:0 below) ----- ipoib_mcast_join_task(work) ipoib_ib_dev_flush_light(work) spin_lock_irq(&priv->lock) __ipoib_ib_dev_flush(priv, ...) list_for_each_entry(mcast, ipoib_mcast_dev_flush(dev = priv->dev) &priv->multicast_list, list) ipoib_mcast_join(dev, mcast) spin_unlock_irq(&priv->lock) spin_lock_irqsave(&priv->lock, flags) list_for_each_entry_safe(mcast, tmcast, &priv->multicast_list, list) list_del(&mcast->list); list_add_tail(&mcast->list, &remove_list) spin_unlock_irqrestore(&priv->lock, flags) spin_lock_irq(&priv->lock) ipoib_mcast_remove_list(&remove_list) (Here, `mcast` is no longer on the list_for_each_entry_safe(mcast, tmcast, `priv->multicast_list` and we keep remove_list, list) spinning on the `remove_list` of >>> wait_for_completion(&mcast->done) the other thread which is blocked and the list is still valid on it's stack.) Fix this by keeping the lock held and changing to GFP_ATOMIC to prevent eventual sleeps. Unfortunately we could not reproduce the lockup and confirm this fix but based on the code review I think this fix should address such lockups. crash> bc 31 PID: 747 TASK: ff1c6a1a007e8000 CPU: 31 COMMAND: "kworker/u72:2" -- [exception RIP: ipoib_mcast_join_task+0x1b1] RIP: ffffffff0944ac1 RSP: ff646f199a8c7e00 RFLAGS: 00000002 RAX: 0000000000000000 RBX: ff1c6a1a04dc82f8 RCX: 0000000000000000 work (&priv->mcast_task{,work}) RDX: ff1c6a192d60ac68 RSI: 0000000000000286 RDI: ff1c6a1a04dc8000 &mcast->list RBP: ff646f199a8c7e90 R8: ff1c699980019420 R9: ff1c6a1920c9a000 R10: ff646f199a8c7e00 R11: ff1c6a191a7d9800 R12: ff1c6a192d60ac00 mcast R13: ff1c6a1d82200000 R14: ff1c6a1a04dc8000 R15: ff1c6a1a04dc82d8 dev priv (&priv->lock) &priv->multicast_list (aka head) ORIG_RAX: ffffffff0944ac1 CS: 0010 SS: 0018 --- <NMI exception stack> --- #5 [ff646f199a8c7e00] ipoib_mcast_join_task+0x1b1 at ffffffff0944ac1 [ib_ipoib] #6 [ff646f199a8c7e98] process_one_work+0x1a7 at ffffffff9bf10967 crash> rx ff646f199a8c7e68 ff646f199a8c7e68: ff1c6a1a04dc82f8 <<< work = &priv->mcast_task.work crash> list -hO ipoib_dev_priv.multicast_list ff1c6a1a04dc8000 (empty) crash> ipoib_dev_priv.mcast_task.work.func,mcast_mutex.owner.counter ff1c6a1a04dc8000 mcast_task.work.func = 0xffffffff0944910 <ipoib_mcast_join_task>, mcast_mutex.owner.counter = 0xff1c69998efec000 crash> b 8 PID: 8 TASK: ff1c69998efec000 CPU: 33 COMMAND: "kworker/u72:0" -- #3 [ff646f1980153d50] wait_for_completion+0x96 at ffffffff9c7d7646 #4 [ff646f1980153d90] ipoib_mcast_remove_list+0x56 at ffffffff0944dc6 [ib_ipoib] #5 [ff646f1980153de8] ipoib_mcast_dev_flush+0x1a7 at ffffffff09455a7 [ib_ipoib] #6 [ff646f1980153e58] __ipoib_ib_dev_flush+0x1a4 at ffffffff09431a4 [ib_ipoib] #7 [ff ---truncated---
CVE-2023-52614	In the Linux kernel, the following vulnerability has been resolved: PM / devfreq: Fix buffer overflow in trans_stat_show Fix buffer overflow in trans_stat_show(). Convert simple snprintf to the more secure scnprintf with size of PAGE_SIZE. Add condition checking if we are exceeding PAGE_SIZE and exit early from loop. Also add at the end a warning that we exceeded PAGE_SIZE and that stats is disabled. Return -EFBIG in the case where we don't have enough space to write the full transition table. Also document in the ABI that this function can return -EFBIG error.
CVE-2023-52621	In the Linux kernel, the following vulnerability has been resolved: bpf: Check rcu_read_lock_trace_held() before calling bpf map helpers These three bpf_map_{lookup,update,delete}_elem() helpers are also available for sleepable bpf program, so add the corresponding lock assertion for sleepable bpf program, otherwise the following warning will be reported when a sleepable bpf program manipulates bpf map under interpreter mode (aka bpf_jit_enable=0): WARNING: CPU: 3 PID: 4985 at kernel/bpf/helpers.c:40 CPU: 3 PID: 4985 Comm: test_progs Not tainted 6.6.0+ #2 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) RIP: 0010:bpf_map_lookup_elem+0x54/0x60 Call Trace: <TASK> ? __warn+0xa5/0x240 ? bpf_map_lookup_elem+0x54/0x60 ? report_bug+0x1ba/0x1f0 ? handle_bug+0x40/0x80 ? exc_invalid_op+0x18/0x50 ? asm_exc_invalid_op+0x1b/0x20 ? __pfx_bpf_map_lookup_elem+0x10/0x10 ? rcu_lockdep_current_cpu_online+0x65/0xb0 ? rcu_is_watching+0x23/0x50 ? bpf_map_lookup_elem+0x54/0x60 ? __pfx_bpf_map_lookup_elem+0x10/0x10 __bpf_prog_run+0x513/0x3b70 __bpf_prog_run32+0x9d/0xd0 ? __bpf_prog_enter_sleepable_recur+0xad/0x120 ? __bpf_prog_enter_sleepable_recur+0x3e/0x120 bpf_trampoline_6442580665+0x4d/0x1000 __x64_sys_getpgid+0x5/0x30 ? do_syscall_64+0x36/0xb0 entry_SYSCALL_64_after_hwframe+0x6e/0x76 </TASK>
CVE-2023-52624	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Wake DMCUB before executing GPINT commands [Why] DMCUB can be in idle when we attempt to interface with the HW through the GPINT mailbox resulting in a system hang. [How] Add dc_wake_and_execute_gpint() to wrap the wake, execute, sleep sequence. If the GPINT executes successfully then DMCUB will be put back into sleep after the optional response is returned. It functions similar to the inbox command interface.

CVE ID	Description
CVE-2023-52625	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Refactor DMCUB enter/exit idle interface [Why] We can hang in place trying to send commands when the DMCUB isn't powered on. [How] We need to exit out of the idle state prior to sending a command, but the process that performs the exit also invokes a command itself. Fixing this issue involves the following: 1. Using a software state to track whether or not we need to start the process to exit idle or notify idle. It's possible for the hardware to have exited an idle state without driver knowledge, but entering one is always restricted to a driver allow - which makes the SW state vs HW state mismatch issue purely one of optimization, which should seldomly be hit, if at all. 2. Refactor any instances of exit/notify idle to use a single wrapper that maintains this SW state. This works similar to dc_allow_idle_optimizations, but works at the DMCUB level and makes sure the state is marked prior to any notify/exit idle so we don't enter an infinite loop. 3. Make sure we exit out of idle prior to sending any commands or waiting for DMCUB idle. This patch takes care of 1/2. A future patch will take care of wrapping DMCUB command submission with calls to this new interface.
CVE-2023-52632	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix lock dependency warning with srcu ===== WARNING: possible circular locking dependency detected 6.5.0-kfd-yangp #2289 Not tainted ----- kworker/0:2/996 is trying to acquire lock: (srcu){+.+.}-.{0:0}, at: __synchronize_srcu+0x5/0x1a0 but task is already holding lock: ((work_completion)(&svms->deferred_list_work)){+.+.}-.{0:0}, at: process_one_work+0x211/0x560 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #3 ((work_completion)(&svms->deferred_list_work)){+.+.}-.{0:0}: __flush_work+0x88/0x4f0 svm_range_list_lock_and_flush_work+0x3d/0x110 [amdgpu] svm_range_set_attr+0xd6/0x14c0 [amdgpu] kfd_ioctl+0x1d1/0x630 [amdgpu] __x64_sys_ioctl+0x88/0xc0 -> #2 (&info->lock#2){+.+.}-.{3:3}: __mutex_lock+0x99/0xc70 amdgpu_amdkfd_gpumv_restore_process_bos+0x54/0x740 [amdgpu] restore_process_helper+0x22/0x80 [amdgpu] restore_process_worker+0x2d/0xa0 [amdgpu] process_one_work+0x29b/0x560 worker_thread+0x3d/0x3d0 -> #1 ((work_completion)(&(&process->restore_work)->work)){+.+.}-.{0:0}: __flush_work+0x88/0x4f0 __cancel_work_timer+0x12c/0x1c0 kfd_process_notifier_release_internal+0x37/0x1f0 [amdgpu] __mmu_notifier_release+0xad/0x240 exit_mmap+0x6a/0x3a0 mmpu+0x6a/0x120 do_exit+0x322/0xb90 do_group_exit+0x37/0xa0 __x64_sys_exit_group+0x18/0x20 do_syscall_64+0x38/0x80 -> #0 (srcu){+.+.}-.{0:0}: __lock_acquire+0x1521/0x2510 lock_sync+0x5f/0x90 __synchronize_srcu+0x4f/0x1a0 __mmu_notifier_release+0x128/0x240 exit_mmap+0x6a/0x3a0 mmpu+0x6a/0x120 svm_range_deferred_list_work+0x19f/0x350 [amdgpu] process_one_work+0x29b/0x560 worker_thread+0x3d/0x3d0 other info that might help us debug this: Chain exists of: srcu --> &info->lock#2 --> (work_completion)(&svms->deferred_list_work) Possible unsafe locking scenario: CPU0 CPU1 ---- lock((work_completion)(&svms->deferred_list_work)); lock(&info->lock#2); lock((work_completion)(&svms->deferred_list_work)); sync(srcu);
CVE-2023-52634	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix disable_otg_wa logic [Why] When switching to another HDMI mode, we are unnecessarily disabling/enabling FIFO causing both HPO and DIG registers to be set at the same time when only HPO is supposed to be set. This can lead to a system hang the next time we change refresh rates as there are cases when we don't disable OTG/FIFO but FIFO is enabled when it isn't supposed to be. [How] Removing the enable/disable FIFO entirely.
CVE-2023-52639	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: vsie: fix race during shadow creation Right now it is possible to see gmap->private being zero in kvm_s390_vsie_gmap_notifier resulting in a crash. This is due to the fact that we add gmap->private == kvm after creation: static int acquire_gmap_shadow(struct kvm_vcpu *vcpu, struct vsie_page *vsie_page) { [...] gmap = gmap_shadow(vcpu->arch.gmap, asce, edat); if (IS_ERR(gmap)) return PTR_ERR(gmap); gmap->private = vcpu->kvm; Let children inherit the private field of the parent.
CVE-2023-52642	In the Linux kernel, the following vulnerability has been resolved: media: rc: bpf attach/detach requires write permission Note that bpf attach/detach also requires CAP_NET_ADMIN.
CVE-2023-52646	In the Linux kernel, the following vulnerability has been resolved: aio: fix mmap after fork null-deref Commit e4a0d3e720e7 ("aio: Make it possible to remap aio ring") introduced a null-deref if mmap is called on an old aio mapping after fork as mm->iocx_table will be set to NULL. [jmoyer@redhat.com: fix 80 column issue]
CVE-2023-52664	In the Linux kernel, the following vulnerability has been resolved: net: atlantic: eliminate double free in error handling logic Driver has a logic leak in ring data allocation/free, where aq_ring_free could be called multiple times on same ring, if system is under stress and got memory allocation error. Ring pointer was used as an indicator of failure, but this is not correct since only ring data is allocated/deallocated. Ring itself is an array member. Changing ring allocation functions to return error code directly. This simplifies error handling and eliminates aq_ring_free on higher layer.

CVE ID	Description
CVE-2023-52674	In the Linux kernel, the following vulnerability has been resolved: ALSA: scarlett2: Add clamp() in scarlett2_mixer_ctl_put() Ensure the value passed to scarlett2_mixer_ctl_put() is between 0 and SCARLETT2_MIXER_MAX_VALUE so we don't attempt to access outside scarlett2_mixer_values[].
CVE-2023-52676	In the Linux kernel, the following vulnerability has been resolved: bpf: Guard stack limits against 32bit overflow This patch promotes the arithmetic around checking stack bounds to be done in the 64-bit domain, instead of the current 32bit. The arithmetic implies adding together a 64-bit register with a int offset. The register was checked to be below 1<<29 when it was variable, but not when it was fixed. The offset either comes from an instruction (in which case it is 16 bit), from another register (in which case the caller checked it to be below 1<<29 [1]), or from the size of an argument to a kfunc (in which case it can be a u32 [2]). Between the register being inconsistently checked to be below 1<<29, and the offset being up to an u32, it appears that we were open to overflowing the `int`s which were currently used for arithmetic. [1] https://github.com/torvalds/linux/blob/815fb87b753055df2d9e50f6cd80eb10235fe3e9/kernel/bpf/verifier.c#L7494-L7498 [2] https://github.com/torvalds/linux/blob/815fb87b753055df2d9e50f6cd80eb10235fe3e9/kernel/bpf/verifier.c#L11904
CVE-2023-52680	In the Linux kernel, the following vulnerability has been resolved: ALSA: scarlett2: Add missing error checks to *_ctl_get() The *_ctl_get() functions which call scarlett2_update_*(*) were not checking the return value. Fix to check the return value and pass to the caller.
CVE-2023-52689	In the Linux kernel, the following vulnerability has been resolved: ALSA: scarlett2: Add missing mutex lock around get meter levels As scarlett2_meter_ctl_get() uses meter_level_map[], the data_mutex should be locked while accessing it.
CVE-2023-52697	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: sof_sdw_rt_sdca_jack_common: ctx->headset_codec_dev = NULL sof_sdw_rt_sdca_jack_exit() are used by different codecs, and some of them use the same dai name. For example, rt712 and rt713 both use "rt712-sdca-aif1" and sof_sdw_rt_sdca_jack_exit(). As a result, sof_sdw_rt_sdca_jack_exit() will be called twice by mc_dailink_exit_loop(). Set ctx->headset_codec_dev = NULL; after put_device(ctx->headset_codec_dev); to avoid ctx->headset_codec_dev being put twice.
CVE-2023-52698	In the Linux kernel, the following vulnerability has been resolved: calipso: fix memory leak in netlbl_calipso_add_pass() If IPv6 support is disabled at boot (ipv6.disable=1), the calipso_init() -> netlbl_calipso_ops_register() function isn't called, and the netlbl_calipso_ops_get() function always returns NULL. In this case, the netlbl_calipso_add_pass() function allocates memory for the doi_def variable but doesn't free it with the calipso_doi_free(). BUG: memory leak unreferenced object 0xffff888011d68180 (size 64): comm "syz-executor.1", pid 10746, jiffies 4295410986 (age 17.928s) hex dump (first 32 bytes): 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<...>] kmalloc include/linux/slab.h:552 [inline] [<...>] netlbl_calipso_add_pass net/netlabel/netlabel_calipso.c:76 [inline] [<...>] netlbl_calipso_add+0x22e/0x4f0 net/netlabel/netlabel_calipso.c:111 [<...>] genl_family_rcv_msg_doit +0x22f/0x330 net/netlink/genetlink.c:739 [<...>] genl_family_rcv_msg net/netlink/genetlink.c:783 [inline] [<...>] genl_rcv_msg+0x341/0x5a0 net/netlink/genetlink.c:800 [<...>] netlink_rcv_skb +0x14d/0x440 net/netlink/af_netlink.c:2515 [<...>] genl_rcv+0x29/0x40 net/netlink/genetlink.c:811 [<...>] netlink_unicast_kernel net/netlink/af_netlink.c:1313 [inline] [<...>] netlink_unicast+0x54b/0x800 net/netlink/af_netlink.c:1339 [<...>] netlink_sendmsg+0x90a/0xdf0 net/netlink/af_netlink.c:1934 [<...>] sock_sendmsg_nosec net/socket.c:651 [inline] [<...>] sock_sendmsg+0x157/0x190 net/socket.c:671 [<...>] ____sys_sendmsg+0x712/0x870 net/socket.c:2342 [<...>] __sys_sendmsg+0xf8/0x170 net/socket.c:2396 [<...>] __sys_sendmsg+0xea/0x1b0 net/socket.c:2429 [<...>] do_syscall_64+0x30/0x40 arch/x86/entry/common.c:46 [<...>] entry_SYSCALL_64_after_hwframe+0x61/0xc6 Found by InfoTeCS on behalf of Linux Verification Center (linuxtesting.org) with Syzkaller [PM: merged via the LSM tree at Jakub Kicinski request]
CVE-2023-52702	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix possible memory leak in ovs_meter_cmd_set() old_meter needs to be free after it is detached regardless of whether the new meter is successfully attached.
CVE-2023-52731	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix invalid page access after closing deferred I/O devices When a fbdev with deferred I/O is once opened and closed, the dirty pages still remain queued in the pageref list, and eventually later those may be processed in the delayed work. This may lead to a corruption of pages, hitting an Oops. This patch makes sure to cancel the delayed work and clean up the pageref list at closing the device for addressing the bug. A part of the cleanup code is factored out as a new helper function that is called from the common fb_release().

CVE ID	Description
CVE-2023-52732	In the Linux kernel, the following vulnerability has been resolved: ceph: blocklist the kclient when receiving corrupted snap trace When received corrupted snap trace we don't know what exactly has happened in MDS side. And we shouldn't continue IOs and metadatas access to MDS, which may corrupt or get incorrect contents. This patch will just block all the further IO/MDS requests immediately and then evict the kclient itself. The reason why we still need to evict the kclient just after blocking all the further IOs is that the MDS could revoke the caps faster.
CVE-2023-52735	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Don't let sock_map_{close,destroy,unhash} call itself sock_map proto callbacks should never call themselves by design. Protect against bugs like [1] and break out of the recursive loop to avoid a stack overflow in favor of a resource leak. [1] https://lore.kernel.org/all/00000000000073b14905ef2e7401@google.com/
CVE-2023-52736	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: Do not unset preset when cleaning up codec Several functions that take part in codec's initialization and removal are re-used by ASoC codec drivers implementations. Drivers mimic the behavior of hda_codec_driver_probe/remove() found in sound/pci/hda/hda_bind.c with their component->probe/remove() instead. One of the reasons for that is the expectation of snd_hda_codec_device_new() to receive a valid pointer to an instance of struct snd_card. This expectation can be met only once sound card components probing commences. As ASoC sound card may be unbound without codec device being actually removed from the system, unsetting ->preset in snd_hda_codec_cleanup_for_unbind() interferes with module unload -> load scenario causing null-ptr-deref. Preset is assigned only once, during device/driver matching whereas ASoC codec driver's module reloading may occur several times throughout the lifetime of an audio stack.
CVE-2023-52738	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/fence: Fix oops due to non-matching drm_sched init/fini Currently amdgpu calls drm_sched_fini() from the fence driver sw fini routine - such function is expected to be called only after the respective init function - drm_sched_init() - was executed successfully. Happens that we faced a driver probe failure in the Steam Deck recently, and the function drm_sched_fini() was called even without its counter-part had been previously called, causing the following oops: amdgpu: probe of 0000:04:00.0 failed with error -110 BUG: kernel NULL pointer dereference, address: 0000000000000090 PGD 0 P4D 0 Oops: 0002 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 609 Comm: systemd-udevd Not tainted 6.2.0-rc3-gpiccoli #338 Hardware name: Valve Jupiter/Jupiter, BIOS F7A0113 11/04/2022 RIP: 0010:drm_sched_fini+0x84/0xa0 [gpu_sched] [...] Call Trace: <TASK> amdgpu_fence_driver_sw_fini+0xc8/0xd0 [amdgpu] amdgpu_device_fini_sw+0x2b/0x3b0 [amdgpu] amdgpu_driver_release_kms+0x16/0x30 [amdgpu] devm_drm_dev_init_release+0x49/0x70 [...] To prevent that, check if the drm_sched was properly initialized for a given ring before calling its fini counter-part. Notice ideally we'd use sched.ready for that; such field is set as the latest thing on drm_sched_init(). But amdgpu seems to "override" the meaning of such field - in the above oops for example, it was a GFX ring causing the crash, and the sched.ready field was set to true in the ring init routine, regardless of the state of the DRM scheduler. Hence, we ended-up using sched.ops as per Christian's suggestion [0], and also removed the no_scheduler check [1]. [0] https://lore.kernel.org/amd-gfx/984ee981-2906-0eaf-cc9c-9f80975cb136@amd.com/ [1] https://lore.kernel.org/amd-gfx/cd0e2994-f85f-d837-609f-7056d5fb7231@amd.com/
CVE-2023-52739	In the Linux kernel, the following vulnerability has been resolved: Fix page corruption caused by racy check in __free_pages When we upgraded our kernel, we started seeing some page corruption like the following consistently: BUG: Bad page state in process ganesha.nfsd pfn:1304ca page:0000000022261c55 refcount:0 mapcount:-128 mapping:0000000000000000 index:0x0 pfn:0x1304ca flags: 0x17ffffc00000000 raw: 0017ffffc0000000 ffff8a513ffdc98 fffffee24b35ec08 0000000000000000 raw: 0000000000000000 0000000000000001 00000000fffff7f 0000000000000000 page dumped because: nonzero mapcount CPU: 0 PID: 15567 Comm: ganesha.nfsd Kdump: loaded Tainted: P B O 5.10.158-1.nutanix.20221209.el7.x86_64 #1 Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 04/05/2016 Call Trace: dump_stack+0x74/0x96 bad_page.cold+0x63/0x94 check_new_page_bad+0x6d/0x80 rmqueue+0x46e/0x970 get_page_from_freelist+0xcb/0x3f0 ? _cond_resched+0x19/0x40 __alloc_pages_nodemask+0x164/0x300 alloc_pages_current+0x87/0xf0 skb_page_frag_refill+0x84/0x110 ... Sometimes, it would also show up as corruption in the free list pointer and cause crashes. After bisecting the issue, we found the issue started from commit e320d3012d25 ("mm/page_alloc.c: fix freeing non-compound pages"): if (put_page_testzero(page)) free_the_page(page, order); else if (!PageHead(page)) while (order-- > 0) free_the_page(page + (1 << order), order); So the problem is the check PageHead is racy because at this point we already dropped our reference to the page. So even if we came in with compound page, the page can already be freed and PageHead can return false and we will end up freeing all the tail pages causing double free.

CVE ID	Description
CVE-2023-52741	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix use-after-free in rdata->read_into_pages() When the network status is unstable, use-after-free may occur when read data from the server. BUG: KASAN: use-after-free in readpages_fill_pages+0x14c/0x7e0 Call Trace: <TASK> dump_stack_lvl+0x38/0x4c print_report+0x16f/0x4a6 kasan_report+0xb7/0x130 readpages_fill_pages+0x14c/0x7e0 cifs_readv_receive+0x46d/0xa40 cifs_demultiplex_thread+0x121c/0x1490 kthread+0x16b/0x1a0 ret_from_fork+0x2c/0x50 </TASK> Allocated by task 2535: kasan_save_stack+0x22/0x50 kasan_set_track+0x25/0x30 __kasan_kmalloc+0x82/0x90 cifs_readdata_direct_alloc+0x2c/0x110 cifs_readdata_alloc+0x2d/0x60 cifs_readahead+0x393/0xfe0 read_pages+0x12f/0x470 page_cache_ra_unbounded+0x1b1/0x240 filemap_get_pages+0x1c8/0x9a0 filemap_read+0x1c0/0x540 cifs_strict_readv+0x21b/0x240 vfs_read+0x395/0x4b0 ksys_read+0xb8/0x150 do_syscall_64+0x3f/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc Freed by task 79: kasan_save_stack+0x22/0x50 kasan_set_track+0x25/0x30 kasan_save_free_info+0x2e/0x50 __kasan_slab_free+0x10e/0x1a0 __kmem_cache_free+0x7a/0x1a0 cifs_readdata_release+0x49/0x60 process_one_work+0x46c/0x760 worker_thread+0x2a4/0x6f0 kthread+0x16b/0x1a0 ret_from_fork+0x2c/0x50 Last potentially related work creation: kasan_save_stack+0x22/0x50 __kasan_record_aux_stack+0x95/0xb0 insert_work+0x2b/0x130 __queue_work+0x1fe/0x660 queue_work_on+0x4b/0x60 smb2_readv_callback+0x396/0x800 cifs_abort_connection+0x474/0x6a0 cifs_reconnect+0x5cb/0xa50 cifs_readv_from_socket.cold+0x22/0x6c cifs_read_page_from_socket+0xc1/0x100 readpages_fill_pages.cold+0x2f/0x46 cifs_readv_receive+0x46d/0xa40 cifs_demultiplex_thread+0x121c/0x1490 kthread+0x16b/0x1a0 ret_from_fork+0x2c/0x50 The following function calls will cause UAF of the rdata pointer. readpages_fill_pages cifs_read_page_from_socket cifs_readv_from_socket cifs_reconnect __cifs_reconnect cifs_abort_connection mid->callback() --> smb2_readv_callback queue_work(&rdata->work) # if the worker completes first, # the rdata is freed cifs_readv_complete kref_put cifs_readdata_release kfree(rdata) return rdata->... # UAF in readpages_fill_pages() Similarly, this problem also occurs in the uncached_fill_pages(). Fix this by adjusts the order of condition judgment in the return statement.

CVE ID	Description
CVE-2023-52743	In the Linux kernel, the following vulnerability has been resolved: ice: Do not use WQ_MEM_RECLAIM flag for workqueue When both ice and the irdma driver are loaded, a warning in check_flush_dependency is being triggered. This is due to ice driver workqueue being allocated with the WQ_MEM_RECLAIM flag and the irdma one is not. According to kernel documentation, this flag should be set if the workqueue will be involved in the kernel's memory reclamation flow. Since it is not, there is no need for the ice driver's WQ to have this flag set so remove it. Example trace: [+0.000004] workqueue: WQ_MEM_RECLAIM ice:ice_service_task [ice] is flushing !WQ_MEM_RECLAIM infiniband:0x0 [+0.000139] WARNING: CPU: 0 PID: 728 at kernel/workqueue.c:2632 check_flush_dependency+0x178/0x1a0 [+0.000011] Modules linked in: bonding tls xt_CHECKSUM xt_MASQUERADE xt_conntrack ipt_REJECT nf_reject_ipv4 nft_compat nft_cha in_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 nf_tables nfnetlink bridge stp llc rkill vfat fat intel_rapl_msr intel_rapl_common isst_if_common skx_edac nfit libnvdimm x86_pkg_temp_thermal intel_powerclamp coretemp kvm_intel kvm irqbypass crct10dif_pclmul crc32_pclmul ghash_clmulni_intel rapl intel_cstate rpcrdma sunrpc rdma_ucm ib_srpt ib_isert iscsi_target_mod target_core_mod ib_user libiscsi scsi_transport_iscsi rdma_cm ib_cm iw_cm iTCO_wdt iTCO_vendor_support ipmi_ssif irdma mei_me ib_uverbs ib_core intel_uncore joydev pcspkr i2c_i801 acpi_ipmi mei_lpc_ich i2c_smbus intel_pch_thermal ioatdma ipmi_si acpi_power_meter acpi_pad xfs libcrc32c sd_mod t10_pi crc64_rocksoft crc64_sg ahci ixgbe libahci ice_i40e igb crc32c_intel mdio_i2c_algo_bit liba_ta dca wmi dm_mirror dm_region_hash dm_log dm_mod ipmi_devintf ipmi_msghandler fuse [+0.000161] [last unloaded: bonding] [+0.000006] CPU: 0 PID: 728 Comm: kworker/0:2 Tainted: G S 6.2.0-rc2_next-queue-13jan-00458-gc20aabd57164 #1 [+0.000006] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0010.010620200716 01/06/2020 [+0.000003] Workqueue: ice ice_service_task [ice] [+0.000127] RIP: 0010:check_flush_dependency+0x178/0x1a0 [+0.000005] Code: 89 8e 02 01 e8 49 3d 40 00 49 8b 55 18 48 8d 8d d0 00 00 00 48 8d b3 d0 00 00 00 4d 89 e0 48 c7 c7 e0 3b 08 9f e8 bb d3 07 01 <0f> 0b e9 be fe ff ff 80 3d 24 89 8e 02 00 0f 85 6b ff ff ff e9 06 [+0.000004] RSP: 0018:ffff88810a39f990 EFLAGS: 00010282 [+0.000005] RAX: 0000000000000000 RBX: ffff888141bc2400 RCX: 0000000000000000 [+0.000004] RDX: 0000000000000001 RSI: dffffc0000000000 RDI: ffffffff1213a80 [+0.000003] RBP: ffff888194bf3400 R08: ffffd117b306112 R09: ffffd117b306112 [+0.000003] R10: ffff888bd983088b R11: ffffd117b306111 R12: 0000000000000000 [+0.000003] R13: ffff88811f84d00 R14: ffff88810a3943ac R15: ffff888194bf3400 [+0.000004] FS: 0000000000000000(0000) GS:ffff888bd9800000(0000) knlGS:0000000000000000 [+0.000003] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [+0.000003] CR2: 000056035b208b60 CR3: 000000017795e005 CR4: 00000000007706f0 [+0.000003] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [+0.000003] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [+0.000002] PKRU: 55555554 [+0.000003] Call Trace: [+0.000002] <TASK> [+0.000003] __flush_workqueue+0x203/0x840 [+0.000006] ? mutex_unlock+0x84/0xd0 [+0.000008] ? __pfx_mutex_unlock+0x10/0x10 [+0.000004] ? __pfx__flush_workqueue+0x10/0x10 [+0.000006] ? mutex_lock+0xa3/0xf0 [+0.000005] ib_cache_cleanup_one+0x39/0x190 [ib_core] [+0.000174] __ib_unregister_device+0x84/0xf0 [ib_core] [+0.000094] ib_unregister_device+0x25/0x30 [ib_core] [+0.000093] irdma_ib_unregister_device+0x97/0xc0 [irdma] [+0.000064] ? __pfx_irdma_ib_unregister_device+0x10/0x10 [irdma] [+0.000059] ? up_write+0x5c/0x90 [+0.000005] irdma_remove+0x36/0x90 [irdma] [+0.000062] auxiliary_bus_remove+0x32/0x50 [+0.000007] device_r ---truncated---
CVE-2023-52744	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix potential NULL-ptr-dereference in_dev_get() can return NULL which will cause a failure once idev is dereferenced in in_dev_for_each_ifa_rtnl(). This patch adds a check for NULL value in idev beforehand. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2023-52745	In the Linux kernel, the following vulnerability has been resolved: IB/IPoIB: Fix legacy IPoIB due to wrong number of queues The cited commit creates child PKEY interfaces over netlink will multiple tx and rx queues, but some devices doesn't support more than 1 tx and 1 rx queues. This causes to a crash when traffic is sent over the PKEY interface due to the parent having a single queue but the child having multiple queues. This patch fixes the number of queues to 1 for legacy IPoIB at the earliest possible point in time. BUG: kernel NULL pointer dereference, address: 000000000000036b PGD 0 P4D 0 Oops: 0000 [#1] SMP CPU: 4 PID: 209665 Comm: python3 Not tainted 6.1.0_for_upstream_min_debug_2022_12_12_17_02 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:kmem_cache_alloc+0xcb/0x450 Code: ce 7e 49 8b 50 08 49 83 78 10 00 4d 8b 28 0f 84 cb 02 00 00 4d 85 ed 0f 84 c2 02 00 00 41 8b 44 24 28 48 8d 4a 01 49 8b 3c 24 <49> 8b 5c 05 00 4c 89 e8 65 48 0f c7 0f 0f 94 c0 84 c0 74 b8 41 8b RSP: 0018:ffff88822acbbab8 EFLAGS: 00010202 RAX: 0000000000000070 RBX: ffff8881c28e3e00 RCX: 00000000064f8dae RDX: 00000000064f8dad RSI: 0000000000000a20 RDI: 000000000030d000 RBP: 0000000000000a20 R08: ffff8882f5d30d00 R09: ffff888104032f40 R10: ffff88810fade828 R11: 736f6d6570736575 R12: ffff88810081c000 R13: 00000000000002fb R14: ffffffff817fc865 R15: 0000000000000000 FS: 00007f9324ff9700(0000) GS:ffff8882f5d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000000036b CR3: 00000001125af004 CR4: 0000000000370ea0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> skb_clone+0x55/0xd0 ip6_finish_output2+0x3fe/0x690 ip6_finish_output+0xfa/0x310 ip6_send_skb+0x1e/0x60 udp_v6_send_skb+0x1e5/0x420 udpv6_sendmsg+0xb3c/0xe60 ? ip_mc_finish_output+0x180/0x180 ? __switch_to_asm+0x3a/0x60 ? __switch_to_asm+0x34/0x60 sock_sendmsg+0x33/0x40 __sys_sendto+0x103/0x160 ? _copy_to_user+0x21/0x30 ? kvm_clock_get_cycles+0xd/0x10 ? ktime_get_ts64+0x49/0xe0 __x64_sys_sendto+0x25/0x30 do_syscall_64+0x3d/0x90 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7f9374f1ed14 Code: 42 41 f8 ff 44 8b 4c 24 2c 4c 8b 44 24 20 89 c5 44 8b 54 24 28 48 8b 54 24 18 b8 2c 00 00 00 48 8b 74 24 10 8b 7c 24 08 0f 05 <48> 3d 00 f0 ff ff 77 34 89 ef 48 89 44 24 08 e8 68 41 f8 ff 48 8b RSP: 002b:00007f9324ff7bd0 EFLAGS: 00000293 ORIG_RAX: 000000000000002c RAX: ffffffff817fda RAX: 00007f9324ff7cc8 RCX: 00007f9374f1ed14 RDX: 00000000000002fb RSI: 00007f93000052f0 RDI: 0000000000000030 RBP: 0000000000000000 R08: 00007f9324ff7d40 R09: 000000000000001c R10: 0000000000000000 R11: 0000000000000293 R12: 0000000000000000 R13: 000000012a05f200 R14: 0000000000000001 R15: 00007f9374d57bdc </TASK>
CVE-2023-52747	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Restore allocated resources on failed copyout Fix a resource leak if an error occurs.
CVE-2023-52749	In the Linux kernel, the following vulnerability has been resolved: spi: Fix null dereference on suspend A race condition exists where a synchronous (noqueue) transfer can be active during a system suspend. This can cause a null pointer dereference exception to occur when the system resumes. Example order of events leading to the exception: 1. spi_sync() calls __spi_transfer_message_noqueue() which sets ctrl->cur_msg 2. Spi transfer begins via spi_transfer_one_message() 3. System is suspended interrupting the transfer context 4. System is resumed 6. spi_controller_resume() calls spi_start_queue() which resets cur_msg to NULL 7. Spi transfer context resumes and spi_finalize_current_message() is called which dereferences cur_msg (which is now NULL) Wait for synchronous transfers to complete before suspending by acquiring the bus mutex and setting/checking a suspend flag.

CVE ID	Description
CVE-2023-52750	In the Linux kernel, the following vulnerability has been resolved: arm64: Restrict CPU_BIG_ENDIAN to GNU as or LLVM IAS 15.x or newer Prior to LLVM 15.0.0, LLVM's integrated assembler would incorrectly byte-swap NOP when compiling for big-endian, and the resulting series of bytes happened to match the encoding of FNMADD S21, S30, S0, S0. This went unnoticed until commit: 34f66c4c4d5518c1 ("arm64: Use a positive cpucap for FP/SIMD") Prior to that commit, the kernel would always enable the use of FPSIMD early in boot when __cpu_setup() initialized CPACR_EL1, and so usage of FNMADD within the kernel was not detected, but could result in the corruption of user or kernel FPSIMD state. After that commit, the instructions happen to trap during boot prior to FPSIMD being detected and enabled, e.g. Unhandled 64-bit el1h sync exception on CPU0, ESR 0x000000001fe00000 -- ASIMD CPU: 0 PID: 0 Comm: swapper Not tainted 6.6.0-rc3-00013-g34f66c4c4d55 #1 Hardware name: linux,dummy-virt (DT) pstate: 400000c9 (nZcv daIF -PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : __pi_strcmp+0x1c/0x150 lr : populate_properties+0xe4/0x254 sp : ffffd014173d3ad0 x29: ffffd014173d3af0 x28: ffffbffffddffcb8 x27: 0000000000000000 x26: 0000000000000058 x25: ffffbffffddfe054 x24: 0000000000000008 x23: ffffbffffddfe000 x22: ffffbffffddfe000 x21: ffffbffffddfe044 x20: ffffd014173d3b70 x19: 0000000000000001 x18: 0000000000000005 x17: 0000000000000010 x16: 0000000000000000 x15: 00000000413e7000 x14: 0000000000000000 x13: 0000000000001bcc x12: 0000000000000000 x11: 00000000d00dfeed x10: ffffd414193f2cd0 x9 : 0000000000000000 x8 : 0101010101010101 x7 : ffffffffc0 x6 : 0000000000000000 x5 : 0000000000000000 x4 : 0101010101010101 x3 : 000000000000002a x2 : 0000000000000001 x1 : ffffd014171f2988 x0 : ffffbffffddffcb8 Kernel panic - not syncing: Unhandled exception CPU: 0 PID: 0 Comm: swapper Not tainted 6.6.0-rc3-00013-g34f66c4c4d55 #1 Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0xec/0x108 show_stack+0x18/0x2c dump_stack_lvl+0x50/0x68 dump_stack+0x18/0x24 panic+0x13c/0x340 el1t_64_irq_handler+0x0/0x1c el1_abort+0x0/0x5c el1h_64_sync+0x64/0x68 __pi_strcmp+0x1c/0x150 unflatten_dt_nodes+0x1e8/0x2d8 __unflatten_device_tree+0x5c/0x15c unflatten_device_tree+0x38/0x50 setup_arch+0x164/0x1e0 start_kernel+0x64/0x38c __primary_switched+0xbc/0xc4 Restrict CONFIG_CPU_BIG_ENDIAN to a known good assembler, which is either GNU as or LLVM's IAS 15.0.0 and newer, which contains the linked commit.
CVE-2023-52754	In the Linux kernel, the following vulnerability has been resolved: media: imon: fix access to invalid resource for the second interface imon driver probes two USB interfaces, and at the probe of the second interface, the driver assumes blindly that the first interface got bound with the same imon driver. It's usually true, but it's still possible that the first interface is bound with another driver via a malformed descriptor. Then it may lead to a memory corruption, as spotted by syzkaller; imon driver accesses the data from drvdata as struct imon_context object although it's a completely different one that was assigned by another driver. This patch adds a sanity check -- whether the first interface is really bound with the imon driver or not -- for avoiding the problem above at the probe time.
CVE-2023-52760	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix slab-use-after-free in gfs2_qd_dealloc In gfs2_put_super(), whether withdrawn or not, the quota should be cleaned up by gfs2_quota_cleanup(). Otherwise, struct gfs2_sbd will be freed before gfs2_qd_dealloc (rcu callback) has run for all gfs2_quota_data objects, resulting in use-after-free. Also, gfs2_destroy_threads() and gfs2_quota_cleanup() is already called by gfs2_make_fs_ro(), so in gfs2_put_super(), after calling gfs2_make_fs_ro(), there is no need to call them again.

CVE ID	Description
CVE-2023-52782	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Track xmit submission to PTP WQ after populating metadata map Ensure the skb is available in metadata mapping to skbs before tracking the metadata index for detecting undelivered CQEs. If the metadata index is put in the tracking list before putting the skb in the map, the metadata index might be used for detecting undelivered CQEs before the relevant skb is available in the map, which can lead to a null-ptr-deref. Log: general protection fault, probably for non-canonical address 0xdffffc0000000005: 0000 [#1] SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000028-0x000000000000002f] CPU: 0 PID: 1243 Comm: kworker/0:2 Not tainted 6.6.0-rc4+ #108 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 Workqueue: events mlx5e_rx_dim_work [mlx5_core] RIP: 0010:mlx5e_ptp_napi_poll+0x9a4/0x2290 [mlx5_core] Code: 8c 24 38 cc ff ff 4c 8d 3c c1 4c 89 f9 48 c1 e9 03 42 80 3c 31 00 0f 85 97 0f 00 00 4d 8b 3f 49 8d 7f 28 48 89 f9 48 c1 e9 03 <42> 80 3c 31 00 0f 85 8b 0f 00 00 49 8b 47 28 48 85 c0 0f 84 05 07 RSP: 0018:ffff8884d3c09c88 EFLAGS: 00010206 RAX: 0000000000000069 RBX: ffff8881160349d8 RCX: 0000000000000005 RDX: ffffd10218f48cf RSI: 0000000000000004 RDI: 0000000000000028 RBP: ffff888122707700 R08: 0000000000000001 R09: ffffd109a781383 R10: 0000000000000003 R11: 0000000000000003 R12: ffff88810c7a7a40 R13: ffff888122707700 R14: dffffc0000000000 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff8884d3c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f4f878dd6e0 CR3: 000000014d108002 CR4: 0000000000370eb0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <IRQ> ? die_addr+0x3c/0xa0 ? exc_general_protection+0x144/0x210 ? asm_exc_general_protection+0x22/0x30 ? mlx5e_ptp_napi_poll+0x9a4/0x2290 [mlx5_core] ? mlx5e_ptp_napi_poll+0x8f6/0x2290 [mlx5_core] __napi_poll.constprop.0+0xa4/0x580 net_rx_action+0x460/0xb80 ? _raw_spin_unlock_irqrestore+0x32/0x60 ? __napi_poll.constprop.0+0x580/0x580 ? tasklet_action_common.isra.0+0x2ef/0x760 __do_softirq+0x26c/0x827 irq_exit_rcu+0xc2/0x100 common_interrupt+0x7f/0xa0 </IRQ> <TASK> asm_common_interrupt+0x22/0x40 RIP: 0010: __kmem_cache_alloc_node+0xb/0x330 Code: 41 5d 41 5e 41 5f c3 8b 44 24 14 8b 4c 24 10 09 c8 eb d5 e8 b7 43 ca 01 0f 1f 80 00 00 00 0f 1f 44 00 00 55 48 89 e5 41 57 <41> 56 41 89 d6 41 55 41 89 f5 41 54 49 89 fc 53 48 83 e4 f0 48 83 RSP: 0018:ffff88812c4079c0 EFLAGS: 00000246 RAX: 1fffffffff083c7fe RBX: ffff888100042dc0 RCX: 0000000000000218 RDX: 00000000ffffffff RSI: 0000000000000dc0 RDI: ffff888100042dc0 RBP: ffff88812c4079c8 R08: ffffffff0289f96 R09: ffffd1025880ea9 R10: ffff888138839f80 R11: 0000000000000002 R12: 0000000000000dc0 R13: 0000000000000100 R14: 000000000000008c R15: ffff8881271fc450 ? cmd_exec+0x796/0x2200 [mlx5_core] kmalloc_trace+0x26/0xc0 cmd_exec+0x796/0x2200 [mlx5_core] mlx5_cmd_do+0x22/0xc0 [mlx5_core] mlx5_cmd_exec+0x17/0x30 [mlx5_core] mlx5_core_modify_cq_moderation+0x139/0x1b0 [mlx5_core] ? mlx5_add_cq_to_tasklet+0x280/0x280 [mlx5_core] ? lockdep_set_lock_cmp_fn+0x190/0x190 ? process_one_work+0x659/0x1220 mlx5e_rx_dim_work+0x9d/0x100 [mlx5_core] process_one_work+0x730/0x1220 ? lockdep_hardirqs_on_prepare+0x400/0x400 ? max_active_store+0xf0/0xf0 ? assign_work+0x168/0x240 worker_thread+0x70f/0x12d0 ? __kthread_parkme+0xd1/0x1d0 ? process_one_work+0x1220/0x1220 kthread+0x2d9/0x3b0 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x2d/0x70 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork_as ---truncated---
CVE-2023-52788	In the Linux kernel, the following vulnerability has been resolved: i915/perf: Fix NULL deref bugs with drm_dbg() calls When i915 perf interface is not available dereferencing it will lead to NULL dereferences. As returning -ENOTSUPP is pretty clear return when perf interface is not available. [tursulin: added stable tag] (cherry picked from commit 36f27350ff745bd228ab04d7845dfbffc177a889)
CVE-2023-52806	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: Fix possible null-ptr-deref when assigning a stream While AudioDSP drivers assign streams exclusively of HOST or LINK type, nothing blocks a user to attempt to assign a COUPLED stream. As supplied substream instance may be a stub, what is the case when code-loading, such scenario ends with null-ptr-deref.

CVE ID	Description
CVE-2023-52808	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Set debugfs_dir pointer to NULL after removing debugfs If init debugfs failed during device registration due to memory allocation failure, debugfs_remove_recursive() is called, after which debugfs_dir is not set to NULL. debugfs_remove_recursive() will be called again during device removal. As a result, illegal pointer is accessed. [1665.467244] hisi_sas_v3_hw 0000:b4:02:0: failed to init debugfs! ... [1669.836708] Unable to handle kernel NULL pointer dereference at virtual address 00000000000000a0 [1669.872669] pc : down_write+0x24/0x70 [1669.876315] lr : down_write+0x1c/0x70 [1669.879961] sp : ffff000036f53a30 [1669.883260] x29: ffff000036f53a30 x28: ffffa027c31549f8 [1669.888547] x27: ffffa027c3140000 x26: 0000000000000000 [1669.893834] x25: ffffa027bf37c270 x24: ffffa027bf37c270 [1669.899122] x23: ffff0000095406b8 x22: ffff0000095406a8 [1669.904408] x21: 0000000000000000 x20: ffffa027bf37c310 [1669.909695] x19: 00000000000000a0 x18: ffff8027dcd86f10 [1669.914982] x17: 0000000000000000 x16: 0000000000000000 [1669.920268] x15: 0000000000000000 x14: ffffa0274014f870 [1669.925555] x13: 0000000000000040 x12: 0000000000000228 [1669.930842] x11: 0000000000000020 x10: 00000000000000bb0 [1669.936129] x9 : ffff000036f537f0 x8 : ffff80273088ca10 [1669.941416] x7 : 000000000000001d x6 : 00000000ffffff [1669.946702] x5 : ffff000008a36310 x4 : ffff80273088be00 [1669.951989] x3 : ffff000009513e90 x2 : 0000000000000000 [1669.957276] x1 : 00000000000000a0 x0 : ffffffff00000001 [1669.962563] Call trace: [1669.965000] down_write+0x24/0x70 [1669.968301] debugfs_remove_recursive+0x5c/0x1b0 [1669.972905] hisi_sas_debugfs_exit+0x24/0x30 [hisi_sas_main] [1669.978541] hisi_sas_v3_remove+0x130/0x150 [hisi_sas_v3_hw] [1669.984175] pci_device_remove+0x48/0xd8 [1669.988082] device_release_driver_internal+0x1b4/0x250 [1669.993282] device_release_driver+0x28/0x38 [1669.997534] pci_stop_bus_device+0x84/0xb8 [1670.001611] pci_stop_and_remove_bus_device_locked+0x24/0x40 [1670.007244] remove_store+0xfc/0x140 [1670.010802] dev_attr_store+0x44/0x60 [1670.014448] sysfs_kf_write+0x58/0x80 [1670.018095] kernfs_fop_write+0xe8/0x1f0 [1670.022000] __vfs_write+0x60/0x190 [1670.025472] vfs_write+0xac/0x1c0 [1670.028771] ksys_write+0x6c/0xd8 [1670.032071] __arm64_sys_write+0x24/0x30 [1670.035977] el0_svc_common+0x78/0x130 [1670.039710] el0_svc_handler+0x38/0x78 [1670.043442] el0_svc+0x8/0xc To fix this, set debugfs_dir to NULL after debugfs_remove_recursive().
CVE-2023-52814	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential null pointer dereference The amdgpu_ras_get_context may return NULL if device not support ras feature, so add check before using.
CVE-2023-52815	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/vkms: fix a possible null pointer dereference In amdgpu_vkms_conn_get_modes(), the return value of drm_cvt_mode() is assigned to mode, which will lead to a NULL pointer dereference on failure of drm_cvt_mode(). Add a check to avoid null pointer dereference.
CVE-2023-52818	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Fix UBSAN array-index-out-of-bounds for SMU7 For ptable structs that use flexible array sizes, use flexible arrays.
CVE-2023-52819	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Fix UBSAN array-index-out-of-bounds for Polaris and Tonga For ptable structs that use flexible array sizes, use flexible arrays.
CVE-2023-52831	In the Linux kernel, the following vulnerability has been resolved: cpu/hotplug: Don't offline the last non-isolated CPU If a system has isolated CPUs via the "isolcpus=" command line parameter, then an attempt to offline the last housekeeping CPU will result in a WARN_ON() when rebuilding the scheduler domains and a subsequent panic due to and unhandled empty CPU mas in partition_sched_domains_locked(). cpuset_hotplug_workfn() rebuild_sched_domains_locked() ndoms = generate_sched_domains(&doms, &attr); cpumask_and(doms[0], top_cpuset.effective_cpus, housekeeping_cpumask(HK_FLAG_DOMAIN)); Thus results in an empty CPU mask which triggers the warning and then the subsequent crash: WARNING: CPU: 4 PID: 80 at kernel/sched/topology.c:2366 build_sched_domains+0x120c/0x1408 Call trace: build_sched_domains+0x120c/0x1408 partition_sched_domains_locked+0x234/0x880 rebuild_sched_domains_locked+0x37c/0x798 rebuild_sched_domains+0x30/0x58 cpuset_hotplug_workfn+0x2a8/0x930 Unable to handle kernel paging request at virtual address fffe80027ab37080 partition_sched_domains_locked+0x318/0x880 rebuild_sched_domains_locked+0x37c/0x798 Aside of the resulting crash, it does not make any sense to offline the last last housekeeping CPU. Prevent this by masking out the non-housekeeping CPUs when selecting a target CPU for initiating the CPU unplug operation via the work queue.

CVE ID	Description
CVE-2023-52833	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btusb: Add date->evt_skb is NULL check fix crash because of null pointers [6104.969662] BUG: kernel NULL pointer dereference, address: 00000000000000c8 [6104.969667] #PF: supervisor read access in kernel mode [6104.969668] #PF: error_code(0x0000) - not-present page [6104.969670] PGD 0 P4D 0 [6104.969673] Oops: 0000 [#1] SMP NOPTI [6104.969684] RIP: 0010:btusb_mtk_hci_wmt_sync+0x144/0x220 [btusb] [6104.969688] RSP: 0018:ffff8d681533d48 EFLAGS: 00010246 [6104.969689] RAX: 0000000000000000 RBX: ffff8ad560bb2000 RCX: 0000000000000006 [6104.969691] RDX: 0000000000000000 RSI: ffff8d681533d08 RDI: 0000000000000000 [6104.969692] RBP: ffff8d681533d70 R08: 0000000000000001 R09: 0000000000000001 [6104.969694] R10: 0000000000000001 R11: 00000000fa83b2da R12: ffff8ad461d1d7c0 [6104.969695] R13: 0000000000000000 R14: ffff8ad459618c18 R15: ffff8d681533d90 [6104.969697] FS: 00007f5a1cab9d40(0000) GS:ffff8ad578200000(0000) knlGS:00000 [6104.969699] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [6104.969700] CR2: 00000000000000c8 CR3: 000000018620c001 CR4: 0000000000760ef0 [6104.969701] PKRU: 55555554 [6104.969702] Call Trace: [6104.969708] btusb_mtk_shutdown+0x44/0x80 [btusb] [6104.969732] hci_dev_do_close+0x470/0x5c0 [bluetooth] [6104.969748] hci_rfkill_set_block+0x56/0xa0 [bluetooth] [6104.969753] rfkill_set_block+0x92/0x160 [6104.969755] rfkill_fop_write+0x136/0x1e0 [6104.969759] __vfs_write+0x18/0x40 [6104.969761] vfs_write+0xdf/0x1c0 [6104.969763] ksys_write+0xb1/0xe0 [6104.969765] __x64_sys_write+0x1a/0x20 [6104.969769] do_syscall_64+0x51/0x180 [6104.969771] entry_SYSCALL_64_after_hwframe+0x44/0xa9 [6104.969773] RIP: 0033:0x7f5a21f18fef [6104.9] RSP: 002b:00007ffef39010 EFLAGS: 00000293 ORIG_RAX: 0000000000000001 [6104.969780] RAX: ffffffff8d681533d0 RBX: 000055c10a7560a0 RCX: 00007f5a21f18fef [6104.969781] RDX: 0000000000000008 RSI: 00007ffef39060 RDI: 0000000000000012 [6104.969782] RBP: 00007ffef39060 R08: 0000000000000000 R09: 0000000000000017 [6104.969784] R10: 00007ffef39060 R11: 0000000000000293 R12: 0000000000000002 [6104.969785] R13: 00007ffef39220 R14: 00007ffef391a0 R15: 000055c10a72acf0
CVE-2023-52851	In the Linux kernel, the following vulnerability has been resolved: IB/mlx5: Fix init stage error handling to avoid double free of same QP and UAF In the unlikely event that workqueue allocation fails and returns NULL in mlx5_mkey_cache_init(), delete the call to mlx5r_umr_resource_cleanup() (which frees the QP) in mlx5_ib_stage_post_ib_reg_umr_init(). This will avoid attempted double free of the same QP when __mlx5_ib_add() does its cleanup. Resolves a splat: Syzkaller reported a UAF in ib_destroy_qp_user workqueue: Failed to create a rescuer kthread for wq "mkey_cache": -EINTR infiniband mlx5_0: mlx5_mkey_cache_init:981:(pid 1642): failed to create work queue infiniband mlx5_0: mlx5_ib_stage_post_ib_reg_umr_init:4075:(pid 1642): mr cache init failed -12 ===== BUG: KASAN: slab-use-after-free in ib_destroy_qp_user (drivers/infiniband/core/verbs.c:2073) Read of size 8 at addr ffff88810da310a8 by task repro_upstream/1642 Call Trace: <TASK> kasan_report (mm/kasan/report.c:590) ib_destroy_qp_user (drivers/infiniband/core/verbs.c:2073) mlx5r_umr_resource_cleanup (drivers/infiniband/hw/mlx5/umr.c:198) __mlx5_ib_add (drivers/infiniband/hw/mlx5/main.c:4178) mlx5r_probe (drivers/infiniband/hw/mlx5/main.c:4402) ... </TASK> Allocated by task 1642: __kmalloc (/include/linux/kasan.h:198 mm/slab_common.c:1026 mm/slab_common.c:1039) create_qp (/include/linux/slab.h:603 ./include/linux/slab.h:720 ./include/rdma/ib_verbs.h:2795 drivers/infiniband/core/verbs.c:1209) ib_create_qp_kernel (drivers/infiniband/core/verbs.c:1347) mlx5r_umr_resource_init (drivers/infiniband/hw/mlx5/umr.c:164) mlx5_ib_stage_post_ib_reg_umr_init (drivers/infiniband/hw/mlx5/main.c:4070) __mlx5_ib_add (drivers/infiniband/hw/mlx5/main.c:4168) mlx5r_probe (drivers/infiniband/hw/mlx5/main.c:4402) ... Freed by task 1642: __kmem_cache_free (mm/slab.c:1826 mm/slab.c:3809 mm/slab.c:3822) ib_destroy_qp_user (drivers/infiniband/core/verbs.c:2112) mlx5r_umr_resource_cleanup (drivers/infiniband/hw/mlx5/umr.c:198) mlx5_ib_stage_post_ib_reg_umr_init (drivers/infiniband/hw/mlx5/main.c:4076 drivers/infiniband/hw/mlx5/main.c:4065) __mlx5_ib_add (drivers/infiniband/hw/mlx5/main.c:4168) mlx5r_probe (drivers/infiniband/hw/mlx5/main.c:4402) ...

CVE ID	Description
CVE-2023-52854	In the Linux kernel, the following vulnerability has been resolved: padata: Fix refcnt handling in padata_free_shell() In a high-load arm64 environment, the pccrypt_aead01 test in LTP can lead to system UAF (Use-After-Free) issues. Due to the lengthy analysis of the pccrypt_aead01 function call, I'll describe the problem scenario using a simplified model: Suppose there's a user of padata named `user_function` that adheres to the padata requirement of calling `padata_free_shell` after `serial()` has been invoked, as demonstrated in the following code: ``c struct request { struct padata_priv padata; struct completion *done; }; void parallel(struct padata_priv *padata) { do_something(); } void serial(struct padata_priv *padata) { struct request *request = container_of(padata, struct request, padata); complete(request->done); } void user_function() { DECLARE_COMPLETION(done) padata->parallel = parallel; padata->serial = serial; padata_do_parallel(); wait_for_completion(&done); padata_free_shell(); } `` In the corresponding padata.c file, there's the following code: ``c static void padata_serial_worker(struct work_struct *serial_work) { ... cnt = 0; while (!list_empty(&local_list)) { ... padata->serial(padata); cnt++; } local_bh_enable(); if (refcount_sub_and_test(cnt, &pd->refcnt)) padata_free_pd(pd); } `` Because of the high system load and the accumulation of unexecuted softirq at this moment, `local_bh_enable()` in padata takes longer to execute than usual. Subsequently, when accessing `pd->refcnt`, `pd` has already been released by `padata_free_shell()`, resulting in a UAF issue with `pd->refcnt`. The fix is straightforward: add `refcount_dec_and_test` before calling `padata_free_pd` in `padata_free_shell`.
CVE-2023-52867	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: possible buffer overflow Buffer 'afmt_status' of size 6 could overflow, since index 'afmt_idx' is checked after access.
CVE-2023-52872	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: fix race condition in status line change on dead connections gsm_cleanup_mux() cleans up the gsm by closing all DLCIs, stopping all timers, removing the virtual tty devices and clearing the data queues. This procedure, however, may cause subsequent changes of the virtual modem status lines of a DLCI. More data is being added the outgoing data queue and the deleted kick timer is restarted to handle this. At this point many resources have already been removed by the cleanup procedure. Thus, a kernel panic occurs. Fix this by proving in gsm_modem_update() that the cleanup procedure has not been started and the mux is still alive. Note that writing to a virtual tty is already protected by checks against the DLCI specific connection state.
CVE-2023-52879	In the Linux kernel, the following vulnerability has been resolved: tracing: Have trace_event_file have ref counters The following can crash the kernel: # cd /sys/kernel/tracing # echo 'p:sched schedule' > kprobe_events # exec 5>>events/kprobes/sched/enable # > kprobe_events # exec 5>&- The above commands: 1. Change directory to the tracefs directory 2. Create a kprobe event (doesn't matter what one) 3. Open bash file descriptor 5 on the enable file of the kprobe event 4. Delete the kprobe event (removes the files too) 5. Close the bash file descriptor 5 The above causes a crash! BUG: kernel NULL pointer dereference, address: 0000000000000028 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 6 PID: 877 Comm: bash Not tainted 6.5.0-rc4-test-00008-g2c6b6b1029d4-dirty #186 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.2-debian-1.16.2-1 04/01/2014 RIP: 0010:tracing_release_file_tr+0xc/0x50 What happens here is that the kprobe event creates a trace_event_file "file" descriptor that represents the file in tracefs to the event. It maintains state of the event (is it enabled for the given instance?). Opening the "enable" file gets a reference to the event "file" descriptor via the open file descriptor. When the kprobe event is deleted, the file is also deleted from the tracefs system which also frees the event "file" descriptor. But as the tracefs file is still opened by user space, it will not be totally removed until the final dput() is called on it. But this is not true with the event "file" descriptor that is already freed. If the user does a write to or simply closes the file descriptor it will reference the event "file" descriptor that was just freed, causing a use-after-free bug. To solve this, add a ref count to the event "file" descriptor as well as a new flag called "FREED". The "file" will not be freed until the last reference is released. But the FREE flag will be set when the event is removed to prevent any more modifications to that event from happening, even if there's still a reference to the event "file" descriptor.
CVE-2023-52880	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: require CAP_NET_ADMIN to attach N_GSM0710 ldisc Any unprivileged user can attach N_GSM0710 ldisc, but it requires CAP_NET_ADMIN to create a GSM network anyway. Require initial namespace CAP_NET_ADMIN to do that.
CVE-2023-52882	In the Linux kernel, the following vulnerability has been resolved: clk: sunxi-ng: h6: Reparent CPUX during PLL CPUX rate change While PLL CPUX clock rate change when CPU is running from it works in vast majority of cases, now and then it causes instability. This leads to system crashes and other undefined behaviour. After a lot of testing (30+ hours) while also doing a lot of frequency switches, we can't observe any instability issues anymore when doing reparenting to stable clock like 24 MHz oscillator.
CVE-2023-52883	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix possible null pointer dereference abo->tbo.resource may be NULL in amdgpu_vm_bo_update.

CVE ID	Description
CVE-2023-52885	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix UAF in svc_tcp_listen_data_ready() After the listener svc_sock is freed, and before invoking svc_tcp_accept() for the established child sock, there is a window that the newsock retaining a freed listener svc_sock in sk_user_data which cloning from parent. In the race window, if data is received on the newsock, we will observe use-after-free report in svc_tcp_listen_data_ready(). Reproduce by two tasks: 1. while ;; do rpc.nfsd 0 ; rpc.nfsd; done 2. while ;; do echo "" ncat -4 127.0.0.1 2049 ; done KASAN report: ===== BUG: KASAN: slab-use-after-free in svc_tcp_listen_data_ready+0x1cf/0x1f0 [sunrpc] Read of size 8 at addr ffff888139d96228 by task nc/102553 CPU: 7 PID: 102553 Comm: nc Not tainted 6.3.0+ #18 Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 Call Trace: <IRQ> dump_stack_lvl+0x33/0x50 print_address_description.constprop.0+0x27/0x310 print_report+0x3e/0x70 kasan_report+0xae/0xe0 svc_tcp_listen_data_ready+0x1cf/0x1f0 [sunrpc] tcp_data_queue+0x9f4/0x20e0 tcp_rcv_established+0x666/0x1f60 tcp_v4_do_rcv+0x51c/0x850 tcp_v4_rcv+0x23fc/0x2e80 ip_protocol_deliver_rcu+0x62/0x300 ip_local_deliver_finish+0x267/0x350 ip_local_deliver+0x18b/0x2d0 ip_rcv+0x2fb/0x370 __netif_receive_skb_one_core+0x166/0x1b0 process_backlog+0x24c/0x5e0 __napi_poll+0xa2/0x500 net_rx_action+0x854/0xc90 __do_softirq+0x1bb/0x5de do_softirq+0xcb/0x100 </IRQ> <TASK> ... </TASK> Allocated by task 102371: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 __kasan_kmalloc+0x7b/0x90 svc_setup_socket+0x52/0x4f0 [sunrpc] svc_addsock+0x20d/0x400 [sunrpc] __write_ports_addfd+0x209/0x390 [nfsd] write_ports+0x239/0x2c0 [nfsd] nfscntl_transaction_write+0xac/0x110 [nfsd] vfs_write+0x1c3/0xae0 ksys_write+0xed/0x1c0 do_syscall_64+0x38/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc Freed by task 102551: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 kasan_save_free_info+0x2a/0x50 __kasan_slab_free+0x106/0x190 __kmem_cache_free+0x133/0x270 svc_xprt_free+0x1e2/0x350 [sunrpc] svc_xprt_destroy_all+0x25a/0x440 [sunrpc] nfsd_put+0x125/0x240 [nfsd] nfsd_svc+0x2cb/0x3c0 [nfsd] write_threads+0x1ac/0x2a0 [nfsd] nfscntl_transaction_write+0xac/0x110 [nfsd] vfs_write+0x1c3/0xae0 ksys_write+0xed/0x1c0 do_syscall_64+0x38/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc Fix the UAF by simply doing nothing in svc_tcp_listen_data_ready() if state != TCP_LISTEN, that will avoid dereferencing svsk for all child socket.
CVE-2023-52886	In the Linux kernel, the following vulnerability has been resolved: USB: core: Fix race by not overwriting udev->descriptor in hub_port_init() Syzbot reported an out-of-bounds read in sysfs.c:read_descriptors(): BUG: KASAN: slab-out-of-bounds in read_descriptors+0x263/0x280 drivers/usb/core/sysfs.c:883 Read of size 8 at addr ffff88801e78b8c8 by task udevd/5011 CPU: 0 PID: 5011 Comm: udevd Not tainted 6.4.0-rc6-syzkaller-00195-g40f71e7cd3c6 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/27/2023 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xd9/0x150 lib/dump_stack.c:106 print_address_description.constprop.0+0x2c/0x3c0 mm/kasan/report.c:351 print_report mm/kasan/report.c:462 [inline] kasan_report+0x11c/0x130 mm/kasan/report.c:572 read_descriptors+0x263/0x280 drivers/usb/core/sysfs.c:883 ... Allocated by task 758: ... __do_kmalloc_node mm/slab_common.c:966 [inline] __kmalloc+0x5e/0x190 mm/slab_common.c:979 kmalloc include/linux/slab.h:563 [inline] kzalloc include/linux/slab.h:680 [inline] usb_get_configuration+0x1f7/0x5170 drivers/usb/core/config.c:887 usb_enumerate_device drivers/usb/core/hub.c:2407 [inline] usb_new_device+0x12b0/0x19d0 drivers/usb/core/hub.c:2545 As analyzed by Khazhy Kumykov, the cause of this bug is a race between read_descriptors() and hub_port_init(): The first routine uses a field in udev->descriptor, not expecting it to change, while the second overwrites it. Prior to commit 45bf39f8df7f ("USB: core: Don't hold device lock while reading the "descriptors" sysfs file") this race couldn't occur, because the routines were mutually exclusive thanks to the device locking. Removing that locking from read_descriptors() exposed it to the race. The best way to fix the bug is to keep hub_port_init() from changing udev->descriptor once udev has been initialized and registered. Drivers expect the descriptors stored in the kernel to be immutable; we should not undermine this expectation. In fact, this change should have been made long ago. So now hub_port_init() will take an additional argument, specifying a buffer in which to store the device descriptor it reads. (If udev has not yet been initialized, the buffer pointer will be NULL and then hub_port_init() will store the device descriptor in udev as before.) This eliminates the data race responsible for the out-of-bounds read. The changes to hub_port_init() appear more extensive than they really are, because of indentation changes resulting from an attempt to avoid writing to other parts of the usb_device structure after it has been initialized. Similar changes should be made to the code that reads the BOS descriptor, but that can be handled in a separate patch later on. This patch is sufficient to fix the bug found by syzbot.
CVE-2023-52898	In the Linux kernel, the following vulnerability has been resolved: xhci: Fix null pointer dereference when host dies Make sure xhci_free_dev() and xhci_kill_endpoint_urbs() do not race and cause null pointer dereference when host suddenly dies. Usb core may call xhci_free_dev() which frees the xhci->devs[slot_id] virt device at the same time that xhci_kill_endpoint_urbs() tries to loop through all the device's endpoints, checking if there are any cancelled urbs left to give back. hold the xhci spinlock while freeing the virt device

CVE ID	Description
CVE-2023-52901	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Check endpoint is valid before dereferencing it When the host controller is not responding, all URBs queued to all endpoints need to be killed. This can cause a kernel panic if we dereference an invalid endpoint. Fix this by using xhci_get_virt_ep() helper to find the endpoint and checking if the endpoint is valid before dereferencing it. [233311.853271] xhci-hcd xhci-hcd.1.auto: xHCI host controller not responding, assume dead [233311.853393] Unable to handle kernel NULL pointer dereference at virtual address 00000000000000e8 [233311.853964] pc : xhci_hc_died+0x10c/0x270 [233311.853971] lr : xhci_hc_died+0x1ac/0x270 [233311.854077] Call trace: [233311.854085] xhci_hc_died+0x10c/0x270 [233311.854093] xhci_stop_endpoint_command_watchdog+0x100/0x1a4 [233311.854105] call_timer_fn+0x50/0x2d4 [233311.854112] expire_timers+0xac/0x2e4 [233311.854118] run_timer_softirq+0x300/0xabc [233311.854127] __do_softirq+0x148/0x528 [233311.854135] irq_exit+0x194/0x1a8 [233311.854143] __handle_domain_irq+0x164/0x1d0 [233311.854149] gic_handle_irq.22273+0x10c/0x188 [233311.854156] el1_irq+0xfc/0x1a8 [233311.854175] lpm_cpuidle_enter+0x25c/0x418 [msm_pm] [233311.854185] cpuidle_enter_state+0x1f0/0x764 [233311.854194] do_idle+0x594/0x6ac [233311.854201] cpu_startup_entry+0x7c/0x80 [233311.854209] secondary_start_kernel+0x170/0x198
CVE-2023-52906	In the Linux kernel, the following vulnerability has been resolved: net/sched: act_mpls: Fix warning during failed attribute validation The 'TCA_MPLS_LABEL' attribute is of 'NLA_U32' type, but has a validation type of 'NLA_VALIDATE_FUNCTION'. This is an invalid combination according to the comment above 'struct nla_policy': " Meaning of 'validate' field, use via NLA_POLICY_VALIDATE_FN: NLA_BINARY Validation function called for the attribute. All other Unused - but note that it's a union " This can trigger the warning [1] in nla_get_range_unsigned() when validation of the attribute fails. Despite being of 'NLA_U32' type, the associated 'min'/max' fields in the policy are negative as they are aliased by the 'validate' field. Fix by changing the attribute type to 'NLA_BINARY' which is consistent with the above comment and all other users of NLA_POLICY_VALIDATE_FN(). As a result, move the length validation to the validation function. No regressions in MPLS tests: # ./tdc.py -f tc-tests/actions/mpls.json [...] # echo \$? 0 [1] WARNING: CPU: 0 PID: 17743 at lib/nlattr.c:118 nla_get_range_unsigned+0x1d8/0x1e0 lib/nlattr.c:117 Modules linked in: CPU: 0 PID: 17743 Comm: syz-executor.0 Not tainted 6.1.0-rc8 #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-48-gd9c812dda519-prebuilt.qemu.org 04/01/2014 RIP: 0010:nla_get_range_unsigned+0x1d8/0x1e0 lib/nlattr.c:117 [...] Call Trace: <TASK> __netlink_policy_dump_write_attr+0x23d/0x990 net/netlink/policy.c:310 netlink_policy_dump_write_attr+0x22/0x30 net/netlink/policy.c:411 netlink_ack_tlv_fill net/netlink/af_netlink.c:2454 [inline] netlink_ack+0x546/0x760 net/netlink/af_netlink.c:2506 netlink_rcv_skb+0x1b7/0x240 net/netlink/af_netlink.c:2546 rtnetlink_rcv+0x18/0x20 net/core/rtnetlink.c:6109 netlink_unicast_kernel net/netlink/af_netlink.c:1319 [inline] netlink_unicast+0x5e9/0x6b0 net/netlink/af_netlink.c:1345 netlink_sendmsg+0x739/0x860 net/netlink/af_netlink.c:1921 sock_sendmsg_nosec net/socket.c:714 [inline] sock_sendmsg net/socket.c:734 [inline] ____sys_sendmsg+0x38f/0x500 net/socket.c:2482 __sys_sendmsg net/socket.c:2536 [inline] __sys_sendmsg+0x197/0x230 net/socket.c:2565 __do_sys_sendmsg net/socket.c:2574 [inline] __se_sys_sendmsg net/socket.c:2572 [inline] __x64_sys_sendmsg+0x42/0x50 net/socket.c:2572 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x2b/0x70 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2023-52908	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential NULL dereference Fix potential NULL dereference, in the case when "man", the resource manager might be NULL, when/if we print debug information.
CVE-2023-52910	In the Linux kernel, the following vulnerability has been resolved: iommu/iova: Fix alloc iova overflows issue In __alloc_and_insert_iova_range, there is an issue that retry_pfn overflows. The value of ioad->anchor.pfn_hi is ~0UL, then when ioad->cached_node is ioad->anchor, curr_iova->pfn_hi + 1 will overflow. As a result, if the retry logic is executed, low_pfn is updated to 0, and then new_pfn < low_pfn returns false to make the allocation successful. This issue occurs in the following two situations: 1. The first iova size exceeds the domain size. When initializing ioiva domain, ioad->cached_node is assigned as ioad->anchor. For example, the ioiva domain size is 10M, start_pfn is 0x1_F000_0000, and the ioiva size allocated for the first time is 11M. The following is the log information, new->pfn_lo is smaller than ioad->cached_node. Example log as follows: [223.798112][T1705487] sh: [name:iova&]__alloc_and_insert_iova_range start_pfn:0x1f0000, retry_pfn:0x0, size:0xb00, limit_pfn:0x1f0a00 [223.799590][T1705487] sh: [name:iova&]__alloc_and_insert_iova_range success start_pfn:0x1f0000, new->pfn_lo:0x1efe00, new->pfn_hi:0x1f08ff 2. The node with the largest ioiva->pfn_lo value in the ioiva domain is deleted, ioad->cached_node will be updated to ioad->anchor, and then the alloc ioiva size exceeds the maximum ioiva size that can be allocated in the domain. After judging that retry_pfn is less than limit_pfn, call retry_pfn +1 to fix the overflow issue.

CVE ID	Description
CVE-2023-52912	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fixed bug on error when unloading amdgpu Fixed bug on error when unloading amdgpu. The error message is as follows: [377.706202] kernel BUG at drivers/gpu/drm/drm_buddy.c:278! [377.706215] invalid opcode: 0000 [#1] PREEMPT SMP NOPTI [377.706222] CPU: 4 PID: 8610 Comm: modprobe Tainted: G IOE 6.0.0-thomas #1 [377.706231] Hardware name: ASUS System Product Name/PRIME Z390-A, BIOS 2004 11/02/2021 [377.706238] RIP: 0010:drm_buddy_free_block+0x26/0x30 [drm_buddy] [377.706264] Code: 00 00 00 90 0f 1f 44 00 00 48 8b 0e 89 c8 25 00 0c 00 00 3d 00 04 00 00 75 10 48 8b 47 18 48 d3 e0 48 01 47 28 e9 fa fe ff ff <0f> 0b 0f 1f 84 00 00 00 00 0f 1f 44 00 00 41 54 55 48 89 f5 53 [377.706282] RSP: 0018:ffffad2dc4683cb8 EFLAGS: 00010287 [377.706289] RAX: 0000000000000000 RBX: ffff8b1743bd5138 RCX: 0000000000000000 [377.706297] RDX: ffff8b1743bd5160 RSI: ffff8b1743bd5c78 RDI: ffff8b16d1b25f70 [377.706304] RBP: ffff8b1743bd59e0 R08: 0000000000000001 R09: 0000000000000001 [377.706311] R10: ffff8b16c8572400 R11: ffffad2dc4683cf0 R12: ffff8b16d1b25f70 [377.706318] R13: ffff8b16d1b25fd0 R14: ffff8b1743bd59c0 R15: ffff8b16d1b25f70 [377.706325] FS: 00007fec56c72c40(0000) GS: ffff8b1836500000(0000) knlGS:0000000000000000 [377.706334] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [377.706340] CR2: 00007f9b88c1ba50 CR3: 0000000110450004 CR4: 00000000003706e0 [377.706347] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [377.706354] DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 [377.706361] Call Trace: [377.706365] <TASK> [377.706369] drm_buddy_free_list+0x2a/0x60 [drm_buddy] [377.706376] amdgpu_vram_mgr_fini+0xea/0x180 [amdgpu] [377.706572] amdgpu_ttm_fini+0x12e/0x1a0 [amdgpu] [377.706650] amdgpu_bo_fini+0x22/0x90 [amdgpu] [377.706727] gmc_v11_0_sw_fini+0x26/0x30 [amdgpu] [377.706821] amdgpu_device_fini_sw+0xa1/0x3c0 [amdgpu] [377.706897] amdgpu_driver_release_kms+0x12/0x30 [amdgpu] [377.706975] drm_dev_release+0x20/0x40 [drm] [377.707006] release_nodes+0x35/0xb0 [377.707014] devres_release_all+0x8b/0xc0 [377.707020] device_unbind_cleanup+0xe/0x70 [377.707027] device_release_driver_internal+0xee/0x160 [377.707033] driver_detach+0x44/0x90 [377.707039] bus_remove_driver+0x55/0xe0 [377.707045] pci_unregister_driver+0x3b/0x90 [377.707052] amdgpu_exit+0x11/0x6c [amdgpu] [377.707194] __x64_sys_delete_module+0x142/0x2b0 [377.707201] ? fpregs_assert_state_consistent+0x22/0x50 [377.707208] ? exit_to_user_mode_prepare+0x3e/0x190 [377.707215] do_syscall_64+0x38/0x90 [377.707221] entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2023-52915	In the Linux kernel, the following vulnerability has been resolved: media: dvb-usb-v2: af9035: Fix null-ptr-deref in af9035_i2c_master_xfer In af9035_i2c_master_xfer, msg is controlled by user. When msg[i].buf is null and msg[i].len is zero, former checks on msg[i].buf would be passed. Malicious data finally reach af9035_i2c_master_xfer. If accessing msg[i].buf[0] without sanity check, null ptr deref would happen. We add check on msg[i].len to prevent crash. Similar commit: commit 0ed554fd769a ("media: dvb-usb: az6027: fix null-ptr-deref in az6027_i2c_xfer()")
CVE-2023-52918	In the Linux kernel, the following vulnerability has been resolved: media: pci: cx23885: check cx23885_vdev_init() return cx23885_vdev_init() can return a NULL pointer, but that pointer is used in the next line without a check. Add a NULL pointer check and go to the error unwind if it is NULL.
CVE-2023-52920	In the Linux kernel, the following vulnerability has been resolved: bpf: support non-r10 register spill/fill to/from stack in precision tracking Use instruction (jump) history to record instructions that performed register spill/fill to/from stack, regardless if this was done through read-only r10 register, or any other register after copying r10 into it *and* potentially adjusting offset. To make this work reliably, we push extra per-instruction flags into instruction history, encoding stack slot index (spi) and stack frame number in extra 10 bit flags we take away from prev_idx in instruction history. We don't touch idx field for maximum performance, as it's checked most frequently during backtracking. This change removes basically the last remaining practical limitation of precision backtracking logic in BPF verifier. It fixes known deficiencies, but also opens up new opportunities to reduce number of verified states, explored in the subsequent patches. There are only three differences in selftests' BPF object files according to veristat, all in the positive direction (less states). File Program Insns (A) Insns (B) Insns (DIFF) States (A) States (B) States (DIFF) ----- test_cls_redirect_dynptr.bpf.linked3.o cls_redirect 2987 2864 -123 (-4.12%) 240 231 -9 (-3.75%) xdp_synproxy_kern.bpf.linked3.o syncookie_tc 82848 82661 -187 (-0.23%) 5107 5073 -34 (-0.67%) xdp_synproxy_kern.bpf.linked3.o syncookie_xdp 85116 84964 -152 (-0.18%) 5162 5130 -32 (-0.62%) Note, I avoided renaming jmp_history to more generic insn_hist to minimize number of lines changed and potential merge conflicts between bpf and bpf-next trees. Notice also cur_hist_entry pointer reset to NULL at the beginning of instruction verification loop. This pointer avoids the problem of relying on last jump history entry's insn_idx to determine whether we already have entry for current instruction or not. It can happen that we added jump history entry because current instruction is_jmp_point(), but also we need to add instruction flags for stack access. In this case, we don't want to entries, so we need to reuse last added entry, if it is present. Relying on insn_idx comparison has the same ambiguity problem as the one that was fixed recently in [0], so we avoid that. [0] https://patchwork.kernel.org/project/netdevbpf/patch/20231110002638.4168352-3-andrii@kernel.org/

CVE ID	Description
CVE-2023-52921	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix possible UAF in amdgpu_cs_pass1() Since the gang_size check is outside of chunk parsing loop, we need to reset i before we free the chunk data. Suggested by Ye Zhang (@VAR10CK) of Baidu Security.
CVE-2023-52923	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: adapt set backend to use GC transaction API Use the GC transaction API to replace the old and buggy gc API and the busy mark approach. No set elements are removed from async garbage collection anymore, instead the _DEAD bit is set on so the set element is not visible from lookup path anymore. Async GC enqueues transaction work that might be aborted and retried later. rbtree and pipapo set backends does not set on the _DEAD bit from the sync GC path since this runs in control plane path where mutex is held. In this case, set elements are deactivated, removed and then released via RCU callback, sync GC never fails.
CVE-2023-52924	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: don't skip expired elements during walk There is an asymmetry between commit/abort and preparation phase if the following conditions are met: 1. set is a verdict map ("1.2.3.4 : jump foo") 2. timeouts are enabled In this case, following sequence is problematic: 1. element E in set S refers to chain C 2. userspace requests removal of set S 3. kernel does a set walk to decrement chain->use count for all elements from preparation phase 4. kernel does another set walk to remove elements from the commit phase (or another walk to do a chain->use increment for all elements from abort phase) If E has already expired in 1), it will be ignored during list walk, so its use count won't have been changed. Then, when set is culled, ->destroy callback will zap the element via nf_tables_set_elem_destroy(), but this function is only safe for elements that have been deactivated earlier from the preparation phase: lack of earlier deactivate removes the element but leaks the chain use count, which results in a WARN splat when the chain gets removed later, plus a leak of the nft_chain structure. Update pipapo_get() not to skip expired elements, otherwise flush command reports bogus ENOENT errors.
CVE-2023-52925	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: don't fail inserts if duplicate has expired nftables selftests fail: run-tests.sh testcases/sets/0044interval_overlap_0 Expected: 0-2 . 0-3, got: W: [FAILED] ./testcases/sets/0044interval_overlap_0: got 1 Insertion must ignore duplicate but expired entries. Moreover, there is a strange asymmetry in nft_pipapo_activate: It refetches the current element, whereas the other ->activate callbacks (bitmap, hash, rhash, rbtree) use elem->priv. Same for .remove: other set implementations take elem->priv, nft_pipapo_remove fetches elem->priv, then does a relookup, remove this. I suspect this was the reason for the change that prompted the removal of the expired check in pipapo_get() in the first place, but skipping expired elements there makes no sense to me, this helper is used for normal get requests, insertions (duplicate check) and deactivate callback. In first two cases expired elements must be skipped. For ->deactivate(), this gets called for DELSETELEM, so it seems to me that expired elements should be skipped as well, i.e. delete request should fail with - ENOENT error.
CVE-2023-52927	In the Linux kernel, the following vulnerability has been resolved: netfilter: allow exp not to be removed in nf_ct_find_expectation Currently nf_conntrack_in() calling nf_ct_find_expectation() will remove the exp from the hash table. However, in some scenario, we expect the exp not to be removed when the created ct will not be confirmed, like in OVS and TC conntrack in the following patches. This patch allows exp not to be removed by setting IPS_CONFIRMED in the status of the tpl.
CVE-2023-52928	In the Linux kernel, the following vulnerability has been resolved: bpf: Skip invalid kfunc call in backtrack_insn The verifier skips invalid kfunc call in check_kfunc_call(), which would be captured in fixup_kfunc_call() if such insn is not eliminated by dead code elimination. However, this can lead to the following warning in backtrack_insn(), also see [1]: ----- [cut here]----- verifier backtracking bug WARNING: CPU: 6 PID: 8646 at kernel/bpf/verifier.c:2756 backtrack_insn kernel/bpf/verifier.c:2756 __mark_chain_precision kernel/bpf/verifier.c:3065 mark_chain_precision kernel/bpf/verifier.c:3165 adjust_reg_min_max_vals kernel/bpf/verifier.c:10715 check_alu_op kernel/bpf/verifier.c:10928 do_check kernel/bpf/verifier.c:13821 [inline] do_check_common kernel/bpf/verifier.c:16289 [...] So make backtracking conservative with this by returning ENOTSUPP. [1] https://lore.kernel.org/bpf/CACkBJsaXNceR8ZjkLG=dT3P=4A8SBsg0Z5h5PWLryF5=ghKq-g@mail.gmail.com/
CVE-2023-52931	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Avoid potential vm use-after-free Adding the vm to the vm_xa table makes it visible to userspace, which could try to race with us to close the vm. So we need to take our extra reference before putting it in the table. (cherry picked from commit 99343c46d4e2b34c285d3d5f68ff04274c2f9fb4)
CVE-2023-52932	In the Linux kernel, the following vulnerability has been resolved: mm/swapfile: add cond_resched() in get_swap_pages() The softlockup still occurs in get_swap_pages() under memory pressure. 64 CPU cores, 64GB memory, and 28 zram devices, the disksize of each zram device is 50MB with same priority as si. Use the stress-ng tool to increase memory pressure, causing the system to oom frequently. The plist_for_each_entry_safe() loops in get_swap_pages() could reach tens of thousands of times to find available space (extreme case: cond_resched() is not called in scan_swap_map_slots()). Let's add cond_resched() into get_swap_pages() when failed to find available space to avoid softlockup.

CVE ID	Description
CVE-2023-52935	In the Linux kernel, the following vulnerability has been resolved: mm/khugepaged: fix ->anon_vma race If an ->anon_vma is attached to the VMA, collapse_and_free_pmd() requires it to be locked. Page table traversal is allowed under any one of the mmap lock, the anon_vma lock (if the VMA is associated with an anon_vma), and the mapping lock (if the VMA is associated with a mapping); and so to be able to remove page tables, we must hold all three of them. retract_page_tables() bails out if an ->anon_vma is attached, but does this check before holding the mmap lock (as the comment above the check explains). If we racily merged an existing ->anon_vma (shared with a child process) from a neighboring VMA, subsequent rmap traversals on pages belonging to the child will be able to see the page tables that we are concurrently removing while assuming that nothing else can access them. Repeat the ->anon_vma check once we hold the mmap lock to ensure that there really is no concurrent page table access. Hitting this bug causes a lockdep warning in collapse_and_free_pmd(), in the line "lockdep_assert_held_write(&vma->anon_vma->root->rwsem)". It can also lead to use-after-free access.
CVE-2023-52936	In the Linux kernel, the following vulnerability has been resolved: kernel/irq/irqdomain.c: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-52937	In the Linux kernel, the following vulnerability has been resolved: HV: hv_balloon: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-52938	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Don't attempt to resume the ports before they exist This will fix null pointer dereference that was caused by the driver attempting to resume ports that were not yet registered.
CVE-2023-52939	In the Linux kernel, the following vulnerability has been resolved: mm: memcg: fix NULL pointer in mem_cgroup_track_foreign_dirty_slowpath() As commit 18365225f044 ("hwpoison, memcg: forcibly uncharge LRU pages"), hwpoison will forcibly uncharge a LRU hwpoisoned page, the folio_memcg could be NULL, then, mem_cgroup_track_foreign_dirty_slowpath() could occurs a NULL pointer dereference, let's do not record the foreign writebacks for folio memcg is null in mem_cgroup_track_foreign_dirty() to fix it.
CVE-2023-52942	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: Fix wrong check in update_parent_subparts_cpumask() It was found that the check to see if a partition could use up all the cpus from the parent cpuset in update_parent_subparts_cpumask() was incorrect. As a result, it is possible to leave parent with no effective cpu left even if there are tasks in the parent cpuset. This can lead to system panic as reported in [1]. Fix this problem by updating the check to fail the enabling the partition if parent's effective_cpus is a subset of the child's cpus_allowed. Also record the error code when an error happens in update_prstate() and add a test case where parent partition and child have the same cpu list and parent has task. Enabling partition in the child will fail in this case. [1] https://www.spinics.net/lists/cgroups/msg36254.html

CVE ID	Description
CVE-2023-52973	In the Linux kernel, the following vulnerability has been resolved: vc_screen: move load of struct vc_data pointer in vcs_read() to avoid UAF After a call to console_unlock() in vcs_read() the vc_data struct can be freed by vc_deallocate(). Because of that, the struct vc_data pointer load must be done at the top of while loop in vcs_read() to avoid a UAF when vcs_size() is called. Syzkaller reported a UAF in vcs_size(). BUG: KASAN: use-after-free in vcs_size (drivers/tty/vt/vc_screen.c:215) Read of size 4 at addr ffff8881137479a8 by task 4a005ed81e27e65/1537 CPU: 0 PID: 1537 Comm: 4a005ed81e27e65 Not tainted 6.2.0-rc5 #1 Hardware name: Red Hat KVM, BIOS 1.15.0-2.module Call Trace: <TASK> __asan_report_load4_noabort (mm/kasan/report_generic.c:350) vcs_size (drivers/tty/vt/vc_screen.c:215) vcs_read (drivers/tty/vt/vc_screen.c:415) vfs_read (fs/read_write.c:468 fs/read_write.c:450) ... </TASK> Allocated by task 1191: ... kmalloc_trace (mm/slab_common.c:1069) vc_allocate (/include/linux/slab.h:580 ./include/linux/slab.h:720 drivers/tty/vt/vt.c:1128 drivers/tty/vt/vt.c:1108) con_install (drivers/tty/vt/vt.c:3383) tty_init_dev (drivers/tty/tty_io.c:1301 drivers/tty/tty_io.c:1413 drivers/tty/tty_io.c:1390) tty_open (drivers/tty/tty_io.c:2080 drivers/tty/tty_io.c:2126) chrdev_open (fs/char_dev.c:415) do_dentry_open (fs/open.c:883) vfs_open (fs/open.c:1014) ... Freed by task 1548: ... kfree (mm/slab_common.c:1021) vc_port_destruct (drivers/tty/vt/vt.c:1094) tty_port_destructor (drivers/tty/tty_port.c:296) tty_port_put (drivers/tty/tty_port.c:312) vt_disallocate_all (drivers/tty/vt/vt_ioctl.c:662 (discriminator 2)) vt_ioctl (drivers/tty/vt/vt_ioctl.c:903) tty_ioctl (drivers/tty/tty_io.c:2776) ... The buggy address belongs to the object at ffff888113747800 which belongs to the cache kmalloc-1k of size 1024 The buggy address is located 424 bytes inside of 1024-byte region [ffff888113747800, ffff888113747c00) The buggy address belongs to the physical page: page:00000000b3fe6c7c refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x113740 head:00000000b3fe6c7c order:3 compound_mapcount:0 subpages_mapcount:0 compound_pincount:0 anon flags: 0x17ffffc0010200(slab head node=0 zone=2 lastcpupid=0x1fffff) raw: 0017ffffc0010200 ffff888100042dc0 0000000000000000 dead000000000001 raw: 0000000000000000 0000000000100010 00000001fffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff888113747880: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ffff888113747900: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb > ffff888113747980: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ^ ffff888113747a00: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ffff888113747a80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ===== Disabling lock debugging due to kernel taint
CVE-2023-52974	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi_tcp: Fix UAF during login when accessing the shost ipaddress If during iscsi_sw_tcp_session_create() iscsi_tcp_r2tpool_alloc() fails, userspace could be accessing the host's ipaddress attr. If we then free the session via iscsi_session_tearardown() while userspace is still accessing the session we will hit a use after free bug. Set the tcp_sw_host->session after we have completed session creation and can no longer fail.

CVE ID	Description
CVE-2023-52975	In the Linux kernel, the following vulnerability has been resolved: scsi: iscsi_tcp: Fix UAF during logout when accessing the shost ipaddress Bug report and analysis from Ding Hui. During iSCSI session logout, if another task accesses the shost ipaddress attr, we can get a KASAN UAF report like this: [276.942144] BUG: KASAN: use-after-free in _raw_spin_lock_bh+0x78/0xe0 [276.942535] Write of size 4 at addr ffff8881053b45b8 by task cat/4088 [276.943511] CPU: 2 PID: 4088 Comm: cat Tainted: G E 6.1.0-rc8+ #3 [276.943997] Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 [276.944470] Call Trace: [276.944943] <TASK> [276.945397] dump_stack_lvl+0x34/0x48 [276.945887] print_address_description.constprop.0+0x86/0x1e7 [276.946421] print_report+0x36/0x4f [276.947358] kasan_report+0xad/0x130 [276.948234] kasan_check_range+0x35/0x1c0 [276.948674] _raw_spin_lock_bh+0x78/0xe0 [276.949989] iscsi_sw_tcp_host_get_param+0xad/0x2e0 [iscsi_tcp] [276.951765] show_host_param_ISCSI_HOST_PARAM_IPADDRESS+0xe9/0x130 [scsi_transport_iscsi] [276.952185] dev_attr_show+0x3f/0x80 [276.953005] sysfs_kf_seq_show+0x1fb/0x3e0 [276.953401] seq_read_iter+0x402/0x1020 [276.954260] vfs_read+0x532/0x7b0 [276.955113] ksys_read+0xed/0x1c0 [276.955952] do_syscall_64+0x38/0x90 [276.956347] entry_SYSCALL_64_after_hwframe+0x63/0xcd [276.956769] RIP: 0033:0x7f5d3a679222 [276.957161] Code: c0 e9 b2 fe ff ff 50 48 8d 3d 32 c0 0b 00 e8 a5 fe 01 00 0f 1f 44 00 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 0f 05 <48> 3d 00 f0 ff ff 77 56 c3 0f 1f 44 00 00 48 83 ec 28 48 89 54 24 [276.958009] RSP: 002b:00007ffc864d16a8 EFLAGS: 00000246 ORIG_RAX: 0000000000000000 [276.958431] RAX: ffffffff8da RBX: 0000000000200000 RCX: 00007f5d3a679222 [276.958857] RDX: 0000000000020000 RSI: 00007f5d3a4fe000 RDI: 0000000000000003 [276.959281] RBP: 00007f5d3a4fe000 R08: 00000000ffffffff R09: 0000000000000000 [276.959682] R10: 0000000000000022 R11: 0000000000000246 R12: 0000000000020000 [276.960126] R13: 0000000000000003 R14: 0000000000000000 R15: 0000557a26dada58 [276.960536] </TASK> [276.961357] Allocated by task 2209: [276.961756] kasan_save_stack+0x1e/0x40 [276.962170] kasan_set_track+0x21/0x30 [276.962557] __kasan_kmalloc+0x7e/0x90 [276.962923] __kmalloc+0x5b/0x140 [276.963308] iscsi_alloc_session+0x28/0x840 [scsi_transport_iscsi] [276.963712] iscsi_session_setup+0xda/0xba0 [libiscsi] [276.964078] iscsi_sw_tcp_session_create+0x1fd/0x330 [iscsi_tcp] [276.964431] iscsi_if_create_session.isra.0+0x50/0x260 [scsi_transport_iscsi] [276.964793] iscsi_if_recv_msg+0xc5a/0x2660 [scsi_transport_iscsi] [276.965153] iscsi_if_rx+0x198/0x4b0 [scsi_transport_iscsi] [276.965546] netlink_unicast+0x4d5/0x7b0 [276.965905] netlink_sendmsg+0x78d/0xc30 [276.966236] sock_sendmsg+0xe5/0x120 [276.966576] ____sys_sendmsg+0x5fe/0x860 [276.966923] ____sys_sendmsg+0xe0/0x170 [276.967300] ____sys_sendmsg+0xc8/0x170 [276.967666] do_syscall_64+0x38/0x90 [276.968028] entry_SYSCALL_64_after_hwframe+0x63/0xcd [276.968773] Freed by task 2209: [276.969111] kasan_save_stack+0x1e/0x40 [276.969449] kasan_set_track+0x21/0x30 [276.969789] kasan_save_free_info+0x2a/0x50 [276.970146] __kasan_slab_free+0x106/0x190 [276.970470] __kmem_cache_free+0x133/0x270 [276.970816] device_release+0x98/0x210 [276.971145] kobject_cleanup+0x101/0x360 [276.971462] iscsi_session_teardown+0x3fb/0x530 [libiscsi] [276.971775] iscsi_sw_tcp_session_destroy+0xd8/0x130 [iscsi_tcp] [276.972143] iscsi_if_recv_msg+0x1bf1/0x2660 [scsi_transport_iscsi] [276.972485] iscsi_if_rx+0x198/0x4b0 [scsi_transport_iscsi] [276.972808] netlink_unicast+0x4d5/0x7b0 [276.973201] netlink_sendmsg+0x78d/0xc30 [276.973544] sock_sendmsg+0xe5/0x120 [276.973864] ____sys_sendmsg+0x5fe/0x860 [276.974248] ____sys_---truncated---
CVE-2023-52976	In the Linux kernel, the following vulnerability has been resolved: efi: fix potential NULL deref in efi_mem_reserve_persistent When iterating on a linked list, a result of memremap is dereferenced without checking it for NULL. This patch adds a check that falls back on allocating a new page in case memremap doesn't succeed. Found by Linux Verification Center (linuxtesting.org) with SVACE. [ardb: return -ENOMEM instead of breaking out of the loop]
CVE-2023-52983	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix uaf for bfqq in bic_set_bfqq() After commit 64dc8c732f5c ("block, bfq: fix possible uaf for 'bfqq->bic'",) bic->bfqq will be accessed in bic_set_bfqq(), however, in some context bic->bfqq will be freed, and bic_set_bfqq() is called with the freed bic->bfqq. Fix the problem by always freeing bfqq after bic_set_bfqq().
CVE-2023-52984	In the Linux kernel, the following vulnerability has been resolved: net: phy: dp83822: Fix null pointer access on DP83825/DP83826 devices The probe() function is only used for the DP83822 PHY, leaving the private data pointer uninitialized for the smaller DP83825/26 models. While all uses of the private data structure are hidden in 82822 specific callbacks, configuring the interrupt is shared across all models. This causes a NULL pointer dereference on the smaller PHYs as it accesses the private data unchecked. Verifying the pointer avoids that.

CVE ID	Description
CVE-2023-52986	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Check for any of tcp_bpf_protos when cloning a listener A listening socket linked to a sockmap has its sk_prot overridden. It points to one of the struct proto variants in tcp_bpf_protos. The variant depends on the socket's family and which sockmap programs are attached. A child socket cloned from a TCP listener initially inherits their sk_prot. But before cloning is finished, we restore the child's proto to the listener's original non-tcp_bpf_protos one. This happens in tcp_create_openreq_child -> tcp_bpf_clone. Today, in tcp_bpf_clone we detect if the child's proto should be restored by checking only for the TCP_BPF_BASE proto variant. This is not correct. The sk_prot of listening socket linked to a sockmap can point to to any variant in tcp_bpf_protos. If the listeners sk_prot happens to be not the TCP_BPF_BASE variant, then the child socket unintentionally is left if the inherited sk_prot by tcp_bpf_clone. This leads to issues like infinite recursion on close [1], because the child state is otherwise not set up for use with tcp_bpf_prot operations. Adjust the check in tcp_bpf_clone to detect all of tcp_bpf_protos variants. Note that it wouldn't be sufficient to check the socket state when overriding the sk_prot in tcp_bpf_update_proto in order to always use the TCP_BPF_BASE variant for listening sockets. Since commit b8b8315e39ff ("bpf, sockmap: Remove unhash handler for BPF sockmap usage") it is possible for a socket to transition to TCP_LISTEN state while already linked to a sockmap, e.g. connect() -> insert into map -> connect(AF_UNSPEC) -> listen(). [1]: https://lore.kernel.org/all/00000000000073b14905ef2e7401@google.com/
CVE-2023-52987	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-mtrace: prevent underflow in sof_ipc4_priority_mask_dfs_write() The "id" comes from the user. Change the type to unsigned to prevent an array underflow.
CVE-2023-52988	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda/via: Avoid potential array out-of-bound in add_secret_dac_path() snd_hda_get_connections() can return a negative error code. It may lead to accessing 'conn' array at a negative index. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-52989	In the Linux kernel, the following vulnerability has been resolved: firewire: fix memory leak for payload of request subaction to IEC 61883-1 FCP region This patch is fix for Linux kernel v2.6.33 or later. For request subaction to IEC 61883-1 FCP region, Linux FireWire subsystem have had an issue of use-after-free. The subsystem allows multiple user space listeners to the region, while data of the payload was likely released before the listeners execute read(2) to access to it for copying to user space. The issue was fixed by a commit 281e20323ab7 ("firewire: core: fix use-after-free regression in FCP handler"). The object of payload is duplicated in kernel space for each listener. When the listener executes ioctl(2) with FW_CDEV_IOC_SEND_RESPONSE request, the object is going to be released. However, it causes memory leak since the commit relies on call of release_request() in drivers/firewire/core-cdev.c. Against the expectation, the function is never called due to the design of release_client_resource(). The function delegates release task to caller when called with non-NULL fourth argument. The implementation of ioctl_send_response() is the case. It should release the object explicitly. This commit fixes the bug.
CVE-2023-52992	In the Linux kernel, the following vulnerability has been resolved: bpf: Skip task with pid=1 in send_signal_common() The following kernel panic can be triggered when a task with pid=1 attaches a prog that attempts to send killing signal to itself, also see [1] for more details: Kernel panic - not syncing: Attempted to kill init! exitcode=0x0000000b CPU: 3 PID: 1 Comm: systemd Not tainted 6.1.0-09652-g59fe41b5255f #148 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x100/0x178 lib/dump_stack.c:106 panic+0x2c4/0x60f kernel/panic.c:275 do_exit.cold+0x63/0xe4 kernel/exit.c:789 do_group_exit+0xd4/0x2a0 kernel/exit.c:950 get_signal+0x2460/0x2600 kernel/signal.c:2858 arch_do_signal_or_restart+0x78/0x5d0 arch/x86/kernel/signal.c:306 exit_to_user_mode_loop kernel/entry/common.c:168 [inline] exit_to_user_mode_prepare+0x15f/0x250 kernel/entry/common.c:203 __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline] syscall_exit_to_user_mode+0x1d/0x50 kernel/entry/common.c:296 do_syscall_64+0x44/0xb0 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x63/0xcd So skip task with pid=1 in bpf_send_signal_common() to avoid the panic. [1] https://lore.kernel.org/bpf/20221222043507.33037-1-sunhao.th@gmail.com
CVE-2023-52993	In the Linux kernel, the following vulnerability has been resolved: x86/i8259: Mark legacy PIC interrupts with IRQ_LEVEL Baoquan reported that after triggering a crash the subsequent crash-kernel fails to boot about half of the time. It triggers a NULL pointer dereference in the periodic tick code. This happens because the legacy timer interrupt (IRQ0) is resent in software which happens in soft interrupt (tasklet) context. In this context get_irq_regs() returns NULL which leads to the NULL pointer dereference. The reason for the resend is a spurious APIC interrupt on the IRQ0 vector which is captured and leads to a resend when the legacy timer interrupt is enabled. This is wrong because the legacy PIC interrupts are level triggered and therefore should never be resent in software, but nothing ever sets the IRQ_LEVEL flag on those interrupts, so the core code does not know about their trigger type. Ensure that IRQ_LEVEL is set when the legacy PCI interrupts are set up.

CVE ID	Description
CVE-2023-52994	In the Linux kernel, the following vulnerability has been resolved: acpi: Fix suspend with Xen PV Commit f1e525009493 ("x86/boot: Skip realmode init code when running as Xen PV guest") missed one code path accessing real_mode_header, leading to dereferencing NULL when suspending the system under Xen: [348.284004] PM: suspend entry (deep) [348.289532] Filesystems sync: 0.005 seconds [348.291545] Freezing user space processes ... (elapsed 0.000 seconds) done. [348.292457] OOM killer disabled. [348.292462] Freezing remaining freezable tasks ... (elapsed 0.104 seconds) done. [348.396612] printk: Suspending console(s) (use no_console_suspend to debug) [348.749228] PM: suspend devices took 0.352 seconds [348.769713] ACPI: EC: interrupt blocked [348.816077] BUG: kernel NULL pointer dereference, address: 000000000000001c [348.816080] #PF: supervisor read access in kernel mode [348.816081] #PF: error_code(0x0000) - not-present page [348.816083] PGD 0 P4D 0 [348.816086] Oops: 0000 [#1] PREEMPT SMP NOPTI [348.816089] CPU: 0 PID: 6764 Comm: systemd-sleep Not tainted 6.1.3-1.fc32.qubes.x86_64 #1 [348.816092] Hardware name: Star Labs StarBook/StarBook, BIOS 8.01 07/03/2022 [348.816093] RIP: e030:acpi_get_wakeup_address +0xc/0x20 Fix that by adding an optional acpi callback allowing to skip setting the wakeup address, as in the Xen PV case this will be handled by the hypervisor anyway.
CVE-2023-52996	In the Linux kernel, the following vulnerability has been resolved: ipv4: prevent potential spectre v1 gadget in fib_metrics_match() if (!type) continue; if (type > RTAX_MAX) return false; ... fi_val = fi->fib_metrics->metrics[type - 1]; @type being used as an array index, we need to prevent cpu speculation or risk leaking kernel memory content.
CVE-2023-52997	In the Linux kernel, the following vulnerability has been resolved: ipv4: prevent potential spectre v1 gadget in ip_metrics_convert() if (!type) continue; if (type > RTAX_MAX) return -EINVAL; ... metrics[type - 1] = val; @type being used as an array index, we need to prevent cpu speculation or risk leaking kernel memory content.
CVE-2023-53000	In the Linux kernel, the following vulnerability has been resolved: netlink: prevent potential spectre v1 gadgets Most netlink attributes are parsed and validated from __nla_validate_parse() or validate_nla() u16 type = nla_type(nla); if (type == 0 type > maxtype) { /* error or continue */ } @type is then used as an array index and can be used as a Spectre v1 gadget. array_index_nospec() can be used to prevent leaking content of kernel memory to malicious users. This should take care of vast majority of netlink uses, but an audit is needed to take care of others where validation is not yet centralized in core netlink functions.
CVE-2023-53005	In the Linux kernel, the following vulnerability has been resolved: trace_events_hist: add check for return value of 'create_hist_field' Function 'create_hist_field' is called recursively at trace_events_hist.c:1954 and can return NULL-value that's why we have to check it to avoid null pointer dereference. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-53006	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix oops due to uncleaned server->smbd_conn in reconnect In smbd_destroy(), clear the server->smbd_conn pointer after freeing the smbd_connection struct that it points to so that reconnection doesn't get confused.
CVE-2023-53007	In the Linux kernel, the following vulnerability has been resolved: tracing: Make sure trace_printk() can output as soon as it can be used Currently trace_printk() can be used as soon as early_trace_init() is called from start_kernel(). But if a crash happens, and "ftrace_dump_on_oops" is set on the kernel command line, all you get will be: [0.456075] <idle>-0 0dN.2. 347519us : Unknown type 6 [0.456075] <idle>-0 0dN.2. 353141us : Unknown type 6 [0.456075] <idle>-0 0dN.2. 358684us : Unknown type 6 This is because the trace_printk() event (type 6) hasn't been registered yet. That gets done via an early_initcall(), which may be early, but not early enough. Instead of registering the trace_printk() event (and other ftrace events, which are not trace events) via an early_initcall(), have them registered at the same time that trace_printk() can be used. This way, if there is a crash before early_initcall(), then the trace_printk(s) will actually be useful.
CVE-2023-53008	In the Linux kernel, the following vulnerability has been resolved: cifs: fix potential memory leaks in session setup Make sure to free cifs_ses::auth_key.response before allocating it as we might end up leaking memory in reconnect or mounting.
CVE-2023-53009	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Add sync after creating vram bo There will be data corruption on vram allocated by svm if the initialization is not complete and application is writing on the memory. Adding sync to wait for the initialization completion is to resolve this issue.

CVE ID	Description
CVE-2023-53011	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: enable all safety features by default In the original implementation of dwmac5 commit 8bf993a5877e ("net: stmmac: Add support for DWMAC5 and implement Safety Features") all safety features were enabled by default. Later it seems some implementations didn't have support for all the features, so in commit 5ac712dcddef ("net: stmmac: enable platform specific safety features") the safety_feat_cfg structure was added to the callback and defined for some platforms to selectively enable these safety features. The problem is that only certain platforms were given that software support. If the automotive safety package bit is set in the hardware features register the safety feature callback is called for the platform, and for platforms that didn't get a safety_feat_cfg defined this results in the following NULL pointer dereference: [7.933303] Call trace: [7.935812] dwmac5_safety_feat_config+0x20/0x170 [stmmac] [7.941455] __stmmac_open+0x16c/0x474 [stmmac] [7.946117] stmmac_open+0x38/0x70 [stmmac] [7.950414] __dev_open+0x100/0x1dc [7.954006] __dev_change_flags+0x18c/0x204 [7.958297] dev_change_flags+0x24/0x6c [7.962237] do_setlink+0x2b8/0xfa4 [7.965827] __rtnl_newlink+0x4ec/0x840 [7.969766] rtnl_newlink+0x50/0x80 [7.973353] rtnetlink_rcv_msg+0x12c/0x374 [7.977557] netlink_rcv_skb+0x5c/0x130 [7.981500] rtnetlink_rcv+0x18/0x2c [7.985172] netlink_unicast+0x2e8/0x340 [7.989197] netlink_sendmsg+0x1a8/0x420 [7.993222] ____sys_sendmsg+0x218/0x280 [7.997249] ___sys_sendmsg+0xac/0x100 [8.001103] __sys_sendmsg+0x84/0xe0 [8.004776] __arm64_sys_sendmsg+0x24/0x30 [8.008983] invoke_syscall+0x48/0x114 [8.012840] el0_svc_common.constprop.0+0xcc/0xec [8.017665] do_el0_svc+0x38/0xb0 [8.021071] el0_svc+0x2c/0x84 [8.024212] el0t_64_sync_handler+0xf4/0x120 [8.028598] el0t_64_sync+0x190/0x194 Go back to the original behavior, if the automotive safety package is found to be supported in hardware enable all the features unless safety_feat_cfg is passed in saying this particular platform only supports a subset of the features.
CVE-2023-53013	In the Linux kernel, the following vulnerability has been resolved: ptdma: pt_core_execute_cmd() should use spinlock The interrupt handler (pt_core_irq_handler()) of the ptdma driver can be called from interrupt context. The code flow in this function can lead down to pt_core_execute_cmd() which will attempt to grab a mutex, which is not appropriate in interrupt context and ultimately leads to a kernel panic. The fix here changes this mutex to a spinlock, which has been verified to resolve the issue.
CVE-2023-53015	In the Linux kernel, the following vulnerability has been resolved: HID: betop: check shape of output reports betopff_init() only checks the total sum of the report counts for each report field to be at least 4, but hid_betopff_play() expects 4 report fields. A device advertising an output report with one field and 4 report counts would pass the check but crash the kernel with a NULL pointer dereference in hid_betopff_play().
CVE-2023-53016	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix possible deadlock in rfcomm_sk_state_change syzbot reports a possible deadlock in rfcomm_sk_state_change [1]. While rfcomm_sock_connect acquires the sk lock and waits for the rfcomm lock, rfcomm_sock_release could have the rfcomm lock and hit a deadlock for acquiring the sk lock. Here's a simplified flow: rfcomm_sock_connect: lock_sock(sk) rfcomm_dlc_open: rfcomm_lock() rfcomm_sock_release: rfcomm_sock_shutdown: rfcomm_lock() __rfcomm_dlc_close: rfcomm_k_state_change: lock_sock(sk) This patch drops the sk lock before calling rfcomm_dlc_open to avoid the possible deadlock and holds sk's reference count to prevent use-after-free after rfcomm_dlc_open completes.
CVE-2023-53019	In the Linux kernel, the following vulnerability has been resolved: net: mdio: validate parameter addr in mdiobus_get_phy() The caller may pass any value as addr, what may result in an out-of-bounds access to array mdio_map. One existing case is stmmac_init_phy() that may pass -1 as addr. Therefore validate addr before using it.
CVE-2023-53020	In the Linux kernel, the following vulnerability has been resolved: l2tp: close all race conditions in l2tp_tunnel_register() The code in l2tp_tunnel_register() is racy in several ways: 1. It modifies the tunnel socket _after_ publishing it. 2. It calls setup_udp_tunnel_sock() on an existing socket without locking. 3. It changes sock lock class on fly, which triggers many syzbot reports. This patch amends all of them by moving socket initialization code before publishing and under sock lock. As suggested by Jakub, the l2tp lockdep class is not necessary as we can just switch to bh_lock_sock_nested().

CVE ID	Description
CVE-2023-53024	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix pointer-leak due to insufficient speculative store bypass mitigation To mitigate Spectre v4, 2039f26f3aca ("bpf: Fix leakage due to insufficient speculative store bypass mitigation") inserts lfence instructions after 1) initializing a stack slot and 2) spilling a pointer to the stack. However, this does not cover cases where a stack slot is first initialized with a pointer (subject to sanitization) but then overwritten with a scalar (not subject to sanitization because the slot was already initialized). In this case, the second write may be subject to speculative store bypass (SSB) creating a speculative pointer-as-scalar type confusion. This allows the program to subsequently leak the numerical pointer value using, for example, a branch-based cache side channel. To fix this, also sanitize scalars if they write a stack slot that previously contained a pointer. Assuming that pointer-spills are only generated by LLVM on register-pressure, the performance impact on most real-world BPF programs should be small. The following unprivileged BPF bytecode drafts a minimal exploit and the mitigation: [...] // r6 = 0 or 1 (skalar, unknown user input) // r7 = accessible ptr for side channel // r10 = frame pointer (fp), to be leaked // r9 = r10 # fp alias to encourage ssb *(u64 *) (r9 - 8) = r10 // fp[-8] = ptr, to be leaked // lfence added here because of pointer spill to stack. // Omitted: Dummy bpf_ringbuf_output() here to train alias predictor // for no r9-r10 dependency. // *(u64 *) (r10 - 8) = r6 // fp[-8] = scalar, overwrites ptr // 2039f26f3aca: no lfence added because stack slot was not STACK_INVALID, // store may be subject to SSB // fix: also add an lfence when the slot contained a ptr // r8 = *(u64 *) (r9 - 8) // r8 = architecturally a scalar, speculatively a ptr // leak ptr using branch-based cache side channel: r8 &= 1 // choose bit to leak if r8 == 0 goto SLOW // no mispredict // architecturally dead code if input r6 is 0, // only executes speculatively iff ptr bit is 1 r8 = *(u64 *) (r7 + 0) # encode bit in cache (0: slow, 1: fast) SLOW: [...] After running this, the program can time the access to *(r7 + 0) to determine whether the chosen pointer bit was 0 or 1. Repeat this 64 times to recover the whole address on amd64. In summary, sanitization can only be skipped if one scalar is overwritten with another scalar. Scalar-confusion due to speculative store bypass can not lead to invalid accesses because the pointer bounds deducted during verification are enforced using branchless logic. See 979d63d50c0c ("bpf: prevent out of bounds speculation on pointer arithmetic") for details. Do not make the mitigation depend on !env->allow_{uninit_stack,ptr_leaks} because speculative leaks are likely unexpected if these were enabled. For example, leaking the address to a protected log file may be acceptable while disabling the mitigation might unintentionally leak the address into the cached-state of a map that is accessible to unprivileged processes.
CVE-2023-53026	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix ib block iterator counter overflow When registering a new DMA MR after selecting the best aligned page size for it, we iterate over the given sglist to split each entry to smaller, aligned to the selected page size, DMA blocks. In given circumstances where the sg entry and page size fit certain sizes and the sg entry is not aligned to the selected page size, the total size of the aligned pages we need to cover the sg entry is >= 4GB. Under this circumstances, while iterating page aligned blocks, the counter responsible for counting how much we advanced from the start of the sg entry is overflowed because its type is u32 and we pass 4GB in size. This can lead to an infinite loop inside the iterator function because the overflow prevents the counter to be larger than the size of the sg entry. Fix the presented problem by changing the advancement condition to eliminate overflow. Backtrace: [192.374329] efa_reg_user_mr_dmabuf [192.376783] efa_register_mr [192.382579] pgsz_bitmap 0xfffff000 rounddown 0x80000000 [192.386423] pg_sz [0x80000000] umem_length[0xc0000000] [192.392657] start 0x0 length 0xc0000000 params.page_shift 31 params.page_num 3 [192.399559] hp_cnt[3], pages_in_hp[524288] [192.403690] umem->sgt_append.sgt.nents[1] [192.407905] number entries: [1], pg_bit: [31] [192.411397] biter->__sg_nents [1] biter->__sg [0000000008b0c5d8] [192.415601] biter->__sg_advance [665837568] sg_dma_len[3221225472] [192.419823] biter->__sg_nents [1] biter->__sg [0000000008b0c5d8] [192.423976] biter->__sg_advance [2813321216] sg_dma_len[3221225472] [192.428243] biter->__sg_nents [1] biter->__sg [0000000008b0c5d8] [192.432397] biter->__sg_advance [665837568] sg_dma_len[3221225472]

CVE ID	Description
CVE-2023-53031	In the Linux kernel, the following vulnerability has been resolved: powerpc/imc-pmu: Fix use of mutex in IRQs disabled section Current imc-pmu code triggers a WARNING with CONFIG_DEBUG_ATOMIC_SLEEP and CONFIG_PROVE_LOCKING enabled, while running a thread_imc event. Command to trigger the warning: # perf stat -e thread_imc/ CPM_CS_FROM_L4_MEM_X_DPTEG/ sleep 5 Performance counter stats for 'sleep 5': 0 thread_imc/ CPM_CS_FROM_L4_MEM_X_DPTEG/ 5.002117947 seconds time elapsed 0.000131000 seconds user 0.001063000 seconds sys Below is snippet of the warning in dmesg: BUG: sleeping function called from invalid context at kernel/locking/mutex.c:580 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 2869, name: perf-exec preempt_count: 2, expected: 0 4 locks held by perf-exec/2869: #0: c00000004325c540 (&sig->cred_guard_mutex){+..-}{3:3}, at: bprm_execve +0x64/0xa90 #1: c00000004325c5d8 (&sig->exec_update_lock){++++-}{3:3}, at: begin_new_exec +0x460/0xef0 #2: c0000003fa99d4e0 (&cputx_lock){-...-}{2:2}, at: perf_event_exec+0x290/0x510 #3: c000000017ab8418 (&ctx->lock){....-}{2:2}, at: perf_event_exec+0x29c/0x510 irq event stamp: 4806 hardirqs last enabled at (4805): [<c000000000f65b94>] _raw_spin_unlock_irqrestore +0x94/0xd0 hardirqs last disabled at (4806): [<c0000000003fae44>] perf_event_exec+0x394/0x510 softirqs last enabled at (0): [<c00000000013c404>] copy_process+0xc34/0x1ff0 softirqs last disabled at (0): [<0000000000000000>] 0x0 CPU: 36 PID: 2869 Comm: perf-exec Not tainted 6.2.0-rc2-00011-g1247637727f2 #61 Hardware name: 8375-42A POWER9 0x4e1202 opal:v7.0-16-g9b85f7d961 PowerNV Call Trace: dump_stack_lvl+0x98/0xe0 (unreliable) __might_resched +0x2f8/0x310 __mutex_lock+0x6c/0x13f0 thread_imc_event_add+0xf4/0x1b0 event_sched_in +0xe0/0x210 merge_sched_in+0x1f0/0x600 visit_groups_merge.isra.92.constprop.166+0x2bc/0x6c0 ctx_flexible_sched_in+0xcc/0x140 ctx_sched_in+0x20c/0x2a0 ctx_resched+0x104/0x1c0 perf_event_exec+0x340/0x510 begin_new_exec+0x730/0xef0 load_elf_binary+0x3f8/0x1e10 ... do not call blocking ops when !TASK_RUNNING; state=2001 set at [<00000000fd63e7cf>] do_nanosleep +0x60/0x1a0 WARNING: CPU: 36 PID: 2869 at kernel/sched/core.c:9912 __might_sleep+0x9c/0xb0 CPU: 36 PID: 2869 Comm: sleep Tainted: G W 6.2.0-rc2-00011-g1247637727f2 #61 Hardware name: 8375-42A POWER9 0x4e1202 opal:v7.0-16-g9b85f7d961 PowerNV NIP: c000000000194a1c LR: c000000000194a18 CTR: c000000000a78670 REGS: c00000004d2134e0 TRAP: 0700 Tainted: G W (6.2.0-rc2-00011-g1247637727f2) MSR: 9000000000021033 <SF,HV,ME,IR,DR,RI,LE> CR: 48002824 XER: 00000000 CFAR: c00000000013fb64 IRQMASK: 1 The above warning triggered because the current imc-pmu code uses mutex lock in interrupt disabled sections. The function mutex_lock() internally calls __might_resched(), which will check if IRQs are disabled and in case IRQs are disabled, it will trigger the warning. Fix the issue by changing the mutex lock to spinlock. [mpe: Fix comments, trim oops in change log, add reported-by tags]
CVE-2023-53032	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: Fix overflow before widen in the bitmap_ip_create() function. When first_ip is 0, last_ip is 0xFFFFFFFF, and netmask is 31, the value of an arithmetic expression 2 << (netmask - mask_bits - 1) is subject to overflow due to a failure casting operands to a larger data type before performing the arithmetic. Note that it's harmless since the value will be checked at the next step. Found by InfoTeCS on behalf of Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-53034	In the Linux kernel, the following vulnerability has been resolved: ntb_hw_switchtec: Fix shift-out-of-bounds in switchtec_ntb_mw_set_trans There is a kernel API ntb_mw_clear_trans() would pass 0 to both addr and size. This would make xlate_pos negative. [23.734156] switchtec switchtec0: MW 0: part 0 addr 0x0000000000000000 size 0x0000000000000000 [23.734158] ===== [23.734172] UBSAN: shift-out-of-bounds in drivers/ntb/hw/mscc/ntb_hw_switchtec.c:293:7 [23.734418] shift exponent -1 is negative Ensuring xlate_pos is a positive or zero before BIT.
CVE-2023-53036	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix call trace warning and hang when removing amdgpu device On GPUs with RAS enabled, below call trace and hang are observed when shutting down device. v2: use DRM device unplugged flag instead of shutdown flag as the check to prevent memory wipe in shutdown stage. [+0.000000] RIP: 0010:amdgpu_vram_mgr_fini +0x18d/0x1c0 [amdgpu] [+0.000001] PKRU: 55555554 [+0.000001] Call Trace: [+0.000001] <TASK> [+0.000002] amdgpu_ttm_fini+0x140/0x1c0 [amdgpu] [+0.000183] amdgpu_bo_fini +0x27/0xa0 [amdgpu] [+0.000184] gmc_v11_0_sw_fini+0x2b/0x40 [amdgpu] [+0.000163] amdgpu_device_fini_sw+0xb6/0x510 [amdgpu] [+0.000152] amdgpu_driver_release_kms+0x16/0x30 [amdgpu] [+0.000090] drm_dev_release+0x28/0x50 [drm] [+0.000016] devm_drm_dev_init_release +0x38/0x60 [drm] [+0.000011] devm_action_release+0x15/0x20 [+0.000003] release_nodes +0x40/0xc0 [+0.000001] devres_release_all+0x9e/0xe0 [+0.000001] device_unbind_cleanup +0x12/0x80 [+0.000003] device_release_driver_internal+0xff/0x160 [+0.000001] driver_detach +0x4a/0x90 [+0.000001] bus_remove_driver+0x6c/0xf0 [+0.000001] driver_unregister+0x31/0x50 [+0.000001] pci_unregister_driver+0x40/0x90 [+0.000003] amdgpu_exit+0x15/0x120 [amdgpu]

CVE ID	Description
CVE-2023-53037	In the Linux kernel, the following vulnerability has been resolved: <code>scsi: mpi3mr</code> : Bad drive in topology results kernel crash When the SAS Transport Layer support is enabled and a device exposed to the OS by the driver fails INQUIRY commands, the driver frees up the memory allocated for an internal HBA port data structure. However, in some places, the reference to the freed memory is not cleared. When the firmware sends the Device Info change event for the same device again, the freed memory is accessed and that leads to memory corruption and OS crash.
CVE-2023-53038	In the Linux kernel, the following vulnerability has been resolved: <code>scsi: lpfc</code> : Check <code>kzalloc()</code> in <code>lpfc_sli4_cgn_params_read()</code> If <code>kzalloc()</code> fails in <code>lpfc_sli4_cgn_params_read()</code> , then we rely on <code>lpfc_read_object()</code> 's routine to NULL check <code>pdata</code> . Currently, an early return error is thrown from <code>lpfc_read_object()</code> to protect us from NULL ptr dereference, but the <code>errno</code> code is <code>-ENODEV</code> . Change the <code>errno</code> code to a more appropriate <code>-ENOMEM</code> .
CVE-2023-53039	In the Linux kernel, the following vulnerability has been resolved: <code>HID: intel-ish-hid: ipc</code> : Fix potential use-after-free in work function When a reset notify IPC message is received, the ISR schedules a work function and passes the ISHTP device to it via a global pointer <code>ishtp_dev</code> . If <code>ish_probe()</code> fails, the devm-managed device resources including <code>ishtp_dev</code> are freed, but the work is not cancelled, causing a use-after-free when the work function tries to access <code>ishtp_dev</code> . Use <code>devm_work_autocancel()</code> instead, so that the work is automatically cancelled if probe fails.
CVE-2023-53041	In the Linux kernel, the following vulnerability has been resolved: <code>scsi: qla2xxx</code> : Perform lockless command completion in abort path While adding and removing the controller, the following call trace was observed: WARNING: CPU: 3 PID: 623596 at kernel/dma/mapping.c:532 dma_free_attrs+0x33/0x50 CPU: 3 PID: 623596 Comm: sh Kdump: loaded Not tainted 5.14.0-96.el9.x86_64 #1 RIP: 0010:dma_free_attrs+0x33/0x50 Call Trace: qla2x00_async_sns_sp_done+0x107/0x1b0 [qla2xxx] qla2x00_abort_srb+0x8e/0x250 [qla2xxx] ? ql_dbg+0x70/0x100 [qla2xxx] __qla2x00_abort_all_cmds+0x108/0x190 [qla2xxx] qla2x00_abort_all_cmds+0x24/0x70 [qla2xxx] qla2x00_abort_isp_cleanup+0x305/0x3e0 [qla2xxx] qla2x00_remove_one+0x364/0x400 [qla2xxx] pci_device_remove+0x36/0xa0 __device_release_driver+0x17a/0x230 device_release_driver+0x24/0x30 pci_stop_bus_device+0x68/0x90 pci_stop_and_remove_bus_device_locked+0x16/0x30 remove_store+0x75/0x90 kernfs_fop_write_iter+0x11c/0x1b0 new_sync_write+0x11f/0x1b0 vfs_write+0x1eb/0x280 ksys_write+0x5f/0xe0 do_syscall_64+0x5c/0x80 ? do_user_addr_fault+0x1d8/0x680 ? do_syscall_64+0x69/0x80 ? exc_page_fault+0x62/0x140 ? asm_exc_page_fault+0x8/0x30 entry_SYSCALL_64_after_hwframe+0x44/0xae The command was completed in the abort path during driver unload with a lock held, causing the warning in abort path. Hence complete the command without any lock held.
CVE-2023-53044	In the Linux kernel, the following vulnerability has been resolved: <code>dm stats</code> : check for and propagate <code>alloc_percpu</code> failure Check <code>alloc_percpu()</code> 's return value and return an error from <code>dm_stats_init()</code> if it fails. Update <code>alloc_dev()</code> to fail if <code>dm_stats_init()</code> does. Otherwise, a NULL pointer dereference will occur in <code>dm_stats_cleanup()</code> even if <code>dm-stats</code> isn't being actively used.
CVE-2023-53046	In the Linux kernel, the following vulnerability has been resolved: <code>Bluetooth</code> : Fix race condition in <code>hci_cmd_sync_clear</code> There is a potential race condition in <code>hci_cmd_sync_work</code> and <code>hci_cmd_sync_clear</code> , and could lead to use-after-free. For instance, <code>hci_cmd_sync_work</code> is added to the 'req_workqueue' after <code>cancel_work_sync</code> The entry of 'cmd_sync_work_list' may be freed in <code>hci_cmd_sync_clear</code> , and causing kernel panic when it is used in 'hci_cmd_sync_work'. Here's the call trace: dump_stack_lvl+0x49/0x63 print_report.cold+0x5e/0x5d3 ? hci_cmd_sync_work+0x282/0x320 kasan_report+0xaa/0x120 ? hci_cmd_sync_work+0x282/0x320 __asan_report_load8_noabort+0x14/0x20 hci_cmd_sync_work+0x282/0x320 process_one_work+0x77b/0x11c0 ? _raw_spin_lock_irq+0x8e/0xf0 worker_thread+0x544/0x1180 ? poll_idle+0x1e0/0x1e0 kthread+0x285/0x320 ? process_one_work+0x11c0/0x11c0 ? kthread_complete_and_exit+0x30/0x30 ret_from_fork+0x22/0x30 </TASK> Allocated by task 266: kasan_save_stack+0x26/0x50 __kasan_kmalloc+0xae/0xe0 kmem_cache_alloc_trace+0x191/0x350 hci_cmd_sync_queue+0x97/0x2b0 hci_update_passive_scan+0x176/0x1d0 le_conn_complete_evt+0x1b5/0x1a00 hci_le_conn_complete_evt+0x234/0x340 hci_le_meta_evt+0x231/0x4e0 hci_event_packet+0x4c5/0xf00 hci_rx_work+0x37d/0x880 process_one_work+0x77b/0x11c0 worker_thread+0x544/0x1180 kthread+0x285/0x320 ret_from_fork+0x22/0x30 Freed by task 269: kasan_save_stack+0x26/0x50 kasan_set_track+0x25/0x40 kasan_set_free_info+0x24/0x40 ____kasan_slab_free+0x176/0x1c0 __kasan_slab_free+0x12/0x20 slab_free_freelist_hook+0x95/0x1a0 kfree+0xba/0x2f0 hci_cmd_sync_clear+0x14c/0x210 hci_unregister_dev+0xff/0x440 vhci_release+0x7b/0xf0 __fput+0x1f3/0x970 __fput+0xe/0x20 task_work_run+0xd4/0x160 do_exit+0x8b0/0x22a0 do_group_exit+0xba/0x2a0 get_signal+0x1e4a/0x25b0 arch_do_signal_or_restart+0x93/0x1f80 exit_to_user_mode_prepare+0xf5/0x1a0 syscall_exit_to_user_mode+0x26/0x50 ret_from_fork+0x15/0x30

CVE ID	Description
CVE-2023-53048	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tcpm: fix warning when handle discover_identity message Since both source and sink device can send discover_identity message in PD3, kernel may dump below warning: -----[cut here]----- WARNING: CPU: 0 PID: 169 at drivers/usb/typec/tcpm/tcpm.c:1446 tcpm_queue_vdm+0xe0/0xf0 Modules linked in: CPU: 0 PID: 169 Comm: 1-0050 Not tainted 6.1.1-00038-g6a3c36cf1da2-dirty #567 Hardware name: NXP i.MX8MPlus EVK board (DT) pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPEDIT) pc : tcpm_queue_vdm+0xe0/0xf0 lr : tcpm_queue_vdm+0x2c/0xf0 sp : ffff80000c19bcd0 x29: ffff80000c19bcd0 x28: 0000000000000001 x27: ffff0000d11c8ab8 x26: ffff0000d11cc000 x25: 0000000000000000 x24: 00000000ff008081 x23: 0000000000000001 x22: 00000000ff00a081 x21: ffff80000c19bdbc x20: 0000000000000000 x19: ffff0000d11c8080 x18: ffffffff00000000 x17: 0000000000000000 x16: 0000000000000000 x15: ffff0000d716f580 x14: 0000000000000001 x13: ffff0000d716f507 x12: 0000000000000001 x11: 0000000000000000 x10: 0000000000000020 x9 : 000000000000ee098 x8 : 00000000ffffffff x7 : 000000000000001c x6 : ffff0000d716f580 x5 : 0000000000000000 x4 : 0000000000000000 x3 : 0000000000000000 x2 : ffff80000c19bdbc x1 : 00000000ff00a081 x0 : 0000000000000004 Call trace: tcpm_queue_vdm+0xe0/0xf0 tcpm_pd_rx_handler+0x340/0x1ab0 kthread_worker_fn+0xcc/0x18c kthread+0x10c/0x110 ret_from_fork+0x10/0x20 ---[end trace 0000000000000000]--- Below sequences may trigger this warning: tcpm_send_discover_work(work) tcpm_send_vdm(port, USB_SID_PD, CMD_DISCOVER_IDENT, NULL, 0); tcpm_queue_vdm(port, header, data, count); port->vdm_state = VDM_STATE_READY; vdm_state_machine_work(work); <-- received discover_identity from partner vdm_run_state_machine(port); port->vdm_state = VDM_STATE_SEND_MESSAGE; mod_vdm_delayed_work(port, x); tcpm_pd_rx_handler(work); tcpm_pd_data_request(port, msg); tcpm_handle_vdm_request(port, msg->payload, cnt); tcpm_queue_vdm(port, response[0], &response[1], rlen - 1); --> WARN_ON(port->vdm_state > VDM_STATE_DONE); For this case, the state machine could still send out discover identity message later if we skip current discover_identity message. So we should handle the received message firstly and override the pending discover_identity message without warning in this case. Then, a delayed send_discover work will send discover_identity message again.
CVE-2023-53049	In the Linux kernel, the following vulnerability has been resolved: usb: ucsi: Fix NULL pointer deref in ucsi_connector_change() When ucsi_init() fails, ucsi->connector is NULL, yet in case of ucsi_acpi we may still get events which cause the ucs_acpi code to call ucsi_connector_change(), which then derefs the NULL ucsi->connector pointer. Fix this by not setting ucsi->ntfy inside ucsi_init() until ucsi_init() has succeeded, so that ucsi_connector_change() ignores the events because UCSI_ENABLE_NTFY_CONNECTOR_CHANGE is not set in the ntfy mask.
CVE-2023-53052	In the Linux kernel, the following vulnerability has been resolved: cifs: fix use-after-free bug in refresh_cache_worker() The UAF bug occurred because we were putting DFS root sessions in cifs_umount() while DFS cache refresher was being executed. Make DFS root sessions have same lifetime as DFS tcons so we can avoid the use-after-free bug is DFS cache refresher and other places that require IPCs to get new DFS referrals on. Also, get rid of mount group handling in DFS cache as we no longer need it. This fixes below use-after-free bug caught by KASAN [379.946955] BUG: KASAN: use-after-free in __refresh_tcon.isra.0+0x10b/0xc10 [cifs] [379.947642] Read of size 8 at addr ffff888018f57030 by task kworker/u4:3/56 [379.948096] [379.948208] CPU: 0 PID: 56 Comm: kworker/u4:3 Not tainted 6.2.0-rc7-lku #23 [379.948661] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.0-0-gd239552-rebuilt.opensuse.org 04/01/2014 [379.949368] Workqueue: cifs-dfscache refresh_cache_worker [cifs] [379.949942] Call Trace: [379.950113] <TASK> [379.950260] dump_stack_lvl+0x50/0x67 [379.950510] print_report+0x16a/0x48e [379.950759] ? __virt_addr_valid+0xd8/0x160 [379.951040] ? __phys_addr+0x41/0x80 [379.951285] kasan_report+0xdb/0x110 [379.951533] ? __refresh_tcon.isra.0+0x10b/0xc10 [cifs] [379.952056] ? __refresh_tcon.isra.0+0x10b/0xc10 [cifs] [379.952585] __refresh_tcon.isra.0+0x10b/0xc10 [cifs] [379.953096] ? __pfx__refresh_tcon.isra.0+0x10/0x10 [cifs] [379.953637] ? __pfx__mutex_lock+0x10/0x10 [379.953915] ? lock_release+0xb6/0x720 [379.954167] ? __pfx_lock_acquire+0x10/0x10 [379.954443] ? refresh_cache_worker+0x34e/0x6d0 [cifs] [379.954960] ? __pfx_wb_workfn+0x10/0x10 [379.955239] refresh_cache_worker+0x4ad/0x6d0 [cifs] [379.955755] ? __pfx_refresh_cache_worker+0x10/0x10 [cifs] [379.956323] ? __pfx_lock_acquired+0x10/0x10 [379.956615] ? read_word_at_a_time+0xe/0x20 [379.956898] ? lockdep_hardirqs_on_prepare+0x12/0x220 [379.957235] process_one_work+0x535/0x990 [379.957509] ? __pfx_process_one_work+0x10/0x10 [379.957812] ? lock_acquired+0xb7/0x5f0 [379.958069] ? __list_add_valid+0x37/0xd0 [379.958341] ? __list_add_valid+0x37/0xd0 [379.958611] worker_thread+0x8e/0x630 [379.958861] ? __pfx_worker_thread+0x10/0x10 [379.959148] kthread+0x17d/0x1b0 [379.959369] ? __pfx_kthread+0x10/0x10 [379.959630] ret_from_fork+0x2c/0x50 [379.959879] </TASK>

CVE ID	Description
CVE-2023-53053	In the Linux kernel, the following vulnerability has been resolved: erspan: do not use skb_mac_header() in ndo_start_xmit() Drivers should not assume skb_mac_header(skb) == skb->data in their ndo_start_xmit(). Use skb_network_offset() and skb_transport_offset() which better describe what is needed in erspan_fb_xmit() and ip6erspan_tunnel_xmit() syzbot reported: WARNING: CPU: 0 PID: 5083 at include/linux/skbuff.h:2873 skb_mac_header include/linux/skbuff.h:2873 [inline] WARNING: CPU: 0 PID: 5083 at include/linux/skbuff.h:2873 ip6erspan_tunnel_xmit+0x1d9c/0x2d90 net/ipv6/ip6_gre.c:962 Modules linked in: CPU: 0 PID: 5083 Comm: syz-executor406 Not tainted 6.3.0-rc2-syzkaller-00866-gd4671cb96fa3 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/02/2023 RIP: 0010:skb_mac_header include/linux/skbuff.h:2873 [inline] RIP: 0010:ip6erspan_tunnel_xmit+0x1d9c/0x2d90 net/ipv6/ip6_gre.c:962 Code: 04 02 41 01 de 84 c0 74 08 3c 03 0f 8e 1c 0a 00 00 45 89 b4 24 c8 00 00 00 c6 85 77 fe ff 01 e9 33 e7 ff ff e8 b4 27 a1 f8 <0f> 0b e9 b6 e7 ff ff e8 a8 27 a1 f8 49 8d bf f0 0c 00 00 48 b8 00 RSP: 0018:ffffc90003b2f830 EFLAGS: 00010293 RAX: 0000000000000000 RBX: 000000000000ffff RCX: 0000000000000000 RDX: ffff888021273a80 RSI: ffffffff88e1bd4c RDI: 0000000000000003 RBP: ffff90003b2f9d8 R08: 0000000000000003 R09: 000000000000ffff R10: 000000000000ffff R11: 0000000000000000 R12: ffff88802b28da00 R13: 00000000000000d0 R14: ffff88807e25b6d0 R15: ffff888023408000 FS: 0000555556a61300(0000) GS: ffff8880b9800000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055e5b11eb6e8 CR3: 0000000027c1b000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __netdev_start_xmit include/linux/netdevice.h:4900 [inline] netdev_start_xmit include/linux/netdevice.h:4914 [inline] __dev_direct_xmit+0x504/0x730 net/core/dev.c:4300 dev_direct_xmit include/linux/netdevice.h:3088 [inline] packet_xmit+0x20a/0x390 net/packet/af_packet.c:285 packet_snd net/packet/af_packet.c:3075 [inline] packet_sendmsg+0x31a0/0x5150 net/packet/af_packet.c:3107 sock_sendmsg_nosec net/socket.c:724 [inline] sock_sendmsg+0xde/0x190 net/socket.c:747 __sys_sendto+0x23a/0x340 net/socket.c:2142 __do_sys_sendto net/socket.c:2154 [inline] __se_sys_sendto net/socket.c:2150 [inline] __x64_sys_sendto+0xe1/0x1b0 net/socket.c:2150 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7f123aaa1039 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 b1 14 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 c0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007ffc15d12058 EFLAGS: 00000246 ORIG_RAX: 000000000000002c RAX: ffffffff8bda RBX: 0000000000000000 RCX: 00007f123aaa1039 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003 RBP: 0000000000000000 R08: 0000000020000040 R09: 0000000000000014 R10: 0000000000000000 R11: 0000000000000246 R12: 00007f123aa648c0 R13: 431bde82d7b634db R14: 0000000000000000 R15: 0000000000000000

CVE ID	Description
CVE-2023-53057	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: HCI: Fix global-out-of-bounds To loop a variable-length array, hci_init_stage_sync(stage) considers that stage[i] is valid as long as stage[i-1].func is valid. Thus, the last element of stage[].func should be intentionally invalid as hci_init0[], le_init2[], and others did. However, amp_init1[] and amp_init2[] have no invalid element, letting hci_init_stage_sync() keep accessing amp_init1[] over its valid range. This patch fixes this by adding {} in the last of amp_init1[] and amp_init2[]. ===== BUG: KASAN: global-out-of-bounds in hci_dev_open_sync (/v6.2-bzimage/net/bluetooth/hci_sync.c:3154 /v6.2-bzimage/net/bluetooth/hci_sync.c:3343 /v6.2-bzimage/net/bluetooth/hci_sync.c:4418 /v6.2-bzimage/net/bluetooth/hci_sync.c:4609 /v6.2-bzimage/net/bluetooth/hci_sync.c:4689) Read of size 8 at addr ffffffffad1ab70 by task kworker/u5:0/1032 CPU: 0 PID: 1032 Comm: kworker/u5:0 Not tainted 6.2.0 #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04 Workqueue: hci1 hci_power_on Call Trace: <TASK> dump_stack_lvl (/v6.2-bzimage/lib/dump_stack.c:107 (discriminator 1)) print_report (/v6.2-bzimage/mm/kasan/report.c:307 /v6.2-bzimage/mm/kasan/report.c:417) ? hci_dev_open_sync (/v6.2-bzimage/net/bluetooth/hci_sync.c:3154 /v6.2-bzimage/net/bluetooth/hci_sync.c:3343 /v6.2-bzimage/net/bluetooth/hci_sync.c:4418 /v6.2-bzimage/net/bluetooth/hci_sync.c:4609 /v6.2-bzimage/net/bluetooth/hci_sync.c:4689) kasan_report (/v6.2-bzimage/mm/kasan/report.c:184 /v6.2-bzimage/mm/kasan/report.c:519) ? hci_dev_open_sync (/v6.2-bzimage/net/bluetooth/hci_sync.c:3154 /v6.2-bzimage/net/bluetooth/hci_sync.c:3343 /v6.2-bzimage/net/bluetooth/hci_sync.c:4418 /v6.2-bzimage/net/bluetooth/hci_sync.c:4609 /v6.2-bzimage/net/bluetooth/hci_sync.c:4689) hci_dev_open_sync (/v6.2-bzimage/net/bluetooth/hci_sync.c:3154 /v6.2-bzimage/net/bluetooth/hci_sync.c:3343 /v6.2-bzimage/net/bluetooth/hci_sync.c:4418 /v6.2-bzimage/net/bluetooth/hci_sync.c:4609 /v6.2-bzimage/net/bluetooth/hci_sync.c:4689) ? __pfx_hci_dev_open_sync (/v6.2-bzimage/net/bluetooth/hci_sync.c:4635) ? mutex_lock (/v6.2-bzimage/.arch/x86/include/asm/atomic64_64.h:190 /v6.2-bzimage/.include/linux/atomic/atomic-long.h:443 /v6.2-bzimage/.include/linux/atomic/atomic-instrumented.h:1781 /v6.2-bzimage/kernel/locking/mutex.c:171 /v6.2-bzimage/kernel/locking/mutex.c:285) ? __pfx_mutex_lock (/v6.2-bzimage/kernel/locking/mutex.c:282) hci_power_on (/v6.2-bzimage/net/bluetooth/hci_core.c:485 /v6.2-bzimage/net/bluetooth/hci_core.c:984) ? __pfx_hci_power_on (/v6.2-bzimage/net/bluetooth/hci_core.c:969) ? read_word_at_a_time (/v6.2-bzimage/.include/asm-generic/rwonce.h:85) ? strscpy (/v6.2-bzimage/.arch/x86/include/asm/word-at-a-time.h:62 /v6.2-bzimage/lib/string.c:161) process_one_work (/v6.2-bzimage/kernel/workqueue.c:2294) worker_thread (/v6.2-bzimage/.include/linux/list.h:292 /v6.2-bzimage/kernel/workqueue.c:2437) ? __pfx_worker_thread (/v6.2-bzimage/kernel/workqueue.c:2379) kthread (/v6.2-bzimage/kernel/kthread.c:376) ? __pfx_kthread (/v6.2-bzimage/kernel/kthread.c:331) ret_from_fork (/v6.2-bzimage/arch/x86/entry/entry_64.S:314) </TASK> The buggy address belongs to the variable: amp_init1+0x30/0x60 The buggy address belongs to the physical page: page:000000003a157ec6 refcount:1 mapcount:0 mapping:0000000000000000 ia flags: 0x2000000000001000(reserved node=0 zone=2) raw: 0200000000001000 ffffea0005054688 ffffea0005054688 0000000000000000 raw: 0000000000000000 0000000000000000 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffffffffad1aa00: f9 f9 f9 f9 00 00 00 00 f9 f9 f9 f9 00 00 00 00 ffffffffad1aa80: 00 00 00 00 f9 f9 f9 f9 00 00 00 00 00 00 00 00 >fffffffad1ab00: 00 f9 f9 f9 f9 f9 f9 00 00 00 00 00 00 f9 f9 ---truncated---
CVE-2023-53062	In the Linux kernel, the following vulnerability has been resolved: net: usb: smsc95xx: Limit packet length to skb->len Packet length retrieved from descriptor may be larger than the actual socket buffer length. In such case the cloned skb passed up the network stack will leak kernel memory contents.

CVE ID	Description
CVE-2023-53064	In the Linux kernel, the following vulnerability has been resolved: iavf: fix hang on reboot with ice When a system with E810 with existing VFs gets rebooted the following hang may be observed. Pid 1 is hung in iavf_remove(), part of a network driver: PID: 1 TASK: ffff965400e5a340 CPU: 24 COMMAND: "systemd-shutdown" #0 [ffffaad04005fa50] __schedule at ffffffff8b3239cb #1 [ffffaad04005fae8] schedule at ffffffff8b323e2d #2 [ffffaad04005fb00] schedule_hrtimeout_range_clock at ffffffff8b32cebc #3 [ffffaad04005fb80] usleep_range_state at ffffffff8b32c930 #4 [ffffaad04005fbb0] iavf_remove at ffffffff8b32b9b4c [iavf] #5 [ffffaad04005fbf0] pci_device_remove at ffffffff8add7513 #6 [ffffaad04005fc10] device_release_driver_internal at ffffffff8af08baa #7 [ffffaad04005fc40] pci_stop_bus_device at ffffffff8adcc5fc #8 [ffffaad04005fc60] pci_stop_and_remove_bus_device at ffffffff8adcc81e #9 [ffffaad04005fc70] pci_iov_remove_virtfn at ffffffff8adf9429 #10 [ffffaad04005fca8] sriov_disable at ffffffff8adf98e4 #11 [ffffaad04005fcc8] ice_free_vfs at ffffffff8c04bb2c8 [ice] #12 [ffffaad04005fd10] ice_remove at ffffffff8c0477fe [ice] #13 [ffffaad04005fd38] ice_shutdown at ffffffff8c0477946 [ice] #14 [ffffaad04005fd50] pci_device_shutdown at ffffffff8add58f1 #15 [ffffaad04005fd70] device_shutdown at ffffffff8af05386 #16 [ffffaad04005fd98] kernel_restart at ffffffff8a92a870 #17 [ffffaad04005fda8] __do_sys_reboot at ffffffff8a92abd6 #18 [ffffaad04005fee0] do_syscall_64 at ffffffff8b317159 #19 [ffffaad04005ff08] __context_tracking_enter at ffffffff8b31b6fc #20 [ffffaad04005ff18] syscall_exit_to_user_mode at ffffffff8b31b50d #21 [ffffaad04005ff28] do_syscall_64 at ffffffff8b317169 #22 [ffffaad04005ff50] entry_SYSCALL_64_after_hwframe at ffffffff8b40009b RIP: 00007f1baa5c13d7 RSP: 00007ffbcc55a98 RFLAGS: 00000202 RAX: ffffffff8da RBX: 0000000000000000 RCX: 00007f1baa5c13d7 RDX: 0000000001234567 RSI: 00000000028121969 RDI: 00000000fee1dead RBP: 00007ffbcc55ca0 R8: 0000000000000000 R9: 00007ffbcc54e90 R10: 00007ffbcc55050 R11: 0000000000000202 R12: 0000000000000005 R13: 0000000000000000 R14: 00007ffbcc55af0 R15: 0000000000000000 ORIG_RAX: 00000000000000a9 CS: 0033 SS: 002b During reboot all drivers PM shutdown callbacks are invoked. In iavf_shutdown() the adapter state is changed to __IAVF_REMOVE. In ice_shutdown() the call chain above is executed, which at some point calls iavf_remove(). However iavf_remove() expects the VF to be in one of the states __IAVF_RUNNING, __IAVF_DOWN or __IAVF_INIT_FAILED. If that's not the case it sleeps forever. So if iavf_shutdown() gets invoked before iavf_remove() the system will hang indefinitely because the adapter is already in state __IAVF_REMOVE. Fix this by returning from iavf_remove() if the state is __IAVF_REMOVE, as we already went through iavf_shutdown().
CVE-2023-53065	In the Linux kernel, the following vulnerability has been resolved: perf/core: Fix perf_output_begin parameter is incorrectly invoked in perf_event_bpf_output syzcaller reportes a KASAN issue with stack-out-of-bounds. The call trace is as follows: dump_stack+0x9c/0xd3 print_address_description.constprop.0+0x19/0x170 __kasan_report.cold+0x6c/0x84 kasan_report+0x3a/0x50 __perf_event_header__init_id+0x34/0x290 perf_event_header__init_id+0x48/0x60 perf_output_begin+0x4a4/0x560 perf_event_bpf_output+0x161/0x1e0 perf_iterate_sb_cpu+0x29e/0x340 perf_iterate_sb+0x4c/0xc0 perf_event_bpf_event+0x194/0x2c0 __bpf_prog_put.constprop.0+0x55/0xf0 __cls_bpf_delete_prog+0xea/0x120 [cls_bpf] cls_bpf_delete_prog_work+0x1c/0x30 [cls_bpf] process_one_work+0x3c2/0x730 worker_thread+0x93/0x650 kthread+0x1b8/0x210 ret_from_fork+0x1f/0x30 commit 267fb27352b6 ("perf: Reduce stack usage of perf_output_begin()") use on-stack struct perf_sample_data of the caller function. However, perf_event_bpf_output uses incorrect parameter to convert small-sized data (struct perf_bpf_event) into large-sized data (struct perf_sample_data), which causes memory overwriting occurs in __perf_event_header__init_id.
CVE-2023-53066	In the Linux kernel, the following vulnerability has been resolved: qed/qed_sriov: guard against NULL derefs from qed_iiov_get_vf_info We have to make sure that the info returned by the helper is valid before using it. Found by Linux Verification Center (linuxtesting.org) with the SVACE static analysis tool.
CVE-2023-53068	In the Linux kernel, the following vulnerability has been resolved: net: usb: lan78xx: Limit packet length to skb->len Packet length retrieved from descriptor may be larger than the actual socket buffer length. In such case the cloned skb passed up the network stack will leak kernel memory contents. Additionally prevent integer underflow when size is less than ETH_FCS_LEN.

CVE ID	Description
CVE-2023-53070	In the Linux kernel, the following vulnerability has been resolved: ACPI: PPTT: Fix to avoid sleep in the atomic context when PPTT is absent Commit 0c80f9e165f8 ("ACPI: PPTT: Leave the table mapped for the runtime usage") enabled to map PPTT once on the first invocation of acpi_get_pptt() and never unmapped the same allowing it to be used at runtime with out the hassle of mapping and unmapping the table. This was needed to fetch LLC information from the PPTT in the cpufreq path which is executed in the atomic context as the acpi_get_table() might sleep waiting for a mutex. However it missed to handle the case when there is no PPTT on the system which results in acpi_get_pptt() being called from all the secondary CPUs attempting to fetch the LLC information in the atomic context without knowing the absence of PPTT resulting in the splat like below: BUG: sleeping function called from invalid context at kernel/locking/semaphore.c:164 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 0, name: swapper/1 preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 no locks held by swapper/1/0. irq event stamp: 0 hardirqs last enabled at (0): 0x0 hardirqs last disabled at (0): copy_process+0x61c/0x1b40 softirqs last enabled at (0): copy_process+0x61c/0x1b40 softirqs last disabled at (0): 0x0 CPU: 1 PID: 0 Comm: swapper/1 Not tainted 6.3.0-rc1 #1 Call trace: dump_backtrace+0xac/0x138 show_stack+0x30/0x48 dump_stack_lvl+0x60/0xb0 dump_stack+0x18/0x28 __might_resched+0x160/0x270 __might_sleep+0x58/0xb0 down_timeout+0x34/0x98 acpi_os_wait_semaphore+0x7c/0xc0 acpi_ut_acquire_mutex+0x58/0x108 acpi_get_table+0x40/0xe8 acpi_get_pptt+0x48/0xa0 acpi_get_cache_info+0x38/0x140 init_cache_level+0xf4/0x118 detect_cache_attributes+0x2e4/0x640 update_siblings_masks+0x3c/0x330 store_cpu_topology+0x88/0xf0 secondary_start_kernel+0xd0/0x168 __secondary_switched+0xb8/0xc0 Update acpi_get_pptt() to consider the fact that PPTT is once checked and is not available on the system and return NULL avoiding any attempts to fetch PPTT and thereby avoiding any possible sleep waiting for a mutex in the atomic context.
CVE-2023-53071	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: do not run mt76_unregister_device() on unregistered hw Trying to probe a mt7921e pci card without firmware results in a successful probe where ieee80211_register_hw hasn't been called. When removing the driver, ieee80211_unregister_hw is called unconditionally leading to a kernel NULL pointer dereference. Fix the issue running mt76_unregister_device routine just for registered hw.
CVE-2023-53072	In the Linux kernel, the following vulnerability has been resolved: mptcp: use the workqueue to destroy unaccepted sockets Christoph reported a UaF at token lookup time after having refactored the passive socket initialization part: BUG: KASAN: use-after-free in __token_bucket_busy+0x253/0x260 Read of size 4 at addr ffff88810698d5b0 by task syz-executor653/3198 CPU: 1 PID: 3198 Comm: syz-executor653 Not tainted 6.2.0-rc59af4eaa31c1f6c00c8f1e448ed99a45c66340dd5 #6 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x6e/0x91 print_report+0x16a/0x46f kasan_report+0xad/0x130 __token_bucket_busy+0x253/0x260 mptcp_token_new_connect+0x13d/0x490 mptcp_connect+0x4ed/0x860 __inet_stream_connect+0x80e/0xd90 tcp_sendmsg_fastopen+0x3ce/0x710 mptcp_sendmsg+0xff1/0x1a20 inet_sendmsg+0x11d/0x140 __sys_sendto+0x405/0x490 __x64_sys_sendto+0xdc/0x1b0 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc We need to properly clean-up all the paired MPTCP-level resources and be sure to release the msk last, even when the unaccepted subflow is destroyed by the TCP internals via inet_child_forget(). We can re-use the existing MPTCP_WORK_CLOSE_SUBFLOW infra, explicitly checking that for the critical scenario: the closed subflow is the MPC one, the msk is not accepted and eventually going through full cleanup. With such change, __mptcp_destroy_sock() is always called on msk sockets, even on accepted ones. We don't need anymore to transiently drop one sk reference at msk clone time. Please note this commit depends on the parent one: mptcp: refactor passive socket initialization
CVE-2023-53073	In the Linux kernel, the following vulnerability has been resolved: perf/x86/amd/core: Always clear status for idx The variable 'status' (which contains the unhandled overflow bits) is not being properly masked in some cases, displaying the following warning: WARNING: CPU: 156 PID: 475601 at arch/x86/events/amd/core.c:972 amd_pmu_v2_handle_irq+0x216/0x270 This seems to be happening because the loop is being continued before the status bit being unset, in case x86_perf_event_set_period() returns 0. This is also causing an inconsistency because the "handled" counter is incremented, but the status bit is not cleaned. Move the bit cleaning together above, together when the "handled" counter is incremented.

CVE ID	Description
CVE-2023-53074	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix ttm_bo calltrace warning in psp_hw_fini The call trace occurs when the amdgpu is removed after the mode1 reset. During mode1 reset, from suspend to resume, there is no need to reinitialize the ta firmware buffer which caused the bo pin_count increase redundantly. [489.885525] Call Trace: [489.885525] <TASK> [489.885526] amdtm_bo_put+0x34/0x50 [amdtm] [489.885529] amdgpu_bo_free_kernel+0xe8/0x130 [amdgpu] [489.885620] psp_free_shared_bufs+0xb7/0x150 [amdgpu] [489.885720] psp_hw_fini+0xce/0x170 [amdgpu] [489.885815] amdgpu_device_fini_hw+0x2ff/0x413 [amdgpu] [489.885960] ? blocking_notifier_chain_unregister+0x56/0xb0 [489.885962] amdgpu_driver_unload_kms+0x51/0x60 [amdgpu] [489.886049] amdgpu_pci_remove+0x5a/0x140 [amdgpu] [489.886132] ? __pm_runtime_resume+0x60/0x90 [489.886134] pci_device_remove+0x3e/0xb0 [489.886135] __device_release_driver+0x1ab/0x2a0 [489.886137] driver_detach+0xf3/0x140 [489.886138] bus_remove_driver+0x6c/0xf0 [489.886140] driver_unregister+0x31/0x60 [489.886141] pci_unregister_driver+0x40/0x90 [489.886142] amdgpu_exit+0x15/0x451 [amdgpu]
CVE-2023-53075	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix invalid address access in lookup_rec() when index is 0 KASAN reported follow problem: BUG: KASAN: use-after-free in lookup_rec Read of size 8 at addr ffff000199270ff0 by task modprobe CPU: 2 Comm: modprobe Call trace: kasan_report __asan_load8 lookup_rec ftrace_location arch_check_ftrace_location check_kprobe_address_safe register_kprobe When checking pg->records[pg->index - 1].ip in lookup_rec(), it can get a pg which is newly added to ftrace_pages_start in ftrace_process_locs(). Before the first pg->index++, index is 0 and accessing pg->records[-1].ip will cause this problem. Don't check the ip when pg->index is 0.
CVE-2023-53077	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix shift-out-of-bounds in CalculateVMAndRowBytes [WHY] When PTEBufferSizeInRequests is zero, UBSAN reports the following warning because dml_log2 returns an unexpected negative value: shift exponent 4294966273 is too large for 32-bit type 'int' [HOW] In the case PTEBufferSizeInRequests is zero, skip the dml_log2() and assign the result directly.
CVE-2023-53080	In the Linux kernel, the following vulnerability has been resolved: xsk: Add missing overflow check in xdp_umem_reg The number of chunks can overflow u32. Make sure to return -EINVAL on overflow. Also remove a redundant u32 cast assigning umem->npgs.
CVE-2023-53084	In the Linux kernel, the following vulnerability has been resolved: drm/shmem-helper: Remove another errant put in error path drm_gem_shmem_mmap() doesn't own reference in error code path, resulting in the dma-buf shmem GEM object getting prematurely freed leading to a later use-after-free.

CVE ID	Description
CVE-2023-53086	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: connac: do not check WED status for non-mmio devices WED is supported just for mmio devices, so do not check it for usb or sdio devices. This patch fixes the crash reported below: [21.946627] wlp0s3u1i3: authenticate with c4:41:1e:f5:2b:1d [22.525298] wlp0s3u1i3: send auth to c4:41:1e:f5:2b:1d (try 1/3) [22.548274] wlp0s3u1i3: authenticate with c4:41:1e:f5:2b:1d [22.557694] wlp0s3u1i3: send auth to c4:41:1e:f5:2b:1d (try 1/3) [22.565885] wlp0s3u1i3: authenticated [22.569502] wlp0s3u1i3: associate with c4:41:1e:f5:2b:1d (try 1/3) [22.578966] wlp0s3u1i3: RX AssocResp from c4:41:1e:f5:2b:1d (capab=0x11 status=30 aid=3) [22.579113] wlp0s3u1i3: c4:41:1e:f5:2b:1d rejected association temporarily; comeback duration 1000 TU (1024 ms) [23.649518] wlp0s3u1i3: associate with c4:41:1e:f5:2b:1d (try 2/3) [23.752528] wlp0s3u1i3: RX AssocResp from c4:41:1e:f5:2b:1d (capab=0x11 status=0 aid=3) [23.797450] wlp0s3u1i3: associated [24.959527] kernel tried to execute NX-protected page - exploit attempt? (uid: 0) [24.959640] BUG: unable to handle page fault for address: ffff88800c223200 [24.959706] #PF: supervisor instruction fetch in kernel mode [24.959788] #PF: error_code(0x0011) - permissions violation [24.959846] PGD 2c01067 P4D 2c01067 PUD 2c02067 PMD c2a8063 PTE 800000000c223163 [24.959957] Oops: 0011 [#1] PREEMPT SMP [24.960009] CPU: 0 PID: 391 Comm: wpa_supplicant Not tainted 6.2.0-kvm #18 [24.960089] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.1-2.fc37 04/01/2014 [24.960191] RIP: 0010:0xffff88800c223200 [24.960446] RSP: 0018:ffffc90000ff7698 EFLAGS: 00010282 [24.960513] RAX: ffff888028397010 RBX: ffff88800c26e630 RCX: 0000000000000058 [24.960598] RDX: ffff88800c26f844 RSI: 0000000000000006 RDI: ffff888028397010 [24.960682] RBP: ffff88800ea72f00 R08: 18b873fbab2b964c R09: be06b38235f3c63c [24.960766] R10: 18b873fbab2b964c R11: be06b38235f3c63c R12: 0000000000000001 [24.960853] R13: ffff88800c26f84c R14: ffff8880063f0ff8 R15: ffff88800c26e644 [24.960950] FS: 00007effcea327c0(0000) GS: ffff88807dc00000(0000) knlGS: 0000000000000000 [24.961036] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [24.961106] CR2: ffff88800c223200 CR3: 000000000eaa2000 CR4: 00000000000000b0 [24.961190] Call Trace: [24.961219] <TASK> [24.961245] ? mt76_connac_mcu_add_key+0x2cf/0x310 [24.961313] ? mt7921_set_key+0x150/0x200 [24.961365] ? drv_set_key+0xa9/0x1b0 [24.961418] ? ieee80211_key_enable_hw_accel+0xd9/0x240 [24.961485] ? ieee80211_key_replace+0x3f3/0x730 [24.961541] ? crypto_shash_setkey+0x89/0xd0 [24.961597] ? ieee80211_key_link+0x2d7/0x3a0 [24.961664] ? crypto_aead_setauthsize+0x31/0x50 [24.961730] ? sta_info_hash_lookup+0xa6/0xf0 [24.961785] ? ieee80211_add_key+0x1fc/0x250 [24.961842] ? rdev_add_key+0x41/0x140 [24.961882] ? nl80211_parse_key+0x6c/0x2f0 [24.961940] ? nl80211_new_key+0x24a/0x290 [24.961984] ? genl_rcv_msg+0x36c/0x3a0 [24.962036] ? rdev_mod_link_station+0xe0/0xe0 [24.962102] ? nl80211_set_key+0x410/0x410 [24.962143] ? nl80211_pre_doit+0x200/0x200 [24.962187] ? genl_bind+0xc0/0xc0 [24.962217] ? netlink_rcv_skb+0xaa/0xd0 [24.962259] ? genl_rcv+0x24/0x40 [24.962300] ? netlink_unicast+0x224/0x2f0 [24.962345] ? netlink_sendmsg+0x30b/0x3d0 [24.962388] ? __sys_sendmsg+0x109/0x1b0 [24.962388] ? __sys_sendmsg+0x109/0x1b0 [24.962440] ? __import_iovec+0x2e/0x110 [24.962482] ? __sys_sendmsg+0xbe/0xe0 [24.962525] ? mod_objcg_state+0x25c/0x330 [24.962576] ? __dentry_kill+0x19e/0x1d0 [24.962618] ? call_rcu+0x18f/0x270 [24.962660] ? __dentry_kill+0x19e/0x1d0 [24.962702] ? __x64_sys_sendmsg+0x70/0x90 [24.962744] ? do_syscall_64+0x3d/0x80 [24.962796] ? exit_to_user_mode_prepare+0x1b/0x70 [24.962852] ? entry_SYSCA ---truncated---
CVE-2023-53088	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix UaF in listener shutdown As reported by Christoph after having refactored the passive socket initialization, the mptcp listener shutdown path is prone to an UaF issue. BUG: KASAN: use-after-free in _raw_spin_lock_bh +0x73/0xe0 Write of size 4 at addr ffff88810cb23098 by task syz-executor731/1266 CPU: 1 PID: 1266 Comm: syz-executor731 Not tainted 6.2.0-rc59af4eaa31c1f6c00c8f1e448ed99a45c66340dd5 #6 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x6e/0x91 print_report +0x16a/0x46f kasan_report+0xad/0x130 kasan_check_range+0x14a/0x1a0 _raw_spin_lock_bh +0x73/0xe0 subflow_error_report+0x6d/0x110 sk_error_report+0x3b/0x190 tcp_disconnect +0x138c/0x1aa0 inet_child_forget+0x6f/0x2e0 inet_csk_listen_stop+0x209/0x1060 __mptcp_close_ssk+0x52d/0x610 mptcp_destroy_common+0x165/0x640 mptcp_destroy+0x13/0x80 __mptcp_destroy_sock+0xe7/0x270 __mptcp_close+0x70e/0x9b0 mptcp_close+0x2b/0x150 inet_release+0xe9/0x1f0 __sock_release+0xd2/0x280 sock_close+0x15/0x20 __fput+0x252/0xa20 task_work_run+0x169/0x250 exit_to_user_mode_prepare+0x113/0x120 syscall_exit_to_user_mode +0x1d/0x40 do_syscall_64+0x48/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc The msk grace period can legitimately expire in between the last reference count dropped in mptcp_subflow_queue_clean() and the later eventual access in inet_csk_listen_stop(). After the previous patch we don't need anymore special-casing msk listener socket cleanup: the mptcp worker will process each of the unaccepted msk sockets. Just drop the now unnecessary code. Please note this commit depends on the two parent ones: mptcp: refactor passive socket initialization mptcp: use the workqueue to destroy unaccepted sockets

CVE ID	Description
CVE-2023-53089	In the Linux kernel, the following vulnerability has been resolved: ext4: fix task hung in ext4_xattr_delete_inode Syzbot reported a hung task problem: <pre>===== INFO: task syz-executor232:5073 blocked for more than 143 seconds. Not tainted 6.2.0-rc2-syzkaller-00024-g512dee0c00ad #0 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:syz-exec232 state:D stack:21024 pid:5073 ppid:5072 flags:0x00004004 Call Trace: <TASK> context_switch kernel/sched/core.c:5244 [inline] __schedule+0x995/0xe20 kernel/sched/core.c:6555 schedule+0xcb/0x190 kernel/sched/core.c:6631 __wait_on_freeing_inode fs/inode.c:2196 [inline] find_inode_fast+0x35a/0x4c0 fs/inode.c:950 iget_locked+0xb1/0x830 fs/inode.c:1273 __ext4_iget+0x22e/0x3ed0 fs/ext4/inode.c:4861 ext4_xattr_inode_iget +0x68/0x4e0 fs/ext4/xattr.c:389 ext4_xattr_inode_dec_ref_all+0x1a7/0xe50 fs/ext4/xattr.c:1148 ext4_xattr_delete_inode+0xb04/0xcd0 fs/ext4/xattr.c:2880 ext4_evict_inode+0xd7c/0x10b0 fs/ext4/inode.c:296 evict+0x2a4/0x620 fs/inode.c:664 ext4_orphan_cleanup+0xb60/0x1340 fs/ext4/orphan.c:474 __ext4_fill_super fs/ext4/super.c:5516 [inline] ext4_fill_super+0x81cd/0x8700 fs/ext4/super.c:5644 get_tree_bdev+0x400/0x620 fs/super.c:1282 vfs_get_tree+0x88/0x270 fs/super.c:1489 do_new_mount+0x289/0xad0 fs/namespace.c:3145 do_mount fs/namespace.c:3488 [inline] __do_sys_mount fs/namespace.c:3697 [inline] __se_sys_mount+0x2d3/0x3c0 fs/namespace.c:3674 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7fa5406fd5ea RSP: 002b:00007ffc7232f968 EFLAGS: 00000202 ORIG_RAX: 00000000000000a5 RAX: ffffffffda RBX: 0000000000000003 RCX: 00007fa5406fd5ea RDX: 0000000020000440 RSI: 0000000020000000 RDI: 00007ffc7232f970 RBP: 00007ffc7232f970 R08: 00007ffc7232f9b0 R09: 0000000000000432 R10: 0000000000804a03 R11: 0000000000000202 R12: 0000000000000004 R13: 0000555556a7a2c0 R14: 00007ffc7232f9b0 R15: 0000000000000000 </TASK> =====</pre> The problem is that the inode contains an xattr entry with ea_inum of 15 when cleaning up an orphan inode <15>. When evict inode <15>, the reference counting of the corresponding EA inode is decreased. When EA inode <15> is found by find_inode_fast() in __ext4_iget(), it is found that the EA inode holds the I_FREEING flag and waits for the EA inode to complete deletion. As a result, when inode <15> is being deleted, we wait for inode <15> to complete the deletion, resulting in an infinite loop and triggering Hung Task. To solve this problem, we only need to check whether the ino of EA inode and parent is the same before getting EA inode.
CVE-2023-53091	In the Linux kernel, the following vulnerability has been resolved: ext4: update s_journal_inum if it changes after journal replay When mounting a crafted ext4 image, s_journal_inum may change after journal replay, which is obviously unreasonable because we have successfully loaded and replayed the journal through the old s_journal_inum. And the new s_journal_inum bypasses some of the checks in ext4_get_journal(), which may trigger a null pointer dereference problem. So if s_journal_inum changes after the journal replay, we ignore the change, and rewrite the current journal_inum to the superblock.

CVE ID	Description
CVE-2023-53093	In the Linux kernel, the following vulnerability has been resolved: tracing: Do not let histogram values have some modifiers Histogram values can not be strings, stacktraces, graphs, symbols, syscalls, or grouped in buckets or log. Give an error if a value is set to do so. Note, the histogram code was not prepared to handle these modifiers for histograms and caused a bug. Mark Rutland reported: # echo 'p:copy_to_user __arch_copy_to_user n=\$arg2' >> /sys/kernel/tracing/kprobe_events # echo 'hist:keys=n:vals=hitcount.buckets=8:sort=hitcount' > /sys/kernel/tracing/events/kprobes/copy_to_user/trigger # cat /sys/kernel/tracing/events/kprobes/copy_to_user/hist [143.694628] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 [143.695190] Mem abort info: [143.695362] ESR = 0x0000000096000004 [143.695604] EC = 0x25: DABT (current EL), IL = 32 bits [143.695889] SET = 0, FnV = 0 [143.696077] EA = 0, S1PTW = 0 [143.696302] FSC = 0x04: level 0 translation fault [143.702381] Data abort info: [143.702614] ISV = 0, ISS = 0x00000004 [143.702832] CM = 0, WnR = 0 [143.703087] user pgtable: 4k pages, 48-bit VAs, pgdp=00000000448f9000 [143.703407] [0000000000000000] pgd=0000000000000000, p4d=0000000000000000 [143.704137] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [143.704714] Modules linked in: [143.705273] CPU: 0 PID: 133 Comm: cat Not tainted 6.2.0-00003-g6fc512c10a7c #3 [143.706138] Hardware name: linux,dummy-virt (DT) [143.706723] pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) [143.707120] pc : hist_field_name.part.0+0x14/0x140 [143.707504] lr : hist_field_name.part.0+0x104/0x140 [143.707774] sp : ffff800008333a30 [143.707952] x29: ffff800008333a30 x28: 0000000000000001 x27: 0000000000400cc0 [143.708429] x26: ffff7a653b20260 x25: 0000000000000000 x24: ffff10d303ee5800 [143.708776] x23: ffff7a6539b27b0 x22: ffff10d303fb8c00 x21: 0000000000000001 [143.709127] x20: ffff10d303ec2000 x19: 0000000000000000 x18: 0000000000000000 [143.709478] x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 [143.709824] x14: 0000000000000000 x13: 203a6f666e692072 x12: 6567676972742023 [143.710179] x11: 0a230a6d6172676f x10: 000000000000002c x9 : ffff7a6521e018c [143.710584] x8 : 000000000000002c x7 : 7f7f7f7f7f7f7f x6 : 000000000000002c [143.710915] x5 : ffff10d303b0103e x4 : ffff7a653b20261 x3 : 000000000000003d [143.711239] x2 : 0000000000020001 x1 : 0000000000000001 x0 : 0000000000000000 [143.711746] Call trace: [143.712115] hist_field_name.part.0+0x14/0x140 [143.712642] hist_field_name.part.0+0x104/0x140 [143.712925] hist_field_print+0x28/0x140 [143.713125] event_hist_trigger_print+0x174/0x4d0 [143.713348] hist_show+0xf8/0x980 [143.713521] seq_read_iter+0x1bc/0x4b0 [143.713711] seq_read+0x8c/0xc4 [143.713876] vfs_read+0xc8/0x2a4 [143.714043] ksys_read+0x70/0xfc [143.714218] __arm64_sys_read+0x24/0x30 [143.714400] invoke_syscall+0x50/0x120 [143.714587] el0_svc_common.constprop.0+0x4c/0x100 [143.714807] do_el0_svc+0x44/0xd0 [143.714970] el0_svc+0x2c/0x84 [143.715134] el0t_64_sync_handler+0xbc/0x140 [143.715334] el0t_64_sync+0x190/0x194 [143.715742] Code: a9bd7bfd 910003fd a90153f3 aa0003f3 (f9400000) [143.716510] ---[end trace 0000000000000000]--- Segmentation fault
CVE-2023-53097	In the Linux kernel, the following vulnerability has been resolved: powerpc/iommu: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.

CVE ID	Description
CVE-2023-53098	In the Linux kernel, the following vulnerability has been resolved: media: rc: gpio-ir-recv: add remove function In case runtime PM is enabled, do runtime PM clean up to remove cpu latency qos request, otherwise driver removal may have below kernel dump: [19.463299] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000048 [19.472161] Mem abort info: [19.474985] ESR = 0x0000000096000004 [19.478754] EC = 0x25: DABT (current EL), IL = 32 bits [19.484081] SET = 0, FnV = 0 [19.487149] EA = 0, S1PTW = 0 [19.490361] FSC = 0x04: level 0 translation fault [19.495256] Data abort info: [19.498149] ISV = 0, ISS = 0x00000004 [19.501997] CM = 0, WnR = 0 [19.504977] user pgtbl: 4k pages, 48-bit VAs, pgdp=0000000049f81000 [19.511432] [0000000000000048] pgd=0000000000000000, p4d=0000000000000000 [19.518245] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [19.524520] Modules linked in: gpio_ir_recv(+) rc_core [last unloaded: rc_core] [19.531845] CPU: 0 PID: 445 Comm: insmod Not tainted 6.2.0-rc1-00028-g2c397a46d47c #72 [19.531854] Hardware name: FSL i.MX8MM EVK board (DT) [19.531859] pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYP--) [19.551777] pc : cpu_latency_qos_remove_request+0x20/0x110 [19.557277] lr : gpio_ir_recv_runtime_suspend+0x18/0x30 [gpio_ir_recv] [19.557294] sp : ffff800008ce3740 [19.557297] x29: ffff800008ce3740 x28: 0000000000000000 x27: ffff800008ce3d50 [19.574270] x26: ffff7e3e9cea100 x25: 00000000000f4240 x24: ffff7e3f9ef0e30 [19.574284] x23: 0000000000000000 x22: ffff0061803820f4 x21: 0000000000000008 [19.574296] x20: ffff7e3fa75df30 x19: 0000000000000020 x18: ffffffff [19.588570] x17: 0000000000000000 x16: ffff7e3f9efab70 x15: ffffffff [19.595712] x14: ffff800008ce37b8 x13: ffff800008ce37aa x12: 0000000000000001 [19.602853] x11: 0000000000000001 x10: ffffcb3ec0dff87 x9 : 0000000000000008 [19.609991] x8 : 0101010101010101 x7 : 0000000000000000 x6 : 000000000f0bfe9f [19.624261] x5 : 00ffffff [19.624261] x4 : 0025ab8e00000000 x3 : ffff006180382010 [19.631405] x2 : ffff7e3e9ce8030 x1 : ffff7e3fc3eb810 x0 : 0000000000000020 [19.638548] Call trace: [19.640995] cpu_latency_qos_remove_request+0x20/0x110 [19.646142] gpio_ir_recv_runtime_suspend+0x18/0x30 [gpio_ir_recv] [19.652339] pm_generic_runtime_suspend+0x2c/0x44 [19.657055] __rpm_callback+0x48/0x1dc [19.660807] rpm_callback+0x6c/0x80 [19.664301] rpm_suspend+0x10c/0x640 [19.667880] rpm_idle+0x250/0x2d0 [19.671198] update_autosuspend+0x38/0xe0 [19.675213] pm_runtime_set_autosuspend_delay+0x40/0x60 [19.680442] gpio_ir_recv_probe+0x1b4/0x21c [gpio_ir_recv] [19.685941] platform_probe+0x68/0xc0 [19.689610] really_probe+0xc0/0x3dc [19.693189] __driver_probe_device+0x7c/0x190 [19.697550] driver_probe_device+0x3c/0x110 [19.701739] __driver_attach+0xf4/0x200 [19.705578] bus_for_each_dev+0x70/0xd0 [19.709417] driver_attach+0x24/0x30 [19.712998] bus_add_driver+0x17c/0x240 [19.716834] driver_register+0x78/0x130 [19.720676] __platform_driver_register+0x28/0x34 [19.725386] gpio_ir_recv_driver_init+0x20/0x1000 [gpio_ir_recv] [19.731404] do_one_initcall+0x44/0x2ac [19.735243] do_init_module+0x48/0x1d0 [19.739003] load_module+0x19fc/0x2034 [19.742759] __do_sys_finit_module+0xac/0x12c [19.747124] __arm64_sys_finit_module+0x20/0x30 [19.751664] invoke_syscall+0x48/0x114 [19.755420] el0_svc_common.constprop.0+0xcc/0xec [19.760132] do_el0_svc+0x38/0xb0 [19.763456] el0_svc+0x2c/0x84 [19.766516] el0t_64_sync_handler+0xf4/0x120 [19.770789] el0t_64_sync+0x190/0x194 [19.774460] Code: 910003fd a90153f3 aa0003f3 91204021 (f9401400) [19.780556] ---[end trace 0000000000000000]---

CVE ID	Description
CVE-2023-53100	In the Linux kernel, the following vulnerability has been resolved: ext4: fix WARNING in ext4_update_inline_data Syzbot found the following issue: EXT4-fs (loop0): mounted filesystem 00000000-0000-0000-0000-000000000000 without journal. Quota mode: none. fscrypt: AES-256-CTS-CBC using implementation "cts-cbc-aes-aesni" fscrypt: AES-256-XTS using implementation "xts-aes-aesni" -----[cut here]----- WARNING: CPU: 0 PID: 5071 at mm/page_alloc.c:5525 __alloc_pages+0x30a/0x560 mm/page_alloc.c:5525 Modules linked in: CPU: 1 PID: 5071 Comm: syz-executor263 Not tainted 6.2.0-rc1-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 RIP: 0010: __alloc_pages+0x30a/0x560 mm/page_alloc.c:5525 RSP: 0018: ffff90003c2f1c0 EFLAGS: 00010246 RAX: ffff90003c2f220 RBX: 0000000000000014 RCX: 0000000000000000 RDX: 0000000000000028 RSI: 0000000000000000 RDI: ffff90003c2f248 RBP: ffff90003c2f2d8 R08: dffffc0000000000 R09: ffff90003c2f220 R10: fffff52000785e49 R11: fffff92000785e44 R12: 0000000000040d40 R13: fffff92000785e40 R14: dffffc0000000000 R15: fffff92000785e3c FS: 0000555556c0d300(0000) GS: ffff8880b9800000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f95d5e04138 CR3: 00000000793aa000 CR4: 0000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __alloc_pages_node include/linux/gfp.h:237 [inline] alloc_pages_node include/linux/gfp.h:260 [inline] __kmalloclarge_node+0x95/0x1e0 mm/slab_common.c:1113 __do_kmalloclarge_node mm/slab_common.c:956 [inline] __kmalloclarge_node+0x95/0x1e0 mm/slab_common.c:981 kmalloclarge_node include/linux/slab.h:584 [inline] kzalloc include/linux/slab.h:720 [inline] ext4_update_inline_data+0x236/0x6b0 fs/ext4/inline.c:346 ext4_update_inline_dir fs/ext4/inline.c:1115 [inline] ext4_try_add_inline_entry+0x328/0x990 fs/ext4/inline.c:1307 ext4_add_entry+0x5a4/0xeb0 fs/ext4/namei.c:2385 ext4_add_nondir+0x96/0x260 fs/ext4/namei.c:2772 ext4_create+0x36c/0x560 fs/ext4/namei.c:2817 lookup_open fs/namei.c:3413 [inline] open_last_lookups fs/namei.c:3481 [inline] path_openat+0x12ac/0x2dd0 fs/namei.c:3711 do_filp_open+0x264/0x4f0 fs/namei.c:3741 do_sys_openat2+0x124/0x4e0 fs/open.c:1310 do_sys_open fs/open.c:1326 [inline] __do_sys_openat fs/open.c:1342 [inline] __se_sys_openat fs/open.c:1337 [inline] __x64_sys_openat+0x243/0x290 fs/open.c:1337 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd Above issue happens as follows: ext4_iget ext4_find_inline_data_nolock -> i_inline_off=164 i_inline_size=60 ext4_try_add_inline_entry __ext4_mark_inode_dirty ext4_expand_extra_isize_ea -> i_extra_isize=32 s_want_extra_isize=44 ext4_xattr_shift_entries -> after shift i_inline_off is incorrect, actually is change to 176 ext4_try_add_inline_entry ext4_update_inline_dir get_max_inline_xattr_value_size if (EXT4_I(inode)->i_inline_off) entry = (struct ext4_xattr_entry *)((void *)raw_inode + EXT4_I(inode)->i_inline_off); free += EXT4_XATTR_SIZE(le32_to_cpu(entry->e_value_size)); -> As entry is incorrect, then 'free' may be negative ext4_update_inline_data value = kcalloc(len, GFP_NOFS); -> len is unsigned int, maybe very large, then trigger warning when 'kcalloc()' To resolve the above issue we need to update 'i_inline_off' after 'ext4_xattr_shift_entries()'. We do not need to set EXT4_STATE_MAY_INLINE_DATA flag here, since ext4_mark_inode_dirty() already sets this flag if needed. Setting EXT4_STATE_MAY_INLINE_DATA when it is needed may trigger a BUG_ON in ext4_writepages().
CVE-2023-53101	In the Linux kernel, the following vulnerability has been resolved: ext4: zero i_disksize when initializing the bootloader inode If the boot loader inode has never been used before, the EXT4_IOC_SWAP_BOOT inode will initialize it, including setting the i_size to 0. However, if the "never before used" boot loader has a non-zero i_size, then i_disksize will be non-zero, and the inconsistency between i_size and i_disksize can trigger a kernel warning: WARNING: CPU: 0 PID: 2580 at fs/ext4/file.c:319 CPU: 0 PID: 2580 Comm: bb Not tainted 6.3.0-rc1-00004-g703695902cfa RIP: 0010: ext4_file_write_iter+0x7c7/0xd10 Call Trace: vfs_write+0x3b1/0x5c0 ksys_write+0x77/0x160 __x64_sys_write+0x22/0x30 do_syscall_64+0x39/0x80 Reproducer: 1. create corrupted image and mount it: mke2fs -t ext4 /tmp/foo.img 200 debugfs -wR "sif <5> size 25700" /tmp/foo.img mount -t ext4 /tmp/foo.img /mnt cd /mnt echo 123 > file 2. Run the reproducer program: posix_memalign(&buf, 1024, 1024) fd = open("file", O_RDWR O_DIRECT); ioctl(fd, EXT4_IOC_SWAP_BOOT); write(fd, buf, 1024); Fix this by setting i_disksize as well as i_size to zero when initializing the boot loader inode.

CVE ID	Description
CVE-2023-53105	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix cleanup null-ptr deref on encap lock During module is unloaded while a peer tc flow is still offloaded, first the peer uplink rep profile is changed to a nic profile, and so neigh encap lock is destroyed. Next during unload, the VF reps netdevs are unregistered which causes the original non-peer tc flow to be deleted, which deletes the peer flow. The peer flow deletion detaches the encap entry and try to take the already destroyed encap lock, causing the below trace. Fix this by clearing peer flows during tc eswitch cleanup (mlx5e_tc_esw_cleanup()). Relevant trace: [4316.837128] BUG: kernel NULL pointer dereference, address: 00000000000001d8 [4316.842239] RIP: 0010: __mutex_lock+0xb5/0xc40 [4316.851897] Call Trace: [4316.852481] <TASK> [4316.857214] mlx5e_rep_neigh_entry_release+0x93/0x790 [mlx5_core] [4316.858258] mlx5e_rep_encap_entry_detach+0xa7/0xf0 [mlx5_core] [4316.859134] mlx5e_encap_dealloc+0xa3/0xf0 [mlx5_core] [4316.859867] clean_encap_dests.part.0+0x5c/0xe0 [mlx5_core] [4316.860605] mlx5e_tc_del_fdb_flow+0x32a/0x810 [mlx5_core] [4316.862609] __mlx5e_tc_del_fdb_peer_flow+0x1a2/0x250 [mlx5_core] [4316.863394] mlx5e_tc_del_flow+0x/(0x630 [mlx5_core] [4316.864090] mlx5e_flow_put+0x5f/0x100 [mlx5_core] [4316.864771] mlx5e_delete_flow+0x4de/0xa40 [mlx5_core] [4316.865486] tc_setup_cb_reoffload+0x20/0x80 [4316.865905] fl_reoffload+0x47c/0x510 [cls_flowler] [4316.869181] tcf_block_playback_offloads+0x91/0x1d0 [4316.869649] tcf_block_unbind+0xe7/0x1b0 [4316.870049] tcf_block_offload_cmd.isra.0+0x1ee/0x270 [4316.879266] tcf_block_offload_unbind+0x61/0xa0 [4316.879711] __tcf_block_put+0xa4/0x310
CVE-2023-53108	In the Linux kernel, the following vulnerability has been resolved: net/iucv: Fix size of interrupt data iucv_irq_data needs to be 4 bytes larger. These bytes are not used by the iucv module, but written by the z/VM hypervisor in case a CPU is deconfigured. Reported as: BUG dma-kmalloc-64 (Not tainted): kmalloc Redzone overwritten ----- 0x00000000000400564-0x00000000000400567 @offset=1380. First byte 0x80 instead of 0xcc Allocated in iucv_cpu_prepare+0x44/0xd0 age=167839 cpu=2 pid=1 __kmem_cache_alloc_node+0x166/0x450 kmalloc_node_trace+0x3a/0x70 iucv_cpu_prepare+0x44/0xd0 cpuhp_invoke_callback+0x156/0x2f0 cpuhp_issue_call+0xf0/0x298 __cpuhp_setup_state_cpuslocked+0x136/0x338 __cpuhp_setup_state+0xf4/0x288 iucv_init+0xf4/0x280 do_one_initcall+0x78/0x390 do_initcalls+0x11a/0x140 kernel_init_freeable+0x25e/0x2a0 kernel_init+0x2e/0x170 __ret_from_fork+0x3c/0x58 ret_from_fork+0xa/0x40 Freed in iucv_init+0x92/0x280 age=167839 cpu=2 pid=1 __kmem_cache_free+0x308/0x358 iucv_init+0x92/0x280 do_one_initcall+0x78/0x390 do_initcalls+0x11a/0x140 kernel_init_freeable+0x25e/0x2a0 kernel_init+0x2e/0x170 __ret_from_fork+0x3c/0x58 ret_from_fork+0xa/0x40 Slab 0x0000037200010000 objects=32 used=30 fp=0x00000000000400640 flags=0x1ffff00000010200(slab head node=0 zone=0) Object 0x00000000000400540 @offset=1344 fp=0x0000000000000000 Redzone 00000000000400500: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc Redzone 00000000000400510: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc Redzone 00000000000400520: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc Redzone 00000000000400530: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc Object 00000000000400540: 00 01 00 03 00 00 00 00 00 00 00 00 00 00 00 00 Object 00000000000400550: f3 86 81 f2 f4 82 f8 82 f0 f0 f0 f0 f0 f0 f2 Object 00000000000400560: 00 00 00 00 80 00 00 00 cc cc cc cc cc cc cc cc Object 00000000000400570: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc Redzone 00000000000400580: cc cc cc cc cc cc cc cc Padding 000000000004005d4: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a ZZZZZZZZZZZZZZZZ Padding 000000000004005e4: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a ZZZZZZZZZZZZZZZZ Padding 000000000004005f4: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a ZZZZZZZZZZZZZZZZ CPU: 6 PID: 121030 Comm: 116-pai-crypto. Not tainted 6.3.0-20230221.rc0.git4.99b8246b2d71.300.fc37.s390x+debug #1 Hardware name: IBM 3931 A01 704 (z/VM 7.3.0) Call Trace: [<0000000032aa034ec>] dump_stack_lvl+0xac/0x100 [<00000000329f5a6cc>] check_bytes_and_report+0x104/0x140 [<00000000329f5aa78>] check_object +0x370/0x3c0 [<00000000329f5ede6>] free_debug_processing+0x15e/0x348 [<00000000329f5f06a>] free_to_partial_list+0x9a/0x2f0 [<00000000329f5f4a4>] __slab_free+0x1e4/0x3a8 [<00000000329f61768>] __kmem_cache_free+0x308/0x358 [<0000000032a91465c>] iucv_cpu_dead +0x6c/0x88 [<00000000329c2fc66>] cpuhp_invoke_callback+0x156/0x2f0 [<0000000032aa062da>] _cpu_down.constprop.0+0x22a/0x5e0 [<00000000329c3243e>] cpu_device_down+0x4e/0x78 [<0000000032a61dee0>] device_offline+0xc8/0x118 [<0000000032a61e048>] online_store+0x60/0xe0 [<0000000032a08b6b0>] kernfs_fop_write_iter+0x150/0x1e8 [<00000000329fab65c>] vfs_write +0x174/0x360 [<00000000329fab9fc>] ksys_write+0x74/0x100 [<0000000032aa03a5a>] __do_syscall +0x1da/0x208 [<0000000032aa177b2>] system_call+0x82/0xb0 INFO: lockdep is turned off. FIX dma- kmalloc-64: Restoring kmalloc Redzone 0x00000000000400564-0x00000000000400567=0xcc FIX dma- kmalloc-64: Object at 0x00000000000400540 not freed

CVE ID	Description
CVE-2023-53109	In the Linux kernel, the following vulnerability has been resolved: net: tunnels: annotate lockless accesses to dev->needed_headroom IP tunnels can apparently update dev->needed_headroom in their xmit path. This patch takes care of three tunnels xmit, and also the core LL_RESERVED_SPACE() and LL_RESERVED_SPACE_EXTRA() helpers. More changes might be needed for completeness. BUG: KCSAN: data-race in ip_tunnel_xmit / ip_tunnel_xmit read to 0xffff88815b9da0ec of 2 bytes by task 888 on cpu 1: ip_tunnel_xmit+0x1270/0x1730 net/ipv4/ip_tunnel.c:803 __gre_xmit net/ipv4/ip_gre.c:469 [inline] ipgre_xmit+0x516/0x570 net/ipv4/ip_gre.c:661 __netdev_start_xmit include/linux/netdevice.h:4881 [inline] netdev_start_xmit include/linux/netdevice.h:4895 [inline] xmit_one net/core/dev.c:3580 [inline] dev_hard_start_xmit+0x127/0x400 net/core/dev.c:3596 __dev_queue_xmit+0x1007/0x1eb0 net/core/dev.c:4246 dev_queue_xmit include/linux/netdevice.h:3051 [inline] neigh_direct_output+0x17/0x20 net/core/neighbour.c:1623 neigh_output include/net/neighbour.h:546 [inline] ip_finish_output2+0x740/0x840 net/ipv4/ip_output.c:228 ip_finish_output+0xf4/0x240 net/ipv4/ip_output.c:316 NF_HOOK_COND include/linux/netfilter.h:291 [inline] ip_output+0xe5/0x1b0 net/ipv4/ip_output.c:430 dst_output include/net/dst.h:444 [inline] ip_local_out+0x64/0x80 net/ipv4/ip_output.c:126 iptunnel_xmit+0x34a/0x4b0 net/ipv4/ip_tunnel_core.c:82 ip_tunnel_xmit+0x1451/0x1730 net/ipv4/ip_tunnel.c:813 __gre_xmit net/ipv4/ip_gre.c:469 [inline] ipgre_xmit+0x516/0x570 net/ipv4/ip_gre.c:661 __netdev_start_xmit include/linux/netdevice.h:4881 [inline] netdev_start_xmit include/linux/netdevice.h:4895 [inline] xmit_one net/core/dev.c:3580 [inline] dev_hard_start_xmit+0x127/0x400 net/core/dev.c:3596 __dev_queue_xmit+0x1007/0x1eb0 net/core/dev.c:4246 dev_queue_xmit include/linux/netdevice.h:3051 [inline] neigh_direct_output+0x17/0x20 net/core/neighbour.c:1623 neigh_output include/net/neighbour.h:546 [inline] ip_finish_output2+0x740/0x840 net/ipv4/ip_output.c:228 ip_finish_output+0xf4/0x240 net/ipv4/ip_output.c:316 NF_HOOK_COND include/linux/netfilter.h:291 [inline] ip_output+0xe5/0x1b0 net/ipv4/ip_output.c:430 dst_output include/net/dst.h:444 [inline] ip_local_out+0x64/0x80 net/ipv4/ip_output.c:126 iptunnel_xmit+0x34a/0x4b0 net/ipv4/ip_tunnel_core.c:82 ip_tunnel_xmit+0x1451/0x1730 net/ipv4/ip_tunnel.c:813 __gre_xmit net/ipv4/ip_gre.c:469 [inline] ipgre_xmit+0x516/0x570 net/ipv4/ip_gre.c:661 __netdev_start_xmit include/linux/netdevice.h:4881 [inline] netdev_start_xmit include/linux/netdevice.h:4895 [inline] xmit_one net/core/dev.c:3580 [inline] dev_hard_start_xmit+0x127/0x400 net/core/dev.c:3596 __dev_queue_xmit+0x1007/0x1eb0 net/core/dev.c:4246 dev_queue_xmit include/linux/netdevice.h:3051 [inline] neigh_direct_output+0x17/0x20 net/core/neighbour.c:1623 neigh_output include/net/neighbour.h:546 [inline] ip_finish_output2+0x740/0x840 net/ipv4/ip_output.c:228 ip_finish_output+0xf4/0x240 net/ipv4/ip_output.c:316 NF_HOOK_COND include/linux/netfilter.h:291 [inline] ip_output+0xe5/0x1b0 net/i --- truncated---
CVE-2023-53110	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix NULL sndbuf_desc in smc_cdc_tx_handler() When performing a stress test on SMC-R by rmmmod mlx5_ib driver during the wrk/nginx test, we found that there is a probability of triggering a panic while terminating all link groups. This issue dues to the race between smc_smcr_terminate_all() and smc_buf_create(). smc_smcr_terminate_all smc_buf_create /* init */ conn->sndbuf_desc = NULL; ... smc_lgr_terminate smc_conn_kill smc_close_abort smc_cdc_get_slot_and_msg_send __softirqentry_text_start smc_wr_tx_process_cqe smc_cdc_tx_handler READ(conn->sndbuf_desc->len); /* panic dues to NULL sndbuf_desc */ conn->sndbuf_desc = xxx; This patch tries to fix the issue by always to check the sndbuf_desc before send any cdc msg, to make sure that no null pointer is seen during cqe processing.
CVE-2023-53111	In the Linux kernel, the following vulnerability has been resolved: loop: Fix use-after-free issues do_req_filebacked() calls blk_mq_complete_request() synchronously or asynchronously when using asynchronous I/O unless memory allocation fails. Hence, modify loop_handle_cmd() such that it does not dereference 'cmd' nor 'rq' after do_req_filebacked() finished unless we are sure that the request has not yet been completed. This patch fixes the following kernel crash: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000054 Call trace: css_put.42938+0x1c/0x1ac loop_process_work+0xc8c/0xfd4 loop_rootcg_workfn+0x24/0x34 process_one_work+0x244/0x558 worker_thread+0x400/0x8fc kthread+0x16c/0x1e0 ret_from_fork+0x10/0x20

CVE ID	Description
CVE-2023-53114	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix kernel crash during reboot when adapter is in recovery mode If the driver detects during probe that firmware is in recovery mode then i40e_init_recovery_mode() is called and the rest of probe function is skipped including pci_set_drvdata(). Subsequent i40e_shutdown() called during shutdown/reboot dereferences NULL pointer as pci_get_drvdata() returns NULL. To fix call pci_set_drvdata() also during entering to recovery mode. Reproducer: 1) Lets have i40e NIC with firmware in recovery mode 2) Run reboot Result: [139.084698] i40e: Intel(R) Ethernet Connection XL710 Network Driver [139.090959] i40e: Copyright (c) 2013 - 2019 Intel Corporation. [139.108438] i40e 0000:02:00.0: Firmware recovery mode detected. Limiting functionality. [139.116439] i40e 0000:02:00.0: Refer to the Intel(R) Ethernet Adapters and Devices User Guide for details on firmware recovery mode. [139.129499] i40e 0000:02:00.0: fw 8.3.64775 api 1.13 nvm 8.30 0x8000b78d 1.3106.0 [8086:1583] [15d9:084a] [139.215932] i40e 0000:02:00.0 enp2s0f0: renamed from eth0 [139.223292] i40e 0000:02:00.1: Firmware recovery mode detected. Limiting functionality. [139.231292] i40e 0000:02:00.1: Refer to the Intel(R) Ethernet Adapters and Devices User Guide for details on firmware recovery mode. [139.244406] i40e 0000:02:00.1: fw 8.3.64775 api 1.13 nvm 8.30 0x8000b78d 1.3106.0 [8086:1583] [15d9:084a] [139.329209] i40e 0000:02:00.1 enp2s0f1: renamed from eth0 ... [156.311376] BUG: kernel NULL pointer dereference, address: 00000000000006c2 [156.318330] #PF: supervisor write access in kernel mode [156.323546] #PF: error_code(0x0002) - not-present page [156.328679] PGD 0 P4D 0 [156.331210] Oops: 0002 [#1] PREEMPT SMP NOPTI [156.335567] CPU: 26 PID: 15119 Comm: reboot Tainted: G E 6.2.0+ #1 [156.343126] Hardware name: Abacus electric, s.r.o. - servis@abacus.cz Super Server/H12SSW-iN, BIOS 2.4 04/13/2022 [156.353369] RIP: 0010:i40e_shutdown+0x15/0x130 [i40e] [156.358430] Code: c1 fc ff ff 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa 0f 1f 44 00 00 55 48 89 fd 53 48 8b 9f 48 01 00 00 <f0> 80 8b c2 06 00 00 04 f0 80 8b c0 06 00 00 08 48 8d bb 08 08 00 [156.377168] RSP: 0018:ffffb223c8447d90 EFLAGS: 00010282 [156.382384] RAX: ffffffff073ee70 RBX: 0000000000000000 RCX: 0000000000000001 [156.389510] RDX: 0000000080000001 RSI: 0000000000000246 RDI: ffff95db49988000 [156.396634] RBP: ffff95db49988000 R08: ffffffff00000000 R09: ffffffff8bd17d40 [156.403759] R10: 0000000000000001 R11: ffffffff8a5e3d28 R12: ffff95db49988000 [156.410882] R13: ffffffff89a6fe17 R14: ffff95db49988150 R15: 0000000000000000 [156.418007] FS: 00007fe7c0cc3980(0000) GS:ffff95ea8ee80000(0000) knlGS:0000000000000000 [156.426083] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [156.431819] CR2: 00000000000006c2 CR3: 000000003092fc005 CR4: 0000000000770ee0 [156.438944] PKRU: 55555554 [156.441647] Call Trace: [156.444096] <TASK> [156.446199] pci_device_shutdown+0x38/0x60 [156.450297] device_shutdown+0x163/0x210 [156.454215] kernel_restart+0x12/0x70 [156.457872] __do_sys_reboot+0x1ab/0x230 [156.461789] ? vfs_writev+0xa6/0x1a0 [156.465362] ? __pfx_file_free_rcu+0x10/0x10 [156.469635] ? __call_rcu_common.constprop.85+0x109/0x5a0 [156.475034] do_syscall_64+0x3e/0x90 [156.478611] entry_SYSCALL_64_after_hwframe+0x72/0xdc [156.483658] RIP: 0033:0x7fe7bfb37ab7
CVE-2023-53115	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix memory leaks in mpi3mr_init_ioc() Don't allocate memory again when IOC is being reinitialized.
CVE-2023-53116	In the Linux kernel, the following vulnerability has been resolved: nvmet: avoid potential UAF in nvmet_req_complete() An nvme target ->queue_response() operation implementation may free the request passed as argument. Such implementation potentially could result in a use after free of the request pointer when percpu_ref_put() is called in nvmet_req_complete(). Avoid such problem by using a local variable to save the sq pointer before calling __nvmet_req_complete(), thus avoiding dereferencing the req pointer after that function call.
CVE-2023-53117	In the Linux kernel, the following vulnerability has been resolved: fs: prevent out-of-bounds array speculation when closing a file descriptor Google-Bug-Id: 114199369
CVE-2023-53118	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix a procfs host directory removal regression scsi_proc_hostdir_rm() decreases a reference counter and hence must only be called once per host that is removed. This change does not require a scsi_add_host_with_dma() change since scsi_add_host_with_dma() will return 0 (success) if scsi_proc_host_add() is called.
CVE-2023-53121	In the Linux kernel, the following vulnerability has been resolved: tcp: tcp_make_synack() can be called from process context tcp_rtx_synack() now could be called in process context as explained in 0a375c822497 ("tcp: tcp_rtx_synack() can be called from process context"). tcp_rtx_synack() might call tcp_make_synack(), which will touch per-CPU variables with preemption enabled. This causes the following BUG: BUG: using __this_cpu_add() in preemptible [00000000] code: ThriftIO1/5464 caller is tcp_make_synack+0x841/0xac0 Call Trace: <TASK> dump_stack_lvl +0x10d/0x1a0 check_preemption_disabled+0x104/0x110 tcp_make_synack+0x841/0xac0 tcp_v6_send_synack+0x5c/0x450 tcp_rtx_synack+0xeb/0x1f0 inet_rtx_syn_ack+0x34/0x60 tcp_check_req+0x3af/0x9e0 tcp_rcv_state_process+0x59b/0x2030 tcp_v6_do_rcv+0x5f5/0x700 release_sock+0x3a/0xf0 tcp_sendmsg+0x33/0x40 ____sys_sendmsg+0x2f2/0x490 __sys_sendmsg +0x184/0x230 do_syscall_64+0x3d/0x90 Avoid calling __TCP_INC_STATS() which will touch per-cpu variables. Use TCP_INC_STATS() which is safe to be called from context switch.

CVE ID	Description
CVE-2023-53123	In the Linux kernel, the following vulnerability has been resolved: PCI: s390: Fix use-after-free of PCI resources with per-function hotplug On s390 PCI functions may be hotplugged individually even when they belong to a multi-function device. In particular on an SR-IOV device VFs may be removed and later re-added. In commit a50297cf8235 ("s390/pci: separate zbus creation from scanning") it was missed however that struct pci_bus and struct zpci_bus's resource list retained a reference to the PCI functions MMIO resources even though those resources are released and freed on hot-unplug. These stale resources may subsequently be claimed when the PCI function re-appears resulting in use-after-free. One idea of fixing this use-after-free in s390 specific code that was investigated was to simply keep resources around from the moment a PCI function first appeared until the whole virtual PCI bus created for a multi-function device disappears. The problem with this however is that due to the requirement of artificial MMIO addresses (address cookies) extra logic is then needed to keep the address cookies compatible on re-plug. At the same time the MMIO resources semantically belong to the PCI function so tying their lifecycle to the function seems more logical. Instead a simpler approach is to remove the resources of an individually hot-unplugged PCI function from the PCI bus's resource list while keeping the resources of other PCI functions on the PCI bus untouched. This is done by introducing pci_bus_remove_resource() to remove an individual resource. Similarly the resource also needs to be removed from the struct zpci_bus's resource list. It turns out however, that there is really no need to add the MMIO resources to the struct zpci_bus's resource list at all and instead we can simply use the zpci_bar_struct's resource pointer directly.
CVE-2023-53127	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix expander node leak in mpi3mr_remove() Add a missing resource clean up in .remove.
CVE-2023-53128	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix throttle_groups memory leak Add a missing kfree().
CVE-2023-53131	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: Fix a server shutdown leak Fix a race where kthread_stop() may prevent the threadfn from ever getting called. If that happens the svc_rqst will not be cleaned up.
CVE-2023-53133	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix an infinite loop error when len is 0 in tcp_bpf_recvmmsg_parser() When the buffer length of the recvmmsg system call is 0, we got the following soft lockup problem: watchdog: BUG: soft lockup - CPU#3 stuck for 27s! [a.out:6149] CPU: 3 PID: 6149 Comm: a.out Kdump: loaded Not tainted 6.2.0+ #30 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1 04/01/2014 RIP: 0010:remove_wait_queue+0xb/0xc0 Code: 5e 41 5f c3 cc cc cc 0f 1f 80 00 00 00 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa 0f 1f 44 00 00 41 57 <41> 56 41 55 41 54 55 48 89 fd 53 48 89 f3 4c 8d 6b 18 4c 8d 73 20 RSP: 0018:ffff88811b5978b8 EFLAGS: 00000246 RAX: 0000000000000000 RBX: ffff88811a7d3780 RCX: ffffffff7a4d768 RDX: dffffc0000000000 RSI: ffff88811b597908 RDI: ffff888115408040 RBP: 1ffff110236b2f1b R08: 0000000000000000 R09: ffff88811a7d37e7 R10: ffffd10234fa6fc R11: 0000000000000001 R12: ffff88811179b800 R13: 0000000000000001 R14: ffff88811a7d38a8 R15: ffff88811a7d37e0 FS: 00007f6b5398740(0000) GS:ffff888237180000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000000 CR3: 000000010b6ba002 CR4: 0000000000370ee0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: <TASK> tcp_msg_wait_data+0x279/0x2f0 tcp_bpf_recvmmsg_parser+0x3c6/0x490 inet_recvmmsg+0x280/0x290 sock_recvmmsg+0x4c/0x120 ____sys_recvmmsg+0x160/0x3d0 ____sys_recvmmsg+0xf0/0x180 ____sys_recvmmsg+0xea/0x1a0 do_syscall_64+0x3f/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc The logic in tcp_bpf_recvmmsg_parser is as follows: msg_bytes_ready: copied = sk_msg_recvmmsg(sk, psock, msg, len, flags); if (!copied) { wait data; goto msg_bytes_ready; } In this case, "copied" always is 0, the infinite loop occurs. According to the Linux system call man page, 0 should be returned in this case. Therefore, in tcp_bpf_recvmmsg_parser(), if the length is 0, directly return. Also modify several other functions with the same problem.
CVE-2023-53140	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Remove the /proc/scsi/\${proc_name} directory earlier Remove the /proc/scsi/\${proc_name} directory earlier to fix a race condition between unloading and reloading kernel modules. This fixes a bug introduced in 2009 by commit 77c019768f06 ("[SCSI] fix /proc memory leak in the SCSI core"). Fix the following kernel warning: proc_dir_entry 'scsi/scsi_debug' already registered WARNING: CPU: 19 PID: 27986 at fs/proc/generic.c:376 proc_register+0x27d/0x2e0 Call Trace: proc_mkdir+0xb5/0xe0 scsi_proc_hostdir_add+0xb5/0x170 scsi_host_alloc+0x683/0x6c0 sdebug_driver_probe+0x6b/0x2d0 [scsi_debug] really_probe+0x159/0x540 __driver_probe_device+0xdc/0x230 driver_probe_device+0x4f/0x120 __device_attach_driver+0xef/0x180 bus_for_each_drv+0xe5/0x130 __device_attach+0x127/0x290 device_initial_probe+0x17/0x20 bus_probe_device+0x110/0x130 device_add+0x673/0xc80 device_register+0x1e/0x30 sdebug_add_host_helper+0x1a7/0x3b0 [scsi_debug] scsi_debug_init+0x64f/0x1000 [scsi_debug] do_one_initcall+0xd7/0x470 do_init_module+0xe7/0x330 load_module+0x122a/0x12c0 __do_sys_finit_module+0x124/0x1a0 __x64_sys_finit_module+0x46/0x50 do_syscall_64+0x38/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0

CVE ID	Description
CVE-2023-53142	In the Linux kernel, the following vulnerability has been resolved: ice: copy last block omitted in ice_get_module_eeprom() ice_get_module_eeprom() is broken since commit e9c9692c8a81 ("ice: Reimplement module reads used by ethtool") In this refactor, ice_get_module_eeprom() reads the eeprom in blocks of size 8. But the condition that should protect the buffer overflow ignores the last block. The last block always contains zeros. Bug uncovered by ethtool upstream commit 9538f384b535 ("netlink: eeprom: Defer page requests to individual parsers") After this commit, ethtool reads a block with length = 1; to read the SFF-8024 identifier value. unpatched driver: \$ ethtool -m enp65s0f0np0 offset 0x90 length 8 Offset Values ----- 0x0090: 00 00 00 00 00 00 00 00 \$ ethtool -m enp65s0f0np0 offset 0x90 length 12 Offset Values ----- 0x0090: 00 00 01 a0 4d 65 6c 6c 00 00 00 00 \$ \$ ethtool -m enp65s0f0np0 Offset Values ----- 0x0000: 11 06 06 00 00 00 00 00 00 00 00 00 00 00 00 00 0x0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0x0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 08 00 0x0070: 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 patched driver: \$ ethtool -m enp65s0f0np0 offset 0x90 length 8 Offset Values ----- 0x0090: 00 00 01 a0 4d 65 6c 6c \$ ethtool -m enp65s0f0np0 offset 0x90 length 12 Offset Values ----- 0x0090: 00 00 01 a0 4d 65 6c 6c 61 6e 6f 78 \$ ethtool -m enp65s0f0np0 Identifier : 0x11 (QSFP28) Extended identifier : 0x00 Extended identifier description : 1.5W max. Power consumption Extended identifier description : No CDR in TX, No CDR in RX Extended identifier description : High Power Class (> 3.5 W) not enabled Connector : 0x23 (No separable connector) Transceiver codes : 0x88 0x00 0x00 0x00 0x00 0x00 0x00 0x00 Transceiver type : 40G Ethernet: 40G Base-CR4 Transceiver type : 25G Ethernet: 25G Base-CR CA-N Encoding : 0x05 (64B/66B) BR, Nominal : 25500Mbps Rate identifier : 0x00 Length (SMF,km) : 0km Length (OM3 50um) : 0m Length (OM2 50um) : 0m Length (OM1 62.5um) : 0m Length (Copper or Active cable) : 1m Transmitter technology : 0xa0 (Copper cable unequalized) Attenuation at 2.5GHz : 4db Attenuation at 5.0GHz : 5db Attenuation at 7.0GHz : 7db Attenuation at 12.9GHz : 10db

CVE ID	Description
CVE-2023-53143	In the Linux kernel, the following vulnerability has been resolved: ext4: fix another off-by-one fsmmap error on 1k block filesystems Apparently syzbot figured out that issuing this FSMAP call: struct fsmmap_head cmd = { .fsmh_count = ..., .fsmh_keys = { { .fmr_device = /* ext4 dev */, .fmr_physical = 0, }, { .fmr_device = /* ext4 dev */, .fmr_physical = 0, }, ... }, ret = ioctl(fd, FS_IOC_GETFSMAP, &cmd); Produces this crash if the underlying filesystem is a 1k-block ext4 filesystem: kernel BUG at fs/ext4/ext4.h:3331! invalid opcode: 0000 [#1] PREEMPT SMP CPU: 3 PID: 3227965 Comm: xfs_io Tainted: G W O 6.2.0-rc8-achx Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1 04/01/2014 RIP: 0010:ext4_mb_load_buddy_gfp+0x47c/0x570 [ext4] RSP: 0018:ffffc90007c03998 EFLAGS: 00010246 RAX: ffff888004978000 RBX: ffff90007c03a20 RCX: ffff888041618000 RDX: 0000000000000000 RSI: 00000000000005a4 RDI: ffffffff0c99b11 RBP: ffff888012330000 R08: ffffffff0c2b7d0 R09: 0000000000000040 R10: ffff90007c03950 R11: 0000000000000000 R12: 0000000000000001 R13: 00000000ffffffff R14: 00000000000000c4 R15: ffff88802678c398 FS: 00007fd2020c880(0000) GS:ffff88807e100000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fd318a5fe8 CR3: 000000007f80f001 CR4: 00000000001706e0 Call Trace: <TASK> ext4_mballocc_query_range+0x4b/0x210 [ext4 dfa189daddffe8fecdc3dfd00564e0f265a8ab80] ext4_getfsmmap_datadev+0x713/0x890 [ext4 dfa189daddffe8fecdc3dfd00564e0f265a8ab80] ext4_getfsmmap+0x2b7/0x330 [ext4 dfa189daddffe8fecdc3dfd00564e0f265a8ab80] ext4_ioc_getfsmmap+0x153/0x2b0 [ext4 dfa189daddffe8fecdc3dfd00564e0f265a8ab80] __ext4_ioctl+0x2a7/0x17e0 [ext4 dfa189daddffe8fecdc3dfd00564e0f265a8ab80] __x64_sys_ioctl+0x82/0xa0 do_syscall_64+0x2b/0x80 entry_SYSCALL_64_after_hwframe-0x46/0xb0 RIP: 0033:0x7fd20558aff RSP: 002b:00007fd318a9e30 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff0ffffda RBX: 000000000000200c RCX: 00007fd20558aff RDX: 00007fd1feb2010 RSI: 000000000c0c0583b RDI: 0000000000000003 RBP: 00005625c0634be0 R08: 00005625c0634c40 R09: 0000000000000001 R10: 0000000000000000 R11: 0000000000000246 R12: 00007fd1feb2010 R13: 00005625be70d994 R14: 0000000000000080 R15: 0000000000000000 For GETFSMAP calls, the caller selects a physical block device by writing its block number into fsmmap_head.fmh_keys[0].fmr_device. To query mappings for a subrange of the device, the starting byte of the range is written to fsmmap_head.fmh_keys[0].fmr_physical and the last byte of the range goes in fsmmap_head.fmh_keys[1].fmr_physical. IOWs, to query what mappings overlap with bytes 3-14 of /dev/sda, you'd set the inputs as follows: fmh_keys[0] = { .fmr_device = major(8, 0), .fmr_physical = 3}, fmh_keys[1] = { .fmr_device = major(8, 0), .fmr_physical = 14}, Which would return you whatever is mapped in the 12 bytes starting at physical offset 3. The crash is due to insufficient range validation of keys[1] in ext4_getfsmmap_datadev. On 1k-block filesystems, block 0 is not part of the filesystem, which means that s_first_data_block is nonzero. ext4_get_group_no_and_offset subtracts this quantity from the blocknr argument before cracking it into a group number and a block number within a group. IOWs, block group 0 spans blocks 1-8192 (1-based) instead of 0-8191 (0-based) like what happens with larger block sizes. The net result of this encoding is that blocknr < s_first_data_block is not a valid input to this function. The end_fsb variable is set from the keys that are copied from userspace, which means that in the above example, its value is zero. That leads to an underflow here: blocknr = blocknr - le32_to_cpu(es->s_first_data_block); The division then operates on -1: offset = do_div(blocknr, EXT4_BLOCKS_PER_GROUP(sb)) >> EXT4_SB(sb)->s_cluster_bits; Leaving an impossibly large group number (2^32-1) in blocknr. ext4_getfsmmap_check_keys checked that keys[0] ---truncated---
CVE-2023-53145	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: bttdio: fix use after free bug in bttdio_remove due to race condition In bttdio_probe, the data->work is bound with bttdio_work. It will be started in bttdio_send_frame. If the bttdio_remove runs with a unfinished work, there may be a race condition that hdev is freed but used in bttdio_work. Fix it by canceling the work before do cleanup in bttdio_remove.
CVE-2023-53146	In the Linux kernel, the following vulnerability has been resolved: media: dw2102: Fix null-ptr-deref in dw2102_i2c_transfer() In dw2102_i2c_transfer, msg is controlled by user. When msg[i].buf is null and msg[i].len is zero, former checks on msg[i].buf would be passed. Malicious data finally reach dw2102_i2c_transfer. If accessing msg[i].buf[0] without sanity check, null ptr deref would happen. We add check on msg[i].len to prevent crash. Similar commit: commit 950e252cb469 ("[media] dw2102: limit messages to buffer size")

CVE ID	Description
CVE-2023-53147	In the Linux kernel, the following vulnerability has been resolved: xfrm: add NULL check in xfrm_update_ae_params Normally, x->replay_esn and x->preplay_esn should be allocated at xfrm_alloc_replay_state_esn(...) in xfrm_state_construct(...), hence the xfrm_update_ae_params(...) is okay to update them. However, the current implementation of xfrm_new_ae(...) allows a malicious user to directly dereference a NULL pointer and crash the kernel like below. BUG: kernel NULL pointer dereference, address: 0000000000000000 PGD 8253067 P4D 8253067 PUD 8e0e067 PMD 0 Oops: 0002 [#1] PREEMPT SMP KASAN NOPTI CPU: 0 PID: 98 Comm: poc.npd Not tainted 6.4.0-rc7-00072-gdad9774deaf1 #8 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.o4 RIP: 0010:memcpy_orig+0xad/0x140 Code: e8 4c 89 5f e0 48 8d 7f e0 73 d2 83 c2 20 48 29 d6 48 29 d7 83 fa 10 72 34 4c 8b 06 4c 8b 4e 08 c RSP: 0018:ffff888008f57658 EFLAGS: 00000202 RAX: 0000000000000000 RBX: ffff888008bd0000 RCX: ffffffff8238e571 RDX: 0000000000000018 RSI: ffff888007f64844 RDI: 0000000000000000 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: ffff888008f57818 R13: ffff888007f64aa4 R14: 0000000000000000 R15: 0000000000000000 FS: 00000000014013c0(0000) GS:ffff88806d600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000000054d8000 CR4: 00000000000006f0 Call Trace: <TASK> ? __die+0x1f/0x70 ? page_fault_oops+0x1e8/0x500 ? __pfx_is_prefetch.constprop.0+0x10/0x10 ? __pfx_page_fault_oops+0x10/0x10 ? __raw_spin_unlock_irqrestore+0x11/0x40 ? fixup_exception+0x36/0x460 ? __raw_spin_unlock_irqrestore+0x11/0x40 ? exc_page_fault+0x5e/0xc0 ? asm_exc_page_fault+0x26/0x30 ? xfrm_update_ae_params+0xd1/0x260 ? memcpy_orig+0xad/0x140 ? __pfx_raw_spin_lock_bh+0x10/0x10 xfrm_update_ae_params+0xe7/0x260 xfrm_new_ae+0x298/0x4e0 ? __pfx_xfrm_new_ae+0x10/0x10 ? __pfx_xfrm_new_ae+0x10/0x10 xfrm_user_rcv_msg+0x25a/0x410 ? __pfx_xfrm_user_rcv_msg+0x10/0x10 ? __alloc_skb+0xcf/0x210 ? stack_trace_save+0x90/0xd0 ? filter_irq_stacks+0x1c/0x70 ? __stack_depot_save+0x39/0x4e0 ? __kasan_slab_free+0x10a/0x190 ? kmem_cache_free+0x9c/0x340 ? netlink_recvmmsg+0x23c/0x660 ? sock_recvmmsg+0xeb/0xf0 ? __sys_recvfrom+0x13c/0x1f0 ? __x64_sys_recvfrom+0x71/0x90 ? do_syscall_64+0x3f/0x90 ? entry_SYSCALL_64_after_hwframe+0x72/0xdc ? copyout+0x3e/0x50 netlink_rcv_skb+0xd6/0x210 ? __pfx_xfrm_user_rcv_msg+0x10/0x10 ? __pfx_netlink_rcv_skb+0x10/0x10 ? __pfx_sock_has_perm+0x10/0x10 ? mutex_lock+0x8d/0xe0 ? __pfx_mutex_lock+0x10/0x10 xfrm_netlink_rcv+0x44/0x50 netlink_unicast+0x36f/0x4c0 ? __pfx_netlink_unicast+0x10/0x10 ? netlink_recvmmsg+0x500/0x660 netlink_sendmsg+0x3b7/0x700 This Null-ptr-deref bug is assigned CVE-2023-3772. And this commit adds additional NULL check in xfrm_update_ae_params to fix the NPD.
CVE-2023-53148	In the Linux kernel, the following vulnerability has been resolved: igb: Fix igb_down hung on surprise removal In a setup where a Thunderbolt hub connects to Ethernet and a display through USB Type-C, users may experience a hung task timeout when they remove the cable between the PC and the Thunderbolt hub. This is because the igb_down function is called multiple times when the Thunderbolt hub is unplugged. For example, the igb_io_error_detected triggers the first call, and the igb_remove triggers the second call. The second call to igb_down will block at napi_synchronize. Here's the call trace: __schedule+0x3b0/0xd0 ? __mod_timer+0x164/0x5d3 schedule+0x44/0xa8 schedule_timeout+0xb2/0x2a4 ? run_local_timers+0x4e/0x4e msleep+0x31/0x38 igb_down+0x12c/0x22a [igb 6615058754948bfde0bf01429257eb59f13030d4] __igb_close+0x6f/0x9c [igb 6615058754948bfde0bf01429257eb59f13030d4] igb_close+0x23/0x2b [igb 6615058754948bfde0bf01429257eb59f13030d4] __dev_close_many+0x95/0xec dev_close_many+0x6e/0x103 unregister_netdevice_many+0x105/0x5b1 unregister_netdevice_queue+0xc2/0x10d unregister_netdev+0x1c/0x23 igb_remove+0xa7/0x11c [igb 6615058754948bfde0bf01429257eb59f13030d4] pci_device_remove+0x3f/0x9c device_release_driver_internal+0xfe/0x1b4 pci_stop_bus_device+0x5b/0x7f pci_stop_bus_device+0x30/0x7f pci_stop_bus_device+0x30/0x7f pci_stop_and_remove_bus_device+0x12/0x19 pciehp_unconfigure_device+0x76/0xe9 pciehp_disable_slot+0x6e/0x131 pciehp_handle_presence_or_link_change+0x7a/0x3f7 pciehp_ist+0xbe/0x194 irq_thread_fn+0x22/0x4d ? irq_thread+0x1fd/0x1fd irq_thread+0x17b/0x1fd ? irq_forced_thread_fn+0x5f/0x5f kthread+0x142/0x153 ? __irq_get_irqchip_state+0x46/0x46 ? kthread_associate_blkcg+0x71/0x71 ret_from_fork+0x1f/0x30 In this case, igb_io_error_detected detaches the network interface and requests a PCIE slot reset, however, the PCIE reset callback is not being invoked and thus the Ethernet connection breaks down. As the PCIE error in this case is a non-fatal one, requesting a slot reset can be avoided. This patch fixes the task hung issue and preserves Ethernet connection by ignoring non-fatal PCIE errors.

CVE ID	Description
CVE-2023-53149	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid deadlock in fs reclaim with page writeback Ext4 has a filesystem wide lock protecting ext4_writepages() calls to avoid races with switching of journalled data flag or inode format. This lock can however cause a deadlock like: CPU0 CPU1 ext4_writepages() percpu_down_read(sbi->s_writepages_rwsem); ext4_change_inode_journal_flag() percpu_down_write(sbi->s_writepages_rwsem); - blocks, all readers block from now on ext4_do_writepages() ext4_init_io_end() kmem_cache_zalloc(io_end_cachep, GFP_KERNEL) fs_reclaim frees dentry... dentry_unlink_inode() iput() - last ref => iput_final() - inode dirty => write_inode_now()... ext4_writepages() tries to acquire sbi->s_writepages_rwsem and blocks forever Make sure we cannot recurse into filesystem reclaim from writeback code to avoid the deadlock.
CVE-2023-53150	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Pointer may be dereferenced Klocwork tool reported pointer 'rport' returned from call to function fc_bsg_to_rport() may be NULL and will be dereferenced. Add a fix to validate rport before dereferencing.
CVE-2023-53151	In the Linux kernel, the following vulnerability has been resolved: md/raid10: prevent soft lockup while flush writes Currently, there is no limit for raid1/raid10 plugged bio. While flushing writes, raid1 has cond_resched() while raid10 doesn't, and too many writes can cause soft lockup. Follow up soft lockup can be triggered easily with writeback test for raid10 with ramdisks: watchdog: BUG: soft lockup - CPU#10 stuck for 27s! [md0_raid10:1293] Call Trace: <TASK> call_rcu+0x16/0x20 put_object+0x41/0x80 __delete_object+0x50/0x90 delete_object_full+0x2b/0x40 kmemleak_free+0x46/0xa0 slab_free_freelist_hook.constprop.0+0xed/0x1a0 kmem_cache_free+0xfd/0x300 mempool_free_slab+0x1f/0x30 mempool_free+0x3a/0x100 bio_free+0x59/0x80 bio_put+0xc/0x2c0 free_r10bio+0xbf/0xf0 raid_end_bio_io+0x78/0xb0 one_write_done+0x8a/0xa0 raid10_end_write_request+0x1b4/0x430 bio_endio+0x175/0x320 brd_submit_bio+0x3b9/0x9b7 [brd] __submit_bio+0x69/0xe0 submit_bio_noacct_nocheck+0x1e6/0x5a0 submit_bio_noacct+0x38c/0x7e0 flush_pending_writes+0xf0/0x240 raid10d+0xac/0x1ed0 Fix the problem by adding cond_resched() to raid10 like what raid1 did. Note that unlimited plugged bio still need to be optimized, for example, in the case of lots of dirty pages writeback, this will take lots of memory and io will spend a long time in plug, hence io latency is bad.
CVE-2023-53153	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: Fix use after free for wext Key information in wext.connect is not reset on (re)connect and can hold data from a previous connection. Reset key data to avoid that drivers or mac80211 incorrectly detect a WEP connection request and access the freed or already reused memory. Additionally optimize cfg80211_sme_connect() and avoid an useless schedule of conn_work.
CVE-2023-53165	In the Linux kernel, the following vulnerability has been resolved: udf: Fix uninitialized array access for some pathnames For filenames that begin with . and are between 2 and 5 characters long, UDF charset conversion code would read uninitialized memory in the output buffer. The only practical impact is that the name may be prepended a "unification hash" when it is not actually needed but still it is good to fix this.
CVE-2023-53174	In the Linux kernel, the following vulnerability has been resolved: scsi: core: Fix possible memory leak if device_add() fails If device_add() returns error, the name allocated by dev_set_name() needs be freed. As the comment of device_add() says, put_device() should be used to decrease the reference count in the error path. So fix this by calling put_device(), then the name can be freed in kobject_cleanup().
CVE-2023-53176	In the Linux kernel, the following vulnerability has been resolved: serial: 8250: Reinit port->pm on port specific driver unbind When we unbind a serial port hardware specific 8250 driver, the generic serial8250 driver takes over the port. After that we see an oops about 10 seconds later. This can produce the following at least on some TI SoCs: Unhandled fault: imprecise external abort (0x1406) Internal error: : 1406 [#1] SMP ARM Turns out that we may still have the serial port hardware specific driver port->pm in use, and serial8250_pm() tries to call it after the port specific driver is gone: serial8250_pm [8250_base] from uart_change_pm+0x54/0x8c [serial_base] uart_change_pm [serial_base] from uart_hangup+0x154/0x198 [serial_base] uart_hangup [serial_base] from __tty_hangup.part.0+0x328/0x37c __tty_hangup.part.0 from disassociate_ctty+0x154/0x20c disassociate_ctty from do_exit+0x744/0xaac do_exit from do_group_exit+0x40/0x8c do_group_exit from __wake_up_parent+0x0/0x1c Let's fix the issue by calling serial8250_set_defaults() in serial8250_unregister_port(). This will set the port back to using the serial8250 default functions, and sets the port->pm to point to serial8250_pm.

CVE ID	Description
CVE-2023-53178	In the Linux kernel, the following vulnerability has been resolved: mm: fix zswap writeback race condition The zswap writeback mechanism can cause a race condition resulting in memory corruption, where a swapped out page gets swapped in with data that was written to a different page. The race unfolds like this: 1. a page with data A and swap offset X is stored in zswap 2. page A is removed off the LRU by zpool driver for writeback in zswap-shrink work, data for A is mapped by zpool driver 3. user space program faults and invalidates page entry A, offset X is considered free 4. kswapd stores page B at offset X in zswap (zswap could also be full, if so, page B would then be IOed to X, then skip step 5.) 5. entry A is replaced by B in tree->rbrroot, this doesn't affect the local reference held by zswap-shrink work 6. zswap-shrink work writes back A at X, and frees zswap entry A 7. swapin of slot X brings A in memory instead of B The fix: Once the swap page cache has been allocated (case ZSWAP_SWAPCACHE_NEW), zswap-shrink work just checks that the local zswap_entry reference is still the same as the one in the tree. If it's not the same it means that it's either been invalidated or replaced, in both cases the writeback is aborted because the local entry contains stale data. Reproducer: I originally found this by running `stress` overnight to validate my work on the zswap writeback mechanism, it manifested after hours on my test machine. The key to make it happen is having zswap writebacks, so whatever setup pumps /sys/kernel/debug/zswap/written_back_pages should do the trick. In order to reproduce this faster on a vm, I setup a system with ~100M of available memory and a 500M swap file, then running `stress --vm 1 --vm-bytes 300000000 --vm-stride 4000` makes it happen in matter of tens of minutes. One can speed things up even more by swinging /sys/module/zswap/parameters/max_pool_percent up and down between, say, 20 and 1; this makes it reproduce in tens of seconds. It's crucial to set `--vm-stride` to something other than 4096 otherwise `stress` won't realize that memory has been corrupted because all pages would have the same data.
CVE-2023-53179	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: add the missing IP_SET_HASH_WITH_NET0 macro for ip_set_hash_netportnet.c The missing IP_SET_HASH_WITH_NET0 macro in ip_set_hash_netportnet can lead to the use of wrong `CIDR_POS(c)` for calculating array offsets, which can lead to integer underflow. As a result, it leads to slab out-of-bound access. This patch adds back the IP_SET_HASH_WITH_NET0 macro to ip_set_hash_netportnet to address the issue.
CVE-2023-53181	In the Linux kernel, the following vulnerability has been resolved: dma-buf/dma-resv: Stop leaking on krealloc() failure Currently dma_resv_get_fences() will leak the previously allocated array if the fence iteration got restarted and the krealloc_array() fails. Free the old array by hand, and make sure we still clear the returned *fences so the caller won't end up accessing freed memory. Some (but not all) of the callers of dma_resv_get_fences() seem to still trawl through the array even when dma_resv_get_fences() failed. And let's zero out *num_fences as well for good measure.

CVE ID	Description
CVE-2023-53182	In the Linux kernel, the following vulnerability has been resolved: ACPICA: Avoid undefined behavior: applying zero offset to null pointer ACPICA commit 770653e3ba67c30a629ca7d12e352d83c2541b1e Before this change we see the following UBSAN stack trace in Fuchsia: #0 0x000021e4213b3302 in acpi_ds_init_aml_walk(struct acpi_walk_state*, union acpi_parse_object*, struct acpi_namespace_node*, u8*, u32, struct acpi_evaluate_info*, u8) ../../third_party/acpica/source/components/dispatcher/dswstate.c:682 <platform-bus-x86.so>+0x233302 #1.2 0x000020d0f660777f in ubsan_get_stack_trace() compiler-rt/lib/ubsan/ubsan_diag.cpp:41 <libclang_rt.asan.so>+0x3d77f #1.1 0x000020d0f660777f in maybe_print_stack_trace() compiler-rt/lib/ubsan/ubsan_diag.cpp:51 <libclang_rt.asan.so>+0x3d77f #1 0x000020d0f660777f in ~scoped_report() compiler-rt/lib/ubsan/ubsan_diag.cpp:387 <libclang_rt.asan.so>+0x3d77f #2 0x000020d0f660b96d in handlepointer_overflow_impl() compiler-rt/lib/ubsan/ubsan_handlers.cpp:809 <libclang_rt.asan.so>+0x4196d #3 0x000020d0f660b50d in compiler-rt/lib/ubsan/ubsan_handlers.cpp:815 <libclang_rt.asan.so>+0x4150d #4 0x000021e4213b3302 in acpi_ds_init_aml_walk(struct acpi_walk_state*, union acpi_parse_object*, struct acpi_namespace_node*, u8*, u32, struct acpi_evaluate_info*, u8) ../../third_party/acpica/source/components/dispatcher/dswstate.c:682 <platform-bus-x86.so>+0x233302 #5 0x000021e4213e2369 in acpi_ds_call_control_method(struct acpi_thread_state*, struct acpi_walk_state*, union acpi_parse_object*) ../../third_party/acpica/source/components/dispatcher/dsmethod.c:605 <platform-bus-x86.so>+0x262369 #6 0x000021e421437fac in acpi_ps_parse_aml(struct acpi_walk_state*) ../../third_party/acpica/source/components/parser/psparse.c:550 <platform-bus-x86.so>+0x2b7fac #7 0x000021e4214464d2 in acpi_ps_execute_method(struct acpi_evaluate_info*) ../../third_party/acpica/source/components/parser/psxfac.c:244 <platform-bus-x86.so>+0x2c64d2 #8 0x000021e4213aa052 in acpi_ns_evaluate(struct acpi_evaluate_info*) ../../third_party/acpica/source/components/namespace/nseval.c:250 <platform-bus-x86.so>+0x22a052 #9 0x000021e421413dd8 in acpi_ns_init_one_device(acpi_handle, u32, void*, void**) ../../third_party/acpica/source/components/namespace/nsinit.c:735 <platform-bus-x86.so>+0x293dd8 #10 0x000021e421429e98 in acpi_ns_walk_namespace(acpi_object_type, acpi_handle, u32, u32, acpi_walk_callback, acpi_walk_callback, void*, void**) ../../third_party/acpica/source/components/namespace/nswalk.c:298 <platform-bus-x86.so>+0x2a9e98 #11 0x000021e4214131ac in acpi_ns_initialize_devices(u32) ../../third_party/acpica/source/components/namespace/nsinit.c:268 <platform-bus-x86.so>+0x2931ac #12 0x000021e42147c40d in acpi_initialize_objects(u32) ../../third_party/acpica/source/components/utilities/utxfinit.c:304 <platform-bus-x86.so>+0x2fc40d #13 0x000021e42126d603 in acpi::acpi_impl::initialize_acpi(acpi::acpi_impl*) ../../src/devices/board/lib/acpi/acpi-impl.cc:224 <platform-bus-x86.so>+0xed603 Add a simple check that avoids incrementing a pointer by zero, but otherwise behaves as before. Note that our findings are against ACPICA 20221020, but the same code exists on master.
CVE-2023-53184	In the Linux kernel, the following vulnerability has been resolved: arm64/sme: Set new vector length before reallocating As part of fixing the allocation of the buffer for SVE state when changing SME vector length we introduced an immediate reallocation of the SVE state, this is also done when changing the SVE vector length for consistency. Unfortunately this reallocation is done prior to writing the new vector length to the task struct, meaning the allocation is done with the old vector length and can lead to memory corruption due to an undersized buffer being used. Move the update of the vector length before the allocation to ensure that the new vector length is taken into account. For some reason this isn't triggering any problems when running tests on the arm64 fixes branch (even after repeated tries) but is triggering issues very often after merge into mainline.
CVE-2023-53185	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k: don't allow to overwrite ENDPOINT0 attributes A bad USB device is able to construct a service connection response message with target endpoint being ENDPOINT0 which is reserved for HTC_CTRL_RSVD_SVC and should not be modified to be used for any other services. Reject such service connection responses. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.

CVE ID	Description
CVE-2023-53188	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix race on port output assume the following setup on a single machine: 1. An openvswitch instance with one bridge and default flows 2. two network namespaces "server" and "client" 3. two ovs interfaces "server" and "client" on the bridge 4. for each ovs interface a veth pair with a matching name and 32 rx and tx queues 5. move the ends of the veth pairs to the respective network namespaces 6. assign ip addresses to each of the veth ends in the namespaces (needs to be the same subnet) 7. start some http server on the server network namespace 8. test if a client in the client namespace can reach the http server when following the actions below the host has a chance of getting a cpu stuck in a infinite loop: 1. send a large amount of parallel requests to the http server (around 3000 curls should work) 2. in parallel delete the network namespace (do not delete interfaces or stop the server, just kill the namespace) there is a low chance that this will cause the below kernel cpu stuck message. If this does not happen just retry. Below there is also the output of bpftrace for the functions mentioned in the output. The series of events happening here is: 1. the network namespace is deleted calling `unregister_netdevice_many_notify` somewhere in the process 2. this sets first `NETREG_UNREGISTERING` on both ends of the veth and then runs `synchronize_net` 3. it then calls `call_netdevice_notifiers` with `NETDEV_UNREGISTER` 4. this is then handled by `dp_device_event` which calls `ovs_netdev_detach_dev` (if a vport is found, which is the case for the veth interface attached to ovs) 5. this removes the rx_handlers of the device but does not prevent packages to be sent to the device 6. `dp_device_event` then queues the vport deletion to work in background as a ovs_lock is needed that we do not hold in the unregistration path 7. `unregister_netdevice_many_notify` continues to call `netdev_unregister_kobject` which sets `real_num_tx_queues` to 0 8. port deletion continues (but details are not relevant for this issue) 9. at some future point the background task deletes the vport If after 7. but before 9. a packet is send to the ovs vport (which is not deleted at this point in time) which forwards it to the `dev_queue_xmit` flow even though the device is unregistering. In `skb_tx_hash` (which is called in the `dev_queue_xmit`) path there is a while loop (if the packet has a rx_queue recorded) that is infinite if `dev->real_num_tx_queues` is zero. To prevent this from happening we update `do_output` to handle devices without carrier the same as if the device is not found (which would be the code path after 9. is done). Additionally we now produce a warning in `skb_tx_hash` if we will hit the infinite loop. bpftrace (first word is function name): __dev_queue_xmit server: real_num_tx_queues: 1, cpu: 2, pid: 28024, tid: 28024, skb_addr: 0xffff9edb6f207000, reg_state: 1 netdev_core_pick_tx server: addr: 0xffff9f0a46d4a000 real_num_tx_queues: 1, cpu: 2, pid: 28024, tid: 28024, skb_addr: 0xffff9edb6f207000, reg_state: 1 dp_device_event server: real_num_tx_queues: 1 cpu 9, pid: 21024, tid: 21024, event 2, reg_state: 1 synchronize_rcu_expedited: cpu 9, pid: 21024, tid: 21024 synchronize_rcu_expedited: cpu 9, pid: 21024, tid: 21024 synchronize_rcu_expedited: cpu 9, pid: 21024, tid: 21024 dp_device_event server: real_num_tx_queues: 1 cpu 9, pid: 21024, tid: 21024, event 6, reg_state: 2 ovs_netdev_detach_dev server: real_num_tx_queues: 1 cpu 9, pid: 21024, tid: 21024, reg_state: 2 netdev_rx_handler_unregister server: real_num_tx_queues: 1, cpu: 9, pid: 21024, tid: 21024, reg_state: 2 synchronize_rcu_expedited: cpu 9, pid: 21024, tid: 21024 netdev_rx_handler_unregister ret server: real_num_tx_queues: 1, cpu: 9, pid: 21024, tid: 21024, reg_state: 2 dp_ ---truncated---
CVE-2023-53189	In the Linux kernel, the following vulnerability has been resolved: ipv6/addrconf: fix a potential refcount underflow for idev Now in addrconf_mod_rs_timer(), reference idev depends on whether rs_timer is not pending. Then modify rs_timer timeout. There is a time gap in [1], during which if the pending rs_timer becomes not pending. It will miss to hold idev, but the rs_timer is activated. Thus rs_timer callback function addrconf_rs_timer() will be executed and put idev later without holding idev. A refcount underflow issue for idev can be caused by this. if (!timer_pending(&idev->rs_timer)) in6_dev_hold(idev); <-----[1] mod_timer(&idev->rs_timer, jiffies + when); To fix the issue, hold idev if mod_timer() return 0.
CVE-2023-53197	In the Linux kernel, the following vulnerability has been resolved: USB: uhci: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.

CVE ID	Description
CVE-2023-53198	In the Linux kernel, the following vulnerability has been resolved: raw: Fix NULL deref in raw_get_next(). Dae R. Jeong reported a NULL deref in raw_get_next() [0]. It seems that the repro was running these sequences in parallel so that one thread was iterating on a socket that was being freed in another netns. unshare(0x40060200) r0 = syz_open_procs(0x0, &(0x7f0000002080)='net/raw\x00') socket\$inet_icmp_raw(0x2, 0x3, 0x1) pread64(r0, &(0x7f0000000000)='"/10, 0xa, 0x10000000007f) After commit 0daf07e52709 ("raw: convert raw sockets to RCU"), we use RCU and hlist_nulls_for_each_entry() to iterate over SOCK_RAW sockets. However, we should use spinlock for slow paths to avoid the NULL deref. Also, SOCK_RAW does not use SLAB_TYPESAFE_BY_RCU, and the slab object is not reused during iteration in the grace period. In fact, the lockless readers do not check the nulls marker with get_nulls_value(). So, SOCK_RAW should use hlist instead of hlist_nulls. Instead of adding an unnecessary barrier by sk_nulls_for_each_rcu(), let's convert hlist_nulls to hlist and use sk_for_each_rcu() for fast paths and sk_for_each() and spinlock for /proc/net/raw. [0]: general protection fault, probably for non-canonical address 0xdffffc0000000005: 0000 [#1] PREEMPT SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000028-0x000000000000002f] CPU: 2 PID: 20952 Comm: syz-executor.0 Not tainted 6.2.0-g048ec869bafd-dirty #7 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014 RIP: 0010:read_pnet include/net/net_namespace.h:383 [inline] RIP: 0010:sock_net include/net/sock.h:649 [inline] RIP: 0010:raw_get_next net/ipv4/raw.c:974 [inline] RIP: 0010:raw_get_idx net/ipv4/raw.c:986 [inline] RIP: 0010:raw_seq_start+0x431/0x800 net/ipv4/raw.c:995 Code: ef e8 33 3d 94 f7 49 8b 6d 00 4c 89 ef e8 b7 65 f7 49 89 ed 49 83 c5 98 0f 84 9a 00 00 00 48 83 c5 c8 48 89 e8 48 c1 e8 03 <42> 80 3c 30 00 74 08 48 89 ef e8 00 3d 94 f7 4c 8b 7d 00 48 89 ef RSP: 0018:ffffc9001154f9b0 EFLAGS: 00010206 RAX: 0000000000000005 RBX: 1ffff1100302c8fd RCX: 0000000000000000 RDX: 0000000000000028 RSI: fffffc9001154f988 RDI: fffffc9000f77a338 RBP: 0000000000000029 R08: ffffff8a50ffb4 R09: fffffbfff24b6bd9 R10: fffffbfff24b6bd9 R11: 0000000000000000 R12: ffff88801db73b78 R13: ffffffff9 R14: dffffc0000000000 R15: 0000000000000030 FS: 00007f843ae8e700(0000) GS:ffff888063700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055bb9614b35f CR3: 000000003c672000 CR4: 00000000003506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> seq_read_iter+0x4c6/0x10f0 fs/seq_file.c:225 seq_read+0x224/0x320 fs/seq_file.c:162 pde_read fs/proc/inode.c:316 [inline] proc_reg_read+0x23f/0x330 fs/proc/inode.c:328 vfs_read+0x31e/0xd30 fs/read_write.c:468 ksys_pread64 fs/read_write.c:665 [inline] __do_sys_pread64 fs/read_write.c:675 [inline] __se_sys_pread64 fs/read_write.c:672 [inline] __x64_sys_pread64+0x1e9/0x280 fs/read_write.c:672 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x4e/0xa0 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x478d29 Code: f7 d8 64 89 02 b8 ff ff ff ff c3 66 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 bc ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f843ae8dbe8 EFLAGS: 00000246 ORIG_RAX: 0000000000000011 RAX: ffffffff8da RBX: 0000000000791408 RCX: 0000000000478d29 RDX: 000000000000000a RSI: 0000000020000000 RDI: 0000000000000003 RBP: 00000000f477909a R08: 0000000000000000 R09: 0000000000000000 R10: 000010000000007f R11: 0000000000000246 R12: 0000000000791740 R13: 0000000000791414 R14: 0000000000791408 R15: 00007ffc2eb48a50 </TASK> Modules linked in: --- [end trace 0000000000000000]--- RIP: 0010 ---truncated---
CVE-2023-53199	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k: hif_usb: clean up skbs if ath9k_hif_usb_rx_stream() fails Syzkaller detected a memory leak of skbs in ath9k_hif_usb_rx_stream(). While processing skbs in ath9k_hif_usb_rx_stream(), the already allocated skbs in skb_pool are not freed if ath9k_hif_usb_rx_stream() fails. If we have an incorrect pkt_len or pkt_tag, the input skb is considered invalid and dropped. All the associated packets already in skb_pool should be dropped and freed. Added a comment describing this issue. The patch also makes remain_skb NULL after being processed so that it cannot be referenced after potential free. The initialization of hif_dev fields which are associated with remain_skb (rx_remain_len, rx_transfer_len and rx_pad_len) is moved after a new remain_skb is allocated. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2023-53200	In the Linux kernel, the following vulnerability has been resolved: netfilter: x_tables: fix percpu counter block leak on error path when creating new netns Here is the stack where we allocate percpu counter block: +-< __alloc_percpu +-< xt_percpu_counter_alloc +-< find_check_entry # {arp,ip,ip6}_tables.c +-< translate_table And it can be leaked on this code path: +> ip6t_register_table +> translate_table # allocates percpu counter block +> xt_register_table # fails there is no freeing of the counter block on xt_register_table fail. Note: xt_percpu_counter_free should be called to free it like we do in do_replace through cleanup_entry helper (or in __ip6t_unregister_table). Probability of hitting this error path is low AFAICS (xt_register_table can only return ENOMEM here, as it is not replacing anything, as we are creating new netns, and it is hard to imagine that all previous allocations succeeded and after that one in xt_register_table failed). But it's worth fixing even the rare leak.

CVE ID	Description
CVE-2023-53201	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: wraparound mbox producer index Driver is not handling the wraparound of the mbox producer index correctly. Currently the wraparound happens once u32 max is reached. Bit 31 of the producer index register is special and should be set only once for the first command. Because the producer index overflow setting bit31 after a long time, FW goes to initialization sequence and this causes FW hang. Fix is to wraparound the mbox producer index once it reaches u16 max.
CVE-2023-53202	In the Linux kernel, the following vulnerability has been resolved: PM: domains: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-53204	In the Linux kernel, the following vulnerability has been resolved: af_unix: Fix data-races around user->unix_inflight. user->unix_inflight is changed under spin_lock(unix_gc_lock), but too_many_unix_fds() reads it locklessly. Let's annotate the write/read accesses to user->unix_inflight. BUG: KCSAN: data-race in unix_attach_fds / unix_inflight write to 0xffffffff8546f2d0 of 8 bytes by task 44798 on cpu 1: unix_inflight+0x157/0x180 net/unix/scm.c:66 unix_attach_fds+0x147/0x1e0 net/unix/scm.c:123 unix_scm_to_skb net/unix/af_unix.c:1827 [inline] unix_dgram_sendmsg +0x46a/0x14f0 net/unix/af_unix.c:1950 unix_seqpacket_sendmsg net/unix/af_unix.c:2308 [inline] unix_seqpacket_sendmsg+0xba/0x130 net/unix/af_unix.c:2292 sock_sendmsg_nosec net/socket.c:725 [inline] sock_sendmsg+0x148/0x160 net/socket.c:748 ____sys_sendmsg+0x4e4/0x610 net/socket.c:2494 ____sys_sendmsg+0xc6/0x140 net/socket.c:2548 ____sys_sendmsg+0x94/0x140 net/socket.c:2577 __do_sys_sendmsg net/socket.c:2586 [inline] __se_sys_sendmsg net/socket.c:2584 [inline] __x64_sys_sendmsg+0x45/0x50 net/socket.c:2584 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3b/0x90 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe +0x6e/0xd8 read to 0xffffffff8546f2d0 of 8 bytes by task 44814 on cpu 0: too_many_unix_fds net/unix/scm.c:101 [inline] unix_attach_fds+0x54/0x1e0 net/unix/scm.c:110 unix_scm_to_skb net/unix/af_unix.c:1827 [inline] unix_dgram_sendmsg+0x46a/0x14f0 net/unix/af_unix.c:1950 unix_seqpacket_sendmsg net/unix/af_unix.c:2308 [inline] unix_seqpacket_sendmsg+0xba/0x130 net/unix/af_unix.c:2292 sock_sendmsg_nosec net/socket.c:725 [inline] sock_sendmsg+0x148/0x160 net/socket.c:748 ____sys_sendmsg+0x4e4/0x610 net/socket.c:2494 ____sys_sendmsg+0xc6/0x140 net/socket.c:2548 ____sys_sendmsg+0x94/0x140 net/socket.c:2577 __do_sys_sendmsg net/socket.c:2586 [inline] __se_sys_sendmsg net/socket.c:2584 [inline] __x64_sys_sendmsg+0x45/0x50 net/socket.c:2584 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3b/0x90 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x6e/0xd8 value changed: 0x000000000000000c -> 0x000000000000000d Reported by Kernel Concurrency Sanitizer on: CPU: 0 PID: 44814 Comm: systemd-coredum Not tainted 6.4.0-11989-g6843306689af #6 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014
CVE-2023-53205	In the Linux kernel, the following vulnerability has been resolved: KVM: s390/diag: fix racy access of physical cpu number in diag 9c handler We do check for target CPU == -1, but this might change at the time we are going to use it. Hold the physical target CPU in a local variable to avoid out-of-bound accesses to the cpu arrays.
CVE-2023-53208	In the Linux kernel, the following vulnerability has been resolved: KVM: nSVM: Load L1's TSC multiplier based on L1 state, not L2 state When emulating nested VM-Exit, load L1's TSC multiplier if L1's desired ratio doesn't match the current ratio, not if the ratio L1 is using for L2 diverges from the default. Functionally, the end result is the same as KVM will run L2 with L1's multiplier if L2's multiplier is the default, i.e. checking that L1's multiplier is loaded is equivalent to checking if L2 has a non-default multiplier. However, the assertion that TSC scaling is exposed to L1 is flawed, as userspace can trigger the WARN at will by writing the MSR and then updating guest CPUID to hide the feature (modifying guest CPUID is allowed anytime before KVM_RUN). E.g. hacking KVM's state_test selftest to do vcpu_set_msr(vcpu, MSR_AMD64_TSC_RATIO, 0); vcpu_clear_cpuid_feature(vcpu, X86_FEATURE_TSCRATEMSR); after restoring state in a new VM+vCPU yields an endless supply of: -----[cut here]----- WARNING: CPU: 10 PID: 206939 at arch/x86/kvm/svm/nested.c:1105 nested_svm_vmexit+0x6af/0x720 [kvm_amd] Call Trace: nested_svm_exit_handled+0x102/0x1f0 [kvm_amd] svm_handle_exit+0xb9/0x180 [kvm_amd] kvm_arch_vcpu_ioctl_run+0x1eab/0x2570 [kvm] kvm_vcpu_ioctl+0x4c9/0x5b0 [kvm] ? trace_hardirqs_off+0x4d/0xa0 __se_sys_ioctl+0x7a/0xc0 __x64_sys_ioctl+0x21/0x30 do_syscall_64+0x41/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd Unlike the nested VMRUN path, hoisting the svm->tsc_scaling_enabled check into the if-statement is wrong as KVM needs to ensure L1's multiplier is loaded in the above scenario. Alternatively, the WARN_ON() could simply be deleted, but that would make KVM's behavior even more subtle, e.g. it's not immediately obvious why it's safe to write MSR_AMD64_TSC_RATIO when checking only tsc_ratio_msr.
CVE-2023-53209	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211_hwsim: Fix possible NULL dereference In a call to mac80211_hwsim_select_tx_link() the sta pointer might be NULL, thus need to check that it is not NULL before accessing it.

CVE ID	Description
CVE-2023-53211	In the Linux kernel, the following vulnerability has been resolved: driver core: location: Free struct acpi_pld_info *pld before return false struct acpi_pld_info *pld should be freed before the return of allocation failure, to prevent memory leak, add the ACPI_FREE() to fix it.
CVE-2023-53213	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: slab-out-of-bounds read in brcmf_get_assoc_ies() Fix a slab-out-of-bounds read that occurs in kmemdup() called from brcmf_get_assoc_ies(). The bug could occur when assoc_info->req_len, data from a URB provided by a USB device, is bigger than the size of buffer which is defined as WL_EXTRA_BUF_MAX. Add the size check for req_len/resp_len of assoc_info. Found by a modified version of syzkaller. [46.592467] [T7] ===== [46.594687][T7] BUG: KASAN: slab-out-of-bounds in kmemdup+0x3e/0x50 [46.596572] [T7] Read of size 3014656 at addr ffff888019442000 by task kworker/0:1/7 [46.598575][T7] [46.599157][T7] CPU: 0 PID: 7 Comm: kworker/0:1 Tainted: G O 5.14.0+ #145 [46.601333][T7] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org 04/01/2014 [46.604360][T7] Workqueue: events brcmf_fw_event_worker [46.605943][T7] Call Trace: [46.606584][T7] dump_stack_lvl+0x8e/0xd1 [46.607446] [T7] print_address_description.constprop.0.cold+0x93/0x334 [46.608610][T7] ? kmemdup+0x3e/0x50 [46.609341][T7] kasan_report.cold+0x79/0xd5 [46.610151][T7] ? kmemdup+0x3e/0x50 [46.610796][T7] kasan_check_range+0x14e/0x1b0 [46.611691][T7] memcpy+0x20/0x60 [46.612323][T7] kmemdup+0x3e/0x50 [46.612987][T7] brcmf_get_assoc_ies+0x967/0xf60 [46.613904][T7] ? brcmf_notify_vif_event+0x3d0/0x3d0 [46.614831][T7] ? lock_chain_count+0x20/0x20 [46.615683][T7] ? mark_lock.part.0+0xfc/0x2770 [46.616552][T7] ? lock_chain_count+0x20/0x20 [46.617409][T7] ? mark_lock.part.0+0xfc/0x2770 [46.618244][T7] ? lock_chain_count+0x20/0x20 [46.619024][T7] brcmf_bss_connect_done.constprop.0+0x241/0x2e0 [46.620019][T7] ? brcmf_parse_configure_security.isra.0+0x2a0/0x2a0 [46.620818] [T7] ? __lock_acquire+0x181f/0x5790 [46.621462][T7] brcmf_notify_connect_status+0x448/0x1950 [46.622134][T7] ? rcu_read_lock_bh_held+0xb0/0xb0 [46.622736] [T7] ? brcmf_cfg80211_join_ibss+0x7b0/0x7b0 [46.623390][T7] ? find_held_lock+0x2d/0x110 [46.623962][T7] ? brcmf_fw_event_worker+0x19f/0xc60 [46.624603] [T7] ? mark_held_locks+0x9f/0xe0 [46.625145][T7] ? lockdep_hardirqs_on_prepare+0x3e0/0x3e0 [46.625871][T7] ? brcmf_cfg80211_join_ibss+0x7b0/0x7b0 [46.626545][T7] brcmf_fw_event_handler.isra.0+0x90/0x100 [46.627338][T7] brcmf_fw_event_worker+0x557/0xc60 [46.627962][T7] ? brcmf_fw_event_handler.isra.0+0x100/0x100 [46.628736] [T7] ? rcu_read_lock_sched_held+0xa1/0xd0 [46.629396][T7] ? rcu_read_lock_bh_held+0xb0/0xb0 [46.629970][T7] ? lockdep_hardirqs_on_prepare+0x273/0x3e0 [46.630649][T7] process_one_work+0x92b/0x1460 [46.631205][T7] ? pwq_dec_nr_in_flight+0x330/0x330 [46.631821][T7] ? rwlock_bug.part.0+0x90/0x90 [46.632347][T7] worker_thread+0x95/0xe00 [46.632832][T7] ? __kthread_parkme+0x115/0x1e0 [46.633393][T7] ? process_one_work+0x1460/0x1460 [46.633957] [T7] kthread+0x3a1/0x480 [46.634369][T7] ? set_kthread_struct+0x120/0x120 [46.634933][T7] ret_from_fork+0x1f/0x30 [46.635431][T7] [46.635687][T7] Allocated by task 7: [46.636151] [T7] kasan_save_stack+0x1b/0x40 [46.636628][T7] __kasan_kmalloc+0x7c/0x90 [46.637108][T7] kmem_cache_alloc_trace+0x19e/0x330 [46.637696][T7] brcmf_cfg80211_attach+0x4a0/0x4040 [46.638275][T7] brcmf_attach+0x389/0xd40 [46.638739][T7] brcmf_usb_probe+0x12de/0x1690 [46.639279][T7] usb_probe_interface+0x2aa/0x760 [46.639820][T7] really_probe+0x205/0xb70 [46.640342][T7] __driver_probe_device+0 ---truncated---

CVE ID	Description
CVE-2023-53215	In the Linux kernel, the following vulnerability has been resolved: sched/fair: Don't balance task to its current running CPU We've run into the case that the balancer tries to balance a migration disabled task and trigger the warning in set_task_cpu() like below: -----[cut here]----- WARNING: CPU: 7 PID: 0 at kernel/sched/core.c:3115 set_task_cpu+0x188/0x240 Modules linked in: hclgevf xt_CHECKSUM ipt_REJECT nf_reject_ipv4 <...snip> CPU: 7 PID: 0 Comm: swapper/7 Kdump: loaded Tainted: G O 6.1.0-rc4+ #1 Hardware name: Huawei TaiShan 2280 V2/BC82AMDC, BIOS 2280-V2 CS V5.B221.01 12/09/2021 pstate: 604000c9 (nZCv daIF +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : set_task_cpu+0x188/0x240 lr : load_balance+0x5d0/0xc60 sp : ffff80000803bc70 x29: ffff80000803bc70 x28: ffff004089e190e8 x27: ffff004089e19040 x26: ffff007effcab38 x25: 0000000000000000 x24: 0000000000000001 x23: ffff80000803be84 x22: 000000000000000c x21: ffff093e79e2a78 x20: 000000000000000c x19: ffff004089e19040 x18: 0000000000000000 x17: 00000000000001fad x16: 0000000000000030 x15: 0000000000000000 x14: 0000000000000003 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000001 x10: 0000000000000400 x9 : ffff093e4cee530 x8 : 00000000ffffffe x7 : 0000000000ce168a x6 : 000000000000013e x5 : 00000000fffffe1 x4 : 0000000000000001 x3 : 0000000000000b2a x2 : 0000000000000b2a x1 : ffff093e6d6c510 x0 : 0000000000000001 Call trace: set_task_cpu+0x188/0x240 load_balance+0x5d0/0xc60 rebalance_domains+0x26c/0x380 _nohz_idle_balance.isra.0+0x1e0/0x370 run_rebalance_domains+0x6c/0x80 __do_softirq+0x128/0x3d8 __do_softirq+0x18/0x24 call_on_irq_stack+0x2c/0x38 do_softirq_own_stack+0x24/0x3c __irq_exit_rcu+0xcc/0xf4 irq_exit_rcu+0x18/0x24 el1_interrupt+0x4c/0xe4 el1h_64_irq_handler+0x18/0x2c el1h_64_irq+0x74/0x78 arch_cpu_idle+0x18/0x4c default_idle_call+0x58/0x194 do_idle+0x244/0x2b0 cpu_startup_entry+0x30/0x3c secondary_start_kernel+0x14c/0x190 __secondary_switched+0xb0/0xb4 ---[end trace 0000000000000000]--- Further investigation shows that the warning is superfluous, the migration disabled task is just going to be migrated to its current running CPU. This is because that on load balance if the dst_cpu is not allowed by the task, we'll re-select a new_dst_cpu as a candidate. If no task can be balanced to dst_cpu we'll try to balance the task to the new_dst_cpu instead. In this case when the migration disabled task is not on CPU it only allows to run on its current CPU, load balance will select its current CPU as new_dst_cpu and later triggers the warning above. The new_dst_cpu is chosen from the env->dst_grpmask. Currently it contains CPUs in sched_group_span() and if we have overlapped groups it's possible to run into this case. This patch makes env->dst_grpmask of group_balance_mask() which exclude any CPUs from the busiest group and solve the issue. For balancing in a domain with no overlapped groups the behaviour keeps same as before.
CVE-2023-53220	In the Linux kernel, the following vulnerability has been resolved: media: az6007: Fix null-ptr-deref in az6007_i2c_xfer() In az6007_i2c_xfer, msg is controlled by user. When msg[i].buf is null and msg[i].len is zero, former checks on msg[i].buf would be passed. Malicious data finally reach az6007_i2c_xfer. If accessing msg[i].buf[0] without sanity check, null ptr deref would happen. We add check on msg[i].len to prevent crash. Similar commit: commit 0ed554fd769a ("media: dvb-usb: az6027: fix null-ptr-deref in az6027_i2c_xfer()")
CVE-2023-53221	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix memleak due to fentry attach failure If it fails to attach fentry, the allocated bpf trampoline image will be left in the system. That can be verified by checking /proc/kallsyms. This meamleak can be verified by a simple bpf program as follows: SEC("fentry/trap_init") int fentry_run() { return 0; } It will fail to attach trap_init because this function is freed after kernel init, and then we can find the trampoline image is left in the system by checking /proc/kallsyms. \$ tail /proc/kallsyms ffffffff0613000 t bpf_trampoline_6442453466_1 [bpf] ffffffff06c3000 t bpf_trampoline_6442453466_1 [bpf] \$ bpftool btf dump file /sys/kernel/btf/vmlinux grep "FUNC 'trap_init'" [2522] FUNC 'trap_init' type_id=119 linkage=static \$ echo \$((6442453466 & 0x7fffffff)) 2522 Note that there are two left bpf trampoline images, that is because the libbpf will fallback to raw tracepoint if -EINVAL is returned.
CVE-2023-53224	In the Linux kernel, the following vulnerability has been resolved: ext4: Fix function prototype mismatch for ext4_feat_ktype With clang's kernel control flow integrity (kCFI, CONFIG_CFI_CLANG), indirect call targets are validated against the expected function pointer prototype to make sure the call target is valid to help mitigate ROP attacks. If they are not identical, there is a failure at run time, which manifests as either a kernel panic or thread getting killed. ext4_feat_ktype was setting the "release" handler to "kfree", which doesn't have a matching function prototype. Add a simple wrapper with the correct prototype. This was found as a result of Clang's new -Wcast-function-type-strict flag, which is more sensitive than the simpler -Wcast-function-type, which only checks for type width mismatches. Note that this code is only reached when ext4 is a loadable module and it is being unloaded: CFI failure at kobject_put+0xbb/0x1b0 (target: kfree+0x0/0x180; expected type: 0x7c4aa698) ... RIP: 0010:kobject_put+0xbb/0x1b0 ... Call Trace: <TASK> ext4_exit_sysfs+0x14/0x60 [ext4] cleanup_module+0x67/0xedb [ext4]
CVE-2023-53226	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Fix OOB and integer underflow when rx packets Make sure mwifiex_process_mgmt_packet, mwifiex_process_sta_rx_packet and mwifiex_process_uap_rx_packet, mwifiex_uap_queue_bridged_pkt and mwifiex_process_rx_packet not out-of-bounds access the skb->data buffer.

CVE ID	Description
CVE-2023-53229	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix invalid drv_sta_pre_rcu_remove calls for non-uploaded sta Avoid potential data corruption issues caused by uninitialized driver private data structures.
CVE-2023-53230	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix warning in cifs_smb3_do_mount() This fixes the following warning reported by kernel test robot fs/smb/client/cifsfs.c:982 cifs_smb3_do_mount() warn: possible memory leak of 'cifs_sb'
CVE-2023-53232	In the Linux kernel, the following vulnerability has been resolved: mt76: mt7921: fix kernel panic by accessing unallocated eeprom.data The MT7921 driver no longer uses eeprom.data, but the relevant code has not been removed completely since commit 16d98b548365 ("mt76: mt7921: rely on mcu_get_nic_capability"). This could result in potential invalid memory access. To fix the kernel panic issue in mt7921, it is necessary to avoid accessing unallocated eeprom.data which can lead to invalid memory access. Furthermore, it is possible to entirely eliminate the mt7921_mcu_parse_eeprom function and solely depend on mt7921_mcu_parse_response to divide the RxID header. [2.702735] BUG: kernel NULL pointer dereference, address: 0000000000000550 [2.702740] #PF: supervisor write access in kernel mode [2.702741] #PF: error_code(0x0002) - not-present page [2.702743] PGD 0 P4D 0 [2.702747] Oops: 0002 [#1] PREEMPT SMP NOPTI [2.702755] RIP: 0010:mt7921_mcu_parse_response+0x147/0x170 [mt7921_common] [2.702758] RSP: 0018:ffffae7c00fef828 EFLAGS: 00010286 [2.702760] RAX: ffffa367f57be024 RBX: ffffa367cc7bf500 RCX: 0000000000000000 [2.702762] RDX: 0000000000000550 RSI: 0000000000000000 RDI: ffffa367cc7bf500 [2.702763] RBP: ffffae7c00fef840 R08: ffffa367cb167000 R09: 0000000000000005 [2.702764] R10: 0000000000000000 R11: ffffffffc04702e4 R12: ffffa367e8329f40 [2.702766] R13: 0000000000000000 R14: 0000000000000001 R15: ffffa367e8329f40 [2.702768] FS: 000079ee6cf20c40(0000) GS:ffffa36b2f940000(0000) knlGS:0000000000000000 [2.702769] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [2.702775] CR2: 0000000000000550 CR3: 00000001233c6004 CR4: 0000000000770ee0 [2.702776] PKRU: 55555554 [2.702777] Call Trace: [2.702782] mt76_mcu_skb_send_and_get_msg+0xc3/0x11e [mt76 <HASH:1bc4 5>] [2.702785] mt7921_run_firmware+0x241/0x853 [mt7921_common <HASH:6a2f 6>] [2.702789] mt7921e_mcu_init+0x2b/0x56 [mt7921e <HASH:d290 7>] [2.702792] mt7921_register_device+0x2eb/0x5a5 [mt7921_common <HASH:6a2f 6>] [2.702795] ? mt7921_irq_tasklet+0x1d4/0x1d4 [mt7921e <HASH:d290 7>] [2.702797] mt7921_pci_probe+0x2d6/0x319 [mt7921e <HASH:d290 7>] [2.702799] pci_device_probe+0x9f/0x12a

CVE ID	Description
CVE-2023-53233	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix deadlock triggered by cancel_delayed_work_syn() The following LOCKDEP was detected: Workqueue: events smc_lgr_free_work [smc] WARNING: possible circular locking dependency detected 6.1.0-20221027.rc2.git8.56bc5b569087.300.fc36.s390x+debug #1 Not tainted <pre>----- kworker/3:0/176251 is trying to acquire lock: 00000000f1467148 ((wq_completion)smc_tx_wq-00000000#2){+.+.}-{0:0}, at: __flush_workqueue +0x7a/0x4f0 but task is already holding lock: 0000037fffe97dc8 ((work_completion)((&lgr- >free_work)->work)){+.+.}-{0:0}, at: process_one_work+0x232/0x730 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #4 ((work_completion)((&lgr- >free_work)->work)){+.+.}-{0:0}: __lock_acquire+0x58e/0xbd8 lock_acquire.part.0+0xe2/0x248 lock_acquire+0xac/0x1c8 __flush_work+0x76/0xf0 __cancel_work_timer+0x170/0x220 __smc_lgr_terminate.part.0+0x34/0x1c0 [smc] smc_connect_rdma+0x15e/0x418 [smc] __smc_connect +0x234/0x480 [smc] smc_connect+0x1d6/0x230 [smc] __sys_connect+0x90/0xc0 __do_sys_socketcall +0x186/0x370 __do_syscall+0x1da/0x208 system_call+0x82/0xb0 -> #3 (smc_client_lgr_pending){+. +.}-{3:3}: __lock_acquire+0x58e/0xbd8 lock_acquire.part.0+0xe2/0x248 lock_acquire+0xac/0x1c8 __mutex_lock+0x96/0x8e8 mutex_lock_nested+0x32/0x40 smc_connect_rdma+0xa4/0x418 [smc] __smc_connect+0x234/0x480 [smc] smc_connect+0x1d6/0x230 [smc] __sys_connect+0x90/0xc0 __do_sys_socketcall+0x186/0x370 __do_syscall+0x1da/0x208 system_call+0x82/0xb0 -> #2 (sk_lock- AF_SMC){+.+.}-{0:0}: __lock_acquire+0x58e/0xbd8 lock_acquire.part.0+0xe2/0x248 lock_acquire +0xac/0x1c8 lock_sock_nested+0x46/0xa8 smc_tx_work+0x34/0x50 [smc] process_one_work +0x30c/0x730 worker_thread+0x62/0x420 kthread+0x138/0x150 __ret_from_fork+0x3c/0x58 ret_from_fork+0xa/0x40 -> #1 ((work_completion)((&smc->conn.tx_work)->work)){+.+.}-{ 0:0}: __lock_acquire+0x58e/0xbd8 lock_acquire.part.0+0xe2/0x248 lock_acquire+0xac/0x1c8 process_one_work+0x2bc/0x730 worker_thread+0x62/0x420 kthread+0x138/0x150 __ret_from_fork +0x3c/0x58 ret_from_fork+0xa/0x40 -> #0 ((wq_completion)smc_tx_wq-00000000#2){+.+.}-{ 0:0}: check_prev_add+0xd8/0xe88 validate_chain+0x70c/0xb20 __lock_acquire+0x58e/0xbd8 lock_acquire.part.0+0xe2/0x248 lock_acquire+0xac/0x1c8 __flush_workqueue+0xaa/0x4f0 drain_workqueue+0xaa/0x158 destroy_workqueue+0x44/0x2d8 smc_lgr_free+0x9e/0xf8 [smc] process_one_work+0x30c/0x730 worker_thread+0x62/0x420 kthread+0x138/0x150 __ret_from_fork +0x3c/0x58 ret_from_fork+0xa/0x40 other info that might help us debug this: Chain exists of: (wq_completion)smc_tx_wq-00000000#2 -> smc_client_lgr_pending -> (work_completion)((&lgr- >free_work)->work) Possible unsafe locking scenario: CPU0 CPU1 ---- lock((work_completion) (&(&lgr->free_work)->work)); lock(smc_client_lgr_pending); lock((work_completion)((&(&lgr- >free_work)->work))); lock((wq_completion)smc_tx_wq-00000000#2); *** DEADLOCK *** 2 locks held by kworker/3:0/176251: #0: 0000000080183548 ((wq_completion)events){+.+.}-{0:0}, at: process_one_work+0x232/0x730 #1: 0000037fffe97dc8 ((work_completion)((&lgr->free_work)- >work)){+.+.}-{0:0}, at: process_one_work+0x232/0x730 stack backtr ---truncated---</pre>
CVE-2023-53234	In the Linux kernel, the following vulnerability has been resolved: watchdog: Fix kmemleak in watchdog_cdev_register kmemleak reports memory leaks in watchdog_dev_register, as follows: unreferenced object 0xffff888116233000 (size 2048): comm ""modprobe"", pid 28147, jiffies 4353426116 (age 61.741s) hex dump (first 32 bytes): 80 fa b9 05 81 88 ff ff 08 30 23 16 81 88 ff ff 0#..... 08 30 23 16 81 88 ff ff 00 00 00 00 00 00 00 00 0#..... backtrace: [000000007f001ffd] __kmem_cache_alloc_node+0x157/0x220 [0000000006a389304] kcalloc_trace+0x21/0x110 [0000000008d640eea] watchdog_dev_register+0x4e/0x780 [watchdog] [00000000053c9f248] __watchdog_register_device+0x4f0/0x680 [watchdog] [00000000b2979824] watchdog_register_device+0xd2/0x110 [watchdog] [000000001f730178] 0xfffffffcc10880ae [0000000007a1a8bcc] do_one_initcall+0xcb/0x4d0 [00000000b98be325] do_init_module+0x1ca/0x5f0 [00000000046d08e7c] load_module+0x6133/0x70f0 ... unreferenced object 0xffff8881105b9fa80 (size 16): comm ""modprobe"", pid 28147, jiffies 4353426116 (age 61.741s) hex dump (first 16 bytes): 77 61 74 63 68 64 6f 67 31 00 b9 05 81 88 ff ff watchdog1..... backtrace: [0000000007f001ffd] __kmem_cache_alloc_node+0x157/0x220 [000000000486ab89b] __kmalloct_node_track_caller+0x44/0x1b0 [0000000005a39aab0] kvasprintf+0xb5/0x140 [00000000024806f85] kvasprintf_const+0x55/0x180 [0000000009276cb7f] kobject_set_name_vars+0x56/0x150 [000000000a92e820b] dev_set_name+0xab/0xe0 [000000000cec812c6] watchdog_dev_register+0x285/0x780 [watchdog] [00000000053c9f248] __watchdog_register_device+0x4f0/0x680 [watchdog] [00000000b2979824] watchdog_register_device+0xd2/0x110 [watchdog] [000000001f730178] 0xfffffffcc10880ae [0000000007a1a8bcc] do_one_initcall+0xcb/0x4d0 [00000000b98be325] do_init_module+0x1ca/0x5f0 [00000000046d08e7c] load_module+0x6133/0x70f0 ... The reason is that put_device is not be called if cdev_device_add fails and wdd->id != 0. watchdog_cdev_register wd_data = kzalloc [1] err = dev_set_name [2] .. err = cdev_device_add if (err) { if (wdd->id == 0) { // wdd->id != 0 .. } return err; // [1],[2] would be leaked To fix it, call put_device in all wdd->id cases.

CVE ID	Description
CVE-2023-53235	In the Linux kernel, the following vulnerability has been resolved: drm/tests: helpers: Avoid a driver uaf when using <code>__drm_kunit_helper_alloc_drm_device()</code> the driver may be dereferenced by device-managed resources up until the device is freed, which is typically later than the kunit-managed resource code frees it. Fix this by simply make the driver device-managed as well. In short, the sequence leading to the UAF is as follows: INIT: Code allocates a struct device as a kunit-managed resource. Code allocates a drm driver as a kunit-managed resource. Code allocates a drm device as a device-managed resource. EXIT: Kunit resource cleanup frees the drm driver Kunit resource cleanup puts the struct device, which starts a device-managed resource cleanup device-managed cleanup calls <code>drm_dev_put()</code> <code>drm_dev_put()</code> dereferences the (now freed) drm driver -> Boom. Related KASAN message: [55272.551542] <pre>===== [55272.551551] BUG: KASAN: slab-use-after-free in drm_dev_put.part.0+0xd4/0xe0 [drm] [55272.551603] Read of size 8 at addr ffff888127502828 by task kunit_try_catch/10353 [55272.551612] CPU: 4 PID: 10353 Comm: kunit_try_catch Tainted: G U N 6.5.0-rc7+ #155 [55272.551620] Hardware name: ASUS System Product Name/PRIME B560M-A AC, BIOS 0403 01/26/2021 [55272.551626] Call Trace: [55272.551629] <TASK> [55272.551633] dump_stack_lvl+0x57/0x90 [55272.551639] print_report+0xcf/0x630 [55272.551645] ? _raw_spin_lock_irqsave+0x5f/0x70 [55272.551652] ? drm_dev_put.part.0+0xd4/0xe0 [drm] [55272.551694] kasan_report+0xd7/0x110 [55272.551699] ? drm_dev_put.part.0+0xd4/0xe0 [drm] [55272.551742] drm_dev_put.part.0+0xd4/0xe0 [drm] [55272.551783] devres_release_all +0x15d/0x1f0 [55272.551790] ? __pfx_devres_release_all+0x10/0x10 [55272.551797] device_unbind_cleanup+0x16/0x1a0 [55272.551802] device_release_driver_internal+0x3e5/0x540 [55272.551808] ? kobject_put+0x5d/0x4b0 [55272.551814] bus_remove_device+0x1f1/0x3f0 [55272.551819] device_del+0x342/0x910 [55272.551826] ? __pfx_device_del+0x10/0x10 [55272.551830] ? lock_release+0x339/0x5e0 [55272.551836] ? kunit_remove_resource +0x128/0x290 [kunit] [55272.551845] ? __pfx_lock_release+0x10/0x10 [55272.551851] platform_device_del.part.0+0x1f/0x1e0 [55272.551856] ? _raw_spin_unlock_irqrestore +0x30/0x60 [55272.551863] kunit_remove_resource+0x195/0x290 [kunit] [55272.551871] ? _raw_spin_unlock_irqrestore+0x30/0x60 [55272.551877] kunit_cleanup+0x78/0x120 [kunit] [55272.551885] ? __kthread_parkme+0xc1/0x1f0 [55272.551891] ? __pfx_kunit_try_run_case_cleanup +0x10/0x10 [kunit] [55272.551900] ? __pfx_kunit_generic_run_threadfn_adapter+0x10/0x10 [kunit] [55272.551909] kunit_generic_run_threadfn_adapter+0x4a/0x90 [kunit] [55272.551919] kthread +0x2e7/0x3c0 [55272.551924] ? __pfx_kthread+0x10/0x10 [55272.551929] ret_from_fork+0x2d/0x70 [55272.551935] ? __pfx_kthread+0x10/0x10 [55272.551940] ret_from_fork_asm+0x1b/0x30 [55272.551948] </TASK> [55272.551953] Allocated by task 10351: [55272.551956] kasan_save_stack +0x1c/0x40 [55272.551962] kasan_set_track+0x21/0x30 [55272.551966] __kasan_kmalloc +0x8b/0x90 [55272.551970] __kmalloc+0x5e/0x160 [55272.551976] kunit_kmalloc_array +0x1c/0x50 [kunit] [55272.551984] drm_exec_test_init+0xfa/0x2c0 [drm_exec_test] [55272.551991] kunit_try_run_case+0xdd/0x250 [kunit] [55272.551999] kunit_generic_run_threadfn_adapter +0x4a/0x90 [kunit] [55272.552008] kthread+0x2e7/0x3c0 [55272.552012] ret_from_fork+0x2d/0x70 [55272.552017] ret_from_fork_asm+0x1b/0x30 [55272.552024] Freed by task 10353: [55272.552027] kasan_save_stack+0x1c/0x40 [55272.552032] kasan_set_track+0x21/0x30 [55272.552036] kasan_save_free_info+0x27/0x40 [55272.552041] __kasan_slab_free+0x106/0x180 [55272.552046] slab_free_freelist_hook+0xb3/0x160 [55272.552051] __kmem_cache_free+0xb2/0x290 [55272.552056] kunit_remove_resource+0x195/0x290 [kunit] [55272.552064] kunit_cleanup+0x7 ---truncated---</pre>
CVE-2023-53245	In the Linux kernel, the following vulnerability has been resolved: scsi: storvsc: Fix handling of virtual Fibre Channel timeouts Hyper-V provides the ability to connect Fibre Channel LUNs to the host system and present them in a guest VM as a SCSI device. I/O to the vFC device is handled by the storvsc driver. The storvsc driver includes a partial integration with the FC transport implemented in the generic portion of the Linux SCSI subsystem so that FC attributes can be displayed in /sys. However, the partial integration means that some aspects of vFC don't work properly. Unfortunately, a full and correct integration isn't practical because of limitations in what Hyper-V provides to the guest. In particular, in the context of Hyper-V storvsc, the FC transport timeout function <code>fc_ah_timed_out()</code> causes a kernel panic because it can't find the rport and dereferences a NULL pointer. The original patch that added the call from <code>storvsc_ah_timed_out()</code> to <code>fc_ah_timed_out()</code> is faulty in this regard. In many cases a timeout is due to a transient condition, so the situation can be improved by just continuing to wait like with other I/O requests issued by storvsc, and avoiding the guaranteed panic. For a permanent failure, continuing to wait may result in a hung thread instead of a panic, which again may be better. So fix the panic by removing the storvsc call to <code>fc_ah_timed_out()</code>. This allows storvsc to keep waiting for a response. The change has been tested by users who experienced a panic in <code>fc_ah_timed_out()</code> due to transient timeouts, and it solves their problem. In the future we may want to deprecate the vFC functionality in storvsc since it can't be fully fixed. But it has current users for whom it is working well enough, so it should probably stay for a while longer.
CVE-2023-53251	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: pcie: fix NULL pointer dereference in <code>iwlmwifi_pcie_rx_msix_handler()</code> <code>rxq</code> can be NULL only when <code>trans_pcie->rxq</code> is NULL and <code>entry->entry</code> is zero. For the case when <code>entry->entry</code> is not equal to 0, <code>rxq</code> won't be NULL even if <code>trans_pcie->rxq</code> is NULL. Modify checker to check for <code>trans_pcie->rxq</code>.

CVE ID	Description
CVE-2023-53252	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: use RCU for hci_conn_params and iterate safely in hci_sync hci_update_accept_list_sync iterates over hdev->pend_le_conns and hdev->pend_le_reports, and waits for controller events in the loop body, without holding hdev lock. Meanwhile, these lists and the items may be modified e.g. by le_scan_cleanup. This can invalidate the list cursor or any other item in the list, resulting to invalid behavior (eg use-after-free). Use RCU for the hci_conn_params action lists. Since the loop bodies in hci_sync block and we cannot use RCU or hdev->lock for the whole loop, copy list items first and then iterate on the copy. Only the flags field is written from elsewhere, so READ_ONCE/WRITE_ONCE should guarantee we read valid values. Free params everywhere with hci_conn_params_free so the cleanup is guaranteed to be done properly. This fixes the following, which can be triggered e.g. by BlueZ new mgmt-tester case "Add + Remove Device Nowait - Success", or by changing hci_le_set_cig_params to always return false, and running iso-tester: ===== BUG: KASAN: slab-use-after-free in hci_update_passive_scan_sync (net/bluetooth/hci_sync.c:2536 net/bluetooth/hci_sync.c:2723 net/bluetooth/hci_sync.c:2841) Read of size 8 at addr ffff888001265018 by task kworker/u3:0/32 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.2-1.fc38 04/01/2014 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> dump_stack_lvl (/arch/x86/include/asm/irqflags.h:134 lib/dump_stack.c:107) print_report (mm/kasan/report.c:320 mm/kasan/report.c:430) ? __virt_addr_valid (/include/linux/mmzone.h:1915 ./include/linux/mmzone.h:2011 arch/x86/mm/physaddr.c:65) ? hci_update_passive_scan_sync (net/bluetooth/hci_sync.c:2536 net/bluetooth/hci_sync.c:2723 net/bluetooth/hci_sync.c:2841) kasan_report (mm/kasan/report.c:538) ? hci_update_passive_scan_sync (net/bluetooth/hci_sync.c:2536 net/bluetooth/hci_sync.c:2723 net/bluetooth/hci_sync.c:2841) hci_update_passive_scan_sync (net/bluetooth/hci_sync.c:2536 net/bluetooth/hci_sync.c:2723 net/bluetooth/hci_sync.c:2841) ? __pfx_hci_update_passive_scan_sync (net/bluetooth/hci_sync.c:2780) ? mutex_lock (kernel/locking/mutex.c:282) ? __pfx_mutex_lock (kernel/locking/mutex.c:282) ? __pfx_mutex_unlock (kernel/locking/mutex.c:538) ? __pfx_update_passive_scan_sync (net/bluetooth/hci_sync.c:2861) hci_cmd_sync_work (net/bluetooth/hci_sync.c:306) process_one_work (/arch/x86/include/asm/preempt.h:27 kernel/workqueue.c:2399) worker_thread (/include/linux/list.h:292 kernel/workqueue.c:2538) ? __pfx_worker_thread (kernel/workqueue.c:2480) kthread (kernel/kthread.c:376) ? __pfx_kthread (kernel/kthread.c:331) ret_from_fork (arch/x86/entry/entry_64.S:314) </TASK> Allocated by task 31: kasan_save_stack (mm/kasan/common.c:46) kasan_set_track (mm/kasan/common.c:52) __kasan_kmalloc (mm/kasan/common.c:374 mm/kasan/common.c:383) hci_conn_params_add (/include/linux/slab.h:580 ./include/linux/slab.h:720 net/bluetooth/hci_core.c:2277) hci_connect_le_scan (net/bluetooth/hci_conn.c:1419 net/bluetooth/hci_conn.c:1589) hci_connect_cis (net/bluetooth/hci_conn.c:2266) iso_connect_cis (net/bluetooth/iso.c:390) iso_sock_connect (net/bluetooth/iso.c:899) __sys_connect (net/socket.c:2003 net/socket.c:2020) __x64_sys_connect (net/socket.c:2027) do_syscall_64 (arch/x86/entry/common.c:50 arch/x86/entry/common.c:80) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120) Freed by task 15: kasan_save_stack (mm/kasan/common.c:46) kasan_set_track (mm/kasan/common.c:52) kasan_save_free_info (mm/kasan/generic.c:523) __kasan_slab_free (mm/kasan/common.c:238 mm/kasan/common.c:200 mm/kasan/common.c:244) __kmem_cache_free (mm/slub.c:1807 mm/slub.c:3787 mm/slub.c:3800) hci_conn_params_del (net/bluetooth/hci_core.c:2323) le_scan_cleanup (net/bluetooth/hci_conn.c:202) process_one_work (/arch/x86/include/asm/preempt. --- truncated---
CVE-2023-53254	In the Linux kernel, the following vulnerability has been resolved: cacheinfo: Fix shared_cpu_map to handle shared caches at different levels The cacheinfo sets up the shared_cpu_map by checking whether the caches with the same index are shared between CPUs. However, this will trigger slab-out-of-bounds access if the CPUs do not have the same cache hierarchy. Another problem is the mismatched shared_cpu_map when the shared cache does not have the same index between CPUs. CPU0 I D L3 index 0 1 2 x ^ ^ ^ ^ index 0 1 2 3 CPU1 I D L2 L3 This patch checks each cache is shared with all caches on other CPUs.
CVE-2023-53257	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: check S1G action frame size Before checking the action code, check that it even exists in the frame.
CVE-2023-53259	In the Linux kernel, the following vulnerability has been resolved: VMCI: check context->notify_page after call to get_user_pages_fast() to avoid GPF The call to get_user_pages_fast() in vmci_host_setup_notify() can return NULL context->notify_page causing a GPF. To avoid GPF check if context->notify_page == NULL and return error if so. general protection fault, probably for non-canonical address 0xe0009d1000000060: 0000 [1] PREEMPT SMP KASAN NOPTI KASAN: maybe wild-memory-access in range [0x0005088000000300- 0x0005088000000307] CPU: 2 PID: 26180 Comm: repro_34802241 Not tainted 6.1.0-rc4 #1 Hardware name: Red Hat KVM, BIOS 1.15.0-2.module+el8.6.0 04/01/2014 RIP: 0010:vmci_ctx_check_signal_notify+0x91/0xe0 Call Trace: <TASK> vmci_host_unlocked_ioctl+0x362/0x1f40 __x64_sys_ioctl+0x1a1/0x230 do_syscall_64+0x3a/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd

CVE ID	Description
CVE-2023-53265	In the Linux kernel, the following vulnerability has been resolved: ubi: ensure that VID header offset + VID header size <= alloc, size Ensure that the VID header offset + VID header size does not exceed the allocated area to avoid slab OOB. BUG: KASAN: slab-out-of-bounds in crc32_body lib/crc32.c:111 [inline] BUG: KASAN: slab-out-of-bounds in crc32_le_generic lib/crc32.c:179 [inline] BUG: KASAN: slab-out-of-bounds in crc32_le_base+0x58c/0x626 lib/crc32.c:197 Read of size 4 at addr ffff88802bb36f00 by task syz-executor136/1555 CPU: 2 PID: 1555 Comm: syz-executor136 Tainted: G W 6.0.0-1868 #1 Hardware name: Red Hat KVM, BIOS 1.13.0-2.module+el8.3.0+7860+a7792d29 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x85/0xad lib/dump_stack.c:106 print_address_description mm/kasan/report.c:317 [inline] print_report.cold.13+0xb6/0x6bb mm/kasan/report.c:433 kasan_report+0xa7/0x11b mm/kasan/report.c:495 crc32_body lib/crc32.c:111 [inline] crc32_le_generic lib/crc32.c:179 [inline] crc32_le_base+0x58c/0x626 lib/crc32.c:197 ubi_io_write_vid_hdr+0x1b7/0x472 drivers/mtd/ubi/io.c:1067 create_vtbl+0x4d5/0x9c4 drivers/mtd/ubi/vtbl.c:317 create_empty_lvol drivers/mtd/ubi/vtbl.c:500 [inline] ubi_read_volume_table+0x67b/0x288a drivers/mtd/ubi/vtbl.c:812 ubi_attach+0xf34/0x1603 drivers/mtd/ubi/attach.c:1601 ubi_attach_mtd_dev+0x6f3/0x185e drivers/mtd/ubi/build.c:965 ctrl_cdev_ioctl+0x2db/0x347 drivers/mtd/ubi/cdev.c:1043 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:870 [inline] __se_sys_ioctl fs/ioctl.c:856 [inline] __x64_sys_ioctl+0x193/0x213 fs/ioctl.c:856 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3e/0x86 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0x0 RIP: 0033:0x7f96d5cf753d Code: RSP: 002b:00007fffd72206f8 EFLAGS: 00000246 ORIG_RAX: 000000000000010 RAX: ffffffffda RBX: 0000000000000000 RCX: 00007f96d5cf753d RDX: 0000000020000080 RSI: 0000000040186f40 RDI: 0000000000000003 RBP: 000000000400cd0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 000000000400be0 R13: 00007fffd72207e0 R14: 0000000000000000 R15: 0000000000000000 </TASK> Allocated by task 1555: kasan_save_stack+0x20/0x3d mm/kasan/common.c:38 kasan_set_track mm/kasan/common.c:45 [inline] set_alloc_info mm/kasan/common.c:437 [inline] __kasan_kmalloc mm/kasan/common.c:516 [inline] __kasan_kmalloc+0x88/0xa3 mm/kasan/common.c:525 kasan_kmalloc include/linux/kasan.h:234 [inline] __kmalloc+0x138/0x257 mm/slab.c:4429 kmalloc include/linux/slab.h:605 [inline] ubi_alloc_vid_buf drivers/mtd/ubi/ubi.h:1093 [inline] create_vtbl+0x9cc/0x9c4 drivers/mtd/ubi/vtbl.c:295 create_empty_lvol drivers/mtd/ubi/vtbl.c:500 [inline] ubi_read_volume_table+0x67b/0x288a drivers/mtd/ubi/vtbl.c:812 ubi_attach+0xf34/0x1603 drivers/mtd/ubi/attach.c:1601 ubi_attach_mtd_dev+0x6f3/0x185e drivers/mtd/ubi/build.c:965 ctrl_cdev_ioctl+0x2db/0x347 drivers/mtd/ubi/cdev.c:1043 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:870 [inline] __se_sys_ioctl fs/ioctl.c:856 [inline] __x64_sys_ioctl+0x193/0x213 fs/ioctl.c:856 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3e/0x86 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0x0 The buggy address belongs to the object at ffff88802bb36e00 which belongs to the cache kmalloc-256 of size 256 The buggy address is located 0 bytes to the right of 256-byte region [ffff88802bb36e00, ffff88802bb36f00) The buggy address belongs to the physical page: page:00000000ea4d1263 refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x2bb36 head:00000000ea4d1263 order:1 compound_mapcount:0 compound_pincount:0 flags: 0xffffc0010200(slab)head[node=0]zone=1 lastcpupid=0x1fffff raw: 000ffffc0010200 ffffea000066c300 dead000000000003 ffff888100042b40 raw: 0000000000000000 00000000001 ---truncated---
CVE-2023-53266	In the Linux kernel, the following vulnerability has been resolved: arm64: acpi: Fix possible memory leak of ffh_ctxt Allocated 'ffh_ctxt' memory leak is possible if the SMCCC version and conduit checks fail and -EOPNOTSUPP is returned without freeing the allocated memory. Fix the same by moving the allocation after the SMCCC version and conduit checks.
CVE-2023-53270	In the Linux kernel, the following vulnerability has been resolved: ext4: fix i_dsksize exceeding i_size problem in partially written case It is possible for i_dsksize can exceed i_size, triggering a warning. generic_perform_write copied = iov_iter_copy_from_user_atomic(len) // copied < len ext4_da_write_end ext4_update_i_dsksize new_i_size = pos + copied; WRITE_ONCE(EXT4_I(inode)->i_dsksize, newsize) // update i_dsksize generic_write_end copied = block_write_end(copied, len) // copied = 0 if (unlikely(copied < len)) if (!PageUptodate(page)) copied = 0; if (pos + copied > inode->i_size) // return false if (unlikely(copied == 0)) goto again; if (unlikely(iov_iter_fault_in_readable(i, bytes))) { status = -EFAULT; break; } We get i_dsksize greater than i_size here, which could trigger WARNING check 'i_size_read(inode) < EXT4_I(inode)->i_dsksize' while doing dio: ext4_dio_write_iter iomap_dio_rw __iomap_dio_rw // return err, length is not aligned to 512 ext4_handle_inode_extension WARN_ON_ONCE(i_size_read(inode) < EXT4_I(inode)->i_dsksize) // Oops WARNING: CPU: 2 PID: 2609 at fs/ext4/file.c:319 CPU: 2 PID: 2609 Comm: aa Not tainted 6.3.0-rc2 RIP: 0010:ext4_file_write_iter+0xbc7 Call Trace: vfs_write+0x3b1 ksys_write+0x77 do_syscall_64+0x39 Fix it by updating 'copied' value before updating i_dsksize just like ext4_write_inline_data_end() does. A reproducer can be found in the buganizer link below.

CVE ID	Description
CVE-2023-53271	In the Linux kernel, the following vulnerability has been resolved: ubi: Fix unreferenced object reported by kmemleak in ubi_resize_volume() There is a memory leaks problem reported by kmemleak: unreferenced object 0xffff888102007a00 (size 128): comm "ubirsvol", pid 32090, jiffies 4298464136 (age 2361.231s) hex dump (first 32 bytes): ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff backtrace: [<fffffffb176cec>] __kmalloca+0x4d/0x150 [<fffffffa02a9a36>] ubi_eba_create_table+0x76/0x170 [ubi] [<fffffffa029764e>] ubi_resize_volume+0x1be/0xbc0 [ubi] [<fffffffa02a3321>] ubi_cdev_ioctl+0x701/0x1850 [ubi] [<fffffffb1975d2d>] __x64_sys_ioctl+0x11d/0x170 [<fffffffb83c142a5>] do_syscall_64+0x35/0x80 [<fffffffb83e0006a>] entry_SYSCALL_64_after_hwframe+0x46/0xb0 This is due to a mismatch between create and destroy interfaces, and in detail that "new_eba_tbl" created by ubi_eba_create_table() but destroyed by kfree(), while will causing "new_eba_tbl->entries" not freed. Fix it by replacing kfree(new_eba_tbl) with ubi_eba_destroy_table(new_eba_tbl)
CVE-2023-53272	In the Linux kernel, the following vulnerability has been resolved: net: ena: fix shift-out-of-bounds in exponential backoff The ENA adapters on our instances occasionally reset. Once recently logged a UBSAN failure to console in the process: UBSAN: shift-out-of-bounds in build/linux/drivers/net/ethernet/amazon/ena/ena_com.c:540:13 shift exponent 32 is too large for 32-bit type 'unsigned int' CPU: 28 PID: 70012 Comm: kworker/u72:2 Kdump: loaded not tainted 5.15.117 Hardware name: Amazon EC2 c5d.9xlarge/, BIOS 1.0 10/16/2017 Workqueue: ena ena_fw_reset_device [ena] Call Trace: <TASK> dump_stack_lvl+0xa4/0x63 dump_stack+0x10/0x16 ubsan_epilogue+0x9/0x36 __ubsan_handle_shift_out_of_bounds.cold+0x61/0x10e ? __const_udelay+0x43/0x50 ena_delay_exponential_backoff_us.cold+0x16/0x1e [ena] wait_for_reset_state+0x54/0xa0 [ena] ena_com_dev_reset+0xc8/0x110 [ena] ena_down+0x3fe/0x480 [ena] ena_destroy_device+0xeb/0xf0 [ena] ena_fw_reset_device+0x30/0x50 [ena] process_one_work+0x22b/0x3d0 worker_thread+0x4d/0x3f0 ? process_one_work+0x3d0/0x3d0 kthread+0x12a/0x150 ? set_kthread_struct+0x50/0x50 ret_from_fork+0x22/0x30 </TASK> Apparently, the reset delays are getting so large they can trigger a UBSAN panic. Looking at the code, the current timeout is capped at 5000us. Using a base value of 100us, the current code will overflow after (1<<29). Even at values before 32, this function wraps around, perhaps unintentionally. Cap the value of the exponent used for this backoff at (1<<16) which is larger than currently necessary, but large enough to support bigger values in the future.
CVE-2023-53273	In the Linux kernel, the following vulnerability has been resolved: Drivers: vmbus: Check for channel allocation before looking up relids relid2channel() assumes vmbus channel array to be allocated when called. However, in cases such as kdump/kexec, not all relids will be reset by the host. When the second kernel boots and if the guest receives a vmbus interrupt during vmbus driver initialization before vmbus_connect() is called, before it finishes, or if it fails, the vmbus interrupt service routine is called which in turn calls relid2channel() and can cause a null pointer dereference. Print a warning and error out in relid2channel() for a channel id that's invalid in the second kernel.
CVE-2023-53275	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: fix a possible null-pointer dereference due to data race in snd_hdac_regmap_sync() The variable codec->regmap is often protected by the lock codec->regmap_lock when is accessed. However, it is accessed without holding the lock when is accessed in snd_hdac_regmap_sync(): if (codec->regmap) In my opinion, this may be a harmful race, because if codec->regmap is set to NULL right after the condition is checked, a null-pointer dereference can occur in the called function regcache_sync(): map->lock(map->lock_arg); --> Line 360 in drivers/base/regmap/regcache.c To fix this possible null-pointer dereference caused by data race, the mutex_lock coverage is extended to protect the if statement as well as the function call to regcache_sync(). [Note: the lack of the regmap_lock itself is harmless for the current codec driver implementations, as snd_hdac_regmap_sync() is only for PM runtime resume that is prohibited during the codec probe. But the change makes the whole code more consistent, so it's merged as is -- tiwai]
CVE-2023-53279	In the Linux kernel, the following vulnerability has been resolved: misc: vmw_balloon: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.

CVE ID	Description
CVE-2023-53280	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Remove unused nvme_ls_waitq wait queue System crash when qla2x00_start_sp(sp) returns error code EGAIN and wake_up gets called for uninitialized wait queue sp->nvme_ls_waitq. qla2xxx [0000:37:00.1]-2121:5: Returning existing qpair of ffff8ae2c0513400 for idx=0 qla2xxx [0000:37:00.1]-700e:5: qla2x00_start_sp failed = 11 BUG: unable to handle kernel NULL pointer dereference at 0000000000000000 PGD 0 P4D 0 Oops: 0000 [#1] SMP NOPTI Hardware name: HPE ProLiant DL360 Gen10/ProLiant DL360 Gen10, BIOS U32 09/03/2021 Workqueue: nvme-wq nvme_fc_connect_ctrl_work [nvme_fc] RIP: 0010: __wake_up_common+0x4c/0x190 RSP: 0018:ffff95f3e0cb7cd0 EFLAGS: 00010086 RAX: 0000000000000000 RBX: ffff8b08d3b26328 RCX: 0000000000000000 RDX: 0000000000000001 RSI: 0000000000000003 RDI: ffff8b08d3b26320 RBP: 0000000000000001 R08: 0000000000000000 R09: ffffffff88e8 R10: 0000000000000000 R11: ffff95f3e0cb7a60 R12: ffff95f3e0cb7d20 R13: 0000000000000003 R14: 0000000000000000 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff8b2fd6c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 0000002f1e410002 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: __wake_up_common_lock+0x7c/0xc0 qla_nvme_ls_req+0x355/0x4c0 [qla2xxx] ? __nvme_fc_send_ls_req+0x260/0x380 [nvme_fc] ? nvme_fc_send_ls_req.constprop.42+0x1a/0x45 [nvme_fc] ? nvme_fc_connect_ctrl_work.cold.63+0x1e3/0xa7d [nvme_fc] Remove unused nvme_ls_waitq wait queue. nvme_ls_waitq logic was removed previously in the commits tagged Fixed: below.
CVE-2023-53282	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix use-after-free KFENCE violation during sysfs firmware write During the sysfs firmware write process, a use-after-free read warning is logged from the lpfc_wr_object() routine: BUG: KFENCE: use-after-free read in lpfc_wr_object+0x235/0x310 [lpfc] Use-after-free read at 0x0000000000cf164d (in kfence-#111): lpfc_wr_object+0x235/0x310 [lpfc] lpfc_write_firmware.cold+0x206/0x30d [lpfc] lpfc_sli4_request_firmware_update+0xa6/0x100 [lpfc] lpfc_request_firmware_upgrade_store+0x66/0xb0 [lpfc] kernfs_fop_write_iter+0x121/0x1b0 new_sync_write+0x11c/0x1b0 vfs_write+0x1ef/0x280 ksys_write+0x5f/0xe0 do_syscall_64+0x59/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd The driver accessed wr_object pointer data, which was initialized into mailbox payload memory, after the mailbox object was released back to the mailbox pool. Fix by moving the mailbox free calls to the end of the routine ensuring that we don't reference internal mailbox memory after release.
CVE-2023-53285	In the Linux kernel, the following vulnerability has been resolved: ext4: add bounds checking in get_max_inline_xattr_value_size() Normally the extended attributes in the inode body would have been checked when the inode is first opened, but if someone is writing to the block device while the file system is mounted, it's possible for the inode table to get corrupted. Add bounds checking to avoid reading beyond the end of allocated memory if this happens.

CVE ID	Description
CVE-2023-53286	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Return the firmware result upon destroying QP/RQ Previously when destroying a QP/RQ, the result of the firmware destruction function was ignored and upper layers weren't informed about the failure. Which in turn could lead to various problems since when upper layer isn't aware of the failure it continues its operation thinking that the related QP/RQ was successfully destroyed while it actually wasn't, which could lead to the below kernel WARN. Currently, we return the correct firmware destruction status to upper layers which in case of the RQ would be mlx5_ib_destroy_wq() which was already capable of handling RQ destruction failure or in case of a QP to destroy_qp_common(), which now would actually warn upon qp destruction failure. WARNING: CPU: 3 PID: 995 at drivers/infiniband/core/rdma_core.c:940 uverbs_destroy_ufile_hw+0xcb/0xe0 [ib_uverbs] Modules linked in: xt_contrack xt_MASQUERADE nf_contrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter rpcrdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi rdma_cm ib_umad ib_ipoib iw_cm ib_cm mlx5_ib ib_uverbs ib_core overlay mlx5_core fuse CPU: 3 PID: 995 Comm: python3 Not tainted 5.16.0-rc5+ #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:uverbs_destroy_ufile_hw+0xcb/0xe0 [ib_uverbs] Code: 41 5c 41 5d 41 5e e9 44 34 f0 e0 48 89 df e8 4c 77 ff ff 49 8b 86 10 01 00 00 48 85 c0 74 a1 4c 89 e7 ff d0 eb 9a 0f 0b eb c1 <0f> 0b be 04 00 00 00 48 89 df e8 b6 f6 ff ff e9 75 ff ff 90 0f RSP: 0018:ffff8881533e3e78 EFLAGS: 00010287 RAX: ffff88811b2cf3e0 RBX: ffff888106209700 RCX: 0000000000000000 RDX: ffff888106209780 RSI: ffff8881533e3d30 RDI: ffff888109b101a0 RBP: 0000000000000001 R08: ffff888127cb381c R09: 0de9890000000009 R10: ffff888127cb3800 R11: 0000000000000000 R12: ffff888106209780 R13: ffff888106209750 R14: ffff888100f20660 R15: 0000000000000000 FS: 00007f8be353b740(0000) GS: ffff88852c980000(0000) kmlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f8bd5b117c0 CR3: 000000012cd8a004 CR4: 0000000000370ea0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: <TASK> ib_uverbs_close+0x1a/0x90 [ib_uverbs] __fput+0x82/0x230 task_work_run+0x59/0x90 exit_to_user_mode_prepare+0x138/0x140 syscall_exit_to_user_mode+0x1d/0x50 ? __x64_sys_close+0xe/0x40 do_syscall_64+0x4a/0x90 entry_SYSCALL_64_after_hwframe+0x44/0xae RIP: 0033:0x7f8be3ae0abb Code: 03 00 00 00 0f 05 48 3d 00 f0 ff ff 77 41 c3 48 83 ec 18 89 7c 24 0c e8 83 43 f9 ff 8b 7c 24 0c 41 89 c0 b8 03 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 35 44 89 c7 89 44 24 0c e8 c1 43 f9 ff 8b 44 RSP: 002b:00007f8db51909c0 EFLAGS: 00000293 ORIG_RAX: 0000000000000003 RAX: 0000000000000000 RBX: 0000557bb7f7c020 RCX: 00007f8be3ae0abb RDX: 0000557bb7c74010 RSI: 0000557bb7f14ca0 RDI: 0000000000000005 RBP: 0000557bb7fbd598 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000293 R12: 0000557bb7fbd5b8 R13: 0000557bb7fbd5a8 R14: 0000000000001000 R15: 0000557bb7f7c020 <TASK>
CVE-2023-53288	In the Linux kernel, the following vulnerability has been resolved: drm/client: Fix memory leak in drm_client_modeset_probe When a new mode is set to modeset->mode, the previous mode should be freed. This fixes the following kmemleak report: drm_mode_duplicate+0x45/0x220 [drm] drm_client_modeset_probe+0x944/0xf50 [drm] __drm_fb_helper_initial_config_and_unlock+0xb4/0x2c0 [drm_kms_helper] drm_fbdev_client_hotplug+0x2bc/0x4d0 [drm_kms_helper] drm_client_register+0x169/0x240 [drm] ast_pci_probe+0x142/0x190 [ast] local_pci_probe+0xdc/0x180 work_for_cpu_fn+0x4e/0xa0 process_one_work+0x8b7/0x1540 worker_thread+0x70a/0xed0 kthread+0x29f/0x340 ret_from_fork+0x1f/0x30
CVE-2023-53292	In the Linux kernel, the following vulnerability has been resolved: blk-mq: fix NULL dereference on q->elevator in blk_mq_elv_switch_none After grabbing q->sysfs_lock, q->elevator may become NULL because of elevator switch. Fix the NULL dereference on q->elevator by checking it with lock.
CVE-2023-53293	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btrtl: check for NULL in btrtl_set_quirks() The btrtl_set_quirks() has accessed btrtl_dev->ic_info->imp_subver since b8e482d02513. However, if installing a Realtek Bluetooth controller without the driver supported, it will hit the NULL point accessed. Add a check for NULL to avoid the Kernel Oops.
CVE-2023-53295	In the Linux kernel, the following vulnerability has been resolved: udf: Do not update file length for failed writes to inline files When write to inline file fails (or happens only partly), we still updated length of inline data as if the whole write succeeded. Fix the update of length of inline data to happen only if the write succeeds.
CVE-2023-53297	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: fix "bad unlock balance" in l2cap_disconnect_rsp conn->chan_lock isn't acquired before l2cap_get_chan_by_scid, if l2cap_get_chan_by_scid returns NULL, then 'bad unlock balance' is triggered.

CVE ID	Description
CVE-2023-53307	In the Linux kernel, the following vulnerability has been resolved: rbd: avoid use-after-free in do_rbd_add() when rbd_dev_create() fails If getting an ID or setting up a work queue in rbd_dev_create() fails, use-after-free on rbd_dev->rbd_client, rbd_dev->spec and rbd_dev->opts is triggered in do_rbd_add(). The root cause is that the ownership of these structures is transferred to rbd_dev prematurely and they all end up getting freed when rbd_dev_create() calls rbd_dev_free() prior to returning to do_rbd_add(). Found by Linux Verification Center (linuxtesting.org) with SVACE, an incomplete patch submitted by Natalia Petrova <n.petrova@fintech.ru>.
CVE-2023-53309	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: Fix integer overflow in radeon_cs_parser_init The type of size is unsigned, if size is 0x40000000, there will be an integer overflow, size will be zero after size *= sizeof(uint32_t), will cause uninitialized memory to be referenced later
CVE-2023-53313	In the Linux kernel, the following vulnerability has been resolved: md/raid10: fix wrong setting of max_corr_read_errors There is no input check when echo md/max_read_errors and overflow might occur. Add check of input number.
CVE-2023-53315	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: Fix SKB corruption in REO destination ring While running traffics for a long time, randomly an RX descriptor filled with value "0" from REO destination ring is received. This descriptor which is invalid causes the wrong SKB (SKB stored in the IDR lookup with buffer id "0") to be fetched which in turn causes SKB memory corruption issue and the same leads to crash after some time. Changed the start id for idr allocation to "1" and the buffer id "0" is reserved for error validation. Introduced Sanity check to validate the descriptor, before processing the SKB. Crash Signature : Unable to handle kernel paging request at virtual address 3f004900 PC points to "b15_dma_inv_range+0x30/0x50" LR points to "dma_cache_maint_page+0x8c/0x128". The Backtrace obtained is as follows: [<8031716c>] (b15_dma_inv_range) from [<80313a4c>] (dma_cache_maint_page+0x8c/0x128) [<80313a4c>] (dma_cache_maint_page) from [<80313b90>] (__dma_page_dev_to_cpu+0x28/0x40) [<80313b90>] (__dma_page_dev_to_cpu) from [<7fb5dd68>] (ath11k_dp_process_rx+0x1e8/0x4a4 [ath11k]) [<7fb5dd68>] (ath11k_dp_process_rx [ath11k]) from [<7fb53c20>] (ath11k_dp_service_srng+0xb0/0x2ac [ath11k]) [<7fb53c20>] (ath11k_dp_service_srng [ath11k]) from [<7f67bba4>] (ath11k_pci_ext_grp_napi_poll+0x1c/0x78 [ath11k_pci]) [<7f67bba4>] (ath11k_pci_ext_grp_napi_poll [ath11k_pci]) from [<807d5cf4>] (__napi_poll+0x28/0xb8) [<807d5cf4>] (__napi_poll) from [<807d5f28>] (net_rx_action+0xf0/0x280) [<807d5f28>] (net_rx_action) from [<80302148>] (__do_softirq+0xd0/0x280) [<80302148>] (__do_softirq) from [<80320408>] (irq_exit+0x74/0xd4) [<80320408>] (irq_exit) from [<803638a4>] (__handle_domain_irq+0x90/0xb4) [<803638a4>] (__handle_domain_irq) from [<805bedec>] (gic_handle_irq+0x58/0x90) [<805bedec>] (gic_handle_irq) from [<80301a78>] (__irq_svc+0x58/0x8c) Tested-on: IPQ8074 hw2.0 AHB WLAN.HK.2.7.0.1-01744-QCAHKSUPL_SILICONZ-1

CVE ID	Description
CVE-2023-53317	In the Linux kernel, the following vulnerability has been resolved: ext4: fix WARNING in mb_find_extent Syzbot found the following issue: EXT4-fs: Warning: mounting with data=journal disables delayed allocation, dioread_nolock, O_DIRECT and fast_commit support! EXT4-fs (loop0): orphan cleanup on readonly fs -----[cut here]----- WARNING: CPU: 1 PID: 5067 at fs/ext4/mballo.c:1869 mb_find_extent+0x8a1/0xe30 Modules linked in: CPU: 1 PID: 5067 Comm: syz-executor307 Not tainted 6.2.0-rc1-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 RIP: 0010:mb_find_extent+0x8a1/0xe30 fs/ext4/mballo.c:1869 RSP: 0018:ffffc90003c9e098 EFLAGS: 00010293 RAX: ffffffff82405731 RBX: 0000000000000041 RCX: ffff8880783457c0 RDX: 0000000000000000 RSI: 0000000000000041 RDI: 0000000000000040 RBP: 0000000000000040 R08: fffffff82405723 R09: ffffed10053c9402 R10: ffffed10053c9402 R11: 1ffff110053c9401 R12: 0000000000000000 R13: fffffc90003c9e538 R14: dffffc0000000000 R15: fffffc90003c9e2cc FS: 0000555556665300(0000) GS:ffff8880b9900000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000008005003 CR2: 000056312f6796f8 CR3: 0000000022437000 CR4: 00000000003506e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ext4_mb_complex_scan_group+0x353/0x1100 fs/ext4/mballo.c:2307 ext4_mb_regular_allocator+0x1533/0x3860 fs/ext4/mballo.c:2735 ext4_mb_new_blocks+0xddf/0x3db0 fs/ext4/mballo.c:5605 ext4_ext_map_blocks+0x1868/0x6880 fs/ext4/extents.c:4286 ext4_map_blocks+0xa49/0x1cc0 fs/ext4/inode.c:651 ext4_getblk+0x1b9/0x770 fs/ext4/inode.c:864 ext4_bread+0x2a/0x170 fs/ext4/inode.c:920 ext4_quota_write+0x225/0x570 fs/ext4/super.c:7105 write_blk fs/quota/quota_tree.c:64 [inline] get_free_dqblk+0x34a/0x6d0 fs/quota/quota_tree.c:130 do_insert_tree+0x26b/0x1aa0 fs/quota/quota_tree.c:340 do_insert_tree+0x722/0x1aa0 fs/quota/quota_tree.c:375 do_insert_tree+0x722/0x1aa0 fs/quota/quota_tree.c:375 dq_insert_tree fs/quota/quota_tree.c:401 [inline] qtree_write_dquot+0x3b6/0x530 fs/quota/quota_tree.c:420 v2_write_dquot+0x11b/0x190 fs/quota/quota_v2.c:358 dquot_acquire+0x348/0x670 fs/quota/dquot.c:444 ext4_acquire_dquot+0x2dc/0x400 fs/ext4/super.c:6740 dqget+0x999/0xcd0 fs/quota/dquot.c:914 __dquot_initialize+0x3d0/0xcfc0 fs/quota/dquot.c:1492 ext4_process_orphan+0x57/0x2d0 fs/ext4/orphan.c:329 ext4_orphan_cleanup+0xb60/0x1340 fs/ext4/orphan.c:474 __ext4_fill_super fs/ext4/super.c:5516 [inline] ext4_fill_super+0x81cd/0x8700 fs/ext4/super.c:5644 get_tree_bdev+0x400/0x620 fs/super.c:1282 vfs_get_tree+0x88/0x270 fs/super.c:1489 do_new_mount+0x289/0xad0 fs/namespaces.c:3145 do_mount fs/namespaces.c:3488 [inline] __do_sys_mount fs/namespaces.c:3697 [inline] __se_sys_mount+0x2d3/0x3c0 fs/namespaces.c:3674 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd Add some debug information: mb_find_extent: mb_find_extent block=41, order=0 needed=64 next=0 ex=0/41/1@3735929054 64 64 7 block_bitmap: ff 3f 0c 00 fc 01 00 00 d2 3d 00 00 00 00 00 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff Actually, blocks per group is 64, but block bitmap indicate at least has 128 blocks. Now, ext4_validate_block_bitmap() didn't check invalid block's bitmap if set. To resolve above issue, add check like fsck "Padding at end of block bitmap is not set".
CVE-2023-53320	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix issues in mpi3mr_get_all_tgt_info() The function mpi3mr_get_all_tgt_info() has four issues: 1) It calculates valid entry length in alltgt_info assuming the header part of the struct mpi3mr_device_map_info would equal to sizeof(u32). The correct size is sizeof(u64). 2) When it calculates the valid entry length kern_entrylen, it excludes one entry by subtracting 1 from num_devices. 3) It copies num_device by calling memcpy(). Substitution is enough. 4) It does not specify the calculated length to sg_copy_from_buffer(). Instead, it specifies the payload length which is larger than the alltgt_info size. It causes "BUG: KASAN: slab-out-of-bounds". Fix the issues by using the correct header size, removing the subtraction from num_devices, replacing the memcpy() with substitution and specifying the correct length to sg_copy_from_buffer().
CVE-2023-53321	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211_hwsim: drop short frames While technically some control frames like ACK are shorter and end after Address 1, such frames shouldn't be forwarded through wmediumd or similar userspace, so require the full 3-address header to avoid accessing invalid memory if shorter frames are passed in.
CVE-2023-53322	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Wait for io return on terminate rport System crash due to use after free. Current code allows terminate_rport_io to exit before making sure all IOs has returned. For FCP-2 device, IO's can hang on in HW because driver has not tear down the session in FW at first sign of cable pull. When dev_loss_tmo timer pops, terminate_rport_io is called and upper layer is about to free various resources. Terminate_rport_io trigger qla to do the final cleanup, but the cleanup might not be fast enough where it leave qla still holding on to the same resource. Wait for IO's to return to upper layer before resources are freed.

CVE ID	Description
CVE-2023-53326	In the Linux kernel, the following vulnerability has been resolved: powerpc: Don't try to copy PPR for task with NULL pt_regs powerpc sets up PF_KTHREAD and PF_IO_WORKER with a NULL pt_regs, which from my (arguably very short) checking is not commonly done for other archs. This is fine, except when PF_IO_WORKER's have been created and the task does something that causes a coredump to be generated. Then we get this crash: Kernel attempted to read user page (160) - exploit attempt? (uid: 1000) BUG: Kernel NULL pointer dereference on read at 0x00000160 Faulting instruction address: 0xc000000000c3a60 Oops: Kernel access of bad area, sig: 11 [#1] LE PAGE_SIZE=64K MMU=Radix SMP NR_CPUS=32 NUMA pSeries Modules linked in: bochs drm_vram_helper drm_kms_helper xts binfmt_misc ecb ctr syscopyarea sysfillrect cbc sysimgblt drm_ttm_helper aes_generic ttm sg libaes evdev joydev virtio_balloon vmx_crypto gf128mul drm dm_mod fuse loop configfs drm_panel_orientation_quirks ip_tables x_tables autofs4 hid_generic usbhid hid xhci_pci xhci_hcd usbcore usb_common sd_mod CPU: 1 PID: 1982 Comm: ppc-crash Not tainted 6.3.0-rc2+ #88 Hardware name: IBM pSeries (emulated by qemu) POWER9 (raw) 0x4e1202 0xf000005 of:SLOF,HEAD hv:linux,kvm pSeries NIP: c000000000c3a60 LR: c00000000039944 CTR: c000000000398e0 REGS: c0000000041833b0 TRAP: 0300 Not tainted (6.3.0-rc2+) MSR: 800000000280b033 <SF,VEC,VSX,EE,FP,ME,IR,DR,RI,LE> CR: 88082828 XER: 200400f8 ... NIP memcpy_power7+0x200/0x7d0 LR ppr_get+0x64/0xb0 Call Trace: ppr_get+0x40/0xb0 (unreliable) __regset_get+0x180/0x1f0 regset_get_alloc+0x64/0x90 elf_core_dump+0xb98/0x1b60 do_coredump+0x1c34/0x24a0 get_signal+0x71c/0x1410 do_notify_resume+0x140/0x6f0 interrupt_exit_user_prepare_main+0x29c/0x320 interrupt_exit_user_prepare+0x6c/0xa0 interrupt_return_srr_user+0x8/0x138 Because ppr_get() is trying to copy from a PF_IO_WORKER with a NULL pt_regs. Check for a valid pt_regs in both ppc_get/ppr_set, and return an error if not set. The actual error value doesn't seem to be important here, so just pick -EINVAL. [mpe: Trim oops in change log, add Fixes & Cc stable]
CVE-2023-53332	In the Linux kernel, the following vulnerability has been resolved: genirq/ipi: Fix NULL pointer deref in irq_data_get_affinity_mask() If ipi_send_{mask single}() is called with an invalid interrupt number, all the local variables there will be NULL. ipi_send_verify() which is invoked from these functions does verify its 'data' parameter, resulting in a kernel oops in irq_data_get_affinity_mask() as the passed NULL pointer gets dereferenced. Add a missing NULL pointer check in ipi_send_verify()... Found by Linux Verification Center (linuxtesting.org) with the SVACE static analysis tool.
CVE-2023-53333	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: dccp: copy entire header to stack buffer, not just basic one Eric Dumazet says: nf_conntrack_dccp_packet() has an unique: dh = skb_header_pointer(skb, dataoff, sizeof(_dh), &_dh); And nothing more is 'pulled' from the packet, depending on the content. dh->dccph_doff, and/or dh->dccph_x ...) So dccp_ack_seq() is happily reading stuff past the _dh buffer. BUG: KASAN: stack-out-of-bounds in nf_conntrack_dccp_packet +0x1134/0x11c0 Read of size 4 at addr ffff000128f66e0c by task syz-executor.2/29371 [...] Fix this by increasing the stack buffer to also include room for the extra sequence numbers and all the known dccp packet type headers, then pull again after the initial validation of the basic header. While at it, mark packets invalid that lack 48bit sequence bit but where RFC says the type MUST use them. Compile tested only. v2: first skb_header_pointer() now needs to adjust the size to only pull the generic header. (Eric) Heads-up: I intend to remove dccp conntrack support later this year.
CVE-2023-53335	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: Fix potential null-ptr-deref in pass_establish() If get_ep_from_tid() fails to lookup non-NULL value for ep, ep is dereferenced later regardless of whether it is empty. This patch adds a simple sanity check to fix the issue. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-53338	In the Linux kernel, the following vulnerability has been resolved: lwt: Fix return values of BPF xmit ops BPF encap ops can return different types of positive values, such like NET_RX_DROP, NET_XMIT_CN, NETDEV_TX_BUSY, and so on, from function skb_do_redirect and bpf_lwt_xmit_reroute. At the xmit hook, such return values would be treated implicitly as LWTUNNEL_XMIT_CONTINUE in ip(6)_finish_output2. When this happens, skbs that have been freed would continue to the neighbor subsystem, causing use-after-free bug and kernel crashes. To fix the incorrect behavior, skb_do_redirect return values can be simply discarded, the same as tc-egress behavior. On the other hand, bpf_lwt_xmit_reroute returns useful errors to local senders, e.g. PMTU information. Thus convert its return values to avoid the conflict with LWTUNNEL_XMIT_CONTINUE.
CVE-2023-53340	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Collect command failures data only for known commands DEVX can issue a general command, which is not used by mlx5 driver. In case such command is failed, mlx5 is trying to collect the failure data, However, mlx5 doesn't create a storage for this command, since mlx5 doesn't use it. This lead to array-index-out-of-bounds error. Fix it by checking whether the command is known before collecting the failure data.

CVE ID	Description
CVE-2023-53341	In the Linux kernel, the following vulnerability has been resolved: of/fdt: run soc memory setup when early_init_dt_scan_memory fails If memory has been found early_init_dt_scan_memory now returns 1. If it hasn't found any memory it will return 0, allowing other memory setup mechanisms to carry on. Previously early_init_dt_scan_memory always returned 0 without distinguishing between any kind of memory setup being done or not. Any code path after the early_init_dt_scan memory call in the ramips plat_mem_setup code wouldn't be executed anymore. Making early_init_dt_scan_memory the only way to initialize the memory. Some boards, including my mt7621 based Cudy X6 board, depend on memory initialization being done via the soc_info.mem_detect function pointer. Those wouldn't be able to obtain memory and panic the kernel during early bootup with the message "early_init_dt_alloc_memory_arch: Failed to allocate 12416 bytes align=0x40".
CVE-2023-53343	In the Linux kernel, the following vulnerability has been resolved: icmp6: Fix null-ptr-deref of ip6_null_entry->rt6i_iddev in icmp6_dev(). With some IPv6 Ext Hdr (RPL, SRv6, etc.), we can send a packet that has the link-local address as src and dst IP and will be forwarded to an external IP in the IPv6 Ext Hdr. For example, the script below generates a packet whose src IP is the link-local address and dst is updated to 11:: # for f in \$(find /proc/sys/net/ -name *seg6_enabled*); do echo 1 > \$f; done # python3 >>> from socket import * >>> from scapy.all import * >>> SRC_ADDR = DST_ADDR = "fe80::5054:ff:fe12:3456" >>> >>> pkt = IPv6(src=SRC_ADDR, dst=DST_ADDR) >>> pkt /= IPv6ExtHdrSegmentRouting(type=4, addresses=["11::", "22::"], segleft=1) >>> >>> sk = socket(AF_INET6, SOCK_RAW, IPPROTO_RAW) >>> sk.sendto(bytes(pkt), (DST_ADDR, 0)) For such a packet, we call ip6_route_input() to look up a route for the next destination in these three functions depending on the header type. * ipv6_rthdr_rcv() * ipv6_rpl_srh_rcv() * ipv6_srh_rcv() If no route is found, ip6_null_entry is set to skb, and the following dst_input(skb) calls ip6_pkt_drop(). Finally, in icmp6_dev(), we dereference skb_rt6_info(skb)->rt6i_iddev->dev as the input device is the loopback interface. Then, we have to check if skb_rt6_info(skb)->rt6i_iddev is NULL or not to avoid NULL pointer deref for ip6_null_entry. BUG: kernel NULL pointer dereference, address: 0000000000000000 PF: supervisor read access in kernel mode PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 0 PID: 157 Comm: python3 Not tainted 6.4.0-11996-gb121d614371c #35 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 RIP: 0010:icmp6_send (net/ipv6/icmp.c:436 net/ipv6/icmp.c:503) Code: fe ff ff 48 c7 40 30 c0 86 5d 83 e8 c6 44 1c 00 e9 c8 fc ff ff 49 8b 46 58 48 83 e0 fe 0f 84 4a fb ff ff 48 8b 80 d0 00 00 00 <48> 8b 00 44 8b 88 e0 00 00 00 e9 34 fb ff ff 4d 85 ed 0f 85 69 01 RSP: 0018:ffff900000003c70 EFLAGS: 00000286 RAX: 0000000000000000 RBX: 0000000000000001 RCX: 00000000000000e0 RDX: 0000000000000021 RSI: 0000000000000000 RDI: ffff888006d72a18 RBP: ffff900000003d80 R08: 0000000000000000 R09: 0000000000000001 R10: ffff900000003d98 R11: 0000000000000040 R12: ffff888006d72a10 R13: 0000000000000000 R14: ffff8880057fb800 R15: ffffffff835d86c0 FS: 00007f9dc72ee740(0000) GS: ffff88807dc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000000057b2000 CR4: 00000000007506f0 PKRU: 55555554 Call Trace: <IRQ> ip6_pkt_drop (net/ipv6/route.c:4513) ipv6_rthdr_rcv (net/ipv6/exthdrs.c:640 net/ipv6/exthdrs.c:686) ip6_protocol_deliver_rcu (net/ipv6/ip6_input.c:437 (discriminator 5)) ip6_input_finish (/include/linux/rcupdate.h:781 net/ipv6/ip6_input.c:483) __netif_receive_skb_one_core (net/core/dev.c:5455) process_backlog (/include/linux/rcupdate.h:781 net/core/dev.c:5895) __napi_poll (net/core/dev.c:6460) net_rx_action (net/core/dev.c:6529 net/core/dev.c:6660) __do_softirq (/arch/x86/include/asm/jump_label.h:27 ./include/linux/jump_label.h:207 ./include/trace/events/irq.h:142 kernel/softirq.c:554) do_softirq (kernel/softirq.c:454 kernel/softirq.c:441) </IRQ> <TASK> __local_bh_enable_ip (kernel/softirq.c:381) __dev_queue_xmit (net/core/dev.c:4231) ip6_finish_output2 (/include/net/neighbor.h:544 net/ipv6/ip6_output.c:135) rawv6_sendmsg (/include/net/dst.h:458 ./include/linux/netfilter.h:303 net/ipv6/raw.c:656 net/ipv6/raw.c:914) sock_sendmsg (net/socket.c:725 net/socket.c:748) __sys_sendto (net/socket.c:2134) __x64_sys_sendto (net/socket.c:2146 net/socket.c:2142 net/socket.c:2142) do_syscall_64 (arch/x86/entry/common.c:50 arch/x86/entry/common.c:80) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:120) RIP: 0033:0xf9dc751baea Code: d8 64 89 02 48 c7 c0 ff f ---truncated---

CVE ID	Description
CVE-2023-53344	In the Linux kernel, the following vulnerability has been resolved: can: bcm: bcm_tx_setup(): fix KMSAN uninit-value in vfs_write Syzkaller reported the following issue: <pre>===== BUG: KMSAN: uninit-value in aio_rw_done fs/aio.c:1520 [inline] BUG: KMSAN: uninit-value in aio_write+0x899/0x950 fs/aio.c:1600 aio_rw_done fs/aio.c:1520 [inline] aio_write+0x899/0x950 fs/aio.c:1600 io_submit_one+0x1d1c/0x3bf0 fs/aio.c:2019 __do_sys_io_submit fs/aio.c:2078 [inline] __se_sys_io_submit+0x293/0x770 fs/aio.c:2048 __x64_sys_io_submit+0x92/0xd0 fs/aio.c:2048 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd Uninit was created at: slab_post_alloc_hook mm/slab.h:766 [inline] slab_alloc_node mm/slab.c:3452 [inline] __kmem_cache_alloc_node+0x71f/0xce0 mm/slab.c:3491 __do_kmalloc_node mm/slab_common.c:967 [inline] __kmalloc+0x11d/0x3b0 mm/slab_common.c:981 kmalloc_array include/linux/slab.h:636 [inline] bcm_tx_setup+0x80e/0x29d0 net/can/bcm.c:930 bcm_sendmsg+0x3a2/0xce0 net/can/bcm.c:1351 sock_sendmsg_nosec net/socket.c:714 [inline] sock_sendmsg net/socket.c:734 [inline] sock_write_iter+0x495/0x5e0 net/socket.c:1108 call_write_iter include/linux/fs.h:2189 [inline] aio_write+0x63a/0x950 fs/aio.c:1600 io_submit_one+0x1d1c/0x3bf0 fs/aio.c:2019 __do_sys_io_submit fs/aio.c:2078 [inline] __se_sys_io_submit+0x293/0x770 fs/aio.c:2048 __x64_sys_io_submit+0x92/0xd0 fs/aio.c:2048 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd CPU: 1 PID: 5034 Comm: syz-executor350 Not tainted 6.2.0-rc6-syzkaller-80422-geda666ff2276 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/12/2023 ===== We can follow the call chain and find that 'bcm_tx_setup' function calls 'memcpy_from_msg' to copy some content to the newly allocated frame of 'op->frames'. After that the 'len' field of copied structure being compared with some constant value (64 or 8). However, if 'memcpy_from_msg' returns an error, we will compare some uninitialized memory. This triggers 'uninit-value' issue. This patch will add 'memcpy_from_msg' possible errors processing to avoid uninit-value issue. Tested via syzkaller</pre>
CVE-2023-53347	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Handle pairing of E-switch via uplink un/load APIs In case user switch a device from switchdev mode to legacy mode, mlx5 first unpair the E-switch and afterwards unload the uplink vport. From the other hand, in case user remove or reload a device, mlx5 first unload the uplink vport and afterwards unpair the E-switch. The latter is causing a bug[1], hence, handle pairing of E-switch as part of uplink un/load APIs. [1] In case VF_LAG is used, every tc fdb flow is duplicated to the peer esw. However, the original esw keeps a pointer to this duplicated flow, not the peer esw. e.g.: if user create tc fdb flow over esw0, the flow is duplicated over esw1, in FW/HW, but in SW, esw0 keeps a pointer to the duplicated flow. During module unload while a peer tc fdb flow is still offloaded, in case the first device to be removed is the peer device (esw1 in the example above), the peer net-dev is destroyed, and so the mlx5e_priv is memset to 0. Afterwards, the peer device is trying to unpair himself from the original device (esw0 in the example above). Unpair API invoke the original device to clear peer flow from its eswitch (esw0), but the peer flow, which is stored over the original eswitch (esw0), is trying to use the peer mlx5e_priv, which is memset to 0 and result in bellow kernel-oops. [157.964081] BUG: unable to handle page fault for address: 000000000002ce60 [157.964662] #PF: supervisor read access in kernel mode [157.965123] #PF: error_code(0x0000) - not-present page [157.965582] PGD 0 P4D 0 [157.965866] Oops: 0000 [#1] SMP [157.967670] RIP: 0010:mlx5e_tc_del_fdb_flow+0x48/0x460 [mlx5_core] [157.976164] Call Trace: [157.976437] <TASK> [157.976690] __mlx5e_tc_del_fdb_peer_flow+0xe6/0x100 [mlx5_core] [157.977230] mlx5e_tc_clean_fdb_peer_flows+0x67/0x90 [mlx5_core] [157.977767] mlx5_esw_offloads_unpair+0x2d/0x1e0 [mlx5_core] [157.984653] mlx5_esw_offloads_devcom_event+0xbf/0x130 [mlx5_core] [157.985212] mlx5_devcom_send_event+0xa3/0xb0 [mlx5_core] [157.985714] esw_offloads_disable+0x5a/0x110 [mlx5_core] [157.986209] mlx5_eswitch_disable_locked+0x152/0x170 [mlx5_core] [157.986757] mlx5_eswitch_disable+0x51/0x80 [mlx5_core] [157.987248] mlx5_unload+0x2a/0xb0 [mlx5_core] [157.987678] mlx5_uninit_one+0x5f/0xd0 [mlx5_core] [157.988127] remove_one+0x64/0xe0 [mlx5_core] [157.988549] pci_device_remove+0x31/0xa0 [157.988933] device_release_driver_internal+0x18f/0x1f0 [157.989402] driver_detach+0x3f/0x80 [157.989754] bus_remove_driver+0x70/0xf0 [157.990129] pci_unregister_driver+0x34/0x90 [157.990537] mlx5_cleanup+0xc/0x1c [mlx5_core] [157.990972] __x64_sys_delete_module+0x15a/0x250 [157.991398] ? exit_to_user_mode_prepare+0xea/0x110 [157.991840] do_syscall_64+0x3d/0x90 [157.992198] entry_SYSCALL_64_after_hwframe+0x46/0xb0

CVE ID	Description
CVE-2023-53351	In the Linux kernel, the following vulnerability has been resolved: drm/sched: Check scheduler work queue before calling timeout handling During an IGT GPU reset test we see again oops despite of commit 0c8c901aaebc9 (drm/sched: Check scheduler ready before calling timeout handling). It uses ready condition whether to call drm_sched_fault which unwind the TDR leads to GPU reset. However it looks the ready condition is overloaded with other meanings, for example, for the following stack is related GPU reset : 0 gfx_v9_0_cp_gfx_start 1 gfx_v9_0_cp_gfx_resume 2 gfx_v9_0_cp_resume 3 gfx_v9_0_hw_init 4 gfx_v9_0_resume 5 amdgpu_device_ip_resume_phase2 does the following: / * start the ring */ gfx_v9_0_cp_gfx_start(adev); ring->sched.ready = true; The same approach is for other ASICs as well : gfx_v8_0_cp_gfx_resume gfx_v10_0_kiq_resume, etc... As a result, our GPU reset test causes GPU fault which calls unconditionally gfx_v9_0_fault and then drm_sched_fault. However now it depends on whether the interrupt service routine drm_sched_fault is executed after gfx_v9_0_cp_gfx_start is completed which sets the ready field of the scheduler to true even for uninitialized schedulers and causes oops vs no fault or when ISR drm_sched_fault is completed prior gfx_v9_0_cp_gfx_start and NULL pointer dereference does not occur. Use the field timeout_wq to prevent oops for uninitialized schedulers. The field could be initialized by the work queue of resetting the domain. v1: Corrections to commit message (Luben)
CVE-2023-53352	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: check null pointer before accessing when swapping Add a check to avoid null pointer dereference as below: [90.002283] general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN NOPTI [90.002292] KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] [90.002346] ? exc_general_protection +0x159/0x240 [90.002352] ? asm_exc_general_protection+0x26/0x30 [90.002357] ? ttm_bo_evict_swapout_allowable+0x322/0x5e0 [ttm] [90.002365] ? ttm_bo_evict_swapout_allowable +0x42e/0x5e0 [ttm] [90.002373] ttm_bo_swapout+0x134/0x7f0 [ttm] [90.002383] ? __pfx_ttm_bo_swapout+0x10/0x10 [ttm] [90.002391] ? lock_acquire+0x44d/0x4f0 [90.002398] ? ttm_device_swapout+0xa5/0x260 [ttm] [90.002412] ? lock_acquired+0x355/0xa00 [90.002416] ? do_raw_spin_trylock+0xb6/0x190 [90.002421] ? __pfx_lock_acquired+0x10/0x10 [90.002426] ? ttm_global_swapout+0x25/0x210 [ttm] [90.002442] ttm_device_swapout +0x198/0x260 [ttm] [90.002456] ? __pfx_ttm_device_swapout+0x10/0x10 [ttm] [90.002472] ttm_global_swapout+0x75/0x210 [ttm] [90.002486] ttm_tt_populate+0x187/0x3f0 [ttm] [90.002501] ttm_bo_handle_move_mem+0x437/0x590 [ttm] [90.002517] ttm_bo_validate+0x275/0x430 [ttm] [90.002530] ? __pfx_ttm_bo_validate+0x10/0x10 [ttm] [90.002544] ? kasan_save_stack+0x33/0x60 [90.002550] ? kasan_set_track+0x25/0x30 [90.002554] ? __kasan_kmalloc+0x8f/0xa0 [90.002558] ? amdgpu_gtt_mgr_new+0x81/0x420 [amdgpu] [90.003023] ? ttm_resource_alloc+0xf6/0x220 [ttm] [90.003038] amdgpu_bo_pin_restricted+0x2dd/0x8b0 [amdgpu] [90.003210] ? __x64_sys_ioctl +0x131/0x1a0 [90.003210] ? do_syscall_64+0x60/0x90
CVE-2023-53354	In the Linux kernel, the following vulnerability has been resolved: skbuff: skb_segment, Call zero copy functions before using skbuff frags Commit bf5c25d60861 ("skbuff: in skb_segment, call zerocopy functions once per nskb") added the call to zero copy functions in skb_segment(). The change introduced a bug in skb_segment() because skb_orphan_frags() may possibly change the number of fragments or allocate new fragments altogether leaving nrfrags and frag to point to the old values. This can cause a panic with stacktrace like the one below. [193.894380] BUG: kernel NULL pointer dereference, address: 00000000000000bc [193.895273] CPU: 13 PID: 18164 Comm: vh-net-17428 Kdump: loaded Tainted: G O 5.15.123+ #26 [193.903919] RIP: 0010:skb_segment+0xb0e/0x12f0 [194.021892] Call Trace: [194.027422] <TASK> [194.072861] tcp_gso_segment+0x107/0x540 [194.082031] inet_gso_segment+0x15c/0x3d0 [194.090783] skb_mac_gso_segment+0x9f/0x110 [194.095016] __skb_gso_segment+0xc1/0x190 [194.103131] netem_enqueue+0x290/0xb10 [sch_netem] [194.107071] dev_qdisc_enqueue+0x16/0x70 [194.110884] __dev_queue_xmit+0x63b/0xb30 [194.121670] bond_start_xmit+0x159/0x380 [bonding] [194.128506] dev_hard_start_xmit+0xc3/0x1e0 [194.131787] __dev_queue_xmit+0x8a0/0xb30 [194.138225] macvlan_start_xmit+0x4f/0x100 [macvlan] [194.141477] dev_hard_start_xmit+0xc3/0x1e0 [194.144622] sch_direct_xmit+0xe3/0x280 [194.147748] __dev_queue_xmit+0x54a/0xb30 [194.154131] tap_get_user+0x2a8/0x9c0 [tap] [194.157358] tap_sendmsg+0x52/0x8e0 [tap] [194.167049] handle_tx_zerocopy+0x14e/0x4c0 [vhost_net] [194.173631] handle_tx+0xcd/0xe0 [vhost_net] [194.176959] vhost_worker+0x76/0xb0 [vhost] [194.183667] kthread+0x118/0x140 [194.190358] ret_from_fork+0x1f/0x30 [194.193670] </TASK> In this case calling skb_orphan_frags() updated nr_frags leaving nrfrags local variable in skb_segment() stale. This resulted in the code hitting i >= nrfrags prematurely and trying to move to next frag_skb using list_skb pointer, which was NULL, and caused kernel panic. Move the call to zero copy functions before using frags and nr_frags.
CVE-2023-53357	In the Linux kernel, the following vulnerability has been resolved: md/raid10: check slab-out-of-bounds in md_bitmap_get_counter If we write a large number to md_bitmap_set_bits, md_bitmap_checkpage() will return -EINVAL because 'page >= bitmap->pages', but the return value was not checked immediately in md_bitmap_get_counter() in order to set *blocks value and slab-out-of-bounds occurs. Move check of 'page >= bitmap->pages' to md_bitmap_get_counter() and return directly if true.

CVE ID	Description
CVE-2023-53359	In the Linux kernel, the following vulnerability has been resolved: USB: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-53365	In the Linux kernel, the following vulnerability has been resolved: ip6mr: Fix skb_under_panic in ip6mr_cache_report() skbuff: skb_under_panic: text:ffff887f1f69 len:56 put:-4 head:ffff88805f86a800 data:ffff887f5f86a850 tail:0x88 end:0x2c0 dev:pim6reg -----[cut here]----- kernel BUG at net/core/skbuff.c:192! invalid opcode: 0000 [#1] PREEMPT SMP KASAN CPU: 2 PID: 22968 Comm: kworker/2:11 Not tainted 6.5.0-rc3-00044-g0a8db05b571a #236 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Workqueue: ipv6_addrconf addrconf_dad_work RIP: 0010:skb_panic+0x152/0x1d0 Call Trace: <TASK> skb_push+0xc4/0xe0 ip6mr_cache_report+0xd69/0x19b0 reg_vif_xmit+0x406/0x690 dev_hard_start_xmit+0x17e/0x6e0 __dev_queue_xmit+0x2d6a/0x3d20 vlan_dev_hard_start_xmit+0x3ab/0x5c0 dev_hard_start_xmit+0x17e/0x6e0 __dev_queue_xmit+0x2d6a/0x3d20 neigh_connected_output+0x3ed/0x570 ip6_finish_output2+0x5b5/0x1950 ip6_finish_output+0x693/0x11c0 ip6_output+0x24b/0x880 NF_HOOK.constprop.0+0xfd/0x530 ndisc_send_skb+0x9db/0x1400 ndisc_send_rs+0x12a/0x6c0 addrconf_dad_completed+0x3c9/0xea0 addrconf_dad_work+0x849/0x1420 process_one_work+0xa22/0x16e0 worker_thread+0x679/0x10c0 ret_from_fork+0x28/0x60 ret_from_fork_asm+0x11/0x20 When setup a vlan device on dev pim6reg, DAD ns packet may sent on reg_vif_xmit(). reg_vif_xmit() ip6mr_cache_report() skb_push(skb, -skb_network_offset(pkt)); // skb_network_offset(pkt) is 4 And skb_push declared as: void *skb_push(struct sk_buff *skb, unsigned int len); skb->data -= len; //0xffff88805f86a84c - 0xffffffff = 0xffff887f5f86a850 skb->data is set to 0xffff887f5f86a850, which is invalid mem addr, lead to skb_push() fails.
CVE-2023-53368	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix race issue between cpu buffer write and swap Warning happened in rb_end_commit() at code: if (RB_WARN_ON(cpu_buffer, !local_read(&cpu_buffer->committing))) WARNING: CPU: 0 PID: 139 at kernel/trace/ring_buffer.c:3142 rb_commit+0x402/0x4a0 Call Trace: ring_buffer_unlock_commit+0x42/0x250 trace_buffer_unlock_commit_regs+0x3b/0x250 trace_event_buffer_commit+0xe5/0x440 trace_event_buffer_reserve+0x11c/0x150 trace_event_raw_event_sched_switch+0x23c/0x2c0 __traceiter_sched_switch+0x59/0x80 __schedule+0x72b/0x1580 schedule+0x92/0x120 worker_thread+0xa0/0x6f0 It is because the race between writing event into cpu buffer and swapping cpu buffer through file per_cpu/cpu0/snapshot: Write on CPU 0 Swap buffer by per_cpu/cpu0/snapshot on CPU 1 ----- tracing_snapshot_write() [...] ring_buffer_lock_reserve() cpu_buffer = buffer->buffers[cpu]; // 1. Suppose find 'cpu_buffer_a'; [...] rb_reserve_next_event() [...] ring_buffer_swap_cpu() if (local_read(&cpu_buffer_a->committing)) goto out_dec; if (local_read(&cpu_buffer_b->committing)) goto out_dec; buffer_a->buffers[cpu] = cpu_buffer_b; buffer_b->buffers[cpu] = cpu_buffer_a; // 2. cpu_buffer has swapped here. rb_start_commit(cpu_buffer); if (unlikely(READ_ONCE(cpu_buffer->buffer) != buffer)) { // 3. This check passed due to 'cpu_buffer->buffer' [...] // has not changed here. return NULL; } cpu_buffer_b->buffer = buffer_a; cpu_buffer_a->buffer = buffer_b; [...] // 4. Reserve event from 'cpu_buffer_a'. ring_buffer_unlock_commit() [...] cpu_buffer = buffer->buffers[cpu]; // 5. Now find 'cpu_buffer_b' !!! rb_commit(cpu_buffer) rb_end_commit() // 6. WARN for the wrong 'committing' state !!! Based on above analysis, we can easily reproduce by following testcase: `` bash #!/bin/bash dmesg -n 7 sysctl -w kernel.panic_on_warn=1 TR=/sys/kernel/tracing echo 7 > \${TR}/buffer_size_kb echo "sched:sched_switch" > \${TR}/set_event while [true]; do echo 1 > \${TR}/per_cpu/cpu0/snapshot done & while [true]; do echo 1 > \${TR}/per_cpu/cpu0/snapshot done & while [true]; do echo 1 > \${TR}/per_cpu/cpu0/snapshot done & `` To fix it, IIUC, we can use smp_call_function_single() to do the swap on the target cpu where the buffer is located, so that above race would be avoided.

CVE ID	Description
CVE-2023-53369	In the Linux kernel, the following vulnerability has been resolved: net: dcb: choose correct policy to parse DCB_ATTR_BCN The dcbnl_bcn_setcfg uses erroneous policy to parse tb[DCB_ATTR_BCN], which is introduced in commit 859ee3c43812 ("DCB: Add support for DCB BCN"). Please see the comment in below code static int dcbnl_bcn_setcfg(...) { ... ret = nla_parse_nested_deprecated(..., dcbnl_pfc_up_nest, ...) // !!! dcbnl_pfc_up_nest for attributes // DCB_PFC_UP_ATTR_0 to DCB_PFC_UP_ATTR_ALL in enum dcbnl_pfc_up_attrs ... for (i = DCB_BCN_ATTR_RP_0; i <= DCB_BCN_ATTR_RP_7; i++) { // !!! DCB_BCN_ATTR_RP_0 to DCB_BCN_ATTR_RP_7 in enum dcbnl_bcn_attrs ... value_byte = nla_get_u8(data[i]); ... } ... for (i = DCB_BCN_ATTR_BCNA_0; i <= DCB_BCN_ATTR_RI; i++) { // !!! DCB_BCN_ATTR_BCNA_0 to DCB_BCN_ATTR_RI in enum dcbnl_bcn_attrs ... value_int = nla_get_u32(data[i]); ... } ... } That is, the nla_parse_nested_deprecated uses dcbnl_pfc_up_nest attributes to parse nlattr defined in dcbnl_pfc_up_attrs. But the following access code fetch each nlattr as dcbnl_bcn_attrs attributes. By looking up the associated nla_policy for dcbnl_bcn_attrs. We can find the beginning part of these two policies are "same", static const struct nla_policy dcbnl_pfc_up_nest[...] = { [DCB_PFC_UP_ATTR_0] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_1] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_2] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_3] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_4] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_5] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_6] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_7] = {.type = NLA_U8}, [DCB_PFC_UP_ATTR_ALL] = {.type = NLA_FLAG}, }; static const struct nla_policy dcbnl_bcn_nest[...] = { [DCB_BCN_ATTR_RP_0] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_1] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_2] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_3] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_4] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_5] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_6] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_7] = {.type = NLA_U8}, [DCB_BCN_ATTR_RP_ALL] = {.type = NLA_FLAG}, // from here is somewhat different [DCB_BCN_ATTR_BCNA_0] = {.type = NLA_U32}, ... [DCB_BCN_ATTR_ALL] = {.type = NLA_FLAG}, }; Therefore, the current code is buggy and this nla_parse_nested_deprecated could overflow the dcbnl_pfc_up_nest and use the adjacent nla_policy to parse attributes from DCB_BCN_ATTR_BCNA_0. Hence use the correct policy dcbnl_bcn_nest to parse the nested tb[DCB_ATTR_BCN] TLV.
CVE-2023-53371	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: fix memory leak in mlx5e_fs_tt_redirect_any_create The memory pointed to by the fs->any pointer is not freed in the error path of mlx5e_fs_tt_redirect_any_create, which can lead to a memory leak. Fix by freeing the memory in the error path, thereby making the error path identical to mlx5e_fs_tt_redirect_any_destroy().
CVE-2023-53372	In the Linux kernel, the following vulnerability has been resolved: sctp: fix a potential overflow in sctp_ifwdtsn_skip Currently, when traversing ifwdtsn skips with _sctp_walk_ifwdtsn, it only checks the pos against the end of the chunk. However, the data left for the last pos may be < sizeof(struct sctp_ifwdtsn_skip), and dereference it as struct sctp_ifwdtsn_skip may cause overflow. This patch fixes it by checking the pos against "the end of the chunk - sizeof(struct sctp_ifwdtsn_skip)" in sctp_ifwdtsn_skip, similar to sctp_fwdsn_skip.
CVE-2023-53376	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Use number of bits to manage bitmap sizes To allocate bitmaps, the mpi3mr driver calculates sizes of bitmaps using byte as unit. However, bitmap helper functions assume that bitmaps are allocated using unsigned long as unit. This gap causes memory access beyond the bitmap sizes and results in "BUG: KASAN: slab-out-of-bounds". The BUG was observed at firmware download to eHBA-9600. Call trace indicated that the out-of-bounds access happened in find_first_zero_bit() called from mpi3mr_send_event_ack() for miroc->evtask_cmds_bitmap. To fix the BUG, do not use bytes to manage bitmap sizes. Instead, use number of bits, and call bitmap helper functions which take number of bits as arguments. For memory allocation, call bitmap_zalloc() instead of kzalloc() and kcalloc(). For memory free, call bitmap_free() instead of kfree(). For zero clear, call bitmap_clear() instead of memset(). Remove three fields for bitmap byte sizes in struct cmd_priv which are no longer required. Replace the field dev_handle_bitmap_sz with dev_handle_bitmap_bits to keep number of bits of removepend_bitmap across resize.
CVE-2023-53378	In the Linux kernel, the following vulnerability has been resolved: drm/i915/dpt: Treat the DPT BO as a framebuffer Currently i915_gem_object_is_framebuffer() doesn't treat the BO containing the framebuffer's DPT as a framebuffer itself. This means eg. that the shrinker can evict the DPT BO while leaving the actual FB BO bound, when the DPT is allocated from regular shmem. That causes an immediate oops during hibernate as we try to rewrite the PTEs inside the already evicted DPT obj. TODO: presumably this might also be the reason for the DPT related display faults under heavy memory pressure, but I'm still not sure how that would happen as the object should be pinned by intel_dpt_pin() while in active use by the display engine... (cherry picked from commit 779cb5ba64ec7df80675a956c9022929514f517a)

CVE ID	Description
CVE-2023-53382	In the Linux kernel, the following vulnerability has been resolved: net/smc: Reset connection when trying to use SMCRv2 fails. We found a crash when using SMCRv2 with 2 Mellanox ConnectX-4. It can be reproduced by: - smc_run nginx - smc_run wrk -t 32 -c 500 -d 30 http://<ip>:<port> BUG: kernel NULL pointer dereference, address: 0000000000000014 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 8000000108713067 P4D 8000000108713067 PUD 151127067 PMD 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 4 PID: 2441 Comm: kworker/4:249 Kdump: loaded Tainted: G W E 6.4.0-rc1+ #42 Workqueue: smc_hs_wq smc_listen_work [smc] RIP: 0010:smc_clc_send_confirm_accept+0x284/0x580 [smc] RSP: 0018:ffff8294b2d7c78 EFLAGS: 00010a06 RAX: ffff8f1873238880 RBX: ffff8294b2d7dc8 RCX: 0000000000000000 RDX: 00000000000000b4 RSI: 0000000000000001 RDI: 000000000b40c00 RBP: ffff8294b2d7db8 R08: ffff8f1815c5860c R09: 0000000000000000 R10: 0000000000000400 R11: 0000000000000000 R12: ffff8f1846f56180 R13: ffff8f1815c5860c R14: 0000000000000001 R15: 0000000000000001 FS: 0000000000000000(0000) GS:ffff8f1aefd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000014 CR3: 00000001027a0001 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? mlx5_ib_map_mr_sg+0xa1/0xd0 [mlx5_ib] ? smc_buf_map_link+0x24b/0x290 [smc] ? __smc_buf_create+0x4ee/0x9b0 [smc] smc_clc_send_accept+0x4c/0xb0 [smc] smc_listen_work+0x346/0x650 [smc] ? __schedule+0x279/0x820 process_one_work+0x1e5/0x3f0 worker_thread+0x4d/0x2f0 ? __pfx_worker_thread+0x10/0x10 kthread+0xe5/0x120 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2c/0x50 <TASK> During the CLC handshake, server sequentially tries available SMCRv2 and SMCRv1 devices in smc_listen_work(). If an SMCRv2 device is found, SMCv2 based link group and link will be assigned to the connection. Then assumed that some buffer assignment errors happen later in the CLC handshake, such as RMB registration failure, server will give up SMCRv2 and try SMCRv1 device instead. But the resources assigned to the connection won't be reset. When server tries SMCRv1 device, the connection creation process will be executed again. Since conn->lnk has been assigned when trying SMCRv2, it will not be set to the correct SMCRv1 link in smc_lgr_conn_assign_link(). So in such situation, conn->lgr points to correct SMCRv1 link group but conn->lnk points to the SMCRv2 link mistakenly. Then in smc_clc_send_confirm_accept(), conn->rmb_desc->mr[link->link_idx] will be accessed. Since the link->link_idx is not correct, the related MR may not have been initialized, so crash happens. Try SMCRv2 device first -> conn->lgr: assign existed SMCRv2 link group; -> conn->link: assign existed SMCRv2 link (link_idx may be 1 in SMC_LGR_SYMMETRIC); -> sndbuf & RMB creation fails, quit; Try SMCRv1 device then -> conn->lgr: create SMCRv1 link group and assign; -> conn->link: keep SMCRv2 link mistakenly; -> sndbuf & RMB creation succeed, only RMB->mr[link_idx = 0] initialized. Then smc_clc_send_confirm_accept() accesses conn->rmb_desc->mr[conn->link->link_idx, which is 1], then crash. v This patch tries to fix this by cleaning conn->lnk before assigning link. In addition, it is better to reset the connection and clean the resources assigned if trying SMCRv2 failed in buffer creation or registration.
CVE-2023-53383	In the Linux kernel, the following vulnerability has been resolved: irqchip/gicv3: Workaround for NVIDIA erratum T241-FABRIC-4 The T241 platform suffers from the T241-FABRIC-4 erratum which causes unexpected behavior in the GIC when multiple transactions are received simultaneously from different sources. This hardware issue impacts NVIDIA server platforms that use more than two T241 chips interconnected. Each chip has support for 320 {E}SPIs. This issue occurs when multiple packets from different GICs are incorrectly interleaved at the target chip. The erratum text below specifies exactly what can cause multiple transfer packets susceptible to interleaving and GIC state corruption. GIC state corruption can lead to a range of problems, including kernel panics, and unexpected behavior. >From the erratum text: "In some cases, inter-socket AXI4 Stream packets with multiple transfers, may be interleaved by the fabric when presented to ARM Generic Interrupt Controller. GIC expects all transfers of a packet to be delivered without any interleaving. The following GICv3 commands may result in multiple transfer packets over inter-socket AXI4 Stream interface: - Register reads from GICD_I* and GICD_N* - Register writes to 64-bit GICD registers other than GICD_IROUTERn* - ITS command MOVALL Multiple commands in GICv4+ utilize multiple transfer packets, including VMOVVP, VMOVVI, VMAPP, and 64-bit register accesses." This issue impacts system configurations with more than 2 sockets, that require multi-transfer packets to be sent over inter-socket AXI4 Stream interface between GIC instances on different sockets. GICv4 cannot be supported. GICv3 SW model can only be supported with the workaround. Single and Dual socket configurations are not impacted by this issue and support GICv3 and GICv4." Writing to the chip alias region of the GICD_In{E} registers except GICD_ICENABLERN has an equivalent effect as writing to the global distributor. The SPI interrupt deactivate path is not impacted by the erratum. To fix this problem, implement a workaround that ensures read accesses to the GICD_In{E} registers are directed to the chip that owns the SPI, and disable GICv4.x features. To simplify code changes, the gic_configure_irq() function uses the same alias region for both read and write operations to GICD_ICFGR.

CVE ID	Description
CVE-2023-53384	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: avoid possible NULL skb pointer dereference In 'mwifiex_handle_uap_rx_forward()', always check the value returned by 'skb_copy()' to avoid potential NULL pointer dereference in 'mwifiex_uap_queue_bridged_pkt()', and drop original skb in case of copying failure. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-53386	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix potential use-after-free when clear keys Similar to commit c5d2b6fa26b5 ("Bluetooth: Fix use-after-free in hci_remove_ltk/hci_remove_irk"). We can not access k after kfree_rcu() call.
CVE-2023-53390	In the Linux kernel, the following vulnerability has been resolved: drivers: base: dd: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-53391	In the Linux kernel, the following vulnerability has been resolved: shmem: use ramfs_kill_sb() for kill_sb method of ramfs-based tmpfs As the ramfs-based tmpfs uses ramfs_init_fs_context() for the init_fs_context method, which allocates fc->s_fs_info, use ramfs_kill_sb() to free it and avoid a memory leak.
CVE-2023-53395	In the Linux kernel, the following vulnerability has been resolved: ACPICA: Add AML_NO_OPERAND_RESOLVE flag to Timer ACPICA commit 90310989a0790032f5a0140741ff09b545af4bc5 According to the ACPI specification 19.6.134, no argument is required to be passed for ASL Timer instruction. For taking care of no argument, AML_NO_OPERAND_RESOLVE flag is added to ASL Timer instruction opcode. When ASL timer instruction interpreted by ACPI interpreter, getting error. After adding AML_NO_OPERAND_RESOLVE flag to ASL Timer instruction opcode, issue is not observed. ===== UBSAN: array-index-out-of-bounds in acpica/dswexec.c:401:12 index -1 is out of range for type 'union acpi_operand_object *[9]' CPU: 37 PID: 1678 Comm: cat Not tainted 6.0.0-dev-th500-6.0.y-1+bcf8c46459e407-generic-64k HW name: NVIDIA BIOS v1.1.1-d7acbf-dirty 12/19/2022 Call trace: dump_backtrace+0xe0/0x130 show_stack+0x20/0x60 dump_stack_lvl+0x68/0x84 dump_stack+0x18/0x34 ubsan_epilogue+0x10/0x50 __ubsan_handle_out_of_bounds+0x80/0x90 acpi_ds_exec_end_op+0x1bc/0x6d8 acpi_ps_parse_loop+0x57c/0x618 acpi_ps_parse_aml+0x1e0/0x4b4 acpi_ps_execute_method+0x24c/0x2b8 acpi_ns_evaluate+0x3a8/0x4bc acpi_evaluate_object+0x15c/0x37c acpi_evaluate_integer+0x54/0x15c show_power+0x8c/0x12c [acpi_power_meter]
CVE-2023-53398	In the Linux kernel, the following vulnerability has been resolved: mlx5: fix possible ptp queue fifo use-after-free Fifo indexes are not checked during pop operations and it leads to potential use-after-free when popping from empty queue. Such case was possible during re-sync action. WARN_ON_ONCE covers future cases. There were out-of-order cq spotted which lead to drain of the queue and use-after-free because of lack of fifo pointers check. Special check and counter are added to avoid resync operation if SKB could not exist in the fifo because of OOO cq (skb_id must be between consumer and producer index).
CVE-2023-53400	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: Fix Oops by 9.1 surround channel names get_line_out_pfx() may trigger an Oops by overflowing the static array with more than 8 channels. This was reported for MacBookPro 12,1 with Cirrus codec. As a workaround, extend for the 9.1 channels and also fix the potential Oops by unifying the code paths accessing the same array with the proper size check.

CVE ID	Description
CVE-2023-53401	In the Linux kernel, the following vulnerability has been resolved: mm: kmem: fix a NULL pointer dereference in obj_stock_flush_required() KCSAN found an issue in obj_stock_flush_required(): stock->cached_objcg can be reset between the check and dereference: ===== BUG: KCSAN: data-race in drain_all_stock / drain_obj_stock write to 0xffff888237c2a2f8 of 8 bytes by task 19625 on cpu 0: drain_obj_stock+0x408/0x4e0 mm/memcontrol.c:3306 refill_obj_stock+0x9c/0x1e0 mm/memcontrol.c:3340 obj_cgroup_uncharge+0xe/0x10 mm/memcontrol.c:3408 memcg_slab_free_hook mm/slab.h:587 [inline] __cache_free mm/slab.c:3373 [inline] __do_kmem_cache_free mm/slab.c:3577 [inline] kmem_cache_free+0x105/0x280 mm/slab.c:3602 __d_free fs/dcache.c:298 [inline] dentry_free fs/dcache.c:375 [inline] __dentry_kill+0x422/0x4a0 fs/dcache.c:621 dentry_kill+0x8d/0x1e0 dput+0x118/0x1f0 fs/dcache.c:913 __fput+0x3bf/0x570 fs/file_table.c:329 __fput+0x15/0x20 fs/file_table.c:349 task_work_run+0x123/0x160 kernel/task_work.c:179 resume_user_mode_work include/linux/resume_user_mode.h:49 [inline] exit_to_user_mode_loop+0xcf/0xe0 kernel/entry/common.c:171 exit_to_user_mode_prepare+0x6a/0xa0 kernel/entry/common.c:203 __syscall_exit_to_user_mode_work kernel/entry/common.c:285 [inline] syscall_exit_to_user_mode+0x26/0x140 kernel/entry/common.c:296 do_syscall_64+0x4d/0xc0 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x63/0xcd read to 0xffff888237c2a2f8 of 8 bytes by task 19632 on cpu 1: obj_stock_flush_required mm/memcontrol.c:3319 [inline] drain_all_stock+0x174/0x2a0 mm/memcontrol.c:2361 try_charge_memcg+0x6d0/0xd10 mm/memcontrol.c:2703 try_charge mm/memcontrol.c:2837 [inline] mem_cgroup_charge_skmem+0x51/0x140 mm/memcontrol.c:7290 sock_reserve_memory+0xb1/0x390 net/core/sock.c:1025 sk_setsockopt+0x800/0x1e70 net/core/sock.c:1525 udp_lib_setsockopt+0x99/0x6c0 net/ipv4/udp.c:2692 udp_setsockopt+0x73/0xa0 net/ipv4/udp.c:2817 sock_common_setsockopt+0x61/0x70 net/core/sock.c:3668 __sys_setsockopt+0x1c3/0x230 net/socket.c:2271 __do_sys_setsockopt net/socket.c:2282 [inline] __se_sys_setsockopt net/socket.c:2279 [inline] __x64_sys_setsockopt+0x66/0x80 net/socket.c:2279 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x41/0xc0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd value changed: 0xffff8881382d52c0 -> 0xffff888138893740 Reported by Kernel Concurrency Sanitizer on: CPU: 1 PID: 19632 Comm: syz-executor.0 Not tainted 6.3.0-rc2-syzkaller-00387-g534293368afa #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/02/2023 Fix it by using READ_ONCE()/WRITE_ONCE() for all accesses to stock->cached_objcg.
CVE-2023-53408	In the Linux kernel, the following vulnerability has been resolved: trace/blktrace: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-53409	In the Linux kernel, the following vulnerability has been resolved: drivers: base: component: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-53410	In the Linux kernel, the following vulnerability has been resolved: USB: ULP: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.
CVE-2023-53411	In the Linux kernel, the following vulnerability has been resolved: PM: EM: fix memory leak with using debugfs_lookup() When calling debugfs_lookup() the result must have dput() called on it, otherwise the memory will leak over time. To make things simpler, just call debugfs_lookup_and_remove() instead which handles all of the logic at once.

CVE ID	Description
CVE-2023-53427	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix warning and UAF when destroy the MR list If the MR allocate failed, the MR recovery work not initialized and list not cleared. Then will be warning and UAF when release the MR: WARNING: CPU: 4 PID: 824 at kernel/workqueue.c:3066 __flush_work.isra.0+0xf7/0x110 CPU: 4 PID: 824 Comm: mount.cifs Not tainted 6.1.0-rc5+ #82 RIP: 0010: __flush_work.isra.0+0xf7/0x110 Call Trace: <TASK> __cancel_work_timer+0x2ba/0x2e0 smbd_destroy+0x4e1/0x990 __smbd_get_connection+0x1cbd/0x2110 smbd_get_connection+0x21/0x40 cifs_get_tcp_session+0x8ef/0xda0 mount_get_conns+0x60/0x750 cifs_mount+0x103/0xd00 cifs_smb3_do_mount+0x1dd/0xcb0 smb3_get_tree+0x1d5/0x300 vfs_get_tree+0x41/0xf0 path_mount+0x9b3/0xdd0 __x64_sys_mount+0x190/0x1d0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 BUG: KASAN: use-after-free in smbd_destroy+0x4fc/0x990 Read of size 8 at addr ffff88810b156a08 by task mount.cifs/824 CPU: 4 PID: 824 Comm: mount.cifs Tainted: G W 6.1.0-rc5+ #82 Call Trace: dump_stack_lvl+0x34/0x44 print_report+0x171/0x472 kasan_report+0xad/0x130 smbd_destroy+0x4fc/0x990 __smbd_get_connection+0x1cbd/0x2110 smbd_get_connection+0x21/0x40 cifs_get_tcp_session+0x8ef/0xda0 mount_get_conns+0x60/0x750 cifs_mount+0x103/0xd00 cifs_smb3_do_mount+0x1dd/0xcb0 smb3_get_tree+0x1d5/0x300 vfs_get_tree+0x41/0xf0 path_mount+0x9b3/0xdd0 __x64_sys_mount+0x190/0x1d0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 Allocated by task 824: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 __kasan_kmalloc+0x7a/0x90 __smbd_get_connection+0x1b6f/0x2110 smbd_get_connection+0x21/0x40 cifs_get_tcp_session+0x8ef/0xda0 mount_get_conns+0x60/0x750 cifs_mount+0x103/0xd00 cifs_smb3_do_mount+0x1dd/0xcb0 smb3_get_tree+0x1d5/0x300 vfs_get_tree+0x41/0xf0 path_mount+0x9b3/0xdd0 __x64_sys_mount+0x190/0x1d0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 Freed by task 824: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 kasan_save_free_info+0x2a/0x40 ____kasan_slab_free+0x143/0x1b0 __kmem_cache_free+0xc8/0x330 __smbd_get_connection+0x1c6a/0x2110 smbd_get_connection+0x21/0x40 cifs_get_tcp_session+0x8ef/0xda0 mount_get_conns+0x60/0x750 cifs_mount+0x103/0xd00 cifs_smb3_do_mount+0x1dd/0xcb0 smb3_get_tree+0x1d5/0x300 vfs_get_tree+0x41/0xf0 path_mount+0x9b3/0xdd0 __x64_sys_mount+0x190/0x1d0 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 Let's initialize the MR recovery work before MR allocate to prevent the warning, remove the MRs from the list to prevent the UAF.
CVE-2023-53430	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: dma: fix memory leak running mt76_dma_tx_cleanup Fix device unregister memory leak and always cleanup all configured rx queues in mt76_dma_tx_cleanup routine.
CVE-2023-53431	In the Linux kernel, the following vulnerability has been resolved: scsi: ses: Don't attach if enclosure has no components An enclosure with no components can't usefully be operated by the driver (since effectively it has nothing to manage), so report the problem and don't attach. Not attaching also fixes an oops which could occur if the driver tries to manage a zero component enclosure. [mkp: Switched to KERN_WARNING since this scenario is common]
CVE-2023-53432	In the Linux kernel, the following vulnerability has been resolved: firewire: net: fix use after free in fwnet_finish_incoming_packet() The netif_rx() function frees the skb so we can't dereference it to save the skb->len.

CVE ID	Description
CVE-2023-53433	In the Linux kernel, the following vulnerability has been resolved: net: add vlan_get_protocol_and_depth() helper Before blamed commit, pskb_may_pull() was used instead of skb_header_pointer() in __vlan_get_protocol() and friends. Few callers depended on skb->head being populated with MAC header, syzbot caught one of them (skb_mac_gso_segment()) Add vlan_get_protocol_and_depth() to make the intent clearer and use it where sensible. This is a more generic fix than commit e9d3f80935b6 ("net/af_packet: make sure to pull mac header") which was dealing with a similar issue. kernel BUG at include/linux/skbuff.h:2655 ! invalid opcode: 0000 [#1] SMP KASAN CPU: 0 PID: 1441 Comm: syz-executor199 Not tainted 6.1.24-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/14/2023 RIP: 0010: __skb_pull include/linux/skbuff.h:2655 [inline] RIP: 0010:skb_mac_gso_segment+0x68f/0x6a0 net/core/gro.c:136 Code: fd 48 8b 5c 24 10 44 89 6b 70 48 c7 c0 ae 0d 86 44 89 e6 e8 a1 91 d0 00 48 c7 c7 00 af 0d 86 48 89 de 31 d2 e8 d1 4a e9 ff <0f> 0b 66 2e 0f 1f 84 00 00 00 00 0f 1f 44 00 00 55 48 89 e5 41 RSP: 0018:ffffc90001bd7520 EFLAGS: 00010286 RAX: ffffffff8469736a RBX: ffff88810f31dac0 RCX: ffff888115a18b00 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ffff88810f31bd75e8 R08: ffffffff84697183 R09: ffff5200037adf9 R10: 0000000000000000 R11: dffffc0000000001 R12: 0000000000000012 R13: 00000000000000fe5 R14: 0000000000005865 R15: 000000000000fed7 FS: 000055555633f300(0000) GS:ffff8881f6a00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000000 CR3: 0000000116fea000 CR4: 0000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> [<fffffff847018dd>] __skb_gso_segment+0x32d/0x4c0 net/core/dev.c:3419 [<fffffff8470398a>] skb_gso_segment include/linux/netdevice.h:4819 [inline] [<fffffff8470398a>] validate_xmit_skb+0x3aa/0xee0 net/core/dev.c:3725 [<fffffff84707042>] __dev_queue_xmit+0x1332/0x3300 net/core/dev.c:4313 [<fffffff851a9ec7>] dev_queue_xmit+0x17/0x20 include/linux/netdevice.h:3029 [<fffffff851b4a82>] packet_snd net/packet/af_packet.c:3111 [inline] [<fffffff851b4a82>] packet_sendmsg+0x49d2/0x6470 net/packet/af_packet.c:3142 [<fffffff84669a12>] sock_sendmsg_nosec net/socket.c:716 [inline] [<fffffff84669a12>] sock_sendmsg net/socket.c:736 [inline] [<fffffff84669a12>] __sys_sendto+0x472/0x5f0 net/socket.c:2139 [<fffffff84669c75>] __do_sys_sendto net/socket.c:2151 [inline] [<fffffff84669c75>] __se_sys_sendto net/socket.c:2147 [inline] [<fffffff84669c75>] __x64_sys_sendto+0xe5/0x100 net/socket.c:2147 [<fffffff8551d40f>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<fffffff8551d40f>] do_syscall_64+0x2f/0x50 arch/x86/entry/common.c:80 [<fffffff85600087>] entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2023-53437	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Handle cameras with invalid descriptors If the source entity does not contain any pads, do not create a link.
CVE-2023-53438	In the Linux kernel, the following vulnerability has been resolved: x86/MCE: Always save CS register on AMD Zen IF Poison errors The Instruction Fetch (IF) units on current AMD Zen-based systems do not guarantee a synchronous #MC is delivered for poison consumption errors. Therefore, MCG_STATUS[EIPV RIPV] will not be set. However, the microarchitecture does guarantee that the exception is delivered within the same context. In other words, the exact rIP is not known, but the context is known to not have changed. There is no architecturally-defined method to determine this behavior. The Code Segment (CS) register is always valid on such IF unit poison errors regardless of the value of MCG_STATUS[EIPV RIPV]. Add a quirk to save the CS register for poison consumption from the IF unit banks. This is needed to properly determine the context of the error. Otherwise, the severity grading function will assume the context is IN_KERNEL due to the m->cs value being 0 (the initialized value). This leads to unnecessary kernel panics on data poison errors due to the kernel believing the poison consumption occurred in kernel context.
CVE-2023-5344	Heap-based Buffer Overflow in GitHub repository vim/vim prior to 9.0.1969.

CVE ID	Description
CVE-2023-53441	In the Linux kernel, the following vulnerability has been resolved: bpf: cpumap: Fix memory leak in cpu_map_update_elem Syzkaller reported a memory leak as follows: BUG: memory leak unreferenced object 0xff110001198ef748 (size 192): comm "syz-executor.3", pid 17672, jiffies 4298118891 (age 9.906s) hex dump (first 32 bytes): 00 00 00 00 4a 19 00 00 80 ad e3 e4 fe ff c0 00 J..... 00 b2 d3 0c 01 00 11 ff 28 f5 8e 19 01 00 11 ff(..... backtrace: [<ffffffffffadd28087>] __cpu_map_entry_alloc+0xf7/0xb00 [<ffffffffffadd28d8e>] cpu_map_update_elem+0x2fe/0x3d0 [<ffffffffffadc6d0fd>] bpf_map_update_value.isra.0+0x2bd/0x520 [<ffffffffffadc7349b>] map_update_elem+0x4cb/0x720 [<ffffffffffadc7d983>] __se_sys_bpf+0x8c3/0xb90 [<ffffffffffb029cc80>] do_syscall_64+0x30/0x40 [<ffffffffffb0400099>] entry_SYSCALL_64_after_hwframe+0x61/0xc6 BUG: memory leak unreferenced object 0xff110001198ef528 (size 192): comm "syz-executor.3", pid 17672, jiffies 4298118891 (age 9.906s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<ffffffffffadd281f0>] __cpu_map_entry_alloc+0x260/0xb00 [<ffffffffffadd28d8e>] cpu_map_update_elem+0x2fe/0x3d0 [<ffffffffffadc6d0fd>] bpf_map_update_value.isra.0+0x2bd/0x520 [<ffffffffffadc7349b>] map_update_elem+0x4cb/0x720 [<ffffffffffadc7d983>] __se_sys_bpf+0x8c3/0xb90 [<ffffffffffb029cc80>] do_syscall_64+0x30/0x40 [<ffffffffffb0400099>] entry_SYSCALL_64_after_hwframe+0x61/0xc6 BUG: memory leak unreferenced object 0xff1100010fd93d68 (size 8): comm "syz-executor.3", pid 17672, jiffies 4298118891 (age 9.906s) hex dump (first 8 bytes): 00 00 00 00 00 00 00 00 backtrace: [<ffffffffffade5db3e>] kvmalloc_node+0x11e/0x170 [<ffffffffffadd28280>] __cpu_map_entry_alloc+0x2f0/0xb00 [<ffffffffffadd28d8e>] cpu_map_update_elem+0x2fe/0x3d0 [<ffffffffffadc6d0fd>] bpf_map_update_value.isra.0+0x2bd/0x520 [<ffffffffffadc7349b>] map_update_elem+0x4cb/0x720 [<ffffffffffadc7d983>] __se_sys_bpf+0x8c3/0xb90 [<ffffffffffb029cc80>] do_syscall_64+0x30/0x40 [<ffffffffffb0400099>] entry_SYSCALL_64_after_hwframe+0x61/0xc6 In the cpu_map_update_elem flow, when kthread_stop is called before calling the threadfn of rcpu->kthread, since the KTHREAD_SHOULD_STOP bit of kthread has been set by kthread_stop, the threadfn of rcpu->kthread will never be executed, and rcpu->refcnt will never be 0, which will lead to the allocated rcpu, rcpu->queue and rcpu->queue->queue cannot be released. Calling kthread_stop before executing kthread's threadfn will return -EINTR. We can complete the release of memory resources in this state.</ffffffffffb0400099></ffffffffffb029cc80></ffffffffffadc7d983></ffffffffffadc7349b></ffffffffffadc6d0fd></ffffffffffadd28d8e></ffffffffffadd28280></ffffffffffade5db3e></ffffffffffb0400099></ffffffffffb029cc80></ffffffffffadc7d983></ffffffffffadc7349b></ffffffffffadc6d0fd></ffffffffffadd28d8e></ffffffffffadd281f0></ffffffffffb0400099></ffffffffffb029cc80></ffffffffffadc7d983></ffffffffffadc7349b></ffffffffffadc6d0fd></ffffffffffadd28d8e></ffffffffffadd28087>
CVE-2023-53443	In the Linux kernel, the following vulnerability has been resolved: mfd: arizona: Use pm_runtime_resume_and_get() to prevent refcnt leak In arizona_clk32k_enable(), we should use pm_runtime_resume_and_get() as pm_runtime_get_sync() will increase the refcnt even when it returns an error.
CVE-2023-53444	In the Linux kernel, the following vulnerability has been resolved: drm/ttm: fix bulk_move corruption when adding a entry When the resource is the first in the bulk_move range, adding it again (thus moving it to the tail) will corrupt the list since the first pointer is not moved. This eventually lead to null pointer deref in ttm_lru_bulk_move_del()
CVE-2023-53446	In the Linux kernel, the following vulnerability has been resolved: PCI/ASPM: Disable ASPM on MFD function removal to avoid use-after-free Struct pcie_link_state->downstream is a pointer to the pci_dev of function 0. Previously we retained that pointer when removing function 0, and subsequent ASPM policy changes dereferenced it, resulting in a use-after-free warning from KASAN, e.g.: # echo 1 > /sys/bus/pci/devices/0000:03:00.0/remove # echo powersave > /sys/module/pcie_aspm/parameters/policy BUG: KASAN: slab-use-after-free in pcie_config_aspm_link+0x42d/0x500 Call Trace: kasan_report+0xae/0xe0 pcie_config_aspm_link+0x42d/0x500 pcie_aspm_set_policy+0x8e/0x1a0 param_attr_store+0x162/0x2c0 module_attr_store+0x3e/0x80 PCIe spec r6.0, sec 7.5.3.7, recommends that software program the same ASPM Control value in all functions of multi-function devices. Disable ASPM and free the pcie_link_state when any child function is removed so we can discard the dangling pcie_link_state->downstream pointer and maintain the same ASPM Control configuration for all functions. [bhelgaas: commit log and comment]
CVE-2023-53450	In the Linux kernel, the following vulnerability has been resolved: ext4: remove a BUG_ON in ext4_mb_release_group_pa() If a malicious fuzzer overwrites the ext4 superblock while it is mounted such that the s_first_data_block is set to a very large number, the calculation of the block group can underflow, and trigger a BUG_ON check. Change this to be an ext4_warning so that we don't crash the kernel.
CVE-2023-53451	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix potential NULL pointer dereference Klocwork tool reported 'cur_dsd' may be dereferenced. Add fix to validate pointer before dereferencing the pointer.

CVE ID	Description
CVE-2023-53452	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix potential race condition between napi_init and napi_enable A race condition can happen if netdev is registered, but NAPI isn't initialized yet, and meanwhile user space starts the netdev that will enable NAPI. Then, it hits BUG_ON(): kernel BUG at net/core/dev.c:6423! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 417 Comm: iwd Not tainted 6.2.7-slab-dirty #3 eb0f5a8a9d91 Hardware name: LENOVO 21DL/LNVNB161216, BIOS JPCN20WW(V1.06) 09/20/2022 RIP: 0010:napi_enable+0x3f/0x50 Code: 48 89 c2 48 83 e2 f6 f6 81 89 08 00 00 02 74 0d 48 83 ... RSP: 0018:ffffada1414f3548 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffffa01425802080 RCX: 0000000000000000 RDX: 000000000000002f RSI: ffffa014e50c614 RDI: ffffa01425808dc0 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000100 R12: ffffa01425808f58 R13: 0000000000000000 R14: ffffa01423498940 R15: 0000000000000001 FS: 00007f5577c0a740(0000) GS:ffffa0169fc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f5577a19972 CR3: 0000000125a7a000 CR4: 0000000000750ef0 PKRU: 55555554 Call Trace: <TASK> rtw89_pci_ops_start+0x1c/0x70 [rtw89_pci 6cbc75429515c181cbc386478d5cfb32ffc5a0f8] rtw89_core_start+0xb6/0x160 [rtw89_core fe07ecb874820b6d778370d4acb6ef8a37847f22] rtw89_ops_start+0x26/0x40 [rtw89_core fe07ecb874820b6d778370d4acb6ef8a37847f22] drv_start+0x42/0x100 [mac80211 c07fa22af8c3cf3f7d7ab3884ca990784d72e2d2] ieee80211_do_open+0x311/0x7d0 [mac80211 c07fa22af8c3cf3f7d7ab3884ca990784d72e2d2] ieee80211_open+0x6a/0x90 [mac80211 c07fa22af8c3cf3f7d7ab3884ca990784d72e2d2] __dev_open+0xe0/0x180 __dev_change_flags+0x1da/0x250 dev_change_flags+0x26/0x70 do_setlink+0x37c/0x12c0 ? ep_poll_callback+0x246/0x290 ? __nla_validate_parse+0x61/0xd00 ? __wake_up_common_lock+0x8f/0xd0 To fix this, follow Jonas' suggestion to switch the order of these functions and move register_netdev to be the last step of PCI probe. Also, correct the error handling of rtw89_core_register_hw().
CVE-2023-53453	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: free iio for atombios when driver shutdown Fix below kmemleak when unload radeon driver: unreferenced object 0xffff9f8608ede200 (size 512): comm "systemd-udevd", pid 326, jiffies 4294682822 (age 716.338s) hex dump (first 32 bytes): 00 00 00 00 c4 aa ec aa 14 ab 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<0000000062fadebe>] kmem_cache_alloc_trace+0x2f1/0x500 [<00000000b6883cea>] atom_parse+0x117/0x230 [radeon] [<00000000158c23fd>] radeon_atombios_init+0xab/0x170 [radeon] [<00000000683f672e>] si_init+0x57/0x750 [radeon] [<00000000566cc31f>] radeon_device_init+0x559/0x9c0 [radeon] [<0000000046efabb3>] radeon_driver_load_kms+0xc1/0x1a0 [radeon] [<00000000b5155064>] drm_dev_register+0xdd/0x1d0 [<0000000045fec835>] radeon_pci_probe+0xb0/0x100 [radeon] [<00000000e69ecca3>] pci_device_probe+0xe1/0x160 [<0000000019484b76>] really_probe.part.0+0xc1/0x2c0 [<000000003f2649da>] __driver_probe_device+0x96/0x130 [<00000000231c5bb1>] driver_probe_device+0x24/0xf0 [<000000000a42377>] __driver_attach+0x77/0x190 [<00000000d7574da6>] bus_for_each_dev+0x7f/0xd0 [<00000000633166d2>] driver_attach+0x1e/0x30 [<00000000313b05b8>] bus_add_driver+0x12c/0x1e0 iio was allocated in atom_index_iio() called by atom_parse(), but it doesn't get released when the driver is shutdown. Fix this kmemleak by free it in radeon_atombios_fini().
CVE-2023-53454	In the Linux kernel, the following vulnerability has been resolved: HID: multitouch: Correct devm device reference for hidinput input_dev name Reference the HID device rather than the input device for the devm allocation of the input_dev name. Referencing the input_dev would lead to a use-after-free when the input_dev was unregistered and subsequently fires a uevent that depends on the name. At the point of firing the uevent, the name would be freed by devres management. Use devm_kasprintf to simplify the logic for allocating memory and formatting the input_dev name string.
CVE-2023-53456	In the Linux kernel, the following vulnerability has been resolved: scsi: qla4xxx: Add length check when parsing nlattr There are three places that qla4xxx parses nlattrs: - qla4xxx_set_chap_entry() - qla4xxx_iface_set_param() - qla4xxx_sysfs_ddb_set_param() and each of them directly converts the nlattr to specific pointer of structure without length checking. This could be dangerous as those attributes are not validated and a malformed nlattr (e.g., length 0) could result in an OOB read that leaks heap dirty data. Add the nla_len check before accessing the nlattr data and return EINVAL if the length check fails.
CVE-2023-53457	In the Linux kernel, the following vulnerability has been resolved: FS: JFS: Fix null-ptr-deref Read in txBegin Syzkaller reported an issue where txBegin may be called on a superblock in a read-only mounted filesystem which leads to NULL pointer deref. This could be solved by checking if the filesystem is read-only before calling txBegin, and returning with appropriate error code.
CVE-2023-53458	In the Linux kernel, the following vulnerability has been resolved: media: cx23885: Fix a null-ptr-deref bug in buffer_prepare() and buffer_finish() When the driver calls cx23885_risc_buffer() to prepare the buffer, the function call dma_alloc_coherent may fail, resulting in an empty buffer risc->cpu. Later when we free the buffer or access the buffer, null ptr deref is triggered. This bug is similar to the following one: https://git.linuxtv.org/media_stage.git/commit/?id=2b064d91440b33fba5b452f2d1b31f13ae911d71 . We believe the bug can be also dynamically triggered from user side. Similarly, we fix this by checking the return value of cx23885_risc_buffer() and the value of risc->cpu before buffer free.

CVE ID	Description
CVE-2023-53460	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: fix memory leak in <code>rtw_usb_probe()</code> drivers/net/wireless/realtek/rtw88/usb.c:876 <code>rtw_usb_probe()</code> warn: 'hw' from <code>ieee80211_alloc_hw()</code> not released on lines: 811 Fix this by modifying return to a goto statement.
CVE-2023-53463	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Do not reset dql stats on NON_FATAL err All ibmvnic resets, make a call to <code>netdev_tx_reset_queue()</code> when re-opening the device. <code>netdev_tx_reset_queue()</code> resets the <code>num_queued</code> and <code>num_completed</code> byte counters. These stats are used in Byte Queue Limit (BQL) algorithms. The difference between these two stats tracks the number of bytes currently sitting on the physical NIC. ibmvnic increases the number of queued bytes though calls to <code>netdev_tx_sent_queue()</code> in the drivers <code>xmit</code> function. When, VIOS reports that it is done transmitting bytes, the ibmvnic device increases the number of completed bytes through calls to <code>netdev_tx_completed_queue()</code> . It is important to note that the driver batches its transmit calls and <code>num_queued</code> is increased every time that an <code>skb</code> is added to the next batch, not necessarily when the batch is sent to VIOS for transmission. Unlike other reset types, a NON_FATAL reset will not flush the sub <code>crq</code> tx buffers. Therefore, it is possible for the batched <code>skb</code> array to be partially full. So if there is call to <code>netdev_tx_reset_queue()</code> when re-opening the device, the value of <code>num_queued</code> (0) would not account for the <code>skb</code> 's that are currently batched. Eventually, when the batch is sent to VIOS, the call to <code>netdev_tx_completed_queue()</code> would increase <code>num_completed</code> to a value greater than the <code>num_queued</code> . This causes a BUG_ON crash: ibmvnic 30000002: Firmware reports error, cause: adapter problem. Starting recovery... ibmvnic 30000002: tx error 600 ibmvnic 30000002: tx error 600 ibmvnic 30000002: tx error 600 ibmvnic 30000002: tx error 600 ibmvnic 30000002: tx error 600 -----[cut here]----- kernel BUG at lib/dynamic_queue_limits.c:27! Oops: Exception in kernel mode, sig: 5 [...] NIP <code>dql_completed+0x28/0x1c0</code> LR <code>ibmvnic_complete_tx.isra.0+0x23c/0x420</code> [ibmvnic] Call Trace: <code>ibmvnic_complete_tx.isra.0+0x3f8/0x420</code> [ibmvnic] (unreliable) <code>ibmvnic_interrupt_tx+0x40/0x70</code> [ibmvnic] <code>__handle_irq_event_percpu+0x98/0x270</code> ---[end trace]--- Therefore, do not reset the dql stats when performing a NON_FATAL reset.
CVE-2023-53467	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix potential leak in <code>rtw89_append_probe_req_ie()</code> Do <code>'kfree_skb(new)'</code> before <code>'goto out'</code> to prevent potential leak.
CVE-2023-53469	In the Linux kernel, the following vulnerability has been resolved: af_unix: Fix null-ptr-deref in <code>unix_stream_sendpage()</code> . Bing-Jhong Billy Jheng reported null-ptr-deref in <code>unix_stream_sendpage()</code> with detailed analysis and a nice repro. <code>unix_stream_sendpage()</code> tries to add data to the last skb in the peer's <code>recv</code> queue without locking the queue. If the peer's FD is passed to another socket and the socket's FD is passed to the peer, there is a loop between them. If we close both sockets without receiving FD, the sockets will be cleaned up by garbage collection. The garbage collection iterates such sockets and unlinks <code>skb</code> with FD from the socket's receive queue under the queue's lock. So, there is a race where <code>unix_stream_sendpage()</code> could access an <code>skb</code> locklessly that is being released by garbage collection, resulting in use-after-free. To avoid the issue, <code>unix_stream_sendpage()</code> must lock the peer's <code>recv</code> queue. Note the issue does not exist in 6.5+ thanks to the recent <code>sendpage()</code> refactoring. This patch is originally written by Linus Torvalds. BUG: unable to handle page fault for address: ffff988004dd6870 PF: supervisor read access in kernel mode PF: error_code(0x0000) - not-present page PGD 0 P4D 0 PREEMPT SMP PTI CPU: 4 PID: 297 Comm: garbage_uaf Not tainted 6.1.46 #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 RIP: 0010:kmem_cache_alloc_node+0xa2/0x1e0 Code: c0 0f 84 32 01 00 00 41 83 fd ff 74 10 48 8b 00 48 c1 e8 3a 41 39 c5 0f 85 1c 01 00 00 41 8b 44 24 28 49 8b 3c 24 48 8d 4a 40 <49> 8b 1c 06 4c 89 f0 65 48 0f c7 0f 0f 94 c0 84 c0 74 a1 41 8b 44 RSP: 0018:ffff9000079fac0 EFLAGS: 00000246 RAX: 0000000000000070 RBX: 0000000000000005 RCX: 0000000000001a284 RDX: 0000000000001a244 RSI: 0000000000400cc0 RDI: 00000000002eee0 RBP: 0000000000400cc0 R08: 0000000000400cc0 R09: 0000000000000003 R10: 0000000000000001 R11: 0000000000000000 R12: ffff888003970f00 R13: 00000000ffffffff R14: ffff988004dd6800 R15: 00000000000000e8 FS: 00007f174d6f3600(0000) GS:ffff88807db00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 00000000080050033 CR2: ffff988004dd6870 CR3: 000000000092be000 CR4: 00000000007506e0 PKRU: 55555554 Call Trace: <TASK> ? <code>__die_body.cold+0x1a/0x1f</code> ? <code>page_fault_oops+0xa9/0x1e0</code> ? <code>fixup_exception+0x1d/0x310</code> ? <code>exc_page_fault+0xa8/0x150</code> ? <code>asm_exc_page_fault+0x22/0x30</code> ? <code>kmem_cache_alloc_node+0xa2/0x1e0</code> ? <code>__alloc_skb</code> +0x16c/0x1e0 <code>__alloc_skb+0x16c/0x1e0</code> <code>alloc_skb_with_frags+0x48/0x1e0</code> <code>sock_alloc_send_pskb</code> +0x234/0x270 <code>unix_stream_sendmsg+0x1f5/0x690</code> <code>sock_sendmsg+0x5d/0x60</code> <code>__sys_sendmsg</code> +0x210/0x260 <code>__sys_sendmsg+0x83/0xd0</code> ? <code>kmem_cache_alloc+0xc6/0x1c0</code> ? <code>avc_disable</code> +0x20/0x20 ? <code>percpu_counter_add_batch+0x53/0xc0</code> ? <code>alloc_empty_file+0x5d/0xb0</code> ? <code>alloc_file</code> +0x91/0x170 ? <code>alloc_file_pseudo+0x94/0x100</code> ? <code>__fget_light+0x9f/0x120</code> <code>__sys_sendmsg+0x54/0xa0</code> <code>do_syscall_64+0x3b/0x90</code> entry_SYSCALL_64_after_hwframe+0x69/0xd3 RIP: 0033:0x7f174d639a7d Code: 28 89 54 24 1c 48 89 74 24 10 89 7c 24 08 e8 8a c1 f4 ff 8b 54 24 1c 48 8b 74 24 10 41 89 c0 8b 7c 24 08 b8 2e 00 00 0f 05 <48> 3d 00 f0 ff ff 77 33 44 89 c7 48 89 44 24 08 e8 de c1 f4 ff 48 RSP: 002b:00007fcb563ea50 EFLAGS: 00000293 ORIG_RAX: 000000000000002e RAX: ffffffffda RBX: 0000000000000000 RCX: 00007f174d639a7d RDX: 0000000000000000 RSI: 00007fcb563ea50 RDI: 0000000000000007 RBP: 00007fcb563eb10 R08: 0000000000000000 R09: 0000000000000000 R10: 00000000004040a0 R11: 0000000000000293 R12: 00007fcb563ec28 R13: 0000000000401398 R14: 0000000000403e00 R15: 00007f174d72c000 </TASK>

CVE ID	Description
CVE-2023-53470	In the Linux kernel, the following vulnerability has been resolved: ionic: catch failure from devlink_alloc Add a check for NULL on the alloc return. If devlink_alloc() fails and we try to use devlink_priv() on the NULL return, the kernel gets very unhappy and panics. With this fix, the driver load will still fail, but at least it won't panic the kernel.
CVE-2023-53473	In the Linux kernel, the following vulnerability has been resolved: ext4: improve error handling from ext4_dirhash() The ext4_dirhash() will *almost* never fail, especially when the hash tree feature was first introduced. However, with the addition of support of encrypted, casefolded file names, that function can most certainly fail today. So make sure the callers of ext4_dirhash() properly check for failures, and reflect the errors back up to their callers.
CVE-2023-53474	In the Linux kernel, the following vulnerability has been resolved: x86/MCE/AMD: Use an u64 for bank_map The maximum number of MCA banks is 64 (MAX_NR_BANKS), see a0bc32b3cacf ("x86/mce: Increase maximum number of banks to 64"). However, the bank_map which contains a bitfield of which banks to initialize is of type unsigned int and that overflows when those bit numbers are >= 32, leading to UBSAN complaining correctly: UBSAN: shift-out-of-bounds in arch/x86/kernel/cpu/mce/amd.c:1365:38 shift exponent 32 is too large for 32-bit type 'int' Change the bank_map to a u64 and use the proper BIT_U64() macro when modifying bits in there. [bp: Rewrite commit message.]
CVE-2023-53476	In the Linux kernel, the following vulnerability has been resolved: iw_cxgb4: Fix potential NULL dereference in c4iw_fill_res_cm_id_entry() This condition needs to match the previous "if (epcp->state == LISTEN) {" exactly to avoid a NULL dereference of either "listen_ep" or "ep". The problem is that "epcp" has been re-assigned so just testing "if (epcp->state == LISTEN) {" a second time is not sufficient.
CVE-2023-53477	In the Linux kernel, the following vulnerability has been resolved: ipv6: Add lwtunnel encaps size of all siblings in nexthop calculation In function rt6_nmsg_size(), the length of nexthop is calculated by multiplying the nexthop length of fib6_info and the number of siblings. However if the fib6_info has no lwtunnel but the siblings have lwtunnels, the nexthop length is less than it should be, and it will trigger a warning in inet6_rt_notify() as follows: WARNING: CPU: 0 PID: 6082 at net/ipv6/route.c:6180 inet6_rt_notify+0x120/0x130 Call Trace: <TASK> fib6_add_rt2node+0x685/0xa30 fib6_add+0x96/0x1b0 ip6_route_add+0x50/0xd0 inet6_rtm_newroute+0x97/0xa0 rtnetlink_rcv_msg+0x156/0x3d0 netlink_rcv_skb+0x5a/0x110 netlink_unicast+0x246/0x350 netlink_sendmsg+0x250/0x4c0 sock_sendmsg+0x66/0x70 __sys_sendmsg+0x7c/0xd0 __sys_sendmsg+0x5d/0xb0 do_syscall_64+0x3f/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc This bug can be reproduced by script: ip -6 addr add 2002::2/64 dev ens2 ip -6 route add 100::/64 via 2002::1 dev ens2 metric 100 for i in 10 20 30 40 50 60 70; do ip link add link ens2 name ipv_\$i type ipvlan ip -6 addr add 2002::\$/64 dev ipv_\$i ifconfig ipv_\$i up done for i in 10 20 30 40 50 60; do ip -6 route append 100::/64 encap ip6 dst 2002::\$/1 via 2002::1 dev ipv_\$i metric 100 done ip -6 route append 100::/64 via 2002::1 dev ipv_\$i metric 100 This patch fixes it by adding nexthop_len of every siblings using rt6_nh_nmsg_size().
CVE-2023-53480	In the Linux kernel, the following vulnerability has been resolved: kobject: Add sanity check for kset->kobj.ktype in kset_register() When I register a kset in the following way: static struct kset my_kset; kobject_set_name(&my_kset.kobj, "my_kset"); ret = kset_register(&my_kset); A null pointer dereference exception is occurred: [4453.568337] Unable to handle kernel NULL pointer dereference at \ virtual address 0000000000000028 ... [4453.810361] Call trace: [4453.813062] kobject_get_ownership+0xc/0x34 [4453.817493] kobject_add_internal+0x98/0x274 [4453.822005] kset_register+0x5c/0xb4 [4453.825820] my_kobj_init+0x44/0x1000 [my_kset] ... Because I didn't initialize my_kset.kobj.ktype. According to the description in Documentation/core-api/kobject.rst: - A ktype is the type of object that embeds a kobject. Every structure that embeds a kobject needs a corresponding ktype. So add sanity check to make sure kset->kobj.ktype is not NULL.
CVE-2023-53481	In the Linux kernel, the following vulnerability has been resolved: ubi: ubi_wl_put_peb: Fix infinite loop when wear-leveling work failed Following process will trigger an infinite loop in ubi_wl_put_peb(): ubifs_bgt ubi_bgt ubifs_leb_unmap ubi_leb_unmap ubi_eba_unmap_leb ubi_wl_put_peb wear_leveling_worker e1 = rb_entry(rb_first(&ubi->used) e2 = get_peb_for_wl(ubi) ubi_io_read_vid_hdr // return err (flash fault) out_error: ubi->move_from = ubi->move_to = NULL wl_entry_destroy(ubi, e1) ubi->lookuptbl[e->pnum] = NULL retry: e = ubi->lookuptbl[pnum]; // return NULL if (e == ubi->move_from) { // NULL == NULL gets true goto retry; // infinite loop !!! \$ top PID USER PR NI VIRT RES SHR S %CPU %MEM COMMAND 7676 root 20 0 0 0 R 100.0 0.0 ubifs_bgt0_0 Fix it by: 1) Letting ubi_wl_put_peb() returns directly if wearl leveling entry has been removed from 'ubi->lookuptbl'. 2) Using 'ubi->wl_lock' protecting wl entry deletion to preventing an use-after-free problem for wl entry in ubi_wl_put_peb(). Fetch a reproducer in [Link].
CVE-2023-53482	In the Linux kernel, the following vulnerability has been resolved: iommu: Fix error unwind in iommu_group_alloc() If either iommu_group_grate_file() fails then the iommu_group is leaked. Destroy it on these error paths. Found by kselftest/iommu/iommufd_fail_nth

CVE ID	Description
CVE-2023-53484	In the Linux kernel, the following vulnerability has been resolved: lib: cpu_rmap: Avoid use after free on rmap->obj array entries When calling irq_set_affinity_notifier() with NULL at the notify argument, it will cause freeing of the glue pointer in the corresponding array entry but will leave the pointer in the array. A subsequent call to free_irq_cpu_rmap() will try to free this entry again leading to possible use after free. Fix that by setting NULL to the array entry and checking that we have non-zero at the array entry when iterating over the array in free_irq_cpu_rmap(). The current code does not suffer from this since there are no cases where irq_set_affinity_notifier(irq, NULL) (note the NULL passed for the notify arg) is called, followed by a call to free_irq_cpu_rmap() so we don't hit and issue. Subsequent patches in this series excersize this flow, hence the required fix.
CVE-2023-53487	In the Linux kernel, the following vulnerability has been resolved: powerpc/rtas_flash: allow user copy to flash block cache objects With hardened usercopy enabled (CONFIG_HARDENED_USERCOPY=y), using the /proc/powerpc/rtas/firmware_update interface to prepare a system firmware update yields a BUG(): kernel BUG at mm/usercopy.c:102! Oops: Exception in kernel mode, sig: 5 [#1] LE PAGE_SIZE=64K MMU=Hash SMP NR_CPUS=2048 NUMA pSeries Modules linked in: CPU: 0 PID: 2232 Comm: dd Not tainted 6.5.0-rc3+ #2 Hardware name: IBM,8408-E8E POWER8E (raw) 0x4b0201 0xf000004 of:IBM,FW860.50 (SV860_146) hv:phyp pSeries NIP: c0000000005991d0 LR: c0000000005991cc CTR: 0000000000000000 REGS: c0000000148c76a0 TRAP: 0700 Not tainted (6.5.0-rc3+) MSR: 800000000029033 <SF,EE,ME,IR,DR,RI,LE> CR: 24002242 XER: 0000000c CFAR: c0000000001fbd34 IRQMASK: 0 [... GPRs omitted ...] NIP usercopy_abort+0xa0/0xb0 LR usercopy_abort+0x9c/0xb0 Call Trace: usercopy_abort+0x9c/0xb0 (unreliable) __check_heap_object+0x1b4/0x1d0 __check_object_size+0x2d0/0x380 rtas_flash_write+0xe4/0x250 proc_reg_write+0xfc/0x160 vfs_write+0xfc/0x4e0 ksys_write+0x90/0x160 system_call_exception+0x178/0x320 system_call_common+0x160/0x2c4 The blocks of the firmware image are copied directly from user memory to objects allocated from flash_block_cache, so flash_block_cache must be created using kmem_cache_create_usercopy() to mark it safe for user access. [mpe: Trim and indent oops]
CVE-2023-53488	In the Linux kernel, the following vulnerability has been resolved: IB/hfi1: Fix possible panic during hotplug remove During hotplug remove it is possible that the update counters work might be pending, and may run after memory has been freed. Cancel the update counters work before freeing memory.

CVE ID	Description
CVE-2023-53489	In the Linux kernel, the following vulnerability has been resolved: tcp/udp: Fix memleaks of sk and zerocopy skbs with TX timestamp. syzkaller reported [0] memory leaks of an UDP socket and ZEROCOPY skbs. We can reproduce the problem with these sequences: sk = socket(AF_INET, SOCK_DGRAM, 0) sk.setsockopt(SOL_SOCKET, SO_TIMESTAMPING, SOF_TIMESTAMPING_TX_SOFTWARE) sk.setsockopt(SOL_SOCKET, SO_ZEROCOPY, 1) sk.sendto(b'', MSG_ZEROCOPY, ('127.0.0.1', 53)) sk.close() sendmsg() calls msg_zerocopy_alloc(), which allocates a skb, sets skb->cb->ubuf.refcnt to 1, and calls sock_hold(). Here, struct ubuf_info_msgzrc indirectly holds a refcnt of the socket. When the skb is sent, __skb_tstamp_tx() clones it and puts the clone into the socket's error queue with the TX timestamp. When the original skb is received locally, skb_copy_ubufs() calls skb_unclone(), and pskb_expand_head() increments skb->cb->ubuf.refcnt. This additional count is decremented while freeing the skb, but struct ubuf_info_msgzrc still has a refcnt, so __msg_zerocopy_callback() is not called. The last refcnt is not released unless we retrieve the TX timestamped skb by rcvmsg(). Since we clear the error queue in inet_sock_destruct() after the socket's refcnt reaches 0, there is a circular dependency. If we close() the socket holding such skbs, we never call sock_put() and leak the count, sk, and skb. TCP has the same problem, and commit e0c8bccd40fc ("net: stream: purge sk_error_queue in sk_stream_kill_queues()") tried to fix it by calling skb_queue_purge() during close(). However, there is a small chance that skb queued in a qdisc or device could be put into the error queue after the skb_queue_purge() call. In __skb_tstamp_tx(), the cloned skb should not have a reference to the ubuf to remove the circular dependency, but skb_clone() does not call skb_copy_ubufs() for zerocopy skb. So, we need to call skb_orphan_frags_rx() for the cloned skb to call skb_copy_ubufs(). [0]: BUG: memory leak unreferenced object 0xffff88800c6d2d00 (size 1152): comm "syz-executor392", pid 264, jiffies 4294785440 (age 13.044s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 cd af e8 81 00 00 00 00 02 00 07 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...@..... backtrace: [<0000000055636812>] sk_prot_alloc+0x64/0x2a0 net/core/sock.c:2024 [<0000000054d77b7a>] sk_alloc+0x3b/0x800 net/core/sock.c:2083 [<0000000066f3c7e0>] inet_create net/ipv4/af_inet.c:319 [inline] [<0000000066f3c7e0>] inet_create+0x31e/0xe40 net/ipv4/af_inet.c:245 [<000000009b83af97>] __sock_create+0x2ab/0x550 net/socket.c:1515 [<00000000b9b11231>] sock_create net/socket.c:1566 [inline] [<00000000b9b11231>] __sys_socket_create net/socket.c:1603 [inline] [<00000000b9b11231>] __sys_socket_create net/socket.c:1588 [inline] [<00000000b9b11231>] __sys_socket+0x138/0x250 net/socket.c:1636 [<000000004fb45142>] __do_sys_socket net/socket.c:1649 [inline] [<000000004fb45142>] __se_sys_socket net/socket.c:1647 [inline] [<000000004fb45142>] __x64_sys_socket+0x73/0xb0 net/socket.c:1647 [<0000000066999e0e>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<0000000066999e0e>] do_syscall_64+0x38/0x90 arch/x86/entry/common.c:80 [<0000000017f238c1>] entry_SYSCALL_64_after_hwframe+0x63/0xcd BUG: memory leak unreferenced object 0xffff888017633a00 (size 240): comm "syz-executor392", pid 264, jiffies 4294785440 (age 13.044s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2d 6d 0c 80 88 ff ff-m..... backtrace: [<000000002b1c4368>] __alloc_skb+0x229/0x320 net/core/skbuff.c:497 [<00000000143579a6>] alloc_skb include/linux/skbuff.h:1265 [inline] [<00000000143579a6>] sock_omalloc+0xaa/0x190 net/core/sock.c:2596 [<00000000be626478>] msg_zerocopy_alloc net/core/skbuff.c:1294 [inline] [<00000000be626478>] ---truncated---
CVE-2023-53491	In the Linux kernel, the following vulnerability has been resolved: start_kernel: Add __no_stack_protector function attribute Back during the discussion of commit a9a3ed1eff36 ("x86: Fix early boot crash on gcc-10, third try") we discussed the need for a function attribute to control the omission of stack protectors on a per-function basis; at the time Clang had support for no_stack_protector but GCC did not. This was fixed in gcc-11. Now that the function attribute is available, let's start using it. Callers of boot_init_stack_canary need to use this function attribute unless they're compiled with -fno-stack-protector, otherwise the canary stored in the stack slot of the caller will differ upon the call to boot_init_stack_canary. This will lead to a call to __stack_chk_fail() then panic.
CVE-2023-53496	In the Linux kernel, the following vulnerability has been resolved: x86/platform/uv: Use alternate source for socket to node data The UV code attempts to build a set of tables to allow it to do bidirectional socket<=>node lookups. But when nr_cpus is set to a smaller number than actually present, the cpu_to_node() mapping information for unused CPUs is not available to build_socket_tables(). This results in skipping some nodes or sockets when creating the tables and leaving some -1's for later code to trip. over, causing oopses. The problem is that the socket<=>node lookups are created by doing a loop over all CPUs, then looking up the CPU's APICID and socket. But if a CPU is not present, there is no way to start this lookup. Instead of looping over all CPUs, take CPUs out of the equation entirely. Loop over all APICIDs which are mapped to a valid NUMA node. Then just extract the socket-id from the APICID. This avoid tripping over disabled CPUs.
CVE-2023-53499	In the Linux kernel, the following vulnerability has been resolved: virtio_net: Fix error unwinding of XDP initialization When initializing XDP in virtnet_open(), some rq xdp initialization may hit an error causing net device open failed. However, previous rqs have already initialized XDP and enabled NAPI, which is not the expected behavior. Need to roll back the previous rq initialization to avoid leaks in error unwinding of init code. Also extract helper functions of disable and enable queue pairs. Use newly introduced disable helper function in error unwinding and virtnet_close. Use enable helper function in virtnet_open.

CVE ID	Description
CVE-2023-53500	In the Linux kernel, the following vulnerability has been resolved: xfrm: fix slab-use-after-free in decode_session6 When the xfrm device is set to the qdisc of the sfb type, the cb field of the sent skb may be modified during enqueueing. Then, slab-use-after-free may occur when the xfrm device sends IPv6 packets. The stack information is as follows: BUG: KASAN: slab-use-after-free in decode_session6+0x103f/0x1890 Read of size 1 at addr ffff8881111458ef by task swapper/3/0 CPU: 3 PID: 0 Comm: swapper/3 Not tainted 6.4.0-next-20230707 #409 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.14.0-1.fc33 04/01/2014 Call Trace: <IRQ> dump_stack_lvl+0xd9/0x150 print_address_description.constprop.0+0x2c/0x3c0 kasan_report+0x11d/0x130 decode_session6+0x103f/0x1890 __xfrm_decode_session+0x54/0xb0 xfrm_i_xmit+0x173/0x1ca0 dev_hard_start_xmit+0x187/0x700 sch_direct_xmit+0x1a3/0xc30 __qdisc_run+0x510/0x17a0 __dev_queue_xmit+0x2215/0x3b10 neigh_connected_output+0x3c2/0x550 ip6_finish_output2+0x55a/0x1550 ip6_finish_output+0x6b9/0x1270 ip6_output+0x1f1/0x540 ndisc_send_skb+0xa63/0x1890 ndisc_send_rs+0x132/0x6f0 addrconf_rs_timer+0x3f1/0x870 call_timer_fn+0x1a0/0x580 expire_timers+0x29b/0x4b0 run_timer_softirq+0x326/0x910 __do_softirq+0x1d4/0x905 irq_exit_rcu+0xb7/0x120 sysvec_apic_timer_interrupt+0x97/0xc0 </IRQ> <TASK> asm_sysvec_apic_timer_interrupt+0x1a/0x20 RIP: 0010:intel_idle_hlt+0x23/0x30 Code: 1f 84 00 00 00 00 f3 0f 1e fa 41 54 41 89 d4 0f 1f 44 00 00 66 90 0f 1f 44 00 00 0f 02 d4 9f ab 00 0f 1f 44 00 00 fb f4 <fa> 44 89 e0 41 5c c3 66 0f 1f 44 00 00 f3 0f 1e fa 41 54 41 89 d4 RSP: 0018:ffff90000197d78 EFLAGS: 00000246 RAX: 00000000000a83c3 RBX: fffff8ffffd09c50 RCX: ffffffff8a22d8e5 RDX: 0000000000000001 RSI: ffffffff8d3f8080 RDI: fffff8ffffd09c50 RBP: ffffffff8d3f8080 R08: 0000000000000001 R09: fffffd1026ba6d9d R10: fffff888135d36ceb R11: 0000000000000001 R12: 0000000000000001 R13: ffffffff8d3f8100 R14: 0000000000000001 R15: 0000000000000000 cpuidle_enter_state+0xd3/0x6f0 cpuidle_enter+0x4e/0xa0 do_idle+0x2fe/0x3c0 cpu_startup_entry+0x18/0x20 start_secondary+0x200/0x290 secondary_startup_64_no_verify+0x167/0x16b </TASK> Allocated by task 939: kasan_save_stack+0x22/0x40 kasan_set_track+0x25/0x30 __kasan_slab_alloc+0x7f/0x90 kmem_cache_alloc_node+0x1cd/0x410 kmalloc_reserve+0x165/0x270 __alloc_skb+0x129/0x330 inet6_ifa_notify+0x118/0x230 __ipv6_ifa_notify+0x177/0xbe0 addrconf_dad_completed+0x133/0xe00 addrconf_dad_work+0x764/0x1390 process_one_work+0xa32/0x16f0 worker_thread+0x67d/0x10c0 kthread+0x344/0x440 ret_from_fork+0x1f/0x30 The buggy address belongs to the object at ffff888111145800 which belongs to the cache skbuff_small_head of size 640 The buggy address is located 239 bytes inside of freed 640-byte region [ffff888111145800, ffff888111145a80) As commit f855691975bb ("xfrm6: Fix the nexthdr offset in _decode_session6.") showed, xfrm_decode_session was originally intended only for the receive path. IP6CB(skb)->nhoff is not set during transmission. Therefore, set the cb field in the skb to 0 before sending packets.
CVE-2023-53501	In the Linux kernel, the following vulnerability has been resolved: iommu/amd/iommu_v2: Fix pasid_state refcount dec hit 0 warning on pasid unbind When unbinding pasid - a race condition exists vs outstanding page faults. To prevent this, the pasid_state object contains a refcount. * set to 1 on pasid bind * incremented on each ppr notification start * decremented on each ppr notification done * decremented on pasid unbind Since refcount_dec assumes that refcount will never reach 0: the current implementation causes the following to be invoked on pasid unbind: REFCOUNT_WARN("decrement hit 0; leaking memory") Fix this issue by changing refcount_dec to refcount_dec_and_test to explicitly handle refcount=1.
CVE-2023-53503	In the Linux kernel, the following vulnerability has been resolved: ext4: allow ext4_get_group_info() to fail Previously, ext4_get_group_info() would treat an invalid group number as BUG(), since in theory it should never happen. However, if a malicious attacker (or fuzzer) modifies the superblock via the block device while it is the file system is mounted, it is possible for s_first_data_block to get set to a very large number. In that case, when calculating the block group of some block number (such as the starting block of a preallocation region), could result in an underflow and very large block group number. Then the BUG_ON check in ext4_get_group_info() would fire, resulting in a denial of service attack that can be triggered by root or someone with write access to the block device. For a quality of implementation perspective, it's best that even if the system administrator does something that they shouldn't, that it will not trigger a BUG. So instead of BUG'ing, ext4_get_group_info() will call ext4_error and return NULL. We also add fallback code in all of the callers of ext4_get_group_info() that it might NULL. Also, since ext4_get_group_info() was already borderline to be an inline function, un-inline it. The results in a next reduction of the compiled text size of ext4 by roughly 2k.
CVE-2023-53506	In the Linux kernel, the following vulnerability has been resolved: udf: Do not bother merging very long extents When merging very long extents we try to push as much length as possible to the first extent. However this is unnecessarily complicated and not really worth the trouble. Furthermore there was a bug in the logic resulting in corrupting extents in the file as syzbot reproducer shows. So just don't bother with the merging of extents that are too long together.

CVE ID	Description
CVE-2023-53509	In the Linux kernel, the following vulnerability has been resolved: qed: allow sleep in qed_mcp_trace_dump() By default, qed_mcp_cmd_and_union() delays 10us at a time in a loop that can run 500K times, so calls to qed_mcp_nvmm_rd_cmd() may block the current thread for over 5s. We observed thread scheduling delays over 700ms in production, with stacktraces pointing to this code as the culprit. qed_mcp_trace_dump() is called from ethtool, so sleeping is permitted. It already can sleep in qed_mcp_halt(), which calls qed_mcp_cmd(). Add a "can sleep" parameter to qed_find_nvram_image() and qed_nvram_read() so they can sleep during qed_mcp_trace_dump(). qed_mcp_trace_get_meta_info() and qed_mcp_trace_read_meta(), called only by qed_mcp_trace_dump(), allow these functions to sleep. I can't tell if the other caller (qed_grc_dump_mcp_hw_dump()) can sleep, so keep b_can_sleep set to false when it calls these functions. An example stacktrace from a custom warning we added to the kernel showing a thread that has not scheduled despite long needing resched: [2745.362925,17] -----[cut here]----- [2745.362941,17] WARNING: CPU: 23 PID: 5640 at arch/x86/kernel/irq.c:233 do_IRQ+0x15e/0x1a0() [2745.362946,17] Thread not rescheduled for 744 ms after irq 99 [2745.362956,17] Modules linked in: ... [2745.363339,17] CPU: 23 PID: 5640 Comm: lldpd Tainted: P O 4.4.182+ #202104120910+6d1da174272d.61x [2745.363343,17] Hardware name: FOXCONN MercuryB/Quicksilver Controller, BIOS H11P1N09 07/08/2020 [2745.363346,17] 0000000000000000 ffff885ec07c3ed8 ffffffff8131eb2f ffff885ec07c3f20 [2745.363358,17] ffffffff81d14f64 ffff885ec07c3f10 ffffffff81072ac2 ffff88be98ed0000 [2745.363369,17] 00000000000000063 0000000000000174 0000000000000074 0000000000000000 [2745.363379,17] Call Trace: [2745.363382,17] <IRQ> [<ffffffffff8131eb2f>] dump_stack+0x8e/0xcfc [2745.363393,17] [<ffffffffff81072ac2>] warn_slowpath_common+0x82/0xc0 [2745.363398,17] [<ffffffffff81072b4c>] warn_slowpath_fmt+0x4c/0x50 [2745.363404,17] [<ffffffffff810d5a8e>] ? rcu_irq_exit+0xae/0xc0 [2745.363408,17] [<ffffffffff817c99fe>] do_IRQ+0x15e/0x1a0 [2745.363413,17] [<ffffffffff817c7ac9>] common_interrupt+0x89/0x89 [2745.363416,17] <EOI> [<ffffffffff8132aa74>] ? delay_tsc+0x24/0x50 [2745.363425,17] [<ffffffffff8132aa04>] __udelay+0x34/0x40 [2745.363457,17] [<ffffffffffa04d45ff>] qed_mcp_cmd_and_union+0x36f/0x7d0 [qed] [2745.363473,17] [<ffffffffffa04d5ced>] qed_mcp_nvmm_rd_cmd+0x4d/0x90 [qed] [2745.363490,17] [<ffffffffffa04e1dc7>] qed_mcp_trace_dump+0x4a7/0x630 [qed] [2745.363504,17] [<ffffffffffa04e2556>] ? qed_fw_asserts_dump+0x1d6/0x1f0 [qed] [2745.363520,17] [<ffffffffffa04e4ea7>] qed_dbg_mcp_trace_get_dump_buf_size+0x37/0x80 [qed] [2745.363536,17] [<ffffffffffa04e4881>] qed_dbg_feature_size+0x61/0xa0 [qed] [2745.363551,17] [<ffffffffffa04eb427>] qed_dbg_all_data_size+0x247/0x260 [qed] [2745.363560,17] [<ffffffffffa0482c10>] qed_get_regs_len+0x30/0x40 [qed] [2745.363566,17] [<ffffffffff816c9783>] ethtool_get_drvinfo+0xe3/0x190 [2745.363570,17] [<ffffffffff816cc152>] dev_ethtool+0x1362/0x2140 [2745.363575,17] [<ffffffffff8109bcc6>] ? finish_task_switch+0x76/0x260 [2745.363580,17] [<ffffffffff817c2116>] ? __schedule+0x3c6/0x9d0 [2745.363585,17] [<ffffffffff810dbd50>] ? hrtimer_start_range_ns+0x1d0/0x370 [2745.363589,17] [<ffffffffff816c1e5b>] ? dev_get_by_name_rcu+0x6b/0x90 [2745.363594,17] [<ffffffffff816de6a8>] dev_ioctl+0xe8/0x710 [2745.363599,17] [<ffffffffff816a58a8>] sock_do_ioctl+0x48/0x60 [2745.363603,17] [<ffffffffff816a5d87>] sock_ioctl+0x1c7/0x280 [2745.363608,17] [<ffffffffff8111f393>] ? seccomp_phase1+0x83/0x220 [2745.363612,17] [<ffffffffff811e3503>] do_vfs_ioctl+0x2b3/0x4e0 [2745.363616,17] [<ffffffffff811e3771>] SyS_ioctl+0x41/0x70 [2745.363619,17] [<ffffffffff817c6ffe>] entry_SYSCALL_64_fastpath+0x1e/0x79 [2745.363622,17] ---[end trace f6954aa440266421]---
CVE-2023-53512	In the Linux kernel, the following vulnerability has been resolved: scsi: mpt3sas: Fix a memory leak Add a forgotten kfree().
CVE-2023-53513	In the Linux kernel, the following vulnerability has been resolved: nbd: fix incomplete validation of ioctl arg We tested and found an alarm caused by nbd_ioctl arg without verification. The UBSAN warning calltrace like below: UBSAN: Undefined behaviour in fs/buffer.c:1709:35 signed integer overflow: -9223372036854775808 - 1 cannot be represented in type 'long long int' CPU: 3 PID: 2523 Comm: syz-executor.0 Not tainted 4.19.90 #1 Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0x0/0x3f0 arch/arm64/kernel/time.c:78 show_stack+0x28/0x38 arch/arm64/kernel/traps.c:158 __dump_stack lib/dump_stack.c:77 [inline] dump_stack+0x170/0x1dc lib/dump_stack.c:118 ubsan_epilogue+0x18/0xb4 lib/ubsan.c:161 handle_overflow+0x188/0x1dc lib/ubsan.c:192 __ubsan_handle_sub_overflow+0x34/0x44 lib/ubsan.c:206 __block_write_full_page+0x94c/0xa20 fs/buffer.c:1709 block_write_full_page+0x1f0/0x280 fs/buffer.c:2934 blkdev_writepage+0x34/0x40 fs/block_dev.c:607 __writepage+0x68/0xe8 mm/page-writeback.c:2305 write_cache_pages+0x44c/0xc70 mm/page-writeback.c:2240 generic_writepages+0xdc/0x148 mm/page-writeback.c:2329 blkdev_writepages+0x2c/0x38 fs/block_dev.c:2114 do_writepages+0xd4/0x250 mm/page-writeback.c:2344 The reason for triggering this warning is __block_write_full_page() -> i_size_read(inode) - 1 overflow. inode->i_size is assigned in __nbd_ioctl() -> nbd_set_size() -> bytesize. We think it is necessary to limit the size of arg to prevent errors. Moreover, __nbd_ioctl() -> nbd_add_socket(), arg will be cast to int. Assuming the value of arg is 0x8000000000000001 (on a 64-bit machine), it will become 1 after the coercion, which will return unexpected results. Fix it by adding checks to prevent passing in too large numbers.

CVE ID	Description
CVE-2023-53515	In the Linux kernel, the following vulnerability has been resolved: virtio-mmio: don't break lifecycle of vm_dev vm_dev has a separate lifecycle because it has a 'struct device' embedded. Thus, having a release callback for it is correct. Allocating the vm_dev struct with devres totally breaks this protection, though. Instead of waiting for the vm_dev release callback, the memory is freed when the platform_device is removed. Resulting in a use-after-free when finally the callback is to be called. To easily see the problem, compile the kernel with CONFIG_DEBUG_KOBJECT_RELEASE and unbind with sysfs. The fix is easy, don't use devres in this case. Found during my research about object lifetime problems.
CVE-2023-53516	In the Linux kernel, the following vulnerability has been resolved: macvlan: add forgotten nla_policy for IFLA_MACVLAN_BC_CUTOFF The previous commit 954d1fa1ac93 ("macvlan: Add netlink attribute for broadcast cutoff") added one additional attribute named IFLA_MACVLAN_BC_CUTOFF to allow broadcast cutoff. However, it forgot to describe the nla_policy at macvlan_policy (drivers/net/macvlan.c). Hence, this suppose NLA_S32 (4 bytes) integer can be faked as empty (0 bytes) by a malicious user, which could leads to OOB in heap just like CVE-2023-3773. To fix it, this commit just completes the nla_policy description for IFLA_MACVLAN_BC_CUTOFF. This enforces the length check and avoids the potential OOB read.
CVE-2023-53517	In the Linux kernel, the following vulnerability has been resolved: tipc: do not update mtu if msg_max is too small in mtu negotiation When doing link mtu negotiation, a malicious peer may send Activate msg with a very small mtu, e.g. 4 in Shuang's testing, without checking for the minimum mtu, l->mtu will be set to 4 in tipc_link_proto_rcv(), then n->links[bearer_id].mtu is set to 4294967228, which is a overflow of '4 - INT_H_SIZE - EMSG_OVERHEAD' in tipc_link_mss(). With tipc_link.mtu = 4, tipc_link_xmit() kept printing the warning: tipc: Too large msg, purging xmit list 1 5 0 40 4! tipc: Too large msg, purging xmit list 1 15 0 60 4! And with tipc_link_entry.mtu 4294967228, a huge skb was allocated in named_distribute(), and when purging it in tipc_link_xmit(), a crash was even caused: general protection fault, probably for non-canonical address 0x2100001011000dd: 0000 [#1] PREEMPT SMP PTI CPU: 0 PID: 0 Comm: swapper/0 Kdump: loaded Not tainted 6.3.0.neta #19 RIP: 0010:kfree_skb_list_reason+0x7e/0x1f0 Call Trace: <IRQ> skb_release_data+0xf9/0x1d0 kfree_skb_reason+0x40/0x100 tipc_link_xmit+0x57a/0x740 [tipc] tipc_node_xmit+0x16c/0x5c0 [tipc] tipc_named_node_up+0x27f/0x2c0 [tipc] tipc_node_write_unlock+0x149/0x170 [tipc] tipc_rcv+0x608/0x740 [tipc] tipc_udp_rcv+0xdc/0x1f0 [tipc] udp_queue_rcv_one_skb+0x33e/0x620 udp_unicast_rcv_skb.isra.72+0x75/0x90 __udp4_lib_rcv+0x56d/0xc20 ip_protocol_deliver_rcu+0x100/0x2d0 This patch fixes it by checking the new mtu against tipc_bearer_min_mtu(), and not updating mtu if it is too small.
CVE-2023-53520	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix hci_suspend_sync crash If hci_unregister_dev() frees the hci_dev object but hci_suspend_notifier may still be accessing it, it can cause the program to crash. Here's the call trace: <4>[102152.653246] Call Trace: <4>[102152.653254] hci_suspend_sync+0x109/0x301 [bluetooth] <4>[102152.653259] hci_suspend_dev+0x78/0xcd [bluetooth] <4>[102152.653263] hci_suspend_notifier+0x42/0x7a [bluetooth] <4>[102152.653268] notifier_call_chain+0x43/0x6b <4>[102152.653271] __blocking_notifier_call_chain+0x48/0x69 <4>[102152.653273] __pm_notifier_call_chain+0x22/0x39 <4>[102152.653276] pm_suspend+0x287/0x57c <4>[102152.653278] state_store+0xae/0xe5 <4>[102152.653281] kernfs_fop_write+0x109/0x173 <4>[102152.653284] __vfs_write+0x16f/0x1a2 <4>[102152.653287] ? selinux_file_permission+0xca/0x16f <4>[102152.653289] ? security_file_permission+0x36/0x109 <4>[102152.653291] vfs_write+0x114/0x21d <4>[102152.653293] __x64_sys_write+0x7b/0xdb <4>[102152.653296] do_syscall_64+0x59/0x194 <4>[102152.653299] entry_SYSCALL_64_after_hwframe+0x5c/0xc1 This patch holds the reference count of the hci_dev object while processing it in hci_suspend_notifier to avoid potential crash caused by the race condition.
CVE-2023-53521	In the Linux kernel, the following vulnerability has been resolved: scsi: ses: Fix slab-out-of-bounds in ses_intf_remove() A fix for: BUG: KASAN: slab-out-of-bounds in ses_intf_remove+0x23f/0x270 [ses] Read of size 8 at addr ffff88a10d32e5d8 by task rmmmod/12013 When edev->components is zero, accessing edev->component[0] members is wrong.
CVE-2023-53524	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: pcie: Fix integer overflow in iwl_write_to_user_buf An integer overflow occurs in the iwl_write_to_user_buf() function, which is called by the iwl_dbgfs_monitor_data_read() function. static bool iwl_write_to_user_buf(char __user *user_buf, ssize_t count, void *buf, ssize_t *size, ssize_t *bytes_copied) { int buf_size_left = count - *bytes_copied; buf_size_left = buf_size_left - (buf_size_left % sizeof(u32)); if (*size > buf_size_left) *size = buf_size_left; If the user passes a SIZE_MAX value to the "ssize_t count" parameter, the ssize_t count parameter is assigned to "int buf_size_left". Then compare "*size" with "buf_size_left". Here, "buf_size_left" is a negative number, so "*size" is assigned "buf_size_left" and goes into the third argument of the copy_to_user function, causing a heap overflow. This is not a security vulnerability because iwl_dbgfs_monitor_data_read() is a debugfs operation with 0400 privileges.

CVE ID	Description
CVE-2023-53525	In the Linux kernel, the following vulnerability has been resolved: RDMA/cma: Allow UD qp_type to join multicast only As for multicast: - The SIDR is the only mode that makes sense; - Besides PS_UDP, other port spaces like PS_IB is also allowed, as it is UD compatible. In this case qkey also needs to be set [1]. This patch allows only UD qp_type to join multicast, and set qkey to default if it's not set, to fix an uninit-value error: the ib->rec.qkey field is accessed without being initialized. ===== BUG: KMSAN: uninit-value in cma_set_qkey drivers/infiniband/core/cma.c:510 [inline] BUG: KMSAN: uninit-value in cma_make_mc_event+0xb73/0xe00 drivers/infiniband/core/cma.c:4570 cma_set_qkey drivers/infiniband/core/cma.c:510 [inline] cma_make_mc_event+0xb73/0xe00 drivers/infiniband/core/cma.c:4570 cma_iboe_join_multicast drivers/infiniband/core/cma.c:4782 [inline] rdma_join_multicast+0x2b83/0x30a0 drivers/infiniband/core/cma.c:4814 ucma_process_join+0xa76/0xf60 drivers/infiniband/core/ucma.c:1479 ucma_join_multicast+0x1e3/0x250 drivers/infiniband/core/ucma.c:1546 ucma_write+0x639/0x6d0 drivers/infiniband/core/ucma.c:1732 vfs_write+0x8ce/0x2030 fs/read_write.c:588 ksys_write+0x28c/0x520 fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __ia32_sys_write+0xdb/0x120 fs/read_write.c:652 do_syscall_32_irqs_on arch/x86/entry/common.c:114 [inline] __do_fast_syscall_32+0x96/0xf0 arch/x86/entry/common.c:180 do_fast_syscall_32+0x34/0x70 arch/x86/entry/common.c:205 do_SYSENTER_32+0x1b/0x20 arch/x86/entry/common.c:248 entry_SYSENTER_compat_after_hwframe+0x4d/0x5c Local variable ib.i created at: cma_iboe_join_multicast drivers/infiniband/core/cma.c:4737 [inline] rdma_join_multicast+0x586/0x30a0 drivers/infiniband/core/cma.c:4814 ucma_process_join+0xa76/0xf60 drivers/infiniband/core/ucma.c:1479 CPU: 0 PID: 29874 Comm: syz-executor.3 Not tainted 5.16.0-rc3-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 ===== [1] https://lore.kernel.org/linux-rdma/20220117183832.GD84788@nvidia.com/
CVE-2023-53529	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: Fix memory leak in rtw88_usb Kmemleak shows the following leak arising from routine in the usb probe routine: unreferenced object 0xffff895cb29bba00 (size 512): comm "(udev-worker)", pid 534, jiffies 4294903932 (age 102751.088s) hex dump (first 32 bytes): 77 30 30 30 00 00 00 02 2f 2d 2b 30 00 00 00 w000...../-+0... 02 00 2a 28 00 00 00 00 ff 55 ff ff ff 00 00 00 ..*(.....U..... backtrace: [<ffffffff9265fa36>] kcalloc_trace+0x26/0x90 [<ffffffffc17eec41>] rtw_usb_probe+0x2f1/0x680 [rtw_usb] [<ffffffffc03e19fd>] usb_probe_interface+0xdd/0x2e0 [usbcore] [<ffffffff92b4f2fe>] really_probe+0x18e/0x3d0 [<ffffffff92b4f5b8>] __driver_probe_device+0x78/0x160 [<ffffffff92b4f6bf>] driver_probe_device+0x1f/0x90 [<ffffffff92b4f8df>] __driver_attach+0xbf/0x1b0 [<ffffffff92b4d350>] bus_for_each_dev+0x70/0xc0 [<ffffffff92b4e51e>] bus_add_driver+0x10e/0x210 [<ffffffff92b50935>] driver_register+0x55/0xf0 [<ffffffffc03e0708>] usb_register_driver+0x88/0x140 [usbcore] [<ffffffff92401153>] do_one_initcall+0x43/0x210 [<ffffffff9254f42a>] do_init_module+0x4a/0x200 [<ffffffff92551d1c>] __do_sys_finit_module+0xac/0x120 [<ffffffff92ee6626>] do_syscall_64+0x56/0x80 [<ffffffff9300006a>] entry_SYSCALL_64_after_hwframe+0x46/0xb0 The leak was verified to be real by unloading the driver, which resulted in a dangling pointer to the allocation. The allocated memory is freed in rtw_usb_intf_deinit().</ffffffff9300006a></ffffffff92ee6626></ffffffff92551d1c></ffffffff9254f42a></ffffffff92401153></ffffffffc03e0708></ffffffff92b50935></ffffffff92b4e51e></ffffffff92b4d350></ffffffff92b4f8df></ffffffff92b4f6bf></ffffffff92b4f5b8></ffffffff92b4f2fe></ffffffffc03e19fd></ffffffffc17eec41></ffffffff9265fa36>
CVE-2023-53530	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Use raw_smp_processor_id() instead of smp_processor_id() The following call trace was observed: localhost kernel: nvme nvme0: NVME-FC{0}: controller connect complete localhost kernel: BUG: using smp_processor_id() in preemptible [00000000] code: kworker/u129:4/75092 localhost kernel: nvme nvme0: NVME-FC{0}: new ctrl: NQN "nqn.1992-08.com.netapp:sn.b42d198afb4d11ecad6d00a098d6abfa:subsystem.PR_Channel2022_RH84_subsystem_291" localhost kernel: caller is qla_nvme_post_cmd+0x216/0x1380 [qla2xxx] localhost kernel: CPU: 6 PID: 75092 Comm: kworker/u129:4 Kdump: loaded Tainted: G B W OE ----- --- 5.14.0-70.22.1.el9_0.x86_64+debug #1 localhost kernel: Hardware name: HPE ProLiant XL420 Gen10/ProLiant XL420 Gen10, BIOS U39 01/13/2022 localhost kernel: Workqueue: nvme-wq nvme_async_event_work [nvme_core] localhost kernel: Call Trace: localhost kernel: dump_stack_lvl+0x57/0x7d localhost kernel: check_preemption_disabled+0xc8/0xd0 localhost kernel: qla_nvme_post_cmd+0x216/0x1380 [qla2xxx] Use raw_smp_processor_id() instead of smp_processor_id(). Also use queue_work() across the driver instead of queue_work_on() thus avoiding usage of smp_processor_id() when CONFIG_DEBUG_PREEMPT is enabled.

CVE ID	Description
CVE-2023-53532	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix deinitialization of firmware resources Currently, in ath11k_ahb_fw_resources_init(), iommu domain mapping is done only for the chipsets having fixed firmware memory. Also, for such chipsets, mapping is done only if it does not have TrustZone support. During deinitialization, only if TrustZone support is not there, iommu is unmapped back. However, for non fixed firmware memory chipsets, TrustZone support is not there and this makes the condition check to true and it tries to unmap the memory which was not mapped during initialization. This leads to the following trace - [83.198790] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 [83.259537] Modules linked in: ath11k_ahb ath11k_qmi_helpers .. snip .. [83.280286] pstate: 20000005 (nCv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) [83.287228] pc : __iommu_unmap+0x30/0x140 [83.293907] lr : iommu_unmap+0x5c/0xa4 [83.298072] sp : ffff80000b3abad0 .. snip .. [83.369175] Call trace: [83.376282] __iommu_unmap+0x30/0x140 [83.378541] iommu_unmap+0x5c/0xa4 [83.382360] ath11k_ahb_fw_resource_deinit.part.12+0x2c/0xac [ath11k_ahb] [83.385666] ath11k_ahb_free_resources+0x140/0x17c [ath11k_ahb] [83.392521] ath11k_ahb_shutdown+0x34/0x40 [ath11k_ahb] [83.398248] platform_shutdown+0x20/0x2c [83.403455] device_shutdown+0x16c/0x1c4 [83.407621] kernel_restart_prepare+0x34/0x3c [83.411529] kernel_restart+0x14/0x74 [83.415781] __do_sys_reboot+0x1c4/0x22c [83.419427] __arm64_sys_reboot+0x1c/0x24 [83.423420] invoke_syscall+0x44/0xfc [83.427326] el0_svc_common.constprop.3+0xac/0xe8 [83.430974] do_el0_svc+0xa0/0xa8 [83.435659] el0_svc+0x1c/0x44 [83.438957] el0t_64_sync_handler+0x60/0x144 [83.441910] el0t_64_sync+0x15c/0x160 [83.446343] Code: aa0103f4 f9400001 f90027a1 d2800001 (f94006a0) [83.449903] ---[end trace 0000000000000000]--- This can be reproduced by probing an AHB chipset which is not having a fixed memory region. During reboot (or rmmmod) trace can be seen. Fix this issue by adding a condition check on firmware fixed memory hw_param as done in the counter initialization function. Tested-on: IPQ8074 hw2.0 AHB WLAN.HK.2.7.0.1-01744-QCAHKSUPL_SILICONZ-1
CVE-2023-53539	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix incomplete state save in rxe_requester If a send packet is dropped by the IP layer in rxe_requester() the call to rxe_xmit_packet() can fail with err == -EAGAIN. To recover, the state of the wqe is restored to the state before the packet was sent so it can be resent. However, the routines that save and restore the state miss a significant part of the variable state in the wqe, the dma struct which is used to process through the sge table. And, the state is not saved before the packet is built which modifies the dma struct. Under heavy stress testing with many QPs on a fast node sending large messages to a slow node dropped packets are observed and the resent packets are corrupted because the dma struct was not restored. This patch fixes this behavior and allows the test cases to succeed.
CVE-2023-53540	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: reject auth/assoc to AP with our address If the AP uses our own address as its MLD address or BSSID, then clearly something's wrong. Reject such connections so we don't try and fail later.
CVE-2023-53543	In the Linux kernel, the following vulnerability has been resolved: vdpa: Add max vqp attr to vdpa_nl_policy for nlattr length check The vdpa_nl_policy structure is used to validate the nlattr when parsing the incoming nlmsg. It will ensure the attribute being described produces a valid nlattr pointer in info->attrs before entering into each handler in vdpa_nl_ops. That is to say, the missing part in vdpa_nl_policy may lead to illegal nlattr after parsing, which could lead to OOB read just like CVE-2023-3773. This patch adds the missing nla_policy for vdpa max vqp attr to avoid such bugs.
CVE-2023-53546	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: DR, fix memory leak in mlx5dr_cmd_create_reformat_ctx when mlx5_cmd_exec failed in mlx5dr_cmd_create_reformat_ctx, the memory pointed by 'in' is not released, which will cause memory leak. Move memory release after mlx5_cmd_exec.

CVE ID	Description
CVE-2023-53548	In the Linux kernel, the following vulnerability has been resolved: net: usbnet: Fix WARNING in usbnet_start_xmit/usb_submit_urb The syzbot fuzzer identified a problem in the usbnet driver: usb 1-1: BOGUS urb xfer, pipe 3 != type 1 WARNING: CPU: 0 PID: 754 at drivers/usb/core/urb.c:504 usb_submit_urb+0xed6/0x1880 drivers/usb/core/urb.c:504 Modules linked in: CPU: 0 PID: 754 Comm: kworker/0:2 Not tainted 6.4.0-rc7-syzkaller-00014-g692b7dc87ca6 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/27/2023 Workqueue: mld mld_ifc_work RIP: 0010:usb_submit_urb+0xed6/0x1880 drivers/usb/core/urb.c:504 Code: 7c 24 18 e8 2c b4 5b fb 48 8b 7c 24 18 e8 42 07 f0 fe 41 89 d8 44 89 e1 4c 89 ea 48 89 c6 48 c7 c7 a0 c9 fc 8a e8 5a 6f 23 fb <0f> 0b e9 58 f8 ff ff e8 fe b3 5b fb 48 81 c5 c0 05 00 00 e9 84 f7 RSP: 0018:ffff9000463f568 EFLAGS: 00010086 RAX: 0000000000000000 RBX: 0000000000000001 RCX: 0000000000000000 RDX: ffff88801eb28000 RSI: ffffffff814c03b7 RDI: 0000000000000001 RBP: ffff8881443b7190 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000001 R12: 0000000000000003 R13: ffff88802a77cb18 R14: 0000000000000003 R15: ffff888018262500 FS: 0000000000000000(0000) GS: ffff8880b9800000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 0000000080050033 CR2: 0000556a99c15a18 CR3: 0000000028c71000 CR4: 0000000000350ef0 Call Trace: <TASK> usbnet_start_xmit+0xfe5/0x2190 drivers/net/usb/usbnet.c:1453 __netdev_start_xmit include/linux/netdevice.h:4918 [inline] netdev_start_xmit include/linux/netdevice.h:4932 [inline] xmit_one net/core/dev.c:3578 [inline] dev_hard_start_xmit+0x187/0x700 net/core/dev.c:3594 ... This bug is caused by the fact that usbnet trusts the bulk endpoint addresses its probe routine receives in the driver_info structure, and it does not check to see that these endpoints actually exist and have the expected type and directions. The fix is simply to add such a check.
CVE-2023-53549	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: Rework long task execution when adding/deleting entries When adding/deleting large number of elements in one step in ipset, it can take a reasonable amount of time and can result in soft lockup errors. The patch 5f7b51bf09ba ("netfilter: ipset: Limit the maximal range of consecutive elements to add/delete") tried to fix it by limiting the max elements to process at all. However it was not enough, it is still possible that we get hung tasks. Lowering the limit is not reasonable, so the approach in this patch is as follows: rely on the method used at resizing sets and save the state when we reach a smaller internal batch limit, unlock/lock and proceed from the saved state. Thus we can avoid long continuous tasks and at the same time removed the limit to add/delete large number of elements in one step. The nfnl mutex is held during the whole operation which prevents one to issue other ipset commands in parallel.
CVE-2023-53550	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: fix global sysfs attribute type In commit 3666062b87ec ("cpufreq: amd-pstate: move to use bus_get_dev_root()") the "amd_pstate" attributes were moved from a dedicated kobject to the cpu root kobject. While the dedicated kobject expects to contain kobj_attributes the root kobject needs device_attributes. As the changed arguments are not used by the callbacks it works most of the time. However CFI will detect this issue: [4947.849350] CFI failure at dev_attr_show+0x24/0x60 (target: show_status+0x0/0x70; expected type: 0x8651b1de) ... [4947.849409] Call Trace: [4947.849410] <TASK> [4947.849411] ? __warn+0xcf/0x1c0 [4947.849414] ? dev_attr_show+0x24/0x60 [4947.849415] ? report_cfi_failure+0x4e/0x60 [4947.849417] ? handle_cfi_failure+0x14c/0x1d0 [4947.849419] ? __cfi_show_status+0x10/0x10 [4947.849420] ? handle_bug+0x4f/0x90 [4947.849421] ? exc_invalid_op+0x1a/0x60 [4947.849422] ? asm_exc_invalid_op+0x1a/0x20 [4947.849424] ? __cfi_show_status+0x10/0x10 [4947.849425] ? dev_attr_show+0x24/0x60 [4947.849426] sysfs_kf_seq_show+0xa6/0x110 [4947.849433] seq_read_iter+0x16c/0x4b0 [4947.849436] vfs_read+0x272/0x2d0 [4947.849438] ksys_read+0x72/0xe0 [4947.849439] do_syscall_64+0x76/0xb0 [4947.849440] ? do_user_addr_fault+0x252/0x650 [4947.849442] ? exc_page_fault+0x7a/0x1b0 [4947.849443] entry_SYSCALL_64_after_hwframe+0x72/0xdc
CVE-2023-53552	In the Linux kernel, the following vulnerability has been resolved: drm/i915: mark requests for GuC virtual engines to avoid use-after-free References to i915_requests may be trapped by userspace inside a sync_file or dmabuf (dma-resv) and held indefinitely across different processes. To counter-act the memory leaks, we try to not to keep references from the request past their completion. On the other side on fence release we need to know if rq->engine is valid and points to hw engine (true for non-virtual requests). To make it possible extra bit has been added to rq->execution_mask, for marking virtual engines. (cherry picked from commit 280410677af763f3871b93e794a199cfc6fb580)
CVE-2023-53553	In the Linux kernel, the following vulnerability has been resolved: HID: hyperv: avoid struct mempcy overrun warning A previous patch addressed the fortified mempcy warning for most builds, but I still see this one with gcc-9: In file included from include/linux/string.h:254, from drivers/hid/hid-hyperv.c:8: In function 'fortify_mempcy_chk', inlined from 'mousevsc_on_receive' at drivers/hid/hid-hyperv.c:272:3: include/linux/fortify-string.h:583:4: error: call to '__write_overflow_field' declared with attribute warning: detected write beyond size of field (1st parameter); maybe use struct_group()? [-Werror=attribute-warning] 583 __write_overflow_field(p_size_field, size); ~~~~~ My guess is that the WARN_ON() itself is what confuses gcc, so it no longer sees that there is a correct range check. Rework the code in a way that helps readability and avoids the warning.

CVE ID	Description
CVE-2023-53559	In the Linux kernel, the following vulnerability has been resolved: ip_vti: fix potential slab-use-after-free in decode_session6 When ip_vti device is set to the qdisc of the sfb type, the cb field of the sent skb may be modified during enqueueing. Then, slab-use-after-free may occur when ip_vti device sends IPv6 packets. As commit f855691975bb ("xfrm6: Fix the nexthdr offset in _decode_session6.") showed, xfrm_decode_session was originally intended only for the receive path. IP6CB(skb)->nhoff is not set during transmission. Therefore, set the cb field in the skb to 0 before sending packets.
CVE-2023-53560	In the Linux kernel, the following vulnerability has been resolved: tracing/histograms: Add histograms to hist_vars if they have referenced variables Hist triggers can have referenced variables without having direct variables fields. This can be the case if referenced variables are added for trigger actions. In this case the newly added references will not have field variables. Not taking such referenced variables into consideration can result in a bug where it would be possible to remove hist trigger with variables being referenced. This will result in a bug that is easily reproducible like so \$ cd /sys/kernel/tracing \$ echo 'synthetic_sys_enter char[] comm; long id' >> synthetic_events \$ echo 'hist:keys=common_pid.execname,id.syscall:vals=hitcount:comm=common_pid.execname' >> events/raw_syscalls/sys_enter/trigger \$ echo 'hist:keys=common_pid.execname,id.syscall:onmatch(raw_syscalls.sys_enter).synthetic_sys_enter(\$comm,id)' >> events/raw_syscalls/sys_enter/trigger \$ echo '! hist:keys=common_pid.execname,id.syscall:vals=hitcount:comm=common_pid.execname' >> events/raw_syscalls/sys_enter/trigger [100.263533] ===== [100.264634] BUG: KASAN: slab-use-after-free in resolve_var_refs+0xc7/0x180 [100.265520] Read of size 8 at addr ffff88810375d0f0 by task bash/439 [100.266320] [100.266533] CPU: 2 PID: 439 Comm: bash Not tainted 6.5.0-rc1 #4 [100.267277] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.0-20220807_005459-localhost 04/01/2014 [100.268561] Call Trace: [100.268902] <TASK> [100.269189] dump_stack_lvl+0x4c/0x70 [100.269680] print_report+0xc5/0x600 [100.270165] ? resolve_var_refs+0xc7/0x180 [100.270697] ? kasan_complete_mode_report_info+0x80/0x1f0 [100.271389] ? resolve_var_refs+0xc7/0x180 [100.271913] kasan_report+0xbd/0x100 [100.272380] ? resolve_var_refs+0xc7/0x180 [100.272920] __asan_load8+0x71/0xa0 [100.273377] resolve_var_refs+0xc7/0x180 [100.273888] event_hist_trigger+0x749/0x860 [100.274505] ? kasan_save_stack+0x2a/0x50 [100.275024] ? kasan_set_track+0x29/0x40 [100.275536] ? __pfx_event_hist_trigger+0x10/0x10 [100.276138] ? ksys_write+0xd1/0x170 [100.276607] ? do_syscall_64+0x3c/0x90 [100.277099] ? entry_SYSCALL_64_after_hwframe+0x6e/0xd8 [100.277771] ? destroy_hist_data+0x446/0x470 [100.278324] ? event_hist_trigger_parse+0xa6c/0x3860 [100.278962] ? __pfx_event_hist_trigger_parse+0x10/0x10 [100.279627] ? __kasan_check_write+0x18/0x20 [100.280177] ? mutex_unlock+0x85/0xd0 [100.280660] ? __pfx_mutex_unlock+0x10/0x10 [100.281200] ? kfree+0x7b/0x120 [100.281619] ? __kasan_slab_free+0x15d/0x1d0 [100.282197] ? event_trigger_write+0xac/0x100 [100.282764] ? __kasan_slab_free+0x16/0x20 [100.283293] ? __kmem_cache_free+0x153/0x2f0 [100.283844] ? sched_mm_cid_remote_clear+0xb1/0x250 [100.284550] ? __pfx_sched_mm_cid_remote_clear+0x10/0x10 [100.285221] ? event_trigger_write+0xbc/0x100 [100.285781] ? __kasan_check_read+0x15/0x20 [100.286321] ? __bitmap_weight+0x66/0xa0 [100.286833] ? _find_next_bit+0x46/0xe0 [100.287334] ? task_mm_cid_work+0x37f/0x450 [100.287872] event_triggers_call+0x84/0x150 [100.288408] trace_event_buffer_commit+0x339/0x430 [100.289073] ? ring_buffer_event_data+0x3f/0x60 [100.292189] trace_event_raw_event_sys_enter+0x8b/0xe0 [100.295434] syscall_trace_enter.constprop.0+0x18f/0x1b0 [100.298653] syscall_enter_from_user_mode+0x32/0x40 [100.301808] do_syscall_64+0x1a/0x90 [100.304748] entry_SYSCALL_64_after_hwframe+0x6e/0xd8 [100.307775] RIP: 0033:0x7f686c75c1cb [100.310617] Code: 73 01 c3 48 8b 0d 65 3c 10 00 f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 90 f3 0f 1e fa b8 21 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 35 3c 10 00 f7 d8 64 89 01 48 [100.317847] RSP: 002b:00007ffc60137a38 EFLAGS: 00000246 ORIG_RAX: 0000000000000021 [100.321200] RA --- truncated---
CVE-2023-53567	In the Linux kernel, the following vulnerability has been resolved: spi: qup: Don't skip cleanup in remove's error path Returning early in a platform driver's remove callback is wrong. In this case the dma resources are not released in the error path. this is never retried later and so this is a permanent leak. To fix this, only skip hardware disabling if waking the device fails.
CVE-2023-53570	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: fix integer overflow in nl80211_parse_mbssid_elems() nl80211_parse_mbssid_elems() uses a u8 variable num_elems to count the number of MBSSID elements in the nested netlink attribute attrs, which can lead to an integer overflow if a user of the nl80211 interface specifies 256 or more elements in the corresponding attribute in userspace. The integer overflow can lead to a heap buffer overflow as num_elems determines the size of the trailing array in elems, and this array is thereafter written to for each element in attrs. Note that this vulnerability only affects devices with the wiphy->mbssid_max_interfaces member set for the wireless physical device struct in the device driver, and can only be triggered by a process with CAP_NET_ADMIN capabilities. Fix this by checking for a maximum of 255 elements in attrs.

CVE ID	Description
CVE-2023-53574	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: delete timer and free skb queue when unloading Fix possible crash and memory leak on driver unload by deleting TX purge timer and freeing C2H queue in 'rtw_core_deinit()', shrink critical section in the latter by freeing COEX queue out of TX report lock scope.
CVE-2023-53575	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: fix potential array out of bounds access Account for IWL_SEC_WEP_KEY_OFFSET when needed while verifying key_len size in iwl_mvm_sec_key_add().
CVE-2023-53576	In the Linux kernel, the following vulnerability has been resolved: null_blk: Always check queue mode setting from configs Make sure to check device queue mode in the null_validate_conf() and return error for NULL_Q_RQ as we don't allow legacy I/O path, without this patch we get OOPs when queue mode is set to 1 from configs, following are repro steps :- modprobe null_blk nr_devices=0 mkdir config/nullb/nullb0 echo 1 > config/nullb/nullb0/memory_backed echo 4096 > config/nullb/nullb0/blocksize echo 20480 > config/nullb/nullb0/size echo 1 > config/nullb/nullb0/queue_mode echo 1 > config/nullb/nullb0/power Entering kdb (current=0xfffff88810acdd080, pid 2372) on processor 42 Oops: (null) due to oops @ 0xfffff88810c329 CPU: 42 PID: 2372 Comm: sh Tainted: G O N 6.3.0-rc5lblk+ #5 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014 RIP: 0010:null_add_dev.part.0+0xd9/0x720 [null_blk] Code: 01 00 00 85 d2 0f 85 a1 03 00 00 48 83 bb 08 01 00 00 00 0f 85 f7 03 00 00 80 bb 62 01 00 00 00 48 8b 75 20 0f 85 6d 02 00 00 <48> 89 6e 60 48 8b 75 20 bf 06 00 00 00 e8 f5 37 2c c1 48 8b 75 20 RSP: 0018:ffffc900052cbde0 EFLAGS: 00010246 RAX: 0000000000000001 RBX: ffff88811084d800 RCX: 0000000000000001 RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff888100042e00 RBP: ffff8881053d8200 R08: ffff8881053d8200 R09: ffff8881053d8200 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000002 R13: ffff888104765200 R14: ffff88810e0c1748 R15: ffff88810e0c1740 FS: 00007fd445fd1740(0000) GS:ffff8897dfc80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000000000033 CR2: 0000000000000060 CR3: 0000000166a00000 CR4: 0000000000350ee0 DR0: ffffffff8437a488 DR1: ffffffff8437a489 DR2: ffffffff8437a48a DR3: ffffffff8437a48b DR6: 00000000ffff0ff0 DR7: 0000000000000400 Call Trace: <TASK> nullb_device_power_store +0xd1/0x120 [null_blk] configs_write_iter+0xb4/0x120 vfs_write+0x2ba/0x3c0 ksys_write+0x5f/0xe0 do_syscall_64+0x3b/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc RIP: 0033:0x7fd4460c57a7 Code: 0d 00 f7 d8 64 89 02 48 c7 c0 ff ff ff eb b7 0f 1f 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 51 c3 48 83 ec 28 48 89 54 24 18 48 89 74 24 RSP: 002b:00007fd3792a4a8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 RAX: ffffffff8437a488 RBX: 0000000000000002 RCX: 00007fd4460c57a7 RDX: 0000000000000002 RSI: 000055b43c02e4c0 RDI: 0000000000000001 RBP: 000055b43c02e4c0 R08: 000000000000000a R09: 00007fd44615b4e0 R10: 00007fd44615b3e0 R11: 0000000000000246 R12: 0000000000000002 R13: 00007fd446198520 R14: 0000000000000002 R15: 00007fd446198700 </TASK>
CVE-2023-53577	In the Linux kernel, the following vulnerability has been resolved: bpf, cpumap: Make sure kthread is running before map update returns The following warning was reported when running stress-mode enabled xdp_redirect_cpu with some RT threads: -----[cut here]----- WARNING: CPU: 4 PID: 65 at kernel/bpf/cpumap.c:135 CPU: 4 PID: 65 Comm: kworker/4:1 Not tainted 6.5.0-rc2+ #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) Workqueue: events cpu_map_kthread_stop RIP: 0010:put_cpu_map_entry+0xda/0x220 Call Trace: <TASK> ? show_regs+0x65/0x70 ? __warn+0xa5/0x240 ? put_cpu_map_entry+0xda/0x220 cpu_map_kthread_stop+0x41/0x60 process_one_work+0x6b0/0xb80 worker_thread+0x96/0x720 kthread+0x1a5/0x1f0 ret_from_fork +0x3a/0x70 ret_from_fork_asm+0x1b/0x30 </TASK> The root cause is the same as commit 436901649731 ("bpf: cpumap: Fix memory leak in cpu_map_update_elem"). The kthread is stopped prematurely by kthread_stop() in cpu_map_kthread_stop(), and kthread() doesn't call cpu_map_kthread_run() at all but XDP program has already queued some frames or skbs into ptr_ring. So when __cpu_map_ring_cleanup() checks the ptr_ring, it will find it was not emptied and report a warning. An alternative fix is to use __cpu_map_ring_cleanup() to drop these pending frames or skbs when kthread_stop() returns -EINTR, but it may confuse the user, because these frames or skbs have been handled correctly by XDP program. So instead of dropping these frames or skbs, just make sure the per-cpu kthread is running before __cpu_map_entry_alloc() returns. After apply the fix, the error handle for kthread_stop() will be unnecessary because it will always return 0, so just remove it.

CVE ID	Description
CVE-2023-53578	In the Linux kernel, the following vulnerability has been resolved: net: qrtr: Fix an uninit variable access bug in qrtr_tx_resume() Syzbot reported a bug as following: <pre>===== BUG: KMSAN: uninit- value in qrtr_tx_resume+0x185/0x1f0 net/qrtr/af_qrtr.c:230 qrtr_tx_resume+0x185/0x1f0 net/ qrtr/af_qrtr.c:230 qrtr_endpoint_post+0xf85/0x11b0 net/qrtr/af_qrtr.c:519 qrtr_tun_write_iter +0x270/0x400 net/qrtr/tun.c:108 call_write_iter include/linux/fs.h:2189 [inline] aio_write+0x63a/0x950 fs/aio.c:1600 io_submit_one+0x1d1c/0x3bf0 fs/aio.c:2019 __do_sys_io_submit fs/aio.c:2078 [inline] __se_sys_io_submit+0x293/0x770 fs/aio.c:2048 __x64_sys_io_submit+0x92/0xd0 fs/ aio.c:2048 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/ x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd Uninit was created at: slab_post_alloc_hook mm/slab.h:766 [inline] slab_alloc_node mm/slub.c:3452 [inline] __kmem_cache_alloc_node+0x71f/0xce0 mm/slub.c:3491 __do_kmalloc_node mm/slab_common.c:967 [inline] __kmalloc_node_track_caller+0x114/0x3b0 mm/slab_common.c:988 kmalloc_reserve net/core/ skbuff.c:492 [inline] __alloc_skb+0x3af/0x8f0 net/core/skbuff.c:565 __netdev_alloc_skb+0x120/0x7d0 net/core/skbuff.c:630 qrtr_endpoint_post+0xbd/0x11b0 net/qrtr/af_qrtr.c:446 qrtr_tun_write_iter +0x270/0x400 net/qrtr/tun.c:108 call_write_iter include/linux/fs.h:2189 [inline] aio_write+0x63a/0x950 fs/aio.c:1600 io_submit_one+0x1d1c/0x3bf0 fs/aio.c:2019 __do_sys_io_submit fs/aio.c:2078 [inline] __se_sys_io_submit+0x293/0x770 fs/aio.c:2048 __x64_sys_io_submit+0x92/0xd0 fs/ aio.c:2048 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/ x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd It is because that skb- >len requires at least sizeof(struct qrtr_ctrl_pkt) in qrtr_tx_resume(). And skb->len equals to size in qrtr_endpoint_post(). But size is less than sizeof(struct qrtr_ctrl_pkt) when qrtr_cb->type equals to QRTR_TYPE_RESUME_TX in qrtr_endpoint_post() under the syzbot scenario. This triggers the uninit variable access bug. Add size check when qrtr_cb->type equals to QRTR_TYPE_RESUME_TX in qrtr_endpoint_post() to fix the bug.</pre>

CVE ID	Description
CVE-2023-53581	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Check for NOT_READY flag state after locking Currently the check for NOT_READY flag is performed before obtaining the necessary lock. This opens a possibility for race condition when the flow is concurrently removed from unready_flows list by the workqueue task, which causes a double-removal from the list and a crash[0]. Fix the issue by moving the flag check inside the section protected by uplink_priv->unready_flows_lock mutex. [0]: [44376.389654] general protection fault, probably for non-canonical address 0xdead00000000108: 0000 [#1] SMP [44376.391665] CPU: 7 PID: 59123 Comm: tc Not tainted 6.4.0-rc4+ #1 [44376.392984] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 [44376.395342] RIP: 0010:mlx5e_tc_del_fdb_flow+0xb3/0x340 [mlx5_core] [44376.396857] Code: 00 48 b8 b8 68 ce 02 00 e8 8a 4d 02 00 4c 8d a8 a8 01 00 00 4c 89 ef e8 8b 79 88 e1 48 8b 83 98 06 00 00 48 8b 93 90 06 00 00 <48> 89 42 08 48 89 10 48 b8 00 01 00 00 00 00 ad de 48 89 83 90 06 [44376.399167] RSP: 0018:ffff88812cc97570 EFLAGS: 00010246 [44376.399680] RAX: dead000000000122 RBX: ffff8881088e3800 RCX: ffff8881881bac00 [44376.400337] RDX: dead000000000100 RSI: ffff88812cc97500 RDI: ffff8881242f71b0 [44376.401001] RBP: ffff88811cbb0940 R08: 0000000000000400 R09: 0000000000000001 [44376.401663] R10: 0000000000000001 R11: 0000000000000000 R12: ffff88812c944000 [44376.402342] R13: ffff8881242f71a8 R14: ffff8881222b4000 R15: 0000000000000000 [44376.402999] FS: 00007f0451104800(0000) GS:ffff88852cb80000(0000) knlGS:0000000000000000 [44376.403787] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [44376.404343] CR2: 0000000000489108 CR3: 0000000123a79003 CR4: 0000000000370ea0 [44376.405004] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [44376.405665] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [44376.406339] Call Trace: [44376.406651] <TASK> [44376.406939] ? die_addr+0x33/0x90 [44376.407311] ? exc_general_protection+0x192/0x390 [44376.407795] ? asm_exc_general_protection+0x22/0x30 [44376.408292] ? mlx5e_tc_del_fdb_flow+0xb3/0x340 [mlx5_core] [44376.408876] __mlx5e_tc_del_fdb_peer_flow+0xbc/0xe0 [mlx5_core] [44376.409482] mlx5e_tc_del_flow+0x42/0x210 [mlx5_core] [44376.410055] mlx5e_flow_put+0x25/0x50 [mlx5_core] [44376.410529] mlx5e_delete_flow+0x24b/0x350 [mlx5_core] [44376.411043] tc_setup_cb_reoffload+0x22/0x80 [44376.411462] fl_reoffload+0x261/0x2f0 [cls_flow] [44376.411907] ? mlx5e_rep_indr_setup_ft_cb+0x160/0x160 [mlx5_core] [44376.412481] ? mlx5e_rep_indr_setup_ft_cb+0x160/0x160 [mlx5_core] [44376.413044] tcf_block_playback_offloads+0x76/0x170 [44376.413497] tcf_block_unbind+0x7b/0xd0 [44376.413881] tcf_block_setup+0x17d/0x1c0 [44376.414269] tcf_block_offload_cmd.isra.0+0xf1/0x130 [44376.414725] tcf_block_offload_unbind+0x43/0x70 [44376.415153] __tcf_block_put+0x82/0x150 [44376.415532] ingress_destroy+0x22/0x30 [sch_ingress] [44376.415986] qdisc_destroy+0x3b/0xd0 [44376.416343] qdisc_graft+0x4d0/0x620 [44376.416706] tc_get_qdisc+0x1c9/0x3b0 [44376.417074] rtnetlink_rcv_msg+0x29c/0x390 [44376.419978] ? rep_movs_alternative+0x3a/0xa0 [44376.420399] ? rtnl_calcit.isra.0+0x120/0x120 [44376.420813] netlink_rcv_skb+0x54/0x100 [44376.421192] netlink_unicast+0x1f6/0x2c0 [44376.421573] netlink_sendmsg+0x232/0x4a0 [44376.421980] sock_sendmsg+0x38/0x60 [44376.422328] ____sys_sendmsg+0x1d0/0x1e0 [44376.422709] ? copy_msghdr_from_user+0x6d/0xa0 [44376.423127] __sys_sendmsg+0x80/0xc0 [44376.423495] ? __sys_recvmsg+0x8b/0xc0 [44376.423869] __sys_sendmsg+0x51/0x90 [44376.424226] do_syscall_64+0x3d/0x90 [44376.424587] entry_SYSCALL_64_after_hwframe+0x46/0xb0 [44376.425046] RIP: 0033:0x7f045134f887 [44376.425403] Code: 0a 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff b9 0f 1f 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 2e 00 ---truncated---

CVE ID	Description
CVE-2023-53582	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: ensure CLM version is null-terminated to prevent stack-out-of-bounds Fix a stack-out-of-bounds read in brcmfmac that occurs when 'buf' that is not null-terminated is passed as an argument of streplace() in brcmf_c_preinit_dcmds(). This buffer is filled with a CLM version string by memcpy() in brcmf_fil_iovar_data_get(). Ensure buf is null-terminated. Found by a modified version of syzkaller. [33.004414][T1896] brcmfmac: brcmf_c_process_clm_blob: no clm_blob available (err=-2), device may have limited channels available [33.013486][T1896] brcmfmac: brcmf_c_preinit_dcmds: Firmware: BCM43236/3 wl0: Nov 30 2011 17:33:42 version 5.90.188.22 [33.021554][T1896] ===== [33.022379][T1896] BUG: KASAN: stack-out-of-bounds in streplace+0xf2/0x110 [33.023122][T1896] Read of size 1 at addr fffff90001d6efc8 by task kworker/0:2/1896 [33.023852][T1896] [33.024096][T1896] CPU: 0 PID: 1896 Comm: kworker/0:2 Tainted: G O 5.14.0+ #132 [33.024927][T1896] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org 04/01/2014 [33.026065][T1896] Workqueue: usb_hub_wq hub_event [33.026581][T1896] Call Trace: [33.026896][T1896] dump_stack_lvl+0x57/0x7d [33.027372][T1896] print_address_description.constprop.0.cold+0xf/0x334 [33.028037][T1896] ? streplace+0xf2/0x110 [33.028403][T1896] ? streplace+0xf2/0x110 [33.028807][T1896] kasan_report.cold+0x83/0xdf [33.029283][T1896] ? streplace+0xf2/0x110 [33.029666][T1896] streplace+0xf2/0x110 [33.029966][T1896] brcmf_c_preinit_dcmds+0xab1/0xc40 [33.030351][T1896] ? brcmf_c_set_joinpref_default+0x100/0x100 [33.030787][T1896] ? rcu_read_lock_sched_held+0xa1/0xd0 [33.031223][T1896] ? rcu_read_lock_bh_held+0xb0/0xb0 [33.031661][T1896] ? lock_acquire+0x19d/0x4e0 [33.032091][T1896] ? find_held_lock+0x2d/0x110 [33.032605][T1896] ? brcmf_usb_deq+0x1a7/0x260 [33.033087][T1896] ? brcmf_usb_rx_fill_all+0x5a/0xf0 [33.033582][T1896] brcmf_attach+0x246/0xd40 [33.034022][T1896] ? wiphy_new_nm+0x1476/0x1d50 [33.034383][T1896] ? kmemdup+0x30/0x40 [33.034722][T1896] brcmf_usb_probe+0x12de/0x1690 [33.035223][T1896] ? brcmf_usbdev_qinit.constprop.0+0x470/0x470 [33.035833][T1896] usb_probe_interface+0x25f/0x710 [33.036315][T1896] really_probe+0x1be/0xa90 [33.036656][T1896] __driver_probe_device+0x2ab/0x460 [33.037026][T1896] ? usb_match_id.part.0+0x88/0xc0 [33.037383][T1896] driver_probe_device+0x49/0x120 [33.037790][T1896] __device_attach_driver+0x18a/0x250 [33.038300][T1896] ? driver_allows_async_probing+0x120/0x120 [33.038986][T1896] bus_for_each_drv+0x123/0x1a0 [33.039906][T1896] ? bus_rescan_devices+0x20/0x20 [33.041412][T1896] ? lockdep_hardirqs_on_prepare+0x273/0x3e0 [33.041861][T1896] ? trace_hardirqs_on+0x1c/0x120 [33.042330][T1896] __device_attach+0x207/0x330 [33.042664][T1896] ? device_bind_driver+0xb0/0xb0 [33.043026][T1896] ? kobject_uevent_env+0x230/0x12c0 [33.043515][T1896] bus_probe_device+0x1a2/0x260 [33.043914][T1896] device_add+0xa61/0x1ce0 [33.044227][T1896] ? __mutex_unlock_slowpath+0xe7/0x660 [33.044891][T1896] ? __fw_devlink_link_to_suppliers+0x550/0x550 [33.045531][T1896] usb_set_configuration+0x984/0x1770 [33.046051][T1896] ? kernfs_create_link+0x175/0x230 [33.046548][T1896] usb_generic_driver_probe+0x69/0x90 [33.046931][T1896] usb_probe_device+0x9c/0x220 [33.047434][T1896] really_probe+0x1be/0xa90 [33.047760][T1896] __driver_probe_device+0x2ab/0x460 [33.048134][T1896] driver_probe_device+0x49/0x120 [33.048516][T1896] __device_attach_driver+0x18a/0x250 [33.048910][T1896] ? driver_allows_async_probing+0x120/0x120 ---truncated---
CVE-2023-53585	In the Linux kernel, the following vulnerability has been resolved: bpf: reject unhashed sockets in bpf_sk_assign The semantics for bpf_sk_assign are as follows: sk = some_lookup_func() bpf_sk_assign(skb, sk) bpf_sk_release(sk) That is, the sk is not consumed by bpf_sk_assign. The function therefore needs to make sure that sk lives long enough to be consumed from __inet_lookup_skb. The path through the stack for a TCPv4 packet is roughly: netif_receive_skb_core: takes RCU read lock __netif_receive_skb_core: sch_handle_ingress: tcf_classify: bpf_sk_assign() deliver_ptype_list_skb: deliver_skb: ip_packet_type->func == ip_rcv: ip_rcv_core: ip_rcv_finish_core: dst_input: ip_local_deliver: ip_local_deliver_finish: ip_protocol_deliver_rcu: tcp_v4_rcv: __inet_lookup_skb: skb_steal_sock The existing helper takes advantage of the fact that everything happens in the same RCU critical section: for sockets with SOCK_RCU_FREE set bpf_sk_assign never takes a reference. skb_steal_sock then checks SOCK_RCU_FREE again and does sock_put if necessary. This approach assumes that SOCK_RCU_FREE is never set on a sk between bpf_sk_assign and skb_steal_sock, but this invariant is violated by unhashed UDP sockets. A new UDP socket is created in TCP_CLOSE state but without SOCK_RCU_FREE set. That flag is only added in udp_lib_get_port() which happens when a socket is bound. When bpf_sk_assign was added it wasn't possible to access unhashed UDP sockets from BPF, so this wasn't a problem. This changed in commit 0c48eefae712 ("sock_map: Lift socket state restriction for datagram sockets"), but the helper wasn't adjusted accordingly. The following sequence of events will therefore lead to a refcount leak: 1. Add socket(AF_INET, SOCK_DGRAM) to a sockmap. 2. Pull socket out of sockmap and bpf_sk_assign it. Since SOCK_RCU_FREE is not set we increment the refcount. 3. bind() or connect() the socket, setting SOCK_RCU_FREE. 4. skb_steal_sock will now set refcounted = false due to SOCK_RCU_FREE. 5. tcp_v4_rcv() skips sock_put(). Fix the problem by rejecting unhashed sockets in bpf_sk_assign(). This matches the behaviour of __inet_lookup_skb which is ultimately the goal of bpf_sk_assign().

CVE ID	Description
CVE-2023-53586	In the Linux kernel, the following vulnerability has been resolved: scsi: target: Fix multiple LUN_RESET handling This fixes a bug where an initiator thinks a LUN_RESET has cleaned up running commands when it hasn't. The bug was added in commit 51ec502a3266 ("target: Delete tmr from list before processing"). The problem occurs when: 1. We have N I/O cmds running in the target layer spread over 2 sessions. 2. The initiator sends a LUN_RESET for each session. 3. session1's LUN_RESET loops over all the running commands from both sessions and moves them to its local drain_task_list. 4. session2's LUN_RESET does not see the LUN_RESET from session1 because the commit above has it remove itself. session2 also does not see any commands since the other reset moved them off the state lists. 5. sessions2's LUN_RESET will then complete with a successful response. 6. sessions2's initior believes the running commands on its session are now cleaned up due to the successful response and cleans up the running commands from its side. It then restarts them. 7. The commands do eventually complete on the backend and the target starts to return aborted task statuses for them. The initiator will either throw a invalid ITT error or might accidentally lookup a new task if the ITT has been reallocated already. Fix the bug by reverting the patch, and serialize the execution of LUN_RESETs and Preempt and Aborts. Also prevent us from waiting on LUN_RESETs in core_tmr_drain_tmr_list, because it turns out the original patch fixed a bug that was not mentioned. For LUN_RESET1 core_tmr_drain_tmr_list can see a second LUN_RESET and wait on it. Then the second reset will run core_tmr_drain_tmr_list and see the first reset and wait on it resulting in a deadlock.
CVE-2023-53587	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Sync IRQ works before buffer destruction If something was written to the buffer just before destruction, it may be possible (maybe not in a real system, but it did happen in ARCH=um with time-travel) to destroy the ringbuffer before the IRQ work ran, leading this KASAN report (or a crash without KASAN): BUG: KASAN: slab-use-after-free in irq_work_run_list+0x11a/0x13a Read of size 8 at addr 000000006d640a48 by task swapper/0 CPU: 0 PID: 0 Comm: swapper Tainted: G W O 6.3.0-rc1 #7 Stack: 60c4f20f 0c203d48 41b58ab3 60f224fc 600477fa 60f35687 60c4f20f 601273dd 00000008 6101eb00 6101eab0 615be548 Call Trace: [<60047a58>] show_stack+0x25e/0x282 [<60c609e0>] dump_stack_lvl+0x96/0xfd [<60c50d4c>] print_report+0x1a7/0x5a8 [<603078d3>] kasan_report+0xc1/0xe9 [<60308950>] __asan_report_load8_noabort+0x1b/0x1d [<60232844>] irq_work_run_list+0x11a/0x13a [<602328b4>] irq_work_tick+0x24/0x34 [<6017f9dc>] update_process_times+0x162/0x196 [<6019f335>] tick_sched_handle+0x1a4/0x1c3 [<6019fd9e>] tick_sched_timer+0x79/0x10c [<601812b9>] __hrtimer_run_queues.constprop.0+0x425/0x695 [<60182913>] hrtimer_interrupt+0x16c/0x2c4 [<600486a3>] um_timer+0x164/0x183 [...] Allocated by task 411: save_stack_trace+0x99/0xb5 stack_trace_save+0x81/0x9b kasan_save_stack+0x2d/0x54 kasan_set_track+0x34/0x3e kasan_save_alloc_info+0x25/0x28 ____kasan_kmalloc+0x8b/0x97 __kasan_kmalloc+0x10/0x12 ____kmalloc+0xb2/0xe8 load_elf_phdrs+0xee/0x182 [...] The buggy address belongs to the object at 000000006d640800 which belongs to the cache kmalloc-1k of size 1024 The buggy address is located 584 bytes inside of freed 1024-byte region [000000006d640800, 000000006d640c00) Add the appropriate irq_work_sync() so the work finishes before the buffers are destroyed. Prior to the commit in the Fixes tag below, there was only a single global IRQ work, so this issue didn't exist.
CVE-2023-53588	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: check for station first in client probe When probing a client, first check if we have it, and then check for the channel context, otherwise you can trigger the warning there easily by probing when the AP isn't even started yet. Since a client existing means the AP is also operating, we can then keep the warning. Also simplify the moved code a bit.
CVE-2023-53589	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: mvnm: don't trust firmware n_channels If the firmware sends us a corrupted MCC response with n_channels much larger than the command response can be, we might copy far too much (uninitialized) memory and even crash if the n_channels is large enough to make it run out of the one page allocated for the FW response. Fix that by checking the lengths. Doing a < comparison would be sufficient, but the firmware should be doing it correctly, so check more strictly.

CVE ID	Description
CVE-2023-53591	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix deadlock in tc route query code Cited commit causes ABBA deadlock[0] when peer flows are created while holding the devcom rw semaphore. Due to peer flows offload implementation the lock is taken much higher up the call chain and there is no obvious way to easily fix the deadlock. Instead, since tc route query code needs the peer eswitch structure only to perform a lookup in xarray and doesn't perform any sleeping operations with it, refactor the code for lockless execution in following ways: - RCUify the devcom 'data' pointer. When resetting the pointer synchronously wait for RCU grace period before returning. This is fine since devcom is currently only used for synchronization of pairing/unpairing of eswitches which is rare and already expensive as-is. - Wrap all usages of 'paired' boolean in {READ WRITE}_ONCE(). The flag has already been used in some unlocked contexts without proper annotations (e.g. users of mlx5_devcom_is_paired() function), but it wasn't an issue since all relevant code paths checked it again after obtaining the devcom semaphore. Now it is also used by mlx5_devcom_get_peer_data_rcu() as "best effort" check to return NULL when devcom is being unpaired. Note that while RCU read lock doesn't prevent the unpaired flag from being changed concurrently it still guarantees that reader can continue to use 'data'. - Refactor mlx5e_tc_query_route_vport() function to use new mlx5_devcom_get_peer_data_rcu() API which fixes the deadlock. [0]: [164.599612] ===== [164.600142] WARNING: possible circular locking dependency detected [164.600667] 6.3.0-rc3+ #1 Not tainted [164.601021] ----- [164.601557] handler1/3456 is trying to acquire lock: [164.601998] ffff88811f1714b0 (&esw->offloads.encap_tbl_lock){+.+.}-{3:3}, at: mlx5e_attach_encap+0xd8/0x8b0 [mlx5_core] [164.603078] but task is already holding lock: [164.603617] ffff88810137fc98 (&comp->sem){++++}-{3:3}, at: mlx5_devcom_get_peer_data+0x37/0x80 [mlx5_core] [164.604459] which lock already depends on the new lock. [164.605190] the existing dependency chain (in reverse order) is: [164.605848] -> #1 (&comp->sem){+++ +}-{3:3}: [164.606380] down_read+0x39/0x50 [164.606772] mlx5_devcom_get_peer_data+0x37/0x80 [mlx5_core] [164.607336] mlx5e_tc_query_route_vport+0x86/0xc0 [mlx5_core] [164.607914] mlx5e_tc_tun_route_lookup+0x1a4/0x1d0 [mlx5_core] [164.608495] mlx5e_attach_decap_route+0xc6/0x1e0 [mlx5_core] [164.609063] mlx5e_tc_add_fdb_flow+0x1ea/0x360 [mlx5_core] [164.609627] __mlx5e_add_fdb_flow+0x2d2/0x430 [mlx5_core] [164.610175] mlx5e_configure_flow+0x952/0x1a20 [mlx5_core] [164.610741] tc_setup_cb_add+0xd4/0x200 [164.611146] fl_hw_replace_filter+0x14c/0x1f0 [cls_flow] [164.611661] fl_change+0xc95/0x18a0 [cls_flow] [164.612116] tc_new_tfilter+0x3fc/0xd20 [164.612516] rtnetlink_rcv_msg+0x418/0x5b0 [164.612936] netlink_rcv_skb+0x54/0x100 [164.613339] netlink_unicast+0x190/0x250 [164.613746] netlink_sendmsg+0x245/0x4a0 [164.614150] sock_sendmsg+0x38/0x60 [164.614522] ____sys_sendmsg+0x1d0/0x1e0 [164.614934] __sys_sendmsg+0x80/0xc0 [164.615320] __sys_sendmsg+0x51/0x90 [164.615701] do_syscall_64+0x3d/0x90 [164.616083] entry_SYSCALL_64_after_hwframe+0x46/0xb0 [164.616568] -> #0 (&esw->offloads.encap_tbl_lock){+.+.}-{3:3}: [164.617210] __lock_acquire+0x159e/0x26e0 [164.617638] lock_acquire+0xc2/0x2a0 [164.618018] __mutex_lock+0x92/0xcd0 [164.618401] mlx5e_attach_encap+0xd8/0x8b0 [mlx5_core] [164.618943] post_process_attr+0x153/0x2d0 [---truncated---
CVE-2023-53593	In the Linux kernel, the following vulnerability has been resolved: cifs: Release folio lock on fscache read hit. Under the current code, when cifs_readpage_worker is called, the call contract is that the callee should unlock the page. This is documented in the read_folio section of Documentation/filesystems/vfs.rst as: > The filesystem should unlock the folio once the read has completed, > whether it was successful or not. Without this change, when fscache is in use and cache hit occurs during a read, the page lock is leaked, producing the following stack on subsequent reads (via mmap) to the page: \$ cat /proc/3890/task/12864/stack [<0>] folio_wait_bit_common+0x124/0x350 [<0>] filemap_read_folio+0xad/0xf0 [<0>] filemap_fault+0x8b1/0xab0 [<0>] __do_fault+0x39/0x150 [<0>] do_fault+0x25c/0x3e0 [<0>] __handle_mm_fault+0x6ca/0xc70 [<0>] handle_mm_fault+0xe9/0x350 [<0>] do_user_addr_fault+0x225/0x6c0 [<0>] exc_page_fault+0x84/0x1b0 [<0>] asm_exc_page_fault+0x27/0x30 This requires a reboot to resolve; it is a deadlock. Note however that the call to cifs_readpage_from_fscache does mark the page clean, but does not free the folio lock. This happens in __cifs_readpage_from_fscache on success. Releasing the lock at that point however is not appropriate as cifs_readahead also calls cifs_readpage_from_fscache and *does* unconditionally release the lock after its return. This change therefore effectively makes cifs_readpage_worker work like cifs_readahead.

CVE ID	Description
CVE-2023-53594	In the Linux kernel, the following vulnerability has been resolved: driver core: fix resource leak in device_add() When calling kobject_add() failed in device_add(), it will call cleanup_glue_dir() to free resource. But in kobject_add(), dev->kobj.parent has been set to NULL. This will cause resource leak. The process is as follows: device_add() get_device_parent() class_dir_create_and_add() kobject_add() //kobject_get() ... dev->kobj.parent = kobj; ... kobject_add() //failed, but set dev->kobj.parent = NULL ... glue_dir = get_glue_dir(dev) //glue_dir = NULL, and goto //"Error" label ... cleanup_glue_dir() //because glue_dir is NULL, not call //kobject_put() The preceding problem may cause insmod mac80211_hwsim.ko to failed. sysfs: cannot create duplicate filename '/devices/virtual/mac80211_hwsim' Call Trace: <TASK> dump_stack_lvl+0x8e/0xd1 sysfs_warn_dup.cold+0x1c/0x29 sysfs_create_dir_ns+0x224/0x280 kobject_add_internal+0x2aa/0x880 kobject_add+0x135/0x1a0 get_device_parent+0x3d7/0x590 device_add+0x2aa/0x1cb0 device_create_groups_vargs+0x1eb/0x260 device_create+0xdc/0x110 mac80211_hwsim_new_radio+0x31e/0x4790 [mac80211_hwsim] init_mac80211_hwsim+0x48d/0x1000 [mac80211_hwsim] do_one_initcall+0x10f/0x630 do_init_module+0x19f/0x5e0 load_module+0x64b7/0x6eb0 __do_sys_finit_module+0x140/0x200 do_syscall_64+0x35/0x80 entry_SYSCALL_64_after_hwframe+0x46/0xb0 </TASK> kobject_add_internal failed for mac80211_hwsim with -EEXIST, don't try to register things with the same name in the same directory.
CVE-2023-53596	In the Linux kernel, the following vulnerability has been resolved: drivers: base: Free devm resources when unregistering a device In the current code, devres_release_all() only gets called if the device has a bus and has been probed. This leads to issues when using bus-less or driver-less devices where the device might never get freed if a managed resource holds a reference to the device. This is happening in the DRM framework for example. We should thus call devres_release_all() in the device_del() function to make sure that the device-managed actions are properly executed when the device is unregistered, even if it has neither a bus nor a driver. This is effectively the same change than commit 2f8d16a996da ("devres: release resources on device_del()") that got reverted by commit a525a3ddeaca ("driver core: free devres in device_release") over memory leaks concerns. This patch effectively combines the two commits mentioned above to release the resources both on device_del() and device_release() and get the best of both worlds.
CVE-2023-53597	In the Linux kernel, the following vulnerability has been resolved: cifs: fix mid leak during reconnection after timeout threshold When the number of responses with status of STATUS_IO_TIMEOUT exceeds a specified threshold (NUM_STATUS_IO_TIMEOUT), we reconnect the connection. But we do not return the mid, or the credits returned for the mid, or reduce the number of in-flight requests. This bug could result in the server->in_flight count to go bad, and also cause a leak in the mids. This change moves the check to a few lines below where the response is decrypted, even of the response is read from the transform header. This way, the code for returning the mids can be reused. Also, the cifs_reconnect was reconnecting just the transport connection before. In case of multi-channel, this may not be what we want to do after several timeouts. Changed that to reconnect the session and the tree too. Also renamed NUM_STATUS_IO_TIMEOUT to a more appropriate name MAX_STATUS_IO_TIMEOUT.
CVE-2023-53598	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: host: Range check CHDBOFF and ERDBOFF If the value read from the CHDBOFF and ERDBOFF registers is outside the range of the MHI register space then an invalid address might be computed which later causes a kernel panic. Range check the read value to prevent a crash due to bad data from the device.
CVE-2023-53600	In the Linux kernel, the following vulnerability has been resolved: tunnels: fix kasan splat when generating ipv4 pmtu error If we try to emit an icmp error in response to a nonlinear skb, we get BUG: KASAN: slab-out-of-bounds in ip_compute_csum+0x134/0x220 Read of size 4 at addr ffff88811c50db00 by task iperf3/1691 CPU: 2 PID: 1691 Comm: iperf3 Not tainted 6.5.0-rc3+ #309 [...] kasan_report+0x105/0x140 ip_compute_csum+0x134/0x220 iptunnel_pmtud_build_icmp+0x554/0x1020 skb_tunnel_check_pmtu+0x513/0xb80 vxlan_xmit_one+0x139e/0x2ef0 vxlan_xmit+0x1867/0x2760 dev_hard_start_xmit+0x1ee/0x4f0 br_dev_queue_push_xmit+0x4d1/0x660 [...] ip_compute_csum() cannot deal with nonlinear skbs, so avoid it. After this change, splat is gone and iperf3 is no longer stuck.

CVE ID	Description
CVE-2023-53601	In the Linux kernel, the following vulnerability has been resolved: bonding: do not assume skb mac_header is set Drivers must not assume in their ndo_start_xmit() that skbs have their mac_header set. skb->data is all what is needed. bonding seems to be one of the last offender as caught by syzbot: WARNING: CPU: 1 PID: 12155 at include/linux/skbuff.h:2907 skb_mac_offset include/linux/skbuff.h:2913 [inline] WARNING: CPU: 1 PID: 12155 at include/linux/skbuff.h:2907 bond_xmit_hash drivers/net/bonding/bond_main.c:4170 [inline] WARNING: CPU: 1 PID: 12155 at include/linux/skbuff.h:2907 bond_xmit_3ad_xor_slave_get drivers/net/bonding/bond_main.c:5149 [inline] WARNING: CPU: 1 PID: 12155 at include/linux/skbuff.h:2907 bond_3ad_xor_xmit drivers/net/bonding/bond_main.c:5186 [inline] WARNING: CPU: 1 PID: 12155 at include/linux/skbuff.h:2907 __bond_start_xmit drivers/net/bonding/bond_main.c:5442 [inline] WARNING: CPU: 1 PID: 12155 at include/linux/skbuff.h:2907 bond_start_xmit+0x14ab/0x19d0 drivers/net/bonding/bond_main.c:5470 Modules linked in: CPU: 1 PID: 12155 Comm: syz-executor.3 Not tainted 6.1.30-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/25/2023 RIP: 0010:skb_mac_header include/linux/skbuff.h:2907 [inline] RIP: 0010:skb_mac_offset include/linux/skbuff.h:2913 [inline] RIP: 0010:bond_xmit_hash drivers/net/bonding/bond_main.c:4170 [inline] RIP: 0010:bond_xmit_3ad_xor_slave_get drivers/net/bonding/bond_main.c:5149 [inline] RIP: 0010:bond_3ad_xor_xmit drivers/net/bonding/bond_main.c:5186 [inline] RIP: 0010:__bond_start_xmit drivers/net/bonding/bond_main.c:5442 [inline] RIP: 0010:bond_start_xmit+0x14ab/0x19d0 drivers/net/bonding/bond_main.c:5470 Code: 8b 7c 24 30 e8 76 dd 1a 01 48 85 c0 74 0d 48 89 c3 e8 29 67 2e fe e9 15 ef ff ff e8 1f 67 2e fe e9 10 ef ff ff e8 15 67 2e fe <0f> 0b e9 45 f8 ff ff e8 09 67 2e fe e9 dc fa ff ff e8 ff 66 2e fe RSP: 0018:ffffc90002ffff6e0 EFLAGS: 00010283 RAX: ffffffff835874db RBX: 000000000000ffff RCX: 0000000000040000 RDX: fffffc90004dcf000 RSI: 00000000000000b5 RDI: 00000000000000b6 RBP: fffffc90002ffff8b8 R08: ffffffff83586d16 R09: ffffffff83586584 R10: 0000000000000007 R11: ffff8881599fc780 R12: ffff88811b6a7b7e R13: 1ffff110236d4f6f R14: ffff88811b6a7ac0 R15: 1ffff110236d4f76 FS: 00007f2e9eb47700(0000) GS:ffff8881f6b00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000001b2e421000 CR3: 000000010e6d4000 CR4: 0000000003526e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff00ff DR7: 0000000000000400 Call Trace: <TASK> [<ffffffffff8471a49f>] netdev_start_xmit include/linux/netdevice.h:4925 [inline] [<ffffffffff8471a49f>] __dev_direct_xmit+0x4ef/0x850 net/core/dev.c:4380 [<ffffffffff851d845b>] dev_direct_xmit include/linux/netdevice.h:3043 [inline] [<ffffffffff851d845b>] packet_direct_xmit+0x18b/0x300 net/packet/af_packet.c:284 [<ffffffffff851c7472>] packet_snd net/packet/af_packet.c:3112 [inline] [<ffffffffff851c7472>] packet_sendmsg+0x4a22/0x64d0 net/packet/af_packet.c:3143 [<ffffffffff8467a4b2>] sock_sendmsg_nosec net/socket.c:716 [inline] [<ffffffffff8467a4b2>] sock_sendmsg net/socket.c:736 [inline] [<ffffffffff8467a4b2>] __sys_sendto+0x472/0x5f0 net/socket.c:2139 [<ffffffffff8467a715>] __do_sys_sendto net/socket.c:2151 [inline] [<ffffffffff8467a715>] __se_sys_sendto net/socket.c:2147 [inline] [<ffffffffff8467a715>] __x64_sys_sendto+0xe5/0x100 net/socket.c:2147 [<ffffffffff8553071f>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<ffffffffff8553071f>] do_syscall_64+0x2f/0x50 arch/x86/entry/common.c:80 [<ffffffffff85600087>] entry_SYSCALL_64_after_hwframe+0x63/0xcd
CVE-2023-53602	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix memory leak in WMI firmware stats Memory allocated for firmware pdev, vdev and beacon statistics are not released during rmmmod. Fix it by calling ath11k_fw_stats_free() function before hardware unregister. While at it, avoid calling ath11k_fw_stats_free() while processing the firmware stats received in the WMI event because the local list is getting spliced and reinitialised and hence there are no elements in the list after splicing. Tested-on: QCN9074 hw1.0 PCI WLAN.HK.2.7.0.1-01744-QCAHKSUPL_SILICONZ-1
CVE-2023-53603	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Avoid fcport pointer dereference Klocwork reported warning of NULL pointer may be dereferenced. The routine exits when sa_ctl is NULL and fcport is allocated after the exit call thus causing NULL fcport pointer to dereference at the time of exit. To avoid fcport pointer dereference, exit the routine when sa_ctl is NULL.
CVE-2023-53604	In the Linux kernel, the following vulnerability has been resolved: dm integrity: call kmem_cache_destroy() in dm_integrity_init() error path Otherwise the journal_io_cache will leak if dm_register_target() fails.
CVE-2023-53605	In the Linux kernel, the following vulnerability has been resolved: drm: amd: display: Fix memory leakage This commit fixes memory leakage in dc_construct_ctx() function.

CVE ID	Description
CVE-2023-53606	In the Linux kernel, the following vulnerability has been resolved: nfsd: clean up potential nfsd_file refcount leaks in COPY codepath There are two different flavors of the nfsd4_copy struct. One is embedded in the compound and is used directly in synchronous copies. The other is dynamically allocated, refcounted and tracked in the client struture. For the embedded one, the cleanup just involves releasing any nfsd_files held on its behalf. For the async one, the cleanup is a bit more involved, and we need to dequeue it from lists, unhash it, etc. There is at least one potential refcount leak in this code now. If the kthread_create call fails, then both the src and dst nfsd_files in the original nfsd4_copy object are leaked. The cleanup in this codepath is also sort of weird. In the async copy case, we'll have up to four nfsd_file references (src and dst for both flavors of copy structure). They are both put at the end of nfsd4_do_async_copy, even though the ones held on behalf of the embedded one outlive that structure. Change it so that we always clean up the nfsd_file refs held by the embedded copy structure before nfsd4_copy returns. Rework cleanup_async_copy to handle both inter and intra copies. Eliminate nfsd4_cleanup_intra_ssc since it now becomes a no-op.
CVE-2023-53607	In the Linux kernel, the following vulnerability has been resolved: ALSA: ymfpci: Fix BUG_ON in probe function The snd_dma_buffer.bytes field now contains the aligned size, which this snd_BUG_ON() did not account for, resulting in the following: [9.625915] -----[cut here]----- [9.633440] WARNING: CPU: 0 PID: 126 at sound/pci/ymfpci/ymfpci_main.c:2168 snd_ymfpci_create+0x681/0x698 [snd_ymfpci] [9.648926] Modules linked in: snd_ymfpci(+) snd_intel_dspcfg kvm(+) snd_intel_sdw_acpi snd_ac97_codec snd_mpu401_uart snd_opl3_lib irqbypass snd_hda_codec gameport snd_rawmidi crct10dif_pclmul crc32_pclmul cfg80211 snd_hda_core polyval_cmln polyval_generic gf128mul snd_seq_device ghash_clmulni_intel snd_hwdep ac97_bus sha512_ssse3 rkill snd_pcm aesni_intel tg3 snd_timer crypto_simd snd_mxm_wmi libphy cryptd k10temp fam15h_power pcspkr soundcore sp5100_tco wmi acpi_cpufreq mac_hid dm_multipath sg loop fuse dm_mod bpf_preload ip_tables x_tables ext4 crc32c_generic crc16 mbcache jbd2 sr_mod cdrom ata_generic pata_acpi firewire_ohci crc32c_intel firewire_core xhci_pci crc_itu_t pata_via xhci_pci_renesas floppy [9.711849] CPU: 0 PID: 126 Comm: kworker/0:2 Not tainted 6.1.21-lts #1 08d2e5ece03136efa7c6aeea9a9c40916b1bd8da [9.722200] Hardware name: To Be Filled By O.E.M. To Be Filled By O.E.M./990FX Extreme4, BIOS P2.70 06/05/2014 [9.732204] Workqueue: events work_for_cpu_fn [9.736580] RIP: 0010:snd_ymfpci_create+0x681/0x698 [snd_ymfpci] [9.742594] Code: 8c c0 4c 89 e2 48 89 df 48 c7 c6 92 c6 8c c0 e8 15 d0 e9 ff 48 83 c4 08 44 89 e8 5b 5d 41 5c 41 5d 41 5e 41 5f e9 d3 7a 33 e3 <0f> 0b e9 cb fd ff ff 41 bd fb ff ff eb db 41 bd f4 ff ff eb [9.761358] RSP: 0018:ffffb64804e7da0 EFLAGS: 00010287 [9.766594] RAX: ffff8fa2df06c400 RBX: ffff8fa3073a8000 RCX: ffff8fa303fbc4a8 [9.773734] RDX: ffff8fa2df06d000 RSI: 0000000000000010 RDI: 0000000000000020 [9.780876] RBP: ffff8fa300b5d0d0 R08: ffff8fa3073a8e50 R09: 00000000df06bf00 [9.788018] R10: ffff8fa2df06bf00 R11: 00000000df068200 R12: ffff8fa3073a8918 [9.795159] R13: 0000000000000000 R14: 0000000000000080 R15: ffff8fa2df068200 [9.802317] FS: 0000000000000000(0000) GS:ffff8fa9fec00000(0000) knlGS:0000000000000000 [9.810414] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [9.816158] CR2: 000055febaf66500 CR3: 0000000101a2e000 CR4: 0000000000040f60 [9.823301] Call Trace: [9.825747] <TASK> [9.827889] snd_card_ymfpci_probe+0x194/0x950 [snd_ymfpci b78a5fe64b5663a6390a909c67808567e3e73615] [9.837030] ? finish_task_switch.isra.0+0x90/0x2d0 [9.841918] local_pci_probe+0x45/0x80 [9.845680] work_for_cpu_fn+0x1a/0x30 [9.849431] process_one_work+0x1c7/0x380 [9.853464] worker_thread+0x1af/0x390 [9.857225] ? rescuer_thread+0x3b0/0x3b0 [9.861254] kthread+0xde/0x110 [9.864414] ? kthread_complete_and_exit+0x20/0x20 [9.869210] ret_from_fork+0x22/0x30 [9.872792] </TASK> [9.874985] ---[end trace 0000000000000000]---
CVE-2023-53611	In the Linux kernel, the following vulnerability has been resolved: ipmi_si: fix a memleak in try_smi_init() Kmemleak reported the following leak info in try_smi_init(): unreferenced object 0xffff00018ecf9400 (size 1024): comm "modprobe", pid 2707763, jiffies 4300851415 (age 773.308s) backtrace: [<000000004ca5b312>] __kmalloc+0x4b8/0x7b0 [<00000000953b1072>] try_smi_init+0x148/0x5dc [ipmi_si] [<000000006460d325>] 0xffff800081b10148 [<0000000039206ea5>] do_one_initcall+0x64/0x2a4 [<00000000601399ce>] do_init_module+0x50/0x300 [<000000003c12ba3c>] load_module+0x7a8/0x9e0 [<00000000c246fffe>] __se_sys_init_module+0x104/0x180 [<00000000eea99093>] __arm64_sys_init_module+0x24/0x30 [<0000000021b1ef87>] el0_svc_common.constprop.0+0x94/0x250 [<0000000070f4f8b7>] do_el0_svc+0x48/0xe0 [<000000005a05337f>] el0_svc+0x24/0x3c [<000000005eb248d6>] el0_sync_handler+0x160/0x164 [<0000000030a59039>] el0_sync+0x160/0x180 The problem was that when an error occurred before handlers registration and after allocating 'new_smi->si_sm', the variable wouldn't be freed in the error handling afterwards since 'shutdown_smi()' hadn't been registered yet. Fix it by adding a 'kfree()' in the error handling path in 'try_smi_init()'.

CVE ID	Description
CVE-2023-53612	In the Linux kernel, the following vulnerability has been resolved: hwmon: (coretemp) Simplify platform device handling Coretemp's platform driver is unconventional. All the real work is done globally by the initcall and CPU hotplug notifiers, while the "driver" effectively just wraps an allocation and the registration of the hwmon interface in a long-winded round-trip through the driver core. The whole logic of dynamically creating and destroying platform devices to bring the interfaces up and down is error prone, since it assumes platform_device_add() will synchronously bind the driver and set drvdata before it returns, thus results in a NULL dereference if drivers_autoprobe is turned off for the platform bus. Furthermore, the unusual approach of doing that from within a CPU hotplug notifier, already commented in the code that it deadlocks suspend, also causes lockdep issues for other drivers or subsystems which may want to legitimately register a CPU hotplug notifier from a platform bus notifier. All of these issues can be solved by ripping this unusual behaviour out completely, simply tying the platform devices to the lifetime of the module itself, and directly managing the hwmon interfaces from the hotplug notifiers. There is a slight user-visible change in that /sys/bus/platform/drivers/coretemp will no longer appear, and /sys/devices/platform/coretemp.n will remain present if package n is hotplugged off, but hwmon users should really only be looking for the presence of the hwmon interfaces, whose behaviour remains unchanged.
CVE-2023-53615	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix deletion race condition System crash when using debug kernel due to link list corruption. The cause of the link list corruption is due to session deletion was allowed to queue up twice. Here's the internal trace that show the same port was allowed to double queue for deletion on different cpu. 20808683956 015 qla2xxx [0000:13:00.1]-e801:4: Scheduling sess ffff93ebf9306800 for deletion 50:06:0e:80:12:48:ff:50 fc4_type 1 20808683957 027 qla2xxx [0000:13:00.1]-e801:4: Scheduling sess ffff93ebf9306800 for deletion 50:06:0e:80:12:48:ff:50 fc4_type 1 Move the clearing/setting of deleted flag lock.
CVE-2023-53619	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: Avoid nf_ct_helper_hash uses after free If nf_conntrack_init_start() fails (for example due to a register_nf_conntrack_bpf() failure), the nf_conntrack_helper_fini() clean-up path frees the nf_ct_helper_hash map. When built with NF_CONNTRACK=y, further netfilter modules (e.g: netfilter_conntrack_ftp) can still be loaded and call nf_conntrack_helpers_register(), independently of whether nf_conntrack initialized correctly. This accesses the nf_ct_helper_hash dangling pointer and causes a uaf, possibly leading to random memory corruption. This patch guards nf_conntrack_helper_register() from accessing a freed or uninitialized nf_ct_helper_hash pointer and fixes possible uses-after-free when loading a conntrack module.
CVE-2023-53620	In the Linux kernel, the following vulnerability has been resolved: md: fix soft lockup in status_resync status_resync() will calculate 'curr_resync - recovery_active' to show user a progress bar like following: [=====>.....] resync = 61.4% 'curr_resync' and 'recovery_active' is updated in md_do_sync(), and status_resync() can read them concurrently, hence it's possible that 'curr_resync - recovery_active' can overflow to a huge number. In this case status_resync() will be stuck in the loop to print a large amount of '=', which will end up soft lockup. Fix the problem by setting 'resync' to MD_RESYNC_ACTIVE in this case, this way resync in progress will be reported to user.
CVE-2023-53622	In the Linux kernel, the following vulnerability has been resolved: gfs2: Fix possible data races in gfs2_show_options() Some fields such as gt_logd_secs of the struct gfs2_tune are accessed without holding the lock gt_spin in gfs2_show_options(): val = sdp->sd_tune.gt_logd_secs; if (val != 30) seq_printf(s, ",commit=%d", val); And thus can cause data races when gfs2_show_options() and other functions such as gfs2_reconfigure() are concurrently executed: spin_lock(&gt->gt_spin); gt->gt_logd_secs = newargs->ar_commit; To fix these possible data races, the lock sdp->sd_tune.gt_spin is acquired before accessing the fields of gfs2_tune and released after these accesses. Further changes by Andreas: - Don't hold the spin lock over the seq_printf operations.
CVE-2023-53624	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_fq: fix integer overflow of "credit" if sch_fq is configured with "initial quantum" having values greater than INT_MAX, the first assignment of "credit" does signed integer overflow to a very negative value. In this situation, the syzkaller script provided by Cristoph triggers the CPU soft-lockup warning even with few sockets. It's not an infinite loop, but "credit" wasn't probably meant to be minus 2Gb for each new flow. Capping "initial quantum" to INT_MAX proved to fix the issue. v2: validation of "initial quantum" is done in fq_policy, instead of open coding in fq_change() _ suggested by Jakub Kicinski

CVE ID	Description
CVE-2023-53625	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gvt: fix vgpu debugfs clean in remove Check carefully on root debugfs available when destroying vgpu, e.g in remove case drm minor's debugfs root might already be destroyed, which led to kernel oops like below. Console: switching to colour dummy device 80x25 i915 0000:00:02.0: MDEV: Unregistering intel_vgpu_mdev b1338b2d-a709-4c23-b766-cc436c36cdf0: Removing from iommu group 14 BUG: kernel NULL pointer dereference, address: 0000000000000150 PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP CPU: 3 PID: 1046 Comm: driverctl Not tainted 6.1.0-rc2+ #6 Hardware name: HP HP ProDesk 600 G3 MT/829D, BIOS P02 Ver. 02.44 09/13/2022 RIP: 0010: __lock_acquire+0x5e2/0x1f90 Code: 87 ad 09 00 00 39 05 e1 1e cc 02 0f 82 f1 09 00 00 ba 01 00 00 00 48 83 c4 48 89 d0 5b 5d 41 5c 41 5d 41 5e 41 5f c3 45 31 ff <48> 81 3f 60 9e c2 b6 45 0f 45 f8 83 fe 01 0f 87 55 fa ff ff 89 f0 RSP: 0018:ffff9f770274f948 EFLAGS: 00010046 RAX: 0000000000000003 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000150 RBP: 0000000000000000 R08: 0000000000000001 R09: 0000000000000000 R10: ffff8895d1173300 R11: 0000000000000001 R12: 0000000000000000 R13: 0000000000000150 R14: 0000000000000000 R15: 0000000000000000 FS: 00007fc9b2ba0740(0000) GS:ffff889cdfcc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000150 CR3: 000000010fd93005 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> lock_acquire+0xbff/0x2b0 ? simple_recursive_removal+0xa5/0x2b0 ? lock_release+0x13d/0x2d0 down_write+0x2a/0xd0 ? simple_recursive_removal+0xa5/0x2b0 simple_recursive_removal+0xa5/0x2b0 ? start_creating.part.0+0x110/0x110 ? _raw_spin_unlock+0x29/0x40 debugfs_remove+0x40/0x60 intel_gvt_debugfs_remove_vgpu+0x15/0x30 [kvmgt] intel_gvt_destroy_vgpu+0x60/0x100 [kvmgt] intel_vgpu_release_dev+0xe/0x20 [kvmgt] device_release+0x30/0x80 kobject_put+0x79/0x1b0 device_release_driver_internal+0x1b8/0x230 bus_remove_device+0xec/0x160 device_del+0x189/0x400 ? up_write+0x9c/0x1b0 ? mdev_device_remove_common+0x60/0x60 [mdev] mdev_device_remove_common+0x22/0x60 [mdev] mdev_device_remove_cb+0x17/0x20 [mdev] device_for_each_child+0x56/0x80 mdev_unregister_parent+0x5a/0x81 [mdev] intel_gvt_clean_device+0x2d/0xe0 [kvmgt] intel_gvt_driver_remove+0x2e/0xb0 [i915] i915_driver_remove+0xac/0x100 [i915] i915_pci_remove+0x1a/0x30 [i915] pci_device_remove+0x31/0xa0 device_release_driver_internal+0x1b8/0x230 unbind_store+0xd8/0x100 kernfs_fop_write_iter+0x156/0x210 vfs_write+0x236/0x4a0 ksys_write+0x61/0xd0 do_syscall_64+0x55/0x80 ? find_held_lock+0x2b/0x80 ? lock_release+0x13d/0x2d0 ? up_read+0x17/0x20 ? lock_is_held_type+0xe3/0x140 ? asm_exc_page_fault+0x22/0x30 ? lockdep_hardirqs_on+0x7d/0x100 entry_SYSCALL_64_after_hwframe+0x46/0xb0 RIP: 0033:0x7fc9b2c9e0c4 Code: 15 71 7d 0d 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b7 0f 1f 00 f3 0f 1e fa 80 3d 3d 05 0e 00 00 74 13 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 54 c3 0f 1f 00 48 83 ec 28 48 89 54 24 18 48 RSP: 002b:00007ffec29c81c8 EFLAGS: 00000202 ORIG_RAX: 0000000000000001 RAX: ffffffff8b5f48a0 RBX: 000000000000000d RCX: 00007fc9b2c9e0c4 RDX: 000000000000000d RSI: 0000559f8b5f48a0 RDI: 0000000000000001 RBP: 0000559f8b5f48a0 R08: 0000559f8b5f3540 R09: 00007fc9b2d76d30 R10: 0000000000000000 R11: 0000000000000202 R12: 000000000000000d R13: 00007fc9b2d77780 R14: 000000000000000d R15: 00007fc9b2d72a00 </TASK> Modules linked in: sunrpc intel_rapl_msr intel_rapl_common intel_pmc_core_pltdrv intel_pmc_core intel_tcc_cooling x86_pkg_temp_thermal intel_powerclamp coretemp kvm_intel ee1004 igbvf rapl vfat fat intel_cstate intel_uncore pktcdvd i2c_i801 pcspkr wmi_bmof i2c_smbus acpi_pad vfio_pci vfio_pci_core vfio_virqfd zram fuse dm --- truncated---
CVE-2023-53626	In the Linux kernel, the following vulnerability has been resolved: ext4: fix possible double unlock when moving a directory

CVE ID	Description
CVE-2023-53627	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Grab sas_dev lock when traversing the members of sas_dev.list When freeing slots in function slot_complete_v3_hw(), it is possible that sas_dev.list is being traversed elsewhere, and it may trigger a NULL pointer exception, such as follows: ==>cq thread ==>scsi_eh_6 ==>scsi_error_handler() ==>sas_eh_handle_sas_errors() ==>sas_scsi_find_task() ==>lld_abort_task() ==>slot_complete_v3_hw() ==>hisi_sas_abort_task() ==>hisi_sas_slot_task_free() ==>dereg_device_v3_hw() ==>list_del_init() ==>list_for_each_entry_safe() [7165.434918] sas: Enter sas_scsi_recover_host busy: 32 failed: 32 [7165.434926] sas: trying to find task 0x00000000769b5ba5 [7165.434927] sas: sas_scsi_find_task: aborting task 0x00000000769b5ba5 [7165.434940] hisi_sas_v3_hw 0000:b4:02.0: slot complete: task(00000000769b5ba5) aborted [7165.434964] hisi_sas_v3_hw 0000:b4:02.0: slot complete: task(00000000c9f7aa07) ignored [7165.434965] hisi_sas_v3_hw 0000:b4:02.0: slot complete: task(00000000e2a1cf01) ignored [7165.434968] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 [7165.434972] hisi_sas_v3_hw 0000:b4:02.0: slot complete: task(0000000022d52d93) ignored [7165.434975] hisi_sas_v3_hw 0000:b4:02.0: slot complete: task(0000000066a7516c) ignored [7165.434976] Mem abort info: [7165.434982] ESR = 0x96000004 [7165.434991] Exception class = DABT (current EL), IL = 32 bits [7165.434992] SET = 0, FnV = 0 [7165.434993] EA = 0, S1PTW = 0 [7165.434994] Data abort info: [7165.434994] ISV = 0, ISS = 0x00000004 [7165.434995] CM = 0, WnR = 0 [7165.434997] user pgtable: 4k pages, 48-bit VAs, pgdp = 00000000f29543f2 [7165.434998] [0000000000000000] pgd=0000000000000000 [7165.435003] Internal error: Oops: 96000004 [#1] SMP [7165.439863] Process scsi_eh_6 (pid: 4109, stack limit = 0x00000000c43818d5) [7165.468862] pstate: 00c00009 (nzcw daif +PAN +UAO) [7165.473637] pc : dereg_device_v3_hw +0x68/0xa8 [hisi_sas_v3_hw] [7165.479443] lr : dereg_device_v3_hw+0x2c/0xa8 [hisi_sas_v3_hw] [7165.485247] sp : ffff00001d623bc0 [7165.488546] x29: ffff00001d623bc0 x28: ffffa027d03b9508 [7165.493835] x27: ffff80278ed50af0 x26: ffffa027dd31e0a8 [7165.499123] x25: ffffa027d9b27f88 x24: ffffa027d9b209f8 [7165.504411] x23: ffffa027c45b0d60 x22: ffff80278ec07c00 [7165.509700] x21: 0000000000000008 x20: ffffa027d9b209f8 [7165.514988] x19: ffffa027d9b27f88 x18: ffffffff [7165.520276] x17: 0000000000000000 x16: 0000000000000000 [7165.525564] x15: ffff0000091d9708 x14: ffff0000093b7dc8 [7165.530852] x13: ffff0000093b7a23 x12: 6e7265746e692067 [7165.536140] x11: 0000000000000000 x10: 0000000000000bb0 [7165.541429] x9 : ffff00001d6238f0 x8 : ffffa027d877af00 [7165.546718] x7 : ffffa027d6329600 x6 : ffff7e809f58ca00 [7165.552006] x5 : 0000000000001f8a x4 : 000000000000088e [7165.557295] x3 : ffffa027d9b27fa8 x2 : 0000000000000000 [7165.562583] x1 : 0000000000000000 x0 : 000000003000188e [7165.567872] Call trace: [7165.570309] dereg_device_v3_hw+0x68/0xa8 [hisi_sas_v3_hw] [7165.575775] hisi_sas_abort_task+0x248/0x358 [hisi_sas_main] [7165.581415] sas_eh_handle_sas_errors+0x258/0x8e0 [libsas] [7165.586876] sas_scsi_recover_host+0x134/0x458 [libsas] [7165.592082] scsi_error_handler+0xb4/0x488 [7165.596163] kthread+0x134/0x138 [7165.599380] ret_from_fork+0x10/0x18 [7165.602940] Code: d5033e9f b9000040 aa0103e2 eb03003f (f9400021) [7165.609004] kernel fault(0x1) notification starting on CPU 75 [7165.700728] ---[end trace fc042cbbea224efc]--- [7165.705326] Kernel panic - not syncing: Fatal exception To fix the issue, grab sas_dev lock when traversing the members of sas_dev.list in dereg_device_v3_hw() and hisi_sas_release_tasks() to avoid concurrency of adding and deleting member. When ---truncated---
CVE-2023-53635	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: fix wrong ct->timeout value (struct nf_conn)->timeout is an interval before the conntrack confirmed. After confirmed, it becomes a timestamp. It is observed that timeout of an unconfirmed conntrack: - Set by calling ctnetlink_change_timeout(). As a result, `nfct_time_stamp` was wrongly added to `ct->timeout` twice. - Get by calling ctnetlink_dump_timeout(). As a result, `nfct_time_stamp` was wrongly subtracted. Call Trace: <TASK> dump_stack_lvl ctnetlink_dump_timeout __ctnetlink_glue_build ctnetlink_glue_build __nfqnl_enqueue_packet nf_queue nf_hook_slow ip_mc_output ? __pfx_ip_finish_output ip_send_skb ? __pfx_dst_output udp_send_skb udp_sendmsg ? __pfx_ip_generic_getfrag sock_sendmsg Separate the 2 cases in: - Setting `ct->timeout` in __nf_ct_set_timeout(). - Getting `ct->timeout` in ctnetlink_dump_timeout(). Pablo appends: Update ctnetlink to set up the timeout_after the IPS_CONFIRMED flag is set on, otherwise conntrack creation via ctnetlink breaks. Note that the problem described in this patch occurs since the introduction of the nfnetlink_queue conntrack support, select a sufficiently old Fixes: tag for -stable kernel to pick up this fix.
CVE-2023-53639	In the Linux kernel, the following vulnerability has been resolved: wifi: ath6kl: reduce WARN to dev_dbg() in callback The warn is triggered on a known race condition, documented in the code above the test, that is correctly handled. Using WARN() hinders automated testing. Reducing severity.

CVE ID	Description
CVE-2023-53642	In the Linux kernel, the following vulnerability has been resolved: x86: fix clear_user_rep_good() exception handling annotation This code no longer exists in mainline, because it was removed in commit d2c95f9d6802 ("x86: don't use REP_GOOD or ERMS for user memory clearing") upstream. However, rather than backport the full range of x86 memory clearing and copying cleanups, fix the exception table annotation placement for the final 'rep movsb' in clear_user_rep_good(): rather than pointing at the actual instruction that did the user space access, it pointed to the register move just before it. That made sense from a code flow standpoint, but not from an actual usage standpoint: it means that if user access takes an exception, the exception handler won't actually find the instruction in the exception tables. As a result, rather than fixing it up and returning -EFAULT, it would then turn it into a kernel oops report instead, something like: BUG: unable to handle page fault for address: 0000000020081000 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page ... RIP: 0010:clear_user_rep_good+0x1c/0x30 arch/x86/lib/clear_page_64.S:147 ... Call Trace: __clear_user arch/x86/include/asm/uaccess_64.h:103 [inline] clear_user arch/x86/include/asm/uaccess_64.h:124 [inline] iov_iter_zero+0x709/0x1290 lib/iov_iter.c:800 iomap_dio_hole_iter fs/iomap/direct-io.c:389 [inline] iomap_dio_iter fs/iomap/direct-io.c:440 [inline] __iomap_dio_rw +0xe3d/0x1cd0 fs/iomap/direct-io.c:601 iomap_dio_rw+0x40/0xa0 fs/iomap/direct-io.c:689 ext4_dio_read_iter fs/ext4/file.c:94 [inline] ext4_file_read_iter+0x4be/0x690 fs/ext4/file.c:145 call_read_iter include/linux/fs.h:2183 [inline] do_iter_readv_writev+0x2e0/0x3b0 fs/read_write.c:733 do_iter_read+0x2f2/0x750 fs/read_write.c:796 vfs_readv+0xe5/0x150 fs/read_write.c:916 do_preadv +0x1b6/0x270 fs/read_write.c:1008 __do_sys_preadv2 fs/read_write.c:1070 [inline] __se_sys_preadv2 fs/read_write.c:1061 [inline] __x64_sys_preadv2+0xef/0x150 fs/read_write.c:1061 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x39/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd which then looks like a filesystem bug rather than the incorrect exception annotation that it is. [The alternative to this one-liner fix is to take the upstream series that cleans this all up: 68674f94ffc9 ("x86: don't use REP_GOOD or ERMS for small memory copies") 20f3337d350c ("x86: don't use REP_GOOD or ERMS for small memory clearing") adfcf4231b8c ("x86: don't use REP_GOOD or ERMS for user memory copies") * d2c95f9d6802 ("x86: don't use REP_GOOD or ERMS for user memory clearing") 3639a535587d ("x86: move stac/clac from user copy routines into callers") 577e6a7fd50d ("x86: inline the 'rep movs' in user copies for the FSRM case") 8c9b6a88b7e2 ("x86: improve on the non-rep 'clear_user' function") 427fda2c8a49 ("x86: improve on the non-rep 'copy_user' function") * e046fe5a36a9 ("x86: set FSRs automatically on AMD CPUs that have FSRM") e1f2750edc4a ("x86: remove 'zerorest' argument from __copy_user_nocache()") 034ff37d3407 ("x86: rewrite '__copy_user_nocache' function") with either the whole series or at a minimum the two marked commits being needed to fix this issue]
CVE-2023-53646	In the Linux kernel, the following vulnerability has been resolved: drm/i915/perf: add sentinel to xehp_oa_b_counters Arrays passed to reg_in_range_table should end with empty record. The patch solves KASAN detected bug with signature: BUG: KASAN: global-out-of-bounds in xehp_is_valid_b_counter_addr+0x2c7/0x350 [i915] Read of size 4 at addr ffffffff1555d90 by task perf/1518 CPU: 4 PID: 1518 Comm: perf Tainted: G U 6.4.0-kasan_438-g3303d06107f3+ #1 Hardware name: Intel Corporation Meteor Lake Client Platform/MTL-P DDR5 SODIMM SBS RVP, BIOS MTLPFW11.R00.3223.D80.2305311348 05/31/2023 Call Trace: <TASK> ... xehp_is_valid_b_counter_addr+0x2c7/0x350 [i915] (cherry picked from commit 2f42c5afb34b5696cf5fe79e744f99be9b218798)
CVE-2023-53648	In the Linux kernel, the following vulnerability has been resolved: ALSA: ac97: Fix possible NULL dereference in snd_ac97_mixer smatch error: sound/pci/ac97/ac97_codec.c:2354 snd_ac97_mixer() error: we previously assumed 'rac97' could be null (see line 2072) remove redundant assignment, return error if rac97 is NULL.

CVE ID	Description
CVE-2023-53655	In the Linux kernel, the following vulnerability has been resolved: rcu: Avoid stack overflow due to <code>__rcu_irq_enter_check_tick()</code> being kprobe-ed Registering a kprobe on <code>__rcu_irq_enter_check_tick()</code> can cause kernel stack overflow as shown below. This issue can be reproduced by enabling <code>CONFIG_NO_HZ_FULL</code> and booting the kernel with argument <code>"nohz_full="</code>, and then giving the following commands at the shell prompt: <code># cd /sys/kernel/tracing/ # echo 'p:mp1 __rcu_irq_enter_check_tick' >> kprobe_events # echo 1 > events/kprobes/enable</code> This commit therefore adds <code>__rcu_irq_enter_check_tick()</code> to the kprobes blacklist using <code>NOKPROBE_SYMBOL()</code>. Insufficient stack space to handle exception! ESR: 0x00000000f2000004 -- BRK (AArch64) FAR: 0x0000ffffcf3e510 Task stack: [0xffff80000ad30000..0xffff80000ad38000] IRQ stack: [0xffff800008050000..0xffff800008058000] Overflow stack: [0xffff809c36f9f310..0xffff809c36fa0310] CPU: 5 PID: 190 Comm: bash Not tainted 6.2.0-rc2-00320-g1f5abbd77e2c #19 Hardware name: linux,dummy-virt (DT) pstate: 400003c5 (nZcv DAIF -PAN -UAO -TCO -DIT -SSBS BTYPE=--): pc : <code>__rcu_irq_enter_check_tick+0x0/0x1b8</code> lr : <code>ct_nmi_enter+0x11c/0x138</code> sp : <code>ffff80000ad30080</code> x29: <code>ffff80000ad30080</code> x28: <code>ffff809c82e20000</code> x27: <code>0000000000000000</code> x26: <code>0000000000000000</code> x25: <code>ffff809c02a8d100</code> x24: <code>0000000000000000</code> x23: <code>00000000400003c5</code> x22: <code>0000ffffcf3e510</code> x21: <code>ffff809c36fae148</code> x20: <code>ffff80000ad30120</code> x19: <code>ffff8da8fcce148</code> x18: <code>0000000000000000</code> x17: <code>0000000000000000</code> x16: <code>0000000000000000</code> x15: <code>ffff8da8e44ea6c</code> x14: <code>ffff8da8e44e968</code> x13: <code>ffff8da8e03136c</code> x12: <code>1ffe113804d6809</code> x11: <code>ffff6113804d6809</code> x10: <code>0000000000000a60</code> x9 : <code>dfff800000000000</code> x8 : <code>ffff809c026b404f</code> x7 : <code>00009eec7fb297f7</code> x6 : <code>0000000000000001</code> x5 : <code>ffff80000ad30120</code> x4 : <code>dfff800000000000</code> x3 : <code>ffff8da8e3016f4</code> x2 : <code>0000000000000003</code> x1 : <code>0000000000000000</code> x0 : <code>0000000000000000</code> Kernel panic - not syncing: kernel stack overflow CPU: 5 PID: 190 Comm: bash Not tainted 6.2.0-rc2-00320-g1f5abbd77e2c #19 Hardware name: linux,dummy-virt (DT) Call trace: <code>dump_backtrace+0xf8/0x108</code> <code>show_stack+0x20/0x30</code> <code>dump_stack_lvl+0x68/0x84</code> <code>dump_stack+0x1c/0x38</code> <code>panic+0x214/0x404</code> <code>add_taint+0x0/0xf8</code> <code>panic_bad_stack+0x144/0x160</code> <code>handle_bad_stack+0x38/0x58</code> <code>__bad_stack+0x78/0x7c</code> <code>__rcu_irq_enter_check_tick+0x0/0x1b8</code> <code>arm64_enter_el1_dbg.isra.0+0x14/0x20</code> <code>el1_dbg+0x2c/0x90</code> <code>el1h_64_sync_handler+0xcc/0xe8</code> <code>el1h_64_sync+0x64/0x68</code> <code>__rcu_irq_enter_check_tick+0x0/0x1b8</code> <code>arm64_enter_el1_dbg.isra.0+0x14/0x20</code> <code>el1_dbg+0x2c/0x90</code> <code>el1h_64_sync_handler+0xcc/0xe8</code> <code>el1h_64_sync+0x64/0x68</code> <code>__rcu_irq_enter_check_tick+0x0/0x1b8</code> <code>arm64_enter_el1_dbg.isra.0+0x14/0x20</code> <code>el1_dbg+0x2c/0x90</code> <code>el1h_64_sync_handler+0xcc/0xe8</code> <code>el1h_64_sync+0x64/0x68</code> <code>__rcu_irq_enter_check_tick+0x0/0x1b8</code> <code>arm64_enter_el1_dbg.isra.0+0x14/0x20</code> <code>el1_dbg+0x2c/0x90</code> <code>el1h_64_sync_handler+0xcc/0xe8</code> <code>el1h_64_sync+0x64/0x68</code> <code>__rcu_irq_enter_check_tick+0x0/0x1b8</code> <code>el1_interrupt+0x28/0x60</code> <code>el1h_64_irq_handler+0x18/0x28</code> <code>el1h_64_irq+0x64/0x68</code> <code>__ftrace_set_clr_event_nolock+0x98/0x198</code> <code>__ftrace_set_clr_event+0x58/0x80</code> <code>system_enable_write+0x144/0x178</code> <code>vfs_write+0x174/0x738</code> <code>ksys_write+0xd0/0x188</code> <code>__arm64_sys_write+0x4c/0x60</code> <code>invoke_syscall+0x64/0x180</code> <code>el0_svc_common.constprop.0+0x84/0x160</code> <code>do_el0_svc+0x48/0xe8</code> <code>el0_svc+0x34/0xd0</code> <code>el0t_64_sync_handler+0xb8/0xc0</code> <code>el0t_64_sync+0x190/0x194</code> SMP: stopping secondary CPUs Kernel Offset: 0x28da86000000 from 0xffff800008000000 PHYS_OFFSET: 0xfffff76600000000 CPU features: 0x00000,01a00100,0000421b Memory Limit: none

CVE ID	Description
CVE-2023-53659	In the Linux kernel, the following vulnerability has been resolved: iavf: Fix out-of-bounds when setting channels on remove If we set channels greater during iavf_remove(), and waiting reset done would be timeout, then returned with error but changed num_active_queues directly, that will lead to OOB like the following logs. Because the num_active_queues is greater than tx/rx_rings[] allocated actually. Reproducer: [root@host ~]# cat repro.sh #!/bin/bash pf_dbsf="0000:41:00:0" vf0_dbsf="0000:41:02:0" g_pids=() function do_set_numvfs() { echo 2 >/sys/bus/pci/devices/\${pf_dbsf}/sriov_numvfs sleep \$((RANDOM%3+1)) echo 0 >/sys/bus/pci/devices/\${pf_dbsf}/sriov_numvfs sleep \$((RANDOM%3+1)) } function do_set_channel() { local nic=\$(ls -l --indicator-style=none /sys/bus/pci/devices/\${vf0_dbsf}/net/) [-z "\$nic"] && { sleep \$((RANDOM%3)) ; return 1; } ifconfig \$nic 192.168.18.5 netmask 255.255.255.0 ifconfig \$nic up ethtool -L \$nic combined 1 ethtool -L \$nic combined 4 sleep \$((RANDOM%3)) } function on_exit() { local pid for pid in "\${g_pids[@]"; do kill -0 "\$pid" &>/dev/null && kill "\$pid" &>/dev/null done g_pids=() } trap "on_exit; exit" EXIT while ;; do do_set_numvfs ; done & g_pids+=(\$!) while ;; do do_set_channel ; done & g_pids+=(\$!) wait Result: [3506.152887] iavf 0000:41:02:0: Removing device [3510.400799] ===== [3510.400820] BUG: KASAN: slab-out-of-bounds in iavf_free_all_tx_resources+0x156/0x160 [iavf] [3510.400823] Read of size 8 at addr ffff8b6f9311008 by task repro.sh/55536 [3510.400823] [3510.400830] CPU: 101 PID: 55536 Comm: repro.sh Kdump: loaded Tainted: G O ----- t - 4.18.0 #1 [3510.400832] Hardware name: Powerleader PR2008AL/H12DSi-N6, BIOS 2.0 04/09/2021 [3510.400835] Call Trace: [3510.400851] dump_stack+0x71/0xab [3510.400860] print_address_description+0x6b/0x290 [3510.400865] ? iavf_free_all_tx_resources+0x156/0x160 [iavf] [3510.400868] kasan_report+0x14a/0x2b0 [3510.400873] iavf_free_all_tx_resources+0x156/0x160 [iavf] [3510.400880] iavf_remove+0x2b6/0xc70 [iavf] [3510.400884] ? iavf_free_all_rx_resources+0x160/0x160 [iavf] [3510.400891] ? wait_woken+0x1d0/0x1d0 [3510.400895] ? notifier_call_chain+0xc1/0x130 [3510.400903] pci_device_remove+0xa8/0x1f0 [3510.400910] device_release_driver_internal+0x1c6/0x460 [3510.400916] pci_stop_bus_device+0x101/0x150 [3510.400919] pci_stop_and_remove_bus_device+0xe/0x20 [3510.400924] pci_iiov_remove_virtfn+0x187/0x420 [3510.400927] ? pci_iiov_add_virtfn+0xe10/0xe10 [3510.400929] ? pci_get_subsys+0x90/0x90 [3510.400932] sriov_disable+0xed/0x3e0 [3510.400936] ? bus_find_device+0x12d/0x1a0 [3510.400953] i40e_free_vfs+0x754/0x1210 [i40e] [3510.400966] ? i40e_reset_all_vfs+0x880/0x880 [i40e] [3510.400968] ? pci_get_device+0x7c/0x90 [3510.400970] ? pci_get_subsys+0x90/0x90 [3510.400982] ? pci_vfs_assigned.part.7+0x144/0x210 [3510.400987] ? __mutex_lock_slowpath+0x10/0x10 [3510.400996] i40e_pci_sriov_configure+0x1fa/0x2e0 [i40e] [3510.401001] sriov_numvfs_store+0x214/0x290 [3510.401005] ? sriov_totalvfs_show+0x30/0x30 [3510.401007] ? __mutex_lock_slowpath+0x10/0x10 [3510.401011] ? __check_object_size+0x15a/0x350 [3510.401018] kernfs_fop_write+0x280/0x3f0 [3510.401022] vfs_write+0x145/0x440 [3510.401025] ksys_write+0xab/0x160 [3510.401028] ? __ia32_sys_read+0xb0/0xb0 [3510.401031] ? fput_many+0x1a/0x120 [3510.401032] ? filp_close+0xf0/0x130 [3510.401038] do_syscall_64+0xa0/0x370 [3510.401041] ? page_fault+0x8/0x30 [3510.401043] entry_SYSCALL_64_after_hwframe+0x65/0xca [3510.401073] RIP: 0033:0x7f3a9bb842c0 [3510.401079] Code: 73 01 c3 48 8b 0d d8 cb 2c 00 f7 d8 64 89 01 48 83 c8 ff c3 66 0f 1f 44 00 00 83 3d 89 24 2d 00 00 75 10 b8 01 00 00 0f 05 <48> 3d ---truncated---
CVE-2023-53661	In the Linux kernel, the following vulnerability has been resolved: bnxt: avoid overflow in bnxt_get_nvram_directory() The value of an arithmetic expression is subject of possible overflow due to a failure to cast operands to a larger data type before performing arithmetic. Used macro for multiplication instead operator for avoiding overflow. Found by Security Code and Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-53663	In the Linux kernel, the following vulnerability has been resolved: KVM: nSVM: Check instead of asserting on nested TSC scaling support Check for nested TSC scaling support on nested SVM VMRUN instead of asserting that TSC scaling is exposed to L1 if L1's MSR_AMD64_TSC_RATIO has diverged from KVM's default. Userspace can trigger the WARN at will by writing the MSR and then updating guest CPUID to hide the feature (modifying guest CPUID is allowed anytime before KVM_RUN). E.g. hacking KVM's state_test selftest to do vcpu_set_msr(vcpu, MSR_AMD64_TSC_RATIO, 0); vcpu_clear_cpuid_feature(vcpu, X86_FEATURE_TSCRATEMSR); after restoring state in a new VM +vCPU yields an endless supply of: -----[cut here]----- WARNING: CPU: 164 PID: 62565 at arch/x86/kvm/svm/nested.c:699 nested_vmcb02_prepare_control+0x3d6/0x3f0 [kvm_amd] Call Trace: <TASK> enter_svm_guest_mode+0x114/0x560 [kvm_amd] nested_svm_vmrunk+0x260/0x330 [kvm_amd] vmrun_interception+0x29/0x30 [kvm_amd] svm_invoke_exit_handler+0x35/0x100 [kvm_amd] svm_handle_exit+0xe7/0x180 [kvm_amd] kvm_arch_vcpu_ioctl_run+0x1eab/0x2570 [kvm] kvm_vcpu_ioctl+0x4c9/0x5b0 [kvm] __se_sys_ioctl+0x7a/0xc0 __x64_sys_ioctl+0x21/0x30 do_syscall_64+0x41/0x90 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x45ca1b Note, the nested #VMEXIT path has the same flaw, but needs a different fix and will be handled separately.

CVE ID	Description
CVE-2023-53667	In the Linux kernel, the following vulnerability has been resolved: net: cdc_ncm: Deal with too low values of dwNtbOutMaxSize Currently in cdc_ncm_check_tx_max(), if dwNtbOutMaxSize is lower than the calculated "min" value, but greater than zero, the logic sets tx_max to dwNtbOutMaxSize. This is then used to allocate a new SKB in cdc_ncm_fill_tx_frame() where all the data is handled. For small values of dwNtbOutMaxSize the memory allocated during alloc_skb(dwNtbOutMaxSize, GFP_ATOMIC) will have the same size, due to how size is aligned at alloc time: size = SKB_DATA_ALIGN(size); size += SKB_DATA_ALIGN(sizeof(struct skb_shared_info)); Thus we hit the same bug that we tried to squash with commit 2be6d4d16a084 ("net: cdc_ncm: Allow for dwNtbOutMaxSize to be unset or zero") Low values of dwNtbOutMaxSize do not cause an issue presently because at alloc_skb() time more memory (512b) is allocated than required for the SKB headers alone (320b), leaving some space (512b - 320b = 192b) for CDC data (172b). However, if more elements (for example 3 x u64 = [24b]) were added to one of the SKB header structs, say 'struct skb_shared_info', increasing its original size (320b [320b aligned]) to something larger (344b [384b aligned]), then suddenly the CDC data (172b) no longer fits in the spare SKB data area (512b - 384b = 128b). Consequently the SKB bounds checking semantics fails and panics: skbuff: skb_over_panic: text:fffff831f755b len:184 put:172 head:ffff88811f1c6c00 data:ffff88811f1c6c00 tail:0xb8 end:0x80 dev:<NULL> -----[cut here]----- kernel BUG at net/core/skbuff.c:113! invalid opcode: 0000 [#1] PREEMPT SMP KASAN CPU: 0 PID: 57 Comm: kworker/0:2 Not tainted 5.15.106-syzkaller-00249-g19c0ed55a470 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/14/2023 Workqueue: mld mld_ifc_work RIP: 0010:skb_panic net/core/skbuff.c:113 [inline] RIP: 0010:skb_over_panic+0x14c/0x150 net/core/skbuff.c:118 [snip] Call Trace: <TASK> skb_put+0x151/0x210 net/core/skbuff.c:2047 skb_put_zero include/linux/skbuff.h:2422 [inline] cdc_ncm_ndp16 drivers/net/usb/cdc_ncm.c:1131 [inline] cdc_ncm_fill_tx_frame+0x11ab/0x3da0 drivers/net/usb/cdc_ncm.c:1308 cdc_ncm_tx_fixup+0xa3/0x100 Deal with too low values of dwNtbOutMaxSize, clamp it in the range [USB_CDC_NCM NTB_MIN_OUT_SIZE, CDC_NCM_NTb_MAX_SIZE_TX]. We ensure enough data space is allocated to handle CDC data by making sure dwNtbOutMaxSize is not smaller than USB_CDC_NCM NTB_MIN_OUT_SIZE.
CVE-2023-53668	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix deadlock issue on reading trace_pipe Soft lockup occurs when reading file 'trace_pipe': watchdog: BUG: soft lockup - CPU#6 stuck for 22s! [cat:4488] [...] RIP: 0010:ring_buffer_empty_cpu+0xed/0x170 RSP: 0018:ffff88810dd6fc48 EFLAGS: 00000246 RAX: 0000000000000000 RBX: 0000000000000246 RCX: ffffffff93d1aaeb RDX: ffff88810a280040 RSI: 0000000000000008 RDI: ffff88811164b218 RBP: ffff88811164b218 R08: 0000000000000000 R09: ffff88815156600f R10: ffffd102a2acc01 R11: 0000000000000001 R12: 0000000051651901 R13: 0000000000000000 R14: ffff888115e49500 R15: 0000000000000000 [...] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f8d853c2000 CR3: 000000010dcd8000 CR4: 000000000000006e DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000040 Call Trace: __find_next_entry+0x1a8/0x4b0 ? peek_next_entry+0x250/0x250 ? down_write+0xa5/0x120 ? down_write_killable+0x130/0x130 trace_find_next_entry_inc+0x3b/0x1d0 tracing_read_pipe+0x423/0xae0 ? tracing_splice_read_pipe+0xcb0/0xcb0 vfs_read+0x16b/0x490 ksys_read+0x105/0x210 ? __ia32_sys_pwrite64+0x200/0x200 ? switch_fpu_return+0x108/0x220 do_syscall_64+0x33/0x40 entry_SYSCALL_64_after_hwframe+0x61/0xc6 Through the vmcore, I found it's because in tracing_read_pipe(), ring_buffer_empty_cpu() found some buffer is not empty but then it cannot read anything due to "rb_num_of_entries() == 0" always true. Then it infinitely loop the procedure due to user buffer not been filled, see following code path: tracing_read_pipe() { waitagain: tracing_wait_pipe() // 1. find non-empty buffer here trace_find_next_entry_inc() // 2. loop here try to find an entry __find_next_entry() ring_buffer_empty_cpu(); // 3. find non-empty buffer peek_next_entry() // 4. but peek always return NULL ring_buffer_peek() rb_buffer_peek() rb_get_reader_page() // 5. because rb_num_of_entries() == 0 always true here // then return NULL // 6. user buffer not been filled so goto 'waitagain' // and eventually leads to an deadlock in kernel!!! } By some analyzing, I found that when resetting ringbuffer, the 'entries' of its pages are not all cleared (see rb_reset_cpu()). Then when reducing the ringbuffer, and if some reduced pages exist dirty 'entries' data, they will be added into 'cpu_buffer->overrun' (see rb_remove_pages()), which cause wrong 'overrun' count and eventually cause the deadlock issue. To fix it, we need to clear every pages in rb_reset_cpu().

CVE ID	Description
CVE-2023-53671	In the Linux kernel, the following vulnerability has been resolved: srcu: Delegate work to the boot cpu if using SRCU_SIZE_SMALL Commit 994f706872e6 ("srcu: Make Tree SRCU able to operate without snp_node array") assumes that cpu 0 is always online. However, there really are situations when some other CPU is the boot CPU, for example, when booting a kdump kernel with the maxcpus=1 boot parameter. On PowerPC, the kdump kernel can hang as follows: ... [1.740036] systemd[1]: Hostname set to <xyz.com> [243.686240] INFO: task systemd:1 blocked for more than 122 seconds. [243.686264] Not tainted 6.1.0-rc1 #1 [243.686272] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [243.686281] task:systemd state:D stack:0 pid:1 ppid:0 flags:0x00042000 [243.686296] Call Trace: [243.686301] [c000000016657640] [c000000016657670] 0xc000000016657670 (unreliable) [243.686317] [c000000016657830] [c00000001001dec0] __switch_to+0x130/0x220 [243.686333] [c000000016657890] [c000000010f607b8] __schedule+0x1f8/0x580 [243.686347] [c000000016657940] [c000000010f60bb4] schedule+0x74/0x140 [243.686361] [c0000000166579b0] [c000000010f699b8] schedule_timeout+0x168/0x1c0 [243.686374] [c000000016657a80] [c000000010f61de8] __wait_for_common+0x148/0x360 [243.686387] [c000000016657b20] [c000000010176bb0] __flush_work.isra.0+0x1c0/0x3d0 [243.686401] [c000000016657bb0] [c0000000105f2768] fsnotify_wait_marks_destroyed+0x28/0x40 [243.686415] [c000000016657bd0] [c0000000105f21b8] fsnotify_destroy_group+0x68/0x160 [243.686428] [c000000016657c40] [c0000000105f6500] inotify_release+0x30/0xa0 [243.686440] [c000000016657cb0] [c0000000105751a8] __fput+0xc8/0x350 [243.686452] [c000000016657d00] [c00000001017d524] task_work_run+0xe4/0x170 [243.686464] [c000000016657d50] [c000000010020e94] do_notify_resume+0x134/0x140 [243.686478] [c000000016657d80] [c00000001002eb18] interrupt_exit_user_prepare_main+0x198/0x270 [243.686493] [c000000016657de0] [c00000001002ec60] syscall_exit_prepare+0x70/0x180 [243.686505] [c000000016657e10] [c00000001000bf7c] system_call_vectored_common+0xfc/0x280 [243.686520] --- interrupt: 3000 at 0x7ffa47d5ba4 [243.686528] NIP: 00007ffa47d5ba4 LR: 0000000000000000 CTR: 0000000000000000 [243.686538] REGS: c000000016657e80 TRAP: 3000 Not tainted (6.1.0-rc1) [243.686548] MSR: 800000000000d033 <SF,EE,PR,ME,IR,DR,RI,LE> CR: 42044440 XER: 00000000 [243.686572] IRQMASK: 0 [243.686572] GPR00: 0000000000000000 00007ffa606710 00007ffa48e7200 0000000000000000 [243.686572] GPR04: 0000000000000000 000000000000000a 0000000000000000 0000000000000001 [243.686572] GPR08: 00001000c172dd0 0000000000000000 0000000000000000 0000000000000000 [243.686572] GPR12: 0000000000000000 00007ffa4ff4bc0 0000000000000000 0000000000000000 [243.686572] GPR16: 0000000000000000 0000000000000000 0000000000000000 0000000000000000 [243.686572] GPR20: 0000000132dfdc50 000000000000000e 0000000000189375 0000000000000000 [243.686572] GPR24: 00007fffa606ae0 0000000000000005 00001000c185490 00001000c172570 [243.686572] GPR28: 00001000c172990 00001000c184850 00001000c172e00 00007ffa4fedd98 [243.686683] NIP [00007ffa47d5ba4] 0x7ffa47d5ba4 [243.686691] LR [0000000000000000] 0x0 [243.686698] --- interrupt: 3000 [243.686708] INFO: task kworker/u16:1:24 blocked for more than 122 seconds. [243.686717] Not tainted 6.1.0-rc1 #1 [243.686724] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [243.686733] task:kworker/u16:1 state:D stack:0 pid:24 ppid:2 flags:0x00000800 [243.686747] Workqueue: events_unbound fsnotify_mark_destroy_workfn [243.686758] Call Trace: [243.686762] [c0000000166736e0] [c00000004fd91000] 0xc00000004fd91000 (unreliable) [243.686775] [c0000000166738d0] [c00000001001dec0] __switch_to+0x130/0x220 [243.686788] [c000000016673930] [c000000010f607b8] __schedule+0x1f8/0x ---truncated---

CVE ID	Description
CVE-2023-53673	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: call disconnect callback before deleting conn In hci_cs_disconnect, we do hci_conn_del even if disconnection failed. ISO, L2CAP and SCO connections refer to the hci_conn without hci_conn_get, so disconn_cfm must be called so they can clean up their conn, otherwise use-after-free occurs. ISO: ===== iso_sock_connect:880: sk 00000000eabd6557 iso_connect_cis:356: 70:1a:b8:98:ff:a2 -> 28:3d:c2:4a:7e:da ... iso_conn_add:140: hcon 000000001696f1fd conn 00000000b6251073 hci_dev_put:1487: hci0 orig refcnt 17 __iso_chan_add:214: conn 00000000b6251073 iso_sock_clear_timer:117: sock 00000000eabd6557 state 3 ... hci_rx_work:4085: hci0 Event packet hci_event_packet:7601: hci0: event 0x0f hci_cmd_status_evt:4346: hci0: opcode 0x0406 hci_cs_disconnect:2760: hci0: status 0x0c hci_sent_cmd_data:3107: hci0 opcode 0x0406 hci_conn_del:1151: hci0 hcon 000000001696f1fd handle 2560 hci_conn_unlink:1102: hci0: hcon 000000001696f1fd hci_conn_drop:1451: hcon 00000000d8521aaf orig refcnt 2 hci_chan_list_flush:2780: hcon 000000001696f1fd hci_dev_put:1487: hci0 orig refcnt 21 hci_dev_put:1487: hci0 orig refcnt 20 hci_req_cmd_complete:3978: opcode 0x0406 status 0x0c ... <no iso_* activity on sk/conn> ... iso_sock_sendmsg:1098: sock 00000000dea5e2e0, sk 00000000eabd6557 BUG: kernel NULL pointer dereference, address: 0000000000000668 PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP PTI Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.2-1.fc38 04/01/2014 RIP: 0010:iso_sock_sendmsg (net/bluetooth/iso.c:1112) bluetooth ===== L2CAP: ===== hci_cmd_status_evt:4359: hci0: opcode 0x0406 hci_cs_disconnect:2760: hci0: status 0x0c hci_sent_cmd_data:3085: hci0 opcode 0x0406 hci_conn_del:1151: hci0 hcon ffff88800c999000 handle 3585 hci_conn_unlink:1102: hci0: hcon ffff88800c999000 hci_chan_list_flush:2780: hcon ffff88800c999000 hci_chan_del:2761: hci0 hcon ffff88800c999000 chan ffff888018ddd280 ... BUG: KASAN: slab-use-after-free in hci_send_acl+0x2d/0x540 [bluetooth] Read of size 8 at addr ffff888018ddd298 by task bluetoothd/1175 CPU: 0 PID: 1175 Comm: bluetoothd Tainted: G E 6.4.0-rc4+ #2 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.2-1.fc38 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x5b/0x90 print_report+0xcf/0x670 ? __virt_addr_valid+0xf8/0x180 ? hci_send_acl+0x2d/0x540 [bluetooth] kasan_report+0xa8/0xe0 ? hci_send_acl+0x2d/0x540 [bluetooth] hci_send_acl+0x2d/0x540 [bluetooth] ? __pfx__lock_acquire+0x10/0x10 l2cap_chan_send+0x1fd/0x1300 [bluetooth] ? l2cap_sock_sendmsg+0xf2/0x170 [bluetooth] ? __pfx_l2cap_chan_send+0x10/0x10 [bluetooth] ? lock_release+0x1d5/0x3c0 ? mark_held_locks+0x1a/0x90 l2cap_sock_sendmsg+0x100/0x170 [bluetooth] sock_write_iter+0x275/0x280 ? __pfx_sock_write_iter+0x10/0x10 ? __pfx__lock_acquire+0x10/0x10 do_iter_readv_writev+0x176/0x220 ? __pfx_do_iter_readv_writev+0x10/0x10 ? find_held_lock+0x83/0xa0 ? selinux_file_permission+0x13e/0x210 do_iter_write+0xda/0x340 vfs_writev+0x1b4/0x400 ? __pfx_vfs_writev+0x10/0x10 ? __seccomp_filter+0x112/0x750 ? populate_seccomp_data+0x182/0x220 ? __fget_light+0xdf/0x100 ? do_writev+0x19d/0x210 do_writev+0x19d/0x210 ? __pfx_do_writev+0x10/0x10 ? mark_held_locks+0x1a/0x90 do_syscall_64+0x60/0x90 ? lockdep_hardirqs_on_prepare+0x149/0x210 ? do_syscall_64+0x6c/0x90 ? lockdep_hardirqs_on_prepare+0x149/0x210 entry_SYSCALL_64_after_hwframe+0x72/0xdc RIP: 0033:0x7ff45cb23e64 Code: 15 d1 1f 0d 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b8 0f 1f 00 f3 0f 1e fa 80 3d 9d a7 0d 00 00 74 13 b8 14 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 54 c3 0f 1f 00 48 83 ec 28 89 54 24 1c 48 89 RSP: 002b:00007fff21ae09b8 EFLAGS: 00000202 ORIG_RAX: 0000000000000014 RAX: ffffffffefda RBX: ---truncated---
CVE-2023-53675	In the Linux kernel, the following vulnerability has been resolved: scsi: ses: Fix possible desc_ptr out-of-bounds accesses Sanitize possible desc_ptr out-of-bounds accesses in ses_enclosure_data_process().
CVE-2023-53676	In the Linux kernel, the following vulnerability has been resolved: scsi: target: iscsi: Fix buffer overflow in lio_target_nacl_info_show() The function lio_target_nacl_info_show() uses sprintf() in a loop to print details for every iSCSI connection in a session without checking for the buffer length. With enough iSCSI connections it's possible to overflow the buffer provided by configs and corrupt the memory. This patch replaces sprintf() with sysfs_emit_at() that checks for buffer boundaries.

567

CVE ID	Description
CVE-2023-53679	In the Linux kernel, the following vulnerability has been resolved: wifi: mt7601u: fix an integer underflow Fix an integer underflow that leads to a null pointer dereference in 'mt7601u_rx_skb_from_seg()'. The variable 'dma_len' in the URB packet could be manipulated, which could trigger an integer underflow of 'seg_len' in 'mt7601u_rx_process_seg()'. This underflow subsequently causes the 'bad_frame' checks in 'mt7601u_rx_skb_from_seg()' to be bypassed, eventually leading to a dereference of the pointer 'p', which is a null pointer. Ensure that 'dma_len' is greater than 'min_seg_len'. Found by a modified version of syzkaller. KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] CPU: 0 PID: 12 Comm: ksoftirqd/0 Tainted: G W O 5.14.0+ #139 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.1-0-ga5cab58e9a3f-prebuilt.qemu.org 04/01/2014 RIP: 0010:skb_add_rx_frag+0x143/0x370 Code: e2 07 83 c2 03 38 ca 7c 08 84 c9 0f 85 86 01 00 00 4c 8d 7d 08 44 89 68 08 48 b8 00 00 00 00 00 fc ff df 4c 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 cd 01 00 00 48 8b 45 08 a8 01 0f 85 3d 01 00 00 RSP: 0018:ffffc9000000cfc90 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: ffff888115520dc0 RCX: 0000000000000000 RDX: 0000000000000001 RSI: ffff8881118430c0 RDI: ffff8881118430f8 RBP: 0000000000000000 R08: 00000000000000e09 R09: 00000000000000010 R10: ffff888111843017 R11: ffffd1022308602 R12: 0000000000000000 R13: 00000000000000e09 R14: 00000000000000010 R15: 0000000000000008 FS: 0000000000000000(0000) GS:ffff88811a800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000004035af40 CR3: 00000001157f2000 CR4: 0000000000750ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 PKRU: 55555554 Call Trace: mt7601u_rx_tasklet+0xc73/0x1270 ? mt7601u_submit_rx_buf.isra.0+0x510/0x510 ? tasklet_action_common.isra.0+0x79/0x2f0 tasklet_action_common.isra.0+0x206/0x2f0 __do_softirq+0x1b5/0x880 ? tasklet_unlock+0x30/0x30 run_ksoftirqd+0x26/0x50 smpboot_thread_fn+0x34f/0x7d0 ? smpboot_register_percpu_thread+0x370/0x370 kthread+0x3a1/0x480 ? set_kthread_struct+0x120/0x120 ret_from_fork+0x1f/0x30 Modules linked in: 88XXau(O) 88x2bu(O) ---[end trace 57f34f93b4da0f9b]--- RIP: 0010:skb_add_rx_frag+0x143/0x370 Code: e2 07 83 c2 03 38 ca 7c 08 84 c9 0f 85 86 01 00 00 4c 8d 7d 08 44 89 68 08 48 b8 00 00 00 00 00 fc ff df 4c 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 cd 01 00 00 48 8b 45 08 a8 01 0f 85 3d 01 00 00 RSP: 0018:ffffc9000000cfc90 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: ffff888115520dc0 RCX: 0000000000000000 RDX: 0000000000000001 RSI: ffff8881118430c0 RDI: ffff8881118430f8 RBP: 0000000000000000 R08: 00000000000000e09 R09: 00000000000000010 R10: ffff888111843017 R11: ffffd1022308602 R12: 0000000000000000 R13: 00000000000000e09 R14: 00000000000000010 R15: 0000000000000008 FS: 0000000000000000(0000) GS:ffff88811a800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000004035af40 CR3: 00000001157f2000 CR4: 0000000000750ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 PKRU: 55555554
CVE-2023-53680	In the Linux kernel, the following vulnerability has been resolved: NFSD: Avoid calling OPDESC() with ops->opnum == OP_ILLEGAL OPDESC() simply indexes into nfsd4_ops[] by the op's operation number, without range checking that value. It assumes callers are careful to avoid calling it with an out-of-bounds opnum value. nfsd4_decode_compound() is not so careful, and can invoke OPDESC() with opnum set to OP_ILLEGAL, which is 10044 -- well beyond the end of nfsd4_ops[].
CVE-2023-53681	In the Linux kernel, the following vulnerability has been resolved: bcache: Fix __bch_btree_node_alloc to make the failure behavior consistent In some specific situations, the return value of __bch_btree_node_alloc may be NULL. This may lead to a potential NULL pointer dereference in caller function like a calling chain : btree_split->bch_btree_node_alloc->__bch_btree_node_alloc. Fix it by initializing the return value in __bch_btree_node_alloc.
CVE-2023-53682	In the Linux kernel, the following vulnerability has been resolved: hwmon: (xgene) Fix ioremap and memremap leak Smatch reports: drivers/hwmon/xgene-hwmon.c:757 xgene_hwmon_probe() warn: 'ctx->pcc_comm_addr' from ioremap() not released on line: 757. This is because in drivers/hwmon/xgene-hwmon.c:701 xgene_hwmon_probe(), ioremap and memremap is not released, which may cause a leak. To fix this, ioremap and memremap is modified to devm_ioremap and devm_memremap. [groeck: Fixed formatting and subject]
CVE-2023-53684	In the Linux kernel, the following vulnerability has been resolved: xfrm: Zero padding when dumping algos and encap When copying data to user-space we should ensure that only valid data is copied over. Padding in structures may be filled with random (possibly sensitive) data and should never be given directly to user-space. This patch fixes the copying of xfrm algorithms and the encap template in xfrm_user so that padding is zeroed.

CVE ID	Description
CVE-2023-53685	In the Linux kernel, the following vulnerability has been resolved: tun: Fix memory leak for detached NAPI queue. syzkaller reported [0] memory leaks of sk and skb related to the TUN device with no repro, but we can reproduce it easily with: struct ifreq ifr = {}; int fd_tun, fd_tmp; char buf[4] = {}; fd_tun = openat(AT_FDCWD, "/dev/net/tun", O_WRONLY, 0); ifr.ifr_flags = IFF_TUN IFF_NAPI IFF_MULTI_QUEUE; ioctl(fd_tun, TUNSETIFF, &ifr); ifr.ifr_flags = IFF_DETACH_QUEUE; ioctl(fd_tun, TUNSETQUEUE, &ifr); fd_tmp = socket(AF_PACKET, SOCK_PACKET, 0); ifr.ifr_flags = IFF_UP; ioctl(fd_tmp, SIOCSIFFLAGS, &ifr); write(fd_tun, buf, sizeof(buf)); close(fd_tun); If we enable NAPI and multi-queue on a TUN device, we can put skb into tfile->sk.sk_write_queue after the queue is detached. We should prevent it by checking tfile->detached before queuing skb. Note this must be done under tfile->sk.sk_write_queue.lock because write() and ioctl(IFF_DETACH_QUEUE) can run concurrently. Otherwise, there would be a small race window: write() ioctl(IFF_DETACH_QUEUE) - tun_get_user - __tun_detach - if (tfile->detached) - tun_disable_queue -> false - tfile->detached = tun - tun_queue_purge - spin_lock_bh(&queue->lock) - __skb_queue_tail(queue, skb) Another solution is to call tun_queue_purge() when closing and reattaching the detached queue, but it could paper over another problems. Also, we do the same kind of test for IFF_NAPI_FRAGS. [0]: unreferenced object 0xffff88801edbc800 (size 2048): comm "syz-executor.1", pid 33269, jiffies 4295743834 (age 18.756s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 40 00 00 00 00 00 00 00 00 00 00 00 00 ...@..... backtrace: [<000000008c16ea3d>] __do_kmalloc_node mm/slab_common.c:965 [inline] [<000000008c16ea3d>] __kmalloc+0x4a/0x130 mm/slab_common.c:979 [<000000003addde56>] kmalloc include/linux/slab.h:563 [inline] [<000000003addde56>] sk_prot_alloc+0xef/0x1b0 net/core/sock.c:2035 [<000000003e20621f>] sk_alloc+0x36/0x2f0 net/core/sock.c:2088 [<0000000028e43843>] tun_chr_open+0x3d/0x190 drivers/net/tun.c:3438 [<000000001b0f1f28>] misc_open+0x1a6/0x1f0 drivers/char/misc.c:165 [<000000004376f706>] chrdev_open+0x111/0x300 fs/char_dev.c:414 [<00000000614d379f>] do_dentry_open+0x2f9/0x750 fs/open.c:920 [<000000008eb24774>] do_open fs/namei.c:3636 [inline] [<000000008eb24774>] path_openat+0x143f/0x1a30 fs/namei.c:3791 [<000000000955077b5>] do_filp_open+0xce/0x1c0 fs/namei.c:3818 [<00000000b78973b0>] do_sys_openat2+0xf0/0x260 fs/open.c:1356 [<00000000057be699>] do_sys_open fs/open.c:1372 [inline] [<00000000057be699>] __do_sys_openat fs/open.c:1388 [inline] [<00000000057be699>] __se_sys_openat fs/open.c:1383 [inline] [<00000000057be699>] __x64_sys_openat+0x83/0xf0 fs/open.c:1383 [<00000000a7d2182d>] do_syscall_x64 arch/x86/entry/common.c:50 [inline] [<00000000a7d2182d>] do_syscall_64+0x3c/0x90 arch/x86/entry/common.c:80 [<000000004cc4e8c4>] entry_SYSCALL_64_after_hwframe+0x72/0xdc unreferenced object 0xffff88802f671700 (size 240): comm "syz-executor.1", pid 33269, jiffies 4295743854 (age 18.736s) hex dump (first 32 bytes): 68 c9 db 1e 80 88 ff ff 68 c9 db 1e 80 88 ff ff h.....h..... 00 c0 7b 2f 80 88 ff ff 00 c8 db 1e 80 88 ff ff ..{/..... backtrace: [<00000000e9d9fdb6>] __alloc_skb+0x223/0x250 net/core/skbuff.c:644 [<000000002c3e4e0b>] alloc_skb include/linux/skbuff.h:1288 [inline] [<000000002c3e4e0b>] alloc_skb_with_frags+0x6f/0x350 net/core/skbuff.c:6378 [<00000000825f98d7>] sock_alloc_send_skb+0x3ac/0x3e0 net/core/sock.c:2729 [<00000000e9b3df3>] tun_alloc_skb drivers/net/tun.c:1529 [inline] [< ---truncated---

CVE ID	Description
CVE-2023-53692	In the Linux kernel, the following vulnerability has been resolved: ext4: fix use-after-free read in ext4_find_extent for bigalloc + inline Syzbot found the following issue: loop0: detected capacity change from 0 to 2048 EXT4-fs (loop0): mounted filesystem 00000000-0000-0000-0000-000000000000 without journal. Quota mode: none. ===== BUG: KASAN: use-after-free in ext4_ext_binsearch_idx fs/ext4/extents.c:768 [inline] BUG: KASAN: use-after-free in ext4_find_extent+0x76e/0xd90 fs/ext4/extents.c:931 Read of size 4 at addr ffff888073644750 by task syz-executor420/5067 CPU: 0 PID: 5067 Comm: syz-executor420 Not tainted 6.2.0-rc1-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/26/2022 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x1b1/0x290 lib/dump_stack.c:106 print_address_description+0x74/0x340 mm/kasan/report.c:306 print_report+0x107/0x1f0 mm/kasan/report.c:417 kasan_report+0xcd/0x100 mm/kasan/report.c:517 ext4_ext_binsearch_idx fs/ext4/extents.c:768 [inline] ext4_find_extent+0x76e/0xd90 fs/ext4/extents.c:931 ext4_clu_mapped+0x117/0x970 fs/ext4/extents.c:5809 ext4_insert_delayed_block fs/ext4/inode.c:1696 [inline] ext4_da_map_blocks fs/ext4/inode.c:1806 [inline] ext4_da_get_block_prep+0x9e8/0x13c0 fs/ext4/inode.c:1870 ext4_block_write_begin+0x6a8/0x2290 fs/ext4/inode.c:1098 ext4_da_write_begin+0x539/0x760 fs/ext4/inode.c:3082 generic_perform_write+0x2e4/0x5e0 mm/filemap.c:3772 ext4_buffered_write_iter+0x122/0x3a0 fs/ext4/file.c:285 ext4_file_write_iter+0x1d0/0x18f0 call_write_iter include/linux/fs.h:2186 [inline] new_sync_write fs/read_write.c:491 [inline] vfs_write+0x7dc/0xc50 fs/read_write.c:584 ksys_write+0x177/0x2a0 fs/read_write.c:637 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0x7f4b7a9737b9 RSP: 002b:00007ffc5cac3668 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 RAX: ffffffffda RBX: 0000000000000000 RCX: 00007f4b7a9737b9 RDX: 00000000175d9003 RSI: 0000000020000200 RDI: 0000000000000004 RBP: 00007f4b7a933050 R08: 0000000000000000 R09: 0000000000000000 R10: 000000000000079f R11: 0000000000000246 R12: 00007f4b7a9330e0 R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 </TASK> Above issue is happens when enable bigalloc and inline data feature. As commit 131294c35ed6 fixed delayed allocation bug in ext4_clu_mapped for bigalloc + inline. But it only resolved issue when has inline data, if inline data has been converted to extent(ext4_da_convert_inline_data_to_extent) before writepages, there is no EXT4_STATE_MAY_INLINE_DATA flag. However i_data is still store inline data in this scene. Then will trigger UAF when find extent. To resolve above issue, there is need to add judge "ext4_has_inline_data(inode)" in ext4_clu_mapped().
CVE-2023-53695	In the Linux kernel, the following vulnerability has been resolved: udf: Detect system inodes linked into directory hierarchy When UDF filesystem is corrupted, hidden system inodes can be linked into directory hierarchy which is an avenue for further serious corruption of the filesystem and kernel confusion as noticed by syzbot fuzzed images. Refuse to access system inodes linked into directory hierarchy and vice versa.
CVE-2023-53696	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix memory leak in qla2x00_probe_one() There is a memory leak reported by kmemleak: unreferenced object 0xffffc900003f0000 (size 12288): comm "modprobe", pid 19117, jiffies 4299751452 (age 42490.264s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<00000000629261a8>] __vmalloc_node_range+0xe56/0x1110 [<0000000001906886>] __vmalloc_node+0xbd/0x150 [<000000005bb4dc34>] vmalloc+0x25/0x30 [<00000000a2dc1194>] qla2x00_create_host+0x7a0/0xe30 [qla2xxx] [<0000000062b14b47>] qla2x00_probe_one+0x2eb8/0xd160 [qla2xxx] [<00000000641ccc04>] local_pci_probe+0xeb/0x1a0 The root cause is traced to an error-handling path in qla2x00_probe_one() when the adapter "base_vha" initialize failed. The fab_scan_rp "scan.l" is used to record the port information and it is allocated in qla2x00_create_host(). However, it is not released in the error handling path "probe_failed". Fix this by freeing the memory of "scan.l" when an error occurs in the adapter initialization process.
CVE-2023-53697	In the Linux kernel, the following vulnerability has been resolved: nvdim: Fix memleak of pmu attr_groups in unregister_nvdim_pmu() Memory pointed by 'nd_pmu->pmu.attr_groups' is allocated in function 'register_nvdim_pmu' and is lost after 'kfree(nd_pmu)' call in function 'unregister_nvdim_pmu'.

CVE ID	Description
CVE-2023-53698	In the Linux kernel, the following vulnerability has been resolved: xsk: fix refcount underflow in error path Fix a refcount underflow problem reported by syzbot that can happen when a system is running out of memory. If <code>xp_alloc_tx_descs()</code> fails, and it can only fail due to not having enough memory, then the error path is triggered. In this error path, the refcount of the pool is decremented as it has incremented before. However, the reference to the pool in the socket was not nulled. This means that when the socket is closed later, the socket teardown logic will think that there is a pool attached to the socket and try to decrease the refcount again, leading to a refcount underflow. I chose this fix as it involved adding just a single line. Another option would have been to move <code>xp_get_pool()</code> and the assignment of <code>xs->pool</code> to after the if-statement and using <code>xs_umem->pool</code> instead of <code>xs->pool</code> in the whole if-statement resulting in somewhat simpler code, but this would have led to much more churn in the code base perhaps making it harder to backport.
CVE-2023-53705	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix out-of-bounds access in <code>ipv6_find_tlv()</code> <code>optlen</code> is fetched without checking whether there is more than one byte to parse. It can lead to out-of-bounds access. Found by InfoTeCS on behalf of Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2023-53706	In the Linux kernel, the following vulnerability has been resolved: mm/vmemmap/devdax: fix kernel crash when probing devdax devices commit 4917f55b4ef9 ("mm/sparse-vmemmap: improve memory savings for compound devmaps") added support for using optimized vmemmap for devdax devices. But how vmemmap mappings are created are architecture specific. For example, powerpc with hash translation doesn't have vmemmap mappings in <code>init_mm</code> page table instead they are bolted table entries in the hardware page table <code>vmemmap_populate_compound_pages()</code> used by vmemmap optimization code is not aware of these architecture-specific mapping. Hence allow architecture to opt for this feature. I selected architectures supporting <code>HUGETLB_PAGE_OPTIMIZE_VMEMMAP</code> option as also supporting this feature. This patch fixes the below crash on ppc64. BUG: Unable to handle kernel data access on write at 0xc00c000100400038 Faulting instruction address: 0xc000000001269d90 Oops: Kernel access of bad area, sig: 11 [#1] LE PAGE_SIZE=64K MMU=Hash SMP NR_CPUS=2048 NUMA pSeries Modules linked in: CPU: 7 PID: 1 Comm: swapper/0 Not tainted 6.3.0-rc5-150500.34-default+ #2 5c90a668b6bbd142599890245c2fb5de19d7d28a Hardware name: IBM,9009-42G POWER9 (raw) 0x4e0202 0xf0000005 of:IBM,FW950.40 (VL950_099) hv:phyp pSeries NIP: c000000001269d90 LR: c0000000004c57d4 CTR: 0000000000000000 REGS: c0000000003632c30 TRAP: 0300 Not tainted (6.3.0-rc5-150500.34-default+) MSR: 8000000000009033 <SF,EE,ME,IR,DR,RI,LE> CR: 24842228 XER: 00000000 CFAR: c0000000004c57d0 DAR: c00c000100400038 DSISR: 42000000 IRQMASK: 0 NIP [c000000001269d90] __init_single_page.isra.74+0x14/0x4c LR [c0000000004c57d4] __init_zone_device_page+0x44/0xd0 Call Trace: [c000000003632ed0] [c000000003632f60] 0xc000000003632f60 (unreliable) [c000000003632f10] [c0000000004c5ca0] memmap_init_zone_device+0x170/0x250 [c000000003632fe0] [c0000000000557f8] memremap_pages+0x2c8/0x7f0 [c0000000036330c0] [c000000000557b5c] devm_memremap_pages+0x3c/0xa0 [c000000003633100] [c00000000d458a8] dev_dax_probe+0x108/0x3e0 [c0000000036331a0] [c00000000d41430] dax_bus_probe+0xb0/0x140 [c0000000036331d0] [c00000000cef27c] really_probe+0x19c/0x520 [c000000003633260] [c00000000cef6b4] __driver_probe_device+0xb4/0x230 [c0000000036332e0] [c00000000cef888] driver_probe_device+0x58/0x120 [c000000003633320] [c00000000cef6a6] __device_attach_driver+0x11c/0x1e0 [c0000000036333a0] [c00000000cebc58] bus_for_each_drv+0xa8/0x130 [c000000003633400] [c00000000ceefcc] __device_attach+0x15c/0x250 [c0000000036334a0] [c00000000ced458] bus_probe_device+0x108/0x110 [c0000000036334f0] [c00000000ce92dc] device_add+0x7fc/0xa10 [c0000000036335b0] [c00000000d447c8] devm_create_dev_dax+0x1d8/0x530 [c000000003633640] [c00000000d46b60] __dax_pmem_probe+0x200/0x270 [c0000000036337b0] [c00000000d46bf0] dax_pmem_probe+0x20/0x70 [c0000000036337d0] [c00000000d2279c] nvdimmbus_probe+0xac/0x2b0 [c000000003633860] [c00000000cef27c] really_probe+0x19c/0x520 [c0000000036338f0] [c00000000cef6b4] __driver_probe_device+0xb4/0x230 [c000000003633970] [c00000000cef888] driver_probe_device+0x58/0x120 [c0000000036339b0] [c00000000cefd08] __driver_attach+0x1d8/0x240 [c000000003633a30] [c00000000cebb04] bus_for_each_dev+0xb4/0x130 [c000000003633a90] [c00000000cee564] driver_attach+0x34/0x50 [c000000003633ab0] [c00000000ced78] bus_add_driver+0x218/0x300 [c000000003633b40] [c00000000cf1144] driver_register+0xa4/0x1b0 [c000000003633bb0] [c00000000d21a0c] __nd_driver_register+0x5c/0x100 [c000000003633c10] [c00000000206a2e8] dax_pmem_init+0x34/0x48 [c000000003633c30] [c000000000132d0] do_one_initcall+0x60/0x320 [c000000003633d00] [c0000000020051b0] kernel_init_freeable+0x360/0x400 [c000000003633de0] [c00000000013764] kernel_init+0x34/0x1d0 [c000000003633e50] [c0000000000de14] ret_from_kernel_thread+0x5c/0x64
CVE-2023-53708	In the Linux kernel, the following vulnerability has been resolved: ACPI: x86: s2idle: Catch multiple <code>ACPI_TYPE_PACKAGE</code> objects If a badly constructed firmware includes multiple <code>`ACPI_TYPE_PACKAGE`</code> objects while evaluating the AMD LPS0 <code>_DSM</code> , there will be a memory leak. Explicitly guard against this.

CVE ID	Description
CVE-2023-53709	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Handle race between rb_move_tail and rb_check_pages It seems a data race between ring_buffer writing and integrity check. That is, RB_FLAG of head_page is been updating, while at same time RB_FLAG was cleared when doing integrity check rb_check_pages(): rb_check_pages() rb_handle_head_page(): ----- <pre> ----- rb_head_page_deactivate() rb_head_page_set_normal() rb_head_page_activate() We do integrity test of the list to check if the list is corrupted and it is still worth doing it. So, let's refactor rb_check_pages() such that we no longer clear and set flag during the list sanity checking. [1] and [2] are the test to reproduce and the crash report respectively. 1: `` read_trace.sh while true; do # the "trace" file is closed after read head -1 /sys/kernel/tracing/trace > /dev/null done `` `` repro.sh sysctl -w kernel.panic_on_warn=1 # function tracer will writing enough data into ring_buffer echo function > /sys/kernel/tracing/current_tracer ./read_trace.sh & ./read_trace.sh & ./read_trace.sh & ./read_trace.sh & ./read_trace.sh & ./read_trace.sh & ./read_trace.sh & `` 2: -----[cut here]----- WARNING: CPU: 9 PID: 62 at kernel/trace/ring_buffer.c:2653 rb_move_tail+0x450/0x470 Modules linked in: CPU: 9 PID: 62 Comm: ksoftirqd/9 Tainted: G W 6.2.0-rc6+ Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.15.0-0- g2dd4b9b3f840-prebuilt.qemu.org 04/01/2014 RIP: 0010:rb_move_tail+0x450/0x470 Code: ff ff 4c 89 c8 f0 4d 0f b1 02 48 89 c2 48 83 e2 fc 49 39 d0 75 24 83 e0 03 83 f8 02 0f 84 e1 fb ff ff 48 8b 57 10 f0 ff 42 08 <0f> 0b 83 f8 02 0f 84 ce fb ff ff e9 db RSP: 0018:ffff5564089bd00 EFLAGS: 00000203 RAX: 0000000000000000 RBX: ffff9db385a2bf81 RCX: ffff5564089bd18 RDX: ffff9db281110100 RSI: 00000000000000fe4 RDI: ffff9db380145400 RBP: ffff9db385a2bf80 R08: ffff9db385a2bfc0 R09: ffff9db385a2bfc2 R10: ffff9db385a6c000 R11: ffff9db385a2bf80 R12: 0000000000000000 R13: 000000000000003e8 R14: ffff9db281110100 R15: ffffffffbb006108 FS: 0000000000000000(0000) GS:ffff9db3bdcc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005602323024c8 CR3: 0000000022e0c000 CR4: 000000000000006e0 Call Trace: <TASK> ring_buffer_lock_reserve+0x136/0x360 ? __do_softirq +0x287/0x2df ? __pfx_rcu_softirq_qs+0x10/0x10 trace_function+0x21/0x110 ? __pfx_rcu_softirq_qs +0x10/0x10 ? __do_softirq+0x287/0x2df function_trace_call+0xf6/0x120 0xfffffffcc038f097 ? rcu_softirq_qs+0x5/0x140 rcu_softirq_qs+0x5/0x140 __do_softirq+0x287/0x2df run_ksoftirqd +0x2a/0x30 smpboot_thread_fn+0x188/0x220 ? __pfx_smpboot_thread_fn+0x10/0x10 kthread +0xe7/0x110 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2c/0x50 </TASK> ---[end trace 0000000000000000]--- [crash report and test reproducer credit goes to Zheng Yejian]</pre>
CVE-2023-53710	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921: fix error code of return in mt7921_acpi_read Kernel NULL pointer dereference when ACPI SAR table isn't implemented well. Fix the error code of return to mark the ACPI SAR table as invalid. <pre> [5.077128] mt7921e 0000:06:00.0: sar cnt = 0 [5.077381] BUG: kernel NULL pointer dereference, address: 0000000000000004 [5.077630] #PF: supervisor read access in kernel mode [5.077883] #PF: error_code(0x0000) - not-present page [5.078138] PGD 0 P4D 0 [5.078398] Oops: 0000 [#1] PREEMPT SMP NOPTI [5.079202] RIP: 0010:mt7921_init_acpi_sar+0x106/0x220 [mt7921_common] ... [5.080786] Call Trace: [5.080786] <TASK> [5.080786] mt7921_register_device +0x37d/0x490 [mt7921_common] [5.080786] mt7921_pci_probe.part.0+0x2ee/0x310 [mt7921e] [5.080786] mt7921_pci_probe+0x52/0x70 [mt7921e] [5.080786] local_pci_probe+0x47/0x90 [5.080786] pci_call_probe+0x55/0x190 [5.080786] pci_device_probe+0x84/0x120</pre>
CVE-2023-53711	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix a potential data corruption We must ensure that the subrequests are joined back into the head before we can retransmit a request. If the head was not on the commit lists, because the server wrote it synchronously, we still need to add it back to the retransmission list. Add a call that mirrors the effect of nfs_cancel_remove_inode() for O_DIRECT.
CVE-2023-53712	In the Linux kernel, the following vulnerability has been resolved: ARM: 9317/1: kexec: Make smp stop calls asynchronous If a panic is triggered by a hrtimer interrupt all online cpus will be notified and set offline. But as highlighted by commit 19dbdc8b8039c ("smp: Warn on function calls from softirq context") this call should not be made synchronous with disabled interrupts: softdog: Initiating panic Kernel panic - not syncing: Software Watchdog Timer expired WARNING: CPU: 1 PID: 0 at kernel/smp.c:753 smp_call_function_many_cond unwind_backtrace: show_stack dump_stack_lvl __warn warn_slowpath_fmt smp_call_function_many_cond smp_call_function_crash_smp_send_stop.part.0 machine_crash_shutdown __crash_kexec panic softdog_fire __hrtimer_run_queues hrtimer_interrupt Make the smp call for machine_crash_nonpanic_core() asynchronous.

CVE ID	Description
CVE-2023-53713	In the Linux kernel, the following vulnerability has been resolved: arm64: sme: Use STR P to clear FFR context field in streaming SVE mode The FFR is a predicate register which can vary between 16 and 256 bits in size depending upon the configured vector length. When saving the SVE state in streaming SVE mode, the FFR register is inaccessible and so commit 9f5848665788 ("arm64/sve: Make access to FFR optional") simply clears the FFR field of the in-memory context structure. Unfortunately, it achieves this using an unconditional 8-byte store and so if the SME vector length is anything other than 64 bytes in size we will either fail to clear the entire field or, worse, we will corrupt memory immediately following the structure. This has led to intermittent kfence splats in CI [1] and can trigger kmalloc Redzone corruption messages when running the 'fp-stress' kselftest: ===== BUG kmalloc-1k (Not tainted): kmalloc Redzone overwritten ----- 0xfffff000809bf1e22-0xfffff000809bf1e27 @offset=7714. First byte 0x0 instead of 0xcc Allocated in do_sme_acc+0x9c/0x220 age=2613 cpu=1 pid=531 __kmalloc+0x8c/0xcc do_sme_acc+0x9c/0x220 ... Replace the 8-byte store with a store of a predicate register which has been zero-initialised with PFALSE, ensuring that the entire field is cleared in memory. [1] https://lore.kernel.org/r/CA+G9fYtU7HsV0R0dp4XEH5xXHSJFw8KyDf5VQrLLfMxWfxQkag@mail.gmail.com
CVE-2023-53715	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: cfg80211: Pass the PMK in binary instead of hex Apparently the hex passphrase mechanism does not work on newer chips/firmware (e.g. BCM4387). It seems there was a simple way of passing it in binary all along, so use that and avoid the hexification. OpenBSD has been doing it like this from the beginning, so this should work on all chips. Also clear the structure before setting the PMK. This was leaking uninitialized stack contents to the device.
CVE-2023-53716	In the Linux kernel, the following vulnerability has been resolved: net: fix skb leak in __skb_tstamp_tx() Commit 50749f2dd685 ("tcp/udp: Fix memleaks of sk and zerocopy skbs with TX timestamp.") added a call to skb_orphan_frags_rx() to fix leaks with zerocopy skbs. But it ended up adding a leak of its own. When skb_orphan_frags_rx() fails, the function just returns, leaking the skb it just cloned. Free it before returning. This bug was discovered and resolved using Coverity Static Analysis Security Testing (SAST) by Synopsys, Inc.
CVE-2023-53717	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k: Fix potential stack-out-of-bounds write in ath9k_wmi_rsp_callback() Fix a stack-out-of-bounds write that occurs in a WMI response callback function that is called after a timeout occurs in ath9k_wmi_cmd(). The callback writes to wmi->cmd_rsp_buf, a stack-allocated buffer that could no longer be valid when a timeout occurs. Set wmi->last_seq_id to 0 when a timeout occurred. Found by a modified version of syzkaller. BUG: KASAN: stack-out-of-bounds in ath9k_wmi_ctrl_rx Write of size 4 Call Trace: memcpy ath9k_wmi_ctrl_rx ath9k_hic_rx_msg ath9k_hif_usb_reg_in_cb __usb_hcd_giveback_urb usb_hcd_giveback_urb dummy_timer call_timer_fn run_timer_softirq __do_softirq irq_exit_rcu sysvec_apic_timer_interrupt

CVE ID	Description
CVE-2023-53718	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Do not swap cpu_buffer during resize process When ring_buffer_swap_cpu was called during resize process, the cpu buffer was swapped in the middle, resulting in incorrect state. Continuing to run in the wrong state will result in oops. This issue can be easily reproduced using the following two scripts: /tmp # cat test1.sh /#!/ /bin/sh for i in `seq 0 100000` do echo 2000 > /sys/kernel/debug/tracing/buffer_size_kb sleep 0.5 echo 5000 > /sys/kernel/debug/tracing/buffer_size_kb sleep 0.5 done /tmp # cat test2.sh /#!/ /bin/sh for i in `seq 0 100000` do echo irqsoff > /sys/kernel/debug/tracing/current_tracer sleep 1 echo nop > /sys/kernel/debug/tracing/current_tracer sleep 1 done /tmp # ./test1.sh & /tmp # ./test2.sh & A typical oops log is as follows, sometimes with other different oops logs. [231.711293] WARNING: CPU: 0 PID: 9 at kernel/trace/ring_buffer.c:2026 rb_update_pages+0x378/0x3f8 [231.713375] Modules linked in: [231.714735] CPU: 0 PID: 9 Comm: kworker/0:1 Tainted: G W 6.5.0-rc1-00276-g20edcec23f92 #15 [231.716750] Hardware name: linux,dummy-virt (DT) [231.718152] Workqueue: events update_pages_handler [231.719714] pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPED=) [231.721171] pc : rb_update_pages+0x378/0x3f8 [231.722212] lr : rb_update_pages+0x25c/0x3f8 [231.723248] sp : ffff800082b9bd50 [231.724169] x29: ffff800082b9bd50 x28: ffff8000825f7000 x27: 0000000000000000 [231.726102] x26: 0000000000000001 x25: ffffffff00000000 x24: 00000000000000ff0 [231.728122] x23: ffff0000c3a0b600 x22: ffff0000c3a0b5c0 x21: ffffffff00000000 [231.730203] x20: ffff0000c3a0b600 x19: ffff0000c0102400 x18: 0000000000000000 [231.732329] x17: 0000000000000000 x16: 0000000000000000 x15: 0000ffff7aa8510 [231.734212] x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000002 [231.736291] x11: ffff8000826998a8 x10: ffff800082b9baf0 x9 : ffff800081137558 [231.738195] x8 : fffffc00030e82c8 x7 : 0000000000000000 x6 : 0000000000000001 [231.740192] x5 : ffff0000fbaf00 x4 : 0000000000000000 x3 : 0000000000000000 [231.742118] x2 : 000000000000006aa x1 : 0000000000000001 x0 : ffff0000c0007208 [231.744196] Call trace: [231.744892] rb_update_pages+0x378/0x3f8 [231.745893] update_pages_handler+0x1c/0x38 [231.746893] process_one_work+0x1f0/0x468 [231.747852] worker_thread+0x54/0x410 [231.748737] kthread+0x124/0x138 [231.749549] ret_from_fork+0x10/0x20 [231.750434] ---[end trace 0000000000000000]--- [233.720486] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 [233.721696] Mem abort info: [233.721935] ESR = 0x0000000096000004 [233.722283] EC = 0x25: DABT (current EL), IL = 32 bits [233.722596] SET = 0, FnV = 0 [233.722805] EA = 0, S1PTW = 0 [233.723026] FSC = 0x04: level 0 translation fault [233.723458] Data abort info: [233.723734] ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 [233.724176] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [233.724589] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [233.725075] user pgtable: 4k pages, 48-bit VAs, pgdp=0000000104943000 [233.725592] [0000000000000000] pgd=0000000000000000, p4d=0000000000000000 [233.726231] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [233.726720] Modules linked in: [233.727007] CPU: 0 PID: 9 Comm: kworker/0:1 Tainted: G W 6.5.0-rc1-00276-g20edcec23f92 #15 [233.727777] Hardware name: linux,dummy-virt (DT) [233.728225] Workqueue: events update_pages_handler [233.728655] pstate: 200000c5 (nzCv daIF -PAN -UAO -TCO -DIT -SSBS BTYPED=) [233.729054] pc : rb_update_pages+0x1a8/0x3f8 [233.729334] lr : rb_update_pages+0x154/0x3f8 [233.729592] sp : ffff800082b9bd50 [233.729792] x29: ffff800082b9bd50 x28: ffff8000825f7000 x27: 00000000 ---truncated---
CVE-2023-53719	In the Linux kernel, the following vulnerability has been resolved: serial: arc_uart: fix of_iomap leak in `arc_serial_probe` Smatch reports: drivers/tty/serial/arc_uart.c:631 arc_serial_probe() warn: 'port->membase' from of_iomap() not released on lines: 631. In arc_serial_probe(), if uart_add_one_port() fails, port->membase is not released, which would cause a resource leak. To fix this, I replace of_iomap with devm_platform_ioremap_resource.
CVE-2023-53720	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Release the label when replacing existing ct entry Cited commit doesn't release the label mapping when replacing existing ct entry which leads to following memleak report: unreferenced object 0xffff8881854cf280 (size 96): comm "kworker/u48:74", pid 23093, jiffies 4296664564 (age 175.944s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace: [<000000002722d368>] __kmalloc+0x4b/0x1c0 [<00000000cc44e18f>] mapping_add+0x6e8/0xc90 [mlx5_core] [<000000003ad942a7>] mlx5_get_label_mapping+0x66/0xe0 [mlx5_core] [<00000000266308ac>] mlx5_tc_ct_entry_create_mod_hdr+0x1c4/0xf50 [mlx5_core] [<000000009a768b4f>] mlx5_tc_ct_entry_add_rule+0x16f/0xaf0 [mlx5_core] [<00000000a178f3e5>] mlx5_tc_ct_block_flow_offload_add+0x10cb/0x1f90 [mlx5_core] [<000000007b46c496>] mlx5_tc_ct_block_flow_offload+0x14a/0x630 [mlx5_core] [<00000000a9a18ac5>] nf_flow_offload_tuple+0x1a3/0x390 [nf_flow_table] [<00000000d0881951>] flow_offload_work_handler+0x257/0xd30 [nf_flow_table] [<000000009e4935a4>] process_one_work+0x7c2/0x13e0 [<00000000f5cd36a7>] worker_thread+0x59d/0xec0 [<00000000baed1daf>] kthread+0x28f/0x330 [<0000000063d282a4>] ret_from_fork+0x1f/0x30 Fix the issue by correctly releasing the label mapping.

CVE ID	Description
CVE-2023-53722	In the Linux kernel, the following vulnerability has been resolved: md: raid1: fix potential OOB in raid1_remove_disk() If rddev->raid_disk is greater than mddev->raid_disks, there will be an out-of-bounds in raid1_remove_disk(). We have already found similar reports as follows: 1) commit d17f744e883b ("md-raid10: fix KASAN warning") 2) commit 1ebc2cec0b7d ("dm raid: fix KASAN warning in raid5_remove_disk") Fix this bug by checking whether the "number" variable is valid.
CVE-2023-53726	In the Linux kernel, the following vulnerability has been resolved: arm64: csum: Fix OoB access in IP checksum code for negative lengths Although commit c2c24edb1d9c ("arm64: csum: Fix pathological zero-length calls") added an early return for zero-length input, syzkaller has popped up with an example of a _negative_length which causes an undefined shift and an out-of-bounds read: BUG: KASAN: slab-out-of-bounds in do_csum+0x44/0x254 arch/arm64/lib/csum.c:39 Read of size 4294966928 at addr ffff0000d7ac0170 by task syz-executor412/5975 CPU: 0 PID: 5975 Comm: syz-executor412 Not tainted 6.4.0-rc4-syzkaller-g908f31f2a05b #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/25/2023 Call trace: dump_backtrace+0x1b8/0x1e4 arch/arm64/kernel/stacktrace.c:233 show_stack+0x2c/0x44 arch/arm64/kernel/stacktrace.c:240 __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xd0/0x124 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:351 [inline] print_report+0x174/0x514 mm/kasan/report.c:462 kasan_report+0xd4/0x130 mm/kasan/report.c:572 kasan_check_range+0x264/0x2a4 mm/kasan/generic.c:187 __kasan_check_read+0x20/0x30 mm/kasan/shadow.c:31 do_csum+0x44/0x254 arch/arm64/lib/csum.c:39 csum_partial+0x30/0x58 lib/checksum.c:128 gso_make_checksum include/linux/skbuff.h:4928 [inline] __udp_gso_segment+0xaf4/0x1bc4 net/ipv4/udp_offload.c:332 udp6_ufo_fragment+0x540/0xca0 net/ipv6/udp_offload.c:47 ipv6_gso_segment+0x5cc/0x1760 net/ipv6/ipv6_offload.c:119 skb_mac_gso_segment+0x2b4/0x5b0 net/core/gro.c:141 __skb_gso_segment+0x250/0x3d0 net/core/dev.c:3401 skb_gso_segment include/linux/netdevice.h:4859 [inline] validate_xmit_skb+0x364/0xdb0 net/core/dev.c:3659 validate_xmit_skb_list+0x94/0x130 net/core/dev.c:3709 sch_direct_xmit+0xe8/0x548 net/sched/sch_generic.c:327 __dev_xmit_skb net/core/dev.c:3805 [inline] __dev_queue_xmit+0x147c/0x3318 net/core/dev.c:4210 dev_queue_xmit include/linux/netdevice.h:3085 [inline] packet_xmit+0x6c/0x318 net/packet/af_packet.c:276 packet_snd net/packet/af_packet.c:3081 [inline] packet_sendmsg+0x376c/0x4c98 net/packet/af_packet.c:3113 sock_sendmsg_nosec net/socket.c:724 [inline] sock_sendmsg net/socket.c:747 [inline] __sys_sendto+0x3b4/0x538 net/socket.c:2144 Extend the early return to reject negative lengths as well, aligning our implementation with the generic code in lib/checksum.c
CVE-2023-53728	In the Linux kernel, the following vulnerability has been resolved: posix-timers: Ensure timer ID search-loop limit is valid posix_timer_add() tries to allocate a posix timer ID by starting from the cached ID which was stored by the last successful allocation. This is done in a loop searching the ID space for a free slot one by one. The loop has to terminate when the search wrapped around to the starting point. But that's racy vs. establishing the starting point. That is read out lockless, which leads to the following problem: CPU0 CPU1 posix_timer_add() start = sig->posix_timer_id; lock(hash_lock); ... posix_timer_add() if (++sig->posix_timer_id < 0) start = sig->posix_timer_id; sig->posix_timer_id = 0; So CPU1 can observe a negative start value, i.e. -1, and the loop break never happens because the condition can never be true: if (sig->posix_timer_id == start) break; While this is unlikely to ever turn into an endless loop as the ID space is huge (INT_MAX), the racy read of the start value caught the attention of KCSAN and Dmitry unearthed that incorrectness. Rewrite it so that all id operations are under the hash lock.
CVE-2023-53729	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: qmi_encdec: Restrict string length in decode The QMI TLV value for strings in a lot of qmi element info structures account for null terminated strings with MAX_LEN + 1. If a string is actually MAX_LEN + 1 length, this will cause an out of bounds access when the NULL character is appended in decoding.
CVE-2023-53731	In the Linux kernel, the following vulnerability has been resolved: netlink: fix potential deadlock in netlink_set_err() syzbot reported a possible deadlock in netlink_set_err() [1] A similar issue was fixed in commit 1d482e666b8e ("netlink: disable IRQs for netlink_lock_table()") in netlink_lock_table() This patch adds IRQ safety to netlink_set_err() and __netlink_diag_dump() which were not covered by cited commit. [1] WARNING: possible irq lock inversion dependency detected 6.4.0-rc6-syzkaller-00240-g4e9f0ec38852 #0 Not tainted syz-executor.2/23011 just changed the state of lock: ffffffff8e1a7a58 (nl_table_lock){+.?-}{2:2}, at: netlink_set_err+0x2e/0x3a0 net/netlink/af_netlink.c:1612 but this lock was taken by another, SOFTIRQ-safe lock in the past: (&local->queue_stop_reason_lock){...-}{2:2} and interrupts could create inverse lock ordering between them. other info that might help us debug this: Possible interrupt unsafe locking scenario: CPU0 CPU1 ---- lock(nl_table_lock); local_irq_disable(); lock(&local->queue_stop_reason_lock); lock(nl_table_lock); <Interrupt> lock(&local->queue_stop_reason_lock); *** DEADLOCK ***
CVE-2023-5441	NULL Pointer Dereference in GitHub repository vim/vim prior to 20d161ace307e28690229b68584f2d84556f8960.
CVE-2023-5535	Use After Free in GitHub repository vim/vim prior to v9.0.2010.

CVE ID	Description
CVE-2023-5574	A use-after-free flaw was found in xorg-x11-server-Xvfb. This issue occurs in Xvfb with a very specific and legacy configuration (a multi-screen setup with multiple protocol screens, also known as Zaphod mode). If the pointer is warped from a screen 1 to a screen 0, a use-after-free issue may be triggered during shutdown or reset of the Xvfb server, allowing for possible escalation of privileges or denial of service.
CVE-2023-5752	When installing a package from a Mercurial VCS URL (ie "pip install hg+...") with pip prior to v23.3, the specified Mercurial revision could be used to inject arbitrary configuration options to the "hg clone" call (ie "--config"). Controlling the Mercurial configuration can modify how and which repository is installed. This vulnerability does not affect users who aren't installing from Mercurial.
CVE-2023-5868	A memory disclosure vulnerability was found in PostgreSQL that allows remote users to access sensitive information by exploiting certain aggregate function calls with 'unknown'-type arguments. Handling 'unknown'-type values from string literals without type designation can disclose bytes, potentially revealing notable and confidential information. This issue exists due to excessive data output in aggregate function calls, enabling remote users to read some portion of system memory.
CVE-2023-5870	A flaw was found in PostgreSQL involving the pg_cancel_backend role that signals background workers, including the logical replication launcher, autovacuum workers, and the autovacuum launcher. Successful exploitation requires a non-core extension with a less-resilient background worker and would affect that specific background worker only. This issue may allow a remote high privileged user to launch a denial of service (DoS) attack.
CVE-2023-5981	A vulnerability was found that the response times to malformed ciphertexts in RSA-PSK ClientKeyExchange differ from response times of ciphertexts with correct PKCS#1 v1.5 padding.
CVE-2023-6238	A buffer overflow vulnerability was found in the NVM Express (NVMe) driver in the Linux kernel. Only privileged user could specify a small meta buffer and let the device perform larger Direct Memory Access (DMA) into the same buffer, overwriting unrelated kernel memory, causing random kernel crashes and memory corruption.
CVE-2023-6277	An out-of-memory flaw was found in libtiff. Passing a crafted tiff file to TIFFOpen() API may allow a remote attacker to cause a denial of service via a craft input with size smaller than 379 KB.
CVE-2023-6601	A flaw was found in FFmpeg's HLS demuxer. This vulnerability allows bypassing unsafe file extension checks and triggering arbitrary demuxers via base64-encoded data URIs appended with specific file extensions.
CVE-2023-6602	A flaw was found in FFmpeg's TTY Demuxer. This vulnerability allows possible data exfiltration via improper parsing of non-TTY-compliant input files in HLS playlists.
CVE-2023-6603	A flaw was found in FFmpeg's HLS playlist parsing. This vulnerability allows a denial of service via a maliciously crafted HLS playlist that triggers a null pointer dereference during initialization.
CVE-2023-6604	A flaw was found in FFmpeg. This vulnerability allows unexpected additional CPU load and storage consumption, potentially leading to degraded performance or denial of service via the demuxing of arbitrary data as XBIN-formatted data without proper format validation.
CVE-2023-6605	A flaw was found in FFmpeg's DASH playlist support. This vulnerability allows arbitrary HTTP GET requests to be made on behalf of the machine running FFmpeg via a crafted DASH playlist containing malicious URLs.
CVE-2023-7042	A null pointer dereference vulnerability was found in ath10k_wmi_tlv_op_pull_mgmt_tx_compl_ev() in drivers/net/wireless/ath/ath10k/wmi-tlv.c in the Linux kernel. This issue could be exploited to trigger a denial of service.
CVE-2023-7216	A path traversal vulnerability was found in the CPIO utility. This issue could allow a remote unauthenticated attacker to trick a user into opening a specially crafted archive. During the extraction process, the archiver could follow symlinks outside of the intended directory, which allows files to be written in arbitrary directories through symlinks.
CVE-2024-0232	A heap use-after-free issue has been identified in SQLite in the jsonParseAddNodeArray() function in sqlite3.c. This flaw allows a local attacker to leverage a victim to pass specially crafted malicious input to the application, potentially causing a crash and leading to a denial of service.
CVE-2024-0397	A defect was discovered in the Python "ssl" module where there is a memory race condition with the ssl.SSLContext methods "cert_store_stats()" and "get_ca_certs()". The race condition can be triggered if the methods are called at the same time as certificates are loaded into the SSLContext, such as during the TLS handshake with a certificate directory configured. This issue is fixed in CPython 3.10.14, 3.11.9, 3.12.3, and 3.13.0a5.

CVE ID	Description
CVE-2024-0553	A vulnerability was found in GnuTLS. The response times to malformed ciphertexts in RSA-PSK ClientKeyExchange differ from the response times of ciphertexts with correct PKCS#1 v1.5 padding. This issue may allow a remote attacker to perform a timing side-channel attack in the RSA-PSK key exchange, potentially leading to the leakage of sensitive data. CVE-2024-0553 is designated as an incomplete resolution for CVE-2023-5981.
CVE-2024-0562	A use-after-free flaw was found in the Linux Kernel. When a disk is removed, bdi_unregister is called to stop further write-back and waits for associated delayed work to complete. However, wb_inode_writeback_end() may schedule bandwidth estimation work after this has completed, which can result in the timer attempting to access the recently freed bdi_writeback.
CVE-2024-0564	A flaw was found in the Linux kernel's memory deduplication mechanism. The max page sharing of Kernel Samepage Merging (KSM), added in Linux kernel version 4.4.0-96.119, can create a side channel. When the attacker and the victim share the same host and the default setting of KSM is "max page sharing=256", it is possible for the attacker to time the unmap to merge with the victim's page. The unmapping time depends on whether it merges with the victim's page and additional physical pages are created beyond the KSM's "max page share". Through these operations, the attacker can leak the victim's page.
CVE-2024-0567	A vulnerability was found in GnuTLS, where a cockpit (which uses gnuTLS) rejects a certificate chain with distributed trust. This issue occurs when validating a certificate chain with cockpit-certificate-ensure. This flaw allows an unauthenticated, remote client or attacker to initiate a denial of service attack.
CVE-2024-0607	A flaw was found in the Netfilter subsystem in the Linux kernel. The issue is in the nft_byteorder_eval() function, where the code iterates through a loop and writes to the `dst` array. On each iteration, 8 bytes are written, but `dst` is an array of u32, so each element only has space for 4 bytes. That means every iteration overwrites part of the previous element corrupting this array of u32. This flaw allows a local user to cause a denial of service or potentially break NetFilter functionality.
CVE-2024-0727	Issue summary: Processing a maliciously formatted PKCS12 file may lead OpenSSL to crash leading to a potential Denial of Service attack Impact summary: Applications loading files in the PKCS12 format from untrusted sources might terminate abruptly. A file in PKCS12 format can contain certificates and keys and may come from an untrusted source. The PKCS12 specification allows certain fields to be NULL, but OpenSSL does not correctly check for this case. This can lead to a NULL pointer dereference that results in OpenSSL crashing. If an application processes PKCS12 files from an untrusted source using the OpenSSL APIs then that application will be vulnerable to this issue. OpenSSL APIs that are vulnerable to this are: PKCS12_parse(), PKCS12_unpack_p7data(), PKCS12_unpack_p7encdata(), PKCS12_unpack_authsafes() and PKCS12_newpass(). We have also fixed a similar issue in SMIME_write_PKCS7(). However since this function is related to writing data we do not consider it security significant. The FIPS modules in 3.2, 3.1 and 3.0 are not affected by this issue.
CVE-2024-10041	A vulnerability was found in PAM. The secret information is stored in memory, where the attacker can trigger the victim program to execute by sending characters to its standard input (stdin). As this occurs, the attacker can train the branch predictor to execute an ROP chain speculatively. This flaw could result in leaked passwords, such as those found in /etc/shadow while performing authentications.
CVE-2024-10220	The Kubernetes kubelet component allows arbitrary command execution via specially crafted gitRepo volumes. This issue affects kubelet: through 1.28.11, from 1.29.0 through 1.29.6, from 1.30.0 through 1.30.2.
CVE-2024-10524	Applications that use Wget to access a remote resource using shorthand URLs and pass arbitrary user credentials in the URL are vulnerable. In these cases attackers can enter crafted credentials which will cause Wget to access an arbitrary host.
CVE-2024-10963	A flaw was found in pam_access, where certain rules in its configuration file are mistakenly treated as hostnames. This vulnerability allows attackers to trick the system by pretending to be a trusted hostname, gaining unauthorized access. This issue poses a risk for systems that rely on this feature to control who can access certain services or terminals.

CVE ID	Description
CVE-2024-10976	Incomplete tracking in PostgreSQL of tables with row security allows a reused query to view or change different rows from those intended. CVE-2023-2455 and CVE-2016-2193 fixed most interaction between row security and user ID changes. They missed cases where a subquery, WITH query, security invoker view, or SQL-language function references a table with a row-level security policy. This has the same consequences as the two earlier CVEs. That is to say, it leads to potentially incorrect policies being applied in cases where role-specific policies are used and a given query is planned under one role and then executed under other roles. This scenario can happen under security definer functions or when a common user and query is planned initially and then re-used across multiple SET ROLES. Applying an incorrect policy may permit a user to complete otherwise-forbidden reads and modifications. This affects only databases that have used CREATE POLICY to define a row security policy. An attacker must tailor an attack to a particular application's pattern of query plan reuse, user ID changes, and role-specific row security policies. Versions before PostgreSQL 17.1, 16.5, 15.9, 14.14, 13.17, and 12.21 are affected.
CVE-2024-10977	Client use of server error message in PostgreSQL allows a server not trusted under current SSL or GSS settings to furnish arbitrary non-NUL bytes to the libpq application. For example, a man-in-the-middle attacker could send a long error message that a human or screen-scraper user of psql mistakes for valid query results. This is probably not a concern for clients where the user interface unambiguously indicates the boundary between one error message and other text. Versions before PostgreSQL 17.1, 16.5, 15.9, 14.14, 13.17, and 12.21 are affected.
CVE-2024-10978	Incorrect privilege assignment in PostgreSQL allows a less-privileged application user to view or change different rows from those intended. An attack requires the application to use SET ROLE, SET SESSION AUTHORIZATION, or an equivalent feature. The problem arises when an application query uses parameters from the attacker or conveys query results to the attacker. If that query reacts to current_setting('role') or the current user ID, it may modify or return data as though the session had not used SET ROLE or SET SESSION AUTHORIZATION. The attacker does not control which incorrect user ID applies. Query text from less-privileged sources is not a concern here, because SET ROLE and SET SESSION AUTHORIZATION are not sandboxes for unvetted queries. Versions before PostgreSQL 17.1, 16.5, 15.9, 14.14, 13.17, and 12.21 are affected.
CVE-2024-10979	Incorrect control of environment variables in PostgreSQL PL/Perl allows an unprivileged database user to change sensitive process environment variables (e.g. PATH). That often suffices to enable arbitrary code execution, even if the attacker lacks a database server operating system user. Versions before PostgreSQL 17.1, 16.5, 15.9, 14.14, 13.17, and 12.21 are affected.
CVE-2024-11041	vllm-project vllm version v0.6.2 contains a vulnerability in the MessageQueue.dequeue() API function. The function uses pickle.loads to parse received sockets directly, leading to a remote code execution vulnerability. An attacker can exploit this by sending a malicious payload to the MessageQueue, causing the victim's machine to execute arbitrary code.
CVE-2024-11053	When asked to both use a `.netrc` file for credentials and to follow HTTP redirects, curl could leak the password used for the first host to the followed-to host under certain circumstances. This flaw only manifests itself if the netrc file has an entry that matches the redirect target hostname but the entry either omits just the password or omits both login and password.
CVE-2024-11392	Hugging Face Transformers MobileViTV2 Deserialization of Untrusted Data Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the handling of configuration files. The issue results from the lack of proper validation of user-supplied data, which can result in deserialization of untrusted data. An attacker can leverage this vulnerability to execute code in the context of the current user. Was ZDI-CAN-24322.
CVE-2024-11393	Hugging Face Transformers MaskFormer Model Deserialization of Untrusted Data Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the parsing of model files. The issue results from the lack of proper validation of user-supplied data, which can result in deserialization of untrusted data. An attacker can leverage this vulnerability to execute code in the context of the current user. Was ZDI-CAN-25191.
CVE-2024-11394	Hugging Face Transformers Trax Model Deserialization of Untrusted Data Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the handling of model files. The issue results from the lack of proper validation of user-supplied data, which can result in deserialization of untrusted data. An attacker can leverage this vulnerability to execute code in the context of the current user. Was ZDI-CAN-25012.

CVE ID	Description
CVE-2024-11403	There exists an out of bounds read/write in LibJXL versions prior to commit 9cc451b91b74ba470fd72bd48c121e9f33d24c99. The JPEG decoder used by the JPEG XL encoder when doing JPEG recompression (i.e. if using JxlEncoderAddJPEGFrame on untrusted input) does not properly check bounds in the presence of incomplete codes. This could lead to an out-of-bounds write. In jpegli which is released as part of the same project, the same vulnerability is present. However, the relevant buffer is part of a bigger structure, and the code makes no assumptions on the values that could be overwritten. The issue could however cause jpegli to read uninitialised memory, or addresses of functions.
CVE-2024-11498	There exists a stack buffer overflow in libjxl. A specifically-crafted file can cause the JPEG XL decoder to use large amounts of stack space (up to 256mb is possible, maybe 512mb), potentially exhausting the stack. An attacker can craft a file that will cause excessive memory usage. We recommend upgrading past commit 65fbc56bc578b6b6ee02a527be70787bbd053b0.
CVE-2024-1151	A vulnerability was reported in the Open vSwitch sub-component in the Linux Kernel. The flaw occurs when a recursive operation of code push recursively calls into the code block. The OVS module does not validate the stack depth, pushing too many frames and causing a stack overflow. As a result, this can lead to a crash or other related issues.
CVE-2024-12086	A flaw was found in rsync. It could allow a server to enumerate the contents of an arbitrary file from the client's machine. This issue occurs when files are being copied from a client to a server. During this process, the rsync server will send checksums of local data to the client to compare with in order to determine what data needs to be sent to the server. By sending specially constructed checksum values for arbitrary files, an attacker may be able to reconstruct the data of those files byte-by-byte based on the responses from the client.
CVE-2024-12254	Starting in Python 3.12.0, the <code>asyncio.SelectorSocketTransport.writelines()</code> method would not "pause" writing and signal to the Protocol to drain the buffer to the wire once the write buffer reached the "high-water mark". Because of this, Protocols would not periodically drain the write buffer potentially leading to memory exhaustion. This vulnerability likely impacts a small number of users, you must be using Python 3.12.0 or later, on macOS or Linux, using the <code>asyncio</code> module with protocols, and using <code>.writelines()</code> method which had new zero-copy-on-write behavior in Python 3.12.0 and later. If not all of these factors are true then your usage of Python is unaffected.
CVE-2024-12539	An issue was discovered where improper authorization controls affected certain queries that could allow a malicious actor to circumvent Document Level Security in Elasticsearch and get access to documents that their roles would normally not allow.
CVE-2024-12718	Allows modifying some file metadata (e.g. last modified) with <code>filter="data"</code> or file permissions (<code>chmod</code>) with <code>filter="tar"</code> of files outside the extraction directory. You are affected by this vulnerability if using the <code>tarfile</code> module to extract untrusted tar archives using <code>TarFile.extractall()</code> or <code>TarFile.extract()</code> using the <code>filter=</code> parameter with a value of "data" or "tar". See the <code>tarfile</code> extraction filters documentation https://docs.python.org/3/library/tarfile.html#tarfile-extraction-filter for more information. Only Python versions 3.12 or later are affected by these vulnerabilities, earlier versions don't include the extraction filter feature. Note that for Python 3.14 or later the default value of <code>filter=</code> changed from "no filtering" to "data", so if you are relying on this new default behavior then your usage is also affected. Note that none of these vulnerabilities significantly affect the installation of source distributions which are tar archives as source distributions already allow arbitrary code execution during the build process. However when evaluating source distributions it's important to avoid installing source distributions with suspicious links.
CVE-2024-12720	A Regular Expression Denial of Service (ReDoS) vulnerability was identified in the <code>huggingface/transformers</code> library, specifically in the <code>file_tokenization_nougat_fast.py</code> . The vulnerability occurs in the <code>post_process_single()</code> function, where a regular expression processes specially crafted input. The issue stems from the regex exhibiting exponential time complexity under certain conditions, leading to excessive backtracking. This can result in significantly high CPU usage and potential application downtime, effectively creating a Denial of Service (DoS) scenario. The affected version is v4.46.3 (latest).
CVE-2024-12745	A SQL injection in the Amazon Redshift Python Connector v2.1.4 allows a user to gain escalated privileges via the <code>get_schemas</code> , <code>get_tables</code> , or <code>get_columns</code> Metadata APIs. Users are recommended to upgrade to the driver version 2.1.5 or revert to driver version 2.1.3.

CVE ID	Description
CVE-2024-12797	Issue summary: Clients using RFC7250 Raw Public Keys (RPKs) to authenticate a server may fail to notice that the server was not authenticated, because handshakes don't abort as expected when the SSL_VERIFY_PEER verification mode is set. Impact summary: TLS and DTLS connections using raw public keys may be vulnerable to man-in-middle attacks when server authentication failure is not detected by clients. RPKs are disabled by default in both TLS clients and TLS servers. The issue only arises when TLS clients explicitly enable RPK use by the server, and the server, likewise, enables sending of an RPK instead of an X.509 certificate chain. The affected clients are those that then rely on the handshake to fail when the server's RPK fails to match one of the expected public keys, by setting the verification mode to SSL_VERIFY_PEER. Clients that enable server-side raw public keys can still find out that raw public key verification failed by calling SSL_get_verify_result(), and those that do, and take appropriate action, are not affected. This issue was introduced in the initial implementation of RPK support in OpenSSL 3.2. The FIPS modules in 3.4, 3.3, 3.2, 3.1 and 3.0 are not affected by this issue.
CVE-2024-12798	ACE vulnerability in JaninoEventEvaluator by QOS.CH logback-core upto including version 0.1 to 1.3.14 and 1.4.0 to 1.5.12 in Java applications allows attacker to execute arbitrary code by compromising an existing logback configuration file or by injecting an environment variable before program execution. Malicious logback configuration files can allow the attacker to execute arbitrary code using the JaninoEventEvaluator extension. A successful attack requires the user to have write access to a configuration file. Alternatively, the attacker could inject a malicious environment variable pointing to a malicious configuration file. In both cases, the attack requires existing privilege.
CVE-2024-12801	Server-Side Request Forgery (SSRF) in SaxEventRecorder by QOS.CH logback version 0.1 to 1.3.14 and 1.4.0 to 1.5.12 on the Java platform, allows an attacker to forge requests by compromising logback configuration files in XML. The attacks involves the modification of DOCTYPE declaration in XML configuration files.
CVE-2024-13009	In Eclipse Jetty versions 9.4.0 to 9.4.56 a buffer can be incorrectly released when confronted with a gzip error when inflating a request body. This can result in corrupted and/or inadvertent sharing of data between requests.
CVE-2024-13176	Issue summary: A timing side-channel which could potentially allow recovering the private key exists in the ECDSA signature computation. Impact summary: A timing side-channel in ECDSA signature computations could allow recovering the private key by an attacker. However, measuring the timing would require either local access to the signing application or a very fast network connection with low latency. There is a timing signal of around 300 nanoseconds when the top word of the inverted ECDSA nonce value is zero. This can happen with significant probability only for some of the supported elliptic curves. In particular the NIST P-521 curve is affected. To be able to measure this leak, the attacker process must either be located in the same physical computer or must have a very fast network connection with low latency. For that reason the severity of this vulnerability is Low. The FIPS modules in 3.4, 3.3, 3.2, 3.1 and 3.0 are affected by this issue.
CVE-2024-13978	A vulnerability was found in LibTIFF up to 4.7.0. It has been declared as problematic. Affected by this vulnerability is the function t2p_read_tiff_init of the file tools/tiff2pdf.c of the component fax2ps. The manipulation leads to null pointer dereference. The attack needs to be approached locally. The complexity of an attack is rather high. The exploitation appears to be difficult. The patch is named 2ebfffb0e8836bfb1cd7d85c059cd285c59761a4. It is recommended to apply a patch to fix this issue.
CVE-2024-1597	pgjdbc, the PostgreSQL JDBC Driver, allows attacker to inject SQL if using PreferQueryMode=SIMPLE. Note this is not the default. In the default mode there is no vulnerability. A placeholder for a numeric value must be immediately preceded by a minus. There must be a second placeholder for a string value after the first placeholder; both must be on the same line. By constructing a matching string payload, the attacker can inject SQL to alter the query, bypassing the protections that parameterized queries bring against SQL Injection attacks. Versions before 42.7.2, 42.6.1, 42.5.5, 42.4.4, 42.3.9, and 42.2.28 are affected.
CVE-2024-21538	Versions of the package cross-spawn before 6.0.6, from 7.0.0 and before 7.0.5 are vulnerable to Regular Expression Denial of Service (ReDoS) due to improper input sanitization. An attacker can increase the CPU usage and crash the program by crafting a very large and well crafted string.
CVE-2024-21634	Amazon Ion is a Java implementation of the Ion data notation. Prior to version 1.10.5, a potential denial-of-service issue exists in 'ion-java' for applications that use 'ion-java' to deserialize Ion text encoded data, or deserialize Ion text or binary encoded data into the 'IonValue' model and then invoke certain 'IonValue' methods on that in-memory representation. An actor could craft Ion data that, when loaded by the affected application and/or processed using the 'IonValue' model, results in a 'StackOverflowError' originating from the 'ion-java' library. The patch is included in 'ion-java' 1.10.5. As a workaround, do not load data which originated from an untrusted source or that could have been tampered with.

CVE ID	Description
CVE-2024-22099	NULL Pointer Dereference vulnerability in Linux Linux kernel kernel on Linux, x86, ARM (net, bluetooth modules) allows Overflow Buffers. This vulnerability is associated with program files /net/bluetooth/rfcomm/core.C. This issue affects Linux kernel: v2.6.12-rc2.
CVE-2024-22243	Applications that use UriComponentsBuilder to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to an open redirect https://cwe.mitre.org/data/definitions/601.html attack or to a SSRF attack if the URL is used after passing validation checks.
CVE-2024-22259	Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to an open redirect https://cwe.mitre.org/data/definitions/601.html attack or to a SSRF attack if the URL is used after passing validation checks. This is the same as CVE-2024-22243 https://spring.io/security/cve-2024-22243 , but with different input.
CVE-2024-22262	Applications that use UriComponentsBuilder to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to an open redirect https://cwe.mitre.org/data/definitions/601.html attack or to a SSRF attack if the URL is used after passing validation checks. This is the same as CVE-2024-22259 https://spring.io/security/cve-2024-22259 and CVE-2024-22243 https://spring.io/security/cve-2024-22243 , but with different input.
CVE-2024-2236	A timing-based side-channel flaw was found in libcrypto's RSA implementation. This issue may allow a remote attacker to initiate a Bleichenbacher-style attack, which can lead to the decryption of RSA ciphertexts.
CVE-2024-22667	Vim before 9.0.2142 has a stack-based buffer overflow because did_set_langmap in map.c calls sprintf to write to the error buffer that is passed down to the option callback functions.
CVE-2024-23196	A race condition was found in the Linux kernel's sound/hda device driver in snd_hdac_regmap_sync() function. This can result in a null pointer dereference issue, possibly leading to a kernel panic or denial of service issue.
CVE-2024-23454	Apache Hadoop's RunJar.run() does not set permissions for temporary directory by default. If sensitive data will be present in this file, all the other local users may be able to view the content. This is because, on unix-like systems, the system temporary directory is shared between all local users. As such, files written in this directory, without setting the correct posix permissions explicitly, may be viewable by all other local users.
CVE-2024-23775	Integer Overflow vulnerability in Mbed TLS 2.x before 2.28.7 and 3.x before 3.5.2, allows attackers to cause a denial of service (DoS) via mbedtls_x509_set_extension().
CVE-2024-23944	Information disclosure in persistent watchers handling in Apache ZooKeeper due to missing ACL check. It allows an attacker to monitor child znodes by attaching a persistent watcher (addWatch command) to a parent which the attacker has already access to. ZooKeeper server doesn't do ACL check when the persistent watcher is triggered and as a consequence, the full path of znodes that a watch event gets triggered upon is exposed to the owner of the watcher. It's important to note that only the path is exposed by this vulnerability, not the data of znode, but since znode path can contain sensitive information like user name or login ID, this issue is potentially critical. Users are recommended to upgrade to version 3.9.2, 3.8.4 which fixes the issue.
CVE-2024-23945	Signing cookies is an application security feature that adds a digital signature to cookie data to verify its authenticity and integrity. The signature helps prevent malicious actors from modifying the cookie value, which can lead to security vulnerabilities and exploitation. Apache Hive's service component accidentally exposes the signed cookie to the end user when there is a mismatch in signature between the current and expected cookie. Exposing the correct cookie signature can lead to further exploitation. The vulnerable CookieSigner logic was introduced in Apache Hive by HIVE-9710 (1.2.0) and in Apache Spark by SPARK-14987 (2.0.0). The affected components are the following: * org.apache.hive:hive-service * org.apache.spark:spark-hive-thriftserver_2.11 * org.apache.spark:spark-hive-thriftserver_2.12
CVE-2024-23953	Use of Arrays.equals() in LlapSignerImpl in Apache Hive to compare message signatures allows attacker to forge a valid signature for an arbitrary message byte by byte. The attacker should be an authorized user of the product to perform this attack. Users are recommended to upgrade to version 4.0.0, which fixes this issue. The problem occurs when an application doesn't use a constant-time algorithm for validating a signature. The method Arrays.equals() returns false right away when it sees that one of the input's bytes are different. It means that the comparison time depends on the contents of the arrays. This little thing may allow an attacker to forge a valid signature for an arbitrary message byte by byte. So it might allow malicious users to submit splits/work with selected signatures to LLAP without running as a privileged user, potentially leading to DDoS attack. More details in the reference section.

CVE ID	Description
CVE-2024-24783	Verifying a certificate chain which contains a certificate with an unknown public key algorithm will cause Certificate.Verify to panic. This affects all crypto/tls clients, and servers that set Config.ClientAuth to VerifyClientCertIfGiven or RequireAndVerifyClientCert. The default behavior is for TLS servers to not verify client certificates.
CVE-2024-24784	The ParseAddressList function incorrectly handles comments (text within parentheses) within display names. Since this is a misalignment with conforming address parsers, it can result in different trust decisions being made by programs using different parsers.
CVE-2024-24785	If errors returned from MarshalJSON methods contain user controlled data, they may be used to break the contextual auto-escaping behavior of the html/template package, allowing for subsequent actions to inject unexpected content into templates.
CVE-2024-24786	The protojson.Unmarshal function can enter an infinite loop when unmarshaling certain forms of invalid JSON. This condition can occur when unmarshaling into a message which contains a google.protobuf.Any value, or when the UnmarshalOptions.DiscardUnknown option is set.
CVE-2024-24788	A malformed DNS message in response to a query can cause the Lookup functions to get stuck in an infinite loop.
CVE-2024-24789	The archive/zip package's handling of certain types of invalid zip files differs from the behavior of most zip implementations. This misalignment could be exploited to create a zip file with contents that vary depending on the implementation reading the file. The archive/zip package now rejects files containing these errors.
CVE-2024-24790	The various Is methods (IsPrivate, IsLoopback, etc) did not work as expected for IPv4-mapped IPv6 addresses, returning false for addresses which would return true in their traditional IPv4 forms.
CVE-2024-24791	The net/http HTTP/1.1 client mishandled the case where a server responds to a request with an "Expect: 100-continue" header with a non-informational (200 or higher) status. This mishandling could leave a client connection in an invalid state, where the next request sent on the connection will fail. An attacker sending a request to a net/http/httputil.ReverseProxy proxy can exploit this mishandling to cause a denial of service by sending "Expect: 100-continue" requests which elicit a non-informational response from the backend. Each such request leaves the proxy with an invalid connection, and causes one subsequent request using that connection to fail.
CVE-2024-24795	HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious response headers into backend applications to cause an HTTP desynchronization attack. Users are recommended to upgrade to version 2.4.59, which fixes this issue.
CVE-2024-24855	A race condition was found in the Linux kernel's scsi device driver in lpfc_unregister_fcf_rescan() function. This can result in a null pointer dereference issue, possibly leading to a kernel panic or denial of service issue.
CVE-2024-24858	A race condition was found in the Linux kernel's net/bluetooth in {conn,adv}_min,max_interval_set() function. This can result in I2cap connection or broadcast abnormality issue, possibly leading to denial of service.
CVE-2024-24859	A race condition was found in the Linux kernel's net/bluetooth in sniff_min,max_interval_set() function. This can result in a bluetooth sniffing exception issue, possibly leading denial of service.
CVE-2024-2511	Issue summary: Some non-default TLS server configurations can cause unbounded memory growth when processing TLSv1.3 sessions Impact summary: An attacker may exploit certain server configurations to trigger unbounded memory growth that would lead to a Denial of Service This problem can occur in TLSv1.3 if the non-default SSL_OP_NO_TICKET option is being used (but not if early_data support is also configured and the default anti-replay protection is in use). In this case, under certain conditions, the session cache can get into an incorrect state and it will fail to flush properly as it fills. The session cache will continue to grow in an unbounded manner. A malicious client could deliberately create the scenario for this failure to force a Denial of Service. It may also happen by accident in normal operation. This issue only affects TLS servers supporting TLSv1.3. It does not affect TLS clients. The FIPS modules in 3.2, 3.1 and 3.0 are not affected by this issue. OpenSSL 1.0.2 is also not affected by this issue.
CVE-2024-25126	Rack is a modular Ruby web server interface. Carefully crafted content type headers can cause Rack's media type parser to take much longer than expected, leading to a possible denial of service vulnerability (ReDos 2nd degree polynomial). This vulnerability is patched in 3.0.9.1 and 2.2.8.1.
CVE-2024-25260	elfutils v0.189 was discovered to contain a NULL pointer dereference via the handle_verdef() function at readelf.c.

CVE ID	Description
CVE-2024-25638	dnsjava is an implementation of DNS in Java. Records in DNS replies are not checked for their relevance to the query, allowing an attacker to respond with RRs from different zones. This vulnerability is fixed in 3.6.0.
CVE-2024-25710	Loop with Unreachable Exit Condition ('Infinite Loop') vulnerability in Apache Commons Compress. This issue affects Apache Commons Compress: from 1.3 through 1.25.0. Users are recommended to upgrade to version 1.26.0 which fixes the issue.
CVE-2024-25740	A memory leak flaw was found in the UBI driver in drivers/mtd/ubi/attach.c in the Linux kernel through 6.7.4 for UBI_IOCATT, because kobj->name is not released.
CVE-2024-26130	cryptography is a package designed to expose cryptographic primitives and recipes to Python developers. Starting in version 38.0.0 and prior to version 42.0.4, if <code>`pkcs12.serialize_key_and_certificates`</code> is called with both a certificate whose public key did not match the provided private key and an <code>`encryption_algorithm`</code> with <code>`hmac_hash`</code> set (via <code>`PrivateFormat.PKCS12.encryption_builder().hmac_hash(...)`</code>), then a NULL pointer dereference would occur, crashing the Python process. This has been resolved in version 42.0.4, the first version in which a <code>`ValueError`</code> is properly raised.
CVE-2024-26141	Rack is a modular Ruby web server interface. Carefully crafted Range headers can cause a server to respond with an unexpectedly large response. Responding with such large responses could lead to a denial of service issue. Vulnerable applications will use the <code>`Rack::File`</code> middleware or the <code>`Rack::Utils.byte_ranges`</code> methods (this includes Rails applications). The vulnerability is fixed in 3.0.9.1 and 2.2.8.1.
CVE-2024-26146	Rack is a modular Ruby web server interface. Carefully crafted headers can cause header parsing in Rack to take longer than expected resulting in a possible denial of service issue. Accept and Forwarded headers are impacted. Ruby 3.2 has mitigations for this problem, so Rack applications using Ruby 3.2 or newer are unaffected. This vulnerability is fixed in 2.0.9.4, 2.1.4.4, 2.2.8.1, and 3.0.9.1.
CVE-2024-26308	Allocation of Resources Without Limits or Throttling vulnerability in Apache Commons Compress. This issue affects Apache Commons Compress: from 1.21 before 1.26. Users are recommended to upgrade to version 1.26, which fixes the issue.
CVE-2024-26458	Kerberos 5 (aka krb5) 1.21.2 contains a memory leak in <code>/krb5/src/lib/krb5/pmap_rmt.c</code> .
CVE-2024-26461	Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in <code>/krb5/src/lib/gssapi/krb5/k5sealv3.c</code> .
CVE-2024-26462	Kerberos 5 (aka krb5) 1.21.2 contains a memory leak vulnerability in <code>/krb5/src/kdc/ndr.c</code> .
CVE-2024-26589	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject variable offset alu on PTR_TO_FLOW_KEYS For PTR_TO_FLOW_KEYS, check_flow_keys_access() only uses fixed off for validation. However, variable offset ptr alu is not prohibited for this ptr kind. So the variable offset is not checked. The following prog is accepted: func#0 @0 0: R1=ctx() R10=fp0 0: (bf) r6 = r1 ; R1=ctx() R6_w=ctx() 1: (79) r7 = *(u64*)(r6 +144) ; R6_w=ctx() R7_w=flow_keys() 2: (b7) r8 = 1024 ; R8_w=1024 3: (37) r8 /= 1 ; R8_w=scalar() 4: (57) r8 &= 1024 ; R8_w=scalar(smin=smin32=0, smax=umax=smax32=umax32=1024,var_off=(0x0; 0x400)) 5: (0f) r7 += r8 mark_precise: frame0: last_idx 5 first_idx 0 subseq_idx -1 mark_precise: frame0: regs=r8 stack= before 4: (57) r8 &= 1024 mark_precise: frame0: regs=r8 stack= before 3: (37) r8 /= 1 mark_precise: frame0: regs=r8 stack= before 2: (b7) r8 = 1024 6: R7_w=flow_keys(smin=smin32=0,smax=umax=smax32=umax32=1024,var_off=(0x0; 0x400)) R8_w=scalar(smin=smin32=0,smax=umax=smax32=umax32=1024, var_off=(0x0; 0x400)) 6: (79) r0 = *(u64*)(r7 +0) ; R0_w=scalar() 7: (95) exit This prog loads flow_keys to r7, and adds the variable offset r8 to r7, and finally causes out-of-bounds access: BUG: unable to handle page fault for address: fffff90014c80038 [...] Call Trace: <TASK> bpf_dispatcher_nop_func include/linux/bpf.h:1231 [inline] __bpf_prog_run include/linux/filter.h:651 [inline] bpf_prog_run include/linux/filter.h:658 [inline] bpf_prog_run_pin_on_cpu include/linux/filter.h:675 [inline] bpf_flow_dissect +0x15f/0x350 net/core/flow_dissector.c:991 bpf_prog_test_run_flow_dissector+0x39d/0x620 net/bpf/test_run.c:1359 bpf_prog_test_run kernel/bpf/syscall.c:4107 [inline] __sys_bpf+0xf8f/0x4560 kernel/bpf/syscall.c:5475 __do_sys_bpf kernel/bpf/syscall.c:5561 [inline] __se_sys_bpf kernel/bpf/syscall.c:5559 [inline] __x64_sys_bpf+0x73/0xb0 kernel/bpf/syscall.c:5559 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x3f/0x110 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0xb6 Fix this by rejecting ptr alu with variable offset on flow_keys. Applying the patch rejects the program with "R7 pointer arithmetic on flow_keys prohibited".

CVE ID	Description
CVE-2024-26591	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix re-attachment branch in bpf_tracing_prog_attach The following case can cause a crash due to missing attach_btf: 1) load rawtp program 2) load fentry program with rawtp as target_fd 3) create tracing link for fentry program with target_fd = 0 4) repeat 3 In the end we have: - prog->aux->dst_trampoline == NULL - tgt_prog == NULL (because we did not provide target_fd to link_create) - prog->aux->attach_btf == NULL (the program was loaded with attach_prog_fd=X) - the program was loaded for tgt_prog but we have no way to find out which one BUG: kernel NULL pointer dereference, address: 0000000000000058 Call Trace: <TASK> ? __die+0x20/0x70 ? page_fault_oops+0x15b/0x430 ? fixup_exception+0x22/0x330 ? exc_page_fault+0x6f/0x170 ? asm_exc_page_fault+0x22/0x30 ? bpf_tracing_prog_attach+0x279/0x560 ? btf_obj_id+0x5/0x10 bpf_tracing_prog_attach+0x439/0x560 __sys_bpf+0x1cf4/0x2de0 __x64_sys_bpf+0x1c/0x30 do_syscall_64+0x41/0xf0 entry_SYSCALL_64_after_hwframe+0x6e/0x76 Return -EINVAL in this situation.
CVE-2024-26598	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: vgic-its: Avoid potential UAF in LPI translation cache There is a potential UAF scenario in the case of an LPI translation cache hit racing with an operation that invalidates the cache, such as a DISCARD ITS command. The root of the problem is that vgic_its_check_cache() does not elevate the refcount on the vgic_irq before dropping the lock that serializes refcount changes. Have vgic_its_check_cache() raise the refcount on the returned vgic_irq and add the corresponding decrement after queueing the interrupt.
CVE-2024-26611	In the Linux kernel, the following vulnerability has been resolved: xsk: fix usage of multi-buffer BPF helpers for ZC XDP Currently when packet is shrunk via bpf_xdp_adjust_tail() and memory type is set to MEM_TYPE_XSK_BUFF_POOL, null ptr dereference happens: [1136314.192256] BUG: kernel NULL pointer dereference, address: 0000000000000034 [1136314.203943] #PF: supervisor read access in kernel mode [1136314.213768] #PF: error_code(0x0000) - not-present page [1136314.223550] PGD 0 P4D 0 [1136314.230684] Oops: 0000 [#1] PREEMPT SMP NOPTI [1136314.239621] CPU: 8 PID: 54203 Comm: xdpsock Not tainted 6.6.0+ #257 [1136314.250469] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0008.031920191559 03/19/2019 [1136314.265615] RIP: 0010: __xdp_return+0x6c/0x210 [1136314.274653] Code: ad 00 48 8b 47 08 49 89 f8 a8 01 0f 85 9b 01 00 00 0f 1f 44 00 00 f0 41 ff 48 34 75 32 4c 89 c7 e9 79 cd 80 ff 83 fe 03 75 17 <f6> 41 34 01 0f 85 02 01 00 00 48 89 cf e9 22 cc 1e 00 e9 3d d2 86 [1136314.302907] RSP: 0018:ffffc900089f8db0 EFLAGS: 00010246 [1136314.312967] RAX: fffffc9003168aed0 RBX: fffffc9003168aed0 RCX: 0000000000000000 [1136314.324953] RDX: 0000000000000000 RSI: 0000000000000000 RDI: fffffc9003168c000 [1136314.336929] RBP: 00000000000000ae0 R08: 0000000000000002 R09: 0000000000010000 [1136314.348844] R10: fffffc9000e495000 R11: 0000000000000040 R12: 0000000000000001 [1136314.360706] R13: 0000000000000524 R14: fffffc9003168aec0 R15: 0000000000000001 [1136314.373298] FS: 00007f8df8bbcb80(0000) GS:ffff8897e0e00000(0000) knlGS:0000000000000000 [1136314.386105] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [1136314.396532] CR2: 0000000000000034 CR3: 00000001aa912002 CR4: 00000000007706f0 [1136314.408377] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [1136314.420173] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [1136314.431890] PKRU: 55555554 [1136314.439143] Call Trace: [1136314.446058] <IRQ> [1136314.452465] ? __die+0x20/0x70 [1136314.459881] ? page_fault_oops+0x15b/0x440 [1136314.468305] ? exc_page_fault+0x6a/0x150 [1136314.476491] ? asm_exc_page_fault+0x22/0x30 [1136314.484927] ? __xdp_return+0x6c/0x210 [1136314.492863] bpf_xdp_adjust_tail+0x155/0x1d0 [1136314.501269] bpf_prog_ccc47ae29d3b6570_xdp_sock_prog+0x15/0x60 [1136314.511263] ice_clean_rx_irq_zc+0x206/0xc60 [ice] [1136314.520222] ? ice_xmit_zc+0x6e/0x150 [ice] [1136314.528506] ice_napi_poll+0x467/0x670 [ice] [1136314.536858] ? ttwu_do_activate.constprop.0+0x8f/0x1a0 [1136314.546010] __napi_poll+0x29/0x1b0 [1136314.553462] net_rx_action+0x133/0x270 [1136314.561619] __do_softirq+0xbe/0x28e [1136314.569303] do_softirq+0x3f/0x60 This comes from __xdp_return() call with xdp_buff argument passed as NULL which is supposed to be consumed by xsk_buff_free() call. To address this properly, in ZC case, a node that represents the frag being removed has to be pulled out of xskb_list. Introduce appropriate xsk helpers to do such node operation and use them accordingly within bpf_xdp_adjust_tail().

CVE ID	Description
CVE-2024-26633	In the Linux kernel, the following vulnerability has been resolved: ip6_tunnel: fix NEXTHDR_FRAGMENT handling in ip6_tnl_parse_tlv_enc_lim() syzbot pointed out [1] that NEXTHDR_FRAGMENT handling is broken. Reading frag_off can only be done if we pulled enough bytes to skb->head. Currently we might access garbage. [1] BUG: KMSAN: uninit-value in ip6_tnl_parse_tlv_enc_lim+0x94f/0xbb0 ip6_tnl_parse_tlv_enc_lim+0x94f/0xbb0 ipxip6_tnl_xmit net/ipv6/ip6_tunnel.c:1326 [inline] ip6_tnl_start_xmit+0xab2/0x1a70 net/ipv6/ip6_tunnel.c:1432 __netdev_start_xmit include/linux/netdevice.h:4940 [inline] netdev_start_xmit include/linux/netdevice.h:4954 [inline] xmit_one net/core/dev.c:3548 [inline] dev_hard_start_xmit+0x247/0xa10 net/core/dev.c:3564 __dev_queue_xmit+0x33b8/0x5130 net/core/dev.c:4349 dev_queue_xmit include/linux/netdevice.h:3134 [inline] neigh_connected_output+0x569/0x660 net/core/neighbour.c:1592 neigh_output include/net/neighbour.h:542 [inline] ip6_finish_output2+0x23a9/0x2b30 net/ipv6/ip6_output.c:137 ip6_finish_output+0x855/0x12b0 net/ipv6/ip6_output.c:222 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip6_output+0x323/0x610 net/ipv6/ip6_output.c:243 dst_output include/net/dst.h:451 [inline] ip6_local_out+0xe9/0x140 net/ipv6/output_core.c:155 ip6_send_skb net/ipv6/ip6_output.c:1952 [inline] ip6_push_pending_frames+0x1f9/0x560 net/ipv6/ip6_output.c:1972 rawv6_push_pending_frames+0xbe8/0xdf0 net/ipv6/raw.c:582 rawv6_sendmsg+0x2b66/0x2e70 net/ipv6/raw.c:920 inet_sendmsg+0x105/0x190 net/ipv4/af_inet.c:847 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] ____sys_sendmsg+0x9c2/0xd60 net/socket.c:2584 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2638 __sys_sendmsg net/socket.c:2667 [inline] __do_sys_sendmsg net/socket.c:2676 [inline] __se_sys_sendmsg net/socket.c:2674 [inline] __x64_sys_sendmsg+0x307/0x490 net/socket.c:2674 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x44/0x110 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0xb Uninit was created at: slab_post_alloc_hook+0x129/0xa70 mm/slab.h:768 slab_alloc_node mm/slub.c:3478 [inline] __kmem_cache_alloc_node+0x5c9/0x970 mm/slub.c:3517 __do_kmalloc_node mm/slab_common.c:1006 [inline] __kmalloc_node_track_caller+0x118/0x3c0 mm/slab_common.c:1027 kmalloc_reserve+0x249/0x4a0 net/core/skbuff.c:582 pskb_expand_head+0x226/0x1a00 net/core/skbuff.c:2098 __pskb_pull_tail+0x13b/0x2310 net/core/skbuff.c:2655 pskb_may_pull_reason include/linux/skbuff.h:2673 [inline] pskb_may_pull include/linux/skbuff.h:2681 [inline] ip6_tnl_parse_tlv_enc_lim+0x901/0xbb0 net/ipv6/ip6_tunnel.c:408 ipxip6_tnl_xmit net/ipv6/ip6_tunnel.c:1326 [inline] ip6_tnl_start_xmit+0xab2/0x1a70 net/ipv6/ip6_tunnel.c:1432 __netdev_start_xmit include/linux/netdevice.h:4940 [inline] netdev_start_xmit include/linux/netdevice.h:4954 [inline] xmit_one net/core/dev.c:3548 [inline] dev_hard_start_xmit+0x247/0xa10 net/core/dev.c:3564 __dev_queue_xmit+0x33b8/0x5130 net/core/dev.c:4349 dev_queue_xmit include/linux/netdevice.h:3134 [inline] neigh_connected_output+0x569/0x660 net/core/neighbour.c:1592 neigh_output include/net/neighbour.h:542 [inline] ip6_finish_output2+0x23a9/0x2b30 net/ipv6/ip6_output.c:137 ip6_finish_output+0x855/0x12b0 net/ipv6/ip6_output.c:222 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip6_output+0x323/0x610 net/ipv6/ip6_output.c:243 dst_output include/net/dst.h:451 [inline] ip6_local_out+0xe9/0x140 net/ipv6/output_core.c:155 ip6_send_skb net/ipv6/ip6_output.c:1952 [inline] ip6_push_pending_frames+0x1f9/0x560 net/ipv6/ip6_output.c:1972 rawv6_push_pending_frames+0xbe8/0xdf0 net/ipv6/raw.c:582 rawv6_sendmsg+0x2b66/0x2e70 net/ipv6/raw.c:920 inet_sendmsg+0x105/0x190 net/ipv4/af_inet.c:847 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] ____sys_sendmsg+0x9c2/0xd60 net/socket.c:2584 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2638 __sys_sendmsg net/socket.c:2667 [inline] __do_sys_sendms ---truncated---

CVE ID	Description
CVE-2024-26641	In the Linux kernel, the following vulnerability has been resolved: ip6_tunnel: make sure to pull inner header in __ip6_tnl_rcv() syzbot found __ip6_tnl_rcv() could access unutilized data [1]. Call pskb_inet_may_pull() to fix this, and initialize ipv6h variable after this call as it can change skb->head. [1] BUG: KMSAN: uninit-value in __INET_ECN_decapsulate include/net/inet_ecn.h:253 [inline] BUG: KMSAN: uninit-value in INET_ECN_decapsulate include/net/inet_ecn.h:275 [inline] BUG: KMSAN: uninit-value in IP6_ECN_decapsulate+0x7df/0x1e50 include/net/inet_ecn.h:321 __INET_ECN_decapsulate include/net/inet_ecn.h:253 [inline] INET_ECN_decapsulate include/net/inet_ecn.h:275 [inline] IP6_ECN_decapsulate+0x7df/0x1e50 include/net/inet_ecn.h:321 ip6ip6_dscp_ecn_decapsulate+0x178/0x1b0 net/ipv6/ip6_tunnel.c:727 __ip6_tnl_rcv+0xd4e/0x1590 net/ipv6/ip6_tunnel.c:845 ip6_tnl_rcv+0xce/0x100 net/ipv6/ip6_tunnel.c:888 gre_rcv+0x143f/0x1870 ip6_protocol_deliver_rcu+0xda6/0x2a60 net/ipv6/ip6_input.c:438 ip6_input_finish net/ipv6/ip6_input.c:483 [inline] NF_HOOK include/linux/netfilter.h:314 [inline] ip6_input+0x15d/0x430 net/ipv6/ip6_input.c:492 ip6_mc_input+0xa7e/0xc80 net/ipv6/ip6_input.c:586 dst_input include/net/dst.h:461 [inline] ip6_rcv_finish+0x5db/0x870 net/ipv6/ip6_input.c:79 NF_HOOK include/linux/netfilter.h:314 [inline] ipv6_rcv+0xda/0x390 net/ipv6/ip6_input.c:310 __netif_receive_skb_one_core net/core/dev.c:5532 [inline] __netif_receive_skb+0x5a0 net/core/dev.c:5646 netif_receive_skb_internal net/core/dev.c:5732 [inline] netif_receive_skb+0x58/0x660 net/core/dev.c:5791 tun_rx_batched+0x3ee/0x980 drivers/net/tun.c:1555 tun_get_user+0x53af/0x66d0 drivers/net/tun.c:2002 tun_chr_write_iter+0x3af/0x5d0 drivers/net/tun.c:2048 call_write_iter include/linux/fs.h:2084 [inline] new_sync_write fs/read_write.c:497 [inline] vfs_write+0x786/0x1200 fs/read_write.c:590 ksys_write+0x20f/0x4c0 fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __x64_sys_write+0x93/0xd0 fs/read_write.c:652 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x6d/0x140 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b Uninit was created at: slab_post_alloc_hook+0x129/0xa70 mm/slab.h:768 slab_alloc_node mm/slab.c:3478 [inline] kmem_cache_alloc_node+0x5e9/0xb10 mm/slab.c:3523 kmmalloc_reserve+0x13d/0x4a0 net/core/skbuff.c:560 __alloc_skb+0x318/0x740 net/core/skbuff.c:651 alloc_skb include/linux/skbuff.h:1286 [inline] alloc_skb_with_frags+0xc8/0xb0 net/core/skbuff.c:6334 sock_alloc_send_pskb+0xa80/0xb0 net/core/sock.c:2787 tun_alloc_skb drivers/net/tun.c:1531 [inline] tun_get_user+0x1e8a/0x66d0 drivers/net/tun.c:1846 tun_chr_write_iter+0x3af/0x5d0 drivers/net/tun.c:2048 call_write_iter include/linux/fs.h:2084 [inline] new_sync_write fs/read_write.c:497 [inline] vfs_write+0x786/0x1200 fs/read_write.c:590 ksys_write+0x20f/0x4c0 fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __x64_sys_write+0x93/0xd0 fs/read_write.c:652 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x6d/0x140 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b CPU: 0 PID: 5034 Comm: syz-executor331 Not tainted 6.7.0-syzkaller-00562-g9f8413c4a66f #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 11/17/2023
CVE-2024-26646	In the Linux kernel, the following vulnerability has been resolved: thermal: intel: hfi: Add syscore callbacks for system-wide PM The kernel allocates a memory buffer and provides its location to the hardware, which uses it to update the HFI table. This allocation occurs during boot and remains constant throughout runtime. When resuming from hibernation, the restore kernel allocates a second memory buffer and reprograms the HFI hardware with the new location as part of a normal boot. The location of the second memory buffer may differ from the one allocated by the image kernel. When the restore kernel transfers control to the image kernel, its HFI buffer becomes invalid, potentially leading to memory corruption if the hardware writes to it (the hardware continues to use the buffer from the restore kernel). It is also possible that the hardware "forgets" the address of the memory buffer when resuming from "deep" suspend. Memory corruption may also occur in such a scenario. To prevent the described memory corruption, disable HFI when preparing to suspend or hibernate. Enable it when resuming. Add syscore callbacks to handle the package of the boot CPU (packages of non-boot CPUs are handled via CPU offline). Syscore ops always run on the boot CPU. Additionally, HFI only needs to be disabled during "deep" suspend and hibernation. Syscore ops only run in these cases. [rjw: Comment adjustment, subject and changelog edits]
CVE-2024-26647	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix late dereference 'dsc' check in 'link_set_dsc_pps_packet()' In link_set_dsc_pps_packet(), 'struct display_stream_compressor *dsc' was dereferenced in a DC_LOGGER_INIT(dsc->ctx->logger); before the 'dsc' NULL pointer check. Fixes the below: drivers/gpu/drm/amd/amdgpu/./display/dc/link/link_dpms.c:905 link_set_dsc_pps_packet() warn: variable dereferenced before check 'dsc' (see line 903)
CVE-2024-26648	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix variable dereferencing before NULL check in edp_setup_replay() In edp_setup_replay(), 'struct dc *dc' & 'struct dmub_replay *replay' was dereferenced before the pointer 'link' & 'replay' NULL check. Fixes the below: drivers/gpu/drm/amd/amdgpu/./display/dc/link/protocols/link_edp_panel_control.c:947 edp_setup_replay() warn: variable dereferenced before check 'link' (see line 933)

CVE ID	Description
CVE-2024-26663	In the Linux kernel, the following vulnerability has been resolved: tipc: Check the bearer type before calling tipc_udp_nl_bearer_add() syzbot reported the following general protection fault [1]: general protection fault, probably for non-canonical address 0xdffffc0000000010: 0000 [#1] PREEMPT SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000080-0x0000000000000087] ... RIP: 0010:tipc_udp_is_known_peer+0x9c/0x250 net/tipc/udp_media.c:291 ... Call Trace: <TASK> tipc_udp_nl_bearer_add+0x212/0x2f0 net/tipc/udp_media.c:646 tipc_nl_bearer_add+0x21e/0x360 net/tipc/bearer.c:1089 genl_family_rcv_msg_doit+0x1fc/0x2e0 net/netlink/genetlink.c:972 genl_family_rcv_msg net/netlink/genetlink.c:1052 [inline] genl_rcv_msg+0x561/0x800 net/netlink/genetlink.c:1067 netlink_rcv_skb+0x16b/0x440 net/netlink/af_netlink.c:2544 genl_rcv+0x28/0x40 net/netlink/genetlink.c:1076 netlink_unicast_kernel net/netlink/af_netlink.c:1341 [inline] netlink_unicast+0x53b/0x810 net/netlink/af_netlink.c:1367 netlink_sendmsg+0x8b7/0xd70 net/netlink/af_netlink.c:1909 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0xd5/0x180 net/socket.c:745 ____sys_sendmsg+0x6ac/0x940 net/socket.c:2584 ____sys_sendmsg+0x135/0x1d0 net/socket.c:2638 __sys_sendmsg+0x117/0x1e0 net/socket.c:2667 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x40/0x110 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0xb6 The cause of this issue is that when tipc_nl_bearer_add() is called with the TIPC_NLA_BEARER_UDP_OPTS attribute, tipc_udp_nl_bearer_add() is called even if the bearer is not UDP. tipc_udp_is_known_peer() called by tipc_udp_nl_bearer_add() assumes that the media_ptr field of the tipc_bearer has an udp_bearer type object, so the function goes crazy for non-UDP bearers. This patch fixes the issue by checking the bearer type before calling tipc_udp_nl_bearer_add() in tipc_nl_bearer_add().
CVE-2024-26668	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_limit: reject configurations that cause integer overflow Reject bogus configs where internal token counter wraps around. This only occurs with very very large requests, such as 17gbyte/s. Its better to reject this rather than having incorrect ratelimit.
CVE-2024-26672	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix variable 'mca_funcs' dereferenced before NULL check in 'amdgpu_mca_smu_get_mca_entry()' Fixes the below: drivers/gpu/drm/amd/amdgpu/amdgpu_mca.c:377 amdgpu_mca_smu_get_mca_entry() warn: variable dereferenced before check 'mca_funcs' (see line 368) 357 int amdgpu_mca_smu_get_mca_entry(struct amdgpu_device *adev, enum amdgpu_mca_error_type type, 358 int idx, struct mca_bank_entry *entry) 359 { 360 const struct amdgpu_mca_smu_funcs *mca_funcs = adev->mca_mca_funcs; 361 int count; 362 363 switch (type) { 364 case AMDGPU_MCA_ERROR_TYPE_UE: 365 count = mca_funcs->max_ue_count; mca_funcs is dereferenced here. 366 break; 367 case AMDGPU_MCA_ERROR_TYPE_CE: 368 count = mca_funcs->max_ce_count; mca_funcs is dereferenced here. 369 break; 370 default: 371 return -EINVAL; 372 } 373 374 if (idx >= count) 375 return -EINVAL; 376 377 if (mca_funcs && mca_funcs->mca_get_mca_entry) ^^^^^^^^^ Checked too late!
CVE-2024-26680	In the Linux kernel, the following vulnerability has been resolved: net: atlantic: Fix DMA mapping for PTP hwts ring Function aq_ring_hwts_rx_alloc() maps extra AQ_CFG_RXDS_DEF bytes for PTP HWTS ring but then generic aq_ring_free() does not take this into account. Create and use a specific function to free HWTS ring to fix this issue. Trace: [215.351607] ----- [cut here]----- [215.351612] DMA-API: atlantic 0000:4b:00.0: device driver frees DMA memory with different size [device address=0x00000000fbdd0000] [map size=34816 bytes] [unmap size=32768 bytes] [215.351635] WARNING: CPU: 33 PID: 10759 at kernel/dma/debug.c:988 check_unmap+0xa6f/0x2360 ... [215.581176] Call Trace: [215.583632] <TASK> [215.585745] ? show_trace_log_lvl+0x1c4/0x2df [215.590114] ? show_trace_log_lvl+0x1c4/0x2df [215.594497] ? debug_dma_free_coherent+0x196/0x210 [215.599305] ? check_unmap+0xa6f/0x2360 [215.603147] ? __warn+0xca/0x1d0 [215.606391] ? check_unmap+0xa6f/0x2360 [215.610237] ? report_bug+0x1ef/0x370 [215.613921] ? handle_bug+0x3c/0x70 [215.617423] ? exc_invalid_op+0x14/0x50 [215.621269] ? asm_exc_invalid_op+0x16/0x20 [215.625480] ? check_unmap+0xa6f/0x2360 [215.629331] ? mark_lock.part.0+0xca/0xa40 [215.633445] debug_dma_free_coherent+0x196/0x210 [215.638079] ? __pfx_debug_dma_free_coherent+0x10/0x10 [215.643242] ? slab_free_freelist_hook+0x11d/0x1d0 [215.648060] dma_free_attrs+0x6d/0x130 [215.651834] aq_ring_free+0x193/0x290 [atlantic] [215.656487] aq_ptp_ring_free+0x67/0x110 [atlantic] ... [216.127540] ---[end trace 6467e5964dd2640b]--- [216.132160] DMA-API: Mapped at: [216.132162] debug_dma_alloc_coherent+0x66/0x2f0 [216.132165] dma_alloc_attrs+0xf5/0x1b0 [216.132168] aq_ring_hwts_rx_alloc+0x150/0x1f0 [atlantic] [216.132193] aq_ptp_ring_alloc+0x1bb/0x540 [atlantic] [216.132213] aq_nic_init+0x4a1/0x760 [atlantic]

CVE ID	Description
CVE-2024-26687	In the Linux kernel, the following vulnerability has been resolved: xen/events: close evtchn after mapping cleanup shutdown_pirq and startup_pirq are not taking the irq_mapping_update_lock because they can't due to lock inversion. Both are called with the irq_desc->lock being taking. The lock order, however, is first irq_mapping_update_lock and then irq_desc->lock. This opens multiple races: - shutdown_pirq can be interrupted by a function that allocates an event channel: CPU0 CPU1 shutdown_pirq { xen_evtchn_close(e) __startup_pirq { EVTCHNOP_bind_pirq -> returns just freed evtchn e set_evtchn_to_irq(e, irq) } xen_irq_info_cleanup() { set_evtchn_to_irq(e, -1) } } Assume here event channel e refers here to the same event channel number. After this race the evtchn_to_irq mapping for e is invalid (-1). - __startup_pirq races with __unbind_from_irq in a similar way. Because __startup_pirq doesn't take irq_mapping_update_lock it can grab the evtchn that __unbind_from_irq is currently freeing and cleaning up. In this case even though the event channel is allocated, its mapping can be unset in evtchn_to_irq. The fix is to first cleanup the mappings and then close the event channel. In this way, when an event channel gets allocated it's potential previous evtchn_to_irq mappings are guaranteed to be unset already. This is also the reverse order of the allocation where first the event channel is allocated and then the mappings are setup. On a 5.10 kernel prior to commit 3fcdaf3d7634 ("xen/events: modify internal [un]bind interfaces"), we hit a BUG like the following during probing of NVMe devices. The issue is that during nvme_setup_io_queues, pci_free_irq is called for every device which results in a call to shutdown_pirq. With many nvme devices it's therefore likely to hit this race during boot because there will be multiple calls to shutdown_pirq and startup_pirq are running potentially in parallel. -----[cut here]----- blkfront: xvda: barrier or flush: disabled; persistent grants: enabled; indirect descriptors: enabled; bounce buffer: enabled kernel BUG at drivers/xen/events/events_base.c:499! invalid opcode: 0000 [#1] SMP PTI CPU: 44 PID: 375 Comm: kworker/u257:23 Not tainted 5.10.201-191.748.amzn2.x86_64 #1 Hardware name: Xen HVM domU, BIOS 4.11.amazon 08/24/2006 Workqueue: nvme-reset-wq nvme_reset_work RIP: 0010:bind_evtchn_to_cpu+0xdf/0xf0 Code: 5d 41 5e c3 cc cc cc cc 44 89 f7 e8 2b 55 ad ff 49 89 c5 48 85 c0 0f 84 64 ff ff 4c 8b 68 30 41 83 fe ff 0f 85 60 ff ff ff <0f> 0b 66 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 0f 1f 44 00 00 RSP: 0000:ffffc9000d533b08 EFLAGS: 00010046 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000006 RDX: 0000000000000028 RSI: 00000000ffffff RDI: 00000000ffffff RBP: ffff888107419680 R08: 0000000000000000 R09: ffffffff82d72b00 R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000001 R13: 0000000000000000 R14: 00000000ffffff R15: 0000000000000002 FS: 0000000000000000(0000) GS: ffff88bc8b500000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 0000000002610001 CR4: 00000000001706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: ? show_trace_log_lvl+0x1c1/0x2d9 ? show_trace_log_lvl+0x1c1/0x2d9 ? set_affinity_irq+0xdc/0x1c0 ? __die_body.cold+0x8/0xd ? die+0x2b/0x50 ? do_trap+0x90/0x110 ? bind_evtchn_to_cpu+0xdf/0xf0 ? do_error_trap+0x65/0x80 ? bind_evtchn_to_cpu+0xdf/0xf0 ? exc_invalid_op+0x4e/0x70 ? bind_evtchn_to_cpu+0xdf/0xf0 ? asm_exc_invalid_op+0x12/0x20 ? bind_evtchn_to_cpu+0xdf/0x --- truncated---

CVE ID	Description
CVE-2024-26688	In the Linux kernel, the following vulnerability has been resolved: fs,hugetlb: fix NULL pointer dereference in hugetlbfills_fill_super When configuring a hugetlb filesystem via the fsconfig() syscall, there is a possible NULL dereference in hugetlbfills_fill_super() caused by assigning NULL to ctx->hstate in hugetlbfills_parse_param() when the requested pagesize is non valid. E.g: Taking the following steps: fd = fsopen("hugetlbfills", FSOPEN_CLOEXEC); fsconfig(fd, FSCONFIG_SET_STRING, "pagesize", "1024", 0); fsconfig(fd, FSCONFIG_CMD_CREATE, NULL, NULL, 0); Given that the requested "pagesize" is invalid, ctx->hstate will be replaced with NULL, losing its previous value, and we will print an error: case Opt_pagesize: ps = memparse(param->string, &rest); ctx->hstate = h; if (!ctx->hstate) { pr_err("Unsupported page size %lu MB\n", ps / SZ_1M); return -EINVAL; } return 0; sb->s_blocksize = huge_page_size(ctx->hstate); Causing below Oops. Fix this by replacing ctx->hstate value only when then pagesize is known to be valid. kernel: hugetlbfills: Unsupported page size 0 MB kernel: BUG: kernel NULL pointer dereference, address: 0000000000000028 kernel: #PF: supervisor read access in kernel mode kernel: #PF: error_code(0x0000) - not-present page kernel: PGD 800000010f66c067 P4D 800000010f66c067 PUD 1b22f8067 PMD 0 kernel: Oops: 0000 [#1] PREEMPT SMP PTI kernel: CPU: 4 PID: 5659 Comm: syscall Tainted: G E 6.8.0-rc2-default+ #22 5a47c3fef76212addcc6eb71344aabc35190ae8f kernel: Hardware name: Intel Corp. GROVEPORT/GROVEPORT, BIOS GVPRCRB1.86B.0016.D04.1705030402 05/03/2017 kernel: RIP: 0010:hugetlbfills_fill_super+0xb4/0x1a0 kernel: Code: 48 8b 3b e8 3e c6 ed ff 48 85 c0 48 89 45 20 0f 84 d6 00 00 00 48 b8 ff ff ff ff ff 7f 4c 89 e7 49 89 44 24 20 48 8b 03 <8b> 48 28 b8 00 10 00 00 48 d3 e0 49 89 44 24 18 48 8b 03 8b 40 28 kernel: RSP: 0018:ffffbe9960fcbd48 EFLAGS: 00010246 kernel: RAX: 0000000000000000 RBX: ffff9af5272ae780 RCX: 0000000000372004 kernel: RDX: ffffffff ffffffff RSI: ffffffff ffffffff RDI: ffff9af555e9b000 kernel: RBP: ffff9af52ee66b00 R08: 0000000000000040 R09: 0000000000370004 kernel: R10: ffffbe9960fcbd48 R11: 0000000000000040 R12: ffff9af555e9b000 kernel: R13: ffffffff66b86c0 R14: ffff9af507d2f400 R15: ffff9af507d2f400 kernel: FS: 00007ffbc0ba4740(0000) GS: ffff9b0bd7000000(0000) knlGS: 0000000000000000 kernel: CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 kernel: CR2: 0000000000000028 CR3: 00000001b1ee0000 CR4: 00000000001506f0 kernel: Call Trace: kernel: <TASK> kernel: ? __die_body+0x1a/0x60 kernel: ? page_fault_oops+0x16f/0x4a0 kernel: ? search_bpf_extables+0x65/0x70 kernel: ? fixup_exception+0x22/0x310 kernel: ? exc_page_fault+0x69/0x150 kernel: ? asm_exc_page_fault+0x22/0x30 kernel: ? __pfx_hugetlbfills_fill_super+0x10/0x10 kernel: ? hugetlbfills_fill_super+0xb4/0x1a0 kernel: ? hugetlbfills_fill_super+0x28/0x1a0 kernel: ? __pfx_hugetlbfills_fill_super+0x10/0x10 kernel: vfs_get_super+0x40/0xa0 kernel: ? __pfx_bpf_lsm_capable+0x10/0x10 kernel: vfs_get_tree+0x25/0xd0 kernel: vfs_cmd_create+0x64/0xe0 kernel: __x64_sys_fsconfig+0x395/0x410 kernel: do_syscall_64+0x80/0x160 kernel: ? syscall_exit_to_user_mode+0x82/0x240 kernel: ? do_syscall_64+0x8d/0x160 kernel: ? syscall_exit_to_user_mode+0x82/0x240 kernel: ? do_syscall_64+0x8d/0x160 kernel: ? exc_page_fault+0x69/0x150 kernel: entry_SYSCALL_64_after_hwframe+0x6e/0x76 kernel: RIP: 0033:0x7ffbc0cb87c9 kernel: Code: 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 66 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 97 96 0d 00 f7 d8 64 89 01 48 kernel: RSP: 002b:00007ffc29d2f388 EFLAGS: 00000206 ORIG_RAX: 000000000000001af kernel: RAX: ffffffff ---truncated---
CVE-2024-26689	In the Linux kernel, the following vulnerability has been resolved: ceph: prevent use-after-free in encode_cap_msg() In fs/ceph/caps.c, in encode_cap_msg(), "use after free" error was caught by KASAN at this line - 'ceph_buffer_get(arg->xattr_buf);'. This implies before the refcount could be incremented here, it was freed. In same file, in "handle_cap_grant()" refcount is decremented by this line - 'ceph_buffer_put(ci->i_xattrs.blob);'. It appears that a race occurred and resource was freed by the latter line before the former line could increment it. encode_cap_msg() is called by __send_cap() and __send_cap() is called by ceph_check_caps() after calling __prep_cap(). __prep_cap() is where arg->xattr_buf is assigned to ci->i_xattrs.blob. This is the spot where the refcount must be increased to prevent "use after free" error.

CVE ID	Description
CVE-2024-26695	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp - Fix null pointer dereference in __sev_platform_shutdown_locked The SEV platform device can be shutdown with a null psp_master, e.g., using DEBUG_TEST_DRIVER_REMOVE. Found using KASAN: [137.148210] ccp 0000:23:00.1: enabling device (0000 -> 0002) [137.162647] ccp 0000:23:00.1: no command queues available [137.170598] ccp 0000:23:00.1: sev enabled [137.174645] ccp 0000:23:00.1: psp enabled [137.178890] general protection fault, probably for non-canonical address 0xdffffc000000001e: 0000 [#1] PREEMPT SMP DEBUG_PAGEALLOC KASAN NOPTI [137.182693] KASAN: null-ptr-deref in range [0x00000000000000f0-0x00000000000000f7] [137.182693] CPU: 93 PID: 1 Comm: swapper/0 Not tainted 6.8.0-rc1+ #311 [137.182693] RIP: 0010: __sev_platform_shutdown_locked +0x51/0x180 [137.182693] Code: 08 80 3c 08 00 0f 85 0e 01 00 00 48 8b 1d 67 b6 01 08 48 b8 00 00 00 00 fc ff df 48 8d bb f0 00 00 00 48 89 f9 48 c1 e9 03 <80> 3c 01 00 0f 85 fe 00 00 00 48 8b 9b f0 00 00 00 48 85 db 74 2c [137.182693] RSP: 0018:ffffc900000cf9b0 EFLAGS: 00010216 [137.182693] RAX: dffffc0000000000 RBX: 0000000000000000 RCX: 000000000000001e [137.182693] RDX: 0000000000000000 RSI: 0000000000000008 RDI: 00000000000000f0 [137.182693] RBP: fffffc900000cf9c R08: 0000000000000000 R09: fffffbfff58f5a66 [137.182693] R10: fffffc900000cf9c R11: ffffffffac7ad32f R12: ffff8881e5052c28 [137.182693] R13: ffff8881e5052c28 R14: ffff8881758e43e8 R15: ffffffffac64abf8 [137.182693] FS: 0000000000000000(0000) GS:ffff889de7000000(0000) knlGS:0000000000000000 [137.182693] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [137.182693] CR2: 0000000000000000 CR3: 0000001cf7c7e000 CR4: 000000000350ef0 [137.182693] Call Trace: [137.182693] <TASK> [137.182693] ? show_regs+0x6c/0x80 [137.182693] ? __die_body+0x24/0x70 [137.182693] ? die_addr+0x4b/0x80 [137.182693] ? exc_general_protection+0x126/0x230 [137.182693] ? asm_exc_general_protection+0x2b/0x30 [137.182693] ? __sev_platform_shutdown_locked+0x51/0x180 [137.182693] sev_firmware_shutdown.isra.0+0x1e/0x80 [137.182693] sev_dev_destroy+0x49/0x100 [137.182693] psp_dev_destroy+0x47/0xb0 [137.182693] sp_destroy+0xbb/0x240 [137.182693] sp_pci_remove+0x45/0x60 [137.182693] pci_device_remove+0xaa/0x1d0 [137.182693] device_remove+0xc7/0x170 [137.182693] really_probe+0x374/0xbe0 [137.182693] ? srso_return_thunk+0x5/0x5f [137.182693] __driver_probe_device+0x199/0x460 [137.182693] driver_probe_device+0x4e/0xd0 [137.182693] __driver_attach+0x191/0x3d0 [137.182693] ? __pfx__driver_attach+0x10/0x10 [137.182693] bus_for_each_dev+0x100/0x190 [137.182693] ? __pfx_bus_for_each_dev+0x10/0x10 [137.182693] ? __kasan_check_read+0x15/0x20 [137.182693] ? srso_return_thunk+0x5/0x5f [137.182693] ? _raw_spin_unlock+0x27/0x50 [137.182693] driver_attach+0x41/0x60 [137.182693] bus_add_driver+0x2a8/0x580 [137.182693] driver_register+0x141/0x480 [137.182693] __pci_register_driver+0x1d6/0x2a0 [137.182693] ? srso_return_thunk+0x5/0x5f [137.182693] ? esrt_sysfs_init+0x1cd/0x5d0 [137.182693] ? __pfx_sp_mod_init+0x10/0x10 [137.182693] sp_pci_init+0x22/0x30 [137.182693] sp_mod_init+0x14/0x30 [137.182693] ? __pfx_sp_mod_init+0x10/0x10 [137.182693] do_one_initcall+0xd1/0x470 [137.182693] ? __pfx_do_one_initcall+0x10/0x10 [137.182693] ? parameq+0x80/0xf0 [137.182693] ? srso_return_thunk+0x5/0x5f [137.182693] ? __kmalloc+0x3b0/0x4e0 [137.182693] ? kernel_init_freeable+0x92d/0x1050 [137.182693] ? kasan_populate_vmalloc_pte+0x171/0x190 [137.182693] ? srso_return_thunk+0x5/0x5f [137.182693] kernel_init_freeable+0xa64/0x1050 [137.182693] ? __pfx_kernel_init+0x10/0x10 [137.182693] kernel_init+0x24/0x160 [137.182693] ? __switch_to_asm+0x3e/0x70 [137.182693] ret_from_fork+0x40/0x80 [137.182693] ? __pfx_kernel_init+0x1 ---truncated---

CVE ID	Description
CVE-2024-26700	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix MST Null Ptr for RV The change try to fix below error specific to RV platform: BUG: kernel NULL pointer dereference, address: 0000000000000008 PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 4 PID: 917 Comm: sway Not tainted 6.3.9-arch1-1 #1 124dc55df4f5272ccb409f39ef4872fc2b3376a2 Hardware name: LENOVO 20NKS01Y00/20NKS01Y00, BIOS R12ET61W(1.31) 07/28/2022 RIP: 0010:drm_dp_atomic_find_time_slots+0x5e/0x260 [drm_display_helper] Code: 01 00 00 48 8b 85 60 05 00 00 48 63 80 88 00 00 00 3b 43 28 0f 8d 2e 01 00 00 48 8b 53 30 48 8d 04 80 48 8d 04 c2 48 8b 40 18 <48> 8> RSP: 0018:ffff960cc2df77d8 EFLAGS: 00010293 RAX: 0000000000000000 RBX: ffff8afb87e81280 RCX: 0000000000000224 RDX: ffff8afb9ee37c00 RSI: ffff8afb8da1a578 RDI: ffff8afb87e81280 RBP: ffff8afb83d67000 R08: 0000000000000001 R09: ffff8afb9652f850 R10: ffff960cc2df7908 R11: 0000000000000002 R12: 0000000000000000 R13: ffff8afb8d7688a0 R14: ffff8afb8da1a578 R15: 0000000000000224 FS: 00007f4dac35ce00(0000) GS:ffff8afe30b00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000008 CR3: 000000010ddc6000 CR4: 00000000003506e0 Call Trace: <TASK> ? __die+0x23/0x70 ? page_fault_oops+0x171/0x4e0 ? plist_add+0xbe/0x100 ? exc_page_fault+0x7c/0x180 ? asm_exc_page_fault+0x26/0x30 ? drm_dp_atomic_find_time_slots+0x5e/0x260 [drm_display_helper 0e67723696438d8e02b741593dd50d80b44c2026] ? drm_dp_atomic_find_time_slots+0x28/0x260 [drm_display_helper 0e67723696438d8e02b741593dd50d80b44c2026] compute_mst_dsc_configs_for_link+0x2ff/0xa40 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] ? fill_plane_buffer_attributes+0x419/0x510 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] compute_mst_dsc_configs_for_state+0x1e1/0x250 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] amdgpu_dm_atomic_check+0xecd/0x1190 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] drm_atomic_check_only+0x5c5/0xa40 drm_mode_atomic_ioctl+0x76e/0xabc0 ? _copy_to_user+0x25/0x30 ? drm_ioctl+0x296/0x4b0 ? __pfx_drm_mode_atomic_ioctl+0x10/0x10 drm_ioctl_kernel+0xcd/0x170 drm_ioctl+0x26d/0x4b0 ? __pfx_drm_mode_atomic_ioctl+0x10/0x10 amdgpu_drm_ioctl+0x4e/0x90 [amdgpu 62e600d2a75e9158e1cd0a243bdc8e6da040c054] __x64_sys_ioctl+0x94/0xd0 do_syscall_64+0x60/0x90 ? do_syscall_64+0x6c/0x90 entry_SYSCALL_64_after_hwframe+0x72/0xdc RIP: 0033:0x7f4dad17f76f Code: 00 48 89 44 24 18 31 c0 48 8d 44 24 60 c7 04 24 10 00 00 00 48 89 44 24 08 48 8d 44 24 20 48 89 44 24 10 b8 10 00 00 00 0f 05 <89> c> RSP: 002b:00007ffd9ae859f0 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff9ffda RBX: 000055e255a55900 RCX: 00007f4dad17f76f RDX: 00007ffd9ae85a90 RSI: 00000000c03864bc RDI: 000000000000000b RBP: 00007ffd9ae85a90 R08: 0000000000000003 R09: 0000000000000003 R10: 0000000000000000 R11: 0000000000000246 R12: 00000000c03864bc R13: 000000000000000b R14: 000055e255a7fc60 R15: 000055e255a01eb0 </TASK> Modules linked in: rfcomm snd_seq_dummy snd_hrtimer snd_seq snd_seq_device ccm cmac algif_hash algif_skcipher af_alg joydev mousedev bnep > typec libphy k10temp ipmi_msghandler roles i2c_scmi acpi_cpufreq mac_hid nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_mas> CR2: 0000000000000008 ---[end trace 0000000000000000]--- RIP: 0010:drm_dp_atomic_find_time_slots+0x5e/0x260 [drm_display_helper] Code: 01 00 00 48 8b 85 60 05 00 00 48 63 80 88 00 00 00 3b 43 28 0f 8d 2e 01 00 00 48 8b 53 30 48 8d 04 80 48 8d 04 c2 48 8b 40 18 <48> 8> RSP: 0018:ffff960cc2df77d8 EFLAGS: 00010293 RAX: 0000000000000000 RBX: ffff8afb87e81280 RCX: 0000000000000224 RDX: ffff8afb9ee37c00 RSI: ffff8afb8da1a578 RDI: ffff8afb87e81280 RBP: ffff8afb83d67000 R08: 0000000000000001 R09: ffff8afb9652f850 R10: ffff960cc2df7908 R11: 0000000000000002 R12: 0000000000000000 R13: ffff8afb8d7688a0 R14: ffff8afb8da1a578 R15: 0000000000000224 FS: 00007f4dac35ce00(0000) GS:ffff8afe30b00000(0000) ---truncated---
CVE-2024-26715	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: Fix NULL pointer dereference in dwc3_gadget_suspend In current scenario if Plug-out and Plug-In performed continuously there could be a chance while checking for dwc->gadget_driver in dwc3_gadget_suspend, a NULL pointer dereference may occur. Call Stack: CPU1: CPU2: gadget_unbind_driver dwc3_suspend_common dwc3_gadget_stop dwc3_gadget_suspend dwc3_disconnect_gadget CPU1 basically clears the variable and CPU2 checks the variable. Consider CPU1 is running and right before gadget_driver is cleared and in parallel CPU2 executes dwc3_gadget_suspend where it finds dwc->gadget_driver which is not NULL and resumes execution and then CPU1 completes execution. CPU2 executes dwc3_disconnect_gadget where it checks dwc->gadget_driver is already NULL because of which the NULL pointer dereference occur.

CVE ID	Description
CVE-2024-26718	In the Linux kernel, the following vulnerability has been resolved: dm-crypt, dm-verity: disable tasklets Tasklets have an inherent problem with memory corruption. The function tasklet_action_common calls tasklet_trylock, then it calls the tasklet callback and then it calls tasklet_unlock. If the tasklet callback frees the structure that contains the tasklet or if it calls some code that may free it, tasklet_unlock will write into free memory. The commits 8e14f610159d and d9a02e016aaf try to fix it for dm-crypt, but it is not a sufficient fix and the data corruption can still happen [1]. There is no fix for dm-verity and dm-verity will write into free memory with every tasklet-processed bio. There will be atomic workqueues implemented in the kernel 6.9 [2]. They will have better interface and they will not suffer from the memory corruption problem. But we need something that stops the memory corruption now and that can be backported to the stable kernels. So, I'm proposing this commit that disables tasklets in both dm-crypt and dm-verity. This commit doesn't remove the tasklet support, because the tasklet code will be reused when atomic workqueues will be implemented. [1] https://lore.kernel.org/all/d390d7ee-f142-44d3-822a-87949e14608b@suse.de/T/ [2] https://lore.kernel.org/lkml/20240130091300.2968534-1-tj@kernel.org/
CVE-2024-26719	In the Linux kernel, the following vulnerability has been resolved: nouveau: offload fence uevents work to workqueue This should break the deadlock between the fctx lock and the irq lock. This offloads the processing off the work from the irq into a workqueue.
CVE-2024-26721	In the Linux kernel, the following vulnerability has been resolved: drm/i915/dsc: Fix the macro that calculates DSCC _DSCA_PPS reg address Commit bd077259d0a9 ("drm/i915/vdsc: Add function to read any PPS register") defines a new macro to calculate the DSC PPS register addresses with PPS number as an input. This macro correctly calculates the addresses till PPS 11 since the addresses increment by 4. So in that case the following macro works correctly to give correct register address: _MMIO(_DSCA_PPS_0 + (pps) * 4) However after PPS 11, the register address for PPS 12 increments by 12 because of RC Buffer memory allocation in between. Because of this discontinuity in the address space, the macro calculates wrong addresses for PPS 12 - 16 resulting into incorrect DSC PPS parameter value read/writes causing DSC corruption. This fixes it by correcting this macro to add the offset of 12 for PPS >=12. v3: Add correct paranthesis for pps argument (Jani Nikula) (cherry picked from commit 6074be620c31dc2ae11af96a1a5ea95580976fb5)
CVE-2024-26742	In the Linux kernel, the following vulnerability has been resolved: scsi: smartpqi: Fix disable_managed_interrupts Correct blk-mq registration issue with module parameter disable_managed_interrupts enabled. When we turn off the default PCI_IRQ_AFFINITY flag, the driver needs to register with blk-mq using blk_mq_map_queues(). The driver is currently calling blk_mq_pci_map_queues() which results in a stack trace and possibly undefined behavior. Stack Trace: [7.860089] scsi host2: smartpqi [7.871934] WARNING: CPU: 0 PID: 238 at block/blk-mq-pci.c:52 blk_mq_pci_map_queues+0xca/0xd0 [7.889231] Modules linked in: sd_mod t10_pi sg uas smartpqi(+) crc32c_intel scsi_transport_sas usb_storage dm_mirror dm_region_hash dm_log dm_mod ipmi_devintf ipmi_msghandler fuse [7.924755] CPU: 0 PID: 238 Comm: kworker/0:3 Not tainted 4.18.0-372.88.1.el8_6_smartpqi_test.x86_64 #1 [7.944336] Hardware name: HPE ProLiant DL380 Gen10/ProLiant DL380 Gen10, BIOS U30 03/08/2022 [7.963026] Workqueue: events work_for_cpu_fn [7.978275] RIP: 0010:blk_mq_pci_map_queues+0xca/0xd0 [7.978278] Code: 48 89 de 89 c7 e8 f6 0f 4f 00 3b 05 c4 b7 8e 01 72 e1 5b 31 c0 5d 41 5c 41 5d 41 5e 41 5f e9 7d df 73 00 31 c0 e9 76 df 73 00 <0f> 0b eb bc 90 90 0f 1f 44 00 00 41 57 49 89 ff 41 56 41 55 41 54 [7.978280] RSP: 0018:ffffa95fc3707d50 EFLAGS: 00010216 [7.978283] RAX: 00000000ffffffff RBX: 0000000000000000 RCX: 0000000000000010 [7.978284] RDX: 0000000000000004 RSI: 0000000000000000 RDI: ffff9190c32d4310 [7.978286] RBP: 0000000000000000 R08: fffffa95fc3707d3 R09: ffff91929b81ac00 [7.978287] R10: 0000000000000001 R11: fffffa95fc3707ac0 R12: 0000000000000000 [7.978288] R13: ffff9190c32d4000 R14: 00000000ffffffff R15: ffff9190c4c950a8 [7.978290] FS: 0000000000000000(0000) GS:ffff9193efc00000(0000) knlGS:0000000000000000 [7.978292] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [8.172814] CR2: 000055d11166c000 CR3: 00000002dae10002 CR4: 00000000007706f0 [8.172816] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [8.172817] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [8.172818] PKRU: 55555554 [8.172819] Call Trace: [8.172823] blk_mq_alloc_tag_set+0x12e/0x310 [8.264339] scsi_add_host_with_dma.cold.9+0x30/0x245 [8.279302] pqi_ctl_init+0xacf/0xc8e [smartpqi] [8.294085] ? pqi_pci_probe+0x480/0x4c8 [smartpqi] [8.309015] pqi_pci_probe+0x480/0x4c8 [smartpqi] [8.323286] local_pci_probe+0x42/0x80 [8.337855] work_for_cpu_fn+0x16/0x20 [8.351193] process_one_work+0x1a7/0x360 [8.364462] ? create_worker+0x1a0/0x1a0 [8.379252] worker_thread+0x1ce/0x390 [8.392623] ? create_worker+0x1a0/0x1a0 [8.406295] kthread+0x10a/0x120 [8.418428] ? set_kthread_struct+0x50/0x50 [8.431532] ret_from_fork+0x1f/0x40 [8.444137] ---[end trace 1bf0173d39354506]---

CVE ID	Description
CVE-2024-26747	In the Linux kernel, the following vulnerability has been resolved: usb: roles: fix NULL pointer issue when put module's reference In current design, usb role class driver will get usb_role_switch parent's module reference after the user get usb_role_switch device and put the reference after the user put the usb_role_switch device. However, the parent device of usb_role_switch may be removed before the user put the usb_role_switch. If so, then, NULL pointer issue will be met when the user put the parent module's reference. This will save the module pointer in structure of usb_role_switch. Then, we don't need to find module by iterating long relations.
CVE-2024-26751	In the Linux kernel, the following vulnerability has been resolved: ARM: ep93xx: Add terminator to gpiod_lookup_table Without the terminator, if a con_id is passed to gpio_find() that does not exist in the lookup table the function will not stop looping correctly, and eventually cause an oops.
CVE-2024-26756	In the Linux kernel, the following vulnerability has been resolved: md: Don't register sync_thread for reshape directly Currently, if reshape is interrupted, then reassemble the array will register sync_thread directly from pers->run(), in this case 'MD_RECOVERY_RUNNING' is set directly, however, there is no guarantee that md_do_sync() will be executed, hence stop_sync_thread() will hang because 'MD_RECOVERY_RUNNING' can't be cleared. Last patch make sure that md_do_sync() will set MD_RECOVERY_DONE, however, following hang can still be triggered by dm-raid test shell/lvconvert-raid-reshape.sh occasionally: [root@fedora ~]# cat /proc/1982/stack [<0>] stop_sync_thread+0x1ab/0x270 [md_mod] [<0>] md_frozen_sync_thread+0x5c/0xa0 [md_mod] [<0>] raid_presuspend+0x1e/0x70 [dm_raid] [<0>] dm_table_presuspend_targets+0x40/0xb0 [dm_mod] [<0>] __dm_destroy+0x2a5/0x310 [dm_mod] [<0>] dm_destroy+0x16/0x30 [dm_mod] [<0>] dev_remove+0x165/0x290 [dm_mod] [<0>] ctl_ioctl+0x4bb/0x7b0 [dm_mod] [<0>] dm_ctl_ioctl+0x11/0x20 [dm_mod] [<0>] vfs_ioctl+0x21/0x60 [<0>] __x64_sys_ioctl+0xb9/0xe0 [<0>] do_syscall_64+0xc6/0x230 [<0>] entry_SYSCALL_64_after_hwframe+0x6c/0x74 Meanwhile mddev->recovery is: MD_RECOVERY_RUNNING MD_RECOVERY_INTR MD_RECOVERY_RESHAPE MD_RECOVERY_FROZEN Fix this problem by remove the code to register sync_thread directly from raid10 and raid5. And let md_check_recovery() to register sync_thread.
CVE-2024-26757	In the Linux kernel, the following vulnerability has been resolved: md: Don't ignore read-only array in md_check_recovery() Usually if the array is not read-write, md_check_recovery() won't register new sync_thread in the first place. And if the array is read-write and sync_thread is registered, md_set_readonly() will unregister sync_thread before setting the array read-only. md/raid follow this behavior hence there is no problem. After commit f52f5c71f3d4 ("md: fix stopping sync thread"), following hang can be triggered by test shell/integrity-caching.sh: 1) array is read-only. dm-raid update super block: rs_update_sbs ro = mddev->ro mddev->ro = 0 -> set array read-write md_update_sb 2) register new sync thread concurrently. 3) dm-raid set array back to read-only: rs_update_sbs mddev->ro = ro 4) stop the array: raid_dtr md_stop stop_sync_thread set_bit(MD_RECOVERY_INTR, &mddev->recovery); md_wakeup_thread_directly(mddev->sync_thread); wait_event(..., !test_bit(MD_RECOVERY_RUNNING, &mddev->recovery)) 5) sync thread done: md_do_sync set_bit(MD_RECOVERY_DONE, &mddev->recovery); md_wakeup_thread(mddev->thread); 6) daemon thread can't unregister sync thread: md_check_recovery if (!md_is_rdw(mddev) && !test_bit(MD_RECOVERY_NEEDED, &mddev->recovery)) return; -> MD_RECOVERY_RUNNING can't be cleared, hence step 4 hang; The root cause is that dm-raid manipulate 'mddev->ro' by itself, however, dm-raid really should stop sync thread before setting the array read-only. Unfortunately, I need to read more code before I can refactor the handler of 'mddev->ro' in dm-raid, hence let's fix the problem the easy way for now to prevent dm-raid regression.
CVE-2024-26758	In the Linux kernel, the following vulnerability has been resolved: md: Don't ignore suspended array in md_check_recovery() mddev_suspend() never stop sync_thread, hence it doesn't make sense to ignore suspended array in md_check_recovery(), which might cause sync_thread can't be unregistered. After commit f52f5c71f3d4 ("md: fix stopping sync thread"), following hang can be triggered by test shell/integrity-caching.sh: 1) suspend the array: raid_postsuspend mddev_suspend 2) stop the array: raid_dtr md_stop __md_stop_writes stop_sync_thread set_bit(MD_RECOVERY_INTR, &mddev->recovery); md_wakeup_thread_directly(mddev->sync_thread); wait_event(..., !test_bit(MD_RECOVERY_RUNNING, &mddev->recovery)) 3) sync thread done: md_do_sync set_bit(MD_RECOVERY_DONE, &mddev->recovery); md_wakeup_thread(mddev->thread); 4) daemon thread can't unregister sync thread: md_check_recovery if (mddev->suspended) return; -> return directly md_read_sync_thread clear_bit(MD_RECOVERY_RUNNING, &mddev->recovery); -> MD_RECOVERY_RUNNING can't be cleared, hence step 2 hang; This problem is not just related to dm-raid, fix it by ignoring suspended array in md_check_recovery(). And follow up patches will improve dm-raid better to frozen sync thread during suspend.
CVE-2024-26767	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fixed integer types and null check locations [why]: issues fixed: - comparison with wider integer type in loop condition which can cause infinite loops - pointer dereference before null check

CVE ID	Description
CVE-2024-26771	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: edma: Add some null pointer checks to the edma_probe devm_kasprintf() returns a pointer to dynamically allocated memory which can be NULL upon failure. Ensure the allocation was successful by checking the pointer validity.
CVE-2024-26775	In the Linux kernel, the following vulnerability has been resolved: aoe: avoid potential deadlock at set_capacity Move set_capacity() outside of the section protected by (&d->lock). To avoid possible interrupt unsafe locking scenario: CPU0 CPU1 ---- [1] lock(&bdev->bd_size_lock); local_irq_disable(); [2] lock(&d->lock); [3] lock(&bdev->bd_size_lock); <Interrupt> [4] lock(&d->lock); *** DEADLOCK *** Where [1](&bdev->bd_size_lock) hold by zram_add()->set_capacity(). [2]lock(&d->lock) hold by aoeblk_gdalloc(). And aoeblk_gdalloc() is trying to acquire [3](&bdev->bd_size_lock) at set_capacity() call. In this situation an attempt to acquire [4]lock(&d->lock) from aoe_cmd_cfg_rsp() will lead to deadlock. So the simplest solution is breaking lock dependency [2](&d->lock) -> [3](&bdev->bd_size_lock) by moving set_capacity() outside.
CVE-2024-26782	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix double-free on socket dismantle when MPTCP server accepts an incoming connection, it clones its listener socket. However, the pointer to 'inet_opt' for the new socket has the same value as the original one: as a consequence, on program exit it's possible to observe the following splat: BUG: KASAN: double-free in inet_sock_destruct+0x54f/0x8b0 Free of addr ffff888485950880 by task swapper/25/0 CPU: 25 PID: 0 Comm: swapper/25 Kdump: loaded Not tainted 6.8.0-rc1+ #609 Hardware name: Supermicro SYS-6027R-72RF/X9DRH-7TF/7F/iTF/iF, BIOS 3.0 07/26/2013 Call Trace: <IRQ> dump_stack_lvl+0x32/0x50 print_report+0xca/0x620 kasan_report_invalid_free+0x64/0x90 __kasan_slab_free+0x1aa/0x1f0 kfree+0xed/0x2e0 inet_sock_destruct+0x54f/0x8b0 __sk_destruct+0x48/0x5b0 rcu_do_batch+0x34e/0xd90 rcu_core+0x559/0xa00 __do_softirq+0x183/0x5a4 irq_exit_rcu+0x12d/0x170 sysvec_apic_timer_interrupt+0x6b/0x80 </IRQ> <TASK> asm_sysvec_apic_timer_interrupt+0x16/0x20 RIP: 0010:cpuidle_enter_state+0x175/0x300 Code: 30 00 0f 84 1f 01 00 00 83 e8 01 83 f8 ff 75 e5 48 83 c4 18 44 89 e8 5b 5d 41 5c 41 5d 41 5e 41 5f c3 cc cc cc cc cc fb 45 85 ed <0f> 89 60 ff ff 48 c1 e5 06 48 c7 43 18 00 00 00 48 83 44 2b RSP: 0018:ffff888481cf7d90 EFLAGS: 00000202 RAX: 0000000000000000 RBX: ffff88887facddc8 RCX: 0000000000000000 RDX: 1ffff110ff588b1 RSI: 0000000000000019 RDI: ffff88887fac4588 RBP: 0000000000000004 R08: 0000000000000002 R09: 0000000000043080 R10: 0009b02ea273363f R11: ffff88887fabf42b R12: ffffffff932592e0 R13: 0000000000000004 R14: 0000000000000000 R15: 00000022c880ec80 cpuidle_enter+0x4a/0xa0 do_idle+0x310/0x410 cpu_startup_entry+0x51/0x60 start_secondary+0x211/0x270 secondary_startup_64_no_verify+0x184/0x18b </TASK> Allocated by task 6853: kasan_save_stack+0x1c/0x40 kasan_save_track+0x10/0x30 __kasan_kmalloc+0xa6/0xb0 __kmalloc+0x1eb/0x450 cipso_v4_sock_setattr+0x96/0x360 netlbl_sock_setattr+0x132/0x1f0 selinux_netlbl_socket_post_create+0x6c/0x110 selinux_socket_post_create+0x37b/0x7f0 security_socket_post_create+0x63/0xb0 __sock_create+0x305/0x450 __sys_socket_create.part.23+0xbd/0x130 __sys_socket+0x37/0xb0 __x64_sys_socket+0x6f/0xb0 do_syscall_64+0x83/0x160 entry_SYSCALL_64_after_hwframe+0x6e/0x76 Freed by task 6858: kasan_save_stack+0x1c/0x40 kasan_save_track+0x10/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x12c/0x1f0 kfree+0xed/0x2e0 inet_sock_destruct+0x54f/0x8b0 __sk_destruct+0x48/0x5b0 subflow_ulp_release+0x1f0/0x250 tcp_cleanup_ulp+0x6e/0x110 tcp_v4_destroy_sock+0x5a/0x3a0 inet_csk_destroy_sock+0x135/0x390 tcp_fin+0x416/0x5c0 tcp_data_queue+0x1bc8/0x4310 tcp_rcv_state_process+0x15a3/0x47b0 tcp_v4_do_rcv+0x2c1/0x990 tcp_v4_rcv+0x41fb/0x5ed0 ip_protocol_deliver_rcu+0x6d/0x9f0 ip_local_deliver_finish+0x278/0x360 ip_local_deliver+0x182/0x2c0 ip_rcv+0xb5/0x1c0 __netif_receive_skb_one_core+0x16e/0x1b0 process_backlog+0x1e3/0x650 __napi_poll+0xa6/0x500 net_rx_action+0x740/0xbb0 __do_softirq+0x183/0x5a4 The buggy address belongs to the object at ffff888485950880 which belongs to the cache kmalloc-64 of size 64 The buggy address is located 0 bytes inside of 64-byte region [ffff888485950880, ffff8884859508c0) The buggy address belongs to the physical page: page:0000000056d1e95e refcount:1 mapcount:0 mapping:0000000000000000 index:0xffff888485950700 pfn:0x485950 flags: 0x57ffffc0000800(slab node=1 zone=2 lastcpupid=0x1fffff) page_type: 0xffffffffff() raw: 0057ffffc0000800 ffff88810004c640 ffffea00121b8ac0 dead000000000000 raw: ffff888485950700 000000000200019 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff888485950780: fa fb fb ---truncated---

CVE ID	Description
CVE-2024-26803	In the Linux kernel, the following vulnerability has been resolved: net: veth: clear GRO when clearing XDP even when down veth sets NETIF_F_GRO automatically when XDP is enabled, because both features use the same NAPI machinery. The logic to clear NETIF_F_GRO sits in veth_disable_xdp() which is called both on ndo_stop and when XDP is turned off. To avoid the flag from being cleared when the device is brought down, the clearing is skipped when IFF_UP is not set. Bringing the device down should indeed not modify its features. Unfortunately, this means that clearing is also skipped when XDP is disabled _while_ the device is down. And there's nothing on the open path to bring the device features back into sync. IOW if user enables XDP, disables it and then brings the device up we'll end up with a stray GRO flag set but no NAPI instances. We don't depend on the GRO flag on the datapath, so the datapath won't crash. We will crash (or hang), however, next time features are sync'd (either by user via ethtool or peer changing its config). The GRO flag will go away, and veth will try to disable the NAPIs. But the open path never created them since XDP was off, the GRO flag was a stray. If NAPI was initialized before we'll hang in napi_disable(). If it never was we'll crash trying to stop uninitialized hrtimer. Move the GRO flag updates to the XDP enable / disable paths, instead of mixing them with the ndo_open / ndo_close paths.
CVE-2024-26812	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Create persistent INTx handler A vulnerability exists where the eventfd for INTx signaling can be deconfigured, which unregisters the IRQ handler but still allows eventfds to be signaled with a NULL context through the SET_IRQS ioctl or through unmask irqfd if the device interrupt is pending. Ideally this could be solved with some additional locking; the igate mutex serializes the ioctl and config space accesses, and the interrupt handler is unregistered relative to the trigger, but the irqfd path runs asynchronous to those. The igate mutex cannot be acquired from the atomic context of the eventfd wake function. Disabling the irqfd relative to the eventfd registration is potentially incompatible with existing userspace. As a result, the solution implemented here moves configuration of the INTx interrupt handler to track the lifetime of the INTx context object and irq_type configuration, rather than registration of a particular trigger eventfd. Synchronization is added between the ioctl path and eventfd_signal() wrapper such that the eventfd trigger can be dynamically updated relative to in-flight interrupts or irqfd callbacks.
CVE-2024-26816	In the Linux kernel, the following vulnerability has been resolved: x86, relocs: Ignore relocations in .notes section When building with CONFIG_XEN_PV=y, .text symbols are emitted into the .notes section so that Xen can find the "startup_xen" entry point. This information is used prior to booting the kernel, so relocations are not useful. In fact, performing relocations against the .notes section means that the KASLR base is exposed since /sys/kernel/notes is world-readable. To avoid leaking the KASLR base without breaking unprivileged tools that are expecting to read /sys/kernel/notes, skip performing relocations in the .notes section. The values readable in .notes are then identical to those found in System.map.
CVE-2024-26828	In the Linux kernel, the following vulnerability has been resolved: cifs: fix underflow in parse_server_interfaces() In this loop, we step through the buffer and after each item we check if the size_left is greater than the minimum size we need. However, the problem is that "bytes_left" is type ssize_t while sizeof() is type size_t. That means that because of type promotion, the comparison is done as an unsigned and if we have negative bytes left the loop continues instead of ending.

CVE ID	Description
CVE-2024-26837	In the Linux kernel, the following vulnerability has been resolved: net: bridge: switchdev: Skip MDB replays of deferred events on offload Before this change, generation of the list of MDB events to replay would race against the creation of new group memberships, either from the IGMP/MLD snooping logic or from user configuration. While new memberships are immediately visible to walkers of br->mdb_list, the notification of their existence to switchdev event subscribers is deferred until a later point in time. So if a replay list was generated during a time that overlapped with such a window, it would also contain a replay of the not-yet-delivered event. The driver would thus receive two copies of what the bridge internally considered to be one single event. On destruction of the bridge, only a single membership deletion event was therefore sent. As a consequence of this, drivers which reference count memberships (at least DSA), would be left with orphan groups in their hardware database when the bridge was destroyed. This is only an issue when replaying additions. While deletion events may still be pending on the deferred queue, they will already have been removed from br->mdb_list, so no duplicates can be generated in that scenario. To a user this meant that old group memberships, from a bridge in which a port was previously attached, could be reanimated (in hardware) when the port joined a new bridge, without the new bridge's knowledge. For example, on an mv88e6xxx system, create a snooping bridge and immediately add a port to it: root@infix-06-0b-00:~\$ ip link add dev br0 up type bridge mcast_snooping 1 && \> ip link set dev x3 up master br0 And then destroy the bridge: root@infix-06-0b-00:~\$ ip link del dev br0 root@infix-06-0b-00:~\$ mvls atu ADDRESS FID STATE Q F 0 1 2 3 4 5 6 7 8 9 a DEV:0 Marvell 88E6393X 33:33:00:00:00:6a 1 static -- 0 33:33:ff:87:e4:3f 1 static -- 0 ff:ff:ff:ff:ff:ff 1 static -- 0 1 2 3 4 5 6 7 8 9 a root@infix-06-0b-00:~\$ The two IPv6 groups remain in the hardware database because the port (x3) is notified of the host's membership twice: once via the original event and once via a replay. Since only a single delete notification is sent, the count remains at 1 when the bridge is destroyed. Then add the same port (or another port belonging to the same hardware domain) to a new bridge, this time with snooping disabled: root@infix-06-0b-00:~\$ ip link add dev br1 up type bridge mcast_snooping 0 && \> ip link set dev x3 up master br1 All multicast, including the two IPv6 groups from br0, should now be flooded, according to the policy of br1. But instead the old memberships are still active in the hardware database, causing the switch to only forward traffic to those groups towards the CPU (port 0). Eliminate the race in two steps: 1. Grab the write-side lock of the MDB while generating the replay list. This prevents new memberships from showing up while we are generating the replay list. But it leaves the scenario in which a deferred event was already generated, but not delivered, before we grabbed the lock. Therefore: 2. Make sure that no deferred version of a replay event is already enqueued to the switchdev deferred queue, before adding it to the replay list, when replaying additions.
CVE-2024-26838	In the Linux kernel, the following vulnerability has been resolved: RDMA/irdma: Fix KASAN issue with tasklet KASAN testing revealed the following issue associated with freeing an IRQ. [50006.466686] Call Trace: [50006.466691] <IRQ> [50006.489538] dump_stack+0x5c/0x80 [50006.493475] print_address_description.constprop.6+0x1a/0x150 [50006.499872] ? irdma_sc_process_ceq+0x483/0x790 [irdma] [50006.505742] ? irdma_sc_process_ceq+0x483/0x790 [irdma] [50006.511644] kasan_report.cold.11+0x7f/0x118 [50006.516572] ? irdma_sc_process_ceq+0x483/0x790 [irdma] [50006.522473] irdma_sc_process_ceq+0x483/0x790 [irdma] [50006.528232] irdma_process_ceq+0xb2/0x400 [irdma] [50006.533601] ? irdma_hw_flush_wqes_callback+0x370/0x370 [irdma] [50006.540298] irdma_ceq_dpc+0x44/0x100 [irdma] [50006.545306] tasklet_action_common.isra.14+0x148/0x2c0 [50006.551096] __do_softirq+0x1d0/0xaf8 [50006.555396] irq_exit_rcu+0x219/0x260 [50006.559670] irq_exit+0xa/0x20 [50006.563320] smp_apic_timer_interrupt+0x1bf/0x690 [50006.568645] apic_timer_interrupt+0xf/0x20 [50006.573341] </IRQ> The issue is that a tasklet could be pending on another core racing the delete of the irq. Fix by insuring any scheduled tasklet is killed after deleting the irq.
CVE-2024-26845	In the Linux kernel, the following vulnerability has been resolved: scsi: target: core: Add TMF to tmr_list handling An abort that is responded to by iSCSI itself is added to tmr_list but does not go to target core. A LUN_RESET that goes through tmr_list takes a refcounter on the abort and waits for completion. However, the abort will be never complete because it was not started in target core. Unable to locate ITT: 0x05000000 on CID: 0 Unable to locate RefTaskTag: 0x05000000 on CID: 0. wait_for_tasks: Stopping tmf LUN_RESET with tag 0x0 ref_task_tag 0x0 i_state 34 t_state ISTATE_PROCESSING refcnt 2 transport_state active,stop,fabric_stop wait for tasks: tmf LUN_RESET with tag 0x0 ref_task_tag 0x0 i_state 34 t_state ISTATE_PROCESSING refcnt 2 transport_state active,stop,fabric_stop ... INFO: task kworker/0:2:49 blocked for more than 491 seconds. task:kworker/0:2 state:D stack: 0 pid: 49 ppid: 2 flags:0x00000800 Workqueue: events target_tmr_work [target_core_mod] Call Trace: __switch_to+0x2c4/0x470 _schedule+0x314/0x1730 schedule+0x64/0x130 schedule_timeout+0x168/0x430 wait_for_completion+0x140/0x270 target_put_cmd_and_wait+0x64/0xb0 [target_core_mod] core_tmr_lun_reset+0x30/0xa0 [target_core_mod] target_tmr_work+0xc8/0x1b0 [target_core_mod] process_one_work+0x2d4/0x5d0 worker_thread+0x78/0x6c0 To fix this, only add abort to tmr_list if it will be handled by target core.

CVE ID	Description
CVE-2024-26857	In the Linux kernel, the following vulnerability has been resolved: geneve: make sure to pull inner header in geneve_rx() syzbot triggered a bug in geneve_rx() [1] Issue is similar to the one I fixed in commit 8d975c15c0cd ("ip6_tunnel: make sure to pull inner header in __ip6_tnl_rcv()") We have to save skb->network_header in a temporary variable in order to be able to recompute the network_header pointer after a pskb_inet_may_pull() call. pskb_inet_may_pull() makes sure the needed headers are in skb->head. [1] BUG: KMSAN: uninit-value in IP_ECN_decapsulate include/net/inet_ecn.h:302 [inline] BUG: KMSAN: uninit-value in geneve_rx drivers/net/geneve.c:279 [inline] BUG: KMSAN: uninit-value in geneve_udp_encap_rcv+0x36f9/0x3c10 drivers/net/geneve.c:391 IP_ECN_decapsulate include/net/inet_ecn.h:302 [inline] geneve_rx drivers/net/geneve.c:279 [inline] geneve_udp_encap_rcv+0x36f9/0x3c10 drivers/net/geneve.c:391 udp_queue_rcv_one_skb+0x1d39/0x1f20 net/ipv4/udp.c:2108 udp_queue_rcv_skb+0x6ae/0x6e0 net/ipv4/udp.c:2186 udp_unicast_rcv_skb+0x184/0x4b0 net/ipv4/udp.c:2346 __udp4_lib_rcv+0x1c6b/0x3010 net/ipv4/udp.c:2422 udp_rcv+0x7d/0xa0 net/ipv4/udp.c:2604 ip_protocol_deliver_rcu+0x264/0x1300 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x2b8/0x440 net/ipv4/ip_input.c:233 NF_HOOK include/linux/netfilter.h:314 [inline] ip_local_deliver+0x21f/0x490 net/ipv4/ip_input.c:254 dst_input include/net/dst.h:461 [inline] ip_rcv_finish net/ipv4/ip_input.c:449 [inline] NF_HOOK include/linux/netfilter.h:314 [inline] ip_rcv+0x46f/0x760 net/ipv4/ip_input.c:569 __netif_receive_skb_one_core net/core/dev.c:5534 [inline] __netif_receive_skb+0x1a6/0x5a0 net/core/dev.c:5648 process_backlog+0x480/0x8b0 net/core/dev.c:5976 __napi_poll+0xe3/0x980 net/core/dev.c:6576 napi_poll net/core/dev.c:6645 [inline] net_rx_action+0x8b8/0x1870 net/core/dev.c:6778 __do_softirq+0x1b7/0x7c5 kernel/softirq.c:553 do_softirq+0x9a/0xf0 kernel/softirq.c:454 __local_bh_enable_ip+0x9b/0xa0 kernel/softirq.c:381 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:820 [inline] __dev_queue_xmit+0x2768/0x51c0 net/core/dev.c:4378 dev_queue_xmit include/linux/netdevice.h:3171 [inline] packet_xmit+0x9c/0x6b0 net/packet/af_packet.c:276 packet_snd net/packet/af_packet.c:3081 [inline] packet_sendmsg+0x8aef/0x9f10 net/packet/af_packet.c:3113 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] __sys_sendto+0x735/0xa10 net/socket.c:2191 __do_sys_sendto net/socket.c:2203 [inline] __se_sys_sendto net/socket.c:2199 [inline] __x64_sys_sendto+0x125/0x1c0 net/socket.c:2199 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcf/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b Uninit was created at: slab_post_alloc_hook mm/slub.c:3819 [inline] slab_alloc_node mm/slub.c:3860 [inline] kmem_cache_alloc_node+0x5cb/0xabc mm/slub.c:3903 kmemalloc_reserve+0x13d/0x4a0 net/core/skbuff.c:560 __alloc_skb+0x352/0x790 net/core/skbuff.c:651 alloc_skb include/linux/skbuff.h:1296 [inline] alloc_skb_with_frags+0xc8/0xbd0 net/core/skbuff.c:6394 sock_alloc_send_pskb+0xa80/0xbf0 net/core/sock.c:2783 packet_alloc_skb net/packet/af_packet.c:2930 [inline] packet_snd net/packet/af_packet.c:3024 [inline] packet_sendmsg+0x70c2/0x9f10 net/packet/af_packet.c:3113 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] __sys_sendto+0x735/0xa10 net/socket.c:2191 __do_sys_sendto net/socket.c:2203 [inline] __se_sys_sendto net/socket.c:2199 [inline] __x64_sys_sendto+0x125/0x1c0 net/socket.c:2199 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcf/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b

CVE ID	Description
CVE-2024-26882	In the Linux kernel, the following vulnerability has been resolved: net: ip_tunnel: make sure to pull inner header in ip_tunnel_rcv() Apply the same fix than ones found in : 8d975c15c0cd ("ip6_tunnel: make sure to pull inner header in __ip6_tnl_rcv()") 1ca1ba465e55 ("geneve: make sure to pull inner header in geneve_rx()") We have to save skb->network_header in a temporary variable in order to be able to recompute the network_header pointer after a pskb_inet_may_pull() call. pskb_inet_may_pull() makes sure the needed headers are in skb->head. syzbot reported: BUG: KMSAN: uninit-value in __INET_ECN_decapsulate include/net/inet_ecn.h:253 [inline] BUG: KMSAN: uninit-value in INET_ECN_decapsulate include/net/inet_ecn.h:275 [inline] BUG: KMSAN: uninit-value in IP_ECN_decapsulate include/net/inet_ecn.h:302 [inline] BUG: KMSAN: uninit-value in ip_tunnel_rcv+0xed9/0x2ed0 net/ipv4/ip_tunnel.c:409 __INET_ECN_decapsulate include/net/inet_ecn.h:253 [inline] INET_ECN_decapsulate include/net/inet_ecn.h:275 [inline] IP_ECN_decapsulate include/net/inet_ecn.h:302 [inline] ip_tunnel_rcv+0xed9/0x2ed0 net/ipv4/ip_tunnel.c:409 __ipgre_rcv+0x9bc/0xabc0 net/ipv4/ip_gre.c:389 ipgre_rcv net/ipv4/ip_gre.c:411 [inline] gre_rcv+0x423/0x19f0 net/ipv4/ip_gre.c:447 gre_rcv+0x2a4/0x390 net/ipv4/gre_demux.c:163 ip_protocol_deliver_rcu+0x264/0x1300 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x2b8/0x440 net/ipv4/ip_input.c:233 NF_HOOK include/linux/netfilter.h:314 [inline] ip_local_deliver+0x21f/0x490 net/ipv4/ip_input.c:254 dst_input include/net/dst.h:461 [inline] ip_rcv_finish net/ipv4/ip_input.c:449 [inline] NF_HOOK include/linux/netfilter.h:314 [inline] ip_rcv+0x46f/0x760 net/ipv4/ip_input.c:569 __netif_receive_skb_one_core net/core/dev.c:5534 [inline] __netif_receive_skb+0x1a6/0x5a0 net/core/dev.c:5648 netif_receive_skb_internal net/core/dev.c:5734 [inline] netif_receive_skb+0x58/0x660 net/core/dev.c:5793 tun_rx_batched+0x3ee/0x980 drivers/net/tun.c:1556 tun_get_user+0x53b9/0x66e0 drivers/net/tun.c:2009 tun_chr_write_iter+0x3af/0x5d0 drivers/net/tun.c:2055 call_write_iter include/linux/fs.h:2087 [inline] new_sync_write fs/read_write.c:497 [inline] vfs_write+0xb6b/0x1520 fs/read_write.c:590 ksys_write+0x20f/0x4c0 fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __x64_sys_write+0x93/0xd0 fs/read_write.c:652 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcf/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b Uninit was created at: __alloc_pages+0x9a6/0xe00 mm/page_alloc.c:4590 alloc_pages_mpol+0x62b/0x9d0 mm/mempolicy.c:2133 alloc_pages+0x1be/0x1e0 mm/mempolicy.c:2204 skb_page_frag_refill+0x2bf/0x7c0 net/core/sock.c:2909 tun_build_skb drivers/net/tun.c:1686 [inline] tun_get_user+0xe0a/0x66e0 drivers/net/tun.c:1826 tun_chr_write_iter+0x3af/0x5d0 drivers/net/tun.c:2055 call_write_iter include/linux/fs.h:2087 [inline] new_sync_write fs/read_write.c:497 [inline] vfs_write+0xb6b/0x1520 fs/read_write.c:590 ksys_write+0x20f/0x4c0 fs/read_write.c:643 __do_sys_write fs/read_write.c:655 [inline] __se_sys_write fs/read_write.c:652 [inline] __x64_sys_write+0x93/0xd0 fs/read_write.c:652 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcf/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b
CVE-2024-26883	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix stackmap overflow check on 32-bit arches The stackmap code relies on roundup_pow_of_two() to compute the number of hash buckets, and contains an overflow check by checking if the resulting value is 0. However, on 32-bit arches, the roundup code itself can overflow by doing a 32-bit left-shift of an unsigned long value, which is undefined behaviour, so it is not guaranteed to truncate neatly. This was triggered by syzbot on the DEVMAP_HASH type, which contains the same check, copied from the hashtable code. The commit in the fixes tag actually attempted to fix this, but the fix did not account for the UB, so the fix only works on CPUs where an overflow does result in a neat truncation to zero, which is not guaranteed. Checking the value before rounding does not have this problem.
CVE-2024-26886	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: af_bluetooth: Fix deadlock Attempting to do sock_lock on .recvmsg may cause a deadlock as shown below, so instead of using sock_sock this uses sk_receive_queue.lock on bt_sock_ioctl to avoid the UAF: INFO: task kworker/u9:1:121 blocked for more than 30 seconds. Not tainted 6.7.6-lemon #183 Workqueue: hci0 hci_rx_work Call Trace: <TASK> __schedule+0x37d/0xa00 schedule+0x32/0xe0 __lock_sock+0x68/0xa0 ? __pfx_autoremove_wake_function+0x10/0x10 lock_sock_nested+0x43/0x50 l2cap_sock_recv_cb+0x21/0xa0 l2cap_recv_frame+0x55b/0x30a0 ? psi_task_switch+0xeb/0x270 ? finish_task_switch.isra.0+0x93/0x2a0 hci_rx_work+0x33a/0x3f0 process_one_work+0x13a/0x2f0 worker_thread+0x2f0/0x410 ? __pfx_worker_thread+0x10/0x10 kthread+0xe0/0x110 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2c/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1b/0x30 </TASK>

CVE ID	Description
CVE-2024-26900	In the Linux kernel, the following vulnerability has been resolved: md: fix kmemleak of rdev->serial If kobject_add() is fail in bind_rdev_to_array(), 'rdev->serial' will be alloc not be freed, and kmemleak occurs. unreferenced object 0xffff88815a350000 (size 49152): comm "mdadm", pid 789, jiffies 4294716910 hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace (crc f773277a): [<0000000058b0a453>] kmemleak_alloc+0x61/0xe0 [<00000000366adf14>] __kmalloclarge_node+0x15e/0x270 [<000000002e82961b>] __kmalloclnode.cold+0x11/0x7f [<00000000f206d60a>] kvmalloc_node+0x74/0x150 [<0000000034bf3363>] rdev_init_serial+0x67/0x170 [<0000000010e08fe9>] mddev_create_serial_pool+0x62/0x220 [<00000000c3837bf0>] bind_rdev_to_array+0x2af/0x630 [<0000000073c28560>] md_add_new_disk+0x400/0x9f0 [<000000000770e30ff>] md_ioctl+0x15bf/0x1c10 [<0000000006cfab718>] blkdev_ioctl+0x191/0x3f0 [<00000000085086a11>] vfs_ioctl+0x22/0x60 [<0000000018b656fe>] __x64_sys_ioctl+0xba/0xe0 [<00000000e54e675e>] do_syscall_64+0x71/0x150 [<000000008b0ad622>] entry_SYSCALL_64_after_hwframe+0x6c/0x74
CVE-2024-26902	In the Linux kernel, the following vulnerability has been resolved: perf: RISCv: Fix panic on pmu overflow handler (1 <= idx) of int is not desired when setting bits in unsigned long overflowed_ctrs, use BIT() instead. This panic happens when running 'perf record -e branches' on sophgo sg2042. [273.311852] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000098 [273.320851] Oops [#1] [273.323179] Modules linked in: [273.326303] CPU: 0 PID: 1475 Comm: perf Not tainted 6.6.0-rc3+ #9 [273.332521] Hardware name: Sophgo Mango (DT) [273.336878] epc : riscv_pmu_ctr_get_width_mask+0x8/0x62 [273.342291] ra : pmu_sbi_ovf_handler+0x2e0/0x34e [273.347091] epc : ffffffff80aecd98 ra : ffffffff80aee056 sp : ffffffff6e36928b0 [273.354454] gp : ffffffff821f82d0 tp : ffffffff90c353200 t0 : 0000002ade4f9978 [273.361815] t1 : 0000000000504d55 t2 : ffffffff8016cd8c s0 : ffffffff6e3692a70 [273.369180] s1 : 0000000000000020 a0 : 0000000000000000 a1 : 00001a8e81800000 [273.376540] a2 : 0000003c00070198 a3 : 0000003c00db75a4 a4 : 0000000000000015 [273.383901] a5 : ffffffff7ff8804b0 a6 : 0000000000000015 a7 : 0000000000000002a [273.391327] s2 : 000000000000ffff s3 : 0000000000000000 s4 : ffffffff7ff8803b0 [273.398773] s5 : 0000000000504d55 s6 : ffffffff905069800 s7 : ffffffff821fe210 [273.406139] s8 : 0000000007ffffff s9 : ffffffff7ff8803b0 s10 : ffffffff903f29098 [273.413660] s11 : 0000000008000000 t3 : 0000000000000003 t4 : ffffffff8017a0ca [273.421022] t5 : ffffffff8023cfc2 t6 : ffffffff9040780e8 [273.426437] status: 0000000200000100 badaddr: 0000000000000098 cause: 000000000000000d [273.434512] [<fffffff80aecd98>] riscv_pmu_ctr_get_width_mask+0x8/0x62 [273.441169] [<fffffff80076bd8>] handle_percpu_devid_irq+0x98/0x1ee [273.447562] [<fffffff80071158>] generic_handle_domain_irq+0x28/0x36 [273.454151] [<fffffff8047a99a>] riscv_intc_irq+0x36/0x4e [273.459659] [<fffffff80c944de>] handle_riscv_irq+0x4a/0x74 [273.465442] [<fffffff80c94c48>] do_irq+0x62/0x92 [273.470360] Code: 0420 60a2 6402 5529 0141 8082 0013 0000 0013 0000 (6d5c) b783 [273.477921] ---[end trace 0000000000000000]--- [273.482630] Kernel panic - not syncing: Fatal exception in interrupt
CVE-2024-26903	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: rfcomm: Fix null-ptr-deref in rfcomm_check_security During our fuzz testing of the connection and disconnection process at the RFCOMM layer, we discovered this bug. By comparing the packets from a normal connection and disconnection process with the testcase that triggered a KASAN report. We analyzed the cause of this bug as follows: 1. In the packets captured during a normal connection, the host sends a 'Read Encryption Key Size' type of 'HCI_CMD' packet (Command Opcode: 0x1408) to the controller to inquire the length of encryption key. After receiving this packet, the controller immediately replies with a Command Complete packet (Event Code: 0x0e) to return the Encryption Key Size. 2. In our fuzz test case, the timing of the controller's response to this packet was delayed to an unexpected point: after the RFCOMM and L2CAP layers had disconnected but before the HCI layer had disconnected. 3. After receiving the Encryption Key Size Response at the time described in point 2, the host still called the rfcomm_check_security function. However, by this time 'struct l2cap_conn *conn = l2cap_pi(sk)->chan->conn;' had already been released, and when the function executed 'return hci_conn_security(conn->hcon, d->sec_level, auth_type, d->out);', specifically when accessing 'conn->hcon', a null-ptr-deref error occurred. To fix this bug, check if 'sk->sk_state' is BT_CLOSED before calling rfcomm_recv_frame in rfcomm_process_rx.
CVE-2024-26915	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Reset IH OVERFLOW_CLEAR bit Allows us to detect subsequent IH ring buffer overflows as well.
CVE-2024-26916	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/amd: flush any delayed gfxoff on suspend entry" commit ab4750332dbe ("drm/amdgpu/sdma5.2: add begin/end_use ring callbacks") caused GFXOFF control to be used more heavily and the codepath that was removed from commit 0dee72639533 ("drm/amd: flush any delayed gfxoff on suspend entry") now can be exercised at suspend again. Users report that by using GNOME to suspend the lockscreen trigger will cause SDMA traffic and the system can deadlock. This reverts commit 0dee72639533fea833eaf838bc80962df886c8.

CVE ID	Description
CVE-2024-26922	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: validate the parameters of bo mapping operations more clearly Verify the parameters of amdgpu_vm_bo_(map/replace_map/clearing_mappings) in one common place.
CVE-2024-26926	In the Linux kernel, the following vulnerability has been resolved: binder: check offset alignment in binder_get_object() Commit 6d98eb95b450 ("binder: avoid potential data leakage when copying txn") introduced changes to how binder objects are copied. In doing so, it unintentionally removed an offset alignment check done through calls to binder_alloc_copy_from_buffer() -> check_buffer(). These calls were replaced in binder_get_object() with copy_from_user(), so now an explicit offset alignment check is needed here. This avoids later complications when unwinding the objects gets harder. It is worth noting this check existed prior to commit 7a67a39320df ("binder: add function to copy binder object from buffer"), likely removed due to redundancy at the time.
CVE-2024-26927	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Add some bounds checking to firmware data Smatch complains about "head->full_size - head->header_size" can underflow. To some extent, we're always going to have to trust the firmware a bit. However, it's easy enough to add a check for negatives, and let's add a upper bounds check as well.
CVE-2024-26928	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in cifs_debug_files_proc_show() Skip sessions that are being teared down (status == SES_EXITING) to avoid UAF.
CVE-2024-26930	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix double free of the ha->vp_map pointer Coverity scan reported potential risk of double free of the pointer ha->vp_map. ha->vp_map was freed in qla2x00_mem_alloc(), and again freed in function qla2x00_mem_free(ha). Assign NULL to vp_map and kfree take care of NULL.
CVE-2024-26931	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix command flush on cable pull System crash due to command failed to flush back to SCSI layer. BUG: unable to handle kernel NULL pointer dereference at 0000000000000000 PGD 0 P4D 0 Oops: 0000 [#1] SMP NOPTI CPU: 27 PID: 793455 Comm: kworker/u130:6 Kdump: loaded Tainted: G OE ----- - - 4.18.0-372.9.1.el8.x86_64 #1 Hardware name: HPE ProLiant DL360 Gen10/ProLiant DL360 Gen10, BIOS U32 09/03/2021 Workqueue: nvme-wq nvme_fc_connect_ctrl_work [nvme_fc] RIP: 0010: __wake_up_common+0x4c/0x190 Code: 24 10 4d 85 c9 74 0a 41 f6 01 04 0f 85 9d 00 00 00 48 8b 43 08 48 83 c3 08 4c 8d 48 e8 49 8d 41 18 48 39 c3 0f 84 f0 00 00 00 <49> 8b 41 18 89 54 24 08 31 ed 4c 8d 70 e8 45 8b 29 41 f6 c5 04 75 RSP: 0018:ffff95f3e0cb7cd0 EFLAGS: 00010086 RAX: 0000000000000000 RBX: ffff8b08d3b26328 RCX: 0000000000000000 RDX: 0000000000000001 RSI: 0000000000000003 RDI: ffff8b08d3b26320 RBP: 0000000000000001 R08: 0000000000000000 R09: ffffffff8b08d3b2 R10: 0000000000000000 R11: ffff95f3e0cb7a60 R12: ffff95f3e0cb7d20 R13: 0000000000000003 R14: 0000000000000000 R15: 0000000000000000 FS: 0000000000000000(0000) GS: ffff8b2fdf6c0000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 0000002f1e410002 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: __wake_up_common_lock+0x7c/0xc0 qla_nvme_ls_req +0x355/0x4c0 [qla2xxx] qla2xxx [0000:12:00.1]-f084:3: qlt_free_session_done: se_sess 0000000000000000 / sess ffff8ae1407ca000 from port 21:32:00:02:ac:07:ee:b8 loop_id 0x02 s_id 01:02:00 logout 1 keep 0 els_logo 0 ? __nvme_fc_send_ls_req+0x260/0x380 [nvme_fc] qla2xxx [0000:12:00.1]-207d:3: FCPort 21:32:00:02:ac:07:ee:b8 state transitioned from ONLINE to LOST - portid=010200. ? nvme_fc_send_ls_req.constprop.42+0x1a/0x45 [nvme_fc] qla2xxx [0000:12:00.1]-2109:3: qla2x00_schedule_rport_del 21320002ac07eeb8. rport ffff8ae598122000 roles 1 ? nvme_fc_connect_ctrl_work.cold.63+0x1e3/0xa7d [nvme_fc] qla2xxx [0000:12:00.1]-f084:3: qlt_free_session_done: se_sess 0000000000000000 / sess ffff8ae14801e000 from port 21:32:01:02:ad:f7:ee:b8 loop_id 0x04 s_id 01:02:01 logout 1 keep 0 els_logo 0 ? __switch_to +0x10c/0x450 ? process_one_work+0x1a7/0x360 qla2xxx [0000:12:00.1]-207d:3: FCPort 21:32:01:02:ad:f7:ee:b8 state transitioned from ONLINE to LOST - portid=010201. ? worker_thread+0x1ce/0x390 ? create_worker+0x1a0/0x1a0 qla2xxx [0000:12:00.1]-2109:3: qla2x00_schedule_rport_del 21320102adf7eeb8. rport ffff8ae3b2312800 roles 70 ? kthread +0x10a/0x120 qla2xxx [0000:12:00.1]-2112:3: qla_nvme_unregister_remote_port: unregister remoteport on ffff8ae14801e000 21320102adf7eeb8 ? set_kthread_struct+0x40/0x40 qla2xxx [0000:12:00.1]-2110:3: remoteport_delete of ffff8ae14801e000 21320102adf7eeb8 completed. ? ret_from_fork+0x1f/0x40 qla2xxx [0000:12:00.1]-f086:3: qlt_free_session_done: waiting for sess ffff8ae14801e000 logout The system was under memory stress where driver was not able to allocate an SRB to carry out error recovery of cable pull. The failure to flush causes upper layer to start modifying scsi_cmnd. When the system frees up some memory, the subsequent cable pull trigger another command flush. At this point the driver access a null pointer when attempting to DMA unmap the SGL. Add a check to make sure commands are flush back on session tear down to prevent the null pointer access.

CVE ID	Description
CVE-2024-26938	In the Linux kernel, the following vulnerability has been resolved: drm/i915/bios: Tolerate devdata==NULL in intel_bios_encoder_supports_dp_dual_mode() If we have no VBT, or the VBT didn't declare the encoder in question, we won't have the 'devdata' for the encoder. Instead of oopsing just bail early. We won't be able to tell whether the port is DP++ or not, but so be it. (cherry picked from commit 26410896206342c8a80d2b027923e9ee7d33b733)
CVE-2024-26962	In the Linux kernel, the following vulnerability has been resolved: dm-raid456, md/raid456: fix a deadlock for dm-raid456 while io concurrent with reshape For raid456, if reshape is still in progress, then IO across reshape position will wait for reshape to make progress. However, for dm-raid, in following cases reshape will never make progress hence IO will hang: 1) the array is read-only; 2) MD_RECOVERY_WAIT is set; 3) MD_RECOVERY_FROZEN is set; After commit c467e97f079f ("md/raid6: use valid sector values to determine if an I/O should wait on the reshape") fix the problem that IO across reshape position doesn't wait for reshape, the dm-raid test shell/lvconvert-raid-reshape.sh start to hang: [root@fedora ~]# cat /proc/979/stack [<0>] wait_woken+0x7d/0x90 [<0>] raid5_make_request+0x929/0x1d70 [raid456] [<0>] md_handle_request+0xc2/0x3b0 [md_mod] [<0>] raid_map+0x2c/0x50 [dm_raid] [<0>] __map_bio+0x251/0x380 [dm_mod] [<0>] dm_submit_bio+0x1f0/0x760 [dm_mod] [<0>] __submit_bio+0xc2/0x1c0 [<0>] submit_bio_noacct_nocheck+0x17f/0x450 [<0>] submit_bio_noacct+0x2bc/0x780 [<0>] submit_bio+0x70/0xc0 [<0>] mpage_readahead+0x169/0x1f0 [<0>] blkdev_readahead+0x18/0x30 [<0>] read_pages+0x7c/0x3b0 [<0>] page_cache_ra_unbounded+0x1ab/0x280 [<0>] force_page_cache_ra+0x9e/0x130 [<0>] page_cache_sync_ra+0x3b/0x110 [<0>] filemap_get_pages+0x143/0xa30 [<0>] filemap_read+0xdc/0x4b0 [<0>] blkdev_read_iter+0x75/0x200 [<0>] vfs_read+0x272/0x460 [<0>] ksys_read+0x7a/0x170 [<0>] __x64_sys_read+0x1c/0x30 [<0>] do_syscall_64+0xc6/0x230 [<0>] entry_SYSCALL_64_after_hwframe+0x6c/0x74 This is because reshape can't make progress. For md/raid, the problem doesn't exist because register new sync_thread doesn't rely on the IO to be done any more: 1) If array is read-only, it can switch to read-write by ioctl/sysfs; 2) md/raid never set MD_RECOVERY_WAIT; 3) If MD_RECOVERY_FROZEN is set, mddev_suspend() doesn't hold 'reconfig_mutex', hence it can be cleared and reshape can continue by sysfs api 'sync_action'. However, I'm not sure yet how to avoid the problem in dm-raid yet. This patch on the one hand make sure raid_message() can't change sync_thread() through raid_message() after presuspend(), on the other hand detect the above 3 cases before wait for IO do be done in dm_suspend(), and let dm-raid requeue those IO.
CVE-2024-26982	In the Linux kernel, the following vulnerability has been resolved: Squashfs: check the inode number is not the invalid value of zero Syskiller has produced an out of bounds access in fill_meta_index(). That out of bounds access is ultimately caused because the inode has an inode number with the invalid value of zero, which was not checked. The reason this causes the out of bounds access is due to following sequence of events: 1. Fill_meta_index() is called to allocate (via empty_meta_index()) and fill a metadata index. It however suffers a data read error and aborts, invalidating the newly returned empty metadata index. It does this by setting the inode number of the index to zero, which means unused (zero is not a valid inode number). 2. When fill_meta_index() is subsequently called again on another read operation, locate_meta_index() returns the previous index because it matches the inode number of 0. Because this index has been returned it is expected to have been filled, and because it hasn't been, an out of bounds access is performed. This patch adds a sanity check which checks that the inode number is not zero when the inode is created and returns -EINVAL if it is. [phillip@squashfs.org.uk: whitespace fix]

CVE ID	Description
CVE-2024-26984	In the Linux kernel, the following vulnerability has been resolved: nouveau: fix instmem race condition around ptr stores Running a lot of VK CTS in parallel against nouveau, once every few hours you might see something like this crash. BUG: kernel NULL pointer dereference, address: 0000000000000008 PGD 8000000114e6e067 P4D 8000000114e6e067 PUD 109046067 PMD 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 7 PID: 53891 Comm: deqp-vk Not tainted 6.8.0-rc6+ #27 Hardware name: Gigabyte Technology Co., Ltd. Z390 I AORUS PRO WIFI/Z390 I AORUS PRO WIFI-CF, BIOS F8 11/05/2021 RIP: 0010:gp100_vmm_pgt_mem+0xe3/0x180 [nouveau] Code: c7 48 01 c8 49 89 45 58 85 d2 0f 84 95 00 00 00 41 0f b7 46 12 49 8b 7e 08 89 da 42 8d 2c f8 48 8b 47 08 41 83 c7 01 48 89 ee <48> 8b 40 08 ff d0 0f 1f 00 49 8b 7e 08 48 89 d9 48 8d 75 04 48 c1 RSP: 0000:ffffac20c5857838 EFLAGS: 00010202 RAX: 0000000000000000 RBX: 00000000004d8001 RCX: 0000000000000001 RDX: 00000000004d8001 RSI: 000000000000006d8 RDI: ffffa07afe332180 RBP: 000000000000006d8 R08: ffffac20c5857ad0 R09: 000000000fffff10 R10: 0000000000000001 R11: ffffa07af27e2de0 R12: 0000000000000001c R13: ffffac20c5857ad0 R14: ffffa07a96fe9040 R15: 000000000000001c FS: 00007fe395eed7c0(0000) GS:ffffa07e2c980000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000008 CR3: 000000011febe001 CR4: 00000000003706f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: ... ? gp100_vmm_pgt_mem+0xe3/0x180 [nouveau] ? gp100_vmm_pgt_mem+0x37/0x180 [nouveau] nvkm_vmm_iter+0x351/0xa20 [nouveau] ? __pfx_nvkm_vmm_ref_ptes+0x10/0x10 [nouveau] ? __pfx_gp100_vmm_pgt_mem+0x10/0x10 [nouveau] ? __pfx_gp100_vmm_pgt_mem+0x10/0x10 [nouveau] ? __lock_acquire+0x3ed/0x2170 ? __pfx_gp100_vmm_pgt_mem+0x10/0x10 [nouveau] nvkm_vmm_ptes_get_map+0xc2/0x100 [nouveau] ? __pfx_nvkm_vmm_ref_ptes+0x10/0x10 [nouveau] ? __pfx_gp100_vmm_pgt_mem+0x10/0x10 [nouveau] nvkm_vmm_map_locked +0x224/0x3a0 [nouveau] Adding any sort of useful debug usually makes it go away, so I hand wrote the function in a line, and debugged the asm. Every so often pt->memory->ptrs is NULL. This ptrs ptr is set in the nv50_instobj_acquire called from nvkm_kmap. If Thread A and Thread B both get to nv50_instobj_acquire around the same time, and Thread A hits the refcount_set line, and in lockstep thread B succeeds at refcount_inc_not_zero, there is a chance the ptrs value won't have been stored since refcount_set is unordered. Force a memory barrier here, I picked smp_mb, since we want it on all CPUs and it's write followed by a read. v2: use paired smp_rmb/smp_wmb.
CVE-2024-26999	In the Linux kernel, the following vulnerability has been resolved: serial/pmac_zilog: Remove flawed mitigation for rx irq flood The mitigation was intended to stop the irq completely. That may be better than a hard lock-up but it turns out that you get a crash anyway if you're using pmac_zilog as a serial console: ttyPZ0: pmz: rx irq flood ! BUG: spinlock recursion on CPU#0, swapper/0 That's because the pr_err() call in pmz_receive_chars() results in pmz_console_write() attempting to lock a spinlock already locked in pmz_interrupt(). With CONFIG_DEBUG_SPINLOCK=y, this produces a fatal BUG splat. The spinlock in question is the one in struct uart_port. Even when it's not fatal, the serial port rx function ceases to work. Also, the iteration limit doesn't play nicely with QEMU, as can be seen in the bug report linked below. A web search for other reports of the error message "pmz: rx irq flood" didn't produce anything. So I don't think this code is needed any more. Remove it.

CVE ID	Description
CVE-2024-27004	In the Linux kernel, the following vulnerability has been resolved: clk: Get runtime PM before walking tree during disable_unused Doug reported [1] the following hung task: INFO: task swapper/0:1 blocked for more than 122 seconds. Not tainted 5.15.149-21875-gf795ebc40eb8 #1 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:swapper/0 state:D stack: 0 pid: 1 ppid: 0 flags:0x00000008 Call trace: __switch_to+0xf4/0x1f4 __schedule+0x418/0xb80 schedule+0x5c/0x10c rpm_resume+0xe0/0x52c rpm_resume+0x178/0x52c __pm_runtime_resume+0x58/0x98 clk_pm_runtime_get+0x30/0xb0 clk_disable_unused_subtree+0x58/0x208 clk_disable_unused_subtree+0x38/0x208 clk_disable_unused_subtree+0x38/0x208 clk_disable_unused_subtree+0x38/0x208 clk_disable_unused_subtree+0x38/0x208 do_initcall_level+0xa4/0x148 do_initcalls+0x5c/0x9c do_basic_setup+0x24/0x30 kernel_init_freeable+0xec/0x164 kernel_init+0x28/0x120 ret_from_fork+0x10/0x20 INFO: task kworker/u16:0:9 blocked for more than 122 seconds. Not tainted 5.15.149-21875-gf795ebc40eb8 #1 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:kworker/u16:0 state:D stack: 0 pid: 9 ppid: 2 flags:0x00000008 Workqueue: events_unbound deferred_probe_work_func Call trace: __switch_to+0xf4/0x1f4 __schedule+0x418/0xb80 schedule+0x5c/0x10c schedule_preempt_disabled+0x2c/0x48 __mutex_lock+0x238/0x488 __mutex_lock_slowpath+0x1c/0x28 mutex_lock+0x50/0x74 clk_prepare_lock+0x7c/0x9c clk_core_prepare_lock+0x20/0x44 clk_prepare+0x24/0x30 clk_bulk_prepare+0x40/0xb0 mdss_runtime_resume+0x54/0x1c8 pm_generic_runtime_resume+0x30/0x44 __genpd_runtime_resume+0x68/0x7c genpd_runtime_resume+0x108/0x1f4 __rpm_callback+0x84/0x144 rpm_callback+0x30/0x88 rpm_resume+0x1f4/0x52c rpm_resume+0x178/0x52c __pm_runtime_resume+0x58/0x98 __device_attach+0xe0/0x170 device_initial_probe+0x1c/0x28 bus_probe_device+0x3c/0x9c device_add+0x644/0x814 mipi_dsi_device_register_full+0xe4/0x170 devm_mipi_dsi_device_register_full+0x28/0x70 ti_sn_bridge_probe+0x1dc/0x2c0 auxiliary_bus_probe+0x4c/0x94 really_probe+0xcc/0x2c8 __driver_probe_device+0xa8/0x130 driver_probe_device+0x48/0x110 __device_attach_driver+0xa4/0xcc bus_for_each_drv+0x8c/0xd8 __device_attach+0xf8/0x170 device_initial_probe+0x1c/0x28 bus_probe_device+0x3c/0x9c deferred_probe_work_func+0x9c/0xd8 process_one_work+0x148/0x518 worker_thread+0x138/0x350 kthread+0x138/0x1e0 ret_from_fork+0x10/0x20 The first thread is walking the clk tree and calling clk_pm_runtime_get() to power on devices required to read the clk hardware via struct clk_ops::is_enabled(). This thread holds the clk prepare_lock, and is trying to runtime PM resume a device, when it finds that the device is in the process of resuming so the thread schedule(s) away waiting for the device to finish resuming before continuing. The second thread is runtime PM resuming the same device, but the runtime resume callback is calling clk_prepare(), trying to grab the prepare_lock waiting on the first thread. This is a classic ABBA deadlock. To properly fix the deadlock, we must never runtime PM resume or suspend a device with the clk prepare_lock held. Actually doing that is near impossible today because the global prepare_lock would have to be dropped in the middle of the tree, the device runtime PM resumed/suspended, and then the prepare_lock grabbed again to ensure consistency of the clk tree topology. If anything changes with the clk tree in the meantime, we've lost and will need to start the operation all over again. Luckily, most of the time we're simply incrementing or decrementing the runtime PM count on an active device, so we don't have the chance to schedule away with the prepare_lock held. Let's fix this immediate problem that can be ---truncated---
CVE-2024-27008	In the Linux kernel, the following vulnerability has been resolved: drm: nv04: Fix out of bounds access When Output Resource (dcb->or) value is assigned in fabricate_dcb_output(), there may be out of bounds access to dac_users array in case dcb->or is zero because ffs(dcb->or) is used as index there. The 'or' argument of fabricate_dcb_output() must be interpreted as a number of bit to set, not value. Utilize macros from 'enum nouveau_or' in calls instead of hardcoding. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-27022	In the Linux kernel, the following vulnerability has been resolved: fork: defer linking file vma until vma is fully initialized Thorvald reported a WARNING [1]. And the root cause is below race: CPU 1 CPU 2 fork hugetlbfs_fallocate dup_mmap hugetlbfs_punch_hole i_mmap_lock_write(mapping); vma_interval_tree_insert_after -- Child vma is visible through i_mmap tree. i_mmap_unlock_write(mapping); hugetlb_dup_vma_private -- Clear vma_lock outside i_mmap_rwlock! i_mmap_lock_write(mapping); hugetlb_vmdelete_list vma_interval_tree_foreach hugetlb_vma_trylock_write -- Vma_lock is cleared. tmp->vm_ops->open -- Alloc new vma_lock outside i_mmap_rwlock! hugetlb_vma_unlock_write -- Vma_lock is assigned!!! i_mmap_unlock_write(mapping); hugetlb_dup_vma_private() and hugetlb_vm_op_open() are called outside i_mmap_rwlock lock while vma lock can be used in the same time. Fix this by deferring linking file vma until vma is fully initialized. Those vmas should be initialized first before they can be used.
CVE-2024-27023	In the Linux kernel, the following vulnerability has been resolved: md: Fix missing release of 'active_io' for flush submit_flushes atomic_set(&mddev->flush_pending, 1); rdev_for_each_rcu(rdev, mddev) atomic_inc(&mddev->flush_pending); bi->bi_end_io = md_end_flush submit_bio(bi); /* flush io is done first */ md_end_flush if (atomic_dec_and_test(&mddev->flush_pending)) percpu_ref_put(&mddev->active_io) -> active_io is not released if (atomic_dec_and_test(&mddev->flush_pending)) -> missing release of active_io For consequence, mddev_suspend() will wait for 'active_io' to be zero forever. Fix this problem by releasing 'active_io' in submit_flushes() if 'flush_pending' is decreased to zero.

CVE ID	Description
CVE-2024-27025	In the Linux kernel, the following vulnerability has been resolved: nbd: null check for nla_nest_start nla_nest_start() may fail and return NULL. Insert a check and set errno based on other call sites within the same source code.
CVE-2024-27057	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-pcm: Workaround for crashed firmware on system suspend When the system is suspended while audio is active, the sof_ipc4_pcm_hw_free() is invoked to reset the pipelines since during suspend the DSP is turned off, streams will be re-started after resume. If the firmware crashes during while audio is running (or when we reset the stream before suspend) then the sof_ipc4_set_multi_pipeline_state() will fail with IPC error and the state change is interrupted. This will cause misalignment between the kernel and firmware state on next DSP boot resulting errors returned by firmware for IPC messages, eventually failing the audio resume. On stream close the errors are ignored so the kernel state will be corrected on the next DSP boot, so the second boot after the DSP panic. If sof_ipc4_trigger_pipelines() is called from sof_ipc4_pcm_hw_free() then state parameter is SOF_IPC4_PIPE_RESET and only in this case. Treat a forced pipeline reset similarly to how we treat a pcm_free by ignoring error on state sending to allow the kernel's state to be consistent with the state the firmware will have after the next boot.
CVE-2024-27073	In the Linux kernel, the following vulnerability has been resolved: media: ttpci: fix two memleaks in budget_av_attach When saa7146_register_device and saa7146_vv_init fails, budget_av_attach should free the resources it allocates, like the error-handling of ttpci_budget_init does. Besides, there are two fixme comment refers to such deallocations.
CVE-2024-27075	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: avoid stack overflow warnings with clang A previous patch worked around a KASAN issue in stv0367, now a similar problem showed up with clang: drivers/media/dvb-frontends/stv0367.c:1222:12: error: stack frame size (3624) exceeds limit (2048) in 'stv0367ter_set_frontend' [-Werror,-Wframe-larger-than] 1214 static int stv0367ter_set_frontend(struct dvb_frontend *fe) Rework the stv0367_writereg() function to be simpler and mark both register access functions as noline_for_stack so the temporary i2c_msg structures do not get duplicated on the stack when KASAN_STACK is enabled.
CVE-2024-27351	In Django 3.2 before 3.2.25, 4.2 before 4.2.11, and 5.0 before 5.0.3, the django.utils.text.Truncator.words() method (with html=True) and the truncatewords_html template filter are subject to a potential regular expression denial-of-service attack via a crafted string. NOTE: this issue exists because of an incomplete fix for CVE-2019-14232 and CVE-2023-43665.

CVE ID	Description
CVE-2024-27398	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix use-after-free bugs caused by sco_sock_timeout When the sco connection is established and then, the sco socket is releasing, timeout_work will be scheduled to judge whether the sco disconnection is timeout. The sock will be deallocated later, but it is dereferenced again in sco_sock_timeout. As a result, the use-after-free bugs will happen. The root cause is shown below: Cleanup Thread Worker Thread sco_sock_release sco_sock_close __sco_sock_close sco_sock_set_timer schedule_delayed_work sco_sock_kill (wait a time) sock_put(sk) //FREE sco_sock_timeout sock_hold(sk) //USE The KASAN report triggered by POC is shown below: [95.890016] <pre>===== [95.890496] BUG: KASAN: slab-use-after-free in sco_sock_timeout+0x5e/0x1c0 [95.890755] Write of size 4 at addr ffff88800c388080 by task kworker/0:0/7 ... [95.890755] Workqueue: events sco_sock_timeout [95.890755] Call Trace: [95.890755] <TASK> [95.890755] dump_stack_lvl +0x45/0x110 [95.890755] print_address_description+0x78/0x390 [95.890755] print_report +0x11b/0x250 [95.890755] ? __virt_addr_valid+0xbe/0xf0 [95.890755] ? sco_sock_timeout +0x5e/0x1c0 [95.890755] kasan_report+0x139/0x170 [95.890755] ? update_load_avg+0xe5/0x9f0 [95.890755] ? sco_sock_timeout+0x5e/0x1c0 [95.890755] kasan_check_range+0x2c3/0x2e0 [95.890755] sco_sock_timeout+0x5e/0x1c0 [95.890755] process_one_work+0x561/0xc50 [95.890755] worker_thread+0xab2/0x13c0 [95.890755] ? pr_cont_work+0x490/0x490 [95.890755] kthread+0x279/0x300 [95.890755] ? pr_cont_work+0x490/0x490 [95.890755] ? kthread_blkcg +0xa0/0xa0 [95.890755] ret_from_fork+0x34/0x60 [95.890755] ? kthread_blkcg+0xa0/0xa0 [95.890755] ret_from_fork_asm+0x11/0x20 [95.890755] </TASK> [95.890755] [95.890755] Allocated by task 506: [95.890755] kasan_save_track+0x3f/0x70 [95.890755] __kasan_kmalloc +0x86/0x90 [95.890755] __kmalloc+0x17f/0x360 [95.890755] sk_prot_alloc+0xe1/0x1a0 [95.890755] sk_alloc+0x31/0x4e0 [95.890755] bt_sock_alloc+0x2b/0x2a0 [95.890755] sco_sock_create+0xad/0x320 [95.890755] bt_sock_create+0x145/0x320 [95.890755] __sock_create +0x2e1/0x650 [95.890755] __sys_socket+0xd0/0x280 [95.890755] __x64_sys_socket+0x75/0x80 [95.890755] do_syscall_64+0xc4/0x1b0 [95.890755] entry_SYSCALL_64_after_hwframe +0x67/0x6f [95.890755] [95.890755] Freed by task 506: [95.890755] kasan_save_track+0x3f/0x70 [95.890755] kasan_save_free_info+0x40/0x50 [95.890755] poison_slab_object+0x118/0x180 [95.890755] __kasan_slab_free+0x12/0x30 [95.890755] kfree+0xb2/0x240 [95.890755] __sk_destruct+0x317/0x410 [95.890755] sco_sock_release+0x232/0x280 [95.890755] sock_close +0xb2/0x210 [95.890755] __fput+0x37f/0x770 [95.890755] task_work_run+0x1ae/0x210 [95.890755] get_signal+0xe17/0xf70 [95.890755] arch_do_signal_or_restart+0x3f/0x520 [95.890755] syscall_exit_to_user_mode+0x55/0x120 [95.890755] do_syscall_64+0xd1/0x1b0 [95.890755] entry_SYSCALL_64_after_hwframe+0x67/0x6f [95.890755] [95.890755] The buggy address belongs to the object at ffff88800c388000 [95.890755] which belongs to the cache kmalloc-1k of size 1024 [95.890755] The buggy address is located 128 bytes inside of [95.890755] freed 1024-byte region [ffff88800c388000, ffff88800c388400] [95.890755] [95.890755] The buggy address belongs to the physical page: [95.890755] page: refcount:1 mapcount:0 mapping:0000000000000000 index:0xffff88800c38a800 pfn:0xc388 [95.890755] head: order:3 entire_mapcount:0 nr_pages_mapped:0 pincount:0 [95.890755] ano ---truncated---</pre>
CVE-2024-27403	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_flow_offload: reset dst in route object after setting up flow dst is transferred to the flow object, route object does not own it anymore. Reset dst in route object, otherwise if flow_offload_add() fails, error path releases dst twice, leading to a refcount underflow.
CVE-2024-27413	In the Linux kernel, the following vulnerability has been resolved: efi/capsule-loader: fix incorrect allocation size gcc-14 notices that the allocation with sizeof(void) on 32-bit architectures is not enough for a 64-bit phys_addr_t: drivers/firmware/efi/capsule-loader.c: In function 'efi_capsule_open': drivers/firmware/efi/capsule-loader.c:295:24: error: allocation of insufficient size '4' for type 'phys_addr_t' {aka 'long long unsigned int'} with size '8' [-Werror=alloc-size] 295 cap_info->phys = kzalloc(sizeof(void *), GFP_KERNEL); ^ Use the correct type instead here.

CVE ID	Description
CVE-2024-27415	In the Linux kernel, the following vulnerability has been resolved: netfilter: bridge: confirm multicast packets before passing them up the stack conntrack nf_confirm logic cannot handle cloned skbs referencing the same nf_conn entry, which will happen for multicast (broadcast) frames on bridges. Example: macvlan0 br0 / \ ethX ethY ethX (or Y) receives a L2 multicast or broadcast packet containing an IP packet, flow is not yet in conntrack table. 1. skb passes through bridge and fake-ip (br_netfilter)Prerouting. -> skb->_nfct now references a unconfirmed entry 2. skb is broad/mcast packet. bridge now passes clones out on each bridge interface. 3. skb gets passed up the stack. 4. In macvlan case, macvlan driver retains clone(s) of the mcast skb and schedules a work queue to send them out on the lower devices. The clone skb->_nfct is not a copy, it is the same entry as the original skb. The macvlan rx handler then returns RX_HANDLER_PASS. 5. Normal conntrack hooks (in NF_INET_LOCAL_IN) confirm the orig skb. The Macvlan broadcast worker and normal confirm path will race. This race will not happen if step 2 already confirmed a clone. In that case later steps perform skb_clone() with skb->_nfct already confirmed (in hash table). This works fine. But such confirmation won't happen when eb/ip/nftables rules dropped the packets before they reached the nf_confirm step in postrouting. Pablo points out that nf_conntrack_bridge doesn't allow use of stateful nat, so we can safely discard the nf_conn entry and let inet call conntrack again. This doesn't work for bridge netfilter: skb could have a nat transformation. Also bridge nf prevents re-invocation of inet prerouting via 'sabotage_in' hook. Work around this problem by explicit confirmation of the entry at LOCAL_IN time, before upper layer has a chance to clone the unconfirmed entry. The downside is that this disables NAT and conntrack helpers. Alternative fix would be to add locking to all code parts that deal with unconfirmed packets, but even if that could be done in a sane way this opens up other problems, for example: -m physdev --physdev-out eth0 -j SNAT --snat-to 1.2.3.4 -m physdev --physdev-out eth1 -j SNAT --snat-to 1.2.3.5 For multicast case, only one of such conflicting mappings will be created, conntrack only handles 1:1 NAT mappings. Users should set create a setup that explicitly marks such traffic NOTRACK (conntrack bypass) to avoid this, but we cannot auto-bypass them, ruleset might have accept rules for untracked traffic already, so user-visible behaviour would change.
CVE-2024-27417	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix potential "struct net" leak in inet6_rtm_getaddr() It seems that if userspace provides a correct IFA_TARGET_NETNSID value but no IFA_ADDRESS and IFA_LOCAL attributes, inet6_rtm_getaddr() returns -EINVAL with an elevated "struct net" refcount.
CVE-2024-27431	In the Linux kernel, the following vulnerability has been resolved: cpumap: Zero-initialise xdp_rxq_info struct before running XDP program When running an XDP program that is attached to a cpumap entry, we don't initialise the xdp_rxq_info data structure being used in the xdp_buff that backs the XDP program invocation. Tobias noticed that this leads to random values being returned as the xdp_md->rx_queue_index value for XDP programs running in a cpumap. This means we're basically returning the contents of the uninitialised memory, which is bad. Fix this by zero-initialising the rxq data structure before running the XDP program.
CVE-2024-27435	In the Linux kernel, the following vulnerability has been resolved: nvme: fix reconnection fail due to reserved tag allocation We found a issue on production environment while using NVMe over RDMA, admin_q reconnect failed forever while remote target and network is ok. After dig into it, we found it may caused by a ABBA deadlock due to tag allocation. In my case, the tag was hold by a keep alive request waiting inside admin_q, as we quiesced admin_q while reset ctrl, so the request maked as idle and will not process before reset success. As fabric_q shares tagset with admin_q, while reconnect remote target, we need a tag for connect command, but the only one reserved tag was held by keep alive command which waiting inside admin_q. As a result, we failed to reconnect admin_q forever. In order to fix this issue, I think we should keep two reserved tags for admin queue.
CVE-2024-27436	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Stop parsing channels bits when all channels are found. If a usb audio device sets more bits than the amount of channels it could write outside of the map array.
CVE-2024-27437	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Disable auto-enable of exclusive INTx IRQ Currently for devices requiring masking at the irqchip for INTx, ie. devices without DisINTx support, the IRQ is enabled in request_irq() and subsequently disabled as necessary to align with the masked status flag. This presents a window where the interrupt could fire between these events, resulting in the IRQ incrementing the disable depth twice. This would be unrecoverable for a user since the masked flag prevents nested enables through vfio. Instead, invert the logic using IRQF_NO_AUTOEN such that exclusive INTx is never auto-enabled, then unmask as required.
CVE-2024-28180	Package jose aims to provide an implementation of the Javascript Object Signing and Encryption set of standards. An attacker could send a JWE containing compressed data that used large amounts of memory and CPU when decompressed by Decrypt or DecryptMulti. Those functions now return an error if the decompressed data would exceed 250kB or 10x the compressed size (whichever is larger). This vulnerability has been patched in versions 4.0.1, 3.0.3 and 2.6.3.
CVE-2024-28219	In _imagingcms.c in Pillow before 10.3.0, a buffer overflow exists because strcpy is used instead of strncpy.

CVE ID	Description
CVE-2024-28956	Exposure of Sensitive Information in Shared Microarchitectural Structures during Transient Execution for some Intel(R) Processors may allow an authenticated user to potentially enable information disclosure via local access.
CVE-2024-28960	An issue was discovered in Mbed TLS 2.18.0 through 2.28.x before 2.28.8 and 3.x before 3.6.0, and Mbed Crypto. The PSA Crypto API mishandles shared memory.
CVE-2024-29025	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. The `HttpPostRequestDecoder` can be tricked to accumulate data. While the decoder can store items on the disk if configured so, there are no limits to the number of fields the form can have, an attacker can send a chunked post consisting of many small fields that will be accumulated in the `bodyListHttpData` list. The decoder cumulates bytes in the `undecodedChunk` buffer until it can decode a field, this field can cumulate data without limits. This vulnerability is fixed in 4.1.108.Final.
CVE-2024-29131	Out-of-bounds Write vulnerability in Apache Commons Configuration. This issue affects Apache Commons Configuration: from 2.0 before 2.10.1. Users are recommended to upgrade to version 2.10.1, which fixes the issue.
CVE-2024-29133	Out-of-bounds Write vulnerability in Apache Commons Configuration. This issue affects Apache Commons Configuration: from 2.0 before 2.10.1. Users are recommended to upgrade to version 2.10.1, which fixes the issue.
CVE-2024-29857	An issue was discovered in ECCurve.java and ECCurve.cs in Bouncy Castle Java (BC Java) before 1.78, BC Java LTS before 2.73.6, BC-FJA before 1.0.2.5, and BC C#.Net before 2.3.1. Importing an EC certificate with crafted F2m parameters can lead to excessive CPU consumption during the evaluation of the curve parameters.
CVE-2024-29869	Hive creates a credentials file to a temporary directory in the file system with permissions 644 by default when the file permissions are not set explicitly. Any unauthorized user having access to the directory can read the sensitive information written into this file. Users are recommended to upgrade to version 4.0.1, which fixes this issue.
CVE-2024-30171	An issue was discovered in Bouncy Castle Java TLS API and JSSE Provider before 1.78. Timing-based leakage may occur in RSA based handshakes because of exception processing.
CVE-2024-30172	An issue was discovered in Bouncy Castle Java Cryptography APIs before 1.78. An Ed25519 verification code infinite loop can occur via a crafted signature and public key.
CVE-2024-30204	In Emacs before 29.3, LaTeX preview is enabled by default for e-mail attachments.
CVE-2024-3056	A flaw was found in Podman. This issue may allow an attacker to create a specially crafted container that, when configured to share the same IPC with at least one other container, can create a large number of IPC resources in /dev/shm. The malicious container will continue to exhaust resources until it is out-of-memory (OOM) killed. While the malicious container's cgroup will be removed, the IPC resources it created are not. Those resources are tied to the IPC namespace that will not be removed until all containers using it are stopped, and one non-malicious container is holding the namespace open. The malicious container is restarted, either automatically or by attacker control, repeating the process and increasing the amount of memory consumed. With a container configured to restart always, such as `podman run --restart=always`, this can result in a memory-based denial of service of the system.
CVE-2024-31141	Files or Directories Accessible to External Parties, Improper Privilege Management vulnerability in Apache Kafka Clients. Apache Kafka Clients accept configuration data for customizing behavior, and includes ConfigProvider plugins in order to manipulate these configurations. Apache Kafka also provides FileConfigProvider, DirectoryConfigProvider, and EnvVarConfigProvider implementations which include the ability to read from disk or environment variables. In applications where Apache Kafka Clients configurations can be specified by an untrusted party, attackers may use these ConfigProviders to read arbitrary contents of the disk and environment variables. In particular, this flaw may be used in Apache Kafka Connect to escalate from REST API access to filesystem/ environment access, which may be undesirable in certain environments, including SaaS products. This issue affects Apache Kafka Clients: from 2.3.0 through 3.5.2, 3.6.2, 3.7.0. Users with affected applications are recommended to upgrade kafka-clients to version >=3.8.0, and set the JVM system property "org.apache.kafka.automatic.config.providers=none". Users of Kafka Connect with one of the listed ConfigProvider implementations specified in their worker config are also recommended to add appropriate "allowlist.pattern" and "allowed.paths" to restrict their operation to appropriate bounds. For users of Kafka Clients or Kafka Connect in environments that trust users with disk and environment variable access, it is not recommended to set the system property. For users of the Kafka Broker, Kafka MirrorMaker 2.0, Kafka Streams, and Kafka command-line tools, it is not recommended to set the system property.

CVE ID	Description
CVE-2024-31578	FFmpeg version n6.1.1 was discovered to contain a heap use-after-free via the <code>av_hwframe_ctx_init</code> function.
CVE-2024-31582	FFmpeg version n6.1 was discovered to contain a heap buffer overflow vulnerability in the <code>draw_block_rectangle</code> function of <code>libavfilter/vf_codecview.c</code> . This vulnerability allows attackers to cause undefined behavior or a Denial of Service (DoS) via crafted input.
CVE-2024-31755	cJSON v1.7.17 was discovered to contain a segmentation violation, which can trigger through the second parameter of function <code>cJSON_SetValueString</code> at <code>cJSON.c</code> .
CVE-2024-3177	A security issue was discovered in Kubernetes where users may be able to launch containers that bypass the mountable secrets policy enforced by the ServiceAccount admission plugin when using containers, init containers, and ephemeral containers with the <code>envFrom</code> field populated. The policy ensures pods running with a service account may only reference secrets specified in the service account's secrets field. Kubernetes clusters are only affected if the ServiceAccount admission plugin and the <code>kubernetes.io/enforce-mountable-secrets</code> annotation are used together with containers, init containers, and ephemeral containers with the <code>envFrom</code> field populated.
CVE-2024-32228	FFmpeg 7.0 is vulnerable to Buffer Overflow. There is a SEGV at <code>libavcodec/hevcdec.c:2947:22</code> in <code>hevc_frame_end</code> .
CVE-2024-32230	FFmpeg 7.0 is vulnerable to Buffer Overflow. There is a negative-size-param bug at <code>libavcodec/mpegvideo_enc.c:1216:21</code> in <code>load_input_picture</code> in FFmpeg7.0
CVE-2024-32473	Moby is an open source container framework that is a key component of Docker Engine, Docker Desktop, and other distributions of container tooling or runtimes. In 26.0.0, IPv6 is not disabled on network interfaces, including those belonging to networks where <code>--ipv6=false</code> . A container with an <code>'ipvlan'</code> or <code>'macvlan'</code> interface will normally be configured to share an external network link with the host machine. Because of this direct access, (1) Containers may be able to communicate with other hosts on the local network over link-local IPv6 addresses, (2) if router advertisements are being broadcast over the local network, containers may get SLAAC-assigned addresses, and (3) the interface will be a member of IPv6 multicast groups. This means interfaces in IPv4-only networks present an unexpectedly and unnecessarily increased attack surface. The issue is patched in 26.0.2. To completely disable IPv6 in a container, use <code>--sysctl=net.ipv6.conf.all.disable_ipv6=1</code> in the <code>'docker create'</code> or <code>'docker run'</code> command. Or, in the service configuration of a <code>'compose'</code> file.
CVE-2024-34062	tqdm is an open source progress bar for Python and CLI. Any optional non-boolean CLI arguments (e.g. <code>--delim</code> , <code>--buf-size</code> , <code>--manpath</code>) are passed through python's <code>'eval'</code> , allowing arbitrary code execution. This issue is only locally exploitable and had been addressed in release version 4.66.3. All users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2024-34069	Werkzeug is a comprehensive WSGI web application library. The debugger in affected versions of Werkzeug can allow an attacker to execute code on a developer's machine under some circumstances. This requires the attacker to get the developer to interact with a domain and subdomain they control, and enter the debugger PIN, but if they are successful it allows access to the debugger even if it is only running on localhost. This also requires the attacker to guess a URL in the developer's application that will trigger the debugger. This vulnerability is fixed in 3.0.3.
CVE-2024-34155	Calling any of the Parse functions on Go source code which contains deeply nested literals can cause a panic due to stack exhaustion.
CVE-2024-34156	Calling <code>Decoder.Decode</code> on a message which contains deeply nested structures can cause a panic due to stack exhaustion. This is a follow-up to CVE-2022-30635.
CVE-2024-34158	Calling <code>Parse</code> on a <code>'// +build'</code> build tag line with deeply nested expressions can cause a panic due to stack exhaustion.
CVE-2024-34459	An issue was discovered in <code>xmllint</code> (from <code>libxml2</code>) before 2.11.8 and 2.12.x before 2.12.7. Formatting error messages with <code>xmllint --htmlout</code> can result in a buffer over-read in <code>xmlHTMLPrintFileContext</code> in <code>xmllint.c</code> .
CVE-2024-35176	REXML is an XML toolkit for Ruby. The REXML gem before 3.2.6 has a denial of service vulnerability when it parses an XML that has many <code>'<'</code> s in an attribute value. Those who need to parse untrusted XMLs may be impacted to this vulnerability. The REXML gem 3.2.7 or later include the patch to fix this vulnerability. As a workaround, don't parse untrusted XMLs.
CVE-2024-35195	Requests is a HTTP library. Prior to 2.32.0, when making requests through a Requests <code>'Session'</code> , if the first request is made with <code>'verify=False'</code> to disable cert verification, all subsequent requests to the same host will continue to ignore cert verification regardless of changes to the value of <code>'verify'</code> . This behavior will continue for the lifecycle of the connection in the connection pool. This vulnerability is fixed in 2.32.0.

CVE ID	Description
CVE-2024-35255	Azure Identity Libraries and Microsoft Authentication Library Elevation of Privilege Vulnerability
CVE-2024-35365	FFmpeg version n6.1.1 has a double-free vulnerability in the fftools/ffmpeg_mux_init.c component of FFmpeg, specifically within the new_stream_audio function.
CVE-2024-35366	FFmpeg n6.1.1 is Integer Overflow. The vulnerability exists in the parse_options function of sbgdec.c within the libavformat module. When parsing certain options, the software does not adequately validate the input. This allows for negative duration values to be accepted without proper bounds checking.
CVE-2024-35367	FFmpeg n6.1.1 has an Out-of-bounds Read via libavcodec/ppc/vp8dsp_altivec.c, static const vec_s8 h_subpel_filters_outer
CVE-2024-35368	FFmpeg n7.0 is affected by a Double Free via the rkmpv_retrieve_frame function within libavcodec/rkmpvdec.c.
CVE-2024-35791	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Flush pages under kvm->lock to fix UAF in svm_register_enc_region() Do the cache flush of converted pages in svm_register_enc_region() before dropping kvm->lock to fix use-after-free issues where region and/or its array of pages could be freed by a different task, e.g. if userspace has __unregister_enc_region_locked() already queued up for the region. Note, the "obvious" alternative of using local variables doesn't fully resolve the bug, as region->pages is also dynamically allocated. I.e. the region structure itself would be fine, but region->pages could be freed. Flushing multiple pages under kvm->lock is unfortunate, but the entire flow is a rare slow path, and the manual flush is only needed on CPUs that lack coherency for encrypted memory.
CVE-2024-35794	In the Linux kernel, the following vulnerability has been resolved: dm-raid: really frozen sync_thread during suspend 1) commit f52f5c71f3d4 ("md: fix stopping sync thread") remove MD_RECOVERY_FROZEN from __md_stop_writes() and doesn't realize that dm-raid relies on __md_stop_writes() to frozen sync_thread indirectly. Fix this problem by adding MD_RECOVERY_FROZEN in md_stop_writes(), and since stop_sync_thread() is only used for dm-raid in this case, also move stop_sync_thread() to md_stop_writes(). 2) The flag MD_RECOVERY_FROZEN doesn't mean that sync thread is frozen, it only prevent new sync_thread to start, and it can't stop the running sync thread; In order to frozen sync_thread, after setting the flag, stop_sync_thread() should be used. 3) The flag MD_RECOVERY_FROZEN doesn't mean that writes are stopped, use it as condition for md_stop_writes() in raid_postsuspend() doesn't look correct. Consider that reentrant stop_sync_thread() do nothing, always call md_stop_writes() in raid_postsuspend(). 4) raid_message can set/clear the flag MD_RECOVERY_FROZEN at anytime, and if MD_RECOVERY_FROZEN is cleared while the array is suspended, new sync_thread can start unexpected. Fix this by disallow raid_message() to change sync_thread status during suspend. Note that after commit f52f5c71f3d4 ("md: fix stopping sync thread"), the test shell/lvconvert-raid-reshape.sh start to hang in stop_sync_thread(), and with previous fixes, the test won't hang there anymore, however, the test will still fail and complain that ext4 is corrupted. And with this patch, the test won't hang due to stop_sync_thread() or fail due to ext4 is corrupted anymore. However, there is still a deadlock related to dm-raid456 that will be fixed in following patches.
CVE-2024-35803	In the Linux kernel, the following vulnerability has been resolved: x86/efistub: Call mixed mode boot services on the firmware's stack Normally, the EFI stub calls into the EFI boot services using the stack that was live when the stub was entered. According to the UEFI spec, this stack needs to be at least 128k in size - this might seem large but all asynchronous processing and event handling in EFI runs from the same stack and so quite a lot of space may be used in practice. In mixed mode, the situation is a bit different: the bootloader calls the 32-bit EFI stub entry point, which calls the decompressor's 32-bit entry point, where the boot stack is set up, using a fixed allocation of 16k. This stack is still in use when the EFI stub is started in 64-bit mode, and so all calls back into the EFI firmware will be using the decompressor's limited boot stack. Due to the placement of the boot stack right after the boot heap, any stack overruns have gone unnoticed. However, commit 5c4feadb0011983b ("x86/decompressor: Move global symbol references to C code") moved the definition of the boot heap into C code, and now the boot stack is placed right at the base of BSS, where any overruns will corrupt the end of the .data section. While it would be possible to work around this by increasing the size of the boot stack, doing so would affect all x86 systems, and mixed mode systems are a tiny (and shrinking) fraction of the x86 installed base. So instead, record the firmware stack pointer value when entering from the 32-bit firmware, and switch to this stack every time a EFI boot service call is made.
CVE-2024-35805	In the Linux kernel, the following vulnerability has been resolved: dm snapshot: fix lockup in dm_exception_table_exit There was reported lockup when we exit a snapshot with many exceptions. Fix this by adding "cond_resched" to the loop that frees the exceptions.

CVE ID	Description
CVE-2024-35808	In the Linux kernel, the following vulnerability has been resolved: md/dm-raid: don't call md_reap_sync_thread() directly Currently md_reap_sync_thread() is called from raid_message() directly without holding 'reconfig_mutex', this is definitely unsafe because md_reap_sync_thread() can change many fields that is protected by 'reconfig_mutex'. However, hold 'reconfig_mutex' here is still problematic because this will cause deadlock, for example, commit 130443d60b1b ("md: refactor idle/frozen_sync_thread() to fix deadlock"). Fix this problem by using stop_sync_thread() to unregister sync_thread, like md/raid did.
CVE-2024-35817	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: amdgpu_ttm_gart_bind set gtt bound flag Otherwise after the GTT bo is released, the GTT and gart space is freed but amdgpu_ttm_backend_unbind will not clear the gart page table entry and leave valid mapping entry pointing to the stale system page. Then if GPU access the gart address mistakenly, it will read undefined value instead page fault, harder to debug and reproduce the real issue.
CVE-2024-35840	In the Linux kernel, the following vulnerability has been resolved: mptcp: use OPTION_MPTCP_MPJ_SYNACK in subflow_finish_connect() subflow_finish_connect() uses four fields (backup, join_id, thmac, none) that may contain garbage unless OPTION_MPTCP_MPJ_SYNACK has been set in mptcp_parse_option()
CVE-2024-35843	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Use device rbtree in iopf reporting path The existing I/O page fault handler currently locates the PCI device by calling pci_get_domain_bus_and_slot(). This function searches the list of all PCI devices until the desired device is found. To improve lookup efficiency, replace it with device_rbtree_find() to search the device within the probed device rbtree. The I/O page fault is initiated by the device, which does not have any synchronization mechanism with the software to ensure that the device stays in the probed device tree. Theoretically, a device could be released by the IOMMU subsystem after device_rbtree_find() and before iopf_get_dev_fault_param(), which would cause a use-after-free problem. Add a mutex to synchronize the I/O page fault reporting path and the IOMMU release device path. This lock doesn't introduce any performance overhead, as the conflict between I/O page fault reporting and device releasing is very rare.
CVE-2024-35860	In the Linux kernel, the following vulnerability has been resolved: bpf: support deferring bpf_link dealloc to after RCU grace period BPF link for some program types is passed as a "context" which can be used by those BPF programs to look up additional information. E.g., for multi-kprobes and multi-uprobes, link is used to fetch BPF cookie values. Because of this runtime dependency, when bpf_link refcnt drops to zero there could still be active BPF programs running accessing link data. This patch adds generic support to defer bpf_link dealloc callback to after RCU GP, if requested. This is done by exposing two different deallocation callbacks, one synchronous and one deferred. If deferred one is provided, bpf_link_free() will schedule dealloc_deferred() callback to happen after RCU GP. BPF is using two flavors of RCU: "classic" non-sleepable one and RCU tasks trace one. The latter is used when sleepable BPF programs are used. bpf_link_free() accommodates that by checking underlying BPF program's sleepable flag, and goes either through normal RCU GP only for non-sleepable, or through RCU tasks trace GP *and* then normal RCU GP (taking into account rcu_trace_implies_rcu_gp() optimization), if BPF program is sleepable. We use this for multi-kprobe and multi-uprobe links, which dereference link during program run. We also preventively switch raw_tp link to use deferred dealloc callback, as upcoming changes in bpf-next tree expose raw_tp link data (specifically, cookie value) to BPF program at runtime as well.
CVE-2024-35864	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential UAF in smb2_is_valid_lease_break() Skip sessions that are being teared down (status == SES_EXITING) to avoid UAF.

CVE ID	Description
CVE-2024-35870	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix UAF in smb2_reconnect_server() The UAF bug is due to smb2_reconnect_server() accessing a session that is already being torn down by another thread that is executing __cifs_put_smb_ses(). This can happen when (a) the client has connection to the server but no session or (b) another thread ends up setting @ses->ses_status again to something different than SES_EXITING. To fix this, we need to make sure to unconditionally set @ses->ses_status to SES_EXITING and prevent any other threads from setting a new status while we're still tearing it down. The following can be reproduced by adding some delay to right after the ipc is freed in __cifs_put_smb_ses() - which will give smb2_reconnect_server() worker a chance to run and then accessing @ses->ipc: kinit ... mount.cifs //srv/share /mnt/1 -o sec=krb5,nohandlecache,echo_interval=10 [disconnect srv] ls /mnt/1 &>/dev/null sleep 30 kdestroy [reconnect srv] sleep 10 umount /mnt/1 ... CIFS: VFS: Verify user has a krb5 ticket and keyutils is installed CIFS: VFS: \\srv Send error in SessSetup = -126 CIFS: VFS: Verify user has a krb5 ticket and keyutils is installed CIFS: VFS: \\srv Send error in SessSetup = -126 general protection fault, probably for non-canonical address 0x6b6b6b6b6b6b6b6b: 0000 [#1] PREEMPT SMP NOPTI CPU: 3 PID: 50 Comm: kworker/3:1 Not tainted 6.9.0-rc2 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-1.fc39 04/01/2014 Workqueue: cifsiod smb2_reconnect_server [cifs] RIP: 0010: __list_del_entry_valid_or_report+0x33/0xf0 Code: 4f 08 48 85 d2 74 42 48 85 c9 74 59 48 b8 00 01 00 00 00 00 ad de 48 39 c2 74 61 48 b8 22 01 00 00 00 00 74 69 <48> 8b 01 48 39 f8 75 7b 48 8b 72 08 48 39 c6 0f 85 88 00 00 00 b8 RSP: 0018:ffffc900001bfd70 EFLAGS: 00010a83 RAX: dead000000000122 RBX: ffff88810da53838 RCX: 6b6b6b6b6b6b6b6b RDX: 6b6b6b6b6b6b6b6b RSI: ffffffff02f6878 RDI: ffff88810da53800 RBP: ffff88810da53800 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000001 R12: ffff88810c064000 R13: 0000000000000001 R14: ffff88810c064000 R15: ffff8881039cc000 FS: 0000000000000000(0000) GS:ffff888157c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fe3728b1000 CR3: 000000010caa4000 CR4: 0000000000750ef0 PKRU: 55555554 Call Trace: <TASK> ? die_addr+0x36/0x90 ? exc_general_protection+0x1c1/0x3f0 ? asm_exc_general_protection+0x26/0x30 ? __list_del_entry_valid_or_report+0x33/0xf0 __cifs_put_smb_ses+0x1ae/0x500 [cifs] smb2_reconnect_server+0x4ed/0x710 [cifs] process_one_work+0x205/0x6b0 worker_thread+0x191/0x360 ? __pfx_worker_thread+0x10/0x10 kthread+0xe2/0x110 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x34/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK>
CVE-2024-35875	In the Linux kernel, the following vulnerability has been resolved: x86/coco: Require seeding RNG with RDRAND on CoCo systems There are few uses of CoCo that don't rely on working cryptography and hence a working RNG. Unfortunately, the CoCo threat model means that the VM host cannot be trusted and may actively work against guests to extract secrets or manipulate computation. Since a malicious host can modify or observe nearly all inputs to guests, the only remaining source of entropy for CoCo guests is RDRAND. If RDRAND is broken -- due to CPU hardware fault -- the RNG as a whole is meant to gracefully continue on gathering entropy from other sources, but since there aren't other sources on CoCo, this is catastrophic. This is mostly a concern at boot time when initially seeding the RNG, as after that the consequences of a broken RDRAND are much more theoretical. So, try at boot to seed the RNG using 256 bits of RDRAND output. If this fails, panic(). This will also trigger if the system is booted without RDRAND, as RDRAND is essential for a safe CoCo boot. Add this deliberately to be "just a CoCo x86 driver feature" and not part of the RNG itself. Many device drivers and platforms have some desire to contribute something to the RNG, and add_device_randomness() is specifically meant for this purpose. Any driver can call it with seed data of any quality, or even garbage quality, and it can only possibly make the quality of the RNG better or have no effect, but can never make it worse. Rather than trying to build something into the core of the RNG, consider the particular CoCo issue just a CoCo issue, and therefore separate it all out into driver (well, arch/platform) code. [bp: Massage commit message.]

CVE ID	Description
CVE-2024-35885	In the Linux kernel, the following vulnerability has been resolved: mlxbf_gige: stop interface during shutdown The mlxbf_gige driver intermittantly encounters a NULL pointer exception while the system is shutting down via "reboot" command. The mlxbf_driver will experience an exception right after executing its shutdown() method. One example of this exception is: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000070 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004 CM = 0, WnR = 0 user pgtable: 4k pages, 48-bit VAs, pgdp=000000011d373000 [0000000000000070] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 96000004 [#1] SMP CPU: 0 PID: 13 Comm: ksoftirqd/0 Tainted: G S OE 5.15.0-bf.6.gef6992a #1 Hardware name: https://www.mellanox.com BlueField SoC/BlueField SoC, BIOS 4.0.2.12669 Apr 21 2023 pstate: 20400009 (nzCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=--) pc : mlxbf_gige_handle_tx_complete+0xc8/0x170 [mlxbf_gige] lr : mlxbf_gige_poll+0x54/0x160 [mlxbf_gige] sp : ffff8000080d3c10 x29: ffff8000080d3c10 x28: ffffce72cbb7000 x27: ffff8000080d3d58 x26: ffff0000814e7340 x25: ffff331cd1a05000 x24: ffffce72c4ea008 x23: ffff0000814e4b40 x22: ffff0000814e4d10 x21: ffff0000814e4128 x20: 0000000000000000 x19: ffff0000814e4a80 x18: ffffffff00000000 x17: 0000000000000001c x16: ffffce72b4553f4 x15: ffff80008805b8a7 x14: 0000000000000000 x13: 0000000000000030 x12: 0101010101010101 x11: 7f7f7f7f7f7f7f7f x10: c2ac898b17576267 x9: ffffce720fa5404 x8: ffff000080812138 x7: 00000000000002e9a x6: 0000000000000080 x5: ffff00008de3b000 x4: 0000000000000000 x3: 0000000000000001 x2: 0000000000000000 x1: 0000000000000000 x0: 0000000000000000 Call trace: mlxbf_gige_handle_tx_complete+0xc8/0x170 [mlxbf_gige] mlxbf_gige_poll+0x54/0x160 [mlxbf_gige] __napi_poll+0x40/0x1c8 net_rx_action+0x314/0x3a0 __do_softirq+0x128/0x334 run_ksoftirqd+0x54/0x6c smpboot_thread_fn+0x14c/0x190 kthread+0x10c/0x110 ret_from_fork+0x10/0x20 Code: 8b070000 f9000ea0 f95056c0 f86178a1 (b9407002) ---[end trace 7cc3941aa0d8e6a4]--- Kernel panic - not syncing: Oops: Fatal exception in interrupt Kernel Offset: 0x4ce722520000 from 0xffff800008000000 PHYS_OFFSET: 0x80000000 CPU features: 0x000005c1,a3330e5a Memory Limit: none ---[end Kernel panic - not syncing: Oops: Fatal exception in interrupt]--- During system shutdown, the mlxbf_gige driver's shutdown() is always executed. However, the driver's stop() method will only execute if networking interface configuration logic within the Linux distribution has been setup to do so. If shutdown() executes but stop() does not execute, NAPI remains enabled and this can lead to an exception if NAPI is scheduled while the hardware interface has only been partially deinitialized. The networking interface managed by the mlxbf_gige driver must be properly stopped during system shutdown so that IFF_UP is cleared, the hardware interface is put into a clean state, and NAPI is fully deinitialized.

CVE ID	Description
CVE-2024-35886	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix infinite recursion in fib6_dump_done(). syzkaller reported infinite recursive calls of fib6_dump_done() during netlink socket destruction. [1] From the log, syzkaller sent an AF_UNSPEC RTM_GETROUTE message, and then the response was generated. The following recvmsg() resumed the dump for IPv6, but the first call of inet6_dump_fib() failed at kzalloc() due to the fault injection. [0] 12:01:34 executing program 3: r0 = socket\$nl_route(0x10, 0x3, 0x0) sendmsg\$nl_route(r0, ... snip ...) recvmsg(r0, ... snip ...) (fail_nth: 8) Here, fib6_dump_done() was set to nlk_sk(sk)->cb.done, and the next call of inet6_dump_fib() set it to nlk_sk(sk)->cb.args[3]. syzkaller stopped receiving the response halfway through, and finally netlink_sock_destruct() called nlk_sk(sk)->cb.done(). fib6_dump_done() calls fib6_dump_end() and nlk_sk(sk)->cb.done() if it is still not NULL. fib6_dump_end() rewrites nlk_sk(sk)->cb.done() by nlk_sk(sk)->cb.args[3], but it has the same function, not NULL, calling itself recursively and hitting the stack guard page. To avoid the issue, let's set the destructor after kzalloc(). [0]: FAULT_INJECTION: forcing a failure. name failslab, interval 1, probability 0, space 0, times 0 CPU: 1 PID: 432110 Comm: syz-executor.3 Not tainted 6.8.0-12821-g537c2e91d354-dirty #11 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:117) should_fail_ex (lib/fault-inject.c:52 lib/fault-inject.c:153) should_failslab (mm/slab.c:3733) kmallocc_trace (mm/slab.c:3748 mm/slab.c:3827 mm/slab.c:3992) inet6_dump_fib (/include/linux/slab.h:628 ./include/linux/slab.h:749 net/ipv6/ip6_fib.c:662) rtm_dump_all (net/core/rtnetlink.c:4029) netlink_dump (net/netlink/af_netlink.c:2269) netlink_recvmsg (net/netlink/af_netlink.c:1988) __sys_recvmsg (net/socket.c:1046 do_recvmsg (net/socket.c:2846) do_recvmsg (net/socket.c:2943) __x64_sys_recvmsg (net/socket.c:3041 net/socket.c:3034 net/socket.c:3034) [1]: BUG: TASK stack guard page was hit at 00000000f2fa9af1 (stack is 00000000b7912430..000000009a436beb) stack guard page: 0000 [1] PREEMPT SMP KASAN CPU: 1 PID: 223719 Comm: kworker/1:3 Not tainted 6.8.0-12821-g537c2e91d354-dirty #11 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 Workqueue: events netlink_sock_destruct_work RIP: 0010:fib6_dump_done (net/ipv6/ip6_fib.c:570) Code: 3c 24 e8 f3 e9 51 fd e9 28 fd ff 66 66 2e 0f 1f 84 00 00 00 00 0f 1f 00 f3 0f 1e fa 41 57 41 56 41 55 41 54 55 48 89 fd <53> 48 8d 5d 60 e8 b6 4d 07 fd 48 89 da 48 b8 00 00 00 00 00 fc ff RSP: 0018:ffffc9000d980000 EFLAGS: 00010293 RAX: 0000000000000000 RBX: ffffffff84405990 RCX: ffffffff844059d3 RDX: ffff8881028e0000 RSI: ffffffff84405ac2 RDI: ffff88810c02f358 RBP: ffff88810c02f358 R08: 0000000000000007 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000224 R12: 0000000000000000 R13: ffff888007c82c78 R14: ffff888007c82c68 R15: ffff888007c82c68 FS: 0000000000000000(0000) GS:ffff88811b100000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffff88800d97ff8 CR3: 0000000102309002 CR4: 00000000000770ef0 PKRU: 55555554 Call Trace: <#DF> <#DF> <TASK> fib6_dump_done (net/ipv6/ip6_fib.c:572 (discriminator 1)) fib6_dump_done (net/ipv6/ip6_fib.c:572 (discriminator 1)) ... fib6_dump_done (net/ipv6/ip6_fib.c:572 (discriminator 1)) fib6_dump_done (net/ipv6/ip6_fib.c:572 (discriminator 1)) netlink_sock_destruct (net/netlink/af_netlink.c:401) __sk_destruct (net/core/sock.c:2177 (discriminator 2)) sk_destruct (net/core/sock.c:2224) __sk_free (net/core/sock.c:2235) sk_free (net/core/sock.c:2246) process_one_work (kernel/workqueue.c:3259) worker_thread (kernel/workqueue.c:3329 kernel/workqueue. ---truncated---
CVE-2024-35895	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Prevent lock inversion deadlock in map delete elem syzkaller started using corpses where a BPF tracing program deletes elements from a sockmap/sockhash map. Because BPF tracing programs can be invoked from any interrupt context, locks taken during a map_delete_elem operation must be hardirq-safe. Otherwise a deadlock due to lock inversion is possible, as reported by lockdep: CPU0 CPU1 ---- lock(&htab->buckets[i].lock); local_irq_disable(); lock(&host->lock); lock(&htab->buckets[i].lock); <Interrupt> lock(&host->lock); Locks in sockmap are hardirq-unsafe by design. We expects elements to be deleted from sockmap/sockhash only in task (normal) context with interrupts enabled, or in softirq context. Detect when map_delete_elem operation is invoked from a context which is _not_ hardirq-unsafe, that is interrupts are disabled, and bail out with an error. Note that map updates are not affected by this issue. BPF verifier does not allow updating sockmap/sockhash from a BPF tracing program today.
CVE-2024-35901	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix Rx DMA datasize and skb_over_panic mana_get_rxbuf_cfg() aligns the RX buffer's DMA datasize to be multiple of 64. So a packet slightly bigger than mtu+14, say 1536, can be received and cause skb_over_panic. Sample dmesg: [5325.237162] skbuff: skb_over_panic: text:fffffc043277a len:1536 put:1536 head:ff1100018b517000 data:ff1100018b517100 tail:0x700 end:0x6ea dev:<NULL> [5325.243689] -----[cut here]----- [5325.245748] kernel BUG at net/core/skbuff.c:192! [5325.247838] invalid opcode: 0000 [1] PREEMPT SMP NOPTI [5325.258374] RIP: 0010:skb_panic+0x4f/0x60 [5325.302941] Call Trace: [5325.304389] <IRQ> [5325.315794] ? skb_panic+0x4f/0x60 [5325.317457] ? asm_exc_invalid_op+0x1f/0x30 [5325.319490] ? skb_panic+0x4f/0x60 [5325.321161] skb_put+0x4e/0x50 [5325.322670] mana_poll+0x6fa/0xb50 [mana] [5325.324578] __napi_poll+0x33/0x1e0 [5325.326328] net_rx_action+0x12e/0x280 As discussed internally, this alignment is not necessary. To fix this bug, remove it from the code. So oversized packets will be marked as CQE_RX_TRUNCATED by NIC, and dropped.

CVE ID	Description
CVE-2024-35904	In the Linux kernel, the following vulnerability has been resolved: selinux: avoid dereference of garbage after mount failure In case kern_mount() fails and returns an error pointer return in the error branch instead of continuing and dereferencing the error pointer. While on it drop the never read static variable selinuxfs_mount.
CVE-2024-35907	In the Linux kernel, the following vulnerability has been resolved: mlxbf_gige: call request_irq() after NAPI initialized The mlxbf_gige driver encounters a NULL pointer exception in mlxbf_gige_open() when kdump is enabled. The sequence to reproduce the exception is as follows: a) enable kdump b) trigger kdump via "echo c > /proc/sysrq-trigger" c) kdump kernel executes d) kdump kernel loads mlxbf_gige module e) the mlxbf_gige module runs its open() as the "oob_net0" interface is brought up f) mlxbf_gige module will experience an exception during its open(), something like: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x0000000086000004 EC = 0x21: IABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault user pgttable: 4k pages, 48-bit VAs, pgdp=00000000e29a4000 [0000000000000000] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000086000004 [#1] SMP CPU: 0 PID: 812 Comm: NetworkManager Tainted: G OE 5.15.0-1035-bluefield #37-Ubuntu Hardware name: https://www.mellanox.com BlueField-3 SmartNIC Main Card/BlueField-3 SmartNIC Main Card, BIOS 4.6.0.13024 Jan 19 2024 pstate: 80400009 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : 0x0 lr : __napi_poll+0x40/0x230 sp : ffff800008003e00 x29: ffff800008003e00 x28: 0000000000000000 x27: 00000000ffffff x26: ffff000066027238 x25: ffff00007cedec00 x24: ffff800008003ec8 x23: 000000000000012c x22: ffff800008003eb7 x21: 0000000000000000 x20: 0000000000000001 x19: ffff000066027238 x18: 0000000000000000 x17: ffff578fcb450000 x16: ffffa870b083c7c0 x15: 0000aaab010441d0 x14: 0000000000000001 x13: 00726f7272655f65 x12: 6769675f6662786c x11: 0000000000000000 x10: 0000000000000000 x9 : ffffa870b0842398 x8 : 0000000000000004 x7 : fe5a48b9069706ea x6 : 17fdb11fc84ae0d2 x5 : d94a82549d594f35 x4 : 0000000000000000 x3 : 0000000000400100 x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff000066027238 Call trace: 0x0 net_rx_action+0x178/0x360 __do_softirq+0x15c/0x428 __irq_exit_rcu+0xac/0xec irq_exit+0x18/0x2c handle_domain_irq+0x6c/0xa0 gic_handle_irq+0xec/0x1b0 call_on_irq_stack+0x20/0x2c do_interrupt_handler+0x5c/0x70 el1_interrupt+0x30/0x50 el1h_64_irq_handler+0x18/0x2c el1h_64_irq+0x7c/0x80 __setup_irq+0x4c0/0x950 request_threaded_irq+0xf4/0x1bc mlxbf_gige_request_irqs+0x68/0x110 [mlxbf_gige] mlxbf_gige_open+0x5c/0x170 [mlxbf_gige] __dev_open+0x100/0x220 __dev_change_flags+0x16c/0x1f0 dev_change_flags+0x2c/0x70 do_setlink+0x220/0xa40 __rtnl_newlink+0x56c/0x8a0 rtnl_newlink+0x58/0x84 rtnetlink_rcv_msg+0x138/0x3c4 netlink_rcv_skb+0x64/0x130 rtnetlink_rcv+0x20/0x30 netlink_unicast+0x2ec/0x360 netlink_sendmsg+0x278/0x490 __sock_sendmsg+0x5c/0x6c ____sys_sendmsg+0x290/0x2d4 __sys_sendmsg+0x84/0xd0 __sys_sendmsg+0x70/0xd0 __arm64_sys_sendmsg+0x2c/0x40 invoke_syscall+0x78/0x100 el0_svc_common.constprop.0+0x54/0x184 do_el0_svc+0x30/0xac el0_svc+0x48/0x160 el0t_64_sync_handler+0xa4/0x12c el0t_64_sync+0x1a4/0x1a8 Code: bad PC value --- [end trace 7d1c3f3bf9d81885]--- Kernel panic - not syncing: Oops: Fatal exception in interrupt Kernel Offset: 0x2870a7a00000 from 0xffff800008000000 PHYS_OFFSET: 0x80000000 CPU features: 0x0,000005c1,a3332a5a Memory Limit: none ---[end Kernel panic - not syncing: Oops: Fatal exception in interrupt]--- The exception happens because there is a pending RX interrupt before the call to request_irq(RX IRQ) executes. Then, the RX IRQ handler fires immediately after this request_irq() completes. The ---truncated---

CVE ID	Description
CVE-2024-35911	In the Linux kernel, the following vulnerability has been resolved: ice: fix memory corruption bug with suspend and rebuild The ice driver would previously panic after suspend. This is caused from the driver *only* calling the ice_vsi_free_q_vectors() function by itself, when it is suspending. Since commit b3e7b3a6ee92 ("ice: prevent NULL pointer deref during reload") the driver has zeroed out num_q_vectors, and only restored it in ice_vsi_cfg_def(). This further causes the ice_rebuild() function to allocate a zero length buffer, after which num_q_vectors is updated, and then the new value of num_q_vectors is used to index into the zero length buffer, which corrupts memory. The fix entails making sure all the code referencing num_q_vectors only does so after it has been reset via ice_vsi_cfg_def(). I didn't perform a full bisect, but I was able to test against 6.1.77 kernel and that ice driver works fine for suspend/resume with no panic, so sometime since then, this problem was introduced. Also clean up an un-needed init of a local variable in the function being modified. PANIC from 6.8.0-rc1: [1026674.915596] PM: suspend exit [1026675.664697] ice 0000:17:00.1: PTP reset successful [1026675.664707] ice 0000:17:00.1: 2755 msec passed between update to cached PHC time [1026675.667660] ice 0000:b1:00.0: PTP reset successful [1026675.675944] ice 0000:b1:00.0: 2832 msec passed between update to cached PHC time [1026677.137733] ixgbe 0000:31:00.0 ens787: NIC Link is Up 1 Gbps, Flow Control: None [1026677.190201] BUG: kernel NULL pointer dereference, address: 0000000000000010 [1026677.192753] ice 0000:17:00.0: PTP reset successful [1026677.192764] ice 0000:17:00.0: 4548 msec passed between update to cached PHC time [1026677.197928] #PF: supervisor read access in kernel mode [1026677.197933] #PF: error_code(0x0000) - not-present page [1026677.197937] PGD 1557a7067 P4D 0 [1026677.212133] ice 0000:b1:00.1: PTP reset successful [1026677.212143] ice 0000:b1:00.1: 4344 msec passed between update to cached PHC time [1026677.212575] [1026677.243142] Oops: 0000 [#1] PREEMPT SMP NOPTI [1026677.247918] CPU: 23 PID: 42790 Comm: kworker/23:0 Kdump: loaded Tainted: G W 6.8.0-rc1+ #1 [1026677.257989] Hardware name: Intel Corporation M50CYP2SBSTD/M50CYP2SBSTD, BIOS SE5C620.86B.01.01.0005.2202160810 02/16/2022 [1026677.269367] Workqueue: ice ice_service_task [ice] [1026677.274592] RIP: 0010:ice_vsi_rebuild_set_coalesce+0x130/0x1e0 [ice] [1026677.281421] Code: 0f 84 3a ff ff ff 41 0f b7 74 ec 02 66 89 b0 22 02 00 00 81 e6 ff 1f 00 00 e8 ec fd ff ff e9 35 ff ff ff 48 8b 43 30 49 63 ed <41> 0f b7 34 24 41 83 c5 01 48 8b 3c e8 66 89 b7 aa 02 00 00 81 e6 [1026677.300877] RSP: 0018:ff3be62a6399bcc0 EFLAGS: 00010202 [1026677.306556] RAX: ff28691e28980828 RBX: ff28691e41099828 RCX: 0000000000188000 [1026677.314148] RDX: 0000000000000000 RSI: 0000000000000010 RDI: ff28691e41099828 [1026677.321730] RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 [1026677.329311] R10: 0000000000000007 R11: ffffffffefc0 R12: 0000000000000010 [1026677.336896] R13: 0000000000000000 R14: 0000000000000000 R15: ff28691e0eaa81a0 [1026677.344472] FS: 0000000000000000(0000) GS:ff28693cbffc0000(0000) knlGS:0000000000000000 [1026677.353000] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [1026677.359195] CR2: 0000000000000010 CR3: 0000000128df4001 CR4: 0000000000771ef0 [1026677.366779] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [1026677.374369] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [1026677.381952] PKRU: 55555554 [1026677.385116] Call Trace: [1026677.388023] <TASK> [1026677.390589] ? __die+0x20/0x70 [1026677.394105] ? page_fault_oops+0x82/0x160 [1026677.398576] ? do_user_addr_fault+0x65/0x6a0 [1026677.403307] ? exc_page_fault+0x6a/0x150 [1026677.407694] ? asm_exc_page_fault+0x22/0x30 [1026677.412349] ? ice_vsi_rebuild_set_coalesce+0x130/0x1e0 [ice] [1026677.4186 ---truncated---
CVE-2024-35922	In the Linux kernel, the following vulnerability has been resolved: fbmon: prevent division by zero in fb_videomode_from_videomode() The expression htotal * vtotal can have a zero value on overflow. It is necessary to prevent division by zero like in fb_var_to_videomode(). Found by Linux Verification Center (linuxtesting.org) with Svace.

CVE ID	Description
CVE-2024-35927	In the Linux kernel, the following vulnerability has been resolved: drm: Check output polling initialized before disabling In <code>drm_kms_helper_poll_disable()</code> check if output polling support is initialized before disabling polling. If not flag this as a warning. Additionally in <code>drm_mode_config_helper_suspend()</code> and <code>drm_mode_config_helper_resume()</code> calls, that re the callers of these functions, avoid invoking them if polling is not initialized. For drivers like <code>hyperv-drm</code>, that do not initialize connector polling, if suspend is called without this check, it leads to suspend failure with following stack [770.719392] Freezing remaining freezable tasks ... (elapsed 0.001 seconds) done. [770.720592] printk: Suspending console(s) (use no_console_suspend to debug) [770.948823] -----[cut here]----- [770.948824] WARNING: CPU: 1 PID: 17197 at kernel/workqueue.c:3162 __flush_work.isra.0+0x212/0x230 [770.948831] Modules linked in: rfkill nft_counter xt_conntrack xt_owner udf nft_compat crc_itu_t nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 ip_set nf_tables nfnetlink vfat fat mlx5_ib ib_uverbs ib_core mlx5_core intel_rapl_msr intel_rapl_common kvm_amd ccp mlxfw kvm psample hyperv_drm tls drm_shmem_helper drm_kms_helper irqbypass pcspkr syscopyarea sysfillrect sysimgblt hv_balloon hv_utils joydev drm fuse xfs libcrc32c pci_hyperv pci_hyperv_intf sr_mod sd_mod cdrom t10_pi sg hv_storvsc scsi_transport_fc hv_netvsc serio_raw hyperv_keyboard hid_hyperv crct10dif_pclmul crc32_pclmul crc32c_intel hv_vmbus ghash_clmulni_intel dm_mirror dm_region_hash dm_log dm_mod [770.948863] CPU: 1 PID: 17197 Comm: systemd-sleep Not tainted 5.14.0-362.2.1.el9_3.x86_64 #1 [770.948865] Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS Hyper-V UEFI Release v4.1 05/09/2022 [770.948866] RIP: 0010: __flush_work.isra.0+0x212/0x230 [770.948869] Code: 8b 4d 00 4c 8b 45 08 89 ca 48 c1 e9 04 83 e2 08 83 e1 0f 83 ca 02 89 c8 48 0f ba 6d 00 03 e9 25 ff ff 0f 0b e9 4e ff ff ff <0f> 0b 45 31 ed e9 44 ff ff ff e8 8f 89 b2 00 66 66 2e 0f 1f 84 00 [770.948870] RSP: 0018:ffffaf4ac213fb10 EFLAGS: 00010246 [770.948871] RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffffff8c992857 [770.948872] RDX: 0000000000000001 RSI: 0000000000000001 RDI: ffff9aad82b00330 [770.948873] RBP: ffff9aad82b00330 R08: 0000000000000000 R09: ffff9aad87ee3d10 [770.948874] R10: 0000000000000200 R11: 0000000000000000 R12: ffff9aad82b00330 [770.948874] R13: 0000000000000001 R14: 0000000000000000 R15: 0000000000000001 [770.948875] FS: 00007ff1b2f6bb40(0000) GS: ffff9aaf37d00000(0000) knlGS: 0000000000000000 [770.948878] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [770.948878] CR2: 0000555f345cb666 CR3: 00000001462dc005 CR4: 0000000000370ee0 [770.948879] Call Trace: [770.948880] <TASK> [770.948881] ? show_trace_log_lvl+0x1c4/0x2df [770.948884] ? show_trace_log_lvl+0x1c4/0x2df [770.948886] ? __cancel_work_timer+0x103/0x190 [770.948887] ? __flush_work.isra.0+0x212/0x230 [770.948889] ? __warn+0x81/0x110 [770.948891] ? __flush_work.isra.0+0x212/0x230 [770.948892] ? report_bug+0x10a/0x140 [770.948895] ? handle_bug+0x3c/0x70 [770.948898] ? exc_invalid_op+0x14/0x70 [770.948899] ? asm_exc_invalid_op+0x16/0x20 [770.948903] ? __flush_work.isra.0+0x212/0x230 [770.948905] __cancel_work_timer+0x103/0x190 [770.948907] ? _raw_spin_unlock_irqrestore+0xa/0x30 [770.948910] drm_kms_helper_poll_disable+0x1e/0x40 [drm_kms_helper] [770.948923] drm_mode_config_helper_suspend+0x1c/0x80 [drm_kms_helper] [770.948933] ? __pfx_vmbus_suspend+0x10/0x10 [hv_vmbus] [770.948942] hyperv_vmbus_suspend+0x17/0x40 [hyperv_drm] [770.948944] ? __pfx_vmbus_suspend+0x10/0x10 [hv_vmbus] [770.948951] dpm_run_callback+0x4c/0x140 [770.948954] __device_suspend_noir ---truncated---

CVE ID	Description
CVE-2024-35931	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Skip do PCI error slot reset during RAS recovery Why: The PCI error slot reset maybe triggered after inject ue to UMC multi times, this caused system hang. [557.371857] amdgpu 0000:af:00.0: amdgpu: GPU reset succeeded, trying to resume [557.373718] [drm] PCIE GART of 512M enabled. [557.373722] [drm] PTB located at 0x0000031FED700000 [557.373788] [drm] VRAM is lost due to GPU reset! [557.373789] [drm] PSP is resuming... [557.547012] mlx5_core 0000:55:00.0: mlx5_pci_err_detected Device state = 1 pci_status: 0. Exit, result = 3, need reset [557.547067] [drm] PCI error: detected callback, state(1)!! [557.547069] [drm] No support for XGMI hive yet... [557.548125] mlx5_core 0000:55:00.0: mlx5_pci_slot_reset Device state = 1 pci_status: 0. Enter [557.607763] mlx5_core 0000:55:00.0: wait vital counter value 0x16b5b after 1 iterations [557.607777] mlx5_core 0000:55:00.0: mlx5_pci_slot_reset Device state = 1 pci_status: 1. Exit, err = 0, result = 5, recovered [557.610492] [drm] PCI error: slot reset callback!! ... [560.689382] amdgpu 0000:3f:00.0: amdgpu: GPU reset(2) succeeded! [560.689546] amdgpu 0000:5a:00.0: amdgpu: GPU reset(2) succeeded! [560.689562] general protection fault, probably for non-canonical address 0x5f080b54534f611f: 0000 [#1] SMP NOPTI [560.701008] CPU: 16 PID: 2361 Comm: kworker/u448:9 Tainted: G OE 5.15.0-91-generic #101-Ubuntu [560.712057] Hardware name: Microsoft C278A/C278A, BIOS C2789.5.BS.1C11.AG.1 11/08/2023 [560.720959] Workqueue: amdgpu-reset-hive amdgpu_ras_do_recovery [amdgpu] [560.728887] RIP: 0010:amdgpu_device_gpu_recover.cold+0xbff1/0xcf5 [amdgpu] [560.736891] Code: ff 41 89 c6 e9 1b ff ff ff 44 0f b6 45 b0 e9 4f ff ff ff be 01 00 00 00 4c 89 e7 e8 76 c9 8b ff 44 0f b6 45 b0 e9 3c fd ff ff <48> 83 ba 18 02 00 00 0f 84 6a ff ff ff 48 d7 7a 78 be 01 00 00 [560.757967] RSP: 0018:ffa0000032e53d80 EFLAGS: 00010202 [560.763848] RAX: ffa00000001dfd10 RBX: ffa0000000197090 RCX: ffa0000032e53db0 [560.771856] RDX: 5f080b54534f5f07 RSI: 0000000000000000 RDI: ff11000128100010 [560.779867] RBP: ffa0000032e53df0 R08: 0000000000000000 R09: ffffffff77f08 [560.787879] R10: 0000000000ffff0a R11: 0000000000000001 R12: 0000000000000000 [560.795889] R13: ffa0000032e53e00 R14: 0000000000000000 R15: 0000000000000000 [560.803889] FS: 0000000000000000(0000) GS:ff11007e7e800000(0000) knlGS:0000000000000000 [560.812973] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [560.819422] CR2: 000055a04c118e68 CR3: 0000000007410005 CR4: 0000000000771ee0 [560.827433] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [560.835433] DR3: 0000000000000000 DR6: 00000000fffe07f0 DR7: 00000000000000400 [560.843444] PKRU: 55555554 [560.846480] Call Trace: [560.849225] <TASK> [560.851580] ? show_trace_log_lvl+0x1d6/0x2ea [560.856488] ? show_trace_log_lvl+0x1d6/0x2ea [560.861379] ? amdgpu_ras_do_recovery+0x1b2/0x210 [amdgpu] [560.867778] ? show_regs.part.0+0x23/0x29 [560.872293] ? __die_body.cold+0x8/0xd [560.876502] ? die_addr+0x3e/0x60 [560.880238] ? exc_general_protection+0x1c5/0x410 [560.885532] ? asm_exc_general_protection+0x27/0x30 [560.891025] ? amdgpu_device_gpu_recover.cold+0xbff1/0xcf5 [amdgpu] [560.898323] amdgpu_ras_do_recovery+0x1b2/0x210 [amdgpu] [560.904520] process_one_work+0x228/0x3d0 How: In RAS recovery, mode-1 reset is issued from RAS fatal error handling and expected all the nodes in a hive to be reset. no need to issue another mode-1 during this procedure.
CVE-2024-35933	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btintel: Fix null ptr deref in btintel_read_version If hci_cmd_sync_complete() is triggered and skb is NULL, then hdev->req_skb is NULL, which will cause this issue.

CVE ID	Description
CVE-2024-35934	In the Linux kernel, the following vulnerability has been resolved: net/smc: reduce rtnl pressure in smc_pnet_create_pnetids_list() Many syzbot reports show extreme rtnl pressure, and many of them hint that smc acquires rtnl in netns creation for no good reason [1] This patch returns early from smc_pnet_net_init() if there is no netdevice yet. I am not even sure why smc_pnet_create_pnetids_list() even exists, because smc_pnet_netdev_event() is also calling smc_pnet_add_base_pnetid() when handling NETDEV_UP event. [1] extract of typical syzbot reports 2 locks held by syz-executor.3/12252: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns+0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init +0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.4/12253: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns+0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.1/12257: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns+0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.2/12261: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns+0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.0/12265: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns +0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.3/12268: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns +0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.4/12271: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns +0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.1/12274: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns +0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878 2 locks held by syz-executor.2/12280: #0: ffffffff8f369610 (pernet_ops_rwsem){++++}-{3:3}, at: copy_net_ns +0x4c7/0x7b0 net/core/net_namespace.c:491 #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_create_pnetids_list net/smc/smc_pnet.c:809 [inline] #1: ffffffff8f375b88 (rtnl_mutex){+.+}-{3:3}, at: smc_pnet_net_init+0x10a/0x1e0 net/smc/smc_pnet.c:878
CVE-2024-35945	In the Linux kernel, the following vulnerability has been resolved: net: phy: phy_device: Prevent nullptr exceptions on ISR If phydev->irq is set unconditionally, check for valid interrupt handler or fall back to polling mode to prevent nullptr exceptions in interrupt service routine.
CVE-2024-35948	In the Linux kernel, the following vulnerability has been resolved: bcachefs: Check for journal entries overrunning end of sb clean section Fix a missing bounds check in superblock validation. Note that we don't yet have repair code for this case - repair code for individual items is generally low priority, since the whole superblock is checksummed, validated prior to write, and we have backups.
CVE-2024-35950	In the Linux kernel, the following vulnerability has been resolved: drm/client: Fully protect modes[] with dev->mode_config.mutex The modes[] array contains pointers to modes on the connectors' mode lists, which are protected by dev->mode_config.mutex. Thus we need to extend modes[] the same protection or by the time we use it the elements may already be pointing to freed/reused memory.

CVE ID	Description
CVE-2024-35957	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Fix WARN_ON in iommu probe path Commit 1a75cc710b95 ("iommu/vt-d: Use rbtree to track iommu probed devices") adds all devices probed by the iommu driver in a rbtree indexed by the source ID of each device. It assumes that each device has a unique source ID. This assumption is incorrect and the VT-d spec doesn't state this requirement either. The reason for using a rbtree to track devices is to look up the device with PCI bus and devfunc in the paths of handling ATS invalidation time out error and the PRI I/O page faults. Both are PCI ATS feature related. Only track the devices that have PCI ATS capabilities in the rbtree to avoid unnecessary WARN_ON in the iommu probe path. Otherwise, on some platforms below kernel splat will be displayed and the iommu probe results in failure. WARNING: CPU: 3 PID: 166 at drivers/iommu/intel/iommu.c:158 intel_iommu_probe_device+0x319/0xd90 Call Trace: <TASK> ? __warn+0x7e/0x180 ? intel_iommu_probe_device+0x319/0xd90 ? report_bug+0x1f8/0x200 ? handle_bug+0x3c/0x70 ? exc_invalid_op+0x18/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? intel_iommu_probe_device+0x319/0xd90 ? debug_mutex_init+0x37/0x50 __iommu_probe_device+0xf2/0x4f0 iommu_probe_device+0x22/0x70 iommu_bus_notifier+0x1e/0x40 notifier_call_chain+0x46/0x150 blocking_notifier_call_chain+0x42/0x60 bus_notify+0x2f/0x50 device_add+0x5ed/0x7e0 platform_device_add+0xf5/0x240 mfd_add_devices+0x3f9/0x500 ? preempt_count_add+0x4c/0xa0 ? up_write+0xa2/0x1b0 ? __debugfs_create_file+0xe3/0x150 intel_lpss_probe+0x49f/0x5b0 ? pci_conf1_write+0xa3/0xf0 intel_lpss_pci_probe+0xcf/0x110 [intel_lpss_pci] pci_device_probe+0x95/0x120 really_probe+0xd9/0x370 ? pfx__driver_attach+0x10/0x10 __driver_probe_device+0x73/0x150 driver_probe_device+0x19/0xa0 __driver_attach+0xb6/0x180 ? pfx__driver_attach+0x10/0x10 bus_for_each_dev+0x77/0xd0 bus_add_driver+0x114/0x210 driver_register+0x5b/0x110 ? pfx_intel_lpss_pci_driver_init+0x10/0x10 [intel_lpss_pci] do_one_initcall+0x57/0x2b0 ? kmalloc_trace+0x21e/0x280 ? do_init_module+0x1e/0x210 do_init_module+0x5f/0x210 load_module+0x1d37/0x1fc0 ? init_module_from_file+0x86/0xd0 init_module_from_file+0x86/0xd0 idempotent_init_module+0x17c/0x230 __x64_sys_finit_module+0x56/0xb0 do_syscall_64+0x6e/0x140 entry_SYSCALL_64_after_hwframe+0x71/0x79
CVE-2024-35961	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Register devlink first under devlink lock In case device is having a non fatal FW error during probe, the driver will report the error to user via devlink. This will trigger a WARN_ON, since mlx5 is calling devlink_register() last. In order to avoid the WARN_ON[1], change mlx5 to invoke devl_register() first under devlink lock. [1] WARNING: CPU: 5 PID: 227 at net/devlink/health.c:483 devlink_recover_notify.constprop.0+0xb8/0xc0 CPU: 5 PID: 227 Comm: kworker/u16:3 Not tainted 6.4.0-rc5_for_upstream_min_debug_2023_06_12_12_38 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 Workqueue: mlx5_health0000:08:00.0 mlx5_fw_reporter_err_work [mlx5_core] RIP: 0010:devlink_recover_notify.constprop.0+0xb8/0xc0 Call Trace: <TASK> ? __warn+0x79/0x120 ? devlink_recover_notify.constprop.0+0xb8/0xc0 ? report_bug+0x17c/0x190 ? handle_bug+0x3c/0x60 ? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 ? devlink_recover_notify.constprop.0+0xb8/0xc0 devlink_health_report+0x4a/0x1c0 mlx5_fw_reporter_err_work+0xa4/0xd0 [mlx5_core] process_one_work+0x1bb/0x3c0 ? process_one_work+0x3c0/0x3c0 worker_thread+0x4d/0x3c0 ? process_one_work+0x3c0/0x3c0 kthread+0xc6/0xf0 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x1f/0x30 </TASK>
CVE-2024-35963	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sock: Fix not validating setsockopt user input Check user input length before copying data.
CVE-2024-35965	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix not validating setsockopt user input Check user input length before copying data.
CVE-2024-35966	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: Fix not validating setsockopt user input syzbot reported rfcomm_sock_setsockopt_old() is copying data without checking user input length. BUG: KASAN: slab-out-of-bounds in copy_from_sockptr_offset include/linux/sockptr.h:49 [inline] BUG: KASAN: slab-out-of-bounds in copy_from_sockptr include/linux/sockptr.h:55 [inline] BUG: KASAN: slab-out-of-bounds in rfcomm_sock_setsockopt_old net/bluetooth/rfcomm/sock.c:632 [inline] BUG: KASAN: slab-out-of-bounds in rfcomm_sock_setsockopt +0x893/0xa70 net/bluetooth/rfcomm/sock.c:673 Read of size 4 at addr ffff8880209a8bc3 by task syz-executor632/5064
CVE-2024-35967	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SCO: Fix not validating setsockopt user input syzbot reported sco_sock_setsockopt() is copying data without checking user input length. BUG: KASAN: slab-out-of-bounds in copy_from_sockptr_offset include/linux/sockptr.h:49 [inline] BUG: KASAN: slab-out-of-bounds in copy_from_sockptr include/linux/sockptr.h:55 [inline] BUG: KASAN: slab-out-of-bounds in sco_sock_setsockopt+0xc0b/0xf90 net/bluetooth/sco.c:893 Read of size 4 at addr ffff88805f7b15a3 by task syz-executor.5/12578

CVE ID	Description
CVE-2024-35969	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix race condition between <code>ipv6_get_ifaddr</code> and <code>ipv6_del_addr</code> Although <code>ipv6_get_ifaddr</code> walks <code>inet6_addr_lst</code> under the RCU lock, it still means <code>hlist_for_each_entry_rcu</code> can return an item that got removed from the list. The memory itself of such item is not freed thanks to RCU but nothing guarantees the actual content of the memory is sane. In particular, the reference count can be zero. This can happen if <code>ipv6_del_addr</code> is called in parallel. <code>ipv6_del_addr</code> removes the entry from <code>inet6_addr_lst</code> (<code>hlist_del_init_rcu(&ifp->addr_lst)</code>) and drops all references (<code>__in6_ifa_put(ifp) + in6_ifa_put(ifp)</code>). With bad enough timing, this can happen: 1. In <code>ipv6_get_ifaddr</code>, <code>hlist_for_each_entry_rcu</code> returns an entry. 2. Then, the whole <code>ipv6_del_addr</code> is executed for the given entry. The reference count drops to zero and <code>kfree_rcu</code> is scheduled. 3. <code>ipv6_get_ifaddr</code> continues and tries to increments the reference count (<code>in6_ifa_hold</code>). 4. The rcu is unlocked and the entry is freed. 5. The freed entry is returned. Prevent increasing of the reference count in such case. The name <code>in6_ifa_hold_safe</code> is chosen to mimic the existing <code>fib6_info_hold_safe</code>. [41.506330] refcount_t: addition on 0; use-after-free. [41.506760] WARNING: CPU: 0 PID: 595 at lib/refcount.c:25 refcount_warn_saturate+0xa5/0x130 [41.507413] Modules linked in: veth bridge stp llc [41.507821] CPU: 0 PID: 595 Comm: python3 Not tainted 6.9.0-rc2.main-00208-g49563be82afa #14 [41.508479] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) [41.509163] RIP: 0010:refcount_warn_saturate+0xa5/0x130 [41.509586] Code: ad ff 90 0f 0b 90 90 c3 cc cc cc cc 80 3d c0 30 ad 01 00 75 a0 c6 05 b7 30 ad 01 01 90 48 c7 c7 38 cc 7a 8c e8 cc 18 ad ff 90 <0f> 0b 90 90 c3 cc cc cc cc 80 3d 98 30 ad 01 00 0f 85 75 ff ff ff [41.510956] RSP: 0018:ffffbda3c026baf0 EFLAGS: 00010282 [41.511368] RAX: 0000000000000000 RBX: ffff9e9c46914800 RCX: 0000000000000000 [41.511910] RDX: ffff9e9c7ec29c00 RSI: ffff9e9c7ec1c900 RDI: ffff9e9c7ec1c900 [41.512445] RBP: ffff9e9c43660c9c R08: 00000000000009ffb R09: 00000000ffffdfff [41.512998] R10: 00000000ffffdfff R11: ffffffff8ca58a40 R12: ffff9e9c4339a000 [41.513534] R13: 0000000000000001 R14: ffff9e9c438a0000 R15: ffff9e9c438a0000 [41.514086] FS: 00007fbc4cda1740(0000) GS:ffff9e9c7ec00000(0000) knlGS:0000000000000000 [41.514726] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [41.515176] CR2: 000056233b337d88 CR3: 000000000376e006 CR4: 0000000000370ef0 [41.515713] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [41.516252] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [41.516799] Call Trace: [41.517037] <TASK> [41.517249] ? __warn +0x7b/0x120 [41.517535] ? refcount_warn_saturate+0xa5/0x130 [41.517923] ? report_bug +0x164/0x190 [41.518240] ? handle_bug+0x3d/0x70 [41.518541] ? exc_invalid_op+0x17/0x70 [41.520972] ? asm_exc_invalid_op+0x1a/0x20 [41.521325] ? refcount_warn_saturate+0xa5/0x130 [41.521708] ipv6_get_ifaddr+0xda/0xe0 [41.522035] inet6_rtm_getaddr+0x342/0x3f0 [41.522376] ? __pfx_inet6_rtm_getaddr+0x10/0x10 [41.522758] rtnetlink_rcv_msg+0x334/0x3d0 [41.523102] ? netlink_unicast+0x30f/0x390 [41.523445] ? __pfx_rtnetlink_rcv_msg+0x10/0x10 [41.523832] netlink_rcv_skb+0x53/0x100 [41.524157] netlink_unicast+0x23b/0x390 [41.524484] netlink_sendmsg +0x1f2/0x440 [41.524826] __sys_sendto+0x1d8/0x1f0 [41.525145] __x64_sys_sendto+0x1f/0x30 [41.525467] do_syscall_64+0xa5/0x1b0 [41.525794] entry_SYSCALL_64_after_hwframe+0x72/0x7a [41.526213] RIP: 0033:0x7fbc4cfcea9a [41.526528] Code: d8 64 89 02 48 c7 c0 ff ff ff eb b8 0f 1f 00 f3 0f 1e fa 41 89 ca 64 8b 04 25 18 00 00 00 85 c0 75 15 b8 2c 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 7e c3 0f 1f 44 00 00 41 54 48 83 ec 30 44 89 [41.527942] RSP: 002b:00007f ---truncated---

CVE ID	Description
CVE-2024-35976	In the Linux kernel, the following vulnerability has been resolved: xsk: validate user input for XDP_{UMEM COMPLETION}_FILL_RING syzbot reported an illegal copy in xsk_setsockopt() [1] Make sure to validate setsockopt() @optlen parameter. [1] BUG: KASAN: slab-out-of-bounds in copy_from_sockptr_offset include/linux/sockptr.h:49 [inline] BUG: KASAN: slab-out-of-bounds in copy_from_sockptr include/linux/sockptr.h:55 [inline] BUG: KASAN: slab-out-of-bounds in xsk_setsockopt+0x909/0xa40 net/xdp/xsk.c:1420 Read of size 4 at addr ffff888028c6cde3 by task syz-executor.0/7549 CPU: 0 PID: 7549 Comm: syz-executor.0 Not tainted 6.8.0-syzkaller-08951-gfe46a7dd189e #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/27/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:114 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 copy_from_sockptr_offset include/linux/sockptr.h:49 [inline] copy_from_sockptr include/linux/sockptr.h:55 [inline] xsk_setsockopt+0x909/0xa40 net/xdp/xsk.c:1420 do_sock_setsockopt+0x3af/0x720 net/socket.c:2311 __sys_setsockopt+0x1ae/0x250 net/socket.c:2334 __do_sys_setsockopt net/socket.c:2343 [inline] __se_sys_setsockopt net/socket.c:2340 [inline] __x64_sys_setsockopt+0xb5/0xd0 net/socket.c:2340 do_syscall_64+0xfb/0x240 entry_SYSCALL_64_after_hwframe+0x6d/0x75 RIP: 0033:0x7fb40587de69 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 e1 20 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fb40665a0c8 EFLAGS: 00000246 ORIG_RAX: 0000000000000036 RAX: ffffffffda RBX: 00007fb4059abf80 RCX: 00007fb40587de69 RDX: 0000000000000005 RSI: 000000000000011b RDI: 0000000000000006 RBP: 00007fb4058ca47a R08: 0000000000000002 R09: 0000000000000000 R10: 0000000020001980 R11: 0000000000000246 R12: 0000000000000000 R13: 000000000000000b R14: 00007fb4059abf80 R15: 00007fff57ee4d08 </TASK> Allocated by task 7549: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:370 [inline] __kasan_kmalloc+0x98/0xb0 mm/kasan/common.c:387 kasan_kmalloc include/linux/kasan.h:211 [inline] __do_kmalloc_node mm/slub.c:3966 [inline] __kmalloc+0x233/0x4a0 mm/slub.c:3979 kmalloc include/linux/slab.h:632 [inline] __cgroup_bpf_run_filter_setsockopt+0xd2f/0x1040 kernel/bpf/cgroup.c:1869 do_sock_setsockopt+0x6b4/0x720 net/socket.c:2293 __sys_setsockopt+0x1ae/0x250 net/socket.c:2334 __do_sys_setsockopt net/socket.c:2343 [inline] __se_sys_setsockopt net/socket.c:2340 [inline] __x64_sys_setsockopt+0xb5/0xd0 net/socket.c:2340 do_syscall_64+0xfb/0x240 entry_SYSCALL_64_after_hwframe+0x6d/0x75 The buggy address belongs to the object at ffff888028c6cde0 which belongs to the cache kmalloc-8 of size 8 The buggy address is located 1 bytes to the right of allocated 2-byte region [fff888028c6cde0, ffff888028c6cde2) The buggy address belongs to the physical page: page:ffffea0000a31b00 refcount:1 mapcount:0 mapping:0000000000000000 index:0xffff888028c6c9c0 pfn:0x28c6c anon flags: 0xfff00000000800(slab node=0 zone=1 lastcpupid=0x7ff) page_type: 0xffffffffff raw: 00ffff0000000800 ffff888014c41280 0000000000000000 dead000000000001 raw: ffff888028c6c9c0 00000000080800057 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected page_owner tracks the page as allocated page last allocated via order 0, migratetype Unmovable, gfp_mask 0x112cc0(GFP_USER __GFP_NOWARN __GFP_NORETRY), pid 6648, tgid 6644 (syz-executor.0), ts 133906047828, free_ts 133859922223 set_page_owner include/linux/page_owner.h:31 [inline] post_alloc_hook+0x1ea/0x210 mm/page_alloc.c:1533 prep_new_page mm/page_alloc.c: ---truncated---
CVE-2024-35978	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix memory leak in hci_req_sync_complete() In 'hci_req_sync_complete()', always free the previous sync request state before assigning reference to a new one.

CVE ID	Description
CVE-2024-35979	In the Linux kernel, the following vulnerability has been resolved: raid1: fix use-after-free for original bio in raid1_write_request() r1_bio->bios[] is used to record new bios that will be issued to underlying disks, however, in raid1_write_request(), r1_bio->bios[] will set to the original bio temporarily. Meanwhile, if blocked_rdev is set, free_r1bio() will be called causing that all r1_bio->bios[] to be freed: raid1_write_request() r1_bio = alloc_r1bio(mddev, bio); -> r1_bio->bios[] is NULL for (i = 0; i < disks; i++) -> for each rdev in conf // first rdev is normal r1_bio->bios[0] = bio; -> set to original bio // second rdev is blocked if (test_bit(Blocked, &rdev->flags)) break if (blocked_rdev) free_r1bio() put_all_bios() bio_put(r1_bio->bios[0]) -> original bio is freed Test scripts: mdadm -CR /dev/md0 -l1 -n4 /dev/sd[abcd] --assume-clean fio -filename=/dev/md0 -ioengine=libaio -rw=write -bs=4k -numjobs=1 \ -iodepth=128 -name=test -direct=1 echo blocked > /sys/block/md0/md/rd2/state Test result: BUG bio-264 (Not tainted): Object already free ----- Allocated in mempool_alloc_slab +0x24/0x50 age=1 cpu=1 pid=869 kmem_cache_alloc+0x324/0x480 mempool_alloc_slab +0x24/0x50 mempool_alloc+0x6e/0x220 bio_alloc_bioset+0x1af/0x4d0 blkdev_direct_IO +0x164/0x8a0 blkdev_write_iter+0x309/0x440 aio_write+0x139/0x2f0 io_submit_one+0x5ca/0xb70 __do_sys_io_submit+0x86/0x270 __x64_sys_io_submit+0x22/0x30 do_syscall_64+0xb1/0x210 entry_SYSCALL_64_after_hwframe+0x6c/0x74 Freed in mempool_free_slab+0x1f/0x30 age=1 cpu=1 pid=869 kmem_cache_free+0x28c/0x550 mempool_free_slab+0x1f/0x30 mempool_free +0x40/0x100 bio_free+0x59/0x80 bio_put+0xf0/0x220 free_r1bio+0x74/0xb0 raid1_make_request +0xadf/0x1150 md_handle_request+0xc7/0x3b0 md_submit_bio+0x76/0x130 __submit_bio +0xd8/0x1d0 submit_bio_noacct_nocheck+0x1eb/0x5c0 submit_bio_noacct+0x169/0xd40 submit_bio +0xee/0x1d0 blkdev_direct_IO+0x322/0x8a0 blkdev_write_iter+0x309/0x440 aio_write+0x139/0x2f0 Since that bios for underlying disks are not allocated yet, fix this problem by using mempool_free() directly to free the r1_bio.
CVE-2024-35983	In the Linux kernel, the following vulnerability has been resolved: bounds: Use the right number of bits for power-of-two CONFIG_NR_CPUS bits_per() rounds up to the next power of two when passed a power of two. This causes crashes on some machines and configurations.
CVE-2024-35995	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Use access_width over bit_width for system memory accesses To align with ACPI 6.3+, since bit_width can be any 8-bit value, it cannot be depended on to be always on a clean 8b boundary. This was uncovered on the Cobalt 100 platform. SError Interrupt on CPU26, code 0xbe000011 -- SError CPU: 26 PID: 1510 Comm: systemd-udevd Not tainted 5.15.2.1-13 #1 Hardware name: MICROSOFT CORPORATION, BIOS MICROSOFT CORPORATION pstate: 62400009 (nZCv daif +PAN -UAO +TCO -DIT -SSBS BTYPE=) pc : cppc_get_perf_caps+0xec/0x410 lr : cppc_get_perf_caps+0xe8/0x410 sp : ffff8000155ab730 x29: ffff8000155ab730 x28: ffff0080139d0038 x27: ffff0080139d0078 x26: 0000000000000000 x25: ffff0080139d0058 x24: 00000000ffffffffff x23: ffff0080139d0298 x22: ffff0080139d0278 x21: 0000000000000000 x20: ffff00802b251910 x19: ffff0080139d0000 x18: ffffffff x17: 0000000000000000 x16: ffffd7e111bad04 x15: ffff00802b251008 x14: ffffffff x13: ffff013f1fd63300 x12: 0000000000000006 x11: ffffd7e128f4420 x10: 0000000000000000 x9: ffffd7e111badec x8: ffff00802b251980 x7: 0000000000000000 x6: ffff0080139d0028 x5: 0000000000000000 x4: ffff0080139d0018 x3: 00000000ffffffffff x2: 0000000000000008 x1: ffff8000155ab7a0 x0: 0000000000000000 Kernel panic - not syncing: Asynchronous SError Interrupt CPU: 26 PID: 1510 Comm: systemd-udevd Not tainted 5.15.2.1-13 #1 Hardware name: MICROSOFT CORPORATION, BIOS MICROSOFT CORPORATION Call trace: dump_backtrace+0x0/0x1e0 show_stack+0x24/0x30 dump_stack_lvl+0x8c/0xb8 dump_stack+0x18/0x34 panic+0x16c/0x384 add_taint+0x0/0xc0 arm64_serror_panic+0x7c/0x90 arm64_is_fatal_ras_serror+0x34/0xa4 do_serror+0x50/0x6c el1h_64_error_handler+0x40/0x74 el1h_64_error+0x7c/0x80 cppc_get_perf_caps+0xec/0x410 cppc_cpufreq_cpu_init+0x74/0x400 [cppc_cpufreq] cpufreq_online+0x2dc/0xa30 cpufreq_add_dev+0xc0/0xd4 subsys_interface_register +0x134/0x14c cpufreq_register_driver+0x1b0/0x354 cppc_cpufreq_init+0x1a8/0x1000 [cppc_cpufreq] do_one_initcall+0x50/0x250 do_init_module+0x60/0x27c load_module+0x2300/0x2570 __do_sys_finit_module+0xa8/0x114 __arm64_sys_finit_module+0x2c/0x3c invoke_syscall +0x78/0x100 el0_svc_common.constprop.0+0x180/0x1a0 do_el0_svc+0x84/0xa0 el0_svc+0x2c/0xc0 el0t_64_sync_handler+0xa4/0x12c el0t_64_sync+0x1a4/0x1a8 Instead, use access_width to determine the size and use the offset and width to shift and mask the bits to read/write out. Make sure to add a check for system memory since pcc redefines the access_width to subspace id. If access_width is not set, then fall back to using bit_width. [rjw: Subject and changelog edits, comment adjustments]
CVE-2024-36012	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: msft: fix slab-use-after-free in msft_do_close() Tying the msft->data lifetime to hdev by freeing it in hci_release_dev() to fix the following case: [use] msft_do_close() msft = hdev->msft_data; if (!msft) ... (1) <- passed. return; mutex_lock(&msft->filter_lock); ... (4) <- used after freed. [free] msft_unregister() msft = hdev->msft_data; hdev->msft_data = NULL; ... (2) kfree(msft); ... (3) <- msft is freed. ===== BUG: KASAN: slab-use-after-free in __mutex_lock_common kernel/locking/mutex.c:587 [inline] BUG: KASAN: slab-use-after-free in __mutex_lock+0x8f/0xc30 kernel/locking/mutex.c:752 Read of size 8 at addr ffff888106cbbca8 by task kworker/u5:2/309

CVE ID	Description
CVE-2024-36013	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix slab-use-after-free in l2cap_connect() Extend a critical section to prevent chan from early freeing. Also make the l2cap_connect() return type void. Nothing is using the returned value but it is ugly to return a potentially freed pointer. Making it void will help with backports because earlier kernels did use the return value. Now the compile will break for kernels where this patch is not a complete fix. Call stack summary: [use] l2cap_bredr_sig_cmd l2cap_connect # mutex_lock(&conn->chan_lock); # chan = pchan->ops->new_connection(pchan); <- alloc chan # __l2cap_chan_add(conn, chan); # l2cap_chan_hold(chan); # list_add(&chan->list, &conn->chan_l); ... (1) # mutex_unlock(&conn->chan_lock); chan->conf_state ... (4) <- use after free [free] l2cap_conn_del # mutex_lock(&conn->chan_lock); # foreach chan in conn->chan_l: ... (2) # l2cap_chan_put(chan); # l2cap_chan_destroy # kfree(chan) ... (3) <- chan freed # mutex_unlock(&conn->chan_lock); ===== BUG: KASAN: slab-use-after-free in instrument_atomic_read include/linux/instrumented.h:68 [inline] BUG: KASAN: slab-use-after-free in _test_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline] BUG: KASAN: slab-use-after-free in l2cap_connect+0xa67/0x11a0 net/bluetooth/l2cap_core.c:4260 Read of size 8 at addr ffff88810bf040a0 by task kworker/u3:1/311
CVE-2024-36015	In the Linux kernel, the following vulnerability has been resolved: ppdev: Add an error check in register_device In register_device, the return value of ida_simple_get is unchecked, in which ida_simple_get will use an invalid index value. To address this issue, index should be checked after ida_simple_get. When the index value is abnormal, a warning message should be printed, the port should be dropped, and the value should be recorded.
CVE-2024-36021	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when devlink reload during pf initialization The devlink reload process will access the hardware resources, but the register operation is done before the hardware is initialized. So, processing the devlink reload during initialization may lead to kernel crash. This patch fixes this by taking devl_lock during initialization.
CVE-2024-36114	Aircompressor is a library with ports of the Snappy, LZO, LZ4, and Zstandard compression algorithms to Java. All decompressor implementations of Aircompressor (LZ4, LZO, Snappy, Zstandard) can crash the JVM for certain input, and in some cases also leak the content of other memory of the Java process (which could contain sensitive information). When decompressing certain data, the decompressors try to access memory outside the bounds of the given byte arrays or byte buffers. Because Aircompressor uses the JDK class `sun.misc.Unsafe` to speed up memory access, no additional bounds checks are performed and this has similar security consequences as out-of-bounds access in C or C++, namely it can lead to non-deterministic behavior or crash the JVM. Users should update to Aircompressor 0.27 or newer where these issues have been fixed. When decompressing data from untrusted users, this can be exploited for a denial-of-service attack by crashing the JVM, or to leak other sensitive information from the Java process. There are no known workarounds for this issue.
CVE-2024-36331	Improper initialization of CPU cache memory could allow a privileged attacker with hypervisor access to overwrite SEV-SNP guest memory resulting in loss of data integrity.
CVE-2024-36347	Improper signature verification in AMD CPU ROM microcode patch loader may allow an attacker with local administrator privilege to load malicious microcode, potentially resulting in loss of integrity of x86 instruction execution, loss of confidentiality and integrity of data in x86 CPU privileged context and compromise of SMM execution environment.
CVE-2024-36350	A transient execution vulnerability in some AMD processors may allow an attacker to infer data from previous stores, potentially resulting in the leakage of privileged information.
CVE-2024-36357	A transient execution vulnerability in some AMD processors may allow an attacker to infer data in the L1D cache, potentially resulting in the leakage of sensitive information across privileged boundaries.
CVE-2024-36476	In the Linux kernel, the following vulnerability has been resolved: RDMA/rtrs: Ensure 'ib_sge list' is accessible Move the declaration of the 'ib_sge list' variable outside the 'always_invalidate' block to ensure it remains accessible for use throughout the function. Previously, 'ib_sge list' was declared within the 'always_invalidate' block, limiting its accessibility, then caused a 'BUG: kernel NULL pointer dereference'[1]. ? __die_body.cold+0x19/0x27 ? page_fault_oops+0x15a/0x2d0 ? search_module_extables+0x19/0x60 ? search_bpf_extables+0x5f/0x80 ? exc_page_fault+0x7e/0x180 ? asm_exc_page_fault+0x26/0x30 ? memcpy_orig+0xd5/0x140 rxe_mr_copy+0x1c3/0x200 [rdma_rxe] ? rxe_pool_get_index+0x4b/0x80 [rdma_rxe] copy_data+0xa5/0x230 [rdma_rxe] rxe_requester+0xd9b/0xf70 [rdma_rxe] ? finish_task_switch.isra.0+0x99/0x2e0 rxe_sender+0x13/0x40 [rdma_rxe] do_task+0x68/0x1e0 [rdma_rxe] process_one_work+0x177/0x330 worker_thread +0x252/0x390 ? __pfx_worker_thread+0x10/0x10 This change ensures the variable is available for subsequent operations that require it. [1] https://lore.kernel.org/linux-rdma/6a1f3e8f-deb0-49f9-bc69-a9b03ecfcd7@fujitsu.com/

CVE ID	Description
CVE-2024-36478	In the Linux kernel, the following vulnerability has been resolved: null_blk: fix null-ptr-dereference while configuring 'power' and 'submit_queues' Writing 'power' and 'submit_queues' concurrently will trigger kernel panic: Test script: modprobe null_blk nr_devices=0 mkdir -p /sys/kernel/config/nullb/nullb0 while true; do echo 1 > submit_queues; echo 4 > submit_queues; done & while true; do echo 1 > power; echo 0 > power; done Test result: BUG: kernel NULL pointer dereference, address: 0000000000000148 Oops: 0000 [#1] PREEMPT SMP RIP: 0010: __lock_acquire+0x41d/0x28f0 Call Trace: <TASK> lock_acquire+0x121/0x450 down_write+0x5f/0x1d0 simple_recursive_removal+0x12f/0x5c0 blk_mq_debugfs_unregister_hctxs+0x7c/0x100 blk_mq_update_nr_hw_queues+0x4a3/0x720 nullb_update_nr_hw_queues+0x71/0xf0 [null_blk] nullb_device_submit_queues_store+0x79/0xf0 [null_blk] configfs_write_iter+0x119/0x1e0 vfs_write+0x326/0x730 ksys_write+0x74/0x150 This is because del_gendisk() can concurrent with blk_mq_update_nr_hw_queues(): nullb_device_power_store nullb_apply_submit_queues null_del_dev del_gendisk nullb_update_nr_hw_queues if (!dev->nullb) // still set while gendisk is deleted return 0 blk_mq_update_nr_hw_queues dev->nullb = NULL Fix this problem by resuing the global mutex to protect nullb_device_power_store() and nullb_update_nr_hw_queues() from configfs.
CVE-2024-3651	A vulnerability was identified in the kjd/idna library, specifically within the `idna.encode()` function, affecting version 3.6. The issue arises from the function's handling of crafted input strings, which can lead to quadratic complexity and consequently, a denial of service condition. This vulnerability is triggered by a crafted input that causes the `idna.encode()` function to process the input with considerable computational load, significantly increasing the processing time in a quadratic manner relative to the input size.
CVE-2024-36613	FFmpeg n6.1.1 has a vulnerability in the DXA demuxer of the libavformat library allowing for an integer overflow, potentially resulting in a denial-of-service (DoS) condition or other undefined behavior.
CVE-2024-36616	An integer overflow in the component /libavformat/westwood_vqa.c of FFmpeg n6.1.1 allows attackers to cause a denial of service in the application via a crafted VQA file.
CVE-2024-36617	FFmpeg n6.1.1 has an integer overflow vulnerability in the FFmpeg CAF decoder.
CVE-2024-36618	FFmpeg n6.1.1 has a vulnerability in the AVI demuxer of the libavformat library which allows for an integer overflow, potentially resulting in a denial-of-service (DoS) condition.
CVE-2024-36619	FFmpeg n6.1.1 has a vulnerability in the WAVARC decoder of the libavcodec library which allows for an integer overflow when handling certain block types, leading to a denial-of-service (DoS) condition.
CVE-2024-36620	moby v25.0.0 - v26.0.2 is vulnerable to NULL Pointer Dereference via daemon/images/image_history.go.
CVE-2024-36621	moby v25.0.5 is affected by a Race Condition in builder/builder-next/adapters/snapshot/layer.go. The vulnerability could be used to trigger concurrent builds that call the EnsureLayer function resulting in resource leaks/exhaustion.
CVE-2024-36623	moby through v25.0.3 has a Race Condition vulnerability in the streamformatter package which can be used to trigger multiple concurrent write operations resulting in data corruption or application crashes.
CVE-2024-36880	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: qca: add missing firmware sanity checks Add the missing sanity checks when parsing the firmware files before downloading them to avoid accessing and corrupting memory beyond the vmallocated buffer.
CVE-2024-36881	In the Linux kernel, the following vulnerability has been resolved: mm/userfaultfd: reset ptes when close() for wr-protected ones Userfaultfd unregister includes a step to remove wr-protect bits from all the relevant pgtable entries, but that only covered an explicit UFFDIO_UNREGISTER ioctl, not a close() on the userfaultfd itself. Cover that too. This fixes a WARN trace. The only user visible side effect is the user can observe leftover wr-protect bits even if the user close()ed on a userfaultfd when releasing the last reference of it. However hopefully that should be harmless, and nothing bad should happen even if so. This change is now more important after the recent page-table-check patch we merged in mm-unstable (446dd9ad37d0 ("mm/page_table_check: support userfault wr-protect entries")), as we'll do sanity check on uffd-wp bits without vma context. So it's better if we can 100% guarantee no uffd-wp bit leftovers, to make sure each report will be valid.

CVE ID	Description
CVE-2024-36882	In the Linux kernel, the following vulnerability has been resolved: mm: use memalloc_nofs_save() in page_cache_ra_order() See commit f2c817bed58d ("mm: use memalloc_nofs_save in readahead path"), ensure that page_cache_ra_order() do not attempt to reclaim file-backed pages too, or it leads to a deadlock, found issue when test ext4 large folio. INFO: task DataXceiver for:7494 blocked for more than 120 seconds. "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:DataXceiver for state:D stack:0 pid:7494 ppid:1 flags:0x00000200 Call trace: __switch_to+0x14c/0x240 __schedule+0x82c/0xdd0 schedule+0x58/0xf0 io_schedule+0x24/0xa0 __folio_lock+0x130/0x300 migrate_pages_batch+0x378/0x918 migrate_pages+0x350/0x700 compact_zone+0x63c/0xb38 compact_zone_order+0xc0/0x118 try_to_compact_pages+0xb0/0x280 __alloc_pages_direct_compact+0x98/0x248 __alloc_pages+0x510/0x1110 alloc_pages+0x9c/0x130 folio_alloc+0x20/0x78 filemap_alloc_folio+0x8c/0x1b0 page_cache_ra_order+0x174/0x308 ondemand_readahead+0x1c8/0x2b8 page_cache_async_ra+0x68/0xb8 filemap_readahead.isra.0+0x64/0xa8 filemap_get_pages+0x3fc/0x5b0 filemap_splice_read+0xf4/0x280 ext4_file_splice_read+0x2c/0x48 [ext4] vfs_splice_read.part.0+0xa8/0x118 splice_direct_to_actor+0xbc/0x288 do_splice_direct+0x9c/0x108 do_sendfile+0x328/0x468 __arm64_sys_sendfile64+0x8c/0x148 invoke_syscall+0x4c/0x118 el0_svc_common.constprop.0+0xc8/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x4c/0x1f8 el0t_64_sync_handler+0xc0/0xc8 el0t_64_sync+0x188/0x190
CVE-2024-36884	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Use the correct type in nvidia_smmu_context_fault() This was missed because of the function pointer indirection. nvidia_smmu_context_fault() is also installed as a irq function, and the 'void *' was changed to a struct arm_smmu_domain. Since the iommu_domain is embedded at a non-zero offset this causes nvidia_smmu_context_fault() to miscompute the offset. Fixup the types. Unable to handle kernel NULL pointer dereference at virtual address 0000000000000120 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x000000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtbl: 4k pages, 48-bit VAs, pgdp=0000000107c9f000 [0000000000000120] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] SMP Modules linked in: CPU: 1 PID: 47 Comm: kworker/u25:0 Not tainted 6.9.0-0.rc7.58.eln136.aarch64 #1 Hardware name: Unknown NVIDIA Jetson Orin NX/NVIDIA Jetson Orin NX, BIOS 3.1-32827747 03/19/2023 Workqueue: events_unbound deferred_probe_work_func pstate: 604000c9 (nZCv daIF +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : nvidia_smmu_context_fault+0x1c/0x158 lr : __free_irq+0x1d4/0x2e8 sp : ffff80008044b6f0 x29: ffff80008044b6f0 x28: ffff000080a60b18 x27: fffffd32b5172e970 x26: 0000000000000000 x25: ffff0000802f5aac x24: ffff0000802f5a30 x23: ffff0000802f5b60 x22: 0000000000000057 x21: 0000000000000000 x20: ffff0000802f5a00 x19: ffff000087d4cd80 x18: ffffffff x17: 6234362066666666 x16: 6630303078302d30 x15: ffff00008156d888 x14: 0000000000000000 x13: ffff0000801db910 x12: ffff00008156d6d0 x11: 0000000000000003 x10: ffff0000801db918 x9: ffff32b50f94d9c x8: 1ffff0001032fda1 x7: ffff00008197ed00 x6: 000000000000000f x5: 000000000000010e x4: 000000000000010e x3: 0000000000000000 x2: ffff32b51720cd8 x1: ffff000087e6f700 x0: 0000000000000057 Call trace: nvidia_smmu_context_fault+0x1c/0x158 __free_irq+0x1d4/0x2e8 free_irq+0x3c/0x80 devm_free_irq+0x64/0xa8 arm_smmu_domain_free+0xc4/0x158 iommu_domain_free+0x44/0xa0 iommu_deinit_device+0xd0/0xf8 __iommu_group_remove_device+0xcc/0xe0 iommu_bus_notifier+0x64/0xa8 notifier_call_chain+0x78/0x148 blocking_notifier_call_chain+0x4c/0x90 bus_notify+0x44/0x70 device_del+0x264/0x3e8 pci_remove_bus_device+0x84/0x120 pci_remove_root_bus+0x5c/0xc0 dw_pcie_host_deinit+0x38/0xe0 tegra_pcie_config_rp+0xc0/0x1f0 tegra_pcie_dw_probe+0x34c/0x700 platform_probe+0x70/0xe8 really_probe+0xc8/0x3a0 __driver_probe_device+0x84/0x160 driver_probe_device+0x44/0x130 __device_attach_driver+0xc4/0x170 bus_for_each_drv+0x90/0x100 __device_attach+0xa8/0x1c8 device_initial_probe+0x1c/0x30 bus_probe_device+0xb0/0xc0 deferred_probe_work_func+0xbc/0x120 process_one_work+0x194/0x490 worker_thread+0x284/0x3b0 kthread+0xf4/0x108 ret_from_fork+0x10/0x20 Code: a9b97bfd 910003fd a9025bf5 f85a0035 (b94122a1)
CVE-2024-36903	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix potential uninit-value access in __ip6_make_skb() As it was done in commit fc1092f51567 ("ipv4: Fix uninit-value access in __ip_make_skb()") for IPv4, check FLOWI_FLAG_KNOWN_NH on fl6->flowi6_flags instead of testing HDRINCL on the socket to avoid a race condition which causes uninit-value access.
CVE-2024-36910	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Don't free decrypted memory In CoCo VMs it is possible for the untrusted host to cause set_memory_encrypted() or set_memory_decrypted() to fail such that an error is returned and the resulting memory is shared. Callers need to take care to handle these errors to avoid returning decrypted (shared) memory to the page allocator, which could lead to functional or security issues. The VMBus device UIO driver could free decrypted/shared pages if set_memory_decrypted() fails. Check the decrypted field in the gpadl to decide whether to free the memory.

CVE ID	Description
CVE-2024-36911	In the Linux kernel, the following vulnerability has been resolved: hv_netvsc: Don't free decrypted memory In CoCo VMs it is possible for the untrusted host to cause set_memory_encrypted() or set_memory_decrypted() to fail such that an error is returned and the resulting memory is shared. Callers need to take care to handle these errors to avoid returning decrypted (shared) memory to the page allocator, which could lead to functional or security issues. The netvsc driver could free decrypted/shared pages if set_memory_decrypted() fails. Check the decrypted field in the gpadl to decide whether to free the memory.
CVE-2024-36937	In the Linux kernel, the following vulnerability has been resolved: xdp: use flags field to disambiguate broadcast redirect When redirecting a packet using XDP, the bpf_redirect_map() helper will set up the redirect destination information in struct bpf_redirect_info (using the __bpf_xdp_redirect_map() helper function), and the xdp_do_redirect() function will read this information after the XDP program returns and pass the frame on to the right redirect destination. When using the BPF_F_BROADCAST flag to do multicast redirect to a whole map, __bpf_xdp_redirect_map() sets the 'map' pointer in struct bpf_redirect_info to point to the destination map to be broadcast. And xdp_do_redirect() reacts to the value of this map pointer to decide whether it's dealing with a broadcast or a single-value redirect. However, if the destination map is being destroyed before xdp_do_redirect() is called, the map pointer will be cleared out (by bpf_clear_redirect_map()) without waiting for any XDP programs to stop running. This causes xdp_do_redirect() to think that the redirect was to a single target, but the target pointer is also NULL (since broadcast redirects don't have a single target), so this causes a crash when a NULL pointer is passed to dev_map_enqueue(). To fix this, change xdp_do_redirect() to react directly to the presence of the BPF_F_BROADCAST flag in the 'flags' value in struct bpf_redirect_info to disambiguate between a single-target and a broadcast redirect. And only read the 'map' pointer if the broadcast flag is set, aborting if that has been cleared out in the meantime. This prevents the crash, while keeping the atomic (cmpxchg-based) clearing of the map pointer itself, and without adding any more checks in the non-broadcast fast path.
CVE-2024-36938	In the Linux kernel, the following vulnerability has been resolved: bpf, skmsg: Fix NULL pointer dereference in sk_psock_skb_ingress_enqueue Fix NULL pointer data-races in sk_psock_skb_ingress_enqueue() which syzbot reported [1]. [1] BUG: KCSAN: data-race in sk_psock_drop / sk_psock_skb_ingress_enqueue write to 0xffff88814b3278b8 of 8 bytes by task 10724 on cpu 1: sk_psock_stop_verdict net/core/skmsg.c:1257 [inline] sk_psock_drop +0x13e/0x1f0 net/core/skmsg.c:843 sk_psock_put include/linux/skmsg.h:459 [inline] sock_map_close +0x1a7/0x260 net/core/sock_map.c:1648 unix_release+0x4b/0x80 net/unix/af_unix.c:1048 __sock_release net/socket.c:659 [inline] sock_close+0x68/0x150 net/socket.c:1421 __fput +0x2c1/0x660 fs/file_table.c:422 __fput_sync+0x44/0x60 fs/file_table.c:507 __do_sys_close fs/open.c:1556 [inline] __se_sys_close+0x101/0x1b0 fs/open.c:1541 __x64_sys_close+0x1f/0x30 fs/open.c:1541 do_syscall_64+0xd3/0x1d0 entry_SYSCALL_64_after_hwframe+0x6d/0x75 read to 0xffff88814b3278b8 of 8 bytes by task 10713 on cpu 0: sk_psock_data_ready include/linux/skmsg.h:464 [inline] sk_psock_skb_ingress_enqueue+0x32d/0x390 net/core/skmsg.c:555 sk_psock_skb_ingress_self +0x185/0x1e0 net/core/skmsg.c:606 sk_psock_verdict_apply net/core/skmsg.c:1008 [inline] sk_psock_verdict_recv+0x3e4/0x4a0 net/core/skmsg.c:1202 unix_read_skb net/unix/af_unix.c:2546 [inline] unix_stream_read_skb+0x9e/0xf0 net/unix/af_unix.c:2682 sk_psock_verdict_data_ready +0x77/0x220 net/core/skmsg.c:1223 unix_stream_sendmsg+0x527/0x860 net/unix/af_unix.c:2339 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x140/0x180 net/socket.c:745 __sys_sendmsg+0x312/0x410 net/socket.c:2584 __sys_sendmsg net/socket.c:2638 [inline] __sys_sendmsg+0x1e9/0x280 net/socket.c:2667 __do_sys_sendmsg net/socket.c:2676 [inline] __se_sys_sendmsg net/socket.c:2674 [inline] __x64_sys_sendmsg+0x46/0x50 net/socket.c:2674 do_syscall_64+0xd3/0x1d0 entry_SYSCALL_64_after_hwframe+0x6d/0x75 value changed: 0xffffffff83d7feb0 -> 0x0000000000000000 Reported by Kernel Concurrency Sanitizer on: CPU: 0 PID: 10713 Comm: syz-executor.4 Tainted: G W 6.8.0-syzkaller-08951-gfe46a7dd189e #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/29/2024 Prior to this, commit 4cd12c6065df ("bpf, sockmap: Fix NULL pointer dereference in sk_psock_verdict_data_ready()") fixed one NULL pointer similarly due to no protection of saved_data_ready. Here is another different caller causing the same issue because of the same reason. So we should protect it with sk_callback_lock read lock because the writer side in the sk_psock_drop() uses "write_lock_bh(&sk->sk_callback_lock);". To avoid errors that could happen in future, I move those two pairs of lock into the sk_psock_data_ready(), which is suggested by John Fastabend.
CVE-2024-36944	In the Linux kernel, the following vulnerability has been resolved: Reapply "drm/pxl: simplify qxl_fence_wait" This reverts commit 07ed11afb68d94eadd4ffc082b97c2331307c5ea. Stephen Rostedt reports: "I went to run my tests on my VMs and the tests hung on boot up. Unfortunately, the most I ever got out was: [93.607888] Testing event system initcall: OK [93.667730] Running tests on all trace events: [93.669757] Testing all events: OK [95.631064] -----[cut here]----- Timed out after 60 seconds" and further debugging points to a possible circular locking dependency between the console_owner locking and the worker pool locking. Reverting the commit allows Steve's VM to boot to completion again. [This may obviously result in the "[TTM] Buffer eviction failed" messages again, which was the reason for that original revert. But at this point this seems preferable to a non-booting system...]

CVE ID	Description
CVE-2024-36955	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: intel-sdw-acpi: fix usage of device_get_named_child_node() The documentation for device_get_named_child_node() mentions this important point: " The caller is responsible for calling fwnode_handle_put() on the returned fwnode pointer. " Add fwnode_handle_put() to avoid a leaked reference.
CVE-2024-36962	In the Linux kernel, the following vulnerability has been resolved: net: ks8851: Queue RX packets in IRQ handler instead of disabling BHs Currently the driver uses local_bh_disable()/local_bh_enable() in its IRQ handler to avoid triggering net_rx_action() softirq on exit from netif_rx(). The net_rx_action() could trigger this driver .start_xmit callback, which is protected by the same lock as the IRQ handler, so calling the .start_xmit from netif_rx() from the IRQ handler critical section protected by the lock could lead to an attempt to claim the already claimed lock, and a hang. The local_bh_disable()/local_bh_enable() approach works only in case the IRQ handler is protected by a spinlock, but does not work if the IRQ handler is protected by mutex, i.e. this works for KS8851 with Parallel bus interface, but not for KS8851 with SPI bus interface. Remove the BH manipulation and instead of calling netif_rx() inside the IRQ handler code protected by the lock, queue all the received SKBs in the IRQ handler into a queue first, and once the IRQ handler exits the critical section protected by the lock, dequeue all the queued SKBs and push them all into netif_rx(). At this point, it is safe to trigger the net_rx_action() softirq, since the netif_rx() call is outside of the lock that protects the IRQ handler.
CVE-2024-36968	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix div-by-zero in l2cap_le_flowctl_init() l2cap_le_flowctl_init() can cause both div-by-zero and an integer overflow since hdev->le_mtu may not fall in the valid range. Move MTU from hci_dev to hci_conn to validate MTU and stop the connection process earlier if MTU is invalid. Also, add a missing validation in read_buffer_size() and make it return an error value if the validation fails. Now hci_conn_add() returns ERR_PTR() as it can fail due to the both a kzalloc failure and invalid MTU value. divide error: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 0 PID: 67 Comm: kworker/u5:0 Tainted: G W 6.9.0-rc5+ #20 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Workqueue: hci0 hci_rx_work RIP: 0010:l2cap_le_flowctl_init+0x19e/0x3f0 net/bluetooth/l2cap_core.c:547 Code: e8 17 17 0c 00 66 41 89 9f 84 00 00 00 bf 01 00 00 00 41 b8 02 00 00 00 4c 89 fe 4c 89 e2 89 d9 e8 27 17 0c 00 44 89 f0 31 d2 <66> f7 f3 89 c3 ff c3 4d 8d b7 88 00 00 00 4c 89 f0 48 c1 e8 03 42 RSP: 0018:ffff88810bc0f858 EFLAGS: 00010246 RAX: 00000000000002a0 RBX: 0000000000000000 RCX: dffffc0000000000 RDX: 0000000000000000 RSI: ffff88810bc0f7c0 RDI: ffff88810bc0f7c0 RBP: ffff88810bc0f880 R08: aa69db2dda70ff01 R09: 0000ffa000000000 R10: 0084000000ffa000 R11: 0000000000000000 R12: ffff88810d65a084 R13: dffffc0000000000 R14: 00000000000002a0 R15: ffff88810d65a000 FS: 0000000000000000(0000) GS:ffff88811ac00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000100 CR3: 0000000103268003 CR4: 0000000000770ef0 PKRU: 55555554 Call Trace: <TASK> l2cap_le_connect_req net/bluetooth/l2cap_core.c:4902 [inline] l2cap_le_sig_cmd net/bluetooth/l2cap_core.c:5420 [inline] l2cap_le_sig_channel net/bluetooth/l2cap_core.c:5486 [inline] l2cap_rcv_frame+0xe59d/0x11710 net/bluetooth/l2cap_core.c:6809 l2cap_rcv_acldata+0x544/0x10a0 net/bluetooth/l2cap_core.c:7506 hci_acldata_packet net/bluetooth/hci_core.c:3939 [inline] hci_rx_work+0x5e5/0xb20 net/bluetooth/hci_core.c:4176 process_one_work kernel/workqueue.c:3254 [inline] process_scheduled_works+0x90f/0x1530 kernel/workqueue.c:3335 worker_thread+0x926/0xe70 kernel/workqueue.c:3416 kthread+0x2e3/0x380 kernel/kthread.c:388 ret_from_fork+0x5c/0x90 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Modules linked in: ---[end trace 0000000000000000]---
CVE-2024-36970	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlmwifi: Use request_module_nowait This appears to work around a deadlock regression that came in with the LED merge in 6.9. The deadlock happens on my system with 24 iwlmwifi radios, so maybe it something like all worker threads are busy and some work that needs to complete cannot complete. [also remove unnecessary "load_module" var and now-wrong comment]

CVE ID	Description
CVE-2024-37891	urllib3 is a user-friendly HTTP client library for Python. When using urllib3's proxy support with `ProxyManager`, the `Proxy-Authorization` header is only sent to the configured proxy, as expected. However, when sending HTTP requests *without* using urllib3's proxy support, it's possible to accidentally configure the `Proxy-Authorization` header even though it won't have any effect as the request is not using a forwarding proxy or a tunneling proxy. In those cases, urllib3 doesn't treat the `Proxy-Authorization` HTTP header as one carrying authentication material and thus doesn't strip the header on cross-origin redirects. Because this is a highly unlikely scenario, we believe the severity of this vulnerability is low for almost all users. Out of an abundance of caution urllib3 will automatically strip the `Proxy-Authorization` header during cross-origin redirects to avoid the small chance that users are doing this on accident. Users should use urllib3's proxy support or disable automatic redirects to achieve safe processing of the `Proxy-Authorization` header, but we still decided to strip the header by default in order to further protect users who aren't using the correct approach. We believe the number of usages affected by this advisory is low. It requires all of the following to be true to be exploited: 1. Setting the `Proxy-Authorization` header without using urllib3's built-in proxy support. 2. Not disabling HTTP redirects. 3. Either not using an HTTPS origin server or for the proxy or target origin to redirect to a malicious origin. Users are advised to update to either version 1.26.19 or version 2.2.2. Users unable to upgrade may use the `Proxy-Authorization` header with urllib3's `ProxyManager`, disable HTTP redirects using `redirects=False` when sending requests, or not use the `Proxy-Authorization` header as mitigations.
CVE-2024-38384	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: fix list corruption from reorder of WRITE ->lqueued __blkcg_rstat_flush() can be run anytime, especially when blk_cgroup_bio_start is being executed. If WRITE of `->lqueued` is re-ordered with READ of `bisc->lnode.next` in the loop of __blkcg_rstat_flush(), `next_bisc` can be assigned with one stat instance being added in blk_cgroup_bio_start(), then the local list in __blkcg_rstat_flush() could be corrupted. Fix the issue by adding one barrier.
CVE-2024-38388	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda/cs_dsp_ctl: Use private_free for control cleanup Use the control private_free callback to free the associated data block. This ensures that the memory won't leak, whatever way the control gets destroyed. The original implementation didn't actually remove the ALSA controls in hda_cs_dsp_control_remove(). It only freed the internal tracking structure. This meant it was possible to remove/unload the amp driver while leaving its ALSA controls still present in the soundcard. Obviously attempting to access them could cause segfaults or at least dereferencing stale pointers.
CVE-2024-38543	In the Linux kernel, the following vulnerability has been resolved: lib/test_hmm.c: handle src_pfns and dst_pfns allocation failure The kcalloc() in dmirror_device_evict_chunk() will return null if the physical memory has run out. As a result, if src_pfns or dst_pfns is dereferenced, the null pointer dereference bug will happen. Moreover, the device is going away. If the kcalloc() fails, the pages mapping a chunk could not be evicted. So add a __GFP_NOFAIL flag in kcalloc(). Finally, as there is no need to have physically contiguous memory, Switch kcalloc() to kvmalloc() in order to avoid failing allocations.
CVE-2024-38544	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix seg fault in rxe_comp_queue_pkt In rxe_comp_queue_pkt() an incoming response packet skb is enqueued to the resp_pkts queue and then a decision is made whether to run the completer task inline or schedule it. Finally the skb is dereferenced to bump a 'hw' performance counter. This is wrong because if the completer task is already running in a separate thread it may have already processed the skb and freed it which can cause a seg fault. This has been observed infrequently in testing at high scale. This patch fixes this by changing the order of enqueueing the packet until after the counter is accessed.
CVE-2024-38556	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Add a timeout to acquire the command queue semaphore Prevent forced completion handling on an entry that has not yet been assigned an index, causing an out of bounds access on idx = -22. Instead of waiting indefinitely for the sem, blocking flow now waits for index to be allocated or a sem acquisition timeout before beginning the timer for FW completion. Kernel log example: mlx5_core 0000:06:00.0: wait_func_handle_exec_timeout:1128:(pid 185911): cmd[-22]: CREATE_UCTX(0xa04) No done completion
CVE-2024-38561	In the Linux kernel, the following vulnerability has been resolved: kunit: Fix kthread reference There is a race condition when a kthread finishes after the deadline and before the call to kthread_stop(), which may lead to use after free.
CVE-2024-38580	In the Linux kernel, the following vulnerability has been resolved: epoll: be better about file lifetimes epoll can call out to vfs_poll() with a file pointer that may race with the last 'fput()'. That would make f_count go down to zero, and while the ep->mtx locking means that the resulting file pointer tear-down will be blocked until the poll returns, it means that f_count is already dead, and any use of it won't actually get a reference to the file any more: it's dead regardless. Make sure we have a valid ref on the file pointer before we call down to vfs_poll() from the epoll routines.

CVE ID	Description
CVE-2024-38581	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/mes: fix use-after-free issue Delete fence fallback timer to fix the ramdom use-after-free issue. v2: move to amdgpu_mes.c
CVE-2024-38588	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix possible use-after-free issue in ftrace_location() KASAN reports a bug: BUG: KASAN: use-after-free in ftrace_location +0x90/0x120 Read of size 8 at addr ffff888141d40010 by task insmod/424 CPU: 8 PID: 424 Comm: insmod Tainted: G W 6.9.0-rc2+ [...] Call Trace: <TASK> dump_stack_lvl+0x68/0xa0 print_report +0xcf/0x610 kasan_report+0xb5/0xe0 ftrace_location+0x90/0x120 register_kprobe+0x14b/0xa40 kprobe_init+0x2d/0xff0 [kprobe_example] do_one_initcall+0x8f/0x2d0 do_init_module+0x13a/0x3c0 load_module+0x3082/0x33d0 init_module_from_file+0xd2/0x130 __x64_sys_finit_module +0x306/0x440 do_syscall_64+0x68/0x140 entry_SYSCALL_64_after_hwframe+0x71/0x79 The root cause is that, in lookup_rec(), ftrace record of some address is being searched in ftrace pages of some module, but those ftrace pages at the same time is being freed in ftrace_release_mod() as the corresponding module is being deleted: CPU1 CPU2 register_kprobes() { delete_module() { check_kprobe_address_safe() { arch_check_ftrace_location() { ftrace_location() { lookup_rec() // USE! ftrace_release_mod() // Free! To fix this issue: 1. Hold rcu lock as accessing ftrace pages in ftrace_location_range(); 2. Use ftrace_location_range() instead of lookup_rec() in ftrace_location(); 3. Call synchronize_rcu() before freeing any ftrace pages both in ftrace_process_locs()/ ftrace_release_mod()/ftrace_free_mem().
CVE-2024-38594	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: move the EST lock to struct stmmac_priv Reinitialize the whole EST structure would also reset the mutex lock which is embedded in the EST structure, and then trigger the following warning. To address this, move the lock to struct stmmac_priv. We also need to reacquire the mutex lock when doing this initialization. DEBUG_LOCKS_WARN_ON(lock->magic != lock) WARNING: CPU: 3 PID: 505 at kernel/locking/mutex.c:587 __mutex_lock+0xd84/0x1068 Modules linked in: CPU: 3 PID: 505 Comm: tc Not tainted 6.9.0-rc6-00053-g0106679839f7-dirty #29 Hardware name: NXP i.MX8MPlus EVK board (DT) pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=---) pc : __mutex_lock+0xd84/0x1068 lr : __mutex_lock+0xd84/0x1068 sp : fffffffc0864e3570 x29: fffffffc0864e3570 x28: fffffffc0817bdc78 x27: 0000000000000003 x26: fffffffc08c54f1808 x25: fffffffc08c9164080 x24: fffffffc080d723ac x23: 0000000000000000 x22: 0000000000000002 x21: 0000000000000000 x20: 0000000000000000 x19: fffffffc083bc3000 x18: fffffffc08117b080 x16: 0000000000000002 x15: fffffffc08d2d40000 x14: 000000000000002da x13: fffffffc08d2d404b8 x12: fffffffc082b5a5c8 x11: fffffffc082bca680 x10: fffffffc082bb2640 x9 : fffffffc082bb2698 x8 : 00000000000017fe8 x7 : c0000000ffffefff x6 : 0000000000000001 x5 : fffffff8178fe0d48 x4 : 0000000000000000 x3 : 0000000000000027 x2 : fffffff8178fe0d50 x1 : 0000000000000000 x0 : 0000000000000000 Call trace: __mutex_lock+0xd84/0x1068 mutex_lock_nested+0x28/0x34 tc_setup_taprio+0x118/0x68c stmmac_setup_tc+0x50/0xf0 taprio_change+0x868/0xc9c
CVE-2024-38600	In the Linux kernel, the following vulnerability has been resolved: ALSA: Fix deadlocks with kctl removals at disconnection In snd_card_disconnect(), we set card->shutdown flag at the beginning, call callbacks and do sync for card->power_ref_sleep waiters at the end. The callback may delete a kctl element, and this can lead to a deadlock when the device was in the suspended state. Namely: * A process waits for the power up at snd_power_ref_and_wait() in snd_ctl_info() or read/write() inside card->controls_rwsem. * The system gets disconnected meanwhile, and the driver tries to delete a kctl via snd_ctl_remove*(); it tries to take card->controls_rwsem again, but this is already locked by the above. Since the sleeper isn't woken up, this deadlocks. An easy fix is to wake up sleepers before processing the driver disconnect callbacks but right after setting the card->shutdown flag. Then all sleepers will abort immediately, and the code flows again. So, basically this patch moves the wait_event() call at the right timing. While we're at it, just to be sure, call wait_event_all() instead of wait_event(), although we don't use exclusive events on this queue for now.

CVE ID	Description
CVE-2024-38601	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix a race between readers and resize checks The reader code in <code>rb_get_reader_page()</code> swaps a new reader page into the ring buffer by doing <code>cmpxchg</code> on <code>old->list.prev->next</code> to point it to the new page. Following that, if the operation is successful, <code>old->list.next->prev</code> gets updated too. This means the underlying doubly-linked list is temporarily inconsistent, <code>page->prev->next</code> or <code>page->next->prev</code> might not be equal back to page for some page in the ring buffer. The <code>resize</code> operation in <code>ring_buffer_resize()</code> can be invoked in parallel. It calls <code>rb_check_pages()</code> which can detect the described inconsistency and stop further tracing: [190.271762] -----[cut here]----- [190.271771] WARNING: CPU: 1 PID: 6186 at kernel/trace/ring_buffer.c:1467 rb_check_pages.isra.0+0x6a/0xa0 [190.271789] Modules linked in: [...] [190.271991] Unloaded tainted modules: intel_uncore_frequency(E):1 skx_edac(E):1 [190.272002] CPU: 1 PID: 6186 Comm: cmd.sh Kdump: loaded Tainted: G E 6.9.0-rc6-default #5 158d3e1e6d0b091c34c3b96bfd99a1c58306d79f [190.272011] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.0-0-gd239552c-rebuilt.opensuse.org 04/01/2014 [190.272015] RIP: 0010:rb_check_pages.isra.0+0x6a/0xa0 [190.272023] Code: [...] [190.272028] RSP: 0018:ffff9c37463abb70 EFLAGS: 00010206 [190.272034] RAX: ffff8eba04b6cb80 RBX: 0000000000000007 RCX: ffff8eba01f13d80 [190.272038] RDX: ffff8eba01f130c0 RSI: ffff8eba04b6cd00 RDI: ffff8eba0004c700 [190.272042] RBP: ffff8eba0004c700 R08: 0000000000010002 R09: 0000000000000000 [190.272045] R10: 00000000ffff7f52 R11: ffff8eba7f600000 R12: ffff8eba0004c720 [190.272049] R13: ffff8eba00223a00 R14: 0000000000000008 R15: ffff8eba067a8000 [190.272053] FS: 00007f1bd64752c0(0000) GS:ffff8eba7f680000(0000) knlGS:0000000000000000 [190.272057] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [190.272061] CR2: 00007f1bd6662590 CR3: 000000010291e001 CR4: 0000000000370ef0 [190.272070] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [190.272073] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [190.272077] Call Trace: [190.272098] <TASK> [190.272189] ring_buffer_resize+0x2ab/0x460 [190.272199] __tracing_resize_ring_buffer.part.0+0x23/0xa0 [190.272206] tracing_resize_ring_buffer+0x65/0x90 [190.272216] tracing_entries_write+0x74/0xc0 [190.272225] vfs_write+0xf5/0x420 [190.272248] ksys_write+0x67/0xe0 [190.272256] do_syscall_64+0x82/0x170 [190.272363] entry_SYSCALL_64_after_hwframe+0x76/0x7e [190.272373] RIP: 0033:0x7f1bd657d263 [190.272381] Code: [...] [190.272385] RSP: 002b:00007ffe72b643f8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 [190.272391] RAX: ffffffff8bda RBX: 0000000000000002 RCX: 00007f1bd657d263 [190.272395] RDX: 0000000000000002 RSI: 0000555a6eb538e0 RDI: 0000000000000001 [190.272398] RBP: 0000555a6eb538e0 R08: 000000000000000a R09: 0000000000000000 [190.272401] R10: 0000555a6eb55190 R11: 0000000000000246 R12: 00007f1bd6662500 [190.272404] R13: 0000000000000002 R14: 00007f1bd6667c00 R15: 0000000000000002 [190.272412] </TASK> [190.272414] ---[end trace 0000000000000000]--- Note that <code>ring_buffer_resize()</code> calls <code>rb_check_pages()</code> only if the parent trace_buffer has recording disabled. Recent commit d78ab792705c ("tracing: Stop current tracer when resizing buffer") causes that it is now always the case which makes it more likely to experience this issue. The window to hit this race is nonetheless very small. To help reproducing it, one can add a delay loop in <code>rb_get_reader_page()</code>: <code>ret = rb_head_page_replace(reader, cpu_buffer->reader_page); if (!ret) goto spin; for (unsigned i = 0; i < 1U << 26; i++) /* inserted delay loop */ __asm__ __volatile__ (" : : \"memory\"); rb_list_head(reader->list.next->prev = &cpu_buffer->reader_page->list; .. ---truncated---</code>
CVE-2024-38605	In the Linux kernel, the following vulnerability has been resolved: ALSA: core: Fix NULL module pointer assignment at card init The commit 81033c6b584b ("ALSA: core: Warn on empty module") introduced a <code>WARN_ON()</code> for a NULL module pointer passed at <code>snd_card</code> object creation, and it also wraps the code around it with <code>'#ifdef MODULE'</code>. This works in most cases, but the devils are always in details. "MODULE" is defined when the target code (i.e. the sound core) is built as a module; but this doesn't mean that the caller is also built-in or not. Namely, when only the sound core is built-in (<code>CONFIG_SND=y</code>) while the driver is a module (<code>CONFIG_SND_USB_AUDIO=m</code>), the passed module pointer is ignored even if it's non-NULL, and <code>card->module</code> remains as NULL. This would result in the missing module reference up/down at the device open/close, leading to a race with the code execution after the module removal. For addressing the bug, move the assignment of <code>card->module</code> again out of <code>ifdef</code>. The <code>WARN_ON()</code> is still wrapped with <code>ifdef</code> because the module can be really NULL when all sound drivers are built-in. Note that we keep <code>'#ifdef MODULE'</code> for <code>WARN_ON()</code>, otherwise it would lead to a false-positive NULL module check. Admittedly it won't catch perfectly, i.e. no check is performed when <code>CONFIG_SND=y</code>. But, it's no real problem as it's only for debugging, and the condition is pretty rare.
CVE-2024-38607	In the Linux kernel, the following vulnerability has been resolved: macintosh/via-macii: Fix "BUG: sleeping function called from invalid context" The via-macii ADB driver calls <code>request_irq()</code> after disabling hard interrupts. But disabling interrupts isn't necessary here because the VIA shift register interrupt was masked during VIA1 initialization.

CVE ID	Description
CVE-2024-38608	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix netif state handling mlx5e_suspend cleans resources only if netif_device_present() returns true. However, mlx5e_resume changes the state of netif, via mlx5e_nic_enable, only if reg_state == NETREG_REGISTERED. In the below case, the above leads to NULL-ptr Oops[1] and memory leaks: mlx5e_probe mlx5e_resume mlx5e_attach_netdev mlx5e_nic_enable <-- netdev not reg, not calling netif_device_attach() register_netdev <-- failed for some reason. ERROR_FLOW: _mlx5e_suspend <-- netif_device_present return false, resources aren't freed :(Hence, clean resources in this case as well. [1] BUG: kernel NULL pointer dereference, address: 0000000000000000 PGD 0 P4D 0 Oops: 0010 [#1] SMP CPU: 2 PID: 9345 Comm: test-ovs-ct-gen Not tainted 6.5.0_for_upstream_min_debug_2023_09_05_16_01 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:0x0 Code: Unable to access opcode bytes at 0xfffffffffffffd6. RSP: 0018:ffff888178aaf758 EFLAGS: 00010246 Call Trace: <TASK> ? __die+0x20/0x60 ? page_fault_oops+0x14c/0x3c0 ? exc_page_fault+0x75/0x140 ? asm_exc_page_fault+0x22/0x30 notifier_call_chain+0x35/0xb0 blocking_notifier_call_chain+0x3d/0x60 mlx5_blocking_notifier_call_chain+0x22/0x30 [mlx5_core] mlx5_core_uplink_netdev_event_replay+0x3e/0x60 [mlx5_core] mlx5_mdev_netdev_track+0x53/0x60 [mlx5_ib] mlx5_ib_roce_init+0xc3/0x340 [mlx5_ib] __mlx5_ib_add+0x34/0xd0 [mlx5_ib] mlx5r_probe+0xe1/0x210 [mlx5_ib] ? auxiliary_match_id+0x6a/0x90 auxiliary_bus_probe+0x38/0x80 ? driver_sysfs_add+0x51/0x80 really_probe+0xc9/0x3e0 ? driver_probe_device+0x90/0x90 __driver_probe_device+0x80/0x160 driver_probe_device+0x1e/0x90 __device_attach_driver+0x7d/0x100 bus_for_each_drv+0x80/0xd0 __device_attach+0xbc/0x1f0 bus_probe_device+0x86/0xa0 device_add+0x637/0x840 __auxiliary_device_add+0x3b/0xa0 add_adev+0xc9/0x140 [mlx5_core] mlx5_rescan_drivers_locked+0x22a/0x310 [mlx5_core] mlx5_register_device+0x53/0xa0 [mlx5_core] mlx5_init_one_devl_locked+0x5c4/0x9c0 [mlx5_core] mlx5_init_one+0x3b/0x60 [mlx5_core] probe_one+0x44c/0x730 [mlx5_core] local_pci_probe+0x3e/0x90 pci_device_probe+0xbf/0x210 ? kernfs_create_link+0x5d/0xa0 ? sysfs_do_create_link_sd+0x60/0xc0 really_probe+0xc9/0x3e0 ? driver_probe_device+0x90/0x90 __driver_probe_device+0x80/0x160 driver_probe_device+0x1e/0x90 __device_attach_driver+0x7d/0x100 bus_for_each_drv+0x80/0xd0 __device_attach+0xbc/0x1f0 pci_bus_add_device+0x54/0x80 pci_iov_add_virtfn+0x2e6/0x320 sriov_enable+0x208/0x420 mlx5_core_sriov_configure+0x9e/0x200 [mlx5_core] sriov_numvfs_store+0xae/0x1a0 kernfs_fop_write_iter+0x10c/0x1a0 vfs_write+0x291/0x3c0 ksys_write+0x5f/0xe0 do_syscall_64+0x3d/0x90 entry_SYSCALL_64_after_hwframe+0x46/0xb0 CR2: 0000000000000000 ---[end trace 0000000000000000]---
CVE-2024-38618	In the Linux kernel, the following vulnerability has been resolved: ALSA: timer: Set lower bound of start tick time Currently ALSA timer doesn't have the lower limit of the start tick time, and it allows a very small size, e.g. 1 tick with 1ns resolution for hrtimer. Such a situation may lead to an unexpected RCU stall, where the callback repeatedly queuing the expire update, as reported by fuzzer. This patch introduces a sanity check of the timer start tick time, so that the system returns an error when a too small start size is set. As of this patch, the lower limit is hard-coded to 100us, which is small enough but can still work somehow.
CVE-2024-38635	In the Linux kernel, the following vulnerability has been resolved: soundwire: cadence: fix invalid PDI offset For some reason, we add an offset to the PDI, presumably to skip the PDI0 and PDI1 which are reserved for BPT. This code is however completely wrong and leads to an out-of-bounds access. We were just lucky so far since we used only a couple of PDIs and remained within the PDI array bounds. A Fixes: tag is not provided since there are no known platforms where the out-of-bounds would be accessed, and the initial code had problems as well. A follow-up patch completely removes this useless offset.
CVE-2024-38663	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: fix list corruption from resetting io stat Since commit 3b8cc6298724 ("blk-cgroup: Optimize blkcg_rstat_flush()"), each iostat instance is added to blkcg_percpu list, so blkcg_reset_stats() can't reset the stat instance by memset(), otherwise the llist may be corrupted. Fix the issue by only resetting the counter part.
CVE-2024-38808	In Spring Framework versions 5.3.0 - 5.3.38 and older unsupported versions, it is possible for a user to provide a specially crafted Spring Expression Language (SpEL) expression that may cause a denial of service (DoS) condition. Specifically, an application is vulnerable when the following is true: * The application evaluates user-supplied SpEL expressions.
CVE-2024-38809	Applications that parse ETags from "If-Match" or "If-None-Match" request headers are vulnerable to DoS attack. Users of affected versions should upgrade to the corresponding fixed version. Users of older, unsupported versions could enforce a size limit on "If-Match" and "If-None-Match" headers, e.g. through a Filter.
CVE-2024-38820	The fix for CVE-2022-22968 made disallowedFields patterns in DataBinder case insensitive. However, String.toLowerCase() has some Locale dependent exceptions that could potentially result in fields not protected as expected.

CVE ID	Description
CVE-2024-38827	The usage of String.toLowerCase() and String.toUpperCase() has some Locale dependent exceptions that could potentially result in authorization rules not working properly.
CVE-2024-39282	In the Linux kernel, the following vulnerability has been resolved: net: wwan: t7xx: Fix FSM command timeout issue When driver processes the internal state change command, it use an asynchronous thread to process the command operation. If the main thread detects that the task has timed out, the asynchronous thread will panic when executing the completion notification because the main thread completion object has been released. BUG: unable to handle page fault for address: ffffffff8 PGD 1f283a067 P4D 1f283a067 PUD 1f283c067 PMD 0 Oops: 0000 [#1] PREEMPT SMP NOPTI RIP: 0010:complete_all+0x3e/0xa0 [...] Call Trace: <TASK> ? __die_body+0x68/0xb0 ? page_fault_oops+0x379/0x3e0 ? exc_page_fault+0x69/0xa0 ? asm_exc_page_fault+0x22/0x30 ? complete_all+0x3e/0xa0 fsm_main_thread+0xa3/0x9c0 [mtk_t7xx (HASH:1400 5)] ? __pfx_autoremove_wake_function+0x10/0x10 kthread+0xd8/0x110 ? __pfx_fsm_main_thread+0x10/0x10 [mtk_t7xx (HASH:1400 5)] ? __pfx_kthread+0x10/0x10 ret_from_fork+0x38/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1b/0x30 </TASK> [...] CR2: ffffffff8 ---[end trace 0000000000000000]--- Use the reference counter to ensure safe release as Sergey suggests: https://lore.kernel.org/all/da90f64c-260a-4329-87bf-1f9ff20a5951@gmail.com/
CVE-2024-39293	In the Linux kernel, the following vulnerability has been resolved: Revert "xsk: Support redirect to any socket bound to the same umem" This reverts commit 2863d665ea41282379f108e4da6c8a2366ba66db. This patch introduced a potential kernel crash when multiple napi instances redirect to the same AF_XDP socket. By removing the queue_index check, it is possible for multiple napi instances to access the Rx ring at the same time, which will result in a corrupted ring state which can lead to a crash when flushing the rings in __xsk_flush(). This can happen when the linked list of sockets to flush gets corrupted by concurrent accesses. A quick and small fix is not possible, so let us revert this for now.
CVE-2024-39331	In Emacs before 29.4, org-link-expand-abbrev in lisp/ol.el expands a %(...) link abbrev even when it specifies an unsafe function, such as shell-command-to-string. This affects Org Mode before 9.7.5.
CVE-2024-39473	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-topology: Fix input format query of process modules without base extension If a process module does not have base config extension then the same format applies to all of it's inputs and the process->base_config_ext is NULL, causing NULL dereference when specifically crafted topology and sequences used.
CVE-2024-39479	In the Linux kernel, the following vulnerability has been resolved: drm/i915/hwmon: Get rid of devm When both hwmon and hwmon drvdata (on which hwmon depends) are device managed resources, the expectation, on device unbind, is that hwmon will be released before drvdata. However, in i915 there are two separate code paths, which both release either drvdata or hwmon and either can be released before the other. These code paths (for device unbind) are as follows (see also the bug referenced below): Call Trace: release_nodes+0x11/0x70 devres_release_group+0xb2/0x110 component_unbind_all+0x8d/0xa0 component_del+0xa5/0x140 intel_pxp_tee_component_fini+0x29/0x40 [i915] intel_pxp_fini+0x33/0x80 [i915] i915_driver_remove+0x4c/0x120 [i915] i915_pci_remove+0x19/0x30 [i915] pci_device_remove+0x32/0xa0 device_release_driver_internal+0x19c/0x200 unbind_store+0x9c/0xb0 and Call Trace: release_nodes+0x11/0x70 devres_release_all+0x8a/0xc0 device_unbind_cleanup+0x9/0x70 device_release_driver_internal+0x1c1/0x200 unbind_store+0x9c/0xb0 This means that in i915, if use devm, we cannot guarantee that hwmon will always be released before drvdata. Which means that we have a uaf if hwmon sysfs is accessed when drvdata has been released but hwmon hasn't. The only way out of this seems to be do get rid of devm_ and release/free everything explicitly during device unbind. v2: Change commit message and other minor code changes v3: Cleanup from i915_hwmon_register on error (Armin Wolf) v4: Eliminate potential static analyzer warning (Rodrigo) Eliminate fetch_and_zero (Jani) v5: Restore previous logic for ddat_gt->hwmon_dev error return (Andi)
CVE-2024-39480	In the Linux kernel, the following vulnerability has been resolved: kdb: Fix buffer overflow during tab-complete Currently, when the user attempts symbol completion with the Tab key, kdb will use strncpy() to insert the completed symbol into the command buffer. Unfortunately it passes the size of the source buffer rather than the destination to strncpy() with predictably horrible results. Most obviously if the command buffer is already full but cp, the cursor position, is in the middle of the buffer, then we will write past the end of the supplied buffer. Fix this by replacing the dubious strncpy() calls with memmove()/memcpy() calls plus explicit boundary checks to make sure we have enough space before we start moving characters around.

CVE ID	Description
CVE-2024-39488	In the Linux kernel, the following vulnerability has been resolved: arm64: asm-bug: Add .align 2 to the end of __BUG_ENTRY When CONFIG_DEBUG_BUGVERBOSE=n, we fail to add necessary padding bytes to bug_table entries, and as a result the last entry in a bug table will be ignored, potentially leading to an unexpected panic(). All prior entries in the table will be handled correctly. The arm64 ABI requires that struct fields of up to 8 bytes are naturally-aligned, with padding added within a struct such that struct are suitably aligned within arrays. When CONFIG_DEBUG_BUGVERBOSE=y, the layout of a bug_entry is: struct bug_entry { signed int bug_addr_disp; // 4 bytes signed int file_disp; // 4 bytes unsigned short line; // 2 bytes unsigned short flags; // 2 bytes } ... with 12 bytes total, requiring 4-byte alignment. When CONFIG_DEBUG_BUGVERBOSE=n, the layout of a bug_entry is: struct bug_entry { signed int bug_addr_disp; // 4 bytes unsigned short flags; // 2 bytes < implicit padding > // 2 bytes } ... with 8 bytes total, with 6 bytes of data and 2 bytes of trailing padding, requiring 4-byte alignment. When we create a bug_entry in assembly, we align the start of the entry to 4 bytes, which implicitly handles padding for any prior entries. However, we do not align the end of the entry, and so when CONFIG_DEBUG_BUGVERBOSE=n, the final entry lacks the trailing padding bytes. For the main kernel image this is not a problem as find_bug() doesn't depend on the trailing padding bytes when searching for entries: for (bug = __start__bug_table; bug < __stop__bug_table; ++bug) if (bugaddr == bug_addr(bug)) return bug; However for modules, module_bug_finalize() depends on the trailing bytes when calculating the number of entries: mod->num_bugs = sechdrs[i].sh_size / sizeof(struct bug_entry); ... and as the last bug_entry lacks the necessary padding bytes, this entry will not be counted, e.g. in the case of a single entry: sechdrs[i].sh_size == 6 sizeof(struct bug_entry) == 8; sechdrs[i].sh_size / sizeof(struct bug_entry) == 0; Consequently module_find_bug() will miss the last bug_entry when it does: for (i = 0; i < mod->num_bugs; ++i, ++bug) if (bugaddr == bug_addr(bug)) goto out; ... which can lead to a kernel panic due to an unhandled bug. This can be demonstrated with the following module: static int __init buginit(void) { WARN(1, "hello\n"); return 0; } static void __exit bugexit(void) { } module_init(buginit); module_exit(bugexit); MODULE_LICENSE("GPL"); ... which will trigger a kernel panic when loaded: -----[cut here]----- hello Unexpected kernel BRK exception at EL1 Internal error: BRK handler: 00000000f2000800 [#1] PREEMPT SMP Modules linked in: hello(O+) CPU: 0 PID: 50 Comm: insmod Tainted: G O 6.9.1 #8 Hardware name: linux,dummy-virt (DT) pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : buginit +0x18/0x1000 [hello] lr : buginit+0x18/0x1000 [hello] sp : ffff800080533ae0 x29: ffff800080533ae0 x28: 0000000000000000 x27: 0000000000000000 x26: ffffaba8c4e70510 x25: ffff800080533c30 x24: ffffaba8c4a28a58 x23: 0000000000000000 x22: 0000000000000000 x21: ffff3947c0eab3c0 x20: ffffaba8c4e3f000 x19: ffffaba846464000 x18: 0000000000000006 x17: 0000000000000000 x16: ffffaba8c2492834 x15: 0720072007200720 x14: 0720072007200720 x13: ffffaba8c49b27c8 x12: 0000000000000312 x11: 0000000000000106 x10: ffffaba8c4a0a7c8 x9 : ffffaba8c49b27c8 x8 : 00000000ffffefff x7 : ffffaba8c4a0a7c8 x6 : 80000000ffff0000 x5 : 0000000000000107 x4 : 0000000000000000 x3 : 0000000000000000 x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff3947c0eab3c0 Call trace: buginit+0x18/0x1000 [hello] do_one_initcall+0x80/0x1c8 do_init_module+0x60/0x218 load_module+0x1ba4/0x1d70 __do_sys_init_module+0x198/0x1d0 __arm64_sys_init_module+0x1c/0x28 invoke_syscall+0x48/0x114 el0_svc ---truncated---
CVE-2024-39494	In the Linux kernel, the following vulnerability has been resolved: ima: Fix use-after-free on a dentry's dname.name -> d_name.name can change on rename and the earlier value can be freed; there are conditions sufficient to stabilize it (->d_lock on dentry, ->d_lock on its parent, ->i_rwsem exclusive on the parent's inode, rename_lock), but none of those are met at any of the sites. Take a stable snapshot of the name instead.

CVE ID	Description
CVE-2024-39500	In the Linux kernel, the following vulnerability has been resolved: sock_map: avoid race between sock_map_close and sk_psock_put sk_psock_get will return NULL if the refcount of psock has gone to 0, which will happen when the last call of sk_psock_put is done. However, sk_psock_drop may not have finished yet, so the close callback will still point to sock_map_close despite psock being NULL. This can be reproduced with a thread deleting an element from the sock map, while the second one creates a socket, adds it to the map and closes it. That will trigger the WARN_ON_ONCE: -----[cut here]----- WARNING: CPU: 1 PID: 7220 at net/core/sock_map.c:1701 sock_map_close+0x2a2/0x2d0 net/core/sock_map.c:1701 Modules linked in: CPU: 1 PID: 7220 Comm: syz-executor380 Not tainted 6.9.0-syzkaller-07726-g3c999d1ae3c7 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/02/2024 RIP: 0010:sock_map_close+0x2a2/0x2d0 net/core/sock_map.c:1701 Code: df e8 92 29 88 f8 48 8b 1b 48 89 d8 48 c1 e8 03 42 80 3c 20 00 74 08 48 89 df e8 79 29 88 f8 4c 8b 23 eb 89 e8 4f 15 23 f8 90 <0f> 0b 90 48 83 c4 08 5b 41 5c 41 5d 41 5e 41 5f 5d e9 13 26 3d 02 RSP: 0018:ffffc9000441fda8 EFLAGS: 00010293 RAX: ffffffff89731ae1 RBX: ffffffff94b87540 RCX: ffff888029470000 RDX: 0000000000000000 RSI: ffffffff8bcab5c0 RDI: ffffffff8c1faba0 RBP: 0000000000000000 R08: ffffffff92f9b61f R09: 1fffffffff25f36c3 R10: dffffc0000000000 R11: ffffbf5ff25f36c4 R12: ffffffff89731840 R13: ffff88804b587000 R14: ffff88804b587000 R15: ffffffff89731870 FS: 00005555e080380(0000) GS:ffff8880b9500000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000000207d4000 CR4: 000000000350ef0 Call Trace: <TASK> unix_release+0x87/0xc0 net/unix/af_unix.c:1048 __sock_release net/socket.c:659 [inline] sock_close+0x8e/0x240 net/socket.c:1421 __fput+0x42b/0x8a0 fs/file_table.c:422 __do_sys_close fs/open.c:1556 [inline] __se_sys_close fs/open.c:1541 [inline] __x64_sys_close+0x7f/0x110 fs/open.c:1541 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf5/0x240 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe +0x77/0x7f RIP: 0033:0x7fb37d618070 Code: 00 00 48 c7 c2 b8 ff ff ff f7 d8 64 89 02 b8 ff ff ff ff eb d4 e8 10 2c 00 00 80 3d 31 f0 07 00 00 74 17 b8 03 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 48 c3 0f 1f 80 00 00 00 00 48 83 ec 18 89 7c RSP: 002b:00007ffcd4a525d8 EFLAGS: 00000202 ORIG_RAX: 0000000000000003 RAX: ffffffff8ffda RBX: 0000000000000005 RCX: 00007fb37d618070 RDX: 0000000000000010 RSI: 00000000200001c0 RDI: 0000000000000004 RBP: 0000000000000000 R08: 0000000100000000 R09: 0000000100000000 R10: 0000000000000000 R11: 0000000000000202 R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 </TASK> Use sk_psock, which will only check that the pointer is not been set to NULL yet, which should only happen after the callbacks are restored. If, then, a reference can still be gotten, we may call sk_psock_stop and cancel psock->work. As suggested by Paolo Abeni, reorder the condition so the control flow is less convoluted. After that change, the reproducer does not trigger the WARN_ON_ONCE anymore.
CVE-2024-39509	In the Linux kernel, the following vulnerability has been resolved: HID: core: remove unnecessary WARN_ON() in implement() Syzkaller hit a warning [1] in a call to implement() when trying to write a value into a field of smaller size in an output report. Since implement() already has a warn message printed out with the help of hid_warn() and value in question gets trimmed with: ... value &= m; ... WARN_ON may be considered superfluous. Remove it to suppress future syzkaller triggers. [1] WARNING: CPU: 0 PID: 5084 at drivers/hid/hid-core.c:1451 implement drivers/hid/hid-core.c:1451 [inline] WARNING: CPU: 0 PID: 5084 at drivers/hid/hid-core.c:1451 hid_output_report+0x548/0x760 drivers/hid/hid-core.c:1863 Modules linked in: CPU: 0 PID: 5084 Comm: syz-executor424 Not tainted 6.9.0-rc7-syzkaller-00183-gcf87f46fd34d #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/02/2024 RIP: 0010:implement drivers/hid/hid-core.c:1451 [inline] RIP: 0010:hid_output_report+0x548/0x760 drivers/hid/hid-core.c:1863 ... Call Trace: <TASK> __usbhid_submit_report drivers/hid/usbhid/hid-core.c:591 [inline] usbhid_submit_report+0x43d/0x9e0 drivers/hid/usbhid/hid-core.c:636 hiddev_ioctl+0x138b/0x1f00 drivers/hid/usbhid/hiddev.c:726 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:904 [inline] __se_sys_ioctl+0xfc/0x170 fs/ioctl.c:890 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf5/0x240 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f ...
CVE-2024-39908	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.1 has some DoS vulnerabilities when it parses an XML that has many specific characters such as '<', '&' and '%>'. If you need to parse untrusted XMLs, you may be impacted to these vulnerabilities. The REXML gem 3.3.2 or later include the patches to fix these vulnerabilities. Users are advised to upgrade. Users unable to upgrade should avoid parsing untrusted XML strings.
CVE-2024-39920	The TCP protocol in RFC 9293 has a timing side channel that makes it easier for remote attackers to infer the content of one TCP connection from a client system (to any server), when that client system is concurrently obtaining TCP data at a slow rate from an attacker-controlled server, aka the "SnailLoad" issue. For example, the attack can begin by measuring RTTs via the TCP segments whose role is to provide an ACK control bit and an Acknowledgment Number.

CVE ID	Description
CVE-2024-40635	containerd is an open-source container runtime. A bug was found in containerd prior to versions 1.6.38, 1.7.27, and 2.0.4 where containers launched with a User set as a `UID:GID` larger than the maximum 32-bit signed integer can cause an overflow condition where the container ultimately runs as root (UID 0). This could cause unexpected behavior for environments that require containers to run as a non-root user. This bug has been fixed in containerd 1.6.38, 1.7.27, and 2.0.4. As a workaround, ensure that only trusted images are used and that only trusted users have permissions to import images.
CVE-2024-40905	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix possible race in __fib6_drop_pcpu_from() syzbot found a race in __fib6_drop_pcpu_from() [1] If compiler reads more than once (*ppcpu_rt), second read could read NULL, if another cpu clears the value in rt6_get_pcpu_route(). Add a READ_ONCE() to prevent this race. Also add rcu_read_lock()/rcu_read_unlock() because we rely on RCU protection while dereferencing pcpu_rt. [1] Oops: general protection fault, probably for non-canonical address 0xdffffc0000000012: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000090-0x0000000000000097] CPU: 0 PID: 7543 Comm: kworker/u8:17 Not tainted 6.10.0-rc1-syzkaller-00013-g2bfcfd584ff5 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 04/02/2024 Workqueue: netns cleanup_net RIP: 0010: __fib6_drop_pcpu_from.part.0+0x10a/0x370 net/ipv6/ipv6_fib.c:984 Code: f8 48 c1 e8 03 80 3c 28 00 0f 85 16 02 00 00 4d 8b 3f 4d 85 ff 74 31 e8 74 a7 fa f7 49 8d bf 90 00 00 00 48 89 f8 48 c1 e8 03 <80> 3c 28 00 0f 85 1e 02 00 00 49 8b 87 90 00 00 00 48 8b 0c 24 48 RSP: 0018:ffffc900040df070 EFLAGS: 00010206 RAX: 0000000000000012 RBX: 0000000000000001 RCX: ffffffff89932e16 RDX: ffff888049dd1e00 RSI: ffffffff89932d7c RDI: 0000000000000091 RBP: dffffc0000000000 R08: 0000000000000005 R09: 0000000000000007 R10: 0000000000000001 R11: 0000000000000006 R12: ffff88807fa080b8 R13: ffff8bfff1a9a07d R14: ffffd100ff41022 R15: 0000000000000001 FS: 0000000000000000(0000) GS:ffff8880b9200000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000000005003 CR2: 0000001b32c26000 CR3: 000000005d56e000 CR4: 0000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __fib6_drop_pcpu_from net/ipv6/ipv6_fib.c:966 [inline] fib6_drop_pcpu_from net/ipv6/ipv6_fib.c:1027 [inline] fib6_purge_rt +0x7f2/0x9f0 net/ipv6/ipv6_fib.c:1038 fib6_del_route net/ipv6/ipv6_fib.c:1998 [inline] fib6_del +0xa70/0x17b0 net/ipv6/ipv6_fib.c:2043 fib6_clean_node+0x426/0x5b0 net/ipv6/ipv6_fib.c:2205 fib6_walk_continue+0x44f/0x8d0 net/ipv6/ipv6_fib.c:2127 fib6_walk+0x182/0x370 net/ipv6/ipv6_fib.c:2175 fib6_clean_tree+0xd7/0x120 net/ipv6/ipv6_fib.c:2255 __fib6_clean_all+0x100/0x2d0 net/ipv6/ipv6_fib.c:2271 rt6_sync_down_dev net/ipv6/route.c:4906 [inline] rt6_disable_ip+0x7ed/0xa00 net/ipv6/route.c:4911 addrconf_ifdown.isra.0+0x117/0x1b40 net/ipv6/addrconf.c:3855 addrconf_notify +0x223/0x19e0 net/ipv6/addrconf.c:3778 notifier_call_chain+0xb9/0x410 kernel/notifier.c:93 call_netdevice_notifiers_info+0xbe/0x140 net/core/dev.c:1992 call_netdevice_notifiers_extack net/core/dev.c:2030 [inline] call_netdevice_notifiers net/core/dev.c:2044 [inline] dev_close_many+0x333/0x6a0 net/core/dev.c:1585 unregister_netdevice_many_notify+0x46d/0x19f0 net/core/dev.c:11193 unregister_netdevice_many net/core/dev.c:11276 [inline] default_device_exit_batch+0x85b/0xae0 net/core/dev.c:11759 ops_exit_list+0x128/0x180 net/core/net_namespace.c:178 cleanup_net+0x5b7/0xbf0 net/core/net_namespace.c:640 process_one_work+0x9fb/0x1b60 kernel/workqueue.c:3231 process_scheduled_works kernel/workqueue.c:3312 [inline] worker_thread+0x6c8/0xf70 kernel/workqueue.c:3393 kthread+0x2c1/0x3a0 kernel/kthread.c:389 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244
CVE-2024-40919	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Adjust logging of firmware messages in case of released token in __hwrn_send() In case of token is released due to token->state == BNXT_HWRM_DEFERRED, released token (set to NULL) is used in log messages. This issue is expected to be prevented by HWRM_ERR_CODE_PF_UNAVAILABLE error code. But this error code is returned by recent firmware. So some firmware may not return it. This may lead to NULL pointer dereference. Adjust this issue by adding token pointer check. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2024-40923	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: disable rx data ring on dma allocation failure When vmxnet3_rq_create() fails to allocate memory for rq->data_ring.base, the subsequent call to vmxnet3_rq_destroy_all_rxdataring does not reset rq->data_ring.desc_size for the data ring that failed, which presumably causes the hypervisor to reference it on packet reception. To fix this bug, rq->data_ring.desc_size needs to be set to 0 to tell the hypervisor to disable this feature. [95.436876] kernel BUG at net/core/skbuff.c:207! [95.439074] invalid opcode: 0000 [#1] PREEMPT SMP NOPTI [95.440411] CPU: 7 PID: 0 Comm: swapper/7 Not tainted 6.9.3-dirty #1 [95.441558] Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 12/12/2018 [95.443481] RIP: 0010:skb_panic+0x4d/0x4f [95.444404] Code: 4f 70 50 8b 87 c0 00 00 00 50 8b 87 bc 00 00 00 50 ff b7 d0 00 00 00 4c 8b 8f c8 00 00 00 48 c7 c7 68 e8 be 9f e8 63 58 f9 ff <0f> 0b 48 8b 14 24 48 c7 c1 d0 73 65 9f e8 a1 ff ff ff 48 8b 14 24 [95.447684] RSP: 0018:ffffa13340274dd0 EFLAGS: 00010246 [95.448762] RAX: 0000000000000089 RBX: ffff8bbcb72b02d0 RCX: 0000000000000083f [95.450148] RDX: 0000000000000000 RSI: 00000000000000f6 RDI: 0000000000000083f [95.451520] RBP: 000000000000002d R08: 0000000000000000 R09: fffffa13340274c60 [95.452886] R10: ffffffffa04ed468 R11: 0000000000000002 R12: 0000000000000000 [95.454293] R13: ffff8bbdb3c2d0 R14: ffff8bbdbd829e0 R15: ffff8bbdbd809e0 [95.455682] FS: 0000000000000000(0000) GS:ffff8beefdb0000(0000) knlGS:0000000000000000 [95.457178] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [95.458340] CR2: 00007fd0d1f650c8 CR3: 0000000115f28000 CR4: 00000000000406f0 [95.459791] Call Trace: [95.460515] <IRQ> [95.461180] ? __die_body.cold+0x19/0x27 [95.462150] ? die+0x2e/0x50 [95.462976] ? do_trap+0xca/0x110 [95.463973] ? do_error_trap+0x6a/0x90 [95.464966] ? skb_panic+0x4d/0x4f [95.465901] ? exc_invalid_op+0x50/0x70 [95.466849] ? skb_panic+0x4d/0x4f [95.467718] ? asm_exc_invalid_op+0x1a/0x20 [95.468758] ? skb_panic+0x4d/0x4f [95.469655] skb_put.cold+0x10/0x10 [95.470573] vmxnet3_rq_rx_complete+0x862/0x11e0 [vmxnet3] [95.471853] vmxnet3_poll_rx_only+0x36/0xb0 [vmxnet3] [95.473185] __napi_poll+0x2b/0x160 [95.474145] net_rx_action+0x2c6/0x3b0 [95.475115] handle_softirqs+0xe7/0x2a0 [95.476122] __irq_exit_rcu+0x97/0xb0 [95.477109] common_interrupt+0x85/0xa0 [95.478102] </IRQ> [95.478846] <TASK> [95.479603] asm_common_interrupt+0x26/0x40 [95.480657] RIP: 0010:pv_native_safe_halt+0xf/0x20 [95.481801] Code: 22 d7 e9 54 87 01 00 0f 1f 40 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa eb 07 0f 00 2d 93 ba 3b 00 fb f4 <e9> 2c 87 01 00 66 66 2e 0f 1f 84 00 00 00 00 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 [95.485563] RSP: 0018:ffffa133400ffe58 EFLAGS: 00000246 [95.486882] RAX: 0000000000000400 RBX: ffff8bbcb1d14064 RCX: 0000000000000000 [95.488477] RDX: ffff8beefdb0000 RSI: ffff8bbcb1d14000 RDI: 0000000000000001 [95.490067] RBP: ffff8bbcb1d14064 R08: ffffffffa0652260 R09: 00000000000010d3 [95.491683] R10: 0000000000000018 R11: ffff8beefdb4764 R12: ffffffffa0652260 [95.493389] R13: ffffffffa06522e0 R14: 0000000000000001 R15: 0000000000000000 [95.495035] acpi_safe_halt+0x14/0x20 [95.496127] acpi_idle_do_entry+0x2f/0x50 [95.497221] acpi_idle_enter+0x7f/0xd0 [95.498272] cpuidle_enter_state+0x81/0x420 [95.499375] cpuidle_enter+0x2d/0x40 [95.500400] do_idle+0x1e5/0x240 [95.501385] cpu_startup_entry+0x29/0x30 [95.502422] start_secondary+0x11c/0x140 [95.503454] common_startup_64+0x13e/0x141 [95.504466] </TASK> [95.505197] Modules linked in: nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat nf_nat nf_conntrack nf_defrag_ip ---truncated---
CVE-2024-40937	In the Linux kernel, the following vulnerability has been resolved: gve: Clear napi->skb before dev_kfree_skb_any() gve_rx_free_skb incorrectly leaves napi->skb referencing an skb after it is freed with dev_kfree_skb_any(). This can result in a subsequent call to napi_get_frags returning a dangling pointer. Fix this by clearing napi->skb before the skb is freed.
CVE-2024-40940	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix tainted pointer delete is case of flow rules creation fail In case of flow rule creation fail in mlx5_lag_create_port_sel_table(), instead of previously created rules, the tainted pointer is deleted several times. Fix this bug by using correct flow rules pointers. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-40945	In the Linux kernel, the following vulnerability has been resolved: iommu: Return right value in iommu_sva_bind_device() iommu_sva_bind_device() should return either a sva bond handle or an ERR_PTR value in error cases. Existing drivers (idxd and uacce) only check the return value with IS_ERR(). This could potentially lead to a kernel NULL pointer dereference issue if the function returns NULL instead of an error pointer. In reality, this doesn't cause any problems because iommu_sva_bind_device() only returns NULL when the kernel is not configured with CONFIG_IOMMU_SVA. In this case, iommu_dev_enable_feature(dev, IOMMU_DEV_FEAT_SVA) will return an error, and the device drivers won't call iommu_sva_bind_device() at all.

CVE ID	Description
CVE-2024-40954	In the Linux kernel, the following vulnerability has been resolved: net: do not leave a dangling sk pointer, when socket creation fails It is possible to trigger a use-after-free by: * attaching an fentry probe to __sock_release() and the probe calling the bpf_get_socket_cookie() helper * running traceroute -I 1.1.1.1 on a freshly booted VM A KASAN enabled kernel will log something like below (decoded and stripped): ===== BUG: KASAN: slab-use-after-free in __sock_gen_cookie (/arch/x86/include/asm/atomic64_64.h:15 ./include/linux/atomic/atomic-arch-fallback.h:2583 ./include/linux/atomic/atomic-instrumented.h:1611 net/core/sock_diag.c:29) Read of size 8 at addr ffff888007110dd8 by task traceroute/299 CPU: 2 PID: 299 Comm: traceroute Tainted: G E 6.10.0-rc2+ #2 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.2-debian-1.16.2-1 04/01/2014 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:117 (discriminator 1)) print_report (mm/kasan/report.c:378 mm/kasan/report.c:488) ? __sock_gen_cookie (/arch/x86/include/asm/atomic64_64.h:15 ./include/linux/atomic/atomic-arch-fallback.h:2583 ./include/linux/atomic/atomic-instrumented.h:1611 net/core/sock_diag.c:29) kasan_report (mm/kasan/report.c:603) ? __sock_gen_cookie (/arch/x86/include/asm/atomic64_64.h:15 ./include/linux/atomic/atomic-arch-fallback.h:2583 ./include/linux/atomic/atomic-instrumented.h:1611 net/core/sock_diag.c:29) kasan_check_range (mm/kasan/generic.c:183 mm/kasan/generic.c:189) __sock_gen_cookie (/arch/x86/include/asm/atomic64_64.h:15 ./include/linux/atomic/atomic-arch-fallback.h:2583 ./include/linux/atomic/atomic-instrumented.h:1611 net/core/sock_diag.c:29) bpf_get_socket_ptr_cookie (/arch/x86/include/asm/preempt.h:94 ./include/linux/sock_diag.h:42 net/core/filter.c:5094 net/core/filter.c:5092) bpf_prog_875642cf11f1d139 __sock_release+0x6e/0x8e bpf_trampoline_6442506592+0x47/0xaf __sock_release (net/socket.c:652) __sock_create (net/socket.c:1601) ... Allocated by task 299 on cpu 2 at 78.328492s: kasan_save_stack (mm/kasan/common.c:48) kasan_save_track (mm/kasan/common.c:68) __kasan_slab_alloc (mm/kasan/common.c:312 mm/kasan/common.c:338) kmem_cache_alloc_noprof (mm/slub.c:3941 mm/slub.c:4000 mm/slub.c:4007) sk_prot_alloc (net/core/sock.c:2075) sk_alloc (net/core/sock.c:2134) inet_create (net/ipv4/af_inet.c:327 net/ipv4/af_inet.c:252) __sock_create (net/socket.c:1572) __sys_socket (net/socket.c:1660 net/socket.c:1644 net/socket.c:1706) __x64_sys_socket (net/socket.c:1718) do_syscall_64 (arch/x86/entry/common.c:52 arch/x86/entry/common.c:83) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) Freed by task 299 on cpu 2 at 78.328502s: kasan_save_stack (mm/kasan/common.c:48) kasan_save_track (mm/kasan/common.c:68) kasan_save_free_info (mm/kasan/generic.c:582) poison_slab_object (mm/kasan/common.c:242) __kasan_slab_free (mm/kasan/common.c:256) kmem_cache_free (mm/slub.c:4437 mm/slub.c:4511) __sk_destruct (net/core/sock.c:2117 net/core/sock.c:2208) inet_create (net/ipv4/af_inet.c:397 net/ipv4/af_inet.c:252) __sock_create (net/socket.c:1572) __sys_socket (net/socket.c:1660 net/socket.c:1644 net/socket.c:1706) __x64_sys_socket (net/socket.c:1718) do_syscall_64 (arch/x86/entry/common.c:52 arch/x86/entry/common.c:83) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) Fix this by clearing the struct socket reference in sk_common_release() to cover all protocol families create functions, which may already attached the reference to the sk object with sock_init_data().
CVE-2024-40956	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Fix possible Use-After-Free in irq_process_work_list Use list_for_each_entry_safe() to allow iterating through the list and deleting the entry in the iteration process. The descriptor is freed via idxd_desc_complete() and there's a slight chance may cause issue for the list iterator when the descriptor is reused by another thread without it being deleted from the list.
CVE-2024-40964	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda: cs35l41: Possible null pointer dereference in cs35l41_hda_unbind() The cs35l41_hda_unbind() function clears the hda_component entry matching it's index and then dereferences the codec pointer held in the first element of the hda_component array, this is an issue when the device index was 0. Instead use the codec pointer stashed in the cs35l41_hda structure as it will still be valid.
CVE-2024-40966	In the Linux kernel, the following vulnerability has been resolved: tty: add the option to have a tty reject a new ldisc ... and use it to limit the virtual terminals to just N_TTY. They are kind of special, and in particular, the "con_write()" routine violates the "writes cannot sleep" rule that some ldiscs rely on. This avoids the BUG: sleeping function called from invalid context at kernel/printk/printk.c:2659 when N_GSM has been attached to a virtual console, and gsmld_write() calls con_write() while holding a spinlock, and con_write() then tries to get the console lock.

CVE ID	Description
CVE-2024-40980	In the Linux kernel, the following vulnerability has been resolved: drop_monitor: replace spin_lock by raw_spin_lock trace_drop_common() is called with preemption disabled, and it acquires a spin_lock. This is problematic for RT kernels because spin_locks are sleeping locks in this configuration, which causes the following splat: BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 449, name: rcuc/47 preempt_count: 1, expected: 0 RCU nest depth: 2, expected: 2 5 locks held by rcuc/47/449: #0: ff1100086ec30a60 ((softirq_ctrl.lock)) {+.+.} -{2:2}, at: __local_bh_disable_ip+0x105/0x210 #1: ffffffff394a280 (rcu_read_lock){....} -{1:2}, at: rt_spin_lock+0xbf/0x130 #2: ffffffff394a280 (rcu_read_lock){....} -{1:2}, at: __local_bh_disable_ip+0x11c/0x210 #3: ffffffff394a160 (rcu_callback){....} -{0:0}, at: rcu_do_batch+0x360/0xc70 #4: ff1100086ee07520 (&data->lock) {+.+.} -{2:2}, at: trace_drop_common.constprop.0+0xb5/0x290 irq event stamp: 139909 hardirqs last enabled at (139908): [<ffffffffffb1df2b33>] _raw_spin_unlock_irqrestore+0x63/0x80 hardirqs last disabled at (139909): [<ffffffffffb19bd03d>] trace_drop_common.constprop.0+0x26d/0x290 softirqs last enabled at (139892): [<ffffffffffb07a1083>] __local_bh_enable_ip+0x103/0x170 softirqs last disabled at (139898): [<ffffffffffb0909b33>] rcu_cpu_kthread+0x93/0x1f0 Preemption disabled at: [<ffffffffffb1de786b>] rt_mutex_slowunlock+0xab/0x2e0 CPU: 47 PID: 449 Comm: rcuc/47 Not tainted 6.9.0-rc2-rt1+ #7 Hardware name: Dell Inc. PowerEdge R650/0Y2G81, BIOS 1.6.5 04/15/2022 Call Trace: <TASK> dump_stack_lvl+0x8c/0xd0 dump_stack+0x14/0x20 __might_resched+0x21e/0x2f0 rt_spin_lock+0x5e/0x130 ? trace_drop_common.constprop.0+0xb5/0x290 ? skb_queue_purge_reason.part.0+0x1bf/0x230 trace_drop_common.constprop.0+0xb5/0x290 ? preempt_count_sub+0x1c/0xd0 ? _raw_spin_unlock_irqrestore+0x4a/0x80 ? __pfx_trace_drop_common.constprop.0+0x10/0x10 ? rt_mutex_slowunlock+0x26a/0x2e0 ? skb_queue_purge_reason.part.0+0x1bf/0x230 ? __pfx_rt_mutex_slowunlock+0x10/0x10 ? skb_queue_purge_reason.part.0+0x1bf/0x230 trace_kfree_skb_hit+0x15/0x20 trace_kfree_skb+0xe9/0x150 kfree_skb_reason+0x7b/0x110 skb_queue_purge_reason.part.0+0x1bf/0x230 ? __pfx_skb_queue_purge_reason.part.0+0x10/0x10 ? mark_lock.part.0+0x8a/0x520 ... trace_drop_common() also disables interrupts, but this is a minor issue because we could easily replace it with a local_lock. Replace the spin_lock with raw_spin_lock to avoid sleeping in atomic context.
CVE-2024-40990	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Add check for srq max_sge attribute max_sge attribute is passed by the user, and is inserted and used unchecked, so verify that the value doesn't exceed maximum allowed value before using it.
CVE-2024-40999	In the Linux kernel, the following vulnerability has been resolved: net: ena: Add validation for completion descriptors consistency Validate that `first` flag is set only for the first descriptor in multi-buffer packets. In case of an invalid descriptor, a reset will occur. A new reset reason for RX data corruption has been added.

CVE ID	Description
CVE-2024-41000	In the Linux kernel, the following vulnerability has been resolved: block/ioctl: prefer different overflow check Running syzkaller with the newly reintroduced signed integer overflow sanitizer shows this report: [62.982337] -----[cut here]----- [62.985692] cgroup: Invalid name [62.986211] UBSAN: signed-integer-overflow in ../block/ioctl.c:36:46 [62.989370] 9pnet_fd: p9_fd_create_tcp (7343): problem connecting socket to 127.0.0.1 [62.992992] 9223372036854775807 + 4095 cannot be represented in type 'long long' [62.997827] 9pnet_fd: p9_fd_create_tcp (7345): problem connecting socket to 127.0.0.1 [62.999369] random: crng reseeded on system resumption [63.000634] GUP no longer grows the stack in syz-executor.2 (7353); 20002000-20003000 (20001000) [63.000668] CPU: 0 PID: 7353 Comm: syz-executor.2 Not tainted 6.8.0-rc2-00035-gb3ef86b5a957 #1 [63.000677] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [63.000682] Call Trace: [63.000686] <TASK> [63.000731] dump_stack_lvl+0x93/0xd0 [63.000919] __get_user_pages+0x903/0xd30 [63.001030] __gup_longterm_locked+0x153e/0x1ba0 [63.001041] ? _raw_read_unlock_irqrestore+0x17/0x50 [63.001072] ? try_get_folio+0x29c/0x2d0 [63.001083] internal_get_user_pages_fast+0x1119/0x1530 [63.001109] iov_iter_extract_pages+0x23b/0x580 [63.001206] bio_iov_iter_get_pages+0x4de/0x1220 [63.001235] iomap_dio_bio_iter+0x9b6/0x1410 [63.001297] __iomap_dio_rw+0xab4/0x1810 [63.001316] iomap_dio_rw+0x45/0xa0 [63.001328] ext4_file_write_iter+0xdde/0x1390 [63.001372] vfs_write+0x599/0xbd0 [63.001394] ksys_write+0xc8/0x190 [63.001403] do_syscall_64+0xd4/0x1b0 [63.001421] ? arch_exit_to_user_mode_prepare+0x3a/0x60 [63.001479] entry_SYSCALL_64_after_hwframe+0x6f/0x77 [63.001535] RIP: 0033:0xf7fd3ebf539 [63.001551] Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 f1 14 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff ff f7 d8 64 89 01 48 [63.001562] RSP: 002b:00007f7fd32570c8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 [63.001584] RAX: ffffffffefda RBX: 00007f7fd3ff3f80 RCX: 00007f7fd3ebf539 [63.001590] RDX: 4db6d1e4f7e43360 RSI: 0000000020000000 RDI: 0000000000000004 [63.001595] RBP: 00007f7fd3f1e496 R08: 0000000000000000 R09: 0000000000000000 [63.001599] R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 [63.001604] R13: 0000000000000006 R14: 00007f7fd3ff3f80 R15: 00007ffd415ad2b8 ... [63.018142] ---[end trace]--- Historically, the signed integer overflow sanitizer did not work in the kernel due to its interaction with <code>-fwrapv</code> but this has since been changed [1] in the newest version of Clang; It was re-enabled in the kernel with Commit 557f8c582a9ba8ab ("ubsan: Reintroduce signed overflow sanitizer"). Let's rework this overflow checking logic to not actually perform an overflow during the check itself, thus avoiding the UBSAN splat. [1]: https://github.com/llvm/llvm-project/pull/82432
CVE-2024-41008	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: change vm->task_info handling This patch changes the handling and lifecycle of vm->task_info object. The major changes are: - vm->task_info is a dynamically allocated ptr now, and its usage is reference counted. - introducing two new helper funcs for task_info lifecycle management - amdgpu_vm_get_task_info: reference counts up task_info before returning this info - amdgpu_vm_put_task_info: reference counts down task_info - last put to task_info() frees task_info from the vm. This patch also does logistical changes required for existing usage of vm->task_info. V2: Do not block all the prints when task_info not found (Felix) V3: Fixed review comments from Felix - Fix wrong indentation - No debug message for -ENOMEM - Add NULL check for task_info - Do not duplicate the debug messages (ti vs no ti) - Get first reference of task_info in vm_init(), put last in vm_fini() V4: Fixed review comments from Felix - fix double reference increment in create_task_info - change amdgpu_vm_get_task_info_pasid - additional changes in amdgpu_gem.c while porting

CVE ID	Description
CVE-2024-41010	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix too early release of tcx_entry Pedro Pinto and later independently also Hyunwoo Kim and Wongi Lee reported an issue that the tcx_entry can be released too early leading to a use after free (UAF) when an active old-style ingress or clsact qdisc with a shared tc block is later replaced by another ingress or clsact instance. Essentially, the sequence to trigger the UAF (one example) can be as follows: 1. A network namespace is created 2. An ingress qdisc is created. This allocates a tcx_entry, and &tcx_entry->miniq is stored in the qdisc's miniqp->p_minq. At the same time, a tcf block with index 1 is created. 3. chain0 is attached to the tcf block. chain0 must be connected to the block linked to the ingress qdisc to later reach the function tcf_chain0_head_change_cb_del() which triggers the UAF. 4. Create and graft a clsact qdisc. This causes the ingress qdisc created in step 1 to be removed, thus freeing the previously linked tcx_entry: rtnetlink_rcv_msg() => tc_modify_qdisc() => qdisc_create() => clsact_init() [a] => qdisc_graft() => qdisc_destroy() => __qdisc_destroy() => ingress_destroy() [b] => tcx_entry_free() => kfree_rcu() // tcx_entry freed 5. Finally, the network namespace is closed. This registers the cleanup_net worker, and during the process of releasing the remaining clsact qdisc, it accesses the tcx_entry that was already freed in step 4, causing the UAF to occur: cleanup_net() => ops_exit_list() => default_device_exit_batch() => unregister_netdevice_many() => unregister_netdevice_many_notify() => dev_shutdown() => qdisc_put() => clsact_destroy() [c] => tcf_block_put_ext() => tcf_chain0_head_change_cb_del() => tcf_chain_head_change_item() => clsact_chain_head_change() => mini_qdisc_pair_swap() // UAF There are also other variants, the gist is to add an ingress (or clsact) qdisc with a specific shared block, then to replace that qdisc, waiting for the tcx_entry kfree_rcu() to be executed and subsequently accessing the current active qdisc's miniq one way or another. The correct fix is to turn the miniq_active boolean into a counter. What can be observed, at step 2 above, the counter transitions from 0->1, at step [a] from 1->2 (in order for the miniq object to remain active during the replacement), then in [b] from 2->1 and finally [c] 1->0 with the eventual release. The reference counter in general ranges from [0,2] and it does not need to be atomic since all access to the counter is protected by the rtnl mutex. With this in place, there is no longer a UAF happening and the tcx_entry is freed at the correct time.
CVE-2024-41013	In the Linux kernel, the following vulnerability has been resolved: xfs: don't walk off the end of a directory data block This adds sanity checks for xfs_dir2_data_unused and xfs_dir2_data_entry to make sure don't stray beyond valid memory region. Before patching, the loop simply checks that the start offset of the dup and dep is within the range. So in a crafted image, if last entry is xfs_dir2_data_unused, we can change dup->length to dup->length-1 and leave 1 byte of space. In the next traversal, this space will be considered as dup or dep. We may encounter an out of bound read when accessing the fixed members. In the patch, we make sure that the remaining bytes large enough to hold an unused entry before accessing xfs_dir2_data_unused and xfs_dir2_data_unused is XFS_DIR2_DATA_ALIGN byte aligned. We also make sure that the remaining bytes large enough to hold a dirent with a single-byte name before accessing xfs_dir2_data_entry.
CVE-2024-41014	In the Linux kernel, the following vulnerability has been resolved: xfs: add bounds checking to xlog_recover_process_data There is a lack of verification of the space occupied by fixed members of xlog_op_header in the xlog_recover_process_data. We can create a crafted image to trigger an out of bounds read by following these steps: 1) Mount an image of xfs, and do some file operations to leave records 2) Before unmounting, copy the image for subsequent steps to simulate abnormal exit. Because umount will ensure that tail_blk and head_blk are the same, which will result in the inability to enter xlog_recover_process_data 3) Write a tool to parse and modify the copied image in step 2 4) Make the end of the xlog_op_header entries only 1 byte away from xlog_rec_header->h_size 5) xlog_rec_header->h_num_logops++ 6) Modify xlog_rec_header->h_crc Fix: Add a check to make sure there is sufficient space to access fixed members of xlog_op_header.
CVE-2024-41020	In the Linux kernel, the following vulnerability has been resolved: filelock: Fix fcntl/close race recovery compat path When I wrote commit 3cad1bc01041 ("filelock: Remove locks reliably when fcntl/close race is detected"), I missed that there are two copies of the code I was patching: The normal version, and the version for 64-bit offsets on 32-bit kernels. Thanks to Greg KH for stumbling over this while doing the stable backport... Apply exactly the same fix to the compat path for 32-bit kernels.
CVE-2024-41037	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda: fix null deref on system suspend entry When system enters suspend with an active stream, SOF core calls hw_params_upon_resume(). On Intel platforms with HDA DMA used to manage the link DMA, this leads to call chain of hda_dsp_set_hw_params_upon_resume() -> hda_dsp_dais_suspend() -> hda_dai_suspend() -> hda_ipc4_post_trigger() A bug is hit in hda_dai_suspend() as hda_link_dma_cleanup() is run first, which clears hext_stream->link_substream, and then hda_ipc4_post_trigger() is called with a NULL snd_pcm_substream pointer.

CVE ID	Description
CVE-2024-41047	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix XDP program unloading while removing the driver The commit 6533e558c650 ("i40e: Fix reset path while removing the driver") introduced a new PF state "__I40E_IN_REMOVE" to block modifying the XDP program while the driver is being removed. Unfortunately, such a change is useful only if the ".ndo_bpf()" callback was called out of the rmmmod context because unloading the existing XDP program is also a part of driver removing procedure. In other words, from the rmmmod context the driver is expected to unload the XDP program without reporting any errors. Otherwise, the kernel warning with callstack is printed out to dmesg. Example failing scenario: 1. Load the i40e driver. 2. Load the XDP program. 3. Unload the i40e driver (using "rmmmod" command). The example kernel warning log: [+0.004646] WARNING: CPU: 94 PID: 10395 at net/core/dev.c:9290 unregister_netdevice_many_notify+0x7a9/0x870 [...] [+0.010959] RIP: 0010:unregister_netdevice_many_notify+0x7a9/0x870 [...] [+0.002726] Call Trace: [+0.002457] <TASK> [+0.002119] ? __warn+0x80/0x120 [+0.003245] ? unregister_netdevice_many_notify+0x7a9/0x870 [+0.005586] ? report_bug+0x164/0x190 [+0.003678] ? handle_bug+0x3c/0x80 [+0.003503] ? exc_invalid_op+0x17/0x70 [+0.003846] ? asm_exc_invalid_op+0x1a/0x20 [+0.004200] ? unregister_netdevice_many_notify+0x7a9/0x870 [+0.005579] ? unregister_netdevice_many_notify+0x3cc/0x870 [+0.005586] unregister_netdevice_queue+0xf7/0x140 [+0.004806] unregister_netdev+0x1c/0x30 [+0.003933] i40e_vsi_release+0x87/0x2f0 [i40e] [+0.004604] i40e_remove+0x1a1/0x420 [i40e] [+0.004220] pci_device_remove+0x3f/0xb0 [+0.003943] device_release_driver_internal+0x19f/0x200 [+0.005243] driver_detach+0x48/0x90 [+0.003586] bus_remove_driver+0x6d/0xf0 [+0.003939] pci_unregister_driver+0x2e/0xb0 [+0.004278] i40e_exit_module+0x10/0x5f0 [i40e] [+0.004570] __do_sys_delete_module.isra.0+0x197/0x310 [+0.005153] do_syscall_64+0x85/0x170 [+0.003684] ? syscall_exit_to_user_mode+0x69/0x220 [+0.004886] ? do_syscall_64+0x95/0x170 [+0.003851] ? exc_page_fault+0x7e/0x180 [+0.003932] entry_SYSCALL_64_after_hwframe+0x71/0x79 [+0.005064] RIP: 0033:0x7f59dc9347cb [+0.003648] Code: 73 01 c3 48 8b 0d 65 16 0c 00 f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa b8 b0 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 35 16 0c 00 f7 d8 64 89 01 48 [+0.018753] RSP: 002b:00007ffffac99048 EFLAGS: 00000206 ORIG_RAX: 00000000000000b0 [+0.007577] RAX: ffffffffdfda RBX: 0000559b9bb2f6e0 RCX: 00007f59dc9347cb [+0.007140] RDX: 0000000000000000 RSI: 0000000000000800 RDI: 0000559b9bb2f748 [+0.007146] RBP: 00007ffffac99070 R08: 1999999999999999 R09: 0000000000000000 [+0.007133] R10: 00007f59dc9a5a0 R11: 0000000000000206 R12: 0000000000000000 [+0.007141] R13: 00007ffffac992d8 R14: 0000559b9bb2f6e0 R15: 0000000000000000 [+0.007151] </TASK> [+0.002204] ---[end trace 0000000000000000]--- Fix this by checking if the XDP program is being loaded or unloaded. Then, block only loading a new program while "__I40E_IN_REMOVE" is set. Also, move testing "__I40E_IN_REMOVE" flag to the beginning of XDP_SETUP callback to avoid unnecessary operations and checks.
CVE-2024-41062	In the Linux kernel, the following vulnerability has been resolved: bluetooth/l2cap: sync sock recv cb and release The problem occurs between the system call to close the sock and hci_rx_work, where the former releases the sock and the latter accesses it without lock protection. CPU0 CPU1 ---- sock_close hci_rx_work l2cap_sock_release hci_acldata_packet l2cap_sock_kill l2cap_recv_frame sk_free l2cap_conless_channel l2cap_sock_recv_cb If hci_rx_work processes the data that needs to be received before the sock is closed, then everything is normal; Otherwise, the work thread may access the released sock when receiving data. Add a chan mutex in the rx callback of the sock to achieve synchronization between the sock release and recv cb. Sock is dead, so set chan data to NULL, avoid others use invalid sock pointer.
CVE-2024-41063	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: cancel all works upon hci_unregister_dev() syzbot is reporting that calling hci_release_dev() from hci_error_reset() due to hci_dev_put() from hci_error_reset() can cause deadlock at destroy_workqueue(), for hci_error_reset() is called from hdev->req_workqueue which destroy_workqueue() needs to flush. We need to make sure that hdev->{rx_work,cmd_work,tx_work} which are queued into hdev->workqueue and hdev->{power_on,error_reset} which are queued into hdev->req_workqueue are no longer running by the moment destroy_workqueue(hdev->workqueue); destroy_workqueue(hdev->req_workqueue); are called from hci_release_dev(). Call cancel_work_sync() on these work items from hci_unregister_dev() as soon as hdev->list is removed from hci_dev_list.
CVE-2024-41069	In the Linux kernel, the following vulnerability has been resolved: ASoC: topology: Fix references to freed memory Most users after parsing a topology file, release memory used by it, so having pointer references directly into topology file contents is wrong. Use devm_kmemdup(), to allocate memory as needed.
CVE-2024-41072	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: wext: add extra SIOCSIWSCAN data check In 'cfg80211_wext_siwscan()', add extra check whether number of channels passed via 'ioctl(sock, SIOCSIWSCAN, ...)' doesn't exceed IW_MAX_FREQUENCIES and reject invalid request with -EINVAL otherwise.

CVE ID	Description
CVE-2024-41073	In the Linux kernel, the following vulnerability has been resolved: nvme: avoid double free special payload If a discard request needs to be retried, and that retry may fail before a new special payload is added, a double free will result. Clear the RQF_SPECIAL_LOAD when the request is cleaned.
CVE-2024-41077	In the Linux kernel, the following vulnerability has been resolved: null_blk: fix validation of block size Block size should be between 512 and PAGE_SIZE and be a power of 2. The current check does not validate this, so update the check. Without this patch, null_blk would Oops due to a null pointer deref when loaded with bs=1536 [1]. [axboe: remove unnecessary braces and != 0 check]
CVE-2024-41079	In the Linux kernel, the following vulnerability has been resolved: nvmet: always initialize cq.e.result The spec doesn't mandate that the first two double words (aka results) for the command queue entry need to be set to 0 when they are not used (not specified). Though, the target implementation returns 0 for TCP and FC but not for RDMA. Let's make RDMA behave the same and thus explicitly initializing the result field. This prevents leaking any data from the stack.
CVE-2024-41082	In the Linux kernel, the following vulnerability has been resolved: nvme-fabrics: use reserved tag for reg read/write command In some scenarios, if too many commands are issued by nvme command in the same time by user tasks, this may exhaust all tags of admin_q. If a reset (nvme reset or IO timeout) occurs before these commands finish, reconnect routine may fail to update nvme regs due to insufficient tags, which will cause kernel hang forever. In order to workaround this issue, maybe we can let reg_read32()/reg_read64()/reg_write32() use reserved tags. This maybe safe for nvmmf: 1. For the disable ctrl path, we will not issue connect command 2. For the enable ctrl / fw activate path, since connect and reg_xx() are called serially. So the reserved tags may still be enough while reg_xx() use reserved tags.
CVE-2024-41087	In the Linux kernel, the following vulnerability has been resolved: ata: libata-core: Fix double free on error If e.g. the ata_port_alloc() call in ata_host_alloc() fails, we will jump to the err_out label, which will call devres_release_group(). devres_release_group() will trigger a call to ata_host_release(). ata_host_release() calls kfree(host), so executing the kfree(host) in ata_host_alloc() will lead to a double free: kernel BUG at mm/slub.c:553! Oops: invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 11 PID: 599 Comm: (udev-worker) Not tainted 6.10.0-rc5 #47 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-2.fc40 04/01/2014 RIP: 0010:kfree+0x2cf/0x2f0 Code: 5d 41 5e 41 5f 5d e9 80 d6 ff ff 4d 89 f1 41 b8 01 00 00 00 48 89 d9 48 89 da RSP: 0018:ffff90000f377f0 EFLAGS: 00010246 RAX: ffff888112b1f2c0 RBX: ffff888112b1f2c0 RCX: ffff888112b1f320 RDX: 000000000000400b RSI: ffffffff02c9de5 RDI: ffff888112b1f2c0 RBP: ffff90000f37830 R08: 0000000000000000 R09: 0000000000000000 R10: ffff90000f37610 R11: 617461203a736b6e R12: ffffea00044ac780 R13: ffff888100046400 R14: ffffffff02c9de5 R15: 0000000000000006 FS: 00007f2f1cabe980(0000) GS:ffff88813b380000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f2f1c3acf75 CR3: 0000000111724000 CR4: 0000000000750ef0 PKRU: 55555554 Call Trace: <TASK> ? __die_body.cold+0x19/0x27 ? die+0x2e/0x50 ? do_trap+0xca/0x110 ? do_error_trap+0x6a/0x90 ? kfree+0x2cf/0x2f0 ? exc_invalid_op+0x50/0x70 ? kfree+0x2cf/0x2f0 ? asm_exc_invalid_op+0x1a/0x20 ? ata_host_alloc+0xf5/0x120 [libata] ? ata_host_alloc+0xf5/0x120 [libata] ? kfree+0x2cf/0x2f0 ata_host_alloc+0xf5/0x120 [libata] ata_host_alloc_pinfo+0x14/0xa0 [libata] ahci_init_one+0x6c9/0xd20 [ahci] Ensure that we will not call kfree(host) twice, by performing the kfree() only if the devres_open_group() call failed.
CVE-2024-41089	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/dispnv04: fix null pointer dereference in nv17_tv_get_hd_modes In nv17_tv_get_hd_modes(), the return value of drm_mode_duplicate() is assigned to mode, which will lead to a possible NULL pointer dereference on failure of drm_mode_duplicate(). The same applies to drm_cvt_mode(). Add a check to avoid null pointer dereference.
CVE-2024-41095	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/dispnv04: fix null pointer dereference in nv17_tv_get_ld_modes In nv17_tv_get_ld_modes(), the return value of drm_mode_duplicate() is assigned to mode, which will lead to a possible NULL pointer dereference on failure of drm_mode_duplicate(). Add a check to avoid npd.

CVE ID	Description
CVE-2024-41098	In the Linux kernel, the following vulnerability has been resolved: ata: libata-core: Fix null pointer dereference on error If the ata_port_alloc() call in ata_host_alloc() fails, ata_host_release() will get called. However, the code in ata_host_release() tries to free ata_port struct members unconditionally, which can lead to the following: BUG: unable to handle page fault for address: 0000000000003990 PGD 0 P4D 0 Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 10 PID: 594 Comm: (udev-worker) Not tainted 6.10.0-rc5 #44 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-2.fc40 04/01/2014 RIP: 0010:ata_host_release.cold+0x2f/0x6e [libata] Code: e4 4d 63 f4 44 89 e2 48 c7 c6 90 ad 32 c0 48 c7 c7 d0 70 33 c0 49 83 c6 0e 41 RSP: 0018:ffffc90000ebb968 EFLAGS: 00010246 RAX: 0000000000000041 RBX: ffff88810fb52e78 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffff88813b3218c0 RDI: ffff88813b3218c0 RBP: ffff88810fb52e40 R08: 0000000000000000 R09: 6c65725f74736f68 R10: ffff88810fb52e78 R11: 73692033203a746e R12: 0000000000000004 R13: 0000000000000000 R14: 0000000000000011 R15: 0000000000000006 FS: 00007f6cc55b9980(0000) GS:ffff88813b300000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 0000000000000033 CR2: 0000000000003990 CR3: 00000001122a2000 CR4: 0000000000750ef0 PKRU: 55555554 Call Trace: <TASK> ? __die_body.cold+0x19/0x27 ? page_fault_oops+0x15a/0x2f0 ? exc_page_fault+0x7e/0x180 ? asm_exc_page_fault+0x26/0x30 ? ata_host_release.cold+0x2f/0x6e [libata] ? ata_host_release.cold+0x2f/0x6e [libata] release_nodes+0x35/0xb0 devres_release_group+0x113/0x140 ata_host_alloc+0xed/0x120 [libata] ata_host_alloc_pinfo+0x14/0xa0 [libata] ahci_init_one+0x6c9/0xd20 [ahci] Do not access ata_port struct members unconditionally.
CVE-2024-41110	Moby is an open-source project created by Docker for software containerization. A security vulnerability has been detected in certain versions of Docker Engine, which could allow an attacker to bypass authorization plugins (AuthZ) under specific circumstances. The base likelihood of this being exploited is low. Using a specially-crafted API request, an Engine API client could make the daemon forward the request or response to an authorization plugin without the body. In certain circumstances, the authorization plugin may allow a request which it would have otherwise denied if the body had been forwarded to it. A security issue was discovered In 2018, where an attacker could bypass AuthZ plugins using a specially crafted API request. This could lead to unauthorized actions, including privilege escalation. Although this issue was fixed in Docker Engine v18.09.1 in January 2019, the fix was not carried forward to later major versions, resulting in a regression. Anyone who depends on authorization plugins that introspect the request and/or response body to make access control decisions is potentially impacted. Docker EE v19.03.x and all versions of Mirantis Container Runtime are not vulnerable. docker-ce v27.1.1 contains patches to fix the vulnerability. Patches have also been merged into the master, 19.03, 20.0, 23.0, 24.0, 25.0, 26.0, and 26.1 release branches. If one is unable to upgrade immediately, avoid using AuthZ plugins and/or restrict access to the Docker API to trusted parties, following the principle of least privilege.
CVE-2024-41123	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.2 has some DoS vulnerabilities when it parses an XML that has many specific characters such as whitespace character, '>]' and '>'. The REXML gem 3.3.3 or later include the patches to fix these vulnerabilities.
CVE-2024-41932	In the Linux kernel, the following vulnerability has been resolved: sched: fix warning in sched_setaffinity Commit 8f9ea86fd99b added some logic to sched_setaffinity that included a WARN when a per-task affinity assignment races with a cpuset update. Specifically, we can have a race where a cpuset update results in the task affinity no longer being a subset of the cpuset. That's fine; we have a fallback to instead use the cpuset mask. However, we have a WARN set up that will trigger if the cpuset mask has no overlap at all with the requested task affinity. This shouldn't be a warning condition; its trivial to create this condition. Reproduced the warning by the following setup: - \$PID inside a cpuset cgroup - another thread repeatedly switching the cpuset cpus from 1-2 to just 1 - another thread repeatedly setting the \$PID affinity (via taskset) to 2
CVE-2024-41935	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to shrink read extent node in batches We use rwlock to protect core structure data of extent tree during its shrink, however, if there is a huge number of extent nodes in extent tree, during shrink of extent tree, it may hold rwlock for a very long time, which may trigger kernel hang issue. This patch fixes to shrink read extent node in batches, so that, critical region of the rwlock can be shrunk to avoid its extreme long time hold.
CVE-2024-41946	REXML is an XML toolkit for Ruby. The REXML gem 3.3.2 has a DoS vulnerability when it parses an XML that has many entity expansions with SAX2 or pull parser API. The REXML gem 3.3.3 or later include the patch to fix the vulnerability.
CVE-2024-41965	Vim is an open source command line text editor. double-free in dialog_changed() in Vim < v9.1.0648. When abandoning a buffer, Vim may ask the user what to do with the modified buffer. If the user wants the changed buffer to be saved, Vim may create a new Untitled file, if the buffer did not have a name yet. However, when setting the buffer name to Unnamed, Vim will falsely free a pointer twice, leading to a double-free and possibly later to a heap-use-after-free, which can lead to a crash. The issue has been fixed as of Vim patch v9.1.0648.

CVE ID	Description
CVE-2024-41996	Validating the order of the public keys in the Diffie-Hellman Key Agreement Protocol, when an approved safe prime is used, allows remote attackers (from the client side) to trigger unnecessarily expensive server-side DHE modular-exponentiation calculations. The client may cause asymmetric resource consumption. The basic attack scenario is that the client must claim that it can only communicate with DHE, and the server must be configured to allow DHE and validate the order of the public key.
CVE-2024-42063	In the Linux kernel, the following vulnerability has been resolved: bpf: Mark bpf prog stack with kmsan_unpoison_memory in interpreter mode syzbot reported uninit memory usages during map_{lookup,delete}_elem. ===== BUG: KMSAN: uninit-value in __dev_map_lookup_elem kernel/bpf/devmap.c:441 [inline] BUG: KMSAN: uninit-value in dev_map_lookup_elem +0xf3/0x170 kernel/bpf/devmap.c:796 __dev_map_lookup_elem kernel/bpf/devmap.c:441 [inline] dev_map_lookup_elem+0xf3/0x170 kernel/bpf/devmap.c:796 ____bpf_map_lookup_elem kernel/bpf/helpers.c:42 [inline] bpf_map_lookup_elem+0x5c/0x80 kernel/bpf/helpers.c:38 ____bpf_prog_run +0x13fe/0xe0f0 kernel/bpf/core.c:1997 ____bpf_prog_run256+0xb5/0xe0 kernel/bpf/core.c:2237 ===== The reproducer should be in the interpreter mode. The C reproducer is trying to run the following bpf prog: 0: (18) r0 = 0x0 2: (18) r1 = map[id:49] 4: (b7) r8 = 16777216 5: (7b) *(u64 *) (r10 - 8) = r8 6: (bf) r2 = r10 7: (07) r2 += -229 ^^^^^^^^^ 8: (b7) r3 = 8 9: (b7) r4 = 0 10: (85) call dev_map_lookup_elem#1543472 11: (95) exit It is due to the "void *key" (r2) passed to the helper. bpf allows uninit stack memory access for bpf prog with the right privileges. This patch uses kmsan_unpoison_memory() to mark the stack as initialized. This should address different syzbot reports on the uninit "void *key" argument during map_{lookup,delete}_elem.
CVE-2024-42065	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Add a NULL check in xe_ttm_stolen_mgr_init Add an explicit check to ensure that the mgr is not NULL.
CVE-2024-42067	In the Linux kernel, the following vulnerability has been resolved: bpf: Take return from set_memory_rox() into account with bpf_jit_binary_lock_ro() set_memory_rox() can fail, leaving memory unprotected. Check return and bail out when bpf_jit_binary_lock_ro() returns an error.
CVE-2024-42068	In the Linux kernel, the following vulnerability has been resolved: bpf: Take return from set_memory_ro() into account with bpf_prog_lock_ro() set_memory_ro() can fail, leaving memory unprotected. Check its return and take it into account as an error.
CVE-2024-42069	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix possible double free in error handling path When auxiliary_device_add() returns error and then calls auxiliary_device_uninit(), callback function adev_release calls kfree(madev). We shouldn't call kfree(madev) again in the error handling path. Set 'madev' to NULL.
CVE-2024-42080	In the Linux kernel, the following vulnerability has been resolved: RDMA/restrack: Fix potential invalid address access struct rdma_restrack_entry's kern_name was set to KBUILD_MODNAME in ib_create_cq(), while if the module exited but forgot del this rdma_restrack_entry, it would cause a invalid address access in rdma_restrack_clean() when print the owner of this rdma_restrack_entry. These code is used to help find one forgotten PD release in one of the ULPs. But it is not needed anymore, so delete them.
CVE-2024-42082	In the Linux kernel, the following vulnerability has been resolved: xdp: Remove WARN() from __xdp_reg_mem_model() syzkaller reports a warning in __xdp_reg_mem_model(). The warning occurs only if __mem_id_init_hash_table() returns an error. It returns the error in two cases: 1. memory allocation fails; 2. rhashtable_init() fails when some fields of rhashtable_params struct are not initialized properly. The second case cannot happen since there is a static const rhashtable_params struct with valid fields. So, warning is only triggered when there is a problem with memory allocation. Thus, there is no sense in using WARN() to handle this error and it can be safely removed. WARNING: CPU: 0 PID: 5065 at net/core/xdp.c:299 __xdp_reg_mem_model+0x2d9/0x650 net/core/xdp.c:299 CPU: 0 PID: 5065 Comm: syz-executor883 Not tainted 6.8.0-syzkaller-05271-gf99c5f563c17 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 03/27/2024 RIP: 0010: __xdp_reg_mem_model+0x2d9/0x650 net/core/xdp.c:299 Call Trace: xdp_reg_mem_model+0x22/0x40 net/core/xdp.c:344 xdp_test_run_setup net/bpf/test_run.c:188 [inline] bpf_test_run_xdp_live+0x365/0x1e90 net/bpf/test_run.c:377 bpf_prog_test_run_xdp+0x813/0x11b0 net/bpf/test_run.c:1267 bpf_prog_test_run+0x33a/0x3b0 kernel/bpf/syscall.c:4240 __sys_bpf+0x48d/0x810 kernel/bpf/syscall.c:5649 __do_sys_bpf kernel/bpf/syscall.c:5738 [inline] __se_sys_bpf kernel/bpf/syscall.c:5736 [inline] __x64_sys_bpf+0x7c/0x90 kernel/bpf/syscall.c:5736 do_syscall_64+0xfb/0x240 entry_SYSCALL_64_after_hwframe+0x6d/0x75 Found by Linux Verification Center (linuxtesting.org) with syzkaller.
CVE-2024-42097	In the Linux kernel, the following vulnerability has been resolved: ALSA: emux: improve patch ioctl data validation In load_data(), make the validation of and skipping over the main info block match that in load_guspatch(). In load_guspatch(), add checking that the specified patch length matches the actually supplied data, like load_data() already did.

CVE ID	Description
CVE-2024-42098	In the Linux kernel, the following vulnerability has been resolved: crypto: ecdh - explicitly zeroize private_key private_key is overwritten with the key parameter passed in by the caller (if present), or alternatively a newly generated private key. However, it is possible that the caller provides a key (or the newly generated key) which is shorter than the previous key. In that scenario, some key material from the previous key would not be overwritten. The easiest solution is to explicitly zeroize the entire private_key array first. Note that this patch slightly changes the behavior of this function: previously, if the ecc_gen_privkey failed, the old private_key would remain. Now, the private_key is always zeroized. This behavior is consistent with the case where params.key is set and ecc_is_key_valid fails.
CVE-2024-42101	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: fix null pointer dereference in nouveau_connector_get_modes In nouveau_connector_get_modes(), the return value of drm_mode_duplicate() is assigned to mode, which will lead to a possible NULL pointer dereference on failure of drm_mode_duplicate(). Add a check to avoid npd.
CVE-2024-42106	In the Linux kernel, the following vulnerability has been resolved: inet_diag: Initialize pad field in struct inet_diag_req_v2 KMSAN reported uninit-value access in raw_lookup() [1]. Diag for raw sockets uses the pad field in struct inet_diag_req_v2 for the underlying protocol. This field corresponds to the sdiag_raw_protocol field in struct inet_diag_req_raw. inet_diag_get_exact_compat() converts inet_diag_req to inet_diag_req_v2, but leaves the pad field uninitialized. So the issue occurs when raw_lookup() accesses the sdiag_raw_protocol field. Fix this by initializing the pad field in inet_diag_get_exact_compat(). Also, do the same fix in inet_diag_dump_compat() to avoid the similar issue in the future. [1] BUG: KMSAN: uninit-value in raw_lookup net/ipv4/raw_diag.c:49 [inline] BUG: KMSAN: uninit-value in raw_sock_get+0x657/0x800 net/ipv4/raw_diag.c:71 raw_lookup net/ipv4/raw_diag.c:49 [inline] raw_sock_get+0x657/0x800 net/ipv4/raw_diag.c:71 raw_diag_dump_one+0xa1/0x660 net/ipv4/raw_diag.c:99 inet_diag_cmd_exact+0x7d9/0x980 inet_diag_get_exact_compat net/ipv4/inet_diag.c:1404 [inline] inet_diag_rcv_msg_compat +0x469/0x530 net/ipv4/inet_diag.c:1426 sock_diag_rcv_msg+0x23d/0x740 net/core/sock_diag.c:282 netlink_rcv_skb+0x537/0x670 net/netlink/af_netlink.c:2564 sock_diag_rcv+0x35/0x40 net/core/sock_diag.c:297 netlink_unicast_kernel net/netlink/af_netlink.c:1335 [inline] netlink_unicast +0xe74/0x1240 net/netlink/af_netlink.c:1361 netlink_sendmsg+0x10c6/0x1260 net/netlink/af_netlink.c:1905 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x332/0x3d0 net/socket.c:745 __sys_sendmsg+0x7f0/0xb70 net/socket.c:2585 __sys_sendmsg+0x271/0x3b0 net/socket.c:2639 __sys_sendmsg net/socket.c:2668 [inline] __do_sys_sendmsg net/socket.c:2677 [inline] __se_sys_sendmsg net/socket.c:2675 [inline] __x64_sys_sendmsg+0x27e/0x4a0 net/socket.c:2675 x64_sys_call+0x135e/0x3ce0 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xd9/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was stored to memory at: raw_sock_get +0x650/0x800 net/ipv4/raw_diag.c:71 raw_diag_dump_one+0xa1/0x660 net/ipv4/raw_diag.c:99 inet_diag_cmd_exact+0x7d9/0x980 inet_diag_get_exact_compat net/ipv4/inet_diag.c:1404 [inline] inet_diag_rcv_msg_compat+0x469/0x530 net/ipv4/inet_diag.c:1426 sock_diag_rcv_msg+0x23d/0x740 net/core/sock_diag.c:282 netlink_rcv_skb+0x537/0x670 net/netlink/af_netlink.c:2564 sock_diag_rcv +0x35/0x40 net/core/sock_diag.c:297 netlink_unicast_kernel net/netlink/af_netlink.c:1335 [inline] netlink_unicast+0xe74/0x1240 net/netlink/af_netlink.c:1361 netlink_sendmsg+0x10c6/0x1260 net/netlink/af_netlink.c:1905 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg +0x332/0x3d0 net/socket.c:745 __sys_sendmsg+0x7f0/0xb70 net/socket.c:2585 __sys_sendmsg +0x271/0x3b0 net/socket.c:2639 __sys_sendmsg net/socket.c:2668 [inline] __do_sys_sendmsg net/socket.c:2677 [inline] __se_sys_sendmsg net/socket.c:2675 [inline] __x64_sys_sendmsg +0x27e/0x4a0 net/socket.c:2675 x64_sys_call+0x135e/0x3ce0 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xd9/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Local variable req.i created at: inet_diag_get_exact_compat net/ipv4/inet_diag.c:1396 [inline] inet_diag_rcv_msg_compat +0x2a6/0x530 net/ipv4/inet_diag.c:1426 sock_diag_rcv_msg+0x23d/0x740 net/core/sock_diag.c:282 CPU: 1 PID: 8888 Comm: syz-executor.6 Not tainted 6.10.0-rc4-00217-g35bb670d65fc #32 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-2.fc40 04/01/2014
CVE-2024-42107	In the Linux kernel, the following vulnerability has been resolved: ice: Don't process extts if PTP is disabled The ice_ptp_extts_event() function can race with ice_ptp_release() and result in a NULL pointer dereference which leads to a kernel panic. Panic occurs because the ice_ptp_extts_event() function calls ptp_clock_event() with a NULL pointer. The ice driver has already released the PTP clock by the time the interrupt for the next external timestamp event occurs. To fix this, modify the ice_ptp_extts_event() function to check the PTP state and bail early if PTP is not ready.

CVE ID	Description
CVE-2024-42110	In the Linux kernel, the following vulnerability has been resolved: net: ntb_netdev: Move ntb_netdev_rx_handler() to call netif_rx() from __netif_rx() The following is emitted when using idxd (DSA) dmanegine as the data mover for ntb_transport that ntb_netdev uses. [74412.546922] BUG: using smp_processor_id() in preemptible [00000000] code: irq/52-idxd-por/14526 [74412.556784] caller is netif_rx_internal+0x42/0x130 [74412.562282] CPU: 6 PID: 14526 Comm: irq/52-idxd-por Not tainted 6.9.5 #5 [74412.569870] Hardware name: Intel Corporation ArcherCity/ArcherCity, BIOS EGSDCRB1.E9I.1752.P05.2402080856 02/08/2024 [74412.581699] Call Trace: [74412.584514] <TASK> [74412.586933] dump_stack_lvl+0x55/0x70 [74412.591129] check_preemption_disabled+0xc8/0xf0 [74412.596374] netif_rx_internal+0x42/0x130 [74412.600957] __netif_rx+0x20/0xd0 [74412.604743] ntb_netdev_rx_handler+0x66/0x150 [ntb_netdev] [74412.610985] ntb_complete_rxc+0xed/0x140 [ntb_transport] [74412.617010] ntb_rx_copy_callback+0x53/0x80 [ntb_transport] [74412.623332] idxd_dma_complete_tx+0xe3/0x160 [idxd] [74412.628963] idxd_wq_thread+0x1a6/0x2b0 [idxd] [74412.634046] irq_thread_fn+0x21/0x60 [74412.638134] ? irq_thread+0xa8/0x290 [74412.642218] irq_thread+0x1a0/0x290 [74412.646212] ? __pfx_irq_thread_fn+0x10/0x10 [74412.651071] ? __pfx_irq_thread_dtor+0x10/0x10 [74412.656117] ? __pfx_irq_thread+0x10/0x10 [74412.660686] kthread+0x100/0x130 [74412.664384] ? __pfx_kthread+0x10/0x10 [74412.668639] ret_from_fork+0x31/0x50 [74412.672716] ? __pfx_kthread+0x10/0x10 [74412.676978] ret_from_fork_asm+0x1a/0x30 [74412.681457] </TASK> The cause is due to the idxd driver interrupt completion handler uses threaded interrupt and the threaded handler is not hard or soft interrupt context. However __netif_rx() can only be called from interrupt context. Change the call to netif_rx() in order to allow completion via normal context for dmaengine drivers that utilize threaded irq handling. While the following commit changed from netif_rx() to __netif_rx(), baebdf48c360 ("net: dev: Makes sure netif_rx() can be invoked in any context."), the change should've been a noop instead. However, the code precedes this fix should've been using netif_rx_ni() or netif_rx_any_context().
CVE-2024-42122	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL pointer check for kzalloc [Why & How] Check return pointer of kzalloc before using it.
CVE-2024-42125	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fw: scan offload prohibit all 6 GHz channel if no 6 GHz sband We have some policy via BIOS to block uses of 6 GHz. In this case, 6 GHz sband will be NULL even if it is WiFi 7 chip. So, add NULL handling here to avoid crash.
CVE-2024-42126	In the Linux kernel, the following vulnerability has been resolved: powerpc: Avoid nmi_enter/nmi_exit in real mode interrupt. nmi_enter()/nmi_exit() touches per cpu variables which can lead to kernel crash when invoked during real mode interrupt handling (e.g. early HMI/MCE interrupt handler) if percpu allocation comes from vmalloc area. Early HMI/MCE handlers are called through DEFINE_INTERRUPT_HANDLER_NMI() wrapper which invokes nmi_enter/nmi_exit calls. We don't see any issue when percpu allocation is from the embedded first chunk. However with CONFIG_NEED_PER_CPU_PAGE_FIRST_CHUNK enabled there are chances where percpu allocation can come from the vmalloc area. With kernel command line "percpu_alloc=page" we can force percpu allocation to come from vmalloc area and can see kernel crash in machine_check_early: [1.215714] NIP [c000000000e49eb4] rcu_nmi_enter+0x24/0x110 [1.215717] LR [c0000000000461a0] machine_check_early+0xf0/0x2c0 [1.215719] --- interrupt: 200 [1.215720] [c0000000fffd73180] [0000000000000000] 0x0 (unreliable) [1.215722] [c0000000fffd731b0] [0000000000000000] 0x0 [1.215724] [c0000000fffd73210] [c000000000008364] machine_check_early_common+0x134/0x1f8 Fix this by avoiding use of nmi_enter()/nmi_exit() in real mode if percpu first chunk is not embedded.
CVE-2024-42129	In the Linux kernel, the following vulnerability has been resolved: leds: mlxreg: Use devm_mutex_init() for mutex initialization In this driver LEDs are registered using devm_led_classdev_register() so they are automatically unregistered after module's remove() is done. led_classdev_unregister() calls module's led_set_brightness() to turn off the LEDs and that callback uses mutex which was destroyed already in module's remove() so use devm API instead.
CVE-2024-42135	In the Linux kernel, the following vulnerability has been resolved: vhost_task: Handle SIGKILL by flushing work and exiting Instead of lingering until the device is closed, this has us handle SIGKILL by: 1. marking the worker as killed so we no longer try to use it with new virtqueues and new flush operations. 2. setting the virtqueue to worker mapping so no new works are queued. 3. running all the exiting works.

CVE ID	Description
CVE-2024-42136	In the Linux kernel, the following vulnerability has been resolved: cdrom: rearrange last_media_change check to avoid unintentional overflow When running syzkaller with the newly reintroduced signed integer wrap sanitizer we encounter this splat: [366.015950] UBSAN: signed-integer-overflow in ./drivers/cdrom/cdrom.c:2361:33 [366.021089] -9223372036854775808 - 346321 cannot be represented in type '__s64' (aka 'long long') [366.025894] program syz-executor.4 is using a deprecated SCSI ioctl, please convert it to SG_IO [366.027502] CPU: 5 PID: 28472 Comm: syz-executor.7 Not tainted 6.8.0-rc2-00035-gb3ef86b5a957 #1 [366.027512] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [366.027518] Call Trace: [366.027523] <TASK> [366.027533] dump_stack_lvl+0x93/0xd0 [366.027899] handle_overflow+0x171/0x1b0 [366.038787] ata1.00: invalid multi_count 32 ignored [366.043924] cdrom_ioctl+0x2c3f/0x2d10 [366.063932] ? __pm_runtime_resume+0xe6/0x130 [366.071923] sr_block_ioctl+0x15d/0x1d0 [366.074624] ? __pfx_sr_block_ioctl+0x10/0x10 [366.077642] blkdev_ioctl+0x419/0x500 [366.080231] ? __pfx_blkdev_ioctl+0x10/0x10 ... Historically, the signed integer overflow sanitizer did not work in the kernel due to its interaction with `fwrapv` but this has since been changed [1] in the newest version of Clang. It was re-enabled in the kernel with Commit 557f8c582a9ba8ab ("ubsan: Reintroduce signed overflow sanitizer"). Let's rearrange the check to not perform any arithmetic, thus not tripping the sanitizer.
CVE-2024-42139	In the Linux kernel, the following vulnerability has been resolved: ice: Fix improper extts handling Extts events are disabled and enabled by the application ts2phc. However, in case where the driver is removed when the application is running, a specific extts event remains enabled and can cause a kernel crash. As a side effect, when the driver is reloaded and application is started again, remaining extts event for the channel from a previous run will keep firing and the message "extts on unexpected channel" might be printed to the user. To avoid that, extts events shall be disabled when PTP is released.
CVE-2024-42145	In the Linux kernel, the following vulnerability has been resolved: IB/core: Implement a limit on UMAD receive List The existing behavior of ib_umad, which maintains received MAD packets in an unbounded list, poses a risk of uncontrolled growth. As user-space applications extract packets from this list, the rate of extraction may not match the rate of incoming packets, leading to potential list overflow. To address this, we introduce a limit to the size of the list. After considering typical scenarios, such as OpenSM processing, which can handle approximately 100k packets per second, and the 1-second retry timeout for most packets, we set the list size limit to 200k. Packets received beyond this limit are dropped, assuming they are likely timed out by the time they are handled by user-space. Notably, packets queued on the receive list due to reasons like timed-out sends are preserved even when the list is full.

CVE ID	Description
CVE-2024-42148	In the Linux kernel, the following vulnerability has been resolved: bnx2x: Fix multiple UBSAN array-index-out-of-bounds Fix UBSAN warnings that occur when using a system with 32 physical cpu cores or more, or when the user defines a number of Ethernet queues greater than or equal to FP_SB_MAX_E1x using the num_queues module parameter. Currently there is a read/write out of bounds that occurs on the array "struct stats_query_entry query" present inside the "bnx2x_fw_stats_req" struct in "drivers/net/ethernet/broadcom/bnx2x/bnx2x.h". Looking at the definition of the "struct stats_query_entry query" array: struct stats_query_entry query[FP_SB_MAX_E1x+BNX2X_FIRST_QUEUE_QUERY_IDX]; FP_SB_MAX_E1x is defined as the maximum number of fast path interrupts and has a value of 16, while BNX2X_FIRST_QUEUE_QUERY_IDX has a value of 3 meaning the array has a total size of 19. Since accesses to "struct stats_query_entry query" are offsetted by BNX2X_FIRST_QUEUE_QUERY_IDX, that means that the total number of Ethernet queues should not exceed FP_SB_MAX_E1x (16). However one of these queues is reserved for FCOE and thus the number of Ethernet queues should be set to [FP_SB_MAX_E1x -1] (15) if FCOE is enabled or [FP_SB_MAX_E1x] (16) if it is not. This is also described in a comment in the source code in drivers/net/ethernet/broadcom/bnx2x/bnx2x.h just above the Macro definition of FP_SB_MAX_E1x. Below is the part of this explanation that is important for this patch /* * The total number of L2 queues, MSIX vectors and HW contexts (CIDs) is * control by the number of fast-path status blocks supported by the * device (HW/FW). Each fast-path status block (FP-SB) aka non-default * status block represents an independent interrupts context that can * serve a regular L2 networking queue. However special L2 queues such * as the FCoE queue do not require a FP-SB and other components like * the CNIC may consume FP-SB reducing the number of possible L2 queues * * If the maximum number of FP-SB available is X then: * a. If CNIC is supported it consumes 1 FP-SB thus the max number of * regular L2 queues is Y=X-1 * b. In MF mode the actual number of L2 queues is Y= (X-1/MF_factor) * c. If the FCoE L2 queue is supported the actual number of L2 queues * is Y+1 * d. The number of irqs (MSIX vectors) is either Y+1 (one extra for * slow-path interrupts) or Y+2 if CNIC is supported (one additional * FP interrupt context for the CNIC). * e. The number of HW context (CID count) is always X or X+1 if FCoE * L2 queue is supported. The cid for the FCoE L2 queue is always X. */ However this driver also supports NICs that use the E2 controller which can handle more queues due to having more FP-SB represented by FP_SB_MAX_E2. Looking at the commits when the E2 support was added, it was originally using the E1x parameters: commit f2e0899f0f27 ("bnx2x: Add 57712 support"). Back then FP_SB_MAX_E2 was set to 16 the same as E1x. However the driver was later updated to take full advantage of the E2 instead of having it be limited to the capabilities of the E1x. But as far as we can tell, the array "stats_query_entry query" was still limited to using the FP-SB available to the E1x cards as part of an oversight when the driver was updated to take full advantage of the E2, and now with the driver being aware of the greater queue size supported by E2 NICs, it causes the UBSAN warnings seen in the stack traces below. This patch increases the size of the "stats_query_entry query" array by replacing FP_SB_MAX_E1x with FP_SB_MAX_E2 to be large enough to handle both types of NICs. Stack traces: UBSAN: array-index-out-of-bounds in drivers/net/ethernet/broadcom/bnx2x/bnx2x_stats.c:1529:11 index 20 is out of range for type 'stats_query_entry [19]' CPU: 12 PID: 858 Comm: systemd-network Not tainted 6.9.0-060900rc7-generic #202405052133 Hardware name: HP ProLiant DL360 Gen9/ProLiant DL360 ---truncated---
CVE-2024-42161	In the Linux kernel, the following vulnerability has been resolved: bpf: Avoid uninitialized value in BPF_CORE_READ_BITFIELD [Changes from V1: - Use a default branch in the switch statement to initialize 'val'.] GCC warns that 'val' may be used uninitialized in the BPF_CRE_READ_BITFIELD macro, defined in bpf_core_read.h as: [...] unsigned long long val; \ [...] \ switch (__CORE_RELO(s, field, BYTE_SIZE)) { \ case 1: val = *(const unsigned char *)p; break; \ case 2: val = *(const unsigned short *)p; break; \ case 4: val = *(const unsigned int *)p; break; \ case 8: val = *(const unsigned long long *)p; break; \ } \ [...] val; \ } \ This patch adds a default entry in the switch statement that sets 'val' to zero in order to avoid the warning, and random values to be used in case __builtin_preserve_field_info returns unexpected values for BPF_FIELD_BYTE_SIZE. Tested in bpf-next master. No regressions.
CVE-2024-42223	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: tda10048: Fix integer overflow state->xtal_hz can be up to 16M, so it can overflow a 32 bit integer when multiplied by pll_mfactor. Create a new 64 bit variable to hold the calculations.
CVE-2024-42229	In the Linux kernel, the following vulnerability has been resolved: crypto: aead,cipher - zeroize key buffer after use I.G 9.7.B for FIPS 140-3 specifies that variables temporarily holding cryptographic information should be zeroized once they are no longer needed. Accomplish this by using kfree_sensitive for buffers that previously held the private key.

CVE ID	Description
CVE-2024-42232	In the Linux kernel, the following vulnerability has been resolved: libceph: fix race between delayed_work() and ceph_monc_stop() The way the delayed work is handled in ceph_monc_stop() is prone to races with mon_fault() and possibly also finish_hunting(). Both of these can requeue the delayed work which wouldn't be canceled by any of the following code in case that happens after cancel_delayed_work_sync() runs -- __close_session() doesn't mess with the delayed work in order to avoid interfering with the hunting interval logic. This part was missed in commit b5d91704f53e ("libceph: behave in mon_fault() if cur_mon < 0") and use-after-free can still ensue on monc and objects that hang off of it, with monc->auth and monc->monmap being particularly susceptible to quickly being reused. To fix this: - clear monc->cur_mon and monc->hunting as part of closing the session in ceph_monc_stop() - bail from delayed_work() if monc->cur_mon is cleared, similar to how it's done in mon_fault() and finish_hunting() (based on monc->hunting) - call cancel_delayed_work_sync() after the session is closed
CVE-2024-42245	In the Linux kernel, the following vulnerability has been resolved: Revert "sched/fair: Make sure to try to detach at least one movable task" This reverts commit b0defa7ae03ecf91b8bfd10ede430cff12fcb06. b0defa7ae03ec changed the load balancing logic to ignore env.max_loop if all tasks examined to that point were pinned. The goal of the patch was to make it more likely to be able to detach a task buried in a long list of pinned tasks. However, this has the unfortunate side effect of creating an O(n) iteration in detach_tasks(), as we now must fully iterate every task on a cpu if all or most are pinned. Since this load balance code is done with rq lock held, and often in softirq context, it is very easy to trigger hard lockups. We observed such hard lockups with a user who affinity O(10k) threads to a single cpu. When I discussed this with Vincent he initially suggested that we keep the limit on the number of tasks to detach, but increase the number of tasks we can search. However, after some back and forth on the mailing list, he recommended we instead revert the original patch, as it seems likely no one was actually getting hit by the original issue.
CVE-2024-42259	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gem: Fix Virtual Memory mapping boundaries calculation Calculating the size of the mapped area as the lesser value between the requested size and the actual size does not consider the partial mapping offset. This can cause page fault access. Fix the calculation of the starting and ending addresses, the total size is now deduced from the difference between the end and start addresses. Additionally, the calculations have been rewritten in a clearer and more understandable form. [Joonas: Add Requires: tag] Requires: 60a2066c5005 ("drm/i915/gem: Adjust vma offset for framebuffer mmap offset") (cherry picked from commit 97b6784753da06d9d402328efc5c5367e53417)
CVE-2024-42268	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix missing lock on sync reset reload On sync reset reload work, when remote host updates devlink on reload actions performed on that host, it misses taking devlink lock before calling devlink_remote_reload_actions_performed() which results in triggering lock assert like the following: WARNING: CPU: 4 PID: 1164 at net/devlink/core.c:261 devl_assert_locked+0x3e/0x50 ... CPU: 4 PID: 1164 Comm: kworker/u96:6 Tainted: G S W 6.10.0-rc2+ #116 Hardware name: Supermicro SYS-2028TP-DECTR/X10DRT-PT, BIOS 2.0 12/18/2015 Workqueue: mlx5_fw_reset_events mlx5_sync_reset_reload_work [mlx5_core] RIP: 0010:devl_assert_locked+0x3e/0x50 ... Call Trace: ? __warn+0xa4/0x210 ? devl_assert_locked+0x3e/0x50 ? report_bug+0x160/0x280 ? handle_bug+0x3f/0x80 ? exc_invalid_op+0x17/0x40 ? asm_exc_invalid_op+0x1a/0x20 ? devl_assert_locked+0x3e/0x50 devlink_notify+0x88/0x2b0 ? mlx5_attach_device+0x20c/0x230 [mlx5_core] ? __pfx_devlink_notify+0x10/0x10 ? process_one_work+0x4b6/0xbb0 process_one_work+0x4b6/0xbb0 [...]
CVE-2024-42271	In the Linux kernel, the following vulnerability has been resolved: net/iucv: fix use after free in iucv_sock_close() iucv_sever_path() is called from process context and from bh context. iucv->path is used as indicator whether somebody else is taking care of severing the path (or it is already removed / never existed). This needs to be done with atomic compare and swap, otherwise there is a small window where iucv_sock_close() will try to work with a path that has already been severed and freed by iucv_callback_connrej() called by iucv_tasklet_fn(). Example: [452744.123844] Call Trace: [452744.123845] [<0000001e87f03880>] 0x1e87f03880 [452744.123966] [<00000000d593001e>] iucv_path_sever+0x96/0x138 [452744.124330] [<0000003ff801ddbca>] iucv_sever_path+0xc2/0xd0 [af_iucv] [452744.124336] [<0000003ff801e01b6>] iucv_sock_close+0xa6/0x310 [af_iucv] [452744.124341] [<0000003ff801e08cc>] iucv_sock_release+0x3c/0xd0 [af_iucv] [452744.124345] [<00000000d574794e>] __sock_release+0x5e/0xe8 [452744.124815] [<00000000d5747a0c>] sock_close+0x34/0x48 [452744.124820] [<00000000d5421642>] __fput+0xba/0x268 [452744.124826] [<00000000d51b382c>] task_work_run+0xbc/0xf0 [452744.124832] [<00000000d5145710>] do_notify_resume+0x88/0x90 [452744.124841] [<00000000d5978096>] system_call+0xe2/0x2c8 [452744.125319] Last Breaking-Event-Address: [452744.125321] [<00000000d5930018>] iucv_path_sever+0x90/0x138 [452744.125324] [452744.125325] Kernel panic - not syncing: Fatal exception in interrupt Note that bh_lock_sock() is not serializing the tasklet context against process context, because the check for sock_ownership() and corresponding handling is missing. Ideas for a future clean-up patch: A) Correct usage of bh_lock_sock() in tasklet context, as described in Re-enqueue, if needed. This may require adding return values to the tasklet functions and thus changes to all users of iucv. B) Change iucv tasklet into worker and use only lock_sock() in af_iucv.

CVE ID	Description
CVE-2024-42274	In the Linux kernel, the following vulnerability has been resolved: Revert "ALSA: firewire-lib: operate for period elapse event in process context" Commit 7ba5ca32fe6e ("ALSA: firewire-lib: operate for period elapse event in process context") removed the process context workqueue from amctp_domain_stream_pcm_pointer() and update_pcm_pointers() to remove its overhead. With RME Fireface 800, this lead to a regression since Kernels 5.14.0, causing an AB/BA deadlock competition for the substream lock with eventual system freeze under ALSA operation: thread 0: * (lock A) acquire substream lock by snd_pcm_stream_lock_irq() in snd_pcm_status64() * (lock B) wait for tasklet to finish by calling tasklet_unlock_spin_wait() in tasklet_disable_in_atomic() in ohci_flush_iso_completions() of ohci.c thread 1: * (lock B) enter tasklet * (lock A) attempt to acquire substream lock, waiting for it to be released: snd_pcm_stream_lock_irqsave() in snd_pcm_period_elapsed() in update_pcm_pointers() in process_ctx_payloads() in process_rx_packets() of amctp-stream.c ? tasklet_unlock_spin_wait </NMI> <TASK> ohci_flush_iso_completions firewire_ohci amctp_domain_stream_pcm_pointer snd_firewire_lib snd_pcm_update_hw_ptr0 snd_pcm snd_pcm_status64 snd_pcm ? native_queued_spin_lock_slowpath </NMI> <IRQ> _raw_spin_lock_irqsave snd_pcm_period_elapsed snd_pcm process_rx_packets snd_firewire_lib irq_target_callback snd_firewire_lib handle_it_packet firewire_ohci context_tasklet firewire_ohci Restore the process context work queue to prevent deadlock AB/BA deadlock competition for ALSA substream lock of snd_pcm_stream_lock_irq() in snd_pcm_status64() and snd_pcm_stream_lock_irqsave() in snd_pcm_period_elapsed(). revert commit 7ba5ca32fe6e ("ALSA: firewire-lib: operate for period elapse event in process context") Replace inline description to prevent future deadlock.
CVE-2024-42280	In the Linux kernel, the following vulnerability has been resolved: mISDN: Fix a use after free in hfcmulti_tx() Don't dereference *sp after calling dev_kfree_skb(*sp).
CVE-2024-42281	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix a segment issue when downgrading gso_size Linearize the skb when downgrading gso_size because it may trigger a BUG_ON() later when the skb is segmented as described in [1,2].
CVE-2024-42285	In the Linux kernel, the following vulnerability has been resolved: RDMA/iwcm: Fix a use-after-free related to destroying CM IDs iw_conn_req_handler() associates a new struct rdma_id_private (conn_id) with an existing struct iw_cm_id (cm_id) as follows: conn_id->cm_id.iw = cm_id; cm_id->context = conn_id; cm_id->cm_handler = cma_iw_handler; rdma_destroy_id() frees both the cm_id and the struct rdma_id_private. Make sure that cm_work_handler() does not trigger a use-after-free by only freeing of the struct rdma_id_private after all pending work has finished.
CVE-2024-42286	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: validate nvme_local_port correctly The driver load failed with error message, qla2xxx [0000:04:00.0]-ffff:0: register_localport failed: ret=fffffef and with a kernel crash, BUG: unable to handle kernel NULL pointer dereference at 0000000000000070 Workqueue: events_unbound qla_register_fcport_fn [qla2xxx] RIP: 0010:nvme_fc_register_remoteport+0x16/0x430 [nvme_fc] RSP: 0018:ffffaaa040eb3d98 EFLAGS: 00010282 RAX: 0000000000000000 RBX: ffff9dfb46b78c00 RCX: 0000000000000000 RDX: ffff9dfb46b78da8 RSI: fffffaa040eb3e08 RDI: 0000000000000000 RBP: ffff9dfb612a0a58 R08: ffffffffaf1d6270 R09: 3a34303a30303030 R10: 34303a303030305b R11: 2078787832616c71 R12: ffff9dfb46b78dd4 R13: ffff9dfb46b78c24 R14: ffff9dfb41525300 R15: ffff9dfb46b78da8 FS: 0000000000000000(0000) GS:ffff9dfc67c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000070 CR3: 000000018da10004 CR4: 00000000000206f0 Call Trace: qla_nvme_register_remote+0xeb/0x1f0 [qla2xxx] ? qla2x00_dfs_create_rport+0x231/0x270 [qla2xxx] qla2x00_update_fcport+0x2a1/0x3c0 [qla2xxx] qla_register_fcport_fn+0x54/0xc0 [qla2xxx] Exit the qla_nvme_register_remote() function when qla_nvme_register_hba() fails and correctly validate nvme_local_port.

CVE ID	Description
CVE-2024-42287	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Complete command early within lock A crash was observed while performing NPIV and FW reset, BUG: kernel NULL pointer dereference, address: 000000000000001c #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 1 PREEMPT_RT SMP NOPTI RIP: 0010:dma_direct_unmap_sg+0x51/0x1e0 RSP: 0018:ffff90026f47b88 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 0000000000000021 RCX: 0000000000000002 RDX: 0000000000000021 RSI: 0000000000000000 RDI: ffff8881041130d0 RBP: ffff8881041130d0 R08: 0000000000000000 R09: 0000000000000034 R10: ffff90026f47c48 R11: 0000000000000031 R12: 0000000000000000 R13: 0000000000000000 R14: ffff8881565e4a20 R15: 0000000000000000 FS: 00007f4c69ed3d00(0000) GS:ffff889faac80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000000001c CR3: 0000000288a50002 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> ? __die_body+0x1a/0x60 ? page_fault_oops+0x16f/0x4a0 ? do_user_addr_fault+0x174/0x7f0 ? exc_page_fault+0x69/0x1a0 ? asm_exc_page_fault+0x22/0x30 ? dma_direct_unmap_sg+0x51/0x1e0 ? preempt_count_sub+0x96/0xe0 qla2xxx_qpair_sp_free_dma+0x29f/0x3b0 [qla2xxx] qla2xxx_qpair_sp_compl+0x60/0x80 [qla2xxx] __qla2x00_abort_all_cmds+0xa2/0x450 [qla2xxx] The command completion was done early while aborting the commands in driver unload path but outside lock to avoid the WARN_ON condition of performing dma_free_attr within the lock. However this caused race condition while command completion via multiple paths causing system crash. Hence complete the command early in unload path but within the lock to avoid race condition.
CVE-2024-42288	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix for possible memory corruption Init Control Block is dereferenced incorrectly. Correctly dereference ICB
CVE-2024-42289	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: During vport delete send async logout explicitly During vport delete, it is observed that during unload we hit a crash because of stale entries in outstanding command array. For all these stale I/O entries, eh_abort was issued and aborted (fast_fail_io = 2009h) but I/Os could not complete while vport delete is in process of deleting. BUG: kernel NULL pointer dereference, address: 000000000000001c #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI Workqueue: qla2xxx_wq qla_do_work [qla2xxx] RIP: 0010:dma_direct_unmap_sg+0x51/0x1e0 RSP: 0018:ffffa1e150fc68 EFLAGS: 00010046 RAX: 0000000000000000 RBX: 0000000000000021 RCX: 0000000000000001 RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff8ce208a7a0d0 RBP: ffff8ce208a7a0d0 R08: 0000000000000000 R09: ffff8ce378aac9c8 R10: ffff8ce378aac8a0 R11: fffffa1e150f9d8 R12: 0000000000000000 R13: 0000000000000000 R14: ffff8ce378aac9c8 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff8d21f0000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000000001c CR3: 0000002089acc000 CR4: 0000000000350ee0 Call Trace: <TASK> qla2xxx_qpair_sp_free_dma+0x417/0x4e0 ? qla2xxx_qpair_sp_compl+0x10d/0x1a0 ? qla2x00_status_entry+0x768/0x2830 ? newidle_balance+0x2f0/0x430 ? dequeue_entity+0x100/0x3c0 ? qla24xx_process_response_queue+0x6a1/0x19e0 ? __schedule+0x2d5/0x1140 ? qla_do_work+0x47/0x60 ? process_one_work+0x267/0x440 ? process_one_work+0x440/0x440 ? worker_thread+0x2d/0x3d0 ? process_one_work+0x440/0x440 ? kthread+0x156/0x180 ? set_kthread_struct+0x50/0x50 ? ret_from_fork+0x22/0x30 </TASK> Send out async logout explicitly for all the ports during vport delete.
CVE-2024-42291	In the Linux kernel, the following vulnerability has been resolved: ice: Add a per-VF limit on number of FDIR filters While the iavf driver adds a s/w limit (128) on the number of FDIR filters that the VF can request, a malicious VF driver can request more than that and exhaust the resources for other VFs. Add a similar limit in ice.
CVE-2024-42304	In the Linux kernel, the following vulnerability has been resolved: ext4: make sure the first directory block is not a hole The syzbot constructs a directory that has no dirblock but is non-inline, i.e. the first directory block is a hole. And no errors are reported when creating files in this directory in the following flow. ext4_mknod ... ext4_add_entry // Read block 0 ext4_read_dirblock(dir, block, DIRENT) bh = ext4_bread(NULL, inode, block, 0) if (!bh && (type == INDEX type == DIRENT_HTREE)) // The first directory block is a hole // But type == DIRENT, so no error is reported. After that, we get a directory block without '.' and '.' but with a valid dentry. This may cause some code that relies on dot or dotdot (such as make_indexed_dir()) to crash. Therefore when ext4_read_dirblock() finds that the first directory block is a hole report that the filesystem is corrupted and return an error to avoid loading corrupted data from disk causing something bad.

CVE ID	Description
CVE-2024-42305	In the Linux kernel, the following vulnerability has been resolved: ext4: check dot and dotdot of dx_root before making dir indexed Syzbot reports a issue as follows: <pre>===== BUG: unable to handle page fault for address: ffffed11022e24fe PGD 23ffee067 P4D 23ffee067 PUD 0 Oops: Oops: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 0 PID: 5079 Comm: syz-executor306 Not tainted 6.10.0-rc5-g55027e689933 #0 Call Trace: <TASK> make_indexed_dir+0xdaf/0x13c0 fs/ext4/namei.c:2341 ext4_add_entry+0x222a/0x25d0 fs/ext4/namei.c:2451 ext4_rename fs/ext4/namei.c:3936 [inline] ext4_rename2+0x26e5/0x4370 fs/ext4/namei.c:4214 [...]</pre> ===== The immediate cause of this problem is that there is only one valid dentry for the block to be split during do_split, so split==0 results in out of bounds accesses to the map triggering the issue. do_split unsigned split dx_make_map count = 1 split = count/2 = 0; continued = hash2 == map[split - 1].hash; ---> map[4294967295] The maximum length of a filename is 255 and the minimum block size is 1024, so it is always guaranteed that the number of entries is greater than or equal to 2 when do_split() is called. But syzbot's crafted image has no dot and dotdot in dir, and the dentry distribution in dirblock is as follows: bus dentry1 hole dentry2 free [xx-] xx-----[.....]xx-----[.....] 0 12 (8+248)=256 268 256 524 (8+256)=264 788 236 1024 So when renaming dentry1 increases its name_len length by 1, neither hole nor free is sufficient to hold the new dentry, and make_indexed_dir() is called. In make_indexed_dir() it is assumed that the first two entries of the dirblock must be dot and dotdot, so bus and dentry1 are left in dx_root because they are treated as dot and dotdot, and only dentry2 is moved to the new leaf block. That's why count is equal to 1. Therefore add the ext4_check_dx_root() helper function to add more sanity checks to dot and dotdot before starting the conversion to avoid the above issue.
CVE-2024-42309	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: fix null pointer dereference in psb_intel_lvds_get_modes In psb_intel_lvds_get_modes(), the return value of drm_mode_duplicate() is assigned to mode, which will lead to a possible NULL pointer dereference on failure of drm_mode_duplicate(). Add a check to avoid npd.
CVE-2024-42310	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: fix null pointer dereference in cdv_intel_lvds_get_modes In cdv_intel_lvds_get_modes(), the return value of drm_mode_duplicate() is assigned to mode, which will lead to a NULL pointer dereference on failure of drm_mode_duplicate(). Add a check to avoid npd.
CVE-2024-42321	In the Linux kernel, the following vulnerability has been resolved: net: flow_dissector: use DEBUG_NET_WARN_ON_ONCE The following splat is easy to reproduce upstream as well as in -stable kernels. Florian Westphal provided the following commit: d1dab4f71d37 ("net: add and use __skb_get_hash_symmetric_net") but this complementary fix has been also suggested by Willem de Bruijn and it can be easily backported to -stable kernel which consists in using DEBUG_NET_WARN_ON_ONCE instead to silence the following splat given __skb_get_hash() is used by the nftables tracing infrastructure to to identify packets in traces. [69133.561393] <pre>-----[cut here]----- [69133.561404] WARNING: CPU: 0 PID: 43576 at net/core/ flow_dissector.c:1104 __skb_flow_dissect+0x134f/ [...] [69133.561944] CPU: 0 PID: 43576 Comm: socat Not tainted 6.10.0-rc7+ #379 [69133.561959] RIP: 0010: __skb_flow_dissect +0x134f/0x2ad0 [69133.561970] Code: 83 f9 04 0f 84 b3 00 00 00 45 85 c9 0f 84 aa 00 00 00 41 83 f9 02 0f 84 81 fc ff ff 44 0f b7 b4 24 80 00 00 00 e9 8b f9 ff ff <0f> 0b e9 20 f3 ff ff 41 f6 c6 20 0f 84 e4 ef ff ff 48 8d 7b 12 e8 [69133.561979] RSP: 0018:ffffc90000006fc0 EFLAGS: 00010246 [69133.561988] RAX: 0000000000000000 RBX: ffffffff82f33e20 RCX: ffffffff81ab7e19 [69133.561994] RDX: dffffc0000000000 RSI: fffffc900000007388 RDI: ffff888103a1b418 [69133.562001] RBP: fffffc900000007310 R08: 0000000000000000 R09: 0000000000000000 [69133.562007] R10: fffffc900000007388 R11: ffffffff810cfac0 R12: ffff888103a1b400 [69133.562013] R13: 0000000000000000 R14: ffffffff82f33e2a R15: ffffffff82f33e28 [69133.562020] FS: 00007f40f7131740(0000) GS:ffff888390800000(0000) knlGS:0000000000000000 [69133.562027] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [69133.562033] CR2: 00007f40f7346ee0 CR3: 000000015d200001 CR4: 00000000001706f0 [69133.562040] Call Trace: [69133.562044] <IRQ> [69133.562049] ? __warn+0x9f/0x1a0 [1211.841384] ? __skb_flow_dissect+0x107e/0x2860 [...] [1211.841496] ? bpf_flow_dissect+0x160/0x160 [1211.841753] __skb_get_hash+0x97/0x280 [1211.841765] ? __skb_get_hash_symmetric+0x230/0x230 [1211.841776] ? mod_find+0xbf/0xe0 [1211.841786] ? get_stack_info_noinstr+0x12/0xe0 [1211.841798] ? bpf_ksym_find+0x56/0xe0 [1211.841807] ? __rcu_read_unlock+0x2a/0x70 [1211.841819] nft_trace_init+0x1b9/0x1c0 [nf_tables] [1211.841895] ? nft_trace_notify+0x830/0x830 [nf_tables] [1211.841964] ? get_stack_info+0x2b/0x80 [1211.841975] ? nft_do_chain_arp+0x80/0x80 [nf_tables] [1211.842044] nft_do_chain+0x79c/0x850 [nf_tables]</pre>

CVE ID	Description
CVE-2024-42367	aiohttp is an asynchronous HTTP client/server framework for asyncio and Python. In versions on the 3.10 branch prior to version 3.10.2, static routes which contain files with compressed variants (`.gz` or `.br` extension) are vulnerable to path traversal outside the root directory if those variants are symbolic links. The server protects static routes from path traversal outside the root directory when `follow_symlinks=False` (default). It does this by resolving the requested URL to an absolute path and then checking that path relative to the root. However, these checks are not performed when looking for compressed variants in the `FileResponse` class, and symbolic links are then automatically followed when performing the `Path.stat()` and `Path.open()` to send the file. Version 3.10.2 contains a patch for the issue.
CVE-2024-42516	HTTP response splitting in the core of Apache HTTP Server allows an attacker who can manipulate the Content-Type response headers of applications hosted or proxied by the server can split the HTTP response. This vulnerability was described as CVE-2023-38709 but the patch included in Apache HTTP Server 2.4.59 did not address the issue. Users are recommended to upgrade to version 2.4.64, which fixes this issue.
CVE-2024-43098	In the Linux kernel, the following vulnerability has been resolved: i3c: Use i3cdev->desc->info instead of calling i3c_device_get_info() to avoid deadlock A deadlock may happen since the i3c_master_register() acquires &i3cbus->lock twice. See the log below. Use i3cdev->desc->info instead of calling i3c_device_info() to avoid acquiring the lock twice. v2: - Modified the title and commit message ===== WARNING: possible recursive locking detected 6.11.0-mainline ----- init/1 is trying to acquire lock: f1fff80a6a40dc0 (&i3cbus->lock){++++}-{3:3}, at: i3c_bus_normaluse_lock but task is already holding lock: f1fff80a6a40dc0 (&i3cbus->lock){++++}-{3:3}, at: i3c_master_register other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(&i3cbus->lock); lock(&i3cbus->lock); *** DEADLOCK *** May be due to missing lock nesting notation 2 locks held by init/1: #0: fcfff809b6798f8 (&dev->mutex){....}-{3:3}, at: __driver_attach #1: f1fff80a6a40dc0 (&i3cbus->lock){++++}-{3:3}, at: i3c_master_register stack backtrace: CPU: 6 UID: 0 PID: 1 Comm: init Call trace: dump_backtrace+0xfc/0x17c show_stack+0x18/0x28 dump_stack_lvl+0x40/0xc0 dump_stack+0x18/0x24 print_deadlock_bug+0x388/0x390 __lock_acquire+0x18bc/0x32ec lock_acquire+0x134/0x2b0 down_read+0x50/0x19c i3c_bus_normaluse_lock+0x14/0x24 i3c_device_get_info+0x24/0x58 i3c_device_uevent+0x34/0xa4 dev_uevent+0x310/0x384 kobject_uevent_env+0x244/0x414 kobject_uevent+0x14/0x20 device_add+0x278/0x460 device_register+0x20/0x34 i3c_master_register_new_i3c_devs+0x78/0x154 i3c_master_register+0x6a0/0x6d4 mtk_i3c_master_probe+0x3b8/0x4d8 platform_probe+0xa0/0xe0 really_probe+0x114/0x454 __driver_probe_device+0xa0/0x15c driver_probe_device+0x3c/0x1ac __driver_attach+0xc4/0x1f0 bus_for_each_dev+0x104/0x160 driver_attach+0x24/0x34 bus_add_driver+0x14c/0x294 driver_register+0x68/0x104 __platform_driver_register+0x20/0x30 init_module+0x20/0xfe4 do_one_initcall+0x184/0x464 do_init_module+0x58/0x1ec load_module+0xefc/0x10c8 __arm64_sys_finit_module+0x238/0x33c invoke_syscall+0x58/0x10c el0_svc_common+0xa8/0xdc do_el0_svc+0x1c/0x28 el0_svc+0x50/0xac el0t_64_sync_handler+0x70/0xbc el0t_64_sync+0x1a8/0x1ac
CVE-2024-43204	SSRF in Apache HTTP Server with mod_proxy loaded allows an attacker to send outbound proxy requests to a URL controlled by the attacker. Requires an unlikely configuration where mod_headers is configured to modify the Content-Type request or response header with a value provided in the HTTP request. Users are recommended to upgrade to version 2.4.64 which fixes this issue.
CVE-2024-43374	The UNIX editor Vim prior to version 9.1.0678 has a use-after-free error in argument list handling. When adding a new file to the argument list, this triggers `Buf*` autocommands. If in such an autocommand the buffer that was just opened is closed (including the window where it is shown), this causes the window structure to be freed which contains a reference to the argument list that we are actually modifying. Once the autocommands are completed, the references to the window and argument list are no longer valid and as such cause an use-after-free. Impact is low since the user must either intentionally add some unusual autocommands that wipe a buffer during creation (either manually or by sourcing a malicious plugin), but it will crash Vim. The issue has been fixed as of Vim patch v9.1.0678.
CVE-2024-43382	Snowflake JDBC driver versions >= 3.2.6 and <= 3.19.1 have an Incorrect Security Setting that can result in data being uploaded to an encrypted stage without the additional layer of protection provided by client side encryption.
CVE-2024-43398	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.6 has a DoS vulnerability when it parses an XML that has many deep elements that have same local name attributes. If you need to parse untrusted XMLs with tree parser API like REXML::Document.new, you may be impacted to this vulnerability. If you use other parser APIs such as stream parser API and SAX2 parser API, this vulnerability is not affected. The REXML gem 3.3.6 or later include the patch to fix the vulnerability.

CVE ID	Description
CVE-2024-43802	Vim is an improved version of the unix vi text editor. When flushing the typeahead buffer, Vim moves the current position in the typeahead buffer but does not check whether there is enough space left in the buffer to handle the next characters. So this may lead to the tb_off position within the typebuf variable to point outside of the valid buffer size, which can then later lead to a heap-buffer overflow in e.g. ins_typebuf(). Therefore, when flushing the typeahead buffer, check if there is enough space left before advancing the off position. If not, fall back to flush current typebuf contents. It's not quite clear yet, what can lead to this situation. It seems to happen when error messages occur (which will cause Vim to flush the typeahead buffer) in combination with several long mappings and so it may eventually move the off position out of a valid buffer size. Impact is low since it is not easily reproducible and requires to have several mappings active and run into some error condition. But when this happens, this will cause a crash. The issue has been fixed as of Vim patch v9.1.0697. Users are advised to upgrade. There are no known workarounds for this issue.
CVE-2024-43817	In the Linux kernel, the following vulnerability has been resolved: net: missing check virtio Two missing check in virtio_net_hdr_to_skb() allowed syzbot to crash kernels again 1. After the skb_segment function the buffer may become non-linear (nr_frags != 0), but since the SKBTX_SHARED_FRAG flag is not set anywhere the __skb_linearize function will not be executed, then the buffer will remain non-linear. Then the condition (offset >= skb_headlen(skb)) becomes true, which causes WARN_ON_ONCE in skb_checksum_help. 2. The struct sk_buff and struct virtio_net_hdr members must be mathematically related. (gso_size) must be greater than (needed) otherwise WARN_ON_ONCE. (remainder) must be greater than (needed) otherwise WARN_ON_ONCE. (remainder) may be 0 if division is without remainder. offset+2 (4191) > skb_headlen() (1116) WARNING: CPU: 1 PID: 5084 at net/core/dev.c:3303 skb_checksum_help+0x5e2/0x740 net/core/dev.c:3303 Modules linked in: CPU: 1 PID: 5084 Comm: syz-executor336 Not tainted 6.7.0-rc3-syzkaller-00014-gd6f0cee26a2e #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 11/10/2023 RIP: 0010:skb_checksum_help+0x5e2/0x740 net/core/dev.c:3303 Code: 89 e8 83 e0 07 83 c0 03 38 d0 7c 08 84 d2 0f 85 52 01 00 00 44 89 e2 2b 53 74 4c 89 ee 48 c7 c7 40 57 e9 8b e8 af 8f dd f8 90 <0f> 0b 90 90 e9 87 fe ff ff e8 40 0f 6e f9 e9 4b fa ff ff 48 89 ef RSP: 0018:ffff90003a9f338 EFLAGS: 00010286 RAX: 0000000000000000 RBX: ffff888025125780 RCX: ffffffff814db209 RDX: ffff888015393b80 RSI: ffffffff814db216 RDI: 0000000000000001 RBP: ffff8880251257f4 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000001 R12: 000000000000045c R13: 00000000000105f R14: ffff8880251257f0 R15: 00000000000105d FS: 000055555c24380(0000) GS:ffff8880b9900000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000002000f000 CR3: 0000000023151000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ip_do_fragment+0xa1b/0x18b0 net/ipv4/ip_output.c:777 ip_fragment.constprop.0+0x161/0x230 net/ipv4/ip_output.c:584 ip_finish_output_gso net/ipv4/ip_output.c:286 [inline] __ip_finish_output net/ipv4/ip_output.c:308 [inline] __ip_finish_output+0x49c/0x650 net/ipv4/ip_output.c:295 ip_finish_output+0x31/0x310 net/ipv4/ip_output.c:323 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip_output+0x13b/0x2a0 net/ipv4/ip_output.c:433 dst_output include/net/dst.h:451 [inline] ip_local_out+0xaf/0x1a0 net/ipv4/ip_output.c:129 iptunnel_xmit+0x5b4/0x9b0 net/ipv4/ip_tunnel_core.c:82 ipip6_tunnel_xmit net/ipv6/sit.c:1034 [inline] sit_tunnel_xmit+0xed2/0x28f0 net/ipv6/sit.c:1076 __netdev_start_xmit include/linux/netdevice.h:4940 [inline] netdev_start_xmit include/linux/netdevice.h:4954 [inline] xmit_one net/core/dev.c:3545 [inline] dev_hard_start_xmit+0x13d/0x6d0 net/core/dev.c:3561 __dev_queue_xmit+0x7c1/0x3d60 net/core/dev.c:4346 dev_queue_xmit include/linux/netdevice.h:3134 [inline] packet_xmit+0x257/0x380 net/packet/af_packet.c:276 packet_snd net/packet/af_packet.c:3087 [inline] packet_sendmsg+0x24ca/0x5240 net/packet/af_packet.c:3119 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0xd5/0x180 net/socket.c:745 __sys_sendto+0x255/0x340 net/socket.c:2190 __do_sys_sendto net/socket.c:2202 [inline] __se_sys_sendto net/socket.c:2198 [inline] __x64_sys_sendto+0xe0/0x1b0 net/socket.c:2198 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x40/0x110 arch/x86/entry/common.c:82 entry_SYSCALL_64_after_hwframe+0x63/0xb6 Found by Linux Verification Center (linuxtesting.org) with Syzkaller
CVE-2024-43819	In the Linux kernel, the following vulnerability has been resolved: kvm: s390: Reject memory region operations for ucontrol VMs This change rejects the KVM_SET_USER_MEMORY_REGION and KVM_SET_USER_MEMORY_REGION2 ioctls when called on a ucontrol VM. This is necessary since ucontrol VMs have kvm->arch.gmap set to 0 and would thus result in a null pointer dereference further in. Memory management needs to be performed in userspace and using the ioctls KVM_S390_UCAS_MAP and KVM_S390_UCAS_UNMAP. Also improve s390 specific documentation for KVM_SET_USER_MEMORY_REGION and KVM_SET_USER_MEMORY_REGION2. [frankja@linux.ibm.com: commit message spelling fix, subject prefix fix]

CVE ID	Description
CVE-2024-43821	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix a possible null pointer dereference In function lpfc_xcvt_data_show, the memory allocation with kmalloc might fail, thereby making rdp_context a null pointer. In the following context and functions that use this pointer, there are dereferencing operations, leading to null pointer dereference. To fix this issue, a null pointer check should be added. If it is null, use scnprintf to notify the user and return len.
CVE-2024-43834	In the Linux kernel, the following vulnerability has been resolved: xdp: fix invalid wait context of page_pool_destroy() If the driver uses a page pool, it creates a page pool with page_pool_create(). The reference count of page pool is 1 as default. A page pool will be destroyed only when a reference count reaches 0. page_pool_destroy() is used to destroy page pool, it decreases a reference count. When a page pool is destroyed, ->disconnect() is called, which is mem_allocator_disconnect(). This function internally acquires mutex_lock(). If the driver uses XDP, it registers a memory model with xdp_rxq_info_reg_mem_model(). The xdp_rxq_info_reg_mem_model() internally increases a page pool reference count if a memory model is a page pool. Now the reference count is 2. To destroy a page pool, the driver should call both page_pool_destroy() and xdp_unreg_mem_model(). The xdp_unreg_mem_model() internally calls page_pool_destroy(). Only page_pool_destroy() decreases a reference count. If a driver calls page_pool_destroy() then xdp_unreg_mem_model(), we will face an invalid wait context warning. Because xdp_unreg_mem_model() calls page_pool_destroy() with rcu_read_lock(). The page_pool_destroy() internally acquires mutex_lock(). Splat looks like: <pre> ===== [BUG: Invalid wait context] 6.10.0-rc6+ #4 Tainted: G W ----- ethtool/1806 is trying to lock: ffffffff90387b90 (mem_id_lock){+.,.}- {4:4}, at: mem_allocator_disconnect+0x73/0x150 other info that might help us debug this: context- {5:5} 3 locks held by ethtool/1806: stack backtrace: CPU: 0 PID: 1806 Comm: ethtool Tainted: G W 6.10.0-rc6+ #4 f916f41f172891c800f2fed Hardware name: ASUS System Product Name/ PRIME Z690-P D4, BIOS 0603 11/01/2021 Call Trace: <TASK> dump_stack_lvl+0x7e/0xc0 __lock_acquire+0x1681/0x4de0 ? _printk+0x64/0xe0 ? __pfx_mark_lock.part.0+0x10/0x10 ? __pfx__lock_acquire+0x10/0x10 lock_acquire+0x1b3/0x580 ? mem_allocator_disconnect +0x73/0x150 ? __wake_up_klogd.part.0+0x16/0xc0 ? __pfx_lock_acquire+0x10/0x10 ? dump_stack_lvl+0x91/0xc0 __mutex_lock+0x15c/0x1690 ? mem_allocator_disconnect+0x73/0x150 ? __pfx_prb_read_valid+0x10/0x10 ? mem_allocator_disconnect+0x73/0x150 ? __pfx_llist_add_batch +0x10/0x10 ? console_unlock+0x193/0x1b0 ? lockdep_hardirqs_on+0xbe/0x140 ? __pfx__mutex_lock +0x10/0x10 ? tick_nohz_tick_stopped+0x16/0x90 ? __irq_work_queue_local+0x1e5/0x330 ? irq_work_queue+0x39/0x50 ? __wake_up_klogd.part.0+0x79/0xc0 ? mem_allocator_disconnect +0x73/0x150 mem_allocator_disconnect+0x73/0x150 ? __pfx_mem_allocator_disconnect +0x10/0x10 ? mark_held_locks+0xa5/0xf0 ? rcu_is_watching+0x11/0xb0 page_pool_release +0x36e/0x6d0 page_pool_destroy+0xd7/0x440 xdp_unreg_mem_model+0x1a7/0x2a0 ? __pfx_xdp_unreg_mem_model+0x10/0x10 ? kfree+0x125/0x370 ? bnxt_free_ring.isra.0+0x2eb/0x500 ? bnxt_free_mem+0x5ac/0x2500 xdp_rxq_info_unreg+0x4a/0xd0 bnxt_free_mem+0x1356/0x2500 bnxt_close_nic+0xf0/0x3b0 ? __pfx_bnxt_close_nic+0x10/0x10 ? ethnl_parse_bit+0x2c6/0x6d0 ? __pfx__nla_validate_parse+0x10/0x10 ? __pfx_ethnl_parse_bit+0x10/0x10 bnxt_set_features +0x2a8/0x3e0 __netdev_update_features+0x4dc/0x1370 ? ethnl_parse_bitset+0x4ff/0x750 ? __pfx_ethnl_parse_bitset+0x10/0x10 ? __pfx__netdev_update_features+0x10/0x10 ? mark_held_locks +0xa5/0xf0 ? __raw_spin_unlock_irqrestore+0x42/0x70 ? __pm_runtime_resume+0x7d/0x110 ethnl_set_features+0x32d/0xa20 To fix this problem, it uses rhashtable_lookup_fast() instead of rhashtable_lookup() with rcu_read_lock(). Using xa without rcu_read_lock() here is safe. xa is freed by __xdp_mem_allocator_rcu_free() and this is called by call_rcu() of mem_xa_remove(). The mem_xa_remove() is called by page_pool_destroy() if a reference count reaches 0. The xa is already protected by the reference count mechanism well in the control plane. So removing rcu_read_lock() for page_pool_destroy() is safe.</pre>
CVE-2024-43835	In the Linux kernel, the following vulnerability has been resolved: virtio_net: Fix napi_skb_cache_put warning After the commit bdacf3e34945 ("net: Use nested-BH locking for napi_alloc_cache.") was merged, the following warning began to appear: WARNING: CPU: 5 PID: 1 at net/core/skbuff.c:1451 napi_skb_cache_put+0x82/0x4b0 __warn+0x12f/0x340 napi_skb_cache_put+0x82/0x4b0 napi_skb_cache_put+0x82/0x4b0 report_bug+0x165/0x370 handle_bug+0x3d/0x80 exc_invalid_op+0x1a/0x50 asm_exc_invalid_op+0x1a/0x20 __free_old_xmit+0x1c8/0x510 napi_skb_cache_put+0x82/0x4b0 __free_old_xmit+0x1c8/0x510 __free_old_xmit+0x1c8/0x510 __pfx__free_old_xmit+0x10/0x10 The issue arises because virtio is assuming it's running in NAPI context even when it's not, such as in the netpoll case. To resolve this, modify virtnet_poll_tx() to only set NAPI when budget is available. Same for virtnet_poll_clean_tx(), which always assumed that it was in a NAPI context.

[illegible]

CVE ID	Description
CVE-2024-43853	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: Prevent UAF in proc_cpuset_show() An UAF can happen when /proc/cpuset is read as reported in [1]. This can be reproduced by the following methods: 1.add an mdelay(1000) before acquiring the cgroup_lock In the cgroup_path_ns function. 2.\$cat /proc/<pid>/cpuset repeatedly. 3.\$mount -t cgroup -o cpuset cpuset /sys/fs/cgroup/cpuset/ \$umount /sys/fs/cgroup/cpuset/ repeatedly. The race that cause this bug can be shown as below: (umount) (cat /proc/<pid>/cpuset) css_release proc_cpuset_show css_release_work_fn css = task_get_css(tsk, cpuset_cgrp_id); css_free_rwork_fn cgroup_path_ns(css->cgroup, ...); cgroup_destroy_root mutex_lock(&cgroup_mutex); rebind_subsystems cgroup_free_root // cgrp was freed, UAF cgroup_path_ns_locked(cgrp,...); When the cpuset is initialized, the root node top_cpuset.css.cgrp will point to &cgrp_dfl_root.cgrp. In cgroup v1, the mount operation will allocate cgroup_root, and top_cpuset.css.cgrp will point to the allocated &cgroup_root.cgrp. When the umount operation is executed, top_cpuset.css.cgrp will be rebound to &cgrp_dfl_root.cgrp. The problem is that when rebinding to cgrp_dfl_root, there are cases where the cgroup_root allocated by setting up the root for cgroup v1 is cached. This could lead to a Use-After-Free (UAF) if it is subsequently freed. The descendant cgroups of cgroup v1 can only be freed after the css is released. However, the css of the root will never be released, yet the cgroup_root should be freed when it is unmounted. This means that obtaining a reference to the css of the root does not guarantee that css.cgrp->root will not be freed. Fix this problem by using rcu_read_lock in proc_cpuset_show(). As cgroup_root is kfree_rcu after commit d23b5c577715 ("cgroup: Make operations on the cgroup root_list RCU safe"), css->cgroup won't be freed during the critical section. To call cgroup_path_ns_locked, css_set_lock is needed, so it is safe to replace task_get_css with task_css. [1] https://syzkaller.appspot.com/bug?extid=9b1ff7be974a403aa4cd
CVE-2024-43856	In the Linux kernel, the following vulnerability has been resolved: dma: fix call order in dmam_free_coherent dmam_free_coherent() frees a DMA allocation, which makes the freed vaddr available for reuse, then calls devres_destroy() to remove and free the data structure used to track the DMA allocation. Between the two calls, it is possible for a concurrent task to make an allocation with the same vaddr and add it to the devres list. If this happens, there will be two entries in the devres list with the same vaddr and devres_destroy() can free the wrong entry, triggering the WARN_ON() in dmam_match. Fix by destroying the devres entry before freeing the DMA allocation. kokonut //net/encryption http://sponge2/b9145fe6-0f72-4325-ac2f-a84d81075b03
CVE-2024-43861	In the Linux kernel, the following vulnerability has been resolved: net: usb: qmi_wwan: fix memory leak for not ip packets Free the unused skb when not ip packets arrive.
CVE-2024-43863	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Fix a deadlock in dma buf fence polling Introduce a version of the fence ops that on release doesn't remove the fence from the pending list, and thus doesn't require a lock to fix poll->fence wait->fence unref deadlocks. vmwgfx overwrites the wait callback to iterate over the list of all fences and update their status, to do that it holds a lock to prevent the list modifications from other threads. The fence destroy callback both deletes the fence and removes it from the list of pending fences, for which it holds a lock. dma buf polling cb unrefs a fence after it's been signaled: so the poll calls the wait, which signals the fences, which are being destroyed. The destruction tries to acquire the lock on the pending fences list which it can never get because it's held by the wait from which it was called. Old bug, but not a lot of userspace apps were using dma-buf polling interfaces. Fix those, in particular this fixes KDE stalls/deadlock.
CVE-2024-43866	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Always drain health in shutdown callback There is no point in recovery during device shutdown. if health work started need to wait for it to avoid races and NULL pointer access. Hence, drain health WQ on shutdown callback.
CVE-2024-43872	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix soft lockup under heavy CEQE load CEQEs are handled in interrupt handler currently. This may cause the CPU core staying in interrupt context too long and lead to soft lockup under heavy load. Handle CEQEs in BH workqueue and set an upper limit for the number of CEQE handled by a single call of work handler.

CVE ID	Description
CVE-2024-43882	In the Linux kernel, the following vulnerability has been resolved: exec: Fix ToCToU between perm check and set-uid/gid usage When opening a file for exec via do_filp_open(), permission checking is done against the file's metadata at that moment, and on success, a file pointer is passed back. Much later in the execve() code path, the file metadata (specifically mode, uid, and gid) is used to determine if/how to set the uid and gid. However, those values may have changed since the permissions check, meaning the execution may gain unintended privileges. For example, if a file could change permissions from executable and not set-id: -----x 1 root root 16048 Aug 7 13:16 target to set-id and non-executable: ---S----- 1 root root 16048 Aug 7 13:16 target it is possible to gain root privileges when execution should have been disallowed. While this race condition is rare in real-world scenarios, it has been observed (and proven exploitable) when package managers are updating the setuid bits of installed programs. Such files start with being world-executable but then are adjusted to be group-exec with a set-uid bit. For example, "chmod o-x,u+s target" makes "target" executable only by uid "root" and gid "cdrom", while also becoming setuid-root: -rwxr-xr-x 1 root cdrom 16048 Aug 7 13:16 target becomes: -rwsr-xr-- 1 root cdrom 16048 Aug 7 13:16 target But racing the chmod means users without group "cdrom" membership can get the permission to execute "target" just before the chmod, and when the chmod finishes, the exec reaches brpm_fill_uid(), and performs the setuid to root, violating the expressed authorization of "only cdrom group members can setuid to root". Re-check that we still have execute permissions in case the metadata has changed. It would be better to keep a copy from the perm-check time, but until we can do that refactoring, the least-bad option is to do a full inode_permission() call (under inode lock). It is understood that this is safe against dead-locks, but hardly optimal.
CVE-2024-43884	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Add error handling to pair_device() hci_conn_params_add() never checks for a NULL value and could lead to a NULL pointer dereference causing a crash. Fixed by adding error handling in the function.
CVE-2024-43890	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix overflow in get_free_elt() "tracing_map->next_elt" in get_free_elt() is at risk of overflowing. Once it overflows, new elements can still be inserted into the tracing_map even though the maximum number of elements ('max_elts') has been reached. Continuing to insert elements after the overflow could result in the tracing_map containing "tracing_map->max_size" elements, leaving no empty entries. If any attempt is made to insert an element into a full tracing_map using '__tracing_map_insert()', it will cause an infinite loop with preemption disabled, leading to a CPU hang problem. Fix this by preventing any further increments to "tracing_map->next_elt" once it reaches "tracing_map->max_elt".
CVE-2024-43893	In the Linux kernel, the following vulnerability has been resolved: serial: core: check uartclk for zero to avoid divide by zero Calling ioctl TIOCSSERIAL with an invalid baud_base can result in uartclk being zero, which will result in a divide by zero error in uart_get_divisor(). The check for uartclk being zero in uart_set_info() needs to be done before other settings are made as subsequent calls to ioctl TIOCSSERIAL for the same port would be impacted if the uartclk check was done where uartclk gets set. Oops: divide error: 0000 PREEMPT SMP KASAN PTI RIP: 0010:uart_get_divisor (drivers/tty/serial/serial_core.c:580) Call Trace: <TASK> serial8250_get_divisor (drivers/tty/serial/8250/8250_port.c:2576 drivers/tty/serial/8250/8250_port.c:2589) serial8250_do_set_termios (drivers/tty/serial/8250/8250_port.c:502 drivers/tty/serial/8250/8250_port.c:2741) serial8250_set_termios (drivers/tty/serial/8250/8250_port.c:2862) uart_change_line_settings (./include/linux/spinlock.h:376 ./include/linux/serial_core.h:608 drivers/tty/serial/serial_core.c:222) uart_port_startup (drivers/tty/serial/serial_core.c:342) uart_startup (drivers/tty/serial/serial_core.c:368) uart_set_info (drivers/tty/serial/serial_core.c:1034) uart_set_info_user (drivers/tty/serial/serial_core.c:1059) tty_set_serial (drivers/tty/tty_io.c:2637) tty_ioctl (drivers/tty/tty_io.c:2647 drivers/tty/tty_io.c:2791) __x64_sys_ioctl (fs/ioctl.c:52 fs/ioctl.c:907 fs/ioctl.c:893 fs/ioctl.c:893) do_syscall_64 (arch/x86/entry/common.c:52 (discriminator 1) arch/x86/entry/common.c:83 (discriminator 1)) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) Rule: add
CVE-2024-43906	In the Linux kernel, the following vulnerability has been resolved: drm/admgpu: fix dereferencing null pointer context When user space sets an invalid ta type, the pointer context will be empty. So it need to check the pointer context before using it
CVE-2024-43908	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix the null pointer dereference to ras_manager Check ras_manager before using it

CVE ID	Description
CVE-2024-43910	In the Linux kernel, the following vulnerability has been resolved: bpf: add missing check_func_arg_reg_off() to prevent out-of-bounds memory accesses Currently, it's possible to pass in a modified CONST_PTR_TO_DYNPTR to a global function as an argument. The adverse effects of this is that BPF helpers can continue to make use of this modified CONST_PTR_TO_DYNPTR from within the context of the global function, which can unintentionally result in out-of-bounds memory accesses and therefore compromise overall system stability i.e. [244.157771] BUG: KASAN: slab-out-of-bounds in bpf_dynptr_data+0x137/0x140 [244.161345] Read of size 8 at addr ffff88810914be68 by task test_progs/302 [244.167151] CPU: 0 PID: 302 Comm: test_progs Tainted: G O E 6.10.0-rc3-00131-g66b586715063 #533 [244.174318] Call Trace: [244.175787] <TASK> [244.177356] dump_stack_lvl+0x66/0xa0 [244.179531] print_report+0xce/0x670 [244.182314] ? __virt_addr_valid+0x200/0x3e0 [244.184908] kasan_report+0xd7/0x110 [244.187408] ? bpf_dynptr_data+0x137/0x140 [244.189714] ? bpf_dynptr_data+0x137/0x140 [244.192020] bpf_dynptr_data+0x137/0x140 [244.194264] bpf_prog_b02a02fdd2bdc5fa_global_call_bpf_dynptr_data+0x22/0x26 [244.198044] bpf_prog_b0fe7b9d7dc3abde_callback_adjust_bpf_dynptr_reg_off+0x1f/0x23 [244.202136] bpf_user_ringbuf_drain+0x2c7/0x570 [244.204744] ? 0xffffffffc0009e58 [244.206593] ? __pfx_bpf_user_ringbuf_drain+0x10/0x10 [244.209795] bpf_prog_33ab33f6a804ba2d_user_ringbuf_callback_const_ptr_to_dynptr_reg_off+0x47/0x4b [244.215922] bpf_trampoline_6442502480+0x43/0xe3 [244.218691] __x64_sys_prlimit64+0x9/0xf0 [244.220912] do_syscall_64+0xc1/0x1d0 [244.223043] entry_SYSCALL_64_after_hwframe+0x77/0x7f [244.226458] RIP: 0033:0x7ffa3eb8f059 [244.228582] Code: 08 89 e8 5b 5d c3 66 2e 0f 1f 84 00 00 00 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 8f 1d 0d 00 f7 d8 64 89 01 48 [244.241307] RSP: 002b:00007ffa3e9c6eb8 EFLAGS: 00000206 ORIG_RAX: 000000000000012e [244.246474] RAX: ffffffff8fda RBX: 00007ffa3e9c7cdc RCX: 00007ffa3eb8f059 [244.250478] RDX: 00007ffa3eb162b4 RSI: 0000000000000000 RDI: 00007ffa3e9c7fb0 [244.255396] RBP: 00007ffa3e9c6ed0 R08: 00007ffa3e9c76c0 R09: 0000000000000000 [244.260195] R10: 0000000000000000 R11: 0000000000000206 R12: ffffffff8f80 [244.264201] R13: 000000000000001c R14: 00007ffc5d6b4260 R15: 00007ffa3e1c7000 [244.268303] </TASK> Add a check_func_arg_reg_off() to the path in which the BPF verifier verifies the arguments of global function arguments, specifically those which take an argument of type ARG_PTR_TO_DYNPTR MEM_RDONLY. Also, process_dynptr_func() doesn't appear to perform any explicit and strict type matching on the supplied register type, so let's also enforce that a register either type PTR_TO_STACK or CONST_PTR_TO_DYNPTR is by the caller.

CVE ID	Description
CVE-2024-43911	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix NULL dereference at band check in starting tx ba session In MLD connection, link_data/link_conf are dynamically allocated. They don't point to vif->bss_conf. So, there will be no chanreq assigned to vif->bss_conf and then the chan will be NULL. Tweak the code to check ht_supported/vht_supported/has_he/has_eht on sta deflink. Crash log (with rtw89 version under MLO development): [9890.526087] BUG: kernel NULL pointer dereference, address: 0000000000000000 [9890.526102] #PF: supervisor read access in kernel mode [9890.526105] #PF: error_code(0x0000) - not-present page [9890.526109] PGD 0 P4D 0 [9890.526114] Oops: 0000 [#1] PREEMPT SMP PTI [9890.526119] CPU: 2 PID: 6367 Comm: kworker/u16:2 Kdump: loaded Tainted: G OE 6.9.0 #1 [9890.526123] Hardware name: LENOVO 2356AD1/2356AD1, BIOS G7ETB3WW (2.73) 11/28/2018 [9890.526126] Workqueue: phy2 rtw89_core_ba_work [rtw89_core] [9890.526203] RIP: 0010:ieee80211_start_tx_ba_session (net/mac80211/agg-tx.c:618 (discriminator 1)) mac80211 [9890.526279] Code: f7 e8 d5 93 3e ea 48 83 c4 28 89 d8 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc 49 8b 84 24 e0 f1 ff ff 48 8b 80 90 1b 00 00 <83> 38 03 0f 84 37 fe ff ff bb ea ff ff ff eb cc 49 8b 84 24 10 f3 All code ===== 0: f7 e8 imul %eax 2: d5 (bad) 3: 93 xchg %eax,%ebx 4: 3e ea ds (bad) 6: 48 83 c4 28 add \$0x28, %rsp a: 89 d8 mov %ebx,%eax c: 5b pop %rbx d: 41 5c pop %r12 f: 41 5d pop %r13 11: 41 5e pop %r14 13: 41 5f pop %r15 15: 5d pop %rbp 16: c3 retq 17: cc int3 18: cc int3 19: cc int3 1a: cc int3 1b: 49 8b 84 24 e0 f1 ff mov -0xe20(%r12),%rax 22: ff 23: 48 8b 80 90 1b 00 00 mov 0x1b90(%rax),%rax 2a:* 83 38 03 cmpl \$0x3,(%rax) <-- trapping instruction 2d: 0f 84 37 fe ff ff je 0xfffffffffffe6a 33: bb ea ff ff ff mov \$0xffffffe6,%ebx 38: eb cc jmp 0x6 3a: 49 rex.WB 3b: 8b .byte 0x8b 3c: 84 24 10 test %ah,(%rax,%rdx,1) 3f: f3 repz Code starting with the faulting instruction ===== 0: 83 38 03 cmpl \$0x3,(%rax) 3: 0f 84 37 fe ff ff je 0xfffffffffffe40 9: bb ea ff ff ff mov \$0xffffffe6,%ebx e: eb cc jmp 0xfffffffffffe6a 10: 49 rex.WB 11: 8b .byte 0x8b 12: 84 24 10 test %ah,(%rax,%rdx,1) 15: f3 repz [9890.526285] RSP: 0018:ffff8db09013d68 EFLAGS: 00010246 [9890.526291] RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffff9308e0d656c8 [9890.526295] RDX: 0000000000000000 RSI: ffffffffab99460b RDI: ffffffffab9a7685 [9890.526300] RBP: fffff8db09013db8 R08: 0000000000000000 R09: 00000000000000873 [9890.526304] R10: ffff9308e0d64800 R11: 0000000000000000 R12: ffff9308e5ff6e70 [9890.526308] R13: ffff930952500e20 R14: ffff9309192a8c00 R15: 0000000000000000 [9890.526313] FS: 0000000000000000(0000) GS:ffff930b4e700000(0000) klgS:0000000000000000 [9890.526316] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [9890.526318] CR2: 0000000000000000 CR3: 0000000391c58005 CR4: 00000000001706f0 [9890.526321] Call Trace: [9890.526324] <TASK> [9890.526327] ? show_regs (arch/x86/kernel/dumpstack.c:479) [9890.526335] ? __die (arch/x86/kernel/dumpstack.c:421 arch/x86/kernel/dumpstack.c:434) [9890.526340] ? page_fault_oops (arch/x86/mm/fault.c:713) [9890.526347] ? search_module_extables (kernel/module/main.c:3256 (discriminator ---truncated---
CVE-2024-43912	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: disallow setting special AP channel widths Setting the AP channel width is meant for use with the normal 20/40/... MHz channel width progression, and switching around in S1G or narrow channels isn't supported. Disallow that.
CVE-2024-43914	In the Linux kernel, the following vulnerability has been resolved: md/raid5: avoid BUG_ON() while continue reshape after reassembling Currently, mdadm support --revert-reshape to abort the reshape while reassembling, as the test 07revert-grow. However, following BUG_ON() can be triggered by the test: kernel BUG at drivers/md/raid5.c:6278! invalid opcode: 0000 [#1] PREEMPT SMP PTI irq event stamp: 158985 CPU: 6 PID: 891 Comm: md0_reshape Not tainted 6.9.0-03335-g7592a0b0049a #94 RIP: 0010:reshape_request+0x3f1/0xe60 Call Trace: <TASK> raid5_sync_request+0x43d/0x550 md_do_sync+0xb7a/0x2110 md_thread+0x294/0x2b0 kthread+0x147/0x1c0 ret_from_fork+0x59/0x70 ret_from_fork_asm+0x1a/0x30 </TASK> Root cause is that --revert-reshape update the raid_disks from 5 to 4, while reshape position is still set, and after reassembling the array, reshape position will be read from super block, then during reshape the checking of 'writepos' that is caculated by old reshape position will fail. Fix this panic the easy way first, by converting the BUG_ON() to WARN_ON(), and stop the reshape if checkings fail. Noted that mdadm must fix --revert-shape as well, and probably md/raid should enhance metadata validation as well, however this means reassemble will fail and there must be user tools to fix the wrong metadata.
CVE-2024-44931	In the Linux kernel, the following vulnerability has been resolved: gpio: prevent potential speculation leaks in gpio_device_get_desc() Userspace may trigger a speculative read of an address outside the gpio descriptor array. Users can do that by calling gpio_ioctl() with an offset out of range. Offset is copied from user and then used as an array index to get the gpio descriptor without sanitization in gpio_device_get_desc(). This change ensures that the offset is sanitized by using array_index_nospec() to mitigate any possibility of speculative information leaks. This bug was discovered and resolved using Coverity Static Analysis Security Testing (SAST) by Synopsys, Inc.

CVE ID	Description
CVE-2024-44932	In the Linux kernel, the following vulnerability has been resolved: idpf: fix UAFs when destroying the queues The second tagged commit started sometimes (very rarely, but possible) throwing WARNs from net/core/page_pool.c:page_pool_disable_direct_recycling(). Turned out idpf frees interrupt vectors with embedded NAPIs *before* freeing the queues making page_pools' NAPI pointers lead to freed memory before these pools are destroyed by libeth. It's not clear whether there are other accesses to the freed vectors when destroying the queues, but anyway, we usually free queue/interrupt vectors only when the queues are destroyed and the NAPIs are guaranteed to not be referenced anywhere. Invert the allocation and freeing logic making queue/interrupt vectors be allocated first and freed last. Vectors don't require queues to be present, so this is safe. Additionally, this change allows to remove that useless queue->q_vector pointer cleanup, as vectors are still valid when freeing the queues (+ both are freed within one function, so it's not clear why nullify the pointers at all).
CVE-2024-44934	In the Linux kernel, the following vulnerability has been resolved: net: bridge: mcast: wait for previous gc cycles when removing port syzbot hit a use-after-free[1] which is caused because the bridge doesn't make sure that all previous garbage has been collected when removing a port. What happens is: CPU 1 CPU 2 start gc cycle remove port acquire gc lock first wait for lock call br_multicast_gc() directly acquire lock now but free port the port can be freed while grp timers still running Make sure all previous gc cycles have finished by using flush_work before freeing the port. [1] BUG: KASAN: slab-use-after-free in br_multicast_port_group_expired+0x4c0/0x550 net/bridge/br_multicast.c:861 Read of size 8 at addr ffff888071d6d000 by task syz.5.1232/9699 CPU: 1 PID: 9699 Comm: syz.5.1232 Not tainted 6.10.0-rc5-syzkaller-00021-g24ca36a562d6 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 06/07/2024 Call Trace: <IRQ> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:114 print_address_description mm/kasan/report.c:377 [inline] print_report+0xc3/0x620 mm/kasan/report.c:488 kasan_report+0xd9/0x110 mm/kasan/report.c:601 br_multicast_port_group_expired+0x4c0/0x550 net/bridge/br_multicast.c:861 call_timer_fn+0x1a3/0x610 kernel/time/timer.c:1792 expire_timers kernel/time/timer.c:1843 [inline] __run_timers+0x74b/0xaf0 kernel/time/timer.c:2417 __run_timer_base kernel/time/timer.c:2428 [inline] __run_timer_base kernel/time/timer.c:2421 [inline] run_timer_base+0x111/0x190 kernel/time/timer.c:2437
CVE-2024-44944	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: use helper function to calculate expect ID Delete expectation path is missing a call to the nf_expect_get_id() helper function to calculate the expectation ID, otherwise LSB of the expectation object address is leaked to userspace.
CVE-2024-44947	In the Linux kernel, the following vulnerability has been resolved: fuse: Initialize beyond-EOF page contents before setting uptodate fuse_notify_store(), unlike fuse_do_readpage(), does not enable page zeroing (because it can be used to change partial page contents). So fuse_notify_store() must be more careful to fully initialize page contents (including parts of the page that are beyond end-of-file) before marking the page uptodate. The current code can leave beyond-EOF page contents uninitialized, which makes these uninitialized page contents visible to userspace via mmap(). This is an information leak, but only affects systems which do not enable init-on-alloc (via CONFIG_INIT_ON_ALLOC_DEFAULT_ON=y or the corresponding kernel command line parameter).
CVE-2024-44951	In the Linux kernel, the following vulnerability has been resolved: serial: sc16is7xx: fix TX fifo corruption Sometimes, when a packet is received on channel A at almost the same time as a packet is about to be transmitted on channel B, we observe with a logic analyzer that the received packet on channel A is transmitted on channel B. In other words, the Tx buffer data on channel B is corrupted with data from channel A. The problem appeared since commit 4409df5866b7 ("serial: sc16is7xx: change EFR lock to operate on each channels"), which changed the EFR locking to operate on each channel instead of chip-wise. This commit has introduced a regression, because the EFR lock is used not only to protect the EFR registers access, but also, in a very obscure and undocumented way, to protect access to the data buffer, which is shared by the Tx and Rx handlers, but also by each channel of the IC. Fix this regression first by switching to kfifo_out_linear_ptr() in sc16is7xx_handle_tx() to eliminate the need for a shared Rx/Tx buffer. Secondly, replace the chip-wise Rx buffer with a separate Rx buffer for each channel.
CVE-2024-44954	In the Linux kernel, the following vulnerability has been resolved: ALSA: line6: Fix racy access to midibuf There can be concurrent accesses to line6 midibuf from both the URB completion callback and the rawmidi API access. This could be a cause of KMSAN warning triggered by syzkaller below (so put as reported-by here). This patch protects the midibuf call of the former code path with a spinlock for avoiding the possible races.
CVE-2024-44957	In the Linux kernel, the following vulnerability has been resolved: xen: privcmd: Switch from mutex to spinlock for irqfds irqfd_wakeup() gets EPOLLHUP, when it is called by eventfd_release() by way of wake_up_poll(&ctx->wqh, EPOLLHUP), which gets called under spin_lock_irqsave(). We can't use a mutex here as it will lead to a deadlock. Fix it by switching over to a spin lock.

CVE ID	Description
CVE-2024-44958	In the Linux kernel, the following vulnerability has been resolved: sched/smt: Fix unbalance sched_smt_present dec/inc I got the following warn report while doing stress test: jump label: negative count! WARNING: CPU: 3 PID: 38 at kernel/jump_label.c:263 static_key_slow_try_dec +0x9d/0xb0 Call Trace: <TASK> __static_key_slow_dec_cpuslocked+0x16/0x70 sched_cpu_deactivate +0x26e/0x2a0 cpuhp_invoke_callback+0x3ad/0x10d0 cpuhp_thread_fun+0x3f5/0x680 smpboot_thread_fn+0x56d/0x8d0 kthread+0x309/0x400 ret_from_fork+0x41/0x70 ret_from_fork_asm +0x1b/0x30 </TASK> Because when cpuset_cpu_inactive() fails in sched_cpu_deactivate(), the cpu offline failed, but sched_smt_present is decremented before calling sched_cpu_deactivate(), it leads to unbalanced dec/inc, so fix it by incrementing sched_smt_present in the error path.
CVE-2024-44961	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Forward soft recovery errors to userspace As we discussed before[1], soft recovery should be forwarded to userspace, or we can get into a really bad state where apps will keep submitting hanging command buffers cascading us to a hard reset. 1: https://lore.kernel.org/all/bf23d5ed-9a6b-43e7-84ee-8cbfd0d60f18@froggi.es/ (cherry picked from commit 434967aadbbe3ad9103cc29e9a327de20fdbba01)
CVE-2024-44964	In the Linux kernel, the following vulnerability has been resolved: idpf: fix memory leaks and crashes while performing a soft reset The second tagged commit introduced a UAF, as it removed restoring q_vector->vport pointers after reinitializing the structures. This is due to that all queue allocation functions are performed here with the new temporary vport structure and those functions rewrite the backpointers to the vport. Then, this new struct is freed and the pointers start leading to nowhere. But generally speaking, the current logic is very fragile. It claims to be more reliable when the system is low on memory, but in fact, it consumes two times more memory as at the moment of running this function, there are two vports allocated with their queues and vectors. Moreover, it claims to prevent the driver from running into "bad state", but in fact, any error during the rebuild leaves the old vport in the partially allocated state. Finally, if the interface is down when the function is called, it always allocates a new queue set, but when the user decides to enable the interface later on, vport_open() allocates them once again, IOW there's a clear memory leak here. Just don't allocate a new queue set when performing a reset, that solves crashes and memory leaks. Readd the old queue number and reopen the interface on rollback - that solves limbo states when the device is left disabled and/or without HW queues enabled.
CVE-2024-44967	In the Linux kernel, the following vulnerability has been resolved: drm/mgag200: Bind I2C lifetime to DRM device Managed cleanup with devm_add_action_or_reset() will release the I2C adapter when the underlying Linux device goes away. But the connector still refers to it, so this cleanup leaves behind a stale pointer in struct drm_connector.ddc. Bind the lifetime of the I2C adapter to the connector's lifetime by using DRM's managed release. When the DRM device goes away (after the Linux device) DRM will first clean up the connector and then clean up the I2C adapter.
CVE-2024-44974	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: avoid possible UaF when selecting endp select_local_address() and select_signal_address() both select an endpoint entry from the list inside an RCU protected section, but return a reference to it, to be read later on. If the entry is dereferenced after the RCU unlock, reading info could cause a Use-after-Free. A simple solution is to copy the required info while inside the RCU protected section to avoid any risk of UaF later. The address ID might need to be modified later to handle the ID0 case later, so a copy seems OK to deal with.
CVE-2024-44977	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Validate TA binary size Add TA binary size validation to avoid OOB write. (cherry picked from commit c0a04e3570d72aaf090962156ad085e37c62e442)

CVE ID	Description
CVE-2024-44987	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent UAF in ip6_send_skb() syzbot reported an UAF in ip6_send_skb() [1] After ip6_local_out() has returned, we no longer can safely dereference rt, unless we hold rcu_read_lock(). A similar issue has been fixed in commit a688caa34beb ("ipv6: take rcu lock in rawv6_send_hdrinc()") Another potential issue in ip6_finish_output2() is handled in a separate patch. [1] BUG: KASAN: slab-use-after-free in ip6_send_skb+0x18d/0x230 net/ipv6/ip6_output.c:1964 Read of size 8 at addr ffff88806dde4858 by task syz.1.380/6530 CPU: 1 UID: 0 PID: 6530 Comm: syz.1.380 Not tainted 6.11.0-rc3-syzkaller-00306-gdf6cbc62cc9b #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 ip6_send_skb+0x18d/0x230 net/ipv6/ip6_output.c:1964 rawv6_push_pending_frames+0x75c/0x9e0 net/ipv6/raw.c:588 rawv6_sendmsg+0x19c7/0x23c0 net/ipv6/raw.c:926 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x1a6/0x270 net/socket.c:745 sock_write_iter+0x2dd/0x400 net/socket.c:1160 do_iter_readv_writev+0x60a/0x890 vfs_writev+0x37c/0xbb0 fs/read_write.c:971 do_writev+0x1b1/0x350 fs/read_write.c:1018 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f936bf79e79 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f936cd7f038 EFLAGS: 00000246 ORIG_RAX: 0000000000000014 RAX: ffffffff936c115f80 RBX: 00007f936c115f80 RCX: 00007f936bf79e79 RDX: 0000000000000001 RSI: 0000000020000040 RDI: 0000000000000004 RBP: 00007f936bf7916 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000000 R14: 00007f936c115f80 R15: 00007fff2860a7a8 </TASK> Allocated by task 6530: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 unpoison_slab_object mm/kasan/common.c:312 [inline] __kasan_slab_alloc+0x66/0x80 mm/kasan/common.c:338 kasan_slab_alloc include/linux/kasan.h:201 [inline] slab_post_alloc_hook mm/slab.c:3988 [inline] slab_alloc_node mm/slab.c:4037 [inline] kmem_cache_alloc_noprof+0x135/0x2a0 mm/slab.c:4044 dst_alloc+0x12b/0x190 net/core/dst.c:89 ip6_blackhole_route+0x59/0x340 net/ipv6/route.c:2670 make_blackhole net/xfrm/xfrm_policy.c:3120 [inline] xfrm_lookup_route+0xd1/0x1c0 net/xfrm/xfrm_policy.c:3313 ip6_dst_lookup_flow+0x13e/0x180 net/ipv6/ip6_output.c:1257 rawv6_sendmsg+0x1283/0x23c0 net/ipv6/raw.c:898 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x1a6/0x270 net/socket.c:745 __sys_sendmsg+0x525/0x7d0 net/socket.c:2597 __sys_sendmsg net/socket.c:2651 [inline] __sys_sendmsg+0x2b0/0x3a0 net/socket.c:2680 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 45: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:579 poison_slab_object+0xe0/0x150 mm/kasan/common.c:240 __kasan_slab_free+0x37/0x60 mm/kasan/common.c:256 kasan_slab_free include/linux/kasan.h:184 [inline] slab_free_hook mm/slab.c:2252 [inline] slab_free mm/slab.c:4473 [inline] kmem_cache_free+0x145/0x350 mm/slab.c:4548 dst_destroy+0x2ac/0x460 net/core/dst.c:124 rcu_do_batch kernel/rcu/tree.c:2569 [inline] rcu_core+0xafd/0x1830 kernel/rcu/tree. ---truncated---
CVE-2024-44998	In the Linux kernel, the following vulnerability has been resolved: atm: idt77252: prevent use after free in dequeue_rx() We can't dereference "skb" after calling vcc->push() because the skb is released.
CVE-2024-45001	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix RX buf alloc_size alignment and atomic op panic The MANA driver's RX buffer alloc_size is passed into napi_build_skb() to create SKB. skb_shinfo(skb) is located at the end of skb, and its alignment is affected by the alloc_size passed into napi_build_skb(). The size needs to be aligned properly for better performance and atomic operations. Otherwise, on ARM64 CPU, for certain MTU settings like 4000, atomic operations may panic on the skb_shinfo(skb)->dataref due to alignment fault. To fix this bug, add proper alignment to the alloc_size calculation. Sample panic info: [253.298819] Unable to handle kernel paging request at virtual address ffff000129ba5cce [253.300900] Mem abort info: [253.301760] ESR = 0x0000000096000021 [253.302825] EC = 0x25: DABT (current EL), IL = 32 bits [253.304268] SET = 0, FnV = 0 [253.305172] EA = 0, S1PTW = 0 [253.306103] FSC = 0x21: alignment fault Call trace: __skb_clone+0xfc/0x198 skb_clone+0x78/0xe0 raw6_local_deliver+0xfc/0x228 ip6_protocol_deliver_rcu+0x80/0x500 ip6_input_finish+0x48/0x80 ip6_input+0x48/0xc0 ip6_sublist_rcv_finish+0x50/0x78 ip6_sublist_rcv+0x1cc/0x2b8 ipv6_list_rcv+0x100/0x150 __netif_receive_skb_list_core+0x180/0x220 netif_receive_skb_list_internal+0x198/0x2a8 __napi_poll+0x138/0x250 net_rx_action+0x148/0x330 handle_softirqs+0x12c/0x3a0

CVE ID	Description
CVE-2024-45003	In the Linux kernel, the following vulnerability has been resolved: vfs: Don't evict inode under the inode lru traversing context The inode reclaiming process(See function prune_icache_sb) collects all reclaimable inodes and mark them with I_FREEING flag at first, at that time, other processes will be stuck if they try getting these inodes (See function find_inode_fast), then the reclaiming process destroy the inodes by function dispose_list(). Some filesystems(eg. ext4 with ea_inode feature, ubifs with xattr) may do inode lookup in the inode evicting callback function, if the inode lookup is operated under the inode lru traversing context, deadlock problems may happen. Case 1: In function ext4_evict_inode(), the ea inode lookup could happen if ea_inode feature is enabled, the lookup process will be stuck under the evicting context like this: 1. File A has inode i_reg and an ea inode i_ea 2. getfattr(A, xattr_buf) // i_ea is added into lru // lru->i_ea 3. Then, following three processes running like this: PA PB echo 2 > /proc/sys/vm/drop_caches shrink_slab prune_dcache_sb // i_reg is added into lru, lru->i_ea->i_reg prune_icache_sb list_lru_walk_one inode_lru_isolate i_ea->i_state = I_FREEING // set inode state inode_lru_isolate __iget(i_reg) spin_unlock(&i_reg->i_lock) spin_unlock(lru_lock) rm file A i_reg->nlink = 0 iput(i_reg) // i_reg->nlink is 0, do evict ext4_evict_inode ext4_xattr_delete_inode ext4_xattr_inode_dec_ref_all ext4_xattr_inode_iget ext4_iget(i_ea->i_ino) iget_locked find_inode_fast __wait_on_freeing_inode(i_ea) ----# AA deadlock dispose_list // cannot be executed by prune_icache_sb wake_up_bit(&i_ea->i_state) Case 2: In deleted inode writing function ubifs_jnl_write_inode(), file deleting process holds BASEHD's wbuf->io_mutex while getting the xattr inode, which could race with inode reclaiming process(The reclaiming process could try locking BASEHD's wbuf->io_mutex in inode evicting function), then an ABBA deadlock problem would happen as following: 1. File A has inode ia and a xattr(with inode ixa), regular file B has inode ib and a xattr. 2. getfattr(A, xattr_buf) // ixa is added into lru // lru->ixa 3. Then, following three processes running like this: PA PB PC echo 2 > /proc/sys/vm/drop_caches shrink_slab prune_dcache_sb // ib and ia are added into lru, lru->ixa->ib->ia prune_icache_sb list_lru_walk_one inode_lru_isolate ixa->i_state = I_FREEING // set inode state inode_lru_isolate __iget(ib) spin_unlock(&ib->i_lock) spin_unlock(lru_lock) rm file B ib->nlink = 0 rm file A iput(ia) ubifs_evict_inode(ia) ubifs_jnl_delete_inode(ia) ubifs_jnl_write_inode(ia) make_reservation(BASEHD) // Lock wbuf->io_mutex ubifs_iget(ixa->i_ino) iget_locked find_inode_fast __wait_on_freeing_inode(ixa) iput(ib) // ib->nlink is 0, do evict ubifs_evict_inode ubifs_jnl_delete_inode(ib) # ubifs_jnl_write_inode ABBA deadlock #----make_reservation(BASEHD) dispose_list // cannot be executed by prune_icache_sb wake_up_bit(&ixa->i_state) Fix the possible deadlock by using new inode state flag I_LRU_ISOLATING to pin the inode in memory while inode_lru_isolate(---truncated---

CVE ID	Description
CVE-2024-45005	In the Linux kernel, the following vulnerability has been resolved: KVM: s390: fix validity interception issue when gisa is switched off We might run into a SIE validity if gisa has been disabled either via using kernel parameter "kvm.use_gisa=0" or by setting the related sysfs attribute to N (echo N >/sys/module/kvm/parameters/use_gisa). The validity is caused by an invalid value in the SIE control block's gisa designation. That happens because we pass the uninitialized gisa origin to virt_to_phys() before writing it to the gisa designation. To fix this we return 0 in kvm_s390_get_gisa_desc() if the origin is 0. kvm_s390_get_gisa_desc() is used to determine which gisa designation to set in the SIE control block. A value of 0 in the gisa designation disables gisa usage. The issue surfaces in the host kernel with the following kernel message as soon as a new kvm guest start is attempted. kvm: unhandled validity intercept 0x1011 WARNING: CPU: 0 PID: 781237 at arch/s390/kvm/intercept.c:101 kvm_handle_sie_intercept+0x42e/0x4d0 [kvm] Modules linked in: vhost_net tap tun xt_CHECKSUM xt_MASQUERADE xt_contrack ipt_REJECT xt_tcpudp nft_compat x_tables nf_nat_tftp nf_contrack_tftp vfio_pci_core irqbypass vhost_vsock vmw_vsock_virtio_transport_common vsock vhost vhost_iotlb kvm nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat nf_nat nf_contrack nf_defrag_ipv6 nf_defrag_ipv4 ip_set nf_tables sunrpc mlx5_ib ib_uverbs ib_core mlx5_core uvdevice s390_trng eadm_sch vfio_ccw zcrypt_cex4 mdev vfio_iommu_type1 vfio_sch_fq_codel drm_i2c_core loop drm_panel_orientation_quirks configfs nfnetlink lcs ctm fsm dm_service_time ghash_s390 prng chacha_s390 libchacha aes_s390 des_s390 libdes sha3_512_s390 sha3_256_s390 sha512_s390 sha256_s390 sha1_s390 sha_common dm_mirror dm_region_hash dm_log zfcpc scsi_transport_fc scsi_dh_rdac scsi_dh_emc scsi_dh_alua pkey zcrypt dm_multipath rng_core autofs4 [last unloaded: vfio_pci] CPU: 0 PID: 781237 Comm: CPU 0/KVM Not tainted 6.10.0-08682-gcad9f11498ea #6 Hardware name: IBM 3931 A01 701 (LPAR) Krln PSW : 0704c00180000000 000003d93deb0122 (kvm_handle_sie_intercept+0x432/0x4d0 [kvm]) R:0 T:1 IO:1 EX:1 Key:0 M:1 W:0 P:0 AS:3 CC:0 PM:0 RI:0 EA:3 Krln GPRS: 000003d900000027 000003d900000023 0000000000000028 000002cd00000000 000002d063a00900 00000359c6daf708 000000000000bebb5 0000000000001eff 000002cfd82e9000 000002cfd80bc000 0000000000001011 000003d93deda412 000003ff8962df98 000003d93de77ce0 000003d93deb011e 00000359c6daf960 Krln Code: 000003d93deb0112: c020fffe7259 larl %r2,000003d93de7e5c4 000003d93deb0118: c0e53fa8beac brasl %r14,000003d9bd3c7e70 #000003d93deb011e: af000000 mc 0,0 >000003d93deb0122: a728ffea lhi %r2,-22 000003d93deb0126: a7f4fe24 brc 15,000003d93deafdb6 000003d93deb012a: 9101f0b0 tm 176(%r15),1 000003d93deb012e: a774fe48 brc 7,000003d93deafdb6 000003d93deb0132: 40a0f0ae sth %r10,174(%r15) Call Trace: [<000003d93deb0122>] kvm_handle_sie_intercept +0x432/0x4d0 [kvm] [<000003d93deb011e>] kvm_handle_sie_intercept+0x42e/0x4d0 [kvm] [<000003d93deacc10>] vcpu_post_run+0x1d0/0x3b0 [kvm] [<000003d93deaceda>] __vcpu_run +0xea/0x2d0 [kvm] [<000003d93dead9da>] kvm_arch_vcpu_ioctl_run+0x16a/0x430 [kvm] [<000003d93de93ee0>] kvm_vcpu_ioctl+0x190/0x7c0 [kvm] [<000003d9bd728b4e>] vfs_ioctl +0x2e/0x70 [<000003d9bd72a092>] __s390x_sys_ioctl+0xc2/0xd0 [<000003d9be0e9222>] __do_syscall+0x1f2/0x2e0 [<000003d9be0f9a90>] system_call+0x70/0x98 Last Breaking-Event-Address: [<000003d9bd3c7f58>] __warn_printk+0xe8/0xf0
CVE-2024-45006	In the Linux kernel, the following vulnerability has been resolved: xhci: Fix Panther point NULL pointer deref at full-speed re-enumeration re-enumerating full-speed devices after a failed address device command can trigger a NULL pointer dereference. Full-speed devices may need to reconfigure the endpoint 0 Max Packet Size value during enumeration. Usb core calls usb_ep0_reinit() in this case, which ends up calling xhci_configure_endpoint(). On Panther point xHC the xhci_configure_endpoint() function will additionally check and reserve bandwidth in software. Other hosts do this in hardware If xHC address device command fails then a new xhci_virt_device structure is allocated as part of re-enabling the slot, but the bandwidth table pointers are not set up properly here. This triggers the NULL pointer dereference the next time usb_ep0_reinit() is called and xhci_configure_endpoint() tries to check and reserve bandwidth [46710.713538] usb 3-1: new full-speed USB device number 5 using xhci_hcd [46710.713699] usb 3-1: Device not responding to setup address. [46710.917684] usb 3-1: Device not responding to setup address. [46711.125536] usb 3-1: device not accepting address 5, error -71 [46711.125594] BUG: kernel NULL pointer dereference, address: 0000000000000008 [46711.125600] #PF: supervisor read access in kernel mode [46711.125603] #PF: error_code(0x0000) - not-present page [46711.125606] PGD 0 P4D 0 [46711.125610] Oops: Oops: 0000 [#1] PREEMPT SMP PTI [46711.125615] CPU: 1 PID: 25760 Comm: kworker/1:2 Not tainted 6.10.3_2 #1 [46711.125620] Hardware name: Gigabyte Technology Co., Ltd. [46711.125623] Workqueue: usb_hub_wq hub_event [usbcore] [46711.125668] RIP: 0010:xhci_reserve_bandwidth (drivers/usb/host/xhci.c Fix this by making sure bandwidth table pointers are set up correctly after a failed address device command, and additionally by avoiding checking for bandwidth in cases like this where no actual endpoints are added or removed, i.e. only context for default control endpoint 0 is evaluated.
CVE-2024-45008	In the Linux kernel, the following vulnerability has been resolved: Input: MT - limit max slots syzbot is reporting too large allocation at input_mt_init_slots(), for num_slots is supplied from userspace using ioctl(UI_DEV_CREATE). Since nobody knows possible max slots, this patch chose 1024.

CVE ID	Description
CVE-2024-45009	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: only decrement add_addr_accepted for MPJ req Adding the following warning ... WARN_ON_ONCE(msk->pm.add_addr_accepted == 0) ... before decrementing the add_addr_accepted counter helped to find a bug when running the "remove single subflow" subtest from the mptcp_join.sh selftest. Removing a 'subflow' endpoint will first trigger a RM_ADDR, then the subflow closure. Before this patch, and upon the reception of the RM_ADDR, the other peer will then try to decrement this add_addr_accepted. That's not correct because the attached subflows have not been created upon the reception of an ADD_ADDR. A way to solve that is to decrement the counter only if the attached subflow was an MP_JOIN to a remote id that was not 0, and initiated by the host receiving the RM_ADDR.
CVE-2024-45010	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: only mark 'subflow' endp as available Adding the following warning ... WARN_ON_ONCE(msk->pm.local_addr_used == 0) ... before decrementing the local_addr_used counter helped to find a bug when running the "remove single address" subtest from the mptcp_join.sh selftests. Removing a 'signal' endpoint will trigger the removal of all subflows linked to this endpoint via mptcp_pm_nl_rm_addr_or_subflow() with rm_type == MPTCP_MIB_RMSUBFLOW. This will decrement the local_addr_used counter, which is wrong in this case because this counter is linked to 'subflow' endpoints, and here it is a 'signal' endpoint that is being removed. Now, the counter is decremented, only if the ID is being used outside of mptcp_pm_nl_rm_addr_or_subflow(), only for 'subflow' endpoints, and if the ID is not 0 -- local_addr_used is not taking into account these ones. This marking of the ID as being available, and the decrement is done no matter if a subflow using this ID is currently available, because the subflow could have been closed before.
CVE-2024-45012	In the Linux kernel, the following vulnerability has been resolved: nouveau/firmware: use dma non-coherent allocator Currently, enabling SG_DEBUG in the kernel will cause nouveau to hit a BUG() on startup, when the iommu is enabled: kernel BUG at include/linux/scatterlist.h:187! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 7 PID: 930 Comm: (udev-worker) Not tainted 6.9.0-rc3Lyude-Test+ #30 Hardware name: MSI MS-7A39/A320M GAMING PRO (MS-7A39), BIOS 1.10 01/22/2019 RIP: 0010:sg_init_one+0x85/0xa0 Code: 69 88 32 01 83 e1 03 f6 c3 03 75 20 a8 01 75 1e 48 09 cb 41 89 54 24 08 49 89 1c 24 41 89 6c 24 0c 5b 5d 41 5c e9 7b b9 88 00 <0f> 0b 0f 0b 0f 0b 48 8b 05 5e 46 9a 01 eb b2 66 66 2e 0f 1f 84 00 RSP: 0018:ffffa776017bf6a0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffffa77600d87000 RCX: 000000000000002b RDX: 0000000000000001 RSI: 0000000000000000 RDI: ffffa77680d87000 RBP: 000000000000e000 R08: 0000000000000000 R09: 0000000000000000 R10: ffff98f4c46aa508 R11: 0000000000000000 R12: ffff98f4c46aa508 R13: ffff98f4c46aa008 R14: ffffa77600d4a000 R15: ffffa77600d4a018 FS: 00007feeb5aae980(0000) GS:ffff98f5c4dc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f22cb9a4520 CR3: 00000001043ba000 CR4: 0000000003506f0 Call Trace: <TASK> ? die+0x36/0x90 ? do_trap+0xdd/0x100 ? sg_init_one+0x85/0xa0 ? do_error_trap+0x65/0x80 ? sg_init_one+0x85/0xa0 ? exc_invalid_op+0x50/0x70 ? sg_init_one+0x85/0xa0 ? asm_exc_invalid_op+0x1a/0x20 ? sg_init_one+0x85/0xa0 nvkm_firmware_ctor+0x14a/0x250 [nouveau] nvkm_falcon_fw_ctor+0x42/0x70 [nouveau] ga102_gsp_booter_ctor+0xb4/0x1a0 [nouveau] r535_gsp_oneinit+0xb3/0x15f0 [nouveau] ? srso_return_thunk+0x5/0x5f ? srso_return_thunk+0x5/0x5f ? nvkm_udevice_new+0x95/0x140 [nouveau] ? srso_return_thunk+0x5/0x5f ? srso_return_thunk+0x5/0x5f ? ktime_get+0x47/0xb0 Fix this by using the non-coherent allocator instead, I think there might be a better answer to this, but it involve ripping up some of APIs using sg lists.
CVE-2024-45016	In the Linux kernel, the following vulnerability has been resolved: netem: fix return value if duplicate enqueue fails There is a bug in netem_enqueue() introduced by commit 5845f706388a ("net: netem: fix skb length BUG_ON in __skb_to_sgvec") that can lead to a use-after-free. This commit made netem_enqueue() always return NET_XMIT_SUCCESS when a packet is duplicated, which can cause the parent qdisc's q.qlen to be mistakenly incremented. When this happens qlen_notify() may be skipped on the parent during destruction, leaving a dangling pointer for some classful qdiscs like DRR. There are two ways for the bug happen: - If the duplicated packet is dropped by rootq->enqueue() and then the original packet is also dropped. - If rootq->enqueue() sends the duplicated packet to a different qdisc and the original packet is dropped. In both cases NET_XMIT_SUCCESS is returned even though no packets are enqueued at the netem qdisc. The fix is to defer the enqueue of the duplicate packet until after the original packet has been guaranteed to return NET_XMIT_SUCCESS.
CVE-2024-45019	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Take state lock during tx timeout reporter mlx5e_safe_reopen_channels() requires the state lock taken. The referenced changed in the Fixes tag removed the lock to fix another issue. This patch adds it back but at a later point (when calling mlx5e_safe_reopen_channels()) to avoid the deadlock referenced in the Fixes tag.

CVE ID	Description
CVE-2024-45025	In the Linux kernel, the following vulnerability has been resolved: fix bitmap corruption on close_range() with CLOSE_RANGE_UNSHARE copy_fd_bitmaps(new, old, count) is expected to copy the first count/BITS_PER_LONG bits from old->full_fds_bits[] and fill the rest with zeroes. What it does is copying enough words (BITS_TO_LONGS(count/BITS_PER_LONG)), then memsets the rest. That works fine, *if* all bits past the cutoff point are clear. Otherwise we are risking garbage from the last word we'd copied. For most of the callers that is true - expand_fdtbl() has count equal to old->max_fds, so there's no open descriptors past count, let alone fully occupied words in ->open_fds[], which is what bits in ->full_fds_bits[] correspond to. The other caller (dup_fd()) passes sane_fdtbl_size(old_fdt, max_fds), which is the smallest multiple of BITS_PER_LONG that covers all opened descriptors below max_fds. In the common case (copying on fork()) max_fds is ~0U, so all opened descriptors will be below it and we are fine, by the same reasons why the call in expand_fdtbl() is safe. Unfortunately, there is a case where max_fds is less than that and where we might, indeed, end up with junk in ->full_fds_bits[] - close_range(from, to, CLOSE_RANGE_UNSHARE) with * descriptor table being currently shared * 'to' being above the current capacity of descriptor table * 'from' being just under some chunk of opened descriptors. In that case we end up with observably wrong behaviour - e.g. spawn a child with CLONE_FILES, get all descriptors in range 0..127 open, then close_range(64, ~0U, CLOSE_RANGE_UNSHARE) and watch dup(0) ending up with descriptor #128, despite #64 being observably not open. The minimally invasive fix would be to deal with that in dup_fd(). If this proves to add measurable overhead, we can go that way, but let's try to fix copy_fd_bitmaps() first. * new helper: bitmap_copy_and_expand(to, from, bits_to_copy, size). * make copy_fd_bitmaps() take the bitmap size in words, rather than bits; it's 'count' argument is always a multiple of BITS_PER_LONG, so we are not losing any information, and that way we can use the same helper for all three bitmaps - compiler will see that count is a multiple of BITS_PER_LONG for the large ones, so it'll generate plain memcpy()+memset(). Reproducer added to tools/testing/selftests/core/close_range_test.c
CVE-2024-45033	Insufficient Session Expiration vulnerability in Apache Airflow Fab Provider. This issue affects Apache Airflow Fab Provider: before 1.5.2. When user password has been changed with admin CLI, the sessions for that user have not been cleared, leading to insufficient session expiration, thus logged users could continue to be logged in even after the password was changed. This only happened when the password was changed with CLI. The problem does not happen in case change was done with webserver thus this is different from CVE-2023-40273 https://github.com/advisories/GHSA-pm87-24wq-r8w9 which was addressed in Apache-Airflow 2.7.0 Users are recommended to upgrade to version 1.5.2, which fixes the issue.
CVE-2024-45157	An issue was discovered in Mbed TLS before 2.28.9 and 3.x before 3.6.1, in which the user-selected algorithm is not used. Unlike previously documented, enabling MBEDTLS_PSA_HMAC_DRBG_MD_TYPE does not cause the PSA subsystem to use HMAC_DRBG: it uses HMAC_DRBG only when MBEDTLS_PSA_CRYPTO_EXTERNAL_RNG and MBEDTLS_CTR_DRBG_C are disabled.
CVE-2024-45231	An issue was discovered in Django v5.1.1, v5.0.9, and v4.2.16. The django.contrib.auth.forms.PasswordResetForm class, when used in a view implementing password reset flows, allows remote attackers to enumerate user e-mail addresses by sending password reset requests and observing the outcome (only when e-mail sending is consistently failing).
CVE-2024-45296	path-to-regexp turns path strings into a regular expressions. In certain cases, path-to-regexp will output a regular expression that can be exploited to cause poor performance. Because JavaScript is single threaded and regex matching runs on the main thread, poor performance will block the event loop and lead to a DoS. The bad regular expression is generated any time you have two parameters within a single segment, separated by something that is not a period (.). For users of 0.1, upgrade to 0.1.10. All other users should upgrade to 8.0.0.
CVE-2024-45306	Vim is an open source, command line text editor. Patch v9.1.0038 optimized how the cursor position is calculated and removed a loop, that verified that the cursor position always points inside a line and does not become invalid by pointing beyond the end of a line. Back then we assumed this loop is unnecessary. However, this change made it possible that the cursor position stays invalid and points beyond the end of a line, which would eventually cause a heap-buffer-overflow when trying to access the line pointer at the specified cursor position. It's not quite clear yet, what can lead to this situation that the cursor points to an invalid position. That's why patch v9.1.0707 does not include a test case. The only observed impact has been a program crash. This issue has been addressed in with the patch v9.1.0707. All users are advised to upgrade.
CVE-2024-45336	The HTTP client drops sensitive headers after following a cross-domain redirect. For example, a request to a.com/ containing an Authorization header which is redirected to b.com/ will not send that header to b.com. In the event that the client received a subsequent same-domain redirect, however, the sensitive headers would be restored. For example, a chain of redirects from a.com/, to b.com/1, and finally to b.com/2 would incorrectly send the Authorization header to b.com/2.

CVE ID	Description
CVE-2024-45337	Applications and libraries which misuse connection.serverAuthenticate (via callback field ServerConfig.PublicKeyCallback) may be susceptible to an authorization bypass. The documentation for ServerConfig.PublicKeyCallback says that "A call to this function does not guarantee that the key offered is in fact used to authenticate." Specifically, the SSH protocol allows clients to inquire about whether a public key is acceptable before proving control of the corresponding private key. PublicKeyCallback may be called with multiple keys, and the order in which the keys were provided cannot be used to infer which key the client successfully authenticated with, if any. Some applications, which store the key(s) passed to PublicKeyCallback (or derived information) and make security relevant determinations based on it once the connection is established, may make incorrect assumptions. For example, an attacker may send public keys A and B, and then authenticate with A. PublicKeyCallback would be called only twice, first with A and then with B. A vulnerable application may then make authorization decisions based on key B for which the attacker does not actually control the private key. Since this API is widely misused, as a partial mitigation golang.org/x/crypto/ssh@v0.31.0 enforces the property that, when successfully authenticating via public key, the last key passed to ServerConfig.PublicKeyCallback will be the key used to authenticate the connection. PublicKeyCallback will now be called multiple times with the same key, if necessary. Note that the client may still not control the last key passed to PublicKeyCallback if the connection is then authenticated with a different method, such as PasswordCallback, KeyboardInteractiveCallback, or NoClientAuth. Users should be using the Extensions field of the Permissions return value from the various authentication callbacks to record data associated with the authentication attempt instead of referencing external state. Once the connection is established the state corresponding to the successful authentication attempt can be retrieved via the ServerConn.Permissions field. Note that some third-party libraries misuse the Permissions type by sharing it across authentication attempts; users of third-party libraries should refer to the relevant projects for guidance.
CVE-2024-45338	An attacker can craft an input to the Parse functions that would be processed non-linearly with respect to its length, resulting in extremely slow parsing. This could cause a denial of service.
CVE-2024-45339	When logs are written to a widely-writable directory (the default), an unprivileged attacker may predict a privileged process's log file path and pre-create a symbolic link to a sensitive file in its place. When that privileged process runs, it will follow the planted symlink and overwrite that sensitive file. To fix that, glog now causes the program to exit (with status code 2) when it finds that the configured log file already exists.
CVE-2024-45341	A certificate with a URI which has a IPv6 address with a zone ID may incorrectly satisfy a URI name constraint that applies to the certificate chain. Certificates containing URIs are not permitted in the web PKI, so this only affects users of private PKIs which make use of URIs.
CVE-2024-45828	In the Linux kernel, the following vulnerability has been resolved: i3c: mipi-i3c-hci: Mask ring interrupts before ring stop request Bus cleanup path in DMA mode may trigger a RING_OP_STAT interrupt when the ring is being stopped. Depending on timing between ring stop request completion, interrupt handler removal and code execution this may lead to a NULL pointer dereference in hci_dma_irq_handler() if it gets to run after the io_data pointer is set to NULL in hci_dma_cleanup(). Prevent this by masking the ring interrupts before ring stop request.
CVE-2024-46672	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: cfg80211: Handle SSID based pmksa deletion wpa_supplicant 2.11 sends since 1efdba5fdc2c ("Handle PMKSA flush in the driver for SAE/OWE offload cases") SSID based PMKSA del commands. brcmfmac is not prepared and tries to dereference the NULL bssid and pmkid pointers in cfg80211_pmksa. PMKID_V3 operations support SSID based updates so copy the SSID.
CVE-2024-46673	In the Linux kernel, the following vulnerability has been resolved: scsi: aacraid: Fix double-free on probe failure aac_probe_one() calls hardware-specific init functions through the aac_driver_init pointer, all of which eventually call down to aac_init_adapter(). If aac_init_adapter() fails after allocating memory for aac_dev::queues, it frees the memory but does not clear that member. After the hardware-specific init function returns an error, aac_probe_one() goes down an error path that frees the memory pointed to by aac_dev::queues, resulting in a double-free.

CVE ID	Description
CVE-2024-46678	In the Linux kernel, the following vulnerability has been resolved: bonding: change ipsec_lock from spin lock to mutex In the cited commit, bond->ipsec_lock is added to protect ipsec_list, hence xdo_dev_state_add and xdo_dev_state_delete are called inside this lock. As ipsec_lock is a spin lock and such xfrmdev ops may sleep, "scheduling while atomic" will be triggered when changing bond's active slave. [101.055189] BUG: scheduling while atomic: bash/902/0x00000200 [101.055726] Modules linked in: [101.058211] CPU: 3 PID: 902 Comm: bash Not tainted 6.9.0-rc4+ #1 [101.058760] Hardware name: [101.059434] Call Trace: [101.059436] <TASK> [101.060873] dump_stack_lvl+0x51/0x60 [101.061275] __schedule_bug+0x4e/0x60 [101.061682] __schedule+0x612/0x7c0 [101.062078] ? __mod_timer+0x25c/0x370 [101.062486] schedule+0x25/0xd0 [101.062845] schedule_timeout+0x77/0xf0 [101.063265] ? asm_common_interrupt+0x22/0x40 [101.063724] ? __bpf_trace_itimer_state+0x10/0x10 [101.064215] __wait_for_common+0x87/0x190 [101.064648] ? usleep_range_state+0x90/0x90 [101.065091] cmd_exec+0x437/0xb20 [mlx5_core] [101.065569] mlx5_cmd_do+0x1e/0x40 [mlx5_core] [101.066051] mlx5_cmd_exec+0x18/0x30 [mlx5_core] [101.066552] mlx5_crypto_create_dek_key+0xea/0x120 [mlx5_core] [101.067163] ? bonding_sysfs_store_option+0x4d/0x80 [bonding] [101.067738] ? kmalloc_trace+0x4d/0x350 [101.068156] mlx5_ipsec_create_sa_ctx+0x33/0x100 [mlx5_core] [101.068747] mlx5e_xfrm_add_state+0x47b/0xaa0 [mlx5_core] [101.069312] bond_change_active_slave+0x392/0x900 [bonding] [101.069868] bond_option_active_slave_set+0x1c2/0x240 [bonding] [101.070454] __bond_opt_set+0xa6/0x430 [bonding] [101.070935] __bond_opt_set_notify+0x2f/0x90 [bonding] [101.071453] bond_opt_tryset_rtnl+0x72/0xb0 [bonding] [101.071965] bonding_sysfs_store_option+0x4d/0x80 [bonding] [101.072567] kernfs_fop_write_iter+0x10c/0x1a0 [101.073033] vfs_write+0x2d8/0x400 [101.073416] ? alloc_fd+0x48/0x180 [101.073798] ksys_write+0x5f/0xe0 [101.074175] do_syscall_64+0x52/0x110 [101.074576] entry_SYSCALL_64_after_hwframe+0x4b/0x53 As bond_ipsec_add_sa_all and bond_ipsec_del_sa_all are only called from bond_change_active_slave, which requires holding the RTNL lock. And bond_ipsec_add_sa and bond_ipsec_del_sa are xfrm state xdo_dev_state_add and xdo_dev_state_delete APIs, which are in user context. So ipsec_lock doesn't have to be spin lock, change it to mutex, and thus the above issue can be resolved.
CVE-2024-46681	In the Linux kernel, the following vulnerability has been resolved: pktgen: use cpus_read_lock() in pg_net_init() I have seen the WARN_ON(smp_processor_id() != cpu) firing in pktgen_thread_worker() during tests. We must use cpus_read_lock()/cpus_read_unlock() around the for_each_online_cpu(cpu) loop. While we are at it use WARN_ON_ONCE() to avoid a possible syslog flood.
CVE-2024-46689	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: cmd-db: Map shared memory as WC, not WB Linux does not write into cmd-db region. This region of memory is write protected by XPU. XPU may sometime falsely detect clean cache eviction as "write" into the write protected region leading to secure interrupt which causes an endless loop somewhere in Trust Zone. The only reason it is working right now is because Qualcomm Hypervisor maps the same region as Non-Cacheable memory in Stage 2 translation tables. The issue manifests if we want to use another hypervisor (like Xen or KVM), which does not know anything about those specific mappings. Changing the mapping of cmd-db memory from MEMREMAP_WB to MEMREMAP_WT/WC removes dependency on correct mappings in Stage 2 tables. This patch fixes the issue by updating the mapping to MEMREMAP_WC. I tested this on SA8155P with Xen.
CVE-2024-46707	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Make ICC_*SGI*_EL1 undef in the absence of a vGICv3 On a system with a GICv3, if a guest hasn't been configured with GICv3 and that the host is not capable of GICv2 emulation, a write to any of the ICC_*SGI*_EL1 registers is trapped to EL2. We therefore try to emulate the SGI access, only to hit a NULL pointer as no private interrupt is allocated (no GIC, remember?). The obvious fix is to give the guest what it deserves, in the shape of a UNDEF exception.
CVE-2024-46710	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: Prevent unmapping active read buffers The kms paths keep a persistent map active to read and compare the cursor buffer. These maps can race with each other in simple scenario where: a) buffer "a" mapped for update b) buffer "a" mapped for compare c) do the compare d) unmap "a" for compare e) update the cursor f) unmap "a" for update At step "e" the buffer has been unmapped and the read contents is bogus. Prevent unmapping of active read buffers by simply keeping a count of how many paths have currently active maps and unmap only when the count reaches 0.
CVE-2024-46713	In the Linux kernel, the following vulnerability has been resolved: perf/aux: Fix AUX buffer serialization Ole reported that event->mmap_mutex is strictly insufficient to serialize the AUX buffer, add a per RB mutex to fully serialize it. Note that in the lock order comment the perf_event::mmap_mutex order was already wrong, that is, it nesting under mmap_lock is not new with this patch.

CVE ID	Description
CVE-2024-46715	In the Linux kernel, the following vulnerability has been resolved: driver: iio: add missing checks on iio_info's callback access Some callbacks from iio_info structure are accessed without any check, so if a driver doesn't implement them trying to access the corresponding sysfs entries produce a kernel oops such as: [2203.527791] Unable to handle kernel NULL pointer dereference at virtual address 00000000 when execute [...] [2203.783416] Call trace: [2203.783429] iio_read_channel_info_avail from dev_attr_show+0x18/0x48 [2203.789807] dev_attr_show from sysfs_kf_seq_show+0x90/0x120 [2203.794181] sysfs_kf_seq_show from seq_read_iter+0xd0/0x4e4 [2203.798555] seq_read_iter from vfs_read+0x238/0x2a0 [2203.802236] vfs_read from ksys_read+0xa4/0xd4 [2203.805385] ksys_read from ret_fast_syscall+0x0/0x54 [2203.809135] Exception stack(0xe0badfa8 to 0xe0badff0) [2203.812880] dfa0: 00000003 b6f10f80 00000003 b6eab000 00020000 00000000 [2203.819746] dfc0: 00000003 b6f10f80 7ff00000 00000003 00000000 00020000 00000000 [2203.826619] dfe0: b6e1bc88 bed80958 b6e1bc94 b6e1bcb0 [2203.830363] Code: bad PC value [2203.832695] --- [end trace 0000000000000000]---
CVE-2024-46717	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: SHAMPO, Fix incorrect page release Under the following conditions: 1) No skb created yet 2) header_size == 0 (no SHAMPO header) 3) header_index + 1 % MLX5E_SHAMPO_WQ_HEADER_PER_PAGE == 0 (this is the last page fragment of a SHAMPO header page) a new skb is formed with a page that is NOT a SHAMPO header page (it is a regular data page). Further down in the same function (mlx5e_handle_rx_cqe_mpwq_shampo()), a SHAMPO header page from header_index is released. This is wrong and it leads to SHAMPO header pages being released more than once.
CVE-2024-46719	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Fix null pointer dereference in trace ucsi_register_altmode checks IS_ERR for the alt pointer and treats NULL as valid. When CONFIG_TYPEC_DP_ALTMODE is not enabled, ucsi_register_displayport returns NULL which causes a NULL pointer dereference in trace. Rather than return NULL, call typec_port_register_altmode to register DisplayPort alternate mode as a non-controllable mode when CONFIG_TYPEC_DP_ALTMODE is not enabled.
CVE-2024-46720	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix dereference after null check check the pointer hive before use.
CVE-2024-46722	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix mc_data out-of-bounds read warning Clear warning that read mc_data[i-1] may out-of-bounds.
CVE-2024-46723	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix ucode out-of-bounds read warning Clear warning that read ucode[] may out-of-bounds.
CVE-2024-46724	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix out-of-bounds read of df_v1_7_channel_number Check the fb_channel_number range to avoid the array out-of-bounds read error
CVE-2024-46725	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix out-of-bounds write warning Check the ring type value to fix the out-of-bounds write warning
CVE-2024-46737	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: fix kernel crash if commands allocation fails If the commands allocation fails in nvmet_tcp_alloc_cmds() the kernel crashes in nvmet_tcp_release_queue_work() because of a NULL pointer dereference. nvmet: failed to install queue 0 cntlid 1 ret 6 Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 Fix the bug by setting queue->nr_cmds to zero in case nvmet_tcp_alloc_cmd() fails.

CVE ID	Description
CVE-2024-46738	In the Linux kernel, the following vulnerability has been resolved: VMCI: Fix use-after-free when removing resource in vmci_resource_remove() When removing a resource from vmci_resource_table in vmci_resource_remove(), the search is performed using the resource handle by comparing context and resource fields. It is possible though to create two resources with different types but same handle (same context and resource fields). When trying to remove one of the resources, vmci_resource_remove() may not remove the intended one, but the object will still be freed as in the case of the datagram type in vmci_datagram_destroy_handle(). vmci_resource_table will still hold a pointer to this freed resource leading to a use-after-free vulnerability. BUG: KASAN: use-after-free in vmci_handle_is_equal include/linux/vmw_vmci_defs.h:142 [inline] BUG: KASAN: use-after-free in vmci_resource_remove +0x3a1/0x410 drivers/misc/vmw_vmci/vmci_resource.c:147 Read of size 4 at addr ffff88801c16d800 by task syz-executor197/1592 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x82/0xa9 lib/dump_stack.c:106 print_address_description.constprop.0+0x21/0x366 mm/kasan/report.c:239 __kasan_report.cold+0x7f/0x132 mm/kasan/report.c:425 kasan_report +0x38/0x51 mm/kasan/report.c:442 vmci_handle_is_equal include/linux/vmw_vmci_defs.h:142 [inline] vmci_resource_remove+0x3a1/0x410 drivers/misc/vmw_vmci/vmci_resource.c:147 vmci_qp_broker_detach+0x89a/0x11b9 drivers/misc/vmw_vmci/vmci_queue_pair.c:2182 ctx_free_ctx +0x473/0xbe1 drivers/misc/vmw_vmci/vmci_context.c:444 kref_put include/linux/kref.h:65 [inline] vmci_ctx_put drivers/misc/vmw_vmci/vmci_context.c:497 [inline] vmci_ctx_destroy +0x170/0x1d6 drivers/misc/vmw_vmci/vmci_context.c:195 vmci_host_close+0x125/0x1ac drivers/misc/vmw_vmci/vmci_host.c:143 __fput+0x261/0xa34 fs/file_table.c:282 task_work_run +0xf0/0x194 kernel/task_work.c:164 tracehook_notify_resume include/linux/tracehook.h:189 [inline] exit_to_user_mode_loop+0x184/0x189 kernel/entry/common.c:187 exit_to_user_mode_prepare +0x11b/0x123 kernel/entry/common.c:220 __syscall_exit_to_user_mode_work kernel/entry/common.c:302 [inline] syscall_exit_to_user_mode+0x18/0x42 kernel/entry/common.c:313 do_syscall_64+0x41/0x85 arch/x86/entry/common.c:86 entry_SYSCALL_64_after_hwframe+0x6e/0x0 This change ensures the type is also checked when removing the resource from vmci_resource_table in vmci_resource_remove().
CVE-2024-46739	In the Linux kernel, the following vulnerability has been resolved: uio_hv_generic: Fix kernel NULL pointer dereference in hv_uio_rescind For primary VM Bus channels, primary_channel pointer is always NULL. This pointer is valid only for the secondary channels. Also, rescind callback is meant for primary channels only. Fix NULL pointer dereference by retrieving the device_obj from the parent for the primary channel.
CVE-2024-46742	In the Linux kernel, the following vulnerability has been resolved: smb/server: fix potential null-ptr-deref of lease_ctx_info in smb2_open() null-ptr-deref will occur when (req_op_level == SMB2_OPLOCK_LEVEL_LEASE) and parse_lease_state() return NULL. Fix this by check if 'lease_ctx_info' is NULL. Additionally, remove the redundant parentheses in parse_durable_handle_context().

CVE ID	Description
CVE-2024-46743	In the Linux kernel, the following vulnerability has been resolved: of_irq: Prevent device address out-of-bounds read in interrupt map walk When of_irq_parse_raw() is invoked with a device address smaller than the interrupt parent node (from #address-cells property), KASAN detects the following out-of-bounds read when populating the initial match table (dyndbg="func of_irq_parse_* +p"): OF: of_irq_parse_one: dev=/soc@0/picasso/watchdog, index=0 OF: parent=/soc@0/pci@878000000000/gpio0@17,0, intsize=2 OF: intspec=4 OF: of_irq_parse_raw: ipar=/soc@0/pci@878000000000/gpio0@17,0, size=2 OF: -> addrsize=3 ===== BUG: KASAN: slab-out-of-bounds in of_irq_parse_raw+0x2b8/0x8d0 Read of size 4 at addr ffffff81beca5608 by task bash/764 CPU: 1 PID: 764 Comm: bash Tainted: G O 6.1.67-484c613561-nokia_sm_arm64 #1 Hardware name: Unknown Unknown Product/Unknown Product, BIOS 2023.01-12.24.03-dirty 01/01/2023 Call trace: dump_backtrace+0xdc/0x130 show_stack+0x1c/0x30 dump_stack_lvl+0x6c/0x84 print_report+0x150/0x448 kasan_report+0x98/0x140 __asan_load4+0x78/0xa0 of_irq_parse_raw+0x2b8/0x8d0 of_irq_parse_one+0x24c/0x270 parse_interrupts+0xc0/0x120 of_fwnode_add_links+0x100/0x2d0 fw_devlink_parse_fwtree+0x64/0xc0 device_add+0xb38/0xc30 of_device_add+0x64/0x90 of_platform_device_create_pdata+0xd0/0x170 of_platform_bus_create+0x244/0x600 of_platform_notify+0x1b0/0x254 blocking_notifier_call_chain+0x9c/0xd0 __of_changeset_entry_notify+0x1b8/0x230 __of_changeset_apply_notify+0x54/0xe4 of_overlay_fdt_apply+0xc04/0xd94 ... The buggy address belongs to the object at ffffff81beca5600 which belongs to the cache kmalloc-128 of size 128 The buggy address is located 8 bytes inside of 128-byte region [fffff81beca5600, ffffff81beca5680) The buggy address belongs to the physical page: page:00000000230d3d03 refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x1beca4 head:00000000230d3d03 order:1 compound_mapcount:0 compound_pincount:0 flags: 0x8000000000010200(slab head zone=2) raw: 8000000000010200 0000000000000000 dead000000000122 ffffff810000c300 raw: 0000000000000000 0000000000200020 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffffff81beca5500: 04 fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ffffffff81beca5580: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ffffffff81beca5600: 00 fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ffffffff81beca5680: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc ffffffff81beca5700: 00 00 00 00 00 00 fc fc fc fc fc fc fc fc fc fc ===== OF: -> got it ! Prevent the out-of-bounds read by copying the device address into a buffer of sufficient size.
CVE-2024-46744	In the Linux kernel, the following vulnerability has been resolved: Squashfs: sanity check symbolic link size Syzkiller reports a "KMSAN: uninitialized-value in pick_link" bug. This is caused by an uninitialised page, which is ultimately caused by a corrupted symbolic link size read from disk. The reason why the corrupted symlink size causes an uninitialised page is due to the following sequence of events: 1. squashfs_read_inode() is called to read the symbolic link from disk. This assigns the corrupted value 3875536935 to inode->i_size. 2. Later squashfs_symlink_read_folio() is called, which assigns this corrupted value to the length variable, which being a signed int, overflows producing a negative number. 3. The following loop that fills in the page contents checks that the copied bytes is less than length, which being negative means the loop is skipped, producing an uninitialised page. This patch adds a sanity check which checks that the symbolic link size is not larger than expected. -- V2: fix spelling mistake.
CVE-2024-46745	In the Linux kernel, the following vulnerability has been resolved: Input: uinput - reject requests with unreasonable number of slots When exercising uinput interface syzkaller may try setting up device with a really large number of slots, which causes memory allocation failure in input_mt_init_slots(). While this allocation failure is handled properly and request is rejected, it results in syzkaller reports. Additionally, such request may put undue burden on the system which will try to free a lot of memory for a bogus request. Fix it by limiting allowed number of slots to 100. This can easily be extended if we see devices that can track more than 100 contacts.

CVE ID	Description
CVE-2024-46746	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: free driver_data after destroying hid device HID driver callbacks aren't called anymore once hid_destroy_device() has been called. Hence, hid driver_data should be freed only after the hid_destroy_device() function returned as driver_data is used in several callbacks. I observed a crash with kernel 6.10.0 on my T14s Gen 3, after enabling KASAN to debug memory allocation, I got this output: [13.050438] <pre>===== [13.054060] BUG: KASAN: slab-use-after-free in amd_sfh_get_report+0x3ec/0x530 [amd_sfh] [13.054809] psmouse serio1: trackpoint: Synaptics TrackPoint firmware: 0x02, buttons: 3/3 [13.056432] Read of size 8 at addr ffff88813152f408 by task (udev- worker)/479 [13.060970] CPU: 5 PID: 479 Comm: (udev-worker) Not tainted 6.10.0-arch1-2 #1 893bb55d7f0073f25c46adbb49eb3785fedf74b0 [13.063978] Hardware name: LENOVO 21CQCTO1WW/21CQCTO1WW, BIOS R22ET70W (1.40) 03/21/2024 [13.067860] Call Trace: [13.069383] input: TPPS/2 Synaptics TrackPoint as /devices/platform/i8042/serio1/ input/input8 [13.071486] <TASK> [13.071492] dump_stack_lvl+0x5d/0x80 [13.074870] snd_hda_intel 0000:33:00.6: enabling device (0000 -> 0002) [13.078296] ? amd_sfh_get_report +0x3ec/0x530 [amd_sfh 05f43221435b5205f734cd9da29399130f398a38] [13.082199] print_report+0x174/0x505 [13.085776] ? __pfx__raw_spin_lock_irqsave+0x10/0x10 [13.089367] ? srso_alias_return_thunk+0x5/0xfbfef5 [13.093255] ? amd_sfh_get_report +0x3ec/0x530 [amd_sfh 05f43221435b5205f734cd9da29399130f398a38] [13.097464] kasan_report+0xc8/0x150 [13.101461] ? amd_sfh_get_report+0x3ec/0x530 [amd_sfh 05f43221435b5205f734cd9da29399130f398a38] [13.105802] amd_sfh_get_report +0x3ec/0x530 [amd_sfh 05f43221435b5205f734cd9da29399130f398a38] [13.110303] amddp_hid_request+0xb8/0x110 [amd_sfh 05f43221435b5205f734cd9da29399130f398a38] [13.114879] ? srso_alias_return_thunk+0x5/0xfbfef5 [13.119450] sensor_hub_get_feature +0x1d3/0x540 [hid_sensor_hub 3f13be3016ff415bea03008d45d99da837ee3082] [13.124097] hid_sensor_parse_common_attributes+0x4d0/0xad0 [hid_sensor_iio_common c3a5cbe93969c28b122609768bbe23efe52eb8f5] [13.127404] ? srso_alias_return_thunk+0x5/0xfbfef5 [13.131925] ? __pfx_hid_sensor_parse_common_attributes+0x10/0x10 [hid_sensor_iio_common c3a5cbe93969c28b122609768bbe23efe52eb8f5] [13.136455] ? __raw_spin_lock_irqsave +0x96/0xf0 [13.140197] ? __pfx__raw_spin_lock_irqsave+0x10/0x10 [13.143602] ? devm_iio_device_alloc+0x34/0x50 [industrialio 3d261d5e5765625d2b052be40e526d62b1d2123b] [13.147234] ? srso_alias_return_thunk+0x5/0xfbfef5 [13.150446] ? __devm_add_action +0x167/0x1d0 [13.155061] hid_gyro_3d_probe+0x120/0x7f0 [hid_sensor_gyro_3d 63da36a143b775846ab2dbb86c343b401b5e3172] [13.158581] ? srso_alias_return_thunk +0x5/0xfbfef5 [13.161814] platform_probe+0xa2/0x150 [13.165029] really_probe+0x1e3/0x8a0 [13.168243] __driver_probe_device+0x18c/0x370 [13.171500] driver_probe_device+0x4a/0x120 [13.175000] __driver_attach+0x190/0x4a0 [13.178521] ? __pfx__driver_attach+0x10/0x10 [13.181771] bus_for_each_dev+0x106/0x180 [13.185033] ? __pfx__raw_spin_lock+0x10/0x10 [13.188229] ? __pfx_bus_for_each_dev+0x10/0x10 [13.191446] ? srso_alias_return_thunk +0x5/0xfbfef5 [13.194382] bus_add_driver+0x29e/0x4d0 [13.197328] driver_register+0x1a5/0x360 [13.200283] ? __pfx_hid_gyro_3d_platform_driver_init+0x10/0x10 [hid_sensor_gyro_3d 63da36a143b775846ab2dbb86c343b401b5e3172] [13.203362] do_one_initcall+0xa7/0x380 [13.206432] ? __pfx_do_one_initcall+0x10/0x10 [13.210175] ? srso_alias_return_thunk+0x5/0xfbfef5 [13.213211] ? kasan_unpoison+0x44/0x70 [13.216688] do_init_module+0x238/0x750 [13.2196 --- truncated---</pre>
CVE-2024-46748	In the Linux kernel, the following vulnerability has been resolved: cachefiles: Set the max subreq size for cache writes to MAX_RW_COUNT Set the maximum size of a subrequest that writes to cachefiles to be MAX_RW_COUNT so that we don't overrun the maximum write we can make to the backing filesystem.
CVE-2024-46750	In the Linux kernel, the following vulnerability has been resolved: PCI: Add missing bridge lock to pci_bus_lock() One of the true positives that the cfg_access_lock lockdep effort identified is this sequence: WARNING: CPU: 14 PID: 1 at drivers/pci/pci.c:4886 pci_bridge_secondary_bus_reset+0x5d/0x70 RIP: 0010:pci_bridge_secondary_bus_reset+0x5d/0x70 Call Trace: <TASK> ? __warn+0x8c/0x190 ? pci_bridge_secondary_bus_reset+0x5d/0x70 ? report_bug+0x1f8/0x200 ? handle_bug+0x3c/0x70 ? exc_invalid_op+0x18/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? pci_bridge_secondary_bus_reset+0x5d/0x70 pci_reset_bus+0x1d8/0x270 vmd_probe+0x778/0xa10 pci_device_probe+0x95/0x120 Where pci_reset_bus() users are triggering unlocked secondary bus resets. Ironically pci_bus_reset(), several calls down from pci_reset_bus(), uses pci_bus_lock() before issuing the reset which locks everything *but* the bridge itself. For the same motivation as adding: bridge = pci_upstream_bridge(dev); if (bridge) pci_dev_lock(bridge); to pci_reset_function() for the "bus" and "cxl_bus" reset cases, add pci_dev_lock() for @bus->self to pci_bus_lock(). [bhelgaas: squash in recursive locking deadlock fix from Keith Busch: https://lore.kernel.org/r/20240711193650.701834-1-kbusch@meta.com]

CVE ID	Description
CVE-2024-46754	In the Linux kernel, the following vulnerability has been resolved: bpf: Remove tst_run from lwt_seg6local_prog_ops. The syzbot reported that the lwt_seg6 related BPF ops can be invoked via bpf_test_run() without without entering input_action_end_bpf() first. Martin KaFai Lau said that self test for BPF_PROG_TYPE_LWT_SEG6LOCAL probably didn't work since it was introduced in commit 04d4b274e2a ("ipv6: sr: Add seg6local action End.BPF"). The reason is that the per-CPU variable seg6_bpf_srh_states::srh is never assigned in the self test case but each BPF function expects it. Remove test_run for BPF_PROG_TYPE_LWT_SEG6LOCAL.
CVE-2024-46755	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Do not return unused priv in mwifiex_get_priv_by_id() mwifiex_get_priv_by_id() returns the priv pointer corresponding to the bss_num and bss_type, but without checking if the priv is actually currently in use. Unused priv pointers do not have a wiphy attached to them which can lead to NULL pointer dereferences further down the callstack. Fix this by returning only used priv pointers which have priv->bss_mode set to something else than NL80211_IFTYPE_UNSPECIFIED. Said NULL pointer dereference happened when an Accesspoint was started with wpa_supplicant -i wlan0 with this config: network={ ssid="somessid" mode=2 frequency=2412 key_mgmt=WPA-PSK WPA-PSK-SHA256 proto=RSN group=CCMP pairwise=CCMP psk="12345678" } When waiting for the AP to be established, interrupting wpa_supplicant with <ctrl-c> and starting it again this happens: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000140 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtbl: 4k pages, 48-bit VAs, pgdp=0000000046d96000 [0000000000000140] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP Modules linked in: caam_jr caamhash_desc spidev caamalg_desc crypto_engine authenc libdes mwifiex_sdio +mwifiex crct10dif_ce cdc_acm onboard_usb_hub fsl_imx8_ddr_perf imx8m_ddrc rtc_ds1307 lm75 rtc_snvs +imx_sdma caam imx8mm_thermal spi_imx error imx_cpufreq_dt fuse ip_tables x_tables ipv6 CPU: 0 PID: 8 Comm: kworker/0:1 Not tainted 6.9.0-00007-g937242013fce-dirty #18 Hardware name: somemachine (DT) Workqueue: events sdio_irq_work pstate: 00000005 (nzcw daif -PAN -UAO -TCO -DIT -SSBS BTYP=) pc : mwifiex_get_cfp+0xd8/0x15c [mwifiex] lr : mwifiex_get_cfp+0x34/0x15c [mwifiex] sp : ffff8000818b3a70 x29: ffff8000818b3a70 x28: ffff000006bfd8a5 x27: 0000000000000004 x26: 000000000000002c x25: 0000000000001511 x24: 0000000002e86bc9 x23: ffff000006bfd996 x22: 0000000000000004 x21: ffff000007bec000 x20: 000000000000002c x19: 0000000000000000 x18: 0000000000000000 x17: 000000040044ffff x16: 00500072b5503510 x15: ccc283740681e517 x14: 0201000101006d15 x13: 0000000002e8ff43 x12: 002c01000000ffb1 x11: 0100000000000000 x10: 02e8ff43002c0100 x9: 0000ffb100100157 x8: ffff000003d20000 x7: 000000000000002f x6: 00000000ffffe124 x5: 0000000000000001 x4: 0000000000000003 x3: 0000000000000000 x2: 0000000000000000 x1: 0001000000011001 x0: 0000000000000000 Call trace: mwifiex_get_cfp+0xd8/0x15c [mwifiex] mwifiex_parse_single_response_buf+0x1d0/0x504 [mwifiex] mwifiex_handle_event_ext_scan_report+0x19c/0x2f8 [mwifiex] mwifiex_process_sta_event+0x298/0xf0c [mwifiex] mwifiex_process_event+0x110/0x238 [mwifiex] mwifiex_main_process+0x428/0xa44 [mwifiex] mwifiex_sdio_interrupt+0x64/0x12c [mwifiex_sdio] process_sdio_pending_irqs+0x64/0x1b8 sdio_irq_work+0x4c/0x7c process_one_work+0x148/0x2a0 worker_thread+0x2fc/0x40c kthread+0x110/0x114 ret_from_fork+0x10/0x20 Code: a94153f3 a8c37bfd d50323bf d65f03c0 (f940a000) ---[end trace 0000000000000000]---
CVE-2024-46759	In the Linux kernel, the following vulnerability has been resolved: hwmon: (adc128d818) Fix underflows seen when writing limit attributes DIV_ROUND_CLOSEST() after kstrtol() results in an underflow if a large negative number such as -9223372036854775808 is provided by the user. Fix it by reordering clamp_val() and DIV_ROUND_CLOSEST() operations.
CVE-2024-46760	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: usb: schedule rx work after everything is set up Right now it's possible to hit NULL pointer dereference in rtw_rx_fill_rx_status on hw object and/or its fields because initialization routine can start getting USB replies before rtw_dev is fully setup. The stack trace looks like this: rtw_rx_fill_rx_status rtw8821c_query_rx_desc rtw_usb_rx_handler ... queue_work rtw_usb_read_port_complete ... usb_submit_urb rtw_usb_rx_resubmit rtw_usb_init_rx rtw_usb_probe So while we do the async stuff rtw_usb_probe continues and calls rtw_register_hw, which does all kinds of initialization (e.g. via ieee80211_register_hw) that rtw_rx_fill_rx_status relies on. Fix this by moving the first usb_submit_urb after everything is set up. For me, this bug manifested as: [8.893177] rtw_8821cu 1-1:1.2: band wrong, packet dropped [8.910904] rtw_8821cu 1-1:1.2: hw->conf.chandef.chan NULL in rtw_rx_fill_rx_status because I'm using Larry's backport of rtw88 driver with the NULL checks in rtw_rx_fill_rx_status.

CVE ID	Description
CVE-2024-46761	In the Linux kernel, the following vulnerability has been resolved: pci/hotplug/pnv_php: Fix hotplug driver crash on Powernv The hotplug driver for powerpc (pci/hotplug/pnv_php.c) causes a kernel crash when we try to hot-unplug/disable the PCIe switch/bridge from the PHB. The crash occurs because although the MSI data structure has been released during disable/hot-unplug path and it has been assigned with NULL, still during unregistration the code was again trying to explicitly disable the MSI which causes the NULL pointer dereference and kernel crash. The patch fixes the check during unregistration path to prevent invoking pci_disable_msi/msix() since its data structure is already freed.
CVE-2024-46762	In the Linux kernel, the following vulnerability has been resolved: xen: privcmd: Fix possible access to a freed kirqfd instance Nothing prevents simultaneous ioctl calls to privcmd_irqfd_assign() and privcmd_irqfd_deassign(). If that happens, it is possible that a kirqfd created and added to the irqfds_list by privcmd_irqfd_assign() may get removed by another thread executing privcmd_irqfd_deassign(), while the former is still using it after dropping the locks. This can lead to a situation where an already freed kirqfd instance may be accessed and cause kernel oops. Use SRCU locking to prevent the same, as is done for the KVM implementation for irqfds.
CVE-2024-46765	In the Linux kernel, the following vulnerability has been resolved: ice: protect XDP configuration with a mutex The main threat to data consistency in ice_xdp() is a possible asynchronous PF reset. It can be triggered by a user or by TX timeout handler. XDP setup and PF reset code access the same resources in the following sections: * ice_vsi_close() in ice_prepare_for_reset() - already rtnl-locked * ice_vsi_rebuild() for the PF VSI - not protected * ice_vsi_open() - already rtnl-locked With an unfortunate timing, such accesses can result in a crash such as the one below: [+1.999878] ice 0000:b1:00.0: Registered XDP mem model MEM_TYPE_XSK_BUFF_POOL on Rx ring 14 [+2.002992] ice 0000:b1:00.0: Registered XDP mem model MEM_TYPE_XSK_BUFF_POOL on Rx ring 18 [Mar15 18:17] ice 0000:b1:00.0 ens801f0np0: NETDEV WATCHDOG: CPU: 38: transmit queue 14 timed out 80692736 ms [+0.000093] ice 0000:b1:00.0 ens801f0np0: tx_timeout: VSI_num: 6, Q 14, NTC: 0x0, HW_HEAD: 0x0, NTU: 0x0, INT: 0x4000001 [+0.000012] ice 0000:b1:00.0 ens801f0np0: tx_timeout recovery level 1, txqueue 14 [+0.394718] ice 0000:b1:00.0: PTP reset successful [+0.006184] BUG: kernel NULL pointer dereference, address: 0000000000000098 [+0.000045] #PF: supervisor read access in kernel mode [+0.000023] #PF: error_code(0x0000) - not-present page [+0.000023] PGD 0 P4D 0 [+0.000018] Oops: 0000 [#1] PREEMPT SMP NOPTI [+0.000023] CPU: 38 PID: 7540 Comm: kworker/38:1 Not tainted 6.8.0-rc7 #1 [+0.000031] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0014.082620210524 08/26/2021 [+0.000036] Workqueue: ice ice_service_task [ice] [+0.000183] RIP: 0010:ice_clean_tx_ring+0xa/0xd0 [ice] [...] [+0.000013] Call Trace: [+0.000016] <TASK> [+0.000014] ? __die+0x1f/0x70 [+0.000029] ? page_fault_oops+0x171/0x4f0 [+0.000029] ? schedule+0x3b/0xd0 [+0.000027] ? exc_page_fault+0x7b/0x180 [+0.000022] ? asm_exc_page_fault+0x22/0x30 [+0.000031] ? ice_clean_tx_ring+0xa/0xd0 [ice] [+0.000194] ice_free_tx_ring+0xe/0x60 [ice] [+0.000186] ice_destroy_xdp_rings+0x157/0x310 [ice] [+0.000151] ice_vsi_decfg+0x53/0xe0 [ice] [+0.000180] ice_vsi_rebuild+0x239/0x540 [ice] [+0.000186] ice_vsi_rebuild_by_type+0x76/0x180 [ice] [+0.000145] ice_rebuild+0x18c/0x840 [ice] [+0.000145] ? delay_tsc+0x4a/0xc0 [+0.000022] ? delay_tsc+0x92/0xc0 [+0.000020] ice_do_reset+0x140/0x180 [ice] [+0.000886] ice_service_task+0x404/0x1030 [ice] [+0.000824] process_one_work+0x171/0x340 [+0.000685] worker_thread+0x277/0x3a0 [+0.000675] ? preempt_count_add+0x6a/0xa0 [+0.000677] ? _raw_spin_lock_irqsave+0x23/0x50 [+0.000679] ? __pfx_worker_thread+0x10/0x10 [+0.000653] kthread+0xf0/0x120 [+0.000635] ? __pfx_kthread+0x10/0x10 [+0.000616] ret_from_fork+0x2d/0x50 [+0.000612] ? __pfx_kthread+0x10/0x10 [+0.000604] ret_from_fork_asm+0x1b/0x30 [+0.000604] </TASK> The previous way of handling this through returning -EBUSY is not viable, particularly when destroying AF_XDP socket, because the kernel proceeds with removal anyway. There is plenty of code between those calls and there is no need to create a large critical section that covers all of them, same as there is no need to protect ice_vsi_rebuild() with rtnl_lock(). Add xdp_state_lock mutex to protect ice_vsi_rebuild() and ice_xdp(). Leaving unprotected sections in between would result in two states that have to be considered: 1. when the VSI is closed, but not yet rebuild 2. when VSI is already rebuild, but not yet open The latter case is actually already handled through !netif_running() case, we just need to adjust flag checking a little. The former one is not as trivial, because between ice_vsi_close() and ice_vsi_rebuild(), a lot of hardware interaction happens, this can make adding/deleting rings exit with an error. Luckily, VSI rebuild is pending and can apply new configuration for us in a managed fashion. Therefore, add an additional VSI state flag ICE_VSI_REBUILD_PENDING to indicate that ice_x ---truncated---

CVE ID	Description
CVE-2024-46770	In the Linux kernel, the following vulnerability has been resolved: ice: Add netif_device_attach/detach into PF reset flow Ehtool callbacks can be executed while reset is in progress and try to access deleted resources, e.g. getting coalesce settings can result in a NULL pointer dereference seen below. Reproduction steps: Once the driver is fully initialized, trigger reset: # echo 1 > /sys/class/net/<interface>/device/reset when reset is in progress try to get coalesce settings using ethtool: # ethtool -c <interface> BUG: kernel NULL pointer dereference, address: 0000000000000020 PGD 0 P4D 0 Oops: Oops: 0000 [#1] PREEMPT SMP PTI CPU: 11 PID: 19713 Comm: ethtool Tainted: G S 6.10.0-rc7+ #7 RIP: 0010:ice_get_q_coalesce+0x2e/0xa0 [ice] RSP: 0018:ffffbab1e9bcf6a8 EFLAGS: 00010206 RAX: 000000000000000c RBX: ffff94512305b028 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffff9451c3f2e588 RDI: ffff9451c3f2e588 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: ffff9451c3f2e580 R11: 0000000000000001f R12: ffff945121fa9000 R13: ffff9451e9bcf760 R14: 0000000000000013 R15: ffffffff9e65dd40 FS: 00007faee5fbc740(0000) GS: ffff94546fd80000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000020 CR3: 0000000106c2e005 CR4: 00000000001706f0 Call Trace: <TASK> ice_get_coalesce+0x17/0x30 [ice] coalesce_prepare_data+0x61/0x80 ethnl_default_doit+0xde/0x340 genl_family_rcv_msg_doit+0xf2/0x150 genl_rcv_msg+0x1b3/0x2c0 netlink_rcv_skb+0x5b/0x110 genl_rcv+0x28/0x40 netlink_unicast+0x19c/0x290 netlink_sendmsg+0x222/0x490 __sys_sendto+0x1df/0x1f0 __x64_sys_sendto+0x24/0x30 do_syscall_64+0x82/0x160 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7faee60d8e27 Calling netif_device_detach() before reset makes the net core not call the driver when ethtool command is issued, the attempt to execute an ethtool command during reset will result in the following message: netlink error: No such device instead of NULL pointer dereference. Once reset is done and ice_rebuild() is executing, the netif_device_attach() is called to allow for ethtool operations to occur again in a safe manner.
CVE-2024-46771	In the Linux kernel, the following vulnerability has been resolved: can: bcm: Remove proc entry when dev is unregistered. syzkaller reported a warning in bcm_connect() below. [0] The repro calls connect() to vxcan1, removes vxcan1, and calls connect() with ifindex == 0. Calling connect() for a BCM socket allocates a proc entry. Then, bcm_sk(sk)->bound is set to 1 to prevent further connect(). However, removing the bound device resets bcm_sk(sk)->bound to 0 in bcm_notify(). The 2nd connect() tries to allocate a proc entry with the same name and sets NULL to bcm_sk(sk)->bcm_proc_read, leaking the original proc entry. Since the proc entry is available only for connect(ed) sockets, let's clean up the entry when the bound netdev is unregistered. [0]: proc_dir_entry 'can-bcm/2456' already registered WARNING: CPU: 1 PID: 394 at fs/proc/generic.c:376 proc_register+0x645/0x8f0 fs/proc/generic.c:375 Modules linked in: CPU: 1 PID: 394 Comm: syz-executor403 Not tainted 6.10.0-rc7-g852e42cc2dd4 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:proc_register+0x645/0x8f0 fs/proc/generic.c:375 Code: 00 00 00 00 00 48 85 ed 0f 85 97 02 00 00 4d 85 f6 0f 85 9f 02 00 00 48 c7 c7 9b cb cf 87 48 89 de 4c 89 fa e8 1c 6f eb fe 90 <of> 0b 90 90 48 c7 c7 98 37 99 89 e8 cb 7e 22 05 bb 00 00 00 10 48 RSP: 0018:ffa0000000cd7c30 EFLAGS: 00010246 RAX: 9e129be1950f0200 RBX: ff1100011b51582c RCX: ff1100011857cd80 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000002 RBP: 0000000000000000 R08: ffd400000000000f R09: ff1100013e78cac0 R10: ffac800000cd7980 R11: ff1100013e12b1f0 R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000000 R15: ff1100011a99a2ec FS: 00007fbd7086f740(0000) GS: ff1100013fd00000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000200071c0 CR3: 0000000118556004 CR4: 0000000000771ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe07f0 DR7: 00000000000000400 PKRU: 55555554 Call Trace: <TASK> proc_create_net_single+0x144/0x210 fs/proc/proc_net.c:220 bcm_connect+0x472/0x840 net/can/bcm.c:1673 __sys_connect_file net/socket.c:2049 [inline] __sys_connect+0x5d2/0x690 net/socket.c:2066 __do_sys_connect net/socket.c:2076 [inline] __se_sys_connect net/socket.c:2073 [inline] __x64_sys_connect+0x8f/0x100 net/socket.c:2073 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xd9/0x1c0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x4b/0x53 RIP: 0033:0x7fbd708b0e5d Code: ff c3 66 2e 0f 1f 84 00 00 00 00 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 73 9f 1b 00 f7 d8 64 89 01 48 RSP: 002b:00007fff8cd33f08 EFLAGS: 00000246 ORIG_RAX: 000000000000002a RAX: ffffffff9ffffda RBX: 0000000000000003 RCX: 00007fbd708b0e5d RDX: 0000000000000010 RSI: 0000000020000040 RDI: 0000000000000003 RBP: 0000000000000000 R08: 0000000000000040 R09: 0000000000000040 R10: 0000000000000040 R11: 0000000000000246 R12: 00007fff8cd34098 R13: 0000000000401280 R14: 0000000000406de8 R15: 00007fbd70ab9000 </TASK> remove_proc_entry: removing non-empty directory 'net/can-bcm', leaking at least '2456'
CVE-2024-46774	In the Linux kernel, the following vulnerability has been resolved: powerpc/rtas: Prevent Spectre v1 gadget construction in sys_rtas() Smatch warns: arch/powerpc/kernel/rtas.c:1932 __do_sys_rtas() warn: potential spectre issue 'args.args' [r] (local cap) The 'nargs' and 'nret' locals come directly from a user-supplied buffer and are used as indexes into a small stack-based array and as inputs to copy_to_user() after they are subject to bounds checks. Use array_index_nospec() after the bounds checks to clamp these values for speculative execution.

CVE ID	Description
CVE-2024-46777	In the Linux kernel, the following vulnerability has been resolved: udf: Avoid excessive partition lengths Avoid mounting filesystems where the partition would overflow the 32-bits used for block number. Also refuse to mount filesystems where the partition length is so large we cannot safely index bits in a block bitmap.
CVE-2024-46783	In the Linux kernel, the following vulnerability has been resolved: tcp_bpf: fix return value of tcp_bpf_sendmsg() When we cork messages in psock->cork, the last message triggers the flushing will result in sending a sk_msg larger than the current message size. In this case, in tcp_bpf_send_verdict(), 'copied' becomes negative at least in the following case: 468 case __SK_DROP: 469 default: 470 sk_msg_free_partial(sk, msg, tosend); 471 sk_msg_apply_bytes(psock, tosend); 472 *copied -= (tosend + delta); // <==== HERE 473 return -EACCES; Therefore, it could lead to the following BUG with a proper value of 'copied' (thanks to syzbot). We should not use negative 'copied' as a return value here. -----[cut here]----- kernel BUG at net/socket.c:733! Internal error: Oops - BUG: 0000000f2000800 [#1] PREEMPT SMP Modules linked in: CPU: 0 UID: 0 PID: 3265 Comm: syz-executor510 Not tainted 6.11.0-rc3-syzkaller-00060-gd07b43284ab3 #0 Hardware name: linux,dummy-virt (DT) pstate: 61400009 (nZCv daif +PAN -UAO -TCO +DIT -SSBS BTYP=) pc : sock_sendmsg_nosec net/socket.c:733 [inline] pc : sock_sendmsg_nosec net/socket.c:728 [inline] pc : __sock_sendmsg+0x5c/0x60 net/socket.c:745 lr : sock_sendmsg_nosec net/socket.c:730 [inline] lr : __sock_sendmsg+0x54/0x60 net/socket.c:745 sp : ffff800088ea3b30 x29: ffff800088ea3b30 x28: fbf00000062bc900 x27: 0000000000000000 x26: ffff800088ea3bc0 x25: ffff800088ea3bc0 x24: 0000000000000000 x23: f9f00000048dc000 x22: 0000000000000000 x21: ffff800088ea3d90 x20: f9f00000048dc000 x19: ffff800088ea3d90 x18: 0000000000000001 x17: 0000000000000000 x16: 0000000000000000 x15: 000000002002ffaf x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000000 x10: ffff8000815849c0 x9: ffff8000815b49c0 x8: 0000000000000000 x7: 0000000000000003f x6: 0000000000000000 x5: 0000000000000007e0 x4: fff07fffd239000 x3: fbf00000062bc900 x2: 0000000000000000 x1: 0000000000000000 x0: 00000000fffffdef Call trace: sock_sendmsg_nosec net/socket.c:733 [inline] __sock_sendmsg+0x5c/0x60 net/socket.c:745 __sys_sendmsg+0x274/0x2ac net/socket.c:2597 __sys_sendmsg+0xac/0x100 net/socket.c:2651 __sys_sendmsg+0x84/0xe0 net/socket.c:2680 __do_sys_sendmsg net/socket.c:2689 [inline] __se_sys_sendmsg net/socket.c:2687 [inline] __arm64_sys_sendmsg+0x24/0x30 net/socket.c:2687 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x48/0x110 arch/arm64/kernel/syscall.c:49 el0_svc_common.constprop.0+0x40/0xe0 arch/arm64/kernel/syscall.c:132 do_el0_svc+0x1c/0x28 arch/arm64/kernel/syscall.c:151 el0_svc+0x34/0xec arch/arm64/kernel/entry-common.c:712 el0t_64_sync_handler+0x100/0x12c arch/arm64/kernel/entry-common.c:730 el0t_64_sync+0x19c/0x1a0 arch/arm64/kernel/entry.S:598 Code: f9404463 d63f0060 3108441f 54fffe81 (d4210000) ---[end trace 0000000000000000]---
CVE-2024-46784	In the Linux kernel, the following vulnerability has been resolved: net: mana: Fix error handling in mana_create_txq/rxq's NAPI cleanup Currently napi_disable() gets called during rxq and txq cleanup, even before napi is enabled and hrtimer is initialized. It causes kernel panic. ? page_fault_oops+0x136/0x2b0 ? page_counter_cancel+0x2e/0x80 ? do_user_addr_fault+0x2f2/0x640 ? refill_obj_stock+0xc4/0x110 ? exc_page_fault+0x71/0x160 ? asm_exc_page_fault+0x27/0x30 ? __mmdrop+0x10/0x180 ? __mmdrop+0xec/0x180 ? hrtimer_active+0xd/0x50 hrtimer_try_to_cancel+0x2c/0xf0 hrtimer_cancel+0x15/0x30 napi_disable+0x65/0x90 mana_destroy_rxq+0x4c/0x2f0 mana_create_rxq.isra.0+0x56c/0x6d0 ? mana_uncfg_vport+0x50/0x50 mana_alloc_queues+0x21b/0x320 ? skb_dequeue+0x5f/0x80

CVE ID	Description
CVE-2024-46787	In the Linux kernel, the following vulnerability has been resolved: userfaultfd: fix checks for huge PMDs Patch series "userfaultfd: fix races around pmd_trans_huge() check", v2. The pmd_trans_huge() code in mfill_atomic() is wrong in three different ways depending on kernel version: 1. The pmd_trans_huge() check is racy and can lead to a BUG_ON() (if you hit the right two race windows) - I've tested this in a kernel build with some extra mdelay() calls. See the commit message for a description of the race scenario. On older kernels (before 6.5), I think the same bug can even theoretically lead to accessing transhuge page contents as a page table if you hit the right 5 narrow race windows (I haven't tested this case). 2. As pointed out by Qi Zheng, pmd_trans_huge() is not sufficient for detecting PMDs that don't point to page tables. On older kernels (before 6.5), you'd just have to win a single fairly wide race to hit this. I've tested this on 6.1 stable by racing migration (with a mdelay() patched into try_to_migrate()) against UFFDIO_ZEROPAGE - on my x86 VM, that causes a kernel oops in ptlock_ptr(). 3. On newer kernels (>=6.5), for shmem mappings, khugepaged is allowed to yank page tables out from under us (though I haven't tested that), so I think the BUG_ON() checks in mfill_atomic() are just wrong. I decided to write two separate fixes for these (one fix for bugs 1+2, one fix for bug 3), so that the first fix can be backported to kernels affected by bugs 1+2. This patch (of 2): This fixes two issues. I discovered that the following race can occur: mfill_atomic other thread <pre> ===== <zap PMD> pmdp_get_lockless() [reads none pmd] <bail if trans_huge> <if none:> <pagefault creates transhuge zeropage> __pte_alloc [no-op] <zap PMD> <bail if pmd_trans_huge(*dst_pmd)> BUG_ON(pmd_none(*dst_pmd))</pre> I have experimentally verified this in a kernel with extra mdelay() calls; the BUG_ON(pmd_none(*dst_pmd)) triggers. On kernels newer than commit 0d940a9b270b ("mm/pgtable: allow pte_offset_map[_lock]() to fail"), this can't lead to anything worse than a BUG_ON(), since the page table access helpers are actually designed to deal with page tables concurrently disappearing; but on older kernels (<=6.4), I think we could probably theoretically race past the two BUG_ON() checks and end up treating a hugepage as a page table. The second issue is that, as Qi Zheng pointed out, there are other types of huge PMDs that pmd_trans_huge() can't catch: devmap PMDs and swap PMDs (in particular, migration PMDs). On <=6.4, this is worse than the first issue: If mfill_atomic() runs on a PMD that contains a migration entry (which just requires winning a single, fairly wide race), it will pass the PMD to pte_offset_map_lock(), which assumes that the PMD points to a page table. Breakage follows: First, the kernel tries to take the PTE lock (which will crash or maybe worse if there is no "struct page" for the address bits in the migration entry PMD - I think at least on X86 there usually is no corresponding "struct page" thanks to the PTE inversion mitigation, amd64 looks different). If that didn't crash, the kernel would next try to write a PTE into what it wrongly thinks is a page table. As part of fixing these issues, get rid of the check for pmd_trans_huge() before __pte_alloc() - that's redundant, we're going to have to check for that after the __pte_alloc() anyway. Backport note: pmdp_get_lockless() is pmd_read_atomic() in older kernels.

CVE ID	Description
CVE-2024-46788	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Use a cpumask to know what threads are kthreads The start_kthread() and stop_thread() code was not always called with the interface _lock held. This means that the kthread variable could be unexpectedly changed causing the kthread_stop() to be called on it when it should not have been, leading to: while true; do rtla timerlat top -u -q & PID=\$!; sleep 5; kill -INT \$PID; sleep 0.001; kill -TERM \$PID; wait \$PID; done Causing the following OOPS: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000002: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017] CPU: 5 UID: 0 PID: 885 Comm: timerlatu/5 Not tainted 6.11.0-rc4-test-00002-gbc754cc76d1b-dirty #125 a533010b71dab205ad2f507188ce8c82203b0254 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010:hrtimer_active+0x58/0x300 Code: 48 c1 ee 03 41 54 48 01 d1 48 01 d6 55 53 48 83 ec 20 80 39 00 0f 85 30 02 00 00 49 8b 6f 30 4c 8d 75 10 4c 89 f0 48 c1 e8 03 <0f> b6 3c 10 4c 89 f0 83 e0 07 83 c0 03 40 38 f8 7c 09 40 84 ff 0f RSP: 0018:ffff8881d97f940 EFLAGS: 00010202 RAX: 0000000000000002 RBX: ffff88823c6b5b28 RCX: ffffd10478d6b6b RDX: dffffc0000000000 RSI: ffffd10478d6b6c RDI: ffff88823c6b5b28 RBP: 0000000000000000 R08: ffff88823c6b5b58 R09: ffff88823c6b5b60 R10: ffff8881d97f957 R11: 0000000000000010 R12: 00000000000a801d R13: ffff88810d8b35d8 R14: 0000000000000010 R15: ffff88823c6b5b28 FS: 0000000000000000(0000) GS:ffff88823c680000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000561858ad7258 CR3: 000000007729e001 CR4: 000000000170ef0 Call Trace: <TASK> ? die_addr+0x40/0xa0 ? exc_general_protection+0x154/0x230 ? asm_exc_general_protection+0x26/0x30 ? hrtimer_active+0x58/0x300 ? __pfx_mutex_lock+0x10/0x10 ? __pfx_locks_remove_file+0x10/0x10 hrtimer_cancel+0x15/0x40 timerlat_fd_release+0x8e/0x1f0 ? security_file_release+0x43/0x80 __fput+0x372/0xb10 task_work_run+0x11e/0x1f0 ? _raw_spin_lock+0x85/0xe0 ? __pfx_task_work_run+0x10/0x10 ? poison_slab_object+0x109/0x170 ? do_exit+0x7a0/0x24b0 do_exit+0x7bd/0x24b0 ? __pfx_migrate_enable+0x10/0x10 ? __pfx_do_exit+0x10/0x10 ? __pfx_read_tsc+0x10/0x10 ? ktime_get+0x64/0x140 ? _raw_spin_lock_irq+0x86/0xe0 do_group_exit+0xb0/0x220 get_signal+0x17ba/0x1b50 ? vfs_read+0x179/0xa40 ? timerlat_fd_read+0x30b/0x9d0 ? __pfx_get_signal+0x10/0x10 ? __pfx_timerlat_fd_read+0x10/0x10 arch_do_signal_or_restart+0x8c/0x570 ? __pfx_arch_do_signal_or_restart+0x10/0x10 ? vfs_read+0x179/0xa40 ? ksys_read+0xfe/0x1d0 ? __pfx_ksys_read+0x10/0x10 syscall_exit_to_user_mode+0x9c/0x130 do_syscall_64+0x74/0x110 ? __pfx__rseq_handle_notify_resume+0x10/0x10 ? __pfx_ksys_read+0x10/0x10 ? fpregs_restore_userregs+0xdb/0x1e0 ? fpregs_restore_userregs+0xdb/0x1e0 ? syscall_exit_to_user_mode+0x116/0x130 ? do_syscall_64+0x74/0x110 ? do_syscall_64+0x74/0x110 ? do_syscall_64+0x74/0x110 entry_SYSCALL_64_after_hwframe+0x71/0x79 RIP: 0033:0x7ff0070eca9c Code: Unable to access opcode bytes at 0x7ff0070eca72. RSP: 002b:00007ff006dff8c0 EFLAGS: 00000246 ORIG_RAX: 0000000000000000 RAX: 0000000000000000 RBX: 0000000000000005 RCX: 00007ff0070eca9c RDX: 0000000000000400 RSI: 00007ff006dff9a0 RDI: 0000000000000003 RBP: 00007ff006dffde0 R08: 0000000000000000 R09: 00007ff000000ba0 R10: 00007ff007004b08 R11: 0000000000000246 R12: 0000000000000003 R13: 00007ff006dff9a0 R14: 0000000000000007 R15: 0000000000000008 </TASK> Modules linked in: snd_hda_intel snd_intel_dspcfg snd_intel_sdw_acpi snd_hda_codec snd_hwdsp snd_hda_core ---[end trace 0000000000000000]--- This is because it would mistakenly call kthread_stop() on a user space thread making it "exit" before it actually exits. Since kthread ---truncated---
CVE-2024-46794	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Fix data leak in mmio_read() The mmio_read() function makes a TDVMMCALL to retrieve MMIO data for an address from the VMM. Sean noticed that mmio_read() unintentionally exposes the value of an initialized variable (val) on the stack to the VMM. This variable is only needed as an output value. It did not need to be passed to the VMM in the first place. Do not send the original value of *val to the VMM. [dhansen: clarify what 'val' is used for.]
CVE-2024-46798	In the Linux kernel, the following vulnerability has been resolved: ASoC: dapm: Fix UAF for snd_soc_pcm_runtime object When using kernel with the following extra config, - CONFIG_KASAN=y - CONFIG_KASAN_GENERIC=y - CONFIG_KASAN_INLINE=y - CONFIG_KASAN_VMALLOC=y - CONFIG_FRAME_WARN=4096 kernel detects that snd_pcm_suspend_all() access a freed 'snd_soc_pcm_runtime' object when the system is suspended, which leads to a use-after-free bug: [52.047746] BUG: KASAN: use-after-free in snd_pcm_suspend_all+0x1a8/0x270 [52.047765] Read of size 1 at addr ffff0000b9434d50 by task systemd-sleep/2330 [52.047785] Call trace: [52.047787] dump_backtrace+0x0/0x3c0 [52.047794] show_stack+0x34/0x50 [52.047797] dump_stack_lvl+0x68/0x8c [52.047802] print_address_description.constprop.0+0x74/0x2c0 [52.047809] kasan_report+0x210/0x230 [52.047815] __asan_report_load1_noabort+0x3c/0x50 [52.047820] snd_pcm_suspend_all+0x1a8/0x270 [52.047824] snd_soc_suspend+0x19c/0x4e0 The snd_pcm_sync_stop() has a NULL check on 'substream->runtime' before making any access. So we need to always set 'substream->runtime' to NULL everytime we kfree() it.

CVE ID	Description
CVE-2024-46800	In the Linux kernel, the following vulnerability has been resolved: sch/netem: fix use after free in netem_dequeue If netem_dequeue() enqueues packet to inner qdisc and that qdisc returns __NET_XMIT_STOLEN. The packet is dropped but qdisc_tree_reduce_backlog() is not called to update the parent's q.qlen, leading to the similar use-after-free as Commit e04991a48dbaf382 ("netem: fix return value if duplicate enqueue fails") Commands to trigger KASAN UaF: ip link add type dummy ip link set lo up ip link set dummy0 up tc qdisc add dev lo parent root handle 1: drr tc filter add dev lo parent 1: basic classid 1:1 tc class add dev lo classid 1:1 drr tc qdisc add dev lo parent 1:1 handle 2: netem tc qdisc add dev lo parent 2: handle 3: drr tc filter add dev lo parent 3: basic classid 3:1 action mirrored egress redirect dev dummy0 tc class add dev lo classid 3:1 drr ping -c1 -W0.01 localhost # Trigger bug tc class del dev lo classid 1:1 tc class add dev lo classid 1:1 drr ping -c1 -W0.01 localhost # UaF
CVE-2024-46805	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix the waring dereferencing hive Check the amdgpu_hive_info *hive that maybe is NULL.
CVE-2024-46807	In the Linux kernel, the following vulnerability has been resolved: drm/amd/amdgpu: Check tbo resource pointer Validate tbo resource pointer, skip if NULL
CVE-2024-46819	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: the warning dereferencing obj for nbio_v7_4 if ras_manager obj null, don't print NBIO err data
CVE-2024-46820	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu/vcn: remove irq disabling in vcn 5 suspend We do not directly enable/disable VCN IRQ in vcn 5.0.0. And we do not handle the IRQ state as well. So the calls to disable IRQ and set state are removed. This effectively gets rid of the warning of "WARN_ON(!amdgpu_irq_enabled(adev, src, type))" in amdgpu_irq_put().
CVE-2024-46822	In the Linux kernel, the following vulnerability has been resolved: arm64: acpi: Harden get_cpu_for_acpi_id() against missing CPU entry In a review discussion of the changes to support vCPU hotplug where a check was added on the GICC being enabled if was online, it was noted that there is need to map back to the cpu and use that to index into a cpumask. As such, a valid ID is needed. If an MPIDR check fails in acpi_map_gic_cpu_interface() it is possible for the entry in cpu_madt_gicc[cpu] == NULL. This function would then cause a NULL pointer dereference. Whilst a path to trigger this has not been established, harden this caller against the possibility.
CVE-2024-46825	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvvm: use IWL_FW_CHECK for link ID check The lookup function iwl_mvmm_rcu_fw_link_id_to_link_conf() is normally called with input from the firmware, so it should use IWL_FW_CHECK() instead of WARN_ON().
CVE-2024-46826	In the Linux kernel, the following vulnerability has been resolved: ELF: fix kernel.randomize_va_space double read ELF loader uses "randomize_va_space" twice. It is sysctl and can change at any moment, so 2 loads could see 2 different values in theory with unpredictable consequences. Issue exactly one load for consistent value across one exec.
CVE-2024-46830	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Acquire kvm->srcu when handling KVM_SET_VCPU_EVENTS Grab kvm->srcu when processing KVM_SET_VCPU_EVENTS, as KVM will forcibly leave nested VMX/SVM if SMM mode is being toggled, and leaving nested VMX reads guest memory. Note, kvm_vcpu_ioctl_x86_set_vcpu_events() can also be called from KVM_RUN via sync_regs(), which already holds SRCU. I.e. trying to precisely use kvm_vcpu_srcu_read_lock() around the problematic SMM code would cause problems. Acquiring SRCU isn't all that expensive, so for simplicity, grab it unconditionally for KVM_SET_VCPU_EVENTS. ===== WARNING: suspicious RCU usage 6.10.0-rc7-332d2c1d713e-next-vm #552 Not tainted ----- include/linux/kvm_host.h:1027 suspicious rcu_dereference_check() usage! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by repro/1071: #0: ffff88811e424430 (&vcpu->mutex){+..-}-{3:3}, at: kvm_vcpu_ioctl+0x7d/0x970 [kvm] stack backtrace: CPU: 15 PID: 1071 Comm: repro Not tainted 6.10.0-rc7-332d2c1d713e-next-vm #552 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 0.0.0 02/06/2015 Call Trace: <TASK> dump_stack_lvl+0x7f/0x90 lockdep_rcu_suspicious+0x13f/0x1a0 kvm_vcpu_gfn_to_memslot+0x168/0x190 [kvm] kvm_vcpu_read_guest+0x3e/0x90 [kvm] nested_vmx_load_msr+0x6b/0x1d0 [kvm_intel] load_vmcs12_host_state+0x432/0xb40 [kvm_intel] vmx_leave_nested+0x30/0x40 [kvm_intel] kvm_vcpu_ioctl_x86_set_vcpu_events+0x15d/0x2b0 [kvm] kvm_arch_vcpu_ioctl+0x1107/0x1750 [kvm] ? mark_held_locks+0x49/0x70 ? kvm_vcpu_ioctl+0x7d/0x970 [kvm] ? kvm_vcpu_ioctl+0x497/0x970 [kvm] kvm_vcpu_ioctl+0x497/0x970 [kvm] ? lock_acquire+0xba/0x2d0 ? find_held_lock+0x2b/0x80 ? do_user_addr_fault+0x40c/0x6f0 ? lock_release+0xb7/0x270 __x64_sys_ioctl+0x82/0xb0 do_syscall_64+0x6c/0x170 entry_SYSCALL_64_after_hwframe+0x4b/0x53 RIP: 0033:0x7ff11eb1b539 </TASK>

CVE ID	Description
CVE-2024-46833	In the Linux kernel, the following vulnerability has been resolved: net: hns3: void array out of bound when loop tnl_num When query reg inf of SSU, it loops tnl_num times. However, tnl_num comes from hardware and the length of array is a fixed value. To void array out of bound, make sure the loop time is not greater than the length of array
CVE-2024-46834	In the Linux kernel, the following vulnerability has been resolved: ethtool: fail closed if we can't get max channel used in indirection tables Commit 0d1b7d6c9274 ("bnxt: fix crashes when reducing ring count with active RSS contexts") proves that allowing indirection table to contain channels with out of bounds IDs may lead to crashes. Currently the max channel check in the core gets skipped if driver can't fetch the indirection table or when we can't allocate memory. Both of those conditions should be extremely rare but if they do happen we should try to be safe and fail the channel change.
CVE-2024-46835	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix smatch static checker warning adev->gfx.imu.funcs could be NULL
CVE-2024-46842	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Handle mailbox timeouts in lpfc_get_sfp_info The MBX_TIMEOUT return code is not handled in lpfc_get_sfp_info and the routine unconditionally frees submitted mailbox commands regardless of return status. The issue is that for MBX_TIMEOUT cases, when firmware returns SFP information at a later time, that same mailbox memory region references previously freed memory in its cmpl routine. Fix by adding checks for the MBX_TIMEOUT return code. During mailbox resource cleanup, check the mbox flag to make sure that the wait did not timeout. If the MBOX_WAKE flag is not set, then do not free the resources because it will be freed when firmware completes the mailbox at a later time in its cmpl routine. Also, increase the timeout from 30 to 60 seconds to accommodate boot scripts requiring longer timeouts.
CVE-2024-46845	In the Linux kernel, the following vulnerability has been resolved: tracing/timerlat: Only clear timer if a kthread exists The timerlat tracer can use user space threads to check for osnoise and timer latency. If the program using this is killed via a SIGTERM, the threads are shutdown one at a time and another tracing instance can start up resetting the threads before they are fully closed. That causes the hrtimer assigned to the kthread to be shutdown and freed twice when the dying thread finally closes the file descriptors, causing a use-after-free bug. Only cancel the hrtimer if the associated thread is still around. Also add the interface_lock around the resetting of the dlat_var->kthread. Note, this is just a quick fix that can be backported to stable. A real fix is to have a better synchronization between the shutdown of old threads and the starting of new ones.
CVE-2024-46855	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_socket: fix sk refcount leaks We must put 'sk' reference before returning.
CVE-2024-46857	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix bridge mode operations when there are no VFs Currently, trying to set the bridge mode attribute when numvfs=0 leads to a crash: bridge link set dev eth2 hwmode vepa [168.967392] BUG: kernel NULL pointer dereference, address: 0000000000000030 [...] [168.969989] RIP: 0010:mlx5_add_flow_rules+0x1f/0x300 [mlx5_core] [...] [168.976037] Call Trace: [168.976188] <TASK> [168.978620] _mlx5_eswitch_set_vepa_locked +0x113/0x230 [mlx5_core] [168.979074] mlx5_eswitch_set_vepa+0x7f/0xa0 [mlx5_core] [168.979471] rtnl_bridge_setlink+0xe9/0x1f0 [168.979714] rtnetlink_rcv_msg+0x159/0x400 [168.980451] netlink_rcv_skb+0x54/0x100 [168.980675] netlink_unicast+0x241/0x360 [168.980918] netlink_sendmsg+0x1f6/0x430 [168.981162] ____sys_sendmsg+0x3bb/0x3f0 [168.982155] ____sys_sendmsg+0x88/0xd0 [168.985036] __sys_sendmsg+0x59/0xa0 [168.985477] do_syscall_64+0x79/0x150 [168.987273] entry_SYSCALL_64_after_hwframe+0x76/0x7e [168.987773] RIP: 0033:0x7f8f7950f917 (esw->fdb_table.legacy.vepa_fdb is null) The bridge mode is only relevant when there are multiple functions per port. Therefore, prevent setting and getting this setting when there are no VFs. Note that after this change, there are no settings to change on the PF interface using 'bridge link' when there are no VFs, so the interface no longer appears in the 'bridge link' output.
CVE-2024-46859	In the Linux kernel, the following vulnerability has been resolved: platform/x86: panasonic-laptop: Fix SINF array out of bounds accesses The panasonic laptop code in various places uses the SINF array with index values of 0 - SINF_CUR_BRIGHT(0x0d) without checking that the SINF array is big enough. Not all panasonic laptops have this many SINF array entries, for example the Toughbook CF-18 model only has 10 SINF array entries. So it only supports the AC+DC brightness entries and mute. Check that the SINF array has a minimum size which covers all AC+DC brightness entries and refuse to load if the SINF array is smaller. For higher SINF indexes hide the sysfs attributes when the SINF array does not contain an entry for that attribute, avoiding show()/store() accessing the array out of bounds and add bounds checking to the probe() and resume() code accessing these.
CVE-2024-46860	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7921: fix NULL pointer access in mt7921_ipv6_addr_change When disabling wifi mt7921_ipv6_addr_change() is called as a notifier. At this point mvif->phy is already NULL so we cannot use it here.

CVE ID	Description
CVE-2024-46861	In the Linux kernel, the following vulnerability has been resolved: usbnet: ipheth: do not stop RX on failing RX callback RX callbacks can fail for multiple reasons: * Payload too short * Payload formatted incorrectly (e.g. bad NCM framing) * Lack of memory None of these should cause the driver to seize up. Make such failures non-critical and continue processing further incoming URBs.
CVE-2024-46864	In the Linux kernel, the following vulnerability has been resolved: x86/hyperv: fix kexec crash due to VP assist page corruption commit 9636be85cc5b ("x86/hyperv: Fix hyperv_pcpu_input_arg handling when CPUs go online/offline") introduces a new cpuhp state for hyperv initialization. cpuhp_setup_state() returns the state number if state is CPUHP_AP_ONLINE_DYN or CPUHP_BP_PREPARE_DYN and 0 for all other states. For the hyperv case, since a new cpuhp state was introduced it would return 0. However, in hv_machine_shutdown(), the cpuhp_remove_state() call is conditioned upon "hyperv_init_cpuhp > 0". This will never be true and so hv_cpu_die() won't be called on all CPUs. This means the VP assist page won't be reset. When the kexec kernel tries to setup the VP assist page again, the hypervisor corrupts the memory region of the old VP assist page causing a panic in case the kexec kernel is using that memory elsewhere. This was originally fixed in commit dfe94d4086e4 ("x86/hyperv: Fix kexec panic/hang issues"). Get rid of hyperv_init_cpuhp entirely since we are no longer using a dynamic cpuhp state and use CPUHP_AP_HYPERV_ONLINE directly with cpuhp_remove_state().
CVE-2024-46871	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Correct the defined value for AMDGPU_DMUB_NOTIFICATION_MAX [Why & How] It actually exposes '6' types in enum dmub_notification_type. Not 5. Using smaller number to create array dmub_callback & dmub_thread_offload has potential to access item out of array bound. Fix it.
CVE-2024-47081	Requests is a HTTP library. Due to a URL parsing issue, Requests releases prior to 2.32.4 may leak .netrc credentials to third parties for specific maliciously-crafted URLs. Users should upgrade to version 2.32.4 to receive a fix. For older versions of Requests, use of the .netrc file can be disabled with `trust_env=False` on one's Requests Session.
CVE-2024-47141	In the Linux kernel, the following vulnerability has been resolved: pinmux: Use sequential access to access desc->pinmux data When two client of the same gpio call pinctrl_select_state() for the same functionality, we are seeing NULL pointer issue while accessing desc->mux_owner. Let's say two processes A, B executing in pin_request() for the same pin and process A updates the desc->mux_usecount but not yet updated the desc->mux_owner while process B see the desc->mux_usecount which got updated by A path and further executes strcmp and while accessing desc->mux_owner it crashes with NULL pointer. Serialize the access to mux related setting with a mutex lock. cpu0 (process A) cpu1 (process B) pinctrl_select_state() { pinctrl_select_state() { pin_request() { pin_request() { } else { desc->mux_usecount++; desc->mux_usecount && strcmp(desc->mux_owner, owner)) { if (desc->mux_usecount > 1) return 0; desc->mux_owner = owner; } }
CVE-2024-47143	In the Linux kernel, the following vulnerability has been resolved: dma-debug: fix a possible deadlock on radix_lock radix_lock() shouldn't be held while holding dma_hash_entry[idx].lock otherwise, there's a possible deadlock scenario when dma debug API is called holding rq_lock(): CPU0 CPU1 CPU2 dma_free_attrs() check_unmap() add_dma_entry() __schedule() //out (A) rq_lock() get_hash_bucket() (A) dma_entry_hash check_sync() (A) radix_lock() (W) dma_entry_hash dma_entry_free() (W) radix_lock() // CPU2's one (W) rq_lock() CPU1 situation can happen when it extending radix tree and it tries to wake up kswapd via wake_all_kswapd(). CPU2 situation can happen while perf_event_task_sched_out() (i.e. dma sync operation is called while deleting perf_event using etm and etr tmc which are Arm Coresight hwtracing driver backends). To remove this possible situation, call dma_entry_free() after put_hash_bucket() in check_unmap().
CVE-2024-47220	An issue was discovered in the WEbrick toolkit through 1.8.1 for Ruby. It allows HTTP request smuggling by providing both a Content-Length header and a Transfer-Encoding header, e.g., "GET / admin HTTP/1.1\r\n" inside of a "POST /user HTTP/1.1\r\n" request. NOTE: the supplier's position is "Webrick should not be used in production."
CVE-2024-47408	In the Linux kernel, the following vulnerability has been resolved: net/smc: check smcd_v2_ext_offset when receiving proposal msg When receiving proposal msg in server, the field smcd_v2_ext_offset in proposal msg is from the remote client and can not be fully trusted. Once the value of smcd_v2_ext_offset exceed the max value, there has the chance to access wrong address, and crash may happen. This patch checks the value of smcd_v2_ext_offset before using it.

CVE ID	Description
CVE-2024-4741	Issue summary: Calling the OpenSSL API function <code>SSL_free_buffers</code> may cause memory to be accessed that was previously freed in some situations Impact summary: A use after free can have a range of potential consequences such as the corruption of valid data, crashes or execution of arbitrary code. However, only applications that directly call the <code>SSL_free_buffers</code> function are affected by this issue. Applications that do not call this function are not vulnerable. Our investigations indicate that this function is rarely used by applications. The <code>SSL_free_buffers</code> function is used to free the internal OpenSSL buffer used when processing an incoming record from the network. The call is only expected to succeed if the buffer is not currently in use. However, two scenarios have been identified where the buffer is freed even when still in use. The first scenario occurs where a record header has been received from the network and processed by OpenSSL, but the full record body has not yet arrived. In this case calling <code>SSL_free_buffers</code> will succeed even though a record has only been partially processed and the buffer is still in use. The second scenario occurs where a full record containing application data has been received and processed by OpenSSL but the application has only read part of this data. Again a call to <code>SSL_free_buffers</code> will succeed even though the buffer is still in use. While these scenarios could occur accidentally during normal operation a malicious attacker could attempt to engineer a situation where this occurs. We are not aware of this issue being actively exploited. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue.
CVE-2024-47535	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. An unsafe reading of environment file could potentially cause a denial of service in Netty. When loaded on an Windows application, Netty attempts to load a file that does not exist. If an attacker creates such a large file, the Netty application crashes. This vulnerability is fixed in 4.1.115.
CVE-2024-47554	Uncontrolled Resource Consumption vulnerability in Apache Commons IO. The <code>org.apache.commons.io.input.XmlStreamReader</code> class may excessively consume CPU resources when processing maliciously crafted input. This issue affects Apache Commons IO: from 2.0 before 2.14.0. Users are recommended to upgrade to version 2.14.0 or later, which fixes the issue.
CVE-2024-47561	Schema parsing in the Java SDK of Apache Avro 1.11.3 and previous versions allows bad actors to execute arbitrary code. Users are recommended to upgrade to version 1.11.4 or 1.12.0, which fix this issue.
CVE-2024-47660	In the Linux kernel, the following vulnerability has been resolved: <code>fsnotify</code> : clear <code>PARENT_WATCHED</code> flags lazily In some setups directories can have many (usually negative) dentries. Hence <code>__fsnotify_update_child_dentry_flags()</code> function can take a significant amount of time. Since the bulk of this function happens under <code>inode->i_lock</code> this causes a significant contention on the lock when we remove the watch from the directory as the <code>__fsnotify_update_child_dentry_flags()</code> call from <code>fsnotify_recalc_mask()</code> races with <code>__fsnotify_update_child_dentry_flags()</code> calls from <code>__fsnotify_parent()</code> happening on children. This can lead upto softlockup reports reported by users. Fix the problem by calling <code>fsnotify_update_children_dentry_flags()</code> to set <code>PARENT_WATCHED</code> flags only when parent starts watching children. When parent stops watching children, clear false positive <code>PARENT_WATCHED</code> flags lazily in <code>__fsnotify_parent()</code> for each accessed child.
CVE-2024-47671	In the Linux kernel, the following vulnerability has been resolved: USB: <code>usbtmc</code> : prevent kernel-usb-infoleak The syzbot reported a kernel-usb-infoleak in <code>usbtmc_write</code> , we need to clear the structure before filling fields.
CVE-2024-47672	In the Linux kernel, the following vulnerability has been resolved: wifi: <code>iwlmwifi</code> : mvm: don't wait for tx queues if firmware is dead There is a WARNING in <code>iwl_trans_wait_tx_queues_empty()</code> (that was recently converted from just a message), that can be hit if we wait for TX queues to become empty after firmware died. Clearly, we can't expect anything from the firmware after it's declared dead. Don't call <code>iwl_trans_wait_tx_queues_empty()</code> in this case. While it could be a good idea to stop the flow earlier, the flush functions do some maintenance work that is not related to the firmware, so keep that part of the code running even when the firmware is not running. [edit commit message]
CVE-2024-47673	In the Linux kernel, the following vulnerability has been resolved: wifi: <code>iwlmwifi</code> : mvm: pause TCM when the firmware is stopped Not doing so will make us send a host command to the transport while the firmware is not alive, which will trigger a WARNING. bad state = 0 WARNING: CPU: 2 PID: 17434 at drivers/net/wireless/intel/iwlmwifi/iwl-trans.c:115 iwl_trans_send_cmd+0x1cb/0x1e0 [iwlmwifi] RIP: 0010:iwl_trans_send_cmd+0x1cb/0x1e0 [iwlmwifi] Call Trace: <TASK> iwl_mvm_send_cmd+0x40/0xc0 [iwlmmvm] iwl_mvm_config_scan+0x198/0x260 [iwlmmvm] iwl_mvm_recalc_tcm+0x730/0x11d0 [iwlmmvm] iwl_mvm_tcm_work+0x1d/0x30 [iwlmmvm] process_one_work+0x29e/0x640 worker_thread+0x2df/0x690 ? rescuer_thread+0x540/0x540 kthread+0x192/0x1e0 ? set_kthread_struct+0x90/0x90 ret_from_fork+0x22/0x30

CVE ID	Description
CVE-2024-47674	In the Linux kernel, the following vulnerability has been resolved: mm: avoid leaving partial pfn mappings around in error case As Jann points out, PFN mappings are special, because unlike normal memory mappings, there is no lifetime information associated with the mapping - it is just a raw mapping of PFNs with no reference counting of a 'struct page'. That's all very much intentional, but it does mean that it's easy to mess up the cleanup in case of errors. Yes, a failed mmap() will always eventually clean up any partial mappings, but without any explicit lifetime in the page table mapping itself, it's very easy to do the error handling in the wrong order. In particular, it's easy to mistakenly free the physical backing store before the page tables are actually cleaned up and (temporarily) have stale dangling PTE entries. To make this situation less error-prone, just make sure that any partial pfn mapping is torn down early, before any other error handling.
CVE-2024-47678	In the Linux kernel, the following vulnerability has been resolved: icmp: change the order of rate limits ICMP messages are ratelimited : After the blamed commits, the two rate limiters are applied in this order: 1) host wide ratelimit (icmp_global_allow()) 2) Per destination ratelimit (inetpeer based) In order to avoid side-channels attacks, we need to apply the per destination check first. This patch makes the following change : 1) icmp_global_allow() checks if the host wide limit is reached. But credits are not yet consumed. This is deferred to 3) 2) The per destination limit is checked/updated. This might add a new node in inetpeer tree. 3) icmp_global_consume() consumes tokens if prior operations succeeded. This means that host wide ratelimit is still effective in keeping inetpeer tree small even under DDOS. As a bonus, I removed icmp_global.lock as the fast path can use a lock-free operation.
CVE-2024-47679	In the Linux kernel, the following vulnerability has been resolved: vfs: fix race between evict_inodes() and find_inode()&iput() Hi, all Recently I noticed a bug[1] in btrfs, after digged it into and I believe it's a race in vfs. Let's assume there's a inode (ie ino 261) with i_count 1 is called by iput(), and there's a concurrent thread calling generic_shutdown_super(). cpu0: cpu1: iput() // i_count is 1 ->spin_lock(inode) ->dec i_count to 0 ->iput_final() generic_shutdown_super() ->__inode_add_lru() ->evict_inodes() // cause some reason[2] ->if (atomic_read(inode->i_count)) continue; // return before // inode 261 passed the above check // list_lru_add_obj() // and then schedule out ->spin_unlock() // note here: the inode 261 // was still at sb list and hash list, // and I_FREEING I_WILL_FREE was not been set btrfs_iget() // after some function calls ->find_inode() // found the above inode 261 ->spin_lock(inode) // check I_FREEING I_WILL_FREE // and passed ->__iget() ->spin_unlock(inode) // schedule back ->spin_lock(inode) // check (I_NEW I_FREEING I_WILL_FREE) flags, // passed and set I_FREEING iput() ->spin_unlock(inode) ->spin_lock(inode) ->evict() // dec i_count to 0 ->iput_final() ->spin_unlock() ->evict() Now, we have two threads simultaneously evicting the same inode, which may trigger the BUG(inode->i_state & I_CLEAR) statement both within clear_inode() and iput(). To fix the bug, recheck the inode->i_count after holding i_lock. Because in the most scenarios, the first check is valid, and the overhead of spin_lock() can be reduced. If there is any misunderstanding, please let me know, thanks. [1]: https://lore.kernel.org/linux-btrfs/000000000000eabe1d0619c48986@google.com/ [2]: The reason might be 1. SB_ACTIVE was removed or 2. mapping_shrinkable() return false when I reproduced the bug.

CVE ID	Description
CVE-2024-47684	In the Linux kernel, the following vulnerability has been resolved: tcp: check skb is non-NULL in tcp_rto_delta_us() We have some machines running stock Ubuntu 20.04.6 which is their 5.4.0-174-generic kernel that are running ceph and recently hit a null ptr dereference in tcp_rearm_rto(). Initially hitting it from the TLP path, but then later we also saw it getting hit from the RACK case as well. Here are examples of the oops messages we saw in each of those cases: Jul 26 15:05:02 rx [11061395.780353] BUG: kernel NULL pointer dereference, address: 0000000000000020 Jul 26 15:05:02 rx [11061395.787572] #PF: supervisor read access in kernel mode Jul 26 15:05:02 rx [11061395.792971] #PF: error_code(0x0000) - not-present page Jul 26 15:05:02 rx [11061395.798362] PGD 0 P4D 0 Jul 26 15:05:02 rx [11061395.801164] Oops: 0000 [#1] SMP NOPTI Jul 26 15:05:02 rx [11061395.805091] CPU: 0 PID: 9180 Comm: msgr-worker-1 Tainted: G W 5.4.0-174-generic #193-Ubuntu Jul 26 15:05:02 rx [11061395.814996] Hardware name: Supermicro SMC 2x26 os-gen8 64C NVME-Y 256G/H12SSW-NTR, BIOS 2.5.V1.2U.NVMe.UEFI 05/09/2023 Jul 26 15:05:02 rx [11061395.825952] RIP: 0010:tcp_rearm_rto+0xe4/0x160 Jul 26 15:05:02 rx [11061395.830656] Code: 87 ca 04 00 00 00 5b 41 5c 41 5d 5d c3 c3 49 8b bc 24 40 06 00 00 eb 8d 48 bb cf f7 53 e3 a5 9b c4 20 4c 89 ef e8 0c fe 0e 00 <48> 8b 78 20 48 c1 ef 03 48 89 f8 41 8b bc 24 80 04 00 00 48 f7 e3 Jul 26 15:05:02 rx [11061395.849665] RSP: 0018:ffffb75d40003e08 EFLAGS: 00010246 Jul 26 15:05:02 rx [11061395.855149] RAX: 0000000000000000 RBX: 20c49ba5e353f7cf RCX: 0000000000000000 Jul 26 15:05:02 rx [11061395.862542] RDX: 0000000062177c30 RSI: 0000000000000231c RDI: ffff9874ad283a60 Jul 26 15:05:02 rx [11061395.869933] RBP: ffff75d40003e20 R08: 0000000000000000 R09: ffff987605e20aa8 Jul 26 15:05:02 rx [11061395.877318] R10: ffff75d40003f00 R11: ffff75d4460f740 R12: ffff9874ad283900 Jul 26 15:05:02 rx [11061395.884710] R13: ffff9874ad283a60 R14: ffff9874ad283980 R15: ffff9874ad283d30 Jul 26 15:05:02 rx [11061395.892095] FS: 00007f1ef4a2e700(0000) GS:ffff987605e00000(0000) knlGS:0000000000000000 Jul 26 15:05:02 rx [11061395.900438] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 Jul 26 15:05:02 rx [11061395.906435] CR2: 0000000000000020 CR3: 00000003e450ba003 CR4: 0000000000760ef0 Jul 26 15:05:02 rx [11061395.913822] PKRU: 55555554 Jul 26 15:05:02 rx [11061395.916786] Call Trace: Jul 26 15:05:02 rx [11061395.919488] Jul 26 15:05:02 rx [11061395.921765] ? show_regs.cold+0x1a/0x1f Jul 26 15:05:02 rx [11061395.925859] ? __die+0x90/0xd9 Jul 26 15:05:02 rx [11061395.929169] ? no_context+0x196/0x380 Jul 26 15:05:02 rx [11061395.933088] ? ip6_protocol_deliver_rcu+0x4e0/0x4e0 Jul 26 15:05:02 rx [11061395.938216] ? ip6_sublist_rcv_finish+0x3d/0x50 Jul 26 15:05:02 rx [11061395.943000] ? __bad_area_nosemaphore+0x50/0x1a0 Jul 26 15:05:02 rx [11061395.947873] ? bad_area_nosemaphore+0x16/0x20 Jul 26 15:05:02 rx [11061395.952486] ? do_user_addr_fault+0x267/0x450 Jul 26 15:05:02 rx [11061395.957104] ? ipv6_list_rcv+0x112/0x140 Jul 26 15:05:02 rx [11061395.961279] ? __do_page_fault+0x58/0x90 Jul 26 15:05:02 rx [11061395.965458] ? do_page_fault+0x2c/0xe0 Jul 26 15:05:02 rx [11061395.969465] ? page_fault+0x34/0x40 Jul 26 15:05:02 rx [11061395.973217] ? tcp_rearm_rto+0xe4/0x160 Jul 26 15:05:02 rx [11061395.977313] ? tcp_rearm_rto+0xe4/0x160 Jul 26 15:05:02 rx [11061395.981408] tcp_send_loss_probe+0x10b/0x220 Jul 26 15:05:02 rx [11061395.985937] tcp_write_timer_handler+0x1b4/0x240 Jul 26 15:05:02 rx [11061395.990809] tcp_write_timer+0x9e/0xe0 Jul 26 15:05:02 rx [11061395.994814] ? tcp_write_timer_handler+0x240/0x240 Jul 26 15:05:02 rx [11061395.999866] call_timer_fn+0x32/0x130 Jul 26 15:05:02 rx [11061396.003782] __run_timers.part.0+0x180/0x280 Jul 26 15:05:02 rx [11061396.008309] ? recalibrate_cpu_khz+0x10/0x10 Jul 26 15:05:02 rx [11061396.012841] ? native_x2apic_icr_write+0x30/0x30 Jul 26 15:05:02 rx [11061396.017718] ? lapic_next_even ---truncated---

CVE ID	Description
CVE-2024-47685	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_reject_ipv6: fix nf_reject_ip6_tchdr_put() syzbot reported that nf_reject_ip6_tchdr_put() was possibly sending garbage on the four reserved tcp bits (th->res1) Use skb_put_zero() to clear the whole TCP header, as done in nf_reject_ip_tchdr_put() BUG: KMSAN: uninit-value in nf_reject_ip6_tchdr_put +0x688/0x6c0 net/ipv6/netfilter/nf_reject_ipv6.c:255 nf_reject_ip6_tchdr_put+0x688/0x6c0 net/ipv6/netfilter/nf_reject_ipv6.c:255 nf_send_reset6+0xd84/0x15b0 net/ipv6/netfilter/nf_reject_ipv6.c:344 nft_reject_inet_eval+0x3c1/0x880 net/netfilter/nft_reject_inet.c:48 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x438/0x22a0 net/netfilter/nf_tables_core.c:288 nft_do_chain_inet+0x41a/0x4f0 net/netfilter/nft_chain_filter.c:161 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] NF_HOOK include/linux/netfilter.h:312 [inline] ipv6_rcv+0x29b/0x390 net/ipv6/ip6_input.c:310 __netif_receive_skb_one_core net/core/dev.c:5661 [inline] __netif_receive_skb +0x1da/0xa00 net/core/dev.c:5775 process_backlog+0x4ad/0xa50 net/core/dev.c:6108 __napi_poll +0xe7/0x980 net/core/dev.c:6772 napi_poll net/core/dev.c:6841 [inline] net_rx_action+0xa5a/0x19b0 net/core/dev.c:6963 handle_softirqs+0x1ce/0x800 kernel/softirq.c:554 __do_softirq+0x14/0x1a kernel/softirq.c:588 do_softirq+0x9a/0x100 kernel/softirq.c:455 __local_bh_enable_ip+0x9f/0xb0 kernel/softirq.c:382 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:908 [inline] __dev_queue_xmit+0x2692/0x5610 net/core/dev.c:4450 dev_queue_xmit include/linux/netdevice.h:3105 [inline] neigh_resolve_output+0x9ca/0xae0 net/core/neighbour.c:1565 neigh_output include/net/neighbour.h:542 [inline] ip6_finish_output2+0x2347/0x2ba0 net/ipv6/ip6_output.c:141 __ip6_finish_output net/ipv6/ip6_output.c:215 [inline] ip6_finish_output +0xbb8/0x14b0 net/ipv6/ip6_output.c:226 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip6_output+0x356/0x620 net/ipv6/ip6_output.c:247 dst_output include/net/dst.h:450 [inline] NF_HOOK include/linux/netfilter.h:314 [inline] ip6_xmit+0x1ba6/0x25d0 net/ipv6/ip6_output.c:366 inet6_csk_xmit +0x442/0x530 net/ipv6/inet6_connection_sock.c:135 __tcp_transmit_skb+0x3b07/0x4880 net/ipv4/tcp_output.c:1466 tcp_transmit_skb net/ipv4/tcp_output.c:1484 [inline] tcp_connect +0x35b6/0x7130 net/ipv4/tcp_output.c:4143 tcp_v6_connect+0x1bcc/0x1e40 net/ipv6/tcp_ipv6.c:333 __inet_stream_connect+0x2ef/0x1730 net/ipv4/af_inet.c:679 inet_stream_connect+0x6a/0xd0 net/ipv4/af_inet.c:750 __sys_connect_file net/socket.c:2061 [inline] __sys_connect+0x606/0x690 net/socket.c:2078 __do_sys_connect net/socket.c:2088 [inline] __se_sys_connect net/socket.c:2085 [inline] __x64_sys_connect+0x91/0xe0 net/socket.c:2085 x64_sys_call+0x27a5/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:43 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe +0x77/0x7f Uninit was stored to memory at: nf_reject_ip6_tchdr_put+0x60c/0x6c0 net/ipv6/netfilter/nf_reject_ipv6.c:249 nf_send_reset6+0xd84/0x15b0 net/ipv6/netfilter/nf_reject_ipv6.c:344 nft_reject_inet_eval+0x3c1/0x880 net/netfilter/nft_reject_inet.c:48 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x438/0x22a0 net/netfilter/nf_tables_core.c:288 nft_do_chain_inet+0x41a/0x4f0 net/netfilter/nft_chain_filter.c:161 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] NF_HOOK include/linux/netfilter.h:312 [inline] ipv6_rcv+0x29b/0x390 net/ipv6/ip6_input.c:310 __netif_receive_skb_one_core ---truncated---

CVE ID	Description
CVE-2024-47692	In the Linux kernel, the following vulnerability has been resolved: nfsd: return -EINVAL when namelen is 0 When we have a corrupted main.sqlite in /var/lib/nfs/nfsdclld/, it may result in namelen being 0, which will cause memdup_user() to return ZERO_SIZE_PTR. When we access the name.data that has been assigned the value of ZERO_SIZE_PTR in nfs4_client_to_reclaim(), null pointer dereference is triggered. [T1205] ===== [T1205] BUG: KASAN: null-ptr-deref in nfs4_client_to_reclaim+0xe9/0x260 [T1205] Read of size 1 at addr 0000000000000010 by task nfsdclld/1205 [T1205] [T1205] CPU: 11 PID: 1205 Comm: nfsdclld Not tainted 5.10.0-00003-g2c1423731b8d #406 [T1205] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS ?-20190727_073836-buildvm-ppc64le-16.ppc.fedoraproject.org-3.fc31 04/01/2014 [T1205] Call Trace: [T1205] dump_stack+0x9a/0xd0 [T1205] ? nfs4_client_to_reclaim+0xe9/0x260 [T1205] __kasan_report.cold+0x34/0x84 [T1205] ? nfs4_client_to_reclaim+0xe9/0x260 [T1205] kasan_report+0x3a/0x50 [T1205] nfs4_client_to_reclaim+0xe9/0x260 [T1205] ? nfsd4_release_lockowner+0x410/0x410 [T1205] cld_pipe_downcall+0x5ca/0x760 [T1205] ? nfsd4_cld_tracking_exit+0x1d0/0x1d0 [T1205] ? down_write_killable_nested+0x170/0x170 [T1205] ? avc_policy_seqno+0x28/0x40 [T1205] ? selinux_file_permission+0x1b4/0x1e0 [T1205] rpc_pipe_write+0x84/0xb0 [T1205] vfs_write+0x143/0x520 [T1205] ksys_write+0xc9/0x170 [T1205] ? __ia32_sys_read+0x50/0x50 [T1205] ? ktime_get_coarse_real_ts64+0xfe/0x110 [T1205] ? ktime_get_coarse_real_ts64+0xa2/0x110 [T1205] do_syscall_64+0x33/0x40 [T1205] entry_SYSCALL_64_after_hwframe+0x67/0xd1 [T1205] RIP: 0033:0x7fdbb761bc7 [T1205] Code: 0f 00 f7 d8 64 89 02 48 c7 c0 ff ff ff ff eb b7 0f 1f 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 514 [T1205] RSP: 002b:00007fff8c4b7248 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 [T1205] RAX: ffffffff8c4b7248 RBX: 000000000000042b RCX: 00007fdbb761bc7 [T1205] RDX: 000000000000042b RSI: 00007fff8c4b75f0 RDI: 0000000000000008 [T1205] RBP: 00007fdbb761bb0 R08: 0000000000000000 R09: 0000000000000001 [T1205] R10: 0000000000000000 R11: 0000000000000246 R12: 000000000000042b [T1205] R13: 0000000000000008 R14: 00007fff8c4b75f0 R15: 0000000000000000 [T1205] ===== Fix it by checking namelen.
CVE-2024-47693	In the Linux kernel, the following vulnerability has been resolved: IB/core: Fix ib_cache_setup_one error flow cleanup When ib_cache_update return an error, we exit ib_cache_setup_one instantly with no proper cleanup, even though before this we had already successfully done gid_table_setup_one, that results in the kernel WARN below. Do proper cleanup using gid_table_cleanup_one before returning the err in order to fix the issue. WARNING: CPU: 4 PID: 922 at drivers/infiniband/core/cache.c:806 gid_table_release_one+0x181/0x1a0 Modules linked in: CPU: 4 UID: 0 PID: 922 Comm: c_repro Not tainted 6.11.0-rc1+ #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:gid_table_release_one+0x181/0x1a0 Code: 44 8b 38 75 0c e8 2f cb 34 ff 4d 8b b5 28 05 00 00 e8 23 cb 34 ff 44 89 f9 89 da 4c 89 f6 48 c7 c7 d0 58 14 83 e8 4f de 21 ff <0f> 0b 4c 8b 75 30 e9 54 ff ff ff 48 8 3 c4 10 5b 5d 41 5c 41 5d 41 RSP: 0018:ffff90002b835b0 EFLAGS: 00010286 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffffff811c8527 RDX: 0000000000000000 RSI: ffffffff811c8534 RDI: 0000000000000001 RBP: ffff8881011b3d00 R08: ffff88810b3abe00 R09: 205d303839303631 R10: 666572207972746e R11: 72746e6520444947 R12: 0000000000000001 R13: ffff888106390000 R14: ffff8881011f2110 R15: 0000000000000001 FS: 00007fccc3b70800(0000) GS: ffff88813bd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000340 CR3: 000000010435a001 CR4: 0000000003706b0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? show_regs+0x94/0xa0 ? __warn+0x9e/0x1c0 ? gid_table_release_one+0x181/0x1a0 ? report_bug+0x1f9/0x340 ? gid_table_release_one+0x181/0x1a0 ? handle_bug+0xa2/0x110 ? exc_invalid_op+0x31/0xa0 ? asm_exc_invalid_op+0x16/0x20 ? __warn_printk+0xc7/0x180 ? __warn_printk+0xd4/0x180 ? gid_table_release_one+0x181/0x1a0 ib_device_release+0x71/0xe0 ? __pfx_ib_device_release+0x10/0x10 device_release+0x44/0xd0 kobject_put+0x135/0x3d0 put_device+0x20/0x30 rxe_net_add+0x7d/0xa0 rxe_newlink+0xd7/0x190 nldev_newlink+0x1b0/0x2a0 ? __pfx_nldev_newlink+0x10/0x10 rdma_nl_rcv_msg+0x1ad/0x2e0 rdma_nl_rcv_skb.constprop.0+0x176/0x210 netlink_unicast+0x2de/0x400 netlink_sendmsg+0x306/0x660 __sock_sendmsg+0x110/0x120 __sys_sendmsg+0x30e/0x390 __sys_sendmsg+0x9b/0xf0 ? kstrtoint+0x6e/0xa0 ? kstrtoint_from_user+0x7c/0xb0 ? get_pid_task+0xb0/0xd0 ? proc_fail_nth_write+0x5b/0x140 ? __fget_light+0x9a/0x200 ? preempt_count_add+0x47/0xa0 __sys_sendmsg+0x61/0xd0 do_syscall_64+0x50/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2024-47697	In the Linux kernel, the following vulnerability has been resolved: drivers: media: dvb-frontends/rtl2830: fix an out-of-bounds write error Ensure index in rtl2830_pid_filter does not exceed 31 to prevent out-of-bounds access. dev->filters is a 32-bit value, so set_bit and clear_bit functions should only operate on indices from 0 to 31. If index is 32, it will attempt to access a non-existent 33rd bit, leading to out-of-bounds access. Change the boundary check from index > 32 to index >= 32 to resolve this issue.

CVE ID	Description
CVE-2024-47698	In the Linux kernel, the following vulnerability has been resolved: drivers: media: dvb-frontends/rtl2832: fix an out-of-bounds write error Ensure index in rtl2832_pid_filter does not exceed 31 to prevent out-of-bounds access. dev->filters is a 32-bit value, so set_bit and clear_bit functions should only operate on indices from 0 to 31. If index is 32, it will attempt to access a non-existent 33rd bit, leading to out-of-bounds access. Change the boundary check from index > 32 to index >= 32 to resolve this issue. [hverkuil: added fixes tag, rtl2830_pid_filter -> rtl2832_pid_filter in logmsg]
CVE-2024-47701	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid OOB when system.data xattr changes underneath the filesystem When looking up for an entry in an inlined directory, if e_value_offs is changed underneath the filesystem by some change in the block device, it will lead to an out-of-bounds access that KASAN detects as an UAF. EXT4-fs (loop0): mounted filesystem 00000000-0000-0000-0000-000000000000 r/w without journal. Quota mode: none. loop0: detected capacity change from 2048 to 2047 ===== BUG: KASAN: use-after-free in ext4_search_dir+0xf2/0x1c0 fs/ext4/namei.c:1500 Read of size 1 at addr ffff88803e91130f by task syz-executor269/5103 CPU: 0 UID: 0 PID: 5103 Comm: syz-executor269 Not tainted 6.11.0-rc4-syzkaller #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 ext4_search_dir+0xf2/0x1c0 fs/ext4/namei.c:1500 ext4_find_inline_entry+0x4be/0x5e0 fs/ext4/inline.c:1697 __ext4_find_entry+0x2b4/0x1b30 fs/ext4/namei.c:1573 ext4_lookup_entry fs/ext4/namei.c:1727 [inline] ext4_lookup+0x15f/0x750 fs/ext4/namei.c:1795 lookup_one_qstr_excl+0x11f/0x260 fs/namei.c:1633 filename_create+0x297/0x540 fs/namei.c:3980 do_symlinkat+0xf9/0x3a0 fs/namei.c:4587 __do_sys_symlinkat fs/namei.c:4610 [inline] __se_sys_symlinkat fs/namei.c:4607 [inline] __x64_sys_symlinkat+0x95/0xb0 fs/namei.c:4607 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0xf3e73ced469 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 21 18 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b8 ff ff f7 d8 64 89 01 48 RSP: 002b:00007fff4d40c258 EFLAGS: 00000246 ORIG_RAX: 000000000000010a RAX: ffffffff8da RBX: 0032656c69662f2e RCX: 00007f3e73ced469 RDX: 0000000020000200 RSI: 00000000ffffff9c RDI: 00000000200001c0 RBP: 0000000000000000 R08: 00007fff4d40c290 R09: 00007fff4d40c290 R10: 0023706f6f6c2f76 R11: 0000000000000246 R12: 00007fff4d40c27c R13: 0000000000000003 R14: 431bde82d7b634db R15: 00007fff4d40c2b0 </TASK> Calling ext4_xattr_ibody_find right after reading the inode with ext4_get_inode_loc will lead to a check of the validity of the xattrs, avoiding this problem.
CVE-2024-47703	In the Linux kernel, the following vulnerability has been resolved: bpf, lsm: Add check for BPF LSM return value A bpf prog returning a positive number attached to file_alloc_security hook makes kernel panic. This happens because file system can not filter out the positive number returned by the LSM prog using IS_ERR, and misinterprets this positive number as a file pointer. Given that hook file_alloc_security never returned positive number before the introduction of BPF LSM, and other BPF LSM hooks may encounter similar issues, this patch adds LSM return value check in verifier, to ensure no unexpected value is returned.
CVE-2024-47705	In the Linux kernel, the following vulnerability has been resolved: block: fix potential invalid pointer dereference in blk_add_partition The blk_add_partition() function initially used a single if-condition (IS_ERR(part)) to check for errors when adding a partition. This was modified to handle the specific case of -ENXIO separately, allowing the function to proceed without logging the error in this case. However, this change unintentionally left a path where md_autodetect_dev() could be called without confirming that part is a valid pointer. This commit separates the error handling logic by splitting the initial if-condition, improving code readability and handling specific error scenarios explicitly. The function now distinguishes the general error case from -ENXIO without altering the existing behavior of md_autodetect_dev() calls.

CVE ID	Description
CVE-2024-47706	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix possible UAF for bfqq->bic with merge chain 1) initial state, three tasks: Process 1 Process 2 Process 3 (BIC1) (BIC2) (BIC3) # # # V V V bfqq1 bfqq2 bfqq3 process ref: 1 1 1 2) bfqq1 merged to bfqq2: Process 1 Process 2 Process 3 (BIC1) (BIC2) (BIC3) # \-----\ V bfqq1----->bfqq2 bfqq3 process ref: 0 2 1 3) bfqq2 merged to bfqq3: Process 1 Process 2 Process 3 (BIC1) (BIC2) (BIC3) here -> # \-----\ \-----\ V V bfqq1----->bfqq2----->bfqq3 process ref: 0 1 3 In this case, IO from Process 1 will get bfqq2 from BIC1 first, and then get bfqq3 through merge chain, and finally handle IO by bfqq3. However, current code will think bfqq2 is owned by BIC1, like initial state, and set bfqq2->bic to BIC1. bfq_insert_request -> by Process 1 bfqq = bfq_init_rq(rq) bfqq = bfq_get_bfqq_handle_split bfqq = bic_to_bfqq -> get bfqq2 from BIC1 bfqq->ref++ rq->elv.priv[0] = bic rq->elv.priv[1] = bfqq if (bfqq_process_refs(bfqq) == 1) bfqq->bic = bic -> record BIC1 to bfqq2 __bfq_insert_request new_bfqq = bfq_setup_cooperator -> get bfqq3 from bfqq2->new_bfqq bfqq_request_freed(bfqq) new_bfqq->ref++ rq->elv.priv[1] = new_bfqq -> handle IO by bfqq3 Fix the problem by checking bfqq is from merge chain fist. And this might fix a following problem reported by our syzkaller(unreproducible): ===== BUG: KASAN: slab-use-after-free in bfq_do_early_stable_merge block/bfq-iosched.c:5692 [inline] BUG: KASAN: slab-use-after-free in bfq_do_or_sched_stable_merge block/bfq-iosched.c:5805 [inline] BUG: KASAN: slab-use-after-free in bfq_get_queue+0x25b0/0x2610 block/bfq-iosched.c:5889 Write of size 1 at addr ffff888123839eb8 by task kworker/0:1H/18595 CPU: 0 PID: 18595 Comm: kworker/0:1H Tainted: G L 6.6.0-07439-gba2303cacfda #6 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014 Workqueue: kblockd blk_mq_requeue_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x91/0xf0 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:364 [inline] print_report+0x10d/0x610 mm/kasan/report.c:475 kasan_report +0x8e/0xc0 mm/kasan/report.c:588 bfq_do_early_stable_merge block/bfq-iosched.c:5692 [inline] bfq_do_or_sched_stable_merge block/bfq-iosched.c:5805 [inline] bfq_get_queue+0x25b0/0x2610 block/bfq-iosched.c:5889 bfq_get_bfqq_handle_split+0x169/0x5d0 block/bfq-iosched.c:6757 bfq_init_rq block/bfq-iosched.c:6876 [inline] bfq_insert_request block/bfq-iosched.c:6254 [inline] bfq_insert_requests+0x1112/0x5cf0 block/bfq-iosched.c:6304 blk_mq_insert_request+0x290/0x8d0 block/blk-mq.c:2593 blk_mq_requeue_work+0x6bc/0xa70 block/blk-mq.c:1502 process_one_work kernel/workqueue.c:2627 [inline] process_scheduled_works+0x432/0x13f0 kernel/workqueue.c:2700 worker_thread+0x6f2/0x1160 kernel/workqueue.c:2781 kthread+0x33c/0x440 kernel/kthread.c:388 ret_from_fork+0x4d/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1b/0x30 arch/x86/entry/entry_64.S:305 </TASK> Allocated by task 20776: kasan_save_stack+0x20/0x40 mm/kasan/common.c:45 kasan_set_track+0x25/0x30 mm/kasan/common.c:52 __kasan_slab_alloc+0x87/0x90 mm/kasan/common.c:328 kasan_slab_alloc include/linux/kasan.h:188 [inline] slab_post_alloc_hook mm/slab.h:763 [inline] slab_alloc_node mm/slub.c:3458 [inline] kmem_cache_alloc_node+0x1a4/0x6f0 mm/slub.c:3503 ioc_create_icq block/blk-ioc.c:370 [inline] ---truncated---

CVE ID	Description
CVE-2024-47707	In the Linux kernel, the following vulnerability has been resolved: ipv6: avoid possible NULL deref in rt6_uncached_list_flush_dev() Blamed commit accidentally removed a check for rt->rt6i_iddev being NULL, as spotted by syzbot: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] CPU: 1 UID: 0 PID: 10998 Comm: syz-executor Not tainted 6.11.0-rc6-syzkaller-00208-g625403177711 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 RIP: 0010:rt6_uncached_list_flush_dev net/ipv6/route.c:177 [inline] RIP: 0010:rt6_disable_ip+0x33e/0x7e0 net/ipv6/route.c:4914 Code: 41 80 3c 04 00 74 0a e8 90 d0 9b f7 48 8b 7c 24 08 48 8b 07 48 89 44 24 10 4c 89 f0 48 c1 e8 03 48 b9 00 00 00 00 00 fc ff df <80> 3c 08 00 74 08 4c 89 f7 e8 64 d0 9b f7 48 8b 44 24 18 49 39 06 RSP: 0018:ffff900047374e0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 1ffff1100fd8f33 RCX: dffffc0000000000 RDX: 0000000000000000 RSI: 0000000000000004 RDI: ffff88807efc78c0 RBP: ffff900047375d0 R08: 0000000000000003 R09: ffff520008e6e8c R10: dffffc0000000000 R11: ffff520008e6e8c R12: 1ffff1100fd8f818 R13: ffff88807efc7998 R14: 0000000000000000 R15: ffff88807efc7930 FS: 0000000000000000(0000) GS:ffff8880b8900000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020002a80 CR3: 0000000022f62000 CR4: 0000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> addrconf_ifdown+0x15d/0x1bd0 net/ipv6/addrconf.c:3856 addrconf_notify+0x3cb/0x1020 notifier_call_chain+0x19f/0x3e0 kernel/notifier.c:93 call_netdevice_notifiers_extack net/core/dev.c:2032 [inline] call_netdevice_notifiers net/core/dev.c:2046 [inline] unregister_netdevice_many_notify+0xd81/0x1c40 net/core/dev.c:11352 unregister_netdevice_many net/core/dev.c:11414 [inline] unregister_netdevice_queue+0x303/0x370 net/core/dev.c:11289 unregister_netdevice include/linux/netdevice.h:3129 [inline] __tun_detach +0x6b9/0x1600 drivers/net/tun.c:685 tun_detach drivers/net/tun.c:701 [inline] tun_chr_close +0x108/0x1b0 drivers/net/tun.c:3510 __fput+0x24a/0x8a0 fs/file_table.c:422 task_work_run +0x24f/0x310 kernel/task_work.c:228 exit_task_work include/linux/task_work.h:40 [inline] do_exit +0xa2f/0x27f0 kernel/exit.c:882 do_group_exit+0x207/0x2c0 kernel/exit.c:1031 __do_sys_exit_group kernel/exit.c:1042 [inline] __se_sys_exit_group kernel/exit.c:1040 [inline] __x64_sys_exit_group +0x3f/0x40 kernel/exit.c:1040 x64_sys_call+0x2634/0x2640 arch/x86/include/generated/asm/syscalls_64.h:232 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f1acc77def9 Code: Unable to access opcode bytes at 0x7f1acc77decf. RSP: 002b:00007ffeb26fa738 EFLAGS: 00000246 ORIG_RAX: 000000000000000e RAX: ffffffffda RBX: 0000000000000000 RCX: 00007f1acc77def9 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000043 RBP: 00007f1acc77d508 R08: 00007ffeb26f84d7 R09: 0000000000000003 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000001 R13: 0000000000000003 R14: 00000000ffffff R15: 00007ffeb26fa8e0 </TASK> Modules linked in: ---[end trace 0000000000000000]--- RIP: 0010:rt6_uncached_list_flush_dev net/ipv6/route.c:177 [inline] RIP: 0010:rt6_disable_ip+0x33e/0x7e0 net/ipv6/route.c:4914 Code: 41 80 3c 04 00 74 0a e8 90 d0 9b f7 48 8b 7c 24 08 48 8b 07 48 89 44 24 10 4c 89 f0 48 c1 e8 03 48 b9 00 00 00 00 00 fc ff df <80> 3c 08 00 74 08 4c 89 f7 e8 64 d0 9b f7 48 8b 44 24 18 49 39 06 RSP: 0018:ffff900047374e0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 1ffff1100fd8f33 RCX: dffffc0000000000 RDX: 0000000000000000 RSI: 0000000000000004 RDI: ffff88807efc78c0 R ---truncated---
CVE-2024-47710	In the Linux kernel, the following vulnerability has been resolved: sock_map: Add a cond_resched() in sock_hash_free() Several syzbot soft lockup reports all have in common sock_hash_free() If a map with a large number of buckets is destroyed, we need to yield the cpu when needed.
CVE-2024-47713	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: use two-phase skb reclamation in ieee80211_do_stop() Since '__dev_queue_xmit()' should be called with interrupts enabled, the following backtrace: ieee80211_do_stop() ... spin_lock_irqsave(&local->queue_stop_reason_lock, flags) ... ieee80211_free_txskb() ieee80211_report_used_skb() ieee80211_report_ack_skb() cfg80211_mgmt_tx_status_ext() nl80211_frame_tx_status() genlmsg_multicast_netns() genlmsg_multicast_netns_filtered() nlmsg_multicast_filtered() netlink_broadcast_filtered() do_one_broadcast() netlink_broadcast_deliver() __netlink_sendskb() netlink_deliver_tap() __netlink_deliver_tap_skb() dev_queue_xmit() __dev_queue_xmit() ; with IRQS disabled ... spin_unlock_irqrestore(&local->queue_stop_reason_lock, flags) issues the warning (as reported by syzbot reproducer): WARNING: CPU: 2 PID: 5128 at kernel/softirq.c:362 __local_bh_enable_ip+0xc3/0x120 Fix this by implementing a two-phase skb reclamation in 'ieee80211_do_stop()', where actual work is performed outside of a section with interrupts disabled.
CVE-2024-47718	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: always wait for both firmware loading attempts In 'rtw_wait_firmware_completion()', always wait for both (regular and wowlan) firmware loading attempts. Otherwise if 'rtw_usb_intf_init()' has failed in 'rtw_usb_probe()', 'rtw_usb_disconnect()' may issue 'ieee80211_free_hw()' when one of 'rtw_load_firmware_cb()' (usually the wowlan one) is still in progress, causing UAF detected by KASAN.

CVE ID	Description
CVE-2024-47726	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to wait dio completion It should wait all existing dio write IOs before block removal, otherwise, previous direct write IO may overwrite data in the block which may be reused by other inode.
CVE-2024-47727	In the Linux kernel, the following vulnerability has been resolved: x86/tdx: Fix "in-kernel MMIO" check TDX only supports kernel-initiated MMIO operations. The handle_mmio() function checks if the #VE exception occurred in the kernel and rejects the operation if it did not. However, userspace can deceive the kernel into performing MMIO on its behalf. For example, if userspace can point a syscall to an MMIO address, syscall does get_user() or put_user() on it, triggering MMIO #VE. The kernel will treat the #VE as in-kernel MMIO. Ensure that the target MMIO address is within the kernel before decoding instruction.
CVE-2024-47728	In the Linux kernel, the following vulnerability has been resolved: bpf: Zero former ARG_PTR_TO_{LONG,INT} args in case of error For all non-tracing helpers which formerly had ARG_PTR_TO_{LONG,INT} as input arguments, zero the value for the case of an error as otherwise it could leak memory. For tracing, it is not needed given CAP_PERFMON can already read all kernel memory anyway hence bpf_get_func_arg() and bpf_get_func_ret() is skipped in here. Also, the MTU helpers mtu_len pointer value is being written but also read. Technically, the MEM_UNINIT should not be there in order to always force init. Removing MEM_UNINIT needs more verifier rework though: MEM_UNINIT right now implies two things actually: i) write into memory, ii) memory does not have to be initialized. If we lift MEM_UNINIT, it then becomes: i) read into memory, ii) memory must be initialized. This means that for bpf_*_check_mtu() we're readdressing the issue we're trying to fix, that is, it would then be able to write back into things like .rodata BPF maps. Follow-up work will rework the MEM_UNINIT semantics such that the intent can be better expressed. For now just clear the *mtu_len on error path which can be lifted later again.
CVE-2024-47729	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Use reserved copy engine for user binds on faulting devices User binds map to engines with can fault, faults depend on user binds completion, thus we can deadlock. Avoid this by using reserved copy engine for user binds on faulting devices. While we are here, normalize bind queue creation with a helper. v2: - Pass in extensions to bind queue creation (CI) v3: - s/resevered/reserved (Lucas) - Fix NULL hwe check (Jonathan)
CVE-2024-47736	In the Linux kernel, the following vulnerability has been resolved: erofs: handle overlapped pclusters out of crafted images properly syzbot reported a task hang issue due to a deadlock case where it is waiting for the folio lock of a cached folio that will be used for cache I/Os. After looking into the crafted fuzzed image, I found it's formed with several overlapped big pclusters as below: Ext: logical offset length : physical offset length 0: 0.. 16384 16384 : 151552.. 167936 16384 1: 16384.. 32768 16384 : 155648.. 172032 16384 2: 32768.. 49152 16384 : 537223168.. 537239552 16384 ... Here, extent 0/1 are physically overlapped although it's entirely _impossible_ for normal filesystem images generated by mkfs. First, managed folios containing compressed data will be marked as up-to-date and then unlocked immediately (unlike in-place folios) when compressed I/Os are complete. If physical blocks are not submitted in the incremental order, there should be separate BIOs to avoid dependency issues. However, the current code mis-arranges z_erofs_fill_bio_vec() and BIO submission which causes unexpected BIO waits. Second, managed folios will be connected to their own pclusters for efficient inter-queries. However, this is somewhat hard to implement easily if overlapped big pclusters exist. Again, these only appear in fuzzed images so let's simply fall back to temporary short-lived pages for correctness. Additionally, it justifies that referenced managed folios cannot be truncated for now and reverts part of commit 2080ca1ed3e4 ("erofs: tidy up `struct z_erofs_bvec`") for simplicity although it shouldn't be any difference.
CVE-2024-47737	In the Linux kernel, the following vulnerability has been resolved: nfsd: call cache_put if xdr_reserve_space returns NULL If not enough buffer space available, but idmap_lookup has triggered lookup_fn which calls cache_get and returns successfully. Then we missed to call cache_put here which pairs with cache_get. Reviewed-by: Jeff Layton <jlayton@kernel.org>
CVE-2024-47738	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: don't use rate mask for offchannel TX either Like the commit ab9177d83c04 ("wifi: mac80211: don't use rate mask for scanning"), ignore incorrect settings to avoid no supported rate warning reported by syzbot. The syzbot did bisect and found cause is commit 9df66d5b9f45 ("cfg80211: fix default HE tx bitrate mask in 2G band"), which however corrects bitmask of HE MCS and recognizes correctly settings of empty legacy rate plus HE MCS rate instead of returning -EINVAL. As suggestions [1], follow the change of SCAN TX to consider this case of offchannel TX as well. [1] https://lore.kernel.org/linux-wireless/6ab2dc9c3afe753ca6fdcd1421e7a1f47e87b84.camel@sipsolutions.net/T/#m2ac2a6d2be06a37c9c47a3d8a44b4f647ed4f024

CVE ID	Description
CVE-2024-47739	In the Linux kernel, the following vulnerability has been resolved: padata: use integer wrap around to prevent deadlock on seq_nr overflow When submitting more than 2^32 padata objects to padata_do_serial, the current sorting implementation incorrectly sorts padata objects with overflowed seq_nr, causing them to be placed before existing objects in the reorder list. This leads to a deadlock in the serialization process as padata_find_next cannot match padata->seq_nr and pd->processed because the padata instance with overflowed seq_nr will be selected next. To fix this, we use an unsigned integer wrap around to correctly sort padata objects in scenarios with integer overflow.
CVE-2024-47745	In the Linux kernel, the following vulnerability has been resolved: mm: call the security_mmap_file() LSM hook in remap_file_pages() The remap_file_pages syscall handler calls do_mmap() directly, which doesn't contain the LSM security check. And if the process has called personality(READ_IMPLIES_EXEC) before and remap_file_pages() is called for RW pages, this will actually result in remapping the pages to RWX, bypassing a W^X policy enforced by SELinux. So we should check prot by security_mmap_file LSM hook in the remap_file_pages syscall handler before do_mmap() is called. Otherwise, it potentially permits an attacker to bypass a W^X policy enforced by SELinux. The bypass is similar to CVE-2016-10044, which bypass the same thing via AIO and can be found in [1]. The PoC: \$ cat > test.c int main(void) { size_t pagesz = sysconf(_SC_PAGE_SIZE); int mfd = syscall(SYS_memfd_create, "test", 0); const char *buf = mmap(NULL, 4 * pagesz, PROT_READ PROT_WRITE, MAP_SHARED, mfd, 0); unsigned int old = syscall(SYS_personality, 0xffffffff); syscall(SYS_personality, READ_IMPLIES_EXEC old); syscall(SYS_remap_file_pages, buf, pagesz, 0, 2, 0); syscall(SYS_personality, old); // show the RWX page exists even if W^X policy is enforced int fd = open("/proc/self/maps", O_RDONLY); unsigned char buf2[1024]; while (1) { int ret = read(fd, buf2, 1024); if (ret <= 0) break; write(1, buf2, ret); } close(fd); } \$ gcc test.c -o test \$./test grep rwx 7f1836c34000-7f1836c35000 rwx 00002000 00:01 2050 /memfd:test (deleted) [PM: subject line tweaks]
CVE-2024-47748	In the Linux kernel, the following vulnerability has been resolved: vhost_vdpa: assign irq bypass producer token correctly We used to call irq_bypass_unregister_producer() in vhost_vdpa_setup_vq_irq() which is problematic as we don't know if the token pointer is still valid or not. Actually, we use the eventfd_ctx as the token so the life cycle of the token should be bound to the VHOST_SET_VRING_CALL instead of vhost_vdpa_setup_vq_irq() which could be called by set_status(). Fixing this by setting up irq bypass producer's token when handling VHOST_SET_VRING_CALL and un-registering the producer before calling vhost_vring_ioctl() to prevent a possible use after free as eventfd could have been released in vhost_vring_ioctl(). And such registering and unregistering will only be done if DRIVER_OK is set.
CVE-2024-47749	In the Linux kernel, the following vulnerability has been resolved: RDMA/cxgb4: Added NULL check for lookup_atid The lookup_atid() function can return NULL if the ATID is invalid or does not exist in the identifier table, which could lead to dereferencing a null pointer without a check in the 'act_establish()' and 'act_open_rpl()' functions. Add a NULL check to prevent null pointer dereferencing. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-47794	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent tailcall infinite loop caused by freplace There is a potential infinite loop issue that can occur when using a combination of tail calls and freplace. In an upcoming selftest, the attach target for entry_freplace of tailcall_freplace.c is subprog_tc of tc_bpf2bpf.c, while the tail call in entry_freplace leads to entry_tc. This results in an infinite loop: entry_tc -> subprog_tc -> entry_freplace --tailcall-> entry_tc. The problem arises because the tail_call_cnt in entry_freplace resets to zero each time entry_freplace is executed, causing the tail call mechanism to never terminate, eventually leading to a kernel panic. To fix this issue, the solution is twofold: 1. Prevent updating a program extended by an freplace program to a prog_array map. 2. Prevent extending a program that is already part of a prog_array map with an freplace program. This ensures that: * If a program or its subprogram has been extended by an freplace program, it can no longer be updated to a prog_array map. * If a program has been added to a prog_array map, neither it nor its subprograms can be extended by an freplace program. Moreover, an extension program should not be tailcalled. As such, return -EINVAL if the program has a type of BPF_PROG_TYPE_EXT when adding it to a prog_array map. Additionally, fix a minor code style issue by replacing eight spaces with a tab for proper formatting.
CVE-2024-47809	In the Linux kernel, the following vulnerability has been resolved: dlm: fix possible lkb_resource null dereference This patch fixes a possible null pointer dereference when this function is called from request_lock() as lkb->lkb_resource is not assigned yet, only after validate_lock_args() by calling attach_lkb(). Another issue is that a resource name could be a non printable bytearray and we cannot assume to be ASCII coded. The log functionality is probably never being hit when DLM is used in normal way and no debug logging is enabled. The null pointer dereference can only occur on a new created lkb that does not have the resource assigned yet, it probably never hits the null pointer dereference but we should be sure that other changes might not change this behaviour and we actually can hit the mentioned null pointer dereference. In this patch we just drop the printout of the resource name, the lkb id is enough to make a possible connection to a resource name if this exists.

CVE ID	Description
CVE-2024-47814	Vim is an open source, command line text editor. A use-after-free was found in Vim < 9.1.0764. When closing a buffer (visible in a window) a BufWinLeave auto command can cause an use-after-free if this auto command happens to re-open the same buffer in a new split window. Impact is low since the user must have intentionally set up such a strange auto command and run some buffer unload commands. However this may lead to a crash. This issue has been addressed in version 9.1.0764 and all users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2024-47874	Starlette is an Asynchronous Server Gateway Interface (ASGI) framework/toolkit. Prior to version 0.40.0, Starlette treats `multipart/form-data` parts without a `filename` as text form fields and buffers those in byte strings with no size limit. This allows an attacker to upload arbitrary large form fields and cause Starlette to both slow down significantly due to excessive memory allocations and copy operations, and also consume more and more memory until the server starts swapping and grinds to a halt, or the OS terminates the server process with an OOM error. Uploading multiple such requests in parallel may be enough to render a service practically unusable, even if reasonable request size limits are enforced by a reverse proxy in front of Starlette. This Denial of service (DoS) vulnerability affects all applications built with Starlette (or FastAPI) accepting form requests. Verison 0.40.0 fixes this issue.
CVE-2024-48873	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: check return value of ieee80211_probereq_get() for RNR The return value of ieee80211_probereq_get() might be NULL, so check it before using to avoid NULL pointer access. Addresses-Coverity-ID: 1529805 ("Dereference null return value")
CVE-2024-48875	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't take dev_replace rwsem on task already holding it Running fstests btrfs/011 with MKFS_OPTIONS="-O rst" to force the usage of the RAID stripe-tree, we get the following splat from lockdep: BTRFS info (device sdd): dev_replace from /dev/sdd (devid 1) to /dev/sdb started ===== WARNING: possible recursive locking detected 6.11.0-rc3-btrfs-for-next #599 Not tainted ----- btrfs/2326 is trying to acquire lock: ffff88810f215c98 (&fs_info->dev_replace.rwsem){++++}-{3:3}, at: btrfs_map_block+0x39f/0x2250 but task is already holding lock: ffff88810f215c98 (&fs_info->dev_replace.rwsem){++++}-{3:3}, at: btrfs_map_block+0x39f/0x2250 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(&fs_info->dev_replace.rwsem); lock(&fs_info->dev_replace.rwsem); *** DEADLOCK *** May be due to missing lock nesting notation 1 lock held by btrfs/2326: #0: ffff88810f215c98 (&fs_info->dev_replace.rwsem){++++}-{3:3}, at: btrfs_map_block+0x39f/0x2250 stack backtrace: CPU: 1 UID: 0 PID: 2326 Comm: btrfs Not tainted 6.11.0-rc3-btrfs-for-next #599 Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011 Call Trace: <TASK> dump_stack_lvl+0x5b/0x80 __lock_acquire+0x2798/0x69d0 ? __pfx__lock_acquire+0x10/0x10 ? __pfx__lock_acquire+0x10/0x10 lock_acquire+0x19d/0x4a0 ? btrfs_map_block+0x39f/0x2250 ? __pfx_lock_acquire+0x10/0x10 ? find_held_lock+0x2d/0x110 ? lock_is_held_type+0x8f/0x100 down_read+0x8e/0x440 ? btrfs_map_block+0x39f/0x2250 ? __pfx_down_read+0x10/0x10 ? do_raw_read_unlock+0x44/0x70 ? _raw_read_unlock+0x23/0x40 btrfs_map_block+0x39f/0x2250 ? btrfs_dev_replace_by_ioctl+0xd69/0x1d00 ? btrfs_bio_counter_inc_blocked+0xd9/0x2e0 ? __kasan_slab_alloc+0x6e/0x70 ? __pfx_btrfs_map_block+0x10/0x10 ? __pfx_btrfs_bio_counter_inc_blocked+0x10/0x10 ? kmem_cache_alloc_noprof+0x1f2/0x300 ? mempool_alloc_noprof+0xed/0x2b0 btrfs_submit_chunk+0x28d/0x17e0 ? __pfx_btrfs_submit_chunk+0x10/0x10 ? bvec_alloc+0xd7/0x1b0 ? bio_add_folio+0x171/0x270 ? __pfx_bio_add_folio+0x10/0x10 ? __kasan_check_read+0x20/0x20 btrfs_submit_bio+0x37/0x80 read_extent_buffer_pages+0x3df/0x6c0 btrfs_read_extent_buffer+0x13e/0x5f0 read_tree_block+0x81/0xe0 read_block_for_search+0x4bd/0x7a0 ? __pfx_read_block_for_search+0x10/0x10 btrfs_search_slot+0x78d/0x2720 ? __pfx_btrfs_search_slot+0x10/0x10 ? lock_is_held_type+0x8f/0x100 ? kasan_save_track+0x14/0x30 ? __kasan_slab_alloc+0x6e/0x70 ? kmem_cache_alloc_noprof+0x1f2/0x300 btrfs_get_raid_extent_offset+0x181/0x820 ? __pfx_lock_acquire+0x10/0x10 ? __pfx_btrfs_get_raid_extent_offset+0x10/0x10 ? down_read+0x194/0x440 ? __pfx_down_read+0x10/0x10 ? do_raw_read_unlock+0x44/0x70 ? _raw_read_unlock+0x23/0x40 btrfs_map_block+0x5b5/0x2250 ? __pfx_btrfs_map_block+0x10/0x10 scrub_submit_initial_read+0x8fe/0x11b0 ? __pfx_scrub_submit_initial_read+0x10/0x10 submit_initial_group_read+0x161/0x3a0 ? lock_release+0x20e/0x710 ? __pfx_submit_initial_group_read+0x10/0x10 ? __pfx_lock_release+0x10/0x10 scrub_simple_mirror.isra.0+0x3eb/0x580 scrub_stripe+0xe4d/0x1440 ? lock_release+0x20e/0x710 ? __pfx_scrub_stripe+0x10/0x10 ? __pfx_lock_release+0x10/0x10 ? do_raw_read_unlock+0x44/0x70 ? _raw_read_unlock+0x23/0x40 scrub_chunk+0x257/0x4a0 scrub_enumerate_chunks+0x64c/0xf70 ? __mutex_unlock_slowpath+0x147/0x5f0 ? __pfx_scrub_enumerate_chunks+0x10/0x10 ? bit_wait_timeout+0xb0/0x170 ? __up_read+0x189/0x700 ? scrub_workers_get+0x231/0x300 ? up_write+0x490/0x4f0 btrfs_scrub_dev+0x52e/0xcd0 ? create_pending_snapshots+0x230/0x250 ? __pfx_btrfs_scrub_dev+0x10/0x10 btrfs_dev_replace_by_ioctl+0xd69/0x1d00 ? lock_acquire+0x19d/0x4a0 ? __pfx_btrfs_dev_replace_by_ioctl+0x10/0x10 ? ---truncated---

CVE ID	Description
CVE-2024-48876	In the Linux kernel, the following vulnerability has been resolved: stackdepot: fix stack_depot_save_flags() in NMI context Per documentation, stack_depot_save_flags() was meant to be usable from NMI context if STACK_DEPOT_FLAG_CAN_ALLOC is unset. However, it still would try to take the pool_lock in an attempt to save a stack trace in the current pool (if space is available). This could result in deadlock if an NMI is handled while pool_lock is already held. To avoid deadlock, only try to take the lock in NMI context and give up if unsuccessful. The documentation is fixed to clearly convey this.
CVE-2024-48881	In the Linux kernel, the following vulnerability has been resolved: bcache: revert replacing IS_ERR_OR_NULL with IS_ERR again Commit 028ddcac477b ("bcache: Remove unnecessary NULL point check in node allocations") leads a NULL pointer dereference in cache_set_flush(). 1721 if (!IS_ERR_OR_NULL(c->root)) 1722 list_add(&c->root->list, &c->btree_cache); >From the above code in cache_set_flush(), if previous registration code fails before allocating c->root, it is possible c->root is NULL as what it is initialized. __bch_btree_node_alloc() never returns NULL but c->root is possible to be NULL at above line 1721. This patch replaces IS_ERR() by IS_ERR_OR_NULL() to fix this.
CVE-2024-49568	In the Linux kernel, the following vulnerability has been resolved: net/smc: check v2_ext_offset/eid_cnt/ism_gid_cnt when receiving proposal msg When receiving proposal msg in server, the fields v2_ext_offset/ eid_cnt/ism_gid_cnt in proposal msg are from the remote client and can not be fully trusted. Especially the field v2_ext_offset, once exceed the max value, there has the chance to access wrong address, and crash may happen. This patch checks the fields v2_ext_offset/eid_cnt/ism_gid_cnt before using them.
CVE-2024-49569	In the Linux kernel, the following vulnerability has been resolved: nvme-rdma: unquiesce admin_q before destroy it Kernel will hang on destroy admin_q while we create ctrl failed, such as following calltrace: PID: 23644 TASK: ff2d52b40f439fc0 CPU: 2 COMMAND: "nvme" #0 [ff61d23de260fb78] __schedule at ffffffff8323bc15 #1 [ff61d23de260fc08] schedule at ffffffff8323c014 #2 [ff61d23de260fc28] blk_mq_freeze_queue_wait at ffffffff82a3dba1 #3 [ff61d23de260fc78] blk_freeze_queue at ffffffff82a4113a #4 [ff61d23de260fc90] blk_cleanup_queue at ffffffff82a33006 #5 [ff61d23de260fc0] nvme_rdma_destroy_admin_queue at ffffffff12686ce #6 [ff61d23de260fcc8] nvme_rdma_setup_ctrl at ffffffff1268ced #7 [ff61d23de260fd28] nvme_rdma_create_ctrl at ffffffff126919b #8 [ff61d23de260fd68] nvme_dev_write at ffffffff024f362 #9 [ff61d23de260fe38] vfs_write at ffffffff827d5f25 RIP: 00007fda7891d574 RSP: 00007ffe2ef06958 RFLAGS: 00000202 RAX: ffffffff827d5f25 RBX: 000055e8122a4d90 RCX: 00007fda7891d574 RDX: 000000000000012b RSI: 000055e8122a4d90 RDI: 0000000000000004 RBP: 00007ffe2ef079c0 R8: 000000000000012b R9: 000055e8122a4d90 R10: 0000000000000000 R11: 0000000000000202 R12: 0000000000000004 R13: 000055e8122923c0 R14: 000000000000012b R15: 00007fda78a54500 ORIG_RAX: 0000000000000001 CS: 0033 SS: 002b This due to we have quiesced admin_q before cancel requests, but forgot to unquiesce before destroy it, as a result we fail to drain the pending requests, and hang on blk_mq_freeze_queue_wait() forever. Here try to reuse nvme_rdma_teardown_admin_queue() to fix this issue and simplify the code.
CVE-2024-49571	In the Linux kernel, the following vulnerability has been resolved: net/smc: check iparea_offset and ipv6_prefixes_cnt when receiving proposal msg When receiving proposal msg in server, the field iparea_offset and the field ipv6_prefixes_cnt in proposal msg are from the remote client and can not be fully trusted. Especially the field iparea_offset, once exceed the max value, there has the chance to access wrong address, and crash may happen. This patch checks iparea_offset and ipv6_prefixes_cnt before using them.
CVE-2024-49750	The Snowflake Connector for Python provides an interface for developing Python applications that can connect to Snowflake and perform all standard operations. Prior to version 3.12.3, when the logging level was set by the user to DEBUG, the Connector could have logged Duo passcodes (when specified via the 'passcode' parameter) and Azure SAS tokens. Additionally, the SecretDetector logging formatter, if enabled, contained bugs which caused it to not fully redact JWT tokens and certain private key formats. Snowflake released version 3.12.3 of the Snowflake Connector for Python, which fixes the issue. In addition to upgrading, users should review their logs for any potentially sensitive information that may have been captured.
CVE-2024-49761	REXML is an XML toolkit for Ruby. The REXML gem before 3.3.9 has a ReDoS vulnerability when it parses an XML that has many digits between &# and x...; in a hex numeric character reference (&#x...;). This does not happen with Ruby 3.2 or later. Ruby 3.1 is the only affected maintained Ruby. The REXML gem 3.3.9 or later include the patch to fix the vulnerability.
CVE-2024-49766	Werkzeug is a Web Server Gateway Interface web application library. On Python < 3.11 on Windows, os.path.isabs() does not catch UNC paths like //server/share. Werkzeug's safe_join() relies on this check, and so can produce a path that is not safe, potentially allowing unintended access to data. Applications using Python >= 3.11, or not using Windows, are not vulnerable. Werkzeug version 3.0.6 contains a patch.

CVE ID	Description
CVE-2024-49767	Werkzeug is a Web Server Gateway Interface web application library. Applications using <code>`werkzeug.formparser.MultiPartParser`</code> corresponding to a version of Werkzeug prior to 3.0.6 to parse <code>`multipart/form-data`</code> requests (e.g. all flask applications) are vulnerable to a relatively simple but effective resource exhaustion (denial of service) attack. A specifically crafted form submission request can cause the parser to allocate and block 3 to 8 times the upload size in main memory. There is no upper limit; a single upload at 1 Gbit/s can exhaust 32 GB of RAM in less than 60 seconds. Werkzeug version 3.0.6 fixes this issue.
CVE-2024-49768	Waitress is a Web Server Gateway Interface server for Python 2 and 3. A remote client may send a request that is exactly <code>recv_bytes</code> (defaults to 8192) long, followed by a secondary request using HTTP pipelining. When request lookahead is disabled (default) we won't read any more requests, and when the first request fails due to a parsing error, we simply close the connection. However when request lookahead is enabled, it is possible to process and receive the first request, start sending the error message back to the client while we read the next request and queue it. This will allow the secondary request to be serviced by the worker thread while the connection should be closed. Waitress 3.0.1 fixes the race condition. As a workaround, disable <code>channel_request_lookahead</code> , this is set to 0 by default disabling this feature.
CVE-2024-49769	Waitress is a Web Server Gateway Interface server for Python 2 and 3. When a remote client closes the connection before waitress has had the opportunity to call <code>getpeername()</code> waitress won't correctly clean up the connection leading to the main thread attempting to write to a socket that no longer exists, but not removing it from the list of sockets to attempt to process. This leads to a busy-loop calling the write function. A remote attacker could run waitress out of available sockets with very little resources required. Waitress 3.0.1 contains fixes that remove the race condition.
CVE-2024-49851	In the Linux kernel, the following vulnerability has been resolved: <code>tpm</code> : Clean up TPM space after command failure <code>tpm_dev_transmit</code> prepares the TPM space before attempting command transmission. However if the command fails no rollback of this preparation is done. This can result in transient handles being leaked if the device is subsequently closed with no further commands performed. Fix this by flushing the space in the event of command transmission failure.
CVE-2024-49856	In the Linux kernel, the following vulnerability has been resolved: <code>x86/sgx</code> : Fix deadlock in SGX NUMA node search When the current node doesn't have an EPC section configured by firmware and all other EPC sections are used up, CPU can get stuck inside the while loop that looks for an available EPC page from remote nodes indefinitely, leading to a soft lockup. Note how <code>nid_of_current</code> will never be equal to <code>nid</code> in that while loop because <code>nid_of_current</code> is not set in <code>sgx_numa_mask</code> . Also worth mentioning is that it's perfectly fine for the firmware not to setup an EPC section on a node. While setting up an EPC section on each node can enhance performance, it is not a requirement for functionality. Rework the loop to start and end on <code>*a*</code> node that has SGX memory. This avoids the deadlock looking for the current SGX-lacking node to show up in the loop when it never will.
CVE-2024-49858	In the Linux kernel, the following vulnerability has been resolved: <code>efistub/tpm</code> : Use ACPI reclaim memory for event log to avoid corruption The TPM event log table is a Linux specific construct, where the data produced by the <code>GetEventLog()</code> boot service is cached in memory, and passed on to the OS using an EFI configuration table. The use of <code>EFI_LOADER_DATA</code> here results in the region being left unreserved in the E820 memory map constructed by the EFI stub, and this is the memory description that is passed on to the incoming kernel by <code>kexec</code> , which is therefore unaware that the region should be reserved. Even though the utility of the TPM2 event log after a <code>kexec</code> is questionable, any corruption might send the parsing code off into the weeds and crash the kernel. So let's use <code>EFI_ACPI_RECLAIM_MEMORY</code> instead, which is always treated as reserved by the E820 conversion logic.
CVE-2024-49860	In the Linux kernel, the following vulnerability has been resolved: <code>ACPI: sysfs</code> : validate return type of <code>_STR</code> method Only buffer objects are valid return values of <code>_STR</code> . If something else is returned <code>description_show()</code> will access invalid memory.

CVE ID	Description
CVE-2024-49861	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix helper writes to read-only maps Lonia found an issue that despite user- and BPF-side frozen BPF map (like in case of .rodata), it was still possible to write into it from a BPF program side through specific helpers having ARG_PTR_TO_{LONG,INT} as arguments. In check_func_arg() when the argument is as mentioned, the meta->raw_mode is never set. Later, check_helper_mem_access(), under the case of PTR_TO_MAP_VALUE as register base type, it assumes BPF_READ for the subsequent call to check_map_access_type() and given the BPF map is read-only it succeeds. The helpers really need to be annotated as ARG_PTR_TO_{LONG,INT} MEM_UNINIT when results are written into them as opposed to read out of them. The latter indicates that it's okay to pass a pointer to uninitialized memory as the memory is written to anyway. However, ARG_PTR_TO_{LONG,INT} is a special case of ARG_PTR_TO_FIXED_SIZE_MEM just with additional alignment requirement. So it is better to just get rid of the ARG_PTR_TO_{LONG,INT} special cases altogether and reuse the fixed size memory types. For this, add MEM_ALIGNED to additionally ensure alignment given these helpers write directly into the args via *ptr = val. The .arg_size has been initialized reflecting the actual sizeof(*ptr). MEM_ALIGNED can only be used in combination with MEM_FIXED_SIZE annotated argument types, since in !MEM_FIXED_SIZE cases the verifier does not know the buffer size a priori and therefore cannot blindly write *ptr = val.
CVE-2024-49866	In the Linux kernel, the following vulnerability has been resolved: tracing/timerlat: Fix a race during cpuhp processing There is another found exception that the "timerlat/1" thread was scheduled on CPU0, and lead to timer corruption finally: `` ODEBUG: init active (active state 0) object: ffff888237c2e108 object type: hrtimer hint: timerlat_irq+0x0/0x220 WARNING: CPU: 0 PID: 426 at lib/debugobjects.c:518 debug_print_object+0x7d/0xb0 Modules linked in: CPU: 0 UID: 0 PID: 426 Comm: timerlat/1 Not tainted 6.11.0-rc7+ #45 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 RIP: 0010:debug_print_object+0x7d/0xb0 ... Call Trace: <TASK> ? __warn+0x7c/0x110 ? debug_print_object+0x7d/0xb0 ? report_bug+0xf1/0x1d0 ? prb_read_valid+0x17/0x20 ? handle_bug+0x3f/0x70 ? exc_invalid_op+0x13/0x60 ? asm_exc_invalid_op+0x16/0x20 ? debug_print_object+0x7d/0xb0 ? debug_print_object+0x7d/0xb0 ? __pfx_timerlat_irq+0x10/0x10 __debug_object_init+0x110/0x150 hrtimer_init+0x1d/0x60 timerlat_main+0xab/0x2d0 ? __pfx_timerlat_main+0x10/0x10 kthread+0xb7/0xe0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2d/0x40 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> `` After tracing the scheduling event, it was discovered that the migration of the "timerlat/1" thread was performed during thread creation. Further analysis confirmed that it is because the CPU online processing for osnoise is implemented through workers, which is asynchronous with the offline processing. When the worker was scheduled to create a thread, the CPU may have already been removed from the cpu_online_mask during the offline process, resulting in the inability to select the right CPU: T1 T2 [CPUHP_ONLINE] cpu_device_down() osnoise_hotplug_workfn() cpus_write_lock() takedown_cpu(1) cpus_write_unlock() [CPUHP_OFFLINE] cpus_read_lock() start_kthread(1) cpus_read_unlock() To fix this, skip online processing if the CPU is already offline.

CVE ID	Description
CVE-2024-49875	In the Linux kernel, the following vulnerability has been resolved: nfsd: map the EBADMSG to nfserr_io to avoid warning Ext4 will throw -EBADMSG through ext4_readdir when a checksum error occurs, resulting in the following WARNING. Fix it by mapping EBADMSG to nfserr_io. nfsd_buffered_readdir iterate_dir // -EBADMSG -74 ext4_readdir // .iterate_shared ext4_dx_readdir ext4_htrie_fill_tree htrie_dirblock_to_tree ext4_read_dirblock __ext4_read_dirblock ext4_dirblock_csum_verify warn_no_space_for_csum __warn_no_space_for_csum return ERR_PTR(-EFSBADCRC) // -EBADMSG -74 nfsermo // WARNING [161.115610] ----- [cut here]----- [161.116465] nfsd: non-standard errno: -74 [161.117315] WARNING: CPU: 1 PID: 780 at fs/nfsd/nfsproc.c:878 nfsermo+0x9d/0xd0 [161.118596] Modules linked in: [161.119243] CPU: 1 PID: 780 Comm: nfsd Not tainted 5.10.0-00014-g79679361fd5d #138 [161.120684] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qe mu.org 04/01/2014 [161.123601] RIP: 0010:nfsermo+0x9d/0xd0 [161.124676] Code: 0f 87 da 30 dd 00 83 e3 01 b8 00 00 00 05 75 d7 44 89 ee 48 c7 c7 c0 57 24 98 89 44 24 04 c6 05 ce 2b 61 03 01 e8 99 20 d8 00 <0f> 0b 8b 44 24 04 eb b5 4c 89 e6 48 c7 c7 a0 6d a4 99 e8 cc 15 33 [161.127797] RSP: 0018:ffff90000e2f9c0 EFLAGS: 00010286 [161.128794] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 [161.130089] RDX: 1ffff1103ee16f6d RSI: 0000000000000008 RDI: fffff520001c5f2a [161.131379] RBP: 0000000000000022 R08: 0000000000000001 R09: ffff8881f70c1827 [161.132664] R10: ffffd103ee18304 R11: 0000000000000001 R12: 0000000000000021 [161.133949] R13: 00000000fffffb6 R14: ffff8881317c0000 R15: ffff90000e2fbd8 [161.135244] FS: 0000000000000000(0000) GS:ffff8881f7080000(0000) knlGS:0000000000000000 [161.136695] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [161.137761] CR2: 00007fcaad70b348 CR3: 0000000144256006 CR4: 0000000000770ee0 [161.139041] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [161.140291] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [161.141519] PKRU: 55555554 [161.142076] Call Trace: [161.142575] ? __warn+0x9b/0x140 [161.143229] ? nfsermo+0x9d/0xd0 [161.143872] ? report_bug+0x125/0x150 [161.144595] ? handle_bug+0x41/0x90 [161.145284] ? exc_invalid_op+0x14/0x70 [161.146009] ? asm_exc_invalid_op+0x12/0x20 [161.146816] ? nfsermo+0x9d/0xd0 [161.147487] nfsd_buffered_readdir+0x28b/0x2b0 [161.148333] ? nfsd4_encode_dirent_fattr+0x380/0x380 [161.149258] ? nfsd_buffered_filldir+0xf0/0xf0 [161.150093] ? wait_for_concurrent_writes+0x170/0x170 [161.151004] ? generic_file_llseek_size+0x48/0x160 [161.151895] nfsd_readdir+0x132/0x190 [161.152606] ? nfsd4_encode_dirent_fattr+0x380/0x380 [161.153516] ? nfsd_unlink+0x380/0x380 [161.154256] ? override_creds+0x45/0x60 [161.155006] nfsd4_encode_readdir+0x21a/0x3d0 [161.155850] ? nfsd4_encode_readlink+0x210/0x210 [161.156731] ? write_bytes_to_xdr_buf+0x97/0xe0 [161.157598] ? __write_bytes_to_xdr_buf+0xd0/0xd0 [161.158494] ? lock_downgrade+0x90/0x90 [161.159232] ? nfs4svc_decode_voidarg+0x10/0x10 [161.160092] nfsd4_encode_operation+0x15a/0x440 [161.160959] nfsd4_proc_compound+0x718/0xe90 [161.161818] nfsd_dispatch+0x18e/0x2c0 [161.162586] svc_process_common+0x786/0xc50 [161.163403] ? nfsd_svc+0x380/0x380 [161.164137] ? svc_printk+0x160/0x160 [161.164846] ? svc_xprt_do_enqueue.part.0+0x365/0x380 [161.165808] ? nfsd_svc+0x380/0x380 [161.166523] ? rcu_is_watching+0x23/0x40 [161.167309] svc_process+0x1a5/0x200 [161.168019] nfsd+0x1f5/0x380 [161.168663] ? nfsd_shutdown_threads+0x260/0x260 [161.169554] kthread+0x1c4/0x210 [161.170224] ? kthread_insert_work_sanitary_check+0x80/0x80 [161.171246] ret_from_fork+0x1f/0x30
CVE-2024-49881	In the Linux kernel, the following vulnerability has been resolved: ext4: update orig_path in ext4_find_extent() In ext4_find_extent(), if the path is not big enough, we free it and set *orig_path to NULL. But after reallocating and successfully initializing the path, we don't update *orig_path, in which case the caller gets a valid path but a NULL path, and this may cause a NULL pointer dereference or a path memory leak. For example: ext4_split_extent path = *ppath = 2000 ext4_find_extent if (depth > path[0].p_maxdepth) kfree(path = 2000); *orig_path = path = NULL; path = kcalloc() = 3000 ext4_split_extent_at(*ppath = NULL) path = *ppath; ex = path[depth].p_ext; // NULL pointer dereference! ===== BUG: kernel NULL pointer dereference, address: 0000000000000010 CPU: 6 UID: 0 PID: 576 Comm: fsstress Not tainted 6.11.0-rc2-dirty #847 RIP: 0010:ext4_split_extent_at+0x6d/0x560 Call Trace: <TASK> ext4_split_extent.isra.0+0xcb/0x1b0 ext4_ext_convert_to_initialized+0x168/0x6c0 ext4_ext_handle_unwritten_extents+0x325/0x4d0 ext4_ext_map_blocks+0x520/0xdb0 ext4_map_blocks+0x2b0/0x690 ext4_iomap_begin+0x20e/0x2c0 [...] ===== Therefore, *orig_path is updated when the extent lookup succeeds, so that the caller can safely use path or *ppath.

CVE ID	Description
CVE-2024-49882	In the Linux kernel, the following vulnerability has been resolved: ext4: fix double brelse() the buffer of the extents path In ext4_ext_try_to_merge_up(), set path[1].p_bh to NULL after it has been released, otherwise it may be released twice. An example of what triggers this is as follows: <pre>split2 map split1 ----- ----- ----- ext4_ext_map_blocks ext4_ext_handle_unwritten_extents ext4_split_convert_extents // path->p_depth == 0 ext4_split_extent // 1. do split1 ext4_split_extent_at ext4_ext_insert_extent ext4_ext_create_new_leaf ext4_ext_grow_indepth le16_add_cpu(&neh- >eh_depth, 1) ext4_find_extent // return -ENOMEM // get error and try zeroout path = ext4_find_extent path->p_depth = 1 ext4_ext_try_to_merge ext4_ext_try_to_merge_up path- >p_depth = 0 brelse(path[1].p_bh) ---> not set to NULL here // zeroout success // 2. update path ext4_find_extent // 3. do split2 ext4_split_extent_at ext4_ext_insert_extent ext4_ext_create_new_leaf ext4_ext_grow_indepth le16_add_cpu(&neh->eh_depth, 1) ext4_find_extent path[0].p_bh = NULL; path->p_depth = 1 read_extent_tree_block ---> return err // path[1].p_bh is still the old value ext4_free_ext_path ext4_ext_drop_refs // path->p_depth == 1 brelse(path[1].p_bh) ---> brelse a buffer twice Finally got the following WARNING when removing the buffer from lru: ===== VFS: brelse: Trying to free free buffer WARNING: CPU: 2 PID: 72 at fs/buffer.c:1241 __brelse+0x58/0x90 CPU: 2 PID: 72 Comm: kworker/u19:1 Not tainted 6.9.0-dirty #716 RIP: 0010: __brelse+0x58/0x90 Call Trace: <TASK> __find_get_block+0x6e7/0x810 bdev_getblk+0x2b/0x480 __ext4_get_inode_loc +0x48a/0x1240 ext4_get_inode_loc+0xb2/0x150 ext4_reserve_inode_write+0xb7/0x230 __ext4_mark_inode_dirty+0x144/0x6a0 ext4_ext_insert_extent+0x9c8/0x3230 ext4_ext_map_blocks +0xf45/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...]</pre> =====
CVE-2024-49883	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid use-after-free in ext4_ext_insert_extent() As Ojaswin mentioned in Link, in ext4_ext_insert_extent(), if the path is reallocated in ext4_ext_create_new_leaf(), we'll use the stale path and cause UAF. Below is a sample trace with dummy values: ext4_ext_insert_extent path = *ppath = 2000 ext4_ext_create_new_leaf(ppath) ext4_find_extent(ppath) path = *ppath = 2000 if (depth > path[0].p_maxdepth) kfree(path = 2000); *ppath = path = NULL; path = kcalloc() = 3000 *ppath = 3000; return path; /* here path is still 2000, UAF! */ eh = path[depth].p_hdr ===== BUG: KASAN: slab-use-after-free in ext4_ext_insert_extent+0x26d4/0x3330 Read of size 8 at addr ffff8881027bf7d0 by task kworker/u36:1/179 CPU: 3 UID: 0 PID: 179 Comm: kworker/u6:1 Not tainted 6.11.0-rc2-dirty #866 Call Trace: <TASK> ext4_ext_insert_extent +0x26d4/0x3330 ext4_ext_map_blocks+0xe22/0x2d40 ext4_map_blocks+0x71e/0x1700 ext4_do_writepages+0x1290/0x2800 [...] Allocated by task 179: ext4_find_extent+0x81c/0x1f70 ext4_ext_map_blocks+0x146/0x2d40 ext4_map_blocks+0x71e/0x1700 ext4_do_writepages +0x1290/0x2800 ext4_writepages+0x26d/0x4e0 do_writepages+0x175/0x700 [...] Freed by task 179: kfree+0xcb/0x240 ext4_find_extent+0x7c0/0x1f70 ext4_ext_insert_extent +0xa26/0x3330 ext4_ext_map_blocks+0xe22/0x2d40 ext4_map_blocks+0x71e/0x1700 ext4_do_writepages+0x1290/0x2800 ext4_writepages+0x26d/0x4e0 do_writepages+0x175/0x700 [...] So use *ppath to update the path to avoid the above problem.
CVE-2024-49884	In the Linux kernel, the following vulnerability has been resolved: ext4: fix slab-use-after-free in ext4_split_extent_at() We hit the following use-after-free: ===== BUG: KASAN: slab-use-after-free in ext4_split_extent_at+0xba8/0xcc0 Read of size 2 at addr ffff88810548ed08 by task kworker/u20:0/40 CPU: 0 PID: 40 Comm: kworker/u20:0 Not tainted 6.9.0-dirty #724 Call Trace: <TASK> kasan_report+0x93/0xc0 ext4_split_extent_at +0xba8/0xcc0 ext4_split_extent.isra.0+0x18f/0x500 ext4_split_convert_extents+0x275/0x750 ext4_ext_handle_unwritten_extents+0x73e/0x1580 ext4_ext_map_blocks+0xe20/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...] Allocated by task 40: __kmmalloc_noprof+0x1ac/0x480 ext4_find_extent+0xf3b/0x1e70 ext4_ext_map_blocks +0x188/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...] Freed by task 40: kfree+0xf1/0x2b0 ext4_find_extent+0xa71/0x1e70 ext4_ext_insert_extent+0xa22/0x3260 ext4_split_extent_at+0x3ef/0xcc0 ext4_split_extent.isra.0+0x18f/0x500 ext4_split_convert_extents +0x275/0x750 ext4_ext_handle_unwritten_extents+0x73e/0x1580 ext4_ext_map_blocks +0xe20/0x2dc0 ext4_map_blocks+0x724/0x1700 ext4_do_writepages+0x12d6/0x2a70 [...] ===== The flow of issue triggering is as follows: ext4_split_extent_at path = *ppath ext4_ext_insert_extent(ppath) ext4_ext_create_new_leaf(ppath) ext4_find_extent(orig_path) path = *orig_path read_extent_tree_block // return -ENOMEM or -EIO ext4_free_ext_path(path) kfree(path) *orig_path = NULL a. If err is -ENOMEM: ext4_ext_dirty(path + path->p_depth) // path use-after-free !!! b. If err is -EIO and we have EXT_DEBUG defined: ext4_ext_show_leaf(path) eh = path[depth].p_hdr // path also use-after-free !!! So when trying to zeroout or fix the extent length, call ext4_find_extent() to update the path. In addition we use *ppath directly as an ext4_ext_show_leaf() input to avoid possible use-after-free when EXT_DEBUG is defined, and to avoid unnecessary path updates.

CVE ID	Description
CVE-2024-49886	In the Linux kernel, the following vulnerability has been resolved: platform/x86: ISST: Fix the KASAN report slab-out-of-bounds bug Attaching SST PCI device to VM causes "BUG: KASAN: slab-out-of-bounds". kasan report: [19.411889] <pre> ===== [19.413702] BUG: KASAN: slab-out-of-bounds in _isst_if_get_pci_dev+0x3d5/0x400 [isst_if_common] [19.415634] Read of size 8 at addr ffff888829e65200 by task cpuhp/16/113 [19.417368] [19.418627] CPU: 16 PID: 113 Comm: cpuhp/16 Tainted: G E 6.9.0 #10 [19.420435] Hardware name: VMware, Inc. VMware20,1/440BX Desktop Reference Platform, BIOS VMW201.00V.20192059.B64.2207280713 07/28/2022 [19.422687] Call Trace: [19.424091] <TASK> [19.425448] dump_stack_lvl+0x5d/0x80 [19.426963] ? _isst_if_get_pci_dev +0x3d5/0x400 [isst_if_common] [19.428694] print_report+0x19d/0x52e [19.430206] ? __pfx_raw_spin_lock_irqsave+0x10/0x10 [19.431837] ? _isst_if_get_pci_dev+0x3d5/0x400 [isst_if_common] [19.433539] kasan_report+0xf0/0x170 [19.435019] ? _isst_if_get_pci_dev +0x3d5/0x400 [isst_if_common] [19.436709] _isst_if_get_pci_dev+0x3d5/0x400 [isst_if_common] [19.438379] ? __pfx_sched_clock_cpu+0x10/0x10 [19.439910] isst_if_cpu_online+0x406/0x58f [isst_if_common] [19.441573] ? __pfx_isst_if_cpu_online+0x10/0x10 [isst_if_common] [19.443263] ? ttwu_queue_wakelist+0x2c1/0x360 [19.444797] cpuhp_invoke_callback+0x221/0xec0 [19.446337] cpuhp_thread_fun+0x21b/0x610 [19.447814] ? __pfx_cpuhp_thread_fun+0x10/0x10 [19.449354] smpboot_thread_fn+0x2e7/0x6e0 [19.450859] ? __pfx_smpboot_thread_fn+0x10/0x10 [19.452405] kthread+0x29c/0x350 [19.453817] ? __pfx_kthread+0x10/0x10 [19.455253] ret_from_fork+0x31/0x70 [19.456685] ? __pfx_kthread+0x10/0x10 [19.458114] ret_from_fork_asm+0x1a/0x30 [19.459573] </ TASK> [19.460853] [19.462055] Allocated by task 1198: [19.463410] kasan_save_stack+0x30/0x50 [19.464788] kasan_save_track+0x14/0x30 [19.466139] __kasan_kmalloc+0xaa/0xb0 [19.467465] __kmalloc+0x1cd/0x470 [19.468748] isst_if_cdev_register+0x1da/0x350 [isst_if_common] [19.470233] isst_if_mbox_init+0x108/0xff0 [isst_if_mbox_msr] [19.471670] do_one_initcall +0xa4/0x380 [19.472903] do_init_module+0x238/0x760 [19.474105] load_module+0x5239/0x6f00 [19.475285] init_module_from_file+0xd1/0x130 [19.476506] idempotent_init_module+0x23b/0x650 [19.477725] __x64_sys_finit_module+0xbe/0x130 [19.476506] idempotent_init_module+0x23b/0x650 [19.477725] __x64_sys_finit_module+0xbe/0x130 [19.478920] do_syscall_64+0x82/0x160 [19.480036] entry_SYSCALL_64_after_hwframe+0x76/0x7e [19.481292] [19.482205] The buggy address belongs to the object at ffff888829e65000 which belongs to the cache kmalloc-512 of size 512 [19.484818] The buggy address is located 0 bytes to the right of allocated 512-byte region [ffff888829e65000, ffff888829e65200) [19.487447] [19.488328] The buggy address belongs to the physical page: [19.489569] page: refcount:1 mapcount:0 mapping:0000000000000000 index:0xffff888829e60c00 pfn:0x829e60 [19.491140] head: order:3 entire_mapcount:0 nr_pages_mapped:0 pincount:0 [19.492466] anon flags: 0x57ffffc0000840(slab head node=1 zone=2 lastcpupid=0x1fffff) [19.493914] page_type: 0xffffffffff [19.494988] raw: 0057ffffc0000840 ffff88810004cc80 0000000000000000 0000000000000001 [19.496451] raw: ffff888829e60c00 0000000080200018 00000001ffffff 0000000000000000 [19.497906] head: 0057ffffc0000840 ffff88810004cc80 0000000000000000 0000000000000001 [19.499379] head: ffff888829e60c00 0000000080200018 00000001ffffff 0000000000000000 [19.500844] head: 0057ffffc0000003 ffffea0020a79801 fffffea0020a79848 00000000ffffff [19.502316] head: 0000000800000000 0000000000000000 00000000ffffff 0000000000000000 [19.503784] page dumped because: k --- truncated---</pre>
CVE-2024-49887	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to don't panic system for no free segment fault injection f2fs: fix to don't panic system for no free segment fault injection syzbot reports a f2fs bug as below: F2FS-fs (loop0): inject no free segment in get_new_segment of __allocate_new_segment+0x1ce/0x940 fs/f2fs/segment.c:3167 F2FS-fs (loop0): Stopped filesystem due to reason: 7 -----[cut here]----- kernel BUG at fs/f2fs/segment.c:2748! CPU: 0 UID: 0 PID: 5109 Comm: syz-executor304 Not tainted 6.11.0-rc6-syzkaller-00363-g89f5e14d05b4 #0 RIP: 0010:get_new_segment fs/f2fs/segment.c:2748 [inline] RIP: 0010:new_curseg+0x1f61/0x1f70 fs/f2fs/segment.c:2836 Call Trace: __allocate_new_segment+0x1ce/0x940 fs/f2fs/segment.c:3167 f2fs_allocate_new_section fs/f2fs/segment.c:3181 [inline] f2fs_allocate_pinning_section+0xfa/0x4e0 fs/f2fs/segment.c:3195 f2fs_expand_inode_data+0x5d6/0xbb0 fs/f2fs/file.c:1799 f2fs_fallocate+0x448/0x960 fs/f2fs/file.c:1903 vfs_fallocate+0x553/0x6c0 fs/open.c:334 do_vfs_ioctl+0x2592/0x2e50 fs/ioctl.c:886 __do_sys_ioctl fs/ioctl.c:905 [inline] __se_sys_ioctl+0x81/0x170 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0010:get_new_segment fs/f2fs/segment.c:2748 [inline] RIP: 0010:new_curseg+0x1f61/0x1f70 fs/f2fs/segment.c:2836 The root cause is when we inject no free segment fault into f2fs, we should not panic system, fix it.

CVE ID	Description
CVE-2024-49888	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix a sdiv overflow issue Zac Ecob reported a problem where a bpf program may cause kernel crash due to the following error: Oops: divide error: 0000 [#1] PREEMPT SMP KASAN PTI The failure is due to the below signed divide: LLONG_MIN/-1 where LLONG_MIN equals to -9,223,372,036,854,775,808. LLONG_MIN/-1 is supposed to give a positive number 9,223,372,036,854,775,808, but it is impossible since for 64-bit system, the maximum positive number is 9,223,372,036,854,775,807. On x86_64, LLONG_MIN/-1 will cause a kernel exception. On arm64, the result for LLONG_MIN/-1 is LLONG_MIN. Further investigation found all the following sdiv/smod cases may trigger an exception when bpf program is running on x86_64 platform: - LLONG_MIN/-1 for 64bit operation - INT_MIN/-1 for 32bit operation - LLONG_MIN%-1 for 64bit operation - INT_MIN%-1 for 32bit operation where -1 can be an immediate or in a register. On arm64, there are no exceptions: - LLONG_MIN/-1 = LLONG_MIN - INT_MIN/-1 = INT_MIN - LLONG_MIN%-1 = 0 - INT_MIN%-1 = 0 where -1 can be an immediate or in a register. Insn patching is needed to handle the above cases and the patched codes produced results aligned with above arm64 result. The below are pseudo codes to handle sdiv/smod exceptions including both divisor -1 and divisor 0 and the divisor is stored in a register. sdiv: tmp = rX tmp += 1 /* [-1, 0] -> [0, 1] if tmp >(unsigned) 1 goto L2 if tmp == 0 goto L1 rY = 0 L1: rY = -rY; goto L3 L2: rY /= rX L3: smod: tmp = rX tmp += 1 /* [-1, 0] -> [0, 1] if tmp >(unsigned) 1 goto L1 if tmp == 1 (is64 ? goto L2 : goto L3) rY = 0; goto L2 L1: rY %= rX L2: goto L4 // only when !is64 L3: wY = wY // only when !is64 L4: [1] https://lore.kernel.org/bpf/tPjLTeh7S_DxFEqAI2Ji5MBS0ZVg7_G-Py2iaZpAaWtM961fFTWtsnlzwwTbzBzaUzwQa0NATXKUlt0LZOfgnDcYKcSwAnAGdUF3LBrhGQ=@protonmail.com
CVE-2024-49889	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid use-after-free in ext4_ext_show_leaf() In ext4_find_extent(), path may be freed by error or be reallocated, so using a previously saved *ppath may have been freed and thus may trigger use-after-free, as follows: ext4_split_extent path = *ppath; ext4_split_extent_at(ppath) path = ext4_find_extent(ppath) ext4_split_extent_at(ppath) // ext4_find_extent fails to free path // but zeroout succeeds ext4_ext_show_leaf(inode, path) eh = path[depth].p_hdr // path use-after-free !!! Similar to ext4_split_extent_at(), we use *ppath directly as an input to ext4_ext_show_leaf(). Fix a spelling error by the way. Same problem in ext4_ext_handle_unwritten_extents(). Since 'path' is only used in ext4_ext_show_leaf(), remove 'path' and use *ppath directly. This issue is triggered only when EXT_DEBUG is defined and therefore does not affect functionality.
CVE-2024-49891	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Validate hdwq pointers before dereferencing in reset/errata paths When the HBA is undergoing a reset or is handling an errata event, NULL ptr dereference crashes may occur in routines such as lpfc_sli_flush_io_rings(), lpfc_dev_loss_tmo_callbk(), or lpfc_abort_handler(). Add NULL ptr checks before dereferencing hdwq pointers that may have been freed due to operations colliding with a reset or errata event handler.
CVE-2024-49899	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize denominators' default to 1 [WHAT & HOW] Variables used as denominators and maybe not assigned to other values, should not be 0. Change their default to 1 so they are never 0. This fixes 10 DIVIDE_BY_ZERO issues reported by Coverity.
CVE-2024-49904	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: add list empty check to avoid null pointer issue Add list empty check to avoid null pointer issues in some corner cases. - list_for_each_entry_safe()
CVE-2024-49906	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check null pointer before try to access it [why & how] Change the order of the pipe_ctx->plane_state check to ensure that plane_state is not null before accessing it.
CVE-2024-49908	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null check for 'afb' in amdgpu_dm_update_cursor (v2) This commit adds a null check for the 'afb' variable in the amdgpu_dm_update_cursor function. Previously, 'afb' was assumed to be null at line 8388, but was used later in the code without a null check. This could potentially lead to a null pointer dereference. Changes since v1: - Moved the null check for 'afb' to the line where 'afb' is used. (Alex) Fixes the below: drivers/gpu/drm/amd/amdgpu/./display/amdgpu_dm/amdgpu_dm.c:8433 amdgpu_dm_update_cursor() error: we previously assumed 'afb' could be null (see line 8388)
CVE-2024-49910	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for function pointer in dcn401_set_output_transfer_func This commit adds a null check for the set_output_gamma function pointer in the dcn401_set_output_transfer_func function. Previously, set_output_gamma was being checked for null, but then it was being dereferenced without any null check. This could lead to a null pointer dereference if set_output_gamma is null. To fix this, we now ensure that set_output_gamma is not null before dereferencing it. We do this by adding a null check for set_output_gamma before the call to set_output_gamma.

CVE ID	Description
CVE-2024-49916	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add NULL check for clk_mgr and clk_mgr->funcs in dcn401_init_hw This commit addresses a potential null pointer dereference issue in the `dcn401_init_hw` function. The issue could occur when `dc->clk_mgr` or `dc->clk_mgr->funcs` is null. The fix adds a check to ensure `dc->clk_mgr` and `dc->clk_mgr->funcs` is not null before accessing its functions. This prevents a potential null pointer dereference. Reported by smatch: drivers/gpu/drm/amd/amdgpu/./display/dc/hwss/dcn401/dcn401_hwseq.c:416 dcn401_init_hw() error: we previously assumed 'dc->clk_mgr' could be null (see line 225)
CVE-2024-49925	In the Linux kernel, the following vulnerability has been resolved: fbdev: efifb: Register sysfs groups through driver core The driver core can register and cleanup sysfs groups already. Make use of that functionality to simplify the error handling and cleanup. Also avoid a UAF race during unregistering where the sysctl attributes were usable after the info struct was freed.
CVE-2024-49927	In the Linux kernel, the following vulnerability has been resolved: x86/ioapic: Handle allocation failures gracefully Breno observed panics when using failslab under certain conditions during runtime: can not alloc irq_pin_list (-1,0,20) Kernel panic - not syncing: IO-APIC: failed to add irq-pin. Can not proceed panic+0x4e9/0x590 mp_irqdomain_alloc+0x9ab/0xa80 irq_domain_alloc_irqs_locked+0x25d/0x8d0 __irq_domain_alloc_irqs+0x80/0x110 mp_map_pin_to_irq+0x645/0x890 acpi_register_gsi_ioapic+0xe6/0x150 hpet_open+0x313/0x480 That's a pointless panic which is a leftover of the historic IO/APIC code which panic'ed during early boot when the interrupt allocation failed. The only place which might justify panic is the PIT/HPET timer_check() code which tries to figure out whether the timer interrupt is delivered through the IO/APIC. But that code does not require to handle interrupt allocation failures. If the interrupt cannot be allocated then timer delivery fails and it either panics due to that or falls back to legacy mode. Cure this by removing the panic wrapper around __add_pin_to_irq_node() and making mp_irqdomain_alloc() aware of the failure condition and handle it as any other failure in this function gracefully.
CVE-2024-49928	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid reading out of bounds when loading TX power FW elements Because the loop-expression will do one more time before getting false from cond-expression, the original code copied one more entry size beyond valid region. Fix it by moving the entry copy to loop-body.
CVE-2024-49929	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: avoid NULL pointer dereference iwl_mvm_tx_skb_sta() and iwl_mvm_tx_mpdu() verify that the mvmsta pointer is not NULL. It retrieves this pointer using iwl_mvm_sta_from_mac80211, which is dereferencing the ieee80211_sta pointer. If sta is NULL, iwl_mvm_sta_from_mac80211 will dereference a NULL pointer. Fix this by checking the sta pointer before retrieving the mvmsta from it. If sta is not NULL, then mvmsta isn't either.
CVE-2024-49930	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix array out-of-bound access in SoC stats Currently, the ath11k_soc_dp_stats::hal_reo_error array is defined with a maximum size of DP_REO_DST_RING_MAX. However, the ath11k_dp_process_rx() function access ath11k_soc_dp_stats::hal_reo_error using the REO destination SRNG ring ID, which is incorrect. SRNG ring ID differ from normal ring ID, and this usage leads to out-of-bounds array access. To fix this issue, modify ath11k_dp_process_rx() to use the normal ring ID directly instead of the SRNG ring ID to avoid out-of-bounds array access. Tested-on: QCN9074 hw1.0 PCI WLAN.HK.2.7.0.1-01744-QCAHKS WPL_SILICONZ-1

CVE ID	Description
CVE-2024-49932	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't readahead the relocation inode on RST On relocation we're doing readahead on the relocation inode, but if the filesystem is backed by a RAID stripe tree we can get ENOENT (e.g. due to preallocated extents not being mapped in the RST) from the lookup. But readahead doesn't handle the error and submits invalid reads to the device, causing an assertion in the scatter-gather list code: BTRFS info (device nvme1n1): balance: start -d -m -s BTRFS info (device nvme1n1): relocating block group 6480920576 flags data/raid0 BTRFS error (device nvme1n1): cannot find raid-stripe for logical [6481928192, 6481969152] devid 2, profile raid0 -----[cut here]----- kernel BUG at include/linux/scatterlist.h:115! Oops: invalid opcode: 0000 [#1] PREEMPT SMP PTI CPU: 0 PID: 1012 Comm: btrfs Not tainted 6.10.0-rc7+ #567 RIP: 0010:___blk_rq_map_sg+0x339/0x4a0 RSP: 0018:ffff90001a43820 EFLAGS: 00010202 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffea00045d4802 RDX: 0000000117520000 RSI: 0000000000000000 RDI: ffff8881027d1000 RBP: 0000000000003000 R08: ffffea00045d4902 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000001000 R12: ffff8881003d10b8 R13: ffff90001a438f0 R14: 0000000000000000 R15: 0000000000003000 FS: 00007fcc048a6900(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000002cd11000 CR3: 00000001109ea001 CR4: 0000000000370eb0 Call Trace: <TASK> ? __die_body.cold+0x14/0x25 ? die+0x2e/0x50 ? do_trap+0xca/0x110 ? do_error_trap+0x65/0x80 ? __blk_rq_map_sg+0x339/0x4a0 ? exc_invalid_op+0x50/0x70 ? __blk_rq_map_sg+0x339/0x4a0 ? asm_exc_invalid_op+0x1a/0x20 ? __blk_rq_map_sg+0x339/0x4a0 nvme_prep_rq.part.0+0x9d/0x770 nvme_queue_rq+0x7d/0x1e0 __blk_mq_issue_directly+0x2a/0x90 ? blk_mq_get_budget_and_tag+0x61/0x90 blk_mq_try_issue_list_directly+0x56/0xf0 blk_mq_flush_plug_list.part.0+0x52b/0x5d0 __blk_flush_plug+0xc6/0x110 blk_finish_plug+0x28/0x40 read_pages+0x160/0x1c0 page_cache_ra_unbounded+0x109/0x180 relocate_file_extent_cluster+0x611/0x6a0 ? btrfs_search_slot+0xba4/0xd20 ? balance_dirty_pages_ratelimited_flags+0x26/0xb00 relocate_data_extent.constprop.0+0x134/0x160 relocate_block_group+0x3f2/0x500 btrfs_relocate_block_group+0x250/0x430 btrfs_relocate_chunk+0x3f/0x130 btrfs_balance+0x71b/0xef0 ? kmmalloc_trace_noprof+0x13b/0x280 btrfs_ioctl+0x2c2e/0x3030 ? kvfree_call_rcu+0x1e6/0x340 ? list_lru_add_obj+0x66/0x80 ? mntput_no_expire+0x3a/0x220 __x64_sys_ioctl+0x96/0xc0 do_syscall_64+0x54/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7fcc04514f9b Code: Unable to access opcode bytes at 0x7fcc04514f71. RSP: 002b:00007ffeba923370 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff90001a43820 RBX: 0000000000000003 RCX: 00007fcc04514f9b RDX: 00007ffeba923460 RSI: 00000000c4009420 RDI: 0000000000000003 RBP: 0000000000000000 R08: 0000000000000013 R09: 0000000000000001 R10: 00007fcc043fbba8 R11: 0000000000000246 R12: 00007ffeba924fc5 R13: 00007ffeba923460 R14: 0000000000000002 R15: 0000000004d4bb0 </TASK> Modules linked in: ---[end trace 0000000000000000]--- RIP: 0010:___blk_rq_map_sg+0x339/0x4a0 RSP: 0018:ffff90001a43820 EFLAGS: 00010202 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffea00045d4802 RDX: 0000000117520000 RSI: 0000000000000000 RDI: ffff8881027d1000 RBP: 0000000000003000 R08: ffffea00045d4902 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000001000 R12: ffff8881003d10b8 R13: ffff90001a438f0 R14: 0000000000000000 R15: 0000000000003000 FS: 00007fcc048a6900(0000) GS:ffff88813bc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fcc04514f71 CR3: 00000001109ea001 CR4: 0000000000370eb0 Kernel p ---truncated---

CVE ID	Description
CVE-2024-49934	In the Linux kernel, the following vulnerability has been resolved: fs/inode: Prevent dump_mapping() accessing invalid dentry.d_name.name It's observed that a crash occurs during hot-remove a memory device, in which user is accessing the hugetlb. See calltrace as following: -----[cut here]----- WARNING: CPU: 1 PID: 14045 at arch/x86/mm/fault.c:1278 do_user_addr_fault+0x2a0/0x790 Modules linked in: kmem device_dax cxl_mem cxl_pmem cxl_port cxl_pci dax_hmem dax_pmem nd_pmem cxl_acpi nd_btt cxl_core crc32c_intel nvme virtiofs fuse nvme_core nfit libnvdimm dm_multipath scsi_dh_rdac scsi_dh_emc s_mirror dm_region_hash dm_log dm_mod CPU: 1 PID: 14045 Comm: daxctl Not tainted 6.10.0-rc2-lizhijian+ #492 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:do_user_addr_fault+0x2a0/0x790 Code: 48 8b 00 a8 04 0f 84 b5 fe ff ff e9 1c ff ff ff 4c 89 e9 4c 89 e2 be 01 00 00 00 bf 02 00 00 00 e8 b5 ef 24 00 e9 42 fe ff ff <0f> 0b 48 83 c4 08 4c 89 ea 48 89 ee 4c 89 e7 5b 5d 41 5c 41 5d 41 RSP: 0000:ffffc90000a575f0 EFLAGS: 00010046 RAX: ffff88800c303600 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 00000000000001000 RSI: ffffffff82504162 RDI: ffffffff824b2c36 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: ffff80000a57658 R13: 00000000000001000 R14: ffff88800bc2e040 R15: 0000000000000000 FS: 00007f51cb57d880(0000) GS:ffff88807fd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000000001000 CR3: 00000000072e2004 CR4: 000000000001706f DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __warn+0x8d/0x190 ? do_user_addr_fault+0x2a0/0x790 ? report_bug+0x1c3/0x1d0 ? handle_bug+0x3c/0x70 ? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 ? do_user_addr_fault+0x2a0/0x790 ? exc_page_fault+0x31/0x200 exc_page_fault+0x68/0x200 <...snip...> BUG: unable to handle page fault for address: 00000000000001000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 800000000ad92067 P4D 800000000ad92067 PUD 7677067 PMD 0 Oops: Oops: 0000 [#1] PREEMPT SMP PTI ---[end trace 0000000000000000]--- BUG: unable to handle page fault for address: 00000000000001000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 800000000ad92067 P4D 800000000ad92067 PUD 7677067 PMD 0 Oops: Oops: 0000 [#1] PREEMPT SMP PTI CPU: 1 PID: 14045 Comm: daxctl Kdump: loaded Tainted: G W 6.10.0-rc2-lizhijian+ #492 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:dentry_name+0x1f4/0x440 <...snip...> ? dentry_name+0x2fa/0x440 vsnprintf+0x1f3/0x4f0 vprintk_store+0x23a/0x540 vprintk_emit+0x6d/0x330 _printk+0x58/0x80 dump_mapping+0x10b/0x1a0 ? __pfx_free_object_rcu+0x10/0x10 __dump_page+0x26b/0x3e0 ? vprintk_emit+0xe0/0x330 ? _printk+0x58/0x80 ? dump_page+0x17/0x50 dump_page+0x17/0x50 do_migrate_range+0x2f7/0x7f0 ? do_migrate_range+0x42/0x7f0 ? offline_pages+0x2f4/0x8c0 offline_pages+0x60a/0x8c0 memory_subsys_offline+0x9f/0x1c0 ? lockdep_hardirqs_on+0x77/0x100 ? _raw_spin_unlock_irqrestore+0x38/0x60 device_offline+0xe3/0x110 state_store+0x6e/0xc0 kernfs_fop_write_iter+0x143/0x200 vfs_write+0x39f/0x560 ksys_write+0x65/0xf0 do_syscall_64+0x62/0x130 Previously, some sanity check have been done in dump_mapping() before the print facility parsing '%pd' though, it's still possible to run into an invalid dentry.d_name.name. Since dump_mapping() only needs to dump the filename only, retrieve it by itself in a safer way to prevent an unnecessary crash. Note that either retrieving the filename with '%pd' or strncpy_from_kernel_nofault(), the filename could be unreliable.
CVE-2024-49937	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: Set correct chandef when starting CAC When starting CAC in a mode other than AP mode, it return a "WARNING: CPU: 0 PID: 63 at cfg80211_chandef_dfs_usable+0x20/0xaf [cfg80211]" caused by the chandef.chan being null at the end of CAC. Solution: Ensure the channel definition is set for the different modes when starting CAC to avoid getting a NULL 'chan' at the end of CAC. Call Trace: ? show_regs.part.0+0x14/0x16 ? __warn+0x67/0xc0 ? cfg80211_chandef_dfs_usable+0x20/0xaf [cfg80211] ? report_bug+0xa7/0x130 ? exc_overflow+0x30/0x30 ? handle_bug+0x27/0x50 ? exc_invalid_op+0x18/0x60 ? handle_exception+0xf6/0xf6 ? exc_overflow+0x30/0x30 ? cfg80211_chandef_dfs_usable+0x20/0xaf [cfg80211] ? exc_overflow+0x30/0x30 ? cfg80211_chandef_dfs_usable+0x20/0xaf [cfg80211] ? regulatory_propagate_dfs_state.cold+0x1b/0x4c [cfg80211] ? cfg80211_propagate_cac_done_wk+0x1a/0x30 [cfg80211] ? process_one_work+0x165/0x280 ? worker_thread+0x120/0x3f0 ? kthread+0xc2/0xf0 ? process_one_work+0x280/0x280 ? kthread_complete_and_exit+0x20/0x20 ? ret_from_fork+0x19/0x24 [shorten subject, remove OCB, reorder cases to match previous list]
CVE-2024-49938	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k_htc: Use __skb_set_length() for resetting urb before resubmit Syzbot points out that skb_trim() has a sanity check on the existing length of the skb, which can be uninitialised in some error paths. The intent here is clearly just to reset the length to zero before resubmitting, so switch to calling __skb_set_length(skb, 0) directly. In addition, __skb_set_length() already contains a call to skb_reset_tail_pointer(), so remove the redundant call. The syzbot report came from ath9k_hif_usb_reg_in_cb(), but there's a similar usage of skb_trim() in ath9k_hif_usb_rx_cb(), change both while we're at it.

CVE ID	Description
CVE-2024-49939	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid to add interface to list twice when SER If SER L2 occurs during the WoWLAN resume flow, the add interface flow is triggered by ieee80211_reconfig(). However, due to rtw89_wow_resume() return failure, it will cause the add interface flow to be executed again, resulting in a double add list and causing a kernel panic. Therefore, we have added a check to prevent double adding of the list. list_add double add: new=ffff99d6992e2010, prev=ffff99d6992e2010, next=ffff99d695302628. -----[cut here]----- kernel BUG at lib/list_debug.c:37! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 0 PID: 9 Comm: kworker/0:1 Tainted: G W O 6.6.30-02659-gc18865c4dfbd #1 770df2933251a0e3c888ba69d1053a817a6376a7 Hardware name: HP Grunt/Grunt, BIOS Google_Grunt.11031.169.0 06/24/2021 Workqueue: events_freezable ieee80211_restart_work [mac80211] RIP: 0010: __list_add_valid_or_report+0x5e/0xb0 Code: c7 74 18 48 39 ce 74 13 b0 01 59 5a 5e 5f 41 58 41 59 41 5a 5d e9 e2 d6 03 00 cc 48 c7 c7 8d 4f 17 83 48 89 c2 e8 02 c0 00 00 <0f> 0b 48 c7 c7 aa 8c 1c 83 e8 f4 bf 00 00 0f 0b 48 c7 c7 c8 bc 12 RSP: 0018:ffffa91b8007bc50 EFLAGS: 00010246 RAX: 0000000000000058 RBX: ffff99d6992e0900 RCX: a014d76c70ef3900 RDX: fffffa91b8007bae8 RSI: 00000000ffffdfff RDI: 0000000000000001 RBP: fffffa91b8007bc88 R08: 0000000000000000 R09: fffffa91b8007bae0 R10: 00000000ffffdfff R11: ffffffff83a79800 R12: ffff99d695302060 R13: ffff99d695300900 R14: ffff99d6992e1be0 R15: ffff99d6992e2010 FS: 0000000000000000(0000) GS:ffff99d6aac00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000078fdb43480 CR3: 000000010e464000 CR4: 00000000001506f0 Call Trace: <TASK> ? __die_body+0x1f/0x70 ? die+0x3d/0x60 ? do_trap+0xa4/0x110 ? __list_add_valid_or_report+0x5e/0xb0 ? do_error_trap+0x6d/0x90 ? __list_add_valid_or_report+0x5e/0xb0 ? handle_invalid_op+0x30/0x40 ? __list_add_valid_or_report+0x5e/0xb0 ? exc_invalid_op+0x3c/0x50 ? asm_exc_invalid_op+0x16/0x20 ? __list_add_valid_or_report+0x5e/0xb0 rtw89_ops_add_interface+0x309/0x310 [rtw89_core 7c32b1ee6854761c0321027c8a58c5160e41f48f] drv_add_interface+0x5c/0x130 [mac80211 83e989e6e616bd5b4b8a2b0a9f9352a2c385a3bc] ieee80211_reconfig+0x241/0x13d0 [mac80211 83e989e6e616bd5b4b8a2b0a9f9352a2c385a3bc] ? finish_wait+0x3e/0x90 ? synchronize_rcu_expedited+0x174/0x260 ? sync_rcu_exp_done_unlocked+0x50/0x50 ? wake_bit_function+0x40/0x40 ieee80211_restart_work+0xf0/0x140 [mac80211 83e989e6e616bd5b4b8a2b0a9f9352a2c385a3bc] process_scheduled_works+0x1e5/0x480 worker_thread+0xea/0x1e0 kthread+0xdb/0x110 ? move_linked_works+0x90/0x90 ? kthread_associate_blkcg+0xa0/0xa0 ret_from_fork+0x3b/0x50 ? kthread_associate_blkcg+0xa0/0xa0 ret_from_fork_asm+0x11/0x20 </TASK> Modules linked in: dm_integrity async_xor xor async_tx lz4 lz4_compress zstd zstd_compress zram zsmalloc rfcomm cmac uinput algif_hash algif_skcipher af_alg btusb btrtl iio_trig_hrtimer industrialio_sw_trigger btmtk industrialio_configfs btbcm btintel uvcvideo videobuf2_vmalloc iio_trig_sysfs videobuf2_memops videobuf2_v4l2 videobuf2_common uvc snd_hda_codec_hdmi veth snd_hda_intel snd_intel_dspcfg acpi_als snd_hda_codec industrialio_triggered_buffer kfifo_buf snd_hwdep industrialio i2c_piix4 snd_hda_core designware_i2s ip6table_nat snd_soc_max98357a xt_MASQUERADE xt_cgroup snd_soc_acp_rt5682_mach fuse rtw89_8922ae(O) rtw89_8922a(O) rtw89_pci(O) rtw89_core(O) 8021q mac80211(O) bluetooth ecdh_generic ecc cfg80211 r8152 mii joydev gsmi: Log Shutdown Reason 0x03 ---[end trace 0000000000000000]---
CVE-2024-49940	In the Linux kernel, the following vulnerability has been resolved: l2tp: prevent possible tunnel refcount underflow When a session is created, it sets a backpointer to its tunnel. When the session refcount drops to 0, l2tp_session_free drops the tunnel refcount if session->tunnel is non-NULL. However, session->tunnel is set in l2tp_session_create, before the tunnel refcount is incremented by l2tp_session_register, which leaves a small window where session->tunnel is non-NULL when the tunnel refcount hasn't been bumped. Moving the assignment to l2tp_session_register is trivial but l2tp_session_create calls l2tp_session_set_header_len which uses session->tunnel to get the tunnel's encap. Add an encap arg to l2tp_session_set_header_len to avoid using session->tunnel. If l2tpv3 sessions have colliding IDs, it is possible for l2tp_v3_session_get to race with l2tp_session_register and fetch a session which doesn't yet have session->tunnel set. Add a check for this case.
CVE-2024-49944	In the Linux kernel, the following vulnerability has been resolved: sctp: set sk_state back to CLOSED if autobind fails in sctp_listen_start In sctp_listen_start() invoked by sctp_inet_listen(), it should set the sk_state back to CLOSED if sctp_autobind() fails due to whatever reason. Otherwise, next time when calling sctp_inet_listen(), if sctp_sk(sk)->reuse is already set via setsockopt(SCTP_REUSE_PORT), sctp_sk(sk)->bind_hash will be dereferenced as sk_state is LISTENING, which causes a crash as bind_hash is NULL. KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] RIP: 0010:sctp_inet_listen+0x7f0/0xa20 net/sctp/socket.c:8617 Call Trace: <TASK> __sys_listen_socket net/socket.c:1883 [inline] __sys_listen+0x1b7/0x230 net/socket.c:1894 __do_sys_listen net/socket.c:1902 [inline]

CVE ID	Description
CVE-2024-49948	In the Linux kernel, the following vulnerability has been resolved: net: add more sanity checks to qdisc_pkt_len_init() One path takes care of SKB_GSO_DODGY, assuming skb->len is bigger than hdr_len. virtio_net_hdr_to_skb() does not fully dissect TCP headers, it only make sure it is at least 20 bytes. It is possible for an user to provide a malicious 'GSO' packet, total length of 80 bytes. - 20 bytes of IPv4 header - 60 bytes TCP header - a small gso_size like 8 virtio_net_hdr_to_skb() would declare this packet as a normal GSO packet, because it would see 40 bytes of payload, bigger than gso_size. We need to make detect this case to not underflow qdisc_skb_cb(skb)->pkt_len.
CVE-2024-49950	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix uaf in l2cap_connect [Syzbot reported] BUG: KASAN: slab-use-after-free in l2cap_connect.constprop.0+0x10d8/0x1270 net/bluetooth/l2cap_core.c:3949 Read of size 8 at addr ffff8880241e9800 by task kworker/u9:0/54 CPU: 0 UID: 0 PID: 54 Comm: kworker/u9:0 Not tainted 6.11.0-rc6-syzkaller-00268-g788220eee30d #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 Workqueue: hci2 hci_rx_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl +0x116/0x1f0 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0xc3/0x620 mm/kasan/report.c:488 kasan_report+0xd9/0x110 mm/kasan/report.c:601 l2cap_connect.constprop.0+0x10d8/0x1270 net/bluetooth/l2cap_core.c:3949 l2cap_connect_req net/bluetooth/l2cap_core.c:4080 [inline] l2cap_bredr_sig_cmd net/bluetooth/l2cap_core.c:4772 [inline] l2cap_sig_channel net/bluetooth/l2cap_core.c:5543 [inline] l2cap_rcv_frame+0xf0b/0x8eb0 net/bluetooth/l2cap_core.c:6825 l2cap_rcv_acldata+0x9b4/0xb70 net/bluetooth/l2cap_core.c:7514 hci_acldata_packet net/bluetooth/hci_core.c:3791 [inline] hci_rx_work+0xaab/0x1610 net/bluetooth/hci_core.c:4028 process_one_work+0x9c5/0x1b40 kernel/workqueue.c:3231 process_scheduled_works kernel/workqueue.c:3312 [inline] worker_thread+0x6c8/0xed0 kernel/workqueue.c:3389 kthread +0x2c1/0x3a0 kernel/kthread.c:389 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 ... Freed by task 5245: kasan_save_stack +0x33/0x60 mm/kasan/common.c:47 kasan_save_track+0x14/0x30 mm/kasan/common.c:68 kasan_save_free_info+0x3b/0x60 mm/kasan/generic.c:579 poison_slab_object+0xf7/0x160 mm/kasan/common.c:240 __kasan_slab_free+0x32/0x50 mm/kasan/common.c:256 kasan_slab_free include/linux/kasan.h:184 [inline] slab_free_hook mm/slab.c:2256 [inline] slab_free mm/slab.c:4477 [inline] kfree +0x12a/0x3b0 mm/slab.c:4598 l2cap_conn_free net/bluetooth/l2cap_core.c:1810 [inline] kref_put include/linux/kref.h:65 [inline] l2cap_conn_put net/bluetooth/l2cap_core.c:1822 [inline] l2cap_conn_del +0x59d/0x730 net/bluetooth/l2cap_core.c:1802 l2cap_connect_cfm+0x9e6/0xf80 net/bluetooth/l2cap_core.c:7241 hci_connect_cfm include/net/bluetooth/hci_core.h:1960 [inline] hci_conn_failed +0x1c3/0x370 net/bluetooth/hci_conn.c:1265 hci_abort_conn_sync+0x75a/0xb50 net/bluetooth/hci_sync.c:5583 abort_conn_sync+0x197/0x360 net/bluetooth/hci_conn.c:2917 hci_cmd_sync_work +0x1a4/0x410 net/bluetooth/hci_sync.c:328 process_one_work+0x9c5/0x1b40 kernel/workqueue.c:3231 process_scheduled_works kernel/workqueue.c:3312 [inline] worker_thread+0x6c8/0xed0 kernel/workqueue.c:3389 kthread+0x2c1/0x3a0 kernel/kthread.c:389 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244

CVE ID	Description
CVE-2024-49952	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: prevent nf_skb_duplicated corruption syzbot found that nf_dup_ipv4() or nf_dup_ipv6() could write per-cpu variable nf_skb_duplicated in an unsafe way [1]. Disabling preemption as hinted by the splat is not enough, we have to disable soft interrupts as well. [1] BUG: using __this_cpu_write() in preemptible [00000000] code: syz.4.282/6316 caller is nf_dup_ipv4+0x651/0x8f0 net/ipv4/netfilter/nf_dup_ipv4.c:87 CPU: 0 UID: 0 PID: 6316 Comm: syz.4.282 Not tainted 6.11.0-rc7-syzkaller-00104-g7052622fccb1 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 check_preemption_disabled+0x10e/0x120 lib/smp_processor_id.c:49 nf_dup_ipv4+0x651/0x8f0 net/ipv4/netfilter/nf_dup_ipv4.c:87 nft_dup_ipv4_eval+0x1db/0x300 net/ipv4/netfilter/nft_dup_ipv4.c:30 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x4ad/0x1da0 net/netfilter/nf_tables_core.c:288 nft_do_chain_ipv4+0x202/0x320 net/netfilter/nft_chain_filter.c:23 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xc3/0x220 net/netfilter/core.c:626 nf_hook+0x2c4/0x450 include/linux/netfilter.h:269 NF_HOOK_COND include/linux/netfilter.h:302 [inline] ip_output+0x185/0x230 net/ipv4/ip_output.c:433 ip_local_out net/ipv4/ip_output.c:129 [inline] ip_send_skb+0x74/0x100 net/ipv4/ip_output.c:1495 udp_send_skb+0xacf/0x1650 net/ipv4/udp.c:981 udp_sendmsg+0x1c21/0x2a60 net/ipv4/udp.c:1269 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x1a6/0x270 net/socket.c:745 ____sys_sendmsg+0x525/0x7d0 net/socket.c:2597 __sys_sendmsg net/socket.c:2651 [inline] __sys_sendmmsg+0x3b2/0x740 net/socket.c:2737 __do_sys_sendmmsg net/socket.c:2766 [inline] __se_sys_sendmmsg net/socket.c:2763 [inline] __x64_sys_sendmmsg+0xa0/0xb0 net/socket.c:2763 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f4ce4f7def9 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff f7 d8 64 89 01 48 RSP: 002b:00007f4ce5d4a038 EFLAGS: 00000246 ORIG_RAX: 0000000000000133 RAX: ffffffffda RBX: 00007f4ce5135f80 RCX: 00007f4ce4f7def9 RDX: 0000000000000001 RSI: 0000000020005d40 RDI: 0000000000000006 RBP: 00007f4ce4ff0b76 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000000 R14: 00007f4ce5135f80 R15: 00007ffd4cbc6d68 </TASK>
CVE-2024-49955	In the Linux kernel, the following vulnerability has been resolved: ACPI: battery: Fix possible crash when unregistering a battery hook When a battery hook returns an error when adding a new battery, then the battery hook is automatically unregistered. However the battery hook provider cannot know that, so it will later call battery_hook_unregister() on the already unregistered battery hook, resulting in a crash. Fix this by using the list head to mark already unregistered battery hooks as already being unregistered so that they can be ignored by battery_hook_unregister().
CVE-2024-49959	In the Linux kernel, the following vulnerability has been resolved: jbd2: stop waiting for space when jbd2_cleanup_journal_tail() returns error In __jbd2_log_wait_for_space(), we might call jbd2_cleanup_journal_tail() to recover some journal space. But if an error occurs while executing jbd2_cleanup_journal_tail() (e.g., an EIO), we don't stop waiting for free space right away, we try other branches, and if j_committing_transaction is NULL (i.e., the tid is 0), we will get the following complain: ===== JBD2: I/O error when updating journal superblock for sdd-8. __jbd2_log_wait_for_space: needed 256 blocks and only had 217 space available __jbd2_log_wait_for_space: no way to get more journal space in sdd-8 -----[cut here]----- WARNING: CPU: 2 PID: 139804 at fs/jbd2/checkpoint.c:109 __jbd2_log_wait_for_space+0x251/0x2e0 Modules linked in: CPU: 2 PID: 139804 Comm: kworker/u8:3 Not tainted 6.6.0+ #1 RIP: 0010: __jbd2_log_wait_for_space+0x251/0x2e0 Call Trace: <TASK> add_transaction_credits+0x5d1/0x5e0 start_this_handle+0x1ef/0x6a0 jbd2__journal_start+0x18b/0x340 ext4_dirty_inode+0x5d/0xb0 __mark_inode_dirty+0xe4/0x5d0 generic_update_time+0x60/0x70 [...] ===== So only if jbd2_cleanup_journal_tail() returns 1, i.e., there is nothing to clean up at the moment, continue to try to reclaim free space in other ways. Note that this fix relies on commit 6f6a6fda2945 ("jbd2: fix ocs2 corrupt when updating journal superblock fails") to make jbd2_cleanup_journal_tail return the correct error code.
CVE-2024-49960	In the Linux kernel, the following vulnerability has been resolved: ext4: fix timer use-after-free on failed mount Syzbot has found an ODEBUG bug in ext4_fill_super The del_timer_sync function cancels the s_err_report timer, which reminds about filesystem errors daily. We should guarantee the timer is no longer active before kfree(sbi). When filesystem mounting fails, the flow goes to failed_mount3, where an error occurs when ext4_stop_mmpd is called, causing a read I/O failure. This triggers the ext4_handle_error function that ultimately re-arms the timer, leaving the s_err_report timer active before kfree(sbi) is called. Fix the issue by canceling the s_err_report timer after calling ext4_stop_mmpd.
CVE-2024-49962	In the Linux kernel, the following vulnerability has been resolved: ACPICA: check null return of ACPI_ALLOCATE_ZEROED() in acpi_db_convert_to_package() ACPICA commit 4d4547cf13cca820ff7e0f859ba83e1a610b9fd0 ACPI_ALLOCATE_ZEROED() may fail, elements might be NULL and will cause NULL pointer dereference later. [rjw: Subject and changelog edits]

CVE ID	Description
CVE-2024-49970	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Implement bounds check for stream encoder creation in DCN401 'stream_enc_regs' array is an array of dcn10_stream_enc_registers structures. The array is initialized with four elements, corresponding to the four calls to stream_enc_regs() in the array initializer. This means that valid indices for this array are 0, 1, 2, and 3. The error message 'stream_enc_regs' 4 <= 5 below, is indicating that there is an attempt to access this array with an index of 5, which is out of bounds. This could lead to undefined behavior Here, eng_id is used as an index to access the stream_enc_regs array. If eng_id is 5, this would result in an out-of-bounds access on the stream_enc_regs array. Thus fixing Buffer overflow error in dcn401_stream_encoder_create Found by smatch: drivers/gpu/drm/amd/amdgpu/../display/dc/resource/dcn401/dcn401_resource.c:1209 dcn401_stream_encoder_create() error: buffer overflow 'stream_enc_regs' 4 <= 5
CVE-2024-49971	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Increase array size of dummy_boolean [WHY] dml2_core_shared_mode_support and dml_core_mode_support access the third element of dummy_boolean, i.e. hw_debug5 = &s->dummy_boolean[2], when dummy_boolean has size of 2. Any assignment to hw_debug5 causes an OVERRUN. [HOW] Increase dummy_boolean's array size to 3. This fixes 2 OVERRUN issues reported by Coverity.
CVE-2024-49973	In the Linux kernel, the following vulnerability has been resolved: r8169: add tally counter fields added with RTL8125 RTL8125 added fields to the tally counter, what may result in the chip dma'ing these new fields to unallocated memory. Therefore make sure that the allocated memory area is big enough to hold all of the tally counter values, even if we use only parts of it.
CVE-2024-49974	In the Linux kernel, the following vulnerability has been resolved: NFSD: Limit the number of concurrent async COPY operations Nothing appears to limit the number of concurrent async COPY operations that clients can start. In addition, AFAICT each async COPY can copy an unlimited number of 4MB chunks, so can run for a long time. Thus IMO async COPY can become a DoS vector. Add a restriction mechanism that bounds the number of concurrent background COPY operations. Start simple and try to be fair -- this patch implements a per-namespace limit. An async COPY request that occurs while this limit is exceeded gets NFS4ERR_DELAY. The requesting client can choose to send the request again after a delay or fall back to a traditional read/write style copy. If there is need to make the mechanism more sophisticated, we can visit that in future patches.
CVE-2024-49975	In the Linux kernel, the following vulnerability has been resolved: uprobes: fix kernel info leak via "[uprobes]" vma xol_add_vma() maps the uninitialized page allocated by __create_xol_area() into userspace. On some architectures (x86) this memory is readable even without VM_READ, VM_EXEC results in the same pgprot_t as VM_EXEC VM_READ, although this doesn't really matter, debugger can read this memory anyway.
CVE-2024-49978	In the Linux kernel, the following vulnerability has been resolved: gso: fix udp gso fraglist segmentation after pull from frag_list Detect gso fraglist skbs with corrupted geometry (see below) and pass these to skb_segment instead of skb_segment_list, as the first can segment them correctly. Valid SKB_GSO_FRAGLIST skbs - consist of two or more segments - the head_skb holds the protocol headers plus first gso_size - one or more frag_list skbs hold exactly one segment - all but the last must be gso_size Optional datapath hooks such as NAT and BPF (bpf_skb_pull_data) can modify these skbs, breaking these invariants. In extreme cases they pull all data into skb linear. For UDP, this causes a NULL ptr deref in __udp4_gso_segment_list_csum at udp_hdr(seg->next)->dest. Detect invalid geometry due to pull, by checking head_skb size. Don't just drop, as this may blackhole a destination. Convert to be able to pass to regular skb_segment.
CVE-2024-49990	In the Linux kernel, the following vulnerability has been resolved: drm/xe/hdcp: Check GSC structure validity Sometimes xe_gsc is not initialized when checked at HDCP capability check. Add gsc structure check to avoid null pointer error.
CVE-2024-49991	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: amd_kfd_free_gtt_mem clear the correct pointer Pass pointer reference to amdgpu_bo_unref to clear the correct pointer, otherwise amdgpu_bo_unref clear the local variable, the original pointer not set to NULL, this could cause use-after-free bug.
CVE-2024-49994	In the Linux kernel, the following vulnerability has been resolved: block: fix integer overflow in BLKSECDISCARD I independently rediscovered commit 22d24a544b0d49bbcbdd61c8c0eaf77d3c9297155 block: fix overflow in blk_ioctl_discard() but for secure erase. Same problem: uint64_t r[2] = {512, 18446744073709551104ULL}; iocctl(fd, BLKSECDISCARD, r); will enter near infinite loop inside blkdev_issue_secure_erase(): a.out: attempt to access beyond end of device loop0: rw=5, sector=3399043073, nr_sectors = 1024 limit=2048 bio_check_eod: 3286214 callbacks suppressed

CVE ID	Description
CVE-2024-49996	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix buffer overflow when parsing NFS reparse points ReparseDataLength is sum of the InodeType size and DataBuffer size. So to get DataBuffer size it is needed to subtract InodeType's size from ReparseDataLength. Function cifs_strndup_from_utf16() is currently accessing buf->DataBuffer at position after the end of the buffer because it does not subtract InodeType size from the length. Fix this problem and correctly subtract variable len. Member InodeType is present only when reparse buffer is large enough. Check for ReparseDataLength before accessing InodeType to prevent another invalid memory access. Major and minor rdev values are present also only when reparse buffer is large enough. Check for reparse buffer size before calling reparse_mkdev().
CVE-2024-50000	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix NULL deref in mlx5e_tir_builder_alloc() In mlx5e_tir_builder_alloc() kvzalloc() may return NULL which is dereferenced on the next line in a reference to the modify field. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-50001	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix error path in multi-packet WQE transmit Remove the erroneous unmap in case no DMA mapping was established The multi-packet WQE transmit code attempts to obtain a DMA mapping for the skb. This could fail, e.g. under memory pressure, when the IOMMU driver just can't allocate more memory for page tables. While the code tries to handle this in the path below the err_unmap label it erroneously unmaps one entry from the sq's FIFO list of active mappings. Since the current map attempt failed this unmap is removing some random DMA mapping that might still be required. If the PCI function now presents that IOVA, the IOMMU may assumes a rogue DMA access and e.g. on s390 puts the PCI function in error state. The erroneous behavior was seen in a stress-test environment that created memory pressure.
CVE-2024-50003	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix system hang while resume with TBT monitor [Why] Connected with a Thunderbolt monitor and do the suspend and the system may hang while resume. The TBT monitor HPD will be triggered during the resume procedure and call the drm_client_modeset_probe() while struct drm_connector connector->dev->master is NULL. It will mess up the pipe topology after resume. [How] Skip the TBT monitor HPD during the resume procedure because we currently will probe the connectors after resume by default. (cherry picked from commit 453f86a26945207a16b8f66aaed5962dc2b95b85)
CVE-2024-50004	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: update DML2 policy EnhancedPrefetchScheduleAccelerationFinal DCN35 [WHY & HOW] Mismatch in DCN35 DML2 cause bw validation failed to acquire unexpected DPP pipe to cause grey screen and system hang. Remove EnhancedPrefetchScheduleAccelerationFinal value override to match HW spec. (cherry picked from commit 9dad21f910fcea2bdcff4af46159101d7f9cd8ba)
CVE-2024-50006	In the Linux kernel, the following vulnerability has been resolved: ext4: fix i_data_sem unlock order in ext4_ind_migrate() Fuzzing reports a possible deadlock in jbd2_log_wait_commit. This issue is triggered when an EXT4_IOC_MIGRATE ioctl is set to require synchronous updates because the file descriptor is opened with O_SYNC. This can lead to the jbd2_journal_stop() function calling jbd2_might_wait_for_commit(), potentially causing a deadlock if the EXT4_IOC_MIGRATE call races with a write(2) system call. This problem only arises when CONFIG_PROVE_LOCKING is enabled. In this case, the jbd2_might_wait_for_commit macro locks jbd2_handle in the jbd2_journal_stop function while i_data_sem is locked. This triggers lockdep because the jbd2_journal_start function might also lock the same jbd2_handle simultaneously. Found by Linux Verification Center (linuxtesting.org) with syzkaller. Rule: add
CVE-2024-50007	In the Linux kernel, the following vulnerability has been resolved: ALSA: asihpi: Fix potential OOB array access ASIHPI driver stores some values in the static array upon a response from the driver, and its index depends on the firmware. We shouldn't trust it blindly. This patch adds a sanity check of the array index to fit in the array size.
CVE-2024-50008	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Fix memcpy() field-spanning write warning in mwifiex_cmd_802_11_scan_ext() Replace one-element array with a flexible-array member in 'struct host_cmd_ds_802_11_scan_ext'. With this, fix the following warning: elo 16 17:51:58 surfacebook kernel: -----[cut here]----- elo 16 17:51:58 surfacebook kernel: memcpy: detected field-spanning write (size 243) of single field "ext_scan->tlv_buffer" at drivers/net/wireless/marvell/mwifiex/scan.c:2239 (size 1) elo 16 17:51:58 surfacebook kernel: WARNING: CPU: 0 PID: 498 at drivers/net/wireless/marvell/mwifiex/scan.c:2239 mwifiex_cmd_802_11_scan_ext +0x83/0x90 [mwifiex]
CVE-2024-50009	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: add check for cpufreq_cpu_get's return value cpufreq_cpu_get may return NULL. To avoid NULL-dereference check it and return in case of error. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2024-50010	In the Linux kernel, the following vulnerability has been resolved: exec: don't WARN for racy path_noexec check Both i_mode and noexec checks wrapped in WARN_ON stem from an artifact of the previous implementation. They used to legitimately check for the condition, but that got moved up in two commits: 633fb6ac3980 ("exec: move S_ISREG() check earlier") 0fd338b2d2cd ("exec: move path_noexec() check earlier") Instead of being removed said checks are WARN_ON'ed instead, which has some debug value. However, the spurious path_noexec check is racy, resulting in unwarranted warnings should someone race with setting the noexec flag. One can note there is more to perm-checking whether execve is allowed and none of the conditions are guaranteed to still hold after they were tested for. Additionally this does not validate whether the code path did any perm checking to begin with -- it will pass if the inode happens to be regular. Keep the redundant path_noexec() check even though it's mindless nonsense checking for guarantee that isn't given so drop the WARN. Reword the commentary and do small tidy ups while here. [brauner: keep redundant path_noexec() check]
CVE-2024-50019	In the Linux kernel, the following vulnerability has been resolved: kthread: unpark only parked kthread Calling into kthread unparking unconditionally is mostly harmless when the kthread is already unparked. The wake up is then simply ignored because the target is not in TASK_PARKED state. However if the kthread is per CPU, the wake up is preceded by a call to kthread_bind() which expects the task to be inactive and in TASK_PARKED state, which obviously isn't the case if it is unparked. As a result, calling kthread_stop() on an unparked per-cpu kthread triggers such a warning: WARNING: CPU: 0 PID: 11 at kernel/kthread.c:525 __kthread_bind_mask kernel/kthread.c:525 <TASK> kthread_stop+0x17a/0x630 kernel/kthread.c:707 destroy_workqueue+0x136/0xc40 kernel/workqueue.c:5810 wg_destruct+0x1e2/0x2e0 drivers/net/wireguard/device.c:257 netdev_run_todo+0xe1a/0x1000 net/core/dev.c:10693 default_device_exit_batch+0xa14/0xa90 net/core/dev.c:11769 ops_exit_list net/core/net_namespace.c:178 [inline] cleanup_net+0x89d/0xcc0 net/core/net_namespace.c:640 process_one_work kernel/workqueue.c:3231 [inline] process_scheduled_works+0xa2c/0x1830 kernel/workqueue.c:3312 worker_thread+0x86d/0xd70 kernel/workqueue.c:3393 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Fix this with skipping unnecessary unparking while stopping a kthread.
CVE-2024-50024	In the Linux kernel, the following vulnerability has been resolved: net: Fix an unsafe loop on the list The kernel may crash when deleting a genetlink family if there are still listeners for that family: Oops: Kernel access of bad area, sig: 11 [#1] ... NIP [c000000000c080bc] netlink_update_socket_mc+0x3c/0xc0 LR [c000000000c0f764] __netlink_clear_multicast_users+0x74/0xc0 Call Trace: __netlink_clear_multicast_users+0x74/0xc0 genl_unregister_family+0xd4/0x2d0 Change the unsafe loop on the list to a safe one, because inside the loop there is an element removal from this list.
CVE-2024-50028	In the Linux kernel, the following vulnerability has been resolved: thermal: core: Reference count the zone in thermal_zone_get_by_id() There are places in the thermal netlink code where nothing prevents the thermal zone object from going away while being accessed after it has been returned by thermal_zone_get_by_id(). To address this, make thermal_zone_get_by_id() get a reference on the thermal zone device object to be returned with the help of get_device(), under thermal_list_lock, and adjust all of its callers to this change with the help of the cleanup.h infrastructure.

CVE ID	Description
CVE-2024-50033	In the Linux kernel, the following vulnerability has been resolved: slip: make slhc_remember() more robust against malicious packets syzbot found that slhc_remember() was missing checks against malicious packets [1]. slhc_remember() only checked the size of the packet was at least 20, which is not good enough. We need to make sure the packet includes the IPv4 and TCP header that are supposed to be carried. Add iph and th pointers to make the code more readable. [1] BUG: KMSAN: uninit-value in slhc_remember+0x2e8/0x7b0 drivers/net/sliph/slh.c:666 slhc_remember+0x2e8/0x7b0 drivers/net/sliph/slh.c:666 ppp_receive_nonmp_frame+0xe45/0x35e0 drivers/net/ppp/ppp_generic.c:2455 ppp_receive_frame drivers/net/ppp/ppp_generic.c:2372 [inline] ppp_do_rcv+0x65f/0x40d0 drivers/net/ppp/ppp_generic.c:2212 ppp_input+0x7dc/0xe60 drivers/net/ppp/ppp_generic.c:2327 pppoe_rcv_core+0x1d3/0x720 drivers/net/ppp/pppoe.c:379 sk_backlog_rcv+0x13b/0x420 include/net/sock.h:1113 __release_sock+0x1da/0x330 net/core/sock.c:3072 release_sock+0x6b/0x250 net/core/sock.c:3626 pppoe_sendmsg+0x2b8/0xb90 drivers/net/ppp/pppoe.c:903 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 __sys_sendmsg+0x903/0xb60 net/socket.c:2602 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slub.c:4091 [inline] slab_alloc_node mm/slub.c:4134 [inline] kmem_cache_alloc_node_noprof+0x6bf/0xb80 mm/slub.c:4186 kmmalloc_reserve+0x13d/0x4a0 net/core/skbuff.c:587 __alloc_skb+0x363/0x7b0 net/core/skbuff.c:678 alloc_skb include/linux/skbuff.h:1322 [inline] sock_wmalloc+0xfe/0x1a0 net/core/sock.c:2732 pppoe_sendmsg+0x3a7/0xb90 drivers/net/ppp/pppoe.c:867 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 __sys_sendmsg+0x903/0xb60 net/socket.c:2602 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __se_sys_sendmmsg net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f CPU: 0 UID: 0 PID: 5460 Comm: syz.2.33 Not tainted 6.12.0-rc2-syzkaller-00006-g87d6aab2389e #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024
CVE-2024-50034	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix lacks of icsk_syn_mss with IPPROTO_SMC Eric report a panic on IPPROTO_SMC, and give the facts that when INET_PROTOSW_ICSK was set, icsk->icsk_sync_mss must be set too. Bug: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x00000000086000005 EC = 0x21: IABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x05: level 1 translation fault user pgtbl: 4k pages, 48-bit VAs, pgdp=00000001195d1000 [0000000000000000] pgd=0800000109c46003, p4d=0800000109c46003, pud=0000000000000000 Internal error: Oops: 00000000086000005 [#1] PREEMPT SMP Modules linked in: CPU: 1 UID: 0 PID: 8037 Comm: syz.3.265 Not tainted 6.11.0-rc7-syzkaller-g5f5673607153 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : 0x0 lr : cipso_v4_sock_setattr+0x2a8/0x3c0 net/ipv4/cipso_ipv4.c:1910 sp : ffff80009b887a90 x29: ffff80009b887a90 x28: ffff80008db94050 x27: 0000000000000000 x26: 1ffff0001aa6f5b3 x25: dfff800000000000 x24: ffff0000db75da00 x23: 0000000000000000 x22: ffff0000d8b78518 x21: 0000000000000000 x20: ffff0000d537ad80 x19: ffff0000d8b78000 x18: 1ffff000366d79ee x17: ffff8000800614a8 x16: ffff800080569b84 x15: 0000000000000001 x14: 000000008b336894 x13: 00000000cd96feaa x12: 0000000000000003 x11: 0000000000004000 x10: 00000000000020a3 x9: 1ffff0001b16f0f1 x8: 0000000000000000 x7: 0000000000000000 x6: 000000000000003f x5: 0000000000000040 x4: 0000000000000001 x3: 0000000000000000 x2: 0000000000000002 x1: 0000000000000000 x0: ffff0000d8b78000 Call trace: 0x0 netlbl_sock_setattr+0x2e4/0x338 net/netlabel/netlabel_kapi.c:1000 smack_netlbl_add+0xa4/0x154 security/smack/smack_lsm.c:2593 smack_socket_post_create+0xa8/0x14c security/smack/smack_lsm.c:2973 security_socket_post_create+0x94/0xd4 security/security.c:4425 __sock_create+0x4c8/0x884 net/socket.c:1587 sock_create net/socket.c:1622 [inline] __sys_socket_create net/socket.c:1659 [inline] __sys_socket+0x134/0x340 net/socket.c:1706 __do_sys_socket net/socket.c:1720 [inline] __se_sys_socket net/socket.c:1718 [inline] __arm64_sys_socket+0x7c/0x94 net/socket.c:1718 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc+0x54/0x168 arch/arm64/kernel/entry-common.c:712 el0t_64_sync_handler+0x84/0xfc arch/arm64/kernel/entry-common.c:730 el0t_64_sync+0x190/0x194 arch/arm64/kernel/entry.S:598 Code: ??????? ??????? ??????? ??????? (????????) ---[end trace 0000000000000000]--- This patch add a toy implementation that performs a simple return to prevent such panic. This is because MSS can be set in sock_create_kern or smc_setsockopt, similar to how it's done in AF_SMC. However, for AF_SMC, there is currently no way to synchronize MSS within __sys_connect_file. This toy implementation lays the groundwork for us to support such feature for IPPROTO_SMC in the future.

CVE ID	Description
CVE-2024-50035	In the Linux kernel, the following vulnerability has been resolved: ppp: fix ppp_async_encode() illegal access syzbot reported an issue in ppp_async_encode() [1] In this case, pppoe_sendmsg() is called with a zero size. Then ppp_async_encode() is called with an empty skb. BUG: KMSAN: uninit-value in ppp_async_encode drivers/net/ppp/ppp_async.c:545 [inline] BUG: KMSAN: uninit-value in ppp_async_push+0xb4f/0x2660 drivers/net/ppp/ppp_async.c:675 ppp_async_encode drivers/net/ppp/ppp_async.c:545 [inline] ppp_async_push+0xb4f/0x2660 drivers/net/ppp/ppp_async.c:675 ppp_async_send+0x130/0x1b0 drivers/net/ppp/ppp_async.c:634 ppp_channel_bridge_input drivers/net/ppp/ppp_generic.c:2280 [inline] ppp_input+0x1f1/0xe60 drivers/net/ppp/ppp_generic.c:2304 pppoe_rcv_core+0x1d3/0x720 drivers/net/ppp/pppoe.c:379 sk_backlog_rcv+0x13b/0x420 include/net/sock.h:1113 __release_sock+0x1da/0x330 net/core/sock.c:3072 release_sock+0x6b/0x250 net/core/sock.c:3626 pppoe_sendmsg+0x2b8/0xb90 drivers/net/ppp/pppoe.c:903 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 __sys_sendmsg+0x903/0xb60 net/socket.c:2602 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slab.c:4092 [inline] slab_alloc_node mm/slab.c:4135 [inline] kmem_cache_alloc_node_noprof+0x6bf/0xb80 mm/slab.c:4187 kmallocc_reserve+0x13d/0x4a0 net/core/skbuff.c:587 __alloc_skb+0x363/0x7b0 net/core/skbuff.c:678 alloc_skb include/linux/skbuff.h:1322 [inline] sock_wmalloc+0xfe/0x1a0 net/core/sock.c:2732 pppoe_sendmsg+0x3a7/0xb90 drivers/net/ppp/pppoe.c:867 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:744 __sys_sendmsg+0x903/0xb60 net/socket.c:2602 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2656 __sys_sendmmsg+0x3c1/0x960 net/socket.c:2742 __do_sys_sendmmsg net/socket.c:2771 [inline] __se_sys_sendmmsg net/socket.c:2768 [inline] __x64_sys_sendmmsg+0xbc/0x120 net/socket.c:2768 x64_sys_call+0xb6e/0x3ba0 arch/x86/include/generated/asm/syscalls_64.h:308 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f CPU: 1 UID: 0 PID: 5411 Comm: syz.1.14 Not tainted 6.12.0-rc1-syzkaller-00165-g360c1f1f24c6 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024
CVE-2024-50036	In the Linux kernel, the following vulnerability has been resolved: net: do not delay dst_entries_add() in dst_release() dst_entries_add() uses per-cpu data that might be freed at netns dismantle from ip6_route_net_exit() calling dst_entries_destroy() Before ip6_route_net_exit() can be called, we release all the dsts associated with this netns, via calls to dst_release(), which waits an rcu grace period before calling dst_destroy() dst_entries_add() use in dst_destroy() is racy, because dst_entries_destroy() could have been called already. Decrementing the number of dsts must happen sooner. Notes: 1) in CONFIG_XFRM case, dst_destroy() can call dst_release_immediate(child), this might also cause UAF if the child does not have DST_NOCOUNT set. IPSEC maintainers might take a look and see how to address this. 2) There is also discussion about removing this count of dst, which might happen in future kernels.
CVE-2024-50038	In the Linux kernel, the following vulnerability has been resolved: netfilter: xttables: avoid NFPROTO_UNSPEC where needed syzbot managed to call xt_cluster match via ebttables: WARNING: CPU: 0 PID: 11 at net/netfilter/xt_cluster.c:72 xt_cluster_mt+0x196/0x780 [...] ebt_do_table+0x174b/0x2a40 Module registers to NFPROTO_UNSPEC, but it assumes ipv4/ipv6 packet processing. As this is only useful to restrict locally terminating TCP/UDP traffic, register this for ipv4 and ipv6 family only. Pablo points out that this is a general issue, direct users of the set/getsockopt interface can call into targets/matches that were only intended for use with ip(6)tables. Check all UNSPEC matches and targets for similar issues: - matches and targets are fine except if they assume skb_network_header() is valid -- this is only true when called from inet layer: ip(6) stack pulls the ip/ipv6 header into linear data area. - targets that return XT_CONTINUE or other xttables verdicts must be restricted too, they are incompatible with the ebttables traverser, e.g. EBT_CONTINUE is a completely different value than XT_CONTINUE. Most matches/targets are changed to register for NFPROTO_IPV4/IPV6, as they are provided for use by ip(6)tables. The MARK target is also used by arptables, so register for NFPROTO_ARP too. While at it, bail out if connbytes fails to enable the corresponding conntrack family. This change passes the selftests in iptables.git.

CVE ID	Description
CVE-2024-50039	In the Linux kernel, the following vulnerability has been resolved: net/sched: accept TCA_STAB only for root qdisc Most qdiscs maintain their backlog using qdisc_pkt_len(skb) on the assumption it is invariant between the enqueue() and dequeue() handlers. Unfortunately syzbot can crash a host rather easily using a TBF + SFQ combination, with an STAB on SFQ [1] We can't support TCA_STAB on arbitrary level, this would require to maintain per-qdisc storage. [1] [88.796496] BUG: kernel NULL pointer dereference, address: 0000000000000000 [88.798611] #PF: supervisor read access in kernel mode [88.799014] #PF: error_code(0x0000) - not-present page [88.799506] PGD 0 P4D 0 [88.799829] Oops: Oops: 0000 [#1] SMP NOPTI [88.800569] CPU: 14 UID: 0 PID: 2053 Comm: b371744477 Not tainted 6.12.0-rc1-virtme #1117 [88.801107] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [88.801779] RIP: 0010:sfq_dequeue (net/sched/sch_sfq.c:272 net/sched/sch_sfq.c:499) sch_sfq [88.802544] Code: 0f b7 50 12 48 8d 04 d5 00 00 00 00 48 89 d6 48 29 d0 48 8b 91 c0 01 00 00 48 c1 e0 03 48 01 c2 66 83 7a 1a 00 7e c0 48 8b 3a <4c> 8b 07 4c 89 02 49 89 50 08 48 c7 47 08 00 00 00 00 48 c7 07 00 All code ===== 0: 0f b7 50 12 movzwl 0x12(%rax),%edx 4: 48 8d 04 d5 00 00 00 lea 0x0(%rdx,8),%rax b: 00 c: 48 89 d6 mov %rdx,%rsi f: 48 29 d0 sub %rdx,%rax 12: 48 8b 91 c0 01 00 00 mov 0x1c0(%rcx),%rdx 19: 48 c1 e0 03 shl \$0x3,%rax 1d: 48 01 c2 add %rax,%rdx 20: 66 83 7a 1a 00 cmpw \$0x0,0x1a(%rdx) 25: 7e c0 jle 0xffffffffffffe7 27: 48 8b 3a mov (%rdx),%rdi 2a:* 4c 8b 07 mov (%rdi),%r8 <-- trapping instruction 2d: 4c 89 02 mov %r8,%rdx 30: 49 89 50 08 mov %rdx,0x8(%r8) 34: 48 c7 47 08 00 00 00 movq \$0x0,0x8(%rdi) 3b: 00 3c: 48 rex.W 3d: c7 .byte 0xc7 3e: 07 (bad) ... Code starting with the faulting instruction ===== 0: 4c 8b 07 mov (%rdi),%r8 3: 4c 89 02 mov %r8,%rdx 6: 49 89 50 08 mov %rdx,0x8(%r8) a: 48 c7 47 08 00 00 00 movq \$0x0,0x8(%rdi) 11: 00 12: 48 rex.W 13: c7 .byte 0xc7 14: 07 (bad) ... [88.803721] RSP: 0018:ffff9a1f892b7d58 EFLAGS: 00000206 [88.804032] RAX: 0000000000000000 RBX: ffff9a1f8420c800 RCX: ffff9a1f8420c800 [88.804560] RDX: ffff9a1f81bc1440 RSI: 0000000000000000 RDI: 0000000000000000 [88.805056] RBP: ffffffff04bb0e0 R08: 0000000000000001 R09: 00000000ff7f9a1f [88.805473] R10: 000000000001001b R11: 00000000000009a1f R12: 0000000000000140 [88.806194] R13: 0000000000000001 R14: ffff9a1f886df400 R15: ffff9a1f886df4ac [88.806734] FS: 00007f445601a740(0000) GS:ffff9a2e7fd80000(0000) knlGS:0000000000000000 [88.807225] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [88.807672] CR2: 0000000000000000 CR3: 000000050cc46000 CR4: 00000000000006f0 [88.808165] Call Trace: [88.808459] <TASK> [88.808710] ? __die (arch/x86/kernel/dumpstack.c:421 arch/x86/kernel/dumpstack.c:434) [88.809261] ? page_fault_oops (arch/x86/mm/fault.c:715) [88.809561] ? exc_page_fault (./arch/x86/include/asm/irqflags.h:26 ./arch/x86/include/asm/irqflags.h:87 ./arch/x86/include/asm/irqflags.h:147 arch/x86/mm/fault.c:1489 arch/x86/mm/fault.c:1539) [88.809806] ? asm_exc_page_fault (./arch/x86/include/asm/idtentry.h:623) [88.810074] ? sfq_dequeue (net/sched/sch_sfq.c:272 net/sched/sch_sfq.c:499) sch_sfq [88.810411] sfq_reset (net/sched/sch_sfq.c:525) sch_sfq [88.810671] qdisc_reset (./include/linux/skbuff.h:2135 ./include/linux/skbuff.h:2441 ./include/linux/skbuff.h:3304 ./include/linux/skbuff.h:3310 net/sched/sch_g ---truncated---

CVE ID	Description
CVE-2024-50040	In the Linux kernel, the following vulnerability has been resolved: igb: Do not bring the device up after non-fatal error Commit 004d25060c78 ("igb: Fix igb_down hung on surprise removal") changed igb_io_error_detected() to ignore non-fatal pcie errors in order to avoid hung task that can happen when igb_down() is called multiple times. This caused an issue when processing transient non-fatal errors. igb_io_resume(), which is called after igb_io_error_detected(), assumes that device is brought down by igb_io_error_detected() if the interface is up. This resulted in panic with stacktrace below. <pre>[T3256] igb 0000:09:00.0 haeth0: igb: haeth0 NIC Link is Down [T292] pcieport 0000:00:1c.5: AER: Uncorrected (Non-Fatal) error received: 0000:09:00.0 [T292] igb 0000:09:00.0: PCIe Bus Error: severity=Uncorrected (Non-Fatal), type=Transaction Layer, (Requester ID) [T292] igb 0000:09:00.0: device [8086:1537] error status/mask=00004000/00000000 [T292] igb 0000:09:00.0: [14] CmplTO [200.105524,009][T292] igb 0000:09:00.0: AER: TLP Header: 00000000 00000000 00000000 00000000 [T292] pcieport 0000:00:1c.5: AER: broadcast error_detected message [T292] igb 0000:09:00.0: Non-correctable non-fatal error reported. [T292] pcieport 0000:00:1c.5: AER: broadcast mmio_enabled message [T292] pcieport 0000:00:1c.5: AER: broadcast resume message [T292] -----[cut here]----- [T292] kernel BUG at net/core/dev.c:6539! [T292] invalid opcode: 0000 [#1] PREEMPT SMP [T292] RIP: 0010:napi_enable+0x37/0x40 [T292] Call Trace: [T292] <TASK> [T292] ? die+0x33/0x90 [T292] ? do_trap+0xdc/0x110 [T292] ? napi_enable+0x37/0x40 [T292] ? do_error_trap+0x70/0xb0 [T292] ? napi_enable+0x37/0x40 [T292] ? napi_enable+0x37/0x40 [T292] ? exc_invalid_op+0x4e/0x70 [T292] ? napi_enable+0x37/0x40 [T292] ? asm_exc_invalid_op +0x16/0x20 [T292] ? napi_enable+0x37/0x40 [T292] igb_up+0x41/0x150 [T292] igb_io_resume +0x25/0x70 [T292] report_resume+0x54/0x70 [T292] ? report_frozen_detected+0x20/0x20 [T292] pci_walk_bus+0x6c/0x90 [T292] ? aer_print_port_info+0xa0/0xa0 [T292] pcie_do_recovery +0x22f/0x380 [T292] aer_process_err_devices+0x110/0x160 [T292] aer_isr+0x1c1/0x1e0 [T292] ? disable_irq_nosync+0x10/0x10 [T292] irq_thread_fn+0x1a/0x60 [T292] irq_thread+0xe3/0x1a0 [T292] ? irq_set_affinity_notifier+0x120/0x120 [T292] ? irq_affinity_notify+0x100/0x100 [T292] kthread+0xe2/0x110 [T292] ? kthread_complete_and_exit+0x20/0x20 [T292] ret_from_fork +0x2d/0x50 [T292] ? kthread_complete_and_exit+0x20/0x20 [T292] ret_from_fork_asm+0x11/0x20 [T292] </TASK> To fix this issue igb_io_resume() checks if the interface is running and the device is not down this means igb_io_error_detected() did not bring the device down and there is no need to bring it up. </pre>
CVE-2024-50041	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix macvlan leak by synchronizing access to mac_filter_hash This patch addresses a macvlan leak issue in the i40e driver caused by concurrent access to vsi->mac_filter_hash. The leak occurs when multiple threads attempt to modify the mac_filter_hash simultaneously, leading to inconsistent state and potential memory leaks. To fix this, we now wrap the calls to i40e_del_mac_filter() and zeroing vf->default_lan_addr.addr with spin_lock/unlock_bh(&vsi->mac_filter_hash_lock), ensuring atomic operations and preventing concurrent access. Additionally, we add lockdep_assert_held(&vsi->mac_filter_hash_lock) in i40e_add_mac_filter() to help catch similar issues in the future. Reproduction steps: 1. Spawn VFs and configure port vlan on them. 2. Trigger concurrent macvlan operations (e.g., adding and deleting portvlan and/or mac filters). 3. Observe the potential memory leak and inconsistent state in the mac_filter_hash. This synchronization ensures the integrity of the mac_filter_hash and prevents the described leak.
CVE-2024-50044	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: FIX possible deadlock in rfcomm_sk_state_change rfcomm_sk_state_change attempts to use sock_lock so it must never be called with it locked but rfcomm_sock_ioctl always attempt to lock it causing the following trace: ===== WARNING: possible circular locking dependency detected 6.8.0-syzkaller-08951-gfe46a7dd189e #0 Not tainted ----- syz-executor386/5093 is trying to acquire lock: ffff88807c396258 (sk_lock-AF_BLUETOOTH-BTPROTO_RFCOMM){+..}-{0:0}, at: lock_sock include/net/sock.h:1671 [inline] ffff88807c396258 (sk_lock-AF_BLUETOOTH- BTPROTO_RFCOMM){+..}-{0:0}, at: rfcomm_sk_state_change+0x5b/0x310 net/bluetooth/ rfcomm/sock.c:73 but task is already holding lock: ffff88807badfd28 (&d->lock){+..}-{3:3}, at: __rfcomm_dlc_close+0x226/0x6a0 net/bluetooth/rfcomm/core.c:491

CVE ID	Description
CVE-2024-50045	In the Linux kernel, the following vulnerability has been resolved: netfilter: br_netfilter: fix panic with metadata_dst skb Fix a kernel panic in the br_netfilter module when sending untagged traffic via a VxLAN device. This happens during the check for fragmentation in br_nf_dev_queue_xmit. It is dependent on: 1) the br_netfilter module being loaded; 2) net.bridge.bridge-nf-call-iptables set to 1; 3) a bridge with a VxLAN (single-vxlan-device) netdevice as a bridge port; 4) untagged frames with size higher than the VxLAN MTU forwarded/flooded When forwarding the untagged packet to the VxLAN bridge port, before the netfilter hooks are called, br_handle_egress_vlan_tunnel is called and changes the skb_dst to the tunnel dst. The tunnel_dst is a metadata type of dst, i.e., skb_valid_dst(skb) is false, and metadata->dst.dev is NULL. Then in the br_netfilter hooks, in br_nf_dev_queue_xmit, there's a check for frames that needs to be fragmented: frames with higher MTU than the VxLAN device end up calling br_nf_ip_fragment, which in turns call ip_skb_dst_mtu. The ip_skb_dst_mtu tries to use the skb_dst(skb) as if it was a valid dst with valid dst->dev, thus the crash. This case was never supported in the first place, so drop the packet instead. PING 10.0.0.2 (10.0.0.2) from 0.0.0.0 h1-eth0: 2000(2028) bytes of data. [176.291791] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000110 [176.292101] Mem abort info: [176.292184] ESR = 0x0000000096000004 [176.292322] EC = 0x25: DABT (current EL), IL = 32 bits [176.292530] SET = 0, FnV = 0 [176.292709] EA = 0, S1PTW = 0 [176.292862] FSC = 0x04: level 0 translation fault [176.293013] Data abort info: [176.293104] ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 [176.293488] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [176.293787] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [176.293995] user pgtable: 4k pages, 48-bit VAs, pgdp=000000043ef5000 [176.294166] [0000000000000110] pgd=0000000000000000, p4d=0000000000000000 [176.294827] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [176.295252] Modules linked in: vxlan ip6_udp_tunnel udp_tunnel veth br_netfilter bridge stp llc ipv6 crc10dif_ce [176.295923] CPU: 0 PID: 188 Comm: ping Not tainted 6.8.0-rc3-g5b3fbd61b9d1 #2 [176.296314] Hardware name: linux,dummy-virt (DT) [176.296535] pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYP--) [176.296808] pc : br_nf_dev_queue_xmit+0x390/0x4ec [br_netfilter] [176.297382] lr : br_nf_dev_queue_xmit+0x2ac/0x4ec [br_netfilter] [176.297636] sp : ffff800080003630 [176.297743] x29: ffff800080003630 x28: 0000000000000008 x27: ffff6828c49ad9f8 [176.298093] x26: ffff6828c49ad000 x25: 0000000000000000 x24: 00000000000003e8 [176.298430] x23: 0000000000000000 x22: ffff6828c4960b40 x21: ffff6828c3b16d28 [176.298652] x20: ffff6828c3167048 x19: ffff6828c3b16d00 x18: 0000000000000014 [176.298926] x17: ffff60476322f000 x16: ffff67e164023730 x15: 0000000095744632 [176.299296] x14: ffff6828c3f1c880 x13: 0000000000000002 x12: ffff67e137926a70 [176.299574] x11: 0000000000000001 x10: ffff6828c3f1c898 x9 : 0000000000000000 [176.300049] x8 : ffff6828c49bf070 x7 : 0008460f18d5f20e x6 : f20e0100bebafece [176.300302] x5 : ffff6828c7f918fe x4 : ffff6828c49bf070 x3 : 0000000000000000 [176.300586] x2 : 0000000000000000 x1 : ffff6828c3c7ad00 x0 : ffff6828c7f918f0 [176.300889] Call trace: [176.301123] br_nf_dev_queue_xmit+0x390/0x4ec [br_netfilter] [176.301411] br_nf_post_routing+0x2a8/0x3e4 [br_netfilter] [176.301703] nf_hook_slow+0x48/0x124 [176.302060] br_forward_finish+0xc8/0xe8 [bridge] [176.302371] br_nf_hook_thresh+0x124/0x134 [br_netfilter] [176.302605] br_nf_forward_finish+0x118/0x22c [br_netfilter] [176.302824] br_nf_forward_ip.part.0+0x264/0x290 [br_netfilter] [176.303136] br_nf_forward+0x2b8/0x4e0 [br_netfilter] [176.303359] nf_hook_slow+0x48/0x124 [176.303 ---truncated---
CVE-2024-50048	In the Linux kernel, the following vulnerability has been resolved: fbcon: Fix a NULL pointer dereference issue in fbcon_puts syzbot has found a NULL pointer dereference bug in fbcon. Here is the simplified C reproducer: struct param { uint8_t type; struct tiocl_selection ts; }; int main() { struct fb_con2fbmap con2fb; struct param param; int fd = open("/dev/fb1", 0, 0); con2fb.console = 0x19; con2fb.framebuffer = 0; ioctl(fd, FBIOPUT_CON2FBMAP, &con2fb); param.type = 2; param.ts.xs = 0; param.ts.ys = 0; param.ts.xe = 0; param.ts.ye = 0; param.ts.sel_mode = 0; int fd1 = open("/dev/tty1", O_RDWR, 0); ioctl(fd1, TIOCLINUX, &param); con2fb.console = 1; con2fb.framebuffer = 0; ioctl(fd, FBIOPUT_CON2FBMAP, &con2fb); return 0; } After calling ioctl(fd1, TIOCLINUX, &param), the subsequent ioctl(fd, FBIOPUT_CON2FBMAP, &con2fb) causes the kernel to follow a different execution path: set_con2fb_map -> con2fb_init_display -> fbcon_set_disp -> redraw_screen -> hide_cursor -> clear_selection -> highlight -> invert_screen -> do_update_region -> fbcon_puts -> ops->puts Since ops->puts is a NULL pointer, this leads to a kernel panic. To prevent this, we need to call set_blitting_type() within set_con2fb_map() to properly initialize ops->puts.
CVE-2024-50051	In the Linux kernel, the following vulnerability has been resolved: spi: mpc52xx: Add cancel_work_sync before module remove If we remove the module which will call mpc52xx_spi_remove it will free 'ms' through spi_unregister_controller. while the work ms->work will be used. The sequence of operations that may lead to a UAF bug. Fix it by ensuring that the work is canceled before proceeding with the cleanup in mpc52xx_spi_remove.
CVE-2024-50055	In the Linux kernel, the following vulnerability has been resolved: driver core: bus: Fix double free in driver API bus_register() For bus_register(), any error which happens after kset_register() will cause that @priv are freed twice, fixed by setting @priv with NULL after the first free.

CVE ID	Description
CVE-2024-50057	In the Linux kernel, the following vulnerability has been resolved: usb: typec: tipd: Free IRQ only if it was requested before In polling mode, if no IRQ was requested there is no need to free it. Call devm_free_irq() only if client->irq is set. This fixes the warning caused by the tps6598x module removal: WARNING: CPU: 2 PID: 333 at kernel/irq/devres.c:144 devm_free_irq+0x80/0x8c ... Call trace: devm_free_irq+0x80/0x8c tps6598x_remove+0x28/0x88 [tps6598x] i2c_device_remove+0x2c/0x9c device_remove+0x4c/0x80 device_release_driver_internal+0x1cc/0x228 driver_detach+0x50/0x98 bus_remove_driver+0x6c/0xbc driver_unregister+0x30/0x60 i2c_del_driver+0x54/0x64 tps6598x_i2c_driver_exit+0x18/0xc3c [tps6598x] __arm64_sys_delete_module+0x184/0x264 invoke_syscall+0x48/0x110 el0_svc_common.constprop.0+0xc8/0xe8 do_el0_svc+0x20/0x2c el0_svc+0x28/0x98 el0t_64_sync_handler+0x13c/0x158 el0t_64_sync+0x190/0x194
CVE-2024-50058	In the Linux kernel, the following vulnerability has been resolved: serial: protect uart_port_dtr_rts() in uart_shutdown() too Commit af224ca2df29 (serial: core: Prevent unsafe uart port access, part 3) added few uport == NULL checks. It added one to uart_shutdown(), so the commit assumes, uport can be NULL in there. But right after that protection, there is an unprotected "uart_port_dtr_rts(uport, false);" call. That is invoked only if HUPCL is set, so I assume that is the reason why we do not see lots of these reports. Or it cannot be NULL at this point at all for some reason :P. Until the above is investigated, stay on the safe side and move this dereference to the if too. I got this inconsistency from Coverity under CID 1585130. Thanks.
CVE-2024-50063	In the Linux kernel, the following vulnerability has been resolved: bpf: Prevent tail call between progs attached to different hooks bpf progs can be attached to kernel functions, and the attached functions can take different parameters or return different return values. If prog attached to one kernel function tail calls prog attached to another kernel function, the ctx access or return value verification could be bypassed. For example, if prog1 is attached to func1 which takes only 1 parameter and prog2 is attached to func2 which takes two parameters. Since verifier assumes the bpf ctx passed to prog2 is constructed based on func2's prototype, verifier allows prog2 to access the second parameter from the bpf ctx passed to it. The problem is that verifier does not prevent prog1 from passing its bpf ctx to prog2 via tail call. In this case, the bpf ctx passed to prog2 is constructed from func1 instead of func2, that is, the assumption for ctx access verification is bypassed. Another example, if BPF LSM prog1 is attached to hook file_alloc_security, and BPF LSM prog2 is attached to hook bpf_lsm_audit_rule_known. Verifier knows the return value rules for these two hooks, e.g. it is legal for bpf_lsm_audit_rule_known to return positive number 1, and it is illegal for file_alloc_security to return positive number. So verifier allows prog2 to return positive number 1, but does not allow prog1 to return positive number. The problem is that verifier does not prevent prog1 from calling prog2 via tail call. In this case, prog2's return value 1 will be used as the return value for prog1's hook file_alloc_security. That is, the return value rule is bypassed. This patch adds restriction for tail call to prevent such bypasses.

CVE ID	Description
CVE-2024-50067	In the Linux kernel, the following vulnerability has been resolved: uprobe: avoid out-of-bounds memory access of fetching args Uprobe needs to fetch args into a percpu buffer, and then copy to ring buffer to avoid non-atomic context problem. Sometimes user-space strings, arrays can be very large, but the size of percpu buffer is only page size. And store_trace_args() won't check whether these data exceeds a single page or not, caused out-of-bounds memory access. It could be reproduced by following steps: 1. build kernel with CONFIG_KASAN enabled 2. save follow program as test.c ```\n#include <stdio.h> \n#include <stdlib.h> \n#include <string.h> // If string length large than MAX_STRING_SIZE, the fetch_store_strlen() // will return 0, cause __get_data_size() return shorter size, and // store_trace_args() will not trigger out-of-bounds access. // So make string length less than 4096. \n#define STRLEN 4093 void generate_string(char *str, int n) { int i; for (i = 0; i < n; ++i) { char c = i % 26 + 'a'; str[i] = c; } str[n-1] = '\\0'; } void print_string(char *str) { printf("%s\\n", str); } int main() { char tmp[STRLEN]; generate_string(tmp, STRLEN); print_string(tmp); return 0; } ``` 3. compile program `gcc -o test test.c` 4. get the offset of `print_string` `objdump -t test   grep -w print_string 0000000000401199 g F .text 000000000000001b print_string` 5. configure uprobe with offset 0x1199 `off=0x1199 cd /sys/kernel/debug/tracing/ echo "p /root/test:\${off} arg1=+0(%di):uststring arg2=\\\$comm arg3=+0(%di):uststring" > uprobe_events echo 1 > events/uprobes/enable echo 1 > tracing_on` 6. run `test`, and kasan will report error. ===== BUG: KASAN: use-after-free in strncpy_from_user+0x1d6/0x1f0 Write of size 8 at addr ffff88812311c004 by task test/499CPU: 0 UID: 0 PID: 499 Comm: test Not tainted 6.12.0-rc3+ #18 Hardware name: Red Hat KVM, BIOS 1.16.0-4.al8 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x55/0x70 print_address_description.constprop.0+0x27/0x310 kasan_report+0x10f/0x120 ? strncpy_from_user+0x1d6/0x1f0 strncpy_from_user+0x1d6/0x1f0 ? rmqueue.constprop.0+0x70d/0x2ad0 process_fetch_insn+0xb26/0x1470 ? __pfx_process_fetch_insn+0x10/0x10 ? __raw_spin_lock+0x85/0xe0 ? __pfx__raw_spin_lock+0x10/0x10 ? __pte_offset_map+0x1f/0x2d0 ? unwind_next_frame+0xc5f/0x1f80 ? arch_stack_walk+0x68/0xf0 ? is_bpf_text_address+0x23/0x30 ? kernel_text_address.part.0+0xbb/0xd0 ? __kernel_text_address+0x66/0xb0 ? unwind_get_return_address+0x5e/0xa0 ? __pfx_stack_trace_consume_entry+0x10/0x10 ? arch_stack_walk+0xa2/0xf0 ? __raw_spin_lock_irqsave+0x8b/0xf0 ? __pfx__raw_spin_lock_irqsave+0x10/0x10 ? depot_alloc_stack+0x4c/0x1f0 ? __raw_spin_unlock_irqrestore+0xe/0x30 ? stack_depot_save_flags+0x35d/0x4f0 ? kasan_save_stack+0x34/0x50 ? kasan_save_stack+0x24/0x50 ? mutex_lock+0x91/0xe0 ? __pfx_mutex_lock+0x10/0x10 prepare_uprobe_buffer.part.0+0x2cd/0x500 uprobe_dispatcher+0x2c3/0x6a0 ? __pfx_uprobe_dispatcher+0x10/0x10 ? __kasan_slab_alloc+0x4d/0x90 handler_chain+0xdd/0x3e0 handle_swbp+0x26e/0x3d0 ? __pfx_handle_swbp+0x10/0x10 ? uprobe_pre_sstep_notifier+0x151/0x1b0 irqentry_exit_to_user_mode+0xe2/0x1b0 asm_exc_int3+0x39/0x40 RIP: 0033:0x401199 Code: 01 c2 0f b6 45 fb 88 02 83 45 fc 01 8b 45 fc 3b 45 e4 7c b7 8b 45 e4 48 98 48 8d 50 ff 48 8b 45 e8 48 01 d0 ce RSP: 002b:00007ffdf00576a8 EFLAGS: 00000206 RAX: 00007ffdf00576b0 RBX: 0000000000000000 RCX: 00000000000000ff RDX: 00000000000000ff RSI: 00000000000000ff RDI: 00007ffdf00576b0 RBP: 00007ffdf00586b0 R08: 00007feb2f9c0d20 R09: 00007feb2f9c0d20 R10: 0000000000000001 R11: 0000000000000202 R12: 0000000000401040 R13: 00007ffdf0058780 R14: 0000000000000000 R15: 0000000000000000 </TASK> This commit enforces the buffer's maxlen less than a page-size to avoid store_trace_args() out-of-memory access.

CVE ID	Description
CVE-2024-50073	In the Linux kernel, the following vulnerability has been resolved: tty: n_gsm: Fix use-after-free in gsm_cleanup_mux BUG: KASAN: slab-use-after-free in gsm_cleanup_mux+0x77b/0x7b0 drivers/tty/n_gsm.c:3160 [n_gsm] Read of size 8 at addr ffff88815fe99c00 by task poc/3379 CPU: 0 UID: 0 PID: 3379 Comm: poc Not tainted 6.11.0+ #56 Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 Call Trace: <TASK> gsm_cleanup_mux+0x77b/0x7b0 drivers/tty/n_gsm.c:3160 [n_gsm] __pfx_gsm_cleanup_mux+0x10/0x10 drivers/tty/n_gsm.c:3124 [n_gsm] __pfx_sched_clock_cpu+0x10/0x10 kernel/sched/clock.c:389 update_load_avg+0x1c1/0x27b0 kernel/sched/fair.c:4500 __pfx_min_vruntime_cb_rotate+0x10/0x10 kernel/sched/fair.c:846 __rb_insert_augmented+0x492/0xbf0 lib/rbtree.c:161 gsmld_ioctl+0x395/0x1450 drivers/tty/n_gsm.c:3408 [n_gsm] _raw_spin_lock_irqsave+0x92/0xf0 arch/x86/include/asm/atomic.h:107 __pfx_gsmld_ioctl+0x10/0x10 drivers/tty/n_gsm.c:3822 [n_gsm] ktime_get+0x5e/0x140 kernel/time/timekeeping.c:195 ldsem_down_read+0x94/0x4e0 arch/x86/include/asm/atomic64_64.h:79 __pfx_ldsem_down_read+0x10/0x10 drivers/tty/tty_ldsem.c:338 __pfx_do_vfs_ioctl+0x10/0x10 fs/ioctl.c:805 tty_ioctl+0x643/0x1100 drivers/tty/tty_io.c:2818 Allocated by task 65: gsm_data_alloc.constprop.0+0x27/0x190 drivers/tty/n_gsm.c:926 [n_gsm] gsm_send+0x2c/0x580 drivers/tty/n_gsm.c:819 [n_gsm] gsm1_receive+0x547/0xad0 drivers/tty/n_gsm.c:3038 [n_gsm] gsmld_receive_buf+0x176/0x280 drivers/tty/n_gsm.c:3609 [n_gsm] tty_ldisc_receive_buf+0x101/0x1e0 drivers/tty/tty_buffer.c:391 tty_port_default_receive_buf+0x61/0xa0 drivers/tty/tty_port.c:39 flush_to_ldisc+0x1b0/0x750 drivers/tty/tty_buffer.c:445 process_scheduled_works+0x2b0/0x10d0 kernel/workqueue.c:3229 worker_thread+0x3dc/0x950 kernel/workqueue.c:3391 kthread+0x2a3/0x370 kernel/kthread.c:389 ret_from_fork+0x2d/0x70 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:257 Freed by task 3367: kfree+0x126/0x420 mm/slub.c:4580 gsm_cleanup_mux+0x36c/0x7b0 drivers/tty/n_gsm.c:3160 [n_gsm] gsmld_ioctl+0x395/0x1450 drivers/tty/n_gsm.c:3408 [n_gsm] tty_ioctl+0x643/0x1100 drivers/tty/tty_io.c:2818 [Analysis] gsm_msg on the tx_ctrl_list or tx_data_list of gsm_mux can be freed by multi threads through ioctl, which leads to the occurrence of uaf. Protect it by gsm tx lock.
CVE-2024-50074	In the Linux kernel, the following vulnerability has been resolved: parport: Proper fix for array out-of-bounds access The recent fix for array out-of-bounds accesses replaced sprintf() calls blindly with snprintf(). However, since snprintf() returns the would-be-printed size, not the actually output size, the length calculation can still go over the given limit. Use scnprintf() instead of snprintf(), which returns the actually output letters, for addressing the potential out-of-bounds access properly.

CVE ID	Description
CVE-2024-50083	In the Linux kernel, the following vulnerability has been resolved: tcp: fix mptcp DSS corruption due to large pmtu xmit Syzkaller was able to trigger a DSS corruption: TCP: request_sock_subflow_v4: Possible SYN flooding on port [:]:20002. Sending cookies. -----[cut here]----- WARNING: CPU: 0 PID: 5227 at net/mptcp/protocol.c:695 __mptcp_move_skbs_from_subflow+0x20a9/0x21f0 net/mptcp/protocol.c:695 Modules linked in: CPU: 0 UID: 0 PID: 5227 Comm: syz-executor350 Not tainted 6.11.0-syzkaller-08829-gaf9c191ac2a0 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 RIP: 0010: __mptcp_move_skbs_from_subflow+0x20a9/0x21f0 net/mptcp/protocol.c:695 Code: 0f b6 dc 31 ff 89 de e8 b5 dd ea f5 89 d8 48 81 c4 50 01 00 00 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc e8 98 da ea f5 90 <0f> 0b 90 e9 47 ff ff e8 8a da ea f5 90 0f 0b 90 e9 99 e0 ff ff RSP: 0018:ffffc90000006db8 EFLAGS: 00010246 RAX: ffffffff8ba9df18 RBX: 00000000000005f0 RCX: ffff888030023c00 RDX: 0000000000000100 RSI: 000000000000081e5 RDI: 00000000000005f0 RBP: 1ffff110062bf1ae R08: ffffffff8ba9cf12 R09: 1ffff110062bf1b8 R10: dffffc0000000000 R11: ffffed10062bf1b9 R12: 0000000000000000 R13: dffffc0000000000 R14: 00000000700cec61 R15: 000000000000081e5 FS: 000055556679c380(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000020287000 CR3: 0000000077892000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff0fff DR7: 00000000000000400 Call Trace: <IRQ> move_skbs_to_msk net/mptcp/protocol.c:811 [inline] mptcp_data_ready+0x29c/0xa90 net/mptcp/protocol.c:854 subflow_data_ready+0x34a/0x920 net/mptcp/subflow.c:1490 tcp_data_queue+0x20fd/0x76c0 net/ipv4/tcp_input.c:5283 tcp_rcv_established+0xfba/0x2020 net/ipv4/tcp_input.c:6237 tcp_v4_do_rcv+0x96d/0xc70 net/ipv4/tcp_ipv4.c:1915 tcp_v4_rcv+0x2dc0/0x37f0 net/ipv4/tcp_ipv4.c:2350 ip_protocol_deliver_rcu+0x22e/0x440 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x341/0x5f0 net/ipv4/ip_input.c:233 NF_HOOK+0x3a4/0x450 include/linux/netfilter.h:314 NF_HOOK+0x3a4/0x450 include/linux/netfilter.h:314 __netif_receive_skb_one_core net/core/dev.c:5662 [inline] __netif_receive_skb+0x2bf/0x650 net/core/dev.c:5775 process_backlog+0x662/0x15b0 net/core/dev.c:6107 __napi_poll+0xcb/0x490 net/core/dev.c:6771 napi_poll net/core/dev.c:6840 [inline] net_rx_action+0x89b/0x1240 net/core/dev.c:6962 handle_softirqs+0x2c5/0x980 kernel/softirq.c:554 do_softirq+0x11b/0x1e0 kernel/softirq.c:455 <IRQ> <TASK> __local_bh_enable_ip+0x1bb/0x200 kernel/softirq.c:382 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:919 [inline] __dev_queue_xmit+0x1764/0x3e80 net/core/dev.c:4451 dev_queue_xmit include/linux/netdevice.h:3094 [inline] neigh_hh_output include/net/neighbour.h:526 [inline] neigh_output include/net/neighbour.h:540 [inline] ip_finish_output2+0xd41/0x1390 net/ipv4/ip_output.c:236 ip_local_out net/ipv4/ip_output.c:130 [inline] __ip_queue_xmit+0x118c/0x1b80 net/ipv4/ip_output.c:536 __tcp_transmit_skb+0x2544/0x3b30 net/ipv4/tcp_output.c:1466 tcp_transmit_skb net/ipv4/tcp_output.c:1484 [inline] tcp_mtu_probe net/ipv4/tcp_output.c:2547 [inline] tcp_write_xmit+0x641d/0x6bf0 net/ipv4/tcp_output.c:2752 __tcp_push_pending_frames+0x9b/0x360 net/ipv4/tcp_output.c:3015 tcp_push_pending_frames include/net/tcp.h:2107 [inline] tcp_data_snd_check net/ipv4/tcp_input.c:5714 [inline] tcp_rcv_established+0x1026/0x2020 net/ipv4/tcp_input.c:6239 tcp_v4_do_rcv+0x96d/0xc70 net/ipv4/tcp_ipv4.c:1915 sk_backlog_rcv include/net/sock.h:1113 [inline] __release_sock+0x214/0x350 net/core/sock.c:3072 release_sock+0x61/0x1f0 net/core/sock.c:3626 mptcp_push_ ---truncated---
CVE-2024-50091	In the Linux kernel, the following vulnerability has been resolved: dm vdo: don't refer to dedupe_context after releasing it Clear the dedupe_context pointer in a data_vio whenever ownership of the context is lost, so that vdo can't examine it accidentally.
CVE-2024-50093	In the Linux kernel, the following vulnerability has been resolved: thermal: intel: int340x: processor: Fix warning during module unload The processor_thermal driver uses pcim_device_enable() to enable a PCI device, which means the device will be automatically disabled on driver detach. Thus there is no need to call pci_disable_device() again on it. With recent PCI device resource management improvements, e.g. commit f748a07a0b64 ("PCI: Remove legacy pcim_release()"), this problem is exposed and triggers the warning below. [224.010735] proc_thermal_pci 0000:00:04.0: disabling already-disabled device [224.010747] WARNING: CPU: 8 PID: 4442 at drivers/pci/pci.c:2250 pci_disable_device+0xe5/0x100 ... [224.010844] Call Trace: [224.010845] <TASK> [224.010847] ? show_regs+0x6d/0x80 [224.010851] ? __warn+0x8c/0x140 [224.010854] ? pci_disable_device+0xe5/0x100 [224.010856] ? report_bug+0x1c9/0x1e0 [224.010859] ? handle_bug+0x46/0x80 [224.010862] ? exc_invalid_op+0x1d/0x80 [224.010863] ? asm_exc_invalid_op+0x1f/0x30 [224.010867] ? pci_disable_device+0xe5/0x100 [224.010869] ? pci_disable_device+0xe5/0x100 [224.010871] ? kfree+0x21a/0x2b0 [224.010873] pcim_disable_device+0x20/0x30 [224.010875] devm_action_release+0x16/0x20 [224.010878] release_nodes+0x47/0xc0 [224.010880] devres_release_all+0x9f/0xe0 [224.010883] device_unbind_cleanup+0x12/0x80 [224.010885] device_release_driver_internal+0x1ca/0x210 [224.010887] driver_detach+0x4e/0xa0 [224.010889] bus_remove_driver+0x6f/0xf0 [224.010890] driver_unregister+0x35/0x60 [224.010892] pci_unregister_driver+0x44/0x90 [224.010894] proc_thermal_pci_driver_exit+0x14/0x5f0 [processor_thermal_device_pci] ... [224.010921] ---[end trace 0000000000000000]--- Remove the excess pci_disable_device() calls. [rjw: Subject and changelog edits]

CVE ID	Description
CVE-2024-50095	In the Linux kernel, the following vulnerability has been resolved: RDMA/mad: Improve handling of timed out WRs of mad agent Current timeout handler of mad agent acquires/releases mad_agent_priv lock for every timed out WRs. This causes heavy locking contention when higher no. of WRs are to be handled inside timeout handler. This leads to softlockup with below trace in some use cases where rdma-cm path is used to establish connection between peer nodes Trace: ----- BUG: soft lockup - CPU#4 stuck for 26s! [kworker/u128:3:19767] CPU: 4 PID: 19767 Comm: kworker/u128:3 Kdump: loaded Tainted: G OE ----- --- 5.14.0-427.13.1.el9_4.x86_64 #1 Hardware name: Dell Inc. PowerEdge R740/01YM03, BIOS 2.4.8 11/26/2019 Workqueue: ib_mad1 timeout_sends [ib_core] RIP: 0010: __do_softirq+0x78/0x2ac RSP: 0018:ffffb253449e4f98 EFLAGS: 00000246 RAX: 00000000fffffffb RBX: 0000000000000000 RCX: 000000000000001f RDX: 000000000000001d RSI: 000000003d1879ab RDI: fff363b66fd3a86b RBP: fffffb253604cbcd8 R08: 0000009065635f3b R09: 0000000000000000 R10: 0000000000000040 R11: fffffb253449e4ff8 R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000040 FS: 0000000000000000(0000) GS:ffff8caa1fc80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fd9ec9db900 CR3: 0000000891934006 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 PKRU: 55555554 Call Trace: <IRQ> ? show_trace_log_lvl+0x1c4/0x2df ? show_trace_log_lvl+0x1c4/0x2df ? __irq_exit_rcu+0xa1/0xc0 ? watchdog_timer_fn+0x1b2/0x210 ? __pfx_watchdog_timer_fn+0x10/0x10 ? __hrtimer_run_queues+0x127/0x2c0 ? hrtimer_interrupt+0xfc/0x210 ? __sysvec_apic_timer_interrupt+0x5c/0x110 ? sysvec_apic_timer_interrupt+0x37/0x90 ? asm_sysvec_apic_timer_interrupt+0x16/0x20 ? __do_softirq+0x78/0x2ac ? __do_softirq+0x60/0x2ac __irq_exit_rcu+0xa1/0xc0 sysvec_call_function_single+0x72/0x90 </IRQ> <TASK> asm_sysvec_call_function_single+0x16/0x20 RIP: 0010: _raw_spin_unlock_irq+0x14/0x30 RSP: 0018:ffffb253604cbd88 EFLAGS: 00000247 RAX: 00000000001960d RBX: 0000000000000002 RCX: ffff8cad2a064800 RDX: 000000008020001b RSI: 0000000000000001 RDI: ffff8cad5d39f66c RBP: ffff8cad5d39f600 R08: 0000000000000001 R09: 0000000000000000 R10: ffff8caa443e0c00 R11: fffffb253604cbcd8 R12: ffff8cacb8682538 R13: 0000000000000005 R14: fffffb253604cbd90 R15: ffff8cad5d39f66c cm_process_send_error+0x122/0x1d0 [ib_cm] timeout_sends+0x1dd/0x270 [ib_core] process_one_work+0x1e2/0x3b0 ? __pfx_worker_thread+0x10/0x10 worker_thread+0x50/0x3a0 ? __pfx_worker_thread+0x10/0x10 kthread+0xdd/0x100 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x29/0x50 </TASK> Simplified timeout handler by creating local list of timed out WRs and invoke send handler post creating the list. The new method acquires/ releases lock once to fetch the list and hence helps to reduce locking contention when processing higher no. of WRs
CVE-2024-50102	In the Linux kernel, the following vulnerability has been resolved: x86: fix user address masking non-canonical speculation issue It turns out that AMD has a "Meltdown Lite(tm)" issue with non-canonical accesses in kernel space. And so using just the high bit to decide whether an access is in user space or kernel space ends up with the good old "leak speculative data" if you have the right gadget using the result: CVE-2020-12965 "Transient Execution of Non-Canonical Accesses" Now, the kernel surrounds the access with a STAC/CLAC pair, and those instructions end up serializing execution on older Zen architectures, which closes the speculation window. But that was true only up until Zen 5, which renames the AC bit [1]. That improves performance of STAC/CLAC a lot, but also means that the speculation window is now open. Note that this affects not just the new address masking, but also the regular valid_user_address() check used by access_ok(), and the asm version of the sign bit check in the get_user() helpers. It does not affect put_user() or clear_user() variants, since there's no speculative result to be used in a gadget for those operations.
CVE-2024-50103	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: Fix NULL Dereference in asoc_qcom_lpass_cpu_platform_probe() A devm_kzalloc() in asoc_qcom_lpass_cpu_platform_probe() could possibly return NULL pointer. NULL Pointer Dereference may be triggered without additional check. Add a NULL check for the returned pointer.

CVE ID	Description
CVE-2024-50104	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: sdm845: add missing soundwire runtime stream alloc During the migration of Soundwire runtime stream allocation from the Qualcomm Soundwire controller to SoC's soundcard drivers the sdm845 soundcard was forgotten. At this point any playback attempt or audio daemon startup, for instance on sdm845-db845c (Qualcomm RB3 board), will result in stream pointer NULL dereference: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000020 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtable: 4k pages, 48-bit VAs, pgdp=0000000101ecf000 [0000000000000020] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP Modules linked in: ... CPU: 5 UID: 0 PID: 1198 Comm: aplay Not tainted 6.12.0-rc2-qcoml-arm64-00059-g9d78f315a362-dirty #18 Hardware name: Thundercomm Dragonboard 845c (DT) pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : sdw_stream_add_slave+0x44/0x380 [soundwire_bus] lr : sdw_stream_add_slave+0x44/0x380 [soundwire_bus] sp : ffff80008a2035c0 x29: ffff80008a2035c0 x28: ffff80008a203978 x27: 0000000000000000 x26: 00000000000000c0 x25: 0000000000000000 x24: ffff1676025f4800 x23: ffff167600ff1cb8 x22: ffff167600ff1c98 x21: 0000000000000003 x20: ffff167607316000 x19: ffff167604e64e80 x18: 0000000000000000 x17: 0000000000000000 x16: ffffcec265074160 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000000 x10: 0000000000000000 x9 : 0000000000000000 x8 : 0000000000000000 x7 : 0000000000000000 x6 : ffff167600ff1cec x5 : ffffcec22cfa2010 x4 : 0000000000000000 x3 : 0000000000000003 x2 : ffff167613f836c0 x1 : 0000000000000000 x0 : ffff16761feb60b8 Call trace: sdw_stream_add_slave+0x44/0x380 [soundwire_bus] wsa881x_hw_params+0x68/0x80 [snd_soc_wsa881x] snd_soc_dai_hw_params+0x3c/0xa4 __soc_pcm_hw_params+0x230/0x660 dpcm_be_dai_hw_params+0x1d0/0x3f8 dpcm_fe_dai_hw_params+0x98/0x268 snd_pcm_hw_params+0x124/0x460 snd_pcm_common_ioctl+0x998/0x16e8 snd_pcm_ioctl+0x34/0x58 __arm64_sys_ioctl+0xac/0xf8 invoke_syscall+0x48/0x104 el0_svc_common.constprop.0+0x40/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x34/0xe0 el0t_64_sync_handler+0x120/0x12c el0t_64_sync+0x190/0x194 Code: aa0403fb f9418400 9100e000 9400102f (f8420f22) ---[end trace 0000000000000000]--- 00000000000006108 <sdw_stream_add_slave>: 6108: d503233f paciasp 610c: a9b97bfd stp x29, x30, [sp, #-112]! 6110: 910003fd mov x29, sp 6114: a90153f3 stp x19, x20, [sp, #16] 6118: a9025bf5 stp x21, x22, [sp, #32] 611c: aa0103f6 mov x22, x1 6120: 2a0303f5 mov w21, w3 6124: a90363f7 stp x23, x24, [sp, #48] 6128: aa0003f8 mov x24, x0 612c: aa0203f7 mov x23, x2 6130: a9046bf9 stp x25, x26, [sp, #64] 6134: aa0403f9 mov x25, x4 <-- x4 copied to x25 6138: a90573fb stp x27, x28, [sp, #80] 613c: aa0403fb mov x27, x4 6140: f9418400 ldr x0, [x0, #776] 6144: 9100e000 add x0, x0, #0x38 6148: 94000000 bl 0 <mutex_lock> 614c: f8420f22 ldr x2, [x25, #32]! <-- offset 0x44 ^^ This is 0x6108 + offset 0x44 from the beginning of sdw_stream_add_slave() where data abort happens. wsa881x_hw_params() is called with stream = NULL and passes it further in register x4 (5th argu --- truncated---
CVE-2024-50105	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: sc7280: Fix missing Soundwire runtime stream alloc Commit 15c7fab0e047 ("ASoC: qcom: Move Soundwire runtime stream alloc to soundcards") moved the allocation of Soundwire stream runtime from the Qualcomm Soundwire driver to each individual machine sound card driver, except that it forgot to update SC7280 card. Just like for other Qualcomm sound cards using Soundwire, the card driver should allocate and release the runtime. Otherwise sound playback will result in a NULL pointer dereference or other effect of uninitialized memory accesses (which was confirmed on SDM845 having similar issue).

CVE ID	Description
CVE-2024-50106	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix race between laundromat and free_stateid There is a race between laundromat handling of revoked delegations and a client sending free_stateid operation. Laundromat thread finds that delegation has expired and needs to be revoked so it marks the delegation stid revoked and it puts it on a reaper list but then it unlock the state lock and the actual delegation revocation happens without the lock. Once the stid is marked revoked a racing free_stateid processing thread does the following (1) it calls list_del_init() which removes it from the reaper list and (2) frees the delegation stid structure. The laundromat thread ends up not calling the revoke_delegation() function for this particular delegation but that means it will no release the lock lease that exists on the file. Now, a new open for this file comes in and ends up finding that lease list isn't empty and calls nfsd_breaker_owns_lease() which ends up trying to dereference a freed delegation stateid. Leading to the followint use-after-free KASAN warning: <pre>kernel: ===== kernel: BUG: KASAN: slab-use-after-free in nfsd_breaker_owns_lease+0x140/0x160 [nfsd] kernel: Read of size 8 at addr ffff0000e73cd0c8 by task nfsd/6205 kernel: kernel: CPU: 2 UID: 0 PID: 6205 Comm: nfsd Kdump: loaded Not tainted 6.11.0-rc7+ #9 kernel: Hardware name: Apple Inc. Apple Virtualization Generic Platform, BIOS 2069.0.0.0.0 08/03/2024 kernel: Call trace: kernel: dump_backtrace+0x98/0x120 kernel: show_stack+0x1c/0x30 kernel: dump_stack_lvl +0x80/0xe8 kernel: print_address_description.constprop.0+0x84/0x390 kernel: print_report +0xa4/0x268 kernel: kasan_report+0xb4/0xf8 kernel: __asan_report_load8_noabort+0x1c/0x28 kernel: nfsd_breaker_owns_lease+0x140/0x160 [nfsd] kernel: nfsd_file_do_acquire+0xb3c/0x11d0 [nfsd] kernel: nfsd_file_acquire_opened+0x84/0x110 [nfsd] kernel: nfs4_get_vfs_file+0x634/0x958 [nfsd] kernel: nfsd4_process_open2+0xa40/0x1a40 [nfsd] kernel: nfsd4_open+0xa08/0xe80 [nfsd] kernel: nfsd4_proc_compound+0xb8c/0x2130 [nfsd] kernel: nfsd_dispatch+0x22c/0x718 [nfsd] kernel: svc_process_common+0x8e8/0x1960 [sunrpc] kernel: svc_process+0x3d4/0x7e0 [sunrpc] kernel: svc_handle_xprt+0x828/0xe10 [sunrpc] kernel: svc_rcv+0x2cc/0x6a8 [sunrpc] kernel: nfsd +0x270/0x400 [nfsd] kernel: kthread+0x288/0x310 kernel: ret_from_fork+0x10/0x20 This patch proposes a fixed that's based on adding 2 new additional stid's sc_status values that help coordinate between the laundromat and other operations (nfsd4_free_stateid() and nfsd4_delegreturn()). First to make sure, that once the stid is marked revoked, it is not removed by the nfsd4_free_stateid(), the laundromat take a reference on the stateid. Then, coordinating whether the stid has been put on the cl_revoked list or we are processing FREE_STATEID and need to make sure to remove it from the list, each check that state and act accordingly. If laundromat has added to the cl_revoke list before the arrival of FREE_STATEID, then nfsd4_free_stateid() knows to remove it from the list. If nfsd4_free_stateid() finds that operations arrived before laundromat has placed it on cl_revoke list, it marks the state freed and then laundromat will no longer add it to the list. Also, for nfsd4_delegreturn() when looking for the specified stid, we need to access stid that are marked removed or freeable, it means the laundromat has started processing it but hasn't finished and this delegreturn needs to return nfserr_deleg_revoked and not nfserr_bad_stateid. The latter will not trigger a FREE_STATEID and the lack of it will leave this stid on the cl_revoked list indefinitely.</pre>

CVE ID	Description
CVE-2024-50107	In the Linux kernel, the following vulnerability has been resolved: platform/x86/intel/pmc: Fix pmc_core_iounmap to call iounmap for valid addresses Commit 50c6dbdf16e ("x86/ioremap: Improve iounmap() address range checks") introduces a WARN when address ranges of iounmap are invalid. On Thinkpad P1 Gen 7 (Meteor Lake-P) this caused the following warning to appear: WARNING: CPU: 7 PID: 713 at arch/x86/mm/ioremap.c:461 iounmap+0x58/0x1f0 Modules linked in: rfkill(+) snd_timer(+) fjes(+) snd soundcore intel_pmc_core(+) int3403_thermal(+) int340x_thermal_zone intel_vsec pmt_telemetry acpi_pad pmt_class acpi_tad int3400_thermal acpi_thermal_rel joydev loop nfnetlink zram xe drm_suballoc_helper nouveau i915 mxm_wmi drm_ttm_helper gpu_sched drm_gpuvm drm_exec drm_buddy i2c_algo_bit crct10dif_pclmul crc32_pclmul ttm crc32c_intel polyval_clmulni rtsx_pci_sdmmc ucsi_acpi polyval_generic mmc_core hid_multitouch drm_display_helper ghash_clmulni_intel typec_ucsi nvme sha512_ssse3 video sha256_ssse3 nvme_core intel_vpu sha1_ssse3 rtsx_pci_cec typec nvme_auth i2c_hid_acpi i2c_hid wmi pinctrl_meteorlake serio_raw ip6_tables ip_tables fuse CPU: 7 UID: 0 PID: 713 Comm: (udev-worker) Not tainted 6.12.0-rc2iounmap+ #42 Hardware name: LENOVO 21KWCTO1WW/21KWCTO1WW, BIOS N48ET19W (1.06) 07/18/2024 RIP: 0010:iounmap+0x58/0x1f0 Code: 85 6a 01 00 00 48 8b 05 e6 e2 28 04 48 39 c5 72 19 eb 26 cc cc cc 48 ba 00 00 00 00 00 00 32 00 48 8d 44 02 ff 48 39 c5 72 23 <0f> 0b 48 83 c4 08 5b 5d 41 5c c3 cc cc cc cc 48 ba 00 00 00 00 00 RSP: 0018:ffff888131eff038 EFLAGS: 00010207 RAX: fffffc9000000000 RBX: 0000000000000000 RCX: ffff888e33b80000 RDX: dffffc0000000000 RSI: ffff888e33bc29c0 RDI: 0000000000000000 RBP: 0000000000000000 R08: ffff8881598a8000 R09: ffff888e2ccedc10 R10: 0000000000000003 R11: ffffffff3367634 R12: 00000000fe000000 R13: ffff888101d0da28 R14: ffffffff2e437e0 R15: ffff888110b03b28 FS: 00007f3c1d4b3980(0000) GS:ffff888e33b80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005651cf93578 CR3: 0000000124e4c002 CR4: 0000000000f70ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff07f0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> ? __warn.cold+0xb6/0x176 ? iounmap+0x58/0x1f0 ? report_bug+0x1f4/0x2b0 ? handle_bug+0x58/0x90 ? exc_invalid_op+0x17/0x40 ? asm_exc_invalid_op+0x1a/0x20 ? iounmap+0x58/0x1f0 pmc_core_ssram_get_pmc+0x477/0x6c0 [intel_pmc_core] ? __pfx_pmc_core_ssram_get_pmc+0x10/0x10 [intel_pmc_core] ? __pfx_do_pci_enable_device+0x10/0x10 ? pci_wait_for_pending+0x60/0x110 ? pci_enable_device_flags+0x1e3/0x2e0 ? __pfx_mtl_core_init+0x10/0x10 [intel_pmc_core] pmc_core_ssram_init+0x7f/0x110 [intel_pmc_core] mtl_core_init+0xda/0x130 [intel_pmc_core] ? __mutex_init+0xb9/0x130 pmc_core_probe+0x27e/0x10b0 [intel_pmc_core] ? _raw_spin_lock_irqsave+0x96/0xf0 ? __pfx_pmc_core_probe+0x10/0x10 [intel_pmc_core] ? __pfx_mutex_unlock+0x10/0x10 ? __pfx_mutex_lock+0x10/0x10 ? device_pm_check_callbacks+0x82/0x370 ? acpi_dev_pm_attach+0x234/0x2b0 platform_probe+0x9f/0x150 really_probe+0x1e0/0x8a0 __driver_probe_device+0x18c/0x370 ? __pfx__driver_attach+0x10/0x10 driver_probe_device+0x4a/0x120 __driver_attach+0x190/0x4a0 ? __pfx__driver_attach+0x10/0x10 bus_for_each_dev+0x103/0x180 ? __pfx_bus_for_each_dev+0x10/0x10 ? klist_add_tail+0x136/0x270 bus_add_driver+0x2fc/0x540 driver_register+0x1a5/0x360 ? __pfx_pmc_core_driver_init+0x10/0x10 [intel_pmc_core] do_one_initcall+0xa4/0x380 ? __pfx_do_one_initcall+0x10/0x10 ? kasan_unpoison+0x44/0x70 do_init_module+0x296/0x800 load_module+0x5090/0x6ce0 ? __pfx_load_module+0x10/0x10 ? ima_post_read_file+0x193/0x200 ? __pfx_ima_post_read_file+0x10/0x10 ? rw_verify_area+0x152/0x4c0 ? kernel_read_file+0x257/0x750 ? __pfx_kernel_read_file+0x10/0x10 ? __pfx_filemap_get_read_batch+0x10/0x10 ? init_module_from_file+0xd1/0x130 init_module_from_file+0xd1/0x130 ? __pfx_init_module_from_file+0x10/0 ---truncated---
CVE-2024-50108	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Disable PSR-SU on Parade 08-01 TCON too Stuart Hayhurst has found that both at bootup and fullscreen VA-API video is leading to black screens for around 1 second and kernel WARNING [1] traces when calling dmub_psr_enable() with Parade 08-01 TCON. These symptoms all go away with PSR-SU disabled for this TCON, so disable it for now while DMUB traces [2] from the failure can be analyzed and the failure state properly root caused. (cherry picked from commit afb634a6823d8d9db23c5fb04f79c5549349628b)

CVE ID	Description
CVE-2024-50110	In the Linux kernel, the following vulnerability has been resolved: xfrm: fix one more kernel-infoleak in algo dumping During fuzz testing, the following issue was discovered: BUG: KMSAN: kernel-infoleak in _copy_to_iter+0x598/0x2a30 _copy_to_iter+0x598/0x2a30 __skb_datagram_iter+0x168/0x1060 skb_copy_datagram_iter+0x5b/0x220 netlink_rcvmsg+0x362/0x1700 sock_rcvmsg+0x2dc/0x390 __sys_recvfrom+0x381/0x6d0 __x64_sys_recvfrom+0x130/0x200 x64_sys_call+0x32c8/0x3cc0 do_syscall_64+0xd8/0x1c0 entry_SYSCALL_64_after_hwframe+0x79/0x81 Uninit was stored to memory at: copy_to_user_state_extra+0xcc1/0x1e00 dump_one_state+0x28c/0x5f0 xfrm_state_walk+0x548/0x11e0 xfrm_dump_sa+0x1e0/0x840 netlink_dump+0x943/0x1c40 __netlink_dump_start+0x746/0xdb0 xfrm_user_rcv_msg+0x429/0xc00 netlink_rcv_skb+0x613/0x780 xfrm_netlink_rcv+0x77/0xc0 netlink_unicast+0xe90/0x1280 netlink_sendmsg+0x126d/0x1490 __sock_sendmsg+0x332/0x3d0 __sys_sendmsg+0x863/0xc30 __sys_sendmsg+0x285/0x3e0 __x64_sys_sendmsg+0x2d6/0x560 x64_sys_call+0x1316/0x3cc0 do_syscall_64+0xd8/0x1c0 entry_SYSCALL_64_after_hwframe+0x79/0x81 Uninit was created at: __kmallocc+0x571/0xd30 attach_auth+0x106/0x3e0 xfrm_add_sa+0x2aa0/0x4230 xfrm_user_rcv_msg+0x832/0xc00 netlink_rcv_skb+0x613/0x780 xfrm_netlink_rcv+0x77/0xc0 netlink_unicast+0xe90/0x1280 netlink_sendmsg+0x126d/0x1490 __sock_sendmsg+0x332/0x3d0 __sys_sendmsg+0x863/0xc30 __sys_sendmsg+0x285/0x3e0 __x64_sys_sendmsg+0x2d6/0x560 x64_sys_call+0x1316/0x3cc0 do_syscall_64+0xd8/0x1c0 entry_SYSCALL_64_after_hwframe+0x79/0x81 Bytes 328-379 of 732 are uninitialized Memory access of size 732 starts at ffff88800e18e000 Data copied to user address 00007ff30f48aff0 CPU: 2 PID: 18167 Comm: syz-executor.0 Not tainted 6.8.11 #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Fixes copying of xfrm algorithms where some random data of the structure fields can end up in userspace. Padding in structures may be filled with random (possibly sensitive) data and should never be given directly to user-space. A similar issue was resolved in the commit 8222d5910dae ("xfrm: Zero padding when dumping algos and encap") Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2024-50111	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Enable IRQ if do_ale() triggered in irq-enabled context Unaligned access exception can be triggered in irq-enabled context such as user mode, in this case do_ale() may call get_user() which may cause sleep. Then we will get: BUG: sleeping function called from invalid context at arch/loongarch/kernel/access-helper.h:7 in_atomic(): 0, irqs_disabled(): 1, non_block: 0, pid: 129, name: modprobe preempt_count: 0, expected: 0 RCU nest depth: 0, expected: 0 CPU: 0 UID: 0 PID: 129 Comm: modprobe Tainted: G W 6.12.0-rc1+ #1723 Tainted: [W]=WARN Stack : 9000000105e0bd48 0000000000000000 9000000003803944 9000000105e08000 9000000105e0bc70 9000000105e0bc78 0000000000000000 0000000000000000 9000000105e0bc78 0000000000000001 9000000185e0ba07 9000000105e0b890 ffffffff00000000 9000000105e0bc78 73924b81763be05b 9000000100194500 000000000000020c 000000000000000a 0000000000000000 0000000000000003 000000000000023f0 00000000000e1401 00000000072f8000 0000007fbb0e260 0000000000000000 0000000000000000 9000000005437650 90000000055d5000 0000000000000000 0000000000000003 0000007fbb0e1f0 0000000000000000 0000005567b00490 0000000000000000 9000000003803964 0000007fbb0dfec 00000000000000b0 0000000000000007 0000000000000003 0000000000071c1d ... Call Trace: [<9000000003803964>] show_stack+0x64/0x1a0 [<9000000004c57464>] dump_stack_lvl+0x74/0xb0 [<9000000003861ab4>] __might_resched+0x154/0x1a0 [<900000000380c96c>] emulate_load_store_insn+0x6c/0xf60 [<9000000004c58118>] do_ale+0x78/0x180 [<9000000003801bc8>] handle_ale+0x128/0x1e0 So enable IRQ if unaligned access exception is triggered in irq-enabled context to fix it.
CVE-2024-50112	In the Linux kernel, the following vulnerability has been resolved: x86/lam: Disable ADDRESS_MASKING in most cases Linear Address Masking (LAM) has a weakness related to transient execution as described in the SLAM paper[1]. Unless Linear Address Space Separation (LASS) is enabled this weakness may be exploitable. Until kernel adds support for LASS[2], only allow LAM for COMPILE_TEST, or when speculation mitigations have been disabled at compile time, otherwise keep LAM disabled. There are no processors in market that support LAM yet, so currently nobody is affected by this issue. [1] SLAM: https://download.vusec.net/papers/slam_sp24.pdf [2] LASS: https://lore.kernel.org/lkml/20230609183632.48706-1-alexander.shishkin@linux.intel.com/ [dhansen: update SPECULATION_MITIGATIONS -> CPU_MITIGATIONS]
CVE-2024-50115	In the Linux kernel, the following vulnerability has been resolved: KVM: nSVM: Ignore nCR3[4:0] when loading PDPTEs from memory Ignore nCR3[4:0] when loading PDPTEs from memory for nested SVM, as bits 4:0 of CR3 are ignored when PAE paging is used, and thus VMRUN doesn't enforce 32-byte alignment of nCR3. In the absolute worst case scenario, failure to ignore bits 4:0 can result in an out-of-bounds read, e.g. if the target page is at the end of a memslot, and the VMM isn't using guard pages. Per the APM: The CR3 register points to the base address of the page-directory-pointer table. The page-directory-pointer table is aligned on a 32-byte boundary, with the low 5 address bits 4:0 assumed to be 0. And the SDM's much more explicit: 4:0 Ignored Note, KVM gets this right when loading PDPTRs, it's only the nSVM flow that is broken.

CVE ID	Description
CVE-2024-50116	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix kernel bug due to missing clearing of buffer delay flag Syzbot reported that after nilfs2 reads a corrupted file system image and degrades to read-only, the BUG_ON check for the buffer delay flag in submit_bh_wbc() may fail, causing a kernel bug. This is because the buffer delay flag is not cleared when clearing the buffer state flags to discard a page/folio or a buffer head. So, fix this. This became necessary when the use of nilfs2's own page clear routine was expanded. This state inconsistency does not occur if the buffer is written normally by log writing.
CVE-2024-50117	In the Linux kernel, the following vulnerability has been resolved: drm/amd: Guard against bad data for ATIF ACPI method If a BIOS provides bad data in response to an ATIF method call this causes a NULL pointer dereference in the caller. `` ? show_regs (arch/x86/kernel/dumpstack.c:478 (discriminator 1)) ? __die (arch/x86/kernel/dumpstack.c:423 arch/x86/kernel/dumpstack.c:434) ? page_fault_oops (arch/x86/mm/fault.c:544 (discriminator 2) arch/x86/mm/fault.c:705 (discriminator 2)) ? do_user_addr_fault (arch/x86/mm/fault.c:440 (discriminator 1) arch/x86/mm/fault.c:1232 (discriminator 1)) ? acpi_ut_update_object_reference (drivers/acpi/acpica/utdelete.c:642) ? exc_page_fault (arch/x86/mm/fault.c:1542) ? asm_exc_page_fault (./arch/x86/include/asm/identrty.h:623) ? amdgpu_atif_query_backlight_caps.constprop.0 (drivers/gpu/drm/amd/amdgpu/amdgpu_acpi.c:387 (discriminator 2)) amdgpu ? amdgpu_atif_query_backlight_caps.constprop.0 (drivers/gpu/drm/amd/amdgpu/amdgpu_acpi.c:386 (discriminator 1)) amdgpu `` It has been encountered on at least one system, so guard for it. (cherry picked from commit c9b7c809b89f24e9372a4e7f02d64c950b07fdee)
CVE-2024-50118	In the Linux kernel, the following vulnerability has been resolved: btrfs: reject ro->rw reconfiguration if there are hard ro requirements [BUG] Syzbot reports the following crash: BTRFS info (device loop0 state MCS): disabling free space tree BTRFS info (device loop0 state MCS): clearing compat-ro feature flag for FREE_SPACE_TREE (0x1) BTRFS info (device loop0 state MCS): clearing compat-ro feature flag for FREE_SPACE_TREE_VALID (0x2) Oops: general protection fault, probably for non-canonical address 0xdffffc0000000003: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000018-0x000000000000001f] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:backup_super_roots fs/btrfs/disk-io.c:1691 [inline] RIP: 0010:write_all_supers+0x97a/0x40f0 fs/btrfs/disk-io.c:4041 Call Trace: <TASK> btrfs_commit_transaction+0x1eae/0x3740 fs/btrfs/transaction.c:2530 btrfs_delete_free_space_tree+0x383/0x730 fs/btrfs/free-space-tree.c:1312 btrfs_start_pre_rw_mount+0xf28/0x1300 fs/btrfs/disk-io.c:3012 btrfs_remount_rw fs/btrfs/super.c:1309 [inline] btrfs_reconfigure+0xae6/0x2d40 fs/btrfs/super.c:1534 btrfs_reconfigure_for_mount fs/btrfs/super.c:2020 [inline] btrfs_get_tree_subvol fs/btrfs/super.c:2079 [inline] btrfs_get_tree +0x918/0x1920 fs/btrfs/super.c:2115 vfs_get_tree+0x90/0x2b0 fs/super.c:1800 do_new_mount +0x2be/0xb40 fs/namespace.c:3472 do_mount fs/namespace.c:3812 [inline] __do_sys_mount fs/namespace.c:4020 [inline] __se_sys_mount+0x2d6/0x3c0 fs/namespace.c:3997 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f [CAUSE] To support mounting different subvolume with different RO/RW flags for the new mount APIs, btrfs introduced two workaround to support this feature: - Skip mount option/feature checks if we are mounting a different subvolume - Reconfigure the fs to RW if the initial mount is RO Combining these two, we can have the following sequence: - Mount the fs ro,rescue=all,clear_cache,space_cache=v1 rescue=all will mark the fs as hard read-only, so no v2 cache clearing will happen. - Mount a subvolume rw of the same fs. We go into btrfs_get_tree_subvol(), but fc_mount() returns EBUSY because our new fc is RW, different from the original fs. Now we enter btrfs_reconfigure_for_mount(), which switches the RO flag first so that we can grab the existing fs_info. Then we reconfigure the fs to RW. - During reconfiguration, option/features check is skipped This means we will restart the v2 cache clearing, and convert back to v1 cache. This will trigger fs writes, and since the original fs has "rescue=all" option, it skips the csum tree read. And eventually causing NULL pointer dereference in super block writeback. [FIX] For reconfiguration caused by different subvolume RO/RW flags, ensure we always run btrfs_check_options() to ensure we have proper hard RO requirements met. In fact the function btrfs_check_options() doesn't really do many complex checks, but hard RO requirement and some feature dependency checks, thus there is no special reason not to do the check for mount reconfiguration.
CVE-2024-50120	In the Linux kernel, the following vulnerability has been resolved: smb: client: Handle kstrdup failures for passwords In smb3_reconfigure(), after duplicating ctx->password and ctx->password2 with kstrdup(), we need to check for allocation failures. If ses->password allocation fails, return -ENOMEM. If ses->password2 allocation fails, free ses->password, set it to NULL, and return -ENOMEM.

CVE ID	Description
CVE-2024-50121	In the Linux kernel, the following vulnerability has been resolved: nfsd: cancel nfsd_shrinker_work using sync mode in nfs4_state_shutdown_net In the normal case, when we excute `echo 0 > /proc/fs/nfsd/threads`, the function `nfs4_state_destroy_net` in `nfs4_state_shutdown_net` will release all resources related to the hashed `nfs4_client`. If the `nfsd_client_shrinker` is running concurrently, the `expire_client` function will first unhash this client and then destroy it. This can lead to the following warning. Additionally, numerous use-after-free errors may occur as well. <pre>nfsd_client_shrinker echo 0 > /proc/fs/nfsd/threads expire_client nfsd_shutdown_net unhash_client ... nfs4_state_shutdown_net /* won't wait shrinker exit */ /* cancel_work(&nn->nfsd_shrinker_work) * nfsd_file for this /* won't destroy unhashed client1 */ /* client1 still alive nfs4_state_destroy_net */ nfsd_file_cache_shutdown /* trigger warning */ kmem_cache_destroy(nfsd_file_slab) kmem_cache_destroy(nfsd_file_mark_slab) /* release nfsd_file and mark */ __destroy_client ===== BUG nfsd_file (Not tainted): Objects remaining in nfsd_file on __kmem_cache_shutdown() ----- CPU: 4 UID: 0 PID: 764 Comm: sh Not tainted 6.12.0-rc3+ #1 dump_stack_lvl+0x53/0x70 slab_err+0xb0/0xf0 __kmem_cache_shutdown +0x15c/0x310 kmem_cache_destroy+0x66/0x160 nfsd_file_cache_shutdown+0xac/0x210 [nfsd] nfsd_destroy_serv+0x251/0x2a0 [nfsd] nfsd_svc+0x125/0x1e0 [nfsd] write_threads +0x16a/0x2a0 [nfsd] nfscctl_transaction_write+0x74/0xa0 [nfsd] vfs_write+0x1a5/0x6d0 ksys_write +0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e ===== BUG nfsd_file_mark (Tainted: G B W): Objects remaining nfsd_file_mark on __kmem_cache_shutdown() ----- dump_stack_lvl+0x53/0x70 slab_err +0xb0/0xf0 __kmem_cache_shutdown+0x15c/0x310 kmem_cache_destroy+0x66/0x160 nfsd_file_cache_shutdown+0xc8/0x210 [nfsd] nfsd_destroy_serv+0x251/0x2a0 [nfsd] nfsd_svc+0x125/0x1e0 [nfsd] write_threads+0x16a/0x2a0 [nfsd] nfscctl_transaction_write +0x74/0xa0 [nfsd] vfs_write+0x1a5/0x6d0 ksys_write+0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e To resolve this issue, cancel `nfsd_shrinker_work` using synchronous mode in nfs4_state_shutdown_net.</pre>
CVE-2024-50124	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: ISO: Fix UAF on iso_sock_timeout conn->sk maybe have been unlinked/freed while waiting for iso_conn_lock so this checks if the conn->sk is still valid by checking if it part of iso_sk_list.
CVE-2024-50125	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: SCO: Fix UAF on sco_sock_timeout conn->sk maybe have been unlinked/freed while waiting for sco_conn_lock so this checks if the conn->sk is still valid by checking if it part of sco_sk_list.

CVE ID	Description
CVE-2024-50126	In the Linux kernel, the following vulnerability has been resolved: net: sched: use RCU read-side critical section in taprio_dump() Fix possible use-after-free in 'taprio_dump()' by adding RCU read-side critical section there. Never seen on x86 but found on a KASAN-enabled arm64 system when investigating https://syzkaller.appspot.com/bug?extid=b65e0af58423fc8a73aa: [T15862] BUG: KASAN: slab-use-after-free in taprio_dump+0xa0c/0xbb0 [T15862] Read of size 4 at addr ffff0000d4bb88f8 by task repro/15862 [T15862] [T15862] CPU: 0 UID: 0 PID: 15862 Comm: repro Not tainted 6.11.0-rc1-00293-gdefaf1a2113a-dirty #2 [T15862] Hardware name: QEMU QEMU Virtual Machine, BIOS edk2-20240524-5.fc40 05/24/2024 [T15862] Call trace: [T15862] dump_backtrace+0x20c/0x220 [T15862] show_stack+0x2c/0x40 [T15862] dump_stack_lvl+0xf8/0x174 [T15862] print_report+0x170/0x4d8 [T15862] kasan_report+0xb8/0x1d4 [T15862] __asan_report_load4_noabort+0x20/0x2c [T15862] taprio_dump+0xa0c/0xbb0 [T15862] tc_fill_qdisc+0x540/0x1020 [T15862] qdisc_notify.isra.0+0x330/0x3a0 [T15862] tc_modify_qdisc+0x7b8/0x1838 [T15862] rtnetlink_rcv_msg+0x3c8/0xc20 [T15862] netlink_rcv_skb+0x1f8/0x3d4 [T15862] rtnetlink_rcv+0x28/0x40 [T15862] netlink_unicast+0x51c/0x790 [T15862] netlink_sendmsg+0x79c/0xc20 [T15862] __sock_sendmsg+0xe0/0x1a0 [T15862] __sys_sendmsg+0x6c0/0x840 [T15862] __sys_sendmsg+0x1ac/0x1f0 [T15862] __sys_sendmsg+0x110/0x1d0 [T15862] __arm64_sys_sendmsg+0x74/0xb0 [T15862] invoke_syscall+0x88/0x2e0 [T15862] el0_svc_common.constprop.0+0xe4/0x2a0 [T15862] do_el0_svc+0x44/0x60 [T15862] el0_svc+0x50/0x184 [T15862] el0t_64_sync_handler+0x120/0x12c [T15862] el0t_64_sync+0x190/0x194 [T15862] [T15862] Allocated by task 15857: [T15862] kasan_save_stack+0x3c/0x70 [T15862] kasan_save_track+0x20/0x3c [T15862] kasan_save_alloc_info+0x40/0x60 [T15862] __kasan_kmalloc+0xd4/0xe0 [T15862] __kmalloc_cache_noprof+0x194/0x334 [T15862] taprio_change+0x45c/0x2fe0 [T15862] tc_modify_qdisc+0x6a8/0x1838 [T15862] rtnetlink_rcv_msg+0x3c8/0xc20 [T15862] netlink_rcv_skb+0x1f8/0x3d4 [T15862] rtnetlink_rcv+0x28/0x40 [T15862] netlink_unicast+0x51c/0x790 [T15862] netlink_sendmsg+0x79c/0xc20 [T15862] __sock_sendmsg+0xe0/0x1a0 [T15862] __sys_sendmsg+0x6c0/0x840 [T15862] __sys_sendmsg+0x1ac/0x1f0 [T15862] __sys_sendmsg+0x110/0x1d0 [T15862] __arm64_sys_sendmsg+0x74/0xb0 [T15862] invoke_syscall+0x88/0x2e0 [T15862] el0_svc_common.constprop.0+0xe4/0x2a0 [T15862] do_el0_svc+0x44/0x60 [T15862] el0_svc+0x50/0x184 [T15862] el0t_64_sync_handler+0x120/0x12c [T15862] el0t_64_sync+0x190/0x194 [T15862] [T15862] Freed by task 6192: [T15862] kasan_save_stack+0x3c/0x70 [T15862] kasan_save_track+0x20/0x3c [T15862] kasan_save_free_info+0x4c/0x80 [T15862] poison_slab_object+0x110/0x160 [T15862] __kasan_slab_free+0x3c/0x74 [T15862] kfree+0x134/0x3c0 [T15862] taprio_free_sched_cb+0x18c/0x220 [T15862] rcu_core+0x920/0x1b7c [T15862] rcu_core_si+0x10/0x1c [T15862] handle_softirqs+0x2e8/0xd64 [T15862] __do_softirq+0x14/0x20
CVE-2024-50127	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix use-after-free in taprio_change() In 'taprio_change()', 'admin' pointer may become dangling due to sched switch / removal caused by 'advance_sched()', and critical section protected by 'q->current_entry_lock' is too small to prevent from such a scenario (which causes use-after-free detected by KASAN). Fix this by prefer 'rcu_replace_pointer()' over 'rcu_assign_pointer()' to update 'admin' immediately before an attempt to schedule freeing.

CVE ID	Description
CVE-2024-50128	In the Linux kernel, the following vulnerability has been resolved: net: wwan: fix global oob in wwan_rtnl_policy The variable wwan_rtnl_link_ops assign a *bigger* maxtype which leads to a global out-of-bounds read when parsing the netlink attributes. Exactly same bug cause as the oob fixed in commit b33fb5b801c6 ("net: qualcomm: rmnet: fix global oob in rmnet_policy"). ===== BUG: KASAN: global-out-of-bounds in validate_nla lib/nlatr.c:388 [inline] BUG: KASAN: global-out-of-bounds in __nla_validate_parse+0x19d7/0x29a0 lib/nlatr.c:603 Read of size 1 at addr ffffffff8b09cb60 by task syz.1.66276/323862 CPU: 0 PID: 323862 Comm: syz.1.66276 Not tainted 6.1.70 #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x177/0x231 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:284 [inline] print_report+0x14f/0x750 mm/kasan/report.c:395 kasan_report+0x139/0x170 mm/kasan/report.c:495 validate_nla lib/nlatr.c:388 [inline] __nla_validate_parse+0x19d7/0x29a0 lib/nlatr.c:603 __nla_parse+0x3c/0x50 lib/nlatr.c:700 nla_parse_nested_deprecated include/net/netlink.h:1269 [inline] __rtnl_newlink net/core/rtnetlink.c:3514 [inline] rtnl_newlink+0x7bc/0x1fd0 net/core/rtnetlink.c:3623 rtnetlink_rcv_msg+0x794/0xef0 net/core/rtnetlink.c:6122 netlink_rcv_skb+0x1de/0x420 net/netlink/af_netlink.c:2508 netlink_unicast_kernel net/netlink/af_netlink.c:1326 [inline] netlink_unicast+0x74b/0x8c0 net/netlink/af_netlink.c:1352 netlink_sendmsg+0x882/0xb90 net/netlink/af_netlink.c:1874 sock_sendmsg_nosec net/socket.c:716 [inline] __sock_sendmsg net/socket.c:728 [inline] ____sys_sendmsg+0x5cc/0x8f0 net/socket.c:2499 __sys_sendmsg+0x21c/0x290 net/socket.c:2553 __sys_sendmsg net/socket.c:2582 [inline] __do_sys_sendmsg net/socket.c:2591 [inline] __se_sys_sendmsg+0x19e/0x270 net/socket.c:2589 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x45/0x90 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x63/0xcd RIP: 0033:0xf67b19a24ad RSP: 002b:00007f67b17febb8 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffffdada RBX: 00007f67b1b45f80 RCX: 00007f67b19a24ad RDX: 0000000000000000 RSI: 0000000020005e40 RDI: 0000000000000004 RBP: 00007f67b1a1e01d R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007ffd2513764f R14: 00007ffd251376e0 R15: 00007f67b17fed40 </TASK> The buggy address belongs to the variable: wwan_rtnl_policy+0x20/0x40 The buggy address belongs to the physical page: page:ffffea00002c2700 refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0xb09c flags: 0xffff0000001000(reserved node=0)zone=1 lastcpupid=0x7ff raw: 00ffff0000001000 ffffea00002c2708 ffffea00002c2708 0000000000000000 raw: 0000000000000000 0000000000000000 00000001ffffff 0000000000000000 page dumped because: kasan: bad access detected page_owner info is not present (never set?) Memory state around the buggy address: ffffffff8b09ca00: 05 f9 f9 f9 05 f9 f9 00 01 f9 f9 00 01 f9 f9 ffffffff8b09ca80: 00 00 00 05 f9 f9 f9 00 03 f9 f9 f9 f9 >ffffffff8b09cb00: 00 00 00 00 05 f9 f9 00 00 00 00 00 00 00 00 00 00 ^ ffffffff8b09cb80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ===== According to the comment of `nla_parse_nested_deprecated`, use correct size `IFLA_WWAN_MAX` here to fix this issue.
CVE-2024-50130	In the Linux kernel, the following vulnerability has been resolved: netfilter: bpf: must hold reference on net namespace BUG: KASAN: slab-use-after-free in __nf_unregister_net_hook+0x640/0x6b0 Read of size 8 at addr ffff8880106fe400 by task repro/72= bpf_nf_link_release+0xda/0x1e0 bpf_link_free+0x139/0x2d0 bpf_link_release+0x68/0x80 __fput+0x414/0xb60 Eric says: It seems that bpf was able to defer the __nf_unregister_net_hook() after exit()/close() time. Perhaps a nets reference is missing, because the nets has been dismantled/freed already. bpf_nf_link_attach() does : link->net = net; But I do not see a reference being taken on net. Add such a reference and release it after hook unreg. Note that I was unable to get syzbot reproducer to work, so I do not know if this resolves this splat.
CVE-2024-50131	In the Linux kernel, the following vulnerability has been resolved: tracing: Consider the NULL character when validating the event length strlen() returns a string length excluding the null byte. If the string length equals to the maximum buffer length, the buffer will have no space for the NULL terminating character. This commit checks this condition and returns failure for it.
CVE-2024-50133	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Don't crash in stack_top() for tasks without vDSO Not all tasks have a vDSO mapped, for example kthreads never do. If such a task ever ends up calling stack_top(), it will dereference the NULL vdso pointer and crash. This can for example happen when using kunit: [<9000000000203874>] stack_top+0x58/0xa8 [<90000000002956cc>] arch_pick_mmap_layout+0x164/0x220 [<90000000003c284c>] kunit_vm_mmap_init+0x108/0x12c [<90000000003c1fbc>] __kunit_add_resource+0x38/0x8c [<90000000003c2704>] kunit_vm_mmap+0x88/0xc8 [<9000000000410b14>] usercopy_test_init+0xbc/0x25c [<90000000003c1db4>] kunit_try_run_case+0x5c/0x184 [<90000000003c3d54>] kunit_generic_run_threadfn_adapter+0x24/0x48 [<9000000000022e48c>] kthread+0xc8/0xd4 [<9000000000200ce8>] ret_from_kernel_thread+0xc/0xa4

CVE ID	Description
CVE-2024-50134	In the Linux kernel, the following vulnerability has been resolved: drm/vboxvideo: Replace fake VLA at end of vbva_mouse_pointer_shape with real VLA Replace the fake VLA at end of the vbva_mouse_pointer_shape shape with a real VLA to fix a "memcpy: detected field-spanning write error" warning: [13.319813] memcpy: detected field-spanning write (size 16896) of single field "p->data" at drivers/gpu/drm/vboxvideo/hgsmi_base.c:154 (size 4) [13.319841] WARNING: CPU: 0 PID: 1105 at drivers/gpu/drm/vboxvideo/hgsmi_base.c:154 hgsmi_update_pointer_shape+0x192/0x1c0 [vboxvideo] [13.320038] Call Trace: [13.320173] hgsmi_update_pointer_shape [vboxvideo] [13.320184] vbox_cursor_atomic_update [vboxvideo] Note as mentioned in the added comment it seems the original length calculation for the allocated and send hgsmi buffer is 4 bytes too large. Changing this is not the goal of this patch, so this behavior is kept.
CVE-2024-50135	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: fix race condition between reset and nvme_dev_disable() nvme_dev_disable() modifies the dev->online_queues field, therefore nvme_pci_update_nr_queues() should avoid racing against it, otherwise we could end up passing invalid values to blk_mq_update_nr_hw_queues(). WARNING: CPU: 39 PID: 61303 at drivers/pci/msi/api.c:347 pci_irq_get_affinity+0x187/0x210 Workqueue: nvme-reset-wq nvme_reset_work [nvme] RIP: 0010:pci_irq_get_affinity+0x187/0x210 Call Trace: <TASK> ? blk_mq_pci_map_queues+0x87/0x3c0 ? pci_irq_get_affinity+0x187/0x210 blk_mq_pci_map_queues+0x87/0x3c0 nvme_pci_map_queues+0x189/0x460 [nvme] blk_mq_update_nr_hw_queues+0x2a/0x40 nvme_reset_work+0x1be/0x2a0 [nvme] Fix the bug by locking the shutdown_lock mutex before using dev->online_queues. Give up if nvme_dev_disable() is running or if it has been executed already.
CVE-2024-50136	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Unregister notifier on eswitch init failure It otherwise remains registered and a subsequent attempt at eswitch enabling might trigger warnings of the sort: [682.589148] -----[cut here]----- [682.590204] notifier callback eswitch_vport_event [mlx5_core] already registered [682.590256] WARNING: CPU: 13 PID: 2660 at kernel/notifier.c:31 notifier_chain_register+0x3e/0x90 [...snipped] [682.610052] Call Trace: [682.610369] <TASK> [682.610663] ? __warn+0x7c/0x110 [682.611050] ? notifier_chain_register+0x3e/0x90 [682.611556] ? report_bug+0x148/0x170 [682.611977] ? handle_bug+0x36/0x70 [682.612384] ? exc_invalid_op+0x13/0x60 [682.612817] ? asm_exc_invalid_op+0x16/0x20 [682.613284] ? notifier_chain_register+0x3e/0x90 [682.613789] atomic_notifier_chain_register+0x25/0x40 [682.614322] mlx5_eswitch_enable_locked+0x1d4/0x3b0 [mlx5_core] [682.614965] mlx5_eswitch_enable+0xc9/0x100 [mlx5_core] [682.615551] mlx5_device_enable_sriov+0x25/0x340 [mlx5_core] [682.616170] mlx5_core_sriov_configure+0x50/0x170 [mlx5_core] [682.616789] sriov_numvfs_store+0xb0/0x1b0 [682.617248] kernfs_fop_write_iter+0x117/0x1a0 [682.617734] vfs_write+0x231/0x3f0 [682.618138] ksys_write+0x63/0xe0 [682.618536] do_syscall_64+0x4c/0x100 [682.618958] entry_SYSCALL_64_after_hwframe+0x4b/0x53
CVE-2024-50137	In the Linux kernel, the following vulnerability has been resolved: reset: starfive: jh71x0: Fix accessing the empty member on JH7110 SoC data->asserted will be NULL on JH7110 SoC since commit 82327b127d41 ("reset: starfive: Add StarFive JH7110 reset driver") was added. Add the judgment condition to avoid errors when calling reset_control_status on JH7110 SoC.
CVE-2024-50138	In the Linux kernel, the following vulnerability has been resolved: bpf: Use raw_spinlock_t in ringbuf The function __bpf_ringbuf_reserve is invoked from a tracepoint, which disables preemption. Using spinlock_t in this context can lead to a "sleep in atomic" warning in the RT variant. This issue is illustrated in the example below: BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 556208, name: test_progs preempt_count: 1, expected: 0 RCU nest depth: 1, expected: 1 INFO: lockdep is turned off. Preemption disabled at: [<ffffd33a5c88ea44>] migrate_enable+0xc0/0x39c CPU: 7 PID: 556208 Comm: test_progs Tainted: G Hardware name: Qualcomm SA8775P Ride (DT) Call trace: dump_backtrace+0xac/0x130 show_stack+0x1c/0x30 dump_stack_lvl+0xac/0xe8 dump_stack+0x18/0x30 __might_resched+0x3bc/0x4fc rt_spin_lock+0x8c/0x1a4 __bpf_ringbuf_reserve+0xc4/0x254 bpf_ringbuf_reserve_dynptr+0x5c/0xdc bpf_prog_ac3d15160d62622a_test_read_write+0x104/0x238 trace_call_bpf+0x238/0x774 perf_call_bpf_enter.isra.0+0x104/0x194 perf_syscall_enter+0x2f8/0x510 trace_sys_enter+0x39c/0x564 syscall_trace_enter+0x220/0x3c0 do_el0_svc+0x138/0x1dc el0_svc+0x54/0x130 el0t_64_sync_handler+0x134/0x150 el0t_64_sync+0x17c/0x180 Switch the spinlock to raw_spinlock_t to avoid this error.

CVE ID	Description
CVE-2024-50139	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Fix shift-out-of-bounds bug Fix a shift-out-of-bounds bug reported by UBSAN when running VM with MTE enabled host kernel. UBSAN: shift-out-of-bounds in arch/arm64/kvm/sys_regs.c:1988:14 shift exponent 33 is too large for 32-bit type 'int' CPU: 26 UID: 0 PID: 7629 Comm: qemu-kvm Not tainted 6.12.0-rc2 #34 Hardware name: IEI NF5280R7/Mitchell MB, BIOS 00.00. 2024-10-12 09:28:54 10/14/2024 Call trace: dump_backtrace+0xa0/0x128 show_stack+0x20/0x38 dump_stack_lvl+0x74/0x90 dump_stack+0x18/0x28 __ubsan_handle_shift_out_of_bounds+0xf8/0x1e0 reset_clidr+0x10c/0x1c8 kvm_reset_sys_regs+0x50/0x1c8 kvm_reset_vcpu+0xec/0x2b0 __kvm_vcpu_set_target+0x84/0x158 kvm_vcpu_set_target+0x138/0x168 kvm_arch_vcpu_ioctl_vcpu_init+0x40/0x2b0 kvm_arch_vcpu_ioctl+0x28c/0x4b8 kvm_vcpu_ioctl+0x4bc/0x7a8 __arm64_sys_ioctl+0xb4/0x100 invoke_syscall+0x70/0x100 el0_svc_common.constprop.0+0x48/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x3c/0x158 el0t_64_sync_handler+0x120/0x130 el0t_64_sync+0x194/0x198
CVE-2024-50140	In the Linux kernel, the following vulnerability has been resolved: sched/core: Disable page allocation in task_tick_mm_cid() With KASAN and PREEMPT_RT enabled, calling task_work_add() in task_tick_mm_cid() may cause the following splat. [63.696416] BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 [63.696416] in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 610, name: modprobe [63.696416] preempt_count: 10001, expected: 0 [63.696416] RCU nest depth: 1, expected: 1 This problem is caused by the following call trace. sched_tick() [acquire rq->__lock] -> task_tick_mm_cid() -> task_work_add() -> __kasan_record_aux_stack() -> kasan_save_stack() -> stack_depot_save_flags() -> alloc_pages_mpol_noprof() -> __alloc_pages_noprof() -> get_page_from_freelist() -> rmqueue() -> rmqueue_pcplist() -> __rmqueue_pcplist() -> rmqueue_bulk() -> rt_spin_lock() The rq lock is a raw_spinlock_t. We can't sleep while holding it. IOW, we can't call alloc_pages() in stack_depot_save_flags(). The task_tick_mm_cid() function with its task_work_add() call was introduced by commit 223baf9d17f2 ("sched: Fix performance regression introduced by mm_cid") in v6.4 kernel. Fortunately, there is a kasan_record_aux_stack_noalloc() variant that calls stack_depot_save_flags() while not allowing it to allocate new pages. To allow task_tick_mm_cid() to use task_work without page allocation, a new TWAF_NO_ALLOC flag is added to enable calling kasan_record_aux_stack_noalloc() instead of kasan_record_aux_stack() if set. The task_tick_mm_cid() function is modified to add this new flag. The possible downside is the missing stack trace in a KASAN report due to new page allocation required when task_work_add_noalloc() is called which should be rare.
CVE-2024-50141	In the Linux kernel, the following vulnerability has been resolved: ACPI: PRM: Find EFI_MEMORY_RUNTIME block for PRM handler and context PRMT needs to find the correct type of block to translate the PA-VA mapping for EFI runtime services. The issue arises because the PRMT is finding a block of type EFI_CONVENTIONAL_MEMORY, which is not appropriate for runtime services as described in Section 2.2.2 (Runtime Services) of the UEFI Specification [1]. Since the PRM handler is a type of runtime service, this causes an exception when the PRM handler is called. [Firmware Bug]: Unable to handle paging request in EFI runtime service WARNING: CPU: 22 PID: 4330 at drivers/firmware/efi/runtime-wrappers.c:341 __efi_queue_work+0x11c/0x170 Call trace: Let PRMT find a block with EFI_MEMORY_RUNTIME for PRM handler and PRM context. If no suitable block is found, a warning message will be printed, but the procedure continues to manage the next PRM handler. However, if the PRM handler is actually called without proper allocation, it would result in a failure during error handling. By using the correct memory types for runtime services, ensure that the PRM handler and the context are properly mapped in the virtual address space during runtime, preventing the paging request error. The issue is really that only memory that has been remapped for runtime by the firmware can be used by the PRM handler, and so the region needs to have the EFI_MEMORY_RUNTIME attribute. [rjw: Subject and changelog edits]
CVE-2024-50142	In the Linux kernel, the following vulnerability has been resolved: xfrm: validate new SA's prefixlen using SA family when sel.family is unset This expands the validation introduced in commit 07bf7908950a ("xfrm: Validate address prefix lengths in the xfrm selector.") syzbot created an SA with usersa.sel.family = AF_UNSPEC usersa.sel.prefixlen_s = 128 usersa.family = AF_INET Because of the AF_UNSPEC selector, verify_newsa_info doesn't put limits on prefixlen_{s,d}. But then copy_from_user_state sets x->sel.family to usersa.family (AF_INET). Do the same conversion in verify_newsa_info before validating prefixlen_{s,d}, since that's how prefixlen is going to be used later on.
CVE-2024-50143	In the Linux kernel, the following vulnerability has been resolved: udf: fix uninit-value use in udf_get_fileshortad Check for overflow when computing alen in udf_current_aext to mitigate later uninit-value use in udf_get_fileshortad KMSAN bug[1]. After applying the patch reproducer did not trigger any issue[2]. [1] https://syzkaller.appspot.com/bug?extid=8901c4560b7ab5c2f9df [2] https://syzkaller.appspot.com/x/log.txt?x=10242227980000

CVE ID	Description
CVE-2024-50145	In the Linux kernel, the following vulnerability has been resolved: octeon_ep: Add SKB allocation failures handling in __octep_oq_process_rx() build_skb() returns NULL in case of a memory allocation failure so handle it inside __octep_oq_process_rx() to avoid NULL pointer dereference. __octep_oq_process_rx() is called during NAPI polling by the driver. If skb allocation fails, keep on pulling packets out of the Rx DMA queue: we shouldn't break the polling immediately and thus falsely indicate to the octep_napi_poll() that the Rx pressure is going down. As there is no associated skb in this case, don't process the packets and don't push them up the network stack - they are skipped. Helper function is implemented to unmap/flush all the fragment buffers used by the dropped packet. 'alloc_failures' counter is incremented to mark the skb allocation error in driver statistics. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-50146	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Don't call cleanup on profile rollback failure When profile rollback fails in mlx5e_netdev_change_profile, the netdev profile var is left set to NULL. Avoid a crash when unloading the driver by not calling profile->cleanup in such a case. This was encountered while testing, with the original trigger that the wq rescuer thread creation got interrupted (presumably due to Ctrl+C-ing modprobe), which gets converted to ENOMEM (-12) by mlx5e_priv_init, the profile rollback also fails for the same reason (signal still active) so the profile is left as NULL, leading to a crash later in _mlx5e_remove. [732.473932] mlx5_core 0000:08:00.1: E-Switch: Unload vfs: mode(OFFLOADS), nvfs(2), necvfs(0), active vports(2) [734.525513] workqueue: Failed to create a rescuer kthread for wq "mlx5e": -EINTR [734.557372] mlx5_core 0000:08:00.1: mlx5e_netdev_init_profile:6235:(pid 6086): mlx5e_priv_init failed, err=-12 [734.559187] mlx5_core 0000:08:00.1 eth3: mlx5e_netdev_change_profile: new profile init failed, -12 [734.560153] workqueue: Failed to create a rescuer kthread for wq "mlx5e": -EINTR [734.589378] mlx5_core 0000:08:00.1: mlx5e_netdev_init_profile:6235:(pid 6086): mlx5e_priv_init failed, err=-12 [734.591136] mlx5_core 0000:08:00.1 eth3: mlx5e_netdev_change_profile: failed to rollback to orig profile, -12 [745.537492] BUG: kernel NULL pointer dereference, address: 0000000000000008 [745.538222] #PF: supervisor read access in kernel mode <snipped> [745.551290] Call Trace: [745.551590] <TASK> [745.551866] ? __die+0x20/0x60 [745.552218] ? page_fault_oops+0x150/0x400 [745.555307] ? exc_page_fault+0x79/0x240 [745.555729] ? asm_exc_page_fault+0x22/0x30 [745.556166] ? mlx5e_remove+0x6b/0xb0 [mlx5_core] [745.556698] auxiliary_bus_remove+0x18/0x30 [745.557134] device_release_driver_internal+0x1df/0x240 [745.557654] bus_remove_device+0xd7/0x140 [745.558075] device_del+0x15b/0x3c0 [745.558456] mlx5_rescan_drivers_locked.part.0+0xb1/0x2f0 [mlx5_core] [745.559112] mlx5_unregister_device+0x34/0x50 [mlx5_core] [745.559686] mlx5_uninit_one+0x46/0xf0 [mlx5_core] [745.560203] remove_one+0x4e/0xd0 [mlx5_core] [745.560694] pci_device_remove+0x39/0xa0 [745.561112] device_release_driver_internal+0x1df/0x240 [745.561631] driver_detach+0x47/0x90 [745.562022] bus_remove_driver+0x84/0x100 [745.562444] pci_unregister_driver+0x3b/0x90 [745.562890] mlx5_cleanup+0xc/0x1b [mlx5_core] [745.563415] __x64_sys_delete_module+0x14d/0x2f0 [745.563886] ? kmem_cache_free+0x1b0/0x460 [745.564313] ? lockdep_hardirqs_on_prepare+0xe2/0x190 [745.564825] do_syscall_64+0x6d/0x140 [745.565223] entry_SYSCALL_64_after_hwframe+0x4b/0x53 [745.565725] RIP: 0033:0x7f1579b1288b
CVE-2024-50147	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix command bitmask initialization Command bitmask have a dedicated bit for MANAGE_PAGES command, this bit isn't Initialize during command bitmask Initialization, only during MANAGE_PAGES. In addition, mlx5_cmd_trigger_completions() is trying to trigger completion for MANAGE_PAGES command as well. Hence, in case health error occurred before any MANAGE_PAGES command have been invoke (for example, during mlx5_enable_hca()), mlx5_cmd_trigger_completions() will try to trigger completion for MANAGE_PAGES command, which will result in null-ptr-deref error.[1] Fix it by Initialize command bitmask correctly. While at it, re-write the code for better understanding. [1] BUG: KASAN: null-ptr-deref in mlx5_cmd_trigger_completions+0x1db/0x600 [mlx5_core] Write of size 4 at addr 0000000000000214 by task kworker/u96:2/12078 CPU: 10 PID: 12078 Comm: kworker/u96:2 Not tainted 6.9.0-rc2_for_upstream_debug_2024_04_07_19_01 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 Workqueue: mlx5_health0000:08:00.0 mlx5_fw_fatal_reporter_err_work [mlx5_core] Call Trace: <TASK> dump_stack_lvl+0x7e/0xc0 kasan_report+0xb9/0xf0 kasan_check_range+0xec/0x190 mlx5_cmd_trigger_completions+0x1db/0x600 [mlx5_core] mlx5_cmd_flush+0x94/0x240 [mlx5_core] enter_error_state+0x6c/0xd0 [mlx5_core] mlx5_fw_fatal_reporter_err_work+0xf3/0x480 [mlx5_core] process_one_work+0x787/0x1490 ? lockdep_hardirqs_on_prepare+0x400/0x400 ? pwq_dec_nr_in_flight+0xda0/0xda0 ? assign_work+0x168/0x240 worker_thread+0x586/0xd30 ? rescuer_thread+0xae0/0xae0 kthread+0x2df/0x3b0 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x2d/0x70 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork_asm+0x11/0x20 </TASK>

CVE ID	Description
CVE-2024-50148	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: bnep: fix wild-memory-access in proto_unregister There's issue as follows: KASAN: maybe wild-memory-access in range [0xdead...108-0xdead...10f] CPU: 3 UID: 0 PID: 2805 Comm: rmmmod Tainted: G W RIP: 0010:proto_unregister+0xee/0x400 Call Trace: <TASK> __do_sys_delete_module+0x318/0x580 do_syscall_64+0xc1/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f As bnep_init() ignore bnep_sock_init()'s return value, and bnep_sock_init() will cleanup all resource. Then when remove bnep module will call bnep_sock_cleanup() to cleanup sock's resource. To solve above issue just return bnep_sock_init()'s return value in bnep_exit().
CVE-2024-50150	In the Linux kernel, the following vulnerability has been resolved: usb: typec: altmode should keep reference to parent The altmode device release refers to its parent device, but without keeping a reference to it. When registering the altmode, get a reference to the parent and put it in the release function. Before this fix, when using CONFIG_DEBUG_KOBJECT_RELEASE, we see issues like this: [43.572860] kobject: 'port0.0' (ffff8880057ba008): kobject_release, parent 0000000000000000 (delayed 3000) [43.573532] kobject: 'port0.1' (ffff8880057bd008): kobject_release, parent 0000000000000000 (delayed 1000) [43.574407] kobject: 'port0' (ffff8880057b9008): kobject_release, parent 0000000000000000 (delayed 3000) [43.575059] kobject: 'port1.0' (ffff8880057ca008): kobject_release, parent 0000000000000000 (delayed 4000) [43.575908] kobject: 'port1.1' (ffff8880057c9008): kobject_release, parent 0000000000000000 (delayed 4000) [43.576908] kobject: 'typec' (ffff8880062dbc00): kobject_release, parent 0000000000000000 (delayed 4000) [43.577769] kobject: 'port1' (ffff8880057bf008): kobject_release, parent 0000000000000000 (delayed 3000) [46.612867] ===== [46.613402] BUG: KASAN: slab-use-after-free in typec_altmode_release+0x38/0x129 [46.614003] Read of size 8 at addr ffff8880057b9118 by task kworker/2:1/48 [46.614538] [46.614668] CPU: 2 UID: 0 PID: 48 Comm: kworker/2:1 Not tainted 6.12.0-rc1-00138-gedbae730ad31 #535 [46.615391] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1 04/01/2014 [46.616042] Workqueue: events kobject_delayed_cleanup [46.616446] Call Trace: [46.616648] <TASK> [46.616820] dump_stack_lvl+0x5b/0x7c [46.617112] ? typec_altmode_release+0x38/0x129 [46.617470] print_report+0x14c/0x49e [46.617769] ? rcu_read_unlock_sched+0x56/0x69 [46.618117] ? __virt_addr_valid+0x19a/0x1ab [46.618456] ? kmem_cache_debug_flags+0xc/0x1d [46.618807] ? typec_altmode_release+0x38/0x129 [46.619161] kasan_report+0x8d/0xb4 [46.619447] ? typec_altmode_release+0x38/0x129 [46.619809] ? process_scheduled_works+0x3cb/0x85f [46.620185] typec_altmode_release+0x38/0x129 [46.620537] ? process_scheduled_works+0x3cb/0x85f [46.620907] device_release+0xaf/0xf2 [46.621206] kobject_delayed_cleanup+0x13b/0x17a [46.621584] process_scheduled_works+0x4f6/0x85f [46.621955] ? __pfx_process_scheduled_works+0x10/0x10 [46.622353] ? hlock_class+0x31/0x9a [46.622647] ? lock_acquired+0x361/0x3c3 [46.622956] ? move_linked_works+0x46/0x7d [46.623277] worker_thread+0x1ce/0x291 [46.623582] ? __kthread_parkme+0xc8/0xdf [46.623900] ? __pfx_worker_thread+0x10/0x10 [46.624236] kthread+0x17e/0x190 [46.624501] ? kthread+0xfb/0x190 [46.624756] ? __pfx_kthread+0x10/0x10 [46.625015] ret_from_fork+0x20/0x40 [46.625268] ? __pfx_kthread+0x10/0x10 [46.625532] ret_from_fork_asm+0x1a/0x30 [46.625805] </TASK> [46.625953] [46.626056] Allocated by task 678: [46.626287] kasan_save_stack+0x24/0x44 [46.626555] kasan_save_track+0x14/0x2d [46.626811] __kasan_kmalloc+0x3f/0x4d [46.627049] __kmalloc_noprof+0x1bf/0x1f0 [46.627362] typec_register_port+0x23/0x491 [46.627698] cros_typec_probe+0x634/0xbb6 [46.628026] platform_probe+0x47/0x8c [46.628311] really_probe+0x20a/0x47d [46.628605] device_driver_attach+0x39/0x72 [46.628940] bind_store+0x87/0xd7 [46.629213] kernfs_fop_write_iter+0x1aa/0x218 [46.629574] vfs_write+0x1d6/0x29b [46.629856] ksys_write+0xcd/0x13b [46.630128] do_syscall_64+0xd4/0x139 [46.630420] entry_SYSCALL_64_after_hwframe+0x76/0x7e [46.630820] [46.630946] Freed by task 48: [46.631182] kasan_save_stack+0x24/0x44 [46.631493] kasan_save_track+0x14/0x2d [46.631799] kasan_save_free_info+0x3f/0x4d [46.632144] __kasan_slab_free+0x37/0x45 [46.632474] --- truncated---

CVE ID	Description
CVE-2024-50151	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix OOBs when building SMB2_IOCTL request When using encryption, either enforced by the server or when using 'seal' mount option, the client will squash all compound request buffers down for encryption into a single iov in smb2_set_next_command(). SMB2_ioctl_init() allocates a small buffer (448 bytes) to hold the SMB2_IOCTL request in the first iov, and if the user passes an input buffer that is greater than 328 bytes, smb2_set_next_command() will end up writing off the end of @rqst->iov[0].iov_base as shown below: mount.cifs //srv/share /mnt -o ...,seal ln -s \$(perl -e "print('a')for 1..1024") /mnt/link BUG: KASAN: slab-out-of-bounds in smb2_set_next_command.cold+0x1d6/0x24c [cifs] Write of size 4116 at addr ffff8881148fcab8 by task ln/859 CPU: 1 UID: 0 PID: 859 Comm: ln Not tainted 6.12.0-rc3 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-2.fc40 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x5d/0x80 ? smb2_set_next_command.cold+0x1d6/0x24c [cifs] print_report+0x156/0x4d9 ? smb2_set_next_command.cold+0x1d6/0x24c [cifs] ? __virt_addr_valid+0x145/0x310 ? __phys_addr+0x46/0x90 ? smb2_set_next_command.cold+0x1d6/0x24c [cifs] kasan_report+0xda/0x110 ? smb2_set_next_command.cold+0x1d6/0x24c [cifs] kasan_check_range+0x10f/0x1f0 __asan_memcpy+0x3c/0x60 smb2_set_next_command.cold+0x1d6/0x24c [cifs] smb2_compound_op+0x238c/0x3840 [cifs] ? kasan_save_track+0x14/0x30 ? kasan_save_free_info+0x3b/0x70 ? vfs_symlink+0x1a1/0x2c0 ? do_symlinkat+0x108/0x1c0 ? __pfx_smb2_compound_op+0x10/0x10 [cifs] ? kmem_cache_free+0x118/0x3e0 ? cifs_get_writable_path+0xeb/0x1a0 [cifs] smb2_get_reparse_inode+0x423/0x540 [cifs] ? __pfx_smb2_get_reparse_inode+0x10/0x10 [cifs] ? rcu_is_watching+0x20/0x50 ? __kmalloc_noprof+0x37c/0x480 ? smb2_create_reparse_symlink+0x257/0x490 [cifs] ? smb2_create_reparse_symlink+0x38f/0x490 [cifs] smb2_create_reparse_symlink+0x38f/0x490 [cifs] ? __pfx_smb2_create_reparse_symlink+0x10/0x10 [cifs] ? find_held_lock+0x8a/0xa0 ? hlock_class+0x32/0xb0 ? __build_path_from_dentry_optional_prefix+0x19d/0x2e0 [cifs] cifs_symlink+0x24f/0x960 [cifs] ? __pfx_make_vfsuid+0x10/0x10 ? __pfx_cifs_symlink+0x10/0x10 [cifs] ? make_vfsuid+0x6b/0xc0 ? generic_permission+0x96/0x2d0 vfs_symlink+0x1a1/0x2c0 do_symlinkat+0x108/0x1c0 ? __pfx_do_symlinkat+0x10/0x10 ? strncpy_from_user+0xaa/0x160 __x64_sys_symlinkat+0xb9/0xf0 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f08d75c13bb
CVE-2024-50152	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix possible double free in smb2_set_ea() Clang static checker(scan-build) warning# fs/smb/client/smb2ops.c:1304:2: Attempt to free released memory. 1304 kfree(ea); ^~~~~~ There is a double free in such case: 'ea' is initialized to NULL -> 'first successful memory allocation for ea' -> 'something failed, goto sea_exit' -> 'first memory release for ea' -> 'goto replay_again' -> 'second goto sea_exit before allocate memory for ea' -> 'second memory release for ea resulted in double free'. Re-initialie 'ea' to NULL near to the replay_again label, it can fix this double free problem.
CVE-2024-50153	In the Linux kernel, the following vulnerability has been resolved: scsi: target: core: Fix null-ptr-deref in target_alloc_device() There is a null-ptr-deref issue reported by KASAN: BUG: KASAN: null-ptr-deref in target_alloc_device+0xbc4/0xbe0 [target_core_mod] ... kasan_report+0xb9/0xf0 target_alloc_device+0xbc4/0xbe0 [target_core_mod] core_dev_setup_virtual_lun0+0xef/0x1f0 [target_core_mod] target_core_init_configfs+0x205/0x420 [target_core_mod] do_one_initcall+0xdd/0x4e0 ... entry_SYSCALL_64_after_hwframe+0x76/0x7e In target_alloc_device(), if allocating memory for dev queues fails, then dev will be freed by dev->transport->free_device(), but dev->transport is not initialized at that time, which will lead to a null pointer reference problem. Fixing this bug by freeing dev with hba->backend->ops->free_device().

CVE ID	Description
CVE-2024-50154	In the Linux kernel, the following vulnerability has been resolved: tcp/dccp: Don't use timer_pending() in reqsk_queue_unlink(). Martin KaFai Lau reported use-after-free [0] in reqsk_timer_handler(). """" We are seeing a use-after-free from a bpf prog attached to trace_tcp_retransmit_synack. The program passes the req->sk to the bpf_sk_storage_get_tracing kernel helper which does check for null before using it. """" The commit 83fccfc3940c ("inet: fix potential deadlock in reqsk_queue_unlink()") added timer_pending() in reqsk_queue_unlink() not to call del_timer_sync() from reqsk_timer_handler(), but it introduced a small race window. Before the timer is called, expire_timers() calls detach_timer(timer, true) to clear timer->entry.pprev and marks it as not pending. If reqsk_queue_unlink() checks timer_pending() just after expire_timers() calls detach_timer(), TCP will miss del_timer_sync(); the reqsk timer will continue running and send multiple SYN+ACKs until it expires. The reported UAF could happen if req->sk is close()d earlier than the timer expiration, which is 63s by default. The scenario would be 1. inet_csk_complete_hashdance() calls inet_csk_reqsk_queue_drop(), but del_timer_sync() is missed 2. reqsk timer is executed and scheduled again 3. req->sk is accept()ed and reqsk_put() decrements rsk_refcnt, but reqsk timer still has another one, and inet_csk_accept() does not clear req->sk for non-TFO sockets 4. sk is close()d 5. reqsk timer is executed again, and BPF touches req->sk Let's not use timer_pending() by passing the caller context to __inet_csk_reqsk_queue_drop(). Note that reqsk timer is pinned, so the issue does not happen in most use cases. [1] [0] BUG: Kfence: use-after-free read in bpf_sk_storage_get_tracing+0x2e/0x1b0 Use-after-free read at 0x00000000a891fb3a (in kfence-#1): bpf_sk_storage_get_tracing+0x2e/0x1b0 bpf_prog_5ea3e95db6da0438_tcp_retransmit_synack+0x1d20/0x1dda bpf_trace_run2+0x4c/0xc0 tcp_rtx_synack+0xf9/0x100 reqsk_timer_handler+0xda/0x3d0 run_timer_softirq+0x292/0x8a0 irq_exit_rcu+0xf5/0x320 sysvec_apic_timer_interrupt+0x6d/0x80 asm_sysvec_apic_timer_interrupt+0x16/0x20 intel_idle_irq+0x5a/0xa0 cpuidle_enter_state+0x94/0x273 cpu_startup_entry+0x15e/0x260 start_secondary+0x8a/0x90 secondary_startup_64_no_verify+0xfa/0xfb kfence-#1: 0x00000000a72cc7b6-0x00000000d97616d9, size=2376, cache=TCPv6 allocated by task 0 on cpu 9 at 260507.901592s: sk_prot_alloc+0x35/0x140 sk_clone_lock+0x1f/0x3f0 inet_csk_clone_lock+0x15/0x160 tcp_create_openreq_child+0x1f/0x410 tcp_v6_syn_recv_sock+0x1da/0x700 tcp_check_req+0x1fb/0x510 tcp_v6_rcv+0x98b/0x1420 ipv6_list_rcv+0x2258/0x26e0 napi_complete_done+0x5b1/0x2990 mlx5e_napi_poll+0x2ae/0x8d0 net_rx_action+0x13e/0x590 irq_exit_rcu+0xf5/0x320 common_interrupt+0x80/0x90 asm_common_interrupt+0x22/0x40 cpuidle_enter_state+0xfb/0x273 cpu_startup_entry+0x15e/0x260 start_secondary+0x8a/0x90 secondary_startup_64_no_verify+0xfa/0xfb freed by task 0 on cpu 9 at 260507.927527s: rcu_core_si+0x4ff/0xf10 irq_exit_rcu+0xf5/0x320 sysvec_apic_timer_interrupt+0x6d/0x80 asm_sysvec_apic_timer_interrupt+0x16/0x20 cpuidle_enter_state+0xfb/0x273 cpu_startup_entry+0x15e/0x260 start_secondary+0x8a/0x90 secondary_startup_64_no_verify+0xfa/0xfb
CVE-2024-50155	In the Linux kernel, the following vulnerability has been resolved: netdevsim: use cond_resched() in nsim_dev_trap_report_work() I am still seeing many syzbot reports hinting that syzbot might fool nsim_dev_trap_report_work() with hundreds of ports [1] Lets use cond_resched(), and system_unbound_wq instead of implicit system_wq. [1] INFO: task syz-executor:20633 blocked for more than 143 seconds. Not tainted 6.12.0-rc2-syzkaller-00205-g1d227fcc7222 #0 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:syz-executor state:D stack:25856 pid:20633 tgid:20633 ppid:1 flags:0x00004006 ... NMI backtrace for cpu 1 CPU: 1 UID: 0 PID: 16760 Comm: kworker/1:0 Not tainted 6.12.0-rc2-syzkaller-00205-g1d227fcc7222 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: events nsim_dev_trap_report_work RIP: 0010: __sanitizer_cov_trace_pc+0x0/0x70 kernel/kcov.c:210 Code: 89 fb e8 23 00 00 00 48 8b 3d 04 fb 9c 0c 48 89 de 5b e9 c3 c7 5d 00 0f 1f 00 90 90 90 90 90 90 90 90 90 90 90 90 90 <f3> 0f 1e fa 48 8b 04 24 65 48 8b 0c 25 c0 d7 03 00 65 8b 15 60 f0 RSP: 0018:ffffc90000a187e0 EFLAGS: 00000246 RAX: 0000000000000100 RBX: fffffc90000a188e0 RCX: fffff888027d3bc0 RDX: fffff888027d3bc0 RSI: 0000000000000000 RDI: 0000000000000000 RBP: fffff88804a2e6000 R08: ffffffff8a4bc495 R09: ffffffff89da3577 R10: 0000000000000004 R11: ffffffff8a4bc2b0 R12: dffffc0000000000 R13: fffff88806573b503 R14: dffffc0000000000 R15: fffff8880663cca0 F5: 0000000000000000(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fc90a747f98 CR3: 000000000e734000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 000000000000002b DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400 Call Trace: <NMI> <NMI> <TASK> __local_bh_enable_ip+0x1bb/0x200 kernel/softirq.c:382 spin_unlock_bh include/linux/spinlock.h:396 [inline] nsim_dev_trap_report drivers/net/netdevsim/dev.c:820 [inline] nsim_dev_trap_report_work+0x75d/0xaa0 drivers/net/netdevsim/dev.c:850 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa63/0x1850 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 <TASK>

CVE ID	Description
CVE-2024-50156	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Avoid NULL dereference in msm_disp_state_print_regs() If the allocation in msm_disp_state_dump_regs() failed then `block->state` can be NULL. The msm_disp_state_print_regs() function _does_ have code to try to handle it with: if (*reg) dump_addr = *reg; ...but since "dump_addr" is initialized to NULL the above is actually a noop. The code then goes on to dereference `dump_addr`. Make the function print "Registers not stored" when it sees a NULL to solve this. Since we're touching the code, fix msm_disp_state_print_regs() not to pointlessly take a double-pointer and properly mark the pointer as `const`. Patchwork: https://patchwork.freedesktop.org/patch/619657/
CVE-2024-50157	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Avoid CPU lockups due fifo occupancy check loop Driver waits indefinitely for the fifo occupancy to go below a threshold as soon as the pacing interrupt is received. This can cause soft lockup on one of the processors, if the rate of DB is very high. Add a loop count for FPGA and exit the __wait_for_fifo_occupancy_below_th if the loop is taking more time. Pacing will be continuing until the occupancy is below the threshold. This is ensured by the checks in bnxt_re_pacing_timer_exp and further scheduling the work for pacing based on the fifo occupancy.
CVE-2024-50158	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix out of bound check Driver exports pacing stats only on GenP5 and P7 adapters. But while parsing the pacing stats, driver has a check for "rdev->dbr_pacing". This caused a trace when KASAN is enabled. BUG: KASAN: slab-out-of-bounds in bnxt_re_get_hw_stats+0x2b6a/0x2e00 [bnxt_re] Write of size 8 at addr ffff8885942a6340 by task modprobe/4809
CVE-2024-50159	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix the double free in scmi_debugfs_common_setup() Clang static checker(scan-build) throws below warning# drivers/firmware/arm_scmi/driver.c:line 2915, column 2 Attempt to free released memory. When devm_add_action_or_reset() fails, scmi_debugfs_common_cleanup() will run twice which causes double free of 'dbg->name'. Remove the redundant scmi_debugfs_common_cleanup() to fix this problem.
CVE-2024-50160	In the Linux kernel, the following vulnerability has been resolved: ALSA: hda/cs8409: Fix possible NULL dereference If snd_hda_gen_add_kctl fails to allocate memory and returns NULL, then NULL pointer dereference will occur in the next line. Since dolphin_fixups function is a hda_fixup function which is not supposed to return any errors, add simple check before dereference, ignore the fail. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2024-50162	In the Linux kernel, the following vulnerability has been resolved: bpf: devmap: provide rxq after redirect rxq contains a pointer to the device from where the redirect happened. Currently, the BPF program that was executed after a redirect via BPF_MAP_TYPE_DEVMAP* does not have it set. This is particularly bad since accessing ingress_ifindex, e.g. SEC("xdp") int prog(struct xdp_md *pkt) { return bpf_redirect_map(&dev_redirect_map, 0, 0); } SEC("xdp/devmap") int prog_after_redirect(struct xdp_md *pkt) { bpf_printk("ifindex %i", pkt->ingress_ifindex); return XDP_PASS; } depends on access to rxq, so a NULL pointer gets dereferenced: <1>[574.475170] BUG: kernel NULL pointer dereference, address: 0000000000000000 <1>[574.475188] #PF: supervisor read access in kernel mode <1>[574.475194] #PF: error_code(0x0000) - not-present page <6>[574.475199] PGD 0 P4D 0 <4>[574.475207] Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI <4>[574.475217] CPU: 4 UID: 0 PID: 217 Comm: kworker/4:1 Not tainted 6.11.0-rc5-reduced-00859-g780801200300 #23 <4>[574.475226] Hardware name: Intel(R) Client Systems NUC13ANHi7/NUC13ANBi7, BIOS ANRPL357.0026.2023.0314.1458 03/14/2023 <4>[574.475231] Workqueue: mld mld_ifc_work <4>[574.475247] RIP: 0010:bpf_prog_5e13354d9cf5018a_prog_after_redirect+0x17/0x3c <4>[574.475257] Code: cc cc cc cc cc cc 80 00 00 00 cc cc cc cc cc cc f3 0f 1e fa 0f 1f 44 00 00 66 90 55 48 89 e5 f3 0f 1e fa 48 8b 57 20 <48> 8b 52 00 8b 92 e0 00 00 00 48 bf f8 a6 d5 c4 5d a0 ff ff be 0b <4>[574.475263] RSP: 0018:ffffa62440280c98 EFLAGS: 00010206 <4>[574.475269] RAX: fffffa62440280cd8 RBX: 0000000000000001 RCX: 0000000000000000 <4>[574.475274] RDX: 0000000000000000 RSI: fffffa62440549048 RDI: fffffa62440280ce0 <4>[574.475278] RBP: fffffa62440280c98 R08: 0000000000000002 R09: 0000000000000001 <4>[574.475281] R10: fffffa05dc8b98000 R11: fffffa05f577fca40 R12: fffffa05dcab24000 <4>[574.475285] R13: fffffa62440280ce0 R14: fffffa62440549048 R15: fffffa62440549000 <4>[574.475289] FS: 0000000000000000(0000) GS:ffffa05f4700000(0000) knlGS:0000000000000000 <4>[574.475294] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 <4>[574.475298] CR2: 0000000000000000 CR3: 000000025522e000 CR4: 0000000000f50ef0 <4>[574.475303] PKRU: 55555554 <4>[574.475306] Call Trace: <4>[574.475313] <IRQ> <4>[574.475318] ? __die+0x23/0x70 <4>[574.475329] ? page_fault_oops+0x180/0x4c0 <4>[574.475339] ? skb_pp_cow_data+0x34c/0x490 <4>[574.475346] ? kmem_cache_free+0x257/0x280 <4>[574.475357] ? exc_page_fault+0x67/0x150 <4>[574.475368] ? asm_exc_page_fault+0x26/0x30 <4>[574.475381] ? bpf_prog_5e13354d9cf5018a_prog_after_redirect+0x17/0x3c <4>[574.475386] bq_xmit_all+0x158/0x420 <4>[574.475397] __dev_flush+0x30/0x90 <4>[574.475407] veth_poll+0x216/0x250 [veth] <4>[574.475421] __napi_poll+0x28/0x1c0 <4>[574.475430] net_rx_action+0x32d/0x3a0 <4>[574.475441] handle_softirqs+0xcb/0x2c0 <4>[574.475451] do_softirq+0x40/0x60 <4>[574.475458] </IRQ> <4>[574.475461] <TASK> <4>[574.475464] __local_bh_enable_ip+0x66/0x70 <4>[574.475471] __dev_queue_xmit+0x268/0xe40 <4>[574.475480] ? selinux_ip_postroute+0x213/0x420 <4>[574.475491] ? alloc_skb_with_frags+0x4a/0x1d0 <4>[574.475502] ip6_finish_output2+0x2be/0x640 <4>[574.475512] ? nf_hook_slow+0x42/0xf0 <4>[574.475521] ip6_finish_output+0x194/0x300 <4>[574.475529] ? __pfx_ip6_finish_output+0x10/0x10 <4>[574.475538] mld_sendpack+0x17c/0x240 <4>[574.475548] mld_ifc_work+0x192/0x410 <4>[574.475557] process_one_work+0x15d/0x380 <4>[574.475566] worker_thread+0x29d/0x3a0 <4>[574.475573] ? __pfx_worker_thread+0x10/0x10 <4>[574.475580] ? __pfx_worker_thread+0x10/0x10 <4>[574.475587] kthread+0xcd/0x100 <4>[574.475597] ? __pfx_kthread+0x10/0x10 <4>[574.475606] ret_from_fork+0x31/0x50 <4>[574.475615] ? __pfx_kthread+0x10/0x10 <4>[574.475623] ret_from_fork_asm+0x1a/0x ---truncated---
CVE-2024-50163	In the Linux kernel, the following vulnerability has been resolved: bpf: Make sure internal and UAPI bpf_redirect flags don't overlap The bpf_redirect_info is shared between the SKB and XDP redirect paths, and the two paths use the same numeric flag values in the ri->flags field (specifically, BPF_F_BROADCAST == BPF_F_NEXTHOP). This means that if skb bpf_redirect_neigh() is used with a non-NULL params argument and, subsequently, an XDP redirect is performed using the same bpf_redirect_info struct, the XDP path will get confused and end up crashing, which syzbot managed to trigger. With the stack-allocated bpf_redirect_info, the structure is no longer shared between the SKB and XDP paths, so the crash doesn't happen anymore. However, different code paths using identically-numbered flag values in the same struct field still seems like a bit of a mess, so this patch cleans that up by moving the flag definitions together and redefining the three flags in BPF_F_REDIRECT_INTERNAL to not overlap with the flags used for XDP. It also adds a BUILD_BUG_ON() check to make sure the overlap is not re-introduced by mistake.

CVE ID	Description
CVE-2024-50164	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix overloading of MEM_UNINIT's meaning Lonial reported an issue in the BPF verifier where check_mem_size_reg() has the following code: if (!tnum_is_const(reg->var_off)) /* For unprivileged variable accesses, disable raw * mode so that the program is required to * initialize all the memory that the helper could * just partially fill up. */ meta = NULL; This means that writes are not checked when the register containing the size of the passed buffer has not a fixed size. Through this bug, a BPF program can write to a map which is marked as read-only, for example, .rodata global maps. The problem is that MEM_UNINIT's initial meaning that "the passed buffer to the BPF helper does not need to be initialized" which was added back in commit 435faee1aae9 ("bpf, verifier: add ARG_PTR_TO_RAW_STACK type") got overloaded over time with "the passed buffer is being written to". The problem however is that checks such as the above which were added later via 06c1c049721a ("bpf: allow helpers access to variable memory") set meta to NULL in order force the user to always initialize the passed buffer to the helper. Due to the current double meaning of MEM_UNINIT, this bypasses verifier write checks to the memory (not boundary checks though) and only assumes the latter memory is read instead. Fix this by reverting MEM_UNINIT back to its original meaning, and having MEM_WRITE as an annotation to BPF helpers in order to then trigger the BPF verifier checks for writing to memory. Some notes: check_arg_pair_ok() ensures that for ARG_CONST_SIZE{,_OR_ZERO} we can access fn->arg_type[arg - 1] since it must contain a preceding ARG_PTR_TO_MEM. For check_mem_reg() the meta argument can be removed altogether since we do check both BPF_READ and BPF_WRITE. Same for the equivalent check_kfunc_mem_size_reg().
CVE-2024-50166	In the Linux kernel, the following vulnerability has been resolved: fsl/fman: Fix refcount handling of fman-related devices In mac_probe() there are multiple calls to of_find_device_by_node(), fman_bind() and fman_port_bind() which takes references to of_dev->dev. Not all references taken by these calls are released later on error path in mac_probe() and in mac_remove() which lead to reference leaks. Add references release.
CVE-2024-50167	In the Linux kernel, the following vulnerability has been resolved: be2net: fix potential memory leak in be_xmit() The be_xmit() returns NETDEV_TX_OK without freeing skb in case of be_xmit_enqueue() fails, add dev_kfree_skb_any() to fix it.
CVE-2024-50169	In the Linux kernel, the following vulnerability has been resolved: vsock: Update rx_bytes on read_skb() Make sure virtio_transport_inc_rx_pkt() and virtio_transport_dec_rx_pkt() calls are balanced (i.e. virtio_vsock_sock::rx_bytes doesn't lie) after vsock_transport::read_skb(). While here, also inform the peer that we've freed up space and it has more credit. Failing to update rx_bytes after packet is dequeued leads to a warning on SOCK_STREAM recv(): [233.396654] rx_queue is empty, but rx_bytes is non-zero [233.396702] WARNING: CPU: 11 PID: 40601 at net/vmw_vsock/virtio_transport_common.c:589
CVE-2024-50170	In the Linux kernel, the following vulnerability has been resolved: net: bcmasp: fix potential memory leak in bcmasp_xmit() The bcmasp_xmit() returns NETDEV_TX_OK without freeing skb in case of mapping fails, add dev_kfree_skb() to fix it.
CVE-2024-50172	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix a possible memory leak In bnxt_re_setup_chip_ctx() when bnxt_qplib_map_db_bar() fails driver is not freeing the memory allocated for "rdev->chip_ctx".
CVE-2024-50179	In the Linux kernel, the following vulnerability has been resolved: ceph: remove the incorrect Fw reference check when dirtying pages When doing the direct-io reads it will also try to mark pages dirty, but for the read path it won't hold the Fw caps and there is case will it get the Fw reference.
CVE-2024-50183	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Ensure DA_ID handling completion before deleting an NPIV instance Deleting an NPIV instance requires all fabric ndlps to be released before an NPIV's resources can be torn down. Failure to release fabric ndlps beforehand opens kref imbalance race conditions. Fix by forcing the DA_ID to complete synchronously with usage of wait_queue.
CVE-2024-50185	In the Linux kernel, the following vulnerability has been resolved: mptcp: handle consistently DSS corruption Bugged peer implementation can send corrupted DSS options, consistently hitting a few warning in the data path. Use DEBUG_NET assertions, to avoid the splat on some builds and handle consistently the error, dumping related MIBs and performing fallback and/or reset according to the subflow type.

CVE ID	Description
CVE-2024-50186	In the Linux kernel, the following vulnerability has been resolved: net: explicitly clear the sk pointer, when pf->create fails We have recently noticed the exact same KASAN splat as in commit 6cd4a78d962b ("net: do not leave a dangling sk pointer, when socket creation fails"). The problem is that commit did not fully address the problem, as some pf->create implementations do not use sk_common_release in their error paths. For example, we can use the same reproducer as in the above commit, but changing ping to arping. arping uses AF_PACKET socket and if packet_create fails, it will just sk_free the allocated sk object. While we could chase all the pf->create implementations and make sure they NULL the freed sk object on error from the socket, we can't guarantee future protocols will not make the same mistake. So it is easier to just explicitly NULL the sk pointer upon return from pf->create in __sock_create. We do know that pf->create always releases the allocated sk object on error, so if the pointer is not NULL, it is definitely dangling.
CVE-2024-50189	In the Linux kernel, the following vulnerability has been resolved: HID: amd_sfh: Switch to device-managed dmam_alloc_coherent() Using the device-managed version allows to simplify clean-up in probe() error path. Additionally, this device-managed ensures proper cleanup, which helps to resolve memory errors, page faults, btrfs going read-only, and btrfs disk corruption.
CVE-2024-50191	In the Linux kernel, the following vulnerability has been resolved: ext4: don't set SB_RDONLY after filesystem errors When the filesystem is mounted with errors=remount-ro, we were setting SB_RDONLY flag to stop all filesystem modifications. We knew this misses proper locking (sb->s_umount) and does not go through proper filesystem remount procedure but it has been the way this worked since early ext2 days and it was good enough for catastrophic situation damage mitigation. Recently, syzbot has found a way (see link) to trigger warnings in filesystem freezing because the code got confused by SB_RDONLY changing under its hands. Since these days we set EXT4_FLAGS_SHUTDOWN on the superblock which is enough to stop all filesystem modifications, modifying SB_RDONLY shouldn't be needed. So stop doing that.
CVE-2024-50194	In the Linux kernel, the following vulnerability has been resolved: arm64: probes: Fix uprobes for big-endian kernels The arm64 uprobes code is broken for big-endian kernels as it doesn't convert the in-memory instruction encoding (which is always little-endian) into the kernel's native endianness before analyzing and simulating instructions. This may result in a few distinct problems: * The kernel may erroneously reject probing an instruction which can safely be probed. * The kernel may erroneously permit stepping an instruction out-of-line when that instruction cannot be stepped out-of-line safely. * The kernel may erroneously simulate instruction incorrectly due to interpreting the byte-swapped encoding. The endianness mismatch isn't caught by the compiler or sparse because: * The arch_uprobe::{insn,ixol} fields are encoded as arrays of u8, so the compiler and sparse have no idea these contain a little-endian 32-bit value. The core uprobes code populates these with a memcpy() which similarly does not handle endianness. * While the uprobe_opcode_t type is an alias for __le32, both arch_uprobe_analyze_insn() and arch_uprobe_skip_sstep() cast from u8[] to the similarly-named probe_opcode_t, which is an alias for u32. Hence there is no endianness conversion warning. Fix this by changing the arch_uprobe::{insn,ixol} fields to __le32 and adding the appropriate __le32_to_cpu() conversions prior to consuming the instruction encoding. The core uprobes copies these fields as opaque ranges of bytes, and so is unaffected by this change. At the same time, remove MAX_UINSN_BYTES and consistently use AARCH64_INSN_SIZE for clarity. Tested with the following: #include <stdio.h> #include <stdbool.h> #define ninline __attribute__((ninline)) static ninline void *adrp_self(void) { void *addr; asm volatile(" adrp %x0, adrp_self\n" " add %x0, %x0, :lo12:adrp_self\n" : "=r" (addr)); } int main(int argc, char *argv) { void *ptr = adrp_self(); bool equal = (ptr == adrp_self); printf("adrp_self => %p\n" "adrp_self() => %p\n" "%s\n", adrp_self, ptr, equal ? "EQUAL" : "NOT EQUAL"); return 0; } where the adrp_self() function was compiled to: 00000000004007e0 <adrp_self>: 4007e0: 90000000 adrp x0, 400000 <__ehdr_start> 4007e4: 911f8000 add x0, x0, #0x7e0 4007e8: d65f03c0 ret Before this patch, the ADRP is not recognized, and is assumed to be steppable, resulting in corruption of the result: # ./adrp-self adrp_self => 0x4007e0 adrp_self() => 0x4007e0 EQUAL # echo 'p /root/adrp-self:0x007e0' > /sys/kernel/tracing/uprobe_events # echo 1 > /sys/kernel/tracing/events/uprobes/enable # ./adrp-self adrp_self => 0x4007e0 adrp_self() => 0xffffffff7e0 NOT EQUAL After this patch, the ADRP is correctly recognized and simulated: # ./adrp-self adrp_self => 0x4007e0 adrp_self() => 0x4007e0 EQUAL # # echo 'p /root/adrp-self:0x007e0' > /sys/kernel/tracing/uprobe_events # echo 1 > /sys/kernel/tracing/events/uprobes/enable # ./adrp-self adrp_self => 0x4007e0 adrp_self() => 0x4007e0 EQUAL

CVE ID	Description
CVE-2024-50195	In the Linux kernel, the following vulnerability has been resolved: posix-clock: Fix missing timespec64 check in pc_clock_settime() As Andrew pointed out, it will make sense that the PTP core checked timespec64 struct's tv_sec and tv_nsec range before calling ptp->info->settime64(). As the man manual of clock_settime() said, if tp.tv_sec is negative or tp.tv_nsec is outside the range [0..999,999,999], it should return EINVAL, which include dynamic clocks which handles PTP clock, and the condition is consistent with timespec64_valid(). As Thomas suggested, timespec64_valid() only check the timespec is valid, but not ensure that the time is in a valid range, so check it ahead using timespec64_valid_strict() in pc_clock_settime() and return -EINVAL if not valid. There are some drivers that use tp->tv_sec and tp->tv_nsec directly to write registers without validity checks and assume that the higher layer has checked it, which is dangerous and will benefit from this, such as hclge_ptp_settime(), igb_ptp_settime_i210(), _rcar_gen4_ptp_settime(), and some drivers can remove the checks of itself.
CVE-2024-50199	In the Linux kernel, the following vulnerability has been resolved: mm/swapfile: skip HugeTLB pages for unuse_vma I got a bad pud error and lost a 1GB HugeTLB when calling swapoff. The problem can be reproduced by the following steps: 1. Allocate an anonymous 1GB HugeTLB and some other anonymous memory. 2. Swapout the above anonymous memory. 3. run swapoff and we will get a bad pud error in kernel message: mm/pgtable-generic.c:42: bad pud 00000000743d215d(84000001400000e7) We can tell that pud_clear_bad is called by pud_none_or_clear_bad in unuse_pud_range() by ftrace. And therefore the HugeTLB pages will never be freed because we lost it from page table. We can skip HugeTLB pages for unuse_vma to fix it.
CVE-2024-50201	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: Fix encoder->possible_clones Include the encoder itself in its possible_clones bitmask. In the past nothing validated that drivers were populating possible_clones correctly, but that changed in commit 74d2aacbe840 ("drm: Validate encoder->possible_clones"). Looks like radeon never got the memo and is still not following the rules 100% correctly. This results in some warnings during driver initialization: Bogus possible_clones: [ENCODER:46:TV-46] possible_clones=0x4 (full encoder_mask=0x7) WARNING: CPU: 0 PID: 170 at drivers/gpu/drm/drm_mode_config.c:615 drm_mode_config_validate+0x113/0x39c ... (cherry picked from commit 3b6e7d40649c0d75572039aff9d0911864c689db)
CVE-2024-50203	In the Linux kernel, the following vulnerability has been resolved: bpf, arm64: Fix address emission with tag-based KASAN enabled When BPF_TRAMP_F_CALL_ORIG is enabled, the address of a bpf_trampoline_image struct on the stack is passed during the size calculation pass and an address on the heap is passed during code generation. This may cause a heap buffer overflow if the heap address is tagged because emit_a64_mov_i64() will emit longer code than it did during the size calculation pass. The same problem could occur without tag-based KASAN if one of the 16-bit words of the stack address happened to be all-ones during the size calculation pass. Fix the problem by assuming the worst case (4 instructions) when calculating the size of the bpf_trampoline_image address emission.
CVE-2024-50205	In the Linux kernel, the following vulnerability has been resolved: ALSA: firewire-lib: Avoid division by zero in apply_constraint_to_size() The step variable is initialized to zero. It is changed in the loop, but if it's not changed it will remain zero. Add a variable check before the division. The observed behavior was introduced by commit 826b5de90c0b ("ALSA: firewire-lib: fix insufficient PCM rule for period/buffer size"), and it is difficult to show that any of the interval parameters will satisfy the snd_interval_test() condition with data from the amdtpt_rate_table[] table. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-50206	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk_eth_soc: fix memory corruption during fq dma init The loop responsible for allocating up to MTK_FQ_DMA_LENGTH buffers must only touch as many descriptors, otherwise it ends up corrupting unrelated memory. Fix the loop iteration count accordingly.
CVE-2024-50207	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix reader locking when changing the sub buffer order The function ring_buffer_subbuf_order_set() updates each ring_buffer_per_cpu and installs new sub buffers that match the requested page order. This operation may be invoked concurrently with readers that rely on some of the modified data, such as the head bit (RB_PAGE_HEAD), or the ring_buffer_per_cpu.pages and reader_page pointers. However, no exclusive access is acquired by ring_buffer_subbuf_order_set(). Modifying the mentioned data while a reader also operates on them can then result in incorrect memory access and various crashes. Fix the problem by taking the reader_lock when updating a specific ring_buffer_per_cpu in ring_buffer_subbuf_order_set().
CVE-2024-50208	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix a bug while setting up Level-2 PBL pages Avoid memory corruption while setting up Level-2 PBL pages for the non MR resources when num_pages > 256K. There will be a single PDE page address (contiguous pages in the case of > PAGE_SIZE), but, current logic assumes multiple pages, leading to invalid memory access after 256K PBL entries in the PDE.

CVE ID	Description
CVE-2024-50209	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Add a check for memory allocation __alloc_pbl() can return error when memory allocation fails. Driver is not checking the status on one of the instances.
CVE-2024-50210	In the Linux kernel, the following vulnerability has been resolved: posix-clock: posix-clock: Fix unbalanced locking in pc_clock_settime() If get_clock_desc() succeeds, it calls fget() for the clockid's fd, and get the clk->rwsem read lock, so the error path should release the lock to make the lock balance and fput the clockid's fd to make the refcount balance and release the fd related resource. However the below commit left the error path locked behind resulting in unbalanced locking. Check timespec64_valid_strict() before get_clock_desc() to fix it, because the "ts" is not changed after that. [pabeni@redhat.com: fixed commit message typo]
CVE-2024-50211	In the Linux kernel, the following vulnerability has been resolved: udf: refactor inode_bmap() to handle error Refactor inode_bmap() to handle error since udf_next_aext() can return error now. On situations like ftruncate, udf_extend_file() can now detect errors and bail out early without resorting to checking for particular offsets and assuming internal behavior of these functions.
CVE-2024-50215	In the Linux kernel, the following vulnerability has been resolved: nvmet-auth: assign dh_key to NULL after kfree_sensitive ctrl->dh_key might be used across multiple calls to nvmet_setup_dhgroup() for the same controller. So it's better to nullify it after release on error path in order to avoid double free later in nvmet_destroy_auth(). Found by Linux Verification Center (linuxtesting.org) with Svace.
CVE-2024-50216	In the Linux kernel, the following vulnerability has been resolved: xfs: fix finding a last resort AG in xfs_filestream_pick_ag When the main loop in xfs_filestream_pick_ag fails to find a suitable AG it tries to just pick the online AG. But the loop for that uses args->pag as loop iterator while the later code expects pag to be set. Fix this by reusing the max_pag case for this last resort, and also add a check for impossible case of no AG just to make sure that the uninitialized pag doesn't even escape in theory.
CVE-2024-50217	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free of block device file in __btrfs_free_extra_devids() Mounting btrfs from two images (which have the same one fsid and two different dev_uuids) in certain executing order may trigger an UAF for variable 'device->bdev_file' in __btrfs_free_extra_devids(). And following are the details: 1. Attach image_1 to loop0, attach image_2 to loop1, and scan btrfs devices by ioctl(BTRFS_IOC_SCAN_DEV): / btrfs_device_1 # loop0 fs_device \ btrfs_device_2 # loop1 2. mount /dev/loop0 /mnt btrfs_open_devices btrfs_device_1->bdev_file = btrfs_get_bdev_and_sb(loop0) btrfs_device_2->bdev_file = btrfs_get_bdev_and_sb(loop1) btrfs_fill_super open_ctree fail: btrfs_close_devices // -ENOMEM btrfs_close_bdev(btrfs_device_1) fput(btrfs_device_1->bdev_file) // btrfs_device_1->bdev_file is freed btrfs_close_bdev(btrfs_device_2) fput(btrfs_device_2->bdev_file) 3. mount /dev/loop1 /mnt btrfs_open_devices btrfs_get_bdev_and_sb(&bdev_file) // EIO, btrfs_device_1->bdev_file is not assigned, // which points to a freed memory area btrfs_device_2->bdev_file = btrfs_get_bdev_and_sb(loop1) btrfs_fill_super open_ctree btrfs_free_extra_devids if (btrfs_device_1->bdev_file) fput(btrfs_device_1->bdev_file) // UAF ! Fix it by setting 'device->bdev_file' as 'NULL' after closing the btrfs_device in btrfs_close_one_device().
CVE-2024-50218	In the Linux kernel, the following vulnerability has been resolved: ocfs2: pass u64 to ocfs2_truncate_inline maybe overflow Syzbot reported a kernel BUG in ocfs2_truncate_inline. There are two reasons for this: first, the parameter value passed is greater than ocfs2_max_inline_data_with_xattr, second, the start and end parameters of ocfs2_truncate_inline are "unsigned int". So, we need to add a sanity check for byte_start and byte_len right before ocfs2_truncate_inline() in ocfs2_remove_inode_range(), if they are greater than ocfs2_max_inline_data_with_xattr return -EINVAL.

CVE ID	Description
CVE-2024-50220	In the Linux kernel, the following vulnerability has been resolved: fork: do not invoke uffd on fork if error occurs Patch series "fork: do not expose incomplete mm on fork". During fork we may place the virtual memory address space into an inconsistent state before the fork operation is complete. In addition, we may encounter an error during the fork operation that indicates that the virtual memory address space is invalidated. As a result, we should not be exposing it in any way to external machinery that might interact with the mm or VMAs, machinery that is not designed to deal with incomplete state. We specifically update the fork logic to defer khugepaged and ksm to the end of the operation and only to be invoked if no error arose, and disallow uffd from observing fork events should an error have occurred. This patch (of 2): Currently on fork we expose the virtual address space of a process to userland unconditionally if uffd is registered in VMAs, regardless of whether an error arose in the fork. This is performed in dup_userfaultfd_complete() which is invoked unconditionally, and performs two duties - invoking registered handlers for the UFFD_EVENT_FORK event via dup_fctx(), and clearing down userfaultfd_fork_ctx objects established in dup_userfaultfd(). This is problematic, because the virtual address space may not yet be correctly initialised if an error arose. The change in commit d24062914837 ("fork: use __mt_dup() to duplicate maple tree in dup_mmap()") makes this more pertinent as we may be in a state where entries in the maple tree are not yet consistent. We address this by, on fork error, ensuring that we roll back state that we would otherwise expect to clean up through the event being handled by userland and perform the memory freeing duty otherwise performed by dup_userfaultfd_complete(). We do this by implementing a new function, dup_userfaultfd_fail(), which performs the same loop, only decrementing reference counts. Note that we perform mmgrab() on the parent and child mm's, however userfaultfd_ctx_put() will mmdrop() this once the reference count drops to zero, so we will avoid memory leaks correctly here.
CVE-2024-50221	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Vangogh: Fix kernel memory out of bounds write KASAN reports that the GPU metrics table allocated in vangogh_tables_init() is not large enough for the memset done in smu_cm_n_init_soft_gpu_metrics(). Condensed report follows: [33.861314] BUG: KASAN: slab-out-of-bounds in smu_cm_n_init_soft_gpu_metrics+0x73/0x200 [amdgpu] [33.861799] Write of size 168 at addr ffff888129f59500 by task mangoapp/1067 ... [33.861808] CPU: 6 UID: 1000 PID: 1067 Comm: mangoapp Tainted: G W 6.12.0-rc4 #356 1a56f59a8b5182eef67eb7cb8b13594dd23b544 [33.861816] Tainted: [W]=WARN [33.861818] Hardware name: Valve Galileo/Galileo, BIOS F7G0107 12/01/2023 [33.861822] Call Trace: [33.861826] <TASK> [33.861829] dump_stack_lvl+0x66/0x90 [33.861838] print_report+0xce/0x620 [33.861853] kasan_report+0xda/0x110 [33.862794] kasan_check_range+0xfd/0x1a0 [33.862799] __asan_memset+0x23/0x40 [33.862803] smu_cm_n_init_soft_gpu_metrics+0x73/0x200 [amdgpu 13b1bc364ec578808f676eba412c20eaab792779] [33.863306] vangogh_get_gpu_metrics_v2_4+0x123/0xad0 [amdgpu 13b1bc364ec578808f676eba412c20eaab792779] [33.864257] vangogh_common_get_gpu_metrics+0xb0c/0xbcc0 [amdgpu 13b1bc364ec578808f676eba412c20eaab792779] [33.865682] amdgpu_dpm_get_gpu_metrics+0xcc/0x110 [amdgpu 13b1bc364ec578808f676eba412c20eaab792779] [33.866160] amdgpu_get_gpu_metrics+0x154/0x2d0 [amdgpu 13b1bc364ec578808f676eba412c20eaab792779] [33.867135] dev_attr_show+0x43/0xc0 [33.867147] sysfs_kf_seq_show+0x1f1/0x3b0 [33.867155] seq_read_iter+0x3f8/0x1140 [33.867173] vfs_read+0x76c/0xc50 [33.867198] ksys_read+0xfb/0x1d0 [33.867214] do_syscall_64+0x90/0x160 ... [33.867353] Allocated by task 378 on cpu 7 at 22.794876s: [33.867358] kasan_save_stack+0x33/0x50 [33.867364] kasan_save_track+0x17/0x60 [33.867367] __kasan_kmalloc+0x87/0x90 [33.867371] vangogh_init_smc_tables+0x3f9/0x840 [amdgpu] [33.867835] smu_sw_init+0xa32/0x1850 [amdgpu] [33.868299] amdgpu_device_init+0x467b/0x8d90 [amdgpu] [33.868733] amdgpu_driver_load_kms+0x19/0xf0 [amdgpu] [33.869167] amdgpu_pci_probe+0x2d6/0xcd0 [amdgpu] [33.869608] local_pci_probe+0xda/0x180 [33.869614] pci_device_probe+0x43f/0x6b0 Empirically we can confirm that the former allocates 152 bytes for the table, while the latter memsets the 168 large block. Root cause appears that when GPU metrics tables for v2_4 parts were added it was not considered to enlarge the table to fit. The fix in this patch is rather "brute force" and perhaps later should be done in a smarter way, by extracting and consolidating the part version to size logic to a common helper, instead of brute forcing the largest possible allocation. Nevertheless, for now this works and fixes the out of bounds write. v2: * Drop impossible v3_0 case. (Mario) (cherry picked from commit 0880f58f9609f0200483a49429af0f050d281703)
CVE-2024-50222	In the Linux kernel, the following vulnerability has been resolved: iov_iter: fix copy_page_from_iter_atomic() if KMAP_LOCAL_FORCE_MAP generic/077 on x86_32 CONFIG_DEBUG_KMAP_LOCAL_FORCE_MAP=y with highmem, on huge=always tmpfs, issues a warning and then hangs (interruptibly): WARNING: CPU: 5 PID: 3517 at mm/highmem.c:622 kunmap_local_indexed+0x62/0xc9 CPU: 5 UID: 0 PID: 3517 Comm: cp Not tainted 6.12.0-rc4 #2 ... copy_page_from_iter_atomic+0xa6/0x5ec generic_perform_write+0xf6/0x1b4 shmem_file_write_iter+0x54/0x67 Fix copy_page_from_iter_atomic() by limiting it in that case (include/linux/skbuff.h skb_frag_must_loop() does similar). But going forward, perhaps CONFIG_DEBUG_KMAP_LOCAL_FORCE_MAP is too surprising, has outlived its usefulness, and should just be removed?

CVE ID	Description
CVE-2024-50223	In the Linux kernel, the following vulnerability has been resolved: sched/numa: Fix the potential null pointer dereference in task_numa_work() When running stress-ng-vm-segv test, we found a null pointer dereference error in task_numa_work(). Here is the backtrace: [323676.066985] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000020 [323676.067108] CPU: 35 PID: 2694524 Comm: stress-ng-vm-se [323676.067113] pstate: 23401009 (nzCv daif +PAN -UAO +TCO +DIT +SSBS BTYPPE=--) [323676.067115] pc : vma_migratable+0x1c/0xd0 [323676.067122] lr : task_numa_work+0x1ec/0x4e0 [323676.067127] sp : ffff8000ada73d20 [323676.067128] x29: ffff8000ada73d20 x28: 0000000000000000 x27: 000000003e89f010 [323676.067130] x26: 0000000000080000 x25: ffff800081b5c0d8 x24: ffff800081b27000 [323676.067133] x23: 0000000000010000 x22: 0000000104d18cc0 x21: ffff0009f7158000 [323676.067135] x20: 0000000000000000 x19: 0000000000000000 x18: ffff8000ada73db8 [323676.067138] x17: 0001400000000000 x16: ffff800080df40b0 x15: 0000000000000035 [323676.067140] x14: ffff8000ada73cc8 x13: 1fffe0017cc72001 x12: ffff8000ada73cc8 [323676.067142] x11: ffff80008001160c x10: ffff000be639000c x9: ffff8000800f4ba4 [323676.067145] x8 : ffff000810375000 x7 : ffff8000ada73974 x6 : 0000000000000001 [323676.067147] x5 : 0068000b33e26707 x4 : 0000000000000001 x3 : ffff0009f7158000 [323676.067149] x2 : 0000000000000041 x1 : 0000000000004400 x0 : 0000000000000000 [323676.067152] Call trace: [323676.067153] vma_migratable+0x1c/0xd0 [323676.067155] task_numa_work+0x1ec/0x4e0 [323676.067157] task_work_run+0x78/0xd8 [323676.067161] do_notify_resume+0x1ec/0x290 [323676.067163] el0_svc+0x150/0x160 [323676.067167] el0t_64_sync_handler+0xf8/0x128 [323676.067170] el0t_64_sync+0x17c/0x180 [323676.067173] Code: d2888001 910003fd f9000bf3 aa0003f3 (f9401000) [323676.067177] SMP: stopping secondary CPUs [323676.070184] Starting crashdump kernel... stress-ng-vm-segv in stress-ng is used to stress test the SIGSEGV error handling function of the system, which tries to cause a SIGSEGV error on return from unmapping the whole address space of the child process. Normally this program will not cause kernel crashes. But before the munmap system call returns to user mode, a potential task_numa_work() for numa balancing could be added and executed. In this scenario, since the child process has no vma after munmap, the vma_next() in task_numa_work() will return a null pointer even if the vma iterator restarts from 0. Recheck the vma pointer before dereferencing it in task_numa_work().
CVE-2024-50224	In the Linux kernel, the following vulnerability has been resolved: spi: spi-fsl-dspi: Fix crash when not using GPIO chip select Add check for the return value of spi_get_csgpiod() to avoid passing a NULL pointer to gpiod_direction_output(), preventing a crash when GPIO chip select is not used. Fix below crash: [4.251960] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 [4.260762] Mem abort info: [4.263556] ESR = 0x0000000096000004 [4.267308] EC = 0x25: DABT (current EL), IL = 32 bits [4.272624] SET = 0, FnV = 0 [4.275681] EA = 0, S1PTW = 0 [4.278822] FSC = 0x04: level 0 translation fault [4.283704] Data abort info: [4.286583] ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 [4.292074] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [4.297130] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [4.302445] [0000000000000000] user address but active_mm is swapper [4.308805] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [4.315072] Modules linked in: [4.318124] CPU: 2 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.12.0-rc4-next-20241023-00008-ga20ec42c5fc1 #359 [4.328130] Hardware name: LS1046A QDS Board (DT) [4.332832] pstate: 40000005 (nZcv daif -PAN -UAO -TCO -DIT -SSBS BTYPPE=--) [4.339794] pc : gpiod_direction_output+0x34/0x5c [4.344505] lr : gpiod_direction_output+0x18/0x5c [4.349208] sp : ffff80008003b8f0 [4.352517] x29: ffff80008003b8f0 x28: 0000000000000000 x27: ffffc96bcc7e9068 [4.359659] x26: ffffc96bcc6e00b0 x25: ffffc96bcc598398 x24: ffff447400132810 [4.366800] x23: 0000000000000000 x22: 0000000011e1a300 x21: 0000000000020002 [4.373940] x20: 0000000000000000 x19: 0000000000000000 x18: ffffffff [4.381081] x17: ffff44740016e600 x16: 0000000500000003 x15: 0000000000000007 [4.388221] x14: 0000000000989680 x13: 0000000000020000 x12: 000000000000001e [4.395362] x11: 0044b82fa09b5a53 x10: 0000000000000019 x9 : 0000000000000008 [4.402502] x8 : 0000000000000002 x7 : 0000000000000007 x6 : 0000000000000000 [4.409641] x5 : 0000000000000200 x4 : 0000000002000000 x3 : 0000000000000000 [4.416781] x2 : 0000000000002220 x1 : 0000000000000000 x0 : 0000000000000000 [4.423921] Call trace: [4.426362] gpiod_direction_output+0x34/0x5c (P) [4.431067] gpiod_direction_output+0x18/0x5c (L) [4.435771] dspi_setup+0x220/0x334

CVE ID	Description
CVE-2024-50225	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix error propagation of split bios The purpose of btrfs_bbio_propagate_error() shall be propagating an error of split bio to its original btrfs_bio, and tell the error to the upper layer. However, it's not working well on some cases. * Case 1. Immediate (or quick) end_bio with an error When btrfs sends btrfs_bio to mirrored devices, btrfs calls btrfs_bio_end_io() when all the mirroring bios are completed. If that btrfs_bio was split, it is from btrfs_clone_bioset and its end_io function is btrfs_orig_write_end_io. For this case, btrfs_bbio_propagate_error() accesses the orig_bbio's bio context to increase the error count. That works well in most cases. However, if the end_io is called enough fast, orig_bbio's (remaining part after split) bio context may not be properly set at that time. Since the bio context is set when the orig_bbio (the last btrfs_bio) is sent to devices, that might be too late for earlier split btrfs_bio's completion. That will result in NULL pointer dereference. That bug is easily reproducible by running btrfs/146 on zoned devices [1] and it shows the following trace. [1] You need raid-stripe-tree feature as it create "-d raid0 -m raid1" FS. BUG: kernel NULL pointer dereference, address: 0000000000000020 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: Oops: 0000 [#1] PREEMPT SMP PTI CPU: 1 UID: 0 PID: 13 Comm: kworker/u32:1 Not tainted 6.11.0-rc7-BTRFS-ZNS+ #474 Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011 Workqueue: writeback wb_workfn (flush-btrfs-5) RIP: 0010:btrfs_bio_end_io+0xae/0xc0 [btrfs] BTRFS error (device dm-0): bdev /dev/mapper/error-test errs: wr 2, rd 0, flush 0, corrupt 0, gen 0 RSP: 0018:ffff90000006f248 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff888005a7f080 RCX: ffff90000006f1dc RDX: 0000000000000000 RSI: 0000000000000000a RDI: ffff888005a7f080 RBP: ffff888011dfc540 R08: 0000000000000000 R09: 0000000000000000 R10: ffffffff82e508e0 R11: 0000000000000005 R12: ffff88800ddfb5e8 R13: ffff888005a7f080 R14: ffff888005a7f158 R15: ffff888005a7f158 FS: 0000000000000000(0000) GS:ffff88803ea80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000020 CR3: 0000000002e22006 CR4: 0000000000370ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __die_body.cold+0x19/0x26 ? page_fault_oops+0x13e/0x2b0 ? _printk+0x58/0x73 ? do_user_addr_fault+0x5f/0x750 ? exc_page_fault+0x76/0x240 ? asm_exc_page_fault+0x22/0x30 ? btrfs_bio_end_io+0xae/0xc0 [btrfs] ? btrfs_log_dev_io_error+0x7f/0x90 [btrfs] btrfs_orig_write_end_io+0x51/0x90 [btrfs] dm_submit_bio+0x5c2/0xa50 [dm_mod] ? find_held_lock+0x2b/0x80 ? blk_try_enter_queue+0x90/0x1e0 __submit_bio+0xe0/0x130 ? ktime_get+0x10a/0x160 ? lockdep_hardirqs_on+0x74/0x100 submit_bio_noacct_nocheck+0x199/0x410 btrfs_submit_bio+0x7d/0x150 [btrfs] btrfs_submit_chunk+0x1a1/0x6d0 [btrfs] ? lockdep_hardirqs_on+0x74/0x100 ? __folio_start_writeback+0x10/0x2c0 btrfs_submit_bbio+0x1c/0x40 [btrfs] submit_one_bio+0x44/0x60 [btrfs] submit_extent_folio+0x13f/0x330 [btrfs] ? btrfs_set_range_writeback+0xa3/0xd0 [btrfs] extent_writepage_io+0x18b/0x360 [btrfs] extent_write_locked_range+0x17c/0x340 [btrfs] ? __pfx_end_bbio_data_write+0x10/0x10 [btrfs] run_delalloc_cow+0x71/0xd0 [btrfs] btrfs_run_delalloc_range+0x176/0x500 [btrfs] ? find_lock_delalloc_range+0x119/0x260 [btrfs] writepage_delalloc+0x2ab/0x480 [btrfs] extent_write_cache_pages+0x236/0x7d0 [btrfs] btrfs_writepages+0x72/0x130 [btrfs] do_writepages+0xd4/0x240 ? find_held_lock+0x2b/0x80 ? wbc_attach_and_unlock_inode+0x12c/0x290 ? wbc_attach_and_unlock_inode+0x12c/0x29 ---truncated---

CVE ID	Description
CVE-2024-50226	In the Linux kernel, the following vulnerability has been resolved: cxl/port: Fix use-after-free, permit out-of-order decoder shutdown In support of investigating an initialization failure report [1], cxl_test was updated to register mock memory-devices after the mock root-port/bus device had been registered. That led to cxl_test crashing with a use-after-free bug with the following signature: cxl_port_attach_region: cxl region3: cxl_host_bridge.0:port3 decoder3.0 add: mem0:decoder7.0 @ 0 next: cxl_switch_uport.0 nr_eps: 1 nr_targets: 1 cxl_port_attach_region: cxl region3: cxl_host_bridge.0:port3 decoder3.0 add: mem4:decoder14.0 @ 1 next: cxl_switch_uport.0 nr_eps: 2 nr_targets: 1 cxl_port_setup_targets: cxl region3: cxl_switch_uport.0:port6 target[0] = cxl_switch_dport.0 for mem0:decoder7.0 @ 0 1) cxl_port_setup_targets: cxl region3: cxl_switch_uport.0:port6 target[1] = cxl_switch_dport.4 for mem4:decoder14.0 @ 1 [...] cxld_unregister: cxl decoder14.0: cxl_region_decode_reset: cxl_region region3: mock_decoder_reset: cxl_port port3: decoder3.0 reset 2) mock_decoder_reset: cxl_port port3: decoder3.0: out of order reset, expected decoder3.1 cxl_endpoint_decoder_release: cxl decoder14.0: [...] cxld_unregister: cxl decoder7.0: 3) cxl_region_decode_reset: cxl_region region3: Oops: general protection fault, probably for non-canonical address 0x6b6b6b6b6b6b6b6c3: 0000 [#1] PREEMPT SMP PTI [...] RIP: 0010:to_cxl_port+0x8/0x60 [cxl_core] [...] Call Trace: <TASK> cxl_region_decode_reset +0x69/0x190 [cxl_core] cxl_region_detach+0xe8/0x210 [cxl_core] cxl_decoder_kill_region+0x27/0x40 [cxl_core] cxld_unregister+0x5d/0x60 [cxl_core] At 1) a region has been established with 2 endpoint decoders (7.0 and 14.0). Those endpoints share a common switch-decoder in the topology (3.0). At teardown, 2), decoder14.0 is the first to be removed and hits the "out of order reset case" in the switch decoder. The effect though is that region3 cleanup is aborted leaving it in-tact and referencing decoder14.0. At 3) the second attempt to teardown region3 trips over the stale decoder14.0 object which has long since been deleted. The fix here is to recognize that the CXL specification places no mandate on in-order shutdown of switch-decoders, the driver enforces in-order allocation, and hardware enforces in-order commit. So, rather than fail and leave objects dangling, always remove them. In support of making cxl_region_decode_reset() always succeed, cxl_region_invalidate_memregion() failures are turned into warnings. Crashing the kernel is ok there since system integrity is at risk if caches cannot be managed around physical address mutation events like CXL region destruction. A new device_for_each_child_reverse_from() is added to cleanup port->commit_end after all dependent decoders have been disabled. In other words if decoders are allocated 0->1->2 and disabled 1->2->0 then port->commit_end only decrements from 2 after 2 has been disabled, and it decrements all the way to zero since 1 was disabled previously.
CVE-2024-50230	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix kernel bug due to missing clearing of checked flag Syzbot reported that in directory operations after nilfs2 detects filesystem corruption and degrades to read-only, __block_write_begin_int(), which is called to prepare block writes, may fail the BUG_ON check for accesses exceeding the folio/page size, triggering a kernel bug. This was found to be because the "checked" flag of a page/folio was not cleared when it was discarded by nilfs2's own routine, which causes the sanity check of directory entries to be skipped when the directory page/folio is reloaded. So, fix that. This was necessary when the use of nilfs2's own page discard routine was applied to more than just metadata files.
CVE-2024-50231	In the Linux kernel, the following vulnerability has been resolved: iio: gts-helper: Fix memory leaks in iio_gts_build_avail_scale_table() modprobe iio-test-gts and rmmid it, then the following memory leak occurs: unreferenced object 0xfffff80c810be00 (size 64): comm "kunit_try_catch", pid 1654, jiffies 4294913981 hex dump (first 32 bytes): 02 00 00 00 08 00 00 00 20 00 00 00 40 00 00 00 @... 80 00 00 00 00 02 00 00 00 04 00 00 00 08 00 00 backtrace (crc a63d875e): [<0000000028c1b3c2>] kmemleak_alloc+0x34/0x40 [<000000001d6ecc87>] __kmallocc_noprof +0x2bc/0x3c0 [<00000000393795c1>] devm_iio_init_iio_gts+0x4b4/0x16f4 [<0000000071bb4b09>] 0xfffff8052a62e0 [<000000000315bc18>] 0xfffff8052a6488 [<000000000f9dc55b5>] kunit_try_run_case+0x13c/0x3ac [<00000000175a3fd4>] kunit_generic_run_threadfn_adapter +0x80/0xec [<00000000f505065d>] kthread+0x2e8/0x374 [<000000000bbfb0e5d>] ret_from_fork +0x10/0x20 unreferenced object 0xfffff80cbfe9e70 (size 16): comm "kunit_try_catch", pid 1658, jiffies 4294914015 hex dump (first 16 bytes): 10 00 00 00 40 00 00 00 80 00 00 00 00 00 00 00@..... backtrace (crc 857f0cb4): [<00000000028c1b3c2>] kmemleak_alloc+0x34/0x40 [<000000001d6ecc87>] __kmallocc_noprof+0x2bc/0x3c0 [<00000000393795c1>] devm_iio_init_iio_gts +0x4b4/0x16f4 [<0000000071bb4b09>] 0xfffff8052a62e0 [<0000000007d089d45>] 0xfffff8052a6864 [<000000000f9dc55b5>] kunit_try_run_case+0x13c/0x3ac [<00000000175a3fd4>] kunit_generic_run_threadfn_adapter+0x80/0xec [<00000000f505065d>] kthread+0x2e8/0x374 [<000000000bbfb0e5d>] ret_from_fork+0x10/0x20 It includes 5*5 times "size 64" memory leaks, which correspond to 5 times test_init_iio_gain_scale() calls with gts_test_gains size 10 (10*size(int)) and gts_test_itimes size 5. It also includes 5*1 times "size 16" memory leak, which correspond to one time __test_init_iio_gain_scale() call with gts_test_gains_gain_low size 3 (3*size(int)) and gts_test_itimes size 5. The reason is that the per_time_gains[i] is not freed which is allocated in the "gts->num_itime" for loop in iio_gts_build_avail_scale_table().

CVE ID	Description
CVE-2024-50232	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ad7124: fix division by zero in ad7124_set_channel_odr() In the ad7124_write_raw() function, parameter val can potentially be zero. This may lead to a division by zero when DIV_ROUND_CLOSEST() is called within ad7124_set_channel_odr(). The ad7124_write_raw() function is invoked through the sequence: iio_write_channel_raw() -> iio_write_channel_attribute() -> iio_channel_write(), with no checks in place to ensure val is non-zero.
CVE-2024-50234	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlegacy: Clear stale interrupts before resuming device iw4965 fails upon resume from hibernation on my laptop. The reason seems to be a stale interrupt which isn't being cleared out before interrupts are enabled. We end up with a race between the resume trying to bring things back up, and the restart work (queued from the interrupt handler) trying to bring things down. Eventually the whole thing blows up. Fix the problem by clearing out any stale interrupts before interrupts get enabled during resume. Here's a debug log of the incident: [12.042589] ieee80211 phy0: il_isr ISR into 0x00000080, enabled 0xaa00008b, fh 0x00000000 [12.042625] ieee80211 phy0: il4965_irq_tasklet into 0x00000080, enabled 0x00000000, fh 0x00000000 [12.042651] iw4965 0000:10:00.0: RF_KILL bit toggled to enable radio. [12.042653] iw4965 0000:10:00.0: On demand firmware reload [12.042690] ieee80211 phy0: il4965_irq_tasklet End into 0x00000000, enabled 0xaa00008b, fh 0x00000000, flags 0x00000282 [12.052207] ieee80211 phy0: il4965_mac_start enter [12.052212] ieee80211 phy0: il_prep_station Add STA to driver ID 31: ff:ff:ff:ff:ff:ff [12.052244] ieee80211 phy0: il4965_set_hw_ready hardware ready [12.052324] ieee80211 phy0: il_apm_init Init card's basic functions [12.052348] ieee80211 phy0: il_apm_init L1 Enabled; Disabling LOS [12.055727] ieee80211 phy0: il4965_load_bsm Begin load bsm [12.056140] ieee80211 phy0: il4965_verify_bsm Begin verify bsm [12.058642] ieee80211 phy0: il4965_verify_bsm BSM bootstrap uCode image OK [12.058721] ieee80211 phy0: il4965_load_bsm BSM write complete, poll 1 iterations [12.058734] ieee80211 phy0: __il4965_up iw4965 is coming up [12.058737] ieee80211 phy0: il4965_mac_start Start UP work done. [12.058757] ieee80211 phy0: __il4965_down iw4965 is going down [12.058761] ieee80211 phy0: il_scan_cancel_timeout Scan cancel timeout [12.058762] ieee80211 phy0: il_do_scan_abort Not performing scan to abort [12.058765] ieee80211 phy0: il_clear_ucode_stations Clearing ucode stations in driver [12.058767] ieee80211 phy0: il_clear_ucode_stations No active stations found to be cleared [12.058819] ieee80211 phy0: _il_apm_stop Stop card, put in low power state [12.058827] ieee80211 phy0: _il_apm_stop_master stop master [12.058864] ieee80211 phy0: il4965_clear_free_frames 0 frames on pre-allocated heap on clear. [12.058869] ieee80211 phy0: Hardware restart was requested [16.132299] iw4965 0000:10:00.0: START_ALIVE timeout after 4000ms. [16.132303] ----- [cut here]----- [16.132304] Hardware became unavailable upon resume. This could be a software issue prior to suspend or a hardware issue. [16.132338] WARNING: CPU: 0 PID: 181 at net/mac80211/util.c:1826 ieee80211_reconfig+0x8f/0x14b0 [mac80211] [16.132390] Modules linked in: ctr ccm sch_fq_codel xt_tcpudp xt_multiport xt_state iptable_filter iptable_nat nf_nat nf_conntrack nf_defrag_ipv4 ip_tables x_tables binfmt_misc joydev mousedev btusb btrtl btmintel btbcm bluetooth ecch_generic ecc iTCO_wdt i2c_dev iw4965 iwlegacy coretemp snd_hda_codec_analog pcspkr psmouse mac80211 snd_hda_codec_generic libarc4 sdhci_pci cqhci sha256_generic sdhci libsha256 firewire_ohci snd_hda_intel snd_intel_dspcfg mmc_core snd_hda_codec snd_hwdep firewire_core led_class iosf_mbi snd_hda_core uhci_hcd lpc_ich crc_itu_t cfg80211 ehci_pci ehci_hcd snd_pcm usbcore mfd_core rfkill snd_timer snd_usb_common soundcore video parport_pc parport intel_agp wmi intel_gtt backlight e1000e agpgart evdev [16.132456] CPU: 0 UID: 0 PID: 181 Comm: kworker/u8:6 Not tainted 6.11.0-cl+ #143 [16.132460] Hardware name: Hewlett-Packard HP Compaq 6910p/30BE, BIOS 68MCU Ver. F.19 07/06/2010 [16.132463] Workqueue: async async_run_entry_fn [16.132469] RIP: 0010:ieee80211_reconfig+0x8f/0x14b0 [mac80211] [16.132501] Code: da 02 00 0 ---truncated---
CVE-2024-50235	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: clear wdev->qcm_config pointer on free When we free wdev->qcm_config when unregistering, we also need to clear out the pointer since the same wdev/netdev may get re-registered in another network namespace, then destroyed later, running this code again, which results in a double-free.
CVE-2024-50236	In the Linux kernel, the following vulnerability has been resolved: wifi: ath10k: Fix memory leak in management tx In the current logic, memory is allocated for storing the MSDU context during management packet TX but this memory is not being freed during management TX completion. Similar leaks are seen in the management TX cleanup logic. Kmemleak reports this problem as below, unreferenced object 0xfffff80b64ed250 (size 16): comm "kworker/u16:7", pid 148, jiffies 4294687130 (age 714.199s) hex dump (first 16 bytes): 00 2b d8 d8 80 ff ff c4 74 e9 fd 07 00 00 00 .+.....t..... backtrace: [<fffffe67b245dc>] __kmem_cache_alloc_node+0x1e4/0x2d8 [<fffffe67adde88>] kmalloc_trace+0x48/0x110 [<fffffe6bbd765fc>] ath10k_wmi_tlv_op_gen_mgmt_tx_send+0xd4/0x1d8 [ath10k_core] [<fffffe6bbd3eed4>] ath10k_mgmt_over_wmi_tx_work+0x134/0x298 [ath10k_core] [<fffffe678d5974>] process_scheduled_works+0x1ac/0x400 [<fffffe678d60b8>] worker_thread+0x208/0x328 [<fffffe678dc890>] kthread+0x100/0x1c0 [<fffffe678166c0>] ret_from_fork+0x10/0x20 Free the memory during completion and cleanup to fix the leak. Protect the mgmt_pending_tx idr_remove() operation in ath10k_wmi_tlv_op_cleanup_mgmt_tx_send() using ar->data_lock similar to other instances. Tested-on: WCN3990 hw1.0 SNOC WLAN.HL.2.0-01387-QCAHLSWMTPLZ-1

CVE ID	Description
CVE-2024-50237	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: do not pass a stopped vif to the driver in .get_txpower Avoid potentially crashing in the driver because of uninitialized private data
CVE-2024-50238	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb: fix NULL-deref on runtime suspend Commit 413db06c05e7 ("phy: qcom-qmp-usb: clean up probe initialisation") removed most users of the platform device driver data from the qcom-qmp-usb driver, but mistakenly also removed the initialisation despite the data still being used in the runtime PM callbacks. This bug was later reproduced when the driver was copied to create the qmp-usb driver. Restore the driver data initialisation at probe to avoid a NULL-pointer dereference on runtime suspend. Apparently no one uses runtime PM, which currently needs to be enabled manually through sysfs, with these drivers.
CVE-2024-50239	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb-legacy: fix NULL-deref on runtime suspend Commit 413db06c05e7 ("phy: qcom-qmp-usb: clean up probe initialisation") removed most users of the platform device driver data from the qcom-qmp-usb driver, but mistakenly also removed the initialisation despite the data still being used in the runtime PM callbacks. This bug was later reproduced when the driver was copied to create the qmp-usb-legacy driver. Restore the driver data initialisation at probe to avoid a NULL-pointer dereference on runtime suspend. Apparently no one uses runtime PM, which currently needs to be enabled manually through sysfs, with these drivers.
CVE-2024-50240	In the Linux kernel, the following vulnerability has been resolved: phy: qcom: qmp-usb: fix NULL-deref on runtime suspend Commit 413db06c05e7 ("phy: qcom-qmp-usb: clean up probe initialisation") removed most users of the platform device driver data, but mistakenly also removed the initialisation despite the data still being used in the runtime PM callbacks. Restore the driver data initialisation at probe to avoid a NULL-pointer dereference on runtime suspend. Apparently no one uses runtime PM, which currently needs to be enabled manually through sysfs, with this driver.
CVE-2024-50242	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Additional check in ntfs_file_release
CVE-2024-50243	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fix general protection fault in run_is_mapped_full Fixed deleating of a non-resident attribute in ntfs_create_inode() rollback.
CVE-2024-50244	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Additional check in ni_clear() Checking of NTFS_FLAGS_LOG_REPLAYING added to prevent access to uninitialized bitmap during replay process.
CVE-2024-50245	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fix possible deadlock in mi_read Mutex lock with another subclass used in ni_lock_dir().
CVE-2024-50246	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Add rough attr alloc_size check
CVE-2024-50247	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Check if more than chunk-size bytes are written A incorrectly formatted chunk may decompress into more than LZNT_CHUNK_SIZE bytes and an index out of bounds will occur in s_max_off.
CVE-2024-50248	In the Linux kernel, the following vulnerability has been resolved: ntfs3: Add bounds checking to mi_enum_attr() Added bounds checking to make sure that every attr don't stray beyond valid memory region.
CVE-2024-50249	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Make rmw_lock a raw_spin_lock The following BUG was triggered: ===== [BUG: Invalid wait context] 6.12.0-rc2-XXX #406 Not tainted ----- kworker/1:1/62 is trying to lock: ffffff8801593030 (&cpc_ptr->rmw_lock){+..-}{3:3}, at: cpc_write+0xcc/0x370 other info that might help us debug this: context-{5:5} 2 locks held by kworker/1:1/62: #0: ffffff897ef5ec98 (&rq->__lock){-.-}{2:2}, at: raw_spin_rq_lock_nested+0x2c/0x50 #1: ffffff880154e238 (&sg_policy->update_lock){....}{2:2}, at: sugov_update_shared+0x3c/0x280 stack backtrace: CPU: 1 UID: 0 PID: 62 Comm: kworker/1:1 Not tainted 6.12.0-rc2-g9654bd3e8806 #406 Workqueue: 0x0 (events) Call trace: dump_backtrace+0xa4/0x130 show_stack+0x20/0x38 dump_stack_lvl+0x90/0xd0 dump_stack+0x18/0x28 __lock_acquire+0x480/0x1ad8 lock_acquire+0x114/0x310 __raw_spin_lock+0x50/0x70 cpc_write+0xcc/0x370 cppc_set_perf+0xa0/0x3a8 cppc_cpufreq_fast_switch+0x40/0xc0 cpufreq_driver_fast_switch+0x4c/0x218 sugov_update_shared+0x234/0x280 update_load_avg+0x6ec/0x7b8 dequeue_entities+0x108/0x830 dequeue_task_fair+0x58/0x408 __schedule+0x4f0/0x1070 schedule+0x54/0x130 worker_thread+0xc0/0x2e8 kthread+0x130/0x148 ret_from_fork+0x10/0x20 sugov_update_shared() locks a raw_spinlock while cpc_write() locks a spinlock. To have a correct wait-type order, update rmw_lock to a raw spinlock and ensure that interrupts will be disabled on the CPU holding it. [rjw: Changelog edits]

CVE ID	Description
CVE-2024-50250	In the Linux kernel, the following vulnerability has been resolved: fsdax: dax_unshare_iter needs to copy entire blocks The code that copies data from srcmap to iomap in dax_unshare_iter is very very broken, which fosters recent fsx changes have exposed. If the pos and len passed to dax_file_unshare are not aligned to an fsblock boundary, the iter pos and length in the _iter function will reflect this unalignment. dax_iomap_direct_access always returns a pointer to the start of the kmapped fsdax page, even if its pos argument is in the middle of that page. This is catastrophic for data integrity when iter->pos is not aligned to a page, because daddr/saddr do not point to the same byte in the file as iter->pos. Hence we corrupt user data by copying it to the wrong place. If iter->pos + iomap_length() in the _iter function not aligned to a page, then we fail to copy a full block, and only partially populate the destination block. This is catastrophic for data confidentiality because we expose stale pmem contents. Fix both of these issues by aligning copy_pos/copy_len to a page boundary (remember, this is fsdax so 1 fsblock == 1 base page) so that we always copy full blocks. We're not done yet -- there's no call to invalidate_inode_pages2_range, so programs that have the file range mmap'd will continue accessing the old memory mapping after the file metadata updates have completed. Be careful with the return value -- if the unshare succeeds, we still need to return the number of bytes that the iomap iter thinks we're operating on.
CVE-2024-50251	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_payload: sanitize offset and length before calling skb_checksum() If access to offset + length is larger than the skbuff length, then skb_checksum() triggers BUG_ON(). skb_checksum() internally subtracts the length parameter while iterating over skbuff, BUG_ON(len) at the end of it checks that the expected length to be included in the checksum calculation is fully consumed.
CVE-2024-50252	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_ipip: Fix memory leak when changing remote IPv6 address The device stores IPv6 addresses that are used for encapsulation in linear memory that is managed by the driver. Changing the remote address of an ip6gre net device never worked properly, but since cited commit the following reproducer [1] would result in a warning [2] and a memory leak [3]. The problem is that the new remote address is never added by the driver to its hash table (and therefore the device) and the old address is never removed from it. Fix by programming the new address when the configuration of the ip6gre net device changes and removing the old one. If the address did not change, then the above would result in increasing the reference count of the address and then decreasing it. [1] # ip link add name bla up type ip6gre local 2001:db8:1::1 remote 2001:db8:2::1 tos inherit ttl inherit # ip link set dev bla type ip6gre remote 2001:db8:3::1 # ip link del dev bla # devlink dev reload pci/0000:01:00.0 [2] WARNING: CPU: 0 PID: 1682 at drivers/net/ethernet/mellanox/mlxsw/spectrum.c:3002 mlxsw_sp_ipv6_addr_put+0x140/0x1d0 Modules linked in: CPU: 0 UID: 0 PID: 1682 Comm: ip Not tainted 6.12.0-rc3-custom-g86b5b55bc835 #151 Hardware name: Nvidia SN5600/VMOD0013, BIOS 5.13 05/31/2023 RIP: 0010:mlxsw_sp_ipv6_addr_put+0x140/0x1d0 [...] Call Trace: <TASK> mlxsw_sp_router_netdevice_event+0x55f/0x1240 notifier_call_chain+0x5a/0xd0 call_netdevice_notifiers_info+0x39/0x90 unregister_netdevice_many_notify+0x63e/0x9d0 rtnl_dellink+0x16b/0x3a0 rtnetlink_rcv_msg+0x142/0x3f0 netlink_rcv_skb+0x50/0x100 netlink_unicast+0x242/0x390 netlink_sendmsg+0x1de/0x420 __sys_sendmsg+0x2bd/0x320 __sys_sendmsg+0x9a/0xe0 __sys_sendmsg+0x7a/0xd0 do_syscall_64+0x9e/0x1a0 entry_SYSCALL_64_after_hwframe+0x77/0x7f [3] unreferenced object 0xffff898081f597a0 (size 32): comm "ip", pid 1626, jiffies 4294719324 hex dump (first 32 bytes): 20 01 0d b8 00 02 00 00 00 00 00 00 00 00 01 21 49 61 83 80 89 ff ff 00 00 00 01 00 00 00 ! Ia..... backtrace (crc fd9be911): [<00000000df89c55d>] __kmalloc_cache_noprof+0x1da/0x260 [<00000000ff2a1ddb>] mlxsw_sp_ipv6_addr_kvdl_index_get+0x281/0x340 [<000000009ddd445d>] mlxsw_sp_router_netdevice_event+0x47b/0x1240 [<00000000743e7757>] notifier_call_chain+0x5a/0xd0 [<000000007c7b9e13>] call_netdevice_notifiers_info+0x39/0x90 [<000000002509645d>] register_netdevice+0x5f7/0x7a0 [<00000000c2e7d2a9>] ip6gre_newlink_common.isra.0+0x65/0x130 [<0000000087cd6d8d>] ip6gre_newlink+0x72/0x120 [<000000004df7c7cc>] rtnl_newlink+0x471/0xa20 [<0000000057ed632a>] rtnetlink_rcv_msg+0x142/0x3f0 [<0000000032e0d5b5>] netlink_rcv_skb+0x50/0x100 [<00000000908bca63>] netlink_unicast+0x242/0x390 [<00000000cdbe1c87>] netlink_sendmsg+0x1de/0x420 [<0000000011db153e>] __sys_sendmsg+0x2bd/0x320 [<000000003b6d53eb>] __sys_sendmsg+0x9a/0xe0 [<00000000cae27c62>] __sys_sendmsg+0x7a/0xd0

CVE ID	Description
CVE-2024-50255	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci: fix null-ptr-deref in hci_read_supported_codecs Fix __hci_cmd_sync_sk() to return not NULL for unknown opcodes. __hci_cmd_sync_sk() returns NULL if a command returns a status event. However, it also returns NULL where an opcode doesn't exist in the hci_cc table because hci_cmd_complete_evt() assumes status = skb->data[0] for unknown opcodes. This leads to null-ptr-deref in cmd_sync for HCI_OP_READ_LOCAL_CODECS as there is no hci_cc for HCI_OP_READ_LOCAL_CODECS, which always assumes status = skb->data[0]. KASAN: null-ptr-deref in range [0x0000000000000070-0x0000000000000077] CPU: 1 PID: 2000 Comm: kworker/u9:5 Not tainted 6.9.0-ga6bcb805883c-dirty #10 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Workqueue: hci7 hci_power_on RIP: 0010:hci_read_supported_codecs+0xb9/0x870 net/bluetooth/hci_codec.c:138 Code: 08 48 89 ef e8 b8 c1 8f fd 48 8b 75 00 e9 96 00 00 00 49 89 c6 48 ba 00 00 00 00 00 fc ff df 4c 8d 60 70 4c 89 e3 48 c1 eb 03 <0f> b6 04 13 84 c0 0f 85 82 06 00 00 41 83 3c 24 02 77 0a e8 bf 78 RSP: 0018:ffff888120bafac8 EFLAGS: 00010212 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffff8881173f0040 RDX: dffffc0000000000 RSI: ffffffff58496c0 RDI: ffff88810b9ad1e4 RBP: ffff88810b9ac000 R08: ffffffff77882a7 R09: 1fffffff4ef1054 R10: dffffc0000000000 R11: ffff88810b9ad1e4 R12: 0000000000000070 R13: 0000000000000000 R14: 0000000000000000 R15: ffff88810b9ac000 FS: 0000000000000000(0000) GS:ffff8881f6c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f6ddaa3439e CR3: 0000000139764003 CR4: 0000000000770ef0 PKRU: 55555554 Call Trace: <TASK> hci_read_local_codecs_sync net/bluetooth/hci_sync.c:4546 [inline] hci_init_stage_sync net/bluetooth/hci_sync.c:3441 [inline] hci_init4_sync net/bluetooth/hci_sync.c:4706 [inline] hci_init_sync net/bluetooth/hci_sync.c:4742 [inline] hci_dev_init_sync net/bluetooth/hci_sync.c:4912 [inline] hci_dev_open_sync+0x19a9/0x2d30 net/bluetooth/hci_sync.c:4994 hci_dev_do_open net/bluetooth/hci_core.c:483 [inline] hci_power_on+0x11e/0x560 net/bluetooth/hci_core.c:1015 process_one_work kernel/workqueue.c:3267 [inline] process_scheduled_works+0x8ef/0x14f0 kernel/workqueue.c:3348 worker_thread+0x91f/0xe50 kernel/workqueue.c:3429 kthread+0x2cb/0x360 kernel/kthread.c:388 ret_from_fork+0x4d/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244

CVE ID	Description
CVE-2024-50256	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_reject_ipv6: fix potential crash in nf_send_reset6() I got a syzbot report without a repro [1] crashing in nf_send_reset6() I think the issue is that dev->hard_header_len is zero, and we attempt later to push an Ethernet header. Use LL_MAX_HEADER, as other functions in net/ipv6/netfilter/nf_reject_ipv6.c. [1] skbuff: skb_under_panic: text:fffff8b1d008 len:74 put:14 head:ffff88803123aa00 data:ffff88803123a9f2 tail:0x3c end:0x140 dev:syz_tun kernel BUG at net/core/skbuff.c:206 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 0 UID: 0 PID: 7373 Comm: syz.1.568 Not tainted 6.12.0-rc2-syzkaller-00631-g6d858708d465 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010:skb_panic net/core/skbuff.c:206 [inline] RIP: 0010:skb_under_panic+0x14b/0x150 net/core/skbuff.c:216 Code: 0d 8d 48 c7 c6 60 a6 29 8e 48 8b 54 24 08 8b 0c 24 44 8b 44 24 04 4d 89 e9 50 41 54 41 57 41 56 e8 ba 30 38 02 48 83 c4 20 90 <0f> 0b 0f 1f 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 RSP: 0018:ffffc900045269b0 EFLAGS: 00010282 RAX: 0000000000000088 RBX: dffffc0000000000 RCX: cd66dacdc5d8e800 RDX: 0000000000000000 RSI: 0000000000000200 RDI: 0000000000000000 RBP: ffff88802d39a3d0 R08: ffffffff8174afec R09: 1ffff920008a4ccc R10: dffffc0000000000 R11: fffff520008a4ccd R12: 0000000000000140 R13: ffff88803123aa00 R14: ffff88803123a9f2 R15: 000000000000003c FS: 00007fdee5ff6c0(0000) GS: ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 000000005d322000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> skb_push+0xe5/0x100 net/core/skbuff.c:2636 eth_header+0x38/0x1f0 ethernet/eth.c:83 dev_hard_header include/linux/netdevice.h:3208 [inline] nf_send_reset6+0xce6/0x1270 net/ipv6/netfilter/nf_reject_ipv6.c:358 nft_reject_inet_eval+0x3b9/0x690 net/netfilter/nft_reject_inet.c:48 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x4ad/0x1da0 net/netfilter/nf_tables_core.c:288 nft_do_chain_inet+0x418/0x6b0 net/netfilter/nft_chain_filter.c:161 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xc3/0x220 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] NF_HOOK include/linux/netfilter.h:312 [inline] br_nf_pre_routing_ipv6+0x63e/0x770 net/bridge/br_netfilter_ipv6.c:184 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_bridge_pre net/bridge/br_input.c:277 [inline] br_handle_frame +0x9fd/0x1530 net/bridge/br_input.c:424 __netif_receive_skb_core+0x13e8/0x4570 net/core/dev.c:5562 __netif_receive_skb_one_core net/core/dev.c:5666 [inline] __netif_receive_skb+0x12f/0x650 net/core/dev.c:5781 netif_receive_skb_internal net/core/dev.c:5867 [inline] netif_receive_skb+0x1e8/0x890 net/core/dev.c:5926 tun_rx_batched+0x1b7/0x8f0 drivers/net/tun.c:1550 tun_get_user+0x3056/0x47e0 drivers/net/tun.c:2007 tun_chr_write_iter+0x10d/0x1f0 drivers/net/tun.c:2053 new_sync_write fs/read_write.c:590 [inline] vfs_write+0xa6d/0xc90 fs/read_write.c:683 ksys_write+0x183/0x2b0 fs/read_write.c:736 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fdee5ff6c0 Code: 89 54 24 18 48 89 74 24 10 89 7c 24 08 e8 c9 8d 02 00 48 8b 54 24 18 48 8b 74 24 10 41 89 c0 8b 7c 24 08 b8 01 00 00 0f 05 <48> 3d 00 f0 ff ff 77 31 44 89 c7 48 89 44 24 08 e8 1c 8e 02 00 48 RSP: 002b:00007fdee5ff000 EFLAGS: 00000293 ORIG_RAX: 0000000000000001 RAX: ffffffff8174afec RBX: 00007fdee5ff000 RCX: 00007fdee5ff000 RDX: 000000000000000e RSI: 0000000002000040 RDI: 00000000000000c8 RBP: 00007fdee5ff000 R08: 00000000 ---truncated---
CVE-2024-50257	In the Linux kernel, the following vulnerability has been resolved: netfilter: Fix use-after-free in get_info() ip6table_nat module unload has refcnt warning for UAF. call trace is: WARNING: CPU: 1 PID: 379 at kernel/module/main.c:853 module_put+0x6f/0x80 Modules linked in: ip6table_nat(-) CPU: 1 UID: 0 PID: 379 Comm: ip6tables Not tainted 6.12.0-rc4-00047-gc2ee9f594da8-dirty #205 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:module_put+0x6f/0x80 Call Trace: <TASK> get_info+0x128/0x180 do_ip6t_get_ctl+0x6a/0x430 nf_getsockopt+0x46/0x80 ipv6_getsockopt+0xb9/0x100 rawv6_getsockopt +0x42/0x190 do_sock_getsockopt+0xaa/0x180 __sys_getsockopt+0x70/0xc0 __x64_sys_getsockopt +0x20/0x30 do_syscall_64+0xa2/0x1a0 entry_SYSCALL_64_after_hwframe+0x77/0x7f Concurrent execution of module unload and get_info() triggered the warning. The root cause is as follows: cpu0 cpu1 module_nat //mod->state = MODULE_STATE_GOING ip6table_nat_exit xt_unregister_template kfree(t) //removed from tpl_list getinfo() t = xt_find_table_lock list_for_each_entry(tmpl, &xt_templates[af]...) if (strcmp(tmpl->name, name)) continue; //table not found try_module_get list_for_each_entry(t, &xt_net->tables[af]...) return t; //not get refcnt module_put(t->me) //uaf unregister_pernet_subsys //remove table from xt_net list While xt_table module was going away and has been removed from xt_templates list, we couldnt get refcnt of xt_table->me. Check module in xt_net->tables list re-traversal to fix it.

CVE ID	Description
CVE-2024-50258	In the Linux kernel, the following vulnerability has been resolved: net: fix crash when config small gso_max_size/gso_ipv4_max_size Config a small gso_max_size/gso_ipv4_max_size will lead to an underflow in sk_dst_gso_max_size(), which may trigger a BUG_ON crash, because sk->sk_gso_max_size would be much bigger than device limits. Call Trace: tcp_write_xmit tso_segs = tcp_init_tso_segs(skb, mss_now); tcp_set_skb_tso_segs tcp_skb_pcount_set // skb->len = 524288, mss_now = 8 // u16 tso_segs = 524288/8 = 65535 -> 0 tso_segs = DIV_ROUND_UP(skb->len, mss_now) BUG_ON(!tso_segs) Add check for the minimum value of gso_max_size and gso_ipv4_max_size.
CVE-2024-50259	In the Linux kernel, the following vulnerability has been resolved: netdevsim: Add trailing zero to terminate the string in nsim_nexthop_bucket_activity_write() This was found by a static analyzer. We should not forget the trailing zero after copy_from_user() if we will further do some string operations, sscanf() in this case. Adding a trailing zero will ensure that the function performs properly.
CVE-2024-50261	In the Linux kernel, the following vulnerability has been resolved: macsec: Fix use-after-free while sending the offloading packet KASAN reports the following UAF. The metadata_dst, which is used to store the SCI value for macsec offload, is already freed by metadata_dst_free() in macsec_free_netdev(), while driver still use it for sending the packet. To fix this issue, dst_release() is used instead to release metadata_dst. So it is not freed instantly in macsec_free_netdev() if still referenced by skb. BUG: KASAN: slab-use-after-free in mlx5e_xmit+0x1e8f/0x4190 [mlx5_core] Read of size 2 at addr ffff88813e42e038 by task kworker/7:2/714 [...] Workqueue: mld mld_ifc_work Call Trace: <TASK> dump_stack_lvl+0x51/0x60 print_report+0xc1/0x600 kasan_report+0xab/0xe0 mlx5e_xmit+0x1e8f/0x4190 [mlx5_core] dev_hard_start_xmit +0x120/0x530 sch_direct_xmit+0x149/0x11e0 __qdisc_run+0x3ad/0x1730 __dev_queue_xmit +0x1196/0x2ed0 vlan_dev_hard_start_xmit+0x32e/0x510 [8021q] dev_hard_start_xmit+0x120/0x530 __dev_queue_xmit+0x14a7/0x2ed0 macsec_start_xmit+0x13e9/0x2340 dev_hard_start_xmit +0x120/0x530 __dev_queue_xmit+0x14a7/0x2ed0 ip6_finish_output2+0x923/0x1a70 ip6_finish_output +0x2d7/0x970 ip6_output+0x1ce/0x3a0 NF_HOOK.constprop.0+0x15f/0x190 mld_sendpack +0x59a/0xbd0 mld_ifc_work+0x48a/0xa80 process_one_work+0x5aa/0xe50 worker_thread +0x79c/0x1290 kthread+0x28f/0x350 ret_from_fork+0x2d/0x70 ret_from_fork_asm+0x11/0x20 </TASK> Allocated by task 3922: kasan_save_stack+0x20/0x40 kasan_save_track+0x10/0x30 __kasan_kmalloc+0x77/0x90 __kmalloc_noprof+0x188/0x400 metadata_dst_alloc+0x1f/0x4e0 macsec_newlink+0x914/0x1410 __rtnl_newlink+0xe08/0x15b0 rtnl_newlink+0x5f/0x90 rtnetlink_rcv_msg+0x667/0xa80 netlink_rcv_skb+0x12c/0x360 netlink_unicast+0x551/0x770 netlink_sendmsg+0x72d/0xbd0 __sock_sendmsg+0xc5/0x190 __sys_sendmsg+0x52e/0x6a0 __sys_sendmsg+0xeb/0x170 __sys_sendmsg+0xb5/0x140 do_syscall_64+0x4c/0x100 entry_SYSCALL_64_after_hwframe+0x4b/0x53 Freed by task 4011: kasan_save_stack+0x20/0x40 kasan_save_track+0x10/0x30 kasan_save_free_info+0x37/0x50 poison_slab_object+0x10c/0x190 __kasan_slab_free+0x11/0x30 kfree+0xe0/0x290 macsec_free_netdev+0x3f/0x140 netdev_run_todo +0x450/0xc70 rtnetlink_rcv_msg+0x66f/0xa80 netlink_rcv_skb+0x12c/0x360 netlink_unicast +0x551/0x770 netlink_sendmsg+0x72d/0xbd0 __sock_sendmsg+0xc5/0x190 __sys_sendmsg +0x52e/0x6a0 __sys_sendmsg+0xeb/0x170 __sys_sendmsg+0xb5/0x140 do_syscall_64+0x4c/0x100 entry_SYSCALL_64_after_hwframe+0x4b/0x53
CVE-2024-50262	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix out-of-bounds write in trie_get_next_key() trie_get_next_key() allocates a node stack with size trie->max_prefixlen, while it writes (trie->max_prefixlen + 1) nodes to the stack when it has full paths from the root to leaves. For example, consider a trie with max_prefixlen is 8, and the nodes with key 0x00/0, 0x00/1, 0x00/2, ... 0x00/8 inserted. Subsequent calls to trie_get_next_key with _key with .prefixlen = 8 make 9 nodes be written on the node stack with size 8.
CVE-2024-50263	In the Linux kernel, the following vulnerability has been resolved: fork: only invoke khugepaged, ksm hooks if no error There is no reason to invoke these hooks early against an mm that is in an incomplete state. The change in commit d24062914837 ("fork: use __mt_dup() to duplicate maple tree in dup_mmap()") makes this more pertinent as we may be in a state where entries in the maple tree are not yet consistent. Their placement early in dup_mmap() only appears to have been meaningful for early error checking, and since functionally it'd require a very small allocation to fail (in practice 'too small to fail') that'd only occur in the most dire circumstances, meaning the fork would fail or be OOM'd in any case. Since both khugepaged and KSM tracking are there to provide optimisations to memory performance rather than critical functionality, it doesn't really matter all that much if, under such dire memory pressure, we fail to register an mm with these. As a result, we follow the example of commit d2081b2bf819 ("mm: khugepaged: make khugepaged_enter() void function") and make ksm_fork() a void function also. We only expose the mm to these functions once we are done with them and only if no error occurred in the fork operation.

CVE ID	Description
CVE-2024-50265	In the Linux kernel, the following vulnerability has been resolved: ocfs2: remove entry once instead of null-ptr-dereference in ocfs2_xa_remove() Syzkaller is able to provoke null-ptr-dereference in ocfs2_xa_remove(): [57.319872] (a.out,1161,7):ocfs2_xa_remove:2028 ERROR: status = -12 [57.320420] (a.out,1161,7):ocfs2_xa_cleanup_value_truncate:1999 ERROR: Partial truncate while removing xattr.overlay.upper. Leaking 1 clusters and removing the entry [57.321727] BUG: kernel NULL pointer dereference, address: 0000000000000004 [...] [57.325727] RIP: 0010:ocfs2_xa_block_wipe_namevalue+0x2a/0xc0 [...] [57.331328] Call Trace: [57.331477] <TASK> [...] [57.333511] ? do_user_addr_fault+0x3e5/0x740 [57.333778] ? exc_page_fault+0x70/0x170 [57.334016] ? asm_exc_page_fault+0x2b/0x30 [57.334263] ? __pfx_ocfs2_xa_block_wipe_namevalue+0x10/0x10 [57.334596] ? ocfs2_xa_block_wipe_namevalue+0x2a/0xc0 [57.334913] ocfs2_xa_remove_entry+0x23/0xc0 [57.335164] ocfs2_xa_set+0x704/0xc0 [57.335381] ? _raw_spin_unlock+0x1a/0x40 [57.335620] ? ocfs2_inode_cache_unlock+0x16/0x20 [57.335915] ? trace_preempt_on+0x1e/0x70 [57.336153] ? start_this_handle+0x16c/0x500 [57.336410] ? preempt_count_sub+0x50/0x80 [57.336656] ? _raw_read_unlock+0x20/0x40 [57.336906] ? start_this_handle+0x16c/0x500 [57.337162] ocfs2_xattr_block_set+0xa6/0x1e0 [57.337424] __ocfs2_xattr_set_handle+0x1fd/0x5d0 [57.337706] ? ocfs2_start_trans+0x13d/0x290 [57.337971] ocfs2_xattr_set+0xb13/0xfb0 [57.338207] ? dput+0x46/0x1c0 [57.338393] ocfs2_xattr_trusted_set+0x28/0x30 [57.338665] ? ocfs2_xattr_trusted_set+0x28/0x30 [57.338948] __vfs_remove_xattr+0x92/0xc0 [57.339182] __vfs_remove_xattr_locked+0xd5/0x190 [57.339456] ? preempt_count_sub+0x50/0x80 [57.339705] vfs_remove_xattr+0x5f/0x100 [...] Reproducer uses faultinject facility to fail ocfs2_xa_remove() -> ocfs2_xa_value_truncate() with -ENOMEM. In this case the comment mentions that we can return 0 if ocfs2_xa_cleanup_value_truncate() is going to wipe the entry anyway. But the following 'rc' check is wrong and execution flow do 'ocfs2_xa_remove_entry(loc);' twice: * 1st: in ocfs2_xa_cleanup_value_truncate(); * 2nd: returning back to ocfs2_xa_remove() instead of going to 'out'. Fix this by skipping the 2nd removal of the same entry and making syzkaller repro happy.
CVE-2024-50267	In the Linux kernel, the following vulnerability has been resolved: USB: serial: io_edgeport: fix use after free in debug printk The "dev_dbg(&urb->dev->dev, ...)" which happens after usb_free_urb(urb) is a use after free of the "urb" pointer. Store the "dev" pointer at the start of the function to avoid this issue.
CVE-2024-50268	In the Linux kernel, the following vulnerability has been resolved: usb: typec: fix potential out of bounds in ucsi_ccg_update_set_new_cam_cmd() The "*cmd" variable can be controlled by the user via debugfs. That means "new_cam" can be as high as 255 while the size of the uc->updated[] array is UCSI_MAX_ALTMODES (30). The call tree is: ucsi_cmd() // val comes from simple_attr_write_xsigned() -> ucsi_send_command() -> ucsi_send_command_common() -> ucsi_run_command() // calls ucsi->ops->sync_control() -> ucsi_ccg_sync_control()
CVE-2024-50269	In the Linux kernel, the following vulnerability has been resolved: usb: musb: sunxi: Fix accessing an released usb phy Commit 6ed05c68cbca ("usb: musb: sunxi: Explicitly release USB PHY on exit") will cause that usb phy @glue->xceiv is accessed after released. 1) register platform driver @sunxi_musb_driver // get the usb phy @glue->xceiv sunxi_musb_probe() -> devm_usb_get_phy(). 2) register and unregister platform driver @musb_driver_musb_probe() -> sunxi_musb_init() use the phy here //the phy is released here musb_remove() -> sunxi_musb_exit() -> devm_usb_put_phy() 3) register @musb_driver again musb_probe() -> sunxi_musb_init() use the phy here but the phy has been released at 2). ... Fixed by reverting the commit, namely, removing devm_usb_put_phy() from sunxi_musb_exit().
CVE-2024-50270	In the Linux kernel, the following vulnerability has been resolved: mm/damon/core: avoid overflow in damon_feed_loop_next_input() damon_feed_loop_next_input() is inefficient and fragile to overflows. Specifically, 'score_goal_diff_bp' calculation can overflow when 'score' is high. The calculation is actually unnecessary at all because 'goal' is a constant of value 10,000. Calculation of 'compensation' is again fragile to overflow. Final calculation of return value for under-achieving case is again fragile to overflow when the current score is under-achieving the target. Add two corner cases handling at the beginning of the function to make the body easier to read, and rewrite the body of the function to avoid overflows and the unnecessary bp value calculation.
CVE-2024-50271	In the Linux kernel, the following vulnerability has been resolved: signal: restore the override_rlimit logic Prior to commit d64696905554 ("Reimplement RLIMIT_SIGPENDING on top of ucounts") UCOUNT_RLIMIT_SIGPENDING rlimit was not enforced for a class of signals. However now it's enforced unconditionally, even if override_rlimit is set. This behavior change caused production issues. For example, if the limit is reached and a process receives a SIGSEGV signal, sigqueue_alloc fails to allocate the necessary resources for the signal delivery, preventing the signal from being delivered with siginfo. This prevents the process from correctly identifying the fault address and handling the error. From the user-space perspective, applications are unaware that the limit has been reached and that the siginfo is effectively 'corrupted'. This can lead to unpredictable behavior and crashes, as we observed with java applications. Fix this by passing override_rlimit into inc_rlimit_get_ucounts() and skip the comparison to max there if override_rlimit is set. This effectively restores the old behavior.

CVE ID	Description
CVE-2024-50272	In the Linux kernel, the following vulnerability has been resolved: filemap: Fix bounds checking in filemap_read() If the caller supplies an iocb->ki_pos value that is close to the filesystem upper limit, and an iterator with a count that causes us to overflow that limit, then filemap_read() enters an infinite loop. This behaviour was discovered when testing xfstests generic/525 with the "localio" optimisation for loopback NFS mounts.
CVE-2024-50273	In the Linux kernel, the following vulnerability has been resolved: btrfs: reinitialize delayed ref list after deleting it from the list At insert_delayed_ref() if we need to update the action of an existing ref to BTRFS_DROP_DELAYED_REF, we delete the ref from its ref head's ref_add_list using list_del(), which leaves the ref's add_list member not reinitialized, as list_del() sets the next and prev members of the list to LIST_POISON1 and LIST_POISON2, respectively. If later we end up calling drop_delayed_ref() against the ref, which can happen during merging or when destroying delayed refs due to a transaction abort, we can trigger a crash since at drop_delayed_ref() we call list_empty() against the ref's add_list, which returns false since the list was not reinitialized after the list_del() and as a consequence we call list_del() again at drop_delayed_ref(). This results in an invalid list access since the next and prev members are set to poison pointers, resulting in a splat if CONFIG_LIST_HARDENED and CONFIG_DEBUG_LIST are set or invalid poison pointer dereferences otherwise. So fix this by deleting from the list with list_del_init() instead.
CVE-2024-50274	In the Linux kernel, the following vulnerability has been resolved: idpf: avoid vport access in idpf_get_link_ksettings When the device control plane is removed or the platform running device control plane is rebooted, a reset is detected on the driver. On driver reset, it releases the resources and waits for the reset to complete. If the reset fails, it takes the error path and releases the vport lock. At this time if the monitoring tools tries to access link settings, it call traces for accessing released vport pointer. To avoid it, move link_speed_mbps to netdev_priv structure which removes the dependency on vport pointer and the vport lock in idpf_get_link_ksettings. Also use netif_carrier_ok() to check the link status and adjust the offsetof to use link_up instead of link_speed_mbps.
CVE-2024-50275	In the Linux kernel, the following vulnerability has been resolved: arm64/sve: Discard stale CPU state when handling SVE traps The logic for handling SVE traps manipulates saved FPSIMD/SVE state incorrectly, and a race with preemption can result in a task having TIF_SVE set and TIF_FOREIGN_FPSTATE clear even though the live CPU state is stale (e.g. with SVE traps enabled). This has been observed to result in warnings from do_sve_acc() where SVE traps are not expected while TIF_SVE is set: if (test_and_set_thread_flag(TIF_SVE)) WARN_ON(1); /* SVE access shouldn't have trapped */ Warnings of this form have been reported intermittently, e.g. https://lore.kernel.org/linux-arm-kernel/CA+G9fYtEGe_DhY2Ms7+L7NKsLYUomGsgqpdBj+QwDLLeSg=JhGg@mail.gmail.com/ https://lore.kernel.org/linux-arm-kernel/0000000000000511e9a060ce5a45c@googleg.com/ The race can occur when the SVE trap handler is preempted before and after manipulating the saved FPSIMD/SVE state, starting and ending on the same CPU, e.g. void do_sve_acc(unsigned long esr, struct pt_regs *regs) { // Trap on CPU 0 with TIF_SVE clear, SVE traps enabled // task->fpsimd_cpu is 0. // per_cpu_ptr(&fpsimd_last_state, 0) is task. ... // Preempted; migrated from CPU 0 to CPU 1. // TIF_FOREIGN_FPSTATE is set. get_cpu_fpsimd_context(); if (test_and_set_thread_flag(TIF_SVE)) WARN_ON(1); /* SVE access shouldn't have trapped */ sve_init_regs() { if (!test_thread_flag(TIF_FOREIGN_FPSTATE)) { ... } else { fpsimd_to_sve(current); current->thread.fp_type = FP_STATE_SVE; } } put_cpu_fpsimd_context(); // Preempted; migrated from CPU 1 to CPU 0. // task->fpsimd_cpu is still 0 // If per_cpu_ptr(&fpsimd_last_state, 0) is still task then: // - Stale HW state is reused (with SVE traps enabled) // - TIF_FOREIGN_FPSTATE is cleared // - A return to userspace skips HW state restore } Fix the case where the state is not live and TIF_FOREIGN_FPSTATE is set by calling fpsimd_flush_task_state() to detach from the saved CPU state. This ensures that a subsequent context switch will not reuse the stale CPU state, and will instead set TIF_FOREIGN_FPSTATE, forcing the new state to be reloaded from memory prior to a return to userspace.

[illegible]

CVE ID	Description
CVE-2024-50280	In the Linux kernel, the following vulnerability has been resolved: dm cache: fix flushing uninitialized delayed_work on cache_ctr error An unexpected WARN_ON from flush_work() may occur when cache creation fails, caused by destroying the uninitialized delayed_work waker in the error path of cache_create(). For example, the warning appears on the superbblock checksum error. Reproduce steps: dmsetup create cmeta --table "0 8192 linear /dev/sdc 0" dmsetup create cdata --table "0 65536 linear /dev/sdc 8192" dmsetup create corig --table "0 524288 linear /dev/sdc 262144" dd if=/dev/urandom of=/dev/mapper/cmeta bs=4k count=1 oflag=direct dmsetup create cache --table "0 524288 cache /dev/mapper/cmeta \ /dev/mapper/cdata /dev/mapper/corig 128 2 metadata2 writethrough smq 0" Kernel logs: (snip) WARNING: CPU: 0 PID: 84 at kernel/workqueue.c:4178 __flush_work+0x5d4/0x890 Fix by pulling out the cancel_delayed_work_sync() from the constructor's error path. This patch doesn't affect the use-after-free fix for concurrent dm_resume and dm_destroy (commit 6a459d8edbdb ("dm cache: Fix UAF in destroy()")) as cache_dtr is not changed.
CVE-2024-50282	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: add missing size check in amdgpu_debugfs_gprwave_read() Avoid a possible buffer overflow if size is larger than 4K. (cherry picked from commit f5d873f5825b40d886d03bd2aede91d4cf002434)
CVE-2024-50283	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-use-after-free in smb3_preauth_hash_rsp ksmbd_user_session_put should be called under smb3_preauth_hash_rsp(). It will avoid freeing session before calling smb3_preauth_hash_rsp().
CVE-2024-50284	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Fix the missing xa_store error check xa_store() can fail, it return xa_err(-EINVAL) if the entry cannot be stored in an XArray, or xa_err(-ENOMEM) if memory allocation failed, so check error for xa_store() to fix it.
CVE-2024-50285	In the Linux kernel, the following vulnerability has been resolved: ksmbd: check outstanding simultaneous SMB operations If Client send simultaneous SMB operations to ksmbd, It exhausts too much memory through the "ksmbd_work_cache". It will cause OOM issue. ksmbd has a credit mechanism but it can't handle this problem. This patch add the check if it exceeds max credits to prevent this problem by assuming that one smb request consumes at least one credit.
CVE-2024-50286	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix slab-use-after-free in ksmbd_smb2_session_create There is a race condition between ksmbd_smb2_session_create and ksmbd_expire_session. This patch add missing sessions_table_lock while adding/deleting session from global session table.
CVE-2024-50287	In the Linux kernel, the following vulnerability has been resolved: media: v4l2-tpg: prevent the risk of a division by zero As reported by Coverity, the logic at tpg_precalculate_line() blindly rescales the buffer even when scaled_width is equal to zero. If this ever happens, this will cause a division by zero. Instead, add a WARN_ON_ONCE() to trigger such cases and return without doing any precalculation.
CVE-2024-50288	In the Linux kernel, the following vulnerability has been resolved: media: vivid: fix buffer overwrite when using > 32 buffers The maximum number of buffers that can be requested was increased to 64 for the video capture queue. But video capture used a must_blank array that was still sized for 32 (VIDEO_MAX_FRAME). This caused an out-of-bounds write when using buffer indices >= 32. Create a new define MAX_VID_CAP_BUFFERS that is used to access the must_blank array and set max_num_buffers for the video capture queue. This solves a crash reported by: https://bugzilla.kernel.org/show_bug.cgi?id=219258
CVE-2024-50289	In the Linux kernel, the following vulnerability has been resolved: media: av7110: fix a spectre vulnerability As warned by smatch: drivers/staging/media/av7110/av7110_ca.c:270 dvb_ca_ioctl() warn: potential spectre issue 'av7110->ci_slot' [w] (local cap) There is a spectre-related vulnerability at the code. Fix it.
CVE-2024-50290	In the Linux kernel, the following vulnerability has been resolved: media: cx24116: prevent overflows on SNR calculus as reported by Coverity, if reading SNR registers fail, a negative number will be returned, causing an underflow when reading SNR registers. Prevent that.
CVE-2024-50291	In the Linux kernel, the following vulnerability has been resolved: media: dvb-core: add missing buffer index check dvb_vb2_expbuf() didn't check if the given buffer index was for a valid buffer. Add this check.
CVE-2024-50292	In the Linux kernel, the following vulnerability has been resolved: ASoC: stm32: spdifrx: fix dma channel release in stm32_spdifrx_remove In case of error when requesting ctrl_chan DMA channel, ctrl_chan is not null. So the release of the dma channel leads to the following issue: [4.879000] st,stm32-spdifrx 500d0000.audio-controller: dma_request_slave_channel error -19 [4.888975] Unable to handle kernel NULL pointer dereference at virtual address 000000000000003d [...] [5.096577] Call trace: [5.099099] dma_release_channel+0x24/0x100 [5.103235] stm32_spdifrx_remove+0x24/0x60 [snd_soc_stm32_spdifrx] [5.109494] stm32_spdifrx_probe+0x320/0x4c4 [snd_soc_stm32_spdifrx] To avoid this issue, release channel only if the pointer is valid.

CVE ID	Description
CVE-2024-50294	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix missing locking causing hanging calls If a call gets aborted (e.g. because kafs saw a signal) between it being queued for connection and the I/O thread picking up the call, the abort will be prioritised over the connection and it will be removed from local->new_client_calls by rxrpc_disconnect_client_call() without a lock being held. This may cause other calls on the list to disappear if a race occurs. Fix this by taking the client_call_lock when removing a call from whatever list its ->wait_link happens to be on.
CVE-2024-50295	In the Linux kernel, the following vulnerability has been resolved: net: arc: fix the device for dma_map_single/dma_unmap_single The ndev->dev and pdev->dev aren't the same device, use ndev->dev.parent which has dma_mask, ndev->dev.parent is just pdev->dev. Or it would cause the following issue: [39.933526] -----[cut here]----- [39.938414] WARNING: CPU: 1 PID: 501 at kernel/dma/mapping.c:149 dma_map_page_attrs+0x90/0x1f8
CVE-2024-50296	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when uninstalling driver When the driver is uninstalled and the VF is disabled concurrently, a kernel crash occurs. The reason is that the two actions call function pci_disable_sriov(). The num_VFs is checked to determine whether to release the corresponding resources. During the second calling, num_VFs is not 0 and the resource release function is called. However, the corresponding resource has been released during the first invoking. Therefore, the problem occurs: [15277.839633][T50670] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000020 ... [15278.131557] [T50670] Call trace: [15278.134686][T50670] klist_put+0x28/0x12c [15278.138682][T50670] klist_del+0x14/0x20 [15278.142592][T50670] device_del+0xbc/0x3c0 [15278.146676][T50670] pci_remove_bus_device+0x84/0x120 [15278.151714][T50670] pci_stop_and_remove_bus_device+0x6c/0x80 [15278.157447][T50670] pci_iov_remove_virtfn+0xb4/0x12c [15278.162485][T50670] sriov_disable+0x50/0x11c [15278.166829][T50670] pci_disable_sriov+0x24/0x30 [15278.171433] [T50670] hnae3_unregister_ae_algo_prepare+0x60/0x90 [hnae3] [15278.178039][T50670] hclge_exit+0x28/0xd0 [hclge] [15278.182730][T50670] __se_sys_delete_module.isra.0+0x164/0x230 [15278.188550][T50670] __arm64_sys_delete_module+0x1c/0x30 [15278.193848][T50670] invoke_syscall+0x50/0x11c [15278.198278][T50670] el0_svc_common.constprop.0+0x158/0x164 [15278.203837][T50670] do_el0_svc+0x34/0xcc [15278.207834][T50670] el0_svc+0x20/0x30 For details, see the following figure. rmmmod hclge disable VFs ----- hclge_exit() sriov_numvfs_store() ... device_lock() pci_disable_sriov() hns3_pci_sriov_configure() pci_disable_sriov() sriov_disable() sriov_disable() if !num_VFs : if !num_VFs : return; return; sriov_del_vfs() sriov_del_vfs() ... klist_put() klist_put() ... num_VFs = 0; num_VFs = 0; device_unlock(); In this patch, when driver is removing, we get the device_lock() to protect num_VFs, just like sriov_numvfs_store().
CVE-2024-50297	In the Linux kernel, the following vulnerability has been resolved: net: xilinx: axienet: Enqueue Tx packets in dql before dmaengine starts Enqueue packets in dql after dma engine starts causes race condition. Tx transfer starts once dma engine is started and may execute dql dequeue in completion before it gets queued. It results in following kernel crash while running iperf stress test: kernel BUG at lib/dynamic_queue_limits.c:99! <snip> Internal error: Oops - BUG: 00000000f2000800 [#1] SMP pc : dql_completed+0x238/0x248 lr : dql_completed+0x3c/0x248 Call trace: dql_completed+0x238/0x248 axienet_dma_tx_cb+0xa0/0x170 xilinx_dma_do_tasklet+0xdc/0x290 tasklet_action_common+0xf8/0x11c tasklet_action+0x30/0x3c handle_softirqs+0xf8/0x230 <snip> Start dmaengine after enqueue in dql fixes the crash.
CVE-2024-50298	In the Linux kernel, the following vulnerability has been resolved: net: enetc: allocate vf_state during PF probes In the previous implementation, vf_state is allocated memory only when VF is enabled. However, net_device_ops::ndo_set_vf_mac() may be called before VF is enabled to configure the MAC address of VF. If this is the case, enetc_pf_set_vf_mac() will access vf_state, resulting in access to a null pointer. The simplified error log is as follows. root@ls1028ardb:~# ip link set eno0 vf 1 mac 00:0c:e7:66:77:89 [173.543315] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000004 [173.637254] pc : enetc_pf_set_vf_mac+0x3c/0x80 Message from sy [173.641973] lr : do_setlink+0x4a8/0xec8 [173.732292] Call trace: [173.734740] enetc_pf_set_vf_mac+0x3c/0x80 [173.738847] __rtnl_newlink+0x530/0x89c [173.742692] rtnl_newlink+0x50/0x7c [173.746189] rtnetlink_rcv_msg+0x128/0x390 [173.750298] netlink_rcv_skb+0x60/0x130 [173.754145] rtnetlink_rcv+0x18/0x24 [173.757731] netlink_unicast+0x318/0x380 [173.761665] netlink_sendmsg+0x17c/0x3c8
CVE-2024-50299	In the Linux kernel, the following vulnerability has been resolved: sctp: properly validate chunk size in sctp_sf_ootb() A size validation fix similar to that in Commit 50619dbf8db7 ("sctp: add size validation when walking chunks") is also required in sctp_sf_ootb() to address a crash reported by syzbot: BUG: KMSAN: uninit-value in sctp_sf_ootb+0x7f5/0xce0 net/sctp/sm_statefuns.c:3712 sctp_sf_ootb+0x7f5/0xce0 net/sctp/sm_statefuns.c:3712 sctp_do_sm+0x181/0x93d0 net/sctp/sm_sideeffect.c:1166 sctp_endpoint_bh_rcv+0xc38/0xf90 net/sctp/endpointola.c:407 sctp_inq_push+0x2ef/0x380 net/sctp/inqueue.c:88 sctp_rcv+0x3831/0x3b20 net/sctp/input.c:243 sctp4_rcv+0x42/0x50 net/sctp/protocol.c:1159 ip_protocol_deliver_rcu+0xb51/0x13d0 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x336/0x500 net/ipv4/ip_input.c:233

CVE ID	Description
CVE-2024-50300	In the Linux kernel, the following vulnerability has been resolved: regulator: rtq2208: Fix uninitialized use of regulator_config Fix rtq2208 driver uninitialized use to cause kernel error.
CVE-2024-50301	In the Linux kernel, the following vulnerability has been resolved: security/keys: fix slab-out-of-bounds in key_task_permission KASAN reports an out of bounds read: BUG: KASAN: slab-out-of-bounds in __kuid_val include/linux/uidgid.h:36 BUG: KASAN: slab-out-of-bounds in uid_eq include/linux/uidgid.h:63 [inline] BUG: KASAN: slab-out-of-bounds in key_task_permission +0x394/0x410 security/keys/permission.c:54 Read of size 4 at addr ffff88813c3ab618 by task stress-ng/4362 CPU: 2 PID: 4362 Comm: stress-ng Not tainted 5.10.0-14930-gafbfdd6c3ede #15 Call Trace: __dump_stack lib/dump_stack.c:82 [inline] dump_stack+0x107/0x167 lib/dump_stack.c:123 print_address_description.constprop.0+0x19/0x170 mm/kasan/report.c:400 __kasan_report.cold +0x6c/0x84 mm/kasan/report.c:560 kasan_report+0x3a/0x50 mm/kasan/report.c:585 __kuid_val include/linux/uidgid.h:36 [inline] uid_eq include/linux/uidgid.h:63 [inline] key_task_permission +0x394/0x410 security/keys/permission.c:54 search_nested_keyrings+0x90e/0xe90 security/keys/keyring.c:793 This issue was also reported by syzbot. It can be reproduced by following these steps(more details [1]): 1. Obtain more than 32 inputs that have similar hashes, which ends with the pattern '0xxxxxxx6'. 2. Reboot and add the keys obtained in step 1. The reproducer demonstrates how this issue happened: 1. In the search_nested_keyrings function, when it iterates through the slots in a node(below tag ascend_to_node), if the slot pointer is meta and node->back_pointer != NULL(it means a root), it will proceed to descend_to_node. However, there is an exception. If node is the root, and one of the slots points to a shortcut, it will be treated as a keyring. 2. Whether the ptr is keyring decided by keyring_ptr_is_keyring function. However, KEYRING_PTR_SUBTYPE is 0x2UL, the same as ASSOC_ARRAY_PTR_SUBTYPE_MASK. 3. When 32 keys with the similar hashes are added to the tree, the ROOT has keys with hashes that are not similar (e.g. slot 0) and it splits NODE A without using a shortcut. When NODE A is filled with keys that all hashes are xxe6, the keys are similar, NODE A will split with a shortcut. Finally, it forms the tree as shown below, where slot 6 points to a shortcut. NODE A +----->+----+ ROOT 0 xxe6 +----+ +---+ xxxx 0 shortcut : : xxe6 +----+ +----+ xxe6 : : xxe6 +----+ +----+ 6 ---+ : : xxe6 +----+ +----+ xxe6 : : f xxe6 +----+ +----+ xxe6 f +----+ 4. As mentioned above, If a slot(slot 6) of the root points to a shortcut, it may be mistakenly transferred to a key*, leading to a read out-of-bounds read. To fix this issue, one should jump to descend_to_node if the ptr is a shortcut, regardless of whether the node is root or not. [1] https://lore.kernel.org/linux-kernel/1cfa878e-8c7b-4570-8606-21daf5e13ce7@huaweicloud.com/ [jarkko: tweaked the commit message a bit to have an appropriate closes tag.]
CVE-2024-50302	In the Linux kernel, the following vulnerability has been resolved: HID: core: zero-initialize the report buffer Since the report buffer is used by all kinds of drivers in various ways, let's zero-initialize it during allocation to make sure that it can't be ever used to leak kernel memory via specially-crafted report.

CVE ID	Description
CVE-2024-50303	In the Linux kernel, the following vulnerability has been resolved: resource,kexec: walk_system_ram_res_rev must retain resource flags walk_system_ram_res_rev() erroneously discards resource flags when passing the information to the callback. This causes systems with IORESOURCE_SYSRAM_DRIVER_MANAGED memory to have these resources selected during kexec to store kexec buffers if that memory happens to be at placed above normal system ram. This leads to undefined behavior after reboot. If the kexec buffer is never touched, nothing happens. If the kexec buffer is touched, it could lead to a crash (like below) or undefined behavior. Tested on a system with CXL memory expanders with driver managed memory, TPM enabled, and CONFIG_IMA_KEXEC=y. Adding printk's showed the flags were being discarded and as a result the check for IORESOURCE_SYSRAM_DRIVER_MANAGED passes. find_next_iomem_res: name(System RAM (kmem)) start(10000000000) end(1034ffffff) flags(83000200) locate_mem_hole_top_down: start(10000000000) end(1034ffffff) flags(0) [...] BUG: unable to handle page fault for address: ffff89834ffff000 [...] #PF: supervisor read access in kernel mode [...] #PF: error_code(0x0000) - not-present page [...] PGD c04c8bf067 P4D c04c8bf067 PUD c04c8be067 PMD 0 [...] Oops: 0000 [#1] SMP [...] RIP: 0010:ima_restore_measurement_list+0x95/0x4b0 [...] RSP: 0018:ffffc900000d3ba0 EFLAGS: 00010286 [...] RAX: 0000000000001000 RBX: 0000000000000000 RCX: ffff89834ffff000 [...] RDX: 0000000000000018 RSI: ffff89834ffff000 RDI: ffff89834ffff018 [...] RBP: ffff89834ffff000 R08: 0000000000000020 R09: ffff888132b8a900 [...] R10: 4000000000000000 R11: 000000003a616d69 R12: 0000000000000000 [...] R13: ffffffff8404ac28 R14: 0000000000000000 R15: ffff89834ffff000 [...] FS: 0000000000000000(0000) GS:ffff893d44640000(0000) knlGS:0000000000000000 [...] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [...] ata5: SATA link down (SStatus 0 SControl 300) [...] CR2: ffff89834ffff000 CR3: 000001034d00f001 CR4: 0000000000770ef0 [...] PKRU: 55555554 [...] Call Trace: [...] <TASK> [...] ? __die+0x78/0xc0 [...] ? page_fault_oops+0x2a8/0x3a0 [...] ? exc_page_fault+0x84/0x130 [...] ? asm_exc_page_fault+0x22/0x30 [...] ? ima_restore_measurement_list+0x95/0x4b0 [...] ? template_desc_init_fields+0x317/0x410 [...] ? crypto_alloc_tfm_node+0x9c/0xc0 [...] ? init_ima_lsm+0x30/0x30 [...] ima_load_kexec_buffer+0x72/0xa0 [...] ima_init+0x44/0xa0 [...] __initstub_kmod_ima__373_1201_init_ima7+0x1e/0xb0 [...] ? init_ima_lsm+0x30/0x30 [...] do_one_initcall+0xad/0x200 [...] ? idr_alloc_cyclic+0xaa/0x110 [...] ? new_slab+0x12c/0x420 [...] ? new_slab+0x12c/0x420 [...] ? number+0x12a/0x430 [...] ? sysvec_apic_timer_interrupt+0xa/0x80 [...] ? asm_sysvec_apic_timer_interrupt+0x16/0x20 [...] ? parse_args+0xd4/0x380 [...] ? parse_args+0x14b/0x380 [...] kernel_init_freeable+0x1c1/0x2b0 [...] ? rest_init+0xb0/0xb0 [...] kernel_init+0x16/0x1a0 [...] ret_from_fork+0x2f/0x40 [...] ? rest_init+0xb0/0xb0 [...] ret_from_fork_asm+0x11/0x20 [...] </TASK>
CVE-2024-50304	In the Linux kernel, the following vulnerability has been resolved: ipv4: ip_tunnel: Fix suspicious RCU usage warning in ip_tunnel_find() The per-netns IP tunnel hash table is protected by the RTNL mutex and ip_tunnel_find() is only called from the control path where the mutex is taken. Add a lockdep expression to hlist_for_each_entry_rcu() in ip_tunnel_find() in order to validate that the mutex is held and to silence the suspicious RCU usage warning [1]. [1] WARNING: suspicious RCU usage 6.12.0-rc3-custom-gd95d9a31aceb #139 Not tainted ----- net/ipv4/ip_tunnel.c:221 RCU-list traversed in non-reader section!! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by ip/362: #0: ffffffff86fc7cb0 (rtnl_mutex){+..-}{3:3}, at: rtnetlink_rcv_msg+0x377/0xf60 stack backtrace: CPU: 12 UID: 0 PID: 362 Comm: ip Not tainted 6.12.0-rc3-custom-gd95d9a31aceb #139 Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011 Call Trace: <TASK> dump_stack_lvl+0xba/0x110 lockdep_rcu_suspicious.cold+0x4f/0xd6 ip_tunnel_find+0x435/0x4d0 ip_tunnel_newlink+0x517/0x7a0 ipgre_newlink+0x14c/0x170 __rtnl_newlink+0x1173/0x19c0 rtln_newlink+0x6c/0xa0 rtnetlink_rcv_msg+0x3cc/0xf60 netlink_rcv_skb+0x171/0x450 netlink_unicast+0x539/0x7f0 netlink_sendmsg+0x8c1/0xd80 __sys_sendmsg+0x8f9/0xc20 __sys_sendmsg+0x197/0x1e0 __sys_sendmsg+0x122/0x1f0 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-50349	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals. When Git asks for credentials via a terminal prompt (i.e. without using any credential helper), it prints out the host name for which the user is expected to provide a username and/or a password. At this stage, any URL-encoded parts have been decoded already, and are printed verbatim. This allows attackers to craft URLs that contain ANSI escape sequences that the terminal interpret to confuse users e.g. into providing passwords for trusted Git hosting sites when in fact they are then sent to untrusted sites that are under the attacker's control. This issue has been patch via commits `7725b81` and `c903985` which are included in release versions v2.48.1, v2.47.2, v2.46.3, v2.45.3, v2.44.3, v2.43.6, v2.42.4, v2.41.3, and v2.40.4. Users are advised to upgrade. Users unable to upgrade should avoid cloning from untrusted URLs, especially recursive clones.

CVE ID	Description
CVE-2024-50379	Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability during JSP compilation in Apache Tomcat permits an RCE on case insensitive file systems when the default servlet is enabled for write (non-default configuration). This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.1, from 10.1.0-M1 through 10.1.33, from 9.0.0.M1 through 9.0.97. The following versions were EOL at the time the CVE was created but are known to be affected: 8.5.0 through 8.5.100. Other, older, EOL versions may also be affected. Users are recommended to upgrade to version 11.0.2, 10.1.34 or 9.0.98, which fixes the issue.
CVE-2024-50602	An issue was discovered in libexpat before 2.6.4. There is a crash within the XML_ResumeParser function because XML_StopParser can stop/suspend an unstarted parser.
CVE-2024-50613	libsndfile through 1.2.2 has a reachable assertion, that may lead to application exit, in mpeg_l3_encode.c mpeg_l3_encoder_close.
CVE-2024-51744	golang-jwt is a Go implementation of JSON Web Tokens. Unclear documentation of the error behavior in <code>ParseWithClaims</code> can lead to situation where users are potentially not checking errors in the way they should be. Especially, if a token is both expired and invalid, the errors returned by <code>ParseWithClaims</code> return both error codes. If users only check for the <code>jwt.ErrTokenExpired</code> using <code>error.Is</code> , they will ignore the embedded <code>jwt.ErrTokenSignatureInvalid</code> and thus potentially accept invalid tokens. A fix has been back-ported with the error handling logic from the <code>v5</code> branch to the <code>v4</code> branch. In this logic, the <code>ParseWithClaims</code> function will immediately return in "dangerous" situations (e.g., an invalid signature), limiting the combined errors only to situations where the signature is valid, but further validation failed (e.g., if the signature is valid, but is expired AND has the wrong audience). This fix is part of the 4.5.1 release. We are aware that this changes the behaviour of an established function and is not 100 % backwards compatible, so updating to 4.5.1 might break your code. In case you cannot update to 4.5.0, please make sure that you are properly checking for all errors ("dangerous" ones first), so that you are not running in the case detailed above.
CVE-2024-52005	Git is a source code management tool. When cloning from a server (or fetching, or pushing), informational or error messages are transported from the remote Git process to the client via the so-called "sideband channel". These messages will be prefixed with "remote:" and printed directly to the standard error output. Typically, this standard error output is connected to a terminal that understands ANSI escape sequences, which Git did not protect against. Most modern terminals support control sequences that can be used by a malicious actor to hide and misrepresent information, or to mislead the user into executing untrusted scripts. As requested on the git-security mailing list, the patches are under discussion on the public mailing list. Users are advised to update as soon as possible. Users unable to upgrade should avoid recursive clones unless they are from trusted sources.
CVE-2024-52006	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals. Git defines a line-based protocol that is used to exchange information between Git and Git credential helpers. Some ecosystems (most notably, .NET and node.js) interpret single Carriage Return characters as newlines, which renders the protections against CVE-2020-5260 incomplete for credential helpers that treat Carriage Returns in this way. This issue has been addressed in commit <code>b01b9b8</code> which is included in release versions v2.48.1, v2.47.2, v2.46.3, v2.45.3, v2.44.3, v2.43.6, v2.42.4, v2.41.3, and v2.40.4. Users are advised to upgrade. Users unable to upgrade should avoid cloning from untrusted URLs, especially recursive clones.
CVE-2024-52046	The ObjectSerializationDecoder in Apache MINA uses Java's native deserialization protocol to process incoming serialized data but lacks the necessary security checks and defenses. This vulnerability allows attackers to exploit the deserialization process by sending specially crafted malicious serialized data, potentially leading to remote code execution (RCE) attacks. This issue affects MINA core versions 2.0.X, 2.1.X and 2.2.X, and will be fixed by the releases 2.0.27, 2.1.10 and 2.2.4. It's also important to note that an application using MINA core library will only be affected if the <code>IoBuffer#getObject()</code> method is called, and this specific method is potentially called when adding a <code>ProtocolCodecFilter</code> instance using the <code>ObjectSerializationCodecFactory</code> class in the filter chain. If your application is specifically using those classes, you have to upgrade to the latest version of MINA core library. Upgrading will not be enough: you also need to explicitly allow the classes the decoder will accept in the <code>ObjectSerializationDecoder</code> instance, using one of the three new methods: <code>/** * Accept class names where the supplied ClassNameMatcher matches for * deserialization, unless they are otherwise rejected. * * @param classNameMatcher the matcher to use */ public void accept(ClassNameMatcher classNameMatcher) /** * Accept class names that match the supplied pattern for * deserialization, unless they are otherwise rejected. * * @param pattern standard Java regexp */ public void accept(Pattern pattern) /** * Accept the wildcard specified classes for deserialization, * unless they are otherwise rejected. * * @param patterns Wildcard file name patterns as defined by * {@link org.apache.commons.io.FilenameUtils#wildcardMatch(String, String) FilenameUtils.wildcardMatch} */ public void accept(String... patterns)</code> By default, the decoder will reject <code>*all*</code> classes that will be present in the incoming data. Note: The <code>FtpServer</code> , <code>SSHd</code> and <code>Vysper</code> sub-project are not affected by this issue.

CVE ID	Description
CVE-2024-52303	aiohttp is an asynchronous HTTP client/server framework for asyncio and Python. In versions starting with 3.10.6 and prior to 3.10.11, a memory leak can occur when a request produces a MatchInfoError. This was caused by adding an entry to a cache on each request, due to the building of each MatchInfoError producing a unique cache entry. An attacker may be able to exhaust the memory resources of a server by sending a substantial number (100,000s to millions) of such requests. Those who use any middlewares with aiohttp.web should upgrade to version 3.10.11 to receive a patch.
CVE-2024-52304	aiohttp is an asynchronous HTTP client/server framework for asyncio and Python. Prior to version 3.10.11, the Python parser parses newlines in chunk extensions incorrectly which can lead to request smuggling vulnerabilities under certain conditions. If a pure Python version of aiohttp is installed (i.e. without the usual C extensions) or `AIOHTTP_NO_EXTENSIONS` is enabled, then an attacker may be able to execute a request smuggling attack to bypass certain firewalls or proxy protections. Version 3.10.11 fixes the issue.
CVE-2024-52332	In the Linux kernel, the following vulnerability has been resolved: igb: Fix potential invalid memory access in igb_init_module() The pci_register_driver() can fail and when this happened, the dca_notifier needs to be unregistered, otherwise the dca_notifier can be called when igb fails to install, resulting to invalid memory access.
CVE-2024-52559	In the Linux kernel, the following vulnerability has been resolved: drm/msm/gem: prevent integer overflow in msm_ioctl_gem_submit() The "submit->cmd[i].size" and "submit->cmd[i].offset" variables are u32 values that come from the user via the submit_lookup_cmds() function. This addition could lead to an integer wrapping bug so use size_add() to prevent that. Patchwork: https://patchwork.freedesktop.org/patch/624696/
CVE-2024-52560	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Mark inode as bad as soon as error detected in mi_enum_attr() Extended the `mi_enum_attr()` function interface with an additional parameter, `struct ntfs_inode *ni`, to allow marking the inode as bad as soon as an error is detected.
CVE-2024-52615	A flaw was found in Avahi-daemon, which relies on fixed source ports for wide-area DNS queries. This issue simplifies attacks where malicious DNS responses are injected.
CVE-2024-52616	A flaw was found in the Avahi-daemon, where it initializes DNS transaction IDs randomly only once at startup, incrementing them sequentially after that. This predictable behavior facilitates DNS spoofing attacks, allowing attackers to guess transaction IDs.
CVE-2024-53042	In the Linux kernel, the following vulnerability has been resolved: ipv4: ip_tunnel: Fix suspicious RCU usage warning in ip_tunnel_init_flow() There are code paths from which the function is called without holding the RCU read lock, resulting in a suspicious RCU usage warning [1]. Fix by using l3mdev_master_upper_ifindex_by_index() which will acquire the RCU read lock before calling l3mdev_master_upper_ifindex_by_index_rcu(). [1] WARNING: suspicious RCU usage 6.12.0-rc3-custom-gac8f72681cf2 #141 Not tainted ----- net/core/dev.c:876 RCU-list traversed in non-reader section!! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by ip/361: #0: ffffffff86fc7cb0 (rtnl_mutex){+..-}{3:3}, at: rtnetlink_rcv_msg +0x377/0xf60 stack backtrace: CPU: 3 UID: 0 PID: 361 Comm: ip Not tainted 6.12.0-rc3-custom-gac8f72681cf2 #141 Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011 Call Trace: <TASK> dump_stack_lvl+0xba/0x110 lockdep_rcu_suspicious.cold+0x4f/0xd6 dev_get_by_index_rcu +0x1d3/0x210 l3mdev_master_upper_ifindex_by_index_rcu+0x2b/0xf0 ip_tunnel_bind_dev +0x72f/0xa00 ip_tunnel_newlink+0x368/0x7a0 ipgre_newlink+0x14c/0x170 __rtnl_newlink +0x1173/0x19c0 rtnl_newlink+0x6c/0xa0 rtnetlink_rcv_msg+0x3cc/0xf60 netlink_rcv_skb +0x171/0x450 netlink_unicast+0x539/0x7f0 netlink_sendmsg+0x8c1/0xd80 ____sys_sendmsg +0x8f9/0xc20 __sys_sendmsg+0x197/0x1e0 __sys_sendmsg+0x122/0x1f0 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-53043	In the Linux kernel, the following vulnerability has been resolved: mctp i2c: handle NULL header address daddr can be NULL if there is no neighbour table entry present, in that case the tx packet should be dropped. saddr will usually be set by MCTP core, but check for NULL in case a packet is transmitted by a different protocol.

CVE ID	Description
CVE-2024-53044	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_api: fix xa_insert() error path in tcf_block_get_ext() This command: \$ tc qdisc replace dev eth0 ingress_block 1 egress_block 1 clsact Error: block dev insert failed: -EBUSY. fails because user space requests the same block index to be set for both ingress and egress. [side note, I don't think it even failed prior to commit 913b47d3424e ("net/sched: Introduce tc block netdev tracking infra"), because this is a command from an old set of notes of mine which used to work, but alas, I did not scientifically bisect this] The problem is not that it fails, but rather, that the second time around, it fails differently (and irrecoverably): \$ tc qdisc replace dev eth0 ingress_block 1 egress_block 1 clsact Error: dsa_core: Flow block cb is busy. [another note: the extack is added by me for illustration purposes. the context of the problem is that clsact_init() obtains the same &q->ingress_block pointer as &q->egress_block, and since we call tcf_block_get_ext() on both of them, "dev" will be added to the block->ports xarray twice, thus failing the operation: once through the ingress block pointer, and once again through the egress block pointer. the problem itself is that when xa_insert() fails, we have emitted a FLOW_BLOCK_BIND command through ndo_setup_tc(), but the offload never sees a corresponding FLOW_BLOCK_UNBIND.] Even correcting the bad user input, we still cannot recover: \$ tc qdisc replace dev swp3 ingress_block 1 egress_block 2 clsact Error: dsa_core: Flow block cb is busy. Basically the only way to recover is to reboot the system, or unbind and rebind the net device driver. To fix the bug, we need to fill the correct error teardown path which was missed during code movement, and call tcf_block_offload_unbind() when xa_insert() fails. [last note, fundamentally I blame the label naming convention in tcf_block_get_ext() for the bug. The labels should be named after what they do, not after the error path that jumps to them. This way, it is obviously wrong that two labels pointing to the same code mean something is wrong, and checking the code correctness at the goto site is also easier]
CVE-2024-53045	In the Linux kernel, the following vulnerability has been resolved: ASoC: dapm: fix bounds checker error in dapm_widget_list_create The widgets array in the snd_soc_dapm_widget_list has a __counted_by attribute attached to it, which points to the num_widgets variable. This attribute is used in bounds checking, and if it is not set before the array is filled, then the bounds sanitizer will issue a warning or a kernel panic if CONFIG_UBSAN_TRAP is set. This patch sets the size of the widgets list calculated with list_for_each as the initial value for num_widgets as it is used for allocating memory for the array. It is updated with the actual number of added elements after the array is filled.
CVE-2024-53046	In the Linux kernel, the following vulnerability has been resolved: arm64: dts: imx8ulp: correct the flexspi compatible string The flexspi on imx8ulp only has 16 LUTs, and imx8mm flexspi has 32 LUTs, so correct the compatible string here, otherwise will meet below error: [1.119072] -----[cut here]----- [1.123926] WARNING: CPU: 0 PID: 1 at drivers/spi/spi-nxp-fspi.c:855 nxp_fspi_exec_op+0xb04/0xb64 [1.133239] Modules linked in: [1.136448] CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.11.0-rc6-next-20240902-00001-g131bf9439dd9 #69 [1.146821] Hardware name: NXP i.MX8ULP EVK (DT) [1.151647] pstate: 40000005 (nZcv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) [1.158931] pc : nxp_fspi_exec_op+0xb04/0xb64 [1.163496] lr : nxp_fspi_exec_op+0xa34/0xb64 [1.168060] sp : ffff80008002b2a0 [1.171526] x29: ffff80008002b2d0 x28: 0000000000000000 x27: 0000000000000000 [1.179002] x26: ffff2eb645542580 x25: ffff800080610014 x24: ffff800080610000 [1.186480] x23: ffff2eb645548080 x22: 0000000000000006 x21: ffff2eb6455425e0 [1.193956] x20: 0000000000000000 x19: ffff80008002b5e0 x18: ffffffff92 [1.201432] x17: ffff2eb644467508 x16: 0000000000000138 x15: 0000000000000002 [1.208907] x14: 0000000000000000 x13: ffff2eb6400d8080 x12: 00000000ffffff00 [1.216378] x11: 0000000000000000 x10: ffff2eb6400d8080 x9: ffff2eb697adca80 [1.223850] x8: ffff2eb697ad3cc0 x7: 0000000100000000 x6: 0000000000000001 [1.231324] x5: 0000000000000000 x4: 0000000000000000 x3: 00000000000007a6 [1.238795] x2: 0000000000000000 x1: 0000000000000001 ce x0: 00000000ffffff92 [1.246267] Call trace: [1.248824] nxp_fspi_exec_op+0xb04/0xb64 [1.253031] spi_mem_exec_op+0x3a0/0x430 [1.257139] spi_nor_read_id+0x80/0xccc [1.261065] spi_nor_scan+0x1ec/0xf10 [1.264901] spi_nor_probe+0x108/0x2fc [1.268828] spi_mem_probe+0x6c/0xbc [1.272574] spi_probe+0x84/0xe4 [1.275958] really_probe+0xbc/0x29c [1.279713] __driver_probe_device+0x78/0x12c [1.284277] driver_probe_device+0xd8/0x15c [1.288660] __device_attach_driver+0xb8/0x134 [1.293316] bus_for_each_drv+0x88/0xe8 [1.297337] __device_attach+0xa0/0x190 [1.301353] device_initial_probe+0x14/0x20 [1.305734] bus_probe_device+0xac/0xb0 [1.309752] device_add+0x5d0/0x790 [1.313408] __spi_add_device+0x134/0x204 [1.317606] of_register_spi_device+0x3b4/0x590 [1.322348] spi_register_controller+0x47c/0x754 [1.327181] devm_spi_register_controller+0x4c/0xa4 [1.332289] nxp_fspi_probe+0x1cc/0x2b0 [1.336307] platform_probe+0x68/0xc4 [1.340145] really_probe+0xbc/0x29c [1.343893] __driver_probe_device+0x78/0x12c [1.348457] driver_probe_device+0xd8/0x15c [1.352838] __driver_attach+0x90/0x19c [1.356857] bus_for_each_dev+0x7c/0xdc [1.360877] driver_attach+0x24/0x30 [1.364624] bus_add_driver+0xe4/0x208 [1.368552] driver_register+0x5c/0x124 [1.372573] __platform_driver_register+0x28/0x34 [1.377497] nxp_fspi_driver_init+0x1c/0x28 [1.381888] do_one_initcall+0x80/0x1c8 [1.385908] kernel_init_freeable+0x1c4/0x28c [1.390472] kernel_init+0x20/0x1d8 [1.394138] ret_from_fork+0x10/0x20 [1.397885] ---[end trace 0000000000000000]--- [1.407908] -----[cut here]-----

CVE ID	Description
CVE-2024-53047	In the Linux kernel, the following vulnerability has been resolved: mptcp: init: protect sched with rcu_read_lock Enabling CONFIG_PROVE_RCU_LIST with its dependence CONFIG_RCU_EXPERT creates this splat when an MPTCP socket is created: ===== WARNING: suspicious RCU usage 6.12.0-rc2+ #11 Not tainted ----- net/mptcp/sched.c:44 RCU-list traversed in non-reader section!! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 no locks held by mptcp_connect/176. stack backtrace: CPU: 0 UID: 0 PID: 176 Comm: mptcp_connect Not tainted 6.12.0-rc2+ #11 Hardware name: Bochs Bochs, BIOS Bochs 01/01/2011 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:123) lockdep_rcu_suspicious (kernel/locking/lockdep.c:6822) mptcp_sched_find (net/mptcp/sched.c:44 (discriminator 7)) mptcp_init_sock (net/mptcp/protocol.c:2867 (discriminator 1)) ? sock_init_data_uid (arch/x86/include/asm/atomic.h:28) inet_create.part.0.constprop.0 (net/ipv4/af_inet.c:386) ? __sock_create (include/linux/rcupdate.h:347 (discriminator 1)) __sock_create (net/socket.c:1576) __sys_socket (net/socket.c:1671) ? __pfx__sys_socket (net/socket.c:1712) ? do_user_addr_fault (arch/x86/mm/fault.c:1419 (discriminator 1)) __x64_sys_socket (net/socket.c:1728) do_syscall_64 (arch/x86/entry/common.c:52 (discriminator 1)) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) That's because when the socket is initialised, rcu_read_lock() is not used despite the explicit comment written above the declaration of mptcp_sched_find() in sched.c. Adding the missing lock/unlock avoids the warning.
CVE-2024-53048	In the Linux kernel, the following vulnerability has been resolved: ice: fix crash on probe for DPLL enabled E810 LOM The E810 Lan On Motherboard (LOM) design is vendor specific. Intel provides the reference design, but it is up to vendor on the final product design. For some cases, like Linux DPLL support, the static values defined in the driver does not reflect the actual LOM design. Current implementation of dppll pins is causing the crash on probe of the ice driver for such DPLL enabled E810 LOM designs: WARNING: (...) at drivers/dppll/dppll_core.c:495 dppll_pin_get +0x2c4/0x330 ... Call Trace: <TASK> ? __warn+0x83/0x130 ? dppll_pin_get+0x2c4/0x330 ? report_bug +0x1b7/0x1d0 ? handle_bug+0x42/0x70 ? exc_invalid_op+0x18/0x70 ? asm_exc_invalid_op +0x1a/0x20 ? dppll_pin_get+0x117/0x330 ? dppll_pin_get+0x2c4/0x330 ? dppll_pin_get+0x117/0x330 ice_dppll_get_pins.isra.0+0x52/0xe0 [ice] ... The number of dppll pins enabled by LOM vendor is greater than expected and defined in the driver for Intel designed NICs, which causes the crash. Prevent the crash and allow generic pin initialization within Linux DPLL subsystem for DPLL enabled E810 LOM designs. Newly designed solution for described issue will be based on "per HW design" pin initialization. It requires pin information dynamically acquired from the firmware and is already in progress, planned for next-tree only.
CVE-2024-53050	In the Linux kernel, the following vulnerability has been resolved: drm/i915/hdcp: Add encoder check in hdcp2_get_capability Add encoder check in intel_hdcp2_get_capability to avoid null pointer error.
CVE-2024-53051	In the Linux kernel, the following vulnerability has been resolved: drm/i915/hdcp: Add encoder check in intel_hdcp_get_capability Sometimes during hotplug scenario or suspend/resume scenario encoder is not always initialized when intel_hdcp_get_capability add a check to avoid kernel null pointer dereference.
CVE-2024-53052	In the Linux kernel, the following vulnerability has been resolved: io_uring/rw: fix missing NOWAIT check for O_DIRECT start write When io_uring starts a write, it'll call kiocb_start_write() to bump the super block rwsem, preventing any freezes from happening while that write is in-flight. The freeze side will grab that rwsem for writing, excluding any new writers from happening and waiting for existing writes to finish. But io_uring unconditionally uses kiocb_start_write(), which will block if someone is currently attempting to freeze the mount point. This causes a deadlock where freeze is waiting for previous writes to complete, but the previous writes cannot complete, as the task that is supposed to complete them is blocked waiting on starting a new write. This results in the following stuck trace showing that dependency with the write blocked starting a new write: task: fio state: D stack: 0 pid: 886 tgid: 886 ppid: 876 Call trace: __switch_to+0x1d8/0x348 __schedule+0x8e8/0x2248 schedule+0x110/0x3f0 percpu_rwsem_wait+0x1e8/0x3f8 __percpu_down_read+0xe8/0x500 io_write +0xbb8/0xff8 io_issue_sqe+0x10c/0x1020 io_submit_sqes+0x614/0x2110 __arm64_sys_io_uring_enter +0x524/0x1038 invoke_syscall+0x74/0x268 el0_svc_common.constprop.0+0x160/0x238 do_el0_svc+0x44/0x60 el0_svc+0x44/0xb0 el0t_64_sync_handler+0x118/0x128 el0t_64_sync +0x168/0x170 INFO: task fsfreeze:7364 blocked for more than 15 seconds. Not tainted 6.12.0-rc5-00063-g76aaf945701c #7963 with the attempting freezer stuck trying to grab the rwsem: task: fsfreeze state: D stack: 0 pid: 7364 tgid: 7364 ppid: 995 Call trace: __switch_to+0x1d8/0x348 __schedule+0x8e8/0x2248 schedule+0x110/0x3f0 percpu_down_write+0x2b0/0x680 freeze_super +0x248/0x8a8 do_vfs_ioctl+0x149c/0x1b18 __arm64_sys_ioctl+0xd0/0x1a0 invoke_syscall +0x74/0x268 el0_svc_common.constprop.0+0x160/0x238 do_el0_svc+0x44/0x60 el0_svc+0x44/0xb0 el0t_64_sync_handler+0x118/0x128 el0t_64_sync+0x168/0x170 Fix this by having the io_uring side honor IOCB_NOWAIT, and only attempt a blocking grab of the super block rwsem if it isn't set. For normal issue where IOCB_NOWAIT would always be set, this returns -EAGAIN which will have io_uring core issue a blocking attempt of the write. That will in turn also get completions run, ensuring forward progress. Since freezing requires CAP_SYS_ADMIN in the first place, this isn't something that can be triggered by a regular user.

CVE ID	Description
CVE-2024-53053	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Fix another deadlock during RTC update If ufshcd_rtc_work calls ufshcd_rpm_put_sync() and the pm's usage_count is 0, we will enter the runtime suspend callback. However, the runtime suspend callback will wait to flush ufshcd_rtc_work, causing a deadlock. Replace ufshcd_rpm_put_sync() with ufshcd_rpm_put() to avoid the deadlock.
CVE-2024-53055	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: fix 6 GHz scan construction If more than 255 colocated APs exist for the set of all APs found during 2.4/5 GHz scanning, then the 6 GHz scan construction will loop forever since the loop variable has type u8, which can never reach the number found when that's bigger than 255, and is stored in a u32 variable. Also move it into the loops to have a smaller scope. Using a u32 there is fine, we limit the number of APs in the scan list and each has a limit on the number of RNR entries due to the frame size. With a limit of 1000 scan results, a frame size upper bound of 4096 (really it's more like ~2300) and a TBTT entry size of at least 11, we get an upper bound for the number of ~372k, well in the bounds of a u32.
CVE-2024-53056	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: Fix potential NULL dereference in mtk_crtc_destroy() In mtk_crtc_create(), if the call to mbox_request_channel() fails then we set the "mtk_crtc->cmdq_client.chan" pointer to NULL. In that situation, we do not call cmdq_pkt_create(). During the cleanup, we need to check if the "mtk_crtc->cmdq_client.chan" is NULL first before calling cmdq_pkt_destroy(). Calling cmdq_pkt_destroy() is unnecessary if we didn't call cmdq_pkt_create() and it will result in a NULL pointer dereference.
CVE-2024-53057	In the Linux kernel, the following vulnerability has been resolved: net/sched: stop qdisc_tree_reduce_backlog on TC_H_ROOT In qdisc_tree_reduce_backlog, Qdiscs with major handle ffff: are assumed to be either root or ingress. This assumption is bogus since it's valid to create egress qdiscs with major handle ffff: Budimir Markovic found that for qdiscs like DRR that maintain an active class list, it will cause a UAF with a dangling class pointer. In 066a3b5b2346, the concern was to avoid iterating over the ingress qdisc since its parent is itself. The proper fix is to stop when parent TC_H_ROOT is reached because the only way to retrieve ingress is when a hierarchy which does not contain a ffff: major handle call into qdisc_lookup with TC_H_MAJ(TC_H_ROOT). In the scenario where major ffff: is an egress qdisc in any of the tree levels, the updates will also propagate to TC_H_ROOT, which then the iteration must stop. net/sched/sch_api.c 2 +- 1 file changed, 1 insertion(+), 1 deletion(-)
CVE-2024-53058	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: TSO: Fix unbalanced DMA map/unmap for non-paged SKB data In case the non-paged data of a SKB carries protocol header and protocol payload to be transmitted on a certain platform that the DMA AXI address width is configured to 40-bit/48-bit, or the size of the non-paged data is bigger than TSO_MAX_BUFF_SIZE on a certain platform that the DMA AXI address width is configured to 32-bit, then this SKB requires at least two DMA transmit descriptors to serve it. For example, three descriptors are allocated to split one DMA buffer mapped from one piece of non-paged data: dma_desc[N + 0], dma_desc[N + 1], dma_desc[N + 2]. Then three elements of tx_q->tx_skbuff_dma[] will be allocated to hold extra information to be reused in stmmac_tx_clean(): tx_q->tx_skbuff_dma[N + 0], tx_q->tx_skbuff_dma[N + 1], tx_q->tx_skbuff_dma[N + 2]. Now we focus on tx_q->tx_skbuff_dma[entry].buf, which is the DMA buffer address returned by DMA mapping call. stmmac_tx_clean() will try to unmap the DMA buffer_ONLY_IF tx_q->tx_skbuff_dma[entry].buf is a valid buffer address. The expected behavior that saves DMA buffer address of this non-paged data to tx_q->tx_skbuff_dma[entry].buf is: tx_q->tx_skbuff_dma[N + 0].buf = NULL; tx_q->tx_skbuff_dma[N + 1].buf = NULL; tx_q->tx_skbuff_dma[N + 2].buf = dma_map_single(); Unfortunately, the current code misbehaves like this: tx_q->tx_skbuff_dma[N + 0].buf = dma_map_single(); tx_q->tx_skbuff_dma[N + 1].buf = NULL; tx_q->tx_skbuff_dma[N + 2].buf = NULL; On the stmmac_tx_clean() side, when dma_desc[N + 0] is closed by the DMA engine, tx_q->tx_skbuff_dma[N + 0].buf is a valid buffer address obviously, then the DMA buffer will be unmapped immediately. There may be a rare case that the DMA engine does not finish the pending dma_desc[N + 1], dma_desc[N + 2] yet. Now things will go horribly wrong, DMA is going to access a unmapped/unreferenced memory region, corrupted data will be transmitted or iommu fault will be triggered :(In contrast, the for-loop that maps SKB fragments behaves perfectly as expected, and that is how the driver should do for both non-paged data and paged frags actually. This patch corrects DMA map/unmap sequences by fixing the array index for tx_q->tx_skbuff_dma[entry].buf when assigning DMA buffer address. Tested and verified on DWXGMAC CORE 3.20a
CVE-2024-53059	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: Fix response handling in iwl_mvm_send_recovery_cmd() 1. The size of the response packet is not validated. 2. The response buffer is not freed. Resolve these issues by switching to iwl_mvm_send_cmd_status(), which handles both size validation and frees the buffer.

CVE ID	Description
CVE-2024-53060	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: prevent NULL pointer dereference if ATIF is not supported acpi_evaluate_object() may return AE_NOT_FOUND (failure), which would result in dereferencing buffer.pointer (obj) while being NULL. Although this case may be unrealistic for the current code, it is still better to protect against possible bugs. Bail out also when status is AE_NOT_FOUND. This fixes 1 FORWARD_NULL issue reported by Coverity Report: CID 1600951: Null pointer dereferences (FORWARD_NULL) (cherry picked from commit 91c9e221fe2553edf2db71627d8453f083de87a1)
CVE-2024-53061	In the Linux kernel, the following vulnerability has been resolved: media: s5p-jpeg: prevent buffer overflows The current logic allows word to be less than 2. If this happens, there will be buffer overflows, as reported by smatch. Add extra checks to prevent it. While here, remove an unused word = 0 assignment.
CVE-2024-53062	In the Linux kernel, the following vulnerability has been resolved: media: mgb4: protect driver against spectre Frequency range is set from sysfs via frequency_range_store(), being vulnerable to spectre, as reported by smatch: drivers/media/pci/mgb4/mgb4_cmt.c:231 mgb4_cmt_set_vin_freq_range() warn: potential spectre issue 'cmt_vals_in' [r] drivers/media/pci/mgb4/mgb4_cmt.c:238 mgb4_cmt_set_vin_freq_range() warn: possible spectre second half. 'reg_set' Fix it.
CVE-2024-53063	In the Linux kernel, the following vulnerability has been resolved: media: dvbdev: prevent the risk of out of memory access The dvbdev contains a static variable used to store dvb minors. The behavior of it depends if CONFIG_DVB_DYNAMIC_MINORS is set or not. When not set, dvb_register_device() won't check for boundaries, as it will rely that a previous call to dvb_register_adapter() would already be enforcing it. On a similar way, dvb_device_open() uses the assumption that the register functions already did the needed checks. This can be fragile if some device ends using different calls. This also generate warnings on static check analysers like Coverity. So, add explicit guards to prevent potential risk of OOM issues.
CVE-2024-53064	In the Linux kernel, the following vulnerability has been resolved: idpf: fix idpf_vc_core_init error path In an event where the platform running the device control plane is rebooted, reset is detected on the driver. It releases all the resources and waits for the reset to complete. Once the reset is done, it tries to build the resources back. At this time if the device control plane is not yet started, then the driver timeouts on the virtchnl message and retries to establish the mailbox again. In the retry flow, mailbox is deinitialized but the mailbox workqueue is still alive and polling for the mailbox message. This results in accessing the released control queue leading to null-ptr-deref. Fix it by unrolling the work queue cancellation and mailbox deinitialization in the reverse order which they got initialized.
CVE-2024-53066	In the Linux kernel, the following vulnerability has been resolved: nfs: Fix KMSAN warning in decode_getfattr_attr() Fix the following KMSAN warning: CPU: 1 UID: 0 PID: 7651 Comm: cp Tainted: G B Tainted: [B]=BAD_PAGE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009) ===== BUG: KMSAN: uninit-value in decode_getfattr_attr+0x2d6d/0x2f90 decode_getfattr_attr+0x2d6d/0x2f90 decode_getfattr_generic+0x806/0xb00 nfs4_xdr_dec_getfattr+0x1de/0x240 rpcauth_unwrap_resp_decode+0xab/0x100 rpcauth_unwrap_resp+0x95/0xc0 call_decode+0x4ff/0xb50 __rpc_execute+0x57b/0x19d0 rpc_execute+0x368/0x5e0 rpc_run_task+0xcfe/0xee0 nfs4_proc_getfattr+0x5b5/0x990 __nfs_revalidate_inode+0x477/0xd00 nfs_access_get_cached+0x1021/0x1cc0 nfs_do_access+0x9f/0xae0 nfs_permission+0x1e4/0x8c0 inode_permission+0x356/0x6c0 link_path_walk+0x958/0x1330 path_lookupat+0xce/0x6b0 filename_lookup+0x23e/0x770 vfs_statx+0xe7/0x970 vfs_fstatat+0x1f2/0x2c0 __se_sys_newfstatat+0x67/0x880 __x64_sys_newfstatat+0xbd/0x120 x64_sys_call+0x1826/0x3cf0 do_syscall_64+0xd0/0x1b0 entry_SYSCALL_64_after_hwframe+0x77/0x7f The KMSAN warning is triggered in decode_getfattr_attr(), when calling decode_attr_mdsthreshold(). It appears that fattr->mdsthreshold is not initialized. Fix the issue by initializing fattr->mdsthreshold to NULL in nfs_fattr_init().
CVE-2024-53067	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Start the RTC update work later The RTC update work involves runtime resuming the UFS controller. Hence, only start the RTC update work after runtime power management in the UFS driver has been fully initialized. This patch fixes the following kernel crash: Internal error: Oops: 0000000096000006 [#1] PREEMPT SMP Workqueue: events ufshcd_rtc_work Call trace: _raw_spin_lock_irqsave+0x34/0x8c (P) pm_runtime_get_if_active+0x24/0x9c (L) pm_runtime_get_if_active+0x24/0x9c ufshcd_rtc_work+0x138/0x1b4 process_one_work+0x148/0x288 worker_thread+0x2cc/0x3d4 kthread+0x110/0x114 ret_from_fork+0x10/0x20

CVE ID	Description
CVE-2024-53068	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Fix slab-use-after-free in scmi_bus_notifier() The scmi_dev->name is released prematurely in __scmi_device_destroy(), which causes slab-use-after-free when accessing scmi_dev->name in scmi_bus_notifier(). So move the release of scmi_dev->name to scmi_device_release() to avoid slab-use-after-free. BUG: KASAN: slab-use-after-free in strncmp+0xe4/0xec Read of size 1 at addr ffffff80a482bcc0 by task swapper/0/1 CPU: 1 PID: 1 Comm: swapper/0 Not tainted 6.6.38-debug #1 Hardware name: Qualcomm Technologies, Inc. SA8775P Ride (DT) Call trace: dump_backtrace+0x94/0x114 show_stack+0x18/0x24 dump_stack_lvl+0x48/0x60 print_report+0xf4/0x5b0 kasan_report+0xa4/0xec __asan_report_load1_noabort+0x20/0x2c strncmp+0xe4/0xec scmi_bus_notifier+0x5c/0x54c notifier_call_chain+0xb4/0x31c blocking_notifier_call_chain+0x68/0x9c bus_notify+0x54/0x78 device_del+0x1bc/0x840 device_unregister+0x20/0xb4 __scmi_device_destroy+0xac/0x280 scmi_device_destroy+0x94/0xd0 scmi_chan_setup+0x524/0x750 scmi_probe+0x7fc/0x1508 platform_probe+0xc4/0x19c really_probe+0x32c/0x99c __driver_probe_device+0x15c/0x3c4 driver_probe_device+0x5c/0x170 __driver_attach+0x1c8/0x440 bus_for_each_dev+0xf4/0x178 driver_attach+0x3c/0x58 bus_add_driver+0x234/0x4d4 driver_register+0xf4/0x3c0 __platform_driver_register+0x60/0x88 scmi_driver_init+0xb0/0x104 do_one_initcall+0xb4/0x664 kernel_init_freeable+0x3c8/0x894 kernel_init+0x24/0x1e8 ret_from_fork+0x10/0x20 Allocated by task 1: kasan_save_stack+0x2c/0x54 kasan_set_track+0x2c/0x40 kasan_save_alloc_info+0x24/0x34 __kasan_kmalloc+0xa0/0xb8 __kmalloc_node_track_caller+0x6c/0x104 kstrdup+0x48/0x84 kstrdup_const+0x34/0x40 __scmi_device_create.part.0+0x8c/0x408 scmi_device_create+0x104/0x370 scmi_chan_setup+0x2a0/0x750 scmi_probe+0x7fc/0x1508 platform_probe+0xc4/0x19c really_probe+0x32c/0x99c __driver_probe_device+0x15c/0x3c4 driver_probe_device+0x5c/0x170 __driver_attach+0x1c8/0x440 bus_for_each_dev+0xf4/0x178 driver_attach+0x3c/0x58 bus_add_driver+0x234/0x4d4 driver_register+0xf4/0x3c0 __platform_driver_register+0x60/0x88 scmi_driver_init+0xb0/0x104 do_one_initcall+0xb4/0x664 kernel_init_freeable+0x3c8/0x894 kernel_init+0x24/0x1e8 ret_from_fork+0x10/0x20 Freed by task 1: kasan_save_stack+0x2c/0x54 kasan_set_track+0x2c/0x40 kasan_save_free_info+0x38/0x5c __kasan_slab_free+0xe8/0x164 __kmem_cache_free+0x11c/0x230 kfree+0x70/0x130 kfree_const+0x20/0x40 __scmi_device_destroy+0x70/0x280 scmi_device_destroy+0x94/0xd0 scmi_chan_setup+0x524/0x750 scmi_probe+0x7fc/0x1508 platform_probe+0xc4/0x19c really_probe+0x32c/0x99c __driver_probe_device+0x15c/0x3c4 driver_probe_device+0x5c/0x170 __driver_attach+0x1c8/0x440 bus_for_each_dev+0xf4/0x178 driver_attach+0x3c/0x58 bus_add_driver+0x234/0x4d4 driver_register+0xf4/0x3c0 __platform_driver_register+0x60/0x88 scmi_driver_init+0xb0/0x104 do_one_initcall+0xb4/0x664 kernel_init_freeable+0x3c8/0x894 kernel_init+0x24/0x1e8 ret_from_fork+0x10/0x20
CVE-2024-53072	In the Linux kernel, the following vulnerability has been resolved: platform/x86/amd/pmc: Detect when STB is not available Loading the amd_pmc module as: amd_pmc enable_stb=1 ...can result in the following messages in the kernel ring buffer: amd_pmc AMDI0009:00: SMU cmd failed. err: 0xff ioremap on RAM at 0x0000000000000000 - 0x0000000000ffff WARNING: CPU: 10 PID: 2151 at arch/x86/mm/ioremap.c:217 __ioremap_caller+0x2cd/0x340 Further debugging reveals that this occurs when the requests for S2D_PHYS_ADDR_LOW and S2D_PHYS_ADDR_HIGH return a value of 0, indicating that the STB is inaccessible. To prevent the ioremap warning and provide clarity to the user, handle the invalid address and display an error message.
CVE-2024-53073	In the Linux kernel, the following vulnerability has been resolved: NFSD: Never decrement pending_async_copies on error The error flow in nfsd4_copy() calls cleanup_async_copy(), which already decrements nn->pending_async_copies.
CVE-2024-53076	In the Linux kernel, the following vulnerability has been resolved: iio: gts-helper: Fix memory leaks for the error path of iio_gts_build_avail_scale_table() If per_time_scales[i] or per_time_gains[i] kcalloc fails in the for loop of iio_gts_build_avail_scale_table(), the err_free_out will fail to call kfree() each time when i is reduced to 0, so all the per_time_scales[0] and per_time_gains[0] will not be freed, which will cause memory leaks. Fix it by checking if i >= 0.

CVE ID	Description
CVE-2024-53079	In the Linux kernel, the following vulnerability has been resolved: mm/thp: fix deferred split unqueue naming and locking Recent changes are putting more pressure on THP deferred split queues: under load revealing long-standing races, causing list_del corruptions, "Bad page state"s and worse (I keep BUGs in both of those, so usually don't get to see how badly they end up without). The relevant recent changes being 6.8's mTHP, 6.10's mTHP swapout, and 6.12's mTHP swapin, improved swap allocation, and underused THP splitting. Before fixing locking: rename misleading folio_undo_large_rmappable(), which does not undo large_rmappable, to folio_unqueue_deferred_split(), which is what it does. But that and its out-of-line __callee are mm internals of very limited usability: add comment and WARN_ON_ONCEs to check usage; and return a bool to say if a deferred split was unqueued, which can then be used in WARN_ON_ONCEs around safety checks (sparing callers the arcane conditionals in __folio_unqueue_deferred_split()). Just omit the folio_unqueue_deferred_split() from free_unref_folios(), all of whose callers now call it beforehand (and if any forget then bad_page() will tell) - except for its caller put_pages_list(), which itself no longer has any callers (and will be deleted separately). Swapout: mem_cgroup_swapout() has been resetting folio->memcg_data 0 without checking and unqueueing a THP folio from deferred split list; which is unfortunate, since the split_queue_lock depends on the memcg (when memcg is enabled); so swapout has been unqueueing such THPs later, when freeing the folio, using the pgdat's lock instead: potentially corrupting the memcg's list. __remove_mapping() has frozen refcount to 0 here, so no problem with calling folio_unqueue_deferred_split() before resetting memcg_data. That goes back to 5.4 commit 87eaceb3faa5 ("mm: thp: make deferred split shrinker memcg aware"): which included a check on swapcache before adding to deferred queue, but no check on deferred queue before adding THP to swapcache. That worked fine with the usual sequence of events in reclaim (though there were a couple of rare ways in which a THP on deferred queue could have been swapped out), but 6.12 commit daff3f4c850 ("mm: split underused THPs") avoids splitting underused THPs in reclaim, which makes swapcache THPs on deferred queue commonplace. Keep the check on swapcache before adding to deferred queue? Yes: it is no longer essential, but preserves the existing behaviour, and is likely to be a worthwhile optimization (vmstat showed much more traffic on the queue under swapping load if the check was removed); update its comment. Memcg-v1 move (deprecated): mem_cgroup_move_account() has been changing folio->memcg_data without checking and unqueueing a THP folio from the deferred list, sometimes corrupting "from" memcg's list, like swapout. Refcount is non-zero here, so folio_unqueue_deferred_split() can only be used in a WARN_ON_ONCE to validate the fix, which must be done earlier: mem_cgroup_move_charge_pte_range() first try to split the THP (splitting of course unqueues), or skip it if that fails. Not ideal, but moving charge has been requested, and khugepaged should repair the THP later: nobody wants new custom unqueueing code just for this deprecated case. The 87eaceb3faa5 commit did have the code to move from one deferred list to another (but was not conscious of its unsafety while refcount non-0); but that was removed by 5.6 commit fac0516b5534 ("mm: thp: don't need care deferred split queue in memcg charge move path"), which argued that the existence of a PMD mapping guarantees that the THP cannot be on a deferred list. As above, false in rare cases, and now commonly false. Backport to 6.11 should be straightforward. Earlier backports must take care that other _deferred_list fixes and dependencies are included. There is not a strong case for backports, but they can fix cornercases.
CVE-2024-53081	In the Linux kernel, the following vulnerability has been resolved: media: ar0521: don't overflow when checking PLL values The PLL checks are comparing 64 bit integers with 32 bit ones, as reported by Coverity. Depending on the values of the variables, this may underflow. Fix it ensuring that both sides of the expression are u64.
CVE-2024-53082	In the Linux kernel, the following vulnerability has been resolved: virtio_net: Add hash_key_length check Add hash_key_length check in virtnet_probe() to avoid possible out of bound errors when setting/reading the hash key.
CVE-2024-53083	In the Linux kernel, the following vulnerability has been resolved: usb: typec: qcom-pmic: init value of hdr_len/txbuf_len earlier If the read of USB_PDPHY_RX_ACKNOWLEDGE_REG failed, then hdr_len and txbuf_len are uninitialized. This commit stops to print uninitialized value and misleading/false data.
CVE-2024-53084	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: Break an object reference loop When remaining resources are being cleaned up on driver close, outstanding VM mappings may result in resources being leaked, due to an object reference loop, as shown below, with each object (or set of objects) referencing the object below it: PVR GEM Object GPU scheduler "finished" fence GPU scheduler "scheduled" fence PVR driver "done" fence PVR Context PVR VM Context PVR VM Mappings PVR GEM Object The reference that the PVR VM Context has on the VM mappings is a soft one, in the sense that the freeing of outstanding VM mappings is done as part of VM context destruction; no reference counts are involved, as is the case for all the other references in the loop. To break the reference loop during cleanup, free the outstanding VM mappings before destroying the PVR Context associated with the VM context.

CVE ID	Description
CVE-2024-53085	In the Linux kernel, the following vulnerability has been resolved: tpm: Lock TPM chip in tpm_pm_suspend() first Setting TPM_CHIP_FLAG_SUSPENDED in the end of tpm_pm_suspend() can be racy according, as this leaves window for tpm_hwrng_read() to be called while the operation is in progress. The recent bug report gives also evidence of this behaviour. Address this by locking the TPM chip before checking any chip->flags both in tpm_pm_suspend() and tpm_hwrng_read(). Move TPM_CHIP_FLAG_SUSPENDED check inside tpm_get_random() so that it will be always checked only when the lock is reserved.
CVE-2024-53086	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Drop VM dma-resv lock on xe_sync_in_fence_get failure in exec IOCTL Upon failure all locks need to be dropped before returning to the user. (cherry picked from commit 7d1a4258e602ffdc529f56686925034c1b3b095)
CVE-2024-53087	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix possible exec queue leak in exec IOCTL In a couple of places after an exec queue is looked up the exec IOCTL returns on input errors without dropping the exec queue ref. Fix this ensuring the exec queue ref is dropped on input error. (cherry picked from commit 07064a200b40ac2195cb6b7b779897d9377e5e6f)
CVE-2024-53088	In the Linux kernel, the following vulnerability has been resolved: i40e: fix race condition by adding filter's intermediate sync state Fix a race condition in the i40e driver that leads to MAC/VLAN filters becoming corrupted and leaking. Address the issue that occurs under heavy load when multiple threads are concurrently modifying MAC/VLAN filters by setting mac and port VLAN. 1. Thread T0 allocates a filter in i40e_add_filter() within i40e_ndo_set_vf_port_vlan(). 2. Thread T1 concurrently frees the filter in __i40e_del_filter() within i40e_ndo_set_vf_mac(). 3. Subsequently, i40e_service_task() calls i40e_sync_vsi_filters(), which refers to the already freed filter memory, causing corruption. Reproduction steps: 1. Spawn multiple VFs. 2. Apply a concurrent heavy load by running parallel operations to change MAC addresses on the VFs and change port VLANs on the host. 3. Observe errors in dmesg: "Error I40E_AQ_RC_ENOSPC adding RX filters on VF XX, please set promiscuous on manually for VF XX". Exact code for stable reproduction Intel can't open-source now. The fix involves implementing a new intermediate filter state, I40E_FILTER_NEW_SYNC, for the time when a filter is on a tmp_add_list. These filters cannot be deleted from the hash list directly but must be removed using the full process.
CVE-2024-53089	In the Linux kernel, the following vulnerability has been resolved: LoongArch: KVM: Mark hrtimer to expire in hard interrupt context Like commit 2c0d278f3293f ("KVM: LAPIC: Mark hrtimer to expire in hard interrupt context") and commit 9090825fa9974 ("KVM: arm/arm64: Let the timer expire in hardirq context on RT"), On PREEMPT_RT enabled kernels unmarked hrtimers are moved into soft interrupt expiry mode by default. Then the timers are canceled from a preempt-notifier which is invoked with disabled preemption which is not allowed on PREEMPT_RT. The timer callback is short so in could be invoked in hard-IRQ context. So let the timer expire on hard-IRQ context even on -RT. This fix a "scheduling while atomic" bug for PREEMPT_RT enabled kernels: BUG: scheduling while atomic: qemu-system-loo/1011/0x00000002 Modules linked in: amdgpu rfkill nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat ns CPU: 1 UID: 0 PID: 1011 Comm: qemu-system-loo Tainted: G W 6.12.0-rc2+ #1774 Tainted: [W]=WARN Hardware name: Loongson Loongson-3A5000-7A1000-1w-CRB/Loongson-LS3A5000-7A1000-1w-CRB, BIOS vUDK2018-LoongArch-V2.0.0-prebeta9 10/21/2022 Stack : ffffffff 0000000000000000 9000000004e3ea38 90000000116744000 900000001167475a0 0000000000000000 900000001167475a8 9000000005644830 90000000058dc000 90000000058dbff8 90000000116747420 0000000000000001 0000000000000001 6a613fc938313980 000000000790c000 900000001001c1140 000000000000003fe 0000000000000001 000000000000000d 0000000000000003 0000000000000030 000000000000003f3 000000000790c000 90000000116747830 90000000057ef000 0000000000000000 9000000005644830 0000000000000004 0000000000000000 90000000057f4b58 0000000000000001 90000000116747868 900000000451b600 9000000005644830 9000000003a13998 0000000010000020 00000000000000b0 0000000000000004 0000000000000000 0000000000071c1d ... Call Trace: [<9000000003a13998>] show_stack+0x38/0x180 [<9000000004e3ea34>] dump_stack_lvl+0x84/0xc0 [<9000000003a71708>] __schedule_bug+0x48/0x60 [<9000000004e45734>] __schedule+0x1114/0x1660 [<9000000004e46040>] schedule_rtlock+0x20/0x60 [<9000000004e4e330>] rtlock_slowlock_locked+0x3f0/0x10a0 [<9000000004e4f038>] rt_spin_lock+0x58/0x80 [<9000000003b02d68>] hrtimer_cancel_wait_running+0x68/0xc0 [<9000000003b02e30>] hrtimer_cancel+0x70/0x80 [<ffff80000235eb70>] kvm_restore_timer+0x50/0x1a0 [kvm] [<ffff8000023616c8>] kvm_arch_vcpu_load+0x68/0x2a0 [kvm] [<ffff80000234c2d4>] kvm_sched_in+0x34/0x60 [kvm] [<9000000003a749a0>] finish_task_switch.isra.0+0x140/0x2e0 [<9000000004e44a70>] __schedule+0x450/0x1660 [<9000000004e45cb0>] schedule+0x30/0x180 [<ffff800002354c70>] kvm_vcpu_block+0x70/0x120 [kvm] [<ffff800002354d80>] kvm_vcpu_halt+0x60/0x3e0 [kvm] [<ffff80000235b194>] kvm_handle_gspr+0x3f4/0x4e0 [kvm] [<ffff80000235f548>] kvm_handle_exit+0x1c8/0x260 [kvm]

CVE ID	Description
CVE-2024-53090	In the Linux kernel, the following vulnerability has been resolved: afs: Fix lock recursion afs_wake_up_async_call() can incur lock recursion. The problem is that it is called from AF_RXRPC whilst holding the ->notify_lock, but it tries to take a ref on the afs_call struct in order to pass it to a work queue - but if the afs_call is already queued, we then have an extraneous ref that must be put... calling afs_put_call() may call back down into AF_RXRPC through rxrpc_kernel_shutdown_call(), however, which might try taking the ->notify_lock again. This case isn't very common, however, so defer it to a workqueue. The oops looks something like: BUG: spinlock recursion on CPU#0, krxrpcio/7001/1646 lock: 0xffff888141399b30, .magic: dead4ead, .owner: krxrpcio/7001/1646, .owner_cpu: 0 CPU: 0 UID: 0 PID: 1646 Comm: krxrpcio/7001 Not tainted 6.12.0-rc2-build3+ #4351 Hardware name: ASUS All Series/H97-PLUS, BIOS 2306 10/09/2014 Call Trace: <TASK> dump_stack_lvl+0x47/0x70 do_raw_spin_lock+0x3c/0x90 rxrpc_kernel_shutdown_call+0x83/0xb0 afs_put_call+0xd7/0x180 rxrpc_notify_socket+0xa0/0x190 rxrpc_input_split_jumbo+0x198/0x1d0 rxrpc_input_data+0x14b/0x1e0 ? rxrpc_input_call_packet+0xc2/0x1f0 rxrpc_input_call_event+0xad/0x6b0 rxrpc_input_packet_on_conn+0x1e1/0x210 rxrpc_input_packet+0x3f2/0x4d0 rxrpc_io_thread+0x243/0x410 ? __pfx_rxrpc_io_thread+0x10/0x10 kthread+0xcf/0xe0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x24/0x40 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK>
CVE-2024-53091	In the Linux kernel, the following vulnerability has been resolved: bpf: Add sk_is_inet and IS_ICSK check in tls_sw_has_ctx_tx/rx As the introduction of the support for vsock and unix sockets in sockmap, tls_sw_has_ctx_tx/rx cannot presume the socket passed in must be IS_ICSK. vsock and af_unix sockets have vsock_sock and unix_sock instead of inet_connection_sock. For these sockets, tls_get_ctx may return an invalid pointer and cause page fault in function tls_sw_ctx_rx. BUG: unable to handle page fault for address: 0000000000040030 Workqueue: vsock-loopback vsock_loopback_work RIP: 0010:sk_psock_strp_data_ready+0x23/0x60 Call Trace: ? __die+0x81/0xc3 ? no_context+0x194/0x350 ? do_page_fault+0x30/0x110 ? async_page_fault+0x3e/0x50 ? sk_psock_strp_data_ready+0x23/0x60 virtio_transport_recv_pkt+0x750/0x800 ? update_load_avg+0x7e/0x620 vsock_loopback_work+0xd0/0x100 process_one_work+0x1a7/0x360 worker_thread+0x30/0x390 ? create_worker+0x1a0/0x1a0 kthread+0x112/0x130 ? __kthread_cancel_work+0x40/0x40 ret_from_fork+0x1f/0x40 v2: - Add IS_ICSK check v3: - Update the commits in Fixes
CVE-2024-53093	In the Linux kernel, the following vulnerability has been resolved: nvme-multipath: defer partition scanning We need to suppress the partition scan from occurring within the controller's scan_work context. If a path error occurs here, the IO will wait until a path becomes available or all paths are torn down, but that action also occurs within scan_work, so it would deadlock. Defer the partition scan to a different context that does not block scan_work.
CVE-2024-53094	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Add sendpage_ok() check to disable MSG_SPLICE_PAGES While running ISER over SIW, the initiator machine encounters a warning from skb_splice_from_iter() indicating that a slab page is being used in send_page. To address this, it is better to add a sendpage_ok() check within the driver itself, and if it returns 0, then MSG_SPLICE_PAGES flag should be disabled before entering the network stack. A similar issue has been discussed for NVMe in this thread: https://lore.kernel.org/all/20240530142417.146696-1-ofir.gal@volumez.com/ WARNING: CPU: 0 PID: 5342 at net/core/skbuff.c:7140 skb_splice_from_iter+0x173/0x320 Call Trace: tcp_sendmsg_locked+0x368/0xe40 siw_tx_hdt+0x695/0xa40 [siw] siw_qp_sq_process+0x102/0xb00 [siw] siw_sq_resume+0x39/0x110 [siw] siw_run_sq+0x74/0x160 [siw] kthread+0xd2/0x100 ret_from_fork+0x34/0x40 ret_from_fork_asm+0x1a/0x30

CVE ID	Description
CVE-2024-53095	In the Linux kernel, the following vulnerability has been resolved: smb: client: Fix use-after-free of network namespace. Recently, we got a customer report that CIFS triggers oops while reconnecting to a server. [0] The workload runs on Kubernetes, and some pods mount CIFS servers in non-root network namespaces. The problem rarely happened, but it was always while the pod was dying. The root cause is wrong reference counting for network namespace. CIFS uses kernel sockets, which do not hold refcnt of the netns that the socket belongs to. That means CIFS must ensure the socket is always freed before its netns; otherwise, use-after-free happens. The repro steps are roughly: 1. mount CIFS in a non-root netns 2. drop packets from the netns 3. destroy the netns 4. unmount CIFS We can reproduce the issue quickly with the script [1] below and see the splat [2] if CONFIG_NET_NS_REFCNT_TRACKER is enabled. When the socket is TCP, it is hard to guarantee the netns lifetime without holding refcnt due to async timers. Let's hold netns refcnt for each socket as done for SMC in commit 9744d2bf1976 ("smc: Fix use-after-free in tcp_write_timer_handler()."). Note that we need to move put_net() from cifs_put_tcp_session() to clean_demultiplex_info(); otherwise, __sock_create() still could touch a freed netns while cifsd tries to reconnect from cifs_demultiplex_thread(). Also, maybe_get_net() cannot be put just before __sock_create() because the code is not under RCU and there is a small chance that the same address happened to be reallocated to another netns. [0]: CIFS: VFS: \\XXXXXXXXXXXX has not responded in 15 seconds. Reconnecting... CIFS: Serverclose failed 4 times, giving up Unable to handle kernel paging request at virtual address 14de99e461f84a07 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x000000004 CM = 0, WnR = 0 [14de99e461f84a07] address between user and kernel address ranges Internal error: Oops: 0000000096000004 [#1] SMP Modules linked in: cls_bpf sch_ingress nls_utf8 cifs cifs_arc4 cifs_md4 dns_resolver tcp_diag inet_diag veth xt_state xt_connmark nf_conntrack_netlink xt_nat xt_statistic xt_MASQUERADE xt_mark xt_addrtype ipt_REJECT nf_reject_ipv4 nft_chain_nat nf_nat xt_conntrack nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 xt_comment nft_compat nf_tables nfnetlink overlay nls_ascii nls_cp437 sunrpc vfat fat aes_ce_blk aes_ce_cipher ghash_ce sm4_ce_cipher sm4 sm3_ce sm3 sha3_ce sha512_ce sha512_arm64 sha1_ce ena button sch_fq_codel loop fuse configfs dmi_sysfs sha2_ce sha256_arm64 dm_mirror dm_region_hash dm_log dm_mod dax efivarfs CPU: 5 PID: 2690970 Comm: cifs Not tainted 6.1.103-109.184.amzn2023.aarch64 #1 Hardware name: Amazon EC2 r7g.4xlarge/, BIOS 1.0 11/1/2018 pstate: 00400005 (nzcvc daif +PAN -UAO -DIT -SSBS BTYPEDIT) pc : fib_rules_lookup+0x44/0x238 lr : __fib_lookup+0x64/0xabc sp : ffff8000265db790 x29: ffff8000265db790 x28: 0000000000000000 x27: 000000000000bd01 x26: 0000000000000000 x25: ffff000b4baf8000 x24: ffff00047b5e4580 x23: ffff8000265db7e0 x22: 0000000000000000 x21: ffff00047b5e4500 x20: ffff0010e3f694f8 x19: 14de99e461f849f7 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 3f92800abd010002 x11: 0000000000000001 x10: ffff0010e3f69420 x9: ffff800008a6f294 x8: 0000000000000000 x7: 0000000000000006 x6: 0000000000000000 x5: 0000000000000001 x4: ffff001924354280 x3: ffff8000265db7e0 x2: 0000000000000000 x1: ffff0010e3f694f8 x0: ffff00047b5e4500 Call trace: fib_rules_lookup+0x44/0x238 __fib_lookup+0x64/0xabc ip_route_output_key_hash_rcu+0x2c4/0x398 ip_route_output_key_hash+0x60/0x8c tcp_v4_connect+0x290/0x488 __inet_stream_connect+0x108/0x3d0 inet_stream_connect+0x50/0x78 kernel_connect+0x6c/0xac generic_ip_conne ---truncated---

CVE ID	Description
CVE-2024-53096	In the Linux kernel, the following vulnerability has been resolved: mm: resolve faulty mmap_region() error path behaviour The mmap_region() function is somewhat terrifying, with spaghetti-like control flow and numerous means by which issues can arise and incomplete state, memory leaks and other unpleasantness can occur. A large amount of the complexity arises from trying to handle errors late in the process of mapping a VMA, which forms the basis of recently observed issues with resource leaks and observable inconsistent state. Taking advantage of previous patches in this series we move a number of checks earlier in the code, simplifying things by moving the core of the logic into a static internal function __mmap_region(). Doing this allows us to perform a number of checks up front before we do any real work, and allows us to unwind the writable unmap check unconditionally as required and to perform a CONFIG_DEBUG_VM_MAPLE_TREE validation unconditionally also. We move a number of things here: 1. We preallocate memory for the iterator before we call the file-backed memory hook, allowing us to exit early and avoid having to perform complicated and error-prone close/free logic. We carefully free iterator state on both success and error paths. 2. The enclosing mmap_region() function handles the mapping_map_writable() logic early. Previously the logic had the mapping_map_writable() at the point of mapping a newly allocated file-backed VMA, and a matching mapping_unmap_writable() on success and error paths. We now do this unconditionally if this is a file-backed, shared writable mapping. If a driver changes the flags to eliminate VM_MAYWRITE, however doing so does not invalidate the seal check we just performed, and we in any case always decrement the counter in the wrapper. We perform a debug assert to ensure a driver does not attempt to do the opposite. 3. We also move arch_validate_flags() up into the mmap_region() function. This is only relevant on arm64 and sparc64, and the check is only meaningful for SPARC with ADI enabled. We explicitly add a warning for this arch if a driver invalidates this check, though the code ought eventually to be fixed to eliminate the need for this. With all of these measures in place, we no longer need to explicitly close the VMA on error paths, as we place all checks which might fail prior to a call to any driver mmap hook. This eliminates an entire class of errors, makes the code easier to reason about and more robust.
CVE-2024-53098	In the Linux kernel, the following vulnerability has been resolved: drm/xe/ufence: Prefetch ufence addr to catch bogus address access_ok() only checks for addr overflow so also try to read the addr to catch invalid addr sent from userspace. (cherry picked from commit 9408c4508483ffc60811e910a93d6425b8e63928)
CVE-2024-53099	In the Linux kernel, the following vulnerability has been resolved: bpf: Check validity of link->type in bpf_link_show_fdinfo() If a newly-added link type doesn't invoke BPF_LINK_TYPE(), accessing bpf_link_type_strs[link->type] may result in an out-of-bounds access. To spot such missed invocations early in the future, checking the validity of link->type in bpf_link_show_fdinfo() and emitting a warning when such invocations are missed.

CVE ID	Description
CVE-2024-53100	In the Linux kernel, the following vulnerability has been resolved: nvme: tcp: avoid race between queue_lock lock and destroy Commit 76d54bf20cdc ("nvme-tcp: don't access released socket during error recovery") added a mutex_lock() call for the queue->queue_lock in nvme_tcp_get_address(). However, the mutex_lock() races with mutex_destroy() in nvme_tcp_free_queue(), and causes the WARN below. DEBUG_LOCKS_WARN_ON(lock->magic != lock) WARNING: CPU: 3 PID: 34077 at kernel/locking/mutex.c:587 __mutex_lock+0xc0/0x1220 Modules linked in: nvmet_tcp nvmet nvme_tcp nvme_fabrics iw_cm ib_cm ib_core pktcdvd nft_fib_inet nft_fib_ipv4 nft_fib_ipv6 nft_fib nft_reject_inet nf_reject_ipv4 nf_reject_ipv6 nft_reject nft_ct nft_chain_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 ip_set nf_tables qrtr sunrpc ppdev 9pnet_virtio 9pnet pcspkr netfs parport_pc parport e1000 i2c_piix4 i2c_smbus loop fuse nfnetlink zram bochs drm_vram_helper drm_ttm_helper ttm drm_kms_helper xfs drm sym53c8xx floppy nvme scsi_transport_spi nvme_core nvme_auth serio_raw ata_generic pata_acpi dm_multipath qemu_fw_cfg [last unloaded: ib_uverbs] CPU: 3 UID: 0 PID: 34077 Comm: udisksd Not tainted 6.11.0-rc7 #319 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-2.fc40 04/01/2014 RIP: 0010:__mutex_lock+0xc0/0x1220 Code: 08 84 d2 0f 85 c8 04 00 00 8b 15 ef b6 c8 01 85 d2 0f 85 78 f4 ff ff 48 c7 c6 20 93 ee af 48 c7 c7 60 91 ee af e8 f0 a7 6d fd <0f> 0b e9 5e f4 ff ff 48 b8 00 00 00 00 00 fc ff df 4c 89 f2 48 c1 RSP: 0018:ffff88811305f760 EFLAGS: 00010286 RAX: 0000000000000000 RBX: ffff88812c652058 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000001 RBP: ffff88811305f8b0 R08: 0000000000000001 R09: ffffed1075c36341 R10: ffff8883ae1b1a0b R11: 0000000000010498 R12: 0000000000000000 R13: 0000000000000000 R14: dffffc0000000000 R15: ffff88812c652058 FS: 00007f9713ae4980(0000) GS: ffff8883ae180000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fcd78483c7c CR3: 0000000122c38000 CR4: 0000000000000060 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __warn.cold+0x5b/0x1af ? __mutex_lock+0xc0/0x1220 ? report_bug+0x1ec/0x390 ? handle_bug+0x3c/0x80 ? exc_invalid_op+0x13/0x40 ? asm_exc_invalid_op+0x16/0x20 ? __mutex_lock+0xc0/0x1220 ? nvme_tcp_get_address+0xc2/0x1e0 [nvme_tcp] ? __pfx__mutex_lock+0x10/0x10 ? __lock_acquire+0xd6a/0x59e0 ? nvme_tcp_get_address+0xc2/0x1e0 [nvme_tcp] nvme_tcp_get_address+0xc2/0x1e0 [nvme_tcp] ? __pfx_nvme_tcp_get_address+0x10/0x10 [nvme_tcp] nvme_sysfs_show_address+0x81/0xc0 [nvme_core] dev_attr_show+0x42/0x80 ? __asan_memset+0x1f/0x40 sysfs_kf_seq_show+0x1f0/0x370 seq_read_iter+0x2cb/0x1130 ? rw_verify_area+0x3b1/0x590 ? __mutex_lock+0x433/0x1220 vfs_read+0x6a6/0xa20 ? lockdep_hardirqs_on+0x78/0x100 ? __pfx_vfs_read+0x10/0x10 ksys_read+0xf7/0x1d0 ? __pfx_ksys_read+0x10/0x10 ? __x64_sys_openat+0x105/0x1d0 do_syscall_64+0x93/0x180 ? lockdep_hardirqs_on_prepare+0x16d/0x400 ? do_syscall_64+0x9f/0x180 ? lockdep_hardirqs_on+0x78/0x100 ? do_syscall_64+0x9f/0x180 ? __pfx_ksys_read+0x10/0x10 ? lockdep_hardirqs_on_prepare+0x16d/0x400 ? do_syscall_64+0x9f/0x180 ? lockdep_hardirqs_on+0x78/0x100 ? do_syscall_64+0x9f/0x180 ? lockdep_hardirqs_on_prepare+0x16d/0x400 ? do_syscall_64+0x9f/0x180 ? lockdep_hardirqs_on+0x78/0x100 ? do_syscall_64+0x9f/0x180 ? lockdep_hardirqs_on_prepare+0x16d/0x400 ? do_syscall_64+0x9f/0x180 ? lockdep_hardirqs_on+0x78/0x100 ? do_syscall_64+0x9f/0x180 ? do_syscall_64+0x9f/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f9713f55cfa Code: 55 48 89 e5 48 83 ec 20 48 89 55 e8 48 89 75 f0 89 7d f8 e8 e8 74 f8 ff 48 8b 55 e8 48 8b 75 f0 4 ---truncated---
CVE-2024-53101	In the Linux kernel, the following vulnerability has been resolved: fs: Fix uninitialized value issue in from_kuid and from_kgid ocfs2_setattr() uses attr->ia_mode, attr->ia_uid and attr->ia_gid in a trace point even though ATTR_MODE, ATTR_UID and ATTR_GID aren't set. Initialize all fields of newattr to avoid uninitialized variables, by checking if ATTR_MODE, ATTR_UID, ATTR_GID are initialized, otherwise 0.
CVE-2024-53103	In the Linux kernel, the following vulnerability has been resolved: hv_sock: Initializing vsk->trans to NULL to prevent a dangling pointer When hvs is released, there is a possibility that vsk->trans may not be initialized to NULL, which could lead to a dangling pointer. This issue is resolved by initializing vsk->trans to NULL.

CVE ID	Description
CVE-2024-53105	In the Linux kernel, the following vulnerability has been resolved: mm: page_alloc: move mlocked flag clearance into free_pages_prepare() Syzbot reported a bad page state problem caused by a page being freed using free_page() still having a mlocked flag at free_pages_prepare() stage: BUG: Bad page state in process syz.5.504 pfn:61f45 page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x61f45 flags: 0xfff00000080204(referenced workingset mlocked node=0 zone=1 lastcpupid=0x7ff) raw: 00fff00000080204 0000000000000000 dead000000000122 0000000000000000 raw: 0000000000000000 0000000000000000 00000000ffffff 0000000000000000 page dumped because: PAGE_FLAGS_CHECK_AT_FREE flag(s) set page_owner tracks the page as allocated page last allocated via order 0, migratetype Unmovable, gfp_mask 0x400dc0(GFP_KERNEL_ACCOUNT __GFP_ZERO), pid 8443, tgid 8442 (syz.5.504), ts 201884660643, free_ts 201499827394 set_page_owner include/linux/page_owner.h:32 [inline] post_alloc_hook+0x1f3/0x230 mm/page_alloc.c:1537 prep_new_page mm/page_alloc.c:1545 [inline] get_page_from_freelist+0x303f/0x3190 mm/page_alloc.c:3457 __alloc_pages_noprof+0x292/0x710 mm/page_alloc.c:4733 alloc_pages_mpol_noprof+0x3e8/0x680 mm/mempolicy.c:2265 kvm_coalesced_mmio_init+0x1f/0xf0 virt/kvm/coalesced_mmio.c:99 kvm_create_vm virt/kvm/kvm_main.c:1235 [inline] kvm_dev_ioctl_create_vm virt/kvm/kvm_main.c:5488 [inline] kvm_dev_ioctl+0x12dc/0x2240 virt/kvm/kvm_main.c:5530 __do_compat_sys_ioctl fs/ioctl.c:1007 [inline] __se_compat_sys_ioctl+0x510/0xc90 fs/ioctl.c:950 do_syscall_32_irqs_on arch/x86/entry/common.c:165 [inline] __do_fast_syscall_32+0xb4/0x110 arch/x86/entry/common.c:386 do_fast_syscall_32+0x34/0x80 arch/x86/entry/common.c:411 entry_SYSENTER_compat_after_hwframe+0x84/0x8e page last free pid 8399 tgid 8399 stack trace: reset_page_owner include/linux/page_owner.h:25 [inline] free_pages_prepare mm/page_alloc.c:1108 [inline] free_unref_folios+0xf12/0x18d0 mm/page_alloc.c:2686 folios_put_refs+0x76c/0x860 mm/swap.c:1007 free_pages_and_swap_cache+0x5c8/0x690 mm/swap_state.c:335 __tlb_batch_free_encoded_pages mm/mmu_gather.c:136 [inline] tlb_batch_pages_flush mm/mmu_gather.c:149 [inline] tlb_flush_mmu_free mm/mmu_gather.c:366 [inline] tlb_flush_mmu+0x3a3/0x680 mm/mmu_gather.c:373 tlb_finish_mmu+0xd4/0x200 mm/mmu_gather.c:465 exit_mmap+0x496/0xc40 mm/mmap.c:1926 __mmapput+0x115/0x390 kernel/fork.c:1348 exit_mm+0x220/0x310 kernel/exit.c:571 do_exit+0x9b2/0x28e0 kernel/exit.c:926 do_group_exit+0x207/0x2c0 kernel/exit.c:1088 __do_sys_exit_group kernel/exit.c:1099 [inline] __se_sys_exit_group kernel/exit.c:1097 [inline] __x64_sys_exit_group+0x3f/0x40 kernel/exit.c:1097 x64_sys_call+0x2634/0x2640 arch/x86/include/generated/asm/syscalls_64.h:232 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0xf7 Modules linked in: CPU: 0 UID: 0 PID: 8442 Comm: syz.5.504 Not tainted 6.12.0-rc6-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 bad_page+0x176/0x1d0 mm/page_alloc.c:501 free_page_is_bad mm/page_alloc.c:918 [inline] free_pages_prepare mm/page_alloc.c:1100 [inline] free_unref_page+0xed0/0xf20 mm/page_alloc.c:2638 kvm_destroy_vm virt/kvm/kvm_main.c:1327 [inline] kvm_put_kvm+0x75/0x1350 virt/kvm/kvm_main.c:1386 kvm_vcpu_release+0x54/0x60 virt/kvm/kvm_main.c:4143 __fput+0x23f/0x880 fs/file_table.c:431 task_work_run+0x24f/0x310 kernel/task_work.c:239 exit_task_work include/linux/task_work.h:43 [inline] do_exit+0xa2f/0x28e0 kernel/exit.c:939 do_group_exit+0x207/0x2c0 kernel/exit.c:1088 __do_sys_exit_group kernel/exit.c:1099 [in ---truncated---
CVE-2024-53106	In the Linux kernel, the following vulnerability has been resolved: ima: fix buffer overrun in ima_eventdigest_init_common Function ima_eventdigest_init() calls ima_eventdigest_init_common() with HASH_ALGO_LAST which is then used to access the array hash_digest_size[] leading to buffer overrun. Have a conditional statement to handle this.
CVE-2024-53107	In the Linux kernel, the following vulnerability has been resolved: fs/proc/task_mmu: prevent integer overflow in pagemap_scan_get_args() The "arg->vec_len" variable is a u64 that comes from the user at the start of the function. The "arg->vec_len * sizeof(struct page_region))" multiplication can lead to integer wrapping. Use size_mul() to avoid that. Also the size_add/mul() functions work on unsigned long so for 32bit systems we need to ensure that "arg->vec_len" fits in an unsigned long.
CVE-2024-53108	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Adjust VSDB parser for replay feature At some point, the IEEE ID identification for the replay check in the AMD EDID was added. However, this check causes the following out-of-bounds issues when using KASAN: [27.804016] BUG: KASAN: slab-out-of-bounds in amdgpu_dm_update_freesync_caps+0xefa/0x17a0 [amdgpu] [27.804788] Read of size 1 at addr ffff8881647fdb00 by task systemd-udevvd/383 ... [27.821207] Memory state around the buggy address: [27.821215] ffff8881647fda00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [27.821224] ffff8881647fda80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [27.821234] >ffff8881647fdb00: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [27.821243] ^ [27.821250] ffff8881647fdb80: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc [27.821259] ffff8881647fdc00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [27.821268] ===== This is caused because the ID extraction happens outside of the range of the edid lenght. This commit addresses this issue by considering the amd_vsdb_block size. (cherry picked from commit b7e381b1ccd5e778e3d9c44c669ad38439a861d8)

CVE ID	Description
CVE-2024-53109	In the Linux kernel, the following vulnerability has been resolved: nommu: pass NULL argument to vma_iter_prealloc() When deleting a vma entry from a maple tree, it has to pass NULL to vma_iter_prealloc() in order to calculate internal state of the tree, but it passed a wrong argument. As a result, nommu kernels crashed upon accessing a vma iterator, such as acct_collect() reading the size of vma entries after do_munmap(). This commit fixes this issue by passing a right argument to the preallocation call.
CVE-2024-53110	In the Linux kernel, the following vulnerability has been resolved: vp_vdpa: fix id_table array not null terminated error Allocate one extra virtio_device_id as null terminator, otherwise vdpa_mgmtdev_get_classes() may iterate multiple times and visit undefined memory.
CVE-2024-53111	In the Linux kernel, the following vulnerability has been resolved: mm/mremap: fix address wraparound in move_page_tables() On 32-bit platforms, it is possible for the expression `len + old_addr < old_end` to be false-positive if `len + old_addr` wraps around. `old_addr` is the cursor in the old range up to which page table entries have been moved; so if the operation succeeded, `old_addr` is the `*end*` of the old region, and adding `len` to it can wrap. The overflow causes mremap() to mistakenly believe that PTEs have been copied; the consequence is that mremap() bails out, but doesn't move the PTEs back before the new VMA is unmapped, causing anonymous pages in the region to be lost. So basically if userspace tries to mremap() a private-anon region and hits this bug, mremap() will return an error and the private-anon region's contents appear to have been zeroed. The idea of this check is that `old_end - len` is the original start address, and writing the check that way also makes it easier to read; so fix the check by rearranging the comparison accordingly. (An alternate fix would be to refactor this function by introducing an "orig_old_start" variable or such.) Tested in a VM with a 32-bit X86 kernel; without the patch: `` user@horn:~/big_mremap\$ cat test.c #define _GNU_SOURCE #include <stdlib.h> #include <stdio.h> #include <err.h> #include <sys/mman.h> #define ADDR1 ((void*)0x60000000) #define ADDR2 ((void*)0x10000000) #define SIZE 0x50000000uL int main(void) { unsigned char *p1 = mmap(ADDR1, SIZE, PROT_READ PROT_WRITE, MAP_ANONYMOUS MAP_PRIVATE MAP_FIXED_NOREPLACE, -1, 0); if (p1 == MAP_FAILED) err(1, "mmap 1"); unsigned char *p2 = mmap(ADDR2, SIZE, PROT_NONE, MAP_ANONYMOUS MAP_PRIVATE MAP_FIXED_NOREPLACE, -1, 0); if (p2 == MAP_FAILED) err(1, "mmap 2"); *p1 = 0x41; printf("first char is 0x%02hhx\n", *p1); unsigned char *p3 = mremap(p1, SIZE, SIZE, MREMAP_MAYMOVE MREMAP_FIXED, p2); if (p3 == MAP_FAILED) { printf("mremap() failed; first char is 0x%02hhx\n", *p1); } else { printf("mremap() succeeded; first char is 0x%02hhx\n", *p3); } } user@horn:~/big_mremap\$ gcc -static -o test test.c user@horn:~/big_mremap\$ setarch -R ./test first char is 0x41 mremap() failed; first char is 0x00 `` With the patch: `` user@horn:~/big_mremap\$ setarch -R ./test first char is 0x41 mremap() succeeded; first char is 0x41 ``
CVE-2024-53112	In the Linux kernel, the following vulnerability has been resolved: ocfs2: uncache inode which has failed entering the group Syzbot has reported the following BUG: kernel BUG at fs/ocfs2/uptodate.c:509! ... Call Trace: <TASK> ? __die_body+0x5f/0xb0 ? die+0x9e/0xc0 ? do_trap+0x15a/0x3a0 ? ocfs2_set_new_buffer_uptodate+0x145/0x160 ? do_error_trap+0x1dc/0x2c0 ? ocfs2_set_new_buffer_uptodate+0x145/0x160 ? __pfx_do_error_trap+0x10/0x10 ? handle_invalid_op+0x34/0x40 ? ocfs2_set_new_buffer_uptodate+0x145/0x160 ? exc_invalid_op+0x38/0x50 ? asm_exc_invalid_op+0x1a/0x20 ? ocfs2_set_new_buffer_uptodate+0x2e/0x160 ? ocfs2_set_new_buffer_uptodate+0x144/0x160 ? ocfs2_set_new_buffer_uptodate+0x145/0x160 ocfs2_group_add+0x39f/0x15a0 ? __pfx_ocfs2_group_add+0x10/0x10 ? __pfx_lock_acquire+0x10/0x10 ? mnt_get_write_access+0x68/0x2b0 ? __pfx_lock_release+0x10/0x10 ? rcu_read_lock_any_held+0xb7/0x160 ? __pfx_rcu_read_lock_any_held+0x10/0x10 ? smack_log+0x123/0x540 ? mnt_get_write_access+0x68/0x2b0 ? mnt_get_write_access+0x68/0x2b0 ? mnt_get_write_access+0x226/0x2b0 ocfs2_ioctl+0x65e/0x7d0 ? __pfx_ocfs2_ioctl+0x10/0x10 ? smack_file_ioctl+0x29e/0x3a0 ? __pfx_smack_file_ioctl+0x10/0x10 ? lockdep_hardirqs_on_prepare+0x43d/0x780 ? __pfx_lockdep_hardirqs_on_prepare+0x10/0x10 ? __pfx_ocfs2_ioctl+0x10/0x10 __se_sys_ioctl+0xfb/0x170 do_syscall_64+0xf3/0x230 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... </TASK> When 'ioctl(OCFS2_IOC_GROUP_ADD, ...)' has failed for the particular inode in 'ocfs2_verify_group_and_input()', corresponding buffer head remains cached and subsequent call to the same 'ioctl()' for the same inode issues the BUG() in 'ocfs2_set_new_buffer_uptodate()' (trying to cache the same buffer head of that inode). Fix this by uncaching the buffer head with 'ocfs2_remove_from_cache()' on error path in 'ocfs2_group_add()'.

CVE ID	Description
CVE-2024-53113	In the Linux kernel, the following vulnerability has been resolved: mm: fix NULL pointer dereference in alloc_pages_bulk_noprof We triggered a NULL pointer dereference for ac.preferred_zoneref->zone in alloc_pages_bulk_noprof() when the task is migrated between cpusets. When cpuset is enabled, in prepare_alloc_pages(), ac->nodemask may be &current->mems_allowed. when first_zones_zonelist() is called to find preferred_zoneref, the ac->nodemask may be modified concurrently if the task is migrated between different cpusets. Assuming we have 2 NUMA Node, when traversing Node1 in ac->zonelist, the nodemask is 2, and when traversing Node2 in ac->zonelist, the nodemask is 1. As a result, the ac->preferred_zoneref points to NULL zone. In alloc_pages_bulk_noprof(), for_each_zone_zonelist_nodemask() finds a allowable zone and calls zonelist_node_idx(ac.preferred_zoneref), leading to NULL pointer dereference. __alloc_pages_noprof() fixes this issue by checking NULL pointer in commit ea57485af8f4 ("mm, page_alloc: fix check for NULL preferred_zone") and commit df76cee6bbeb ("mm, page_alloc: remove redundant checks from alloc fastpath"). To fix it, check NULL pointer for preferred_zoneref->zone.
CVE-2024-53114	In the Linux kernel, the following vulnerability has been resolved: x86/CPU/AMD: Clear virtualized VMLOAD/VMSAVE on Zen4 client A number of Zen4 client SoCs advertise the ability to use virtualized VMLOAD/VMSAVE, but using these instructions is reported to be a cause of a random host reboot. These instructions aren't intended to be advertised on Zen4 client so clear the capability.
CVE-2024-53115	In the Linux kernel, the following vulnerability has been resolved: drm/vmwgfx: avoid null_ptr_deref in vmw_framebuffer_surface_create_handle The 'vmw_user_object_buffer' function may return NULL with incorrect inputs. To avoid possible null pointer dereference, add a check whether the 'bo' is NULL in the vmw_framebuffer_surface_create_handle.
CVE-2024-53117	In the Linux kernel, the following vulnerability has been resolved: virtio/vsock: Improve MSG_ZEROCOPY error handling Add a missing kfree_skb() to prevent memory leaks.
CVE-2024-53118	In the Linux kernel, the following vulnerability has been resolved: vsock: Fix sk_error_queue memory leak Kernel queues MSG_ZEROCOPY completion notifications on the error queue. Where they remain, until explicitly rcv(ed). To prevent memory leaks, clean up the queue when the socket is destroyed. unreferenced object 0xffff8881028beb00 (size 224): comm "vsock_test", pid 1218, jiffies 4294694897 hex dump (first 32 bytes): 90 b0 21 17 81 88 ff ff 90 b0 21 17 81 88 ff ff ..!.....!..... 00 00 00 00 00 00 00 00 00 21 17 81 88 ff ff!..... backtrace (crc 6c7031ca): [<fffffff81418ef7>] kmem_cache_alloc_node_noprof+0x2f7/0x370 [<fffffff81d35882>] __alloc_skb+0x132/0x180 [<fffffff81d2d32b>] sock_omalloc+0x4b/0x80 [<fffffff81d3a8ae>] msg_zero_copy_realloc+0x9e/0x240 [<fffffff81fe5cb2>] virtio_transport_send_pkt_info+0x412/0x4c0 [<fffffff81fe6183>] virtio_transport_stream_enqueue+0x43/0x50 [<fffffff81fe0813>] vsock_connectible_sendmsg+0x373/0x450 [<fffffff81d233d5>] ____sys_sendmsg+0x365/0x3a0 [<fffffff81d246f4>] ____sys_sendmsg+0x84/0xd0 [<fffffff81d26f47>] __sys_sendmsg+0x47/0x80 [<fffffff820d3df3>] do_syscall_64+0x93/0x180 [<fffffff8220012b>] entry_SYSCALL_64_after_hwframe+0x76/0xe
CVE-2024-53119	In the Linux kernel, the following vulnerability has been resolved: virtio/vsock: Fix accept_queue memory leak As the final stages of socket destruction may be delayed, it is possible that virtio_transport_rcv_listen() will be called after the accept_queue has been flushed, but before the SOCK_DONE flag has been set. As a result, sockets enqueued after the flush would remain unremoved, leading to a memory leak. vsock_release __vsock_release lock virtio_transport_release virtio_transport_close schedule_delayed_work(close_work) sk_shutdown = SHUTDOWN_MASK (!) flush accept_queue release virtio_transport_rcv_pkt vsock_find_bound_socket lock if flag(SOCK_DONE) return virtio_transport_rcv_listen child = vsock_create_connected (!) vsock_enqueue_accept(child) release close_work lock virtio_transport_do_close set_flag(SOCK_DONE) virtio_transport_remove_sock vsock_remove_sock vsock_remove_bound release Introduce a sk_shutdown check to disallow vsock_enqueue_accept() during socket destruction. unreferenced object 0xffff888109e3f800 (size 2040): comm "kworker/5:2", pid 371, jiffies 4294940105 hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 28 00 0b 40 00 00 00 00 00 00 00 00 00 00 00 00 (...) backtrace (crc 9e5f4e84): [<fffffff81418ff1>] kmem_cache_alloc_noprof+0x2c1/0x360 [<fffffff81d27aa0>] sk_prot_alloc+0x30/0x120 [<fffffff81d2b54c>] sk_alloc+0x2c/0x4b0 [<fffffff81fe049a>] __vsock_create.constprop.0+0x2a/0x310 [<fffffff81fe6dc6>] virtio_transport_rcv_pkt+0x4dc/0x9a0 [<fffffff81fe745d>] vsock_loopback_work+0xfd/0x140 [<fffffff810fc6ac>] process_one_work+0x20c/0x570 [<fffffff810fce3f>] worker_thread+0x1bf/0x3a0 [<fffffff811070dd>] kthread+0xdd/0x110 [<fffffff81044fd>] ret_from_fork+0x2d/0x50 [<fffffff8100785a>] ret_from_fork_asm+0x1a/0x30

CVE ID	Description
CVE-2024-53120	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: CT: Fix null-ptr-deref in add rule err flow In error flow of mlx5_tc_ct_entry_add_rule(), in case ct_rule_add() callback returns error, zone_rule->attr is used uninitialized. Fix it to use attr which has the needed pointer value. Kernel log: BUG: kernel NULL pointer dereference, address: 0000000000000110 RIP: 0010:mlx5_tc_ct_entry_add_rule+0x2b1/0x2f0 [mlx5_core] ... Call Trace: <TASK> ? __die+0x20/0x70 ? page_fault_oops+0x150/0x3e0 ? exc_page_fault+0x74/0x140 ? asm_exc_page_fault+0x22/0x30 ? mlx5_tc_ct_entry_add_rule+0x2b1/0x2f0 [mlx5_core] ? mlx5_tc_ct_entry_add_rule+0x1d5/0x2f0 [mlx5_core] mlx5_tc_ct_block_flow_offload+0xc6a/0xf90 [mlx5_core] ? nf_flow_offload_tuple+0xd8/0x190 [nf_flow_table] nf_flow_offload_tuple+0xd8/0x190 [nf_flow_table] flow_offload_work_handler+0x142/0x320 [nf_flow_table] ? finish_task_switch.isra.0+0x15b/0x2b0 process_one_work+0x16c/0x320 worker_thread+0x28c/0x3a0 ? __pfx_worker_thread+0x10/0x10 kthread+0xb8/0xf0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2d/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK>
CVE-2024-53121	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: fs, lock FTE when checking if active The referenced commits introduced a two-step process for deleting FTEs: - Lock the FTE, delete it from hardware, set the hardware deletion function to NULL and unlock the FTE. - Lock the parent flow group, delete the software copy of the FTE, and remove it from the xarray. However, this approach encounters a race condition if a rule with the same match value is added simultaneously. In this scenario, fs_core may set the hardware deletion function to NULL prematurely, causing a panic during subsequent rule deletions. To prevent this, ensure the active flag of the FTE is checked under a lock, which will prevent the fs_core layer from attaching a new steering rule to an FTE that is in the process of deletion. [438.967589] MOSHE: 2496 mlx5_del_flow_rules del_hw_func [438.968205] ----- [cut here]----- [438.968654] refcount_t: decrement hit 0; leaking memory. [438.969249] WARNING: CPU: 0 PID: 8957 at lib/refcount.c:31 refcount_warn_saturate+0xfb/0x110 [438.970054] Modules linked in: act_mirred cls_flowlet act_gact sch_ingress openvswitch nsh mlx5_vdpa vringh vhost_iotlb vdpa mlx5_ib mlx5_core xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter rpcsec_gss_krb5 auth_rpcgss oid_registry overlay rprcdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi ib_umad rdma_cm ib_ipoib iw_cm ib_cm ib_uverbs ib_core zram zsmalloc fuse [last unloaded: cls_flowlet] [438.973288] CPU: 0 UID: 0 PID: 8957 Comm: tc Not tainted 6.12.0-rc1+ #8 [438.973888] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 [438.974874] RIP: 0010:refcount_warn_saturate+0xfb/0x110 [438.975363] Code: 40 66 3b 82 c6 05 16 e9 4d 01 01 e8 1f 7c a0 ff 0f 0b c3 cc cc cc 48 c7 c7 10 66 3b 82 c6 05 fd e8 4d 01 01 e8 05 7c a0 ff <0f> 0b c3 cc cc cc cc 66 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 00 90 [438.976947] RSP: 0018:ffff888124a53610 EFLAGS: 00010286 [438.977446] RAX: 0000000000000000 RBX: ffff888119d56de0 RCX: 0000000000000000 [438.978090] RDX: ffff88852c828700 RSI: ffff88852c81b3c0 RDI: ffff88852c81b3c0 [438.978721] RBP: ffff888120fa0e88 R08: 0000000000000000 R09: ffff888124a534b0 [438.979353] R10: 0000000000000001 R11: 0000000000000001 R12: ffff888119d56de0 [438.979979] R13: ffff888120fa0ec0 R14: ffff888120fa0ee8 R15: ffff888119d56de0 [438.980607] FS: 00007fe6dccc0f800(0000) GS:ffff88852c800000(0000) knlGS:0000000000000000 [438.983984] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [438.984544] CR2: 00000000004275e0 CR3: 0000000186982001 CR4: 0000000000372eb0 [438.985205] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [438.985842] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [438.986507] Call Trace: [438.986799] <TASK> [438.987070] ? __warn+0x7d/0x110 [438.987426] ? refcount_warn_saturate+0xfb/0x110 [438.987877] ? report_bug+0x17d/0x190 [438.988261] ? prb_read_valid+0x17/0x20 [438.988659] ? handle_bug+0x53/0x90 [438.989054] ? exc_invalid_op+0x14/0x70 [438.989458] ? asm_exc_invalid_op+0x16/0x20 [438.989883] ? refcount_warn_saturate+0xfb/0x110 [438.990348] mlx5_del_flow_rules+0x2f7/0x340 [mlx5_core] [438.990932] __mlx5_eswitch_del_rule+0x49/0x170 [mlx5_core] [438.991519] ? mlx5_lag_is_sriov+0x3c/0x50 [mlx5_core] [438.992054] ? xas_load+0x9/0xb0 [438.992407] mlx5e_tc_rule_unoffload+0x45/0xe0 [mlx5_core] [438.993037] mlx5e_tc_del_fdb_flow+0x2a6/0x2e0 [mlx5_core] [438.993623] mlx5e_flow_put+0x29/0x60 [mlx5_core] [438.994161] mlx5e_delete_flowlet+0x261/0x390 [mlx5_core] [438.994728] tc_setup_cb_destroy+0xb9/0x190 [438.995150] fl_hw_destroy_filter+0x94/0xc0 [cls_flowlet] [438.995650] fl_change+0x11a4/0x13c0 [cls_flowlet] [438.996105] tc_new_tfilter+0x347/0xabc0 [438.996503] ? __---truncated---
CVE-2024-53122	In the Linux kernel, the following vulnerability has been resolved: mptcp: cope racing subflow creation in mptcp_rcv_space_adjust Additional active subflows - i.e. created by the in kernel path manager - are included into the subflow list before starting the 3whs. A racing recvmsg() spooling data received on an already established subflow would unconditionally call tcp_cleanup_rbuf() on all the current subflows, potentially hitting a divide by zero error on the newly created ones. Explicitly check that the subflow is in a suitable state before invoking tcp_cleanup_rbuf().

CVE ID	Description
CVE-2024-53123	In the Linux kernel, the following vulnerability has been resolved: mptcp: error out earlier on disconnect Eric reported a division by zero splat in the MPTCP protocol: Oops: divide error: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 1 UID: 0 PID: 6094 Comm: syz-executor317 Not tainted 6.12.0-rc5-syzkaller-00291-g05b92660cdf #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010: __tcp_select_window +0x5b4/0x1310 net/ipv4/tcp_output.c:3163 Code: f6 44 01 e3 89 df e8 9b 75 09 f8 44 39 f3 0f 8d 11 ff ff ff e8 0d 74 09 f8 45 89 f4 e9 04 ff ff ff e8 00 74 09 f8 44 89 f0 99 <f7> 7c 24 14 41 29 d6 45 89 f4 e9 ec fe ff ff e8 73 09 f8 48 89 RSP: 0018:ffffc900041f7930 EFLAGS: 00010293 RAX: 0000000000017e67 RBX: 0000000000017e67 RCX: ffffffff8983314b RDX: 0000000000000000 RSI: ffffffff898331b0 RDI: 0000000000000004 RBP: 00000000005d6000 R08: 0000000000000004 R09: 000000000000017e67 R10: 00000000000003e8 R11: 0000000000000000 R12: 00000000000003e8 R13: ffff888031d9b440 R14: 0000000000017e67 R15: 00000000002eb000 FS: 00007feb5d7f16c0(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 0000000008005003 CR2: 00007feb5d8adbb8 CR3: 0000000074e4c000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffffe0ff DR7: 0000000000000400 Call Trace: <TASK> __tcp_cleanup_rbuf +0x3e7/0x4b0 net/ipv4/tcp.c:1493 mptcp_rcv_space_adjust net/mptcp/protocol.c:2085 [inline] mptcp_rcvmsg+0x2156/0x2600 net/mptcp/protocol.c:2289 inet_rcvmsg+0x469/0x6a0 net/ipv4/af_inet.c:885 sock_rcvmsg_nosec net/socket.c:1051 [inline] sock_rcvmsg+0x1b2/0x250 net/socket.c:1073 __sys_recvfrom+0x1a5/0x2e0 net/socket.c:2265 __do_sys_recvfrom net/socket.c:2283 [inline] __se_sys_recvfrom net/socket.c:2279 [inline] __x64_sys_recvfrom+0xe0/0x1c0 net/socket.c:2279 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7feb5d857559 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 51 18 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007feb5d7f1208 EFLAGS: 00000246 ORIG_RAX: 000000000000002d RAX: ffffffff81000000da RBX: 00007feb5d8e1318 RCX: 00007feb5d857559 RDX: 000000800000000e RSI: 0000000000000000 RDI: 0000000000000003 RBP: 00007feb5d8e1310 R08: 0000000000000000 R09: ffffffff81000000 R10: 0000000000000100 R11: 0000000000000246 R12: 00007feb5d8e131c R13: 00007feb5d8ae074 R14: 000000800000000e R15: 00000000ffffffdef and provided a nice reproducer. The root cause is the current bad handling of racing disconnect. After the blamed commit below, sk_wait_data() can return (with error) with the underlying socket disconnected and a zero rcv_mss. Catch the error and return without performing any additional operations on the current socket.

CVE ID	Description
CVE-2024-53124	In the Linux kernel, the following vulnerability has been resolved: net: fix data-races around sk->sk_forward_alloc Syzkaller reported this warning: -----[cut here]----- WARNING: CPU: 0 PID: 16 at net/ipv4/af_inet.c:156 inet_sock_destruct+0x1c5/0x1e0 Modules linked in: CPU: 0 UID: 0 PID: 16 Comm: ksoftirqd/0 Not tainted 6.12.0-rc5 #26 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 RIP: 0010:inet_sock_destruct+0x1c5/0x1e0 Code: 24 12 4c 89 e2 5b 48 c7 c7 98 ec bb 82 41 5c e9 d1 18 17 ff 4c 89 e6 5b 48 c7 c7 d0 ec bb 82 41 5c e9 bf 18 17 ff 0f 0b eb 83 <0f> 0b eb 97 0f 0b eb 87 0f 0b e9 68 ff ff ff 66 66 2e 0f 1f 84 00 RSP: 0018:ffffc9000008bd90 EFLAGS: 00010206 RAX: 0000000000000300 RBX: ffff88810b172a90 RCX: 0000000000000007 RDX: 0000000000000002 RSI: 0000000000000300 RDI: ffff88810b172a00 RBP: ffff88810b172a00 R08: ffff888104273c00 R09: 0000000001000007 R10: 0000000000002000 R11: 0000000000000006 R12: ffff88810b172a00 R13: 0000000000000004 R14: 0000000000000000 R15: ffff888237c31f78 FS: 0000000000000000(0000) GS:ffff888237c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffc63fecac8 CR3: 000000000342e000 CR4: 00000000000006f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __warn+0x88/0x130 ? inet_sock_destruct+0x1c5/0x1e0 ? report_bug+0x18e/0x1a0 ? handle_bug+0x53/0x90 ? exc_invalid_op+0x18/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? inet_sock_destruct+0x1c5/0x1e0 __sk_destruct+0x2a/0x200 rcu_do_batch+0x1aa/0x530 ? rcu_do_batch+0x13b/0x530 rcu_core+0x159/0x2f0 handle_softirqs+0xd3/0x2b0 ? __pfx_smpboot_thread_fn+0x10/0x10 run_ksoftirqd+0x25/0x30 smpboot_thread_fn+0xdd/0x1d0 kthread+0xd3/0x100 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x34/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> ---[end trace 0000000000000000]--- Its possible that two threads call tcp_v6_do_rcv()/sk_forward_alloc_add() concurrently when sk->sk_state == TCP_LISTEN with sk->sk_lock unlocked, which triggers a data-race around sk->sk_forward_alloc: tcp_v6_rcv tcp_v6_do_rcv skb_clone_and_charge_r sk_rmem_schedule __sk_mem_schedule sk_forward_alloc_add() skb_set_owner_r sk_mem_charge sk_forward_alloc_add() __kfree_skb skb_release_all skb_release_head_state sock_rfree sk_mem_uncharge sk_forward_alloc_add() sk_mem_reclaim // set local var reclaimable __sk_mem_reclaim sk_forward_alloc_add() In this syzkaller testcase, two threads call tcp_v6_do_rcv() with skb->truesize=768, the sk_forward_alloc changes like this: (cpu 1) (cpu 2) sk_forward_alloc 0 __sk_mem_schedule() +4096 = 4096 __sk_mem_schedule() +4096 = 8192 sk_mem_charge() -768 = 7424 sk_mem_charge() -768 = 6656 sk_mem_uncharge() +768 = 7424 reclaimable=7424 sk_mem_uncharge() +768 = 8192 reclaimable=8192 __sk_mem_reclaim() -4096 = 4096 __sk_mem_reclaim() -8192 = -4096 != 0 The skb_clone_and_charge_r() should not be called in tcp_v6_do_rcv() when sk->sk_state is TCP_LISTEN, it happens later in tcp_v6_syn_recv_sock(). Fix the same issue in dccp_v6_do_rcv().
CVE-2024-53125	In the Linux kernel, the following vulnerability has been resolved: bpf: sync_linked_regs() must preserve subreg_def Range propagation must not affect subreg_def marks, otherwise the following example is rewritten by verifier incorrectly when BPF_F_TEST_RND_HI32 flag is set: 0: call bpf_ktime_get_ns call bpf_ktime_get_ns 1: r0 &= 0x7fffffff after verifier r0 &= 0x7fffffff 2: w1 = w0 rewrites w1 = w0 3: if w0 < 10 goto +0 -----> r11 = 0x2f5674a6 (r) 4: r1 >= 32 r11 <= 32 (r) 5: r0 = r1 r1 = r11 (r) 6: exit; if w0 < 0xa goto pc+0 r1 >= 32 r0 = r1 exit (or zero extension of w1 at (2) is missing for architectures that require zero extension for upper register half). The following happens w/o this patch: - r0 is marked as not a subreg at (0); - w1 is marked as subreg at (2); - w1 subreg_def is overridden at (3) by copy_register_state(); - w1 is read at (5) but mark_insn_zext() does not mark (2) for zero extension, because w1 subreg_def is not set; - because of BPF_F_TEST_RND_HI32 flag verifier inserts random value for hi32 bits of (2) (marked (r)); - this random value is read at (5).
CVE-2024-53126	In the Linux kernel, the following vulnerability has been resolved: vdpa: solidrun: Fix UB bug with devres In psnet_open_pf_bar() and snet_open_vf_bar() a string later passed to pcim_iomap_regions() is placed on the stack. Neither pcim_iomap_regions() nor the functions it calls copy that string. Should the string later ever be used, this, consequently, causes undefined behavior since the stack frame will by then have disappeared. Fix the bug by allocating the strings on the heap through devm_kasprintf().
CVE-2024-53127	In the Linux kernel, the following vulnerability has been resolved: Revert "mmc: dw_mmc: Fix IDMAC operation with pages bigger than 4K" The commit 8396c793ffdf ("mmc: dw_mmc: Fix IDMAC operation with pages bigger than 4K") increased the max_req_size, even for 4K pages, causing various issues: - Panic booting the kernel/rootfs from an SD card on Rockchip RK3566 - Panic booting the kernel/rootfs from an SD card on StarFive JH7100 - "swiotlb buffer is full" and data corruption on StarFive JH7110 At this stage no fix have been found, so it's probably better to just revert the change. This reverts commit 8396c793ffdf28bb8aee7cfe0891080f8cab7890.

CVE ID	Description
CVE-2024-53128	In the Linux kernel, the following vulnerability has been resolved: sched/task_stack: fix object_is_on_stack() for KASAN tagged pointers When CONFIG_KASAN_SW_TAGS and CONFIG_KASAN_STACK are enabled, the object_is_on_stack() function may produce incorrect results due to the presence of tags in the obj pointer, while the stack pointer does not have tags. This discrepancy can lead to incorrect stack object detection and subsequently trigger warnings if CONFIG_DEBUG_OBJECTS is also enabled. Example of the warning: ODEBUG: object 3eff800082ea7bb0 is NOT on stack ffff800082ea0000, but annotated. -----[cut here]----- WARNING: CPU: 0 PID: 1 at lib/debugobjects.c:557 __debug_object_init+0x330/0x364 Modules linked in: CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.12.0-rc5 #4 Hardware name: linux,dummy-virt (DT) pstate: 600000c5 (nZCv daIf -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : __debug_object_init+0x330/0x364 lr : __debug_object_init+0x330/0x364 sp : ffff800082ea7b40 x29: ffff800082ea7b40 x28: 98ff0000c0164518 x27: 98ff0000c0164534 x26: ffff800082d93ec8 x25: 0000000000000001 x24: 1cff0000c00172a0 x23: 0000000000000000 x22: ffff800082d93ed0 x21: ffff800081a24418 x20: 3eff800082ea7bb0 x19: efff800000000000 x18: 0000000000000000 x17: 00000000000000ff x16: 0000000000000047 x15: 206b63617473206e x14: 0000000000000018 x13: ffff800082ea7780 x12: 0fff800082ea78e x11: 0fff800082ea790 x10: 0fff800082ea79d x9 : 34d77febe173e800 x8 : 34d77febe173e800 x7 : 0000000000000001 x6 : 0000000000000001 x5 : feff800082ea74b8 x4 : ffff800082870a90 x3 : ffff80008018d3c4 x2 : 0000000000000001 x1 : ffff800082858810 x0 : 0000000000000050 Call trace: __debug_object_init+0x330/0x364 debug_object_init_on_stack+0x30/0x3c schedule_hrtimeout_range_clock+0xac/0x26c schedule_hrtimeout+0x1c/0x30 wait_task_inactive+0x1d4/0x25c kthread_bind_mask+0x28/0x98 init_rescuer+0x1e8/0x280 workqueue_init+0x1a0/0x3cc kernel_init_freeable+0x118/0x200 kernel_init+0x28/0x1f0 ret_from_fork+0x10/0x20 ---[end trace 0000000000000000]--- ODEBUG: object 3eff800082ea7bb0 is NOT on stack ffff800082ea0000, but annotated. -----[cut here]-----
CVE-2024-53129	In the Linux kernel, the following vulnerability has been resolved: drm/rockchip: vop: Fix a dereferenced before check warning The 'state' can't be NULL, we should check crt_state. Fix warning: drivers/gpu/drm/rockchip/rockchip_drm_vop.c:1096 vop_plane_atomic_async_check() warn: variable dereferenced before check 'state' (see line 1077)
CVE-2024-53130	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix null-ptr-deref in block_dirty_buffer tracepoint When using the "block:block_dirty_buffer" tracepoint, mark_buffer_dirty() may cause a NULL pointer dereference, or a general protection fault when KASAN is enabled. This happens because, since the tracepoint was added in mark_buffer_dirty(), it references the dev_t member bh->b_bdev->bd_dev regardless of whether the buffer head has a pointer to a block_device structure. In the current implementation, nilfs_grab_buffer(), which grabs a buffer to read (or create) a block of metadata, including b-tree node blocks, does not set the block device, but instead does so only if the buffer is not in the "uptodate" state for each of its caller block reading functions. However, if the uptodate flag is set on a folio/page, and the buffer heads are detached from it by try_to_free_buffers(), and new buffer heads are then attached by create_empty_buffers(), the uptodate flag may be restored to each buffer without the block device being set to bh->b_bdev, and mark_buffer_dirty() may be called later in that state, resulting in the bug mentioned above. Fix this issue by making nilfs_grab_buffer() always set the block device of the super block structure to the buffer head, regardless of the state of the buffer's uptodate flag.
CVE-2024-53131	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix null-ptr-deref in block_touch_buffer tracepoint Patch series "nilfs2: fix null-ptr-deref bugs on block tracepoints". This series fixes null pointer dereference bugs that occur when using nilfs2 and two block-related tracepoints. This patch (of 2): It has been reported that when using "block:block_touch_buffer" tracepoint, touch_buffer() called from __nilfs_get_folio_block() causes a NULL pointer dereference, or a general protection fault when KASAN is enabled. This happens because since the tracepoint was added in touch_buffer(), it references the dev_t member bh->b_bdev->bd_dev regardless of whether the buffer head has a pointer to a block_device structure. In the current implementation, the block_device structure is set after the function returns to the caller. Here, touch_buffer() is used to mark the folio/page that owns the buffer head as accessed, but the common search helper for folio/page used by the caller function was optimized to mark the folio/page as accessed when it was reimplemented a long time ago, eliminating the need to call touch_buffer() here in the first place. So this solves the issue by eliminating the touch_buffer() call itself.
CVE-2024-53133	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Handle dml allocation failure to avoid crash [Why] In the case where a dml allocation fails for any reason, the current state's dml contexts would no longer be valid. Then subsequent calls dc_state_copy_internal would shallow copy invalid memory and if the new state was released, a double free would occur. [How] Reset dml pointers in new_state to NULL and avoid invalid pointer (cherry picked from commit bcafdc61529a48f6f06355d78eb41b3aeda5296c)

CVE ID	Description
CVE-2024-53134	In the Linux kernel, the following vulnerability has been resolved: pmdomain: imx93-blk-ctrl: correct remove path The check condition should be 'i < bc->onecell_data.num_domains', not 'bc->onecell_data.num_domains' which will make the look never finish and cause kernel panic. Also disable runtime to address "imx93-blk-ctrl 4ac10000.system-controller: Unbalanced pm_runtime_enable!"
CVE-2024-53135	In the Linux kernel, the following vulnerability has been resolved: KVM: VMX: Bury Intel PT virtualization (guest/host mode) behind CONFIG_BROKEN Hide KVM's pt_mode module param behind CONFIG_BROKEN, i.e. disable support for virtualizing Intel PT via guest/host mode unless BROKEN=y. There are myriad bugs in the implementation, some of which are fatal to the guest, and others which put the stability and health of the host at risk. For guest fatalities, the most glaring issue is that KVM fails to ensure tracing is disabled, and *stays* disabled prior to VM-Enter, which is necessary as hardware disallows loading (the guest's) RTIT_CTL if tracing is enabled (enforced via a VMX consistency check). Per the SDM: If the logical processor is operating with Intel PT enabled (if IA32_RTIT_CTL.TraceEn = 1) at the time of VM entry, the "load IA32_RTIT_CTL" VM-entry control must be 0. On the host side, KVM doesn't validate the guest CPUID configuration provided by userspace, and even worse, uses the guest configuration to decide what MSRs to save/load at VM-Enter and VM-Exit. E.g. configuring guest CPUID to enumerate more address ranges than are supported in hardware will result in KVM trying to passthrough, save, and load non-existent MSRs, which generates a variety of WARNs, ToPA ERRORS in the host, a potential deadlock, etc.
CVE-2024-53136	In the Linux kernel, the following vulnerability has been resolved: mm: revert "mm: shmem: fix data-race in shmem_getattr()" Revert d949d1d14fa2 ("mm: shmem: fix data-race in shmem_getattr()") as suggested by Chuck [1]. It is causing deadlocks when accessing tmpfs over NFS. As Hugh commented, "added just to silence a syzbot sanitizer splat: added where there has never been any practical problem".
CVE-2024-53138	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: kTLS, Fix incorrect page refcounting The kTLS tx handling code is using a mix of get_page() and page_ref_inc() APIs to increment the page reference. But on the release path (mlx5e_ktls_tx_handle_resync_dump_comp()), only put_page() is used. This is an issue when using pages from large folios: the get_page() references are stored on the folio page while the page_ref_inc() references are stored directly in the given page. On release the folio page will be dereferenced too many times. This was found while doing kTLS testing with sendfile() + ZC when the served file was read from NFS on a kernel with NFS large folios support (commit 49b29a573da8 ("nfs: add support for large folios")).

CVE ID	Description
CVE-2024-53139	In the Linux kernel, the following vulnerability has been resolved: sctp: fix possible UAF in sctp_v6_available() A lockdep report [1] with CONFIG_PROVE_RCU_LIST=y hints that sctp_v6_available() is calling dev_get_by_index_rcu() and ipv6_chk_addr() without holding rcu. [1] ===== WARNING: suspicious RCU usage 6.12.0-rc5-virtme #1216 Tainted: G W ----- net/core/dev.c:876 RCU-list traversed in non-reader section!! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by sctp_hello/31495: #0: ffff9f1ebdb7418 (sk_lock-AF_INET6){+.+.}-{0:0}, at: sctp_bind (/arch/x86/include/asm/jump_label.h:27 net/sctp/socket.c:315) sctp stack backtrace: CPU: 7 UID: 0 PID: 31495 Comm: sctp_hello Tainted: G W 6.12.0-rc5-virtme #1216 Tainted: [W]=WARN Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:123) lockdep_rcu_suspicious (kernel/locking/lockdep.c:6822) dev_get_by_index_rcu (net/core/dev.c:876 (discriminator 7)) sctp_v6_available (net/sctp/ipv6.c:701) sctp sctp_do_bind (net/sctp/socket.c:400 (discriminator 1)) sctp sctp_bind (net/sctp/socket.c:320) sctp inet6_bind_sk (net/ipv6/af_inet6.c:465) ? security_socket_bind (security/security.c:4581 (discriminator 1)) __sys_bind (net/socket.c:1848 net/socket.c:1869) ? do_user_addr_fault (/include/linux/rcupdate.h:347 ./include/linux/rcupdate.h:880 ./include/linux/mm.h:729 arch/x86/mm/fault.c:1340) ? do_user_addr_fault (/arch/x86/include/asm/preempt.h:84 (discriminator 13) ./include/linux/rcupdate.h:98 (discriminator 13) ./include/linux/rcupdate.h:882 (discriminator 13) ./include/linux/mm.h:729 (discriminator 13) arch/x86/mm/fault.c:1340 (discriminator 13)) __x64_sys_bind (net/socket.c:1877 (discriminator 1) net/socket.c:1875 (discriminator 1) net/socket.c:1875 (discriminator 1)) do_syscall_64 (arch/x86/entry/common.c:52 (discriminator 1) arch/x86/entry/common.c:83 (discriminator 1)) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) RIP: 0033:0x7f59b934a1e7 Code: 44 00 00 48 b1 15 39 8c 0c 00 f7 d8 64 89 02 b8 ff ff ff ff eb bd 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 00 b8 31 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 09 8c 0c 00 f7 d8 64 89 01 48 All code ===== 0: 44 00 00 add %r8b, (%rax) 3: 48 8b 15 39 8c 0c 00 mov 0xc8c39(%rip), %rdx # 0xc8c43 a: f7 d8 neg %eax c: 64 89 02 mov %eax, %fs:(%rdx) f: b8 ff ff ff ff mov \$0xffffffff, %eax 14: eb bd jmp 0xffffffffffffd3 16: 66 2e 0f 1f 84 00 00 cs nopw 0x0(%rax,%rax,1) 1d: 00 00 00 20: 0f 1f 00 nopl (%rax) 23: b8 31 00 00 00 mov \$0x31, %eax 28: 0f 05 syscall 2a:* 48 3d 01 f0 ff ff cmp \$0xfffffffffffff001, %rax <-- trapping instruction 30: 73 01 jae 0x33 32: c3 ret 33: 48 8b 0d 09 8c 0c 00 mov 0xc8c09(%rip), %rcx # 0xc8c43 3a: f7 d8 neg %eax 3c: 64 89 01 mov %eax, %fs:(%rcx) 3f: 48 rex.W Code starting with the faulting instruction ===== 0: 48 3d 01 f0 ff ff cmp \$0xfffffffffffff001, %rax 6: 73 01 jae 0x9 8: c3 ret 9: 48 8b 0d 09 8c 0c 00 mov 0xc8c09(%rip), %rcx # 0xc8c19 10: f7 d8 neg %eax 12: 64 89 01 mov %eax, %fs:(%rcx) 15: 48 rex.W RSP: 002b:00007ffe2d0ad398 EFLAGS: 00000202 ORIG_RAX: 0000000000000031 RAX: ffffffffddfa RBX: 00007ffe2d0ad3d0 RCX: 00007f59b934a1e7 RDX: 000000000000001c RSI: 00007ffe2d0ad3d0 RDI: 0000000000000005 RBP: 0000000000000005 R08: 1999999999999999 R09: 0000000000000000 R10: 00007f59b9253298 R11: 0000000000000000 ---truncated---
CVE-2024-53140	In the Linux kernel, the following vulnerability has been resolved: netlink: terminate outstanding dump on socket close Netlink supports iterative dumping of data. It provides the families the following ops: - start - (optional) kicks off the dumping process - dump - actual dump helper, keeps getting called until it returns 0 - done - (optional) pairs with .start, can be used for cleanup The whole process is asynchronous and the repeated calls to .dump don't actually happen in a tight loop, but rather are triggered in response to recvmmsg() on the socket. This gives the user full control over the dump, but also means that the user can close the socket without getting to the end of the dump. To make sure .start is always paired with .done we check if there is an ongoing dump before freeing the socket, and if so call .done. The complication is that sockets can get freed from BH and .done is allowed to sleep. So we use a workqueue to defer the call, when needed. Unfortunately this does not work correctly. What we defer is not the cleanup but rather releasing a reference on the socket. We have no guarantee that we own the last reference, if someone else holds the socket they may release it in BH and we're back to square one. The whole dance, however, appears to be unnecessary. Only the user can interact with dumps, so we can clean up when socket is closed. And close always happens in process context. Some async code may still access the socket after close, queue notification skbs to it etc. but no dumps can start, end or otherwise make progress. Delete the workqueue and flush the dump state directly from the release handler. Note that further cleanup is possible in -next, for instance we now always call .done before releasing the main module reference, so dump doesn't have to take a reference of its own.

[illegible]

CVE ID	Description
CVE-2024-53151	In the Linux kernel, the following vulnerability has been resolved: svcrdma: Address an integer overflow Dan Carpenter reports: > Commit 78147ca8b4a9 ("svcrdma: Add a "parsed chunk list" data > structure") from Jun 22, 2020 (linux-next), leads to the following > Smatch static checker warning: > > net/sunrpc/xprtrdma/svc_rdma_recvfrom.c:498 xdr_check_write_chunk() > warn: potential user controlled sizeof overflow 'segcount * 4 * 4' > > net/sunrpc/xprtrdma/svc_rdma_recvfrom.c > 488 static bool xdr_check_write_chunk(struct svc_rdma_recv_ctxt *rctx) > 489 { > 490 u32 segcount; > 491 __be32 *p; > 492 > 493 if (xdr_stream_decode_u32(&rctx->rc_stream, &segcount)) > ^^^^^^ > > 494 return false; > 495 > 496 /* A bogus segcount causes this buffer overflow check to fail. */ > 497 p = xdr_inline_decode(&rctx->rc_stream, > --> 498 segcount * rpcrdma_segment_maxsz * sizeof(*p)); > > > segcount is an untrusted u32. On 32bit systems anything >= SIZE_MAX / 16 will > have an integer overflow and some those values will be accepted by > xdr_inline_decode().
CVE-2024-53154	In the Linux kernel, the following vulnerability has been resolved: clk: clk-apple-nco: Add NULL check in applnco_probe Add NULL check in applnco_probe, to handle kernel NULL pointer dereference error.
CVE-2024-53155	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix uninitialized value in ocfs2_file_read_iter() Syzbot has reported the following KMSAN splat: BUG: KMSAN: uninit-value in ocfs2_file_read_iter+0x9a4/0xf80 ocfs2_file_read_iter+0x9a4/0xf80 __io_read+0x8d4/0x20f0 io_read+0x3e/0xf0 io_issue_sqe+0x42b/0x22c0 io_wq_submit_work+0xaf9/0xdc0 io_worker_handle_work+0xd13/0x2110 io_wq_worker+0x447/0x1410 ret_from_fork+0x6f/0x90 ret_from_fork_asm+0x1a/0x30 Uninit was created at: __alloc_pages_noprof+0x9a7/0xe00 alloc_pages_mpol_noprof+0x299/0x990 alloc_pages_noprof+0x1bf/0x1e0 allocate_slab+0x33a/0x1250 __slab_alloc+0x12ef/0x35e0 kmem_cache_alloc_bulk_noprof+0x486/0x1330 __io_alloc_req_refill+0x84/0x560 io_submit_sqes+0x172f/0x2f30 __se_sys_io_uring_enter+0x406/0x41c0 __x64_sys_io_uring_enter+0x11f/0x1a0 x64_sys_call+0x2b54/0x3ba0 do_syscall_64+0xcd/0x1e0 entry_SYSCALL_64_after_hwframe+0x77/0xf7 Since an instance of 'struct kiocb' may be passed from the block layer with 'private' field uninitialized, introduce 'ocfs2_iocb_init_rw_locked()' and use it from where 'ocfs2_dio_end_io()' might take care, i.e. in 'ocfs2_file_read_iter()' and 'ocfs2_file_write_iter()'.
CVE-2024-53156	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k: add range check for conn_rsp_epid in htc_connect_service() I found the following bug in my fuzzer: UBSAN: array-index-out-of-bounds in drivers/net/wireless/ath/ath9k/htc_hst.c:26:51 index 255 is out of range for type 'htc_endpoint [22]' CPU: 0 UID: 0 PID: 8 Comm: kworker/0:0 Not tainted 6.11.0-rc6-dirty #14 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Workqueue: events request_firmware_work_func Call Trace: <TASK> dump_stack_lvl+0x180/0x1b0 __ubsan_handle_out_of_bounds+0xd4/0x130 htc_issue_send.constprop.0+0x20c/0x230 ? _raw_spin_unlock_irqrestore+0x3c/0x70 ath9k_wmi_cmd+0x41d/0x610 ? mark_held_locks+0x9f/0xe0 ... Since this bug has been confirmed to be caused by insufficient verification of conn_rsp_epid, I think it would be appropriate to add a range check for conn_rsp_epid to htc_connect_service() to prevent the bug from occurring.
CVE-2024-53157	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scp: Check the DVFS OPP count returned by the firmware Fix a kernel crash with the below call trace when the SCPI firmware returns OPP count of zero. dvfs_info.opp_count may be zero on some platforms during the reboot test, and the kernel will crash after dereferencing the pointer to kcalloc(info->count, sizeof(*opp), GFP_KERNEL). Unable to handle kernel NULL pointer dereference at virtual address 0000000000000028 Mem abort info: ESR = 0x96000004 Exception class = DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 Data abort info: ISV = 0, ISS = 0x00000004 CM = 0, WnR = 0 user pgtable: 4k pages, 48-bit VAs, pgdp = 00000000faefa08c [0000000000000028] pgd=0000000000000000 Internal error: Oops: 96000004 [#1] SMP scpi-hwmon: probe of PHYT000D:00 failed with error -110 Process systemd-udev (pid: 1701, stack limit = 0x00000000aaede86c) CPU: 2 PID: 1701 Comm: systemd-udev Not tainted 4.19.90+ #1 Hardware name: PHYTIUM LTD Phytium FT2000/4/Phytium FT2000/4, BIOS pstate: 60000005 (nZCv daif -PAN -UAO) pc : scpi_dvfs_recalc_rate+0x40/0x58 [clk_scp] lr : clk_register+0x438/0x720 Call trace: scpi_dvfs_recalc_rate+0x40/0x58 [clk_scp] devm_clk_hw_register+0x50/0xa0 scpi_clk_ops_init.isra.2+0xa0/0x138 [clk_scp] scpi_clocks_probe+0x528/0x70c [clk_scp] platform_drv_probe+0x58/0xa8 really_probe+0x260/0x3d0 driver_probe_device+0x12c/0x148 device_driver_attach+0x74/0x98 __driver_attach+0xb4/0xe8 bus_for_each_dev+0x88/0xe0 driver_attach+0x30/0x40 bus_add_driver+0x178/0x2b0 driver_register+0x64/0x118 __platform_driver_register+0x54/0x60 scpi_clocks_driver_init+0x24/0x1000 [clk_scp] do_one_initcall+0x54/0x220 do_init_module+0x54/0x1c8 load_module+0x14a4/0x1668 __se_sys_finit_module+0xf8/0x110 __arm64_sys_finit_module+0x24/0x30 el0_svc_common+0x78/0x170 el0_svc_handler+0x38/0x78 el0_svc+0x8/0x340 Code: 937d7c00 a94153f3 a8c27bfd f9400421 (b8606820) ---[end trace 06feb22469d89fa8]--- Kernel panic - not syncing: Fatal exception SMP: stopping secondary CPUs Kernel Offset: disabled CPU features: 0x10,a0002008 Memory Limit: none

CVE ID	Description
CVE-2024-53158	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: geni-se: fix array underflow in geni_se_clk_tbl_get() This loop is supposed to break if the frequency returned from clk_round_rate() is the same as on the previous iteration. However, that check doesn't make sense on the first iteration through the loop. It leads to reading before the start of these->clk_perf_tbl[] array.
CVE-2024-53160	In the Linux kernel, the following vulnerability has been resolved: rcu/kvfree: Fix data-race in __mod_timer / kvfree_call_rcu KCSAN reports a data race when access the krcp->monitor_work.timer.expires variable in the schedule_delayed_monitor_work() function: <snip> BUG: KCSAN: data-race in __mod_timer / kvfree_call_rcu read to 0xffff888237d1cce8 of 8 bytes by task 10149 on cpu 1: schedule_delayed_monitor_work kernel/rcu/tree.c:3520 [inline] kvfree_call_rcu +0x3b8/0x510 kernel/rcu/tree.c:3839 trie_update_elem+0x47c/0x620 kernel/bpf/lpm_trie.c:441 bpf_map_update_value+0x324/0x350 kernel/bpf/syscall.c:203 generic_map_update_batch+0x401/0x520 kernel/bpf/syscall.c:1849 bpf_map_do_batch+0x28c/0x3f0 kernel/bpf/syscall.c:5143 __sys_bpf +0x2e5/0x7a0 __do_sys_bpf kernel/bpf/syscall.c:5741 [inline] __se_sys_bpf kernel/bpf/syscall.c:5739 [inline] __x64_sys_bpf+0x43/0x50 kernel/bpf/syscall.c:5739 x64_sys_call+0x2625/0x2d60 arch/x86/include/generated/asm/syscalls_64.h:322 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xc9/0x1c0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe +0x77/0x7f write to 0xffff888237d1cce8 of 8 bytes by task 56 on cpu 0: __mod_timer+0x578/0x7f0 kernel/time/timer.c:1173 add_timer_global+0x51/0x70 kernel/time/timer.c:1330 __queue_delayed_work +0x127/0x1a0 kernel/workqueue.c:2523 queue_delayed_work_on+0xdf/0x190 kernel/workqueue.c:2552 queue_delayed_work include/linux/workqueue.h:677 [inline] schedule_delayed_monitor_work kernel/rcu/tree.c:3525 [inline] kfree_rcu_monitor+0x5e8/0x660 kernel/rcu/tree.c:3643 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0x483/0x9a0 kernel/workqueue.c:3310 worker_thread+0x51d/0x6f0 kernel/workqueue.c:3391 kthread+0x1d1/0x210 kernel/kthread.c:389 ret_from_fork+0x4b/0x60 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 Reported by Kernel Concurrency Sanitizer on: CPU: 0 UID: 0 PID: 56 Comm: kworker/u8:4 Not tainted 6.12.0-rc2-syzkaller-00050-g5b7c893ed5ed #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: events_unbound kfree_rcu_monitor <snip> kfree_rcu_monitor() rearms the work if a "krcp" has to be still offloaded and this is done without holding krcp->lock, whereas the kvfree_call_rcu() holds it. Fix it by acquiring the "krcp->lock" for kfree_rcu_monitor() so both functions do not race anymore.
CVE-2024-53161	In the Linux kernel, the following vulnerability has been resolved: EDAC/bluefield: Fix potential integer overflow The 64-bit argument for the "get DIMM info" SMC call consists of mem_ctrl_idx left-shifted 16 bits and OR-ed with DIMM index. With mem_ctrl_idx defined as 32-bits wide the left-shift operation truncates the upper 16 bits of information during the calculation of the SMC argument. The mem_ctrl_idx stack variable must be defined as 64-bits wide to prevent any potential integer overflow, i.e. loss of data from upper 16 bits.
CVE-2024-53162	In the Linux kernel, the following vulnerability has been resolved: crypto: qat/qat_4xxx - fix off by one in uof_get_name() The fw_objs[] array has "num_objs" elements so the > needs to be >= to prevent an out of bounds read.
CVE-2024-53163	In the Linux kernel, the following vulnerability has been resolved: crypto: qat/qat_420xx - fix off by one in uof_get_name() This is called from uof_get_name_420xx() where "num_objs" is the ARRAY_SIZE() of fw_objs[]. The > needs to be >= to prevent an out of bounds access.
CVE-2024-53164	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix ordering of qlen adjustment Changes to sch->q.qlen around qdisc_tree_reduce_backlog() need to happen _before_ a call to said function because otherwise it may fail to notify parent qdiscs when the child is about to become empty.

CVE ID	Description
CVE-2024-53166	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix bfqq uaf in bfq_limit_depth() Set new allocated bfqq to bic or remove freed bfqq from bic are both protected by bfqd->lock, however bfq_limit_depth() is deferencing bfqq from bic without the lock, this can lead to UAF if the io_context is shared by multiple tasks. For example, test bfq with io_uring can trigger following UAF in v6.6: ===== BUG: KASAN: slab-use-after-free in bfqq_group+0x15/0x50 Call Trace: <TASK> dump_stack_lvl+0x47/0x80 print_address_description.constprop.0+0x66/0x300 print_report+0x3e/0x70 kasan_report+0xb4/0xf0 bfqq_group+0x15/0x50 bfqq_request_over_limit+0x130/0x9a0 bfq_limit_depth+0x1b5/0x480 __blk_mq_alloc_requests+0x2b5/0xa00 blk_mq_get_new_requests+0x11d/0x1d0 blk_mq_submit_bio+0x286/0xb00 submit_bio_noacct_nocheck+0x331/0x400 __block_write_full_folio+0x3d0/0x640 writepage_cb+0x3b/0xc0 write_cache_pages+0x254/0x6c0 write_cache_pages+0x254/0x6c0 do_writepages+0x192/0x310 filemap_fdatawrite_wbc+0x95/0xc0 __filemap_fdatawrite_range+0x99/0xd0 filemap_write_and_wait_range.part.0+0x4d/0xa0 blkdev_read_iter+0xef/0x1e0 io_read+0x1b6/0x8a0 io_issue_sqe+0x87/0x300 io_wq_submit_work+0xeb/0x390 io_worker_handle_work+0x24d/0x550 io_wq_worker+0x27f/0x6c0 ret_from_fork_asm+0x1b/0x30 </TASK> Allocated by task 808602: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 __kasan_slab_alloc+0x83/0x90 kmem_cache_alloc_node+0x1b1/0x6d0 bfq_get_queue+0x138/0xfa0 bfq_get_bfqq_handle_split+0xe3/0x2c0 bfq_init_rq+0x196/0xbb0 bfq_insert_request.isra.0+0xb5/0x480 bfq_insert_requests+0x156/0x180 blk_mq_insert_request+0x15d/0x440 blk_mq_submit_bio+0x8a4/0xb00 submit_bio_noacct_nocheck+0x331/0x400 __blkdev_direct_IO_async+0x2dd/0x330 blkdev_write_iter+0x39a/0x450 io_write+0x22a/0x840 io_issue_sqe+0x87/0x300 io_wq_submit_work+0xeb/0x390 io_worker_handle_work+0x24d/0x550 io_wq_worker+0x27f/0x6c0 ret_from_fork+0x2d/0x50 ret_from_fork_asm+0x1b/0x30 Freed by task 808589: kasan_save_stack+0x1e/0x40 kasan_set_track+0x21/0x30 kasan_save_free_info+0x27/0x40 __kasan_slab_free+0x126/0x1b0 kmem_cache_free+0x10c/0x750 bfq_put_queue+0x2dd/0x770 __bfq_insert_request.isra.0+0x155/0x7a0 bfq_insert_request.isra.0+0x122/0x480 bfq_insert_requests+0x156/0x180 blk_mq_dispatch_plug_list+0x528/0x7e0 blk_mq_flush_plug_list.part.0+0xe5/0x590 __blk_flush_plug+0x3b/0x90 blk_finish_plug+0x40/0x60 do_writepages+0x19d/0x310 filemap_fdatawrite_wbc+0x95/0xc0 __filemap_fdatawrite_range+0x99/0xd0 filemap_write_and_wait_range.part.0+0x4d/0xa0 blkdev_read_iter+0xef/0x1e0 io_read+0x1b6/0x8a0 io_issue_sqe+0x87/0x300 io_wq_submit_work+0xeb/0x390 io_worker_handle_work+0x24d/0x550 io_wq_worker+0x27f/0x6c0 ret_from_fork+0x2d/0x50 ret_from_fork_asm+0x1b/0x30 Fix the problem by protecting bic_to_bfqq() with bfqd->lock.

CVE ID	Description
CVE-2024-53168	In the Linux kernel, the following vulnerability has been resolved: sunrpc: fix one UAF issue caused by sunrpc kernel tcp socket BUG: KASAN: slab-use-after-free in tcp_write_timer_handler +0x156/0x3e0 Read of size 1 at addr ffff888111f322cd by task swapper/0/0 CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.0-rc4-dirty #7 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 Call Trace: <IRQ> dump_stack_lvl+0x68/0xa0 print_address_description.constprop.0+0x2c/0x3d0 print_report+0xb4/0x270 kasan_report +0xbd/0xf0 tcp_write_timer_handler+0x156/0x3e0 tcp_write_timer+0x66/0x170 call_timer_fn +0xfb/0x1d0 __run_timers+0x3f8/0x480 run_timer_softirq+0x9b/0x100 handle_softirqs+0x153/0x390 __irq_exit_rcu+0x103/0x120 irq_exit_rcu+0xe/0x20 sysvec_apic_timer_interrupt+0x76/0x90 </IRQ> <TASK> asm_sysvec_apic_timer_interrupt+0x1a/0x20 RIP: 0010:default_idle+0xf/0x20 Code: 4c 01 c7 4c 29 c2 e9 72 ff ff ff 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa 66 90 0f 00 2d 33 f8 25 00 fb f4 <fa> c3 cc cc cc cc 66 66 2e 0f 1f 84 00 00 00 00 90 90 90 90 RSP: 0018:fffffa2007e28 EFLAGS: 00000242 RAX: 0000000000000000 RBX: 1ffffff4400fc7 RCX: ffffffffa09c3196 RDX: 0000000000000000 RSI: 0000000000000000 RDI: fffffff9f00590f RBP: 0000000000000000 R08: 0000000000000001 R09: ffffd102360835d R10: ffff88811b041aeb R11: 0000000000000001 R12: 0000000000000000 R13: ffffffffa202d7c0 R14: 0000000000000000 R15: 00000000000147d0 default_idle_call+0x6b/0xa0 cpuidle_idle_call +0x1af/0x1f0 do_idle+0xbc/0x130 cpu_startup_entry+0x33/0x40 rest_init+0x11f/0x210 start_kernel+0x39a/0x420 x86_64_start_reservations+0x18/0x30 x86_64_start_kernel+0x97/0xa0 common_startup_64+0x13e/0x141 </TASK> Allocated by task 595: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 __kasan_slab_alloc+0x87/0x90 kmem_cache_alloc_noprof+0x12b/0x3f0 copy_net_ns+0x94/0x380 create_new_namespaces+0x24c/0x500 unshare_nsproxy_namespaces +0x75/0xf0 ksys_unshare+0x24e/0x4f0 __x64_sys_unshare+0x1f/0x30 do_syscall_64+0x70/0x180 entry_SYSCALL_64_after_hwframe+0x76/0xe Freed by task 100: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x54/0x70 kmem_cache_free+0x156/0x5d0 cleanup_net+0x5d3/0x670 process_one_work+0x776/0xa90 worker_thread+0x2e2/0x560 kthread+0x1a8/0x1f0 ret_from_fork+0x34/0x60 ret_from_fork_asm +0x1a/0x30 Reproduction script: mkdir -p /mnt/nfsshare mkdir -p /mnt/nfs/netns_1 mkfs.ext4 /dev/sdb mount /dev/sdb /mnt/nfsshare systemctl restart nfs-server chmod 777 /mnt/nfsshare exportfs -i -o rw,no_root_squash */mnt/nfsshare ip netns add netns_1 ip link add name veth_1_peer type veth peer veth_1 ifconfig veth_1_peer 11.11.0.254 up ip link set veth_1 netns netns_1 ip netns exec netns_1 ifconfig veth_1 11.11.0.1 ip netns exec netns_1 /root/iptables -A OUTPUT -d 11.11.0.254 -p tcp \ --tcp-flags FIN FIN -j DROP (note: In my environment, a DESTROY_CLIENTID operation is always sent immediately, breaking the nfs tcp connection.) ip netns exec netns_1 timeout -s 9 300 mount -t nfs -o proto=tcp,vers=4.1 \ 11.11.0.254:/mnt/nfsshare /mnt/nfs/netns_1 ip netns del netns_1 The reason here is that the tcp socket in netns_1 (nfs side) has been shutdown and closed (done in xs_destroy), but the FIN message (with ack) is discarded, and the nfsd side keeps sending retransmission messages. As a result, when the tcp sock in netns_1 processes the received message, it sends the message (FIN message) in the sending queue, and the tcp timer is re-established. When the network namespace is deleted, the net structure accessed by tcp's timer handler function causes problems. To fix this problem, let's hold netns refcnt for the tcp kernel socket as done in other modules. This is an ugly hack which can easily be backported to earlier kernels. A proper fix which cleans up the interfaces will follow, but may not be so easy to backport.

CVE ID	Description
CVE-2024-53169	In the Linux kernel, the following vulnerability has been resolved: nvme-fabrics: fix kernel crash while shutting down controller The nvme keep-alive operation, which executes at a periodic interval, could potentially sneak in while shutting down a fabric controller. This may lead to a race between the fabric controller admin queue destroy code path (invoked while shutting down controller) and hw/hctx queue dispatcher called from the nvme keep-alive async request queuing operation. This race could lead to the kernel crash shown below: Call Trace: autoremove_wake_function+0x0/0x0 (unreliable) __blk_mq_sched_dispatch_requests+0x114/0x24c blk_mq_sched_dispatch_requests+0x44/0x84 blk_mq_run_hw_queue+0x140/0x220 nvme_keep_alive_work+0xc8/0x19c [nvme_core] process_one_work+0x200/0x4e0 worker_thread+0x340/0x504 kthread+0x138/0x140 start_kernel_thread+0x14/0x18 While shutting down fabric controller, if nvme keep-alive request sneaks in then it would be flushed off. The nvme_keep_alive_end_io function is then invoked to handle the end of the keep-alive operation which decrements the admin->q_usage_counter and assuming this is the last/only request in the admin queue then the admin->q_usage_counter becomes zero. If that happens then blk-mq destroy queue operation (blk_mq_destroy_queue()) which could be potentially running simultaneously on another cpu (as this is the controller shutdown code path) would forward progress and deletes the admin queue. So, now from this point onward we are not supposed to access the admin queue resources. However the issue here's that the nvme keep-alive thread running hw/hctx queue dispatch operation hasn't yet finished its work and so it could still potentially access the admin queue resource while the admin queue had been already deleted and that causes the above crash. The above kernel crash is regression caused due to changes implemented in commit a54a93d0e359 ("nvme: move stopping keep-alive into nvme_uninit_ctrl()"). Ideally we should stop keep-alive before destroying the admin queue and freeing the admin tagset so that it wouldn't sneak in during the shutdown operation. However we removed the keep alive stop operation from the beginning of the controller shutdown code path in commit a54a93d0e359 ("nvme: move stopping keep-alive into nvme_uninit_ctrl()") and added it under nvme_uninit_ctrl() which executes very late in the shutdown code path after the admin queue is destroyed and its tagset is removed. So this change created the possibility of keep-alive sneaking in and interfering with the shutdown operation and causing observed kernel crash. To fix the observed crash, we decided to move nvme_stop_keep_alive() from nvme_uninit_ctrl() to nvme_remove_admin_tag_set(). This change would ensure that we don't forward progress and delete the admin queue until the keep-alive operation is finished (if it's in-flight) or cancelled and that would help contain the race condition explained above and hence avoid the crash. Moving nvme_stop_keep_alive() to nvme_remove_admin_tag_set() instead of adding nvme_stop_keep_alive() to the beginning of the controller shutdown code path in nvme_stop_ctrl(), as was the case earlier before commit a54a93d0e359 ("nvme: move stopping keep-alive into nvme_uninit_ctrl()"), would help save one callsite of nvme_stop_keep_alive().

CVE ID	Description
CVE-2024-53171	In the Linux kernel, the following vulnerability has been resolved: ubifs: authentication: Fix use-after-free in ubifs_tnc_end_commit After an insertion in TNC, the tree might split and cause a node to change its `zn->parent`. A further deletion of other nodes in the tree (which also could free the nodes), the aforementioned node's `zn->parent` could still point to a freed node. This `zn->parent` may not be updated when getting nodes to commit in `ubifs_tnc_start_commit()`. This could then trigger a use-after-free when accessing the `zn->parent` in `write_index()` in `ubifs_tnc_end_commit()`. This can be triggered by running <code>rm -f /etc/test-file.bin dd if=/dev/urandom of=/etc/test-file.bin bs=1M count=60 conv=fsync</code> in a loop, and with `CONFIG_UBIFS_FS_AUTHENTICATION`. KASAN then reports: BUG: KASAN: use-after-free in ubifs_tnc_end_commit+0xa5c/0x1950 Write of size 32 at addr ffffff800a3af86c by task ubifs_bgt0_20/153 Call trace: dump_backtrace+0x0/0x340 show_stack+0x18/0x24 dump_stack_lvl+0x9c/0xabc print_address_description.constprop.0+0x74/0x2b0 kasan_report+0x1d8/0x1f0 kasan_check_range+0xf8/0x1a0 memcpy+0x84/0xf4 ubifs_tnc_end_commit+0xa5c/0x1950 do_commit+0x4e0/0x1340 ubifs_bg_thread+0x234/0x2e0 kthread+0x36c/0x410 ret_from_fork+0x10/0x20 Allocated by task 401: kasan_save_stack+0x38/0x70 __kasan_kmalloc+0x8c/0xd0 __kmalloc+0x34c/0x5bc tnc_insert+0x140/0x16a4 ubifs_tnc_add+0x370/0x52c ubifs_jnl_write_data+0x5d8/0x870 do_writepage+0x36c/0x510 ubifs_writepage+0x190/0x4dc __writepage+0x58/0x154 write_cache_pages+0x394/0x830 do_writepages+0x1f0/0x5b0 filemap_fdatawrite_wbc+0x170/0x25c file_write_and_wait_range+0x140/0x190 ubifs_fsync+0xe8/0x290 vfs_fsync_range+0xc0/0x1e4 do_fsync+0x40/0x90 __arm64_sys_fsync+0x34/0x50 invoke_syscall.constprop.0+0xa8/0x260 do_el0_svc+0xc8/0x1f0 el0_svc+0x34/0x70 el0t_64_sync_handler+0x108/0x114 el0t_64_sync+0x1a4/0x1a8 Freed by task 403: kasan_save_stack+0x38/0x70 kasan_set_track+0x28/0x40 kasan_set_free_info+0x28/0x4c __kasan_slab_free+0xd4/0x13c kfree+0xc4/0x3a0 tnc_delete+0x3f4/0xe40 ubifs_tnc_remove_range+0x368/0x73c ubifs_tnc_remove_ino+0x29c/0x2e0 ubifs_jnl_delete_inode+0x150/0x260 ubifs_evict_inode+0x1d4/0x2e4 evict+0x1c8/0x450 iput+0x2a0/0x3c4 do_unlinkat+0x2cc/0x490 __arm64_sys_unlinkat+0x90/0x100 invoke_syscall.constprop.0+0xa8/0x260 do_el0_svc+0xc8/0x1f0 el0_svc+0x34/0x70 el0t_64_sync_handler+0x108/0x114 el0t_64_sync+0x1a4/0x1a8 The offending `memcpy()` in `ubifs_copy_hash()` has a use-after-free when a node becomes root in TNC but still has a `parent` to an already freed node. More specifically, consider the following TNC: zroot // zp1 // zn Inserting a new node `zn_new` with a key smaller than `zn` will trigger a split in `tnc_insert()` if `zp1` is full: zroot /\ \ zp1 zp2 /\ \ zn_new zn `zn->parent` has now been moved to `zp2`, *but* `zn->parent` still points to `zp1`. Now, consider a removal of all the nodes `except_` `zn`. Just when `tnc_delete()` is about to delete `zroot` and `zp2`: zroot /\ zp2 /\ zn `zroot` and `zp2` get freed and the tree collapses: zn `zn` now becomes the new `zroot`. `get_znodes_to_commit()` will now only find `zn`, the new `zroot`, and `write_index()` will check its `zn->parent` that wrongly points to the already freed `zp1`. `ubifs_copy_hash()` thus gets wrongly called with `zn->parent->zbranch[znode->iip].hash` that triggers the use-after-free! Fix this by explicitly setting `zn->parent` to `NULL` in `get_znodes_to_commit()` for the root node. The search for the dirty nodes ---truncated---
CVE-2024-53172	In the Linux kernel, the following vulnerability has been resolved: ubi: fastmap: Fix duplicate slab cache names while attaching Since commit 4c39529663b9 ("slab: Warn on duplicate cache names when DEBUG_VM=y"), the duplicate slab cache names can be detected and a kernel WARNING is thrown out. In UBI fast attaching process, <code>alloc_ai()</code> could be invoked twice with the same slab cache name 'ubi_aeb_slab_cache', which will trigger following warning messages: <code>kmem_cache of name 'ubi_aeb_slab_cache' already exists WARNING: CPU: 0 PID: 7519 at mm/slab_common.c:107 __kmem_cache_create_args+0x100/0x5f0 Modules linked in: ubi(+) nandsim [last unloaded: nandsim] CPU: 0 UID: 0 PID: 7519 Comm: modprobe Tainted: G 6.12.0-rc2 RIP: 0010: __kmem_cache_create_args+0x100/0x5f0 Call Trace: __kmem_cache_create_args+0x100/0x5f0 alloc_ai+0x295/0x3f0 [ubi] ubi_attach+0x3c3/0xccc [ubi] ubi_attach_mtd_dev+0x17cf/0x3fa0 [ubi] ubi_init+0x3fb/0x800 [ubi] do_init_module+0x265/0x7d0 __x64_sys_finit_module+0x7a/0xc0 The problem could be easily reproduced by loading UBI device by fastmap with <code>CONFIG_DEBUG_VM=y</code>. Fix it by using different slab names for <code>alloc_ai()</code> callers.</code>
CVE-2024-53173	In the Linux kernel, the following vulnerability has been resolved: NFSv4.0: Fix a use-after-free problem in the asynchronous open() Yang Erkun reports that when two threads are opening files at the same time, and are forced to abort before a reply is seen, then the call to <code>nfs_release_seqid()</code> in <code>nfs4_opendata_free()</code> can result in a use-after-free of the pointer to the defunct rpc task of the other thread. The fix is to ensure that if the RPC call is aborted before the call to <code>nfs_wait_on_sequence()</code> is complete, then we must call <code>nfs_release_seqid()</code> in <code>nfs4_open_release()</code> before the <code>rpc_task</code> is freed.

CVE ID	Description
CVE-2024-53174	In the Linux kernel, the following vulnerability has been resolved: SUNRPC: make sure cache entry active before cache_show The function `c_show` was called with protection from RCU. This only ensures that `cp` will not be freed. Therefore, the reference count for `cp` can drop to zero, which will trigger a refcount use-after-free warning when `cache_get` is called. To resolve this issue, use `cache_get_rcu` to ensure that `cp` remains active. -----[cut here]----- refcount_t: addition on 0; use-after-free. WARNING: CPU: 7 PID: 822 at lib/refcount.c:25 refcount_warn_saturate+0xb1/0x120 CPU: 7 UID: 0 PID: 822 Comm: cat Not tainted 6.12.0-rc3+ #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 RIP: 0010:refcount_warn_saturate+0xb1/0x120 Call Trace: <TASK> c_show+0x2fc/0x380 [sunrpc] seq_read_iter+0x589/0x770 seq_read+0x1e5/0x270 proc_reg_read+0xe1/0x140 vfs_read+0x125/0x530 ksys_read+0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2024-53175	In the Linux kernel, the following vulnerability has been resolved: ipc: fix memleak if msg_init_ns failed in create_ipc_ns Percpu memory allocation may failed during create_ipc_ns however this fail is not handled properly since ipc sysctls and mq sysctls is not released properly. Fix this by release these two resource when failure. Here is the kmemleak stack when percpu failed: unreferenced object 0xffff88819de2a600 (size 512): comm "shmem_2nctest", pid 120711, jiffies 4300542254 hex dump (first 32 bytes): 60 aa 9d 84 ff ff ff fc 18 48 b2 84 88 ff ff H..... 04 00 00 00 a4 01 00 00 20 e4 56 81 ff ff ff V..... backtrace (crc be7cba35): [<ffffffff81b43f83>] __kmalloct_node_track_caller_noprof+0x333/0x420 [<ffffffff81a52e56>] kmempdup_noprof+0x26/0x50 [<ffffffff821b2f37>] setup_mq_sysctls+0x57/0x1d0 [<ffffffff821b29cc>] copy_ipcs+0x29c/0x3b0 [<ffffffff815d6a10>] create_new_namespaces+0x1d0/0x920 [<ffffffff815d7449>] copy_namespaces+0x2e9/0x3e0 [<ffffffff815458f3>] copy_process+0x29f3/0x7ff0 [<ffffffff8154b080>] kernel_clone+0xc0/0x650 [<ffffffff8154b6b1>] __do_sys_clone+0xa1/0xe0 [<ffffffff843df8ff>] do_syscall_64+0xbf/0x1c0 [<ffffffff846000b0>] entry_SYSCALL_64_after_hwframe+0x4b/0x53
CVE-2024-53176	In the Linux kernel, the following vulnerability has been resolved: smb: During unmount, ensure all cached dir instances drop their dentry The unmount process (cifs_kill_sb) calling close_all_cached_dirs() can race with various cached directory operations, which ultimately results in dentries not being dropped and these kernel BUGs: BUG: Dentry ffff88814f37e358{i=1000000000080,n=/} still in use (2) [unmount of cifs cifs] VFS: Busy inodes after unmount of cifs (cifs) -----[cut here]----- kernel BUG at fs/super.c:661! This happens when a cfid is in the process of being cleaned up when, and has been removed from the cfdids->entries list, including: - Receiving a lease break from the server - Server reconnection triggers invalidate_all_cached_dirs(), which removes all the cfdids from the list - The laundromat thread decides to expire an old cfid. To solve these problems, dropping the dentry is done in queued work done in a newly-added cfid_put_wq workqueue, and close_all_cached_dirs() flushes that workqueue after it drops all the dentries of which it's aware. This is a global workqueue (rather than scoped to a mount), but the queued work is minimal. The final cleanup work for cleaning up a cfid is performed via work queued in the serverclose_wq workqueue; this is done separate from dropping the dentries so that close_all_cached_dirs() doesn't block on any server operations. Both of these queued works expect to invoked with a cfid reference and a tcon reference to avoid those objects from being freed while the work is ongoing. While we're here, add proper locking to close_all_cached_dirs(), and locking around the freeing of cfid->dentry.

CVE ID	Description
CVE-2024-53177	In the Linux kernel, the following vulnerability has been resolved: smb: prevent use-after-free due to open_cached_dir error paths If open_cached_dir() encounters an error parsing the lease from the server, the error handling may race with receiving a lease break, resulting in open_cached_dir() freeing the cfid while the queued work is pending. Update open_cached_dir() to drop refs rather than directly freeing the cfid. Have cached_dir_lease_break(), cfids_laundromat_worker(), and invalidate_all_cached_dirs() clear has_lease immediately while still holding cfids->cfid_list_lock, and then use this to also simplify the reference counting in cfids_laundromat_worker() and invalidate_all_cached_dirs(). Fixes this KASAN splat (which manually injects an error and lease break in open_cached_dir()): <pre>===== BUG: KASAN: slab-use-after-free in smb2_cached_lease_break+0x27/0xb0 Read of size 8 at addr ffff88811cc24c10 by task kworker/3:1/65 CPU: 3 UID: 0 PID: 65 Comm: kworker/3:1 Not tainted 6.12.0-rc6-g255cf264e6e5-dirty #87 Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 Workqueue: cifsiod smb2_cached_lease_break Call Trace: <TASK> dump_stack_lvl+0x77/0xb0 print_report+0xce/0x660 kasan_report+0xd3/0x110 smb2_cached_lease_break+0x27/0xb0 process_one_work+0x50a/0xc50 worker_thread+0x2ba/0x530 kthread+0x17c/0x1c0 ret_from_fork+0x34/0x60 ret_from_fork_asm+0x1a/0x30 </TASK> Allocated by task 2464: kasan_save_stack+0x33/0x60 kasan_save_track+0x14/0x30 __kasan_kmalloc +0xaa/0xb0 open_cached_dir+0xa7d/0x1fb0 smb2_query_path_info+0x43c/0x6e0 cifs_get_fattr +0x346/0xf10 cifs_get_inode_info+0x157/0x210 cifs_revalidate_dentry_attr+0x2d1/0x460 cifs_getattr+0x173/0x470 vfs_statx_path+0x10f/0x160 vfs_statx+0xe9/0x150 vfs_fstatat+0x5e/0xc0 __do_sys_newfstatat+0x91/0xf0 do_syscall_64+0x95/0x1a0 entry_SYSCALL_64_after_hwframe +0x76/0x7e Freed by task 2464: kasan_save_stack+0x33/0x60 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x51/0x70 kfree+0x174/0x520 open_cached_dir +0x97f/0x1fb0 smb2_query_path_info+0x43c/0x6e0 cifs_get_fattr+0x346/0xf10 cifs_get_inode_info +0x157/0x210 cifs_revalidate_dentry_attr+0x2d1/0x460 cifs_getattr+0x173/0x470 vfs_statx_path +0x10f/0x160 vfs_statx+0xe9/0x150 vfs_fstatat+0x5e/0xc0 __do_sys_newfstatat+0x91/0xf0 do_syscall_64+0x95/0x1a0 entry_SYSCALL_64_after_hwframe+0x76/0x7e Last potentially related work creation: kasan_save_stack+0x33/0x60 __kasan_record_aux_stack+0xad/0xc0 insert_work+0x32/0x100 __queue_work+0x5c9/0x870 queue_work_on+0x82/0x90 open_cached_dir +0x1369/0x1fb0 smb2_query_path_info+0x43c/0x6e0 cifs_get_fattr+0x346/0xf10 cifs_get_inode_info +0x157/0x210 cifs_revalidate_dentry_attr+0x2d1/0x460 cifs_getattr+0x173/0x470 vfs_statx_path +0x10f/0x160 vfs_statx+0xe9/0x150 vfs_fstatat+0x5e/0xc0 __do_sys_newfstatat+0x91/0xf0 do_syscall_64+0x95/0x1a0 entry_SYSCALL_64_after_hwframe+0x76/0x7e The buggy address belongs to the object at ffff88811cc24c00 which belongs to the cache kmalloc-1k of size 1024 The buggy address is located 16 bytes inside of freed 1024-byte region [ffff88811cc24c00, ffff88811cc25000)</pre>

CVE ID	Description
CVE-2024-53178	In the Linux kernel, the following vulnerability has been resolved: smb: Don't leak cfid when reconnect races with open_cached_dir open_cached_dir() may either race with the tcon reconnection even before compound_send_recv() or directly trigger a reconnection via SMB2_open_init() or SMB_query_info_init(). The reconnection process invokes invalidate_all_cached_dirs() via cifs_mark_open_files_invalid(), which removes all cfids from the cfids->entries list but doesn't drop a ref if has_lease isn't true. This results in the currently-being-constructed cfid not being on the list, but still having a refcount of 2. It leaks if returned from open_cached_dir(). Fix this by setting cfid->has_lease when the ref is actually taken; the cfid will not be used by other threads until it has a valid time. Addresses these kmemleaks: unreferenced object 0xffff8881090c4000 (size 1024): comm "bash", pid 1860, jiffies 4295126592 hex dump (first 32 bytes): 00 01 00 00 00 00 ad de 22 01 00 00 00 00 ad de "..... 00 ca 45 22 81 88 ff ff f8 dc 4f 04 81 88 ff ff ..E".....O..... backtrace (crc 6f58c20f): [<ffffffffff8b895a1e>] __kmalloccache_noprof+0x2be/0x350 [<ffffffffff8bda06e3>] open_cached_dir+0x993/0x1fb0 [<ffffffffff8bdaa750>] cifs_readdir+0x15a0/0x1d50 [<ffffffffff8b9a853f>] iterate_dir+0x28f/0x4b0 [<ffffffffff8b9a9aed>] __x64_sys_getdents64+0xfd/0x200 [<ffffffffff8cf6da05>] do_syscall_64+0x95/0x1a0 [<ffffffffff8d00012f>] entry_SYSCALL_64_after_hwframe+0x76/0x7e unreferenced object 0xffff8881044fdcf8 (size 8): comm "bash", pid 1860, jiffies 4295126592 hex dump (first 8 bytes): 00 cc cc cc cc cc cc backtrace (crc 10c106a9): [<ffffffffff8b89a3d3>] __kmalloccache_node_track_caller_noprof+0x363/0x480 [<ffffffffff8b7d7256>] kstrdup+0x36/0x60 [<ffffffffff8bda0700>] open_cached_dir+0x9b0/0x1fb0 [<ffffffffff8bdaa750>] cifs_readdir+0x15a0/0x1d50 [<ffffffffff8b9a853f>] iterate_dir+0x28f/0x4b0 [<ffffffffff8b9a9aed>] __x64_sys_getdents64+0xfd/0x200 [<ffffffffff8cf6da05>] do_syscall_64+0x95/0x1a0 [<ffffffffff8d00012f>] entry_SYSCALL_64_after_hwframe+0x76/0x7e And addresses these BUG splats when unmounting the SMB filesystem: BUG: Dentry ffff888140590ba0 {i=100000000000080,n=/} still in use (2) [unmount of cifs cifs] WARNING: CPU: 3 PID: 3433 at fs/dcache.c:1536 umount_check+0xd0/0x100 Modules linked in: CPU: 3 UID: 0 PID: 3433 Comm: bash Not tainted 6.12.0-rc4-g850925a8133c-dirty #49 Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 RIP: 0010:umount_check+0xd0/0x100 Code: 8d 7c 24 40 e8 31 5a f4 ff 49 8b 54 24 40 41 56 49 89 e9 45 89 e8 48 89 d9 41 57 48 89 de 48 c7 c7 80 e7 db ac e8 f0 72 9a ff <0f> 0b 58 31 c0 5a 5b 5d 41 5c 41 5d 41 5e 41 5f e9 2b e5 5d 01 41 RSP: 0018:ffff88811cc27978 EFLAGS: 00010286 RAX: 0000000000000000 RBX: ffff888140590ba0 RCX: ffffffff8faaf20bae RDX: dffffc0000000000 RSI: 0000000000000008 RDI: ffff8881f6fb6f40 RBP: ffff8881462ec000 R08: 0000000000000001 R09: ffffd1023984ee3 R10: ffff88811cc2771f R11: 00000000016cfcc0 R12: ffff888134383e08 R13: 0000000000000002 R14: ffff8881462ec668 R15: ffffffff8faceab4c0 FS: 00007f23bfa98740(0000) GS: ffff8881f6f80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000556de4a6f808 CR3: 0000000123c80000 CR4: 0000000000350ef0 Call Trace: <TASK> d_walk+0x6a/0x530 shrink_dcache_for_umount+0x6a/0x200 generic_shutdown_super+0x52/0x2a0 kill_anon_super+0x22/0x40 cifs_kill_sb+0x159/0x1e0 deactivate_locked_super+0x66/0xe0 cleanup_mnt+0x140/0x210 task_work_run+0xfb/0x170 syscall_exit_to_user_mode+0x29f/0x2b0 do_syscall_64+0xa1/0x1a0 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f23bfb93ae7 Code: ff ff ff ff c3 66 0f 1f 44 00 00 48 8b 0d 11 93 0d 00 f7 d8 64 89 01 b8 ff ff ff ff eb bf 0f 1f 44 00 00 b8 50 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d e9 92 0d 00 f7 d8 64 89 ---truncated---
CVE-2024-53179	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix use-after-free of signing key Customers have reported use-after-free in @ses->auth_key.response with SMB2.1 + sign mounts which occurs due to following race: task A task B cifs_mount() dfs_mount_share() get_session() cifs_mount_get_session() cifs_send_recv() cifs_get_smb_ses() compound_send_recv() cifs_setup_session() smb2_setup_request() kfree_sensitive() smb2_calc_signature() crypto_shash_setkey() *UAF* Fix this by ensuring that we have a valid @ses->auth_key.response by checking whether @ses->ses_status is SES_GOOD or SES_EXITING with @ses->ses_lock held. After commit 24a9799aa8ef ("smb: client: fix UAF in smb2_reconnect_server()"), we made sure to call ->logoff() only when @ses was known to be good (e.g. valid ->auth_key.response), so it's safe to access signing key when @ses->ses_status == SES_EXITING.
CVE-2024-53180	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Add sanity NULL check for the default mmap fault handler A driver might allow the mmap access before initializing its runtime->dma_area properly. Add a proper NULL check before passing to virt_to_page() for avoiding a panic.

CVE ID	Description
CVE-2024-53181	In the Linux kernel, the following vulnerability has been resolved: um: vector: Do not use drvdata in release The drvdata is not available in release. Let's just use container_of() to get the vector_device instance. Otherwise, removing a vector device will result in a crash: RIP: 0033:vector_device_release+0xf/0x50 RSP: 00000000e187bc40 EFLAGS: 00010202 RAX: 0000000060028f61 RBX: 00000000600f1baf RCX: 00000000620074e0 RDX: 000000006220b9c0 RSI: 0000000060551c80 RDI: 0000000000000000 RBP: 00000000e187bc50 R08: 00000000603ad594 R09: 00000000e187bb70 R10: 00000000000135a R11: 00000000603ad422 R12: 00000000623ae028 R13: 000000006287a200 R14: 0000000062006d30 R15: 00000000623700b6 Kernel panic - not syncing: Segfault with no mm CPU: 0 UID: 0 PID: 16 Comm: kworker/0:1 Not tainted 6.12.0-rc6-g59b723cd2adb #1 Workqueue: events mc_work_proc Stack: 60028f61 623ae028 e187bc80 60276fcd 6220b9c0 603f5820 623ae028 00000000 e187bcb0 603a2bcd 623ae000 62370010 Call Trace: [<60028f61>] ? vector_device_release+0x0/0x50 [<60276fcd>] device_release+0x70/0xba [<603a2bcd>] kobject_put+0xba/0xe7 [<60277265>] put_device+0x19/0x1c [<60281266>] platform_device_put+0x26/0x29 [<60281e5f>] platform_device_unregister+0x2c/0x2e [<60029422>] vector_remove+0x52/0x58 [<60031316>] ? mconsole_reply+0x0/0x50 [<600310c8>] mconsole_remove+0x160/0x1cc [<603b19f4>] ? strlen+0x0/0x15 [<60066611>] ? __dequeue_entity+0x1a9/0x206 [<600666a7>] ? set_next_entity+0x39/0x63 [<6006666e>] ? set_next_entity+0x0/0x63 [<60038fa6>] ? um_set_signals+0x0/0x43 [<6003070c>] mc_work_proc+0x77/0x91 [<60057664>] process_scheduled_works+0x1b3/0x2dd [<60055f32>] ? assign_work+0x0/0x58 [<60057f0a>] worker_thread+0x1e9/0x293 [<6005406f>] ? set_pf_worker+0x0/0x64 [<6005d65d>] ? arch_local_irq_save+0x0/0x2d [<6005d748>] ? kthread_exit+0x0/0x3a [<60057d21>] ? worker_thread+0x0/0x293 [<6005dbf1>] kthread+0x126/0x12b [<600219c5>] new_thread_handler+0x85/0xb6
CVE-2024-53183	In the Linux kernel, the following vulnerability has been resolved: um: net: Do not use drvdata in release The drvdata is not available in release. Let's just use container_of() to get the uml_net instance. Otherwise, removing a network device will result in a crash: RIP: 0033:net_device_release+0x10/0x6f RSP: 00000000e20c7c40 EFLAGS: 00010206 RAX: 000000006002e4e7 RBX: 00000000600f1baf RCX: 00000000624074e0 RDX: 0000000062778000 RSI: 0000000060551c80 RDI: 00000000627af028 RBP: 00000000e20c7c50 R08: 00000000603ad594 R09: 00000000e20c7b70 R10: 0000000000000135a R11: 00000000603ad422 R12: 0000000000000000 R13: 0000000062c7af00 R14: 0000000062406d60 R15: 00000000627700b6 Kernel panic - not syncing: Segfault with no mm CPU: 0 UID: 0 PID: 29 Comm: kworker/0:2 Not tainted 6.12.0-rc6-g59b723cd2adb #1 Workqueue: events mc_work_proc Stack: 627af028 62c7af00 e20c7c80 60276fcd 62778000 603f5820 627af028 00000000 e20c7cb0 603a2bcd 627af000 62770010 Call Trace: [<60276fcd>] device_release+0x70/0xba [<603a2bcd>] kobject_put+0xba/0xe7 [<60277265>] put_device+0x19/0x1c [<60281266>] platform_device_put+0x26/0x29 [<60281e5f>] platform_device_unregister+0x2c/0x2e [<6002ec9c>] net_remove+0x63/0x69 [<60031316>] ? mconsole_reply+0x0/0x50 [<600310c8>] mconsole_remove+0x160/0x1cc [<60087d40>] ? __remove_hrtimer+0x38/0x74 [<60087ff8>] ? hrtimer_try_to_cancel+0x8c/0x98 [<6006b3cf>] ? dl_server_stop+0x3f/0x48 [<6006b390>] ? dl_server_stop+0x0/0x48 [<600672e8>] ? dequeue_entities+0x327/0x390 [<60038fa6>] ? um_set_signals+0x0/0x43 [<6003070c>] mc_work_proc+0x77/0x91 [<60057664>] process_scheduled_works+0x1b3/0x2dd [<60055f32>] ? assign_work+0x0/0x58 [<60057f0a>] worker_thread+0x1e9/0x293 [<6005406f>] ? set_pf_worker+0x0/0x64 [<6005d65d>] ? arch_local_irq_save+0x0/0x2d [<6005d748>] ? kthread_exit+0x0/0x3a [<60057d21>] ? worker_thread+0x0/0x293 [<6005dbf1>] kthread+0x126/0x12b [<600219c5>] new_thread_handler+0x85/0xb6

CVE ID	Description
CVE-2024-53184	In the Linux kernel, the following vulnerability has been resolved: um: ubd: Do not use drvdata in release The drvdata is not available in release. Let's just use container_of() to get the ubd instance. Otherwise, removing a ubd device will result in a crash: RIP: 0033:blk_mq_free_tag_set +0x1f/0xba RSP: 00000000e2083bf0 EFLAGS: 00010246 RAX: 000000006021463a RBX: 0000000000000348 RCX: 0000000062604d00 RDX: 0000000004208060 RSI: 00000000605241a0 RDI: 0000000000000348 RBP: 00000000e2083c10 R08: 0000000062414010 R09: 00000000601603f7 R10: 000000000000133a R11: 000000006038c4bd R12: 0000000000000000 R13: 0000000060213a5c R14: 0000000062405d20 R15: 00000000604f7aa0 Kernel panic - not syncing: Segfault with no mm CPU: 0 PID: 17 Comm: kworker/0:1 Not tainted 6.8.0-rc3-00107-gba3f67c11638 #1 Workqueue: events mc_work_proc Stack: 00000000 604f7ef0 62c5d000 62405d20 e2083c30 6002c776 6002c755 600e47ff e2083c60 6025ffe3 04208060 603d36e0 Call Trace: [<6002c776>] ubd_device_release+0x21/0x55 [<6002c755>] ? ubd_device_release+0x0/0x55 [<600e47ff>] ? kfree+0x0/0x100 [<6025ffe3>] device_release+0x70/0xba [<60381d6a>] kobject_put+0xb5/0xe2 [<6026027b>] put_device+0x19/0x1c [<6026a036>] platform_device_put+0x26/0x29 [<6026ac5a>] platform_device_unregister+0x2c/0x2e [<6002c52e>] ubd_remove+0xb8/0xd6 [<6002bb74>] ? mconsole_reply+0x0/0x50 [<6002b926>] mconsole_remove+0x160/0x1cc [<6002bbbc>] ? mconsole_reply+0x48/0x50 [<6003379c>] ? um_set_signals+0x3b/0x43 [<60061c55>] ? update_min_vruntime+0x14/0x70 [<6006251f>] ? dequeue_task_fair+0x164/0x235 [<600620aa>] ? update_cfs_group+0x0/0x40 [<603a0e77>] ? __schedule+0x0/0x3ed [<60033761>] ? um_set_signals+0x0/0x43 [<6002af6a>] mc_work_proc+0x77/0x91 [<600520b4>] process_scheduled_works+0x1af/0x2c3 [<6004ede3>] ? assign_work+0x0/0x58 [<600527a1>] worker_thread+0x2f7/0x37a [<6004ee3b>] ? set_pf_worker+0x0/0x64 [<6005765d>] ? arch_local_irq_save+0x0/0x2d [<60058e07>] ? kthread_exit+0x0/0x3a [<600524aa>] ? worker_thread+0x0/0x37a [<60058f9f>] kthread+0x130/0x135 [<6002068e>] new_thread_handler+0x85/0xb6
CVE-2024-53185	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix NULL ptr deref in crypto_aead_setkey() Neither SMB3.0 or SMB3.02 supports encryption negotiate context, so when SMB2_GLOBAL_CAP_ENCRYPTION flag is set in the negotiate response, the client uses AES-128-CCM as the default cipher. See MS-SMB2 3.3.5.4. Commit b0abcd65ec54 ("smb: client: fix UAF in async decryption") added a @server->cipher_type check to conditionally call smb3_crypto_aead_allocate(), but that check would always be false as @server->cipher_type is unset for SMB3.02. Fix the following KASAN splat by setting @server->cipher_type for SMB3.02 as well. mount.cifs //srv/share /mnt -o vers=3.02,seal,... BUG: KASAN: null-ptr-deref in crypto_aead_setkey+0x2c/0x130 Read of size 8 at addr 0000000000000020 by task mount.cifs/1095 CPU: 1 UID: 0 PID: 1095 Comm: mount.cifs Not tainted 6.12.0 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-3.fc41 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x5d/0x80 ? crypto_aead_setkey+0x2c/0x130 kasan_report+0xda/0x110 ? crypto_aead_setkey+0x2c/0x130 crypto_aead_setkey+0x2c/0x130 crypt_message+0x258/0xec0 [cifs] ? __asan_memset+0x23/0x50 ? __pfx_crypt_message+0x10/0x10 [cifs] ? mark_lock+0xb0/0x6a0 ? hlock_class+0x32/0xb0 ? mark_lock+0xb0/0x6a0 smb3_init_transform_rq+0x352/0x3f0 [cifs] ? lock_acquire.part.0+0xf4/0x2a0 smb_send_rqst+0x144/0x230 [cifs] ? __pfx_smb_send_rqst+0x10/0x10 [cifs] ? hlock_class+0x32/0xb0 ? smb2_setup_request+0x225/0x3a0 [cifs] ? __pfx_cifs_compound_last_callback+0x10/0x10 [cifs] compound_send_recv+0x59b/0x1140 [cifs] ? __pfx_compound_send_recv+0x10/0x10 [cifs] ? __create_object+0x5e/0x90 ? hlock_class+0x32/0xb0 ? do_raw_spin_unlock+0x9a/0xf0 cifs_send_recv+0x23/0x30 [cifs] SMB2_tcon+0x3ec/0xb30 [cifs] ? __pfx_SMB2_tcon+0x10/0x10 [cifs] ? lock_acquire.part.0+0xf4/0x2a0 ? __pfx_lock_release+0x10/0x10 ? do_raw_spin_trylock+0xc6/0x120 ? lock_acquire+0x3f/0x90 ? __get_xid+0x16/0xd0 [cifs] ? __pfx_SMB2_tcon+0x10/0x10 [cifs] ? cifs_get_smb_ses+0xcdd/0x10a0 [cifs] cifs_get_smb_ses+0xcdd/0x10a0 [cifs] ? __pfx_cifs_get_smb_ses+0x10/0x10 [cifs] ? cifs_get_tcp_session+0xaa0/0xca0 [cifs] cifs_mount_get_session+0x8a/0x210 [cifs] dfs_mount_share+0x1b0/0x11d0 [cifs] ? __pfx__lock_acquire+0x10/0x10 ? __pfx_dfs_mount_share+0x10/0x10 [cifs] ? lock_acquire.part.0+0xf4/0x2a0 ? find_held_lock+0x8a/0xa0 ? hlock_class+0x32/0xb0 ? lock_release+0x203/0x5d0 cifs_mount+0xb3/0x3d0 [cifs] ? do_raw_spin_trylock+0xc6/0x120 ? __pfx_cifs_mount+0x10/0x10 [cifs] ? lock_acquire+0x3f/0x90 ? find_nls+0x16/0xa0 ? smb3_update_mnt_flags+0x372/0x3b0 [cifs] cifs_smb3_do_mount+0x1e2/0xc80 [cifs] ? __pfx_vfs_parse_fs_string+0x10/0x10 ? __pfx_cifs_smb3_do_mount+0x10/0x10 [cifs] smb3_get_tree+0x1bf/0x330 [cifs] vfs_get_tree+0x4a/0x160 path_mount+0x3c1/0xf0 ? kasan_quarantine_put+0xc7/0x1d0 ? __pfx_path_mount+0x10/0x10 ? kmem_cache_free+0x118/0x3e0 ? user_path_at+0x74/0xa0 __x64_sys_mount+0x1a6/0x1e0 ? __pfx__x64_sys_mount+0x10/0x10 ? mark_held_locks+0x1a/0x90 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f

CVE ID	Description
CVE-2024-53187	In the Linux kernel, the following vulnerability has been resolved: io_uring: check for overflows in io_pin_pages WARNING: CPU: 0 PID: 5834 at io_uring/memmap.c:144 io_pin_pages+0x149/0x180 io_uring/memmap.c:144 CPU: 0 UID: 0 PID: 5834 Comm: syz-executor825 Not tainted 6.12.0-next-20241118-syzkaller #0 Call Trace: <TASK> __io_uaddr_map+0xfb/0x2d0 io_uring/memmap.c:183 io_rings_map io_uring/io_uring.c:2611 [inline] io_allocate_sqc_urings+0x1c0/0x650 io_uring/io_uring.c:3470 io_uring_create+0x5b5/0xc00 io_uring/io_uring.c:3692 io_uring_setup io_uring/io_uring.c:3781 [inline] ... </TASK> io_pin_pages()'s uaddr parameter came directly from the user and can be garbage. Don't just add size to it as it can overflow.
CVE-2024-53188	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix crash when unbinding If there is an error during some initialization related to firmware, the function ath12k_dp_cc_cleanup is called to release resources. However this is released again when the device is unbinded (ath12k_pci), and we get: BUG: kernel NULL pointer dereference, address: 0000000000000020 at RIP: 0010:ath12k_dp_cc_cleanup.part.0+0xb6/0x500 [ath12k] Call Trace: ath12k_dp_cc_cleanup ath12k_dp_free ath12k_core_deinit ath12k_pci_remove ... The issue is always reproducible from a VM because the MSI addressing initialization is failing. In order to fix the issue, just set to NULL the released structure in ath12k_dp_cc_cleanup at the end.
CVE-2024-53190	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: Drastically reduce the attempts to read efuse in case of failures Syzkaller reported a hung task with uevent_show() on stack trace. That specific issue was addressed by another commit [0], but even with that fix applied (for example, running v6.12-rc5) we face another type of hung task that comes from the same reproducer [1]. By investigating that, we could narrow it to the following path: (a) Syzkaller emulates a Realtek USB WiFi adapter using raw-gadget and dummy_hcd infrastructure. (b) During the probe of rtl8192cu, the driver ends-up performing an efuse read procedure (which is related to EEPROM load IIUC), and here lies the issue: the function read_efuse() calls read_efuse_byte() many times, as loop iterations depending on the efuse size (in our example, 512 in total). This procedure for reading efuse bytes relies in a loop that performs an I/O read up to *10k* times in case of failures. We measured the time of the loop inside read_efuse_byte() alone, and in this reproducer (which involves the dummy_hcd emulation layer), it takes 15 seconds each. As a consequence, we have the driver stuck in its probe routine for big time, exposing a stack trace like below if we attempt to reboot the system, for example: task:kworker/0:3 state:D stack:0 pid:662 tgid:662 ppid:2 flags:0x00004000 Workqueue: usb_hub_wq hub_event Call Trace: __schedule+0xe22/0xeb6 schedule_timeout+0xe7/0x132 __wait_for_common+0xb5/0x12e usb_start_wait_urb+0xc5/0x1ef ? usb_alloc_urb+0x95/0xa4 usb_control_msg+0xff/0x184 _usbctrl_vendorreq_sync+0xa0/0x161 _usb_read_sync+0xb3/0xc5 read_efuse_byte+0x13c/0x146 read_efuse+0x351/0x5f0 efuse_read_all_map+0x42/0x52 rtl_efuse_shadow_map_update+0x60/0xef rtl_get_hwinfo+0x5d/0x1c2 rtl92cu_read_eeeprom_info+0x10a/0x8d5 ? rtl92cu_read_chip_version+0x14f/0x17e rtl_usb_probe+0x323/0x851 usb_probe_interface+0x278/0x34b really_probe+0x202/0x4a4 __driver_probe_device+0x166/0x1b2 driver_probe_device+0x2f/0xd8 [...] We propose hereby to drastically reduce the attempts of doing the I/O reads in case of failures, restricted to USB devices (given that they're inherently slower than PCIe ones). By retrying up to 10 times (instead of 10000), we got reponsiveness in the reproducer, while seems reasonable to believe that there's no sane USB device implementation in the field requiring this amount of retries at every I/O read in order to properly work. Based on that assumption, it'd be good to have it backported to stable but maybe not since driver implementation (the 10k number comes from day 0), perhaps up to 6.x series makes sense. [0] Commit 15fffc6a5624 ("driver core: Fix uevent_show() vs driver detach race") [1] A note about that: this syzkaller report presents multiple reproducers that differs by the type of emulated USB device. For this specific case, check the entry from 2024/08/08 06:23 in the list of crashes; the C repro is available at https://syzkaller.appspot.com/text?tag=ReproC&x=1521fc83980000 .
CVE-2024-53191	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix warning when unbinding If there is an error during some initialization related to firmware, the buffers dp->tx_ring[i].tx_status are released. However this is released again when the device is unbinded (ath12k_pci), and we get: WARNING: CPU: 0 PID: 2098 at mm/slub.c:4689 free_large_kmalloc+0x4d/0x80 Call Trace: free_large_kmalloc ath12k_dp_free ath12k_core_deinit ath12k_pci_remove ... The issue is always reproducible from a VM because the MSI addressing initialization is failing. In order to fix the issue, just set the buffers to NULL after releasing in order to avoid the double free.

CVE ID	Description
CVE-2024-53194	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix use-after-free of slot->bus on hot remove Dennis reports a boot crash on recent Lenovo laptops with a USB4 dock. Since commit 0fc70886569c ("thunderbolt: Reset USB4 v2 host router") and commit 59a54c5f3dbd ("thunderbolt: Reset topology created by the boot firmware"), USB4 v2 and v1 Host Routers are reset on probe of the thunderbolt driver. The reset clears the Presence Detect State and Data Link Layer Link Active bits at the USB4 Host Router's Root Port and thus causes hot removal of the dock. The crash occurs when pciehp is unbound from one of the dock's Downstream Ports: pciehp creates a pci_slot on bind and destroys it on unbind. The pci_slot contains a pointer to the pci_bus below the Downstream Port, but a reference on that pci_bus is never acquired. The pci_bus is destroyed before the pci_slot, so a use-after-free ensues when pci_slot_release() accesses slot->bus. In principle this should not happen because pci_stop_bus_device() unbinds pciehp (and therefore destroys the pci_slot) before the pci_bus is destroyed by pci_remove_bus_device(). However the stacktrace provided by Dennis shows that pciehp is unbound from pci_remove_bus_device() instead of pci_stop_bus_device(). To understand the significance of this, one needs to know that the PCI core uses a two step process to remove a portion of the hierarchy: It first unbinds all drivers in the sub-hierarchy in pci_stop_bus_device() and then actually removes the devices in pci_remove_bus_device(). There is no precaution to prevent driver binding in-between pci_stop_bus_device() and pci_remove_bus_device(). In Dennis' case, it seems removal of the hierarchy by pciehp races with driver binding by pci_bus_add_devices(). pciehp is bound to the Downstream Port after pci_stop_bus_device() has run, so it is unbound by pci_remove_bus_device() instead of pci_stop_bus_device(). Because the pci_bus has already been destroyed at that point, accesses to it result in a use-after-free. One might conclude that driver binding needs to be prevented after pci_stop_bus_device() has run. However it seems risky that pci_slot points to pci_bus without holding a reference. Solely relying on correct ordering of driver unbind versus pci_bus destruction is certainly not defensive programming. If pci_slot has a need to access data in pci_bus, it ought to acquire a reference. Amend pci_create_slot() accordingly. Dennis reports that the crash is not reproducible with this change. Abridged stacktrace: pcieport 0000:00:07.0: PME: Signaling with IRQ 156 pcieport 0000:00:07.0: pciehp: Slot #12 AttnBtn- PwrCtrl-MRL- AttnInd- PwrInd- HotPlug+ Surprise+ Interlock- NoCompl+ IbPresDis- LLActRep+ pci_bus 0000:20: dev 00, created physical slot 12 pcieport 0000:00:07.0: pciehp: Slot(12): Card not present ... pcieport 0000:21:02.0: pciehp: pcie_disable_notification: SLOTCTRL d8 write cmd 0 Oops: general protection fault, probably for non-canonical address 0x6b6b6b6b6b6b6b6b: 0000 [#1] PREEMPT SMP NOPTI CPU: 13 UID: 0 PID: 134 Comm: irq/156-pciehp Not tainted 6.11.0-devel+ #1 RIP: 0010:dev_driver_string+0x12/0x40 pci_destroy_slot pciehp_remove pcie_port_remove_service device_release_driver_internal bus_remove_device device_del device_unregister remove_iter device_for_each_child pcie_portdrv_remove pci_device_remove device_release_driver_internal bus_remove_device device_del pci_remove_bus_device (recursive invocation) pci_remove_bus_device pciehp_unconfigure_device pciehp_disable_slot pciehp_handle_presence_or_link_change pciehp_ist
CVE-2024-53195	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Get rid of userspace_irqchip_in_use Improper use of userspace_irqchip_in_use led to syzbot hitting the following WARN_ON() in kvm_timer_update_irq(): WARNING: CPU: 0 PID: 3281 at arch/arm64/kvm/arch_timer.c:459 kvm_timer_update_irq+0x21c/0x394 Call trace: kvm_timer_update_irq+0x21c/0x394 arch/arm64/kvm/arch_timer.c:459 kvm_timer_vcpu_reset+0x158/0x684 arch/arm64/kvm/arch_timer.c:968 kvm_reset_vcpu+0x3b4/0x560 arch/arm64/kvm/reset.c:264 kvm_vcpu_set_target arch/arm64/kvm/arm.c:1553 [inline] kvm_arch_vcpu_ioctl_vcpu_init arch/arm64/kvm/arm.c:1573 [inline] kvm_arch_vcpu_ioctl+0x112c/0x1b3c arch/arm64/kvm/arm.c:1695 kvm_vcpu_ioctl+0x4ec/0xf74 virt/kvm/kvm_main.c:4658 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __arm64_sys_ioctl+0x108/0x184 fs/ioctl.c:893 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x78/0x1b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0xe8/0x1b0 arch/arm64/kernel/syscall.c:132 do_el0_svc+0x40/0x50 arch/arm64/kernel/syscall.c:151 el0_svc+0x54/0x14c arch/arm64/kernel/entry-common.c:712 el0t_64_sync_handler+0x84/0xfc arch/arm64/kernel/entry-common.c:730 el0t_64_sync+0x190/0x194 arch/arm64/kernel/entry.S:598 The following sequence led to the scenario: - Userspace creates a VM and a vCPU. - The vCPU is initialized with KVM_ARM_VCPU_PMU_V3 during KVM_ARM_VCPU_INIT. - Without any other setup, such as vGIC or vPMU, userspace issues KVM_RUN on the vCPU. Since the vPMU is requested, but not setup, kvm_arm_pmu_v3_enable() fails in kvm_arch_vcpu_run_pid_change(). As a result, KVM_RUN returns after enabling the timer, but before incrementing 'userspace_irqchip_in_use': kvm_arch_vcpu_run_pid_change() ret = kvm_arm_pmu_v3_enable() if (!vcpu->arch.pmu.created) return -EINVAL; if (ret) return ret; [...] if (!irqchip_in_kernel(kvm)) static_branch_inc(&userspace_irqchip_in_use); - Userspace ignores the error and issues KVM_ARM_VCPU_INIT again. Since the timer is already enabled, control moves through the following flow, ultimately hitting the WARN_ON(): kvm_timer_vcpu_reset() if (timer->enabled) kvm_timer_update_irq() if (!userspace_irqchip()) ret = kvm_vgic_inject_irq() ret = vgic_lazy_init() if (unlikely(!vgic_initialized(kvm))) if (kvm->arch.vgic.vgic_model != KVM_DEV_TYPE_ARM_VGIC_V2) return -EBUSY; WARN_ON(ret); Theoretically, since userspace_irqchip_in_use's functionality can be simply replaced by 'irqchip_in_kernel()', get rid of the static key to avoid the mismanagement, which also helps with the syzbot issue.

CVE ID	Description
CVE-2024-53196	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Don't retire aborted MMIO instruction Returning an abort to the guest for an unsupported MMIO access is a documented feature of the KVM UAPI. Nevertheless, it's clear that this plumbing has seen limited testing, since userspace can trivially cause a WARN in the MMIO return: WARNING: CPU: 0 PID: 30558 at arch/arm64/include/asm/kvm_emulate.h:536 kvm_handle_mmio_return+0x46c/0x5c4 arch/arm64/include/asm/kvm_emulate.h:536 Call trace: kvm_handle_mmio_return+0x46c/0x5c4 arch/arm64/include/asm/kvm_emulate.h:536 kvm_arch_vcpu_ioctl_run+0x98/0x15b4 arch/arm64/kvm/arm.c:1133 kvm_vcpu_ioctl+0x75c/0xa78 virt/kvm/kvm_main.c:4487 __do_sys_ioctl fs/ioctl.c:51 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __arm64_sys_ioctl+0x14c/0x1c8 fs/ioctl.c:893 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x1e0/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc+0x38/0x68 arch/arm64/kernel/entry-common.c:712 el0t_64_sync_handler+0x90/0xfc arch/arm64/kernel/entry-common.c:730 el0t_64_sync+0x190/0x194 arch/arm64/kernel/entry.S:598 The splat is complaining that KVM is advancing PC while an exception is pending, i.e. that KVM is retiring the MMIO instruction despite a pending synchronous external abort. Womp womp. Fix the glaring UAPI bug by skipping over all the MMIO emulation in case there is a pending synchronous exception. Note that while userspace is capable of pending an asynchronous exception (SError, IRQ, or FIQ), it is still safe to retire the MMIO instruction in this case as (1) they are by definition asynchronous, and (2) KVM relies on hardware support for pending/delivering these exceptions instead of the software state machine for advancing PC.
CVE-2024-53197	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix potential out-of-bound accesses for Extigy and Mbox devices A bogus device can provide a bNumConfigurations value that exceeds the initial value used in usb_get_configuration for allocating dev->config. This can lead to out-of-bounds accesses later, e.g. in usb_destroy_configuration.
CVE-2024-53198	In the Linux kernel, the following vulnerability has been resolved: xen: Fix the issue of resource not being properly released in xenbus_dev_probe() This patch fixes an issue in the function xenbus_dev_probe(). In the xenbus_dev_probe() function, within the if (err) branch at line 313, the program incorrectly returns err directly without releasing the resources allocated by err = drv->probe(dev, id). As the return value is non-zero, the upper layers assume the processing logic has failed. However, the probe operation was performed earlier without a corresponding remove operation. Since the probe actually allocates resources, failing to perform the remove operation could lead to problems. To fix this issue, we followed the resource release logic of the xenbus_dev_remove() function by adding a new block fail_remove before the fail_put block. After entering the branch if (err) at line 313, the function will use a goto statement to jump to the fail_remove block, ensuring that the previously acquired resources are correctly released, thus preventing the reference count leak. This bug was identified by an experimental static analysis tool developed by our team. The tool specializes in analyzing reference count operations and detecting potential issues where resources are not properly managed. In this case, the tool flagged the missing release operation as a potential problem, which led to the development of this patch.
CVE-2024-53200	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null check for pipe_ctx->plane_state in hwss_setup_dpp This commit addresses a null pointer dereference issue in hwss_setup_dpp(). The issue could occur when pipe_ctx->plane_state is null. The fix adds a check to ensure `pipe_ctx->plane_state` is not null before accessing. This prevents a null pointer dereference.
CVE-2024-53201	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null check for pipe_ctx->plane_state in dcn20_program_pipe This commit addresses a null pointer dereference issue in dcn20_program_pipe(). Previously, commit 8e4ed3cf1642 ("drm/amd/display: Add null check for pipe_ctx->plane_state in dcn20_program_pipe") partially fixed the null pointer dereference issue. However, in dcn20_update_dchubp_dpp(), the variable pipe_ctx is passed in, and plane_state is accessed again through pipe_ctx. Multiple if statements directly call attributes of plane_state, leading to potential null pointer dereference issues. This patch adds necessary null checks to ensure stability.
CVE-2024-53202	In the Linux kernel, the following vulnerability has been resolved: firmware_loader: Fix possible resource leak in fw_log_firmware_info() The alg instance should be released under the exception path, otherwise there may be resource leak here. To mitigate this, free the alg instance with crypto_free_shash when kmallocc fails.
CVE-2024-53203	In the Linux kernel, the following vulnerability has been resolved: usb: typec: fix potential array underflow in ucsi_ccg_sync_control() The "command" variable can be controlled by the user via debugfs. The worry is that if con_index is zero then "&ucsi->connector[con_index - 1]" would be an array underflow.

CVE ID	Description
CVE-2024-53207	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix possible deadlocks This fixes possible deadlocks like the following caused by hci_cmd_sync_dequeue causing the destroy function to run: INFO: task kworker/u19:0:143 blocked for more than 120 seconds. Tainted: G W O 6.8.0-2024-03-19-intel-next-iLS-24ww14 #1 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:kworker/u19:0 state:D stack:0 pid:143 tgid:143 ppid:2 flags:0x00004000 Workqueue: hci0 hci_cmd_sync_work [bluetooth] Call Trace: <TASK> __schedule+0x374/0xaf0 schedule+0x3c/0xf0 schedule_preempt_disabled+0x1c/0x30 __mutex_lock.constprop.0+0x3ef/0x7a0 __mutex_lock_slowpath+0x13/0x20 mutex_lock+0x3c/0x50 mgmt_set_connectable_complete+0xa4/0x150 [bluetooth] ? kfree+0x211/0x2a0 hci_cmd_sync_dequeue+0xae/0x130 [bluetooth] ? __pfx_cmd_complete_rsp+0x10/0x10 [bluetooth] cmd_complete_rsp+0x26/0x80 [bluetooth] mgmt_pending_foreach+0x4d/0x70 [bluetooth] __mgmt_power_off+0x8d/0x180 [bluetooth] ? __raw_spin_unlock_irq+0x23/0x40 hci_dev_close_sync+0x445/0x5b0 [bluetooth] hci_set_powered_sync+0x149/0x250 [bluetooth] set_powered_sync+0x24/0x60 [bluetooth] hci_cmd_sync_work+0x90/0x150 [bluetooth] process_one_work+0x13e/0x300 worker_thread+0x2f7/0x420 ? __pfx_worker_thread+0x10/0x10 kthread+0x107/0x140 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x3d/0x60 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1b/0x30 </TASK>
CVE-2024-53208	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix slab-use-after-free Read in set_powered_sync This fixes the following crash: ===== BUG: KASAN: slab-use-after-free in set_powered_sync+0x3a/0xc0 net/bluetooth/mgmt.c:1353 Read of size 8 at addr ffff888029b4dd18 by task kworker/u9:0/54 CPU: 1 UID: 0 PID: 54 Comm: kworker/u9:0 Not tainted 6.11.0-rc6-syzkaller-01155-gf723224742fc #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:93 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:119 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 q kasan_report+0x143/0x180 mm/kasan/report.c:601 set_powered_sync+0x3a/0xc0 net/bluetooth/mgmt.c:1353 hci_cmd_sync_work+0x22b/0x400 net/bluetooth/hci_sync.c:328 process_one_work kernel/workqueue.c:3231 [inline] process_scheduled_works+0xa2c/0x1830 kernel/workqueue.c:3312 worker_thread+0x86d/0xd10 kernel/workqueue.c:3389 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Allocated by task 5247: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:370 [inline] __kasan_kmalloc+0x98/0xb0 mm/kasan/common.c:387 kasan_kmalloc include/linux/kasan.h:211 [inline] __kmalloc_cache_noprof+0x19c/0x2c0 mm/slub.c:4193 kmalloc_noprof include/linux/slub.h:681 [inline] kzalloc_noprof include/linux/slub.h:807 [inline] mgmt_pending_new+0x65/0x250 net/bluetooth/mgmt_util.c:269 mgmt_pending_add+0x36/0x120 net/bluetooth/mgmt_util.c:296 set_powered+0x3cd/0x5e0 net/bluetooth/mgmt.c:1394 hci_mgmt_cmd+0xc47/0x11d0 net/bluetooth/hci_sock.c:1712 hci_sock_sendmsg+0x7b8/0x11c0 net/bluetooth/hci_sock.c:1832 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:745 sock_write_iter+0x2dd/0x400 net/socket.c:1160 new_sync_write fs/read_write.c:497 [inline] vfs_write+0xa72/0xc90 fs/read_write.c:590 ksys_write+0x1a0/0x2c0 fs/read_write.c:643 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 5246: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:579 poison_slab_object+0xe0/0x150 mm/kasan/common.c:240 __kasan_slab_free+0x37/0x60 mm/kasan/common.c:256 kasan_slab_free include/linux/kasan.h:184 [inline] slab_free_hook mm/slub.c:2256 [inline] slab_free mm/slub.c:4477 [inline] kfree+0x149/0x360 mm/slub.c:4598 settings_rsp+0x2bc/0x390 net/bluetooth/mgmt.c:1443 mgmt_pending_foreach+0xd1/0x130 net/bluetooth/mgmt_util.c:259 __mgmt_power_off+0x112/0x420 net/bluetooth/mgmt.c:9455 hci_dev_close_sync+0x665/0x11a0 net/bluetooth/hci_sync.c:5191 hci_dev_do_close net/bluetooth/hci_core.c:483 [inline] hci_dev_close+0x112/0x210 net/bluetooth/hci_core.c:508 sock_do_ioctl+0x158/0x460 net/socket.c:1222 sock_ioctl+0x629/0x8e0 net/socket.c:1341 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xfc/0x170 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83gv entry_SYSCALL_64_after_hwframe+0x77/0x7f

CVE ID	Description
CVE-2024-53209	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix receive ring space parameters when XDP is active The MTU setting at the time an XDP multi-buffer is attached determines whether the aggregation ring will be used and the rx_skb_func handler. This is done in bnxt_set_rx_skb_mode(). If the MTU is later changed, the aggregation ring setting may need to be changed and it may become out-of-sync with the settings initially done in bnxt_set_rx_skb_mode(). This may result in random memory corruption and crashes as the HW may DMA data larger than the allocated buffer size, such as: BUG: kernel NULL pointer dereference, address: 00000000000003c0 PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 17 PID: 0 Comm: swapper/17 Kdump: loaded Tainted: G S OE 6.1.0-226bf9805506 #1 Hardware name: Wiwynn Delta Lake PVT BZA.02601.0150/Delta Lake-Class1, BIOS F0E_3A12 08/26/2021 RIP: 0010:bnxt_rx_pkt+0xe97/0x1ae0 [bnxt_en] Code: 8b 95 70 ff ff 4c 8b 9d 48 ff ff ff 66 41 89 87 b4 00 00 00 e9 0b f7 ff ff 0f b7 43 0a 49 8b 95 a8 04 00 00 25 ff 0f 00 00 <0f> b7 14 42 48 c1 e2 06 49 03 95 a0 04 00 00 0f b6 42 33f RSP: 0018:ffffa19f40cc0d18 EFLAGS: 00010202 RAX: 00000000000001e0 RBX: ffff8e2c805c6100 RCX: 00000000000007ff RDX: 0000000000000000 RSI: ffff8e2c271ab990 RDI: ffff8e2c84f12380 RBP: ffffa19f40cc0e48 R08: 000000000001000d R09: 974ea2fcdffa4cbf R10: 0000000000000000 R11: ffffa19f40cc0ff8 R12: ffff8e2c94b58980 R13: ffff8e2c952d6600 R14: 0000000000000016 R15: ffff8e2c271ab990 FS: 0000000000000000(0000) GS:ffff8e3b3f840000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000000003c0 CR3: 00000000e858a004 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 PKRU: 55555554 Call Trace: <IRQ> __bnxt_poll_work+0x1c2/0x3e0 [bnxt_en] To address the issue, we now call bnxt_set_rx_skb_mode() within bnxt_change_mtu() to properly set the AGG rings configuration and update rx_skb_func based on the new MTU value. Additionally, BNXT_FLAG_NO_AGG_RINGS is cleared at the beginning of bnxt_set_rx_skb_mode() to make sure it gets set or cleared based on the current MTU.
CVE-2024-5321	A security issue was discovered in Kubernetes clusters with Windows nodes where BUILTIN\Users may be able to read container logs and NT AUTHORITY\Authenticated Users may be able to modify container logs.
CVE-2024-53210	In the Linux kernel, the following vulnerability has been resolved: s390/iucv: MSG_PEEK causes memory leak in iucv_sock_destruct() Passing MSG_PEEK flag to skb_recv_datagram() increments skb refcount (skb->users) and iucv_sock_recvmsg() does not decrement skb refcount at exit. This results in skb memory leak in skb_queue_purge() and WARN_ON in iucv_sock_destruct() during socket close. To fix this decrease skb refcount by one if MSG_PEEK is set in order to prevent memory leak and WARN_ON. WARNING: CPU: 2 PID: 6292 at net/iucv/af_iucv.c:286 iucv_sock_destruct+0x144/0x1a0 [af_iucv] CPU: 2 PID: 6292 Comm: afiucv_test_msg Kdump: loaded Tainted: G W 6.10.0-rc7 #1 Hardware name: IBM 3931 A01 704 (z/VM 7.3.0) Call Trace: [<001587c682c4aa98>] iucv_sock_destruct+0x148/0x1a0 [af_iucv] [<001587c682c4a9d0>] iucv_sock_destruct+0x80/0x1a0 [af_iucv] [<001587c704117a32>] __sk_destruct+0x52/0x550 [<001587c704104a54>] __sock_release+0xa4/0x230 [<001587c704104c0c>] sock_close+0x2c/0x40 [<001587c702c5f5a8>] __fput+0x2e8/0x970 [<001587c7024148c4>] task_work_run+0x1c4/0x2c0 [<001587c7023b0716>] do_exit+0x996/0x1050 [<001587c7023b13aa>] do_group_exit+0x13a/0x360 [<001587c7023b1626>] __s390x_sys_exit_group+0x56/0x60 [<001587c7022bcca>] do_syscall +0x27a/0x380 [<001587c7049a6a0c>] __do_syscall+0x9c/0x160 [<001587c7049ce8a8>] system_call +0x70/0x98 Last Breaking-Event-Address: [<001587c682c4a9d4>] iucv_sock_destruct+0x84/0x1a0 [af_iucv]
CVE-2024-53213	In the Linux kernel, the following vulnerability has been resolved: net: usb: lan78xx: Fix double free issue with interrupt buffer allocation In lan78xx_probe(), the buffer `buf` was being freed twice: once implicitly through `usb_free_urb(dev->urb_intr)` with the `URB_FREE_BUFFER` flag and again explicitly by `kfree(buf)`. This caused a double free issue. To resolve this, reordered `kmallocc()` and `usb_alloc_urb()` calls to simplify the initialization sequence and removed the redundant `kfree(buf)`. Now, `buf` is allocated after `usb_alloc_urb()`, ensuring it is correctly managed by `usb_fill_int_urb()` and freed by `usb_free_urb()` as intended.

CVE ID	Description
CVE-2024-53214	In the Linux kernel, the following vulnerability has been resolved: vfio/pci: Properly hide first-in-list PCIe extended capability There are cases where a PCIe extended capability should be hidden from the user. For example, an unknown capability (i.e., capability with ID greater than PCI_EXT_CAP_ID_MAX) or a capability that is intentionally chosen to be hidden from the user. Hiding a capability is done by virtualizing and modifying the 'Next Capability Offset' field of the previous capability so it points to the capability after the one that should be hidden. The special case where the first capability in the list should be hidden is handled differently because there is no previous capability that can be modified. In this case, the capability ID and version are zeroed while leaving the next pointer intact. This hides the capability and leaves an anchor for the rest of the capability list. However, today, hiding the first capability in the list is not done properly if the capability is unknown, as struct vfio_pci_core_device->pci_config_map is set to the capability ID during initialization but the capability ID is not properly checked later when used in vfio_config_do_rw(). This leads to the following warning [1] and to an out-of-bounds access to ecap_perms array. Fix it by checking cap_id in vfio_config_do_rw(), and if it is greater than PCI_EXT_CAP_ID_MAX, use an alternative struct perm_bits for direct read only access instead of the ecap_perms array. Note that this is safe since the above is the only case where cap_id can exceed PCI_EXT_CAP_ID_MAX (except for the special capabilities, which are already checked before). [1] WARNING: CPU: 118 PID: 5329 at drivers/vfio/pci/vfio_pci_config.c:1900 vfio_pci_config_rw+0x395/0x430 [vfio_pci_core] CPU: 118 UID: 0 PID: 5329 Comm: simx-qemu-syste Not tainted 6.12.0+ #1 (snip) Call Trace: <TASK> ? show_regs+0x69/0x80 ? __warn+0x8d/0x140 ? vfio_pci_config_rw+0x395/0x430 [vfio_pci_core] ? report_bug+0x18f/0x1a0 ? handle_bug+0x63/0xa0 ? exc_invalid_op+0x19/0x70 ? asm_exc_invalid_op+0x1b/0x20 ? vfio_pci_config_rw+0x395/0x430 [vfio_pci_core] ? vfio_pci_config_rw+0x244/0x430 [vfio_pci_core] vfio_pci_rw+0x101/0x1b0 [vfio_pci_core] vfio_pci_core_read+0x1d/0x30 [vfio_pci_core] vfio_device_fops_read+0x27/0x40 [vfio] vfs_read+0xbd/0x340 ? vfio_device_fops_unl_ioctl+0xbb/0x740 [vfio] ? __rseq_handle_notify_resume+0xa4/0x4b0 __x64_sys_pread64+0x96/0xc0 x64_sys_call+0x1c3d/0x20d0 do_syscall_64+0x4d/0x120 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2024-53215	In the Linux kernel, the following vulnerability has been resolved: svcrdma: fix miss destroy percpu_counter in svc_rdma_proc_init() There's issue as follows: RPC: Registered rdma transport module. RPC: Registered rdma backchannel transport module. RPC: Unregistered rdma transport module. RPC: Unregistered rdma backchannel transport module. BUG: unable to handle page fault for address: fffffbfff80c609a PGD 123fee067 P4D 123fee067 PUD 123fea067 PMD 10c624067 PTE 0 Oops: Oops: 0000 [#1] PREEMPT SMP KASAN NOPTI RIP: 0010:percpu_counter_destroy_many+0xf7/0x2a0 Call Trace: <TASK> __die+0x1f/0x70 page_fault_oops+0x2cd/0x860 spurious_kernel_fault+0x36/0x450 do_kern_addr_fault+0xca/0x100 exc_page_fault+0x128/0x150 asm_exc_page_fault+0x26/0x30 percpu_counter_destroy_many+0xf7/0x2a0 mmdrop+0x209/0x350 finish_task_switch.isra.0+0x481/0x840 schedule_tail+0xe/0xd0 ret_from_fork+0x23/0x80 ret_from_fork_asm+0x1a/0x30 </TASK> If register_sysctl() return NULL, then svc_rdma_proc_cleanup() will not destroy the percpu counters which init in svc_rdma_proc_init(). If CONFIG_HOTPLUG_CPU is enabled, residual nodes may be in the 'percpu_counters' list. The above issue may occur once the module is removed. If the CONFIG_HOTPLUG_CPU configuration is not enabled, memory leakage occurs. To solve above issue just destroy all percpu counters when register_sysctl() return NULL.

CVE ID	Description
CVE-2024-53216	In the Linux kernel, the following vulnerability has been resolved: nfsd: release svc_expkey/svc_export with rcu_work The last reference for `cache_head` can be reduced to zero in `c_show` and `e_show` (using `rcu_read_lock` and `rcu_read_unlock`). Consequently, `svc_export_put` and `expkey_put` will be invoked, leading to two issues: 1. The `svc_export_put` will directly free ex_uuid. However, `e_show`/`c_show` will access `ex_uuid` after `cache_put`, which can trigger a use-after-free issue, shown below. <pre>===== BUG: KASAN: slab-use-after-free in svc_export_show+0x362/0x430 [nfsd] Read of size 1 at addr ff11000010fdc120 by task cat/870 CPU: 1 UID: 0 PID: 870 Comm: cat Not tainted 6.12.0-rc3+ #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x53/0x70 print_address_description.constprop.0+0x2c/0x3a0 print_report+0xb9/0x280 kasan_report+0xae/0xe0 svc_export_show+0x362/0x430 [nfsd] c_show +0x161/0x390 [sunrpc] seq_read_iter+0x589/0x770 seq_read+0x1e5/0x270 proc_reg_read +0xe1/0x140 vfs_read+0x125/0x530 ksys_read+0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e Allocated by task 830: kasan_save_stack+0x20/0x40 kasan_save_track+0x14/0x30 __kasan_kmalloc+0x8f/0xa0 __kmalloc_node_track_caller_noprof +0x1bc/0x400 kmemdup_noprof+0x22/0x50 svc_export_parse+0x8a9/0xb80 [nfsd] cache_do_downcall +0x71/0xa0 [sunrpc] cache_write_procs+0x8e/0xd0 [sunrpc] proc_reg_write+0xe1/0x140 vfs_write +0x1a5/0x6d0 ksys_write+0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe +0x76/0x7e Freed by task 868: kasan_save_stack+0x20/0x40 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x37/0x50 kfree+0xf3/0x3e0 svc_export_put +0x87/0xb0 [nfsd] cache_purge+0x17f/0x1f0 [sunrpc] nfsd_destroy_serv+0x226/0x2d0 [nfsd] nfsd_svc+0x125/0x1e0 [nfsd] write_threads+0x16a/0x2a0 [nfsd] nfsctl_transaction_write +0x74/0xa0 [nfsd] vfs_write+0x1a5/0x6d0 ksys_write+0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e 2. We cannot sleep while using `rcu_read_lock`/ `rcu_read_unlock`. However, `svc_export_put`/`expkey_put` will call path_put, which subsequently triggers a sleeping operation due to the following `dput`. ===== WARNING: suspicious RCU usage 5.10.0-dirty #141 Not tainted ----- ... Call Trace: dump_stack+0x9a/0xd0 __might_sleep+0x231/0x240 dput+0x39/0x600 path_put +0x1b/0x30 svc_export_put+0x17/0x80 e_show+0x1c9/0x200 seq_read_iter+0x63f/0x7c0 seq_read+0x226/0x2d0 vfs_read+0x113/0x2c0 ksys_read+0xc9/0x170 do_syscall_64+0x33/0x40 entry_SYSCALL_64_after_hwframe+0x67/0xd1 Fix these issues by using `rcu_work` to help release `svc_expkey`/`svc_export`. This approach allows for an asynchronous context to invoke `path_put` and also facilitates the freeing of `uuid/exp/key` after an RCU grace period.</pre>
CVE-2024-53217	In the Linux kernel, the following vulnerability has been resolved: NFSD: Prevent NULL dereference in nfsd4_process_cb_update() @ses is initialized to NULL. If __nfsd4_find_backchannel() finds no available backchannel session, setup_callback_client() will try to dereference @ses and segfault.
CVE-2024-53218	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix race in concurrent f2fs_stop_gc_thread In my test case, concurrent calls to f2fs shutdown report the following stack trace: Oops: general protection fault, probably for non-canonical address 0xc6cfff63bb5513fc: 0000 [#1] PREEMPT SMP PTI CPU: 0 UID: 0 PID: 678 Comm: f2fs_rep_shutdo Not tainted 6.12.0-rc5-next-20241029-g6fb2fa9805c5-dirty #85 Call Trace: <TASK> ? show_regs+0x8b/0xa0 ? __die_body+0x26/0xa0 ? die_addr+0x54/0x90 ? exc_general_protection+0x24b/0x5c0 ? asm_exc_general_protection+0x26/0x30 ? kthread_stop+0x46/0x390 f2fs_stop_gc_thread+0x6c/0x110 f2fs_do_shutdown+0x309/0x3a0 f2fs_ioc_shutdown+0x150/0x1c0 __f2fs_ioctl+0xffd/0x2ac0 f2fs_ioctl+0x76/0xe0 vfs_ioctl+0x23/0x60 __x64_sys_ioctl+0xce/0xf0 x64_sys_call+0x2b1b/0x4540 do_syscall_64+0xa7/0x240 entry_SYSCALL_64_after_hwframe+0x76/0x7e The root cause is a race condition in f2fs_stop_gc_thread() called from different f2fs shutdown paths: [CPU0] [CPU1] ----- f2fs_stop_gc_thread f2fs_stop_gc_thread gc_th = sbi->gc_thread gc_th = sbi->gc_thread kfree(gc_th) sbi->gc_thread = NULL < gc_th != NULL > kthread_stop(gc_th->f2fs_gc_task) //UAF The commit c7f114d864ac ("f2fs: fix to avoid use-after-free in f2fs_stop_gc_thread()") attempted to fix this issue by using a read semaphore to prevent races between shutdown and remount threads, but it fails to prevent all race conditions. Fix it by converting to write lock of s_umount in f2fs_do_shutdown().

CVE ID	Description
CVE-2024-53219	In the Linux kernel, the following vulnerability has been resolved: virtiofs: use pages instead of pointer for kernel direct IO When trying to insert a 10MB kernel module kept in a virtio-fs with cache disabled, the following warning was reported: -----[cut here]----- WARNING: CPU: 1 PID: 404 at mm/page_alloc.c:4551 Modules linked in: CPU: 1 PID: 404 Comm: insmod Not tainted 6.9.0-rc5+ #123 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) RIP: 0010: __alloc_pages+0x2bf/0x380 Call Trace: <TASK> ? __warn+0x8e/0x150 ? __alloc_pages+0x2bf/0x380 __kmalloclarge_node+0x86/0x160 __kmallocl+0x33c/0x480 virtio_fs_enqueue_req+0x240/0x6d0 virtio_fs_wake_pending_and_unlock+0x7f/0x190 queue_request_and_unlock+0x55/0x60 fuse_simple_request+0x152/0x2b0 fuse_direct_io+0x5d2/0x8c0 fuse_file_read_iter+0x121/0x160 __kernel_read+0x151/0x2d0 kernel_read+0x45/0x50 kernel_read_file+0x1a9/0x2a0 init_module_from_file+0x6a/0xe0 idempotent_init_module+0x175/0x230 __x64_sys_finit_module+0x5d/0xb0 x64_sys_call+0x1c3/0x9e0 do_syscall_64+0x3d/0xc0 entry_SYSCALL_64_after_hwframe+0x4b/0x53 </TASK> ---[end trace 0000000000000000]--- The warning is triggered as follows: 1) syscall finit_module() handles the module insertion and it invokes kernel_read_file() to read the content of the module first. 2) kernel_read_file() allocates a 10MB buffer by using vmalloc() and passes it to kernel_read(). kernel_read() constructs a kvec iter by using iov_iter_kvec() and passes it to fuse_file_read_iter(). 3) virtio-fs disables the cache, so fuse_file_read_iter() invokes fuse_direct_io(). As for now, the maximal read size for kvec iter is only limited by fc->max_read. For virtio-fs, max_read is UINT_MAX, so fuse_direct_io() doesn't split the 10MB buffer. It saves the address and the size of the 10MB-sized buffer in out_args[0] of a fuse request and passes the fuse request to virtio_fs_wake_pending_and_unlock(). 4) virtio_fs_wake_pending_and_unlock() uses virtio_fs_enqueue_req() to queue the request. Because virtiofs need DMA-able address, so virtio_fs_enqueue_req() uses kmalloc() to allocate a bounce buffer for all fuse args, copies these args into the bounce buffer and passed the physical address of the bounce buffer to virtiofsd. The total length of these fuse args for the passed fuse request is about 10MB, so copy_args_to_argbuf() invokes kmalloc() with a 10MB size parameter and it triggers the warning in __alloc_pages(): if (WARN_ON_ONCE_GFP(order > MAX_PAGE_ORDER, gfp)) return NULL; 5) virtio_fs_enqueue_req() will retry the memory allocation in a kworker, but it won't help, because kmalloc() will always return NULL due to the abnormal size and finit_module() will hang forever. A feasible solution is to limit the value of max_read for virtio-fs, so the length passed to kmalloc() will be limited. However it will affect the maximal read size for normal read. And for virtio-fs write initiated from kernel, it has the similar problem but now there is no way to limit fc->max_write in kernel. So instead of limiting both the values of max_read and max_write in kernel, introducing use_pages_for_kvec_io in fuse_conn and setting it as true in virtiofs. When use_pages_for_kvec_io is enabled, fuse will use pages instead of pointer to pass the KVEC_IO data. After switching to pages for KVEC_IO data, these pages will be used for DMA through virtio-fs. If these pages are backed by vmalloc(), {flush invalidate}_kernel_vmap_range() are necessary to flush or invalidate the cache before the DMA operation. So add two new fields in fuse_args_pages to record the base address of vmalloc area and the condition indicating whether invalidation is needed. Perform the flush in fuse_get_user_pages() for write operations and the invalidation in fuse_release_user_pages() for read operations. It may seem necessary to introduce another fie ---truncated---
CVE-2024-53220	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to account dirty data in __get_secs_required() It will trigger system panic w/ testcase in [1]: -----[cut here]----- kernel BUG at fs/f2fs/segment.c:2752! RIP: 0010:new_curseg+0xc81/0x2110 Call Trace: f2fs_allocate_data_block+0x1c91/0x4540 do_write_page+0x163/0xdf0 f2fs_outplace_write_data+0x1aa/0x340 f2fs_do_write_data_page+0x797/0x2280 f2fs_write_single_data_page+0x16cd/0x2190 f2fs_write_cache_pages+0x994/0x1c80 f2fs_write_data_pages+0x9cc/0xea0 do_writepages+0x194/0x7a0 filemap_fdatawrite_wbc+0x12b/0x1a0 __filemap_fdatawrite_range+0xbb/0xf0 file_write_and_wait_range+0xa1/0x110 f2fs_do_sync_file+0x26f/0x1c50 f2fs_sync_file+0x12b/0x1d0 vfs_fsync_range+0xfa/0x230 do_fsync+0x3d/0x80 __x64_sys_fsync+0x37/0x50 x64_sys_call+0x1e88/0x20d0 do_syscall_64+0x4b/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e The root cause is if checkpoint_disabling and lfs_mode are both on, it will trigger OPU for all overwritten data, it may cost more free segment than expected, so f2fs must account those data correctly to calculate cosumed free segments later, and return ENOSPC earlier to avoid run out of free segment during block allocation. [1] https://lore.kernel.org/fstests/20241015025106.3203676-1-chao@kernel.org/

CVE ID	Description
CVE-2024-53221	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix null-ptr-deref in f2fs_submit_page_bio() There's issue as follows when concurrently installing the f2fs.ko module and mounting the f2fs file system: KASAN: null-ptr-deref in range [0x0000000000000020-0x0000000000000027] RIP: 0010: __bio_alloc+0x2fb/0x6c0 [f2fs] Call Trace: <TASK> f2fs_submit_page_bio+0x126/0x8b0 [f2fs] __get_meta_page+0x1d4/0x920 [f2fs] get_checkpoint_version.constprop.0+0x2b/0x3c0 [f2fs] validate_checkpoint+0xac/0x290 [f2fs] f2fs_get_valid_checkpoint+0x207/0x950 [f2fs] f2fs_fill_super+0x1007/0x39b0 [f2fs] mount_bdev+0x183/0x250 legacy_get_tree+0xf4/0x1e0 vfs_get_tree+0x88/0x340 do_new_mount+0x283/0x5e0 path_mount+0x2b2/0x15b0 __x64_sys_mount+0x1fe/0x270 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e Above issue happens as the biset of the f2fs file system is not initialized before register "f2fs_fs_type". To address above issue just register "f2fs_fs_type" at the last in init_f2fs_fs(). Ensure that all f2fs file system resources are initialized.
CVE-2024-53222	In the Linux kernel, the following vulnerability has been resolved: zram: fix NULL pointer in comp_algorithm_show() LTP reported a NULL pointer dereference as followed: CPU: 7 UID: 0 PID: 5995 Comm: cat Kdump: loaded Not tainted 6.12.0-rc6+ #3 Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 pstate: 40400005 (nZcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : __pi_strcmp+0x24/0x140 lr : zcomp_available_show+0x60/0x100 [zram] sp : ffff800088b93b90 x29: ffff800088b93b90 x28: 0000000000000001 x27: 0000000000400cc0 x26: 00000000000000fe x25: ffff80007b3e2388 x24: 0000000000000000 x23: ffff80007b3e2390 x22: ffff0004041a9000 x21: ffff80007b3e2900 x20: 0000000000000000 x19: 0000000000000000 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000000 x10: ffff80007b3e2900 x9: ffff80007b3cb280 x8: 0101010101010101 x7: 0000000000000000 x6: 0000000000000000 x5: 0000000000000040 x4: 0000000000000000 x3: 00656c722d6f7a6c x2: 0000000000000000 x1: ffff80007b3e2900 x0: 0000000000000000 Call trace: __pi_strcmp+0x24/0x140 comp_algorithm_show+0x40/0x70 [zram] dev_attr_show+0x28/0x80 sysfs_kf_seq_show+0x90/0x140 kernfs_seq_show+0x34/0x48 seq_read_iter+0x1d4/0x4e8 kernfs_fop_read_iter+0x40/0x58 new_sync_read+0x9c/0x168 vfs_read+0x1a8/0x1f8 ksys_read+0x74/0x108 __arm64_sys_read+0x24/0x38 invoke_syscall+0x50/0x120 el0_svc_common.constprop.0+0xc8/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x38/0x138 el0t_64_sync_handler+0xc0/0xc8 el0t_64_sync+0x188/0x190 The zram->comp_algs[ZRAM_PRIMARY_COMP] can be NULL in zram_add() if comp_algorithm_set() has not been called. User can access the zram device by sysfs after device_add_disk(), so there is a time window to trigger the NULL pointer dereference. Move it ahead device_add_disk() to make sure when user can access the zram device, it is ready. comp_algorithm_set() is protected by zram->init_lock in other places and no such problem.
CVE-2024-53223	In the Linux kernel, the following vulnerability has been resolved: clk: ralink: mtmips: fix clocks probe order in oldest ralink SoCs Base clocks are the first in being probed and are real dependencies of the rest of fixed, factor and peripheral clocks. For old ralink SoCs RT2880, RT305x and RT3883 'xtal' must be defined first since in any other case, when fixed clocks are probed they are delayed until 'xtal' is probed so the following warning appears: WARNING: CPU: 0 PID: 0 at drivers/clk/ralink/clk-mtmips.c:499 rt3883_bus_recalc_rate+0x98/0x138 Modules linked in: CPU: 0 PID: 0 Comm: swapper Not tainted 6.6.43 #0 Stack : 805e58d0 00000000 00000004 8004f950 00000000 00000004 00000000 00000000 80669c54 80830000 80700000 805ae570 80670068 00000001 80669bf8 00000000 00000000 00000000 805ae570 80669b38 00000020 804db7dc 00000000 00000000 203a6d6d 80669b78 80669e48 70617773 00000000 805ae570 00000000 00000009 00000000 00000001 00000004 00000001 00000000 83fe43b0 00000000 ... Call Trace: [<800065d0>] show_stack+0x64/0xf4 [<804bca14>] dump_stack_lvl+0x38/0x60 [<800218ac>] __warn+0x94/0xe4 [<8002195c>] warn_slowpath_fmt+0x60/0x94 [<80259ff8>] rt3883_bus_recalc_rate+0x98/0x138 [<80254530>] __clk_register+0x568/0x688 [<80254838>] of_clk_hw_register+0x18/0x2c [<8070b910>] rt2880_clk_of_clk_init_driver+0x18c/0x594 [<8070b628>] of_clk_init+0x1c0/0x23c [<806fc448>] plat_time_init+0x58/0x18c [<806fdaf0>] time_init+0x10/0x6c [<806f9bc4>] start_kernel+0x458/0x67c ---[end trace 0000000000000000]--- When this driver was mainlined we could not find any active users of old ralink SoCs so we cannot perform any real tests for them. Now, one user of a Belkin f9k1109 version 1 device which uses RT3883 SoC appeared and reported some issues in openWRT: - https://github.com/openwrt/openwrt/issues/16054 Thus, define a 'rt2880_xtal_recalc_rate()' just returning the expected frequency 40Mhz and use it along the old ralink SoCs to have a correct boot trace with no warnings and a working clock plan from the beginning.

CVE ID	Description
CVE-2024-53224	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Move events notifier registration to be after device registration Move pkey change work initialization and cleanup from device resources stage to notifier stage, since this is the stage which handles this work events. Fix a race between the device deregistration and pkey change work by moving MLX5_IB_STAGE_DEVICE_NOTIFIER to be after MLX5_IB_STAGE_IB_REG in order to ensure that the notifier is deregistered before the device during cleanup. Which ensures there are no works that are being executed after the device has already unregistered which can cause the panic below. BUG: kernel NULL pointer dereference, address: 0000000000000000 PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP PTI CPU: 1 PID: 630071 Comm: kworker/1:2 Kdump: loaded Tainted: G W OE ----- 5.14.0-162.6.1.el9_1.x86_64 #1 Hardware name: Microsoft Corporation Virtual Machine/ Virtual Machine, BIOS 090008 02/27/2023 Workqueue: events pkey_change_handler [mlx5_ib] RIP: 0010:setup_qp+0x38/0x1f0 [mlx5_ib] Code: ee 41 54 45 31 e4 55 89 f5 53 48 89 fb 48 83 ec 20 8b 77 08 65 48 8b 04 25 28 00 00 00 48 89 44 24 18 48 8b 07 48 8d 4c 24 16 <4c> 8b 38 49 8b 87 80 0b 00 00 4c 89 ff 48 8b 80 08 05 00 00 8b 40 RSP: 0018:ffffbccc54068be20 EFLAGS: 00010282 RAX: 0000000000000000 RBX: ffff954054494128 RCX: fffffbccc54068be36 RDX: ffff954004934000 RSI: 0000000000000001 RDI: ffff954054494128 RBP: 0000000000000023 R08: ffff954001be2c20 R09: 0000000000000001 R10: ffff954001be2c20 R11: ffff9540260133c0 R12: 0000000000000000 R13: 0000000000000023 R14: 0000000000000000 R15: ffff9540ffcb0905 FS: 0000000000000000(0000) GS:ffff9540ffc80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 000000010625c001 CR4: 00000000003706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: mlx5_ib_gsi_pkey_change+0x20/0x40 [mlx5_ib] process_one_work+0x1e8/0x3c0 worker_thread+0x50/0x3b0 ? rescuer_thread+0x380/0x380 kthread+0x149/0x170 ? set_kthread_struct+0x50/0x50 ret_from_fork+0x22/0x30 Modules linked in: rdma_ucm(OE) rdma_cm(OE) iw_cm(OE) ib_ipoib(OE) ib_cm(OE) ib_umad(OE) mlx5_ib(OE) mlx5_fwctl(OE) fwctl(OE) ib_uverbs(OE) mlx5_core(OE) mlxdevm(OE) ib_core(OE) mlx_compat(OE) psample mlxfw(OE) tls knem(OE) netconsole nfsv3 nfs_acl nfs lockd grace fscache netfs qrtr rkill sunrpc intel_rapl_msr intel_rapl_common rapl hv_balloon hv_utils i2c_piix4 pcspkr joydev fuse ext4 mbcache jbd2 sr_mod sd_mod cdrom t10_pi sg ata_generic pci_hyperv pci_hyperv_intf hyperv_drm drm_shmem_helper drm_kms_helper hv_storvsc syscopyarea hv_netvsc sysfillrect sysimgblt hid_hyperv fb_sys_fops scsi_transport_fc hyperv_keyboard drm ata_piix crc10dif_pclmul crc32_pclmul crc32c_intel libata ghash_clmulni_intel hv_vmbus serio_raw [last unloaded: ib_core] CR2: 0000000000000000 ---[end trace f6f8be4ae12f7bc]---
CVE-2024-53226	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix NULL pointer dereference in hns_roce_map_mr_sg() ib_map_mr_sg() allows ULPs to specify NULL as the sg_offset argument. The driver needs to check whether it is a NULL pointer before dereferencing it.
CVE-2024-53227	In the Linux kernel, the following vulnerability has been resolved: scsi: bfa: Fix use-after-free in bfa_im_module_exit() BUG: KASAN: slab-use-after-free in __lock_acquire+0x2aca/0x3a20 Read of size 8 at addr ffff8881082d80c8 by task modprobe/25303 Call Trace: <TASK> dump_stack_lvl+0x95/0xe0 print_report+0xcb/0x620 kasan_report+0xbd/0xf0 __lock_acquire+0x2aca/0x3a20 lock_acquire+0x19b/0x520 _raw_spin_lock+0x2b/0x40 attribute_container_unregister+0x30/0x160 fc_release_transport+0x19/0x90 [scsi_transport_fc] bfa_im_module_exit+0x23/0x60 [bfa] bfa_im_init+0xdb/0xff0 [bfa] do_one_initcall+0xdc/0x550 do_init_module+0x22d/0x6b0 load_module+0x4e96/0x5ff0 init_module_from_file+0xcd/0x130 idempotent_init_module+0x330/0x620 __x64_sys_finit_module+0xb3/0x110 do_syscall_64+0xc1/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> Allocated by task 25303: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 __kasan_kmalloc+0x7f/0x90 fc_attach_transport+0x4f/0x4740 [scsi_transport_fc] bfa_im_module_init+0x17/0x80 [bfa] bfa_im_init+0x23/0xff0 [bfa] do_one_initcall+0xdc/0x550 do_init_module+0x22d/0x6b0 load_module+0x4e96/0x5ff0 init_module_from_file+0xcd/0x130 idempotent_init_module+0x330/0x620 __x64_sys_finit_module+0xb3/0x110 do_syscall_64+0xc1/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 25303: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x38/0x50 kfree+0x212/0x480 bfa_im_module_init+0x7e/0x80 [bfa] bfa_im_init+0x23/0xff0 [bfa] do_one_initcall+0xdc/0x550 do_init_module+0x22d/0x6b0 load_module+0x4e96/0x5ff0 init_module_from_file+0xcd/0x130 idempotent_init_module+0x330/0x620 __x64_sys_finit_module+0xb3/0x110 do_syscall_64+0xc1/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f Above issue happens as follows: bfa_im_init error = bfa_im_module_init() fc_release_transport(bfa_im_scsi_transport_template); if (error) goto ext; ext: bfa_im_module_exit(); fc_release_transport(bfa_im_scsi_transport_template); --> Trigger double release Don't call bfa_im_module_exit() if bfa_im_module_init() failed.
CVE-2024-53228	In the Linux kernel, the following vulnerability has been resolved: riscv: kvm: Fix out-of-bounds array access In kvm_riscv_vcpu_sbi_init() the entry->ext_idx can contain an out-of-bound index. This is used as a special marker for the base extensions, that cannot be disabled. However, when traversing the extensions, that special marker is not checked prior indexing the array. Add an out-of-bounds check to the function.

CVE ID	Description
CVE-2024-53229	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix the qp flush warnings in req When the qp is in error state, the status of WQEs in the queue should be set to error. Or else the following will appear. [920.617269] WARNING: CPU: 1 PID: 21 at drivers/infiniband/sw/rxe/rxe_completer.c:756 rxe_completer+0x989/0xcc0 [rdma_rxe] [920.617744] Modules linked in: rmbd_client(O) rtrs_client(O) rtrs_core(O) rdma_ucm rdma_cm iw_cm ib_cm crc32_generic rdma_rxe ip6_udp_tunnel udp_tunnel ib_uverbs ib_core loop brd null_blk ipv6 [920.618516] CPU: 1 PID: 21 Comm: ksoftirqd/1 Tainted: G O 6.1.113-storage+ #65 [920.618986] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 [920.619396] RIP: 0010:rxe_completer+0x989/0xcc0 [rdma_rxe] [920.619658] Code: 0f b6 84 24 3a 02 00 00 41 89 84 24 44 04 00 00 e9 2a f7 ff ff 39 ca bb 03 00 00 00 b8 0e 00 00 00 48 0f 45 d8 e9 15 f7 ff ff <0f> 0b e9 cb f8 ff ff 41 bf f5 ff ff ff e9 08 f8 ff ff 49 8d bc 24 [920.620482] RSP: 0018:ffff97b7c00bbc38 EFLAGS: 00010246 [920.620817] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000 [920.621183] RDX: ffff960dc396ebc0 RSI: 0000000000005400 RDI: ffff960dc4e2fbac [920.621548] RBP: 0000000000000000 R08: 0000000000000001 R09: ffffffffac406450 [920.621884] R10: ffffffffac4060c0 R11: 0000000000000001 R12: ffff960dc4e2f800 [920.622254] R13: ffff960dc4e2f928 R14: ffff97b7c029c580 R15: 0000000000000000 [920.622609] FS: 0000000000000000(0000) GS:ffff960ef7d00000(0000) knlGS:0000000000000000 [920.622979] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [920.623245] CR2: 00007fa056965e90 CR3: 00000001107f1000 CR4: 0000000000000060 [920.623680] Call Trace: [920.623815] <TASK> [920.623933] ? __warn+0x79/0xc0 [920.624116] ? rxe_completer+0x989/0xcc0 [rdma_rxe] [920.624356] ? report_bug+0xfb/0x150 [920.624594] ? handle_bug+0x3c/0x60 [920.624796] ? exc_invalid_op+0x14/0x70 [920.624976] ? asm_exc_invalid_op+0x16/0x20 [920.625203] ? rxe_completer+0x989/0xcc0 [rdma_rxe] [920.625474] ? rxe_completer+0x329/0xcc0 [rdma_rxe] [920.625749] rxe_do_task+0x80/0x110 [rdma_rxe] [920.626037] rxe_requester+0x625/0xde0 [rdma_rxe] [920.626310] ? rxe_cq_post+0xe2/0x180 [rdma_rxe] [920.626583] ? do_complete+0x18d/0x220 [rdma_rxe] [920.626812] ? rxe_completer+0x1a3/0xcc0 [rdma_rxe] [920.627050] rxe_do_task+0x80/0x110 [rdma_rxe] [920.627285] tasklet_action_common.constprop.0+0xa4/0x120 [920.627522] handle_softirqs+0xc2/0x250 [920.627728] ? sort_range+0x20/0x20 [920.627942] run_ksoftirqd+0x1f/0x30 [920.628158] smpboot_thread_fn+0xc7/0x1b0 [920.628334] kthread+0xd6/0x100 [920.628504] ? kthread_complete_and_exit+0x20/0x20 [920.628709] ret_from_fork+0x1f/0x30 [920.628892] </TASK>
CVE-2024-53230	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Fix possible null-ptr-deref for cppc_get_cpu_cost() cpufreq_cpu_get_raw() may return NULL if the cpu is not in policy->cpus cpu mask and it will cause null pointer dereference, so check NULL for cppc_get_cpu_cost().
CVE-2024-53231	In the Linux kernel, the following vulnerability has been resolved: cpufreq: CPPC: Fix possible null-ptr-deref for cpufreq_cpu_get_raw() cpufreq_cpu_get_raw() may return NULL if the cpu is not in policy->cpus cpu mask and it will cause null pointer dereference.
CVE-2024-53232	In the Linux kernel, the following vulnerability has been resolved: iommu/s390: Implement blocking domain This fixes a crash when surprise hot-unplugging a PCI device. This crash happens because during hot-unplug __iommu_group_set_domain_nofail() attaching the default domain fails when the platform no longer recognizes the device as it has already been removed and we end up with a NULL domain pointer and UAF. This is exactly the case referred to in the second comment in __iommu_device_set_domain() and just as stated there if we can instead attach the blocking domain the UAF is prevented as this can handle the already removed device. Implement the blocking domain to use this handling. With this change, the crash is fixed but we still hit a warning attempting to change DMA ownership on a blocked device.
CVE-2024-53233	In the Linux kernel, the following vulnerability has been resolved: unicode: Fix utf8_load() error path utf8_load() requests the symbol "utf8_data_table" and then checks if the requested UTF-8 version is supported. If it's unsupported, it tries to put the data table using symbol_put(). If an unsupported version is requested, symbol_put() fails like this: kernel BUG at kernel/module/main.c:786! RIP: 0010:__symbol_put+0x93/0xb0 Call Trace: <TASK> ? __die_body.cold+0x19/0x27 ? die+0x2e/0x50 ? do_trap+0xca/0x110 ? do_error_trap+0x65/0x80 ? __symbol_put+0x93/0xb0 ? exc_invalid_op+0x51/0x70 ? __symbol_put+0x93/0xb0 ? asm_exc_invalid_op+0x1a/0x20 ? __pfx_cmp_name+0x10/0x10 ? __symbol_put+0x93/0xb0 ? __symbol_put+0x62/0xb0 utf8_load+0xf8/0x150 That happens because symbol_put() expects the unique string that identify the symbol, instead of a pointer to the loaded symbol. Fix that by using such string.
CVE-2024-53234	In the Linux kernel, the following vulnerability has been resolved: erofs: handle NONHEAD !delta[1] lclusters gracefully syzbot reported a WARNING in iomap_iter_done: iomap_fiemap+0x73b/0x9b0 fs/iomap/fiemap.c:80 ioctl_fiemap fs/ioctl.c:220 [inline] Generally, NONHEAD lclusters won't have delta[1]==0, except for crafted images and filesystems created by pre-1.0 mkfs versions. Previously, it would immediately bail out if delta[1]==0, which led to inadequate decompressed lengths (thus FIEMAP is impacted). Treat it as delta[1]=1 to work around these legacy mkfs versions. `lclusterbits > 14` is illegal for compact indexes, error out too.

CVE ID	Description
CVE-2024-53236	In the Linux kernel, the following vulnerability has been resolved: xsk: Free skb when TX metadata options are invalid When a new skb is allocated for transmitting an xsk descriptor, i.e., for every non-multibuf descriptor or the first frag of a multibuf descriptor, but the descriptor is later found to have invalid options set for the TX metadata, the new skb is never freed. This can leak skbs until the send buffer is full which makes sending more packets impossible. Fix this by freeing the skb in the error path if we are currently dealing with the first frag, i.e., an skb allocated in this iteration of xsk_build_skb.
CVE-2024-53237	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: fix use-after-free in device_for_each_child() Syzbot has reported the following KASAN splat: BUG: KASAN: slab-use-after-free in device_for_each_child+0x18f/0x1a0 Read of size 8 at addr ffff88801f605308 by task kbnepd bnep0/4980 CPU: 0 UID: 0 PID: 4980 Comm: kbnepd bnep0 Not tainted 6.12.0-rc4-00161-gae90f6a6170d #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-2.fc40 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x100/0x190 ? device_for_each_child+0x18f/0x1a0 print_report+0x13a/0x4cb ? __virt_addr_valid+0x5e/0x590 ? __phys_addr+0xc6/0x150 ? device_for_each_child+0x18f/0x1a0 kasan_report+0xda/0x110 ? device_for_each_child+0x18f/0x1a0 ? __pfx_dev_memalloc_noio+0x10/0x10 device_for_each_child+0x18f/0x1a0 ? __pfx_device_for_each_child+0x10/0x10 pm_runtime_set_memalloc_noio+0xf2/0x180 netdev_unregister_kobject+0x1ed/0x270 unregister_netdevice_many_notify+0x123c/0x1d80 ? __mutex_trylock_common+0xde/0x250 ? __pfx_unregister_netdevice_many_notify+0x10/0x10 ? trace_contention_end+0xe6/0x140 ? __mutex_lock+0x4e7/0x8f0 ? __pfx_lock_acquire.part.0+0x10/0x10 ? rcu_is_watching+0x12/0xc0 ? unregister_netdev+0x12/0x30 unregister_netdevice_queue+0x30d/0x3f0 ? __pfx_unregister_netdevice_queue+0x10/0x10 ? __pfx_down_write+0x10/0x10 unregister_netdev+0x1c/0x30 bnep_session+0x1fb3/0x2ab0 ? __pfx_bnep_session+0x10/0x10 ? __pfx_lock_release+0x10/0x10 ? __pfx_woken_wake_function+0x10/0x10 ? __kthread_parkme+0x132/0x200 ? __pfx_bnep_session+0x10/0x10 ? kthread+0x13a/0x370 ? __pfx_bnep_session+0x10/0x10 kthread+0x2b7/0x370 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x48/0x80 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Allocated by task 4974: kasan_save_stack+0x30/0x50 kasan_save_track+0x14/0x30 __kasan_kmalloc+0xaa/0xb0 __kmalloc_noprof+0x1d1/0x440 hci_alloc_dev_priv+0x1d/0x2820 __vhci_create_device+0xef/0x7d0 vhci_write+0x2c7/0x480 vfs_write+0x6a0/0xfc0 ksys_write+0x12f/0x260 do_syscall_64+0xc7/0x250 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 4979: kasan_save_stack+0x30/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x4f/0x70 kfree+0x141/0x490 hci_release_dev+0x4d9/0x600 bt_host_release+0x6a/0xb0 device_release+0xa4/0x240 kobject_put+0x1ec/0x5a0 put_device+0x1f/0x30 vhci_release+0x81/0xf0 __fput+0x3f6/0xb30 task_work_run+0x151/0x250 do_exit+0xa79/0x2c30 do_group_exit+0xd5/0x2a0 get_signal+0x1fcd/0x2210 arch_do_signal_or_restart+0x93/0x780 syscall_exit_to_user_mode+0x140/0x290 do_syscall_64+0xd4/0x250 entry_SYSCALL_64_after_hwframe+0x77/0x7f In 'hci_conn_del_sysfs()', 'device_unregister()' may be called when an underlying (kobject) reference counter is greater than 1. This means that reparenting (happened when the device is actually freed) is delayed and, during that delay, parent controller device (hciX) may be deleted. Since the latter may create a dangling pointer to freed parent, avoid that scenario by reparenting to NULL explicitly.
CVE-2024-53239	In the Linux kernel, the following vulnerability has been resolved: ALSA: 6fire: Release resources at card release The current 6fire code tries to release the resources right after the call of usb6fire_chip_abort(). But at this moment, the card object might be still in use (as we're calling snd_card_free_when_closed()). For avoid potential UAFs, move the release of resources to the card's private_free instead of the manual call of usb6fire_chip_destroy() at the USB disconnect callback.
CVE-2024-53240	In the Linux kernel, the following vulnerability has been resolved: xen/netfront: fix crash when removing device When removing a netfront device directly after a suspend/resume cycle it might happen that the queues have not been setup again, causing a crash during the attempt to stop the queues another time. Fix that by checking the queues are existing before trying to stop them. This is XSA-465 / CVE-2024-53240.
CVE-2024-53259	quic-go is an implementation of the QUIC protocol in Go. An off-path attacker can inject an ICMP Packet Too Large packet. Since affected quic-go versions used IP_PMTUDISC_DO, the kernel would then return a "message too large" error on sendmsg, i.e. when quic-go attempts to send a packet that exceeds the MTU claimed in that ICMP packet. By setting this value to smaller than 1200 bytes (the minimum MTU for QUIC), the attacker can disrupt a QUIC connection. Crucially, this can be done after completion of the handshake, thereby circumventing any TCP fallback that might be implemented on the application layer (for example, many browsers fall back to HTTP over TCP if they're unable to establish a QUIC connection). The attacker needs to at least know the client's IP and port tuple to mount an attack. This vulnerability is fixed in 0.48.2.
CVE-2024-53427	decNumberCopy in decNumber.c in jq through 1.7.1 does not properly consider that NaN is interpreted as numeric, which has a resultant stack-based buffer overflow and out-of-bounds write, as demonstrated by use of --slurp with subtraction, such as a filter of -. when the input has a certain form of digit string with NaN (e.g., "1 NaN123" immediately followed by many more digits).

CVE ID	Description
CVE-2024-53680	In the Linux kernel, the following vulnerability has been resolved: ipvs: fix UB due to uninitialized stack access in ip_vs_protocol_init() Under certain kernel configurations when building with Clang/LLVM, the compiler does not generate a return or jump as the terminator instruction for ip_vs_protocol_init(), triggering the following objtool warning during build time: vmlinux.o: warning: objtool: ip_vs_protocol_init() falls through to next function __initstub __kmod_ip_vs_rr_935_123_ip_vs_rr_init6() At runtime, this either causes an oops when trying to load the ipvs module or a boot-time panic if ipvs is built-in. This same issue has been reported by the Intel kernel test robot previously. Digging deeper into both LLVM and the kernel code reveals this to be a undefined behavior problem. ip_vs_protocol_init() uses a on-stack buffer of 64 chars to store the registered protocol names and leaves it uninitialized after definition. The function calls strlen() when concatenating protocol names into the buffer. With CONFIG_FORTIFY_SOURCE strlen() performs an extra step to check whether the last byte of the input char buffer is a null character (commit 3009f891bb9f ("fortify: Allow strlen() and strlen() to pass compile-time known lengths")). This, together with possibly other configurations, cause the following IR to be generated: define hidden i32 @ip_vs_protocol_init() local_unnamed_addr #5 section ".init.text" align 16 !kcfi_type !29 { %1 = alloca [64 x i8], align 16 ... 14: ; preds = %11 %15 = getelementptr inbounds i8, ptr %1, i64 63 %16 = load i8, ptr %15, align 1 %17 = tail call i1 @llvm.is.constant.i8(i8 %16) %18 = icmp eq i8 %16, 0 %19 = select i1 %17, i1 %18, i1 false br i1 %19, label %20, label %23 20: ; preds = %14 %21 = call i64 @strlen(ptr noundef nonnull dereferenceable(1) %1) #23 ... 23: ; preds = %14, %11, %20 %24 = call i64 @strlen(ptr noundef nonnull dereferenceable(1) %1, i64 noundef 64) #24 ... } The above code calculates the address of the last char in the buffer (value %15) and then loads from it (value %16). Because the buffer is never initialized, the LLVM GVN pass marks value %16 as undefined: %13 = getelementptr inbounds i8, ptr %1, i64 63 br i1 undef, label %14, label %17 This gives later passes (SCCP, in particular) more DCE opportunities by propagating the undef value further, and eventually removes everything after the load on the uninitialized stack location: define hidden i32 @ip_vs_protocol_init() local_unnamed_addr #0 section ".init.text" align 16 !kcfi_type !11 { %1 = alloca [64 x i8], align 16 ... 12: ; preds = %11 %13 = getelementptr inbounds i8, ptr %1, i64 63 unreachable } In this way, the generated native code will just fall through to the next function, as LLVM does not generate any code for the unreachable IR instruction and leaves the function without a terminator. Zero the on-stack buffer to avoid this possible UB.
CVE-2024-53685	In the Linux kernel, the following vulnerability has been resolved: ceph: give up on paths longer than PATH_MAX If the full path to be built by ceph_mdsc_build_path() happens to be longer than PATH_MAX, then this function will enter an endless (retry) loop, effectively blocking the whole task. Most of the machine becomes unusable, making this a very simple and effective DoS vulnerability. I cannot imagine why this retry was ever implemented, but it seems rather useless and harmful to me. Let's remove it and fail with ENAMETOOLONG instead.

CVE ID	Description
CVE-2024-53687	In the Linux kernel, the following vulnerability has been resolved: riscv: Fix IPIs usage in kfence_protect_page() flush_tlb_kernel_range() may use IPIs to flush the TLBs of all the cores, which triggers the following warning when the irqs are disabled: [3.455330] WARNING: CPU: 1 PID: 0 at kernel/smp.c:815 smp_call_function_many_cond+0x452/0x520 [3.456647] Modules linked in: [3.457218] CPU: 1 UID: 0 PID: 0 Comm: swapper/1 Not tainted 6.12.0-rc7-00010-g91d3de7240b8 #1 [3.457416] Hardware name: QEMU QEMU Virtual Machine, BIOS [3.457633] epc : smp_call_function_many_cond+0x452/0x520 [3.457736] ra : on_each_cpu_cond_mask+0x1e/0x30 [3.457786] epc : ffffffff800b669a ra : ffffffff800b67c2 sp : ff2000000000bb50 [3.457824] gp : ffffffff815212b8 tp : ff6000008014f080 t0 : 000000000000003f [3.457859] t1 : ffffffff815221e0 t2 : 000000000000000f s0 : ff2000000000bc10 [3.457920] s1 : 0000000000000040 a0 : ffffffff815221e0 a1 : 0000000000000001 [3.457953] a2 : 0000000000010000 a3 : 0000000000000003 a4 : 0000000000000000 [3.458006] a5 : 0000000000000000 a6 : ffffffff815221e0 a7 : 0000000000000000 [3.458042] s2 : ffffffff815223be s3 : 0000000000000000 s4 : ff600001ffe38fc0 [3.458076] s5 : ff600001ff950d00 s6 : 0000000200000120 s7 : 0000000000000001 [3.458109] s8 : 0000000000000001 s9 : ff60000080841ef0 s10: 0000000000000001 [3.458141] s11: ffffffff81524812 t3 : 0000000000000001 t4 : ff60000080092bc0 [3.458172] t5 : 0000000000000000 t6 : ff200000000236d0 [3.458203] status: 0000000200000100 badaddr: ffffffff800b669a cause: 0000000000000003 [3.458373] [<ffffffffff800b669a>] smp_call_function_many_cond+0x452/0x520 [3.458593] [<ffffffffff800b67c2>] on_each_cpu_cond_mask+0x1e/0x30 [3.458625] [<ffffffffff8000e4ca>] __flush_tlb_range+0x118/0x1ca [3.458656] [<ffffffffff8000e6b2>] flush_tlb_kernel_range+0x1e/0x26 [3.458683] [<ffffffffff801ea56a>] kfence_protect+0xc0/0xcce [3.458717] [<ffffffffff801e9456>] kfence_guarded_free+0xc6/0x1c0 [3.458742] [<ffffffffff801e9d6c>] __kfence_free+0x62/0xc6 [3.458764] [<ffffffffff801c57d8>] kfree+0x106/0x32c [3.458786] [<ffffffffff80588cf2>] detach_buf_split+0x188/0x1a8 [3.458816] [<ffffffffff8058708c>] virtqueue_get_buf_ctx+0xb6/0x1f6 [3.458839] [<ffffffffff805871da>] virtqueue_get_buf+0xe/0x16 [3.458880] [<ffffffffff80613d6a>] virtblk_done+0x5c/0xe2 [3.458908] [<ffffffffff8058766e>] vring_interrupt+0x6a/0x74 [3.458930] [<ffffffffff800747d8>] __handle_irq_event_percpu+0x7c/0xe2 [3.458956] [<ffffffffff800748f0>] handle_irq_event+0x3c/0x86 [3.458978] [<ffffffffff800786cc>] handle_simple_irq+0x9e/0xbe [3.459004] [<ffffffffff80073934>] generic_handle_domain_irq+0x1c/0x2a [3.459027] [<ffffffffff804bf87c>] imsic_handle_irq+0xba/0x120 [3.459056] [<ffffffffff80073934>] generic_handle_domain_irq+0x1c/0x2a [3.459080] [<ffffffffff804bdb76>] riscv_intc_aia_irq+0x24/0x34 [3.459103] [<ffffffffff809d0452>] handle_riscv_irq+0x2e/0x4c [3.459133] [<ffffffffff809d923e>] call_on_irq_stack+0x32/0x40 So only flush the local TLB and let the lazy kfence page fault handling deal with the faults which could happen when a core has an old protected pte version cached in its TLB. That leads to potential inaccuracies which can be tolerated when using kfence.
CVE-2024-53690	In the Linux kernel, the following vulnerability has been resolved: nilfs2: prevent use of deleted inode syzbot reported a WARNING in nilfs_rmdir. [1] Because the inode bitmap is corrupted, an inode with an inode number that should exist as a ".nilfs" file was reassigned by nilfs_mkdir for "file0", causing an inode duplication during execution. And this causes an underflow of i_nlink in rmdir operations. The inode is used twice by the same task to unmount and remove directories ".nilfs" and "file0", it trigger warning in nilfs_rmdir. Avoid to this issue, check i_nlink in nilfs_iget(), if it is 0, it means that this inode has been deleted, and iput is executed to reclaim it. [1] WARNING: CPU: 1 PID: 5824 at fs/inode.c:407 drop_nlink+0xc4/0x110 fs/inode.c:407 ... Call Trace: <TASK> nilfs_rmdir+0x1b0/0x250 fs/nilfs2/namei.c:342 vfs_rmdir+0x3a3/0x510 fs/namei.c:4394 do_rmdir+0x3b5/0x580 fs/namei.c:4453 __do_sys_rmdir fs/namei.c:4472 [inline] __se_sys_rmdir fs/namei.c:4470 [inline] __x64_sys_rmdir+0x47/0x50 fs/namei.c:4470 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-53899	virtualenv before 20.26.6 allows command injection through the activation scripts for a virtual environment. Magic template strings are not quoted correctly when replacing. NOTE: this is not the same as CVE-2024-9287.
CVE-2024-54193	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix WARN in ivpu_ipc_send_receive_internal() Move pm_runtime_set_active() to ivpu_pm_init() so when ivpu_ipc_send_receive_internal() is executed before ivpu_pm_enable() it already has correct runtime state, even if last resume was not successful.
CVE-2024-54455	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix general protection fault in ivpu_bo_list() Check if ctx is not NULL before accessing its fields.
CVE-2024-54456	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix potential buffer overflow in nfs_sysfs_link_rpc_client() name is char[64] where the size of clnt->cl_program->name remains unknown. Invoking strcat() directly will also lead to potential buffer overflow. Change them to strncpy() and strncat() to fix potential issues.

CVE ID	Description
CVE-2024-54458	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: bsg: Set bsg_queue to NULL after removal Currently, this does not cause any issues, but I believe it is necessary to set bsg_queue to NULL after removing it to prevent potential use-after-free (UAF) access.
CVE-2024-54460	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: iso: Fix circular lock in iso_listen_bis This fixes the circular locking dependency warning below, by releasing the socket lock before entering iso_listen_bis, to avoid any potential deadlock with hdev lock. [75.307983] ===== [75.307984] WARNING: possible circular locking dependency detected [75.307985] 6.12.0-rc6+ #22 Not tainted [75.307987] ----- [75.307987] kworker/u81:2/2623 is trying to acquire lock: [75.307988] ffff8fde1769da58 (sk_lock-AF_BLUETOOTH-BTPROTO_ISO) at: iso_connect_cfm+0x253/0x840 [bluetooth] [75.308021] but task is already holding lock: [75.308022] ffff8fdd61a10078 (&hdev->lock) at: hci_le_per_adv_report_evt+0x47/0x2f0 [bluetooth] [75.308053] which lock already depends on the new lock. [75.308054] the existing dependency chain (in reverse order) is: [75.308055] -> #1 (&hdev->lock){+.+.}-{3:3}: [75.308057] __mutex_lock +0xad/0xc50 [75.308061] mutex_lock_nested+0x1b/0x30 [75.308063] iso_sock_listen+0x143/0x5c0 [bluetooth] [75.308085] __sys_listen_socket+0x49/0x60 [75.308088] __x64_sys_listen+0x4c/0x90 [75.308090] x64_sys_call+0x2517/0x25f0 [75.308092] do_syscall_64+0x87/0x150 [75.308095] entry_SYSCALL_64_after_hwframe+0x76/0x7e [75.308098] -> #0 (sk_lock-AF_BLUETOOTH- BTPROTO_ISO){+.+.}-{0:0}: [75.308100] __lock_acquire+0x155e/0x25f0 [75.308103] lock_acquire+0xc9/0x300 [75.308105] lock_sock_nested+0x32/0x90 [75.308107] iso_connect_cfm +0x253/0x840 [bluetooth] [75.308128] hci_connect_cfm+0x6c/0x190 [bluetooth] [75.308155] hci_le_per_adv_report_evt+0x27b/0x2f0 [bluetooth] [75.308180] hci_le_meta_evt+0xe7/0x200 [bluetooth] [75.308206] hci_event_packet+0x21f/0x5c0 [bluetooth] [75.308230] hci_rx_work +0x3ae/0xb10 [bluetooth] [75.308254] process_one_work+0x212/0x740 [75.308256] worker_thread +0x1bd/0x3a0 [75.308258] kthread+0xe4/0x120 [75.308259] ret_from_fork+0x44/0x70 [75.308261] ret_from_fork_asm+0x1a/0x30 [75.308263] other info that might help us debug this: [75.308264] Possible unsafe locking scenario: [75.308264] CPU0 CPU1 [75.308265] ---- [75.308265] lock(&hdev->lock); [75.308267] lock(sk_lock- AF_BLUETOOTH-BTPROTO_ISO); [75.308268] lock(&hdev->lock); [75.308269] lock(sk_lock-AF_BLUETOOTH-BTPROTO_ISO); [75.308270] *** DEADLOCK *** [75.308271] 4 locks held by kworker/u81:2/2623: [75.308272] #0: ffff8fdd66e52148 (wq_completion)hci0#2){+.+.}-{0:0}, at: process_one_work+0x443/0x740 [75.308276] #1: ffffafb488b7fe48 ((work_completion)(&hdev->rx_work)), at: process_one_work+0x1ce/0x740 [75.308280] #2: ffff8fdd61a10078 (&hdev->lock){+.+.}-{3:3} at: hci_le_per_adv_report_evt +0x47/0x2f0 [bluetooth] [75.308304] #3: ffffffff6ba4900 (rcu_read_lock){....}-{1:2}, at: hci_connect_cfm+0x29/0x190 [bluetooth]
CVE-2024-54683	In the Linux kernel, the following vulnerability has been resolved: netfilter: IDLETIMER: Fix for possible ABBA deadlock Deletion of the last rule referencing a given idletimer may happen at the same time as a read of its file in sysfs: ===== WARNING: possible circular locking dependency detected 6.12.0-rc7-01692-g5e9a28f41134-dirty #594 Not tainted ----- (kn->active#48){++++}-{0:0}, at: __kernfs_remove+0x20 but task is already holding lock: fffffafa0249068 (list_mutex){+.+.}-{3:3}, at: idletimer_tg_destroy_v which lock already depends on the new lock. A simple reproducer is: #!/bin/bash while true; do iptables -A INPUT -i foo - j IDLETIMER --timeout 10 --label "testme" iptables -D INPUT -i foo -j IDLETIMER --timeout 10 --label "testme" done & while true; do cat /sys/class/xt_idletimer/timers/testme >/dev/null done Avoid this by freeing list_mutex right after deleting the element from the list, then continuing with the teardown.
CVE-2024-55069	ffmpeg 7.1 is vulnerable to Null Pointer Dereference in function iamf_read_header in /libavformat/ iamfdec.c.
CVE-2024-55639	In the Linux kernel, the following vulnerability has been resolved: net: renesas: rswitch: avoid use-after- put for a device tree node The device tree node saved in the rswitch_device structure is used at several driver locations. So passing this node to of_node_put() after the first use is wrong. Move of_node_put() for this node to exit paths.
CVE-2024-5569	A Denial of Service (DoS) vulnerability exists in the jaraco/zip library, affecting all versions prior to 3.19.1. The vulnerability is triggered when processing a specially crafted zip file that leads to an infinite loop. This issue also impacts the zipfile module of CPython, as features from the third-party zip library are later merged into CPython, and the affected code is identical in both projects. The infinite loop can be initiated through the use of functions affecting the `Path` module in both zip and zipfile, such as `joinpath`, the overloaded division operator, and `iterdir`. Although the infinite loop is not resource exhaustive, it prevents the application from responding. The vulnerability was addressed in version 3.19.1 of jaraco/zip.

CVE ID	Description
CVE-2024-55881	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Play nice with protected guests in complete_hypermcall_exit() Use is_64_bit_hypermcall() instead of is_64_bit_mode() to detect a 64-bit hypercall when completing said hypercall. For guests with protected state, e.g. SEV-ES and SEV-SNP, KVM must assume the hypercall was made in 64-bit mode as the vCPU state needed to detect 64-bit mode is unavailable. Hacking the sev_smoke_test selftest to generate a KVM_HC_MAP_GPA_RANGE hypercall via VMGEXIT trips the WARN: -----[cut here]----- WARNING: CPU: 273 PID: 326626 at arch/x86/kvm/x86.h:180 complete_hypermcall_exit+0x44/0xe0 [kvm] Modules linked in: kvm_amd kvm ... [last unloaded: kvm] CPU: 273 UID: 0 PID: 326626 Comm: sev_smoke_test Not tainted 6.12.0-smp--392e932fa0f3-feat #470 Hardware name: Google Astoria/astoria, BIOS 0.20240617.0-0 06/17/2024 RIP: 0010:complete_hypermcall_exit+0x44/0xe0 [kvm] Call Trace: <TASK> kvm_arch_vcpu_ioctl_run +0x2400/0x2720 [kvm] kvm_vcpu_ioctl+0x54f/0x630 [kvm] __se_sys_ioctl+0x6b/0xc0 do_syscall_64+0x83/0x160 entry_SYSCALL_64_after_hwframe+0x76/0x7e </TASK> ---[end trace 0000000000000000]---
CVE-2024-55916	In the Linux kernel, the following vulnerability has been resolved: Drivers: hv: util: Avoid accessing a ringbuffer not initialized yet If the KVP (or VSS) daemon starts before the VMBus channel's ringbuffer is fully initialized, we can hit the panic below: hv_utils: Registering HyperV Utility Driver hv_vmbus: registering driver hv_utils ... BUG: kernel NULL pointer dereference, address: 0000000000000000 CPU: 44 UID: 0 PID: 2552 Comm: hv_kvp_daemon Tainted: G E 6.11.0-rc3+ #1 RIP: 0010:hv_pkt_iter_first+0x12/0xd0 Call Trace: ... vmbus_recvpacket hv_kvp_onchannelcallback vmbus_on_event tasklet_action_common tasklet_action_handle_softirqs irq_exit_rcu sysvec_hyperv_stimer0 </IRQ> <TASK> asm_sysvec_hyperv_stimer0 ... kvp_register_done hvt_op_read vfs_read ksys_read __x64_sys_read This can happen because the KVP/VSS channel callback can be invoked even before the channel is fully opened: 1) as soon as hv_kvp_init() -> hvutil_transport_init() creates /dev/vmbus/hv_kvp, the kvp daemon can open the device file immediately and register itself to the driver by writing a message KVP_OP_REGISTER1 to the file (which is handled by kvp_on_msg() -> kvp_handle_handshake()) and reading the file for the driver's response, which is handled by hvt_op_read(), which calls hvt->on_read(), i.e. kvp_register_done(). 2) the problem with kvp_register_done() is that it can cause the channel callback to be called even before the channel is fully opened, and when the channel callback is starting to run, util_probe()-> vmbus_open() may have not initialized the ringbuffer yet, so the callback can hit the panic of NULL pointer dereference. To reproduce the panic consistently, we can add a "sleep(10)" for KVP in __vmbus_open(), just before the first hv_ringbuffer_init(), and then we unload and reload the driver hv_utils, and run the daemon manually within the 10 seconds. Fix the panic by reordering the steps in util_probe() so the char dev entry used by the KVP or VSS daemon is not created until after vmbus_open() has completed. This reordering prevents the race condition from happening.
CVE-2024-56201	Jinja is an extensible templating engine. In versions on the 3.x branch prior to 3.1.5, a bug in the Jinja compiler allows an attacker that controls both the content and filename of a template to execute arbitrary Python code, regardless of if Jinja's sandbox is used. To exploit the vulnerability, an attacker needs to control both the filename and the contents of a template. Whether that is the case depends on the type of application using Jinja. This vulnerability impacts users of applications which execute untrusted templates where the template author can also choose the template filename. This vulnerability is fixed in 3.1.5.
CVE-2024-56326	Jinja is an extensible templating engine. Prior to 3.1.5, An oversight in how the Jinja sandboxed environment detects calls to str.format allows an attacker that controls the content of a template to execute arbitrary Python code. To exploit the vulnerability, an attacker needs to control the content of a template. Whether that is the case depends on the type of application using Jinja. This vulnerability impacts users of applications which execute untrusted templates. Jinja's sandbox does catch calls to str.format and ensures they don't escape the sandbox. However, it's possible to store a reference to a malicious string's format method, then pass that to a filter that calls it. No such filters are built-in to Jinja, but could be present through custom filters in an application. After the fix, such indirect calls are also handled by the sandbox. This vulnerability is fixed in 3.1.5.

CVE ID	Description
CVE-2024-56337	Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability in Apache Tomcat. This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.1, from 10.1.0-M1 through 10.1.33, from 9.0.0.M1 through 9.0.97. The following versions were EOL at the time the CVE was created but are known to be affected: 8.5.0 through 8.5.100. Other, older, EOL versions may also be affected. The mitigation for CVE-2024-50379 was incomplete. Users running Tomcat on a case insensitive file system with the default servlet write enabled (readonly initialisation parameter set to the non-default value of false) may need additional configuration to fully mitigate CVE-2024-50379 depending on which version of Java they are using with Tomcat: - running on Java 8 or Java 11: the system property sun.io.useCanonCaches must be explicitly set to false (it defaults to true) - running on Java 17: the system property sun.io.useCanonCaches, if set, must be set to false (it defaults to false) - running on Java 21 onwards: no further configuration is required (the system property and the problematic cache have been removed) Tomcat 11.0.3, 10.1.35 and 9.0.99 onwards will include checks that sun.io.useCanonCaches is set appropriately before allowing the default servlet to be write enabled on a case insensitive file system. Tomcat will also set sun.io.useCanonCaches to false by default where it can.
CVE-2024-56368	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix overflow in __rb_map_vma An overflow occurred when performing the following calculation: nr_pages = (nr_subbufs + 1) << subbuf_order) - pgoff; Add a check before the calculation to avoid this problem. syzbot reported this as a slab-out-of-bounds in __rb_map_vma: BUG: KASAN: slab-out-of-bounds in __rb_map_vma+0x9ab/0xae0 kernel/trace/ring_buffer.c:7058 Read of size 8 at addr ffff8880767dd2b8 by task syz-executor187/5836 CPU: 0 UID: 0 PID: 5836 Comm: syz-executor187 Not tainted 6.13.0-rc2-syzkaller-00159-gf932fb9b4074 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 11/25/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xc3/0x620 mm/kasan/report.c:489 kasan_report+0xd9/0x110 mm/kasan/report.c:602 __rb_map_vma+0x9ab/0xae0 kernel/trace/ring_buffer.c:7058 ring_buffer_map+0x56e/0x9b0 kernel/trace/ring_buffer.c:7138 tracing_buffers_mmap+0xa6/0x120 kernel/trace/trace.c:8482 call_mmap include/linux/fs.h:2183 [inline] mmap_file mm/internal.h:124 [inline] __mmap_new_file_vma mm/vma.c:2291 [inline] __mmap_new_vma mm/vma.c:2355 [inline] __mmap_region+0x1786/0x2670 mm/vma.c:2456 mmap_region+0x127/0x320 mm/mmap.c:1348 do_mmap+0xc00/0xfc0 mm/mmap.c:496 vm_mmap_pgoff+0x1ba/0x360 mm/util.c:580 ksys_mmap_pgoff+0x32c/0x5c0 mm/mmap.c:542 __do_sys_mmap arch/x86/kernel/sys_x86_64.c:89 [inline] __se_sys_mmap arch/x86/kernel/sys_x86_64.c:82 [inline] __x64_sys_mmap+0x125/0x190 arch/x86/kernel/sys_x86_64.c:82 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f The reproducer for this bug is: -----8----- #include <fcntl.h> #include <stdlib.h> #include <unistd.h> #include <asm/types.h> #include <sys/mman.h> int main(int argc, char **argv) { int page_size = getpagesize(); int fd; void *meta; system("echo 1 > /sys/kernel/tracing/buffer_size_kb"); fd = open("/sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw", O_RDONLY); meta = mmap(NULL, page_size, PROT_READ, MAP_SHARED, fd, page_size * 5); } -----8-----
CVE-2024-56369	In the Linux kernel, the following vulnerability has been resolved: drm/modes: Avoid divide by zero harder in drm_mode_vrefresh() drm_mode_vrefresh() is trying to avoid divide by zero by checking whether htotal or vtotal are zero. But we may still end up with a div-by-zero of vtotal*htotal*...

CVE ID	Description
CVE-2024-56372	In the Linux kernel, the following vulnerability has been resolved: net: tun: fix tun_napi_alloc_frags() syzbot reported the following crash [1] Issue came with the blamed commit. Instead of going through all the iov components, we keep using the first one and end up with a malformed skb. [1] kernel BUG at net/core/skbuff.c:2849 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 0 UID: 0 PID: 6230 Comm: syz-executor132 Not tainted 6.13.0-rc1-syzkaller-00407-g96b6fcc0ee41 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 11/25/2024 RIP: 0010: __pskb_pull_tail+0x1568/0x1570 net/core/skbuff.c:2848 Code: 38 c1 0f 8c 32 f1 ff ff 4c 89 f7 e8 92 96 74 f8 e9 25 f1 ff ff e8 e8 ae 09 f8 48 b5 5c 24 08 e9 eb ff ff e8 d9 ae 09 f8 90 <0f> 0b 66 0f 1f 44 00 00 90 90 90 90 90 90 90 90 90 90 90 90 90 RSP: 0018:ffff90004cbef30 EFLAGS: 00010293 RAX: ffffffff8995c347 RBX: 00000000ffffffff2 RCX: ffff88802cf45a00 RDX: 0000000000000000 RSI: 00000000ffffffff2 RDI: 0000000000000000 RBP: ffff88807df0c06a R08: ffffffff8995b084 R09: 1ffff1100fbe185c R10: dffffc0000000000 R11: fffffd100fbe185d R12: ffff888076e85d50 R13: ffff888076e85c80 R14: ffff888076e85cf4 R15: ffff888076e85c80 FS: 00007f0dca6ea6c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f0dca6ead58 CR3: 00000000119da000 CR4: 000000000003526f DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 Call Trace: <TASK> skb_cow_data+0x2da/0xcb0 net/core/skbuff.c:5284 tipc_aead_decrypt net/tipc/crypto.c:894 [inline] tipc_crypto_rcv+0x402/0x24e0 net/tipc/crypto.c:1844 tipc_rcv+0x57e/0x12a0 net/tipc/node.c:2109 tipc_l2_rcv_msg+0x2bd/0x450 net/tipc/bearer.c:668 __netif_receive_skb_list_ptype net/core/dev.c:5720 [inline] __netif_receive_skb_list_core+0x8b7/0x980 net/core/dev.c:5762 __netif_receive_skb_list net/core/dev.c:5814 [inline] netif_receive_skb_list_internal+0xa51/0xe30 net/core/dev.c:5905 gro_normal_list include/net/gro.h:515 [inline] napi_complete_done+0x2b5/0x870 net/core/dev.c:6256 napi_complete include/linux/netdevice.h:567 [inline] tun_get_user+0x2ea0/0x4890 drivers/net/tun.c:1982 tun_chr_write_iter+0x10d/0x1f0 drivers/net/tun.c:2057 do_iter_readv_writev+0x600/0x880 vfs_writev+0x376/0xba0 fs/read_write.c:1050 do_writev+0x1b6/0x360 fs/read_write.c:1096 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-56374	An issue was discovered in Django 5.1 before 5.1.5, 5.0 before 5.0.11, and 4.2 before 4.2.18. Lack of upper-bound limit enforcement in strings passed when performing IPv6 validation could lead to a potential denial-of-service attack. The undocumented and private functions clean_ipv6_address and is_valid_ipv6_address are vulnerable, as is the django.forms.GenericIPAddressField form field. (The django.db.models.GenericIPAddressField model field is not affected.)
CVE-2024-56406	A heap buffer overflow vulnerability was discovered in Perl. Release branches 5.34, 5.36, 5.38 and 5.40 are affected, including development versions from 5.33.1 through 5.41.10. When there are non-ASCII bytes in the left-hand-side of the 'tr' operator, 'S_do_trans_invmmap' can overflow the destination pointer 'd'. \$ perl -e '\$_ = "\xFF" x 1000000; tr/\xFF/\x{100}/;' Segmentation fault (core dumped) It is believed that this vulnerability can enable Denial of Service and possibly Code Execution attacks on platforms that lack sufficient defenses.
CVE-2024-56433	shadow-utils (aka shadow) 4.4 through 4.17.0 establishes a default /etc/subuid behavior (e.g., uid 100000 through 165535 for the first user account) that can realistically conflict with the uids of users defined on locally administered networks, potentially leading to account takeover, e.g., by leveraging newuidmap for access to an NFS home directory (or same-host resources in the case of remote logins by these local network users). NOTE: it may also be argued that system administrators should not have assigned uids, within local networks, that are within the range that can occur in /etc/subuid.
CVE-2024-56531	In the Linux kernel, the following vulnerability has been resolved: ALSA: caiaq: Use snd_card_free_when_closed() at disconnection The USB disconnect callback is supposed to be short and not too-long waiting. OTOH, the current code uses snd_card_free() at disconnection, but this waits for the close of all used fds, hence it can take long. It eventually blocks the upper layer USB ioctls, which may trigger a soft lockup. An easy workaround is to replace snd_card_free() with snd_card_free_when_closed(). This variant returns immediately while the release of resources is done asynchronously by the card device release at the last close. This patch also splits the code to the disconnect and the free phases; the former is called immediately at the USB disconnect callback while the latter is called from the card destructor.
CVE-2024-56532	In the Linux kernel, the following vulnerability has been resolved: ALSA: us122l: Use snd_card_free_when_closed() at disconnection The USB disconnect callback is supposed to be short and not too-long waiting. OTOH, the current code uses snd_card_free() at disconnection, but this waits for the close of all used fds, hence it can take long. It eventually blocks the upper layer USB ioctls, which may trigger a soft lockup. An easy workaround is to replace snd_card_free() with snd_card_free_when_closed(). This variant returns immediately while the release of resources is done asynchronously by the card device release at the last close. The loop of us122l->mmap_count check is dropped as well. The check is useless for the asynchronous operation with *_when_closed().

CVE ID	Description
CVE-2024-56533	In the Linux kernel, the following vulnerability has been resolved: ALSA: usx2y: Use <code>snd_card_free_when_closed()</code> at disconnection The USB disconnect callback is supposed to be short and not too-long waiting. OTOH, the current code uses <code>snd_card_free()</code> at disconnection, but this waits for the close of all used fds, hence it can take long. It eventually blocks the upper layer USB ioctls, which may trigger a soft lockup. An easy workaround is to replace <code>snd_card_free()</code> with <code>snd_card_free_when_closed()</code> . This variant returns immediately while the release of resources is done asynchronously by the card device release at the last close.
CVE-2024-56538	In the Linux kernel, the following vulnerability has been resolved: drm: zynqmp_kms: Unplug DRM device before removal Prevent userspace accesses to the DRM device from causing use-after-frees by unplugging the device before we remove it. This causes any further userspace accesses to result in an error without further calls into this driver's internals.
CVE-2024-56539	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Fix <code>memcpy()</code> field-spanning write warning in <code>mwifiex_config_scan()</code> Replace one-element array with a flexible-array member in <code>struct mwifiex_ie_types_wildcard_ssid_params</code> to fix the following warning on a MT8173 Chromebook (mt8173-elm-hana): [356.775250] -----[cut here]----- [356.784543] memcpy: detected field-spanning write (size 6) of single field "wildcard_ssid_tlv->ssid" at drivers/net/wireless/marvell/mwifiex/scan.c:904 (size 1) [356.813403] WARNING: CPU: 3 PID: 742 at drivers/net/wireless/marvell/mwifiex/scan.c:904 mwifiex_scan_networks+0x4fc/0xf28 [mwifiex] The "(size 6)" above is exactly the length of the SSID of the network this device was connected to. The source of the warning looks like: <code>ssid_len = user_scan_in->ssid_list[i].ssid_len</code> ; [...] <code>memcpy(wildcard_ssid_tlv->ssid, user_scan_in->ssid_list[i].ssid, ssid_len)</code> ; There is a <code>#define WILDCARD_SSID_TLV_MAX_SIZE</code> that uses <code>sizeof()</code> on this struct, but it already didn't account for the size of the one-element array, so it doesn't need to be changed.
CVE-2024-56540	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Prevent recovery invocation during probe and resume Refactor IPC send and receive functions to allow correct handling of operations that should not trigger a recovery process. Expose <code>ivpu_send_receive_internal()</code> , which is now utilized by the D0i3 entry, DCT initialization, and HWS initialization functions. These functions have been modified to return error codes gracefully, rather than initiating recovery. The updated functions are invoked within <code>ivpu_probe()</code> and <code>ivpu_resume()</code> , ensuring that any errors encountered during these stages result in a proper teardown or shutdown sequence. The previous approach of triggering recovery within these functions could lead to a race condition, potentially causing undefined behavior and kernel crashes due to null pointer dereferences.
CVE-2024-56543	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Skip Rx TID cleanup for self peer During peer create, dp setup for the peer is done where Rx TID is updated for all the TIDs. Peer object for self peer will not go through dp setup. When core halts, dp cleanup is done for all the peers. While cleanup, <code>rx_tid::ab</code> is accessed which causes below stack trace for self peer. WARNING: CPU: 6 PID: 12297 at drivers/net/wireless/ath/ath12k/dp_rx.c:851 Call Trace: <code>__warn+0x7b/0x1a0</code> <code>ath12k_dp_rx_fragments_cleanup+0xd2/0xe0</code> [ath12k] <code>report_bug+0x10b/0x200</code> <code>handle_bug+0x3f/0x70</code> <code>exc_invalid_op+0x13/0x60</code> <code>asm_exc_invalid_op+0x16/0x20</code> <code>ath12k_dp_rx_fragments_cleanup+0xd2/0xe0</code> [ath12k] <code>ath12k_dp_rx_fragments_cleanup+0xca/0xe0</code> [ath12k] <code>ath12k_dp_rx_peer_tid_cleanup+0x39/0xa0</code> [ath12k] <code>ath12k_mac_peer_cleanup_all+0x61/0x100</code> [ath12k] <code>ath12k_core_halt+0x3b/0x100</code> [ath12k] <code>ath12k_core_reset+0x494/0x4c0</code> [ath12k] sta object in peer will be updated when remote peer is created. Hence use <code>peer::sta</code> to detect the self peer and skip the cleanup. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.0.1-00029-QCAHKSXWPL_SILICONZ-1 Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.0.c5-00481-QCAHMTSWPL_V1.0_V2.0_SILICONZ-3
CVE-2024-56544	In the Linux kernel, the following vulnerability has been resolved: udmabuf: change folios array from <code>kmalloc</code> to <code>kvmalloc</code> When <code>PAGE_SIZE</code> 4096, <code>MAX_PAGE_ORDER</code> 10, 64bit machine, <code>page_alloc</code> only support 4MB. If above this, trigger this warn and return NULL. <code>udmabuf</code> can change size limit, if change it to 3072(3GB), and then <code>alloc</code> 3GB <code>udmabuf</code> , will fail create. [4080.876581] -----[cut here]----- [4080.876843] WARNING: CPU: 3 PID: 2015 at mm/page_alloc.c:4556 <code>__alloc_pages+0x2c8/0x350</code> [4080.878839] RIP: 0010: <code>__alloc_pages+0x2c8/0x350</code> [4080.879470] Call Trace: [4080.879473] <TASK> [4080.879473] ? <code>__alloc_pages+0x2c8/0x350</code> [4080.879475] ? <code>__warn.cold+0x8e/0xe8</code> [4080.880647] ? <code>__alloc_pages+0x2c8/0x350</code> [4080.880909] ? <code>report_bug+0xff/0x140</code> [4080.881175] ? <code>handle_bug+0x3c/0x80</code> [4080.881556] ? <code>exc_invalid_op+0x17/0x70</code> [4080.881559] ? <code>asm_exc_invalid_op+0x1a/0x20</code> [4080.882077] ? <code>udmabuf_create+0x131/0x400</code> Because <code>MAX_PAGE_ORDER</code> , <code>kmalloc</code> can max <code>alloc</code> <code>4096 * (1 << 10)</code> , 4MB memory, each array entry is <code>pointer(8byte)</code> , so can save 524288 pages(2GB). Further more, <code>costly order(order 3)</code> may not be guaranteed that it can be applied for, due to fragmentation. This patch change <code>udmabuf</code> array use <code>kvmalloc_array</code> , this can fallback <code>alloc</code> into <code>vmalloc</code> , which can guarantee allocation for any size and does not affect the performance of <code>kmalloc</code> allocations.

CVE ID	Description
CVE-2024-56545	In the Linux kernel, the following vulnerability has been resolved: HID: hyperv: streamline driver probe to avoid devres issues It was found that unloading 'hid_hyperv' module results in a devres complaint: ... hv_vmbus: unregistering driver hid_hyperv -----[cut here]----- WARNING: CPU: 2 PID: 3983 at drivers/base/devres.c:691 devres_release_group+0x1f2/0x2c0 ... Call Trace: <TASK> ? devres_release_group+0x1f2/0x2c0 ? __warn+0xd1/0x1c0 ? devres_release_group+0x1f2/0x2c0 ? report_bug+0x32a/0x3c0 ? handle_bug+0x53/0xa0 ? exc_invalid_op+0x18/0x50 ? asm_exc_invalid_op+0x1a/0x20 ? devres_release_group+0x1f2/0x2c0 ? devres_release_group+0x90/0x2c0 ? rcu_is_watching+0x15/0xb0 ? __pfx_devres_release_group+0x10/0x10 hid_device_remove+0xf5/0x220 device_release_driver_internal+0x371/0x540 ? klist_put+0xf3/0x170 bus_remove_device+0x1f1/0x3f0 device_del+0x33f/0x8c0 ? __pfx_device_del+0x10/0x10 ? cleanup_srcu_struct+0x337/0x500 hid_destroy_device+0xc8/0x130 mousevsc_remove+0xd2/0x1d0 [hid_hyperv] device_release_driver_internal+0x371/0x540 driver_detach+0xc5/0x180 bus_remove_driver+0x11e/0x2a0 ? __mutex_unlock_slowpath+0x160/0x5e0 vmbus_driver_unregister+0x62/0x2b0 [hv_vmbus] ... And the issue seems to be that the corresponding devres group is not allocated. Normally, devres_open_group() is called from __hid_device_probe() but Hyper-V HID driver overrides 'hid_dev->driver' with 'mousevsc_hid_driver' stub and basically re-implements __hid_device_probe() by calling hid_parse() and hid_hw_start() but not devres_open_group(). hid_device_probe() does not call __hid_device_probe() for it. Later, when the driver is removed, hid_device_remove() calls devres_release_group() as it doesn't check whether hdev->driver was initially overridden or not. The issue seems to be related to the commit 62c68e7cee33 ("HID: ensure timely release of driver-allocated resources") but the commit itself seems to be correct. Fix the issue by dropping the 'hid_dev->driver' override and using hid_register_driver()/hid_unregister_driver() instead. Alternatively, it would have been possible to rely on the default handling but HID_CONNECT_DEFAULT implies HID_CONNECT_HIDRAW and it doesn't seem to work for mousevsc as-is.
CVE-2024-56546	In the Linux kernel, the following vulnerability has been resolved: drivers: soc: xilinx: add the missing kfree in xlnx_add_cb_for_suspend() If we fail to allocate memory for cb_data by kmalloc, the memory allocation for eve_data is never freed, add the missing kfree() in the error handling path.

CVE ID	Description
CVE-2024-56548	In the Linux kernel, the following vulnerability has been resolved: hfsplus: don't query the device logical block size multiple times Devices block sizes may change. One of these cases is a loop device by using ioctl LOOP_SET_BLOCK_SIZE. While this may cause other issues like IO being rejected, in the case of hfsplus, it will allocate a block by using that size and potentially write out-of-bounds when hfsplus_read_wrapper calls hfsplus_submit_bio and the latter function reads a different io_size. Using a new min_io_size initially set to sb_min_blocksize works for the purposes of the original fix, since it will be set to the max between HFSPLUS_SECTOR_SIZE and the first seen logical block size. We still use the max between HFSPLUS_SECTOR_SIZE and min_io_size in case the latter is not initialized. Tested by mounting an hfsplus filesystem with loop block sizes 512, 1024 and 4096. The produced KASAN report before the fix looks like this: [419.944641] <pre>===== [419.945655] BUG: KASAN: slab-use-after-free in hfsplus_read_wrapper+0x659/0xa0a [419.946703] Read of size 2 at addr ffff88800721fc00 by task repro/10678 [419.947612] [419.947846] CPU: 0 UID: 0 PID: 10678 Comm: repro Not tainted 6.12.0-rc5-00008-gdf56e0f2f3ca #84 [419.949007] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1 04/01/2014 [419.950035] Call Trace: [419.950384] <TASK> [419.950676] dump_stack_lvl+0x57/0x78 [419.951212] ? hfsplus_read_wrapper+0x659/0xa0a [419.951830] print_report+0x14c/0x49e [419.952361] ? __virt_addr_valid+0x267/0x278 [419.952979] ? kmem_cache_debug_flags+0xc/0x1d [419.953561] ? hfsplus_read_wrapper+0x659/0xa0a [419.954231] kasan_report+0x89/0xb0 [419.954748] ? hfsplus_read_wrapper+0x659/0xa0a [419.955367] hfsplus_read_wrapper+0x659/0xa0a [419.955948] ? __pfx_hfsplus_read_wrapper+0x10/0x10 [419.956618] ? do_raw_spin_unlock+0x59/0x1a9 [419.957214] ? _raw_spin_unlock+0x1a/0x2e [419.957772] hfsplus_fill_super+0x348/0x1590 [419.958355] ? hlock_class+0x4c/0x109 [419.958867] ? __pfx_hfsplus_fill_super+0x10/0x10 [419.959499] ? __pfx_string+0x10/0x10 [419.960006] ? lock_acquire+0x3e2/0x454 [419.960532] ? bdev_name.constprop.0+0xce/0x243 [419.961129] ? __pfx_bdev_name.constprop.0+0x10/0x10 [419.961799] ? pointer+0x3f0/0x62f [419.962277] ? __pfx_pointer+0x10/0x10 [419.962761] ? vsprintf+0x6c4/0xfba [419.963178] ? __pfx_vsnprintf+0x10/0x10 [419.963621] ? setup_bdev_super +0x376/0x3b3 [419.964029] ? snprintf+0x9d/0xd2 [419.964344] ? __pfx_snprintf+0x10/0x10 [419.964675] ? lock_acquired+0x45c/0x5e9 [419.965016] ? set_blocksize+0x139/0x1c1 [419.965381] ? sb_set_blocksize+0x6d/0xae [419.965742] ? __pfx_hfsplus_fill_super+0x10/0x10 [419.966179] mount_bdev+0x12f/0x1bf [419.966512] ? __pfx_mount_bdev+0x10/0x10 [419.966886] ? vfs_parse_fs_string+0xce/0x111 [419.967293] ? __pfx_vfs_parse_fs_string +0x10/0x10 [419.967702] ? __pfx_hfsplus_mount+0x10/0x10 [419.968073] legacy_get_tree +0x104/0x178 [419.968414] vfs_get_tree+0x86/0x296 [419.968751] path_mount+0xba3/0xd0b [419.969157] ? __pfx_path_mount+0x10/0x10 [419.969594] ? kmem_cache_free+0x1e2/0x260 [419.970311] do_mount+0x99/0xe0 [419.970630] ? __pfx_do_mount+0x10/0x10 [419.971008] __do_sys_mount+0x199/0x1c9 [419.971397] do_syscall_64+0xd0/0x135 [419.971761] entry_SYSCALL_64_after_hwframe+0x76/0x7e [419.972233] RIP: 0033:0x7c3cb812972e [419.972564] Code: 48 8b 0d f5 46 0d 00 f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa 49 89 ca b8 a5 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d c2 46 0d 00 f7 d8 64 89 01 48 [419.974371] RSP: 002b:00007ffe30632548 EFLAGS: 00000286 ORIG_RAX: 00000000000000a5 [419.975048] RAX: ffffffffdfda RBX: 00007ffe306328d8 RCX: 00007c3cb812972e [419.975701] RDX: 0000000020000000 RSI: 0000000020000c80 RDI: --- truncated---</pre>
CVE-2024-56549	In the Linux kernel, the following vulnerability has been resolved: cachefiles: Fix NULL pointer dereference in object->file At present, the object->file has the NULL pointer dereference problem in ondemand-mode. The root cause is that the allocated fd and object->file lifetime are inconsistent, and the user-space invocation to anon_fd uses object->file. Following is the process that triggers the issue: [write fd] [umount] cachefiles_ondemand_fd_write_iter fscache_cookie_state_machine cachefiles_withdraw_cookie if (!file) return -ENOBUFFS cachefiles_clean_up_object cachefiles_unmark_inode_in_use fput(object->file) object->file = NULL // file NULL pointer dereference! __cachefiles_write(..., file, ...) Fix this issue by add an additional reference count to the object->file before write/llseek, and decrement after it finished.
CVE-2024-56550	In the Linux kernel, the following vulnerability has been resolved: s390/stacktrace: Use break instead of return statement arch_stack_walk_user_common() contains a return statement instead of a break statement in case store_ip() fails while trying to store a callchain entry of a user space process. This may lead to a missing pagefault_enable() call. If this happens any subsequent page fault of the process won't be resolved by the page fault handler and this in turn will lead to the process being killed. Use a break instead of a return statement to fix this.

CVE ID	Description
CVE-2024-56551	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: fix usage slab after free [+0.000021] BUG: KASAN: slab-use-after-free in drm_sched_entity_flush+0x6cb/0x7a0 [gpu_sched] [+0.000027] Read of size 8 at addr ffff8881b8605f88 by task amd_pci_unplug/2147 [+0.000023] CPU: 6 PID: 2147 Comm: amd_pci_unplug Not tainted 6.10.0+ #1 [+0.000016] Hardware name: ASUS System Product Name/ROG STRIX B550-F GAMING (WI-FI), BIOS 1401 12/03/2020 [+0.000016] Call Trace: [+0.000008] <TASK> [+0.000009] dump_stack_lvl+0x76/0xa0 [+0.000017] print_report+0xce/0x5f0 [+0.000017] ? drm_sched_entity_flush+0x6cb/0x7a0 [gpu_sched] [+0.000019] ? srso_return_thunk+0x5/0x5f [+0.000015] ? kasan_complete_mode_report_info+0x72/0x200 [+0.000016] ? drm_sched_entity_flush+0x6cb/0x7a0 [gpu_sched] [+0.000019] kasan_report+0xbe/0x110 [+0.000015] ? drm_sched_entity_flush+0x6cb/0x7a0 [gpu_sched] [+0.000023] __asan_report_load8_noabort+0x14/0x30 [+0.000014] drm_sched_entity_flush+0x6cb/0x7a0 [gpu_sched] [+0.000020] ? srso_return_thunk+0x5/0x5f [+0.000013] ? __kasan_check_write+0x14/0x30 [+0.000016] ? __pfx_drm_sched_entity_flush+0x10/0x10 [gpu_sched] [+0.000020] ? srso_return_thunk+0x5/0x5f [+0.000013] ? __kasan_check_write+0x14/0x30 [+0.000013] ? srso_return_thunk+0x5/0x5f [+0.000013] ? enable_work+0x124/0x220 [+0.000015] ? __pfx_enable_work+0x10/0x10 [+0.000013] ? srso_return_thunk+0x5/0x5f [+0.000014] ? free_large_kmalloc+0x85/0xf0 [+0.000016] drm_sched_entity_destroy+0x18/0x30 [gpu_sched] [+0.000020] amdgpu_vce_sw_fini+0x55/0x170 [amdgpu] [+0.000735] ? __kasan_check_read+0x11/0x20 [+0.000016] vce_v4_0_sw_fini+0x80/0x110 [amdgpu] [+0.000726] amdgpu_device_fini_sw+0x331/0xfc0 [amdgpu] [+0.000679] ? mutex_unlock+0x80/0xe0 [+0.000017] ? __pfx_amdgpu_device_fini_sw+0x10/0x10 [amdgpu] [+0.000662] ? srso_return_thunk+0x5/0x5f [+0.000014] ? __kasan_check_write+0x14/0x30 [+0.000013] ? srso_return_thunk+0x5/0x5f [+0.000013] ? mutex_unlock+0x80/0xe0 [+0.000016] amdgpu_driver_release_kms+0x16/0x80 [amdgpu] [+0.000663] drm_minor_release+0xc9/0x140 [drm] [+0.000081] drm_release+0x1fd/0x390 [drm] [+0.000082] __fput+0x36c/0xad0 [+0.000018] __fput_sync+0x3c/0x50 [+0.000014] __x64_sys_close+0x7d/0xe0 [+0.000014] x64_sys_call+0x1bc6/0x2680 [+0.000014] do_syscall_64+0x70/0x130 [+0.000014] ? srso_return_thunk+0x5/0x5f [+0.000014] ? irqentry_exit_to_user_mode+0x60/0x190 [+0.000015] ? srso_return_thunk+0x5/0x5f [+0.000014] ? irqentry_exit+0x43/0x50 [+0.000012] ? srso_return_thunk+0x5/0x5f [+0.000013] ? exc_page_fault+0x7c/0x110 [+0.000015] entry_SYSCALL_64_after_hwframe+0x76/0x7e [+0.000014] RIP: 0033:0x7ffff7b14f67 [+0.000013] Code: ff e8 0d 16 02 00 66 2e 0f 1f 84 00 00 00 00 0f 1f 00 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 03 00 00 00 0f 05 <48> 3d 00 f0 ff 77 41 c3 48 83 ec 18 89 7c 24 0c e8 73 ba f7 ff [+0.000026] RSP: 002b:00007fffffe378 EFLAGS: 00000246 ORIG_RAX: 0000000000000003 [+0.000019] RAX: ffffffff8fda RBX: 0000000000000000 RCX: 00007ffff7b14f67 [+0.000014] RDX: 0000000000000000 RSI: 00007ffff7f6f47a RDI: 0000000000000003 [+0.000014] RBP: 00007fffffe3a0 R08: 000055555555569890 R09: 0000000000000000 [+0.000014] R10: 0000000000000000 R11: 00000000000000246 R12: 00007fffffe5c8 [+0.000013] R13: 000055555555552a9 R14: 0000555555557d48 R15: 00007fffffd040 [+0.000020] </TASK> [+0.000016] Allocated by task 383 on cpu 7 at 26.880319s: [+0.000014] kasan_save_stack+0x28/0x60 [+0.000008] kasan_save_track+0x18/0x70 [+0.000007] kasan_save_alloc_info+0x38/0x60 [+0.000007] __kasan_kmalloc+0xc1/0xd0 [+0.000007] kmalloc_trace_noprof+0x180/0x380 [+0.000007] drm_sched_init+0x411/0xec0 [gpu_sched] [+0.000012] amdgpu_device_init+0x695f/0xa610 [amdgpu] [+0.000658] amdgpu_driver_load_kms+0x1a/0x120 [amdgpu] [+0.000662] amdgpu_pci_p ---truncated---
CVE-2024-56552	In the Linux kernel, the following vulnerability has been resolved: drm/xe/guc_submit: fix race around suspend_pending Currently in some testcases we can trigger: xe 0000:03:00.0: [drm] Assertion `exec_queue_destroyed(q)` failed! WARNING: CPU: 18 PID: 2640 at drivers/gpu/drm/xe/xe_guc_submit.c:1826 xe_guc_sched_done_handler+0xa54/0xef0 [xe] xe 0000:03:00.0: [drm] *ERROR* GT1: DEREGISTER_DONE: Unexpected engine state 0x00a1, guc_id=57 Looking at a snippet of corresponding ftrace for this GuC id we can see: 162.673311: xe_sched_msg_add: dev=0000:03:00.0, gt=1 guc_id=57, opcode=3 162.673317: xe_sched_msg_recv: dev=0000:03:00.0, gt=1 guc_id=57, opcode=3 162.673319: xe_exec_queue_scheduling_disable: dev=0000:03:00.0, 1:0x2, gt=1, width=1, guc_id=57, guc_state=0x29, flags=0x0 162.674089: xe_exec_queue_kill: dev=0000:03:00.0, 1:0x2, gt=1, width=1, guc_id=57, guc_state=0x29, flags=0x0 162.674108: xe_exec_queue_close: dev=0000:03:00.0, 1:0x2, gt=1, width=1, guc_id=57, guc_state=0xa9, flags=0x0 162.674488: xe_exec_queue_scheduling_done: dev=0000:03:00.0, 1:0x2, gt=1, width=1, guc_id=57, guc_state=0xa9, flags=0x0 162.678452: xe_exec_queue_deregister: dev=0000:03:00.0, 1:0x2, gt=1, width=1, guc_id=57, guc_state=0xa1, flags=0x0 It looks like we try to suspend the queue (opcode=3), setting suspend_pending and triggering a disable_scheduling. The user then closes the queue. However the close will also forcefully signal the suspend fence after killing the queue, later when the G2H response for disable_scheduling comes back we have now cleared suspend_pending when signalling the suspend fence, so the disable_scheduling now incorrectly tries to also deregister the queue. This leads to warnings since the queue has yet to even be marked for destruction. We also seem to trigger errors later with trying to double unregister the same queue. To fix this tweak the ordering when handling the response to ensure we don't race with a disable_scheduling that didn't actually intend to perform an unregister. The destruction path should now also correctly wait for any pending_disable before marking as destroyed. (cherry picked from commit f161809b362f027b6d72bd998e47f8f0bad60a2e)

CVE ID	Description
CVE-2024-56557	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ad7923: Fix buffer overflow for tx_buf and ring_xfer The AD7923 was updated to support devices with 8 channels, but the size of tx_buf and ring_xfer was not increased accordingly, leading to a potential buffer overflow in ad7923_update_scan_mode().
CVE-2024-56558	In the Linux kernel, the following vulnerability has been resolved: nfsd: make sure exp active before svc_export_show The function `e_show` was called with protection from RCU. This only ensures that `exp` will not be freed. Therefore, the reference count for `exp` can drop to zero, which will trigger a refcount use-after-free warning when `exp_get` is called. To resolve this issue, use `cache_get_rcu` to ensure that `exp` remains active. -----[cut here]----- refcount_t: addition on 0; use-after-free. WARNING: CPU: 3 PID: 819 at lib/refcount.c:25 refcount_warn_saturate+0xb1/0x120 CPU: 3 UID: 0 PID: 819 Comm: cat Not tainted 6.12.0-rc3+ #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 RIP: 0010:refcount_warn_saturate+0xb1/0x120 ... Call Trace: <TASK> e_show+0x20b/0x230 [nfsd] seq_read_iter+0x589/0x770 seq_read+0x1e5/0x270 vfs_read+0x125/0x530 ksys_read+0xc1/0x160 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2024-56561	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: Fix PCI domain ID release in pci_epc_destroy() pci_epc_destroy() invokes pci_bus_release_domain_nr() to release the PCI domain ID, but there are two issues: - 'epc->dev' is passed to pci_bus_release_domain_nr() which was already freed by device_unregister(), leading to a use-after-free issue. - Domain ID corresponds to the EPC device parent, so passing 'epc->dev' is also wrong. Fix these issues by passing 'epc->dev.parent' to pci_bus_release_domain_nr() and also do it before device_unregister(). [mani: reworded subject and description]
CVE-2024-56562	In the Linux kernel, the following vulnerability has been resolved: i3c: master: Fix miss free init_dyn_addr at i3c_master_put_i3c_addrs() if (dev->boardinfo && dev->boardinfo->init_dyn_addr) ^^^ here check "init_dyn_addr" i3c_bus_set_addr_slot_status(&master->bus, dev->info.dyn_addr, ...) ^^^ free "dyn_addr" Fix copy/paste error "dyn_addr" by replacing it with "init_dyn_addr".
CVE-2024-56565	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to drop all discards after creating snapshot on lvm device Piergiorgio reported a bug in bugzilla as below: -----[cut here]----- WARNING: CPU: 2 PID: 969 at fs/f2fs/segment.c:1330 RIP: 0010:__submit_discard_cmd+0x27d/0x400 [f2fs] Call Trace: __issue_discard_cmd+0x1ca/0x350 [f2fs] issue_discard_thread+0x191/0x480 [f2fs] kthread+0xc1/0x100 ret_from_fork+0x31/0x50 ret_from_fork_asm+0x1a/0x30 w/ below testcase, it can reproduce this bug quickly: - pvcreate /dev/vdb - vgcreate myvg1 /dev/vdb - lvcreate -L 1024m -n mylv1 myvg1 - mount /dev/myvg1/mylv1 /mnt/f2fs - dd if=/dev/zero of=/mnt/f2fs/file bs=1M count=20 - sync - rm /mnt/f2fs/file - sync - lvcreate -L 1024m -s -n mylv1-snapshot /dev/myvg1/mylv1 - umount /mnt/f2fs The root cause is: it will update discard_max_bytes of mounted lvm device to zero after creating snapshot on this lvm device, then, __submit_discard_cmd() will pass parameter @nr_sects w/ zero value to __blkdev_issue_discard(), it returns a NULL bio pointer, result in panic. This patch changes as below for fixing: 1. Let's drop all remained discards in f2fs_unfreeze() if snapshot of lvm device is created. 2. Checking discard_max_bytes before submitting discard during __submit_discard_cmd().

CVE ID	Description
CVE-2024-56566	In the Linux kernel, the following vulnerability has been resolved: mm/slub: Avoid list corruption when removing a slab from the full list Boot with slub_debug=UFPZ. If allocated object failed in alloc_consistency_checks, all objects of the slab will be marked as used, and then the slab will be removed from the partial list. When an object belonging to the slab got freed later, the remove_full() function is called. Because the slab is neither on the partial list nor on the full list, it eventually lead to a list corruption (actually a list poison being detected). So we need to mark and isolate the slab page with metadata corruption, do not put it back in circulation. Because the debug caches avoid all the fastpaths, reusing the frozen bit to mark slab page with metadata corruption seems to be fine. [4277.385669] list_del corruption, ffffea00044b3e50->next is LIST_POISON1 (dead00000000100) [4277.387023] -----[cut here]----- [4277.387880] kernel BUG at lib/list_debug.c:56! [4277.388680] invalid opcode: 0000 [#1] PREEMPT SMP PTI [4277.389562] CPU: 5 PID: 90 Comm: kworker/5:1 Kdump: loaded Tainted: G OE 6.6.1-1 #1 [4277.392113] Workqueue: xfs-inodegc/vda1 xfs_inodegc_worker [xfs] [4277.393551] RIP: 0010: __list_del_entry_valid_or_report+0x7b/0xc0 [4277.394518] Code: 48 91 82 e8 37 f9 9a ff 0f 0b 48 89 fe 48 c7 c7 28 49 91 82 e8 26 f9 9a ff 0f 0b 48 89 fe 48 c7 c7 58 49 91 [4277.397292] RSP: 0018:ffffc90000333b38 EFLAGS: 00010082 [4277.398202] RAX: 000000000000004e RBX: ffffea00044b3e50 RCX: 0000000000000000 [4277.399340] RDX: 0000000000000002 RSI: ffffffff828f8715 RDI: 00000000ffffffff [4277.400545] RBP: ffffea00044b3e40 R08: 0000000000000000 R09: ffff900003339f0 [4277.401710] R10: 0000000000000003 R11: ffffffff82d44088 R12: ffff888112cf9910 [4277.402887] R13: 0000000000000001 R14: 0000000000000001 R15: ffff8881000424c0 [4277.404049] FS: 0000000000000000(0000) GS:ffff88842fd40000(0000) knlGS:0000000000000000 [4277.405357] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [4277.406389] CR2: 00007f2ad0b24000 CR3: 0000000102a3a006 CR4: 00000000007706e0 [4277.407589] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [4277.408780] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [4277.410000] PKRU: 55555554 [4277.410645] Call Trace: [4277.411234] <TASK> [4277.411777] ? die+0x32/0x80 [4277.412439] ? do_trap+0xd6/0x100 [4277.413150] ? __list_del_entry_valid_or_report+0x7b/0xc0 [4277.414158] ? do_error_trap+0x6a/0x90 [4277.414948] ? __list_del_entry_valid_or_report+0x7b/0xc0 [4277.415915] ? exc_invalid_op+0x4c/0x60 [4277.416710] ? __list_del_entry_valid_or_report+0x7b/0xc0 [4277.417675] ? asm_exc_invalid_op+0x16/0x20 [4277.418482] ? __list_del_entry_valid_or_report+0x7b/0xc0 [4277.419466] ? __list_del_entry_valid_or_report+0x7b/0xc0 [4277.420410] free_to_partial_list+0x515/0x5e0 [4277.421242] ? xfs_iext_remove+0x41a/0xa10 [xfs] [4277.422298] xfs_iext_remove+0x41a/0xa10 [xfs] [4277.423316] ? xfs_inodegc_worker+0xb4/0x1a0 [xfs] [4277.424383] xfs_bmap_del_extents+0x4fe/0x7d0 [xfs] [4277.425490] __xfs_bunmap+0x50d/0x840 [xfs] [4277.426445] xfs_itruncate_extents_flags+0x13a/0x490 [xfs] [4277.427553] xfs_inactive_truncate+0xa3/0x120 [xfs] [4277.428567] xfs_inactive+0x22d/0x290 [xfs] [4277.429500] xfs_inodegc_worker+0xb4/0x1a0 [xfs] [4277.430479] process_one_work+0x171/0x340 [4277.431227] worker_thread+0x277/0x390 [4277.431962] ? __pfx_worker_thread+0x10/0x10 [4277.432752] kthread+0xf0/0x120 [4277.433382] ? __pfx_kthread+0x10/0x10 [4277.434134] ret_from_fork+0x2d/0x50 [4277.434837] ? __pfx_kthread+0x10/0x10 [4277.435566] ret_from_fork_asm+0x1b/0x30 [4277.436280] </TASK>
CVE-2024-56567	In the Linux kernel, the following vulnerability has been resolved: ad7780: fix division by zero in ad7780_write_raw() In the ad7780_write_raw() , val2 can be zero, which might lead to a division by zero error in DIV_ROUND_CLOSEST(). The ad7780_write_raw() is based on iio_info's write_raw. While val is explicitly declared that can be zero (in read mode), val2 is not specified to be non-zero.
CVE-2024-56568	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu: Defer probe of clients after smmu device bound Null pointer dereference occurs due to a race between smmu driver probe and client driver probe, when of_dma_configure() for client is called after the iommu_device_register() for smmu driver probe has executed but before the driver_bound() for smmu driver has been called. Following is how the race occurs: T1:Smmu device probe T2: Client device probe really_probe() arm_smmu_device_probe() iommu_device_register() really_probe() platform_dma_configure() of_dma_configure() of_dma_configure_id() of_iommu_configure() iommu_probe_device() iommu_init_device() arm_smmu_probe_device() arm_smmu_get_by_fwnode() driver_find_device_by_fwnode() driver_find_device() next_device() klist_next() /* null ptr assigned to smmu */ /* null ptr dereference while smmu->streamid_mask */ driver_bound() klist_add_tail() When this null smmu pointer is dereferenced later in arm_smmu_probe_device, the device crashes. Fix this by deferring the probe of the client device until the smmu device has bound to the arm smmu driver. [will: Add comment]
CVE-2024-56569	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix regression with module command in stack_trace_filter When executing the following command: # echo "write*:mod:ext3" > /sys/kernel/tracing/stack_trace_filter The current mod command causes a null pointer dereference. While commit 0f1797658b3f ("ftrace: Fix regression with module command in stack_trace_filter") has addressed part of the issue, it left a corner case unhandled, which still results in a kernel crash.

CVE ID	Description
CVE-2024-56570	In the Linux kernel, the following vulnerability has been resolved: ovl: Filter invalid inodes with missing lookup function Add a check to the ovl_dentry_weird() function to prevent the processing of directory inodes that lack the lookup function. This is important because such inodes can cause errors in overlaysfs when passed to the lowerstack.
CVE-2024-56572	In the Linux kernel, the following vulnerability has been resolved: media: platform: allegro-dvt: Fix possible memory leak in allocate_buffers_internal() The buffer in the loop should be released under the exception path, otherwise there may be a memory leak here. To mitigate this, free the buffer when allegro_alloc_buffer fails.
CVE-2024-56573	In the Linux kernel, the following vulnerability has been resolved: efi/libstub: Free correct pointer on failure cmdline_ptr is an out parameter, which is not allocated by the function itself, and likely points into the caller's stack. cmdline refers to the pool allocation that should be freed when cleaning up after a failure, so pass this instead to free_pool().
CVE-2024-56574	In the Linux kernel, the following vulnerability has been resolved: media: ts2020: fix null-ptr-deref in ts2020_probe() KASAN reported a null-ptr-deref issue when executing the following command: # echo ts2020 0x20 > /sys/bus/i2c/devices/i2c-0/new_device KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017] CPU: 53 UID: 0 PID: 970 Comm: systemd-udevd Not tainted 6.12.0-rc2+ #24 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009) RIP: 0010:ts2020_probe+0xad/0xe10 [ts2020] RSP: 0018:ffffc9000abbf598 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: 0000000000000000 RCX: ffffffff0714809 RDX: 0000000000000002 RSI: ffff88811550be00 RDI: 0000000000000010 RBP: ffff888109868800 R08: 0000000000000001 R09: fffff52001577eb6 R10: 0000000000000000 R11: ffff888109868800 R12: ffffffff0714790 R13: 1ffff92001577eb8 R14: ffffffff07190d0 R15: 0000000000000001 FS: 00007f95f13b98c0(0000) GS:ffff888149280000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000555d2634b000 CR3: 0000000152236000 CR4: 0000000000000060 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ts2020_probe+0xad/0xe10 [ts2020] i2c_device_probe+0x421/0xb40 really_probe+0x266/0x850 ... The cause of the problem is that when using sysfs to dynamically register an i2c device, there is no platform data, but the probe process of ts2020 needs to use platform data, resulting in a null pointer being accessed. Solve this problem by adding checks to platform data.
CVE-2024-56575	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Ensure power suppliers be suspended before detach them The power suppliers are always requested to suspend asynchronously, dev_pm_domain_detach() requires the caller to ensure proper synchronization of this function with power management callbacks. otherwise the detach may led to kernel panic, like below: [1457.107934] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000040 [1457.116777] Mem abort info: [1457.119589] ESR = 0x0000000096000004 [1457.123358] EC = 0x25: DABT (current EL), IL = 32 bits [1457.128692] SET = 0, FnV = 0 [1457.131764] EA = 0, S1PTW = 0 [1457.134920] FSC = 0x04: level 0 translation fault [1457.139812] Data abort info: [1457.142707] ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 [1457.148196] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [1457.153256] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [1457.158563] user pgtable: 4k pages, 48-bit VAs, pgdp=00000001138b6000 [1457.165000] [0000000000000040] pgd=0000000000000000, p4d=0000000000000000 [1457.171792] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [1457.178045] Modules linked in: v4l2_jpeg wave6_vpu_ctrl(-) [last unloaded: mxc_jpeg_encdec] [1457.186383] CPU: 0 PID: 51938 Comm: kworker/0:3 Not tainted 6.6.36-gd23d64eea511 #66 [1457.194112] Hardware name: NXP i.MX95 19X19 board (DT) [1457.199236] Workqueue: pm pm_runtime_work [1457.203247] pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=--) [1457.210188] pc : genpd_runtime_suspend+0x20/0x290 [1457.214886] lr : __rpm_callback+0x48/0x1d8 [1457.218968] sp : ffff80008250bc50 [1457.222270] x29: ffff80008250bc50 x28: 0000000000000000 x27: 0000000000000000 [1457.229394] x26: 0000000000000000 x25: 0000000000000008 x24: 00000000000f4240 [1457.236518] x23: 0000000000000000 x22: ffff00008590f0e4 x21: 0000000000000008 [1457.243642] x20: ffff80008099c434 x19: ffff00008590f000 x18: ffffffff00000000 [1457.250766] x17: 5300326563697665 x16: 645f676e696c6f6f x15: 63343a6d726f6674 [1457.257890] x14: 0000000000000004 x13: 000000000000003a4 x12: 0000000000000002 [1457.265014] x11: 0000000000000000 x10: 00000000000000a60 x9 : ffff80008250bbb0 [1457.272138] x8 : ffff000092937200 x7 : ffff0003fd6faf80 x6 : 0000000000000000 [1457.279262] x5 : 000000000410fd050 x4 : 0000000000020000 x3 : 0000000000000000 [1457.286386] x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff00008590f000 [1457.293510] Call trace: [1457.295946] genpd_runtime_suspend+0x20/0x290 [1457.300296] __rpm_callback+0x48/0x1d8 [1457.304038] rpm_callback+0x6c/0x78 [1457.307515] rpm_suspend+0x10c/0x570 [1457.311077] pm_runtime_work+0xc4/0xc8 [1457.314813] process_one_work+0x138/0x248 [1457.318816] worker_thread+0x320/0x438 [1457.322552] kthread+0x110/0x114 [1457.325767] ret_from_fork+0x10/0x20

CVE ID	Description
CVE-2024-56576	In the Linux kernel, the following vulnerability has been resolved: media: i2c: tc358743: Fix crash in the probe error path when using polling If an error occurs in the probe() function, we should remove the polling timer that was alarmed earlier, otherwise the timer is called with arguments that are already freed, which results in a crash. -----[cut here]----- WARNING: CPU: 3 PID: 0 at kernel/time/timer.c:1830 __run_timers+0x244/0x268 Modules linked in: CPU: 3 UID: 0 PID: 0 Comm: swapper/3 Not tainted 6.11.0 #226 Hardware name: Diasom DS-RK3568-SOM-EVB (DT) pstate: 804000c9 (Nzcv daIF +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : __run_timers+0x244/0x268 lr : __run_timers+0x1d4/0x268 sp : fffff80eff2baf0 x29: fffff80eff2bb50 x28: 7fffffffffffffff x27: fffff80eff2bb00 x26: fffffc080f669c0 x25: fffff80efef6bf0 x24: fffff80eff2bb00 x23: 0000000000000000 x22: dead00000000122 x21: 0000000000000000 x20: fffff80efef6bf0 x19: fffff80041c8bf8 x18: ffffffffffffffff x17: fffffc06f146000 x16: fffff80eff27dc0 x15: 000000000000003e x14: 0000000000000000 x13: 00000000000054da x12: 0000000000000000 x11: 0000000000639c0 x10: 000000000000000c x9: 0000000000000009 x8: fffff80eff2cb40 x7: fffff80eff2cb40 x6: fffff8002bee480 x5: fffffc080cb2220 x4: fffffc080cb2150 x3: 00000000000f4240 x2: 000000000000102 x1: fffff80eff2bb00 x0: fffff80041c8bf0 Call trace: __run_timers+0x244/0x268 timer_expire_remote+0x50/0x68 tmgr_handle_remote+0x388/0x39c run_timer_softirq+0x38/0x44 handle_softirqs+0x138/0x298 __do_softirq+0x14/0x20 __do_softirq+0x10/0x1c call_on_irq_stack+0x24/0x4c do_softirq_own_stack+0x1c/0x2c irq_exit_rcu+0x9c/0xcc el1_interrupt+0x48/0xc0 el1h_64_irq_handler+0x18/0x24 el1h_64_irq+0x7c/0x80 default_idle_call+0x34/0x68 do_idle+0x23c/0x294 cpu_startup_entry+0x38/0x3c secondary_start_kernel+0x128/0x160 __secondary_switched+0xb8/0xbc ---[end trace 0000000000000000]---
CVE-2024-56577	In the Linux kernel, the following vulnerability has been resolved: media: mtk-jpeg: Fix null-ptr-deref during unload module The workqueue should be destroyed in mtk_jpeg_core.c since commit 09aea13ecf6f ("media: mtk-jpeg: refactor some variables"), otherwise the below calltrace can be easily triggered. [677.862514] Unable to handle kernel paging request at virtual address dfff800000000023 [677.863633] KASAN: null-ptr-deref in range [0x0000000000000118-0x000000000000011f] ... [677.879654] CPU: 6 PID: 1071 Comm: modprobe Tainted: G O 6.8.12-mtk+gfa1a78e5d24b+ #17 ... [677.882838] pc : destroy_workqueue+0x3c/0x770 [677.883413] lr : mtk_jpegdec_destroy_workqueue+0x70/0x88 [mtk_jpeg_dec_hw] [677.884314] sp : ffff80008ad974f0 [677.884744] x29: ffff80008ad974f0 x28: ffff0000d7115580 x27: ffff0000dd691070 [677.885669] x26: ffff0000dd691408 x25: ffff8000844af3e0 x24: ffff80008ad97690 [677.886592] x23: ffff0000e051d400 x22: ffff0000dd691010 x21: dfff800000000000 [677.887515] x20: 0000000000000000 x19: 0000000000000000 x18: ffff800085397ac0 [677.888438] x17: 0000000000000000 x16: ffff8000801b87c8 x15: 1ffff000115b2e10 [677.889361] x14: 00000000f1f1f1f1 x13: 0000000000000000 x12: ffff7000115b2e4d [677.890285] x11: 1ffff000115b2e4c x10: ffff7000115b2e4c x9: ffff80000aa43e90 [677.891208] x8: 00008fffeea4d1b4 x7: ffff80008ad97267 x6: 0000000000000001 [677.892131] x5: ffff80008ad97260 x4: ffff7000115b2e4d x3: 0000000000000000 [677.893054] x2: 0000000000000023 x1: dfff800000000000 x0: 00000000000000118 [677.893977] Call trace: [677.894297] destroy_workqueue+0x3c/0x770 [677.894826] mtk_jpegdec_destroy_workqueue+0x70/0x88 [mtk_jpeg_dec_hw] [677.895677] devm_action_release+0x50/0x90 [677.896211] release_nodes+0xe8/0x170 [677.896688] devres_release_all+0xf8/0x178 [677.897219] device_unbind_cleanup+0x24/0x170 [677.897785] device_release_driver_internal+0x35c/0x480 [677.898461] device_release_driver+0x20/0x38 ... [677.912665] ---[end trace 0000000000000000]---
CVE-2024-56578	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Set video drvdata before register video device The video drvdata should be set before the video device is registered, otherwise video_drvdata() may return NULL in the open() file ops, and led to oops.
CVE-2024-56579	In the Linux kernel, the following vulnerability has been resolved: media: amphion: Set video drvdata before register video device The video drvdata should be set before the video device is registered, otherwise video_drvdata() may return NULL in the open() file ops, and led to oops.
CVE-2024-56580	In the Linux kernel, the following vulnerability has been resolved: media: qcom: camss: fix error path on configuration of power domains There is a chance to meet runtime issues during configuration of CAMSS power domains, because on the error path dev_pm_domain_detach() is unexpectedly called with NULL or error pointer. One of the simplest ways to reproduce the problem is to probe CAMSS driver before registration of CAMSS power domains, for instance if a platform CAMCC driver is simply not built. Warning backtrace example: Unable to handle kernel NULL pointer dereference at virtual address 000000000000001a2 <snip> pc : dev_pm_domain_detach+0x8/0x48 lr : camss_probe+0x374/0x9c0 <snip> Call trace: dev_pm_domain_detach+0x8/0x48 platform_probe+0x70/0xf0 really_probe+0xc4/0x2a8 __driver_probe_device+0x80/0x140 driver_probe_device+0x48/0x170 __device_attach_driver+0xc0/0x148 bus_for_each_drv+0x88/0xf0 __device_attach+0xb0/0x1c0 device_initial_probe+0x1c/0x30 bus_probe_device+0xb4/0xc0 deferred_probe_work_func+0x90/0xd0 process_one_work+0x164/0x3e0 worker_thread+0x310/0x420 kthread+0x120/0x130 ret_from_fork+0x10/0x20

CVE ID	Description
CVE-2024-56581	In the Linux kernel, the following vulnerability has been resolved: btrfs: ref-verify: fix use-after-free after invalid ref action At btrfs_ref_tree_mod() after we successfully inserted the new ref entry (local variable 'ref') into the respective block entry's rbtree (local variable 'be'), if we find an unexpected action of BTRFS_DROP_DELAYED_REF, we error out and free the ref entry without removing it from the block entry's rbtree. Then in the error path of btrfs_ref_tree_mod() we call btrfs_free_ref_cache(), which iterates over all block entries and then calls free_block_entry() for each one, and there we will trigger a use-after-free when we are called against the block entry to which we added the freed ref entry to its rbtree, since the rbtree still points to the block entry, as we didn't remove it from the rbtree before freeing it in the error path at btrfs_ref_tree_mod(). Fix this by removing the new ref entry from the rbtree before freeing it. Syzbot report this with the following stack traces: BTRFS error (device loop0 state EA): Ref action 2, root 5, ref_root 0, parent 8564736, owner 0, offset 0, num_refs 18446744073709551615 __btrfs_mod_ref+0x7dd/0xac0 fs/btrfs/extent-tree.c:2523 update_ref_for_cow+0x9cd/0x11f0 fs/btrfs/ctree.c:512 btrfs_force_cow_block+0x9f6/0x1da0 fs/btrfs/ctree.c:594 btrfs_cow_block+0x35e/0xa40 fs/btrfs/ctree.c:754 btrfs_search_slot+0xbdd/0x30d0 fs/btrfs/ctree.c:2116 btrfs_insert_empty_items+0x9c/0x1a0 fs/btrfs/ctree.c:4314 btrfs_insert_empty_item fs/btrfs/ctree.h:669 [inline] btrfs_insert_orphan_item+0x1f1/0x320 fs/btrfs/orphan.c:23 btrfs_orphan_add +0x6d/0x1a0 fs/btrfs/inode.c:3482 btrfs_unlink+0x267/0x350 fs/btrfs/inode.c:4293 vfs_unlink +0x365/0x650 fs/namei.c:4469 do_unlinkat+0x4ae/0x830 fs/namei.c:4533 __do_sys_unlinkat fs/namei.c:4576 [inline] __se_sys_unlinkat fs/namei.c:4569 [inline] __x64_sys_unlinkat+0xcc/0xf0 fs/namei.c:4569 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0xf7 BTRFS error (device loop0 state EA): Ref action 1, root 5, ref_root 5, parent 0, owner 260, offset 0, num_refs 1 __btrfs_mod_ref+0x76b/0xac0 fs/btrfs/extent-tree.c:2521 update_ref_for_cow+0x96a/0x11f0 btrfs_force_cow_block+0x9f6/0x1da0 fs/btrfs/ctree.c:594 btrfs_cow_block+0x35e/0xa40 fs/btrfs/ctree.c:754 btrfs_search_slot+0xbdd/0x30d0 fs/btrfs/ctree.c:2116 btrfs_lookup_inode+0xdc/0x480 fs/btrfs/inode-item.c:411 __btrfs_update_delayed_inode+0x1e7/0xb90 fs/btrfs/delayed-inode.c:1030 btrfs_update_delayed_inode fs/btrfs/delayed-inode.c:1114 [inline] __btrfs_commit_inode_delayed_items +0x2318/0x24a0 fs/btrfs/delayed-inode.c:1137 __btrfs_run_delayed_items+0x213/0x490 fs/btrfs/delayed-inode.c:1171 btrfs_commit_transaction+0x8a8/0x3740 fs/btrfs/transaction.c:2313 prepare_to_relocate+0x3c4/0x4c0 fs/btrfs/relocation.c:3586 relocate_block_group+0x16c/0xd40 fs/btrfs/relocation.c:3611 btrfs_relocate_block_group+0x77d/0xd90 fs/btrfs/relocation.c:4081 btrfs_relocate_chunk+0x12c/0x3b0 fs/btrfs/volumes.c:3377 __btrfs_balance+0x1b0f/0x26b0 fs/btrfs/volumes.c:4161 btrfs_balance+0xbdc/0x10c0 fs/btrfs/volumes.c:4538 BTRFS error (device loop0 state EA): Ref action 2, root 5, ref_root 0, parent 8564736, owner 0, offset 0, num_refs 18446744073709551615 __btrfs_mod_ref+0x7dd/0xac0 fs/btrfs/extent-tree.c:2523 update_ref_for_cow +0x9cd/0x11f0 fs/btrfs/ctree.c:512 btrfs_force_cow_block+0x9f6/0x1da0 fs/btrfs/ctree.c:594 btrfs_cow_block+0x35e/0xa40 fs/btrfs/ctree.c:754 btrfs_search_slot+0xbdd/0x30d0 fs/btrfs/ctree.c:2116 btrfs_lookup_inode+0xdc/0x480 fs/btrfs/inode-item.c:411 __btrfs_update_delayed_inode+0x1e7/0xb90 fs/btrfs/delayed-inode.c:1030 btrfs_update_delayed_i ---truncated---
CVE-2024-56583	In the Linux kernel, the following vulnerability has been resolved: sched/deadline: Fix warning in migrate_enable for boosted tasks When running the following command: while true; do stress-ng --cyclic 30 --timeout 30s --minimize --quiet done a warning is eventually triggered: WARNING: CPU: 43 PID: 2848 at kernel/sched/deadline.c:794 setup_new_dl_entity+0x13e/0x180 ... Call Trace: <TASK> ? show_trace_log_lvl+0x1c4/0x2df ? enqueue_dl_entity+0x631/0x6e0 ? setup_new_dl_entity +0x13e/0x180 ? __warn+0x7e/0xd0 ? report_bug+0x11a/0x1a0 ? handle_bug+0x3c/0x70 ? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 enqueue_dl_entity+0x631/0x6e0 enqueue_task_dl+0x7d/0x120 __do_set_cpus_allowed+0xe3/0x280 __set_cpus_allowed_ptr_locked +0x140/0x1d0 __set_cpus_allowed_ptr+0x54/0xa0 migrate_enable+0x7e/0x150 rt_spin_unlock +0x1c/0x90 group_send_sig_info+0xf7/0x1a0 ? kill_pid_info+0x1f/0x1d0 kill_pid_info +0x78/0x1d0 kill_proc_info+0x5b/0x110 __x64_sys_kill+0x93/0xc0 do_syscall_64+0x5c/0xf0 entry_SYSCALL_64_after_hwframe+0x6e/0x76 RIP: 0033:0x7f0dab31f92b This warning occurs because set_cpus_allowed dequeues and enqueues tasks with the ENQUEUE_RESTORE flag set. If the task is boosted, the warning is triggered. A boosted task already had its parameters set by rt_mutex_setprio, and a new call to setup_new_dl_entity is unnecessary, hence the WARN_ON call. Check if we are requeueing a boosted task and avoid calling setup_new_dl_entity if that's the case.
CVE-2024-56584	In the Linux kernel, the following vulnerability has been resolved: io_uring/tctx: work around xa_store() allocation error issue syzbot triggered the following WARN_ON: WARNING: CPU: 0 PID: 16 at io_uring/tctx.c:51 __io_uring_free+0xfa/0x140 io_uring/tctx.c:51 which is the WARN_ON_ONCE(! xa_empty(&tctx->xa)); sanity check in __io_uring_free() when a io_uring_task is going through its final put. The syzbot test case includes injecting memory allocation failures, and it very much looks like xa_store() can fail one of its memory allocations and end up with ->head being non-NULL even though no entries exist in the xarray. Until this issue gets sorted out, work around it by attempting to iterate entries in our xarray, and WARN_ON_ONCE() if one is found.

CVE ID	Description
CVE-2024-56586	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix f2fs_bug_on when uninstalling filesystem call f2fs_evict_inode. creating a large files during checkpoint disable until it runs out of space and then delete it, then remount to enable checkpoint again, and then unmount the filesystem triggers the f2fs_bug_on as below: -----[cut here]----- kernel BUG at fs/f2fs/inode.c:896! CPU: 2 UID: 0 PID: 1286 Comm: umount Not tainted 6.11.0-rc7-dirty #360 Oops: invalid opcode: 0000 [#1] PREEMPT SMP NOPTI RIP: 0010:f2fs_evict_inode+0x58c/0x610 Call Trace: __die_body+0x15/0x60 die+0x33/0x50 do_trap+0x10a/0x120 f2fs_evict_inode+0x58c/0x610 do_error_trap+0x60/0x80 f2fs_evict_inode+0x58c/0x610 exc_invalid_op+0x53/0x60 f2fs_evict_inode+0x58c/0x610 asm_exc_invalid_op+0x16/0x20 f2fs_evict_inode+0x58c/0x610 evict+0x101/0x260 dispose_list+0x30/0x50 evict_inodes+0x140/0x190 generic_shutdown_super+0x2f/0x150 kill_block_super+0x11/0x40 kill_f2fs_super+0x7d/0x140 deactivate_locked_super+0x2a/0x70 cleanup_mnt+0xb3/0x140 task_work_run+0x61/0x90 The root cause is: creating large files during disable checkpoint period results in not enough free segments, so when writing back root inode will failed in f2fs_enable_checkpoint. When umount the file system after enabling checkpoint, the root inode is dirty in f2fs_evict_inode function, which triggers BUG_ON. The steps to reproduce are as follows: dd if=/dev/zero of=f2fs.img bs=1M count=55 mount f2fs.img f2fs_dir -o checkpoint=disable:10% dd if=/dev/zero of=big bs=1M count=50 sync rm big mount -o remount,checkpoint=enable f2fs_dir umount f2fs_dir Let's redirty inode when there is not free segments during checkpoint is disable.
CVE-2024-56587	In the Linux kernel, the following vulnerability has been resolved: leds: class: Protect brightness_show() with led_cdev->led_access mutex There is NULL pointer issue observed if from Process A where hid device being added which results in adding a led_cdev addition and later a another call to access of led_cdev attribute from Process B can result in NULL pointer issue. Use mutex led_cdev->led_access to protect access to led->cdev and its attribute inside brightness_show() and max_brightness_show() and also update the comment for mutex that it should be used to protect the led class device fields. Process A Process B kthread+0x114 worker_thread+0x244 process_scheduled_works+0x248 uhid_device_add_worker+0x24 hid_add_device+0x120 device_add+0x268 bus_probe_device+0x94 device_initial_probe+0x14 __device_attach+0xfc bus_for_each_drv+0x10c __device_attach_driver+0x14c driver_probe_device+0x3c __driver_probe_device+0xa0 really_probe+0x190 hid_device_probe+0x130 ps_probe+0x990 ps_led_register+0x94 devm_led_classdev_register_ext+0x58 led_classdev_register_ext+0x1f8 device_create_with_groups+0x48 device_create_groups_vargs+0xc8 device_add+0x244 kobject_uevent+0x14 kobject_uevent_env[jt]+0x224 mutex_unlock[jt]+0xc4 __mutex_unlock_slowpath+0xd4 wake_up_q+0x70 try_to_wake_up[jt]+0x48c preempt_schedule_common+0x28 __schedule+0x628 __switch_to+0x174 el0t_64_sync+0x1a8/0x1ac el0t_64_sync_handler+0x68/0xbc el0_svc+0x38/0x68 do_el0_svc+0x1c/0x28 el0_svc_common+0x80/0xe0 invoke_syscall+0x58/0x114 __arm64_sys_read+0x1c/0x2c ksys_read+0x78/0xe8 vfs_read+0x1e0/0x2c8 kernfs_fop_read_iter+0x68/0x1b4 seq_read_iter+0x158/0x4ec kernfs_seq_show+0x44/0x54 sysfs_kf_seq_show+0xb4/0x130 dev_attr_show+0x38/0x74 brightness_show+0x20/0x4c dualshock4_led_get_brightness+0xc/0x74 [3313.874295][T4013] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000060 [3313.874301][T4013] Mem abort info: [3313.874303][T4013] ESR = 0x0000000009600006 [3313.874305][T4013] EC = 0x25: DABT (current EL), IL = 32 bits [3313.874307][T4013] SET = 0, FnV = 0 [3313.874309][T4013] EA = 0, S1PTW = 0 [3313.874311][T4013] FSC = 0x06: level 2 translation fault [3313.874313][T4013] Data abort info: [3313.874314][T4013] ISV = 0, ISS = 0x000000006, ISS2 = 0x00000000 [3313.874316][T4013] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [3313.874318][T4013] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [3313.874320][T4013] user pgtbl: 4k pages, 39-bit VAs, pgdp=000000008f2b0a000 .. [3313.874332][T4013] Dumping ftrace buffer: [3313.874334][T4013] (ftrace buffer empty) .. [dd3313.874639][T4013] CPU: 6 PID: 4013 Comm: InputReader [3313.874648][T4013] pc : dualshock4_led_get_brightness+0xc/0x74 [3313.874653][T4013] lr : led_update_brightness+0x38/0x60 [3313.874656][T4013] sp : fffffffc0b910bbd0 .. [3313.874685][T4013] Call trace: [3313.874687][T4013] dualshock4_led_get_brightness+0xc/0x74 [3313.874690][T4013] brightness_show+0x20/0x4c [3313.874692][T4013] dev_attr_show+0x38/0x74 [3313.874696][T4013] sysfs_kf_seq_show+0xb4/0x130 [3313.874700][T4013] kernfs_seq_show+0x44/0x54 [3313.874703][T4013] seq_read_iter+0x158/0x4ec [3313.874705][T4013] kernfs_fop_read_iter+0x68/0x1b4 [3313.874708][T4013] vfs_read+0x1e0/0x2c8 [3313.874711][T4013] ksys_read+0x78/0xe8 [3313.874714][T4013] __arm64_sys_read+0x1c/0x2c [3313.874718][T4013] invoke_syscall+0x58/0x114 [3313.874721][T4013] el0_svc_common+0x80/0xe0 [3313.874724][T4013] do_el0_svc+0x1c/0x28 [3313.874727][T4013] el0_svc+0x38/0x68 [3313.874730][T4013] el0t_64_sync_handler+0x68/0xbc [3313.874732][T4013] el0t_64_sync+0x1a8/0x1ac

CVE ID	Description
CVE-2024-56588	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Create all dump files during debugfs initialization For the current debugfs of hisi_sas, after user triggers dump, the driver allocate memory space to save the register information and create debugfs files to display the saved information. In this process, the debugfs files created after each dump. Therefore, when the dump is triggered while the driver is unbind, the following hang occurs: [67840.853907] Unable to handle kernel NULL pointer dereference at virtual address 00000000000000a0 [67840.862947] Mem abort info: [67840.865855] ESR = 0x0000000096000004 [67840.869713] EC = 0x25: DABT (current EL), IL = 32 bits [67840.875125] SET = 0, FnV = 0 [67840.878291] EA = 0, S1PTW = 0 [67840.881545] FSC = 0x04: level 0 translation fault [67840.886528] Data abort info: [67840.889524] ISV = 0, ISS = 0x000000004, ISS2 = 0x00000000 [67840.895117] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [67840.900284] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [67840.905709] user pgtable: 4k pages, 48-bit VAs, pgdp=0000002803a1f000 [67840.912263] [00000000000000a0] pgd=0000000000000000, p4d=0000000000000000 [67840.919177] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [67840.996435] pstate: 80400009 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [67841.003628] pc : down_write+0x30/0x98 [67841.007546] lr : start_creating.part.0+0x60/0x198 [67841.012495] sp : ffff8000b979ba20 [67841.016046] x29: ffff8000b979ba20 x28: 00000000000000010 x27: 0000000000024b40 [67841.023412] x26: 0000000000000012 x25: ffff20202b355ae8 x24: ffff20202b35a8c8 [67841.030779] x23: ffffa36877928208 x22: ffffa368b4972240 x21: ffff8000b979bb18 [67841.038147] x20: ffff00281dc1e3c0 x19: ffffffffefeffe x18: 0000000000000020 [67841.045515] x17: 0000000000000000 x16: ffffa368b128a530 x15: ffffffffefeffe [67841.052888] x14: ffff8000b979bc18 x13: ffffffffefeffe x12: ffff8000b979bb18 [67841.060263] x11: 0000000000000000 x10: 0000000000000000 x9 : ffffa368b1289b18 [67841.067640] x8 : 0000000000000012 x7 : 0000000000000000 x6 : 000000000000003a9 [67841.075014] x5 : 0000000000000000 x4 : ffff002818c5cb00 x3 : 0000000000000001 [67841.082388] x2 : 0000000000000000 x1 : ffff002818c5cb00 x0 : 00000000000000a0 [67841.089759] Call trace: [67841.092456] down_write+0x30/0x98 [67841.096017] start_creating.part.0+0x60/0x198 [67841.100613] debugfs_create_dir+0x48/0x1f8 [67841.104950] debugfs_create_files_v3_hw+0x88/0x348 [hisi_sas_v3_hw] [67841.111447] debugfs_snapshot_regs_v3_hw+0x708/0x798 [hisi_sas_v3_hw] [67841.118111] debugfs_trigger_dump_v3_hw_write+0x9c/0x120 [hisi_sas_v3_hw] [67841.125115] full_proxy_write+0x68/0xc8 [67841.129175] vfs_write+0xd8/0x3f0 [67841.132708] ksys_write+0x70/0x108 [67841.136317] __arm64_sys_write+0x24/0x38 [67841.140440] invoke_syscall+0x50/0x128 [67841.144385] el0_svc_common.constprop.0+0xc8/0xf0 [67841.149273] do_el0_svc+0x24/0x38 [67841.152773] el0_svc+0x38/0xd8 [67841.156009] el0t_64_sync_handler+0xc0/0xc8 [67841.160361] el0t_64_sync+0x1a4/0x1a8 [67841.164189] Code: b9000882 d2800002 d2800023 f9800011 (c85ffc05) [67841.170443] ---[end trace 0000000000000000]--- To fix this issue, create all directories and files during debugfs initialization. In this way, the driver only needs to allocate memory space to save information each time the user triggers dumping.

CVE ID	Description
CVE-2024-56589	In the Linux kernel, the following vulnerability has been resolved: scsi: hisi_sas: Add cond_resched() for no forced preemption model For no forced preemption model kernel, in the scenario where the expander is connected to 12 high performance SAS SSDs, the following call trace may occur: <pre>[214.409199][C240] watchdog: BUG: soft lockup - CPU#240 stuck for 22s! [irq/149-hisi_sa:3211] [214.568533][C240] pstate: 60400009 (nZCv daif +PAN -UAO -TCO BTYPPE=--) [214.575224] [C240] pc : fput_many+0x8c/0xdc [214.579480][C240] lr : fput+0x1c/0xf0 [214.583302][C240] sp : ffff80002de2b900 [214.587298][C240] x29: ffff80002de2b900 x28: ffff1082aa412000 [214.593291] [C240] x27: ffff3062a0348c08 x26: ffff80003a9f6000 [214.599284][C240] x25: ffff1062bbac5c40 x24: 0000000000001000 [214.605277][C240] x23: 000000000000000a x22: 0000000000000001 [214.611270][C240] x21: 0000000000001000 x20: 0000000000000000 [214.617262][C240] x19: ffff3062a41ae580 x18: 0000000000001000 [214.623255][C240] x17: 0000000000000001 x16: ffffd3a6efe5fc0 [214.629248][C240] x15: ffffffff ffff x14: 000000003fffff [214.635241][C240] x13: 000000000000ffff x12: 000000000000029c [214.641234][C240] x11: 0000000000000006 x10: ffff80003a9f7fd0 [214.647226][C240] x9 : ffffd3a6f0482fc x8 : 0000000000000001 [214.653219] [C240] x7 : 0000000000000002 x6 : 0000000000000080 [214.659212][C240] x5 : ffff55480ee9b000 x4 : fffffde7f94c6554 [214.665205][C240] x3 : 0000000000000002 x2 : 0000000000000000 [214.671198][C240] x1 : 0000000000000021 x0 : ffff3062a41ae5b8 [214.677191][C240] Call trace: [214.680320][C240] fput_many+0x8c/0xdc [214.684230][C240] fput+0x1c/0xf0 [214.687707] [C240] aio_complete_rw+0xd8/0x1fc [214.692225][C240] blkdev_bio_end_io+0x98/0x140 [214.696917][C240] bio_endio+0x160/0x1bc [214.701001][C240] blk_update_request+0x1c8/0x3bc [214.705867][C240] scsi_end_request+0x3c/0x1f0 [214.710471][C240] scsi_io_completion +0x7c/0x1a0 [214.715249][C240] scsi_finish_command+0x104/0x140 [214.720200][C240] scsi_softirq_done+0x90/0x180 [214.724892][C240] blk_mq_complete_request+0x5c/0x70 [214.730016][C240] scsi_mq_done+0x48/0xac [214.734194][C240] sas_scsi_task_done+0x1bc/0x16c [libsas] [214.739758][C240] slot_complete_v3_hw+0x260/0x760 [hisi_sas_v3_hw] [214.746185] [C240] cq_thread_v3_hw+0x190 [hisi_sas_v3_hw] [214.752179][C240] irq_thread_fn +0x34/0xa4 [214.756435][C240] irq_thread+0xc4/0x130 [214.760520][C240] kthread+0x108/0x13c [214.764430][C240] ret_from_fork+0x10/0x18 This is because in the hisi_sas driver, both the hardware interrupt handler and the interrupt thread are executed on the same CPU. In the performance test scenario, function irq_wait_for_interrupt() will always return 0 if lots of interrupts occurs and the CPU will be continuously consumed. As a result, the CPU cannot run the watchdog thread. When the watchdog time exceeds the specified time, call trace occurs. To fix it, add cond_resched() to execute the watchdog thread. </pre>
CVE-2024-56590	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Fix not checking skb length on hci_acldata_packet This fixes not checking if skb really contains an ACL header otherwise the code may attempt to access some uninitialized/invalid memory past the valid skb->data.
CVE-2024-56591	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_conn: Use disable_delayed_work_sync This makes use of disable_delayed_work_sync instead cancel_delayed_work_sync as it not only cancel the ongoing work but also disables new submit which is disarable since the object holding the work is about to be freed.

CVE ID	Description
CVE-2024-56592	In the Linux kernel, the following vulnerability has been resolved: bpf: Call free_htab_elem() after htab_unlock_bucket() For htab of maps, when the map is removed from the htab, it may hold the last reference of the map. bpf_map_fd_put_ptr() will invoke bpf_map_free_id() to free the id of the removed map element. However, bpf_map_fd_put_ptr() is invoked while holding a bucket lock (raw_spin_lock_t), and bpf_map_free_id() attempts to acquire map_idr_lock (spinlock_t), triggering the following lockdep warning: ===== [BUG: Invalid wait context] 6.11.0-rc4+ #49 Not tainted ----- test_maps/4881 is trying to lock: ffffffff84884578 (map_idr_lock){+...}-{3:3}, at: bpf_map_free_id.part.0+0x21/0x70 other info that might help us debug this: context-{5:5} 2 locks held by test_maps/4881: #0: ffffffff846caf60 (rcu_read_lock){....}-{1:3}, at: bpf_fd_htab_map_update_elem+0xf9/0x270 #1: ffff888149ced148 (&htab->lockdep_key#2){....}-{2:2}, at: htab_map_update_elem+0x178/0xa80 stack backtrace: CPU: 0 UID: 0 PID: 4881 Comm: test_maps Not tainted 6.11.0-rc4+ #49 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), ... Call Trace: <TASK> dump_stack_lvl+0x6e/0xb0 dump_stack+0x10/0x20 __lock_acquire+0x73e/0x36c0 lock_acquire+0x182/0x450 __raw_spin_lock_irqsave+0x43/0x70 bpf_map_free_id.part.0+0x21/0x70 bpf_map_put+0xcf/0x110 bpf_map_fd_put_ptr+0x9a/0xb0 free_htab_elem+0x69/0xe0 htab_map_update_elem+0x50f/0xa80 bpf_fd_htab_map_update_elem+0x131/0x270 htab_map_update_elem+0x50f/0xa80 bpf_fd_htab_map_update_elem+0x131/0x270 bpf_map_update_value+0x266/0x380 __sys_bpf+0x21bb/0x36b0 __x64_sys_bpf+0x45/0x60 x64_sys_call+0x1b2a/0x20d0 do_syscall_64+0x5d/0x100 entry_SYSCALL_64_after_hwframe+0x76/0x7e One way to fix the lockdep warning is using raw_spinlock_t for map_idr_lock as well. However, bpf_map_alloc_id() invokes idr_alloc_cyclic() after acquiring map_idr_lock, it will trigger a similar lockdep warning because the slab's lock (s->cpu_slab->lock) is still a spinlock. Instead of changing map_idr_lock's type, fix the issue by invoking htab_put_fd_value() after htab_unlock_bucket(). However, only deferring the invocation of htab_put_fd_value() is not enough, because the old map pointers in htab of maps can not be saved during batched deletion. Therefore, also defer the invocation of free_htab_elem(), so these to-be-freed elements could be linked together similar to lru map. There are four callers for ->map_fd_put_ptr: (1) alloc_htab_elem() (through htab_put_fd_value()) It invokes ->map_fd_put_ptr() under a raw_spinlock_t. The invocation of htab_put_fd_value() can not simply move after htab_unlock_bucket(), because the old element has already been stashed in htab->extra_elems. It may be reused immediately after htab_unlock_bucket() and the invocation of htab_put_fd_value() after htab_unlock_bucket() may release the newly-added element incorrectly. Therefore, saving the map pointer of the old element for htab of maps before unlocking the bucket and releasing the map_ptr after unlock. Beside the map pointer in the old element, should do the same thing for the special fields in the old element as well. (2) free_htab_elem() (through htab_put_fd_value()) Its caller includes __htab_map_lookup_and_delete_elem(), htab_map_delete_elem() and __htab_map_lookup_and_delete_batch(). For htab_map_delete_elem(), simply invoke free_htab_elem() after htab_unlock_bucket(). For __htab_map_lookup_and_delete_batch(), just like lru map, linking the to-be-freed element into node_to_free list and invoking free_htab_elem() for these element after unlock. It is safe to reuse batch_flink as the link for node_to_free, because these elements have been removed from the hash llist. Because htab of maps doesn't support lookup_and_delete operation, __htab_map_lookup_and_delete_elem() doesn't have the problem, so kept it as ---truncated---
CVE-2024-56593	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Fix oops due to NULL pointer dereference in brcmf_sdiod_sglist_rw() This patch fixes a NULL pointer dereference bug in brcmfmac that occurs when a high 'sd_sgentry_align' value applies (e.g. 512) and a lot of queued SKBs are sent from the pkt queue. The problem is the number of entries in the pre-allocated sgtable, it is nents = max(rxglom_size, txglom_size) + max(rxglom_size, txglom_size) >> 4 + 1. Given the default [rt]xglom_size=32 it's actually 35 which is too small. Worst case, the pkt queue can end up with 64 SKBs. This occurs when a new SKB is added for each original SKB if tailroom isn't enough to hold tail_pad. At least one sg entry is needed for each SKB. So, eventually the "skb_queue_walk loop" in brcmf_sdiod_sglist_rw may run out of sg entries. This makes sg_next return NULL and this causes the oops. The patch sets nents to max(rxglom_size, txglom_size) * 2 to be able handle the worst-case. Btw. this requires only 64-35=29 * 16 (or 20 if CONFIG_NEED_SG_DMA_LENGTH) = 464 additional bytes of memory.

CVE ID	Description
CVE-2024-56594	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: set the right AMDGPU sg segment limitation The driver needs to set the correct max_segment_size; otherwise debug_dma_map_sg() will complain about the over-mapping of the AMDGPU sg length as following: WARNING: CPU: 6 PID: 1964 at kernel/dma/debug.c:1178 debug_dma_map_sg+0x2dc/0x370 [364.049444] Modules linked in: veth amdgpu(OE) amdxcp drm_exec gpu_sched drm_buddy drm_ttm_helper ttm(OE) drm_suballoc_helper drm_display_helper drm_kms_helper i2c_algo_bit rpcsec_gss_krb5 auth_rpcgss nfsv4 nfs lockd grace netfs xt_contrack xt_MASQUERADE nf_contrack_netlink xfrm_user xfrm_algo iptable_nat xt_addrtype iptable_filter br_netfilter nvme_fabrics overlay nfnetlink_cttimeout nfnetlink_openvswitch nsh nf_conncount nf_nat nf_contrack nf_defrag_ipv6 nf_defrag_ipv4 libcrc32c bridge stp llc amd_atl intel_rapl_msr intel_rapl_common sunrpc sch_fq_codel snd_hda_codec_realtek snd_hda_codec_generic snd_hda_codec_component snd_hda_codec_hdmi snd_hda_intel snd_intel_dspcfg edac_mce_amd binfmt_misc snd_hda_codec snd_pci_acp6x snd_hda_core snd_acp_config snd_hwdep snd_soc_acpi kvm_amd snd_pcm kvm snd_seq_midi snd_seq_midi_event crct10dif_pclmul ghash_clmulni_intel sha512_ssse3 snd_rawmidi sha256_ssse3 sha1_ssse3 aesni_intel snd_seq_nls iso8859_1 crypto_simd snd_seq_device cryptd snd_timer rapl input_leds snd [364.049532] ipmi_devintf wmi_bmof ccp serio_raw k10temp sp5100_tco soundcore ipmi_msghandler cm32181 industrialio mac_hid msr parport_pc ppdev lp parport drm_efi_pstore ip_tables x_tables pci_stub crc32_pclmul nvme ahci libahci i2c_piix4 r8169 nvme_core i2c_designware_pci realtek i2c_ccgx_ucsi video wmi hid_generic cdc_ether usbnet usbhid hid r8152 mii [364.049576] CPU: 6 PID: 1964 Comm: rocminfo Tainted: G OE 6.10.0-custom #492 [364.049579] Hardware name: AMD Majolica-RN/Majolica-RN, BIOS RMJ1009A 06/13/2021 [364.049582] RIP: 0010:debug_dma_map_sg+0x2dc/0x370 [364.049585] Code: 89 4d b8 e8 36 b1 86 00 8b 4d b8 48 b5 55 b0 44 8b 45 a8 4c 8b 4d a0 48 89 c6 48 c7 c7 00 4b 74 bc 4c 89 4d b8 e8 b4 73 f3 ff <0f> 0b 4c 8b 4d b8 8b 15 c8 2c b8 01 85 d2 0f 85 ee fd ff ff 8b 05 [364.049588] RSP: 0018:ffff9ca600b57ac0 EFLAGS: 00010286 [364.049590] RAX: 0000000000000000 RBX: ffff88b7c132b0c8 RCX: 00000000000000027 [364.049592] RDX: ffff88b7c132b0c8 RSI: 0000000000000001 RDI: ffff88b7c132b0c8 [364.049594] RBP: ffff9ca600b57b20 R08: 0000000000000006 R09: ffff9ca600b57930 [364.049596] R10: ffff9ca600b57928 R11: ffffffffcb46328 R12: 0000000000000000 [364.049597] R13: 0000000000000001 R14: ffff88b7c19c0700 R15: ffff88b7c9059800 [364.049599] FS: 00007fb2d3516e80(0000) GS: ffff88b7c19c0700(0000) knlGS:0000000000000000 [364.049601] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [364.049603] CR2: 000055610bd03598 CR3: 00000001049f6000 CR4: 0000000000350ef0 [364.049605] Call Trace: [364.049607] <TASK> [364.049609] ? show_regs+0x6d/0x80 [364.049614] ? __warn+0x8c/0x140 [364.049618] ? debug_dma_map_sg+0x2dc/0x370 [364.049621] ? report_bug+0x193/0x1a0 [364.049627] ? handle_bug+0x46/0x80 [364.049631] ? exc_invalid_op+0x1d/0x80 [364.049635] ? asm_exc_invalid_op+0x1f/0x30 [364.049642] ? debug_dma_map_sg+0x2dc/0x370 [364.049647] __dma_map_sg_attrs+0x90/0xe0 [364.049651] dma_map_sgtable+0x25/0x40 [364.049654] amdgpu_bo_move+0x59a/0x850 [amdgpu] [364.049935] ? srsr_return_thunk+0x5/0x5f [364.049939] ? amdgpu_ttm_tt_populate+0x5d/0xc0 [amdgpu] [364.050095] ttm_bo_handle_move_mem+0xc3/0x180 [ttm] [364.050103] ttm_bo_validate+0xc1/0x160 [ttm] [364.050108] ? amdgpu_ttm_tt_get_user_pages+0xe5/0x1b0 [amdgpu] [364.050263] amdgpu_amdkfd_gpuvm_alloc_memory_of_gpu+0xa12/0xc90 [amdgpu] [364.050473] kfd_ioctl_alloc_memory_of_gpu+0x16b/0x3b0 [amdgpu] [364.050680] kfd_ioctl+0x3c2/0x530 [amdgpu] [364.050866] ? __pfx_kfd_ioctl_alloc_memory_of_gpu+0x10/0x10 [amdgpu] [364.05105 --- truncated---
CVE-2024-56595	In the Linux kernel, the following vulnerability has been resolved: jfs: add a check to prevent array-index-out-of-bounds in dbAdjTree When the value of lp is 0 at the beginning of the for loop, it will become negative in the next assignment and we should bail out.
CVE-2024-56596	In the Linux kernel, the following vulnerability has been resolved: jfs: fix array-index-out-of-bounds in jfs_readdir The stbl might contain some invalid values. Added a check to return error code in that case.
CVE-2024-56597	In the Linux kernel, the following vulnerability has been resolved: jfs: fix shift-out-of-bounds in dbSplit When dmt_budmin is less than zero, it causes errors in the later stages. Added a check to return an error beforehand in dbAllocCtl itself.
CVE-2024-56598	In the Linux kernel, the following vulnerability has been resolved: jfs: array-index-out-of-bounds fix in dtReadFirst The value of stbl can be sometimes out of bounds due to a bad filesystem. Added a check with appropriate return of error code in that case.

CVE ID	Description
CVE-2024-56599	In the Linux kernel, the following vulnerability has been resolved: wifi: ath10k: avoid NULL pointer error during sdio remove When running 'rmmod ath10k', ath10k_sdio_remove() will free sdio workqueue by destroy_workqueue(). But if CONFIG_INIT_ON_FREE_DEFAULT_ON is set to yes, kernel panic will happen: Call trace: destroy_workqueue+0x1c/0x258 ath10k_sdio_remove+0x84/0x94 sdio_bus_remove+0x50/0x16c device_release_driver_internal+0x188/0x25c device_driver_detach+0x20/0x2c This is because during 'rmmod ath10k', ath10k_sdio_remove() will call ath10k_core_destroy() before destroy_workqueue(). wiphy_dev_release() will finally be called in ath10k_core_destroy(). This function will free struct cfg80211_registered_device *rdev and all its members, including wiphy, dev and the pointer of sdio workqueue. Then the pointer of sdio workqueue will be set to NULL due to CONFIG_INIT_ON_FREE_DEFAULT_ON. After device release, destroy_workqueue() will use NULL pointer then the kernel panic happen. Call trace: ath10k_sdio_remove ->ath10k_core_unregister ->ath10k_core_stop ->ath10k_hif_stop ->ath10k_sdio_irq_disable ->ath10k_hif_power_down ->del_timer_sync(&ar_sdio->sleep_timer) ->ath10k_core_destroy ->ath10k_mac_destroy ->ieee80211_free_hw ->wiphy_free ->wiphy_dev_release ->destroy_workqueue Need to call destroy_workqueue() before ath10k_core_destroy(), free the work queue buffer first and then free pointer of work queue by ath10k_core_destroy(). This order matches the error path order in ath10k_sdio_probe(). No work will be queued on sdio workqueue between it is destroyed and ath10k_core_destroy() is called. Based on the call_stack above, the reason is: Only ath10k_sdio_sleep_timer_handler(), ath10k_sdio_hif_tx_sg() and ath10k_sdio_irq_disable() will queue work on sdio workqueue. Sleep timer will be deleted before ath10k_core_destroy() in ath10k_hif_power_down(). ath10k_sdio_irq_disable() only be called in ath10k_hif_stop(). ath10k_core_unregister() will call ath10k_hif_power_down() to stop hif bus, so ath10k_sdio_hif_tx_sg() won't be called anymore. Tested-on: QCA6174 hw3.2 SDIO WLAN.RMH.4.4.1-00189
CVE-2024-56600	In the Linux kernel, the following vulnerability has been resolved: net: inet6: do not leave a dangling sk pointer in inet6_create() sock_init_data() attaches the allocated sk pointer to the provided sock object. If inet6_create() fails later, the sk object is released, but the sock object retains the dangling sk pointer, which may cause use-after-free later. Clear the sock sk pointer on error.
CVE-2024-56601	In the Linux kernel, the following vulnerability has been resolved: net: inet: do not leave a dangling sk pointer in inet_create() sock_init_data() attaches the allocated sk object to the provided sock object. If inet_create() fails later, the sk object is freed, but the sock object retains the dangling pointer, which may create use-after-free later. Clear the sk pointer in the sock object on error.
CVE-2024-56602	In the Linux kernel, the following vulnerability has been resolved: net: ieee802154: do not leave a dangling sk pointer in ieee802154_create() sock_init_data() attaches the allocated sk object to the provided sock object. If ieee802154_create() fails later, the allocated sk object is freed, but the dangling pointer remains in the provided sock object, which may allow use-after-free. Clear the sk pointer in the sock object on error.
CVE-2024-56603	In the Linux kernel, the following vulnerability has been resolved: net: af_can: do not leave a dangling sk pointer in can_create() On error can_create() frees the allocated sk object, but sock_init_data() has already attached it to the provided sock object. This will leave a dangling sk pointer in the sock object and may cause use-after-free later.
CVE-2024-56604	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: RFCOMM: avoid leaving dangling sk pointer in rfcomm_sock_alloc() bt_sock_alloc() attaches allocated sk object to the provided sock object. If rfcomm_dlc_alloc() fails, we release the sk object, but leave the dangling pointer in the sock object, which may cause use-after-free. Fix this by swapping calls to bt_sock_alloc() and rfcomm_dlc_alloc().
CVE-2024-56605	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: do not leave dangling sk pointer on error in l2cap_sock_create() bt_sock_alloc() allocates the sk object and attaches it to the provided sock object. On error l2cap_sock_alloc() frees the sk object, but the dangling pointer is still attached to the sock object, which may create use-after-free in other code.
CVE-2024-56606	In the Linux kernel, the following vulnerability has been resolved: af_packet: avoid erroring out after sock_init_data() in packet_create() After sock_init_data() the allocated sk object is attached to the provided sock object. On error, packet_create() frees the sk object leaving the dangling pointer in the sock object on return. Some other code may try to use this pointer and cause use-after-free.

CVE ID	Description
CVE-2024-56607	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix atomic calls in ath12k_mac_op_set_bitrate_mask() When I try to manually set bitrates: iw wlan0 set bitrates legacy-2.4 I get sleeping from invalid context error, see below. Fix that by switching to use recently introduced ieee80211_iterate_stations_mtx(). Do note that WCN6855 firmware is still crashing, I'm not sure if that firmware even supports bitrate WMI commands and should we consider disabling ath12k_mac_op_set_bitrate_mask() for WCN6855? But that's for another patch. BUG: sleeping function called from invalid context at drivers/net/wireless/ath/ath12k/wmi.c:420 in_atomic(): 0, irqs_disabled(): 0, non_block: 0, pid: 2236, name: iw preempt_count: 0, expected: 0 RCU nest depth: 1, expected: 0 3 locks held by iw/2236: #0: ffffffffabc6f1d8 (cb_lock){++++}-{3:3}, at: genl_rcv+0x14/0x40 #1: ffff888138410810 (&rdev->wiphy.mtx){+.+.}-{3:3}, at: nl80211_pre_doit+0x54d/0x800 [cfg80211] #2: ffffffffab2cf8a0 (rcu_read_lock){....}-{1:2}, at: ieee80211_iterate_stations_atomic+0x2f/0x200 [mac80211] CPU: 3 UID: 0 PID: 2236 Comm: iw Not tainted 6.11.0-rc7-wt-ath+ #1772 Hardware name: Intel(R) Client Systems NUC8i7HVK/NUC8i7HVB, BIOS HNKBLi70.86A.0067.2021.0528.1339 05/28/2021 Call Trace: <TASK> dump_stack_lvl+0xa4/0xe0 dump_stack+0x10/0x20 __might_resched+0x363/0x5a0 ? __alloc_skb+0x165/0x340 __might_sleep+0xad/0x160 ath12k_wmi_cmd_send+0xb1/0x3d0 [ath12k] ? ath12k_wmi_init_wcn7850+0xa40/0xa40 [ath12k] ? __netdev_alloc_skb+0x45/0x7b0 ? __asan_memset+0x39/0x40 ? ath12k_wmi_alloc_skb+0xf0/0x150 [ath12k] ? reacquire_held_locks+0x4d0/0x4d0 ath12k_wmi_set_peer_param+0x340/0x5b0 [ath12k] ath12k_mac_disable_peer_fixed_rate+0xa3/0x110 [ath12k] ? ath12k_mac_vdev_stop+0x4f0/0x4f0 [ath12k] ieee80211_iterate_stations_atomic+0xd4/0x200 [mac80211] ath12k_mac_op_set_bitrate_mask+0x5d2/0x1080 [ath12k] ? ath12k_mac_vif_chan+0x320/0x320 [ath12k] drv_set_bitrate_mask+0x267/0x470 [mac80211] ieee80211_set_bitrate_mask+0x4cc/0x8a0 [mac80211] ? __this_cpu_preempt_check+0x13/0x20 nl80211_set_tx_bitrate_mask+0x2bc/0x530 [cfg80211] ? nl80211_parse_tx_bitrate_mask+0x2320/0x2320 [cfg80211] ? trace_contention_end+0xef/0x140 ? rtnl_unlock+0x9/0x10 ? nl80211_pre_doit+0x557/0x800 [cfg80211] genl_family_rcv_msg_doit+0x1f0/0x2e0 ? genl_family_rcv_msg_attrs_parse.isra.0+0x250/0x250 ? ns_capable+0x57/0xd0 genl_family_rcv_msg+0x34c/0x600 ? genl_family_rcv_msg_dumpit+0x310/0x310 ? __lock_acquire+0xc62/0x1de0 ? he_set_mcs_mask.isra.0+0x8d0/0x8d0 [cfg80211] ? nl80211_parse_tx_bitrate_mask+0x2320/0x2320 [cfg80211] ? cfg80211_external_auth_request+0x690/0x690 [cfg80211] genl_rcv_msg+0xa0/0x130 netlink_rcv_skb+0x14c/0x400 ? genl_family_rcv_msg+0x600/0x600 ? netlink_ack+0xd70/0xd70 ? rwsem_optimistic_spin+0x4f0/0x4f0 ? genl_rcv+0x14/0x40 ? down_read_killable+0x580/0x580 ? netlink_deliver_tap+0x13e/0x350 ? __this_cpu_preempt_check+0x13/0x20 genl_rcv+0x23/0x40 netlink_unicast+0x45e/0x790 ? netlink_attachskb+0x7f0/0x7f0 netlink_sendmsg+0x7eb/0xadb ? netlink_unicast+0x790/0x790 ? __this_cpu_preempt_check+0x13/0x20 ? selinux_socket_sendmsg+0x31/0x40 ? netlink_unicast+0x790/0x790 __sock_sendmsg+0xc9/0x160 __sys_sendmsg+0x620/0x990 ? kernel_sendmsg+0x30/0x30 ? __copy_msghdr+0x410/0x410 ? __kasan_check_read+0x11/0x20 ? mark_lock+0xe6/0x1470 __sys_sendmsg+0xe9/0x170 ? copy_msghdr_from_user+0x120/0x120 ? __lock_acquire+0xc62/0x1de0 ? do_fault_around+0x2c6/0x4e0 ? do_user_addr_fault+0x8c1/0xde0 ? reacquire_held_locks+0x220/0x4d0 ? do_user_addr_fault+0x8c1/0xde0 ? __kasan_check_read+0x11/0x20 ? __fdget+0x4e/0x1d0 ? sockfd_lookup_light+0x1a/0x170 __sys_sendmsg+0xd2/0x180 ? __sys_sendmsg_sock+0x20/0x20 ? reacquire_held_locks+0x4d0/0x4d0 ? debug_smp_processor_id+0x17/0x20 __x64_sys_sendmsg+0x72/0xb0 ? lockdep_hardirqs_on+0x7d/0x100 x64_sys_call+0x894/0x9f0 do_syscall_64+0x64/0x130 entry_SYSCALL_64_after_---truncated---

CVE ID	Description
CVE-2024-56608	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix out-of-bounds access in 'dcn21_link_encoder_create' An issue was identified in the dcn21_link_encoder_create function where an out-of-bounds access could occur when the hpd_source index was used to reference the link_enc_hpd_regs array. This array has a fixed size and the index was not being checked against the array's bounds before accessing it. This fix adds a conditional check to ensure that the hpd_source index is within the valid range of the link_enc_hpd_regs array. If the index is out of bounds, the function now returns NULL to prevent undefined behavior. References: [65.920507] -----[cut here]----- [65.920510] UBSAN: array-index-out-of-bounds in drivers/gpu/drm/amd/amdgpu/./display/dc/resource/dcn21/dcn21_resource.c:1312:29 [65.920519] index 7 is out of range for type 'dcn10_link_enc_hpd_registers [5]' [65.920523] CPU: 3 PID: 1178 Comm: modprobe Tainted: G OE 6.8.0-cleanershafterfeatureresetandtipmi200nv2132 #13 [65.920525] Hardware name: AMD Majolica-RN/Majolica-RN, BIOS WMJ0429N_Weekly_20_04_204/29/2020 [65.920527] Call Trace: [65.920529] <TASK> [65.920532] dump_stack_lvl+0x48/0x70 [65.920541] dump_stack+0x10/0x20 [65.920543] __ubsan_handle_out_of_bounds+0xa2/0xe0 [65.920549] dcn21_link_encoder_create+0xd9/0x140 [amdgpu] [65.921009] link_create+0x6d3/0xed0 [amdgpu] [65.921355] create_links+0x18a/0x4e0 [amdgpu] [65.921679] dc_create+0x360/0x720 [amdgpu] [65.921999] ? dmi_matches+0xa0/0x220 [65.922004] amdgpu_dm_init+0x2b6/0x2c90 [amdgpu] [65.922342] ? console_unlock+0x77/0x120 [65.922348] ? dev_printk_emit+0x86/0xb0 [65.922354] dm_hw_init+0x15/0x40 [amdgpu] [65.922686] amdgpu_device_init+0x26a8/0x33a0 [amdgpu] [65.922921] amdgpu_driver_load_kms+0x1b/0xa0 [amdgpu] [65.923087] amdgpu_pci_probe+0x1b7/0x630 [amdgpu] [65.923087] local_pci_probe+0x4b/0xb0 [65.923087] pci_device_probe+0xc8/0x280 [65.923087] really_probe+0x187/0x300 [65.923087] __driver_probe_device+0x85/0x130 [65.923087] driver_probe_device+0x24/0x110 [65.923087] __driver_attach+0xac/0x1d0 [65.923087] ? __pfx__driver_attach+0x10/0x10 [65.923087] bus_for_each_dev+0x7d/0xd0 [65.923087] driver_attach+0x1e/0x30 [65.923087] bus_add_driver+0xf2/0x200 [65.923087] driver_register+0x64/0x130 [65.923087] ? __pfx_amdgpu_init+0x10/0x10 [amdgpu] [65.923087] __pci_register_driver+0x61/0x70 [65.923087] amdgpu_init+0x7d/0xff0 [amdgpu] [65.923087] do_one_initcall+0x49/0x310 [65.923087] ? kmalloc_trace+0x136/0x360 [65.923087] do_init_module+0x6a/0x270 [65.923087] load_module+0x1fce/0x23a0 [65.923087] init_module_from_file+0x9c/0xe0 [65.923087] ? init_module_from_file+0x9c/0xe0 [65.923087] idempotent_init_module+0x179/0x230 [65.923087] __x64_sys_finit_module+0x5d/0xa0 [65.923087] do_syscall_64+0x76/0x120 [65.923087] entry_SYSCALL_64_after_hwframe+0x6e/0x76 [65.923087] RIP: 0033:0x7f2d80f1e88d [65.923087] Code: 5b 41 5c c3 66 0f 1f 84 00 00 00 00 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 73 b5 0f 00 f7 d8 64 89 01 48 [65.923087] RSP: 002b:00007ffc7bc1aa78 EFLAGS: 00000246 ORIG_RAX: 0000000000000139 [65.923087] RAX: ffffffffda RBX: 0000564c9c1db130 RCX: 00007f2d80f1e88d [65.923087] RDX: 0000000000000000 RSI: 0000564c9c1e5480 RDI: 000000000000000f [65.923087] RBP: 0000000000040000 R08: 0000000000000000 R09: 0000000000000002 [65.923087] R10: 000000000000000f R11: 0000000000000246 R12: 0000564c9c1e5480 [65.923087] R13: 0000564c9c1db260 R14: 0000000000000000 R15: 0000564c9c1e54b0 [65.923087] </TASK> [65.923927] ---[end trace]---

CVE ID	Description
CVE-2024-56609	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: use ieee80211_purge_tx_queue() to purge TX skb When removing kernel modules by: rmmmod rtw88_8723cs rtw88_8703b rtw88_8723x rtw88_sdio rtw88_core Driver uses skb_queue_purge() to purge TX skb, but not report tx status causing "Have pending ack frames!" warning. Use ieee80211_purge_tx_queue() to correct this. Since ieee80211_purge_tx_queue() doesn't take locks, to prevent racing between TX work and purge TX queue, flush and destroy TX work in advance. wlan0: deauthenticating from aa:f5:fd:60:4c:a8 by local choice (Reason: 3=DEAUTH_LEAVING) ----- [cut here]----- Have pending ack frames! WARNING: CPU: 3 PID: 9232 at net/mac80211/main.c:1691 ieee80211_free_ack_frame+0x5c/0x90 [mac80211] CPU: 3 PID: 9232 Comm: rmmmod Tainted: G C 6.10.1-200.fc40.aarch64 #1 Hardware name: pine64 Pine64 PinePhone Braveheart (1.1)/Pine64 PinePhone Braveheart (1.1), BIOS 2024.01 01/01/2024 pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTTYPE=--) pc : ieee80211_free_ack_frame+0x5c/0x90 [mac80211] lr : ieee80211_free_ack_frame+0x5c/0x90 [mac80211] sp : ffff80008c1b37b0 x29: ffff80008c1b37b0 x28: ffff000003be8000 x27: 0000000000000000 x26: 0000000000000000 x25: ffff000003dc14b8 x24: ffff80008c1b37d0 x23: ffff00000ff9f80 x22: 0000000000000000 x21: 000000007fffffff x20: ffff800007c7e93d8 x19: ffff00006e66f400 x18: 0000000000000000 x17: ffff7fffd2b3000 x16: ffff800083fc0000 x15: 0000000000000000 x14: 0000000000000000 x13: 2173656d61726620 x12: 6b636120676e6964 x11: 0000000000000000 x10: 0000000000000005d x9: ffff8000802af2b0 x8: ffff80008c1b3430 x7: 0000000000000001 x6: 0000000000000001 x5: 0000000000000000 x4: 0000000000000000 x3: 0000000000000000 x2: 0000000000000000 x1: 0000000000000000 x0: ffff000003be8000 Call trace: ieee80211_free_ack_frame+0x5c/0x90 [mac80211] idr_for_each+0x74/0x110 ieee80211_free_hw+0x44/0xe8 [mac80211] rtw_sdio_remove+0x9c/0xc0 [rtw88_sdio] sdio_bus_remove+0x44/0x180 device_remove+0x54/0x90 device_release_driver_internal+0x1d4/0x238 driver_detach+0x54/0xc0 bus_remove_driver+0x78/0x108 driver_unregister+0x38/0x78 sdio_unregister_driver+0x2c/0x40 rtw_8723cs_driver_exit+0x18/0x1000 [rtw88_8723cs] __do_sys_delete_module.isra.0+0x190/0x338 __arm64_sys_delete_module+0x1c/0x30 invoke_syscall+0x74/0x100 el0_svc_common.constprop.0+0x48/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x3c/0x158 el0t_64_sync_handler+0x120/0x138 el0t_64_sync+0x194/0x198 ---[end trace 0000000000000000]---
CVE-2024-56610	In the Linux kernel, the following vulnerability has been resolved: kcsan: Turn report_filterlist_lock into a raw_spinlock Ran Xiaokai reports that with a KCSAN-enabled PREEMPT_RT kernel, we can see splats like: BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 0, name: swapper/1 preempt_count: 10002, expected: 0 RCU nest depth: 0, expected: 0 no locks held by swapper/1/0. irq event stamp: 156674 hardirqs last enabled at (156673): [<fffffff81130bd9>] do_idle+0x1f9/0x240 hardirqs last disabled at (156674): [<fffffff82254f84>] sysvec_apic_timer_interrupt+0x14/0xc0 softirqs last enabled at (0): [<fffffff81099f47>] copy_process+0xfc7/0x4b60 softirqs last disabled at (0): [<0000000000000000>] 0x0 Preemption disabled at: [<fffffff814a3e2a>] paint_ptr+0x2a/0x90 CPU: 1 UID: 0 PID: 0 Comm: swapper/1 Not tainted 6.11.0+ #3 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.12.0-0-ga698c8995f-prebuilt.qemu.org 04/01/2014 Call Trace: <IRQ> dump_stack_lvl+0x7e/0xc0 dump_stack+0x1d/0x30 __might_resched+0x1a2/0x270 rt_spin_lock+0x68/0x170 kcsan_skip_report_debugfs+0x43/0xe0 print_report+0xb5/0x590 kcsan_report_known_origin+0x1b1/0x1d0 kcsan_setup_watchpoint+0x348/0x650 __tsan_unaligned_write1+0x16d/0x1d0 hrtimer_interrupt+0x3d6/0x430 __sysvec_apic_timer_interrupt+0xe8/0x3a0 sysvec_apic_timer_interrupt+0x97/0xc0 </IRQ> On a detected data race, KCSAN's reporting logic checks if it should filter the report. That list is protected by the report_filterlist_lock *non-raw* spinlock which may sleep on RT kernels. Since KCSAN may report data races in any context, convert it to a raw_spinlock. This requires being careful about when to allocate memory for the filter list itself which can be done via KCSAN's debugfs interface. Concurrent modification of the filter list via debugfs should be rare: the chosen strategy is to optimistically pre-allocate memory before the critical section and discard if unused.

CVE ID	Description
CVE-2024-56611	In the Linux kernel, the following vulnerability has been resolved: mm/mempolicy: fix migrate_to_node() assuming there is at least one VMA in a MM We currently assume that there is at least one VMA in a MM, which isn't true. So we might end up having find_vma() return NULL, to then de-reference NULL. So properly handle find_vma() returning NULL. This fixes the report: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] CPU: 1 UID: 0 PID: 6021 Comm: syz-executor284 Not tainted 6.12.0-rc7-syzkaller-00187-gf868cd251776 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/30/2024 RIP: 0010:migrate_to_node mm/mempolicy.c:1090 [inline] RIP: 0010:do_migrate_pages+0x403/0x6f0 mm/mempolicy.c:1194 Code: ... RSP: 0018:ffffc9000375fd08 EFLAGS: 00010246 RAX: 0000000000000000 RBX: fffffc9000375fd78 RCX: 0000000000000000 RDX: ffff88807e171300 RSI: dffffc0000000000 RDI: ffff88803390c044 RBP: ffff88807e171428 R08: 0000000000000014 R09: fffffbfff2039ef1 R10: ffffffff901cf78f R11: 0000000000000000 R12: 0000000000000003 R13: fffffc9000375fe90 R14: fffffc9000375fe98 R15: fffffc9000375fd08 FS: 00005555919e1380(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005555919e1ca8 CR3: 000000007f12a000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> kernel_migrate_pages+0x5b2/0x750 mm/mempolicy.c:1709 __do_sys_migrate_pages mm/mempolicy.c:1727 [inline] __se_sys_migrate_pages mm/mempolicy.c:1723 [inline] __x64_sys_migrate_pages+0x96/0x100 mm/mempolicy.c:1723 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f [akpm@linux-foundation.org: add unlikely()]
CVE-2024-56613	In the Linux kernel, the following vulnerability has been resolved: sched/numa: fix memory leak due to the overwritten vma->numab_state [Problem Description] When running the hackbench program of LTP, the following memory leak is reported by kmemleak. # /opt/ltp/testcases/bin/hackbench 20 thread 1000 Running with 20*40 (== 800) tasks. # dmesg grep kmemleak ... kmemleak: 480 new suspected memory leaks (see /sys/kernel/debug/kmemleak) kmemleak: 665 new suspected memory leaks (see /sys/kernel/debug/kmemleak) # cat /sys/kernel/debug/kmemleak unreferenced object 0xfffff888cd8ca2c40 (size 64): comm "hackbench", pid 17142, jiffies 4299780315 hex dump (first 32 bytes): ac 74 49 00 01 00 00 00 4c 84 49 00 01 00 00 00 .tL.....L.I..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace (crc bff18fd4): [<ffffffffff81419a89>] __kmalloccache_noprof+0x2f9/0x3f0 [<ffffffffff8113f715>] task_numa_work+0x725/0xa00 [<ffffffffff8110f878>] task_work_run+0x58/0x90 [<ffffffffff81ddd9f8>] syscall_exit_to_user_mode+0x1c8/0x1e0 [<ffffffffff81dd78d5>] do_syscall_64+0x85/0x150 [<ffffffffff81e0012b>] entry_SYSCALL_64_after_hwframe+0x76/0x7e ... This issue can be consistently reproduced on three different servers: * a 448-core server * a 256-core server * a 192-core server [Root Cause] Since multiple threads are created by the hackbench program (along with the command argument 'thread'), a shared vma might be accessed by two or more cores simultaneously. When two or more cores observe that vma->numab_state is NULL at the same time, vma->numab_state will be overwritten. Although current code ensures that only one thread scans the VMAs in a single 'numa_scan_period', there might be a chance for another thread to enter in the next 'numa_scan_period' while we have not gotten till numab_state allocation [1]. Note that the command '/opt/ltp/testcases/bin/hackbench 50 process 1000' cannot reproduce the issue. It is verified with 200+ test runs. [Solution] Use the cmpxchg atomic operation to ensure that only one thread executes the vma->numab_state assignment. [1] https://lore.kernel.org/lkml/1794be3c-358c-4cdc-a43d-a1f841d91ef7@amd.com/

CVE ID	Description
CVE-2024-56614	In the Linux kernel, the following vulnerability has been resolved: xsk: fix OOB map writes when deleting elements Jordy says: " In the xsk_map_delete_elem function an unsigned integer (map->max_entries) is compared with a user-controlled signed integer (k). Due to implicit type conversion, a large unsigned value for map->max_entries can bypass the intended bounds check: if (k >= map->max_entries) return -EINVAL; This allows k to hold a negative value (between -2147483648 and -2), which is then used as an array index in m->xsk_map[k], which results in an out-of-bounds access. spin_lock_bh(&m->lock); map_entry = &m->xsk_map[k]; // Out-of-bounds map_entry old_xs = unrcu_pointer(xchg(map_entry, NULL)); // Oob write if (old_xs) xsk_map_sock_delete(old_xs, map_entry); spin_unlock_bh(&m->lock); The xchg operation can then be used to cause an out-of-bounds write. Moreover, the invalid map_entry passed to xsk_map_sock_delete can lead to further memory corruption. " It indeed results in following splat: [76612.897343] BUG: unable to handle page fault for address: ffff8fc2e461108 [76612.904330] #PF: supervisor write access in kernel mode [76612.909639] #PF: error_code(0x0002) - not-present page [76612.914855] PGD 0 P4D 0 [76612.917431] Oops: Oops: 0002 [#1] PREEMPT SMP [76612.921859] CPU: 11 UID: 0 PID: 10318 Comm: a.out Not tainted 6.12.0-rc1+ #470 [76612.929189] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0008.031920191559 03/19/2019 [76612.939781] RIP: 0010:xsk_map_delete_elem+0x2d/0x60 [76612.944738] Code: 00 00 41 54 55 53 48 63 2e 3b 6f 24 73 38 4c 8d a7 f8 00 00 00 48 89 fb 4c 89 e7 e8 2d bf 05 00 48 8d b4 eb 00 01 00 00 31 ff <48> 87 3e 48 85 ff 74 05 e8 16 ff ff 4c 89 e7 e8 3e bc 05 00 31 [76612.963774] RSP: 0018:ffff9002e407df8 EFLAGS: 00010246 [76612.969079] RAX: 0000000000000000 RBX: ffff9002e461000 RCX: 0000000000000000 [76612.976323] RDX: 0000000000000000 RSI: ffff8fc2e461108 RDI: 0000000000000000 [76612.983569] RBP: ffffffff80000000 R08: 0000000000000000 R09: 0000000000000000 [76612.990812] R10: ffff9002e407e18 R11: ffff888108a38858 R12: ffff9002e4610f8 [76612.998060] R13: ffff888108a38858 R14: 00007ffd1ae0ac78 R15: ffff9002e4610c0 [76613.005303] FS: 00007f80b6f59740(0000) GS:ffff8897e0ec0000(0000) knlGS:0000000000000000 [76613.013517] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [76613.019349] CR2: ffff8fc2e461108 CR3: 000000011e3ef001 CR4: 00000000007726f0 [76613.026595] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [76613.033841] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 [76613.041086] PKRU: 55555554 [76613.043842] Call Trace: [76613.046331] <TASK> [76613.048468] ? __die+0x20/0x60 [76613.051581] ? page_fault_oops+0x15a/0x450 [76613.055747] ? search_extable+0x22/0x30 [76613.059649] ? search_bpf_extables+0x5f/0x80 [76613.063988] ? exc_page_fault+0xa9/0x140 [76613.067975] ? asm_exc_page_fault+0x22/0x30 [76613.072229] ? xsk_map_delete_elem+0x2d/0x60 [76613.076573] ? xsk_map_delete_elem+0x23/0x60 [76613.080914] __sys_bpf+0x19b7/0x23c0 [76613.084555] __x64_sys_bpf+0x1a/0x20 [76613.088194] do_syscall_64+0x37/0xb0 [76613.091832] entry_SYSCALL_64_after_hwframe+0x4b/0x53 [76613.096962] RIP: 0033:0x7f80b6d1e88d [76613.100592] Code: 5b 41 5c c3 66 0f 1f 84 00 00 00 00 00 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 73 b5 0f 00 f7 d8 64 89 01 48 [76613.119631] RSP: 002b:00007ffd1ae0ac68 EFLAGS: 00000206 ORIG_RAX: 0000000000000141 [76613.131330] RAX: ffffffff80000000 RBX: 0000000000000000 RCX: 00007f80b6d1e88d [76613.142632] RDX: 0000000000000098 RSI: 00007ffd1ae0ad20 RDI: 0000000000000003 [76613.153967] RBP: 00007ffd1ae0adc0 R08: 0000000000000000 R09: 0000000000000000 [76613.166030] R10: 00007f80b6f77040 R11: 0000000000000206 R12: 00007ffd1ae0aed8 [76613.177130] R13: 000055ddf42ce1e9 R14: 000055ddf42d0d98 R15: 00 --- truncated---

CVE ID	Description
CVE-2024-56615	In the Linux kernel, the following vulnerability has been resolved: bpf: fix OOB devmap writes when deleting elements Jordy reported issue against XSKMAP which also applies to DEVMAP - the index used for accessing map entry, due to being a signed integer, causes the OOB writes. Fix is simple as changing the type from int to u32, however, when compared to XSKMAP case, one more thing needs to be addressed. When map is released from system via dev_map_free(), we iterate through all of the entries and an iterator variable is also an int, which implies OOB accesses. Again, change it to be u32. Example splat below: [160.724676] BUG: unable to handle page fault for address: fffff80000000000 [160.731662] #PF: supervisor read access in kernel mode [160.736876] #PF: error_code(0x0000) - not-present page [160.742095] PGD 0 P4D 0 [160.744678] Oops: Oops: 0000 [#1] PREEMPT SMP [160.749106] CPU: 1 UID: 0 PID: 520 Comm: kworker/u145:12 Not tainted 6.12.0-rc1+ #487 [160.757050] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0008.031920191559 03/19/2019 [160.767642] Workqueue: events_unbound bpf_map_free_deferred [160.773308] RIP: 0010:dev_map_free+0x77/0x170 [160.777735] Code: 00 e8 fd 91 ed ff e8 b8 73 ed ff 41 83 7d 18 19 74 6e 41 8b 45 24 49 8b bd f8 00 00 00 31 db 85 c0 74 48 48 63 c3 48 8d 04 c7 <48> 8b 28 48 85 ed 74 30 48 8b 7d 18 48 85 ff 74 05 e8 b3 52 fa ff [160.796777] RSP: 0018:ffff90000ee1fe38 EFLAGS: 00010202 [160.802086] RAX: fffff80000000000 RBX: 0000000080000000 RCX: 0000000000000024 [160.809331] RDX: 0000000000000000 RSI: 0000000000000024 RDI: fffff80000000000 [160.816576] RBP: 0000000000000000 R08: 0000000000000023 R09: 0000000000000001 [160.823823] R10: 0000000000000001 R11: 000000000000ee6b2 R12: dead000000000122 [160.831066] R13: ffff88810c928e00 R14: ffff8881002df405 R15: 0000000000000000 [160.838310] FS: 0000000000000000(0000) GS:ffff8897e0c40000(0000) knlGS:0000000000000000 [160.846528] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [160.852357] CR2: fffff80000000000 CR3: 0000000005c32006 CR4: 00000000007726f0 [160.859604] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [160.866847] DR3: 0000000000000000 DR6: 0000000000000000 DR7: 0000000000000040 [160.874092] PKRU: 55555554 [160.876847] Call Trace: [160.879338] <TASK> [160.881477] ? __die+0x20/0x60 [160.884586] ? page_fault_oops+0x15a/0x450 [160.888746] ? search_extable+0x22/0x30 [160.892647] ? search_bpf_extables+0x5f/0x80 [160.896988] ? exc_page_fault+0xa9/0x140 [160.900973] ? asm_exc_page_fault+0x22/0x30 [160.905232] ? dev_map_free+0x77/0x170 [160.909043] ? dev_map_free+0x58/0x170 [160.912857] bpf_map_free_deferred+0x51/0x90 [160.917196] process_one_work+0x142/0x370 [160.921272] worker_thread+0x29e/0x3b0 [160.925082] ? rescuer_thread+0x4b0/0x4b0 [160.929157] kthread+0xd4/0x110 [160.932355] ? kthread_park+0x80/0x80 [160.936079] ret_from_fork+0x2d/0x50 [160.943396] ? kthread_park+0x80/0x80 [160.950803] ret_from_fork_asm+0x11/0x20 [160.958482] </TASK>
CVE-2024-56616	In the Linux kernel, the following vulnerability has been resolved: drm/dp_mst: Fix MST sideband message body length check Fix the MST sideband message body length check, which must be at least 1 byte accounting for the message body CRC (aka message data CRC) at the end of the message. This fixes a case where an MST branch device returns a header with a correct header CRC (indicating a correctly received body length), with the body length being incorrectly set to 0. This will later lead to a memory corruption in drm_dp_sideband_append_payload() and the following errors in dmesg: UBSAN: array-index-out-of-bounds in drivers/gpu/drm/display/drm_dp_mst_topology.c:786:25 index -1 is out of range for type 'u8 [48]' Call Trace: drm_dp_sideband_append_payload+0x33d/0x350 [drm_display_helper] drm_dp_get_one_sb_msg+0x3ce/0x5f0 [drm_display_helper] drm_dp_mst_hpd_irq_handle_event+0xc8/0x1580 [drm_display_helper] memcpy: detected field-spanning write (size 18446744073709551615) of single field "&msg->msg[msg->curlen]" at drivers/gpu/drm/display/drm_dp_mst_topology.c:791 (size 256) Call Trace: drm_dp_sideband_append_payload+0x324/0x350 [drm_display_helper] drm_dp_get_one_sb_msg+0x3ce/0x5f0 [drm_display_helper] drm_dp_mst_hpd_irq_handle_event+0xc8/0x1580 [drm_display_helper]

CVE ID	Description
CVE-2024-56617	In the Linux kernel, the following vulnerability has been resolved: cacheinfo: Allocate memory during CPU hotplug if not done from the primary CPU Commit 5944ce092b97 ("arch_topology: Build cacheinfo from primary CPU") adds functionality that architectures can use to optionally allocate and build cacheinfo early during boot. Commit 6539cfa9495 ("cacheinfo: Add arch specific early level initializer") lets secondary CPUs correct (and reallocate memory) cacheinfo data if needed. If the early build functionality is not used and cacheinfo does not need correction, memory for cacheinfo is never allocated. x86 does not use the early build functionality. Consequently, during the cacheinfo CPU hotplug callback, last_level_cache_is_valid() attempts to dereference a NULL pointer: BUG: kernel NULL pointer dereference, address: 0000000000000100 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not present page PGD 0 P4D 0 Oops: 0000 [#1] PREPMT SMP NOPTI CPU: 0 PID 19 Comm: cpuhp/0 Not tainted 6.4.0-rc2 #1 RIP: 0010: last_level_cache_is_valid+0x95/0xe0a Allocate memory for cacheinfo during the cacheinfo CPU hotplug callback if not done earlier. Moreover, before determining the validity of the last-level cache info, ensure that it has been allocated. Simply checking for non-zero cache_leaves() is not sufficient, as some architectures (e.g., Intel processors) have non-zero cache_leaves() before allocation. Dereferencing NULL cacheinfo can occur in update_per_cpu_data_slice_size(). This function iterates over all online CPUs. However, a CPU may have come online recently, but its cacheinfo may not have been allocated yet. While here, remove an unnecessary indentation in allocate_cache_info(). [bp: Massage.]
CVE-2024-56619	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix potential out-of-bounds memory access in nilfs_find_entry() Syzbot reported that when searching for records in a directory where the inode's i_size is corrupted and has a large value, memory access outside the folio/page range may occur, or a use-after-free bug may be detected if KASAN is enabled. This is because nilfs_last_byte(), which is called by nilfs_find_entry() and others to calculate the number of valid bytes of directory data in a page from i_size and the page index, loses the upper 32 bits of the 64-bit size information due to an inappropriate type of local variable to which the i_size value is assigned. This caused a large byte offset value due to underflow in the end address calculation in the calling nilfs_find_entry(), resulting in memory access that exceeds the folio/page size. Fix this issue by changing the type of the local variable causing the bit loss from "unsigned int" to "u64". The return value of nilfs_last_byte() is also of type "unsigned int", but it is truncated so as not to exceed PAGE_SIZE and no bit loss occurs, so no change is required.
CVE-2024-56620	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: qcom: Only free platform MSIs when ESI is enabled Otherwise, it will result in a NULL pointer dereference as below: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 Call trace: mutex_lock+0xc/0x54 platform_device_msi_free_irqs_all+0x14/0x20 ufs_qcom_remove+0x34/0x48 [ufs_qcom] platform_remove+0x28/0x44 device_remove+0x4c/0x80 device_release_driver_internal+0xd8/0x178 driver_detach+0x50/0x9c bus_remove_driver+0x6c/0xbc driver_unregister+0x30/0x60 platform_driver_unregister+0x14/0x20 ufs_qcom_pltform_exit+0x18/0xb94 [ufs_qcom] __arm64_sys_delete_module+0x180/0x260 invoke_syscall+0x44/0x100 el0_svc_common.constprop.0+0xc0/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x34/0xdc el0t_64_sync_handler+0xc0/0xc4 el0t_64_sync+0x190/0x194
CVE-2024-56621	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Cancel RTC work during ufshcd_remove() Currently, RTC work is only cancelled during __ufshcd_wl_suspend(). When ufshcd is removed in ufshcd_remove(), RTC work is not cancelled. Due to this, any further trigger of the RTC work after ufshcd_remove() would result in a NULL pointer dereference as below: Unable to handle kernel NULL pointer dereference at virtual address 00000000000002a4 Workqueue: events ufshcd_rtc_work Call trace: _raw_spin_lock_irqsave+0x34/0x8c pm_runtime_get_if_active+0x24/0xb4 ufshcd_rtc_work+0x124/0x19c process_scheduled_works+0x18c/0x2d8 worker_thread+0x144/0x280 kthread+0x11c/0x128 ret_from_fork+0x10/0x20 Since RTC work accesses the ufshcd internal structures, it should be cancelled when ufshcd is removed. So do that in ufshcd_remove(), as per the order in ufshcd_init().
CVE-2024-56622	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: sysfs: Prevent div by zero Prevent a division by 0 when monitoring is not enabled.

CVE ID	Description
CVE-2024-56623	In the Linux kernel, the following vulnerability has been resolved: scsi: qla2xxx: Fix use after free on unload System crash is observed with stack trace warning of use after free. There are 2 signals to tell dpc_thread to terminate (UNLOADING flag and kthread_stop). On setting the UNLOADING flag when dpc_thread happens to run at the time and sees the flag, this causes dpc_thread to exit and clean up itself. When kthread_stop is called for final cleanup, this causes use after free. Remove UNLOADING signal to terminate dpc_thread. Use the kthread_stop as the main signal to exit dpc_thread. [596663.812935] kernel BUG at mm/slub.c:294! [596663.812950] invalid opcode: 0000 [#1] SMP PTI [596663.812957] CPU: 13 PID: 1475935 Comm: rmmmod Kdump: loaded Tainted: G IOE ----- - - 4.18.0-240.el8.x86_64 #1 [596663.812960] Hardware name: HP ProLiant DL380p Gen8, BIOS P70 08/20/2012 [596663.812974] RIP: 0010:___slab_free+0x17d/0x360 ... [596663.813008] Call Trace: [596663.813022] ? __dentry_kill+0x121/0x170 [596663.813030] ? _cond_resched+0x15/0x30 [596663.813034] ? _cond_resched+0x15/0x30 [596663.813039] ? wait_for_completion+0x35/0x190 [596663.813048] ? try_to_wake_up+0x63/0x540 [596663.813055] free_task+0x5a/0x60 [596663.813061] kthread_stop+0xf3/0x100 [596663.813103] qla2x00_remove_one+0x284/0x440 [qla2xxx]
CVE-2024-56625	In the Linux kernel, the following vulnerability has been resolved: can: dev: can_set_termination(): allow sleeping GPIOs In commit 6e86a1543c37 ("can: dev: provide optional GPIO based termination support") GPIO based termination support was added. For no particular reason that patch uses gpiod_set_value() to set the GPIO. This leads to the following warning, if the systems uses a sleeping GPIO, i.e. behind an I2C port expander: WARNING: CPU: 0 PID: 379 at /drivers/gpio/gpiolib.c:3496 gpiod_set_value+0x50/0x6c CPU: 0 UID: 0 PID: 379 Comm: ip Not tainted 6.11.0-20241016-1 #1 823affae360cc91126e4d316d7a614a8bf86236c Replace gpiod_set_value() by gpiod_set_value_cansleep() to allow the use of sleeping GPIOs.
CVE-2024-56626	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix Out-of-Bounds Write in ksmbd_vfs_stream_write An offset from client could be a negative value, It could allows to write data outside the bounds of the allocated buffer. Note that this issue is coming when setting 'vfs objects = streams_xattr parameter' in ksmbd.conf.
CVE-2024-56627	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix Out-of-Bounds Read in ksmbd_vfs_stream_read An offset from client could be a negative value, It could lead to an out-of-bounds read from the stream_buf. Note that this issue is coming when setting 'vfs objects = streams_xattr parameter' in ksmbd.conf.

CVE ID	Description
CVE-2024-56629	In the Linux kernel, the following vulnerability has been resolved: HID: wacom: fix when get product name maybe null pointer Due to incorrect dev->product reporting by certain devices, null pointer dereferences occur when dev->product is empty, leading to potential system crashes. This issue was found on EXCELSIOR DL37-D05 device with Loongson-LS3A6000-7A2000-DL37 motherboard. Kernel logs: [56.470885] usb 4-3: new full-speed USB device number 4 using ohci-pci [56.671638] usb 4-3: string descriptor 0 read error: -22 [56.671644] usb 4-3: New USB device found, idVendor=056a, idProduct=0374, bcdDevice= 1.07 [56.671647] usb 4-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3 [56.678839] hid-generic 0003:056A:0374.0004: hiddev0,hidraw3: USB HID v1.10 Device [HID 056a:0374] on usb-0000:00:05.0-3/input0 [56.697719] CPU 2 Unable to handle kernel paging request at virtual address 0000000000000000, era == 90000000066e35c8, ra == ffff800004f98a80 [56.697732] Oops[#1]: [56.697734] CPU: 2 PID: 2742 Comm: (udev-worker) Tainted: G OE 6.6.0-loong64-desktop #25.00.2000.015 [56.697737] Hardware name: Inspur CE520L2/C09901N000000000, BIOS 2.09.00 10/11/2024 [56.697739] pc 90000000066e35c8 ra ffff800004f98a80 tp 9000000125478000 sp 900000012547b8a0 [56.697741] a0 0000000000000000 a1 ffff800004818b28 a2 0000000000000000 a3 0000000000000000 [56.697743] a4 900000012547b8f0 a5 0000000000000000 a6 0000000000000000 a7 0000000000000000 [56.697745] t0 ffff800004818b2d t1 0000000000000000 t2 0000000000000003 t3 0000000000000005 [56.697747] t4 0000000000000000 t5 0000000000000000 t6 0000000000000000 t7 0000000000000000 [56.697748] t8 0000000000000000 u0 0000000000000000 s9 0000000000000000 s0 900000011aa48028 [56.697750] s1 0000000000000000 s2 0000000000000000 s3 ffff800004818e80 s4 ffff800004810000 [56.697751] s5 90000001000b98d0 s6 ffff800004811f88 s7 ffff800005470440 s8 0000000000000000 [56.697753] ra: ffff800004f98a80 wacom_update_name+0xe0/0x300 [wacom] [56.697802] ERA: 90000000066e35c8 strsr+0x28/0x120 [56.697806] CRMD: 000000b0 (PLV0 -IE -DA +PG DACF=CC DACM=CC -WE) [56.697816] PRMD: 0000000c (PPLV0 +PIE +PWE) [56.697821] EUEN: 00000000 (-FPE -SXE -ASXE -BTE) [56.697827] ECFG: 00071c1d (LIE=0,2-4,10-12 VS=7) [56.697831] ESTAT: 00010000 [PIL] (IS= ECode=1 EsubCode=0) [56.697835] BADV: 0000000000000000 [56.697836] PRID: 0014d000 (Loongson-64bit, Loongson-3A6000) [56.697838] Modules linked in: wacom(+) bnep bluetooth rfkill qrtr nls_iso8859_1 nls_cp437 snd_hda_codec_conexant snd_hda_codec_generic ledtrig_audio snd_hda_codec_hdmi snd_hda_intel snd_intel_dspcfg snd_hda_codec snd_hda_core snd_hwdep snd_pcm snd_timer snd_soundcore input_leds mousedev led_class joydev deepin_netmonitor(OE) fuse nfnetlink dmi_sysfs ip_tables x_tables overlay amdgpu amdxcp drm_exec gpu_sched drm_buddy radeon drm_suballoc_helper i2c_algo_bit drm_ttm_helper r8169 ttm drm_display_helper spi_loongson_pci xhci_pci cec xhci_pci_renesas spi_loongson_core hid_generic realtek gpio_loongson_64bit [56.697887] Process (udev-worker) (pid: 2742, threadinfo=00000000aee0d8b4, task=00000000a9eff1f3) [56.697890] Stack : 0000000000000000 ffff800004817e00 0000000000000000 0000251c00000000 [56.697896] 0000000000000000 00000011fffff000 0000000000000000 0000000000000000 [56.697901] 0000000000000000 1b67a968695184b9 0000000000000000 90000001000b98d0 [56.697906] 90000001000bb8d0 900000011aa48028 0000000000000000 ffff800004f9d74c [56.697911] 90000001000ba000 ffff800004f9ce58 0000000000000000 ffff800005470440 [56.697916] ffff800004811f88 90000001000b98d0 9000000100da2aa8 90000001000bb8d0 [56.697921] 0000000000000000 90000001000ba000 900000011aa48028 ffff800004f9d74c [56.697926] ffff8000054704e8 90000001000bb8b8 90000001000ba000 0000000000000000 [56.697931] 90000001000bb8d0 ---truncated---
CVE-2024-56630	In the Linux kernel, the following vulnerability has been resolved: ocfs2: free inode when ocfs2_get_init_inode() fails syzbot is reporting busy inodes after unmount, for commit 9c89fe0af826 ("ocfs2: Handle error from dquot_initialize()") forgot to call iput() when new_inode() succeeded and dquot_initialize() failed.
CVE-2024-56631	In the Linux kernel, the following vulnerability has been resolved: scsi: sg: Fix slab-use-after-free read in sg_release() Fix a use-after-free bug in sg_release(), detected by syzbot with KASAN: BUG: KASAN: slab-use-after-free in lock_release+0x151/0xa30 kernel/locking/lockdep.c:5838 ___mutex_unlock_slowpath+0xe2/0x750 kernel/locking/mutex.c:912 sg_release+0x1f4/0x2e0 drivers/scsi/sg.c:407 In sg_release(), the function kref_put(&sfp->f_ref, sg_remove_sfp) is called before releasing the open_rel_lock mutex. The kref_put() call may decrement the reference count of sfp to zero, triggering its cleanup through sg_remove_sfp(). This cleanup includes scheduling deferred work via sg_remove_sfp_usercontext(), which ultimately frees sfp. After kref_put(), sg_release() continues to unlock open_rel_lock and may reference sfp or sdp. If sfp has already been freed, this results in a slab-use-after-free error. Move the kref_put(&sfp->f_ref, sg_remove_sfp) call after unlocking the open_rel_lock mutex. This ensures: - No references to sfp or sdp occur after the reference count is decremented. - Cleanup functions such as sg_remove_sfp() and sg_remove_sfp_usercontext() can safely execute without impacting the mutex handling in sg_release(). The fix has been tested and validated by syzbot. This patch closes the bug reported at the following syzkaller link and ensures proper sequencing of resource cleanup and mutex operations, eliminating the risk of use-after-free errors in sg_release().
CVE-2024-56632	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix the memleak while create new ctrl failed Now while we create new ctrl failed, we have not free the tagset occupied by admin_q, here try to fix it.

CVE ID	Description
CVE-2024-56633	In the Linux kernel, the following vulnerability has been resolved: tcp_bpf: Fix the sk_mem_uncharge logic in tcp_bpf_sendmsg The current sk memory accounting logic in __SK_REDIRECT is pre-uncharging tosend bytes, which is either msg->sg.size or a smaller value apply_bytes. Potential problems with this strategy are as follows: - If the actual sent bytes are smaller than tosend, we need to charge some bytes back, as in line 487, which is okay but seems not clean. - When tosend is set to apply_bytes, as in line 417, and (ret < 0), we may miss uncharging (msg->sg.size - apply_bytes) bytes. [...] 415 tosend = msg->sg.size; 416 if (psock->apply_bytes && psock->apply_bytes < tosend) 417 tosend = psock->apply_bytes; [...] 443 sk_msg_return(sk, msg, tosend); 444 release_sock(sk); 446 origsize = msg->sg.size; 447 ret = tcp_bpf_sendmsg_redir(sk_redir, redir_ingress, 448 msg, tosend, flags); 449 sent = origsize - msg->sg.size; [...] 454 lock_sock(sk); 455 if (unlikely(ret < 0)) { 456 int free = sk_msg_free_nocharge(sk, msg); 458 if (!cork) 459 *copied -= free; 460 } [...] 487 if (eval == __SK_REDIRECT) 488 sk_mem_charge(sk, tosend - sent); [...] When running the selftest test_txmsg_redir_wait_sndmem with txmsg_apply, the following warning will be reported: -----[cut here]----- WARNING: CPU: 6 PID: 57 at net/ipv4/af_inet.c:156 inet_sock_destruct+0x190/0x1a0 Modules linked in: CPU: 6 UID: 0 PID: 57 Comm: kworker/6:0 Not tainted 6.12.0-rc1.bm.1-amd64+ #43 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 Workqueue: events sk_psock_destroy RIP: 0010:inet_sock_destruct+0x190/0x1a0 RSP: 0018:ffffad0a8021fe08 EFLAGS: 00010206 RAX: 0000000000000011 RBX: ffff9aab4475b900 RCX: ffff9aab481a0800 RDX: 0000000000000303 RSI: 0000000000000011 RDI: ffff9aab4475b900 RBP: ffff9aab4475b990 R08: 0000000000000000 R09: ffff9aab40050ec0 R10: 0000000000000000 R11: ffff9aae6fdb1d01 R12: ffff9aab49c60400 R13: ffff9aab49c60598 R14: ffff9aab49c60598 R15: dead000000000100 FS: 0000000000000000(0000) GS: ffff9aae6fd80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffec7e47bd8 CR3: 00000001a1a1c004 CR4: 000000000770ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> ? __warn+0x89/0x130 ? inet_sock_destruct+0x190/0x1a0 ? report_bug+0xfc/0x1e0 ? handle_bug+0x5c/0xa0 ? exc_invalid_op+0x17/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? inet_sock_destruct+0x190/0x1a0 __sk_destruct+0x25/0x220 sk_psock_destroy+0x2b2/0x310 process_scheduled_works+0xa3/0x3e0 worker_thread+0x117/0x240 ? __pfx_worker_thread+0x10/0x10 kthread+0xcf/0x100 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x31/0x40 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> ---[end trace 0000000000000000]--- In __SK_REDIRECT, a more concise way is delaying the uncharging after sent bytes are finalized, and uncharge this value. When (ret < 0), we shall invoke sk_msg_free. Same thing happens in case __SK_DROP, when tosend is set to apply_bytes, we may miss uncharging (msg->sg.size - apply_bytes) bytes. The same warning will be reported in selftest. [...] 468 case __SK_DROP: 469 default: 470 sk_msg_free_partial(sk, msg, tosend); 471 sk_msg_apply_bytes(psock, tosend); 472 *copied -= (tosend + delta); 473 return -EACCES; [...] So instead of sk_msg_free_partial we can do sk_msg_free here.
CVE-2024-56634	In the Linux kernel, the following vulnerability has been resolved: gpio: grgpio: Add NULL check in grgpio_probe devm_kasprintf() can return a NULL pointer on failure, but this returned value in grgpio_probe is not checked. Add NULL check in grgpio_probe, to handle kernel NULL pointer dereference error.

CVE ID	Description
CVE-2024-56635	In the Linux kernel, the following vulnerability has been resolved: net: avoid potential UAF in default_operstate() syzbot reported an UAF in default_operstate() [1] Issue is a race between device and netns dismantles. After calling __rtnl_unlock() from netdev_run_todo(), we can not assume the netns of each device is still alive. Make sure the device is not in NETREG_UNREGISTERED state, and add an ASSERT_RTNL() before the call to __dev_get_by_index(). We might move this ASSERT_RTNL() in __dev_get_by_index() in the future. [1] BUG: KASAN: slab-use-after-free in __dev_get_by_index+0x5d/0x110 net/core/dev.c:852 Read of size 8 at addr ffff888043eba1b0 by task syz.0.0/5339 CPU: 0 UID: 0 PID: 5339 Comm: syz.0.0 Not tainted 6.12.0-syzkaller-10296-gaaf20f870da0 #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report+0x143/0x180 mm/kasan/report.c:602 __dev_get_by_index+0x5d/0x110 net/core/dev.c:852 default_operstate net/core/link_watch.c:51 [inline] rfc2863_policy+0x224/0x300 net/core/link_watch.c:67 linkwatch_do_dev+0x3e/0x170 net/core/link_watch.c:170 netdev_run_todo+0x461/0x1000 net/core/dev.c:10894 rtnl_unlock net/core/rtnetlink.c:152 [inline] rtnl_net_unlock include/linux/rtnetlink.h:133 [inline] rtnl_dellink+0x760/0x8d0 net/core/rtnetlink.c:3520 rtnetlink_rcv_msg+0x791/0xc0 net/core/rtnetlink.c:6911 netlink_rcv_skb+0x1e3/0x430 net/netlink/af_netlink.c:2541 netlink_unicast_kernel net/netlink/af_netlink.c:1321 [inline] netlink_unicast+0x7f6/0x990 net/netlink/af_netlink.c:1347 netlink_sendmsg+0x8e4/0xc0 net/netlink/af_netlink.c:1891 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:726 __sys_sendmsg+0x52a/0x7e0 net/socket.c:2583 __sys_sendmsg net/socket.c:2637 [inline] __sys_sendmsg+0x269/0x350 net/socket.c:2669 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f2a3cb80809 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f2a3d9cd058 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffffdfda RBX: 00007f2a3cd45fa0 RCX: 00007f2a3cb80809 RDX: 0000000000000000 RSI: 0000000020000000 RDI: 0000000000000008 RBP: 00007f2a3cbf393e R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000000 R14: 00007f2a3cd45fa0 R15: 00007ffd03bc65c8 </TASK> Allocated by task 5339: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x98/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x243/0x390 mm/slub.c:4314 kmalloc_noprof include/linux/slab.h:901 [inline] kmalloc_array_noprof include/linux/slab.h:945 [inline] netdev_create_hash net/core/dev.c:11870 [inline] netdev_init+0x10c/0x250 net/core/dev.c:11890 ops_init+0x31e/0x590 net/core/net_namespace.c:138 setup_net+0x287/0x9e0 net/core/net_namespace.c:362 copy_net_ns+0x33f/0x570 net/core/net_namespace.c:500 create_new_namespaces+0x425/0x7b0 kernel/nsproxy.c:110 unshare_nsproxy_namespaces+0x124/0x180 kernel/nsproxy.c:228 ksys_unshare+0x57d/0xa70 kernel/fork.c:3314 __do_sys_unshare kernel/fork.c:3385 [inline] __se_sys_unshare kernel/fork.c:3383 [inline] __x64_sys_unshare+0x38/0x40 kernel/fork.c:3383 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86 ---truncated---

CVE ID	Description
CVE-2024-56636	In the Linux kernel, the following vulnerability has been resolved: geneve: do not assume mac header is set in geneve_xmit_skb() We should not assume mac header is set in output path. Use skb_eth_hdr() instead of eth_hdr() to fix the issue. sysbot reported the following : WARNING: CPU: 0 PID: 11635 at include/linux/skbuff.h:3052 skb_mac_header include/linux/skbuff.h:3052 [inline] WARNING: CPU: 0 PID: 11635 at include/linux/skbuff.h:3052 eth_hdr include/linux/if_ether.h:24 [inline] WARNING: CPU: 0 PID: 11635 at include/linux/skbuff.h:3052 geneve_xmit_skb drivers/net/geneve.c:898 [inline] WARNING: CPU: 0 PID: 11635 at include/linux/skbuff.h:3052 geneve_xmit+0x4c38/0x5730 drivers/net/geneve.c:1039 Modules linked in: CPU: 0 UID: 0 PID: 11635 Comm: syz.4.1423 Not tainted 6.12.0-syzkaller-10296-gaaf20f870da0 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010:skb_mac_header include/linux/skbuff.h:3052 [inline] RIP: 0010:eth_hdr include/linux/if_ether.h:24 [inline] RIP: 0010:geneve_xmit_skb drivers/net/geneve.c:898 [inline] RIP: 0010:geneve_xmit+0x4c38/0x5730 drivers/net/geneve.c:1039 Code: 21 c6 02 e9 35 d4 ff ff e8 a5 48 4c fb 90 0f 0b 90 e9 fd f5 ff ff e8 97 48 4c fb 90 0f 0b 90 e9 d8 f5 ff ff e8 89 48 4c fb 90 <0f> 0b 90 e9 41 e4 ff ff e8 7b 48 4c fb 90 0f 0b 90 e9 cd e7 ff ff RSP: 0018:ffff90003b2f870 EFLAGS: 00010283 RAX: 000000000000037a RBX: 000000000000ffff RCX: fffff9000dc3d000 RDX: 0000000000080000 RSI: ffffffff86428417 RDI: 0000000000000003 RBP: fffff90003b2f9f0 R08: 0000000000000003 R09: 000000000000ffff R10: 000000000000ffff R11: 0000000000000002 R12: ffff88806603c000 R13: 0000000000000000 R14: ffff8880685b2780 R15: 0000000000000e23 FS: 00007fdc2deed6c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000001b30a1dff8 CR3: 0000000056b8c000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __netdev_start_xmit include/linux/netdevice.h:5002 [inline] netdev_start_xmit include/linux/netdevice.h:5011 [inline] __dev_direct_xmit+0x58a/0x720 net/core/dev.c:4490 dev_direct_xmit include/linux/netdevice.h:3181 [inline] packet_xmit+0x1e4/0x360 net/packet/af_packet.c:285 packet_snd net/packet/af_packet.c:3146 [inline] packet_sendmsg+0x2700/0x5660 net/packet/af_packet.c:3178 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg net/socket.c:726 [inline] __sys_sendto+0x488/0x4f0 net/socket.c:2197 __do_sys_sendto net/socket.c:2204 [inline] __se_sys_sendto net/socket.c:2200 [inline] __x64_sys_sendto+0xe0/0x1c0 net/socket.c:2200 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-56637	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: Hold module reference while requesting a module User space may unload ip_set.ko while it is itself requesting a set type backend module, leading to a kernel crash. The race condition may be provoked by inserting an mdelay() right after the nfnl_unlock() call.
CVE-2024-56638	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_inner: incorrect percpu area handling under softirq Softirq can interrupt ongoing packet from process context that is walking over the percpu area that contains inner header offsets. Disable bh and perform three checks before restoring the percpu inner header offsets to validate that the percpu area is valid for this skbuff: 1) If the NFT_PKTINFO_INNER_FULL flag is set on, then this skbuff has already been parsed before for inner header fetching to register. 2) Validate that the percpu area refers to this skbuff using the skbuff pointer as a cookie. If there is a cookie mismatch, then this skbuff needs to be parsed again. 3) Finally, validate if the percpu area refers to this tunnel type. Only after these three checks the percpu area is restored to an on-stack copy and bh is enabled again. After inner header fetching, the on-stack copy is stored back to the percpu area.

CVE ID	Description
CVE-2024-56639	In the Linux kernel, the following vulnerability has been resolved: net: hsr: must allocate more bytes for RedBox support Blamed commit forgot to change hsr_init_skb() to allocate larger skb for RedBox case. Indeed, send_hsr_supervision_frame() will add two additional components (struct hsr_sup_tlv and struct hsr_sup_payload) syzbot reported the following crash: skbuff: skb_over_panic: text:ffff880af4d4b0a len:34 put:6 head:ffff88802ad29e00 data:ffff88802ad29f22 tail:0x144 end:0x140 dev:gretap0 ----- [cut here]----- kernel BUG at net/core/skbuff.c:206 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 2 UID: 0 PID: 7611 Comm: syz-executor Not tainted 6.12.0-syzkaller #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:skb_panic+0x157/0x1d0 net/core/skbuff.c:206 Code: b6 04 01 84 c0 74 04 3c 03 7e 21 8b 4b 70 41 56 45 89 e8 48 c7 c7 a0 7d 9b 8c 41 57 56 48 89 ee 52 4c 89 e2 e8 9a 76 79 f8 90 <0f> 0b 4c 89 4c 24 10 48 89 54 24 08 48 89 34 24 e8 94 76 fb f8 4c RSP: 0018:ffff90000858ab8 EFLAGS: 00010282 RAX: 0000000000000087 RBX: ffff8880598c08c0 RCX: ffffffff816d3e69 RDX: 0000000000000000 RSI: ffffffff816de786 RDI: 0000000000000005 RBP: ffffffff8c9b91c0 R08: 0000000000000005 R09: 0000000000000000 R10: 0000000000000302 R11: ffffffff961cc1d0 R12: ffffffff8afd4b0a R13: 0000000000000006 R14: ffff88804b938130 R15: 0000000000000140 FS: 000055558a3d6500(0000) GS:ffff88806a800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f1295974ff8 CR3: 000000002ab6e000 CR4: 0000000000352ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <IRQ> skb_over_panic net/core/skbuff.c:211 [inline] skb_put+0x174/0x1b0 net/core/skbuff.c:2617 send_hsr_supervision_frame+0x6fa/0x9e0 net/hsr/hsr_device.c:342 hsr_proxy_announce+0x1a3/0x4a0 net/hsr/hsr_device.c:436 call_timer_fn+0x1a0/0x610 kernel/time/timer.c:1794 expire_timers kernel/time/timer.c:1845 [inline] __run_timers+0x6e8/0x930 kernel/time/timer.c:2419 __run_timer_base kernel/time/timer.c:2430 [inline] __run_timer_base kernel/time/timer.c:2423 [inline] run_timer_base+0x111/0x190 kernel/time/timer.c:2439 run_timer_softirq+0x1a/0x40 kernel/time/timer.c:2449 handle_softirqs+0x213/0x8f0 kernel/softirq.c:554 __do_softirq kernel/softirq.c:588 [inline] invoke_softirq kernel/softirq.c:428 [inline] __irq_exit_rcu kernel/softirq.c:637 [inline] irq_exit_rcu+0xbb/0x120 kernel/softirq.c:649 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1049 [inline] sysvec_apic_timer_interrupt+0xa4/0xc0 arch/x86/kernel/apic/apic.c:1049 </IRQ>
CVE-2024-56640	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix LGR and link use-after-free issue We encountered a LGR/link use-after-free issue, which manifested as the LGR/link refcnt reaching 0 early and entering the clear process, making resource access unsafe. refcount_t: addition on 0; use-after-free. WARNING: CPU: 14 PID: 107447 at lib/refcount.c:25 refcount_warn_saturate+0x9c/0x140 Workqueue: events smc_lgr_terminate_work [smc] Call trace: refcount_warn_saturate+0x9c/0x140 __smc_lgr_terminate.part.45+0x2a8/0x370 [smc] smc_lgr_terminate_work+0x28/0x30 [smc] process_one_work+0x1b8/0x420 worker_thread+0x158/0x510 kthread+0x114/0x118 or refcount_t: underflow; use-after-free. WARNING: CPU: 6 PID: 93140 at lib/refcount.c:28 refcount_warn_saturate+0xf0/0x140 Workqueue: smc_hs_wq smc_listen_work [smc] Call trace: refcount_warn_saturate+0xf0/0x140 smc_link_put+0x1cc/0x1d8 [smc] smc_conn_free+0x110/0x1b0 [smc] smc_conn_abort+0x50/0x60 [smc] smc_listen_find_device+0x75c/0x790 [smc] smc_listen_work+0x368/0x8a0 [smc] process_one_work+0x1b8/0x420 worker_thread+0x158/0x510 kthread+0x114/0x118 It is caused by repeated release of LGR/link refcnt. One suspect is that smc_conn_free() is called repeatedly because some smc_conn_free() from server listening path are not protected by sock lock. e.g. Calls under socklock smc_listen_work ----- lock_sock(sk) smc_conn_abort smc_conn_free \- smc_conn_free \- smc_link_put \- smc_link_put (duplicated) release_sock(sk) So here add sock lock protection in smc_listen_work() path, making it exclusive with other connection operations.
CVE-2024-56641	In the Linux kernel, the following vulnerability has been resolved: net/smc: initialize close_work early to avoid warning We encountered a warning that close_work was canceled before initialization. WARNING: CPU: 7 PID: 111103 at kernel/workqueue.c:3047 __flush_work+0x19e/0x1b0 Workqueue: events smc_lgr_terminate_work [smc] RIP: 0010: __flush_work+0x19e/0x1b0 Call Trace: ? __wake_up_common+0x7a/0x190 ? work_busy+0x80/0x80 __cancel_work_timer+0xe3/0x160 smc_close_cancel_work+0x1a/0x70 [smc] smc_close_active_abort+0x207/0x360 [smc] __smc_lgr_terminate.part.38+0xc8/0x180 [smc] process_one_work+0x19e/0x340 worker_thread+0x30/0x370 ? process_one_work+0x340/0x340 kthread+0x117/0x130 ? __kthread_cancel_work+0x50/0x50 ret_from_fork+0x22/0x30 This is because when smc_close_cancel_work is triggered, e.g. the RDMA driver is rmod and the LGR is terminated, the conn->close_work is flushed before initialization, resulting in WARN_ON(!work->func). __smc_lgr_terminate smc_connect_{rdma ism} ----- smc_conn_create \- smc_lgr_register_conn for conn in lgr->conns_all \- smc_conn_kill \- smc_close_active_abort \- smc_close_cancel_work \- cancel_work_sync \- __flush_work (close_work) smc_close_init \- INIT_WORK(&close_work) So fix this by initializing close_work before establishing the connection.

CVE ID	Description
CVE-2024-56642	In the Linux kernel, the following vulnerability has been resolved: tipc: Fix use-after-free of kernel socket in cleanup_bearer(). syzkaller reported a use-after-free of UDP kernel socket in cleanup_bearer() without repro. [0][1] When bearer_disable() calls tipc_udp_disable(), cleanup of the UDP kernel socket is deferred by work calling cleanup_bearer(). tipc_exit_net() waits for such works to finish by checking tipc_net(net)->wq_count. However, the work decrements the count too early before releasing the kernel socket, unblocking cleanup_net() and resulting in use-after-free. Let's move the decrement after releasing the socket in cleanup_bearer(). [0]: ref_tracker: net notrefcnt:000000009b3d1faf has 1/1 users at sk_alloc+0x438/0x608 inet_create+0x4c8/0xcb0 __sock_create+0x350/0x6b8 sock_create_kern+0x58/0x78 udp_sock_create4+0x68/0x398 udp_sock_create+0x88/0xc8 tipc_udp_enable+0x5e8/0x848 __tipc_nl_bearer_enable+0x84c/0xed8 tipc_nl_bearer_enable+0x38/0x60 genl_family_rcv_msg_doit+0x170/0x248 genl_rcv_msg+0x400/0x5b0 netlink_rcv_skb+0x1dc/0x398 genl_rcv+0x44/0x68 netlink_unicast+0x678/0x8b0 netlink_sendmsg+0x5e4/0x898 ____sys_sendmsg+0x500/0x830 [1]: BUG: KMSAN: use-after-free in udp_hashslot include/net/udp.h:85 [inline] BUG: KMSAN: use-after-free in udp_lib_unhash+0x3b8/0x930 net/ipv4/udp.c:1979 udp_hashslot include/net/udp.h:85 [inline] udp_lib_unhash+0x3b8/0x930 net/ipv4/udp.c:1979 sk_common_release+0xaf/0x3f0 net/core/sock.c:3820 inet_release+0x1e0/0x260 net/ipv4/af_inet.c:437 inet6_release+0x6f/0xd0 net/ipv6/af_inet6.c:489 __sock_release net/socket.c:658 [inline] sock_release+0xa0/0x210 net/socket.c:686 cleanup_bearer+0x42d/0x4c0 net/tipc/udp_media.c:819 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xcaf/0x1c90 kernel/workqueue.c:3310 worker_thread+0xf6c/0x1510 kernel/workqueue.c:3391 kthread+0x531/0x6b0 kernel/kthread.c:389 ret_from_fork+0x60/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x11/0x20 arch/x86/entry/entry_64.S:244 Uninit was created at: slab_free_hook mm/slub.c:2269 [inline] slab_free mm/slub.c:4580 [inline] kmem_cache_free+0x207/0xc40 mm/slub.c:4682 net_free net/core/net_namespace.c:454 [inline] cleanup_net+0x16f2/0x19d0 net/core/net_namespace.c:647 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xcaf/0x1c90 kernel/workqueue.c:3310 worker_thread+0xf6c/0x1510 kernel/workqueue.c:3391 kthread+0x531/0x6b0 kernel/kthread.c:389 ret_from_fork+0x60/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x11/0x20 arch/x86/entry/entry_64.S:244 CPU: 0 UID: 0 PID: 54 Comm: kworker/0:2 Not tainted 6.12.0-rc1-00131-gf66ebf37d69c #7 91723d6f74857f70725e1583cba3cf4adc716cfa Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 Workqueue: events cleanup_bearer
CVE-2024-56643	In the Linux kernel, the following vulnerability has been resolved: dccp: Fix memory leak in dccp_feat_change_rcv If dccp_feat_push_confirm() fails after new value for SP feature was accepted without reconciliation ('entry == NULL' branch), memory allocated for that value with dccp_feat_clone_sp_val() is never freed. Here is the kmemleak stack for this: unreferenced object 0xffff88801d4ab488 (size 8): comm "syz-executor310", pid 1127, jiffies 4295085598 (age 41.666s) hex dump (first 8 bytes): 01 b4 4a 1d 80 88 ff ff ..J..... backtrace: [<00000000db7cabfe>] kmemdup+0x23/0x50 mm/util.c:128 [<0000000019b38405>] kmemdup include/linux/string.h:465 [inline] [<0000000019b38405>] dccp_feat_clone_sp_val net/dccp/feat.c:371 [inline] [<0000000019b38405>] dccp_feat_clone_sp_val net/dccp/feat.c:367 [inline] [<0000000019b38405>] dccp_feat_change_rcv net/dccp/feat.c:1145 [inline] [<0000000019b38405>] dccp_feat_parse_options+0x1196/0x2180 net/dccp/feat.c:1416 [<00000000b1f6d94a>] dccp_parse_options+0xa2a/0x1260 net/dccp/options.c:125 [<0000000030d7b621>] dccp_rcv_state_process+0x197/0x13d0 net/dccp/input.c:650 [<000000001f74c72e>] dccp_v4_do_rcv+0xf9/0x1a0 net/dccp/ipv4.c:688 [<00000000a6c24128>] sk_backlog_rcv include/net/sock.h:1041 [inline] [<00000000a6c24128>] __release_sock+0x139/0x3b0 net/core/sock.c:2570 [<00000000cf1f3a53>] release_sock+0x54/0x1b0 net/core/sock.c:3111 [<000000008422fa23>] inet_wait_for_connect net/ipv4/af_inet.c:603 [inline] [<000000008422fa23>] __inet_stream_connect+0x5d0/0xf70 net/ipv4/af_inet.c:696 [<0000000015b6f64d>] inet_stream_connect+0x53/0xa0 net/ipv4/af_inet.c:735 [<0000000010122488>] __sys_connect_file+0x15c/0x1a0 net/socket.c:1865 [<00000000b4b70023>] __sys_connect+0x165/0x1a0 net/socket.c:1882 [<00000000f4cb3815>] __do_sys_connect net/socket.c:1892 [inline] [<00000000f4cb3815>] __se_sys_connect net/socket.c:1889 [inline] [<00000000f4cb3815>] __x64_sys_connect+0x6e/0xb0 net/socket.c:1889 [<00000000e7b1e839>] do_syscall_64+0x33/0x40 arch/x86/entry/common.c:46 [<0000000055e91434>] entry_SYSCALL_64_after_hwframe+0x67/0xd1 Clean up the allocated memory in case of dccp_feat_push_confirm() failure and bail out with an error reset code. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.

CVE ID	Description
CVE-2024-56644	In the Linux kernel, the following vulnerability has been resolved: net/ipv6: release expired exception dst cached in socket Dst objects get leaked in ip6_negative_advice() when this function is executed for an expired IPv6 route located in the exception table. There are several conditions that must be fulfilled for the leak to occur: * an ICMPv6 packet indicating a change of the MTU for the path is received, resulting in an exception dst being created * a TCP connection that uses the exception dst for routing packets must start timing out so that TCP begins retransmissions * after the exception dst expires, the FIB6 garbage collector must not run before TCP executes ip6_negative_advice() for the expired exception dst When TCP executes ip6_negative_advice() for an exception dst that has expired and if no other socket holds a reference to the exception dst, the refcount of the exception dst is 2, which corresponds to the increment made by dst_init() and the increment made by the TCP socket for which the connection is timing out. The refcount made by the socket is never released. The refcount of the dst is decremented in sk_dst_reset() but that decrement is counteracted by a dst_hold() intentionally placed just before the sk_dst_reset() in ip6_negative_advice(). After ip6_negative_advice() has finished, there is no other object tied to the dst. The socket lost its reference stored in sk_dst_cache and the dst is no longer in the exception table. The exception dst becomes a leaked object. As a result of this dst leak, an unbalanced refcount is reported for the loopback device of a net namespace being destroyed under kernels that do not contain e5f80fcf869a ("ipv6: give an IPv6 dev to blackhole_netdev"): unregister_netdevice: waiting for lo to become free. Usage count = 2 Fix the dst leak by removing the dst_hold() in ip6_negative_advice(). The patch that introduced the dst_hold() in ip6_negative_advice() was 92f1655aa2b22 ("net: fix __dst_negative_advice() race"). But 92f1655aa2b22 merely refactored the code with regards to the dst refcount so the issue was present even before 92f1655aa2b22. The bug was introduced in 54c1a859efd9f ("ipv6: Don't drop cache route entry unless timer actually expired.") where the expired cached route is deleted and the sk_dst_cache member of the socket is set to NULL by calling dst_negative_advice() but the refcount belonging to the socket is left unbalanced. The IPv4 version - ipv4_negative_advice() - is not affected by this bug. When the TCP connection times out ipv4_negative_advice() merely resets the sk_dst_cache of the socket while decrementing the refcount of the exception dst.
CVE-2024-56645	In the Linux kernel, the following vulnerability has been resolved: can: j1939: j1939_session_new(): fix skb reference counting Since j1939_session_skb_queue() does an extra skb_get() for each new skb, do the same for the initial one in j1939_session_new() to avoid refcount underflow. [mk1: clean up commit message]
CVE-2024-56647	In the Linux kernel, the following vulnerability has been resolved: net: Fix icmp host relookup triggering ip_rt_bug arp link failure may trigger ip_rt_bug while xfrm enabled, call trace is: WARNING: CPU: 0 PID: 0 at net/ipv4/route.c:1241 ip_rt_bug+0x14/0x20 Modules linked in: CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.0-rc6-00077-g2e1b3cc9d7f7 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:ip_rt_bug+0x14/0x20 Call Trace: <IRQ> ip_send_skb+0x14/0x40 __icmp_send+0x42d/0x6a0 ipv4_link_failure+0xe2/0x1d0 arp_error_report+0x3c/0x50 neigh_invalidate+0x8d/0x100 neigh_timer_handler+0x2e1/0x330 call_timer_fn+0x21/0x120 __run_timer_base.part.0+0x1c9/0x270 run_timer_softirq+0x4c/0x80 handle_softirqs+0xac/0x280 irq_exit_rcu+0x62/0x80 sysvec_apic_timer_interrupt+0x77/0x90 The script below reproduces this scenario: ip xfrm policy add src 0.0.0.0/0 dst 0.0.0.0/0 \ dir out priority 0 ptype main flag localok icmp ip l a veth1 type veth ip a 192.168.141.111/24 dev veth0 ip l s veth0 up ping 192.168.141.155 -c 1 icmp_route_lookup() create input routes for locally generated packets while xfrm relookup ICMP traffic. Then it will set input route (dst->out = ip_rt_bug) to skb for DESTUNREACH. For ICMP err triggered by locally generated packets, dst->dev of output route is loopback. Generally, xfrm relookup verification is not required on loopback interfaces (net.ipv4.conf.lo.disable_xfrm = 1). Skip icmp relookup for locally generated packets to fix it.

CVE ID	Description
CVE-2024-56648	In the Linux kernel, the following vulnerability has been resolved: net: hsr: avoid potential out-of-bound access in fill_frame_info() syzbot is able to feed a packet with 14 bytes, pretending it is a vlan one. Since fill_frame_info() is relying on skb->mac_len already, extend the check to cover this case. BUG: KMSAN: uninit-value in fill_frame_info net/hsr/hsr_forward.c:709 [inline] BUG: KMSAN: uninit-value in hsr_forward_skb+0x9ee/0x3b10 net/hsr/hsr_forward.c:724 fill_frame_info net/hsr/hsr_forward.c:709 [inline] hsr_forward_skb+0x9ee/0x3b10 net/hsr/hsr_forward.c:724 hsr_dev_xmit+0x2f0/0x350 net/hsr/hsr_device.c:235 __netdev_start_xmit include/linux/netdevice.h:5002 [inline] netdev_start_xmit include/linux/netdevice.h:5011 [inline] xmit_one net/core/dev.c:3590 [inline] dev_hard_start_xmit+0x247/0xa20 net/core/dev.c:3606 __dev_queue_xmit+0x366a/0x57d0 net/core/dev.c:4434 dev_queue_xmit include/linux/netdevice.h:3168 [inline] packet_xmit+0x9c/0x6c0 net/packet/af_packet.c:276 packet_snd net/packet/af_packet.c:3146 [inline] packet_sendmsg+0x91ae/0xa6f0 net/packet/af_packet.c:3178 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:726 __sys_sendto+0x594/0x750 net/socket.c:2197 __do_sys_sendto net/socket.c:2204 [inline] __se_sys_sendto net/socket.c:2200 [inline] __x64_sys_sendto+0x125/0x1d0 net/socket.c:2200 x64_sys_call+0x346a/0x3c30 arch/x86/include/generated/asm/syscalls_64.h:45 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slub.c:4091 [inline] slab_alloc_node mm/slub.c:4134 [inline] kmem_cache_alloc_node_noprof+0x6bf/0xb80 mm/slub.c:4186 kmem_cache_alloc+0x13d/0x4a0 net/core/skbuff.c:587 __alloc_skb+0x363/0x7b0 net/core/skbuff.c:678 alloc_skb include/linux/skbuff.h:1323 [inline] alloc_skb_with_frags+0xc8/0xd00 net/core/skbuff.c:6612 sock_alloc_send_pskb+0xa81/0xbf0 net/core/sock.c:2881 packet_alloc_skb net/packet/af_packet.c:2995 [inline] packet_snd net/packet/af_packet.c:3089 [inline] packet_sendmsg+0x74c6/0xa6f0 net/packet/af_packet.c:3178 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:726 __sys_sendto+0x594/0x750 net/socket.c:2197 __do_sys_sendto net/socket.c:2204 [inline] __se_sys_sendto net/socket.c:2200 [inline] __x64_sys_sendto+0x125/0x1d0 net/socket.c:2200 x64_sys_call+0x346a/0x3c30 arch/x86/include/generated/asm/syscalls_64.h:45 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-56649	In the Linux kernel, the following vulnerability has been resolved: net: enetc: Do not configure preemptible TCs if SIs do not support Both ENETC PF and VF drivers share enetc_setup_tc_mqprio() to configure MQPRIO. And enetc_setup_tc_mqprio() calls enetc_change_preemptible_tcs() to configure preemptible TCs. However, only PF is able to configure preemptible TCs. Because only PF has related registers, while VF does not have these registers. So for VF, its hw->port pointer is NULL. Therefore, VF will access an invalid pointer when accessing a non-existent register, which will cause a crash issue. The simplified log is as follows. root@ls1028ardb:~# tc qdisc add dev eno0vf0 parent root handle 100: \ mqprio num_tc 4 map 0 0 1 1 2 2 3 3 queues 1@0 1@1 1@2 1@3 hw 1 [187.290775] Unable to handle kernel paging request at virtual address 00000000000001f00 [187.424831] pc : enetc_mm_commit_preemptible_tcs+0x1c4/0x400 [187.430518] lr : enetc_mm_commit_preemptible_tcs+0x30c/0x400 [187.511140] Call trace: [187.513588] enetc_mm_commit_preemptible_tcs+0x1c4/0x400 [187.518918] enetc_setup_tc_mqprio+0x180/0x214 [187.523374] enetc_vf_setup_tc+0x1c/0x30 [187.527306] mqprio_enable_offload+0x144/0x178 [187.531766] mqprio_init+0x3ec/0x668 [187.535351] qdisc_create+0x15c/0x488 [187.539023] tc_modify_qdisc+0x398/0x73c [187.542958] rtnetlink_rcv_msg+0x128/0x378 [187.547064] netlink_rcv_skb+0x60/0x130 [187.550910] rtnetlink_rcv+0x18/0x24 [187.554492] netlink_unicast+0x300/0x36c [187.558425] netlink_sendmsg+0x1a8/0x420 [187.606759] ---[end trace 0000000000000000]--- In addition, some PFs also do not support configuring preemptible TCs, such as eno1 and eno3 on LS1028A. It won't crash like it does for VFs, but we should prevent these PFs from accessing these unimplemented registers.

CVE ID	Description
CVE-2024-56650	In the Linux kernel, the following vulnerability has been resolved: netfilter: x_tables: fix LED ID check in led_tg_check() Syzbot has reported the following BUG detected by KASAN: BUG: KASAN: slab-out-of-bounds in strlen+0x58/0x70 Read of size 1 at addr ffff8881022da0c8 by task repro/5879 ... Call Trace: <TASK> dump_stack_lvl+0x241/0x360 ? __pfx_dump_stack_lvl+0x10/0x10 ? __pfx__printk+0x10/0x10 ? __printk+0xd5/0x120 ? __virt_addr_valid+0x183/0x530 ? __virt_addr_valid+0x183/0x530 print_report+0x169/0x550 ? __virt_addr_valid+0x183/0x530 ? __virt_addr_valid+0x183/0x530 ? __virt_addr_valid+0x45f/0x530 ? __phys_addr+0xba/0x170 ? strlen+0x58/0x70 kasan_report+0x143/0x180 ? strlen+0x58/0x70 strlen+0x58/0x70 kstrdup+0x20/0x80 led_tg_check+0x18b/0x3c0 xt_check_target+0x3bb/0xa40 ? __pfx_xt_check_target+0x10/0x10 ? stack_depot_save_flags+0x6e4/0x830 ? nft_target_init+0x174/0xc30 nft_target_init+0x82d/0xc30 ? __pfx_nft_target_init+0x10/0x10 ? nf_tables_newrule+0x1609/0x2980 ? nf_tables_newrule+0x1609/0x2980 ? rcu_is_watching+0x15/0xb0 ? nf_tables_newrule+0x1609/0x2980 ? nf_tables_newrule+0x1609/0x2980 ? __kmalloc_noprof+0x21a/0x400 nf_tables_newrule+0x1860/0x2980 ? __pfx_nf_tables_newrule+0x10/0x10 ? __nla_parse+0x40/0x60 nfnetlink_rcv+0x14e5/0x2ab0 ? __pfx_validate_chain+0x10/0x10 ? __pfx_nfnetlink_rcv+0x10/0x10 ? __lock_acquire+0x1384/0x2050 ? netlink_deliver_tap+0x2e/0x1b0 ? __pfx_lock_release+0x10/0x10 ? netlink_deliver_tap+0x2e/0x1b0 netlink_unicast+0x7f8/0x990 ? __pfx_netlink_unicast+0x10/0x10 ? __virt_addr_valid+0x183/0x530 ? __check_object_size+0x48e/0x900 netlink_sendmsg+0x8e4/0xcb0 ? __pfx_netlink_sendmsg+0x10/0x10 ? aa_sock_msg_perm+0x91/0x160 ? __pfx_netlink_sendmsg+0x10/0x10 __sock_sendmsg+0x223/0x270 __sys_sendmsg+0x52a/0x7e0 ? __pfx__sys_sendmsg+0x10/0x10 __sys_sendmsg+0x292/0x380 ? __pfx__sys_sendmsg+0x10/0x10 ? lockdep_hardirqs_on_prepare+0x43d/0x780 ? __pfx_lockdep_hardirqs_on_prepare+0x10/0x10 ? exc_page_fault+0x590/0x8c0 ? do_syscall_64+0xb6/0x230 do_syscall_64+0xf3/0x230 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... </TASK> Since an invalid (without '\0' byte at all) byte sequence may be passed from userspace, add an extra check to ensure that such a sequence is rejected as possible ID and so never passed to 'kstrdup()' and further.
CVE-2024-56651	In the Linux kernel, the following vulnerability has been resolved: can: hi311x: hi3110_can_ist(): fix potential use-after-free The commit a22bd630cfff ("can: hi311x: do not report txerr and rxerr during bus-off") removed the reporting of rxerr and txerr even in case of correct operation (i. e. not bus-off). The error count information added to the CAN frame after netif_rx() is a potential use after free, since there is no guarantee that the skb is in the same state. It might be freed or reused. Fix the issue by postponing the netif_rx() call in case of txerr and rxerr reporting.
CVE-2024-56652	In the Linux kernel, the following vulnerability has been resolved: drm/xe/reg_sr: Remove register pool That pool implementation doesn't really work: if the krealloc happens to move the memory and return another address, the entries in the xarray become invalid, leading to use-after-free later: BUG: KASAN: slab-use-after-free in xe_reg_sr_apply_mmio+0x570/0x760 [xe] Read of size 4 at addr ffff8881244b2590 by task modprobe/2753 Allocated by task 2753: kasan_save_stack+0x39/0x70 kasan_save_track+0x14/0x40 kasan_save_alloc_info+0x37/0x60 __kasan_kmalloc+0xc3/0xd0 __kmalloc_node_track_caller_noprof+0x200/0x6d0 krealloc_noprof+0x229/0x380 Simplify the code to fix the bug. A better pooling strategy may be added back later if needed. (cherry picked from commit e5283bd4dfecbd3335f43b62a68e24dae23f59e4)

CVE ID	Description
CVE-2024-56653	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btmtk: avoid UAF in btmtk_process_coredump hci_devcd_append may lead to the release of the skb, so it cannot be accessed once it is called. ===== BUG: KASAN: slab-use-after-free in btmtk_process_coredump+0x2a7/0x2d0 [btmtk] Read of size 4 at addr ffff888033cfabb0 by task kworker/0:3/82 CPU: 0 PID: 82 Comm: kworker/0:3 Tainted: G U 6.6.40-lockdep-03464-g1d8b4eb3060e #1 b0b3c1cc0c842735643fb141799d97921d1f688c Hardware name: Google Yaviks_Ufs/Yaviks_Ufs, BIOS Google_Yaviks_Ufs.15217.552.0 05/07/2024 Workqueue: events btusb_rx_work [btusb] Call Trace: <TASK> dump_stack_lvl +0xfd/0x150 print_report+0x131/0x780 kasan_report+0x177/0x1c0 btmtk_process_coredump +0x2a7/0x2d0 [btmtk 03edd567dd71a65958807c95a65db31d433e1d01] btusb_recv_acl_mtk +0x11c/0x1a0 [btusb 675430d1e87c4f24d0c1f80efe600757af0f32bec] btusb_rx_work+0x9e/0xe0 [btusb 675430d1e87c4f24d0c1f80efe600757af0f32bec] worker_thread+0xe44/0x2cc0 kthread +0x2ff/0x3a0 ret_from_fork+0x51/0x80 ret_from_fork_asm+0x1b/0x30 </TASK> Allocated by task 82: stack_trace_save+0xdc/0x190 kasan_set_track+0x4e/0x80 __kasan_slab_alloc+0x4e/0x60 kmem_cache_alloc+0x19f/0x360 skb_clone+0x132/0xf70 btusb_recv_acl_mtk+0x104/0x1a0 [btusb] btusb_rx_work+0x9e/0xe0 [btusb] worker_thread+0xe44/0x2cc0 kthread+0x2ff/0x3a0 ret_from_fork +0x51/0x80 ret_from_fork_asm+0x1b/0x30 Freed by task 1733: stack_trace_save+0xdc/0x190 kasan_set_track+0x4e/0x80 kasan_save_free_info+0x28/0xb0 ____kasan_slab_free+0xdf/0x170 kmem_cache_free+0x183/0x3f0 hci_devcd_rx+0x91a/0x2060 [bluetooth] worker_thread+0xe44/0x2cc0 kthread+0x2ff/0x3a0 ret_from_fork+0x51/0x80 ret_from_fork_asm+0x1b/0x30 The buggy address belongs to the object at ffff888033cfab40 which belongs to the cache skbuff_head_cache of size 232 The buggy address is located 112 bytes inside of freed 232-byte region [ffff888033cfab40, ffff888033cfac28) The buggy address belongs to the physical page: page:00000000a174ba93 refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x33cfa head:00000000a174ba93 order:1 entire_mapcount:0 nr_pages_mapped:0 pincount:0 anon flags: 0x4000000000000840(slab head zone=1) page_type: 0xffffffffff() raw: 40000000000000840 ffff888100848a00 0000000000000000 0000000000000001 raw: 0000000000000000 0000000080190019 00000001fffffff 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff888033cfaa80: fb fb fb fb fb fb fb fb fb fc fc fc ffff888033cfab00: fc fc fc fc fc fc fc fa fb fb fb fb fb fb fb >ffff888033cfab80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ^ ffff888033cfac00: fb fb fb fb fb fc fc fc fc fc fc fc ffff888033cfac80: fb fb fb fb fb fb fb fb fb fb fb fb ===== Check if we need to call hci_devcd_complete before calling hci_devcd_append. That requires that we check data->cd_info.cnt >= MTK_COREDUMP_NUM instead of data->cd_info.cnt > MTK_COREDUMP_NUM, as we increment data->cd_info.cnt only once the call to hci_devcd_append succeeds.
CVE-2024-56654	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: Fix using rcu_read_(un)lock while iterating The usage of rcu_read_(un)lock while inside list_for_each_entry_rcu is not safe since for the most part entries fetched this way shall be treated as rcu_dereference: Note that the value returned by rcu_dereference() is valid only within the enclosing RCU read-side critical section [1]. For example, the following is **not** legal:: rcu_read_lock(); p = rcu_dereference(head.next); rcu_read_unlock(); x = p->address; /* BUG!!! */ rcu_read_lock(); y = p->data; /* BUG!!! */ rcu_read_unlock();

CVE ID	Description
CVE-2024-56656	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix aggregation ID mask to prevent oops on 5760X chips The 5760X (P7) chip's HW GRO/LRO interface is very similar to that of the previous generation (5750X or P5). However, the aggregation ID fields in the completion structures on P7 have been redefined from 16 bits to 12 bits. The freed up 4 bits are redefined for part of the metadata such as the VLAN ID. The aggregation ID mask was not modified when adding support for P7 chips. Including the extra 4 bits for the aggregation ID can potentially cause the driver to store or fetch the packet header of GRO/LRO packets in the wrong TPA buffer. It may hit the BUG() condition in __skb_pull() because the SKB contains no valid packet header: kernel BUG at include/linux/skbuff.h:2766! Oops: invalid opcode: 0000 1 PREEMPT SMP NOPTI CPU: 4 UID: 0 PID: 0 Comm: swapper/4 Kdump: loaded Tainted: G OE 6.12.0-rc2+ #7 Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: Dell Inc. PowerEdge R760/0VRV9X, BIOS 1.0.1 12/27/2022 RIP: 0010:eth_type_trans+0xda/0x140 Code: 80 00 00 00 eb c1 8b 47 70 2b 47 74 48 8b 97 d0 00 00 00 83 f8 01 7e 1b 48 85 d2 74 06 66 83 3a ff 74 09 b8 00 04 00 00 eb a5 <0f> 0b b8 00 01 00 00 eb 9c 48 85 ff 74 eb 31 f6 b9 02 00 00 00 48 RSP: 0018:ff615003803fcc28 EFLAGS: 00010283 RAX: 00000000000022d2 RBX: 0000000000000003 RCX: ff2e8c25da334040 RDX: 0000000000000040 RSI: ff2e8c25c1ce8000 RDI: ff2e8c25869f9000 RBP: ff2e8c258c31c000 R08: ff2e8c25da334000 R09: 0000000000000001 R10: ff2e8c25da3342c0 R11: ff2e8c25c1ce89c0 R12: ff2e8c258e0990b0 R13: ff2e8c25bb120000 R14: ff2e8c25c1ce89c0 R15: ff2e8c25869f9000 FS: 0000000000000000(0000) GS:ff2e8c34be300000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055f05317e4c8 CR3: 000000108bac6006 CR4: 0000000000773ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe07f0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <IRQ> ? die+0x33/0x90 ? do_trap+0xd9/0x100 ? eth_type_trans+0xda/0x140 ? do_error_trap+0x65/0x80 ? eth_type_trans+0xda/0x140 ? exc_invalid_op+0x4e/0x70 ? eth_type_trans+0xda/0x140 ? asm_exc_invalid_op+0x16/0x20 ? eth_type_trans+0xda/0x140 bnxt_tpa_end+0x10b/0x6b0 [bnxt_en] ? bnxt_tpa_start+0x195/0x320 [bnxt_en] bnxt_rx_pkt+0x902/0xd90 [bnxt_en] ? __bnxt_tx_int.constprop.0+0x89/0x300 [bnxt_en] ? kmem_cache_free+0x343/0x440 ? __bnxt_tx_int.constprop.0+0x24f/0x300 [bnxt_en] __bnxt_poll_work+0x193/0x370 [bnxt_en] bnxt_poll_p5+0x9a/0x300 [bnxt_en] ? try_to_wake_up+0x209/0x670 __napi_poll+0x29/0x1b0 Fix it by redefining the aggregation ID mask for P5_PLUS chips to be 12 bits. This will work because the maximum aggregation ID is less than 4096 on all P5_PLUS chips.
CVE-2024-56657	In the Linux kernel, the following vulnerability has been resolved: ALSA: control: Avoid WARN() for symlink errors Using WARN() for showing the error of symlink creations don't give more information than telling that something goes wrong, since the usual code path is a lregister callback from each control element creation. More badly, the use of WARN() rather confuses fuzzer as if it were serious issues. This patch downgrades the warning messages to use the normal dev_err() instead of WARN(). For making it clearer, add the function name to the prefix, too.

[illegible]

CVE ID	Description
CVE-2024-56659	In the Linux kernel, the following vulnerability has been resolved: net: lapb: increase LAPB_HEADER_LEN It is unclear if net/lapb code is supposed to be ready for 8021q. We can at least avoid crashes like the following : skbuff: skb_under_panic: text:fffff8aabe1f6 len:24 put:20 head:ffff88802824a400 data:ffff88802824a3fe tail:0x16 end:0x140 dev:nr0.2 -----[cut here]----- kernel BUG at net/core/skbuff.c:206 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 1 UID: 0 PID: 5508 Comm: dhcpcd Not tainted 6.12.0-rc7-syzkaller-00144-g66418447d27b #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 10/30/2024 RIP: 0010:skb_panic net/core/skbuff.c:206 [inline] RIP: 0010:skb_under_panic +0x14b/0x150 net/core/skbuff.c:216 Code: 0d 8d 48 c7 c6 2e 9e 29 8e 48 8b 54 24 08 8b 0c 24 44 8b 44 24 04 4d 89 e9 50 41 54 41 57 41 56 e8 1a 6f 37 02 48 83 c4 20 90 <0f> 0b 0f 1f 00 90 90 90 90 90 90 90 90 90 90 90 90 f3 RSP: 0018:ffffc90002ddf638 EFLAGS: 00010282 RAX: 0000000000000086 RBX: dffffc0000000000 RCX: 7a24750e538ff600 RDX: 0000000000000000 RSI: 0000000000000201 RDI: 0000000000000000 RBP: ffff888034a86650 R08: ffffffff8174b13c R09: 1ffff920005bbe60 R10: dffffc0000000000 R11: fffff520005bbe61 R12: 0000000000000140 R13: ffff88802824a400 R14: ffff88802824a3fe R15: 0000000000000016 FS: 00007f2a5990d740(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 00000000080050033 CR2: 000000110c2631fd CR3: 0000000029504000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> skb_push+0xe5/0x100 net/core/skbuff.c:2636 nr_header+0x36/0x320 net/netrom/nr_dev.c:69 dev_hard_header include/linux/netdevice.h:3148 [inline] vlan_dev_hard_header+0x359/0x480 net/8021q/vlan_dev.c:83 dev_hard_header include/linux/netdevice.h:3148 [inline] lapbeth_data_transmit+0x1f6/0x2a0 drivers/net/wan/lapbether.c:257 lapb_data_transmit+0x91/0xb0 net/lapb/lapb_iface.c:447 lapb_transmit_buffer +0x168/0x1f0 net/lapb/lapb_out.c:149 lapb_establish_data_link+0x84/0xd0 lapb_device_event +0x4e0/0x670 notifier_call_chain+0x19f/0x3e0 kernel/notifier.c:93 __dev_notify_flags+0x207/0x400 dev_change_flags+0xf0/0x1a0 net/core/dev.c:8922 devinet_ioctl+0xa4e/0x1aa0 net/ipv4/devinet.c:1188 inet_ioctl+0x3d7/0x4f0 net/ipv4/af_inet.c:1003 sock_do_ioctl+0x158/0x460 net/socket.c:1227 sock_ioctl+0x626/0x8e0 net/socket.c:1346 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xf9/0x170 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83
CVE-2024-56660	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: DR, prevent potential error pointer dereference The dr_domain_add_vport_cap() function generally returns NULL on error but sometimes we want it to return ERR_PTR(-EBUSY) so the caller can retry. The problem here is that "ret" can be either -EBUSY or -ENOMEM and if it's and -ENOMEM then the error pointer is propagated back and eventually dereferenced in dr_ste_v0_build_src_gvmi_qpn_tag().
CVE-2024-56662	In the Linux kernel, the following vulnerability has been resolved: acpi: nfit: vmalloc-out-of-bounds Read in acpi_nfit_ctl Fix an issue detected by syzbot with KASAN: BUG: KASAN: vmalloc-out-of-bounds in cmd_to_func drivers/acpi/nfit/ core.c:416 [inline] BUG: KASAN: vmalloc-out-of-bounds in acpi_nfit_ctl+0x20e8/0x24a0 drivers/acpi/nfit/core.c:459 The issue occurs in cmd_to_func when the call_pkg->nd_reserved2 array is accessed without verifying that call_pkg points to a buffer that is appropriately sized as a struct nd_cmd_pkg. This can lead to out-of-bounds access and undefined behavior if the buffer does not have sufficient space. To address this, a check was added in acpi_nfit_ctl() to ensure that buf is not NULL and that buf_len is less than sizeof(*call_pkg) before accessing it. This ensures safe access to the members of call_pkg, including the nd_reserved2 array.

CVE ID	Description
CVE-2024-56663	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: fix NL80211_ATTR_MLO_LINK_ID off-by-one Since the netlink attribute range validation provides inclusive checking, the *max* of attribute NL80211_ATTR_MLO_LINK_ID should be IEEE80211_MLD_MAX_NUM_LINKS - 1 otherwise causing an off-by-one. One crash stack for demonstration: <pre>===== BUG: KASAN: wild-memory-access in ieee80211_tx_control_port+0x3b6/0xca0 net/mac80211/tx.c:5939 Read of size 6 at addr 001102080000000c by task fuzzer.386/9508 CPU: 1 PID: 9508 Comm: syz.1.386 Not tainted 6.1.70 #2 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x177/0x231 lib/dump_stack.c:106 print_report+0xe0/0x750 mm/kasan/report.c:398 kasan_report+0x139/0x170 mm/kasan/report.c:495 kasan_check_range+0x287/0x290 mm/kasan/ generic.c:189 memcpy+0x25/0x60 mm/kasan/shadow.c:65 ieee80211_tx_control_port+0x3b6/0xca0 net/ mac80211/tx.c:5939 rdev_tx_control_port net/wireless/rdev-ops.h:761 [inline] nl80211_tx_control_port +0x7b3/0xc40 net/wireless/nl80211.c:15453 genl_family_rcv_msg_doit+0x22e/0x320 net/netlink/ genetlink.c:756 genl_family_rcv_msg net/netlink/genetlink.c:833 [inline] genl_rcv_msg+0x539/0x740 net/netlink/genetlink.c:850 netlink_rcv_skb+0x1de/0x420 net/netlink/af_netlink.c:2508 genl_rcv +0x24/0x40 net/netlink/genetlink.c:861 netlink_unicast_kernel net/netlink/af_netlink.c:1326 [inline] netlink_unicast+0x74b/0x8c0 net/netlink/af_netlink.c:1352 netlink_sendmsg+0x882/0xb90 net/netlink/ af_netlink.c:1874 sock_sendmsg_nosec net/socket.c:716 [inline] __sock_sendmsg net/socket.c:728 [inline] __sys_sendmsg+0x5cc/0x8f0 net/socket.c:2499 __sys_sendmsg+0x21c/0x290 net/ socket.c:2553 __sys_sendmsg net/socket.c:2582 [inline] __do_sys_sendmsg net/socket.c:2591 [inline] __se_sys_sendmsg+0x19e/0x270 net/socket.c:2589 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x45/0x90 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe +0x63/0xcd Update the policy to ensure correct validation.</pre>
CVE-2024-56664	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix race between element replace and close() Element replace (with a socket different from the one stored) may race with socket's close() link popping & unlinking. __sock_map_delete() unconditionally unrefs the (wrong) element: // set map[0] = s0 map_update_elem(map, 0, s0) // drop fd of s0 close(s0) sock_map_close() lock_sock(sk) (s0!) sock_map_remove_links(sk) link = sk_psock_link_pop() sock_map_unlink(sk, link) sock_map_delete_from_link // replace map[0] with s1 map_update_elem(map, 0, s1) sock_map_update_elem(s1!) lock_sock(sk) sock_map_update_common psock = sk_psock(sk) spin_lock(&stab->lock) osk = stab->sks[idx] sock_map_add_link(..., &stab->sks[idx]) sock_map_unref(osk, &stab->sks[idx]) psock = sk_psock(osk) sk_psock_put(sk, psock) if (refcount_dec_and_test(&psock)) sk_psock_drop(sk, psock) spin_unlock(&stab->lock) unlock_sock(sk) __sock_map_delete spin_lock(&stab->lock) sk = *psk // s1 replaced s0; sk == s1 if (!sk_test sk_test == sk) // sk_test(s0) != sk(s1); no branch sk = xchg(psk, NULL) if (sk) sock_map_unref(sk, psk) // unref s1; sks[idx] will dangle psock = sk_psock(sk) sk_psock_put(sk, psock) if (refcount_dec_and_test()) sk_psock_drop(sk, psock) spin_unlock(&stab->lock) release_sock(sk) Then close(map) enqueues bpf_map_free_deferred, which finally calls sock_map_free(). This results in some refcount_t warnings along with a KASAN splat [1]. Fix __sock_map_delete(), do not allow sock_map_unref() on elements that may have been replaced. [1]: BUG: KASAN: slab-use-after-free in sock_map_free+0x10e/0x330 Write of size 4 at addr ffff8881f5b9100 by task kworker/u64:12/1063 CPU: 14 UID: 0 PID: 1063 Comm: kworker/u64:12 Not tainted 6.12.0+ #125 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Arch Linux 1.16.3-1-1 04/01/2014 Workqueue: events_unbound bpf_map_free_deferred Call Trace: <TASK> dump_stack_lvl+0x68/0x90 print_report+0x174/0x4f6 kasan_report+0xb9/0x190 kasan_check_range+0x10f/0x1e0 sock_map_free+0x10e/0x330 bpf_map_free_deferred+0x173/0x320 process_one_work+0x846/0x1420 worker_thread+0x5b3/0xf80 kthread+0x29e/0x360 ret_from_fork +0x2d/0x70 ret_from_fork_asm+0x1a/0x30 </TASK> Allocated by task 1202: kasan_save_stack +0x1e/0x40 kasan_save_track+0x10/0x30 __kasan_slab_alloc+0x85/0x90 kmem_cache_alloc_noprof +0x131/0x450 sk_prot_alloc+0x5b/0x220 sk_alloc+0x2c/0x870 unix_create1+0x88/0x8a0 unix_create +0xc5/0x180 __sock_create+0x241/0x650 __sys_socketpair+0x1ce/0x420 __x64_sys_socketpair +0x92/0x100 do_syscall_64+0x93/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e Freed by task 46: kasan_save_stack+0x1e/0x40 kasan_save_track+0x10/0x30 kasan_save_free_info+0x37/0x60 __kasan_slab_free+0x4b/0x70 kmem_cache_free+0x1a1/0x590 __sk_destruct+0x388/0x5a0 sk_psock_destroy+0x73e/0xa50 process_one_work+0x846/0x1420 worker_thread+0x5b3/0xf80 kthread +0x29e/0x360 ret_from_fork+0x2d/0x70 ret_from_fork_asm+0x1a/0x30 The bu ---truncated---
CVE-2024-56665	In the Linux kernel, the following vulnerability has been resolved: bpf,perf: Fix invalid prog_array access in perf_event_detach_bpf_prog Syzbot reported [1] crash that happens for following tracing scenario: - create tracepoint perf event with attr.inherit=1, attach it to the process and set bpf program to it - attached process forks -> chid creates inherited event the new child event shares the parent's bpf program and tp_event (hence prog_array) which is global for tracepoint - exit both process and its child -> release both events - first perf_event_detach_bpf_prog call will release tp_event->prog_array and second perf_event_detach_bpf_prog will crash, because tp_event->prog_array is NULL The fix makes sure the perf_event_detach_bpf_prog checks prog_array is valid before it tries to remove the bpf program from it. [1] https://lore.kernel.org/bpf/Z1MR6dCIKajNS6nU@krava/T/#m91dbf0688221ec7a7fc95e896a7ef9ff93b0b8ad

CVE ID	Description
CVE-2024-56667	In the Linux kernel, the following vulnerability has been resolved: drm/i915: Fix NULL pointer dereference in capture_engine When the intel_context structure contains NULL, it raises a NULL pointer dereference error in drm_info(). (cherry picked from commit 754302a5bc1bd8fd3b7d85c168b0a1af6d4bba4d)
CVE-2024-56670	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: u_serial: Fix the issue that gs_start_io crashed due to accessing null pointer Considering that in some extreme cases, when u_serial driver is accessed by multiple threads, Thread A is executing the open operation and calling the gs_open, Thread B is executing the disconnect operation and calling the gserial_disconnect function, The port->port_usb pointer will be set to NULL. E.g. Thread A Thread B gs_open() gadget_unbind_driver() gs_start_io() composite_disconnect() gs_start_rx() gserial_disconnect() ... spin_unlock(&port->port_lock) status = usb_ep_queue() spin_lock(&port->port_lock) spin_lock(&port->port_lock) port->port_usb = NULL gs_free_requests(port->port_usb->in) spin_unlock(&port->port_lock) Crash This causes thread A to access a null pointer (port->port_usb is null) when calling the gs_free_requests function, causing a crash. If port_usb is NULL, the release request will be skipped as it will be done by gserial_disconnect. So add a null pointer check to gs_start_io before attempting to access the value of the pointer port->port_usb. Call trace: gs_start_io+0x164/0x25c gs_open+0x108/0x13c tty_open+0x314/0x638 chrdev_open+0x1b8/0x258 do_dentry_open+0x2c4/0x700 vfs_open+0x2c/0x3c path_openat+0xa64/0xc60 do_filp_open+0xb8/0x164 do_sys_openat2+0x84/0xf0 __arm64_sys_openat+0x70/0x9c invoke_syscall+0x58/0x114 el0_svc_common+0x80/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x38/0x68
CVE-2024-56671	In the Linux kernel, the following vulnerability has been resolved: gpio: graniterapids: Fix vGPIO driver crash Move setting irq_chip.name from probe() function to the initialization of "irq_chip" struct in order to fix vGPIO driver crash during bootup. Crash was caused by unauthorized modification of irq_chip.name field where irq_chip struct was initialized as const. This behavior is a consequence of suboptimal implementation of gpio_irq_chip_set_chip(), which should be changed to avoid casting away const qualifier. Crash log: BUG: unable to handle page fault for address: ffffffff0ba81c0 /#PF: supervisor write access in kernel mode /#PF: error_code(0x0003) - permissions violation CPU: 33 UID: 0 PID: 1075 Comm: systemd-udevd Not tainted 6.12.0-rc6-00077-g2e1b3cc9d7f7 #1 Hardware name: Intel Corporation Kaseyville RP/Kaseyville RP, BIOS KVLDCRB1.PGS.0026.D73.2410081258 10/08/2024 RIP: 0010:gnr_gpio_probe+0x171/0x220 [gpio_graniterapids]
CVE-2024-56672	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: Fix UAF in blkcg_unpin_online() blkcg_unpin_online() walks up the blkcg hierarchy putting the online pin. To walk up, it uses blkcg_parent(blkcg) but it was calling that after blkcg_destroy_blkgs(blkcg) which could free the blkcg, leading to the following UAF: ===== BUG: KASAN: slab-use-after-free in blkcg_unpin_online+0x15a/0x270 Read of size 8 at addr ffff8881057678c0 by task kworker/9:1/117 CPU: 9 UID: 0 PID: 117 Comm: kworker/9:1 Not tainted 6.13.0-rc1-work-00182-gb8f52214c61a-dirty #48 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS unknown 02/02/2022 Workqueue: cgwb_release cgwb_release_workfn Call Trace: <TASK> dump_stack_lvl+0x27/0x80 print_report+0x151/0x710 kasan_report+0xc0/0x100 blkcg_unpin_online+0x15a/0x270 cgwb_release_workfn+0x194/0x480 process_scheduled_works+0x71b/0xe20 worker_thread+0x82a/0xbd0 kthread+0x242/0x2c0 ret_from_fork+0x33/0x70 ret_from_fork_asm+0x1a/0x30 </TASK> ... Freed by task 1944: kasan_save_track+0x2b/0x70 kasan_save_free_info+0x3c/0x50 __kasan_slab_free+0x33/0x50 kfree+0x10c/0x330 css_free_rwork_fn+0xe6/0xb30 process_scheduled_works+0x71b/0xe20 worker_thread+0x82a/0xbd0 kthread+0x242/0x2c0 ret_from_fork+0x33/0x70 ret_from_fork_asm+0x1a/0x30 Note that the UAF is not easy to trigger as the free path is indirected behind a couple RCU grace periods and a work item execution. I could only trigger it with artificial msleep() injected in blkcg_unpin_online(). Fix it by reading the parent pointer before destroying the blkcg's blkcg's.

CVE ID	Description
CVE-2024-56674	In the Linux kernel, the following vulnerability has been resolved: virtio_net: correct netdev_tx_reset_queue() invocation point When virtnet_close is followed by virtnet_open, some TX completions can possibly remain unconsumed, until they are finally processed during the first NAPI poll after the netdev_tx_reset_queue(), resulting in a crash [1]. Commit b96ed2c97c79 ("virtio_net: move netdev_tx_reset_queue() call before RX napi enable") was not sufficient to eliminate all BQL crash cases for virtio-net. This issue can be reproduced with the latest net-next master by running: <code>`while :: do ip l set DEV down; ip l set DEV up; done`</code> under heavy network TX load from inside the machine. netdev_tx_reset_queue() can actually be dropped from virtnet_open path; the device is not stopped in any case. For BQL core part, it's just like traffic nearly ceases to exist for some period. For stall detector added to BQL, even if virtnet_close could somehow lead to some TX completions delayed for long, followed by virtnet_open, we can just take it as stall as mentioned in commit 6025b9135f7a ("net: dqs: add NIC stall detector based on BQL"). Note also that users can still reset stall_max via sysfs. So, drop netdev_tx_reset_queue() from virtnet_enable_queue_pair(). This eliminates the BQL crashes. As a result, netdev_tx_reset_queue() is now explicitly required in freeze/restore path. This patch adds it to immediately after free_unused_bufs(), following the rule of thumb: netdev_tx_reset_queue() should follow any SKB freeing not followed by netdev_tx_completed_queue(). This seems the most consistent and streamlined approach, and now netdev_tx_reset_queue() runs whenever free_unused_bufs() is done. [1]: -----[cut here]----- kernel BUG at lib/dynamic_queue_limits.c:99! Oops: invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 7 UID: 0 PID: 1598 Comm: ip Tainted: G N 6.12.0net-next_main+ #2 Tainted: [N]=TEST Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), \ BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:dql_completed+0x26b/0x290 Code: b7 c2 49 89 e9 44 89 da 89 c6 4c 89 d7 e8 ed 17 47 00 58 65 ff 0d 4d 27 90 7e 0f 85 fd fe ff ff e8 ea 53 8d ff e9 f3 fe ff ff <0f> 0b 01 d2 44 89 d1 29 d1 ba 00 00 00 00 0f 48 ca e9 28 ff ff ff RSP: 0018:ffff900002b0d08 EFLAGS: 00010297 RAX: 0000000000000000 RBX: ffff888102398c80 RCX: 0000000080190009 RDX: 0000000000000000 RSI: 000000000000006a RDI: 0000000000000000 RBP: ffff888102398c00 R08: 0000000000000000 R09: 0000000000000000 R10: 00000000000000ca R11: 0000000000015681 R12: 0000000000000001 R13: ffff900002b0d68 R14: ffff88811115e000 R15: ffff8881107aca40 FS: 00007f41ded69500(0000) GS:ffff888667dc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000556ccc2dc1a0 CR3: 0000000104fd8003 CR4: 0000000000772ef0 PKRU: 55555554 Call Trace: <IRQ> ? die+0x32/0x80 ? do_trap+0xd9/0x100 ? dql_completed+0x26b/0x290 ? dql_completed+0x26b/0x290 ? do_error_trap+0x6d/0xb0 ? dql_completed+0x26b/0x290 ? exc_invalid_op+0x4c/0x60 ? dql_completed+0x26b/0x290 ? asm_exc_invalid_op+0x16/0x20 ? dql_completed+0x26b/0x290 __free_old_xmit+0xff/0x170 [virtio_net] free_old_xmit+0x54/0xc0 [virtio_net] virtnet_poll+0xf4/0xe30 [virtio_net] ? __update_load_avg_cfs_rq+0x264/0x2d0 ? update_curr+0x35/0x260 ? reweight_entity+0x1be/0x260 __napi_poll.constprop.0+0x28/0x1c0 net_rx_action+0x329/0x420 ? enqueue_hrtimer+0x35/0x90 ? trace_hardirqs_on+0x1d/0x80 ? kvm_sched_clock_read+0xd/0x20 ? sched_clock+0xc/0x30 ? kvm_sched_clock_read+0xd/0x20 ? sched_clock+0xc/0x30 ? sched_clock_cpu+0xd/0x1a0 handle_softirqs+0x138/0x3e0 do_softirq.part.0+0x89/0xc0 </IRQ> <TASK> __local_bh_enable_ip+0xa7/0xb0 virtnet_open+0xc8/0x310 [virtio_net] __dev_open+0xfa/0x1b0 __dev_change_flags+0x1de/0x250 dev_change_flags+0x22/0x60 do_setlink.isra.0+0x2df/0x10b0 ? rtnetlink_rcv_msg+0x34f/0x3f0 ? netlink_rcv_skb+0x54/0x100 ? netlink_unicas ---truncated---
CVE-2024-56675	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix UAF via mismatching bpf_prog/attachment RCU flavors Uprobes always use bpf_prog_run_array_uprobe() under tasks-trace-RCU protection. But it is possible to attach a non-sleepable BPF program to a uprobe, and non-sleepable BPF programs are freed via normal RCU (see __bpf_prog_put_noref()). This leads to UAF of the bpf_prog because a normal RCU grace period does not imply a tasks-trace-RCU grace period. Fix it by explicitly waiting for a tasks-trace-RCU grace period after removing the attachment of a bpf_prog to a perf_event.

CVE ID	Description
CVE-2024-56677	In the Linux kernel, the following vulnerability has been resolved: powerpc/fadump: Move fadump_cma_init to setup_arch() after initmem_init() During early init CMA_MIN_ALIGNMENT_BYTES can be PAGE_SIZE, since pageblock_order is still zero and it gets initialized later during initmem_init() e.g. setup_arch() -> initmem_init() -> sparse_init() -> set_pageblock_order() One such use case where this causes issue is - early_setup() -> early_init_devtree() -> fadump_reserve_mem() -> fadump_cma_init() This causes CMA memory alignment check to be bypassed in cma_init_reserved_mem(). Then later cma_activate_area() can hit a VM_BUG_ON_PAGE(pfn & ((1 << order) - 1)) if the reserved memory area was not pageblock_order aligned. Fix it by moving the fadump_cma_init() after initmem_init(), where other such cma reservations also gets called. <stack trace> ===== page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x10010 flags: 0x13ffff8000000000(node=1 zone=0 lastcpupid=0x7ffff) CMA raw: 013ffff800000000 5deadbeef0000100 5deadbeef0000122 0000000000000000 raw: 0000000000000000 0000000000000000 00000000ffffff 0000000000000000 page dumped because: VM_BUG_ON_PAGE(pfn & ((1 << order) - 1)) -----[cut here]----- kernel BUG at mm/page_alloc.c:778! Call Trace: __free_one_page+0x57c/0x7b0 (unreliable) free_pcppages_bulk+0x1a8/0x2c8 free_unref_page_commit+0x3d4/0x4e4 free_unref_page+0x458/0x6d0 init_cma_reserved_pageblock+0x114/0x198 cma_init_reserved_areas+0x270/0x3e0 do_one_initcall+0x80/0x2f8 kernel_init_freeable+0x33c/0x530 kernel_init+0x34/0x26c ret_from_kernel_user_thread+0x14/0x1c
CVE-2024-56678	In the Linux kernel, the following vulnerability has been resolved: powerpc/mm/fault: Fix kfence page fault reporting copy_from_kernel_nofault() can be called when doing read of /proc/kcore. /proc/kcore can have some unmapped kfence objects which when read via copy_from_kernel_nofault() can cause page faults. Since *_nofault() functions define their own fixup table for handling fault, use that instead of asking kfence to handle such faults. Hence we search the exception tables for the nip which generated the fault. If there is an entry then we let the fixup table handler handle the page fault by returning an error from within __do_page_fault(). This can be easily triggered if someone tries to do dd from /proc/kcore. eg. dd if=/proc/kcore of=/dev/null bs=1M Some example false negatives: ===== BUG: KFENCE: invalid read in copy_from_kernel_nofault+0x9c/0x1a0 Invalid read at 0xc0000000fdff0000: copy_from_kernel_nofault+0x9c/0x1a0 0xc00000000665f950 read_kcore_iter+0x57c/0xa04 proc_reg_read_iter+0xe4/0x16c vfs_read+0x320/0x3ec ksys_read+0x90/0x154 system_call_exception+0x120/0x310 system_call_vectored_common+0x15c/0x2ec BUG: KFENCE: use-after-free read in copy_from_kernel_nofault+0x9c/0x1a0 Use-after-free read at 0xc0000000fe050000 (in kfence-#2): copy_from_kernel_nofault+0x9c/0x1a0 0xc00000000665f950 read_kcore_iter+0x57c/0xa04 proc_reg_read_iter+0xe4/0x16c vfs_read+0x320/0x3ec ksys_read+0x90/0x154 system_call_exception+0x120/0x310 system_call_vectored_common+0x15c/0x2ec
CVE-2024-56679	In the Linux kernel, the following vulnerability has been resolved: octeonx2-pf: handle otx2_mbox_get_rsp errors in otx2_common.c Add error pointer check after calling otx2_mbox_get_rsp().
CVE-2024-56681	In the Linux kernel, the following vulnerability has been resolved: crypto: bcm - add error check in the ahash_hmac_init function The ahash_init functions may return fails. The ahash_hmac_init should not return ok when ahash_init returns error. For an example, ahash_init will return -ENOMEM when allocation memory is error.
CVE-2024-56683	In the Linux kernel, the following vulnerability has been resolved: drm/vc4: hdmi: Avoid hang with debug registers when suspended Trying to read /sys/kernel/debug/dri/1/hdmi1_regs when the hdmi is disconnected results in a fatal system hang. This is due to the pm suspend code disabling the dvp clock. That is just a gate of the 108MHz clock in DVP_HT_RPI_MISC_CONFIG, which results in accesses hanging AXI bus. Protect against this.

CVE ID	Description
CVE-2024-56685	In the Linux kernel, the following vulnerability has been resolved: ASoC: mediatek: Check num_codecs is not zero to avoid panic during probe Following commit 13f58267cda3 ("ASoC: soc.h: don't create dummy Component via COMP_DUMMY()"), COMP_DUMMY() became an array with zero length, and only gets populated with the dummy struct after the card is registered. Since the sound card driver's probe happens before the card registration, accessing any of the members of a dummy component during probe will result in undefined behavior. This can be observed in the mt8188 and mt8195 machine sound drivers. By omitting a dai link subnode in the sound card's node in the Devicetree, the default uninitialized dummy codec is used, and when its dai_name pointer gets passed to strcmp() it results in a null pointer dereference and a kernel panic. In addition to that, set_card_codec_info() in the generic helpers file, mtk-soundcard-driver.c, will populate a dai link with a dummy codec when a dai link node is present in DT but with no codec property. The result is that at probe time, a dummy codec can either be uninitialized with num_codecs = 0, or be an initialized dummy codec, with num_codecs = 1 and dai_name = "snd-soc-dummy-dai". In order to accommodate for both situations, check that num_codecs is not zero before accessing the codecs' fields but still check for the codec's dai name against "snd-soc-dummy-dai" as needed. While at it, also drop the check that dai_name is not null in the mt8192 driver, introduced in commit 4d4e1b6319e5 ("ASoC: mediatek: mt8192: Check existence of dai_name before dereferencing"), as it is actually redundant given the preceding num_codecs != 0 check.
CVE-2024-56687	In the Linux kernel, the following vulnerability has been resolved: usb: musb: Fix hardware lockup on first Rx endpoint request There is a possibility that a request's callback could be invoked from usb_ep_queue() (call trace below, supplemented with missing calls): req->complete from usb_gadget_giveback_request (drivers/usb/gadget/udc/core.c:999) usb_gadget_giveback_request from musb_g_giveback (drivers/usb/musb/musb_gadget.c:147) musb_g_giveback from rxstate (drivers/usb/musb/musb_gadget.c:784) rxstate from musb_ep_restart (drivers/usb/musb/musb_gadget.c:1169) musb_ep_restart from musb_ep_restart_resume_work (drivers/usb/musb/musb_gadget.c:1176) musb_ep_restart_resume_work from musb_queue_resume_work (drivers/usb/musb/musb_core.c:2279) musb_queue_resume_work from musb_gadget_queue (drivers/usb/musb/musb_gadget.c:1241) musb_gadget_queue from usb_ep_queue (drivers/usb/gadget/udc/core.c:300) According to the docstring of usb_ep_queue(), this should not happen: "Note that @req's ->complete() callback must never be called from within usb_ep_queue() as that can create deadlock situations." In fact, a hardware lockup might occur in the following sequence: 1. The gadget is initialized using musb_gadget_enable(). 2. Meanwhile, a packet arrives, and the RXPKTRDY flag is set, raising an interrupt. 3. If IRQs are enabled, the interrupt is handled, but musb_g_rx() finds an empty queue (next_request() returns NULL). The interrupt flag has already been cleared by the glue layer handler, but the RXPKTRDY flag remains set. 4. The first request is enqueued using usb_ep_queue(), leading to the call of req->complete(), as shown in the call trace above. 5. If the callback enables IRQs and another packet is waiting, step (3) repeats. The request queue is empty because usb_g_giveback() removes the request before invoking the callback. 6. The endpoint remains locked up, as the interrupt triggered by hardware setting the RXPKTRDY flag has been handled, but the flag itself remains set. For this scenario to occur, it is only necessary for IRQs to be enabled at some point during the complete callback. This happens with the USB Ethernet gadget, whose rx_complete() callback calls netif_rx(). If called in the task context, netif_rx() disables the bottom halves (BHs). When the BHs are re-enabled, IRQs are also enabled to allow soft IRQs to be processed. The gadget itself is initialized at module load (or at boot if built-in), but the first request is enqueued when the network interface is brought up, triggering rx_complete() in the task context via ioctl(). If a packet arrives while the interface is down, it can prevent the interface from receiving any further packets from the USB host. The situation is quite complicated with many parties involved. This particular issue can be resolved in several possible ways: 1. Ensure that callbacks never enable IRQs. This would be difficult to enforce, as discovering how netif_rx() interacts with interrupts was already quite challenging and u_ether is not the only function driver. Similar "bugs" could be hidden in other drivers as well. 2. Disable MUSB interrupts in musb_g_giveback() before calling the callback and re-enable them afterwards (by calling musb_{dis,en}able_interrupts(), for example). This would ensure that MUSB interrupts are not handled during the callback, even if IRQs are enabled. In fact, it would allow IRQs to be enabled when releasing the lock. However, this feels like an inelegant hack. 3. Modify the interrupt handler to clear the RXPKTRDY flag if the request queue is empty. While this approach also feels like a hack, it wastes CPU time by attempting to handle incoming packets when the software is not ready to process them. 4. Flush the Rx FIFO instead of calling rxstate() in musb_ep_restart(). This ensures that the hardware can receive packets when there is at least one request in the queue. Once I --- truncated---
CVE-2024-56688	In the Linux kernel, the following vulnerability has been resolved: sunrpc: clear XPRT_SOCKET_UPD_TIMEOUT when reset transport Since transport->sock has been set to NULL during reset transport, XPRT_SOCKET_UPD_TIMEOUT also needs to be cleared. Otherwise, the xs_tcp_set_socket_timeouts() may be triggered in xs_tcp_send_request() to dereference the transport->sock that has been set to NULL.

CVE ID	Description
CVE-2024-56689	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: epf-mhi: Avoid NULL dereference if DT lacks 'mmio' If platform_get_resource_byname() fails and returns NULL because DT lacks an 'mmio' property for the MHI endpoint, dereferencing res->start will cause a NULL pointer access. Add a check to prevent it. [kwilczynski: error message update per the review feedback] [bhelgaas: commit log]
CVE-2024-56690	In the Linux kernel, the following vulnerability has been resolved: crypto: pccrypt - Call crypto layer directly when padata_do_parallel() return -EBUSY Since commit 8f4f68e788c3 ("crypto: pccrypt - Fix hungtask for PADATA_RESET"), the pccrypt encryption and decryption operations return -EAGAIN when the CPU goes online or offline. In alg_test(), a WARN is generated when pccrypt_aead_decrypt() or pccrypt_aead_encrypt() returns -EAGAIN, the unnecessary panic will occur when panic_on_warn set 1. Fix this issue by calling crypto layer directly without parallelization in that case.
CVE-2024-56691	In the Linux kernel, the following vulnerability has been resolved: mfd: intel_soc_pmic_bxtwc: Use IRQ domain for USB Type-C device While design wise the idea of converting the driver to use the hierarchy of the IRQ chips is correct, the implementation has (inherited) flaws. This was unveiled when platform_get_irq() had started WARN() on IRQ 0 that is supposed to be a Linux IRQ number (also known as vIRQ). Rework the driver to respect IRQ domain when creating each MFD device separately, as the domain is not the same for all of them.
CVE-2024-56692	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on node blkaddr in truncate_node() syzbot reports a f2fs bug as below: -----[cut here]----- kernel BUG at fs/f2fs/segment.c:2534! RIP: 0010:f2fs_invalidate_blocks+0x35f/0x370 fs/f2fs/segment.c:2534 Call Trace: truncate_node+0x1ae/0x8c0 fs/f2fs/node.c:909 f2fs_remove_inode_page+0x5c2/0x870 fs/f2fs/node.c:1288 f2fs_evict_inode+0x879/0x15c0 fs/f2fs/inode.c:856 evict+0x4e8/0x9b0 fs/inode.c:723 f2fs_handle_failed_inode+0x271/0x2e0 fs/f2fs/inode.c:986 f2fs_create+0x357/0x530 fs/f2fs/namei.c:394 lookup_open fs/namei.c:3595 [inline] open_last_lookups fs/namei.c:3694 [inline] path_openat+0x1c03/0x3590 fs/namei.c:3930 do_filp_open+0x235/0x490 fs/namei.c:3960 do_sys_openat2+0x13e/0x1d0 fs/open.c:1415 do_sys_open fs/open.c:1430 [inline] __do_sys_openat fs/open.c:1446 [inline] __se_sys_openat fs/open.c:1441 [inline] __x64_sys_openat+0x247/0x2a0 fs/open.c:1441 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0010:f2fs_invalidate_blocks+0x35f/0x370 fs/f2fs/segment.c:2534 The root cause is: on a fuzzed image, blkaddr in nat entry may be corrupted, then it will cause system panic when using it in f2fs_invalidate_blocks(), to avoid this, let's add sanity check on nat blkaddr in truncate_node().
CVE-2024-56693	In the Linux kernel, the following vulnerability has been resolved: brd: defer automatic disk creation until module initialization succeeds My colleague Wupeng found the following problems during fault injection: BUG: unable to handle page fault for address: fffffbfff809d073 PGD 6e648067 P4D 123ec8067 PUD 123ec4067 PMD 100e38067 PTE 0 Oops: Oops: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 5 UID: 0 PID: 755 Comm: modprobe Not tainted 6.12.0-rc3+ #17 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 RIP: 0010:__asan_load8+0x4c/0xa0 ... Call Trace: <TASK> blkdev_put_whole+0x41/0x70 bdev_release+0x1a3/0x250 blkdev_release+0x11/0x20 __fput+0x1d7/0x4a0 task_work_run+0xfc/0x180 syscall_exit_to_user_mode+0x1de/0x1f0 do_syscall_64+0x6b/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e loop_init() is calling loop_add() after __register_blkdev() succeeds and is ignoring disk_add() failure from loop_add(), for loop_add() failure is not fatal and successfully created disks are already visible to bdev_open(). brd_init() is currently calling brd_alloc() before __register_blkdev() succeeds and is releasing successfully created disks when brd_init() returns an error. This can cause UAF for the latter two case: case 1: T1: modprobe brd brd_init brd_alloc(0) // success add_disk disk_scan_partitions bdev_file_open_by_dev // alloc file fput // won't free until back to userspace brd_alloc(1) // failed since mem alloc error inject // error path for modprobe will release code segment // back to userspace __fput blkdev_release bdev_release blkdev_put_whole bdev->bd_disk->fops->release // fops is freed now, UAF! case 2: T1: T2: modprobe brd brd_init brd_alloc(0) // success open(/dev/ram0) brd_alloc(1) // fail // error path for modprobe close(/dev/ram0) ... /* UAF! */ bdev->bd_disk->fops->release Fix this problem by following what loop_init() does. Besides, reintroduce brd_devices_mutex to help serialize modifications to brd_list.
CVE-2024-56694	In the Linux kernel, the following vulnerability has been resolved: bpf: fix recursive lock when verdict program return SK_PASS When the stream_verdict program returns SK_PASS, it places the received skb into its own receive queue, but a recursive lock eventually occurs, leading to an operating system deadlock. This issue has been present since v6.9. "" sk_psock_strp_data_ready write_lock_bh(&sk->sk_callback_lock) strp_data_ready strp_read_sock read_sock -> tcp_read_sock strp_rcv cb.rcv_msg -> sk_psock_strp_read # now stream_verdict return SK_PASS without peer sock assign __SK_PASS = sk_psock_map_verd(SK_PASS, NULL) sk_psock_verdict_apply sk_psock_skb_ingress_self sk_psock_skb_ingress_enqueue sk_psock_data_ready read_lock_bh(&sk->sk_callback_lock) <= dead lock "" This topic has been discussed before, but it has not been fixed. Previous discussion: https://lore.kernel.org/all/6684a5864ec86_403d20898@john.notmuch

CVE ID	Description
CVE-2024-56698	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: Fix looping of queued SG entries The dwc3_request->num_queued_sgs is decremented on completion. If a partially completed request is handled, then the dwc3_request->num_queued_sgs no longer reflects the total number of num_queued_sgs (it would be cleared). Correctly check the number of request SG entries remained to be prepare and queued. Failure to do this may cause null pointer dereference when accessing non-existent SG entry.
CVE-2024-56700	In the Linux kernel, the following vulnerability has been resolved: media: wl128x: Fix atomicity violation in fmc_send_cmd() Atomicity violation occurs when the fmc_send_cmd() function is executed simultaneously with the modification of the fmdev->resp_skb value. Consider a scenario where, after passing the validity check within the function, a non-null fmdev->resp_skb variable is assigned a null value. This results in an invalid fmdev->resp_skb variable passing the validity check. As seen in the later part of the function, skb = fmdev->resp_skb; when the invalid fmdev->resp_skb passes the check, a null pointer dereference error may occur at line 478, evt_hdr = (void *)skb->data; To address this issue, it is recommended to include the validity check of fmdev->resp_skb within the locked section of the function. This modification ensures that the value of fmdev->resp_skb does not change during the validation process, thereby maintaining its validity. This possible bug is found by an experimental static analysis tool developed by our team. This tool analyzes the locking APIs to extract function pairs that can be concurrently executed, and then analyzes the instructions in the paired functions to identify possible concurrency bugs including data races and atomicity violations.
CVE-2024-56701	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries: Fix dtl_access_lock to be a rw_semaphore The dtl_access_lock needs to be a rw_semaphore, a sleeping lock, because the code calls kmalloc() while holding it, which can sleep: # echo 1 > /proc/powerpc/vcpudispatch_stats BUG: sleeping function called from invalid context at include/linux/sched/mm.h:337 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 199, name: sh preempt_count: 1, expected: 0 3 locks held by sh/199: #0: c0000000a0743f8 (sb_writers#3){+.+.}-{0:0}, at: vfs_write+0x324/0x438 #1: c0000000028c7058 (dtl_enable_mutex){+.+.}-{3:3}, at: vcpudispatch_stats_write+0xd4/0x5f4 #2: c0000000028c70b8 (dtl_access_lock){+.+.}-{2:2}, at: vcpudispatch_stats_write+0x220/0x5f4 CPU: 0 PID: 199 Comm: sh Not tainted 6.10.0-rc4 #152 Hardware name: IBM pSeries (emulated by qemu) POWER9 (raw) 0x4e1202 0xf000005 of:SLOF,HEAD hv:linux,kvm pSeries Call Trace: dump_stack_lvl+0x130/0x148 (unreliable) __might_resched+0x174/0x410 kmem_cache_alloc_noprof+0x340/0x3d0 alloc_dtl_buffers+0x124/0x1ac vcpudispatch_stats_write+0x2a8/0x5f4 proc_reg_write+0xf4/0x150 vfs_write+0xfc/0x438 ksys_write+0x88/0x148 system_call_exception+0x1c4/0x5a0 system_call_common+0xf4/0x258
CVE-2024-56702	In the Linux kernel, the following vulnerability has been resolved: bpf: Mark raw_tp arguments with PTR_MAYBE_NULL Arguments to a raw tracepoint are tagged as trusted, which carries the semantics that the pointer will be non-NULL. However, in certain cases, a raw tracepoint argument may end up being NULL. More context about this issue is available in [0]. Thus, there is a discrepancy between the reality, that raw_tp arguments can actually be NULL, and the verifier's knowledge, that they are never NULL, causing explicit NULL checks to be deleted, and accesses to such pointers potentially crashing the kernel. To fix this, mark raw_tp arguments as PTR_MAYBE_NULL, and then special case the dereference and pointer arithmetic to permit it, and allow passing them into helpers/kfuncs; these exceptions are made for raw_tp programs only. Ensure that we don't do this when ref_obj_id > 0, as in that case this is an acquired object and doesn't need such adjustment. The reason we do mask_raw_tp_trusted_reg logic is because other will recheck in places whether the register is a trusted_reg, and then consider our register as untrusted when detecting the presence of the PTR_MAYBE_NULL flag. To allow safe dereference, we enable PROBE_MEM marking when we see loads into trusted pointers with PTR_MAYBE_NULL. While trusted raw_tp arguments can also be passed into helpers or kfuncs where such broken assumption may cause issues, a future patch set will tackle their case separately, as PTR_TO_BTF_ID (without PTR_TRUSTED) can already be passed into helpers and causes similar problems. Thus, they are left alone for now. It is possible that these checks also permit passing non-raw_tp args that are trusted PTR_TO_BTF_ID with null marking. In such a case, allowing dereference when pointer is NULL expands allowed behavior, so won't regress existing programs, and the case of passing these into helpers is the same as above and will be dealt with later. Also update the failure case in tp_btf_nullable selftest to capture the new behavior, as the verifier will no longer cause an error when directly dereference a raw tracepoint argument marked as __nullable. [0]: https://lore.kernel.org/bpf/ZrCZS6nirsaEqehw@jlelli-thinkpadt14gen4.remote.csb

CVE ID	Description
CVE-2024-56703	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix soft lockups in fib6_select_path under high next hop churn Soft lockups have been observed on a cluster of Linux-based edge routers located in a highly dynamic environment. Using the 'bird' service, these routers continuously update BGP-advertised routes due to frequently changing nexthop destinations, while also managing significant IPv6 traffic. The lockups occur during the traversal of the multipath circular linked-list in the 'fib6_select_path' function, particularly while iterating through the siblings in the list. The issue typically arises when the nodes of the linked list are unexpectedly deleted concurrently on a different core—indicated by their 'next' and 'previous' elements pointing back to the node itself and their reference count dropping to zero. This results in an infinite loop, leading to a soft lockup that triggers a system panic via the watchdog timer. Apply RCU primitives in the problematic code sections to resolve the issue. Where necessary, update the references to fib6_siblings to annotate or use the RCU APIs. Include a test script that reproduces the issue. The script periodically updates the routing table while generating a heavy load of outgoing IPv6 traffic through multiple iperf3 clients. It consistently induces infinite soft lockups within a couple of minutes. Kernel log: 0 [ffffbd13003e8d30] machine_kexec at ffffffff8ceaf3eb 1 [ffffbd13003e8d90] __crash_kexec at ffffffff8d0120e3 2 [ffffbd13003e8e58] panic at ffffffff8cef65d4 3 [ffffbd13003e8ed8] watchdog_timer_fn at ffffffff8d05cb03 4 [ffffbd13003e8f08] __hrtimer_run_queues at ffffffff8cfec62f 5 [ffffbd13003e8f70] hrtimer_interrupt at ffffffff8cfed756 6 [ffffbd13003e8fd0] __sysvec_apic_timer_interrupt at ffffffff8cea01af 7 [ffffbd13003e8ff0] sysvec_apic_timer_interrupt at ffffffff8df1b83d -- <IRQ stack> -- 8 [ffffbd13003d3708] asm_sysvec_apic_timer_interrupt at ffffffff8e000ecb [exception RIP: fib6_select_path+299] RIP: ffffffff8ddafe7b RSP: fffffbd13003d37b8 RFLAGS: 00000287 RAX: ffff975850b43600 RBX: ffff975850b40200 RCX: 0000000000000000 RDX: 000000003fffffff RSI: 0000000051d383e4 RDI: ffff975850b43618 RBP: fffffbd13003d3800 R8: 0000000000000000 R9: ffff975850b40200 R10: 0000000000000000 R11: 0000000000000000 R12: fffffbd13003d3830 R13: ffff975850b436a8 R14: ffff975850b43600 R15: 0000000000000007 ORIG_RAX: ffffffff8fffffff CS: 0010 SS: 0018 9 [ffffbd13003d3808] ip6_pol_route at ffffffff8ddb030c 10 [ffffbd13003d3888] ip6_pol_route_input at ffffffff8ddb068c 11 [ffffbd13003d3898] fib6_rule_lookup at ffffffff8ddf02b5 12 [ffffbd13003d3928] ip6_route_input at ffffffff8ddb0f47 13 [ffffbd13003d3a18] ip6_rcv_finish_core.constprop.0 at ffffffff8dd950d0 14 [ffffbd13003d3a30] ip6_list_rcv_finish.constprop.0 at ffffffff8dd96274 15 [ffffbd13003d3a98] ip6_sublist_rcv at ffffffff8dd96474 16 [ffffbd13003d3af8] ipv6_list_rcv at ffffffff8dd96615 17 [ffffbd13003d3b60] __netif_receive_skb_list_core at ffffffff8dc16fec 18 [ffffbd13003d3be0] netif_receive_skb_list_internal at ffffffff8dc176b3 19 [ffffbd13003d3c50] napi_gro_receive at ffffffff8dc565b9 20 [ffffbd13003d3c80] ice_receive_skb at ffffffff8c087e4f5 [ice] 21 [ffffbd13003d3c90] ice_clean_rx_irq at ffffffff8c0881b80 [ice] 22 [ffffbd13003d3d20] ice_napi_poll at ffffffff8c088232f [ice] 23 [ffffbd13003d3d80] __napi_poll at ffffffff8dc18000 24 [ffffbd13003d3db8] net_rx_action at ffffffff8dc18581 25 [ffffbd13003d3e40] __do_softirq at ffffffff8df352e9 26 [ffffbd13003d3eb0] run_ksoftirqd at ffffffff8cfe47 27 [ffffbd13003d3ec0] smpboot_thread_fn at ffffffff8cf36a30 28 [ffffbd13003d3ee8] kthread at ffffffff8cf2b39f 29 [ffffbd13003d3f28] ret_from_fork at ffffffff8ce5fa64 30 [ffffbd13003d3f50] ret_from_fork_asm at ffffffff8ce03cbb
CVE-2024-56704	In the Linux kernel, the following vulnerability has been resolved: 9p/xen: fix release of IRQ Kernel logs indicate an IRQ was double-freed. Pass correct device ID during IRQ release. [Dominique: remove confusing variable reset to 0]
CVE-2024-56705	In the Linux kernel, the following vulnerability has been resolved: media: atomisp: Add check for rgby_data memory allocation failure In ia_css_3a_statistics_allocate(), there is no check on the allocation result of the rgby_data memory. If rgby_data is not successfully allocated, it may trigger the assert(host_stats->rgby_data) assertion in ia_css_s3a_hmem_decode(). Adding a check to fix this potential issue.
CVE-2024-56707	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in otx2_dmacflt.c Add error pointer checks after calling otx2_mbox_get_rsp().
CVE-2024-56708	In the Linux kernel, the following vulnerability has been resolved: EDAC/igen6: Avoid segmentation fault on module unload The segmentation fault happens because: During modprobe: 1. In igen6_probe(), igen6_pvt will be allocated with kzalloc() 2. In igen6_register_mci(), mci->pvt_info will point to &igen6_pvt->imc[mc] During rmmmod: 1. In mci_release() in edac_mc.c, it will kfree(mci->pvt_info) 2. In igen6_remove(), it will kfree(igen6_pvt); Fix this issue by setting mci->pvt_info to NULL to avoid the double kfree.
CVE-2024-56709	In the Linux kernel, the following vulnerability has been resolved: io_uring: check if iowq is killed before queuing task work can be executed after the task has gone through io_uring termination, whether it's the final task_work run or the fallback path. In this case, task work will find ->io_wq being already killed and null'ed, which is a problem if it then tries to forward the request to io_queue_iowq(). Make io_queue_iowq() fail requests in this case. Note that it also checks PF_KTHREAD, because the user can first close a DEFER_TASKRUN ring and shortly after kill the task, in which case ->iowq check would race.

CVE ID	Description
CVE-2024-56710	In the Linux kernel, the following vulnerability has been resolved: ceph: fix memory leak in ceph_direct_read_write() The bvecs array which is allocated in iter_get_bvecs_alloc() is leaked and pages remain pinned if ceph_alloc_sparse_ext_map() fails. There is no need to delay the allocation of sparse_ext map until after the bvecs array is set up, so fix this by moving sparse_ext allocation a bit earlier. Also, make a similar adjustment in __ceph_sync_read() for consistency (a leak of the same kind in __ceph_sync_read() has been addressed differently).
CVE-2024-56712	In the Linux kernel, the following vulnerability has been resolved: udmabuf: fix memory leak on last export_udmabuf() error path In export_udmabuf(), if dma_buf_fd() fails because the FD table is full, a dma_buf owning the udmabuf has already been created; but the error handling in udmabuf_create() will tear down the udmabuf without doing anything about the containing dma_buf. This leaves a dma_buf in memory that contains a dangling pointer; though that doesn't seem to lead to anything bad except a memory leak. Fix it by moving the dma_buf_fd() call out of export_udmabuf() so that we can give it different error handling. Note that the shape of this code changed a lot in commit 5e72b2b41a21 ("udmabuf: convert udmabuf driver to use folios"); but the memory leak seems to have existed since the introduction of udmabuf.
CVE-2024-56715	In the Linux kernel, the following vulnerability has been resolved: ionic: Fix netdev notifier unregister on failure If register_netdev() fails, then the driver leaks the netdev notifier. Fix this by calling ionic_lif_unregister() on register_netdev() failure. This will also call ionic_lif_unregister_phc() if it has already been registered.
CVE-2024-56716	In the Linux kernel, the following vulnerability has been resolved: netdevsim: prevent bad user input in nsim_dev_health_break_write() If either a zero count or a large one is provided, kernel can crash.
CVE-2024-56717	In the Linux kernel, the following vulnerability has been resolved: net: msc: ocelot: fix incorrect IFH SRC_PORT field in ocelot_ifh_set_basic() Packets injected by the CPU should have a SRC_PORT field equal to the CPU port module index in the Analyzer block (ocelot->num_phys_ports). The blamed commit copied the ocelot_ifh_set_basic() call incorrectly from ocelot_xmit_common() in net/dsa/tag_ocelot.c. Instead of calling with "x", it calls with BIT_ULL(x), but the field is not a port mask, but rather a single port index. [side note: this is the technical debt of code duplication : () The error used to be silent and doesn't appear to have other user-visible manifestations, but with new changes in the packing library, it now fails loudly as follows: -----[cut here]----- Cannot store 0x40 inside bits 46-43 - will truncate sja1105 spi2.0: xmit timed out WARNING: CPU: 1 PID: 102 at lib/packing.c:98 __pack+0x90/0x198 sja1105 spi2.0: timed out polling for tstamp CPU: 1 UID: 0 PID: 102 Comm: felix_xmit Tainted: G W N 6.13.0-rc1-00372-gf706b85d972d-dirty #2605 Call trace: __pack+0x90/0x198 (P) __pack+0x90/0x198 (L) packing+0x78/0x98 ocelot_ifh_set_basic+0x260/0x368 ocelot_port_inject_frame+0xa8/0x250 felix_port_deferred_xmit+0x14c/0x258 kthread_worker_fn+0x134/0x350 kthread+0x114/0x138 The code path pertains to the ocelot switchdev driver and to the felix secondary DSA tag protocol, ocelot-8021q. Here seen with ocelot-8021q. The messenger (packing) is not really to blame, so fix the original commit instead.
CVE-2024-56718	In the Linux kernel, the following vulnerability has been resolved: net/smc: protect link down work from execute after lgr freed link down work may be scheduled before lgr freed but execute after lgr freed, which may result in crash. So it is need to hold a reference before schedule link down work, and put the reference after work executed or canceled. The relevant crash call stack as follows: list_del corruption. prev->next should be ffff638c9c0fe20, but was 0000000000000000 ----- [cut here]----- kernel BUG at lib/list_debug.c:51! invalid opcode: 0000 [#1] SMP NOPTI CPU: 6 PID: 978112 Comm: kworker/6:119 Kdump: loaded Tainted: G #1 Hardware name: Alibaba Cloud Alibaba Cloud ECS, BIOS 2221b89 04/01/2014 Workqueue: events smc_link_down_work [smc] RIP: 0010: __list_del_entry_valid.cold+0x31/0x47 RSP: 0018:ffff638c9c0fdd8 EFLAGS: 00010086 RAX: 0000000000000054 RBX: ffff942fb75e5128 RCX: 0000000000000000 RDX: ffff943520930aa0 RSI: ffff94352091fc80 RDI: ffff94352091fc80 RBP: 0000000000000000 R08: 0000000000000000 R09: ffff638c9c0fc38 R10: ffff638c9c0fc30 R11: ffffffff015eb28 R12: 0000000000000002 R13: ffff638c9c0fe20 R14: 0000000000000001 R15: ffff942f9cd051c0 FS: 0000000000000000(0000) GS:ffff943520900000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f4f25214000 CR3: 000000025fbae004 CR4: 00000000007706e0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: rwsem_down_write_slowpath+0x17e/0x470 smc_link_down_work+0x3c/0x60 [smc] process_one_work+0x1ac/0x350 worker_thread+0x49/0x2f0 ? rescuer_thread+0x360/0x360 kthread+0x118/0x140 ? __kthread_bind_mask+0x60/0x60 ret_from_fork+0x1f/0x30
CVE-2024-56720	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Several fixes to bpf_msg_pop_data Several fixes to bpf_msg_pop_data, 1. In sk_msg_shift_left, we should put_page 2. if (len == 0), return early is better 3. pop the entire sk_msg (last == msg->sg.size) should be supported 4. Fix for the value of variable "a" 5. In sk_msg_shift_left, after shifting, i has already pointed to the next element. Additional sk_msg_iter_var_next may result in BUG.

CVE ID	Description
CVE-2024-56721	In the Linux kernel, the following vulnerability has been resolved: x86/CPU/AMD: Terminate the erratum_1386_microcode array The erratum_1386_microcode array requires an empty entry at the end. Otherwise x86_match_cpu_with_stepping() will continue iterate the array after it ended. Add an empty entry to erratum_1386_microcode to its end.
CVE-2024-56722	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix cpu stuck caused by printings during reset During reset, cmd to destroy resources such as qp, cq, and mr may fail, and error logs will be printed. When a large number of resources are destroyed, there will be lots of printings, and it may lead to a cpu stuck. Delete some unnecessary printings and replace other printing functions in these paths with the ratelimited version.
CVE-2024-56723	In the Linux kernel, the following vulnerability has been resolved: mfd: intel_soc_pmic_bxtwc: Use IRQ domain for PMIC devices While design wise the idea of converting the driver to use the hierarchy of the IRQ chips is correct, the implementation has (inherited) flaws. This was unveiled when platform_get_irq() had started WARN() on IRQ 0 that is supposed to be a Linux IRQ number (also known as vIRQ). Rework the driver to respect IRQ domain when creating each MFD device separately, as the domain is not the same for all of them.
CVE-2024-56724	In the Linux kernel, the following vulnerability has been resolved: mfd: intel_soc_pmic_bxtwc: Use IRQ domain for TMU device While design wise the idea of converting the driver to use the hierarchy of the IRQ chips is correct, the implementation has (inherited) flaws. This was unveiled when platform_get_irq() had started WARN() on IRQ 0 that is supposed to be a Linux IRQ number (also known as vIRQ). Rework the driver to respect IRQ domain when creating each MFD device separately, as the domain is not the same for all of them.
CVE-2024-56725	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in otx2_dcbnl.c Add error pointer check after calling otx2_mbox_get_rsp().
CVE-2024-56726	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in cn10k.c Add error pointer check after calling otx2_mbox_get_rsp().
CVE-2024-56727	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in otx2_flows.c Adding error pointer check after calling otx2_mbox_get_rsp().
CVE-2024-56728	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: handle otx2_mbox_get_rsp errors in otx2_ethtool.c Add error pointer check after calling otx2_mbox_get_rsp().
CVE-2024-56729	In the Linux kernel, the following vulnerability has been resolved: smb: Initialize cfid->tcon before performing network ops Avoid leaking a tcon ref when a lease break races with opening the cached directory. Processing the leak break might take a reference to the tcon in cached_dir_lease_break() and then fail to release the ref in cached_dir_offload_close, since cfid->tcon is still NULL.
CVE-2024-56739	In the Linux kernel, the following vulnerability has been resolved: rtc: check if __rtc_read_time was successful in rtc_timer_do_work() If the __rtc_read_time call fails,, the struct rtc_time tm; may contain uninitialized data, or an illegal date/time read from the RTC hardware. When calling rtc_tm_to_ktime later, the result may be a very large value (possibly KTIME_MAX). If there are periodic timers in rtc->timerqueue, they will continually expire, may causing kernel softlockup.
CVE-2024-56742	In the Linux kernel, the following vulnerability has been resolved: vfio/mlx5: Fix an unwind issue in mlx5vf_add_migration_pages() Fix an unwind issue in mlx5vf_add_migration_pages(). If a set of pages is allocated but fails to be added to the SG table, they need to be freed to prevent a memory leak. Any pages successfully added to the SG table will be freed as part of mlx5vf_free_data_buffer().

CVE ID	Description
CVE-2024-56744	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid potential deadlock in f2fs_record_stop_reason() syzbot reports deadlock issue of f2fs as below: <pre>===== WARNING: possible circular locking dependency detected 6.12.0-rc3-syzkaller-00087-gc964ced77262 #0 Not tainted ----- kswapd0/79 is trying to acquire lock: ffff888011824088 (&sbi->sb_lock){++++}-{3:3}, at: f2fs_down_write fs/f2fs/f2fs.h:2199 [inline] ffff888011824088 (&sbi->sb_lock){++++}-{3:3}, at: f2fs_record_stop_reason+0x52/0x1d0 fs/ f2fs/super.c:4068 but task is already holding lock: ffff88804bd92610 (sb_internal#2){.+.+}- {0:0}, at: f2fs_evict_inode+0x662/0x15c0 fs/f2fs/inode.c:842 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #2 (sb_internal#2){.+.+}- {0:0}: lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5825 percpu_down_read include/linux/ percpu-rwsem.h:51 [inline] __sb_start_write include/linux/fs.h:1716 [inline] sb_start_intwrite +0x4d/0x1c0 include/linux/fs.h:1899 f2fs_evict_inode+0x662/0x15c0 fs/f2fs/inode.c:842 evict +0x4e8/0x9b0 fs/inode.c:725 f2fs_evict_inode+0x1a4/0x15c0 fs/f2fs/inode.c:807 evict+0x4e8/0x9b0 fs/inode.c:725 dispose_list fs/inode.c:774 [inline] prune_icache_sb+0x239/0x2f0 fs/inode.c:963 super_cache_scan+0x38c/0x4b0 fs/super.c:223 do_shrink_slab+0x701/0x1160 mm/shrinker.c:435 shrink_slab+0x1093/0x14d0 mm/shrinker.c:662 shrink_one+0x43b/0x850 mm/vmscan.c:4818 shrink_many mm/vmscan.c:4879 [inline] lru_gen_shrink_node mm/vmscan.c:4957 [inline] shrink_node+0x3799/0x3de0 mm/vmscan.c:5937 kswapd_shrink_node mm/vmscan.c:6765 [inline] balance_pgdat mm/vmscan.c:6957 [inline] kswapd+0x1ca3/0x3700 mm/vmscan.c:7226 kthread +0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 -> #1 (fs_reclaim){+.+.}-{0:0}: lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5825 __fs_reclaim_acquire mm/page_alloc.c:3834 [inline] fs_reclaim_acquire+0x88/0x130 mm/page_alloc.c:3848 might_alloc include/linux/sched/ mm.h:318 [inline] prepare_alloc_pages+0x147/0x5b0 mm/page_alloc.c:4493 __alloc_pages_noprof +0x16f/0x710 mm/page_alloc.c:4722 alloc_pages_mpol_noprof+0x3e8/0x680 mm/mempolicy.c:2265 alloc_pages_noprof mm/mempolicy.c:2345 [inline] folio_alloc_noprof+0x128/0x180 mm/ mempolicy.c:2352 filemap_alloc_folio_noprof+0xdf/0x500 mm/filemap.c:1010 do_read_cache_folio +0x2eb/0x850 mm/filemap.c:3787 read_mapping_folio include/linux/pagemap.h:1011 [inline] f2fs_commit_super+0x3c0/0x7d0 fs/f2fs/super.c:4032 f2fs_record_stop_reason+0x13b/0x1d0 fs/ f2fs/super.c:4079 f2fs_handle_critical_error+0x2ac/0x5c0 fs/f2fs/super.c:4174 f2fs_write_inode +0x35f/0x4d0 fs/f2fs/inode.c:785 write_inode fs/fs-writeback.c:1503 [inline] __writeback_single_inode +0x711/0x10d0 fs/fs-writeback.c:1723 writeback_single_inode+0x1f3/0x660 fs/fs-writeback.c:1779 sync_inode_metadata+0xc4/0x120 fs/fs-writeback.c:2849 f2fs_release_file+0xa8/0x100 fs/ f2fs/file.c:1941 __fput+0x23f/0x880 fs/file_table.c:431 task_work_run+0x24f/0x310 kernel/ task_work.c:228 resume_user_mode_work include/linux/resume_user_mode.h:50 [inline] exit_to_user_mode_loop kernel/entry/common.c:114 [inline] exit_to_user_mode_prepare include/linux/ entry-common.h:328 [inline] __syscall_exit_to_user_mode_work kernel/entry/common.c:207 [inline] syscall_exit_to_user_mode+0x168/0x370 kernel/entry/common.c:218 do_syscall_64+0x100/0x230 arch/ x86/entry/common.c:89 entry_SYSCALL_64_after_hwframe+0x77/0x7f ---truncated---</pre>
CVE-2024-56745	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix reset_method_store() memory leak In reset_method_store(), a string is allocated via kstrndup() and assigned to the local "options". options is then used in with strsep() to find spaces: while ((name = strsep(&options, " ")) != NULL) { If there are no remaining spaces, then options is set to NULL by strsep(), so the subsequent kfree(options) doesn't free the memory allocated via kstrndup(). Fix by using a separate tmp_options to iterate with strsep() so options is preserved.
CVE-2024-56746	In the Linux kernel, the following vulnerability has been resolved: fbdev: sh7760fb: Fix a possible memory leak in sh7760fb_alloc_mem() When information such as info->screen_base is not ready, calling sh7760fb_free_mem() does not release memory correctly. Call dma_free_coherent() instead.
CVE-2024-56747	In the Linux kernel, the following vulnerability has been resolved: scsi: qedi: Fix a possible memory leak in qedi_alloc_and_init_sb() Hook "qedi_ops->common->sb_init = qed_sb_init" does not release the DMA memory sb_virt when it fails. Add dma_free_coherent() to free it. This is the same way as qedr_alloc_mem_sb() and qede_alloc_mem_sb().
CVE-2024-56748	In the Linux kernel, the following vulnerability has been resolved: scsi: qedf: Fix a possible memory leak in qedf_alloc_and_init_sb() Hook "qedf_ops->common->sb_init = qed_sb_init" does not release the DMA memory sb_virt when it fails. Add dma_free_coherent() to free it. This is the same way as qedr_alloc_mem_sb() and qede_alloc_mem_sb().

CVE ID	Description
CVE-2024-56751	In the Linux kernel, the following vulnerability has been resolved: ipv6: release nexthop on device removal The CI is hitting some aperiodic hangup at device removal time in the pmtu.sh self-test: unregister_netdevice: waiting for veth_A-R1 to become free. Usage count = 6 ref_tracker: veth_A-R1@ffff888013df15d8 has 1/5 users at dst_init+0x84/0x4a0 dst_alloc+0x97/0x150 ip6_dst_alloc+0x23/0x90 ip6_rt_pcpu_alloc+0x1e6/0x520 ip6_pol_route+0x56f/0x840 fib6_rule_lookup+0x334/0x630 ip6_route_output_flags+0x259/0x480 ip6_dst_lookup_tail.constprop.0+0x5c2/0x940 ip6_dst_lookup_flow+0x88/0x190 udp_tunnel6_dst_lookup+0x2a7/0x4c0 vxlan_xmit_one+0xbde/0x4a50 [vxlan] vxlan_xmit+0x9ad/0xf20 [vxlan] dev_hard_start_xmit+0x10e/0x360 __dev_queue_xmit+0xf95/0x18c0 arp_solicit+0x4a2/0xe00 neigh_probe+0xaa/0xf0 While the first suspect is the dst_cache, explicitly tracking the dst owing the last device reference via probes proved such dst is held by the nexthop in the originating fib6_info. Similar to commit f5b51fe804ec ("ipv6: route: purge exception on removal"), we need to explicitly release the originating fib info when disconnecting a to-be-removed device from a live ipv6 dst: move the fib6_info cleanup into ip6_dst_ifdown(). Tested running: ./pmtu.sh cleanup_ipv6_exception in a tight loop for more than 400 iterations with no spat, running an unpatched kernel I observed a splat every ~10 iterations.
CVE-2024-56752	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau/gr/gf100: Fix missing unlock in gf100_gr_chan_new() When the call to gf100_grctx_generate() fails, unlock gr->fecs.mutex before returning the error. Fixes smatch warning: drivers/gpu/drm/nouveau/nvkm/engine/gr/gf100.c:480 gf100_gr_chan_new() warn: inconsistent returns '&gr->fecs.mutex'.
CVE-2024-56754	In the Linux kernel, the following vulnerability has been resolved: crypto: caam - Fix the pointer passed to caam_qi_shutdown() The type of the last parameter given to devm_add_action_or_reset() is "struct caam_drv_private *", but in caam_qi_shutdown(), it is casted to "struct device *". Pass the correct parameter to devm_add_action_or_reset() so that the resources are released as expected.
CVE-2024-56755	In the Linux kernel, the following vulnerability has been resolved: netfs/fscache: Add a memory barrier for FSCACHE_VOLUME_CREATING In fscache_create_volume(), there is a missing memory barrier between the bit-clearing operation and the wake-up operation. This may cause a situation where, after a wake-up, the bit-clearing operation hasn't been detected yet, leading to an indefinite wait. The triggering process is as follows: [cookie1] [cookie2] [volume_work] fscache_perform_lookup fscache_create_volume fscache_perform_lookup fscache_create_volume fscache_create_volume_work cachefiles_acquire_volume clear_and_wake_up_bit test_and_set_bit test_and_set_bit goto maybe_wait goto no_wait In the above process, cookie1 and cookie2 has the same volume. When cookie1 enters the -no_wait- process, it will clear the bit and wake up the waiting process. If a barrier is missing, it may cause cookie2 to remain in the -wait- process indefinitely. In commit 3288666c7256 ("fscache: Use clear_and_wake_up_bit() in fscache_create_volume_work()"), barriers were added to similar operations in fscache_create_volume_work(), but fscache_create_volume() was missed. By combining the clear and wake operations into clear_and_wake_up_bit() to fix this issue.
CVE-2024-56756	In the Linux kernel, the following vulnerability has been resolved: nvme-pci: fix freeing of the HMB descriptor table The HMB descriptor table is sized to the maximum number of descriptors that could be used for a given device, but __nvme_alloc_host_mem could break out of the loop earlier on memory allocation failure and end up using less descriptors than planned for, which leads to an incorrect size passed to dma_free_coherent. In practice this was not showing up because the number of descriptors tends to be low and the dma coherent allocator always allocates and frees at least a page.
CVE-2024-56757	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btusb: mediatek: add intf release flow when usb disconnect MediaTek claim an special usb intr interface for ISO data transmission. The interface need to be released before unregistering hci device when usb disconnect. Removing BT usb dongle without properly releasing the interface may cause Kernel panic while unregister hci device.

CVE ID	Description
CVE-2024-56758	In the Linux kernel, the following vulnerability has been resolved: btrfs: check folio mapping after unlock in relocate_one_folio() When we call btrfs_read_folio() to bring a folio uptodate, we unlock the folio. The result of that is that a different thread can modify the mapping (like remove it with invalidate) before we call folio_lock(). This results in an invalid page and we need to try again. In particular, if we are relocating concurrently with aborting a transaction, this can result in a crash like the following: BUG: kernel NULL pointer dereference, address: 0000000000000000 PGD 0 P4D 0 Oops: 0000 [#1] SMP CPU: 76 PID: 1411631 Comm: kworker/u322:5 Workqueue: events_unbound btrfs_reclaim_bgs_work RIP: 0010:set_page_extent_mapped+0x20/0xb0 RSP: 0018:ffffc900516a7be8 EFLAGS: 00010246 RAX: ffffea009e851d08 RBX: ffffea009e0b1880 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffffc900516a7b90 RDI: ffffea009e0b1880 RBP: 0000000003573000 R08: 0000000000000001 R09: ffff88c07fd2f3f0 R10: 0000000000000000 R11: 0000194754b575be R12: 0000000003572000 R13: 0000000003572fff R14: 000000000100cca R15: 0000000005582fff FS: 0000000000000000(0000) GS:ffff88c07fd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 000000407d00f002 CR4: 00000000007706f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 PKRU: 55555554 Call Trace: <TASK> ? __die+0x78/0xc0 ? page_fault_oops+0x2a8/0x3a0 ? __switch_to+0x133/0x530 ? wq_worker_running+0xa/0x40 ? exc_page_fault+0x63/0x130 ? asm_exc_page_fault+0x22/0x30 ? set_page_extent_mapped+0x20/0xb0 relocate_file_extent_cluster+0x1a7/0x940 relocate_data_extent+0xaf/0x120 relocate_block_group+0x20f/0x480 btrfs_relocate_block_group+0x152/0x320 btrfs_relocate_chunk+0x3d/0x120 btrfs_reclaim_bgs_work+0x2ae/0x4e0 process_scheduled_works+0x184/0x370 worker_thread+0xc6/0x3e0 ? blk_add_timer+0xb0/0xb0 kthread+0xae/0xe0 ? flush_tlb_kernel_range+0x90/0x90 ret_from_fork+0x2f/0x40 ? flush_tlb_kernel_range+0x90/0x90 ret_from_fork_asm+0x11/0x20 </TASK> This occurs because cleanup_one_transaction() calls destroy_delalloc_inodes() which calls invalidate_inode_pages2() which takes the folio_lock before setting mapping to NULL. We fail to check this, and subsequently call set_extent_mapping(), which assumes that mapping != NULL (in fact it asserts that in debug mode) Note that the "fixes" patch here is not the one that introduced the race (the very first iteration of this code from 2009) but a more recent change that made this particular crash happen in practice.
CVE-2024-56759	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free when COWing tree block and tracing is enabled When a COWing a tree block, at btrfs_cow_block(), and we have the tracepoint trace_btrfs_cow_block() enabled and preemption is also enabled (CONFIG_PREEMPT=y), we can trigger a use-after-free in the COWed extent buffer while inside the tracepoint code. This is because in some paths that call btrfs_cow_block(), such as btrfs_search_slot(), we are holding the last reference on the extent buffer @buf so btrfs_force_cow_block() drops the last reference on the @buf extent buffer when it calls free_extent_buffer_stale(buf), which schedules the release of the extent buffer with RCU. This means that if we are on a kernel with preemption, the current task may be preempted before calling trace_btrfs_cow_block() and the extent buffer already released by the time trace_btrfs_cow_block() is called, resulting in a use-after-free. Fix this by moving the trace_btrfs_cow_block() from btrfs_cow_block() to btrfs_force_cow_block() before the COWed extent buffer is freed. This also has a side effect of invoking the tracepoint in the tree defrag code, at defrag.c:btrfs_realloc_node(), since btrfs_force_cow_block() is called there, but this is fine and it was actually missing there.
CVE-2024-56760	In the Linux kernel, the following vulnerability has been resolved: PCI/MSI: Handle lack of irqdomain gracefully Alexandre observed a warning emitted from pci_msi_setup_msi_irqs() on a RISCv platform which does not provide PCI/MSI support: WARNING: CPU: 1 PID: 1 at drivers/pci/msi/msi.h:121 pci_msi_setup_msi_irqs+0x2c/0x32 __pci_enable_msix_range+0x30c/0x596 pci_msi_setup_msi_irqs+0x2c/0x32 pci_alloc_irq_vectors_affinity+0xb8/0xe2 RISCv uses hierarchical interrupt domains and correctly does not implement the legacy fallback. The warning triggers from the legacy fallback stub. That warning is bogus as the PCI/MSI layer knows whether a PCI/MSI parent domain is associated with the device or not. There is a check for MSI-X, which has a legacy assumption. But that legacy fallback assumption is only valid when legacy support is enabled, but otherwise the check should simply return -ENOTSUPP. Loongarch tripped over the same problem and blindly enabled legacy support without implementing the legacy fallbacks. There are weak implementations which return an error, so the problem was papered over. Correct pci_msi_domain_supports() to evaluate the legacy mode and add the missing supported check into the MSI enable path to complete it.

CVE ID	Description
CVE-2024-56761	In the Linux kernel, the following vulnerability has been resolved: x86/fred: Clear WFE in missing-ENDBRANCH #CPs An indirect branch instruction sets the CPU indirect branch tracker (IBT) into WAIT_FOR_ENDBRANCH (WFE) state and WFE stays asserted across the instruction boundary. When the decoder finds an inappropriate instruction while WFE is set ENDBR, the CPU raises a #CP fault. For the "kernel IBT no ENDBR" selftest where #CPs are deliberately triggered, the WFE state of the interrupted context needs to be cleared to let execution continue. Otherwise when the CPU resumes from the instruction that just caused the previous #CP, another missing-ENDBRANCH #CP is raised and the CPU enters a dead loop. This is not a problem with IDT because it doesn't preserve WFE and IRET doesn't set WFE. But FRED provides space on the entry stack (in an expanded CS area) to save and restore the WFE state, thus the WFE state is no longer clobbered, so software must clear it. Clear WFE to avoid dead looping in ibt_clear_fred_wfe() and the !ibt_fatal code path when execution is allowed to continue. Clobbering WFE in any other circumstance is a security-relevant bug. [dhansen: changelog rewording]
CVE-2024-56763	In the Linux kernel, the following vulnerability has been resolved: tracing: Prevent bad count for tracing_cpumask_write If a large count is provided, it will trigger a warning in bitmap_parse_user. Also check zero for it.
CVE-2024-56764	In the Linux kernel, the following vulnerability has been resolved: ublk: detach gendisk from ublk device if add_disk() fails Inside ublk_abort_requests(), gendisk is grabbed for aborting all inflight requests. And ublk_abort_requests() is called when exiting the uring context or handling timeout. If add_disk() fails, the gendisk may have been freed when calling ublk_abort_requests(), so use-after-free can be caused when getting disk's reference in ublk_abort_requests(). Fixes the bug by detaching gendisk from ublk device if add_disk() fails.
CVE-2024-56765	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries/vas: Add close() callback in vas_vm_ops struct The mapping VMA address is saved in VAS window struct when the paste address is mapped. This VMA address is used during migration to unmap the paste address if the window is active. The paste address mapping will be removed when the window is closed or with the munmap(). But the VMA address in the VAS window is not updated with munmap() which is causing invalid access during migration. The KASAN report shows: [16386.254991] BUG: KASAN: slab-use-after-free in reconfig_close_windows+0x1a0/0x4e8 [16386.255043] Read of size 8 at addr c00000014a819670 by task drmgr/696928 [16386.255096] CPU: 29 UID: 0 PID: 696928 Comm: drmgr Kdump: loaded Tainted: G B 6.11.0-rc5-nxgzip #2 [16386.255128] Tainted: [B]=BAD_PAGE [16386.255148] Hardware name: IBM,9080-HEX Power11 (architected) 0x820200 0xf000007 of:IBM,FW1110.00 (NH1110_016) hv:phyp pSeries [16386.255181] Call Trace: [16386.255202] [c00000016b297660] [c0000000018ad0ac] dump_stack_lvl+0x84/0xe8 [16386.255246] [c00000016b297690] [c0000000006e8a90] print_report+0x19c/0x764 [16386.255285] [c00000016b297760] [c0000000006e9490] kasan_report+0x128/0x1f8 [16386.255309] [c00000016b297880] [c0000000006eb5c8] __asan_load8+0xac/0xe0 [16386.255326] [c00000016b2978a0] [c00000000013f898] reconfig_close_windows+0x1a0/0x4e8 [16386.255343] [c00000016b297990] [c000000000140e58] vas_migration_handler+0x3a4/0x3fc [16386.255368] [c00000016b297a90] [c000000000128848] pseries_migrate_partition+0x4c/0x4c4 ... [16386.256136] Allocated by task 696554 on cpu 31 at 16377.277618s: [16386.256149] kasan_save_stack+0x34/0x68 [16386.256163] kasan_save_track+0x34/0x80 [16386.256175] kasan_save_alloc_info+0x58/0x74 [16386.256196] __kasan_slab_alloc+0xb8/0xdc [16386.256209] kmem_cache_alloc_noprof+0x200/0x3d0 [16386.256225] vm_area_alloc+0x44/0x150 [16386.256245] mmap_region+0x214/0x10c4 [16386.256265] do_mmap+0x5fc/0x750 [16386.256277] vm_mmap_pgoff+0x14c/0x24c [16386.256292] ksys_mmap_pgoff+0x20c/0x348 [16386.256303] sys_mmap+0xd0/0x160 ... [16386.256350] Freed by task 0 on cpu 31 at 16386.204848s: [16386.256363] kasan_save_stack+0x34/0x68 [16386.256374] kasan_save_track+0x34/0x80 [16386.256384] kasan_save_free_info+0x64/0x10c [16386.256396] __kasan_slab_free+0x120/0x204 [16386.256415] kmem_cache_free+0x128/0x450 [16386.256428] vm_area_free_rcu_cb+0xa8/0xd8 [16386.256441] rcu_do_batch+0x2c8/0xcf0 [16386.256458] rcu_core+0x378/0x3c4 [16386.256473] handle_softirqs+0x20c/0x60c [16386.256495] do_softirq_own_stack+0x6c/0x88 [16386.256509] do_softirq_own_stack+0x58/0x88 [16386.256521] __irq_exit_rcu+0x1a4/0x20c [16386.256533] irq_exit+0x20/0x38 [16386.256544] interrupt_async_exit_prepare.constprop.0+0x18/0x2c ... [16386.256717] Last potentially related work creation: [16386.256729] kasan_save_stack+0x34/0x68 [16386.256741] __kasan_record_aux_stack+0xcc/0x12c [16386.256753] __call_rcu_common.constprop.0+0x94/0xd04 [16386.256766] vm_area_free+0x28/0x3c [16386.256778] remove_vma+0xf4/0x114 [16386.256797] do_vmi_align_munmap.constprop.0+0x684/0x870 [16386.256811] __vm_munmap+0xe0/0x1f8 [16386.256821] sys_munmap+0x54/0x6c [16386.256830] system_call_exception+0x1a0/0x4a0 [16386.256841] system_call_vectored_common+0x15c/0x2ec [16386.256868] The buggy address belongs to the object at c00000014a819670 which belongs to the cache vm_area_struct of size 168 [16386.256887] The buggy address is located 0 bytes inside of freed 168-byte region [c00000014a819670, c00000014a819718] [16386.256915] The buggy address belongs to the physical page: [16386.256928] page: refcount:0 mapping:0000000000000000 index:0x0 pfn:0x14a81 [16386.256950] memcg:c0000000ba430001 [16386.256961] anon flags: 0x43ffff8000000000(node=4 zone=0 lastcpupid=0x7ffff) [16386.256975] page_type: 0xfdfdfdfdf(slab) [16386 ---truncated---

CVE ID	Description
CVE-2024-56767	In the Linux kernel, the following vulnerability has been resolved: dmaengine: at_xdmac: avoid null_ptr_deref in at_xdmac_prep_dma_memset The at_xdmac_memset_create_desc may return NULL, which will lead to a null pointer dereference. For example, the len input is error, or the atchan->free_descs_list is empty and memory is exhausted. Therefore, add check to avoid this.
CVE-2024-56769	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: dib3000mb: fix uninit-value in dib3000_write_reg Syzbot reports [1] an uninitialized value issue found by KMSAN in dib3000_read_reg(). Local u8 rb[2] is used in i2c_transfer() as a read buffer; in case that call fails, the buffer may end up with some undefined values. Since no elaborate error handling is expected in dib3000_write_reg(), simply zero out rb buffer to mitigate the problem. [1] Syzkaller report dvb-usb: bulk message failed: -22 (6/0) ===== BUG: KMSAN: uninit-value in dib3000mb_attach+0x2d8/0x3c0 drivers/media/dvb-frontends/dib3000mb.c:758 dib3000mb_attach+0x2d8/0x3c0 drivers/media/dvb-frontends/dib3000mb.c:758 dibusb_dib3000mb_frontend_attach+0x155/0x2f0 drivers/media/usb/dvb-usb/dibusb-mb.c:31 dvb_usb_adapter_frontend_init+0xed/0x9a0 drivers/media/usb/dvb-usb/dvb-usb-dvb.c:290 dvb_usb_adapter_init drivers/media/usb/dvb-usb/dvb-usb-init.c:90 [inline] dvb_usb_init drivers/media/usb/dvb-usb/dvb-usb-init.c:186 [inline] dvb_usb_device_init+0x25a8/0x3760 drivers/media/usb/dvb-usb/dvb-usb-init.c:310 dibusb_probe+0x46/0x250 drivers/media/usb/dvb-usb/dibusb-mb.c:110 ... Local variable rb created at: dib3000_read_reg+0x86/0x4e0 drivers/media/dvb-frontends/dib3000mb.c:54 dib3000mb_attach+0x123/0x3c0 drivers/media/dvb-frontends/dib3000mb.c:758 ...
CVE-2024-56770	In the Linux kernel, the following vulnerability has been resolved: net/sched: netem: account for backlog updates from child qdisc In general, 'qlen' of any classful qdisc should keep track of the number of packets that the qdisc itself and all of its children holds. In case of netem, 'qlen' only accounts for the packets in its internal tfifo. When netem is used with a child qdisc, the child qdisc can use 'qdisc_tree_reduce_backlog' to inform its parent, netem, about created or dropped SKBs. This function updates 'qlen' and the backlog statistics of netem, but netem does not account for changes made by a child qdisc. 'qlen' then indicates the wrong number of packets in the tfifo. If a child qdisc creates new SKBs during enqueue and informs its parent about this, netem's 'qlen' value is increased. When netem dequeues the newly created SKBs from the child, the 'qlen' in netem is not updated. If 'qlen' reaches the configured sch->limit, the enqueue function stops working, even though the tfifo is not full. Reproduce the bug: Ensure that the sender machine has GSO enabled. Configure netem as root qdisc and tbf as its child on the outgoing interface of the machine as follows: \$ tc qdisc add dev <oif> root handle 1: netem delay 100ms limit 100 \$ tc qdisc add dev <oif> parent 1:0 tbf rate 50Mbit burst 1542 latency 50ms Send bulk TCP traffic out via this interface, e.g., by running an iPerf3 client on the machine. Check the qdisc statistics: \$ tc -s qdisc show dev <oif> Statistics after 10s of iPerf3 TCP test before the fix (note that netem's backlog > limit, netem stopped accepting packets): qdisc netem 1: root refcnt 2 limit 1000 delay 100ms Sent 2767766 bytes 1848 pkt (dropped 652, overlimits 0 requeues 0) backlog 4294528236b 1155p requeues 0 qdisc tbf 10: parent 1:1 rate 50Mbit burst 1537b lat 50ms Sent 2767766 bytes 1848 pkt (dropped 327, overlimits 7601 requeues 0) backlog 0b 0p requeues 0 Statistics after the fix: qdisc netem 1: root refcnt 2 limit 1000 delay 100ms Sent 37766372 bytes 24974 pkt (dropped 9, overlimits 0 requeues 0) backlog 0b 0p requeues 0 qdisc tbf 10: parent 1:1 rate 50Mbit burst 1537b lat 50ms Sent 37766372 bytes 24974 pkt (dropped 327, overlimits 96017 requeues 0) backlog 0b 0p requeues 0 tbf segments the GSO SKBs (tbf_segment) and updates the netem's 'qlen'. The interface fully stops transferring packets and "locks". In this case, the child qdisc and tfifo are empty, but 'qlen' indicates the tfifo is at its limit and no more packets are accepted. This patch adds a counter for the entries in the tfifo. Netem's 'qlen' is only decreased when a packet is returned by its dequeue function, and not during enqueueing into the child qdisc. External updates to 'qlen' are thus accounted for and only the behavior of the backlog statistics changes. As in other qdiscs, 'qlen' then keeps track of how many packets are held in netem and all of its children. As before, sch->limit remains as the maximum number of packets in the tfifo. The same applies to netem's backlog statistics.
CVE-2024-56771	In the Linux kernel, the following vulnerability has been resolved: mtd: spinand: winbond: Fix 512GW, 01GW, 01JW and 02JW ECC information These four chips: * W25N512GW * W25N01GW * W25N01JW * W25N02JW all require a single bit of ECC strength and thus feature an on-die Hamming-like ECC engine. There is no point in filling a ->get_status() callback for them because the main ECC status bytes are located in standard places, and retrieving the number of bitflips in case of corrected chunk is both useless and unsupported (if there are bitflips, then there is 1 at most, so no need to query the chip for that). Without this change, a kernel warning triggers every time a bit flips.

CVE ID	Description
CVE-2024-56772	In the Linux kernel, the following vulnerability has been resolved: kunit: string-stream: Fix a UAF bug in kunit_init_suite() In kunit_debugfs_create_suite(), if alloc_string_stream() fails in the kunit_suite_for_each_test_case() loop, the "suite->log = stream" has assigned before, and the error path only free the suite->log's stream memory but not set it to NULL, so the later string_stream_clear() of suite->log in kunit_init_suite() will cause below UAF bug. Set stream pointer to NULL after free to fix it. Unable to handle kernel paging request at virtual address 006440150000030d Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [006440150000030d] address between user and kernel address ranges Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP Dumping ftrace buffer: (ftrace buffer empty) Modules linked in: iio_test_gts industrialio_gts_helper cfg80211 rkill ipv6 [last unloaded: iio_test_gts] CPU: 5 UID: 0 PID: 6253 Comm: modprobe Tainted: G B W N 6.12.0-rc4+ #458 Tainted: [B]=BAD_PAGE, [W]=WARN, [N]=TEST Hardware name: linux,dummy-virt (DT) pstate: 40000005 (nZcv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) pc : string_stream_clear+0x54/0x1ac lr : string_stream_clear+0x1a8/0x1ac sp : fffffffc080b47410 x29: fffffffc080b47410 x28: 006440550000030d x27: fffffff80c96b5e98 x26: fffffff80c96b5e80 x25: fffffffe461b3f6c0 x24: 0000000000000003 x23: fffffff80c96b5e88 x22: 1ffffff019cdf4fc x21: dffffffc000000000 x20: fffffff80ce6fa7e0 x19: 032202a80000186d x18: 0000000000001840 x17: 0000000000000000 x16: 0000000000000000 x15: fffffffe45c355cb4 x14: fffffffe45c35589c x13: fffffffe45c03da78 x12: fffffffb810168e75 x11: 1ffffffb810168e74 x10: fffffffb810168e74 x9 : dffffffc000000000 x8 : 0000000000000004 x7 : 0000000000000003 x6 : 0000000000000001 x5 : fffffffc080b473a0 x4 : 0000000000000000 x3 : 0000000000000000 x2 : 0000000000000001 x1 : fffffffc462fbf620 x0 : dffffffc000000000 Call trace: string_stream_clear+0x54/0x1ac __kunit_test_suites_init+0x108/0x1d8 kunit_exec_run_tests+0xb8/0x100 kunit_module_notify+0x400/0x55c notifier_call_chain+0xfc/0x3b4 blocking_notifier_call_chain+0x68/0x9c do_init_module+0x24c/0x5c8 load_module+0x4acc/0x4e90 init_module_from_file+0xd4/0x128 idempotent_init_module+0x2d4/0x57c __arm64_sys_finit_module+0xac/0x100 invoke_syscall+0x6c/0x258 el0_svc_common.constprop.0+0x160/0x22c do_el0_svc+0x44/0x5c el0_svc+0x48/0xb8 el0t_64_sync_handler+0x13c/0x158 el0t_64_sync+0x190/0x194 Code: f9400753 d2dff800 f2fbffe0 d343fe7c (38e06b80) ---[end trace 0000000000000000]--- Kernel panic - not syncing: Oops: Fatal exception
CVE-2024-56773	In the Linux kernel, the following vulnerability has been resolved: kunit: Fix potential null dereference in kunit_device_driver_test() kunit_kzalloc() may return a NULL pointer, dereferencing it without NULL check may lead to NULL dereference. Add a NULL check for test_state.
CVE-2024-56774	In the Linux kernel, the following vulnerability has been resolved: btrfs: add a sanity check for btrfs root in btrfs_search_slot() Syzbot reports a null-ptr-deref in btrfs_search_slot(). The reproducer is using rescue=ibadroots, and the extent tree root is corrupted thus the extent tree is NULL. When scrub tries to search the extent tree to gather the needed extent info, btrfs_search_slot() doesn't check if the target root is NULL or not, resulting the null-ptr-deref. Add sanity check for btrfs root before using it in btrfs_search_slot().
CVE-2024-56775	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix handling of plane refcount [Why] The mechanism to backup and restore plane states doesn't maintain refcount, which can cause issues if the refcount of the plane changes in between backup and restore operations, such as memory leaks if the refcount was supposed to go down, or double frees / invalid memory accesses if the refcount was supposed to go up. [How] Cache and re-apply current refcount when restoring plane states.
CVE-2024-56776	In the Linux kernel, the following vulnerability has been resolved: drm/sti: avoid potential dereference of error pointers The return value of drm_atomic_get_crtc_state() needs to be checked. To avoid use of error pointer 'crtc_state' in case of the failure.
CVE-2024-56777	In the Linux kernel, the following vulnerability has been resolved: drm/sti: avoid potential dereference of error pointers in sti_gdp_atomic_check The return value of drm_atomic_get_crtc_state() needs to be checked. To avoid use of error pointer 'crtc_state' in case of the failure.
CVE-2024-56778	In the Linux kernel, the following vulnerability has been resolved: drm/sti: avoid potential dereference of error pointers in sti_hqvd_atomic_check The return value of drm_atomic_get_crtc_state() needs to be checked. To avoid use of error pointer 'crtc_state' in case of the failure.

CVE ID	Description
CVE-2024-56779	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix nfs4_openowner leak when concurrent nfsd4_open occur The action force umount(umount -f) will attempt to kill all rpc_task even umount operation may ultimately fail if some files remain open. Consequently, if an action attempts to open a file, it can potentially send two rpc_task to nfs server. NFS CLIENT thread1 thread2 open("file") ... nfs4_do_open _nfs4_do_open _nfs4_open_and_get_state _nfs4_proc_open nfs4_run_open_task /* rpc_task1 */ / rpc_run_task rpc_wait_for_completion_task umount -f nfs_umount_begin rpc_killall_tasks rpc_signal_task rpc_task1 been wakeup and return -512 _nfs4_do_open // while loop ... nfs4_run_open_task /* rpc_task2 */ / rpc_run_task rpc_wait_for_completion_task While processing an open request, nfsd will first attempt to find or allocate an nfs4_openowner. If it finds an nfs4_openowner that is not marked as NFS4_OO_CONFIRMED, this nfs4_openowner will be released. Since two rpc_task can attempt to open the same file simultaneously from the client to server, and because two instances of nfsd can run concurrently, this situation can lead to lots of memory leak. Additionally, when we echo 0 to /proc/fs/nfsd/threads, warning will be triggered. NFS SERVER nfsd1 nfsd2 echo 0 > /proc/fs/nfsd/threads nfsd4_open nfsd4_process_open1 find_or_alloc_open_stateowner // alloc oo1, stateid1 nfsd4_open nfsd4_process_open1 find_or_alloc_open_stateowner // find oo1, without NFS4_OO_CONFIRMED release_openowner unhash_openowner_locked list_del_init(&oo->oo_perclient) // cannot find this oo // from client, LEAK!!! alloc_stateowner // alloc oo2 nfsd4_process_open2 init_open_stateid // associate oo1 // with stateid1, stateid1 LEAK!!! nfs4_get_vfs_file // alloc nfsd_file1 and nfsd_file_mark1 // all LEAK!!! nfsd4_process_open2 ... write_threads ... nfsd_destroy_serv nfsd_shutdown_net nfs4_state_shutdown_net nfs4_state_destroy_net destroy_client __destroy_client // won't find oo1!!! nfsd_shutdown_generic nfsd_file_cache_shutdown kmem_cache_destroy for nfsd_file_slab and nfsd_file_mark_slab // bark since nfsd_file1 // and nfsd_file_mark1 // still alive <pre>===== BUG nfsd_file (Not tainted): Objects remaining in nfsd_file on __kmem_cache_shutdown() ----- Slab 0xffd4000004438a80 objects=34 used=1 fp=0xff11000110e2ad28 flags=0x17ffff0000240(workingset head node=0 zone=2 lastcpuid=0x1fffff) CPU: 4 UID: 0 PID: 757 Comm: sh Not tainted 6.12.0-rc6+ #19 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 Call Trace: <TASK> dum ---truncated---</pre>
CVE-2024-56780	In the Linux kernel, the following vulnerability has been resolved: quota: flush quota_release_work upon quota writeback One of the paths quota writeback is called from is: freeze_super() sync_filesystem() ext4_sync_fs() dquot_writeback_dquots() Since we currently don't always flush the quota_release_work queue in this path, we can end up with the following race: 1. dquot are added to releasing_dquots list during regular operations. 2. FS Freeze starts, however, this does not flush the quota_release_work queue. 3. Freeze completes. 4. Kernel eventually tries to flush the workqueue while FS is frozen which hits a WARN_ON since transaction gets started during frozen state: ext4_journal_check_start +0x28/0x110 [ext4] (unreliable) __ext4_journal_start_sb+0x64/0x1c0 [ext4] ext4_release_dquot +0x90/0x1d0 [ext4] quota_release_workfn+0x43c/0x4d0 Which is the following line: WARN_ON(sb->s_writers.frozen == SB_FREEZE_COMPLETE); Which ultimately results in generic/390 failing due to dmesg noise. This was detected on powerpc machine 15 cores. To avoid this, make sure to flush the workqueue during dquot_writeback_dquots() so we dont have any pending workitems after freeze.
CVE-2024-56781	In the Linux kernel, the following vulnerability has been resolved: powerpc/prom_init: Fixup missing powermac #size-cells On some powermacs `esc` nodes are missing `#size-cells` properties, which is deprecated and now triggers a warning at boot since commit 045b14ca5c36 ("of: WARN on deprecated #address-cells/#size-cells handling"). For example: Missing `#size-cells` in /pci@f2000000/mac-io@c/esc@13000 WARNING: CPU: 0 PID: 0 at drivers/of/base.c:133 of_bus_n_size_cells+0x98/0x108 Hardware name: PowerMac3,1 7400 0xc0209 PowerMac ... Call Trace: of_bus_n_size_cells +0x98/0x108 (unreliable) of_bus_default_count_cells+0x40/0x60 __of_get_address+0xc8/0x21c __of_address_to_resource+0x5c/0x228 pmz_init_port+0x5c/0x2ec pmz_probe.isra.0+0x144/0x1e4 pmz_console_init+0x10/0x48 console_init+0xcc/0x138 start_kernel+0x5c4/0x694 As powermacs boot via prom_init it's possible to add the missing properties to the device tree during boot, avoiding the warning. Note that `esc-legacy` nodes are also missing `#size-cells` properties, but they are skipped by the macio driver, so leave them alone. Depends-on: 045b14ca5c36 ("of: WARN on deprecated #address-cells/#size-cells handling")
CVE-2024-56782	In the Linux kernel, the following vulnerability has been resolved: ACPI: x86: Add adev NULL check to acpi_quirk_skip_serdev_enumeration() acpi_dev_hid_match() does not check for adev == NULL, dereferencing it unconditional. Add a check for adev being NULL before calling acpi_dev_hid_match(). At the moment acpi_quirk_skip_serdev_enumeration() is never called with a controller_parent without an ACPI companion, but better safe than sorry.
CVE-2024-56783	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_socket: remove WARN_ON_ONCE on maximum cgroup level cgroup maximum depth is INT_MAX by default, there is a cgroup toggle to restrict this maximum depth to a more reasonable value not to harm performance. Remove unnecessary WARN_ON_ONCE which is reachable from userspace.

CVE ID	Description
CVE-2024-56784	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Adding array index check to prevent memory corruption [Why & How] Array indices out of bound caused memory corruption. Adding checks to ensure that array index stays in bound.
CVE-2024-56785	In the Linux kernel, the following vulnerability has been resolved: MIPS: Loongson64: DTS: Really fix PCIe port nodes for ls7a Fix the dtc warnings: arch/mips/boot/dts/loongson/ls7a-pch.dtsi:68.16-416.5: Warning (interrupt_provider): /bus@10000000/pci@1a000000: '#interrupt-cells' found, but node is not an interrupt provider arch/mips/boot/dts/loongson/ls7a-pch.dtsi:68.16-416.5: Warning (interrupt_provider): /bus@10000000/pci@1a000000: '#interrupt-cells' found, but node is not an interrupt provider arch/mips/boot/dts/loongson/loongson64g_4core_ls7a.dtb: Warning (interrupt_map): Failed prerequisite 'interrupt_provider' And a runtime warning introduced in commit 045b14ca5c36 ("of: WARN on deprecated #address-cells/#size-cells handling"): WARNING: CPU: 0 PID: 1 at drivers/of/base.c:106 of_bus_n_addr_cells+0x9c/0xe0 Missing '#address-cells' in /bus@10000000/pci@1a000000/pci_bridge@9,0 The fix is similar to commit d89a415ff8d5 ("MIPS: Loongson64: DTS: Fix PCIe port nodes for ls7a"), which has fixed the issue for ls2k (despite its subject mentions ls7a).
CVE-2024-56787	In the Linux kernel, the following vulnerability has been resolved: soc: imx8m: Probe the SoC driver as platform driver With driver_async_probe=* on kernel command line, the following trace is produced because on i.MX8M Plus hardware because the soc-imx8m.c driver calls of_clk_get_by_name() which returns -EPROBE_DEFER because the clock driver is not yet probed. This was not detected during regular testing without driver_async_probe. Convert the SoC code to platform driver and instantiate a platform device in its current device_initcall() to probe the platform driver. Rework .soc_revision callback to always return valid error code and return SoC revision via parameter. This way, if anything in the .soc_revision callback return -EPROBE_DEFER, it gets propagated to .probe and the .probe will get retried later. " -----[cut here]----- WARNING: CPU: 1 PID: 1 at drivers/soc/imx/soc-imx8m.c:115 imx8mm_soc_revision+0xdc/0x180 CPU: 1 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.11.0-next-20240924-00002-g2062bb554dea #603 Hardware name: DH electronics i.MX8M Plus DHCOM Premium Developer Kit (3) (DT) pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : imx8mm_soc_revision+0xdc/0x180 lr : imx8mm_soc_revision+0xd0/0x180 sp : ffff8000821fbcc0 x29: ffff8000821fbce0 x28: 0000000000000000 x27: ffff800081810120 x26: ffff8000818a9970 x25: 0000000000000006 x24: 0000000000824311 x23: ffff8000817f42c8 x22: ffff0000df8be210 x21: ffffffffdfdb x20: ffff800082780000 x19: 0000000000000001 x18: ffffffff x17: ffff800081fff418 x16: ffff8000823e1000 x15: ffff0000c03b65e8 x14: ffff0000c00051b0 x13: ffff800082790000 x12: 00000000000000801 x11: ffff80008278ffff x10: ffff80008209d3a6 x9: ffff80008062e95c x8: ffff8000821fb9a0 x7: 0000000000000000 x6: 00000000000080e3 x5: ffff0000df8c03d8 x4: 0000000000000000 x3: 0000000000000000 x2: 0000000000000000 x1: ffffffffdfdb x0: ffffffffdfdb Call trace: imx8mm_soc_revision+0xdc/0x180 imx8_soc_init+0xb0/0x1e0 do_one_initcall+0x94/0x1a8 kernel_init_freeable+0x240/0x2a8 kernel_init+0x28/0x140 ret_from_fork+0x10/0x20 ---[end trace 0000000000000000]--- SoC: i.MX8MP revision 1.1 "

CVE ID	Description
CVE-2024-56788	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: oa_tc6: fix tx skb race condition between reference pointers There are two skb pointers to manage tx skb's enqueued from n/w stack. waiting_tx_skb pointer points to the tx skb which needs to be processed and ongoing_tx_skb pointer points to the tx skb which is being processed. SPI thread prepares the tx data chunks from the tx skb pointed by the ongoing_tx_skb pointer. When the tx skb pointed by the ongoing_tx_skb is processed, the tx skb pointed by the waiting_tx_skb is assigned to ongoing_tx_skb and the waiting_tx_skb pointer is assigned with NULL. Whenever there is a new tx skb from n/w stack, it will be assigned to waiting_tx_skb pointer if it is NULL. Enqueuing and processing of a tx skb handled in two different threads. Consider a scenario where the SPI thread processed an ongoing_tx_skb and it moves next tx skb from waiting_tx_skb pointer to ongoing_tx_skb pointer without doing any NULL check. At this time, if the waiting_tx_skb pointer is NULL then ongoing_tx_skb pointer is also assigned with NULL. After that, if a new tx skb is assigned to waiting_tx_skb pointer by the n/w stack and there is a chance to overwrite the tx skb pointer with NULL in the SPI thread. Finally one of the tx skb will be left as unhandled, resulting packet missing and memory leak. - Consider the below scenario where the TXC reported from the previous transfer is 10 and ongoing_tx_skb holds an tx ethernet frame which can be transported in 20 TXCs and waiting_tx_skb is still NULL. tx_credits = 10; /* 21 are filled in the previous transfer */ ongoing_tx_skb = 20; waiting_tx_skb = NULL; /* Still NULL */ - So, (tc6->ongoing_tx_skb tc6->waiting_tx_skb) becomes true. - After oa_tc6_prepare_spi_tx_buf_for_tx_skbs() ongoing_tx_skb = 10; waiting_tx_skb = NULL; /* Still NULL */ - Perform SPI transfer. - Process SPI rx buffer to get the TXC from footers. - Now let's assume previously filled 21 TXCs are freed so we are good to transport the next remaining 10 tx chunks from ongoing_tx_skb. tx_credits = 21; ongoing_tx_skb = 10; waiting_tx_skb = NULL; - So, (tc6->ongoing_tx_skb tc6->waiting_tx_skb) becomes true again. - In the oa_tc6_prepare_spi_tx_buf_for_tx_skbs() ongoing_tx_skb = NULL; waiting_tx_skb = NULL; - Now the below bad case might happen, Thread1 (oa_tc6_start_xmit) Thread2 (oa_tc6_spi_thread_handler) ----- - if waiting_tx_skb is NULL - if ongoing_tx_skb is NULL - ongoing_tx_skb = waiting_tx_skb - waiting_tx_skb = skb - waiting_tx_skb = NULL ... - ongoing_tx_skb = NULL - if waiting_tx_skb is NULL - waiting_tx_skb = skb To overcome the above issue, protect the moving of tx skb reference from waiting_tx_skb pointer to ongoing_tx_skb pointer and assigning new tx skb to waiting_tx_skb pointer, so that the other thread can't access the waiting_tx_skb pointer until the current thread completes moving the tx skb reference safely.
CVE-2024-57360	https://www.gnu.org/software/binutils/ nm >=2.43 is affected by: Incorrect Access Control. The type of exploitation is: local. The component is: `nm --without-symbol-version` function.
CVE-2024-57791	In the Linux kernel, the following vulnerability has been resolved: net/smc: check return value of sock_recvmsg when draining clc data When receiving clc msg, the field length in smc_clc_msg_hdr indicates the length of msg should be received from network and the value should not be fully trusted as it is from the network. Once the value of length exceeds the value of buflen in function smc_clc_wait_msg it may run into deadlock when trying to drain the remaining data exceeding buflen. This patch checks the return value of sock_recvmsg when draining data in case of deadlock in draining.
CVE-2024-57792	In the Linux kernel, the following vulnerability has been resolved: power: supply: gpio-charger: Fix set charge current limits Fix set charge current limits for devices which allow to set the lowest charge current limit to be greater zero. If requested charge current limit is below lowest limit, the index equals current_limit_map_size which leads to accessing memory beyond allocated memory.
CVE-2024-57793	In the Linux kernel, the following vulnerability has been resolved: virt: tdx-guest: Just leak decrypted memory on unrecoverable errors In CoCo VMs it is possible for the untrusted host to cause set_memory_decrypted() to fail such that an error is returned and the resulting memory is shared. Callers need to take care to handle these errors to avoid returning decrypted (shared) memory to the page allocator, which could lead to functional or security issues. Leak the decrypted memory when set_memory_decrypted() fails, and don't need to print an error since set_memory_decrypted() will call WARN_ONCE().

CVE ID	Description
CVE-2024-57795	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Remove the direct link to net_device The similar patch in siw is in the link: https://git.kernel.org/rdma/rdma/c/16b87037b48889 This problem also occurred in RXE. The following analyze this problem. In the following Call Traces: " BUG: KASAN: slab-use-after-free in dev_get_flags+0x188/0x1d0 net/core/dev.c:8782 Read of size 4 at addr ffff8880554640b0 by task kworker/1:4/5295 CPU: 1 UID: 0 PID: 5295 Comm: kworker/1:4 Not tainted 6.12.0-rc3-syzkaller-00399-g9197b73fd7bb #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: infiniband ib_cache_event_task Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:377 [inline] print_report+0x169/0x550 mm/kasan/report.c:488 kasan_report+0x143/0x180 mm/kasan/report.c:601 dev_get_flags+0x188/0x1d0 net/core/dev.c:8782 rxe_query_port+0x12d/0x260 drivers/infiniband/sw/rxe/rxe_verbs.c:60 __ib_query_port drivers/infiniband/core/device.c:2111 [inline] ib_query_port+0x168/0x7d0 drivers/infiniband/core/device.c:2143 ib_cache_update+0x1a9/0xb80 drivers/infiniband/core/cache.c:1494 ib_cache_event_task+0xf3/0x1e0 drivers/infiniband/core/cache.c:1568 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa65/0x1850 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f2/0x390 kernel/kthread.c:389 ret_from_fork+0x4d/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> " 1). In the link [1], " infiniband syz2: set down " This means that on 839.350575, the event ib_cache_event_task was sent and queued in ib_wq. 2). In the link [1], " team0 (unregistering): Port device team_slave_0 removed " It indicates that before 843.251853, the net device should be freed. 3). In the link [1], " BUG: KASAN: slab-use-after-free in dev_get_flags+0x188/0x1d0 " This means that on 850.559070, this slab-use-after-free problem occurred. In all, on 839.350575, the event ib_cache_event_task was sent and queued in ib_wq, before 843.251853, the net device veth was freed. on 850.559070, this event was executed, and the mentioned freed net device was called. Thus, the above call trace occurred. [1] https://syzkaller.appspot.com/x/log.txt?x=12e7025f980000
CVE-2024-57798	In the Linux kernel, the following vulnerability has been resolved: drm/dp_mst: Ensure mst_primary pointer is valid in drm_dp_mst_handle_up_req() While receiving an MST up request message from one thread in drm_dp_mst_handle_up_req(), the MST topology could be removed from another thread via drm_dp_mst_topology_mgr_set_mst(false), freeing mst_primary and setting drm_dp_mst_topology_mgr::mst_primary to NULL. This could lead to a NULL deref/use-after-free of mst_primary in drm_dp_mst_handle_up_req(). Avoid the above by holding a reference for mst_primary in drm_dp_mst_handle_up_req() while it's used. v2: Fix kfreeing the request if getting an mst_primary reference fails.
CVE-2024-57801	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Skip restore TC rules for vport rep without loaded flag During driver unload, unregister_netdev is called after unloading vport rep. So, the mlx5e_rep_priv is already freed while trying to get rpriv->netdev, or walk rpriv->tc_ht, which results in use-after-free. So add the checking to make sure access the data of vport rep which is still loaded.

CVE ID	Description
CVE-2024-57802	In the Linux kernel, the following vulnerability has been resolved: netrom: check buffer length before accessing it Syzkaller reports an uninit value read from ax25cmp when sending raw message through ieee802154 implementation. ===== BUG: KMSAN: uninit-value in ax25cmp+0x3a5/0x460 net/ax25/ax25_addr.c:119 ax25cmp+0x3a5/0x460 net/ax25/ax25_addr.c:119 nr_dev_get+0x20e/0x450 net/netrom/nr_route.c:601 nr_route_frame+0x1a2/0xfc0 net/netrom/nr_route.c:774 nr_xmit+0x5a/0x1c0 net/netrom/nr_dev.c:144 __netdev_start_xmit include/linux/netdevice.h:4940 [inline] netdev_start_xmit include/linux/netdevice.h:4954 [inline] xmit_one net/core/dev.c:3548 [inline] dev_hard_start_xmit+0x247/0xa10 net/core/dev.c:3564 __dev_queue_xmit+0x33b8/0x5130 net/core/dev.c:4349 dev_queue_xmit include/linux/netdevice.h:3134 [inline] raw_sendmsg+0x654/0xc10 net/ieee802154/socket.c:299 ieee802154_sock_sendmsg+0x91/0xc0 net/ieee802154/socket.c:96 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] ____sys_sendmsg+0x9c2/0xd60 net/socket.c:2584 ____sys_sendmsg+0x28d/0x3c0 net/socket.c:2638 __sys_sendmsg net/socket.c:2667 [inline] __do_sys_sendmsg net/socket.c:2676 [inline] __se_sys_sendmsg net/socket.c:2674 [inline] __x64_sys_sendmsg+0x307/0x490 net/socket.c:2674 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x44/0x110 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b Uninit was created at: slab_post_alloc_hook+0x129/0xa70 mm/slab.h:768 slab_alloc_node mm/slub.c:3478 [inline] kmem_cache_alloc_node+0x5e9/0xb10 mm/slub.c:3523 kmallocc_reserve+0x13d/0x4a0 net/core/skbuff.c:560 __alloc_skb+0x318/0x740 net/core/skbuff.c:651 alloc_skb include/linux/skbuff.h:1286 [inline] alloc_skb_with_frags+0xc8/0xbd0 net/core/skbuff.c:6334 sock_alloc_send_skb+0xa80/0xbf0 net/core/sock.c:2780 sock_alloc_send_skb include/net/sock.h:1884 [inline] raw_sendmsg+0x36d/0xc10 net/ieee802154/socket.c:282 ieee802154_sock_sendmsg+0x91/0xc0 net/ieee802154/socket.c:96 sock_sendmsg_nosec net/socket.c:730 [inline] __sock_sendmsg net/socket.c:745 [inline] ____sys_sendmsg+0x9c2/0xd60 net/socket.c:2584 ____sys_sendmsg+0x28d/0x3c0 net/socket.c:2638 __sys_sendmsg net/socket.c:2667 [inline] __do_sys_sendmsg net/socket.c:2676 [inline] __se_sys_sendmsg net/socket.c:2674 [inline] __x64_sys_sendmsg+0x307/0x490 net/socket.c:2674 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0x44/0x110 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x63/0x6b CPU: 0 PID: 5037 Comm: syz-executor166 Not tainted 6.7.0-rc7-syzkaller-00003-gfbafc3e621c3 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 11/17/2023 ===== This issue occurs because the skb buffer is too small, and it's actual allocation is aligned. This hides an actual issue, which is that nr_route_frame does not validate the buffer size before using it. Fix this issue by checking skb->len before accessing any fields in skb->data. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2024-57804	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix corrupt config pages PHY state is switched in sysfs The driver, through the SAS transport, exposes a sysfs interface to enable/disable PHYs in a controller/expander setup. When multiple PHYs are disabled and enabled in rapid succession, the persistent and current config pages related to SAS IO unit/SAS Expander pages could get corrupted. Use separate memory for each config request.
CVE-2024-57805	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda-dai: Do not release the link DMA on STOP The linkDMA should not be released on stop trigger since a stream restart might happen without closing of the stream. This leaves a short time for other streams to 'steal' the linkDMA since it has been released. This issue is not easy to reproduce under normal conditions as usually after stop the stream is closed, or the same stream is restarted, but if another stream got in between the stop and start, like this: aplay -Dhw:0,3 -c2 -r48000 -fS32_LE /dev/zero -d 120 CTRL+z aplay -Dhw:0,0 -c2 -r48000 -fS32_LE /dev/zero -d 120 then the link DMA channels will be mixed up, resulting firmware error or crash.

CVE ID	Description
CVE-2024-57806	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix transaction atomicity bug when enabling simple quotas Set squash incompat bit before committing the transaction that enables the feature. With the config CONFIG_BTRFS_ASSERT enabled, an assertion failure occurs regarding the simple quota feature. [5.596534] assertion failed: btrfs_fs_incompat(fs_info, SIMPLE_QUOTA), in fs/btrfs/qgroup.c:365 [5.597098] -----[cut here]----- [5.597371] kernel BUG at fs/btrfs/qgroup.c:365! [5.597946] CPU: 1 UID: 0 PID: 268 Comm: mount Not tainted 6.13.0-rc2-00031-gf92f4749861b #146 [5.598450] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.2-debian-1.16.2-1 04/01/2014 [5.599008] RIP: 0010:btrfs_read_qgroup_config+0x74d/0x7a0 [5.604303] <TASK> [5.605230] ? btrfs_read_qgroup_config+0x74d/0x7a0 [5.605538] ? exc_invalid_op+0x56/0x70 [5.605775] ? btrfs_read_qgroup_config+0x74d/0x7a0 [5.606066] ? asm_exc_invalid_op+0x1f/0x30 [5.606441] ? btrfs_read_qgroup_config+0x74d/0x7a0 [5.606741] ? btrfs_read_qgroup_config+0x74d/0x7a0 [5.607038] ? try_to_wake_up+0x317/0x760 [5.607286] open_ctree+0xd9c/0x1710 [5.607509] btrfs_get_tree+0x58a/0x7e0 [5.608002] vfs_get_tree+0x2e/0x100 [5.608224] fc_mount+0x16/0x60 [5.608420] btrfs_get_tree+0x2f8/0x7e0 [5.608897] vfs_get_tree+0x2e/0x100 [5.609121] path_mount+0x4c8/0xabc0 [5.609538] __x64_sys_mount+0x10d/0x150 The issue can be easily reproduced using the following reproducer: root@q:linux# cat repro.sh set -e mkfs.btrfs -q -f /dev/sdb mount /dev/sdb /mnt/btrfs btrfs quota enable -s /mnt/btrfs umount /mnt/btrfs mount /dev/sdb /mnt/btrfs The issue is that when enabling quotas, at btrfs_quota_enable(), we set BTRFS_QGROUP_STATUS_FLAG_SIMPLE_MODE at fs_info->qgroup_flags and persist it in the quota root in the item with the key BTRFS_QGROUP_STATUS_KEY, but we only set the incompat bit BTRFS_FEATURE_INCOMPAT_SIMPLE_QUOTA after we commit the transaction used to enable simple quotas. This means that if after that transaction commit we unmount the filesystem without starting and committing any other transaction, or we have a power failure, the next time we mount the filesystem we will find the flag BTRFS_QGROUP_STATUS_FLAG_SIMPLE_MODE set in the item with the key BTRFS_QGROUP_STATUS_KEY but we will not find the incompat bit BTRFS_FEATURE_INCOMPAT_SIMPLE_QUOTA set in the superblock, triggering an assertion failure at: btrfs_read_qgroup_config() -> qgroup_read_enable_gen() To fix this issue, set the BTRFS_FEATURE_INCOMPAT_SIMPLE_QUOTA flag immediately after setting the BTRFS_QGROUP_STATUS_FLAG_SIMPLE_MODE. This ensures that both flags are flushed to disk within the same transaction.
CVE-2024-57807	In the Linux kernel, the following vulnerability has been resolved: scsi: megasas: Fix for a potential deadlock This fixes a 'possible circular locking dependency detected' warning CPU0 CPU1 ---- lock(&instance->reset_mutex); lock(&shost->scan_mutex); lock(&instance->reset_mutex); lock(&shost->scan_mutex); Fix this by temporarily releasing the reset_mutex.
CVE-2024-57809	In the Linux kernel, the following vulnerability has been resolved: PCI: imx6: Fix suspend/resume support on i.MX6QDL The suspend/resume functionality is currently broken on the i.MX6QDL platform, as documented in the NXP errata (ERR005723): https://www.nxp.com/docs/en/errata/IMX6DQCE.pdf This patch addresses the issue by sharing most of the suspend/resume sequences used by other i.MX devices, while avoiding modifications to critical registers that disrupt the PCIe functionality. It targets the same problem as the following downstream commit: https://github.com/nxp-imx/linux-imx/commit/4e92355e1f79d225ea842511fcfd42b343b32995 Unlike the downstream commit, this patch also resets the connected PCIe device if possible. Without this reset, certain drivers, such as ath10k or iwlmwifi, will crash on resume. The device reset is also done by the driver on other i.MX platforms, making this patch consistent with existing practices. Upon resuming, the kernel will hang and display an error. Here's an example of the error encountered with the ath10k driver: ath10k_pcie 0000:01:00.0: Unable to change power state from D3hot to D0, device inaccessible Unhandled fault: imprecise external abort (0x1406) at 0x0106f944 Without this patch, suspend/resume will fail on i.MX6QDL devices if a PCIe device is connected. [kwilczynski: commit log, added tag for stable releases]

CVE ID	Description
CVE-2024-57834	In the Linux kernel, the following vulnerability has been resolved: media: vidtv: Fix a null-ptr-deref in vidtv_mux_stop_thread syzbot report a null-ptr-deref in vidtv_mux_stop_thread. [1] If dvb->mux is not initialized successfully by vidtv_mux_init() in the vidtv_start_streaming(), it will trigger null pointer dereference about mux in vidtv_mux_stop_thread(). Adjust the timing of streaming initialization and check it before stopping it. [1] KASAN: null-ptr-deref in range [0x0000000000000128-0x000000000000012f] CPU: 0 UID: 0 PID: 5842 Comm: syz-executor248 Not tainted 6.13.0-rc4-syzkaller-00012-g9b2ffa6148b1 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010:vidtv_mux_stop_thread+0x26/0x80 drivers/media/test-drivers/vidtv/vidtv_mux.c:471 Code: 90 90 90 90 66 0f 1f 00 55 53 48 89 fb e8 82 2e c8 f9 48 8d bb 28 01 00 00 48 b8 00 00 00 00 00 ff df 48 89 fa 48 c1 ea 03 <0f> b6 04 02 84 c0 74 02 7e 3b 0f b6 ab 28 01 00 00 31 ff 89 ee e8 RSP: 0018:ffff90003f2faa8 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: 0000000000000000 RCX: ffffffff87cfb125 RDX: 0000000000000025 RSI: ffffffff87d120ce RDI: 0000000000000128 RBP: ffff888029b8d220 R08: 0000000000000005 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000003 R12: ffff888029b8d188 R13: ffffffff8f590aa0 R14: ffff9000581c5c8 R15: ffff888029a17710 FS: 00007fef5156c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fef5e635c CR3: 0000000076ca6000 CR4: 0000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> vidtv_stop_streaming drivers/media/test-drivers/vidtv/vidtv_bridge.c:209 [inline] vidtv_stop_feed+0x151/0x250 drivers/media/test-drivers/vidtv/vidtv_bridge.c:252 dmux_section_feed_stop_filtering+0x90/0x160 drivers/media/dvb-core/dvb_demux.c:1000 dvb_dmxdev_feed_stop.isra.0+0x1ee/0x270 drivers/media/dvb-core/dmxdev.c:486 dvb_dmxdev_filter_stop+0x22a/0x3a0 drivers/media/dvb-core/dmxdev.c:559 dvb_dmxdev_filter_free drivers/media/dvb-core/dmxdev.c:840 [inline] dvb_demux_release+0x92/0x550 drivers/media/dvb-core/dmxdev.c:1246 __fput+0x3f8/0xb60 fs/file_table.c:450 task_work_run+0x14e/0x250 kernel/task_work.c:239 get_signal+0x1d3/0x2610 kernel/signal.c:2790 arch_do_signal_or_restart+0x90/0x7e0 arch/x86/kernel/signal.c:337 exit_to_user_mode_loop kernel/entry/common.c:111 [inline] exit_to_user_mode_prepare include/linux/entry-common.h:329 [inline] __syscall_exit_to_user_mode_work kernel/entry/common.c:207 [inline] syscall_exit_to_user_mode+0x150/0x2a0 kernel/entry/common.c:218 do_syscall_64+0xda/0x250 arch/x86/entry/common.c:89 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2024-57838	In the Linux kernel, the following vulnerability has been resolved: s390/entry: Mark IRQ entries to fix stack depot warnings The stack depot filters out everything outside of the top interrupt context as an uninteresting or irrelevant part of the stack traces. This helps with stack trace de-duplication, avoiding an explosion of saved stack traces that share the same IRQ context code path but originate from different randomly interrupted points, eventually exhausting the stack depot. Filtering uses in_irqentry_text() to identify functions within the .irqentry.text and .softirqentry.text sections, which then become the last stack trace entries being saved. While __do_softirq() is placed into the .softirqentry.text section by common code, populating .irqentry.text is architecture-specific. Currently, the .irqentry.text section on s390 is empty, which prevents stack depot filtering and de-duplication and could result in warnings like: Stack depot reached limit capacity WARNING: CPU: 0 PID: 286113 at lib/stackdepot.c:252 depot_alloc_stack+0x39a/0x3c8 with PREEMPT and KASAN enabled. Fix this by moving the IO/EXT interrupt handlers from .kprobes.text into the .irqentry.text section and updating the kprobes blacklist to include the .irqentry.text section. This is done only for asynchronous interrupts and explicitly not for program checks, which are synchronous and where the context beyond the program check is important to preserve. Despite machine checks being somewhat in between, they are extremely rare, and preserving context when possible is also of value. SVCs and Restart Interrupts are not relevant, one being always at the boundary to user space and the other being a one-time thing. IRQ entries filtering is also optionally used in ftrace function graph, where the same logic applies.
CVE-2024-57841	In the Linux kernel, the following vulnerability has been resolved: net: fix memory leak in tcp_conn_request() If inet_csk_reqsk_queue_hash_add() return false, tcp_conn_request() will return without free the dst memory, which allocated in af_ops->route_req. Here is the kmemleak stack: unreferenced object 0xffff8881198631c0 (size 240): comm "softirq", pid 0, jiffies 4299266571 (age 1802.392s) hex dump (first 32 bytes): 00 10 9b 03 81 88 ff ff 80 98 da bc ff ff ff ff 81 55 18 bb ff ff ff ff 00 00 00 00 00 00 00 00 .U..... backtrace: [<ffffffffff93e8d4c>] kmem_cache_alloc+0x60c/0xa80 [<ffffffffffba11b4c5>] dst_alloc+0x55/0x250 [<ffffffffffba227bf6>] rt_dst_alloc+0x46/0x1d0 [<ffffffffffba23050a>] __mkroute_output+0x29a/0xa50 [<ffffffffffba23456b>] ip_route_output_key_hash+0x10b/0x240 [<ffffffffffba2346bd>] ip_route_output_flow+0x1d/0x90 [<ffffffffffba254855>] inet_csk_route_req+0x2c5/0x500 [<ffffffffffba26b331>] tcp_conn_request+0x691/0x12c0 [<ffffffffffba27bd08>] tcp_rcv_state_process+0x3c8/0x11b0 [<ffffffffffba2965c6>] tcp_v4_do_rcv+0x156/0x3b0 [<ffffffffffba299c98>] tcp_v4_rcv+0x1cf8/0x1d80 [<ffffffffffba239656>] ip_protocol_deliver_rcu+0xf6/0x360 [<ffffffffffba2399a6>] ip_local_deliver_finish+0xe6/0x1e0 [<ffffffffffba239b8e>] ip_local_deliver+0xee/0x360 [<ffffffffffba239ead>] ip_rcv+0xad/0x2f0 [<ffffffffffba110943>] __netif_receive_skb_one_core+0x123/0x140 Call dst_release() to free the dst memory when inet_csk_reqsk_queue_hash_add() return false in tcp_conn_request().

CVE ID	Description
CVE-2024-57843	In the Linux kernel, the following vulnerability has been resolved: virtio-net: fix overflow inside virtnet_rq_alloc When the frag just got a page, then may lead to regression on VM. Specially if the sysctl net.core.high_order_alloc_disable value is 1, then the frag always get a page when do refill. Which could see reliable crashes or scp failure (scp a file 100M in size to VM). The issue is that the virtnet_rq_dma takes up 16 bytes at the beginning of a new frag. When the frag size is larger than PAGE_SIZE, everything is fine. However, if the frag is only one page and the total size of the buffer and virtnet_rq_dma is larger than one page, an overflow may occur. The commit f9dac92ba908 ("virtio_ring: enable premapped mode whatever use_dma_api") introduced this problem. And we reverted some commits to fix this in last linux version. Now we try to enable it and fix this bug directly. Here, when the frag size is not enough, we reduce the buffer len to fix this problem.
CVE-2024-57849	In the Linux kernel, the following vulnerability has been resolved: s390/cpum_sf: Handle CPU hotplug remove during sampling CPU hotplug remove handling triggers the following function call sequence: CPUHP_AP_PERF_S390_SF_ONLINE --> s390_pmu_sf_offline_cpu() ... CPUHP_AP_PERF_ONLINE --> perf_event_exit_cpu() The s390 CPUMF sampling CPU hotplug handler invokes: s390_pmu_sf_offline_cpu() +--> cpusf_pmu_setup() +--> setup_pmc_cpu() +--> deallocate_buffers() This function de-allocates all sampling data buffers (SDBs) allocated for that CPU at event initialization. It also clears the PMU_F_RESERVED bit. The CPU is gone and can not be sampled. With the event still being active on the removed CPU, the CPU event hotplug support in kernel performance subsystem triggers the following function calls on the removed CPU: perf_event_exit_cpu() +--> perf_event_exit_cpu_context() +--> __perf_event_exit_context() +--> __perf_remove_from_context() +--> event_sched_out() +--> cpumfsf_pmu_del() +--> cpumfsf_pmu_stop() +--> hw_perf_event_update() to stop and remove the event. During removal of the event, the sampling device driver tries to read out the remaining samples from the sample data buffers (SDBs). But they have already been freed (and may have been re-assigned). This may lead to a use after free situation in which case the samples are most likely invalid. In the best case the memory has not been reassigned and still contains valid data. Remedy this situation and check if the CPU is still in reserved state (bit PMU_F_RESERVED set). In this case the SDBs have not been released and contain valid data. This is always the case when the event is removed (and no CPU hotplug off occurred). If the PMU_F_RESERVED bit is not set, the SDB buffers are gone.
CVE-2024-57850	In the Linux kernel, the following vulnerability has been resolved: jffs2: Prevent rtime decompress memory corruption The rtime decompression routine does not fully check bounds during the entirety of the decompression pass and can corrupt memory outside the decompression buffer if the compressed data is corrupted. This adds the required check to prevent this failure mode.
CVE-2024-57852	In the Linux kernel, the following vulnerability has been resolved: firmware: qcom: scm: smc: Handle missing SCM device Commit ca61d6836e6f ("firmware: qcom: scm: fix a NULL-pointer dereference") makes it explicit that qcom_scm_get_tzmem_pool() can return NULL, therefore its users should handle this.
CVE-2024-57857	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Remove direct link to net_device Do not manage a per device direct link to net_device. Rely on associated ib_devices net_device management, not doubling the effort locally. A badly managed local link to net_device was causing a 'KASAN: slab-use-after-free' exception during siw_query_port() call.
CVE-2024-57872	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: pltfm: Dellocate HBA during ufshcd_pltfm_remove() This will ensure that the scsi host is cleaned up properly using scsi_host_dev_release(). Otherwise, it may lead to memory leaks.
CVE-2024-57874	In the Linux kernel, the following vulnerability has been resolved: arm64: ptrace: fix partial SETREGSET for NT_ARM_TAGGED_ADDR_CTRL Currently tagged_addr_ctrl_set() doesn't initialize the temporary 'ctrl' variable, and a SETREGSET call with a length of zero will leave this uninitialized. Consequently tagged_addr_ctrl_set() will consume an arbitrary value, potentially leaking up to 64 bits of memory from the kernel stack. The read is limited to a specific slot on the stack, and the issue does not provide a write mechanism. As set_tagged_addr_ctrl() only accepts values where bits [63:4] zero and rejects other values, a partial SETREGSET attempt will randomly succeed or fail depending on the value of the uninitialized value, and the exposure is significantly limited. Fix this by initializing the temporary value before copying the regset from userspace, as for other regsets (e.g. NT_PRSTATUS, NT_PRFPREG, NT_ARM_SYSTEM_CALL). In the case of a zero-length write, the existing value of the tagged address ctrl will be retained. The NT_ARM_TAGGED_ADDR_CTRL regset is only visible in the user_aarch64_view used by a native AArch64 task to manipulate another native AArch64 task. As get_tagged_addr_ctrl() only returns an error value when called for a compat task, tagged_addr_ctrl_get() and tagged_addr_ctrl_set() should never observe an error value from get_tagged_addr_ctrl(). Add a WARN_ON_ONCE() to both to indicate that such an error would be unexpected, and error handling is not missing in either case.

CVE ID	Description
CVE-2024-57875	In the Linux kernel, the following vulnerability has been resolved: block: RCU protect disk->conv_zones_bitmap Ensure that a disk revalidation changing the conventional zones bitmap of a disk does not cause invalid memory references when using the disk_zone_is_conv() helper by RCU protecting the disk->conv_zones_bitmap pointer. disk_zone_is_conv() is modified to operate under the RCU read lock and the function disk_set_conv_zones_bitmap() is added to update a disk conv_zones_bitmap pointer using rcu_replace_pointer() with the disk zone_wplugs_lock spinlock held. disk_free_zone_resources() is modified to call disk_update_zone_resources() with a NULL bitmap pointer to free the disk conv_zones_bitmap. disk_set_conv_zones_bitmap() is also used in disk_update_zone_resources() to set the new (revalidated) bitmap and free the old one.
CVE-2024-57876	In the Linux kernel, the following vulnerability has been resolved: drm/dp_mst: Fix resetting msg rx state after topology removal If the MST topology is removed during the reception of an MST down reply or MST up request sideband message, the drm_dp_mst_topology_mgr::up_req_rcv/down_rep_rcv states could be reset from one thread via drm_dp_mst_topology_mgr_set_mst(false), racing with the reading/parsing of the message from another thread via drm_dp_mst_handle_down_rep() or drm_dp_mst_handle_up_req(). The race is possible since the reader/parser doesn't hold any lock while accessing the reception state. This in turn can lead to a memory corruption in the reader/parser as described by commit bd2fccac61b4 ("drm/dp_mst: Fix MST sideband message body length check"). Fix the above by resetting the message reception state if needed before reading/parsing a message. Another solution would be to hold the drm_dp_mst_topology_mgr::lock for the whole duration of the message reception/parsing in drm_dp_mst_handle_down_rep() and drm_dp_mst_handle_up_req(), however this would require a bigger change. Since the fix is also needed for stable, opting for the simpler solution in this patch.
CVE-2024-57879	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: iso: Always release hdev at the end of iso_listen_bis Since hci_get_route holds the device before returning, the hdev should be released with hci_dev_put at the end of iso_listen_bis even if the function returns with an error.

CVE ID	Description
CVE-2024-57882	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix TCP options overflow. Syzbot reported the following splat: Oops: general protection fault, probably for non-canonical address 0xdffffc0000000001: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] CPU: 1 UID: 0 PID: 5836 Comm: sshd Not tainted 6.13.0-rc3-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 11/25/2024 RIP: 0010:compound_head include/linux/page-flags.h:242 [inline] RIP: 0010:put_page+0x23/0x260 include/linux/mm.h:1552 Code: 90 90 90 90 90 90 55 41 57 41 56 53 49 89 fe 48 bd 00 00 00 00 00 ff df e8 f8 5e 12 f8 49 8d 5e 08 48 89 d8 48 c1 e8 03 <80> 3c 28 00 74 08 48 89 df e8 8f c7 78 f8 48 8b 1b 48 89 de 48 83 RSP: 0000:ffffc90003916c90 EFLAGS: 00010202 RAX: 0000000000000001 RBX: 0000000000000008 RCX: ffff888030458000 RDX: 0000000000000100 RSI: 0000000000000000 RDI: 0000000000000000 RBP: dffffc0000000000 R08: ffffffff898ca81d R09: 1ffff110054414ac R10: dffffc0000000000 R11: ffffed10054414ad R12: 0000000000000007 R13: ffff88802a20a542 R14: 0000000000000000 R15: 0000000000000000 FS: 00007f34f496e800(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f9d6ec9ec28 CR3: 000000004d260000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> skb_page_unref include/linux/skbuff_ref.h:43 [inline] __skb_frag_unref include/linux/skbuff_ref.h:56 [inline] skb_release_data+0x483/0x8a0 net/core/skbuff.c:1119 skb_release_all net/core/skbuff.c:1190 [inline] __kfree_skb+0x55/0x70 net/core/skbuff.c:1204 tcp_clean_rtx_queue net/ipv4/tcp_input.c:3436 [inline] tcp_ack+0x2442/0x6bc0 net/ipv4/tcp_input.c:4032 tcp_rcv_state_process+0x8eb/0x44e0 net/ipv4/tcp_input.c:6805 tcp_v4_do_rcv+0x77d/0xc70 net/ipv4/tcp_ipv4.c:1939 tcp_v4_rcv+0x2dc0/0x37f0 net/ipv4/tcp_ipv4.c:2351 ip_protocol_deliver_rcu+0x22e/0x440 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x341/0x5f0 net/ipv4/ip_input.c:233 NF_HOOK+0x3a4/0x450 include/linux/netfilter.h:314 NF_HOOK+0x3a4/0x450 include/linux/netfilter.h:314 __netif_receive_skb_one_core net/core/dev.c:5672 [inline] __netif_receive_skb+0x2bf/0x650 net/core/dev.c:5785 process_backlog+0x662/0x15b0 net/core/dev.c:6117 __napi_poll+0xcb/0x490 net/core/dev.c:6883 napi_poll net/core/dev.c:6952 [inline] net_rx_action+0x89b/0x1240 net/core/dev.c:7074 handle_softirqs+0x2d4/0x9b0 kernel/softirq.c:561 __do_softirq kernel/softirq.c:595 [inline] invoke_softirq kernel/softirq.c:435 [inline] __irq_exit_rcu+0xf7/0x220 kernel/softirq.c:662 irq_exit_rcu+0x9/0x30 kernel/softirq.c:678 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1049 [inline] sysvec_apic_timer_interrupt+0x57/0xc0 arch/x86/kernel/apic/apic.c:1049 asm_sysvec_apic_timer_interrupt+0x1a/0x20 arch/x86/include/asm/identrty.h:702 RIP: 0033:0x7f34f4519ad5 Code: 85 d2 74 0d 0f 10 02 48 8d 54 24 20 0f 11 44 24 20 64 8b 04 25 18 00 00 00 85 c0 75 27 41 b8 08 00 00 00 b8 0f 01 00 00 0f 05 <48> 3d 00 f0 ff ff 76 75 48 8b 15 24 73 0d 00 f7 d8 64 89 02 48 83 RSP: 002b:00007ffec5b32ce0 EFLAGS: 00000246 RAX: 0000000000000001 RBX: 00000000000668a0 RCX: 00007f34f4519ad5 RDX: 00007ffec5b32d00 RSI: 0000000000000004 RDI: 0000564f4bc6cae0 RBP: 0000564f4bc6b5a0 R08: 0000000000000008 R09: 0000000000000000 R10: 00007ffec5b32de8 R11: 0000000000000246 R12: 0000564f48ea8aa4 R13: 0000000000000001 R14: 0000564f48ea93e8 R15: 00007ffec5b32d68 </TASK> Eric noted a probable shinfo->nr_frags corruption, which indeed occurs. The root cause is a buggy MPTCP option len computation in some circumstances: the ADD_ADDR option should be mutually exclusive with DSS since the blamed commit. Still, mptcp_established_options_add_addr() tries to set the relevant info in mptcp_out_options, if ---truncated---
CVE-2024-57883	In the Linux kernel, the following vulnerability has been resolved: mm: hugetlb: independent PMD page table shared count The folio refcount may be increased unexpectedly through try_get_folio() by caller such as split_huge_pages. In huge_pmd_unshare(), we use refcount to check whether a pmd page table is shared. The check is incorrect if the refcount is increased by the above caller, and this can cause the page table leaked: BUG: Bad page state in process sh pfn:109324 page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x66 pfn:0x109324 flags: 0x17ffff8000000000(node=0 zone=2 lastcpupid=0xfffff) page_type: f2(table) raw: 017ffff800000000 0000000000000000 0000000000000000 0000000000000066 0000000000000000 00000000f2000000 0000000000000000 page dumped because: nonzero mapcount ... CPU: 31 UID: 0 PID: 7515 Comm: sh Kdump: loaded Tainted: G B 6.13.0-rc2master+ #7 Tainted: [B]=BAD_PAGE Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 Call trace: show_stack+0x20/0x38 (C) dump_stack_lvl+0x80/0xf8 dump_stack+0x18/0x28 bad_page+0x8c/0x130 free_page_is_bad_report+0xa4/0xb0 free_unref_page+0x3cc/0x620 __folio_put+0xf4/0x158 split_huge_pages_all+0x1e0/0x3e8 split_huge_pages_write+0x25c/0x2d8 full_proxy_write+0x64/0xd8 vfs_write+0xcc/0x280 ksys_write+0x70/0x110 __arm64_sys_write+0x24/0x38 invoke_syscall+0x50/0x120 el0_svc_common.constprop.0+0xc8/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x34/0x128 el0t_64_sync_handler+0xc8/0xd0 el0t_64_sync+0x190/0x198 The issue may be triggered by damon, offline_page, page_idle, etc, which will increase the refcount of page table. 1. The page table itself will be discarded after reporting the "nonzero mapcount". 2. The HugeTLB page mapped by the page table miss freeing since we treat the page table as shared and a shared page table will not be unmapped. Fix it by introducing independent PMD page table shared count. As described by comment, pt_index/pt_mm/pt_frag_refcount are used for s390 gmap, x86 pgds and powerpc, pt_share_count is used for x86/arm64/riscv pmds, so we can reuse the field as pt_share_count.

CVE ID	Description
CVE-2024-57884	In the Linux kernel, the following vulnerability has been resolved: mm: vmscan: account for free pages to prevent infinite Loop in throttle_direct_reclaim() The task sometimes continues looping in throttle_direct_reclaim() because allow_direct_reclaim(pgdat) keeps returning false. #0 [ffff80002cb6f8d0] __switch_to at ffff8000080095ac #1 [ffff80002cb6f900] __schedule at ffff800008abb1c #2 [ffff80002cb6f990] schedule at ffff800008abc50c #3 [ffff80002cb6f9b0] throttle_direct_reclaim at ffff800008273550 #4 [ffff80002cb6fa20] try_to_free_pages at ffff800008277b68 #5 [ffff80002cb6fae0] __alloc_pages_nodemask at ffff8000082c4660 #6 [ffff80002cb6fc50] alloc_pages_vma at ffff8000082e4a98 #7 [ffff80002cb6fca0] do_anonymous_page at ffff80000829f5a8 #8 [ffff80002cb6fce0] __handle_mm_fault at ffff8000082a5974 #9 [ffff80002cb6fd90] handle_mm_fault at ffff8000082a5bd4 At this point, the pgdat contains the following two zones: NODE: 4 ZONE: 0 ADDR: ffff00817ffe540 NAME: "DMA32" SIZE: 20480 MIN/LOW/HIGH: 11/28/45 VM_STAT: NR_FREE_PAGES: 359 NR_ZONE_INACTIVE_ANON: 18813 NR_ZONE_ACTIVE_ANON: 0 NR_ZONE_INACTIVE_FILE: 50 NR_ZONE_ACTIVE_FILE: 0 NR_ZONE_UNEVICTABLE: 0 NR_ZONE_WRITE_PENDING: 0 NR_MLOCK: 0 NR_BOUNCE: 0 NR_ZSPAGES: 0 NR_FREE_CMA_PAGES: 0 NODE: 4 ZONE: 1 ADDR: ffff00817ffec00 NAME: "Normal" SIZE: 8454144 PRESENT: 98304 MIN/LOW/HIGH: 68/166/264 VM_STAT: NR_FREE_PAGES: 146 NR_ZONE_INACTIVE_ANON: 94668 NR_ZONE_ACTIVE_ANON: 3 NR_ZONE_INACTIVE_FILE: 735 NR_ZONE_ACTIVE_FILE: 78 NR_ZONE_UNEVICTABLE: 0 NR_ZONE_WRITE_PENDING: 0 NR_MLOCK: 0 NR_BOUNCE: 0 NR_ZSPAGES: 0 NR_FREE_CMA_PAGES: 0 In allow_direct_reclaim(), while processing ZONE_DMA32, the sum of inactive/active file-backed pages calculated in zone_reclaimable_pages() based on the result of zone_page_state_snapshot() is zero. Additionally, since this system lacks swap, the calculation of inactive/ active anonymous pages is skipped. crash> p nr_swap_pages nr_swap_pages = \$1937 = { counter = 0 } As a result, ZONE_DMA32 is deemed unreclaimable and skipped, moving on to the processing of the next zone, ZONE_NORMAL, despite ZONE_DMA32 having free pages significantly exceeding the high watermark. The problem is that the pgdat->kswapd_failures hasn't been incremented. crash> px ((struct pglist_data *) 0xffff00817ffe540)->kswapd_failures \$1935 = 0x0 This is because the node deemed balanced. The node balancing logic in balance_pgdat() evaluates all zones collectively. If one or more zones (e.g., ZONE_DMA32) have enough free pages to meet their watermarks, the entire node is deemed balanced. This causes balance_pgdat() to exit early before incrementing the kswapd_failures, as it considers the overall memory state acceptable, even though some zones (like ZONE_NORMAL) remain under significant pressure. The patch ensures that zone_reclaimable_pages() includes free pages (NR_FREE_PAGES) in its calculation when no other reclaimable pages are available (e.g., file-backed or anonymous pages). This change prevents zones like ZONE_DMA32, which have sufficient free pages, from being mistakenly deemed unreclaimable. By doing so, the patch ensures proper node balancing, avoids masking pressure on other zones like ZONE_NORMAL, and prevents infinite loops in throttle_direct_reclaim() caused by allow_direct_reclaim(pgdat) repeatedly returning false. The kernel hangs due to a task stuck in throttle_direct_reclaim(), caused by a node being incorrectly deemed balanced despite pressure in certain zones, such as ZONE_NORMAL. This issue arises from zone_reclaimable_pages ---truncated---

CVE ID	Description
CVE-2024-57885	In the Linux kernel, the following vulnerability has been resolved: mm/kmemleak: fix sleeping function called from invalid context at print message Address a bug in the kernel that triggers a "sleeping function called from invalid context" warning when /sys/kernel/debug/kmemleak is printed under specific conditions: - CONFIG_PREEMPT_RT=y - Set SELinux as the LSM for the system - Set kptr_restrict to 1 - kmemleak buffer contains at least one item BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 136, name: cat preempt_count: 1, expected: 0 RCU nest depth: 2, expected: 2 6 locks held by cat/136: #0: ffff32e64bcbf950 (&p->lock){+..}-{3:3}, at: seq_read_iter+0xb8/0xe30 #1: ffffaf6aa9dea0 (scan_mutex){+..}-{3:3}, at: kmemleak_seq_start+0x34/0x128 #3: ffff32e6546b1cd0 (&object->lock){...}-{2:2}, at: kmemleak_seq_show+0x3c/0x1e0 #4: ffffaf6aa8d8560 (rcu_read_lock){....}-{1:2}, at: has_ns_capability_noaudit+0x8/0x1b0 #5: ffffaf6aabb0cf8 (notif_lock){+..}-{2:2}, at: avc_compute_av+0xc4/0x3d0 irq event stamp: 136660 hardirqs last enabled at (136659): [<ffffaf6a80fd7a0>] _raw_spin_unlock_irqrestore+0xa8/0xd8 hardirqs last disabled at (136660): [<ffffaf6a80fd85c>] _raw_spin_lock_irqsave+0x8c/0xb0 softirqs last enabled at (0): [<ffffaf6a5d50b28>] copy_process+0x11d8/0x3df8 softirqs last disabled at (0): [<0000000000000000>] 0x0 Preemption disabled at: [<ffffaf6a6598a4c>] kmemleak_seq_show+0x3c/0x1e0 CPU: 1 UID: 0 PID: 136 Comm: cat Tainted: G E 6.11.0-rt7+ #34 Tainted: [E]=UNSIGNED_MODULE Hardware name: linux,dummy-virt (DT) Call trace: dump_backtrace+0xa0/0x128 show_stack+0x1c/0x30 dump_stack_lvl+0xe8/0x198 dump_stack+0x18/0x20 rt_spin_lock+0x8c/0x1a8 avc_perm_nnode+0xa0/0x150 cred_has_capability.isra.0+0x118/0x218 selinux_capable+0x50/0x80 security_capable+0x7c/0xd0 has_ns_capability_noaudit+0x94/0x1b0 has_capability_noaudit+0x20/0x30 restricted_pointer+0x21c/0x4b0 pointer+0x298/0x760 vsnprintf+0x330/0xf70 seq_printf+0x178/0x218 print_unreferenced+0x1a4/0x2d0 kmemleak_seq_show+0xd0/0x1e0 seq_read_iter+0x354/0xe30 seq_read+0x250/0x378 full_proxy_read+0xd8/0x148 vfs_read+0x190/0x918 ksys_read+0xf0/0x1e0 __arm64_sys_read+0x70/0xa8 invoke_syscall.constprop.0+0xd4/0x1d8 el0_svc+0x50/0x158 el0t_64_sync+0x17c/0x180 %pS and %pK, in the same back trace line, are redundant, and %pS can void %pK service in certain contexts. %pS alone already provides the necessary information, and if it cannot resolve the symbol, it falls back to printing the raw address voiding the original intent behind the %pK. Additionally, %pK requires a privilege check CAP_SYSLOG enforced through the LSM, which can trigger a "sleeping function called from invalid context" warning under RT_PREEMPT kernels when the check occurs in an atomic context. This issue may also affect other LSMs. This change avoids the unnecessary privilege check and resolves the sleeping function warning without any loss of information.
CVE-2024-57887	In the Linux kernel, the following vulnerability has been resolved: drm: adv7511: Fix use-after-free in adv7533_attach_dsi() The host_node pointer was assigned and freed in adv7533_parse_dt(), and later, adv7533_attach_dsi() uses the same. Fix this use-after-free issue by dropping of_node_put() in adv7533_parse_dt() and calling of_node_put() in error path of probe() and also in the remove().
CVE-2024-57888	In the Linux kernel, the following vulnerability has been resolved: workqueue: Do not warn when cancelling WQ_MEM_RECLAIM work from !WQ_MEM_RECLAIM worker After commit 746ae46c1113 ("drm/sched: Mark scheduler work queues with WQ_MEM_RECLAIM") amdgpu started seeing the following warning: [] workqueue: WQ_MEM_RECLAIM sdma0:drm_sched_run_job_work [gpu_sched] is flushing !WQ_MEM_RECLAIM events:amdgpu_device_delay_enable_gfx_off [amdgpu] ... [] Workqueue: sdma0 drm_sched_run_job_work [gpu_sched] ... [] Call Trace: [] <TASK> ... [] ? check_flush_dependency+0xf5/0x110 ... [] cancel_delayed_work_sync+0x6e/0x80 [] amdgpu_gfx_off_ctrl+0xab/0x140 [amdgpu] [] amdgpu_ring_alloc+0x40/0x50 [amdgpu] [] amdgpu_ib_schedule+0xf4/0x810 [amdgpu] [] ? drm_sched_run_job_work+0x22c/0x430 [gpu_sched] [] amdgpu_job_run+0xaa/0x1f0 [amdgpu] [] drm_sched_run_job_work+0x257/0x430 [gpu_sched] [] process_one_work+0x217/0x720 ... [] </TASK> The intent of the verification done in check_flush_dependency is to ensure forward progress during memory reclaim, by flagging cases when either a memory reclaim process, or a memory reclaim work item is flushed from a context not marked as memory reclaim safe. This is correct when flushing, but when called from the cancel(_delayed)_work_sync() paths it is a false positive because work is either already running, or will not be running at all. Therefore cancelling it is safe and we can relax the warning criteria by letting the helper know of the calling context. References: 746ae46c1113 ("drm/sched: Mark scheduler work queues with WQ_MEM_RECLAIM")

CVE ID	Description
CVE-2024-57889	In the Linux kernel, the following vulnerability has been resolved: pinctrl: mcp23s08: Fix sleeping in atomic context due to regmap locking If a device uses MCP23xxx IO expander to receive IRQs, the following bug can happen: BUG: sleeping function called from invalid context at kernel/locking/mutex.c:283 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, ... preempt_count: 1, expected: 0 ... Call Trace: ... __might_resched+0x104/0x10e __might_sleep+0x3e/0x62 mutex_lock+0x20/0x4c regmap_lock_mutex+0x10/0x18 regmap_update_bits_base+0x2c/0x66 mcp23s08_irq_set_type+0x1ae/0x1d6 __irq_set_trigger+0x56/0x172 __setup_irq+0x1e6/0x646 request_threaded_irq+0xb6/0x160 ... We observed the problem while experimenting with a touchscreen driver which used MCP23017 IO expander (I2C). The regmap in the pinctrl-mcp23s08 driver uses a mutex for protection from concurrent accesses, which is the default for regmaps without .fast_io, .disable_locking, etc. mcp23s08_irq_set_type() calls regmap_update_bits_base(), and the latter locks the mutex. However, __setup_irq() locks desc->lock spinlock before calling these functions. As a result, the system tries to lock the mutex while holding the spinlock. It seems, the internal regmap locks are not needed in this driver at all. mcp->lock seems to protect the regmap from concurrent accesses already, except, probably, in mcp_pinconf_get/set. mcp23s08_irq_set_type() and mcp23s08_irq_mask/unmask() are called under chip_bus_lock(), which calls mcp23s08_irq_bus_lock(). The latter takes mcp->lock and enables regmap caching, so that the potentially slow I2C accesses are deferred until chip_bus_unlock(). The accesses to the regmap from mcp23s08_probe_one() do not need additional locking. In all remaining places where the regmap is accessed, except mcp_pinconf_get/set(), the driver already takes mcp->lock. This patch adds locking in mcp_pinconf_get/set() and disables internal locking in the regmap config. Among other things, it fixes the sleeping in atomic context described above.
CVE-2024-57890	In the Linux kernel, the following vulnerability has been resolved: RDMA/uverbs: Prevent integer overflow issue In the expression "cmd.wqe_size * cmd.wr_count", both variables are u32 values that come from the user so the multiplication can lead to integer wrapping. Then we pass the result to uverbs_request_next_ptr() which also could potentially wrap. The "cmd.sge_count * sizeof(struct ib_uverbs_sge)" multiplication can also overflow on 32bit systems although it's fine on 64bit systems. This patch does two things. First, I've re-arranged the condition in uverbs_request_next_ptr() so that the use controlled variable "len" is on one side of the comparison by itself without any math. Then I've modified all the callers to use size_mul() for the multiplications.
CVE-2024-57892	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix slab-use-after-free due to dangling pointer dq_i_priv When mounting ocfs2 and then remounting it as read-only, a slab-use-after-free occurs after the user uses a syscall to quota_getnextquota. Specifically, sb_dqinfo(sb, type)->dq_i_priv is the dangling pointer. During the remounting process, the pointer dq_i_priv is freed but is never set as null leaving it to be accessed. Additionally, the read-only option for remounting sets the DQUOT_SUSPENDED flag instead of setting the DQUOT_USAGE_ENABLED flags. Moreover, later in the process of getting the next quota, the function ocfs2_get_next_id is called and only checks the quota usage flags and not the quota suspended flags. To fix this, I set dq_i_priv to null when it is freed after remounting with read-only and put a check for DQUOT_SUSPENDED in ocfs2_get_next_id. [akpm@linux-foundation.org: coding-style cleanups]
CVE-2024-57893	In the Linux kernel, the following vulnerability has been resolved: ALSA: seq: oss: Fix races at processing SysEx messages OSS sequencer handles the SysEx messages split in 6 bytes packets, and ALSA sequencer OSS layer tries to combine those. It stores the data in the internal buffer and this access is racy as of now, which may lead to the out-of-bounds access. As a temporary band-aid fix, introduce a mutex for serializing the process of the SysEx message packets.

CVE ID	Description
CVE-2024-57895	In the Linux kernel, the following vulnerability has been resolved: ksmbd: set ATTR_CTIME flags when setting mtime David reported that the new warning from setattr_copy_mtime is coming like the following. [113.215316] -----[cut here]----- [113.215974] WARNING: CPU: 1 PID: 31 at fs/attr.c:300 setattr_copy+0x1ee/0x200 [113.219192] CPU: 1 UID: 0 PID: 31 Comm: kworker/1:1 Not tainted 6.13.0-rc1+ #234 [113.220127] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.2-3-gd478f380-rebuilt.opensuse.org 04/01/2014 [113.221530] Workqueue: ksmbd-io handle_ksmbd_work [ksmbd] [113.222220] RIP: 0010:setattr_copy+0x1ee/0x200 [113.222833] Code: 24 28 49 8b 44 24 30 48 89 53 58 89 43 6c 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc 48 89 df e8 77 d6 ff ff e9 cd fe ff ff <0f> 0b e9 be fe ff ff 66 0 [113.225110] RSP: 0018:ffffaf218010fb68 EFLAGS: 00010202 [113.225765] RAX: 0000000000000120 RBX: ffffa446815f8568 RCX: 0000000000000003 [113.226667] RDX: ffffaf218010fd38 RSI: ffffa446815f8568 RDI: ffffffff94eb03a0 [113.227531] RBP: ffffaf218010fb90 R08: 00000001a251e217d R09: 00000000675259fa [113.228426] R10: 0000000002ba8a6d R11: ffffa4468196c7a8 R12: ffffaf218010fd38 [113.229304] R13: 0000000000000120 R14: ffffffff94eb03a0 R15: 0000000000000000 [113.230210] FS: 0000000000000000(0000) GS:ffffa44739d00000(0000) knlGS:0000000000000000 [113.231215] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [113.232055] CR2: 00007efe0053d27e CR3: 000000000331a000 CR4: 00000000000006b0 [113.232926] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [113.233812] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 [113.234797] Call Trace: [113.235116] <TASK> [113.235393] ? __warn+0x73/0xd0 [113.235802] ? setattr_copy+0x1ee/0x200 [113.236299] ? report_bug+0xf3/0x1e0 [113.236757] ? handle_bug+0x4d/0x90 [113.237202] ? exc_invalid_op+0x13/0x60 [113.237689] ? asm_exc_invalid_op+0x16/0x20 [113.238185] ? setattr_copy+0x1ee/0x200 [113.238692] btrfs_setattr+0x80/0x820 [btrfs] [113.239285] ? get_stack_info_noinstr+0x12/0xf0 [113.239857] ? __module_address+0x22/0xa0 [113.240368] ? handle_ksmbd_work+0x6e/0x460 [ksmbd] [113.240993] ? __module_text_address+0x9/0x50 [113.241545] ? __module_address+0x22/0xa0 [113.242033] ? unwind_next_frame+0x10e/0x920 [113.242600] ? __pfx_stack_trace_consume_entry+0x10/0x10 [113.243268] notify_change+0x2c2/0x4e0 [113.243746] ? stack_depote_save_flags+0x27/0x730 [113.244339] ? set_file_basic_info+0x130/0x2b0 [ksmbd] [113.244993] set_file_basic_info+0x130/0x2b0 [ksmbd] [113.245613] ? process_scheduled_works+0xbe/0x310 [113.246181] ? worker_thread+0x100/0x240 [113.246696] ? kthread+0xc8/0x100 [113.247126] ? ret_from_fork+0x2b/0x40 [113.247606] ? ret_from_fork_asm+0x1a/0x30 [113.248132] smb2_set_info+0x63f/0xa70 [ksmbd] ksmbd is trying to set the atime and mtime via notify_change without also setting the ctime. so This patch add ATTR_CTIME flags when setting mtime to avoid a warning.

CVE ID	Description
CVE-2024-57896	In the Linux kernel, the following vulnerability has been resolved: btrfs: flush delalloc workers queue before stopping cleaner kthread during unmount During the unmount path, at close_tree(), we first stop the cleaner kthread, using kthread_stop() which frees the associated task_struct, and then stop and destroy all the work queues. However after we stopped the cleaner we may still have a worker from the delalloc_workers queue running inode.c:submit_compressed_extents(), which calls btrfs_add_delayed_iput(), which in turn tries to wake up the cleaner kthread - which was already destroyed before, resulting in a use-after-free on the task_struct. Syzbot reported this with the following stack traces: BUG: KASAN: slab-use-after-free in __lock_acquire+0x78/0x2100 kernel/locking/lockdep.c:5089 Read of size 8 at addr ffff8880259d2818 by task kworker/u8:3/52 CPU: 1 UID: 0 PID: 52 Comm: kworker/u8:3 Not tainted 6.13.0-rc1-syzkaller-00002-gcdd30ebb1b9f #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: btrfs-delalloc btrfs_work_helper Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report+0x143/0x180 mm/kasan/report.c:602 __lock_acquire+0x78/0x2100 kernel/locking/lockdep.c:5089 lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5849 __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:110 [inline] _raw_spin_lock_irqsave+0xd5/0x120 kernel/locking/spinlock.c:162 class_raw_spinlock_irqsave_constructor include/linux/spinlock.h:551 [inline] try_to_wake_up+0xc2/0x1470 kernel/sched/core.c:4205 submit_compressed_extents+0xdf/0x16e0 fs/btrfs/inode.c:1615 run_ordered_work fs/btrfs/async-thread.c:288 [inline] btrfs_work_helper+0x96f/0xc40 fs/btrfs/async-thread.c:324 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa66/0x1840 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Allocated by task 2: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 unpoison_slab_object mm/kasan/common.c:319 [inline] __kasan_slab_alloc+0x66/0x80 mm/kasan/common.c:345 kasan_slab_alloc include/linux/kasan.h:250 [inline] slab_post_alloc_hook mm/slub.c:4104 [inline] slab_alloc_node mm/slub.c:4153 [inline] kmem_cache_alloc_node_noprof+0x1d9/0x380 mm/slub.c:4205 alloc_task_struct_node kernel/fork.c:180 [inline] dup_task_struct+0x57/0x8c0 kernel/fork.c:1113 copy_process+0x5d1/0x3d50 kernel/fork.c:2225 kernel_clone+0x223/0x870 kernel/fork.c:2807 kernel_thread+0x1bc/0x240 kernel/fork.c:2869 create_kthread kernel/kthread.c:412 [inline] kthreadd+0x60d/0x810 kernel/kthread.c:767 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 Freed by task 24: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:582 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x59/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2338 [inline] slab_free mm/slub.c:4598 [inline] kmem_cache_free+0x195/0x410 mm/slub.c:4700 put_task_struct include/linux/sched/task.h:144 [inline] delayed_put_task_struct+0x125/0x300 kernel/exit.c:227 rcu_do_batch kernel/rcu/tree.c:2567 [inline] rcu_core+0xaaa/0x17a0 kernel/rcu/tree.c:2823 handle_softirqs+0x2d4/0x9b0 kernel/softirq.c:554 run_ksoftirqd+0xca/0x130 kernel/softirq.c:943 ---truncated---

CVE ID	Description
CVE-2024-57897	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Correct the migration DMA map direction The SVM DMA device map direction should be set the same as the DMA unmap setting, otherwise the DMA core will report the following warning. Before finalize this solution, there're some discussion on the DMA mapping type(stream-based or coherent) in this KFD migration case, followed by https://lore.kernel.org/all/04d4ab32-45a1-4b88-86ee-fb0f35a0ca40@amd.com/T/. As there's no dma_sync_single_for_*(*) in the DMA buffer accessed that because this migration operation should be sync properly and automatically. Give that there's might not be a performance problem in various cache sync policy of DMA sync. Therefore, in order to simplify the DMA direction setting alignment, let's set the DMA map direction as BIDIRECTIONAL. [150.834218] WARNING: CPU: 8 PID: 1812 at kernel/dma/debug.c:1028 check_unmap+0x1cc/0x930 [150.834225] Modules linked in: amdgpu(OE) amdxcp drm_exec(OE) gpu_sched drm_buddy(OE) drm_ttm_helper(OE) ttm(OE) drm_suballoc_helper(OE) drm_display_helper(OE) drm_kms_helper(OE) i2c_algo_bit rpcsec_gss_krb5 auth_rpcgss nfsv4 nfs lockd grace netfs xt_contrack xt_MASQUERADE nf_contrack_netlink xfrm_user xfrm_algo iptable_nat xt_addrtype iptable_filter br_netfilter nvme_fabrics overlay nfnetlink_cttimeout nfnetlink openvswitch nsh nf_conntrack nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 libcrc32c bridge stp llc sch_fq_codel intel_rapl_msr amd_atl intel_rapl_common snd_hda_codec_realtek snd_hda_codec_generic snd_hda_scodec_component snd_hda_codec_hdmi snd_hda_intel snd_intel_dspcfg edac_mce_amd snd_pci_acp6x snd_hda_codec snd_acp_config snd_hda_core snd_hwdep snd_soc_acpi kvm_amd sunrpc snd_pcm kvm binfmt_misc snd_seq_midi crct10dif_pclmul snd_seq_midi_event ghash_clmulni_intel sha512_ssse3 snd_rawmidi nls_iso8859_1 sha256_ssse3 sha1_ssse3 snd_seq_aesni_intel snd_seq_device crypto_simd snd_timer cryptd input_leds [150.834310] wmi_bmof serio_raw k10temp rapl snd_sp5100_tco ipmi_devintf soundcore ccp ipmi_msghandler cm32181 industrialio mac_hid msr parport_pc ppdev lp parport efi_pstore drm(OE) ip_tables x_tables pci_stub crc32_pclmul nvme ahci libahci i2c_piix4 r8169 nvme_core i2c_designware_pci realtek i2c_ccgx_ucsi video wmi hid_generic cdc_ether usbnet usbbid hid_r8152 mii [150.834354] CPU: 8 PID: 1812 Comm: rocrst64 Tainted: G OE 6.10.0-custom #492 [150.834358] Hardware name: AMD Majolica-RN/Majolica-RN, BIOS RMJ1009A 06/13/2021 [150.834360] RIP: 0010:check_unmap+0x1cc/0x930 [150.834363] Code: c0 4c 89 4d c8 e8 34 bf 86 00 4c 8b 4d c8 4c 8b 45 c0 48 8b 4d b8 48 89 c6 41 57 4c 89 ea 48 c7 c7 80 49 b4 84 e8 b4 81 f3 ff <0f> 0b 48 c7 c7 04 83 ac 84 e8 76 ba fc ff 41 8b 76 4c 49 8d 7e 50 [150.834365] RSP: 0018:ffffaac5023739e0 EFLAGS: 00010086 [150.834368] RAX: 0000000000000000 RBX: ffffffff8566a2e0 RCX: 0000000000000027 [150.834370] RDX: ffff8f6a8f621688 RSI: 0000000000000001 RDI: ffff8f6a8f621680 [150.834372] RBP: fffffaac502373a30 R08: 00000000000000c9 R09: fffffaac502373850 [150.834373] R10: fffffaac502373848 R11: ffffffff84f46328 R12: fffffaac502373a40 [150.834375] R13: ffff8f6741045330 R14: ffff8f6741a77700 R15: ffffffff84ac831b [150.834377] FS: 00007faf0fc94c00(0000) GS: ffff8f6a8f600000(0000) knlGS: 0000000000000000 [150.834379] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [150.834381] CR2: 00007faf0b600020 CR3: 000000010a52e000 CR4: 0000000000350ef0 [150.834383] Call Trace: [150.834385] <TASK> [150.834387] ? show_regs+0x6d/0x80 [150.834393] ? __warn+0x8c/0x140 [150.834397] ? check_unmap+0x1cc/0x930 [150.834400] ? report_bug+0x193/0x1a0 [150.834406] ? handle_bug+0x46/0x80 [150.834410] ? exc_invalid_op+0x1d/0x80 [150.834413] ? asm_exc_invalid_op+0x1f/0x30 [150.834420] ? check_unmap+0x1cc/0x930 [150.834425] debug_dma_unmap_page+0x86/0x90 [150.834431] ? srso_return_thunk+0x5/0x5f [150.834435] ---truncated---
CVE-2024-57898	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: clear link ID from bitmap during link delete after clean up Currently, during link deletion, the link ID is first removed from the valid_links bitmap before performing any clean-up operations. However, some functions require the link ID to remain in the valid_links bitmap. One such example is cfg80211_cac_event(). The flow is - nl80211_remove_link() cfg80211_remove_link() ieee80211_del_intf_link() ieee80211_vif_set_links() ieee80211_vif_update_links() ieee80211_link_stop() cfg80211_cac_event() cfg80211_cac_event() requires link ID to be present but it is cleared already in cfg80211_remove_link(). Ultimately, WARN_ON() is hit. Therefore, clear the link ID from the bitmap only after completing the link clean-up.
CVE-2024-57899	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: fix mbss changed flags corruption on 32 bit systems On 32-bit systems, the size of an unsigned long is 4 bytes, while a u64 is 8 bytes. Therefore, when using or_each_set_bit(bit, &bits, sizeof(changed) * BITS_PER_BYTE), the code is incorrectly searching for a bit in a 32-bit variable that is expected to be 64 bits in size, leading to incorrect bit finding. Solution: Ensure that the size of the bits variable is correctly adjusted for each architecture. Call Trace: ? show_regs+0x54/0x58 ? __warn+0x6b/0xd4 ? ieee80211_link_info_change_notify+0xcc/0xd4 [mac80211] ? report_bug+0x113/0x150 ? exc_overflow+0x30/0x30 ? handle_bug+0x27/0x44 ? exc_invalid_op+0x18/0x50 ? handle_exception+0xf6/0xf6 ? exc_overflow+0x30/0x30 ? ieee80211_link_info_change_notify+0xcc/0xd4 [mac80211] ? exc_overflow+0x30/0x30 ? ieee80211_link_info_change_notify+0xcc/0xd4 [mac80211] ? ieee80211_mesh_work+0xff/0x260 [mac80211] ? cfg80211_wiphy_work+0x72/0x98 [cfg80211] ? process_one_work+0xf1/0x1fc ? worker_thread+0x2c0/0x3b4 ? kthread+0xc7/0xf0 ? mod_delayed_work_on+0x4c/0x4c ? kthread_complete_and_exit+0x14/0x14 ? ret_from_fork+0x24/0x38 ? kthread_complete_and_exit+0x14/0x14 ? ret_from_fork_asm+0xf/0x14 ? entry_INT80_32+0xf0/0xf0 [restore no-op path for no changes]

CVE ID	Description
CVE-2024-57900	In the Linux kernel, the following vulnerability has been resolved: ila: serialize calls to nf_register_net_hooks() syzbot found a race in ila_add_mapping() [1] commit 031ae72825ce ("ila: call nf_unregister_net_hooks() sooner") attempted to fix a similar issue. Looking at the syzbot repro, we have concurrent ILA_CMD_ADD commands. Add a mutex to make sure at most one thread is calling nf_register_net_hooks(). [1] BUG: KASAN: slab-use-after-free in rht_key_hashfn include/linux/rhashtable.h:159 [inline] BUG: KASAN: slab-use-after-free in __rhashtable_lookup.constprop.0+0x426/0x550 include/linux/rhashtable.h:604 Read of size 4 at addr ffff888028f40008 by task dhcpcd/5501 CPU: 1 UID: 0 PID: 5501 Comm: dhcpcd Not tainted 6.13.0-rc4-syzkaller-00054-gd6ef8b40d075 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <IRQ> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xc3/0x620 mm/kasan/report.c:489 kasan_report+0xd9/0x110 mm/kasan/report.c:602 rht_key_hashfn include/linux/rhashtable.h:159 [inline] __rhashtable_lookup.constprop.0+0x426/0x550 include/linux/rhashtable.h:604 rhashtable_lookup include/linux/rhashtable.h:646 [inline] rhashtable_lookup_fast include/linux/rhashtable.h:672 [inline] ila_lookup_wildcards net/ipv6/ila/ila_xlat.c:127 [inline] ila_xlat_addr net/ipv6/ila/ila_xlat.c:652 [inline] ila_nf_input+0x1ee/0x620 net/ipv6/ila/ila_xlat.c:185 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xbb/0x200 net/netfilter/core.c:626 nf_hook.constprop.0+0x42e/0x750 include/linux/netfilter.h:269 NF_HOOK include/linux/netfilter.h:312 [inline] ipv6_rcv+0xa4/0x680 net/ipv6/ipv6_input.c:309 __netif_receive_skb_one_core+0x12e/0x1e0 net/core/dev.c:5672 __netif_receive_skb+0x1d/0x160 net/core/dev.c:5785 process_backlog+0x443/0x15f0 net/core/dev.c:6117 __napi_poll.constprop.0+0xb7/0x550 net/core/dev.c:6883 napi_poll net/core/dev.c:6952 [inline] net_rx_action+0xa94/0x1010 net/core/dev.c:7074 handle_softirqs+0x213/0x8f0 kernel/softirq.c:561 __do_softirq kernel/softirq.c:595 [inline] invoke_softirq kernel/softirq.c:435 [inline] __irq_exit_rcu+0x109/0x170 kernel/softirq.c:662 irq_exit_rcu+0x9/0x30 kernel/softirq.c:678 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1049 [inline] sysvec_apic_timer_interrupt+0xa4/0xc0 arch/x86/kernel/apic/apic.c:1049
CVE-2024-57901	In the Linux kernel, the following vulnerability has been resolved: af_packet: fix vlan_get_protocol_dgram() vs MSG_PEEK Blamed commit forgot MSG_PEEK case, allowing a crash [1] as found by syzbot. Rework vlan_get_protocol_dgram() to not touch skb at all, so that it can be used from many cpus on the same skb. Add a const qualifier to skb argument. [1] skbuff: skb_under_panic: text:ffffff8a8ccd05 len:29 put:14 head:ffff88807fc8e400 data:ffff88807fc8e3f4 tail:0x11 end:0x140 dev:<NULL> -----[cut here]----- kernel BUG at net/core/skbuff.c:206 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 1 UID: 0 PID: 5892 Comm: syz-executor883 Not tainted 6.13.0-rc4-syzkaller-00054-gd6ef8b40d075 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010:skb_under_panic net/core/skbuff.c:206 [inline] RIP: 0010:skb_under_panic+0x14b/0x150 net/core/skbuff.c:216 Code: 0b 8d 48 c7 c6 86 d5 25 8e 48 8b 54 24 08 8b 0c 24 44 8b 44 24 04 4d 89 e9 50 41 54 41 57 41 56 e8 5a 69 79 f7 48 83 c4 20 90 <0f> 0b 0f 1f 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 RSP: 0018:ffff900038d7638 EFLAGS: 00010282 RAX: 0000000000000087 RBX: dffffc0000000000 RCX: 609ffd18ea660600 RDX: 0000000000000000 RSI: 0000000080000000 RDI: 0000000000000000 RBP: ffff88802483c8d0 R08: ffffffff817f0a8c R09: 1ffff9200071ae60 R10: dffffc0000000000 R11: fffff5200071ae61 R12: 0000000000000140 R13: ffff88807fc8e400 R14: ffff88807fc8e3f4 R15: 0000000000000011 FS: 00007fbac5e006c0(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fbac5e00d58 CR3: 000000001238e000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> skb_push+0xe5/0x100 net/core/skbuff.c:2636 vlan_get_protocol_dgram+0x165/0x290 net/packet/af_packet.c:585 packet_recvmmsg+0x948/0x1ef0 net/packet/af_packet.c:3552 sock_recvmmsg_nosec net/socket.c:1033 [inline] sock_recvmmsg+0x22f/0x280 net/socket.c:1055 __sys_recvmmsg+0x1c6/0x480 net/socket.c:2803 __sys_recvmmsg net/socket.c:2845 [inline] do_recvmmsg+0x426/0xab0 net/socket.c:2940 __sys_recvmmsg net/socket.c:3014 [inline] __do_sys_recvmmsg net/socket.c:3037 [inline] __se_sys_recvmmsg net/socket.c:3030 [inline] __x64_sys_recvmmsg+0x199/0x250 net/socket.c:3030 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f

CVE ID	Description
CVE-2024-57902	In the Linux kernel, the following vulnerability has been resolved: af_packet: fix vlan_get_tci() vs MSG_PEEK Blamed commit forgot MSG_PEEK case, allowing a crash [1] as found by syzbot. Rework vlan_get_tci() to not touch skb at all, so that it can be used from many cpus on the same skb. Add a const qualifier to skb argument. [1] skbuff: skb_under_panic: text:ffffffff8a8da482 len:32 put:14 head:ffff8807a1d5800 data:ffff8807a1d5810 tail:0x14 end:0x140 dev:<NULL> -----[cut here]----- kernel BUG at net/core/skbuff.c:206 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 0 UID: 0 PID: 5880 Comm: syz-executor172 Not tainted 6.13.0-rc3-syzkaller-00762-g9268abe611b0 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010:skb_panic net/core/skbuff.c:206 [inline] RIP: 0010:skb_under_panic +0x14b/0x150 net/core/skbuff.c:216 Code: 0b 8d 48 c7 c6 9e 6c 26 8e 48 8b 54 24 08 8b 0c 24 44 8b 44 24 04 4d 89 e9 50 41 54 41 57 41 56 e8 3a 5a 79 f7 48 83 c4 20 90 <0f> 0b 0f 1f 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 RSP: 0018:ffffc90003baf5b8 EFLAGS: 00010286 RAX: 0000000000000087 RBX: dffffc0000000000 RCX: 8565c1eec37aa000 RDX: 0000000000000000 RSI: 0000000080000000 RDI: 0000000000000000 RBP: ffff88802616fb50 R08: ffffffff817f0a4c R09: 1ffff92000775e50 R10: dffffc0000000000 R11: fffff52000775e51 R12: 0000000000000140 R13: ffff88807a1d5800 R14: ffff88807a1d5810 R15: 0000000000000014 FS: 00007fa03261f6c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffd65753000 CR3: 0000000031720000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> skb_push+0xe5/0x100 net/core/skbuff.c:2636 vlan_get_tci+0x272/0x550 net/packet/af_packet.c:565 packet_recvmmsg+0x13c9/0x1ef0 net/packet/af_packet.c:3616 sock_recvmmsg_nosec net/socket.c:1044 [inline] sock_recvmmsg+0x22f/0x280 net/socket.c:1066 __sys_recvmmsg+0x1c6/0x480 net/socket.c:2814 __sys_recvmmsg net/socket.c:2856 [inline] do_recvmmsg+0x426/0xab0 net/socket.c:2951 __sys_recvmmsg net/socket.c:3025 [inline] __do_sys_recvmmsg net/socket.c:3048 [inline] __se_sys_recvmmsg net/socket.c:3041 [inline] __x64_sys_recvmmsg+0x199/0x250 net/socket.c:3041 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83
CVE-2024-57903	In the Linux kernel, the following vulnerability has been resolved: net: restrict SO_REUSEPORT to inet sockets After blamed commit, crypto sockets could accidentally be destroyed from RCU call back, as spotted by zyzbot [1]. Trying to acquire a mutex in RCU callback is not allowed. Restrict SO_REUSEPORT socket option to inet sockets. v1 of this patch supported TCP, UDP and SCTP sockets, but fcnal-test.sh test needed RAW and ICMP support. [1] BUG: sleeping function called from invalid context at kernel/locking/mutex.c:562 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 24, name: ksoftirqd/1 preempt_count: 100, expected: 0 RCU nest depth: 0, expected: 0 1 lock held by ksoftirqd/1/24: #0: ffffffff8e937ba0 (rcu_callback){....}-{0:0}, at: rcu_lock_acquire include/linux/rcupdate.h:337 [inline] #0: ffffffff8e937ba0 (rcu_callback){....}-{0:0}, at: rcu_do_batch kernel/rcu/tree.c:2561 [inline] #0: ffffffff8e937ba0 (rcu_callback){....}-{0:0}, at: rcu_core+0xa37/0x17a0 kernel/rcu/tree.c:2823 Preemption disabled at: [<ffffffffff8161c8c8>] softirq_handle_begin kernel/softirq.c:402 [inline] [<ffffffffff8161c8c8>] handle_softirqs+0x128/0x9b0 kernel/softirq.c:537 CPU: 1 UID: 0 PID: 24 Comm: ksoftirqd/1 Not tainted 6.13.0-rc3-syzkaller-00174-ga024e377efed #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 __might_resched+0x5d4/0x780 kernel/sched/core.c:8758 __mutex_lock_common kernel/locking/mutex.c:562 [inline] __mutex_lock+0x131/0xee0 kernel/locking/mutex.c:735 crypto_put_default_null_skcipher+0x18/0x70 crypto/crypto_null.c:179 aead_release+0x3d/0x50 crypto/algif_aead.c:489 alg_do_release crypto/af_alg.c:118 [inline] alg_sock_destruct+0x86/0xc0 crypto/af_alg.c:502 __sk_destruct+0x58/0x5f0 net/core/sock.c:2260 rcu_do_batch kernel/rcu/tree.c:2567 [inline] rcu_core+0xaa0/0x17a0 kernel/rcu/tree.c:2823 handle_softirqs+0x2d4/0x9b0 kernel/softirq.c:561 run_ksoftirqd+0xca/0x130 kernel/softirq.c:950 smpboot_thread_fn+0x544/0xa30 kernel/smpboot.c:164 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK>
CVE-2024-57904	In the Linux kernel, the following vulnerability has been resolved: iio: adc: at91: call input_free_device() on allocated iio_dev Current implementation of at91_ts_register() calls input_free_device() on st->ts_input, however, the err label can be reached before the allocated iio_dev is stored to st->ts_input. Thus call input_free_device() on input instead of st->ts_input.
CVE-2024-57906	In the Linux kernel, the following vulnerability has been resolved: iio: adc: ti-ads8688: fix information leak in triggered buffer The 'buffer' local array is used to push data to user space from a triggered buffer, but it does not set values for inactive channels, as it only uses iio_for_each_active_channel() to assign new values. Initialize the array to zero before using it to avoid pushing uninitialized information to userspace.
CVE-2024-57907	In the Linux kernel, the following vulnerability has been resolved: iio: adc: rockchip_saradc: fix information leak in triggered buffer The 'data' local struct is used to push data to user space from a triggered buffer, but it does not set values for inactive channels, as it only uses iio_for_each_active_channel() to assign new values. Initialize the struct to zero before using it to avoid pushing uninitialized information to userspace.

CVE ID	Description
CVE-2024-57908	In the Linux kernel, the following vulnerability has been resolved: iio: imu: kmx61: fix information leak in triggered buffer The 'buffer' local array is used to push data to user space from a triggered buffer, but it does not set values for inactive channels, as it only uses iio_for_each_active_channel() to assign new values. Initialize the array to zero before using it to avoid pushing uninitialized information to userspace.
CVE-2024-57910	In the Linux kernel, the following vulnerability has been resolved: iio: light: vcnl4035: fix information leak in triggered buffer The 'buffer' local array is used to push data to userspace from a triggered buffer, but it does not set an initial value for the single data element, which is an u16 aligned to 8 bytes. That leaves at least 4 bytes uninitialized even after writing an integer value with regmap_read(). Initialize the array to zero before using it to avoid pushing uninitialized information to userspace.
CVE-2024-57911	In the Linux kernel, the following vulnerability has been resolved: iio: dummy: iio_simply_dummy_buffer: fix information leak in triggered buffer The 'data' array is allocated via kmalloc() and it is used to push data to user space from a triggered buffer, but it does not set values for inactive channels, as it only uses iio_for_each_active_channel() to assign new values. Use kzalloc for the memory allocation to avoid pushing uninitialized information to userspace.
CVE-2024-57912	In the Linux kernel, the following vulnerability has been resolved: iio: pressure: zpa2326: fix information leak in triggered buffer The 'sample' local struct is used to push data to user space from a triggered buffer, but it has a hole between the temperature and the timestamp (u32 pressure, u16 temperature, GAP, u64 timestamp). This hole is never initialized. Initialize the struct to zero before using it to avoid pushing uninitialized information to userspace.
CVE-2024-57913	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_fs: Remove WARN_ON in functionfs_bind This commit addresses an issue related to below kernel panic where panic_on_warn is enabled. It is caused by the unnecessary use of WARN_ON in functionfs_bind, which easily leads to the following scenarios. 1.adb_write in adbd 2. UDC write via configfs <pre> ===== ->usb_ffs_open_thread() ->UDC write ->open_functionfs() ->configfs_write_iter() ->adb_open() ->gadget_dev_desc_UDC_store() ->adb_write() ->usb_gadget_register_driver_owner ->driver_register() ->StartMonitor() - >bus_add_driver() ->adb_read() ->gadget_bind_driver() <times-out without BIND event> - >configfs_composite_bind() ->usb_add_function() ->open_functionfs() ->ffs_func_bind() ->adb_open() ->functionfs_bind() <ffs->state !=FFS_ACTIVE> The adb_open, adb_read, and adb_write operations are invoked from the daemon, but trying to bind the function is a process that is invoked by UDC write through configfs, which opens up the possibility of a race condition between the two paths. In this race scenario, the kernel panic occurs due to the WARN_ON from functionfs_bind when panic_on_warn is enabled. This commit fixes the kernel panic by removing the unnecessary WARN_ON. Kernel panic - not syncing: kernel: panic_on_warn set ... [14.542395] Call trace: [14.542464] ffs_func_bind +0x1c8/0x14a8 [14.542468] usb_add_function+0xcc/0x1f0 [14.542473] configfs_composite_bind +0x468/0x588 [14.542478] gadget_bind_driver+0x108/0x27c [14.542483] really_probe+0x190/0x374 [14.542488] __driver_probe_device+0xa0/0x12c [14.542492] driver_probe_device+0x3c/0x220 [14.542498] __driver_attach+0x11c/0x1fc [14.542502] bus_for_each_dev+0x104/0x160 [14.542506] driver_attach+0x24/0x34 [14.542510] bus_add_driver+0x154/0x270 [14.542514] driver_register+0x68/0x104 [14.542518] usb_gadget_register_driver_owner+0x48/0xf4 [14.542523] gadget_dev_desc_UDC_store+0xf8/0x144 [14.542526] configfs_write_iter+0xf0/0x138 </pre>
CVE-2024-57916	In the Linux kernel, the following vulnerability has been resolved: misc: microchip: pci1xxxx: Resolve kernel panic during GPIO IRQ handling Resolve kernel panic caused by improper handling of IRQs while accessing GPIO values. This is done by replacing generic_handle_irq with handle_nested_irq.
CVE-2024-57917	In the Linux kernel, the following vulnerability has been resolved: topology: Keep the cpumask unchanged when printing cpumap During fuzz testing, the following warning was discovered: different return values (15 and 11) from vsnprintf("%*pbl ", ...) test:keyword is WARNING in kvasprintf WARNING: CPU: 55 PID: 1168477 at lib/kasprintf.c:30 kvasprintf+0x121/0x130 Call Trace: kvasprintf+0x121/0x130 kasprintf+0xa6/0xe0 bitmap_print_to_buf+0x89/0x100 core_siblings_list_read+0x7e/0xb0 kernfs_file_read_iter+0x15b/0x270 new_sync_read+0x153/0x260 vfs_read+0x215/0x290 ksys_read+0xb9/0x160 do_syscall_64+0x56/0x100 entry_SYSCALL_64_after_hwframe+0x78/0xe2 The call trace shows that kvasprintf() reported this warning during the printing of core_siblings_list. kvasprintf() has several steps: (1) First, calculate the length of the resulting formatted string. (2) Allocate a buffer based on the returned length. (3) Then, perform the actual string formatting. (4) Check whether the lengths of the formatted strings returned in steps (1) and (2) are consistent. If the core_cpumask is modified between steps (1) and (3), the lengths obtained in these two steps may not match. Indeed our test includes cpu hotplugging, which should modify core_cpumask while printing. To fix this issue, cache the cpumask into a temporary variable before calling cpumap_print_{list, cpumask}_to_buf(), to keep it unchanged during the printing process.

CVE ID	Description
CVE-2024-57922	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add check for granularity in dml ceil/floor helpers [Why] Wrapper functions for dcn_bw_ceil2() and dcn_bw_floor2() should check for granularity is non zero to avoid assert and divide-by-zero error in dcn_bw_ functions. [How] Add check for granularity 0. (cherry picked from commit f6e09701c3eb2ccb8cb0518e0b67f1c69742a4ec)
CVE-2024-57924	In the Linux kernel, the following vulnerability has been resolved: fs: relax assertions on failure to encode file handles Encoding file handles is usually performed by a filesystem >encode_fh() method that may fail for various reasons. The legacy users of exportfs_encode_fh(), namely, nfsd and name_to_handle_at(2) syscall are ready to cope with the possibility of failure to encode a file handle. There are a few other users of exportfs_encode_{fh,fid}() that currently have a WARN_ON() assertion when ->encode_fh() fails. Relax those assertions because they are wrong. The second linked bug report states commit 16aac5ad1fa9 ("ovl: support encoding non-decodable file handles") in v6.6 as the regressing commit, but this is not accurate. The aforementioned commit only increases the chances of the assertion and allows triggering the assertion with the reproducer using overlays, inotify and drop_caches. Triggering this assertion was always possible with other filesystems and other reasons of ->encode_fh() failures and more particularly, it was also possible with the exact same reproducer using overlays that is mounted with options index=on,nfs_export=on also on kernels < v6.6. Therefore, I am not listing the aforementioned commit as a Fixes commit. Backport hint: this patch will have a trivial conflict applying to v6.6.y, and other trivial conflicts applying to stable kernels < v6.6.
CVE-2024-57925	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix a missing return value check bug In the smb2_send_interim_resp(), if ksmbd_alloc_work_struct() fails to allocate a node, it returns a NULL pointer to the in_work pointer. This can lead to an illegal memory write of in_work->response_buf when allocate_interim_rsp_buf() attempts to perform a kzalloc() on it. To address this issue, incorporating a check for the return value of ksmbd_alloc_work_struct() ensures that the function returns immediately upon allocation failure, thereby preventing the aforementioned illegal memory access.
CVE-2024-57926	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: Set private->all_drm_private[i]->drm to NULL if mtk_drm_bind returns err The pointer need to be set to NULL, otherwise KASAN complains about use-after-free. Because in mtk_drm_bind, all private's drm are set as follows. private->all_drm_private[i]->drm = drm; And drm will be released by drm_dev_put in case mtk_drm_kms_init returns failure. However, the shutdown path still accesses the previous allocated memory in drm_atomic_helper_shutdown. [86.874820] watchdog: watchdog0: watchdog did not stop! [86.512054] ===== [86.513162] BUG: KASAN: use-after-free in drm_atomic_helper_shutdown+0x33c/0x378 [86.514258] Read of size 8 at addr ffff0000d46fc068 by task shutdown/1 [86.515213] [86.515455] CPU: 1 UID: 0 PID: 1 Comm: shutdown Not tainted 6.13.0-rc1-mtk+gfa1a78e5d24b-dirty #55 [86.516752] Hardware name: Unknown Product/Unknown Product, BIOS 2022.10 10/01/2022 [86.517960] Call trace: [86.518333] show_stack+0x20/0x38 (C) [86.518891] dump_stack_lvl+0x90/0xd0 [86.519443] print_report+0xf8/0x5b0 [86.519985] kasan_report+0xb4/0x100 [86.520526] __asan_report_load8_noabort+0x20/0x30 [86.521240] drm_atomic_helper_shutdown+0x33c/0x378 [86.521966] mtk_drm_shutdown+0x54/0x80 [86.522546] platform_shutdown+0x64/0x90 [86.523137] device_shutdown+0x260/0x5b8 [86.523728] kernel_restart+0x78/0xf0 [86.524282] __do_sys_reboot +0x258/0x2f0 [86.524871] __arm64_sys_reboot+0x90/0xd8 [86.525473] invoke_syscall+0x74/0x268 [86.526041] el0_svc_common.constprop.0+0xb0/0x240 [86.526751] do_el0_svc+0x4c/0x70 [86.527251] el0_svc+0x4c/0xc0 [86.527719] el0t_64_sync_handler+0x144/0x168 [86.528367] el0t_64_sync+0x198/0x1a0 [86.528920] [86.529157] The buggy address belongs to the physical page: [86.529972] page: refcount:0 mapcount:0 mapping:0000000000000000 index:0xffff0000d46fd4d0 pfn:0x1146fc [86.531319] flags: 0xbfffc00000000000(node=0 zone=2 lastcpupid=0xffff) [86.532267] raw: 0bffffc0000000000 0000000000000000 dead000000000122 0000000000000000 [86.533390] raw: ffff0000d46fd4d0 0000000000000000 00000000ffffff 0000000000000000 [86.534511] page dumped because: kasan: bad access detected [86.535323] [86.535559] Memory state around the buggy address: [86.536265] ffff0000d46fbf00: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ffff0000d46fbf80: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff [86.538363] >ffff0000d46fc000: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff [86.544733] ^ [86.551057] ffff0000d46fc080: ff ff ff ff ff ff ff ff ff ff ff ff ff ff [86.557510] ffff0000d46fc100: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ===== [86.571093] Disabling lock debugging due to kernel taint [86.577642] Unable to handle kernel paging request at virtual address e0e9c0920000000b [86.581834] KASAN: maybe wild-memory-access in range [0x0752049000000058-0x075204900000005f] ...

CVE ID	Description
CVE-2024-57929	In the Linux kernel, the following vulnerability has been resolved: dm array: fix releasing a faulty array block twice in dm_array_cursor_end When dm_bm_read_lock() fails due to locking or checksum errors, it releases the faulty block implicitly while leaving an invalid output pointer behind. The caller of dm_bm_read_lock() should not operate on this invalid dm_block pointer, or it will lead to undefined result. For example, the dm_array_cursor incorrectly caches the invalid pointer on reading a faulty array block, causing a double release in dm_array_cursor_end(), then hitting the BUG_ON in dm-bufio cache_put(). Reproduce steps: 1. initialize a cache device dmsetup create cmeta --table "0 8192 linear /dev/sdc 0" dmsetup create cdata --table "0 65536 linear /dev/sdc 8192" dmsetup create corig --table "0 524288 linear /dev/sdc \$262144" dd if=/dev/zero of=/dev/mapper/cmeta bs=4k count=1 dmsetup create cache --table "0 524288 cache /dev/mapper/cmeta \ /dev/mapper/cdata /dev/mapper/corig 128 2 metadata2 writethrough smq 0" 2. wipe the second array block offline dmsteup remove cache cmeta cdata corig mapping_root=\$(dd if=/dev/sdc bs=1c count=8 skip=192 \ 2>/dev/null hexdump -e '1/8 "%u\n"') ablock=\$(dd if=/dev/sdc bs=1c count=8 skip=\$((4096*mapping_root+2056)) \ 2>/dev/null hexdump -e '1/8 "%u\n"') dd if=/dev/zero of=/dev/sdc bs=4k count=1 seek=\$ablock 3. try reopen the cache device dmsetup create cmeta --table "0 8192 linear /dev/sdc 0" dmsetup create cdata --table "0 65536 linear /dev/sdc 8192" dmsetup create corig --table "0 524288 linear /dev/sdc \$262144" dmsetup create cache --table "0 524288 cache /dev/mapper/cmeta \ /dev/mapper/cdata /dev/mapper/corig 128 2 metadata2 writethrough smq 0" Kernel logs: (snip) device-mapper: array: array_block_check failed: blocknr 0 != wanted 10 device-mapper: block manager: array validator check failed for block 10 device-mapper: array: get_ablock failed device-mapper: cache metadata: dm_array_cursor_next for mapping failed -----[cut here]----- kernel BUG at drivers/md/dm-bufio.c:638! Fix by setting the cached block pointer to NULL on errors. In addition to the reproducer described above, this fix can be verified using the "array_cursor/damaged" test in dm-unit: dm-unit run /pdata/array_cursor/damaged --kernel-dir <KERNEL_DIR>
CVE-2024-57930	In the Linux kernel, the following vulnerability has been resolved: tracing: Have process_string() also allow arrays In order to catch a common bug where a TRACE_EVENT() TP_fast_assign() assigns an address of an allocated string to the ring buffer and then references it in TP_printk(), which can be executed hours later when the string is free, the function test_event_printk() runs on all events as they are registered to make sure there's no unwanted dereferencing. It calls process_string() to handle cases in TP_printk() format that has "%s". It returns whether or not the string is safe. But it can have some false positives. For instance, xe_bo_move() has: TP_printk("move_lacks_source:%s, migrate object %p [size %zu] from %s to %s device_id:%s", __entry->move_lacks_source ? "yes" : "no", __entry->bo, __entry->size, xe_mem_type_to_name[__entry->old_placement], xe_mem_type_to_name[__entry->new_placement], __get_str(device_id)) Where the "%s" references into xe_mem_type_to_name[]. This is an array of pointers that should be safe for the event to access. Instead of flagging this as a bad reference, if a reference points to an array, where the record field is the index, consider it safe.
CVE-2024-57931	In the Linux kernel, the following vulnerability has been resolved: selinux: ignore unknown extended permissions When evaluating extended permissions, ignore unknown permissions instead of calling BUG(). This commit ensures that future permissions can be added without interfering with older kernels.
CVE-2024-57932	In the Linux kernel, the following vulnerability has been resolved: gve: guard XDP xmit NDO on existence of xdp queues In GVE, dedicated XDP queues only exist when an XDP program is installed and the interface is up. As such, the NDO XDP XMIT callback should return early if either of these conditions are false. In the case of no loaded XDP program, priv->num_xdp_queues=0 which can cause a divide-by-zero error, and in the case of interface down, num_xdp_queues remains untouched to persist XDP queue count for the next interface up, but the TX pointer itself would be NULL. The XDP xmit callback also needs to synchronize with a device transitioning from open to close. This synchronization will happen via the GVE_PRIV_FLAGS_NAPI_ENABLED bit along with a synchronize_net() call, which waits for any RCU critical sections at call-time to complete.
CVE-2024-57933	In the Linux kernel, the following vulnerability has been resolved: gve: guard XSK operations on the existence of queues This patch predicates the enabling and disabling of XSK pools on the existence of queues. As it stands, if the interface is down, disabling or enabling XSK pools would result in a crash, as the RX queue pointer would be NULL. XSK pool registration will occur as part of the next interface up. Similarly, xsk_wakeup needs be guarded against queues disappearing while the function is executing, so a check against the GVE_PRIV_FLAGS_NAPI_ENABLED flag is added to synchronize with the disabling of the bit and the synchronize_net() in gve_turndown.
CVE-2024-57938	In the Linux kernel, the following vulnerability has been resolved: net/sctp: Prevent autoclose integer overflow in sctp_association_init() While by default max_autoclose equals to INT_MAX / HZ, one may set net.sctp.max_autoclose to UINT_MAX. There is code in sctp_association_init() that can consequently trigger overflow.

CVE ID	Description
CVE-2024-57939	In the Linux kernel, the following vulnerability has been resolved: riscv: Fix sleeping in invalid context in die() die() can be called in exception handler, and therefore cannot sleep. However, die() takes spinlock_t which can sleep with PREEMPT_RT enabled. That causes the following warning: BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 285, name: mutex preempt_count: 110001, expected: 0 RCU nest depth: 0, expected: 0 CPU: 0 UID: 0 PID: 285 Comm: mutex Not tainted 6.12.0-rc7-00022-ge19049cf7d56-dirty #234 Hardware name: riscv-virtio,qemu (DT) Call Trace: dump_backtrace+0x1c/0x24 show_stack+0x2c/0x38 dump_stack_lvl+0x5a/0x72 dump_stack+0x14/0x1c __might_resched+0x130/0x13a rt_spin_lock+0x2a/0x5c die+0x24/0x112 do_trap_insn_illegal+0xa0/0xea _new_vmalloc_restore_context_a0+0xcc/0xd8 Oops - illegal instruction [#1] Switch to use raw_spinlock_t, which does not sleep even with PREEMPT_RT enabled.
CVE-2024-57940	In the Linux kernel, the following vulnerability has been resolved: exfat: fix the infinite loop in exfat_readdir() If the file system is corrupted so that a cluster is linked to itself in the cluster chain, and there is an unused directory entry in the cluster, 'dentry' will not be incremented, causing condition 'dentry < max_dentries' unable to prevent an infinite loop. This infinite loop causes s_lock not to be released, and other tasks will hang, such as exfat_sync_fs(). This commit stops traversing the cluster chain when there is unused directory entry in the cluster to avoid this infinite loop.
CVE-2024-57945	In the Linux kernel, the following vulnerability has been resolved: riscv: mm: Fix the out of bound issue of vmemmap address In sparse vmemmap model, the virtual address of vmemmap is calculated as: ((struct page *)VMEMMAP_START - (phys_ram_base >> PAGE_SHIFT)). And the struct page's va can be calculated with an offset: (vmemmap + (pfn)). However, when initializing struct pages, kernel actually starts from the first page from the same section that phys_ram_base belongs to. If the first page's physical address is not (phys_ram_base >> PAGE_SHIFT), then we get an va below VMEMMAP_START when calculating va for it's struct page. For example, if phys_ram_base starts from 0x82000000 with pfn 0x82000, the first page in the same section is actually pfn 0x80000. During init_unavailable_range(), we will initialize struct page for pfn 0x80000 with virtual address ((struct page *)VMEMMAP_START - 0x2000), which is below VMEMMAP_START as well as PCI_IO_END. This commit fixes this bug by introducing a new variable 'vmemmap_start_pfn' which is aligned with memory section size and using it to calculate vmemmap address instead of phys_ram_base.
CVE-2024-57946	In the Linux kernel, the following vulnerability has been resolved: virtio-blk: don't keep queue frozen during system suspend Commit 4ce6e2db00de ("virtio-blk: Ensure no requests in virtqueues before deleting vqs.") replaces queue quiesce with queue freeze in virtio-blk's PM callbacks. And the motivation is to drain inflight IOs before suspending. block layer's queue freeze looks very handy, but it is also easy to cause deadlock, such as, any attempt to call into bio_queue_enter() may run into deadlock if the queue is frozen in current context. There are all kinds of ->suspend() called in suspend context, so keeping queue frozen in the whole suspend context isn't one good idea. And Marek reported lockdep warning[1] caused by virtio-blk's freeze queue in virtblk_freeze(). [1] https://lore.kernel.org/linux-block/ca16370e-d646-4eee-b9cc-87277c89c43c@samsung.com/ Given the motivation is to drain in-flight IOs, it can be done by calling freeze & unfreeze, meantime restore to previous behavior by keeping queue quiesced during suspend.
CVE-2024-57947	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_set_pipapo: fix initial map fill The initial buffer has to be init'd to all-ones, but it must restrict it to the size of the first field, not the total field size. After each round in the map search step, the result and the fill map are swapped, so if we have a set where f->bsize of the first element is smaller than m->bsize_max, those one-bits are leaked into future rounds result map. This makes pipapo find an incorrect matching results for sets where first field size is not the largest. Followup patch adds a test case to nft_concat_range.sh selftest script. Thanks to Stefano Brivio for pointing out that we need to zero out the remainder explicitly, only correcting memset() argument isn't enough.

CVE ID	Description
CVE-2024-57948	In the Linux kernel, the following vulnerability has been resolved: mac802154: check local interfaces before deleting sdata list syzkaller reported a corrupted list in ieee802154_if_remove. [1] Remove an IEEE 802.15.4 network interface after unregister an IEEE 802.15.4 hardware device from the system. CPU0 CPU1 ===== genl_family_rcv_msg_doit ieee802154_unregister_hw ieee802154_del_iface ieee802154_remove_interfaces rdev_del_virtual_intf_deprecated list_del(&sdata->list) ieee802154_if_remove list_del_rcu The net device has been unregistered, since the rcu grace period, unregistration must be run before ieee802154_if_remove. To avoid this issue, add a check for local->interfaces before deleting sdata list. [1] kernel BUG at lib/list_debug.c:58! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 0 UID: 0 PID: 6277 Comm: syz-executor157 Not tainted 6.12.0-rc6-syzkaller-00005-g557329bcecc2 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010: __list_del_entry_valid_or_report+0xf4/0x140 lib/list_debug.c:56 Code: e8 a1 7e 00 07 90 0f 0b 48 c7 c7 e0 37 60 8c 4c 89 fe e8 8f 7e 00 07 90 0f 0b 48 c7 c7 40 38 60 8c 4c 89 fe e8 7d 7e 00 07 90 <0f> 0b 48 c7 c7 a0 38 60 8c 4c 89 fe e8 6b 7e 00 07 90 0f 0b 48 c7 RSP: 0018:ffff9000490f3d0 EFLAGS: 00010246 RAX: 000000000000004e RBX: dead000000000122 RCX: d211eee56bb28d00 RDX: 0000000000000000 RSI: 0000000008000000 RDI: 0000000000000000 RBP: ffff88805b278dd8 R08: ffffffff8174a12c R09: 1fffffff2852f0d R10: dffffc0000000000 R11: fffffbfff2852f0e R12: dffffc0000000000 R13: dffffc0000000000 R14: dead000000000100 R15: ffff88805b278cc0 FS: 0000555572f94380(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 00000000080050033 CR2: 000056262e4a3000 CR3: 0000000078496000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __list_del_entry_valid include/linux/list.h:124 [inline] __list_del_entry include/linux/list.h:215 [inline] list_del_rcu include/linux/rculist.h:157 [inline] ieee802154_if_remove+0x86/0x1e0 net/mac802154/iface.c:687 rdev_del_virtual_intf_deprecated net/ieee802154/rdev-ops.h:24 [inline] ieee802154_del_iface +0x2c0/0x5c0 net/ieee802154/nl-phy.c:323 genl_family_rcv_msg_doit net/netlink/genetlink.c:1115 [inline] genl_family_rcv_msg net/netlink/genetlink.c:1195 [inline] genl_rcv_msg+0xb14/0xec0 net/netlink/genetlink.c:1210 netlink_rcv_skb+0x1e3/0x430 net/netlink/af_netlink.c:2551 genl_rcv +0x28/0x40 net/netlink/genetlink.c:1219 netlink_unicast_kernel net/netlink/af_netlink.c:1331 [inline] netlink_unicast+0x7f6/0x990 net/netlink/af_netlink.c:1357 netlink_sendmsg+0x8e4/0xcb0 net/netlink/af_netlink.c:1901 sock_sendmsg_nosec net/socket.c:729 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:744 ____sys_sendmsg+0x52a/0x7e0 net/socket.c:2607 __sys_sendmsg net/socket.c:2661 [inline] __sys_sendmsg+0x292/0x380 net/socket.c:2690 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe +0x77/0x7f
CVE-2024-57949	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v3-its: Don't enable interrupts in its_irq_set_vcpu_affinity() The following call-chain leads to enabling interrupts in a nested interrupt disabled section: irq_set_vcpu_affinity() irq_get_desc_lock() raw_spin_lock_irqsave() <--- Disable interrupts its_irq_set_vcpu_affinity() guard(raw_spinlock_irq) <--- Enables interrupts when leaving the guard() irq_put_desc_unlock() <--- Warns because interrupts are enabled This was broken in commit b97e8a2f7130, which replaced the original raw_spin_[un]lock() pair with guard(raw_spinlock_irq). Fix the issue by using guard(raw_spinlock). [tglx: Massaged change log]
CVE-2024-57950	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Initialize denominator defaults to 1 [WHAT & HOW] Variables, used as denominators and maybe not assigned to other values, should be initialized to non-zero to avoid DIVIDE_BY_ZERO, as reported by Coverity. (cherry picked from commit e2c4c6c10542ccfe4a0830bb6c9fd5b177b7bbb7)
CVE-2024-57951	In the Linux kernel, the following vulnerability has been resolved: hrtimers: Handle CPU state correctly on hotplug Consider a scenario where a CPU transitions from CPUHP_ONLINE to halfway through a CPU hotunplug down to CPUHP_HRTIMERS_PREPARE, and then back to CPUHP_ONLINE: Since hrtimers_prepare_cpu() does not run, cpu_base.hres_active remains set to 1 throughout. However, during a CPU unplug operation, the tick and the clockevents are shut down at CPUHP_AP_TICK_DYING. On return to the online state, for instance CFS incorrectly assumes that the hrtick is already active, and the chance of the clockevent device to transition to oneshot mode is also lost forever for the CPU, unless it goes back to a lower state than CPUHP_HRTIMERS_PREPARE once. This round-trip reveals another issue; cpu_base.online is not set to 1 after the transition, which appears as a WARN_ON_ONCE in enqueue_hrtimer(). Aside of that, the bulk of the per CPU state is not reset either, which means there are dangling pointers in the worst case. Address this by adding a corresponding startup() callback, which resets the stale per CPU state and sets the online flag. [tglx: Make the new callback unconditionally available, remove the online modification in the prepare() callback and clear the remaining state in the starting callback instead of the prepare callback]

CVE ID	Description
CVE-2024-57952	In the Linux kernel, the following vulnerability has been resolved: Revert "libfs: fix infinite directory reads for offset dir" The current directory offset allocator (based on mtree_alloc_cyclic) stores the next offset value to return in octx->next_offset. This mechanism typically returns values that increase monotonically over time. Eventually, though, the newly allocated offset value wraps back to a low number (say, 2) which is smaller than other already- allocated offset values. Yu Kuai <yukuai3@huawei.com> reports that, after commit 64a7ce76fb90 ("libfs: fix infinite directory reads for offset dir"), if a directory's offset allocator wraps, existing entries are no longer visible via readdir/getdents because offset_readdir() stops listing entries once an entry's offset is larger than octx->next_offset. These entries vanish persistently -- they can be looked up, but will never again appear in readdir(3) output. The reason for this is that the commit treats directory offsets as monotonically increasing integer values rather than opaque cookies, and introduces this comparison: if (dentry2offset(dentry) >= last_index) { On 64-bit platforms, the directory offset value upper bound is $2^{63} - 1$. Directory offsets will monotonically increase for millions of years without wrapping. On 32-bit platforms, however, LONG_MAX is $2^{31} - 1$. The allocator can wrap after only a few weeks (at worst). Revert commit 64a7ce76fb90 ("libfs: fix infinite directory reads for offset dir") to prepare for a fix that can work properly on 32-bit systems and might apply to recent LTS kernels where shmem employs the simple_offset mechanism.
CVE-2024-57953	In the Linux kernel, the following vulnerability has been resolved: rtc: tps6594: Fix integer overflow on 32bit systems The problem is this multiply in tps6594_rtc_set_offset() tmp = offset * TICKS_PER_HOUR; The "tmp" variable is an s64 but "offset" is a long in the (-277774)-277774 range. On 32bit systems a long can hold numbers up to approximately two billion. The number of TICKS_PER_HOUR is really large, (32768 * 3600) or roughly a hundred million. When you start multiplying by a hundred million it doesn't take long to overflow the two billion mark. Probably the safest way to fix this is to change the type of TICKS_PER_HOUR to long long because it's such a large number.
CVE-2024-57970	libarchive through 3.7.7 has a heap-based buffer over-read in header_gnu_longlink in archive_read_support_format_tar.c via a TAR archive because it mishandles truncation in the middle of a GNU long linkname.
CVE-2024-57973	In the Linux kernel, the following vulnerability has been resolved: rdma/cxgb4: Prevent potential integer overflow on 32bit The "gl->tot_len" variable is controlled by the user. It comes from process_responses(). On 32bit systems, the "gl->tot_len + sizeof(struct cpl_pass_accept_req) + sizeof(struct rss_header)" addition could have an integer wrapping bug. Use size_add() to prevent this.

CVE ID	Description
CVE-2024-57974	In the Linux kernel, the following vulnerability has been resolved: udp: Deal with race between UDP socket address change and rehash If a UDP socket changes its local address while it's receiving datagrams, as a result of connect(), there is a period during which a lookup operation might fail to find it, after the address is changed but before the secondary hash (port and address) and the four-tuple hash (local and remote ports and addresses) are updated. Secondary hash chains were introduced by commit 30fff9231fad ("udp: bind() optimisation") and, as a result, a rehash operation became needed to make a bound socket reachable again after a connect(). This operation was introduced by commit 719f835853a9 ("udp: add rehash on connect()") which isn't however a complete fix: the socket will be found once the rehashing completes, but not while it's pending. This is noticeable with a socat(1) server in UDP4-LISTEN mode, and a client sending datagrams to it. After the server receives the first datagram (cf. _xioopen_ipdgram_listen()), it issues a connect() to the address of the sender, in order to set up a directed flow. Now, if the client, running on a different CPU thread, happens to send a (subsequent) datagram while the server's socket changes its address, but is not rehashed yet, this will result in a failed lookup and a port unreachable error delivered to the client, as apparent from the following reproducer: <pre>LEN=\$((cat /proc/sys/net/core/wmem_default) / 4) dd if=/dev/urandom bs=1 count=\${LEN} of=tmp.in while ;; do taskset -c 1 socat UDP4-LISTEN:1337,null-eof OPEN:tmp.out,create,trunc & sleep 0.1 sleep 1 taskset -c 2 socat OPEN:tmp.in UDP4:localhost:1337,shut-null wait done where the client will eventually get ECONNREFUSED on a write() (typically the second or third one of a given iteration): 2024/11/13 21:28:23 socat[46901] E write(6, 0x556db2e3c000, 8192): Connection refused This issue was first observed as a seldom failure in Podman's tests checking UDP functionality while using pasta(1) to connect the container's network namespace, which leads us to a reproducer with the lookup error resulting in an ICMP packet on a tap device: LOCAL_ADDR="\$(ip -j -4 addr showjq -rM '.[] .addr_info[0] select(.scope == "global").local)" while ;; do ./pasta --config-net - p pasta.pcap -u 1337 socat UDP4-LISTEN:1337,null-eof OPEN:tmp.out,create,trunc & sleep 0.2 sleep 1 socat OPEN:tmp.in UDP4:\${LOCAL_ADDR}:1337,shut-null wait cmp tmp.in tmp.out done Once this fails: tmp.in tmp.out differ: char 8193, line 29 we can finally have a look at what's going on: \$ tshark -r pasta.pcap 1 0.000000 :: ? ff02::16 ICMPv6 110 Multicast Listener Report Message v2 2 0.168690 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192 3 0.168767 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192 4 0.168806 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192 5 0.168827 c6:47:05:8d:dc:04 ? Broadcast ARP 42 Who has 88.198.0.161? Tell 88.198.0.164 6 0.168851 9a:55:9a:55:9a:55 ? c6:47:05:8d:dc:04 ARP 42 88.198.0.161 is at 9a:55:9a:55:9a:55 7 0.168875 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192 8 0.168896 88.198.0.164 ? 88.198.0.161 ICMP 590 Destination unreachable (Port unreachable) 9 0.168926 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192 10 0.168959 88.198.0.161 ? 88.198.0.164 UDP 8234 60260 ? 1337 Len=8192 11 0.168989 88.198.0.161 ? 88.198.0.164 UDP 4138 60260 ? 1337 Len=4096 12 0.169010 88.198.0.161 ? 88.198.0.164 UDP 42 60260 ? 1337 Len=0 On the third datagram received, the network namespace of the container initiates an ARP lookup to deliver the ICMP message. In another variant of this reproducer, starting the client with: strace -f pasta --config-net -u 1337 socat UDP4-LISTEN:1337,null-eof OPEN:tmp.out,create,tru ---truncated---</pre>

CVE ID	Description
CVE-2024-57975	In the Linux kernel, the following vulnerability has been resolved: btrfs: do proper folio cleanup when run_delalloc_nocow() failed [BUG] With CONFIG_DEBUG_VM set, test case generic/476 has some chance to crash with the following VM_BUG_ON_FOLIO(): BTRFS error (device dm-3): cow_file_range failed, start 1146880 end 1253375 len 106496 ret -28 BTRFS error (device dm-3): run_delalloc_nocow failed, start 1146880 end 1253375 len 106496 ret -28 page: refcount:4 mapcount:0 mapping:00000000592787cc index:0x12 pfn:0x10664 aops:btrfs_aops [btrfs] ino:101 dentry name(?):"f1774" flags: 0x2fffff80004028(uptodate lru private node=0 zone=2 lastcpupid=0xffff) page dumped because: VM_BUG_ON_FOLIO(!folio_test_locked(folio)) -----[cut here]----- kernel BUG at mm/page-writeback.c:2992! Internal error: Oops - BUG: 00000000f2000800 [#1] SMP CPU: 2 UID: 0 PID: 3943513 Comm: kworker/u24:15 Tainted: G OE 6.12.0-rc7-custom+ #87 Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022 Workqueue: events_unbound btrfs_async_reclaim_data_space [btrfs] pc : folio_clear_dirty_for_io+0x128/0x258 lr : folio_clear_dirty_for_io+0x128/0x258 Call trace: folio_clear_dirty_for_io+0x128/0x258 btrfs_folio_clamp_clear_dirty+0x80/0xd0 [btrfs] __process_folios_contig+0x154/0x268 [btrfs] extent_clear_unlock_delalloc+0x5c/0x80 [btrfs] run_delalloc_nocow+0x5f8/0x760 [btrfs] btrfs_run_delalloc_range+0xa8/0x220 [btrfs] writepage_delalloc+0x230/0x4c8 [btrfs] extent_writepage+0xb8/0x358 [btrfs] extent_write_cache_pages+0x21c/0x4e8 [btrfs] btrfs_writepages+0x94/0x150 [btrfs] do_writepages+0x74/0x190 filemap_fdatawrite_wbc+0x88/0xc8 start_delalloc_inodes+0x178/0x3a8 [btrfs] btrfs_start_delalloc_roots+0x174/0x280 [btrfs] shrink_delalloc+0x114/0x280 [btrfs] flush_space+0x250/0x2f8 [btrfs] btrfs_async_reclaim_data_space+0x180/0x228 [btrfs] process_one_work+0x164/0x408 worker_thread+0x25c/0x388 kthread+0x100/0x118 ret_from_fork+0x10/0x20 Code: 910a8021 a90363f7 a9046bf9 94012379 (d4210000) ---[end trace 0000000000000000]--- [CAUSE] The first two lines of extra debug messages show the problem is caused by the error handling of run_delalloc_nocow(). E.g. we have the following dirtied range (4K blocksize 4K page size): 0 16K 32K ////////////////////////// Pre-allocated And the range [0, 16K) has a preallocated extent. - Enter run_delalloc_nocow() for range [0, 16K) Which found range [0, 16K) is preallocated, can do the proper NOCOW write. - Enter fallback_to_fow() for range [16K, 32K) Since the range [16K, 32K) is not backed by preallocated extent, we have to go COW. - cow_file_range() failed for range [16K, 32K) So cow_file_range() will do the clean up by clearing folio dirty, unlock the folios. Now the folios in range [16K, 32K) is unlocked. - Enter extent_clear_unlock_delalloc() from run_delalloc_nocow() Which is called with PAGE_START_WRITEBACK to start page writeback. But folios can only be marked writeback when it's properly locked, thus this triggered the VM_BUG_ON_FOLIO(). Furthermore there is another hidden but common bug that run_delalloc_nocow() is not clearing the folio dirty flags in its error handling path. This is the common bug shared between run_delalloc_nocow() and cow_file_range(). [FIX] - Clear folio dirty for range [@start, @cur_offset] Introduce a helper, cleanup_dirty_folios(), which will find and lock the folio in the range, clear the dirty flag and start/end the writeback, with the extra handling for the @locked_folio. - Introduce a helper to clear folio dirty, start and end writeback - Introduce a helper to record the last failed COW range end This is to trace which range we should skip, to avoid double unlocking. - Skip the failed COW range for the e --- truncated---

CVE ID	Description
CVE-2024-57976	In the Linux kernel, the following vulnerability has been resolved: btrfs: do proper folio cleanup when cow_file_range() failed [BUG] When testing with COW fixup marked as BUG_ON() (this is involved with the new pin_user_pages*() change, which should not result new out-of-band dirty pages), I hit a crash triggered by the BUG_ON() from hitting COW fixup path. This BUG_ON() happens just after a failed btrfs_run_delalloc_range(): BTRFS error (device dm-2): failed to run delalloc range, root 348 ino 405 folio 65536 submit_bitmap 6-15 start 90112 len 106496: -28 -----[cut here]----- kernel BUG at fs/btrfs/extent_io.c:1444! Internal error: Oops - BUG: 00000000f2000800 [#1] SMP CPU: 0 UID: 0 PID: 434621 Comm: kworker/u24:8 Tainted: G OE 6.12.0-rc7-custom+ #86 Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022 Workqueue: events_unbound btrfs_async_reclaim_data_space [btrfs] pc : extent_writepage_io+0x2d4/0x308 [btrfs] lr : extent_writepage_io+0x2d4/0x308 [btrfs] Call trace: extent_writepage_io+0x2d4/0x308 [btrfs] extent_writepage+0x218/0x330 [btrfs] extent_write_cache_pages+0x1d4/0x4b0 [btrfs] btrfs_writepages+0x94/0x150 [btrfs] do_writepages+0x74/0x190 filemap_fdatawrite_wbc+0x88/0xc8 start_delalloc_inodes+0x180/0x3b0 [btrfs] btrfs_start_delalloc_roots+0x174/0x280 [btrfs] shrink_delalloc+0x114/0x280 [btrfs] flush_space+0x250/0x2f8 [btrfs] btrfs_async_reclaim_data_space+0x180/0x228 [btrfs] process_one_work+0x164/0x408 worker_thread+0x25c/0x388 kthread+0x100/0x118 ret_from_fork+0x10/0x20 Code: aa1403e1 9402f3ef aa1403e0 9402f36f (d4210000) ---[end trace 0000000000000000]--- [CAUSE] That failure is mostly from cow_file_range(), where we can hit -ENOSPC. Although the -ENOSPC is already a bug related to our space reservation code, let's just focus on the error handling. For example, we have the following dirty range [0, 64K) of an inode, with 4K sector size and 4K page size: 0 16K 32K 48K 64K //////////////////// ##### Where // means page are still dirty, and ### means the extent io tree has EXTENT_DEALLOC flag. - Enter extent_writepage() for page 0 - Enter btrfs_run_delalloc_range() for range [0, 64K) - Enter cow_file_range() for range [0, 64K) - Function btrfs_reserve_extent() only reserved one 16K extent So we created extent map and ordered extent for range [0, 16K) 0 16K 32K 48K 64K //////// //////////////////// <- OE -> ##### And range [0, 16K) has its delalloc flag cleared. But since we haven't yet submit any bio, involved 4 pages are still dirty. - Function btrfs_reserve_extent() returns with -ENOSPC Now we have to run error cleanup, which will clear all EXTENT_DEALLOC* flags and clear the dirty flags for the remaining ranges: 0 16K 32K 48K 64K //////// Note that range [0, 16K) still has its pages dirty. - Some time later, writeback is triggered again for the range [0, 16K) since the page range still has dirty flags. - btrfs_run_delalloc_range() will do nothing because there is no EXTENT_DEALLOC flag. - extent_writepage_io() finds page 0 has no ordered flag Which falls into the COW fixup path, triggering the BUG_ON(). Unfortunately this error handling bug dates back to the introduction of btrfs. Thankfully with the abuse of COW fixup, at least it won't crash the kernel. [FIX] Instead of immediately unlocking the extent and folios, we keep the extent and folios locked until either erroring out or the whole delalloc range finished. When the whole delalloc range finished without error, we just unlock the whole range with PAGE_SET_ORDERED (and PAGE_UNLOCK for !keep_locked cases) ---truncated---
CVE-2024-57977	In the Linux kernel, the following vulnerability has been resolved: memcg: fix soft lockup in the OOM process A soft lockup issue was found in the product with about 56,000 tasks were in the OOM cgroup, it was traversing them when the soft lockup was triggered. watchdog: BUG: soft lockup - CPU#2 stuck for 23s! [VM Thread:1503066] CPU: 2 PID: 1503066 Comm: VM Thread Kdump: loaded Tainted: G Hardware name: Huawei Cloud OpenStack Nova, BIOS RIP: 0010:console_unlock+0x343/0x540 RSP: 0000:ffffb751447db9a0 EFLAGS: 00000247 ORIG_RAX: ffffffff13 RAX: 0000000000000001 RBX: 0000000000000000 RCX: 00000000ffffff RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000247 RBP: ffffffff71f90 R08: 0000000000000000 R09: 0000000000000040 R10: 0000000000000080 R11: 0000000000000000 R12: ffffffff74bd0 R13: ffffffff60a220 R14: 0000000000000247 R15: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f2fe6ad91f0 CR3: 00000004b2076003 CR4: 000000000360ee0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000000000ff DR7: 0000000000000400 Call Trace: vprintk_emit+0x193/0x280 printk+0x52/0x6e dump_task+0x114/0x130 mem_cgroup_scan_tasks+0x76/0x100 dump_header+0x1fe/0x210 oom_kill_process+0xd1/0x100 out_of_memory+0x125/0x570 mem_cgroup_out_of_memory+0xb5/0xd0 try_charge+0x720/0x770 mem_cgroup_try_charge+0x86/0x180 mem_cgroup_try_charge_delay+0x1c/0x40 do_anonymous_page+0xb5/0x390 handle_mm_fault+0xc4/0x1f0 This is because thousands of processes are in the OOM cgroup, it takes a long time to traverse all of them. As a result, this lead to soft lockup in the OOM process. To fix this issue, call 'cond_resched' in the 'mem_cgroup_scan_tasks' function per 1000 iterations. For global OOM, call 'touch_softlockup_watchdog' per 1000 iterations to avoid this issue.
CVE-2024-57978	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Fix potential error pointer dereference in detach_pm() The proble is on the first line: if (jpeg->pd_dev[i] && ! pm_runtime_suspended(jpeg->pd_dev[i])) If jpeg->pd_dev[i] is an error pointer, then passing it to pm_runtime_suspended() will lead to an Oops. The other conditions check for both error pointers and NULL, but it would be more clear to use the IS_ERR_OR_NULL() check for that.

CVE ID	Description
CVE-2024-57979	In the Linux kernel, the following vulnerability has been resolved: pps: Fix a use-after-free On a board running ntpd and gpsd, I'm seeing a consistent use-after-free in sys_exit() from gpsd when rebooting: pps1: removed -----[cut here]----- kobject: '(null)' (00000000db4bec24): is not initialized, yet kobject_put() is being called. WARNING: CPU: 2 PID: 440 at lib/kobject.c:734 kobject_put+0x120/0x150 CPU: 2 UID: 299 PID: 440 Comm: gpsd Not tainted 6.11.0-rc6-00308-gb31c44928842 #1 Hardware name: Raspberry Pi 4 Model B Rev 1.1 (DT) pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) pc : kobject_put+0x120/0x150 lr : kobject_put+0x120/0x150 sp : fffffffc0803d3ae0 x29: fffffffc0803d3ae0 x28: fffffff8042dc9738 x27: 0000000000000001 x26: 0000000000000000 x25: fffffff8042dc9040 x24: fffffff8042dc9440 x23: fffffff80402a4620 x22: fffffff8042ef4bd0 x21: fffffff80405cb600 x20: 000000000008001b x19: fffffff8040b3b6e0 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 696e6920746f6e20 x14: 7369203a29343263 x13: 205d303434542020 x12: 0000000000000000 x11: 0000000000000000 x10: 0000000000000000 x9: 0000000000000000 x8: 0000000000000000 x7: 0000000000000000 x6: 0000000000000000 x5: 0000000000000000 x4: 0000000000000000 x3: 0000000000000000 x2: 0000000000000000 x1: 0000000000000000 x0: 0000000000000000 Call trace: kobject_put+0x120/0x150 cdev_put+0x20/0x3c __fput+0x2c4/0x2d8 ____fput+0x1c/0x38 task_work_run+0x70/0xfc do_exit+0x2a0/0x924 do_group_exit+0x34/0x90 get_signal+0x7fc/0x8c0 do_signal+0x128/0x13b4 do_notify_resume+0xdc/0x160 el0_svc+0xd4/0xf8 el0t_64_sync_handler+0x140/0x14c el0t_64_sync+0x190/0x194 ---[end trace 0000000000000000]--- ...followed by more symptoms of corruption, with similar stacks: refcount_t: underflow; use-after-free. kernel BUG at lib/list_debug.c:62! Kernel panic - not syncing: Oops - BUG: Fatal exception This happens because pps_device_destruct() frees the pps_device with the embedded cdev immediately after calling cdev_del(), but, as the comment above cdev_del() notes, fops for previously opened cdevs are still callable even after cdev_del() returns. I think this bug has always been there: I can't explain why it suddenly started happening every time I reboot this particular board. In commit d953e0e837e6 ("pps: Fix a use-after free bug when unregistering a source."), George Spelvin suggested removing the embedded cdev. That seems like the simplest way to fix this, so I've implemented his suggestion, using __register_chrdev() with pps_idr becoming the source of truth for which minor corresponds to which device. But now that pps_idr defines userspace visibility instead of cdev_add(), we need to be sure the pps->dev refcount can't reach zero while userspace can still find it again. So, the idr_remove() call moves to pps_unregister_cdev(), and pps_idr now holds a reference to pps->dev. pps_core: source serial1 got cdev (251:1) <...> pps pps1: removed pps_core: unregistering pps1 pps_core: deallocating pps1
CVE-2024-57980	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Fix double free in error path If the uvc_status_init() function fails to allocate the int_urb, it will free the dev->status pointer but doesn't reset the pointer to NULL. This results in the kfree() call in uvc_status_cleanup() trying to double-free the memory. Fix it by resetting the dev->status pointer to NULL after freeing it. Reviewed by: Ricardo Ribalda <ribalda@chromium.org>
CVE-2024-57981	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Fix NULL pointer dereference on certain command aborts If a command is queued to the final usable TRB of a ring segment, the enqueue pointer is advanced to the subsequent link TRB and no further. If the command is later aborted, when the abort completion is handled the dequeue pointer is advanced to the first TRB of the next segment. If no further commands are queued, xhci_handle_stopped_cmd_ring() sees the ring pointers unequal and assumes that there is a pending command, so it calls xhci_mod_cmd_timer() which crashes if cur_cmd was NULL. Don't attempt timer setup if cur_cmd is NULL. The subsequent doorbell ring likely is unnecessary too, but it's harmless. Leave it alone. This is probably Bug 219532, but no confirmation has been received. The issue has been independently reproduced and confirmed fixed using a USB MCU programmed to NAK the Status stage of SET_ADDRESS forever. Everything continued working normally after several prevented crashes.
CVE-2024-57982	In the Linux kernel, the following vulnerability has been resolved: xfrm: state: fix out-of-bounds read during lookup lookup and resize can run in parallel. The xfrm_state_hash_generation seqlock ensures a retry, but the hash functions can observe a hmask value that is too large for the new hlist array. rehash does: rcu_assign_pointer(net->xfrm.state_bydst, ndst) [...] net->xfrm.state_hmask = nhmask; While state lookup does: h = xfrm_dst_hash(net, daddr, saddr, tmpl->reqid, encap_family); hlist_for_each_entry_rcu(x, net->xfrm.state_bydst + h, bydst) { This is only safe in case the update to state_bydst is larger than net->xfrm.xfrm_state_hmask (or if the lookup function gets serialized via state spinlock again). Fix this by prefetching state_hmask and the associated pointers. The xfrm_state_hash_generation seqlock retry will ensure that the pointer and the hmask will be consistent. The existing helpers, like xfrm_dst_hash(), are now unsafe for RCU side, add lockdep assertions to document that they are only safe for insert side. xfrm_state_lookup_byaddr() uses the spinlock rather than RCU. AFAICS this is an oversight from back when state lookup was converted to RCU, this lock should be replaced with RCU in a future patch.

CVE ID	Description
CVE-2024-57986	In the Linux kernel, the following vulnerability has been resolved: HID: core: Fix assumption that Resolution Multipliers must be in Logical Collections A report in 2019 by the syzbot fuzzer was found to be connected to two errors in the HID core associated with Resolution Multipliers. One of the errors was fixed by commit ea427a222d8b ("HID: core: Fix deadlock in hid_apply_multiplier."), but the other has not been fixed. This error arises because hid_apply_multiplier() assumes that every Resolution Multiplier control is contained in a Logical Collection, i.e., there's no way the routine can ever set multiplier_collection to NULL. This is in spite of the fact that the function starts with a big comment saying: "The Resolution Multiplier control must be contained in the same * Logical Collection as the control(s) to which it is to be applied. ... * If no Logical Collection is * defined, the Resolution Multiplier is associated with all * controls in the report." * HID Usage Table, v1.12, Section 4.3.1, p30 * * Thus, search from the current collection upwards until we find a * logical collection... The comment and the code overlook the possibility that none of the collections found may be a Logical Collection. The fix is to set the multiplier_collection pointer to NULL if the collection found isn't a Logical Collection.
CVE-2024-57990	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7925: fix off by one in mt7925_load_clc() This comparison should be >= instead of > to prevent an out of bounds read and write.
CVE-2024-57993	In the Linux kernel, the following vulnerability has been resolved: HID: hid-thrustmaster: Fix warning in thrustmaster_probe by adding endpoint check syzbot has found a type mismatch between a USB pipe and the transfer endpoint, which is triggered by the hid-thrustmaster driver[1]. There is a number of similar, already fixed issues [2]. In this case as in others, implementing check for endpoint type fixes the issue. [1] https://syzkaller.appspot.com/bug?extid=040e8b3db6a96908d470 [2] https://syzkaller.appspot.com/bug?extid=348331f63b034f89b622
CVE-2024-57994	In the Linux kernel, the following vulnerability has been resolved: ptr_ring: do not block hard interrupts in ptr_ring_resize_multiple() Jakub added a lockdep_assert_no_hardirq() check in __page_pool_put_page() to increase test coverage. syzbot found a splat caused by hard irq blocking in ptr_ring_resize_multiple() [1] As current users of ptr_ring_resize_multiple() do not require hard irqs being masked, replace it to only block BH. Rename helpers to better reflect they are safe against BH only. - ptr_ring_resize_multiple() to ptr_ring_resize_multiple_bh() - skb_array_resize_multiple() to skb_array_resize_multiple_bh() [1] WARNING: CPU: 1 PID: 9150 at net/core/page_pool.c:709 __page_pool_put_page net/core/page_pool.c:709 [inline] WARNING: CPU: 1 PID: 9150 at net/core/page_pool.c:709 page_pool_put_unrefed_netmem+0x157/0xa40 net/core/page_pool.c:780 Modules linked in: CPU: 1 UID: 0 PID: 9150 Comm: syz.1.1052 Not tainted 6.11.0-rc3-syzkaller-00202-gf8669d7b5f5d #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 RIP: 0010:__page_pool_put_page net/core/page_pool.c:709 [inline] RIP: 0010:page_pool_put_unrefed_netmem+0x157/0xa40 net/core/page_pool.c:780 Code: 74 0e e8 7c aa fb f7 eb 43 e8 75 aa fb f7 eb 3c 65 8b 1d 38 a8 6a 76 31 ff 89 de e8 a3 ae fb f7 85 db 74 0b e8 5a aa fb f7 90 <0f> 0b 90 eb 1d 65 8b 1d 15 a8 6a 76 31 ff 89 de e8 84 ae fb f7 85 RSP: 0018:ffffc9000bda6b58 EFLAGS: 00010083 RAX: ffffffff8997e523 RBX: 0000000000000000 RCX: 0000000000040000 RDX: fffffc9000fbd0000 RSI: 0000000000001842 RDI: 0000000000001843 RBP: 0000000000000000 R08: ffffffff8997df2c R09: 1ffffd40003a000d R10: dffffc0000000000 R11: fffff940003a000e R12: fffffea0001d0004 R13: ffff88802e8a4000 R14: dffffc0000000000 R15: 00000000ffffffff FS: 00007fb7aaf716c0(0000) GS:ffff8880b9300000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fa15a0d4b72 CR3: 00000000561b0000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> tun_ptr_free drivers/net/tun.c:617 [inline] __ptr_ring_swap_queue include/linux/ptr_ring.h:571 [inline] ptr_ring_resize_multiple_noprof include/linux/ptr_ring.h:643 [inline] tun_queue_resize drivers/net/tun.c:3694 [inline] tun_device_event+0xaaaf/0x1080 drivers/net/tun.c:3714 notifier_call_chain +0x19f/0x3e0 kernel/notifier.c:93 call_netdevice_notifiers_extack net/core/dev.c:2032 [inline] call_netdevice_notifiers net/core/dev.c:2046 [inline] dev_change_tx_queue_len+0x158/0x2a0 net/core/dev.c:9024 do_setlink+0xff6/0x41f0 net/core/rtnetlink.c:2923 rtnl_setlink+0x40d/0x5a0 net/core/rtnetlink.c:3201 rtnetlink_rcv_msg+0x73f/0xcfc0 net/core/rtnetlink.c:6647 netlink_rcv_skb+0x1e3/0x430 net/netlink/af_netlink.c:2550

CVE ID	Description
CVE-2024-57996	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: don't allow 1 packet limit The current implementation does not work correctly with a limit of 1. iproute2 actually checks for this and this patch adds the check in kernel as well. This fixes the following syzkaller reported crash: UBSAN: array-index-out-of-bounds in net/sched/sch_sfq.c:210:6 index 65535 is out of range for type 'struct sfq_head[128]' CPU: 0 PID: 2569 Comm: syz-executor101 Not tainted 5.10.0-smp-DEV #1 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: __dump_stack lib/dump_stack.c:79 [inline] dump_stack+0x125/0x19f lib/dump_stack.c:120 ubsan_epilogue lib/ubsan.c:148 [inline] __ubsan_handle_out_of_bounds+0xed/0x120 lib/ubsan.c:347 sfq_link net/sched/sch_sfq.c:210 [inline] sfq_dec+0x528/0x600 net/sched/sch_sfq.c:238 sfq_dequeue +0x39b/0x9d0 net/sched/sch_sfq.c:500 sfq_reset+0x13/0x50 net/sched/sch_sfq.c:525 qdisc_reset +0xfe/0x510 net/sched/sch_generic.c:1026 tbf_reset+0x3d/0x100 net/sched/sch_tbf.c:319 qdisc_reset +0xfe/0x510 net/sched/sch_generic.c:1026 dev_reset_queue+0x8c/0x140 net/sched/sch_generic.c:1296 netdev_for_each_tx_queue include/linux/netdevice.h:2350 [inline] dev_deactivate_many+0x6dc/0xc20 net/sched/sch_generic.c:1362 __dev_close_many+0x214/0x350 net/core/dev.c:1468 dev_close_many +0x207/0x510 net/core/dev.c:1506 unregister_netdevice_many+0x40f/0x16b0 net/core/dev.c:10738 unregister_netdevice_queue+0x2be/0x310 net/core/dev.c:10695 unregister_netdevice include/linux/netdevice.h:2893 [inline] __tun_detach+0x6b6/0x1600 drivers/net/tun.c:689 tun_detach drivers/net/tun.c:705 [inline] tun_chr_close+0x104/0x1b0 drivers/net/tun.c:3640 __fput+0x203/0x840 fs/file_table.c:280 task_work_run+0x129/0x1b0 kernel/task_work.c:185 exit_task_work include/linux/task_work.h:33 [inline] do_exit+0x5ce/0x2200 kernel/exit.c:931 do_group_exit+0x144/0x310 kernel/exit.c:1046 __do_sys_exit_group kernel/exit.c:1057 [inline] __se_sys_exit_group kernel/exit.c:1055 [inline] __x64_sys_exit_group+0x3b/0x40 kernel/exit.c:1055 do_syscall_64+0x6c/0xd0 entry_SYSCALL_64_after_hwframe+0x61/0xcb RIP: 0033:0x7fe5e7b52479 Code: Unable to access opcode bytes at RIP 0x7fe5e7b5244f. RSP: 002b:00007ffd3c800398 EFLAGS: 00000246 ORIG_RAX: 00000000000000e7 RAX: ffffffffdfda RBX: 0000000000000000 RCX: 00007fe5e7b52479 RDX: 000000000000003c RSI: 00000000000000e7 RDI: 0000000000000000 RBP: 00007fe5e7bcd2d0 R08: ffffffffbb8 R09: 0000000000000014 R10: 0000000000000000 R11: 0000000000000246 R12: 00007fe5e7bcd2d0 R13: 0000000000000000 R14: 00007fe5e7bcd2d0 R15: 00007fe5e7b24270 The crash can be also be reproduced with the following (with a tc recompiled to allow for sfq limits of 1): tc qdisc add dev dummy0 handle 1: root tbf rate 1Kbit burst 100b lat 1s ../iproute2-6.9.0/tc qdisc add dev dummy0 handle 2: parent 1:10 sfq limit 1 ifconfig dummy0 up ping -l dummy0 -f -c2 -W0.1 8.8.8.8 sleep 1 Scenario that triggers the crash: * the first packet is sent and queued in TBF and SFQ; qdisc qlen is 1 * TBF dequeues: it peeks from SFQ which moves the packet to the gso_skb list and keeps qdisc qlen set to 1. TBF is out of tokens so it schedules itself for later. * the second packet is sent and TBF tries to queue it to SFQ. qdisc qlen is now 2 and because the SFQ limit is 1 the packet is dropped by SFQ. At this point qlen is 1, and all of the SFQ slots are empty, however q->tail is not NULL. At this point, assuming no more packets are queued, when sch_dequeue runs again it will decrement the qlen for the current empty slot causing an underflow and the subsequent out of bounds access.
CVE-2024-57997	In the Linux kernel, the following vulnerability has been resolved: wifi: wcn36xx: fix channel survey memory allocation size KASAN reported a memory allocation issue in wcn->chan_survey due to incorrect size calculation. This commit uses kcalloc to allocate memory for wcn->chan_survey, ensuring proper initialization and preventing the use of uninitialized values when there are no frames on the channel.
CVE-2024-57998	In the Linux kernel, the following vulnerability has been resolved: OPP: add index check to assert to avoid buffer overflow in _read_freq() Pass the freq index to the assert function to make sure we do not read a freq out of the opp->rates[] table when called from the indexed variants: dev_pm_opp_find_freq_exact_indexed() or dev_pm_opp_find_freq_ceil/floor_indexed(). Add a secondary parameter to the assert function, unused for assert_single_clk() then add assert_clk_index() which will check for the clock index when called from the _indexed() find functions.

CVE ID	Description
CVE-2024-57999	In the Linux kernel, the following vulnerability has been resolved: powerpc/pseries/iommu: IOMMU incorrectly marks MMIO range in DDW Power Hypervisor can possibly allocate MMIO window intersecting with Dynamic DMA Window (DDW) range, which is over 32-bit addressing. These MMIO pages needs to be marked as reserved so that IOMMU doesn't map DMA buffers in this range. The current code is not marking these pages correctly which is resulting in LPAR to OOPS while booting. The stack is at below BUG: Unable to handle kernel data access on read at 0xc00800005cd40000 Faulting instruction address: 0xc0000000005cdac Oops: Kernel access of bad area, sig: 11 [#1] LE PAGE_SIZE=64K MMU=Hash SMP NR_CPUS=2048 NUMA pSeries Modules linked in: af_packet rfkill ibmveth(X) lpfc(+) nvmet_fc nvmet nvme_keyring crc10dif_vpmsum nvme_fc nvme_fabrics nvme_core be2net(+) nvme_auth rtc_generic nfsd_auth_rpcgss nfs_acl lockd grace sunrpc fuse configfs ip_tables x_tables xfs libcrc32c dm_service_time ibmvfc(X) scsi_transport_fc vmx_crypto gf128mul crc32c_vpmsum dm_mirror dm_region_hash dm_log dm_multipath dm_mod sd_mod scsi_dh_emc scsi_dh_rdac scsi_dh_alua t10_pi crc64_rocksoft_generic crc64_rocksoft sg crc64_scsi_mod Supported: Yes, External CPU: 8 PID: 241 Comm: kworker/8:1 Kdump: loaded Not tainted 6.4.0-150600.23.14-default #1 SLE15-SP6 b44ee71c81261b9e4bab5e0cde1f2ed891d5359b Hardware name: IBM,9080-M9S POWER9 (raw) 0x4e2103 0xf000005 of:IBM,FW950.B0 (VH950_149) hv:phyp pSeries Workqueue: events work_for_cpu_fn NIP: c0000000005cdac LR: c00000000005e830 CTR: 0000000000000000 REGS: c00001400c9ff770 TRAP: 0300 Not tainted (6.4.0-150600.23.14-default) MSR: 800000000280b033 <SF,VEC,VSX,EE,FP,ME,IR,DR,RI,LE> CR: 24228448 XER: 00000001 CFAR: c0000000005cdd4 DAR: c00800005cd40000 DSISR: 400000000 IRQMASK: 0 GPR00: c00000000005e830 c00001400c9ffa10 c0000000001987400 c00001400c4fe800 GPR04: 0000080000000000 0000000000000001 0000000004000000 0000000000800000 GPR08: 0000000004000000 0000000000000001 c00800005cd40000 ffffffff ffffffff GPR12: 0000000084228882 c00000000a4c4f00 0000000000000010 0000080000000000 GPR16: c00001400c4fe800 0000000004000000 0800000000000000 c00000006088b800 GPR20: c00001401a7be980 c00001400eff3800 c000000002a2da68 0000000000000002b GPR24: c0000000026793a8 c000000002679368 000000000000002a c0000000026793c8 GPR28: 000008007effffff 0000080000000000 0000000000800000 c00001400c4fe800 NIP [c0000000005cdac] iommu_table_reserve_pages+0xac/0x100 LR [c00000000005e830] iommu_init_table +0x80/0x1e0 Call Trace: [c00001400c9ffa10] [c00000000005e810] iommu_init_table+0x60/0x1e0 (unreliable) [c00001400c9ffa90] [c00000000010356c] iommu_bypass_supported_pSeriesLP +0x9cc/0xe40 [c00001400c9ffc30] [c00000000005c300] dma_iommu_dma_supported+0xf0/0x230 [c00001400c9ffc0] [c000000000024b0c4] dma_supported+0x44/0x90 [c00001400c9ffc0] [c000000000024b14c] dma_set_mask+0x3c/0x80 [c00001400c9ffd00] [c0080000555b715c] be_probe +0xc4/0xb90 [be2net] [c00001400c9ffdc0] [c000000000986f3c] local_pci_probe+0x6c/0x110 [c00001400c9ffe40] [c000000000188f28] work_for_cpu_fn+0x38/0x60 [c00001400c9ffe70] [c00000000018e454] process_one_work+0x314/0x620 [c00001400c9fff10] [c00000000018f280] worker_thread+0x2b0/0x620 [c00001400c9fff90] [c00000000019bb18] kthread+0x148/0x150 [c00001400c9fff0] [c00000000000ded8] start_kernel_thread+0x14/0x18 There are 2 issues in the code 1. The index is "int" while the address is "unsigned long". This results in negative value when setting the bitmap. 2. The DMA offset is page shifted but the MMIO range is used as-is (64-bit address). MMIO address needs to be page shifted as well.
CVE-2024-58001	In the Linux kernel, the following vulnerability has been resolved: ocfs2: handle a symlink read error correctly Patch series "Convert ocfs2 to use folios". Mark did a conversion of ocfs2 to use folios and sent it to me as a giant patch for review ;-). So I've redone it as individual patches, and credited Mark for the patches where his code is substantially the same. It's not a bad way to do it; his patch had some bugs and my patches had some bugs. Hopefully all our bugs were different from each other. And hopefully Mark likes all the changes I made to his code! This patch (of 23): If we can't read the buffer, be sure to unlock the page before returning.
CVE-2024-58002	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Remove dangling pointers When an async control is written, we copy a pointer to the file handle that started the operation. That pointer will be used when the device is done. Which could be anytime in the future. If the user closes that file descriptor, its structure will be freed, and there will be one dangling pointer per pending async control, that the driver will try to use. Clean all the dangling pointers during release(). To avoid adding a performance penalty in the most common case (no async operation), a counter has been introduced with some logic to make sure that it is properly handled.
CVE-2024-58003	In the Linux kernel, the following vulnerability has been resolved: media: i2c: ds90ub9x3: Fix extra fwnode_handle_put() The ub913 and ub953 drivers call fwnode_handle_put(priv->sd.fwnode) as part of their remove process, and if the driver is removed multiple times, eventually leads to put "overflow", possibly causing memory corruption or crash. The fwnode_handle_put() is a leftover from commit 905f88cceb1 ("media: i2c: ds90ub9x3: Fix sub-device matching"), which changed the code related to the sd.fwnode, but missed removing these fwnode_handle_put() calls.

CVE ID	Description
CVE-2024-58005	In the Linux kernel, the following vulnerability has been resolved: tpm: Change to kcalloc() in eventlog/acpi.c The following failure was reported on HPE ProLiant D320: [10.693310][T1] tpm_tis STM0925:00: 2.0 TPM (device-id 0x3, rev-id 0) [10.848132][T1] -----[cut here]----- [10.853559][T1] WARNING: CPU: 59 PID: 1 at mm/page_alloc.c:4727 __alloc_pages_noprof +0x2ca/0x330 [10.862827][T1] Modules linked in: [10.866671][T1] CPU: 59 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.12.0-lp155.2.g52785e2-default #1 openSUSE Tumbleweed (unreleased) 588cd98293a7c9eba9013378d807364c088c9375 [10.882741][T1] Hardware name: HPE ProLiant DL320 Gen12/ProLiant DL320 Gen12, BIOS 1.20 10/28/2024 [10.892170][T1] RIP: 0010: __alloc_pages_noprof+0x2ca/0x330 [10.898103][T1] Code: 24 08 e9 4a fe ff ff e8 34 36 fa ff e9 88 fe ff ff 83 fe 0a 0f 86 b3 fd ff ff 80 3d 01 e7 ce 01 00 75 09 c6 05 f8 e6 ce 01 01 <0f> 0b 45 31 ff e9 e5 fe ff ff f7 c2 00 00 08 00 75 42 89 d9 80 e1 [10.917750][T1] RSP: 0000:ffffb7cf40077980 EFLAGS: 00010246 [10.923777][T1] RAX: 0000000000000000 RBX: 0000000000040cc0 RCX: 0000000000000000 [10.931727][T1] RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000040cc0 The above transcript shows that ACPI pointed a 16 MiB buffer for the log events because RSI maps to the 'order' parameter of __alloc_pages_noprof(). Address the bug by moving from devm_kmalloc() to devm_add_action() and kvmalloc() and devm_add_action().
CVE-2024-58006	In the Linux kernel, the following vulnerability has been resolved: PCI: dwc: ep: Prevent changing BAR size/flags in pci_epc_set_bar() In commit 4284c88fff0e ("PCI: designware-ep: Allow pci_epc_set_bar() update inbound map address") set_bar() was modified to support dynamically changing the backing physical address of a BAR that was already configured. This means that set_bar() can be called twice, without ever calling clear_bar() (as calling clear_bar() would clear the BAR's PCI address assigned by the host). This can only be done if the new BAR size/flags does not differ from the existing BAR configuration. Add these missing checks. If we allow set_bar() to set e.g. a new BAR size that differs from the existing BAR size, the new address translation range will be smaller than the BAR size already determined by the host, which would mean that a read past the new BAR size would pass the iATU untranslated, which could allow the host to read memory not belonging to the new struct pci_epf_bar. While at it, add comments which clarifies the support for dynamically changing the physical address of a BAR. (Which was also missing.)
CVE-2024-58007	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: socinfo: Avoid out of bounds read of serial number On MSM8916 devices, the serial number exposed in sysfs is constant and does not change across individual devices. It's always: db410c:/sys/devices/soc0\$ cat serial_number 2644893864 The firmware used on MSM8916 exposes SOCINFO_VERSION(0, 8), which does not have support for the serial_num field in the socinfo struct. There is an existing check to avoid exposing the serial number in that case, but it's not correct: When checking the item_size returned by SMEM, we need to make sure the *end* of the serial_num is within bounds, instead of comparing with the *start* offset. The serial_number currently exposed on MSM8916 devices is just an out of bounds read of whatever comes after the socinfo struct in SMEM. Fix this by changing offsetof() to offsetofend(), so that the size of the field is also taken into account.
CVE-2024-58010	In the Linux kernel, the following vulnerability has been resolved: binfmt_flat: Fix integer overflow bug on 32 bit systems Most of these sizes and counts are capped at 256MB so the math doesn't result in an integer overflow. The "relocs" count needs to be checked as well. Otherwise on 32bit systems the calculation of "full_data" could be wrong. full_data = data_len + relocs * sizeof(unsigned long);
CVE-2024-58011	In the Linux kernel, the following vulnerability has been resolved: platform/x86: int3472: Check for adev == NULL Not all devices have an ACPI companion fwnode, so adev might be NULL. This can e.g. (theoretically) happen when a user manually binds one of the int3472 drivers to another i2c/platform device through sysfs. Add a check for adev not being set and return -ENODEV in that case to avoid a possible NULL pointer deref in skl_int3472_get_acpi_buffer().
CVE-2024-58012	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda-dai: Ensure DAI widget is valid during params Each cpu DAI should associate with a widget. However, the topology might not create the right number of DAI widgets for aggregated amps. And it will cause NULL pointer deference. Check that the DAI widget associated with the CPU DAI is valid to prevent NULL pointer deference due to missing DAI widgets in topologies with aggregated amps.

CVE ID	Description
CVE-2024-58013	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix slab-use-after-free Read in mgmt_remove_adv_monitor_sync This fixes the following crash: <pre>===== BUG: KASAN: slab-use-after-free in mgmt_remove_adv_monitor_sync+0x3a/0xd0 net/bluetooth/mgmt.c:5543 Read of size 8 at addr ffff88814128f898 by task kworker/u9:4/5961 CPU: 1 UID: 0 PID: 5961 Comm: kworker/u9:4 Not tainted 6.12.0-syzkaller-10684-gf1cd565ce577 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report+0x143/0x180 mm/kasan/report.c:602 mgmt_remove_adv_monitor_sync+0x3a/0xd0 net/bluetooth/mgmt.c:5543 hci_cmd_sync_work+0x22b/0x400 net/bluetooth/hci_sync.c:332 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa63/0x1850 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 <TASK> Allocated by task 16026: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x98/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x243/0x390 mm/slub.c:4314 kmalloc_noprof include/linux/slab.h:901 [inline] kzalloc_noprof include/linux/slab.h:1037 [inline] mgmt_pending_new+0x65/0x250 net/bluetooth/mgmt_util.c:269 mgmt_pending_add+0x36/0x120 net/bluetooth/mgmt_util.c:296 remove_adv_monitor+0x102/0x1b0 net/bluetooth/mgmt.c:5568 hci_mgmt_cmd+0xc47/0x11d0 net/bluetooth/hci_sock.c:1712 hci_sock_sendmsg+0x7b8/0x11c0 net/bluetooth/hci_sock.c:1832 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:726 sock_write_iter+0x2d7/0x3f0 net/socket.c:1147 new_sync_write fs/read_write.c:586 [inline] vfs_write+0xaeb/0xd30 fs/read_write.c:679 ksys_write+0x18f/0x2b0 fs/read_write.c:731 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0xf7 Freed by task 16022: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:582 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x59/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2338 [inline] slab_free mm/slub.c:4598 [inline] kfree+0x196/0x420 mm/slub.c:4746 mgmt_pending_foreach+0xd1/0x130 net/bluetooth/mgmt_util.c:259 __mgmt_power_off+0x183/0x430 net/bluetooth/mgmt.c:9550 hci_dev_close_sync+0x6c4/0x11c0 net/bluetooth/hci_sync.c:5208 hci_dev_do_close net/bluetooth/hci_core.c:483 [inline] hci_dev_close+0x112/0x210 net/bluetooth/hci_core.c:508 sock_do_ioctl+0x158/0x460 net/socket.c:1209 sock_ioctl+0x626/0x8e0 net/socket.c:1328 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl+0xf5/0x170 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0xf7</pre>
CVE-2024-58014	In the Linux kernel, the following vulnerability has been resolved: wifi: brcm80211: add gain range check to wlc_phy_iqcal_gainparams_nphy() In 'wlc_phy_iqcal_gainparams_nphy()', add gain range check to WARN() instead of possible out-of-bounds 'tbl_iqcal_gainparams_nphy' access. Compile tested only. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2024-58015	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix for out-of bound access error Selfgen stats are placed in a buffer using print_array_to_buf_index() function. Array length parameter passed to the function is too big, resulting in possible out-of bound memory error. Decreasing buffer size by one fixes faulty upper bound of passed array. Discovered in coverity scan, CID 1600742 and CID 1600758
CVE-2024-58016	In the Linux kernel, the following vulnerability has been resolved: safesetid: check size of policy writes syzbot attempts to write a buffer with a large size to a sysfs entry with writes handled by handle_policy_update(), triggering a warning in kmalloc. Check the size specified for write buffers before allocating. [PM: subject tweak]
CVE-2024-58017	In the Linux kernel, the following vulnerability has been resolved: printk: Fix signed integer overflow when defining LOG_BUF_LEN_MAX Shifting 1 << 31 on a 32-bit int causes signed integer overflow, which leads to undefined behavior. To prevent this, cast 1 to u32 before performing the shift, ensuring well-defined behavior. This change explicitly avoids any potential overflow by ensuring that the shift occurs on an unsigned 32-bit integer.

CVE ID	Description
CVE-2024-58018	In the Linux kernel, the following vulnerability has been resolved: nvkm: correctly calculate the available space of the GSP cmdq buffer r535_gsp_cmdq_push() waits for the available page in the GSP cmdq buffer when handling a large RPC request. When it sees at least one available page in the cmdq, it quits the waiting with the amount of free buffer pages in the queue. Unfortunately, it always takes the [write pointer, buf_size) as available buffer pages before rolling back and wrongly calculates the size of the data should be copied. Thus, it can overwrite the RPC request that GSP is currently reading, which causes GSP hang due to corrupted RPC request: [549.209389] ----- <pre>[cut here]----- [549.214010] WARNING: CPU: 8 PID: 6314 at drivers/gpu/drm/nouveau/nvkm/ subdev/gsp/r535.c:116 r535_gsp_msgq_wait+0xd0/0x190 [nvkm] [549.225678] Modules linked in: nvkm(E+) gsp_log(E) snd_seq_dummy(E) snd_hrtimer(E) snd_seq(E) snd_timer(E) snd_seq_device(E) snd(E) soundcore(E) rfkill(E) qrtr(E) vfat(E) fat(E) ipmi_ssif(E) amd_atl(E) intel_rapl_msr(E) intel_rapl_common(E) mlx5_ib(E) amd64_edac(E) edac_mce_amd(E) kvm_amd(E) ib_uverbs(E) kvm(E) ib_core(E) acpi_ipmi(E) ipmi_si(E) mxm_wmi(E) ipmi_devintf(E) rapl(E) i2c_piix4(E) wmi_bmof(E) joydev(E) ptdma(E) acpi_cpufreq(E) k10temp(E) pcspkr(E) ipmi_msghandler(E) xfs(E) libcrc32c(E) ast(E) i2c_algo_bit(E) crct10dif_pclmul(E) drm_shmem_helper(E) nvme_tcp(E) crc32_pclmul(E) ahci(E) drm_kms_helper(E) libahci(E) nvme_fabrics(E) crc32c_intel(E) nvme(E) cdc_ether(E) mlx5_core(E) nvme_core(E) usbnet(E) drm(E) libata(E) ccp(E) ghash_clmulni_intel(E) mii(E) t10_pi(E) mlxfw(E) sp5100_tco(E) psample(E) pci_hyperv_intf(E) wmi(E) dm_multipath(E) sunrpc(E) dm_mirror(E) dm_region_hash(E) dm_log(E) dm_mod(E) be2iscsi(E) bnxt2i(E) cnic(E) uio(E) cxgb4i(E) cxgb4(E) tls(E) libcxgbi(E) libcxgb(E) qla4xxx(E) [549.225752] iscsi_boot_sysfs(E) iscsi_tcp(E) libiscsi_tcp(E) libiscsi(E) scsi_transport_iscsi(E) fuse(E) [last unloaded: gsp_log(E)] [549.326293] CPU: 8 PID: 6314 Comm: insmod Tainted: G E 6.9.0-rc6+ #1 [549.334039] Hardware name: ASRockRack 1U1G-MILAN/N/ROMED8-NL, BIOS L3.12E 09/06/2022 [549.341781] RIP: 0010:r535_gsp_msgq_wait+0xd0/0x190 [nvkm] [549.347343] Code: 08 00 00 89 da c1 e2 0c 48 8d ac 11 00 10 00 00 48 8b 0c 24 48 85 c9 74 1f c1 e0 0c 4c 8d 6d 30 83 e8 30 89 01 e9 68 ff ff ff <0f> 0b 49 c7 c5 92 ff ff e9 5a ff ff ff ba ff ff ff be c0 0c [549.366090] RSP: 0018:ffffacbccaaeb7d0 EFLAGS: 00010246 [549.371315] RAX: 0000000000000000 RBX: 0000000000000012 RCX: 00000000000923e28 [549.378451] RDX: 0000000000000000 RSI: 0000000055555554 RDI: ffffacbccaaeb730 [549.385590] RBP: 0000000000000001 R08: ffff8bd14d235f70 R09: ffff8bd14d235f70 [549.392721] R10: 0000000000000002 R11: ffff8bd14d233864 R12: 0000000000000002 [549.399854] R13: ffffacbccaaeb818 R14: 0000000000000020 R15: ffff8bb298c67000 [549.406988] FS: 00007f5179244740(0000) GS:ffff8bd14d200000(0000) knlGS:0000000000000000 [549.415076] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [549.420829] CR2: 00007fa844000010 CR3: 00000001567dc005 CR4: 0000000000770ef0 [549.427963] PKRU: 55555554 [549.430672] Call Trace: [549.433126] <TASK> [549.435233] ? __warn+0x7f/0x130 [549.438473] ? r535_gsp_msgq_wait+0xd0/0x190 [nvkm] [549.443426] ? report_bug+0x18a/0x1a0 [549.447098] ? handle_bug+0x3c/0x70 [549.450589] ? exc_invalid_op +0x14/0x70 [549.454430] ? asm_exc_invalid_op+0x16/0x20 [549.458619] ? r535_gsp_msgq_wait +0xd0/0x190 [nvkm] [549.463565] r535_gsp_msg_rcv+0x46/0x230 [nvkm] [549.468257] r535_gsp_rpc_push+0x106/0x160 [nvkm] [549.473033] r535_gsp_rpc_rm_ctrl_push+0x40/0x130 [nvkm] [549.478422] nvvidia_grid_init_vgpu_types+0xbc/0xe0 [nvkm] [549.483899] nvvidia_grid_init +0xb1/0xd0 [nvkm] [549.488420] ? srso_alias_return_thunk+0x5/0xfbef5 [549.493213] nvkm_device_pci_probe+0x305/0x420 [nvkm] [549.498338] local_pci_probe+0x46/ ---truncated---</pre>

CVE ID	Description
CVE-2024-58019	In the Linux kernel, the following vulnerability has been resolved: nvkm/gsp: correctly advance the read pointer of GSP message queue A GSP event message consists three parts: message header, RPC header, message body. GSP calculates the number of pages to write from the total size of a GSP message. This behavior can be observed from the movement of the write pointer. However, nvkm takes only the size of RPC header and message body as the message size when advancing the read pointer. When handling a two-page GSP message in the non rollback case, It wrongly takes the message body of the previous message as the message header of the next message. As the "message length" tends to be zero, in the calculation of size needs to be copied (0 - size of (message header)), the size needs to be copied will be "0xffffffff". It also triggers a kernel panic due to a NULL pointer error. [547.614102] msg: 00000f90: ff ff ff ff ff ff ff ff 40 d7 18 fb 8b 00 00 00@..... [547.622533] msg: 00000fa0: 00 00 00 00 ff ff ff ff ff ff ff 00 00 00 00 [547.630965] msg: 00000fb0: ff ff ff ff ff ff ff ff 00 00 00 00 ff ff ff ff [547.639397] msg: 00000fc0: ff ff ff ff 00 00 00 00 ff ff ff ff ff ff ff [547.647832] nvkm 0000:c1:00.0: gsp: peek msg rpc fn:0 len:0x0/0xfffffffffffff0 [547.655225] nvkm 0000:c1:00.0: gsp: get msg rpc fn:0 len:0x0/0xfffffffffffff0 [547.662532] BUG: kernel NULL pointer dereference, address: 0000000000000020 [547.669485] #PF: supervisor read access in kernel mode [547.674624] #PF: error_code(0x0000) - not-present page [547.679755] PGD 0 P4D 0 [547.682294] Oops: 0000 [#1] PREEMPT SMP NOPTI [547.686643] CPU: 22 PID: 322 Comm: kworker/22:1 Tainted: G E 6.9.0-rc6+ #1 [547.694893] Hardware name: ASRockRack 1U1G-MILAN/N/ROMED8-NL, BIOS L3.12E 09/06/2022 [547.702626] Workqueue: events r535_gsp_msgq_work [nvkm] [547.707921] RIP: 0010:r535_gsp_msg_recv+0x87/0x230 [nvkm] [547.713375] Code: 00 8b 70 08 48 89 e1 31 d2 4c 89 f7 e8 12 f5 ff ff 48 89 c5 48 85 c0 0f 84 cf 00 00 00 48 81 fd 00 f0 ff ff 0f 87 c4 00 00 00 <8b> 55 10 41 8b 46 30 85 d2 0f 85 f6 00 00 00 83 f8 04 76 10 ba 05 [547.732119] RSP: 0018:ffffbabe440f87e10 EFLAGS: 00010203 [547.737335] RAX: 0000000000000010 RBX: 0000000000000008 RCX: 000000000000003f [547.744461] RDX: 0000000000000000 RSI: ffffbabe4480a8030 RDI: 0000000000000010 [547.751585] RBP: 0000000000000010 R08: 0000000000000000 R09: ffffbabe440f87bb0 [547.758707] R10: ffffbabe440f87dc8 R11: 0000000000000010 R12: 0000000000000000 [547.765834] R13: 0000000000000000 R14: ffff9351df1e5000 R15: 0000000000000000 [547.772958] FS: 0000000000000000(0000) GS:ffff93708eb00000(0000) knlGS:0000000000000000 [547.781035] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [547.786771] CR2: 0000000000000020 CR3: 00000003cc220002 CR4: 0000000000770ef0 [547.793896] PKRU: 55555554 [547.796600] Call Trace: [547.799046] <TASK> [547.801152] ? __die+0x20/0x70 [547.804211] ? page_fault_oops+0x75/0x170 [547.808221] ? print_hex_dump+0x100/0x160 [547.812226] ? exc_page_fault+0x64/0x150 [547.816152] ? asm_exc_page_fault+0x22/0x30 [547.820341] ? r535_gsp_msg_recv+0x87/0x230 [nvkm] [547.825184] r535_gsp_msgq_work+0x42/0x50 [nvkm] [547.829845] process_one_work+0x196/0x3d0 [547.833861] worker_thread+0x2fc/0x410 [547.837613] ? __pfx_worker_thread+0x10/0x10 [547.841885] kthread+0xdf/0x110 [547.845031] ? __pfx_kthread+0x10/0x10 [547.848775] ret_from_fork+0x30/0x50 [547.852354] ? __pfx_kthread+0x10/0x10 [547.856097] ret_from_fork_asm+0x1a/0x30 [547.860019] </TASK> [547.862208] Modules linked in: nvkm(E) gsp_log(E) snd_seq_dummy(E) snd_hrtimer(E) snd_seq(E) snd_timer(E) snd_seq_device(E) snd(E) soundcore(E) rkill(E) qrtr(E) vfat(E) fat(E) ipmi_ssif(E) amd_atl(E) intel_rapl_msr(E) intel_rapl_common(E) amd64_edac(E) mlx5_ib(E) edac_mce_amd(E) kvm_amd ---truncated---
CVE-2024-58020	In the Linux kernel, the following vulnerability has been resolved: HID: multitouch: Add NULL check in mt_input_configured devm_kasprintf() can return a NULL pointer on failure, but this returned value in mt_input_configured() is not checked. Add NULL check in mt_input_configured(), to handle kernel NULL pointer dereference error.
CVE-2024-58034	In the Linux kernel, the following vulnerability has been resolved: memory: tegra20-emc: fix an OF node reference bug in tegra_emc_find_node_by_ram_code() As of_find_node_by_name() release the reference of the argument device node, tegra_emc_find_node_by_ram_code() releases some device nodes while still in use, resulting in possible UAFs. According to the bindings and the in-tree DTS files, the "emc-tables" node is always device's child node with the property "nvidia,use-ram-code", and the "lpddr2" node is a child of the "emc-tables" node. Thus utilize the for_each_child_of_node() macro and of_get_child_by_name() instead of of_find_node_by_name() to simplify the code. This bug was found by an experimental verification tool that I am developing. [krzysztof: applied v1, adjust the commit msg to incorporate v2 parts]
CVE-2024-58051	In the Linux kernel, the following vulnerability has been resolved: ipmi: ipmb: Add check devm_kasprintf() returned value devm_kasprintf() can return a NULL pointer on failure but this returned value is not checked.

CVE ID	Description
CVE-2024-58052	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Fix potential NULL pointer dereference in atomctrl_get_smc_sclk_range_table The function atomctrl_get_smc_sclk_range_table() does not check the return value of smu_atom_get_data_table(). If smu_atom_get_data_table() fails to retrieve SMU_Info table, it returns NULL which is later dereferenced. Found by Linux Verification Center (linuxtesting.org) with SVACE. In practice this should never happen as this code only gets called on polaris chips and the vbios data table will always be present on those chips.
CVE-2024-58053	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix handling of received connection abort Fix the handling of a connection abort that we've received. Though the abort is at the connection level, it needs propagating to the calls on that connection. Whilst the propagation bit is performed, the calls aren't then woken up to go and process their termination, and as no further input is forthcoming, they just hang. Also add some tracing for the logging of connection aborts.
CVE-2024-58054	In the Linux kernel, the following vulnerability has been resolved: staging: media: max96712: fix kernel oops when removing module The following kernel oops is thrown when trying to remove the max96712 module: Unable to handle kernel paging request at virtual address 00007375746174db Mem abort info: ESR = 0x0000000009600004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtbl: 4k pages, 48-bit VAs, pgdp=000000010af89000 [00007375746174db] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP Modules linked in: crct10dif_ce polyval_ce mxc_jpeg_encdec flexcan snd_soc_fsl_sai snd_soc_fsl_asoc_card snd_soc_fsl_micfil dwc_mipi_csi2 imx_csi_formatter polyval_generic v4l2_jpeg imx_pcm_dma can_dev snd_soc_imx_audmux snd_soc_wm8962 snd_soc_imx_card snd_soc_fsl_utils max96712(C-) rpmsg_ctrl rpmsg_char pwm_fan fuse [last unloaded: imx8_isi] CPU: 0 UID: 0 PID: 754 Comm: rmmmod Tainted: G C 6.12.0-rc6-06364-g327fec852c31 #17 Tainted: [C]=CRAP Hardware name: NXP i.MX95 19X19 board (DT) pstate: 60400009 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : led_put+0x1c/0x40 lr : v4l2_subdev_put_privacy_led+0x48/0x58 sp : ffff80008699bbb0 x29: ffff80008699bbb0 x28: ffff00008ac233c0 x27: 0000000000000000 x26: 0000000000000000 x25: 0000000000000000 x24: 0000000000000000 x23: ffff000080cf1170 x22: ffff00008b53bd00 x21: ffff8000822ad1c8 x20: ffff000080ff5c00 x19: ffff00008b53be40 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000004 x13: ffff0000800f8010 x12: 0000000000000000 x11: ffff000082acf5c0 x10: ffff000082acf478 x9 : ffff0000800f8010 x8 : 0101010101010101 x7 : 7f7f7f7f7f7f7f7f x6 : fefefeff6364626d x5 : 8080808000000000 x4 : 0000000000000020 x3 : 00000000553a3dc1 x2 : ffff00008ac233c0 x1 : ffff00008ac233c0 x0 : ff00737574617473 Call trace: led_put+0x1c/0x40 v4l2_subdev_put_privacy_led+0x48/0x58 v4l2_async_unregister_subdev+0x2c/0x1a4 max96712_remove+0x1c/0x38 [max96712] i2c_device_remove+0x2c/0x9c device_remove+0x4c/0x80 device_release_driver_internal+0x1cc/0x228 driver_detach+0x4c/0x98 bus_remove_driver+0x6c/0x9c driver_unregister+0x30/0x60 i2c_del_driver+0x54/0x64 max96712_i2c_driver_exit+0x18/0x1d0 [max96712] __arm64_sys_delete_module+0x1a4/0x290 invoke_syscall+0x48/0x10c el0_svc_common.constprop.0+0xc0/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x34/0xd8 el0t_64_sync_handler+0x120/0x12c el0t_64_sync+0x190/0x194 Code: f9000bf3 aa0003f3 f9402800 f9402000 (f9403400) ---[end trace 0000000000000000]--- This happens because in v4l2_i2c_subdev_init(), the i2c_set_cliendata() is called again and the data is overwritten to point to sd, instead of priv. So, in remove(), the wrong pointer is passed to v4l2_async_unregister_subdev(), leading to a crash.
CVE-2024-58055	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_tcm: Don't free command immediately Don't prematurely free the command. Wait for the status completion of the sense status. It can be freed then. Otherwise we will double-free the command.

CVE ID	Description
CVE-2024-58056	In the Linux kernel, the following vulnerability has been resolved: remoteproc: core: Fix ida_free call while not allocated In the rproc_alloc() function, on error, put_device(&rproc->dev) is called, leading to the call of the rproc_type_release() function. An error can occurs before ida_alloc is called. In such case in rproc_type_release(), the condition (rproc->index >= 0) is true as rproc->index has been initialized to 0. ida_free() is called reporting a warning: [4.181906] WARNING: CPU: 1 PID: 24 at lib/idr.c:525 ida_free+0x100/0x164 [4.186378] stm32-display-dsi 5a000000.dsi: Fixed dependency cycle(s) with /soc/dsi@5a000000/panel@0 [4.188854] ida_free called for id=0 which is not allocated. [4.198256] mipi-dsi 5a000000.dsi.0: Fixed dependency cycle(s) with /soc/dsi@5a000000 [4.203556] Modules linked in: panel_orisetech_otm8009a dw_mipi_dsi_stm(+) gpu_sched dw_mipi_dsi stm32_rproc stm32_crc32 stm32_ipcc(+) optee(+) [4.224307] CPU: 1 UID: 0 PID: 24 Comm: kworker/u10:0 Not tainted 6.12.0 #442 [4.231481] Hardware name: STM32 (Device Tree Support) [4.236627] Workqueue: events_unbound deferred_probe_work_func [4.242504] Call trace: [4.242522] unwind_backtrace from show_stack+0x10/0x14 [4.250218] show_stack from dump_stack_lvl+0x50/0x64 [4.255274] dump_stack_lvl from __warn+0x80/0x12c [4.260134] __warn from warn_slowpath_fmt+0x114/0x188 [4.265199] warn_slowpath_fmt from ida_free+0x100/0x164 [4.270565] ida_free from rproc_type_release+0x38/0x60 [4.275832] rproc_type_release from device_release+0x30/0xa0 [4.281601] device_release from kobject_put+0xc4/0x294 [4.286762] kobject_put from rproc_alloc.part.0+0x208/0x28c [4.292430] rproc_alloc.part.0 from devm_rproc_alloc+0x80/0xc4 [4.298393] devm_rproc_alloc from stm32_rproc_probe+0xd0/0x844 [stm32_rproc] [4.305575] stm32_rproc_probe [stm32_rproc] from platform_probe+0x5c/0xbc Calling ida_alloc earlier in rproc_alloc ensures that the rproc->index is properly set.
CVE-2024-58057	In the Linux kernel, the following vulnerability has been resolved: idpf: convert workqueues to unbound When a workqueue is created with 'WQ_UNBOUND', its work items are served by special worker-pools, whose host workers are not bound to any specific CPU. In the default configuration (i.e. when 'queue_delayed_work' and friends do not specify which CPU to run the work item on), 'WQ_UNBOUND' allows the work item to be executed on any CPU in the same node of the CPU it was enqueued on. While this solution potentially sacrifices locality, it avoids contention with other processes that might dominate the CPU time of the processor the work item was scheduled on. This is not just a theoretical problem: in a particular scenario misconfigured process was hogging most of the time from CPU0, leaving less than 0.5% of its CPU time to the kworker. The IDPF workqueues that were using the kworker on CPU0 suffered large completion delays as a result, causing performance degradation, timeouts and eventual system crash. * I have also run a manual test to gauge the performance improvement. The test consists of an antagonist process (./stress --cpu 2') consuming as much of CPU 0 as possible. This process is run under 'taskset 01' to bind it to CPU0, and its priority is changed with 'chrt -pQ 9900 10000 \${pid}' and 'renice -n -20 \${pid}' after start. Then, the IDPF driver is forced to prefer CPU0 by editing all calls to 'queue_delayed_work', 'mod_delayed_work', etc.. to use CPU 0. Finally, 'ktraces' for the workqueue events are collected. Without the current patch, the antagonist process can force arbitrary delays between 'workqueue_queue_work' and 'workqueue_execute_start', that in my tests were as high as '30ms'. With the current patch applied, the workqueue can be migrated to another unloaded CPU in the same node, and, keeping everything else equal, the maximum delay I could see was '6us'.
CVE-2024-58058	In the Linux kernel, the following vulnerability has been resolved: ubifs: skip dumping tnc tree when zroot is null Clearing slab cache will free all znode in memory and make c->zroot.znode = NULL, then dumping tnc tree will access c->zroot.znode which cause null pointer dereference.
CVE-2024-58061	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: prohibit deactivating all links In the internal API this calls this is a WARN_ON, but that should remain since internally we want to know about bugs that may cause this. Prevent deactivating all links in the debugfs write directly.
CVE-2024-58063	In the Linux kernel, the following vulnerability has been resolved: wifi: rtlwifi: fix memory leaks and invalid access at probe error path Deinitialize at reverse order when probe fails. When init_sw_vars fails, rtl_deinit_core should not be called, specially now that it destroys the rtl_wq workqueue. And call rtl_pci_deinit and deinit_sw_vars, otherwise, memory will be leaked. Remove pci_set_drvdata call as it will already be cleaned up by the core driver code and could lead to memory leaks too. cf. commit 8d450935ae7f ("wireless: rtlwifi: remove unnecessary pci_set_drvdata()") and commit 3d86b93064c7 ("rtlwifi: Fix PCI probe error path orphaned memory").
CVE-2024-58068	In the Linux kernel, the following vulnerability has been resolved: OPP: fix dev_pm_opp_find_bw_*(*) when bandwidth table not initialized If a driver calls dev_pm_opp_find_bw_ceil/floor() the retrieve bandwidth from the OPP table but the bandwidth table was not created because the interconnect properties were missing in the OPP consumer node, the kernel will crash with: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000004 ... pc : _read_bw+0x8/0x10 lr : _opp_table_find_key+0x9c/0x174 ... Call trace: _read_bw+0x8/0x10 (P) _opp_table_find_key+0x9c/0x174 (L) _find_key+0x98/0x168 dev_pm_opp_find_bw_ceil+0x50/0x88 ... In order to fix the crash, create an assert function to check if the bandwidth table was created before trying to get a bandwidth with _read_bw().

CVE ID	Description
CVE-2024-58069	In the Linux kernel, the following vulnerability has been resolved: rtc: pcf85063: fix potential OOB write in PCF85063 NVMEM read The nvmem interface supports variable buffer sizes, while the regmap interface operates with fixed-size storage. If an nvmem client uses a buffer size less than 4 bytes, regmap_read will write out of bounds as it expects the buffer to point at an unsigned int. Fix this by using an intermediary unsigned int to hold the value.
CVE-2024-58070	In the Linux kernel, the following vulnerability has been resolved: bpf: bpf_local_storage: Always use bpf_mem_alloc in PREEMPT_RT In PREEMPT_RT, kcalloc(GFP_ATOMIC) is still not safe in non preemptible context. bpf_mem_alloc must be used in PREEMPT_RT. This patch is to enforce bpf_mem_alloc in the bpf_local_storage when CONFIG_PREEMPT_RT is enabled. [35.118559] BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 [35.118566] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1832, name: test_progs [35.118569] preempt_count: 1, expected: 0 [35.118571] RCU nest depth: 1, expected: 1 [35.118577] INFO: lockdep is turned off. ... [35.118647] __might_resched+0x433/0x5b0 [35.118677] rt_spin_lock+0xc3/0x290 [35.118700] __slab_alloc+0x72/0xc40 [35.118723] __kmallocc_noprof+0x13f/0x4e0 [35.118732] bpf_map_kzalloc+0xe5/0x220 [35.118740] bpf_selem_alloc+0x1d2/0x7b0 [35.118755] bpf_local_storage_update+0x2fa/0x8b0 [35.118784] bpf_sk_storage_get_tracing+0x15a/0x1d0 [35.118791] bpf_prog_9a118d86fca78ebb_trace_inet_sock_set_state+0x44/0x66 [35.118795] bpf_trace_run3+0x222/0x400 [35.118820] __bpf_trace_inet_sock_set_state+0x11/0x20 [35.118824] trace_inet_sock_set_state+0x112/0x130 [35.118830] inet_sk_state_store+0x41/0x90 [35.118836] tcp_set_state+0x3b3/0x640 There is no need to adjust the gfp_flags passing to the bpf_mem_cache_alloc_flags() which only honors the GFP_KERNEL. The verifier has ensured GFP_KERNEL is passed only in sleepable context. It has been an old issue since the first introduction of the bpf_local_storage ~5 years ago, so this patch targets the bpf-next. bpf_mem_alloc is needed to solve it, so the Fixes tag is set to the commit when bpf_mem_alloc was first used in the bpf_local_storage.
CVE-2024-58071	In the Linux kernel, the following vulnerability has been resolved: team: prevent adding a device which is already a team device lower Prevent adding a device which is already a team device lower, e.g. adding veth0 if vlan1 was already added and veth0 is a lower of vlan1. This is not useful in practice and can lead to recursive locking: \$ ip link add veth0 type veth peer name veth1 \$ ip link set veth0 up \$ ip link set veth1 up \$ ip link add link veth0 name veth0.1 type vlan protocol 802.1Q id 1 \$ ip link add team0 type team \$ ip link set veth0.1 down \$ ip link set veth0.1 master team0 team0: Port device veth0.1 added \$ ip link set veth0 down \$ ip link set veth0 master team0 ===== WARNING: possible recursive locking detected 6.13.0-rc2-virtme-00441-ga14a429069bb #46 Not tainted ----- ip/7684 is trying to acquire lock: ffff888016848e00 (team->team_lock_key){+..-}{4:4}, at: team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) but task is already holding lock: ffff888016848e00 (team->team_lock_key){+..-}{4:4}, at: team_add_slave (drivers/net/team/team_core.c:1147 drivers/net/team/team_core.c:1977) other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(team->team_lock_key); lock(team->team_lock_key); *** DEADLOCK *** May be due to missing lock nesting notation 2 locks held by ip/7684: stack backtrace: CPU: 3 UID: 0 PID: 7684 Comm: ip Not tainted 6.13.0-rc2-virtme-00441-ga14a429069bb #46 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:122) print_deadlock_bug.cold (kernel/locking/lockdep.c:3040) __lock_acquire (kernel/locking/lockdep.c:3893 kernel/locking/lockdep.c:5226) ? netlink_broadcast_filtered (net/netlink/af_netlink.c:1548) lock_acquire.part.0 (kernel/locking/lockdep.c:467 kernel/locking/lockdep.c:5851) ? team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) ? trace_lock_acquire (/include/trace/events/lock.h:24 (discriminator 2)) ? team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) ? lock_acquire (kernel/locking/lockdep.c:5822) ? team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) __mutex_lock (kernel/locking/mutex.c:587 kernel/locking/mutex.c:735) ? team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) ? team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) ? fib_sync_up (net/ipv4/fib_semantics.c:2167) ? team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) team_device_event (drivers/net/team/team_core.c:2928 drivers/net/team/team_core.c:2951 drivers/net/team/team_core.c:2973) notifier_call_chain (kernel/notifier.c:85) call_netdevice_notifiers_info (net/core/dev.c:1996) __dev_notify_flags (net/core/dev.c:8993) ? __dev_change_flags (net/core/dev.c:8975) dev_change_flags (net/core/dev.c:9027) vlan_device_event (net/8021q/vlan.c:85 net/8021q/vlan.c:470) ? br_device_event (net/bridge/br.c:143) notifier_call_chain (kernel/notifier.c:85) call_netdevice_notifiers_info (net/core/dev.c:1996) dev_open (net/core/dev.c:1519 net/core/dev.c:1505) team_add_slave (drivers/net/team/team_core.c:1219 drivers/net/team/team_core.c:1977) ? __pfx_team_add_slave (drivers/net/team/team_core.c:1972) do_set_master (net/core/rtnetlink.c:2917) do_setlink.isra.0 (net/core/rtnetlink.c:3117)

[illegible]

CVE ID	Description
CVE-2024-58081	In the Linux kernel, the following vulnerability has been resolved: clk: mmp2: call pm_genpd_init() only after genpd.name is set Setting the genpd's struct device's name with dev_set_name() is happening within pm_genpd_init(). If it remains NULL, things can blow up later, such as when crafting the devfs hierarchy for the power domain: Unable to handle kernel NULL pointer dereference at virtual address 00000000 when read ... Call trace: strlen from start_creating+0x90/0x138 start_creating from debugfs_create_dir+0x20/0x178 debugfs_create_dir from genpd_debug_add.part.0+0x4c/0x144 genpd_debug_add.part.0 from genpd_debug_init+0x74/0x90 genpd_debug_init from do_one_initcall+0x5c/0x244 do_one_initcall from kernel_init_freeable+0x19c/0x1f4 kernel_init_freeable from kernel_init+0x1c/0x12c kernel_init from ret_from_fork+0x14/0x28 Bisecting tracks this crash back to commit 899f44531fe6 ("pmdomain: core: Add GENPD_FLAG_DEV_NAME_FW flag"), which exchanges use of genpd->name with dev_name(&genpd->dev) in genpd_debug_add.part().
CVE-2024-58082	In the Linux kernel, the following vulnerability has been resolved: media: nuvoton: Fix an error check in npcm_video_ece_init() When function of_find_device_by_node() fails, it returns NULL instead of an error code. So the corresponding error check logic should be modified to check whether the return value is NULL and set the error code to be returned as -ENODEV.
CVE-2024-58083	In the Linux kernel, the following vulnerability has been resolved: KVM: Explicitly verify target vCPU is online in kvm_get_vcpu() Explicitly verify the target vCPU is fully online _prior_ to clamping the index in kvm_get_vcpu(). If the index is "bad", the nospec clamping will generate '0', i.e. KVM will return vCPU0 instead of NULL. In practice, the bug is unlikely to cause problems, as it will only come into play if userspace or the guest is buggy or misbehaving, e.g. KVM may send interrupts to vCPU0 instead of dropping them on the floor. However, returning vCPU0 when it shouldn't exist per online_vcpus is problematic now that KVM uses an xarray for the vCPUs array, as KVM needs to insert into the xarray before publishing the vCPU to userspace (see commit c5b077549136 ("KVM: Convert the kvm->vcpus array to a xarray")), i.e. before vCPU creation is guaranteed to succeed. As a result, incorrectly providing access to vCPU0 will trigger a use-after-free if vCPU0 is dereferenced and kvm_vm_ioctl_create_vcpu() bails out of vCPU creation due to an error and frees vCPU0. Commit afb2acb2e3a3 ("KVM: Fix vcpu_array[0] races") papered over that issue, but in doing so introduced an unsolvable teardown conundrum. Preventing accesses to vCPU0 before it's fully online will allow reverting commit afb2acb2e3a3, without re-introducing the vcpu_array[0] UAF race.
CVE-2024-58085	In the Linux kernel, the following vulnerability has been resolved: tomoyo: don't emit warning in tomoyo_write_control() syzbot is reporting too large allocation warning at tomoyo_write_control(), for one can write a very very long line without new line character. To fix this warning, I use __GFP_NOWARN rather than checking for KMALLOC_MAX_SIZE, for practically a valid line should be always shorter than 32KB where the "too small to fail" memory-allocation rule applies. One might try to write a valid line that is longer than 32KB, but such request will likely fail with -ENOMEM. Therefore, I feel that separately returning -EINVAL when a line is longer than KMALLOC_MAX_SIZE is redundant. There is no need to distinguish over-32KB and over-KMALLOC_MAX_SIZE.
CVE-2024-58086	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Stop active perfmon if it is being destroyed If the active performance monitor ('v3d->active_perfmon') is being destroyed, stop it first. Currently, the active perfmon is not stopped during destruction, leaving the 'v3d->active_perfmon' pointer stale. This can lead to undefined behavior and instability. This patch ensures that the active perfmon is stopped before being destroyed, aligning with the behavior introduced in commit 7d1fd3638ee3 ("drm/v3d: Stop the active perfmon before being destroyed").
CVE-2024-58087	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix racy issue from session lookup and expire Increment the session reference count within the lock for lookup to avoid racy issue with session expire.

CVE ID	Description
CVE-2024-58088	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix deadlock when freeing cgroup storage The following commit bc235cdb423a ("bpf: Prevent deadlock from recursive bpf_task_storage_[get delete]") first introduced deadlock prevention for fentry/fexit programs attaching on bpf_task_storage helpers. That commit also employed the logic in map free path in its v6 version. Later bpf_cgrp_storage was first introduced in c4bcfb38a95e ("bpf: Implement cgroup storage available to non-cgroup-attached bpf progs") which faces the same issue as bpf_task_storage, instead of its busy counter, NULL was passed to bpf_local_storage_map_free() which opened a window to cause deadlock: <TASK> (acquiring local_storage->lock) _raw_spin_lock_irqsave +0x3d/0x50 bpf_local_storage_update+0xd1/0x460 bpf_cgrp_storage_get+0x109/0x130 bpf_prog_a4d4a370ba857314_cgrp_ptr+0x139/0x170 ? __bpf_prog_enter_recur+0x16/0x80 bpf_trampoline_6442485186+0x43/0xa4 cgroup_storage_ptr+0x9/0x20 (holding local_storage->lock) bpf_selem_unlink_storage_nolock.constprop.0+0x135/0x160 bpf_selem_unlink_storage+0x6f/0x110 bpf_local_storage_map_free+0xa2/0x110 bpf_map_free_deferred+0x5b/0x90 process_one_work +0x17c/0x390 worker_thread+0x251/0x360 kthread+0xd2/0x100 ret_from_fork+0x34/0x50 ret_from_fork_asm+0x1a/0x30 </TASK> Progs: - A: SEC("fentry/cgroup_storage_ptr") - cgid (BPF_MAP_TYPE_HASH) Record the id of the cgroup the current task belonging to in this hash map, using the address of the cgroup as the map key. - cgrpa (BPF_MAP_TYPE_CGRP_STORAGE) If current task is a kworker, lookup the above hash map using function parameter @owner as the key to get its corresponding cgroup id which is then used to get a trusted pointer to the cgroup through bpf_cgroup_from_id(). This trusted pointer can then be passed to bpf_cgrp_storage_get() to finally trigger the deadlock issue. - B: SEC("tp_btf/sys_enter") - cgrpb (BPF_MAP_TYPE_CGRP_STORAGE) The only purpose of this prog is to fill Prog A's hash map by calling bpf_cgrp_storage_get() for as many userspace tasks as possible. Steps to reproduce: - Run A; - while (true) { Run B; Destroy B; } Fix this issue by passing its busy counter to the free procedure so it can be properly incremented before storage/smmap locking.
CVE-2024-58089	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix double accounting race when btrfs_run_delalloc_range() failed [BUG] When running btrfs with block size (4K) smaller than page size (64K, aarch64), there is a very high chance to crash the kernel at generic/750, with the following messages: (before the call traces, there are 3 extra debug messages added) BTRFS warning (device dm-3): read-write for sector size 4096 with page size 65536 is experimental BTRFS info (device dm-3): checking UUID tree hrtimer: interrupt took 5451385 ns BTRFS error (device dm-3): cow_file_range failed, root=4957 inode=257 start=1605632 len=69632: -28 BTRFS error (device dm-3): run_delalloc_nocow failed, root=4957 inode=257 start=1605632 len=69632: -28 BTRFS error (device dm-3): failed to run delalloc range, root=4957 ino=257 folio=1572864 submit_bitmap=8-15 start=1605632 len=69632: -28 -----[cut here]----- WARNING: CPU: 2 PID: 3020984 at ordered-data.c:360 can_finish_ordered_extnt+0x370/0x3b8 [btrfs] CPU: 2 UID: 0 PID: 3020984 Comm: kworker/u24:1 Tainted: G OE 6.13.0-rc1-custom+ #89 Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022 Workqueue: events_unbound btrfs_async_reclaim_data_space [btrfs] pc : can_finish_ordered_extnt+0x370/0x3b8 [btrfs] lr : can_finish_ordered_extnt+0x1ec/0x3b8 [btrfs] Call trace: can_finish_ordered_extnt+0x370/0x3b8 [btrfs] (P) can_finish_ordered_extnt+0x1ec/0x3b8 [btrfs] (L) btrfs_mark_ordered_io_finished+0x130/0x2b8 [btrfs] extnt_writepage+0x10c/0x3b8 [btrfs] extnt_write_cache_pages+0x21c/0x4e8 [btrfs] btrfs_writepages+0x94/0x160 [btrfs] do_writepages +0x74/0x190 filemap_fdatawrite_wbc+0x74/0xa0 start_delalloc_inodes+0x17c/0x3b0 [btrfs] btrfs_start_delalloc_nobg+0x17c/0x288 [btrfs] shrink_delalloc+0x11c/0x280 [btrfs] flush_space +0x288/0x328 [btrfs] btrfs_async_reclaim_data_space+0x180/0x228 [btrfs] process_one_work +0x228/0x680 worker_thread+0x1bc/0x360 kthread+0x100/0x118 ret_from_fork+0x10/0x20 ---[end trace 0000000000000000]--- BTRFS critical (device dm-3): bad ordered extent accounting, root=4957 ino=257 OE offset=1605632 OE len=16384 to_dec=16384 left=0 BTRFS critical (device dm-3): bad ordered extent accounting, root=4957 ino=257 OE offset=1622016 OE len=12288 to_dec=12288 left=0 Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 BTRFS critical (device dm-3): bad ordered extent accounting, root=4957 ino=257 OE offset=1634304 OE len=8192 to_dec=4096 left=0 CPU: 1 UID: 0 PID: 3286940 Comm: kworker/u24:3 Tainted: G W OE 6.13.0-rc1-custom+ #89 Hardware name: QEMU KVM Virtual Machine, BIOS unknown 2/2/2022 Workqueue: btrfs_work_helper [btrfs] (btrfs-endio-write) pstate: 404000c5 (nZcv daIf +PAN-UAO -TCO -DIT -SSBS BTYPED) pc : process_one_work+0x110/0x680 lr : worker_thread+0x1bc/0x360 Call trace: process_one_work+0x110/0x680 (P) worker_thread+0x1bc/0x360 (L) worker_thread+0x1bc/0x360 kthread+0x100/0x118 ret_from_fork+0x10/0x20 Code: f84086a1 f9000fe1 53041c21 b9003361 (f9400661) ---[end trace 0000000000000000]--- Kernel panic - not syncing: Oops: Fatal exception SMP: stopping secondary CPUs SMP: failed to stop secondary CPUs 2-3 Dumping ftrace buffer: (ftrace buffer empty) Kernel Offset: 0x275bb9540000 from 0xffff800080000000 PHYS_OFFSET: 0xffff8fbba0000000 CPU features: 0x100,00000070,00801250,8201720b [CAUSE] The above warning is triggered immediately after the delalloc range failure, this happens in the following sequence: - Range [1568K, 1636K] is dirty 1536K 1568K 1600K 1636K 1664K Where 1536K, 1600K and 1664K are page boundaries (64K page size) - Enter extnt_writepage() for page 1536K - Enter run_delalloc_nocow() with locke ---truncated---

CVE ID	Description
CVE-2024-58090	In the Linux kernel, the following vulnerability has been resolved: sched/core: Prevent rescheduling when interrupts are disabled David reported a warning observed while loop testing kexec jump: Interrupts enabled after irqrouter_resume+0x0/0x50 WARNING: CPU: 0 PID: 560 at drivers/base/syscore.c:103 syscore_resume+0x18a/0x220 kernel_kexec+0xf6/0x180 __do_sys_reboot+0x206/0x250 do_syscall_64+0x95/0x180 The corresponding interrupt flag trace: hardirqs last enabled at (15573): [<ffffffffff8281b8e>] __up_console_sem+0x7e/0x90 hardirqs last disabled at (15580): [<ffffffffff8281b73>] __up_console_sem+0x63/0x90 That means __up_console_sem() was invoked with interrupts enabled. Further instrumentation revealed that in the interrupt disabled section of kexec jump one of the syscore_suspend() callbacks woke up a task, which set the NEED_RESCHED flag. A later callback in the resume path invoked cond_resched() which in turn led to the invocation of the scheduler: __cond_resched+0x21/0x60 down_timeout+0x18/0x60 acpi_os_wait_semaphore+0x4c/0x80 acpi_ut_acquire_mutex+0x3d/0x100 acpi_ns_get_node+0x27/0x60 acpi_ns_evaluate+0x1cb/0x2d0 acpi_rs_set_srs_method_data+0x156/0x190 acpi_pci_link_set+0x11c/0x290 irqrouter_resume+0x54/0x60 syscore_resume+0x6a/0x200 kernel_kexec+0x145/0x1c0 __do_sys_reboot+0xeb/0x240 do_syscall_64+0x95/0x180 This is a long standing problem, which probably got more visible with the recent printk changes. Something does a task wakeup and the scheduler sets the NEED_RESCHED flag. cond_resched() sees it set and invokes schedule() from a completely bogus context. The scheduler enables interrupts after context switching, which causes the above warning at the end. Quite some of the code paths in syscore_suspend()/resume() can result in triggering a wakeup with the exactly same consequences. They might not have done so yet, but as they share a lot of code with normal operations it's just a question of time. The problem only affects the PREEMPT_NONE and PREEMPT_VOLUNTARY scheduling models. Full preemption is not affected as cond_resched() is disabled and the preemption check preemptible() takes the interrupt disabled flag into account. Cure the problem by adding a corresponding check into cond_resched().
CVE-2024-58092	In the Linux kernel, the following vulnerability has been resolved: nfsd: fix legacy client tracking initialization Get rid of the nfsd4_legacy_tracking_ops->init() call in check_for_legacy_methods(). That will be handled in the caller (nfsd4_client_tracking_init()). Otherwise, we'll wind up calling nfsd4_legacy_tracking_ops->init() twice, and the second time we'll trigger the BUG_ON() in nfsd4_init_recdir().
CVE-2024-58093	In the Linux kernel, the following vulnerability has been resolved: PCI/ASPM: Fix link state exit during switch upstream function removal Before 456d8aa37d0f ("PCI/ASPM: Disable ASPM on MFD function removal to avoid use-after-free"), we would free the ASPM link only after the last function on the bus pertaining to the given link was removed. That was too late. If function 0 is removed before sibling function, link->downstream would point to free'd memory after. After above change, we freed the ASPM parent link state upon any function removal on the bus pertaining to a given link. That is too early. If the link is to a PCIe switch with MFD on the upstream port, then removing functions other than 0 first would free a link which still remains parent link to the remaining downstream ports. The resulting GPFs are especially frequent during hot-unplug, because pciehp removes devices on the link bus in reverse order. On that switch, function 0 is the virtual P2P bridge to the internal bus. Free exactly when function 0 is removed -- before the parent link is obsolete, but after all subordinate links are gone. [kwilczynski: commit log]
CVE-2024-58094	In the Linux kernel, the following vulnerability has been resolved: jfs: add check read-only before truncation in jfs_truncate_nolock() Added a check for "read-only" mode in the `jfs_truncate_nolock` function to avoid errors related to writing to a read-only filesystem. Call stack: block_write_begin() { jfs_write_failed() { jfs_truncate() { jfs_truncate_nolock() { txEnd() { ... log = JFS_SBI(tblk->sb)->log; // (log == NULL) If the `isReadOnly(ip)` condition is triggered in `jfs_truncate_nolock`, the function execution will stop, and no further data modification will occur. Instead, the `xtTruncate` function will be called with the "COMMIT_WMAP" flag, preventing modifications in "read-only" mode.
CVE-2024-58095	In the Linux kernel, the following vulnerability has been resolved: jfs: add check read-only before txBeginAnon() call Added a read-only check before calling `txBeginAnon` in `extAlloc` and `extRecord`. This prevents modification attempts on a read-only mounted filesystem, avoiding potential errors or crashes. Call trace: txBeginAnon+0xac/0x154 extAlloc+0xe8/0xddec fs/jfs/jfs_extent.c:78 jfs_get_block+0x340/0xb98 fs/jfs/inode.c:248 __block_write_begin_int+0x580/0x166c fs/buffer.c:2128 __block_write_begin fs/buffer.c:2177 [inline] block_write_begin+0x98/0x11c fs/buffer.c:2236 jfs_write_begin+0x44/0x88 fs/jfs/inode.c:299

CVE ID	Description
CVE-2024-58096	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: add srng->lock for ath11k_hal_srng_* in monitor mode ath11k_hal_srng_* should be used with srng->lock to protect srng data. For ath11k_dp_rx_mon_dest_process() and ath11k_dp_full_mon_process_rx(), they use ath11k_hal_srng_* for many times but never call srng->lock. So when running (full) monitor mode, warning will occur: RIP: 0010:ath11k_hal_srng_dst_peek+0x18/0x30 [ath11k] Call Trace: ? ath11k_hal_srng_dst_peek+0x18/0x30 [ath11k] ath11k_dp_rx_process_mon_status+0xc45/0x1190 [ath11k] ? idr_alloc_u32+0x97/0xd0 ath11k_dp_rx_process_mon_rings+0x32a/0x550 [ath11k] ath11k_dp_service_srng+0x289/0x5a0 [ath11k] ath11k_pcic_ext_grp_napi_poll+0x30/0xd0 [ath11k] __napi_poll+0x30/0x1f0 net_rx_action+0x198/0x320 __do_softirq+0xdd/0x319 So add srng->lock for them to avoid such warnings. In order to fetch the srng->lock, should change srng's definition from 'void' to 'struct hal_srng'. And initialize them elsewhere to prevent one line of code from being too long. This is consistent with other ring process functions, such as ath11k_dp_process_rx(). Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3.6510.30 Tested-on: QCN9074 hw1.0 PCI WLAN.HK.2.7.0.1-01744-QCAHKSPL_SILICONZ-1
CVE-2024-58097	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix RCU stall while reaping monitor destination ring While processing the monitor destination ring, MSDUs are reaped from the link descriptor based on the corresponding buf_id. However, sometimes the driver cannot obtain a valid buffer corresponding to the buf_id received from the hardware. This causes an infinite loop in the destination processing, resulting in a kernel crash. kernel log: ath11k_pci 0000:58:00.0: data msdu_pop: invalid buf_id 309 ath11k_pci 0000:58:00.0: data dp_rx_monitor_link_desc_return failed ath11k_pci 0000:58:00.0: data msdu_pop: invalid buf_id 309 ath11k_pci 0000:58:00.0: data dp_rx_monitor_link_desc_return failed Fix this by skipping the problematic buf_id and reaping the next entry, replacing the break with the next MSDU processing. Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3.6510.30 Tested-on: QCN9074 hw1.0 PCI WLAN.HK.2.7.0.1-01744-QCAHKSPL_SILICONZ-1
CVE-2024-58098	In the Linux kernel, the following vulnerability has been resolved: bpf: track changes_pkt_data property for global functions When processing calls to certain helpers, verifier invalidates all packet pointers in a current state. For example, consider the following program: __attribute__((__noinline__)) long skb_pull_data(struct __sk_buff *sk, __u32 len) { return bpf_skb_pull_data(sk, len); } SEC("tc") int test_invalidate_checks(struct __sk_buff *sk) { int *p = (void *) (long)sk->data; if ((void *) (p + 1) > (void *) (long)sk->data_end) return TCX_DROP; skb_pull_data(sk, 0); *p = 42; return TCX_PASS; } After a call to bpf_skb_pull_data() the pointer 'p' can't be used safely. See function filter.c:bpf_helper_changes_pkt_data() for a list of such helpers. At the moment verifier invalidates packet pointers when processing helper function calls, and does not traverse global sub-programs when processing calls to global sub-programs. This means that calls to helpers done from global sub-programs do not invalidate pointers in the caller state. E.g. the program above is unsafe, but is not rejected by verifier. This commit fixes the omission by computing field bpf_subprog_info->changes_pkt_data for each sub-program before main verification pass. changes_pkt_data should be set if: - subprogram calls helper for which bpf_helper_changes_pkt_data returns true; - subprogram calls a global function, for which bpf_subprog_info->changes_pkt_data should be set. The verifier.c:check_cfg() pass is modified to compute this information. The commit relies on depth first instruction traversal done by check_cfg() and absence of recursive function calls: - check_cfg() would eventually visit every call to subprogram S in a state when S is fully explored; - when S is fully explored: - every direct helper call within S is explored (and thus changes_pkt_data is set if needed); - every call to subprogram S1 called by S was visited with S1 fully explored (and thus S inherits changes_pkt_data from S1). The downside of such approach is that dead code elimination is not taken into account: if a helper call inside global function is dead because of current configuration, verifier would conservatively assume that the call occurs for the purpose of the changes_pkt_data computation.
CVE-2024-58099	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: Fix packet corruption in vmxnet3_xdp_xmit_frame Andrew and Nikolay reported connectivity issues with Cilium's service load-balancing in case of vmxnet3. If a BPF program for native XDP adds an encapsulation header such as IPIP and transmits the packet out the same interface, then in case of vmxnet3 a corrupted packet is being sent and subsequently dropped on the path. vmxnet3_xdp_xmit_frame() which is called e.g. via vmxnet3_run_xdp() through vmxnet3_xdp_xmit_back() calculates an incorrect DMA address: page = virt_to_page(xdpf->data); tbi->dma_addr = page_pool_get_dma_addr(page) + VMXNET3_XDP_HEADROOM; dma_sync_single_for_device(&adapter->pdev->dev, tbi->dma_addr, buf_size, DMA_TO_DEVICE); The above assumes a fixed offset (VMXNET3_XDP_HEADROOM), but the XDP BPF program could have moved xdp->data. While the passed buf_size is correct (xdpf->len), the dma_addr needs to have a dynamic offset which can be calculated as xdpf->data - (void *)xdpf, that is, xdp->data - xdp->data_hard_start.

CVE ID	Description
CVE-2024-58100	In the Linux kernel, the following vulnerability has been resolved: bpf: check changes_pkt_data property for extension programs When processing calls to global sub-programs, verifier decides whether to invalidate all packet pointers in current state depending on the changes_pkt_data property of the global sub-program. Because of this, an extension program replacing a global sub-program must be compatible with changes_pkt_data property of the sub-program being replaced. This commit: - adds changes_pkt_data flag to struct bpf_prog_aux: - this flag is set in check_cfg() for main sub-program; - in jit_subprogs() for other sub-programs; - modifies bpf_check_attach_btf_id() to check changes_pkt_data flag; - moves call to check_attach_btf_id() after the call to check_cfg(), because it needs changes_pkt_data flag to be set: bpf_check: ... - check_attach_btf_id resolve_pseudo_ldimm64 resolve_pseudo_ldimm64 --> bpf_prog_is_offloaded bpf_prog_is_offloaded check_cfg check_cfg + check_attach_btf_id ... The following fields are set by check_attach_btf_id(): - env->ops - prog->aux->attach_btf_trace - prog->aux->attach_func_name - prog->aux->attach_func_proto - prog->aux->dst_trampoline - prog->aux->mod - prog->aux->saved_dst_attach_type - prog->aux->saved_dst_prog_type - prog->expected_attach_type Neither of these fields are used by resolve_pseudo_ldimm64() or bpf_prog_offload_verifier_prep() (for netronome and netdevsim drivers), so the reordering is safe.
CVE-2024-58237	In the Linux kernel, the following vulnerability has been resolved: bpf: consider that tail calls invalidate packet pointers Tail-called programs could execute any of the helpers that invalidate packet pointers. Hence, conservatively assume that each tail call invalidates packet pointers. Making the change in bpf_helper_changes_pkt_data() automatically makes use of check_cfg() logic that computes 'changes_pkt_data' effect for global sub-programs, such that the following program could be rejected: int tail_call(struct __sk_buff *sk) { bpf_tail_call_static(sk, &jmp_table, 0); return 0; } SEC("tc") int not_safe(struct __sk_buff *sk) { int *p = (void *) (long)sk->data; ... make p valid ... tail_call(sk); *p = 42; /* this is unsafe */ ... } The tc_bpf2bpf.c:subprog_tc() needs change: mark it as a function that can invalidate packet pointers. Otherwise, it can't be replaced with tailcall_freplace.c:entry_freplace() that does a tail call.
CVE-2024-58239	In the Linux kernel, the following vulnerability has been resolved: tls: stop recv() if initial process_rx_list gave us non-DATA If we have a non-DATA record on the rx_list and another record of the same type still on the queue, we will end up merging them: - process_rx_list copies the non-DATA record - we start the loop and process the first available record since it's of the same type - we break out of the loop since the record was not DATA Just check the record type and jump to the end in case process_rx_list did some work.
CVE-2024-58240	In the Linux kernel, the following vulnerability has been resolved: tls: separate no-async decryption request handling from async If we're not doing async, the handling is much simpler. There's no reference counting, we just need to wait for the completion to wake us up and return its result. We should preferably also use a separate crypto_wait. I'm not seeing a UAF as I did in the past, I think aec7961916f3 ("tls: fix race between async notify and socket close") took care of it. This will make the next fix easier.
CVE-2024-58241	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Disable works on hci_unregister_dev This make use of disable_work_* on hci_unregister_dev since the hci_dev is about to be freed new submissions are not disarable.
CVE-2024-58251	In netstat in BusyBox through 1.37.0, local users can launch of network application with an argv[0] containing an ANSI terminal escape sequence, leading to a denial of service (terminal locked up) when netstat is used by a victim.
CVE-2024-6104	go-retryablehttp prior to 0.7.7 did not sanitize urls when writing them to its log file. This could lead to go-retryablehttp writing sensitive HTTP basic auth credentials to its log file. This vulnerability, CVE-2024-6104, was fixed in go-retryablehttp 0.7.7.
CVE-2024-6119	Issue summary: Applications performing certificate name checks (e.g., TLS clients checking server certificates) may attempt to read an invalid memory address resulting in abnormal termination of the application process. Impact summary: Abnormal termination of an application can cause a denial of service. Applications performing certificate name checks (e.g., TLS clients checking server certificates) may attempt to read an invalid memory address when comparing the expected name with an `otherName` subject alternative name of an X.509 certificate. This may result in an exception that terminates the application program. Note that basic certificate chain validation (signatures, dates, ...) is not affected, the denial of service can occur only when the application also specifies an expected DNS name, Email address or IP address. TLS servers rarely solicit client certificates, and even when they do, they generally don't perform a name check against a reference identifier (expected identity), but rather extract the presented identity after checking the certificate chain. So TLS servers are generally not affected and the severity of the issue is Moderate. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue.

CVE ID	Description
CVE-2024-6232	There is a MEDIUM severity vulnerability affecting CPython. Regular expressions that allowed excessive backtracking during tarfile.TarFile header parsing are vulnerable to ReDoS via specifically-crafted tar archives.
CVE-2024-6345	A vulnerability in the package_index module of pypa/setuptools versions up to 69.1.1 allows for remote code execution via its download functions. These functions, which are used to download packages from URLs provided by users or retrieved from package index servers, are susceptible to code injection. If these functions are exposed to user-controlled inputs, such as package URLs, they can execute arbitrary commands on the system. The issue is fixed in version 70.0.
CVE-2024-6655	A flaw was found in the GTK library. Under certain conditions, it is possible for a library to be injected into a GTK application from the current working directory.
CVE-2024-6763	Eclipse Jetty is a lightweight, highly scalable, Java-based web server and Servlet engine . It includes a utility class, HttpURI, for URI/URL parsing. The HttpURI class does insufficient validation on the authority segment of a URI. However the behaviour of HttpURI differs from the common browsers in how it handles a URI that would be considered invalid if fully validated against the RRC. Specifically HttpURI and the browser may differ on the value of the host extracted from an invalid URI and thus a combination of Jetty and a vulnerable browser may be vulnerable to a open redirect attack or to a SSRF attack if the URI is used after passing validation checks.
CVE-2024-6923	There is a MEDIUM severity vulnerability affecting CPython. The email module didn't properly quote newlines for email headers when serializing an email message allowing for header injection when an email is serialized.
CVE-2024-7055	A vulnerability was found in FFmpeg up to 7.0.1. It has been classified as critical. This affects the function pnm_decode_frame in the library /libavcodec/pnmdec.c. The manipulation leads to heap-based buffer overflow. It is possible to initiate the attack remotely. The exploit has been disclosed to the public and may be used. Upgrading to version 7.0.2 is able to address this issue. It is recommended to upgrade the affected component. The associated identifier of this vulnerability is VDB-273651.
CVE-2024-7254	Any project that parses untrusted Protocol Buffers data containing an arbitrary number of nested groups / series of SGROUP tags can be corrupted by exceeding the stack limit i.e. StackOverflow. Parsing nested groups as unknown fields with DiscardUnknownFieldsParser or Java Protobuf Lite parser, or against Protobuf map fields, creates unbounded recursions that can be abused by an attacker.
CVE-2024-7264	libcurl's ASN1 parser code has the `GTime2str()` function, used for parsing an ASN.1 Generalized Time field. If given a syntactically incorrect field, the parser might end up using -1 for the length of the *time fraction*, leading to a `strlen()` getting performed on a pointer to a heap buffer area that is not (purposely) null terminated. This flaw most likely leads to a crash, but can also lead to heap contents getting returned to the application when [CURLINFO_CERTINFO](https://curl.se/libcurl/c/CURLINFO_CERTINFO.html) is used.
CVE-2024-7531	Calling `PK11_Encrypt()` in NSS using CKM_CHACHA20 and the same buffer for input and output can result in plaintext on an Intel Sandy Bridge processor. In Firefox this only affects the QUIC header protection feature when the connection is using the ChaCha20-Poly1305 cipher suite. The most likely outcome is connection failure, but if the connection persists despite the high packet loss it could be possible for a network observer to identify packets as coming from the same source despite a network path change. This vulnerability affects Firefox < 129, Firefox ESR < 115.14, and Firefox ESR < 128.1.
CVE-2024-7592	There is a LOW severity vulnerability affecting CPython, specifically the 'http.cookies' standard library module. When parsing cookies that contained backslashes for quoted characters in the cookie value, the parser would use an algorithm with quadratic complexity, resulting in excess CPU resources being used while parsing the value.
CVE-2024-8088	There is a HIGH severity vulnerability affecting the CPython "zipfile" module affecting "zipfile.Path". Note that the more common API "zipfile.ZipFile" class is unaffected. When iterating over names of entries in a zip archive (for example, methods of "zipfile.Path" like "namelist()", "iterdir()", etc) the process can be put into an infinite loop with a maliciously crafted zip archive. This defect applies when reading only metadata or extracting the contents of the zip archive. Programs that are not handling user-controlled zip archives are not affected.
CVE-2024-8096	When curl is told to use the Certificate Status Request TLS extension, often referred to as OCSP stapling, to verify that the server certificate is valid, it might fail to detect some OCSP problems and instead wrongly consider the response as fine. If the returned status reports another error than 'revoked' (like for example 'unauthorized') it is not treated as a bad certificate.

CVE ID	Description
CVE-2024-8176	A stack overflow vulnerability exists in the libexpat library due to the way it handles recursive entity expansion in XML documents. When parsing an XML document with deeply nested entity references, libexpat can be forced to recurse indefinitely, exhausting the stack space and causing a crash. This issue could lead to denial of service (DoS) or, in some cases, exploitable memory corruption, depending on the environment and library usage.
CVE-2024-8184	There exists a security vulnerability in Jetty's ThreadLimitHandler.getRemote() which can be exploited by unauthorized users to cause remote denial-of-service (DoS) attack. By repeatedly sending crafted requests, attackers can trigger OutOfMemory errors and exhaust the server's memory.
CVE-2024-9042	This CVE affects only Windows worker nodes. Your worker node is vulnerable to this issue if it is running one of the affected versions listed below.
CVE-2024-9143	Issue summary: Use of the low-level GF(2 ^m) elliptic curve APIs with untrusted explicit values for the field polynomial can lead to out-of-bounds memory reads or writes. Impact summary: Out of bound memory writes can lead to an application crash or even a possibility of a remote code execution, however, in all the protocols involving Elliptic Curve Cryptography that we're aware of, either only "named curves" are supported, or, if explicit curve parameters are supported, they specify an X9.62 encoding of binary (GF(2 ^m)) curves that can't represent problematic input values. Thus the likelihood of existence of a vulnerable application is low. In particular, the X9.62 encoding is used for ECC keys in X.509 certificates, so problematic inputs cannot occur in the context of processing X.509 certificates. Any problematic use-cases would have to be using an "exotic" curve encoding. The affected APIs include: EC_GROUP_new_curve_GF2m(), EC_GROUP_new_from_params(), and various supporting BN_GF2m_*(*) functions. Applications working with "exotic" explicit binary (GF(2 ^m)) curve parameters, that make it possible to represent invalid field polynomials with a zero constant term, via the above or similar APIs, may terminate abruptly as a result of reading or writing outside of array bounds. Remote code execution cannot easily be ruled out. The FIPS modules in 3.3, 3.2, 3.1 and 3.0 are not affected by this issue.
CVE-2024-9287	A vulnerability has been found in the CPython `venv` module and CLI where path names provided when creating a virtual environment were not quoted properly, allowing the creator to inject commands into virtual environment "activation" scripts (ie "source venv/bin/activate"). This means that attacker-controlled virtual environments are able to run commands when the virtual environment is activated. Virtual environments which are not created by an attacker or which aren't activated before being used (ie "./venv/bin/python") are not affected.
CVE-2024-9681	When curl is asked to use HSTS, the expiry time for a subdomain might overwrite a parent domain's cache entry, making it end sooner or later than otherwise intended. This affects curl using applications that enable HSTS and use URLs with the insecure `HTTP://` scheme and perform transfers with hosts like `x.example.com` as well as `example.com` where the first host is a subdomain of the second host. (The HSTS cache either needs to have been populated manually or there needs to have been previous HTTPS accesses done as the cache needs to have entries for the domains involved to trigger this problem.) When `x.example.com` responds with `Strict-Transport-Security:` headers, this bug can make the subdomain's expiry timeout *bleed over* and get set for the parent domain `example.com` in curl's HSTS cache. The result of a triggered bug is that HTTP accesses to `example.com` get converted to HTTPS for a different period of time than what was asked for by the origin server. If `example.com` for example stops supporting HTTPS at its expiry time, curl might then fail to access `http://example.com` until the (wrongly set) timeout expires. This bug can also expire the parent's entry *earlier*, thus making curl inadvertently switch back to insecure HTTP earlier than otherwise intended.
CVE-2024-9823	There exists a security vulnerability in Jetty's DosFilter which can be exploited by unauthorized users to cause remote denial-of-service (DoS) attack on the server using DosFilter. By repeatedly sending crafted requests, attackers can trigger OutOfMemory errors and exhaust the server's memory finally.
CVE-2025-0167	When asked to use a `.netrc` file for credentials **and** to follow HTTP redirects, curl could leak the password used for the first host to the followed-to host under certain circumstances. This flaw only manifests itself if the netrc file has a `default` entry that omits both login and password. A rare circumstance.
CVE-2025-0395	When the assert() function in the GNU C Library versions 2.13 to 2.40 fails, it does not allocate enough space for the assertion failure message string and size information, which may lead to a buffer overflow if the message string size aligns to page size.
CVE-2025-0426	A security issue was discovered in Kubernetes where a large number of container checkpoint requests made to the unauthenticated kubelet read-only HTTP endpoint may cause a Node Denial of Service by filling the Node's disk.

CVE ID	Description
CVE-2025-0518	Unchecked Return Value, Out-of-bounds Read vulnerability in FFmpeg allows Read Sensitive Constants Within an Executable. This vulnerability is associated with program files https://github.com/FFmpeg/FFmpeg/blob/master/libavfilter/af_pan.c . This issue affects FFmpeg: 7.1. Issue was fixed: https://github.com/FFmpeg/FFmpeg/commit/b5b6391d64807578ab872dc58fb8aa621dcfc38a https://github.com/FFmpeg/FFmpeg/commit/b5b6391d64807578ab872dc58fb8aa621dcfc38a This issue was discovered by: Simcha Kosman
CVE-2025-0665	libcurl would wrongly close the same eventfd file descriptor twice when taking down a connection channel after having completed a threaded name resolve.
CVE-2025-0725	When libcurl is asked to perform automatic gzip decompression of content-encoded HTTP responses with the `CURLOPT_ACCEPT_ENCODING` option, **using zlib 1.2.0.3 or older**, an attacker-controlled integer overflow would make libcurl perform a buffer overflow.
CVE-2025-0913	os.OpenFile(path, os.O_CREATE O_EXCL) behaved differently on Unix and Windows systems when the target path was a dangling symlink. On Unix systems, OpenFile with O_CREATE and O_EXCL flags never follows symlinks. On Windows, when the target path was a symlink to a nonexistent location, OpenFile would create a file in that location. OpenFile now always returns an error when the O_CREATE and O_EXCL flags are both set and the target path is a symlink.
CVE-2025-0938	The Python standard library functions `urllib.parse.urlsplit` and `urlparse` accepted domain names that included square brackets which isn't valid according to RFC 3986. Square brackets are only meant to be used as delimiters for specifying IPv6 and IPvFuture hosts in URLs. This could result in differential parsing across the Python URL parser and other specification-compliant URL parsers.
CVE-2025-10148	curl's websocket code did not update the 32 bit mask pattern for each new outgoing frame as the specification says. Instead it used a fixed mask that persisted and was used throughout the entire connection. A predictable mask pattern allows for a malicious server to induce traffic between the two communicating parties that could be interpreted by an involved proxy (configured or transparent) as genuine, real, HTTP traffic with content and thereby poison its cache. That cached poisoned content could then be served to all users of that proxy.
CVE-2025-10256	A NULL pointer dereference vulnerability exists in FFmpeg's Firequalizer filter (libavfilter/af_firequalizer.c) due to a missing check on the return value of av_malloc_array() in the config_input() function. An attacker could exploit this by tricking a victim into processing a crafted media file with the Firequalizer filter enabled, causing the application to dereference a NULL pointer and crash, leading to denial of service.
CVE-2025-10911	A use-after-free vulnerability was found in libxslt while parsing xsl nodes that may lead to the dereference of expired pointers and application crash.
CVE-2025-1094	Improper neutralization of quoting syntax in PostgreSQL libpq functions PQescapeLiteral(), PQescapeIdentifier(), PQescapeString(), and PQescapeStringConn() allows a database input provider to achieve SQL injection in certain usage patterns. Specifically, SQL injection requires the application to use the function result to construct input to psql, the PostgreSQL interactive terminal. Similarly, improper neutralization of quoting syntax in PostgreSQL command line utility programs allows a source of command line arguments to achieve SQL injection when client_encoding is BIG5 and server_encoding is one of EUC_TW or MULE_INTERNAL. Versions before PostgreSQL 17.3, 16.7, 15.11, 14.16, and 13.19 are affected.
CVE-2025-1097	A security issue was discovered in ingress-nginx https://github.com/kubernetes/ingress-nginx where the `auth-tls-match-cn` Ingress annotation can be used to inject configuration into nginx. This can lead to arbitrary code execution in the context of the ingress-nginx controller, and disclosure of Secrets accessible to the controller. (Note that in the default installation, the controller can access all Secrets cluster-wide.)
CVE-2025-1098	A security issue was discovered in ingress-nginx https://github.com/kubernetes/ingress-nginx where the `mirror-target` and `mirror-host` Ingress annotations can be used to inject arbitrary configuration into nginx. This can lead to arbitrary code execution in the context of the ingress-nginx controller, and disclosure of Secrets accessible to the controller. (Note that in the default installation, the controller can access all Secrets cluster-wide.)
CVE-2025-11081	A vulnerability was detected in GNU Binutils 2.45. This issue affects the function dump_dwarf_section of the file binutils/objdump.c. Performing manipulation results in out-of-bounds read. The attack is only possible with local access. The exploit is now public and may be used. The patch is named f87a66db645caf8cc0e6fc87b0c28c78a38af59b. It is suggested to install a patch to address this issue.

CVE ID	Description
CVE-2025-11082	A flaw has been found in GNU Binutils 2.45. Impacted is the function <code>_bfd_elf_parse_eh_frame</code> of the file <code>bfd/elf-eh-frame.c</code> of the component Linker. Executing manipulation can lead to heap-based buffer overflow. The attack is restricted to local execution. The exploit has been published and may be used. This patch is called <code>ea1a0737c7692737a644af0486b71e4a392cbca8</code> . A patch should be applied to remediate this issue. The code maintainer replied with "[f]ixed for 2.46".
CVE-2025-11083	A vulnerability has been found in GNU Binutils 2.45. The affected element is the function <code>elf_swap_shdr</code> in the library <code>bfd/elfcode.h</code> of the component Linker. The manipulation leads to heap-based buffer overflow. The attack must be carried out locally. The exploit has been disclosed to the public and may be used. The identifier of the patch is <code>9ca499644a21ceb3f946d1c179c38a83be084490</code> . To fix this issue, it is recommended to deploy a patch. The code maintainer replied with "[f]ixed for 2.46".
CVE-2025-11226	ACE vulnerability in conditional configuration file processing by QOS.CH logback-core up to and including version 1.5.18 in Java applications, allows an attacker to execute arbitrary code by compromising an existing logback configuration file or by injecting an environment variable before program execution. A successful attack requires the presence of Janino library and Spring Framework to be present on the user's class path. In addition, the attacker must have write access to a configuration file. Alternatively, the attacker could inject a malicious environment variable pointing to a malicious configuration file. In both cases, the attack requires existing privilege.
CVE-2025-11412	A vulnerability has been found in GNU Binutils 2.45. This impacts the function <code>bfd_elf_gc_record_vtentry</code> of the file <code>bfd/elflink.c</code> of the component Linker. The manipulation leads to out-of-bounds read. Local access is required to approach this attack. The exploit has been disclosed to the public and may be used. The identifier of the patch is <code>047435dd988a3975d40c6626a8f739a0b2e154bc</code> . To fix this issue, it is recommended to deploy a patch.
CVE-2025-11413	A vulnerability was found in GNU Binutils 2.45. Affected is the function <code>elf_link_add_object_symbols</code> of the file <code>bfd/elflink.c</code> of the component Linker. The manipulation results in out-of-bounds read. The attack needs to be approached locally. The exploit has been made public and could be used. Upgrading to version 2.46 is able to address this issue. The patch is identified as <code>72efdf166aa0ed72ecc69fc2349af6591a7a19c0</code> . Upgrading the affected component is advised.
CVE-2025-11414	A vulnerability was determined in GNU Binutils 2.45. Affected by this vulnerability is the function <code>get_link_hash_entry</code> of the file <code>bfd/elflink.c</code> of the component Linker. This manipulation causes out-of-bounds read. The attack can only be executed locally. The exploit has been publicly disclosed and may be utilized. Upgrading to version 2.46 addresses this issue. Patch name: <code>a6aaa9af6359c8e394ce9cf24911fec4f4d23703</code> . It is advisable to upgrade the affected component.
CVE-2025-1147	A vulnerability has been found in GNU Binutils 2.43 and classified as problematic. Affected by this vulnerability is the function <code>__sanitizer::internal_strlen</code> of the file <code>binutils/nm.c</code> of the component nm. The manipulation of the argument <code>const</code> leads to buffer overflow. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used.
CVE-2025-1148	A vulnerability was found in GNU Binutils 2.43 and classified as problematic. Affected by this issue is the function <code>link_order_scan</code> of the file <code>ld/ldelfgen.c</code> of the component ld. The manipulation leads to memory leak. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise ld. All of the reported leaks in this bugzilla have been fixed on binutils master."
CVE-2025-11494	A vulnerability was found in GNU Binutils 2.45. Impacted is the function <code>_bfd_x86_elf_late_size_sections</code> of the file <code>bfd/elfxx-x86.c</code> of the component Linker. The manipulation results in out-of-bounds read. The attack needs to be approached locally. The exploit has been made public and could be used. The patch is identified as <code>b6ac5a8a5b82f0ae6a4642c8d7149b325f4cc60a</code> . A patch should be applied to remediate this issue.
CVE-2025-11495	A vulnerability was determined in GNU Binutils 2.45. The affected element is the function <code>elf_x86_64_relocate_section</code> of the file <code>elf64-x86-64.c</code> of the component Linker. This manipulation causes heap-based buffer overflow. The attack can only be executed locally. The exploit has been publicly disclosed and may be utilized. Patch name: <code>6b21c8b2ecfef5c95142cbc2c32f185cb1c26ab0</code> . To fix this issue, it is recommended to deploy a patch.

CVE ID	Description
CVE-2025-1150	A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. This vulnerability affects the function <code>bfd_malloc</code> of the file <code>libbfd.c</code> of the component <code>ld</code> . The manipulation leads to memory leak. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise <code>ld</code> . All of the reported leaks in this bugzilla have been fixed on binutils master."
CVE-2025-1151	A vulnerability was found in GNU Binutils 2.43. It has been rated as problematic. This issue affects the function <code>xmemdup</code> of the file <code>xmemdup.c</code> of the component <code>ld</code> . The manipulation leads to memory leak. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise <code>ld</code> . All of the reported leaks in this bugzilla have been fixed on binutils master."
CVE-2025-1152	A vulnerability classified as problematic has been found in GNU Binutils 2.43. Affected is the function <code>xstrdup</code> of the file <code>xstrdup.c</code> of the component <code>ld</code> . The manipulation leads to memory leak. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue. The code maintainer explains: "I'm not going to commit some of the leak fixes I've been working on to the 2.44 branch due to concern that would destabilise <code>ld</code> . All of the reported leaks in this bugzilla have been fixed on binutils master."
CVE-2025-1153	A vulnerability classified as problematic was found in GNU Binutils 2.43/2.44. Affected by this vulnerability is the function <code>bfd_set_format</code> of the file <code>format.c</code> . The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. Upgrading to version 2.45 is able to address this issue. The identifier of the patch is <code>8d97c1a53f3dc9fd8e1ccdb039b8a33d50133150</code> . It is recommended to upgrade the affected component.
CVE-2025-11731	A flaw was found in the <code>exsltFuncResultComp()</code> function of <code>libxslt</code> , which handles EXSLT <code><func:result></code> elements during stylesheet parsing. Due to improper type handling, the function may treat an XML document node as a regular XML element node, resulting in a type confusion. This can cause unexpected memory reads and potential crashes. While difficult to exploit, the flaw could lead to application instability or denial of service.
CVE-2025-1176	A vulnerability was found in GNU Binutils 2.43 and classified as critical. This issue affects the function <code>_bfd_elf_gc_mark_rsec</code> of the file <code>elflink.c</code> of the component <code>ld</code> . The manipulation leads to heap-based buffer overflow. The attack may be initiated remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. The patch is named <code>f9978defb6fab0bd8583942d97c112b0932ac814</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1178	A vulnerability was found in GNU Binutils 2.43. It has been declared as problematic. Affected by this vulnerability is the function <code>bfd_putl64</code> of the file <code>libbfd.c</code> of the component <code>ld</code> . The manipulation leads to memory corruption. The attack can be launched remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The identifier of the patch is <code>75086e9de1707281172cc77f178e7949a4414ed0</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1181	A vulnerability classified as critical was found in GNU Binutils 2.43. This vulnerability affects the function <code>_bfd_elf_gc_mark_rsec</code> of the file <code>bfd/elflink.c</code> of the component <code>ld</code> . The manipulation leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is <code>931494c9a89558acb36a03a340c01726545eef24</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1182	A vulnerability, which was classified as critical, was found in GNU Binutils 2.43. Affected is the function <code>bfd_elf_reloc_symbol_deleted_p</code> of the file <code>bfd/elflink.c</code> of the component <code>ld</code> . The manipulation leads to memory corruption. It is possible to launch the attack remotely. The complexity of an attack is rather high. The exploitability is told to be difficult. The exploit has been disclosed to the public and may be used. The patch is identified as <code>b425859021d17adf62f06fb904797cf8642986ad</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-11839	A security flaw has been discovered in GNU Binutils 2.45. Impacted is the function <code>tg_tag_type</code> of the file <code>prdbg.c</code> . Performing manipulation results in unchecked return value. The attack needs to be approached locally. The exploit has been released to the public and may be exploited.

CVE ID	Description
CVE-2025-11840	A weakness has been identified in GNU Binutils 2.45. The affected element is the function <code>vfinfo</code> of the file <code>ldmisc.c</code> . Executing manipulation can lead to out-of-bounds read. The attack can only be executed locally. The exploit has been made available to the public and could be exploited. This patch is called 16357. It is best practice to apply a patch to resolve this issue.
CVE-2025-1194	A Regular Expression Denial of Service (ReDoS) vulnerability was identified in the <code>huggingface/transformers</code> library, specifically in the file <code>tokenization_gpt_neox_japanese.py</code> of the GPT-NeoX-Japanese model. The vulnerability occurs in the <code>SubWordJapaneseTokenizer</code> class, where regular expressions process specially crafted inputs. The issue stems from a regex exhibiting exponential complexity under certain conditions, leading to excessive backtracking. This can result in high CPU usage and potential application downtime, effectively creating a Denial of Service (DoS) scenario. The affected version is v4.48.1 (latest).
CVE-2025-1215	A vulnerability classified as problematic was found in <code>vim</code> up to 9.1.1096. This vulnerability affects unknown code of the file <code>src/main.c</code> . The manipulation of the argument <code>--log</code> leads to memory corruption. It is possible to launch the attack on the local host. Upgrading to version 9.1.1097 is able to address this issue. The patch is identified as <code>c5654b84480822817bb7b69ebc97c174c91185e9</code> . It is recommended to upgrade the affected component.
CVE-2025-1352	A vulnerability has been found in GNU <code>elfutils</code> 0.192 and classified as critical. This vulnerability affects the function <code>__libdw_thread_tail</code> in the library <code>libdw_alloc.c</code> of the component <code>eu-readelf</code> . The manipulation of the argument <code>w</code> leads to memory corruption. The attack can be initiated remotely. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is <code>2636426a091bd6c6f7f02e49ab20d4cdc6bfc753</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1365	A vulnerability, which was classified as critical, was found in GNU <code>elfutils</code> 0.192. This affects the function <code>process_symtab</code> of the file <code>readelf.c</code> of the component <code>eu-readelf</code> . The manipulation of the argument <code>D/a</code> leads to buffer overflow. Local access is required to approach this attack. The exploit has been disclosed to the public and may be used. The identifier of the patch is <code>5e5c0394d82c53e97750fe7b18023e6f84157b81</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1371	A vulnerability has been found in GNU <code>elfutils</code> 0.192 and classified as problematic. This vulnerability affects the function <code>handle_dynamic_symtab</code> of the file <code>readelf.c</code> of the component <code>eu-read</code> . The manipulation leads to null pointer dereference. Attacking locally is a requirement. The exploit has been disclosed to the public and may be used. The patch is identified as <code>b38e562a4c907e08171c76b8b2def8464d5a104a</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1372	A vulnerability was found in GNU <code>elfutils</code> 0.192. It has been declared as critical. Affected by this vulnerability is the function <code>dump_data_section/print_string_section</code> of the file <code>readelf.c</code> of the component <code>eu-readelf</code> . The manipulation of the argument <code>z/x</code> leads to buffer overflow. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. The identifier of the patch is <code>73db9d2021cab9e23fd734b0a76a612d52a6f1db</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1376	A vulnerability classified as problematic was found in GNU <code>elfutils</code> 0.192. This vulnerability affects the function <code>elf_strptr</code> in the library <code>/libelf/elf_strptr.c</code> of the component <code>eu-strip</code> . The manipulation leads to denial of service. It is possible to launch the attack on the local host. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is <code>b16f441cca0a4841050e3215a9f120a6d8aea918</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1377	A vulnerability, which was classified as problematic, has been found in GNU <code>elfutils</code> 0.192. This issue affects the function <code>gelf_getsymshndx</code> of the file <code>strip.c</code> of the component <code>eu-strip</code> . The manipulation leads to denial of service. The attack needs to be approached locally. The exploit has been disclosed to the public and may be used. The identifier of the patch is <code>fbf1df9ca286de3323ae541973b08449f8d03aba</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-1594	A vulnerability, which was classified as critical, was found in <code>FFmpeg</code> up to 7.1. This affects the function <code>ff_aac_search_for_tns</code> of the file <code>libavcodec/aacenc_tns.c</code> of the component <code>AAC Encoder</code> . The manipulation leads to stack-based buffer overflow. It is possible to initiate the attack remotely. The exploit has been disclosed to the public and may be used.
CVE-2025-1632	A vulnerability was found in <code>libarchive</code> up to 3.7.7. It has been classified as problematic. This affects the function <code>list</code> of the file <code>bsdunzip.c</code> . The manipulation leads to null pointer dereference. It is possible to launch the attack on the local host. The exploit has been disclosed to the public and may be used. The vendor was contacted early about this disclosure but did not respond in any way.

CVE ID	Description
CVE-2025-1767	This CVE only affects Kubernetes clusters that utilize the in-tree gitRepo volume to clone git repositories from other pods within the same node. Since the in-tree gitRepo volume feature has been deprecated and will not receive security updates upstream, any cluster still using this feature remains vulnerable.
CVE-2025-1795	During an address list folding when a separating comma ends up on a folded line and that line is to be unicode-encoded then the separator itself is also unicode-encoded. Expected behavior is that the separating comma remains a plain comma. This can result in the address header being misinterpreted by some mail servers.
CVE-2025-1974	A security issue was discovered in Kubernetes where under certain conditions, an unauthenticated attacker with access to the pod network can achieve arbitrary code execution in the context of the ingress-nginx controller. This can lead to disclosure of Secrets accessible to the controller. (Note that in the default installation, the controller can access all Secrets cluster-wide.)
CVE-2025-1979	Versions of the package ray before 2.43.0 are vulnerable to Insertion of Sensitive Information into Log File where the redis password is being logged in the standard logging. If the redis password is passed as an argument, it will be logged and could potentially leak the password. This is only exploitable if: 1) Logging is enabled; 2) Redis is using password authentication; 3) Those logs are accessible to an attacker, who can reach that redis instance. Note: It is recommended that anyone who is running in this configuration should update to the latest version of Ray, then rotate their redis password.
CVE-2025-21502	Vulnerability in the Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition product of Oracle Java SE (component: Hotspot). Supported versions that are affected are Oracle Java SE: 8u431-perf, 11.0.25, 17.0.13, 21.0.5, 23.0.1; Oracle GraalVM for JDK: 17.0.13, 21.0.5, 23.0.1; Oracle GraalVM Enterprise Edition: 20.3.16 and 21.3.12. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition. Successful attacks of this vulnerability can result in unauthorized update, insert or delete access to some of Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition accessible data as well as unauthorized read access to a subset of Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition accessible data. Note: This vulnerability can be exploited by using APIs in the specified Component, e.g., through a web service which supplies data to the APIs. This vulnerability also applies to Java deployments, typically in clients running sandboxed Java Web Start applications or sandboxed Java applets, that load and run untrusted code (e.g., code that comes from the internet) and rely on the Java sandbox for security. CVSS 3.1 Base Score 4.8 (Confidentiality and Integrity impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N).
CVE-2025-21613	go-git is a highly extensible git implementation library written in pure Go. An argument injection vulnerability was discovered in go-git versions prior to v5.13. Successful exploitation of this vulnerability could allow an attacker to set arbitrary values to git-upload-pack flags. This only happens when the file transport protocol is being used, as that is the only protocol that shells out to git binaries. This vulnerability is fixed in 5.13.0.
CVE-2025-21614	go-git is a highly extensible git implementation library written in pure Go. A denial of service (DoS) vulnerability was discovered in go-git versions prior to v5.13. This vulnerability allows an attacker to perform denial of service attacks by providing specially crafted responses from a Git server which triggers resource exhaustion in go-git clients. Users running versions of go-git from v4 and above are recommended to upgrade to v5.13 in order to mitigate this vulnerability.

CVE ID	Description
CVE-2025-21631	In the Linux kernel, the following vulnerability has been resolved: block, bfq: fix waker_bfq UAF after bfq_split_bfq() Our syzkaller report a following UAF for v6.6: BUG: KASAN: slab-use-after-free in bfq_init_rq+0x175d/0x17a0 block/bfq-iosched.c:6958 Read of size 8 at addr ffff8881b57147d8 by task fsstress/232726 CPU: 2 PID: 232726 Comm: fsstress Not tainted 6.6.0-g3629d1885222 #39 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x91/0xf0 lib/dump_stack.c:106 print_address_description.constprop.0+0x66/0x300 mm/kasan/report.c:364 print_report+0x3e/0x70 mm/kasan/report.c:475 kasan_report+0xb8/0xf0 mm/kasan/report.c:588 hlist_add_head include/linux/list.h:1023 [inline] bfq_init_rq+0x175d/0x17a0 block/bfq-iosched.c:6958 bfq_insert_request.isra.0+0xe8/0xa20 block/bfq-iosched.c:6271 bfq_insert_requests+0x27f/0x390 block/bfq-iosched.c:6323 blk_mq_insert_request+0x290/0x8f0 block/blk-mq.c:2660 blk_mq_submit_bio+0x1021/0x15e0 block/blk-mq.c:3143 __submit_bio+0xa0/0x6b0 block/blk-core.c:639 __submit_bio_noacct_mq block/blk-core.c:718 [inline] submit_bio_noacct_noccheck+0x5b7/0x810 block/blk-core.c:747 submit_bio_noacct+0xca0/0x1990 block/blk-core.c:847 __ext4_read_bh fs/ext4/super.c:205 [inline] ext4_read_bh+0x15e/0x2e0 fs/ext4/super.c:230 __read_extent_tree_block+0x304/0x6f0 fs/ext4/extents.c:567 ext4_find_extent+0x479/0xd20 fs/ext4/extents.c:947 ext4_ext_map_blocks+0x1a3/0x2680 fs/ext4/extents.c:4182 ext4_map_blocks+0x929/0x15a0 fs/ext4/inode.c:660 ext4_iomap_begin_report+0x298/0x480 fs/ext4/inode.c:3569 iomap_iter+0x3dd/0x1010 fs/iomap/iter.c:91 iomap_fiemap+0x1f4/0x360 fs/iomap/fiemap.c:80 ext4_fiemap+0x181/0x210 fs/ext4/extents.c:5051 ioctl_fiemap.isra.0+0x1b4/0x290 fs/ioctl.c:220 do_vfs_ioctl+0x31c/0x11a0 fs/ioctl.c:811 __do_sys_ioctl fs/ioctl.c:869 [inline] __se_sys_ioctl+0xae/0x190 fs/ioctl.c:857 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x70/0x120 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x78/0xe2 Allocated by task 232719: kasan_save_stack+0x22/0x50 mm/kasan/common.c:45 kasan_set_track+0x25/0x30 mm/kasan/common.c:52 __kasan_slab_alloc+0x87/0x90 mm/kasan/common.c:328 kasan_slab_alloc include/linux/kasan.h:188 [inline] slab_post_alloc_hook mm/slab.h:768 [inline] slab_alloc_node mm/slub.c:3492 [inline] kmem_cache_alloc_node+0x1b8/0x6f0 mm/slub.c:3537 bfq_get_queue+0x215/0x1f00 block/bfq-iosched.c:5869 bfq_get_bfqq_handle_split+0x167/0x5f0 block/bfq-iosched.c:6776 bfq_init_rq+0x13a4/0x17a0 block/bfq-iosched.c:6938 bfq_insert_request.isra.0+0xe8/0xa20 block/bfq-iosched.c:6271 bfq_insert_requests+0x27f/0x390 block/bfq-iosched.c:6323 blk_mq_insert_request+0x290/0x8f0 block/blk-mq.c:2660 blk_mq_submit_bio+0x1021/0x15e0 block/blk-mq.c:3143 __submit_bio+0xa0/0x6b0 block/blk-core.c:639 __submit_bio_noacct_mq block/blk-core.c:718 [inline] submit_bio_noacct_noccheck+0x5b7/0x810 block/blk-core.c:747 submit_bio_noacct+0xca0/0x1990 block/blk-core.c:847 __ext4_read_bh fs/ext4/super.c:205 [inline] ext4_read_bh_nowait+0x15a/0x240 fs/ext4/super.c:217 ext4_read_bh_lock+0xac/0xd0 fs/ext4/super.c:242 ext4_bread_batch+0x268/0x500 fs/ext4/inode.c:958 __ext4_find_entry+0x448/0x10f0 fs/ext4/namei.c:1671 ext4_lookup_entry fs/ext4/namei.c:1774 [inline] ext4_lookup.part.0+0x359/0x6f0 fs/ext4/namei.c:1842 ext4_lookup+0x72/0x90 fs/ext4/namei.c:1839 __lookup_slow+0x257/0x480 fs/namei.c:1696 lookup_slow fs/namei.c:1713 [inline] walk_component+0x454/0x5c0 fs/namei.c:2004 link_path_walk.part.0+0x773/0xda0 fs/namei.c:2331 link_path_walk fs/namei.c:3826 [inline] path_openat+0x1b9/0x520 fs/namei.c:3826 do_filp_open+0x1b7/0x400 fs/namei.c:3857 do_sys_openat2+0x5dc/0x6e0 fs/open.c:1428 do_sys_open fs/open.c:1443 [inline] __do_sys_openat fs/open.c:1459 [inline] __se_sys_openat fs/open.c:1454 [inline] __x64_sys_openat+0x148/0x200 fs/open.c:1454 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64 ---truncated---
CVE-2025-21632	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Ensure shadow stack is active before "getting" registers The x86 shadow stack support has its own set of registers. Those registers are XSAVE-managed, but they are "supervisor state components" which means that userspace can not touch them with XSAVE/XRSTOR. It also means that they are not accessible from the existing ptrace ABI for XSAVE state. Thus, there is a new ptrace get/set interface for it. The regset code that ptrace uses provides an ->active() handler in addition to the get/set ones. For shadow stack this ->active() handler verifies that shadow stack is enabled via the ARCH_SHSTK_SHSTK bit in the thread struct. The ->active() handler is checked from some call sites of the regset get/set handlers, but not the ptrace ones. This was not understood when shadow stack support was put in place. As a result, both the set/get handlers can be called with XFEATURE_CET_USER in its init state, which would cause get_xsave_addr() to return NULL and trigger a WARN_ON(). The ssp_init() handler luckily has an ssp_active() check to avoid surprising the kernel with shadow stack behavior when the kernel is not ready for it (ARCH_SHSTK_SHSTK==0). That check just happened to avoid the warning. But the ->get() side wasn't so lucky. It can be called with shadow stacks disabled, triggering the warning in practice, as reported by Christina Schimpe: WARNING: CPU: 5 PID: 1773 at arch/x86/kernel/fpu/regset.c:198 ssp_get+0x89/0xa0 [...] Call Trace: <TASK> ? show_regs+0x6e/0x80 ? ssp_get+0x89/0xa0 ? __warn+0x91/0x150 ? ssp_get+0x89/0xa0 ? report_bug+0x19d/0x1b0 ? handle_bug+0x46/0x80 ? exc_invalid_op+0x1d/0x80 ? asm_exc_invalid_op+0x1f/0x30 ? __pfx_ssp_get+0x10/0x10 ? ssp_get+0x89/0xa0 ? ssp_get+0x52/0xa0 __regset_get+0xad/0xf0 copy_regset_to_user+0x52/0xc0 ptrace_regset+0x119/0x140 ptrace_request+0x13c/0x850 ? wait_task_inactive+0x142/0x1d0 ? do_syscall_64+0x6d/0x90 arch_ptrace+0x102/0x300 [...] Ensure that shadow stacks are active in a thread before looking them up in the XSAVE buffer. Since ARCH_SHSTK_SHSTK and user_ssp[SHSTK_EN] are set at the same time, the active check ensures that there will be something to find in the XSAVE buffer. [dhansen: changelog/subject tweaks]

CVE ID	Description
CVE-2025-21634	In the Linux kernel, the following vulnerability has been resolved: cgroup/cpuset: remove kernfs active break A warning was found: WARNING: CPU: 10 PID: 3486953 at fs/kernfs/file.c:828 CPU: 10 PID: 3486953 Comm: rmdir Kdump: loaded Tainted: G RIP: 0010:kernfs_should_drain_open_files +0x1a1/0x1b0 RSP: 0018:ffff8881107ef9e0 EFLAGS: 00010202 RAX: 0000000080000002 RBX: ffff888154738c00 RCX: dffffc0000000000 RDX: 0000000000000007 RSI: 0000000000000004 RDI: ffff888154738c04 RBP: ffff888154738c04 R08: ffffffffaf27fa15 R09: ffffd102a8e7180 R10: ffff888154738c07 R11: 0000000000000000 R12: ffff888154738c08 R13: ffff888750f8c000 R14: ffff888750f8c0e8 R15: ffff888154738ca0 FS: 00007f84cd0be740(0000) GS:ffff8887ddc00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000555f9fbe00c8 CR3: 0000000153eec001 CR4: 0000000000370ee0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: kernfs_drain+0x15e/0x2f0 __kernfs_remove+0x165/0x300 kernfs_remove_by_name_ns+0x7b/0xc0 cgroup_rm_file+0x154/0x1c0 cgroup_addrm_files +0x1c2/0x1f0 css_clear_dir+0x77/0x110 kill_css+0x4c/0x1b0 cgroup_destroy_locked+0x194/0x380 cgroup_rmdir+0x2a/0x140 It can be explained by: rmdir echo 1 > cpuset.cpus kernfs_fop_write_iter // active=0 cgroup_rm_file kernfs_remove_by_name_ns kernfs_get_active // active=1 __kernfs_remove // active=0x80000002 kernfs_drain cpuset_write_resmask wait_event //waiting (active == 0x80000001) kernfs_break_active_protection // active = 0x80000001 // continue kernfs_unbreak_active_protection // active = 0x80000002 ... kernfs_should_drain_open_files // warning occurs kernfs_put_active This warning is caused by 'kernfs_break_active_protection' when it is writing to cpuset.cpus, and the cgroup is removed concurrently. The commit 3a5a6d0c2b03 ("cpuset: don't nest cgroup_mutex inside get_online_cpus()") made cpuset_hotplug_workfn asynchronous, This change involves calling flush_work(), which can create a multiple processes circular locking dependency that involve cgroup_mutex, potentially leading to a deadlock. To avoid deadlock, the commit 76bb5ab8f6e3 ("cpuset: break kernfs active protection in cpuset_write_resmask()") added 'kernfs_break_active_protection' in the cpuset_write_resmask. This could lead to this warning. After the commit 2125c0034c5d ("cgroup/cpuset: Make cpuset hotplug processing synchronous"), the cpuset_write_resmask no longer needs to wait the hotplug to finish, which means that concurrent hotplug and cpuset operations are no longer possible. Therefore, the deadlock doesn't exist anymore and it does not have to 'break active protection' now. To fix this warning, just remove kernfs_break_active_protection operation in the 'cpuset_write_resmask'.
CVE-2025-21635	In the Linux kernel, the following vulnerability has been resolved: rds: sysctl: rds_tcp_{rcv,snd}buf: avoid using current->nsproxy As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons: - Inconsistency: getting info from the reader's/ writer's netns vs only from the opener's netns. - current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2). The per-netns structure can be obtained from the table->data using container_of(), then the 'net' one can be retrieved from the listen socket (if available).
CVE-2025-21636	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: plpmtud_probe_interval: avoid using current->nsproxy As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons: - Inconsistency: getting info from the reader's/ writer's netns vs only from the opener's netns. - current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2). The 'net' structure can be obtained from the table->data using container_of(). Note that table->data could also be used directly, as this is the only member needed from the 'net' structure, but that would increase the size of this fix, to use '*data' everywhere 'net->sctp.probe_interval' is used.
CVE-2025-21637	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: udp_port: avoid using current->nsproxy As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons: - Inconsistency: getting info from the reader's/writer's netns vs only from the opener's netns. - current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2). The 'net' structure can be obtained from the table->data using container_of(). Note that table->data could also be used directly, but that would increase the size of this fix, while 'sctp.ctl_sock' still needs to be retrieved from 'net' structure.
CVE-2025-21638	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: auth_enable: avoid using current->nsproxy As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons: - Inconsistency: getting info from the reader's/writer's netns vs only from the opener's netns. - current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2). The 'net' structure can be obtained from the table->data using container_of(). Note that table->data could also be used directly, but that would increase the size of this fix, while 'sctp.ctl_sock' still needs to be retrieved from 'net' structure.

CVE ID	Description
CVE-2025-21639	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: rto_min/max: avoid using current->nsproxy As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons: - Inconsistency: getting info from the reader's/writer's netns vs only from the opener's netns. - current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2). The 'net' structure can be obtained from the table->data using container_of(). Note that table->data could also be used directly, as this is the only member needed from the 'net' structure, but that would increase the size of this fix, to use '*data' everywhere 'net->sctp.rto_min/max' is used.
CVE-2025-21640	In the Linux kernel, the following vulnerability has been resolved: sctp: sysctl: cookie_hmac_alg: avoid using current->nsproxy As mentioned in a previous commit of this series, using the 'net' structure via 'current' is not recommended for different reasons: - Inconsistency: getting info from the reader's/writer's netns vs only from the opener's netns. - current->nsproxy can be NULL in some cases, resulting in an 'Oops' (null-ptr-deref), e.g. when the current task is exiting, as spotted by syzbot [1] using acct(2). The 'net' structure can be obtained from the table->data using container_of(). Note that table->data could also be used directly, as this is the only member needed from the 'net' structure, but that would increase the size of this fix, to use '*data' everywhere 'net->sctp.cookie_hmac_alg' is used.
CVE-2025-21642	In the Linux kernel, the following vulnerability has been resolved: mptcp: sysctl: sched: avoid using current->nsproxy Using the 'net' structure via 'current' is not recommended for different reasons. First, if the goal is to use it to read or write per-netns data, this is inconsistent with how the "generic" sysctl entries are doing: directly by only using pointers set to the table entry, e.g. table->data. Linked to that, the per-netns data should always be obtained from the table linked to the netns it had been created for, which may not coincide with the reader's or writer's netns. Another reason is that access to current->nsproxy->netns can oops if attempted when current->nsproxy had been dropped when the current task is exiting. This is what syzbot found, when using acct(2): Oops: general protection fault, probably for non-canonical address 0xdffffc0000000005: 0000 [#1] PREEMPT SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000028-0x000000000000002f] CPU: 1 UID: 0 PID: 5924 Comm: syz-executor Not tainted 6.13.0-rc5-syzkaller-00004-gccb98ccef0e5 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010:proc_scheduler+0xc6/0x3c0 net/mptcp/ctrl.c:125 Code: 03 42 80 3c 38 00 0f 85 fe 02 00 00 4d 8b a4 24 08 09 00 00 48 b8 00 00 00 00 00 fc ff df 49 8d 7c 24 28 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 cc 02 00 00 4d 8b 7c 24 28 48 8d 84 24 c8 00 00 RSP: 0018:ffff900034774e8 EFLAGS: 00010206 RAX: dffffc0000000000 RBX: 1ffff920068ee9e RCX: fffff90003477620 RDX: 0000000000000005 RSI: ffffffffb08f91e RDI: 0000000000000028 RBP: 0000000000000001 R08: fffff90003477710 R09: 0000000000000040 R10: 0000000000000040 R11: 00000000726f7475 R12: 0000000000000000 R13: fffff90003477620 R14: fffff90003477710 R15: dffffc0000000000 FS: 0000000000000000(0000) GS:ffff880b87000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fee3cd452d8 CR3: 000000007d116000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> proc_sys_call_handler+0x403/0x5d0 fs/proc/proc_sysctl.c:601 __kernel_write_iter+0x318/0xa80 fs/read_write.c:612 __kernel_write+0xf6/0x140 fs/read_write.c:632 do_acct_process+0xcb0/0x14a0 kernel/acct.c:539 acct_pin_kill+0x2d/0x100 kernel/acct.c:192 pin_kill+0x194/0x7c0 fs/fs_pin.c:44 mnt_pin_kill+0x61/0x1e0 fs/fs_pin.c:81 cleanup_mnt+0x3ac/0x450 fs/namespace.c:1366 task_work_run+0x14e/0x250 kernel/task_work.c:239 exit_task_work include/linux/task_work.h:43 [inline] do_exit+0xad8/0x2d70 kernel/exit.c:938 do_group_exit+0xd3/0x2a0 kernel/exit.c:1087 get_signal+0x2576/0x2610 kernel/signal.c:3017 arch_do_signal_or_restart+0x90/0x7e0 arch/x86/kernel/signal.c:337 exit_to_user_mode_loop kernel/entry/common.c:111 [inline] exit_to_user_mode_prepare include/linux/entry-common.h:329 [inline] __syscall_exit_to_user_mode_work kernel/entry/common.c:207 [inline] syscall_exit_to_user_mode+0x150/0x2a0 kernel/entry/common.c:218 do_syscall_64+0xda/0x250 arch/x86/entry/common.c:89 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fee3cb87a6a Code: Unable to access opcode bytes at 0x7fee3cb87a40. RSP: 002b:00007ffccac688 EFLAGS: 00000202 ORIG_RAX: 0000000000000037 RAX: 0000000000000000 RBX: 00007ffccac710 RCX: 00007fee3cb87a6a RDX: 0000000000000041 RSI: 0000000000000000 RDI: 0000000000000003 RBP: 0000000000000003 R08: 00007ffccac6ac R09: 00007ffccacac7 R10: 00007ffccac710 R11: 0000000000000202 R12: 00007fee3cd49500 R13: 00007ffccac6ac R14: 0000000000000000 R15: 00007fee3cd4b000 </TASK> Modules linked in: --- [end trace 0000000000000000] --- RIP: 0010:proc_scheduler+0xc6/0x3c0 net/mptcp/ctrl.c:125 Code: 03 42 80 3c 38 00 0f 85 fe 02 00 00 4d 8b a4 24 08 09 00 00 48 b8 00 00 00 00 00 fc ---truncated---

CVE ID	Description
CVE-2025-21643	In the Linux kernel, the following vulnerability has been resolved: netfs: Fix kernel async DIO Netfslib needs to be able to handle kernel-initiated asynchronous DIO that is supplied with a bio_vec[] array. Currently, because of the async flag, this gets passed to netfs_extract_user_iter() which throws a warning and fails because it only handles IOVEC and UBUF iterators. This can be triggered through a combination of cifs and a loopback blockdev with something like: mount //my/cifs/share /foo dd if=/dev/zero of=/foo/m0 bs=4K count=1K losetup --sector-size 4096 --direct-io=on /dev/loop2046 /foo/m0 echo hello >/dev/loop2046 This causes the following to appear in syslog: WARNING: CPU: 2 PID: 109 at fs/netfs/iterator.c:50 netfs_extract_user_iter+0x170/0x250 [netfs] and the write to fail. Fix this by removing the check in netfs_unbuffered_write_iter_locked() that causes async kernel DIO writes to be handled as userspace writes. Note that this change relies on the kernel caller maintaining the existence of the bio_vec array (or kvec[] or folio_queue) until the op is complete.
CVE-2025-21645	In the Linux kernel, the following vulnerability has been resolved: platform/x86/amd/pmc: Only disable IRQ1 wakeup where i8042 actually enabled it Wakeup for IRQ1 should be disabled only in cases where i8042 had actually enabled it, otherwise "wake_depth" for this IRQ will try to drop below zero and there will be an unpleasant WARN() logged: kernel: atkbd serio0: Disabling IRQ1 wakeup source to avoid platform firmware bug kernel: -----[cut here]----- kernel: Unbalanced IRQ 1 wake disable kernel: WARNING: CPU: 10 PID: 6431 at kernel/irq/manage.c:920 irq_set_irq_wake+0x147/0x1a0 The PMC driver uses DEFINE_SIMPLE_DEV_PM_OPS() to define its dev_pm_ops which sets amd_pmc_suspend_handler() to the .suspend, .freeze, and .poweroff handlers. i8042_pm_suspend(), however, is only set as the .suspend handler. Fix the issue by call PMC suspend handler only from the same set of dev_pm_ops handlers as i8042_pm_suspend(), which currently means just the .suspend handler. To reproduce this issue try hibernating (S4) the machine after a fresh boot without putting it into s2idle first. [ij: edited the commit message.]
CVE-2025-21646	In the Linux kernel, the following vulnerability has been resolved: afs: Fix the maximum cell name length The kafs filesystem limits the maximum length of a cell to 256 bytes, but a problem occurs if someone actually does that: kafs tries to create a directory under /proc/net/afs/ with the name of the cell, but that fails with a warning: WARNING: CPU: 0 PID: 9 at fs/proc/generic.c:405 because procfs limits the maximum filename length to 255. However, the DNS limits the maximum lookup length and, by extension, the maximum cell name, to 255 less two (length count and trailing NUL). Fix this by limiting the maximum acceptable cellname length to 253. This also allows us to be sure we can create the "/afs/.<cell>/" mountpoint too. Further, split the YFS VL record cell name maximum to be the 256 allowed by the protocol and ignore the record retrieved by YFSVL.GetCellName if it exceeds 253.
CVE-2025-21647	In the Linux kernel, the following vulnerability has been resolved: sched: sch_cake: add bounds checks to host bulk flow fairness counts Even though we fixed a logic error in the commit cited below, syzbot still managed to trigger an underflow of the per-host bulk flow counters, leading to an out of bounds memory access. To avoid any such logic errors causing out of bounds memory accesses, this commit factors out all accesses to the per-host bulk flow counters to a series of helpers that perform bounds-checking before any increments and decrements. This also has the benefit of improving readability by moving the conditional checks for the flow mode into these helpers, instead of having them spread out throughout the code (which was the cause of the original logic error). As part of this change, the flow quantum calculation is consolidated into a helper function, which means that the dithering applied to the ost load scaling is now applied both in the DRR rotation and when a sparse flow's quantum is first initiated. The only user-visible effect of this is that the maximum packet size that can be sent while a flow stays sparse will now vary with +/- one byte in some cases. This should not make a noticeable difference in practice, and thus it's not worth complicating the code to preserve the old behaviour.
CVE-2025-21648	In the Linux kernel, the following vulnerability has been resolved: netfilter: conntrack: clamp maximum hashtable size to INT_MAX Use INT_MAX as maximum size for the conntrack hashtable. Otherwise, it is possible to hit WARN_ON_ONCE in __kvmalloc_node_noprof() when resizing hashtable because __GFP_NOWARN is unset. See: 0708a0afe291 ("mm: Consider __GFP_NOWARN flag for oversized kvmalloc() calls") Note: hashtable resize is only possible from init_netns.

CVE ID	Description
CVE-2025-21649	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix kernel crash when 1588 is sent on HIP08 devices Currently, HIP08 devices does not register the ptp devices, so the hdev->ptp is NULL. But the tx process would still try to set hardware time stamp info with SKBTX_HW_TSTAMP flag and cause a kernel crash. [128.087798] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000018 ... [128.280251] pc : hclge_ptp_set_tx_info+0x2c/0x140 [hclge] [128.286600] lr : hclge_ptp_set_tx_info+0x20/0x140 [hclge] [128.292938] sp : ffff800059b93140 [128.297200] x29: ffff800059b93140 x28: 0000000000003280 [128.303455] x27: ffff800020d48280 x26: ffff0cb9dc814080 [128.309715] x25: ffff0cb9cde93fa0 x24: 0000000000000001 [128.315969] x23: 0000000000000000 x22: 0000000000000194 [128.322219] x21: ffff0cd94f986000 x20: 0000000000000000 [128.328462] x19: ffff0cb9d2a166c0 x18: 0000000000000000 [128.334698] x17: 0000000000000000 x16: ffff0cf1fc523ed24 [128.340934] x15: 0000ffffd530a518 x14: 0000000000000000 [128.347162] x13: ffff0cd6bdb31310 x12: 0000000000000368 [128.353388] x11: ffff0cb9cfbc7070 x10: ffff2cf55dd11e02 [128.359606] x9 : ffff0cf1f85a212b4 x8 : ffff0cd7cf27dab0 [128.365831] x7 : 00000000000000a20 x6 : ffff0cd7cf27d000 [128.372040] x5 : 0000000000000000 x4 : 000000000000ffff [128.378243] x3 : 0000000000000400 x2 : ffff0cf1f85a21294 [128.384437] x1 : ffff0cb9db520080 x0 : ffff0cb9db500080 [128.390626] Call trace: [128.393964] hclge_ptp_set_tx_info+0x2c/0x140 [hclge] [128.399893] hns3_nic_net_xmit+0x39c/0x4c4 [hns3] [128.405468] xmit_one.constprop.0+0xc4/0x200 [128.410600] dev_hard_start_xmit+0x54/0xf0 [128.415556] sch_direct_xmit+0xe8/0x634 [128.420246] __dev_queue_xmit+0x224/0xc70 [128.425101] dev_queue_xmit+0x1c/0x40 [128.429608] ovs_vport_send+0xac/0x1a0 [openvswitch] [128.435409] do_output+0x60/0x17c [openvswitch] [128.440770] do_execute_actions+0x898/0x8c4 [openvswitch] [128.446993] ovs_execute_actions+0x64/0xf0 [openvswitch] [128.453129] ovs_dp_process_packet+0xa0/0x224 [openvswitch] [128.459530] ovs_vport_receive+0x7c/0xfc [openvswitch] [128.465497] internal_dev_xmit+0x34/0xb0 [openvswitch] [128.471460] xmit_one.constprop.0+0xc4/0x200 [128.476561] dev_hard_start_xmit+0x54/0xf0 [128.481489] __dev_queue_xmit+0x968/0xc70 [128.486330] dev_queue_xmit+0x1c/0x40 [128.490856] ip_finish_output2+0x250/0x570 [128.495810] __ip_finish_output+0x170/0x1e0 [128.500832] ip_finish_output+0x3c/0xf0 [128.505504] ip_output+0xbc/0x160 [128.509654] ip_send_skb+0x58/0xd4 [128.513892] udp_send_skb+0x12c/0x354 [128.518387] udp_sendmsg+0x7a8/0x9c0 [128.522793] inet_sendmsg+0x4c/0x8c [128.527116] __sock_sendmsg+0x48/0x80 [128.531609] __sys_sendto+0x124/0x164 [128.536099] __arm64_sys_sendto+0x30/0x5c [128.540935] invoke_syscall+0x50/0x130 [128.545508] el0_svc_common.constprop.0+0x10c/0x124 [128.551205] do_el0_svc+0x34/0xdc [128.555347] el0_svc+0x20/0x30 [128.559227] el0_sync_handler+0xb8/0xc0 [128.563883] el0_sync+0x160/0x180
CVE-2025-21650	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fixed hclge_fetch_pf_reg accesses bar space out of bounds issue The TQP BAR space is divided into two segments. TQPs 0-1023 and TQPs 1024-1279 are in different BAR space addresses. However, hclge_fetch_pf_reg does not distinguish the tcp space information when reading the tcp space information. When the number of TQPs is greater than 1024, access bar space overwriting occurs. The problem of different segments has been considered during the initialization of tcp.io_base. Therefore, tcp.io_base is directly used when the queue is read in hclge_fetch_pf_reg. The error message: Unable to handle kernel paging request at virtual address ffff800037200000 pc : hclge_fetch_pf_reg+0x138/0x250 [hclge] lr : hclge_get_regs+0x84/0x1d0 [hclge] Call trace: hclge_fetch_pf_reg+0x138/0x250 [hclge] hclge_get_regs+0x84/0x1d0 [hclge] hns3_get_regs+0x2c/0x50 [hns3] ethtool_get_regs+0xf4/0x270 dev_ethtool+0x674/0x8a0 dev_ioctl+0x270/0x36c sock_do_ioctl+0x110/0x2a0 sock_ioctl+0x2ac/0x530 __arm64_sys_ioctl+0xa8/0x100 invoke_syscall+0x4c/0x124 el0_svc_common.constprop.0+0x140/0x15c do_el0_svc+0x30/0xd0 el0_svc+0x1c/0x2c el0_sync_handler+0xb0/0xb4 el0_sync+0x168/0x180
CVE-2025-21651	In the Linux kernel, the following vulnerability has been resolved: net: hns3: don't auto enable misc vector Currently, there is a time window between misc irq enabled and service task init. If an interrupt is reported at this time, it will cause warning like below: [16.324639] Call trace: [16.324641] __queue_delayed_work+0xb8/0xe0 [16.324643] mod_delayed_work_on+0x78/0xd0 [16.324655] hclge_errhand_task_schedule+0x58/0x90 [hclge] [16.324662] hclge_misc_irq_handle+0x168/0x240 [hclge] [16.324666] __handle_irq_event_percpu+0x64/0x1e0 [16.324667] handle_irq_event+0x80/0x170 [16.324670] handle_fasteoi_edge_irq+0x110/0x2bc [16.324671] __handle_domain_irq+0x84/0xfc [16.324673] gic_handle_irq+0x88/0x2c0 [16.324674] el1_irq+0xb8/0x140 [16.324677] arch_cpu_idle+0x18/0x40 [16.324679] default_idle_call+0x5c/0x1bc [16.324682] cpuidle_idle_call+0x18c/0x1c4 [16.324684] do_idle+0x174/0x17c [16.324685] cpu_startup_entry+0x30/0x6c [16.324687] secondary_start_kernel+0x1a4/0x280 [16.324688] ---[end trace 6aa0bff672a964aa]--- So don't auto enable misc vector when request irq..

CVE ID	Description
CVE-2025-21652	In the Linux kernel, the following vulnerability has been resolved: ipvlan: Fix use-after-free in ipvlan_get_iflink(). syzbot presented an use-after-free report [0] regarding ipvlan and linkwatch. ipvlan does not hold a refcnt of the lower device unlike vlan and macvlan. If the linkwatch work is triggered for the ipvlan dev, the lower dev might have already been freed, resulting in UAF of ipvlan->phy_dev in ipvlan_get_iflink(). We can delay the lower dev unregistration like vlan and macvlan by holding the lower dev's refcnt in dev->netdev_ops->ndo_init() and releasing it in dev->priv_destructor(). Jakub pointed out calling .ndo_XXX after unregister_netdevice() has returned is error prone and suggested [1] addressing this UAF in the core by taking commit 750e51603395 ("net: avoid potential UAF in default_operstate()") further. Let's assume unregistering devices DOWN and use RCU protection in default_operstate() not to race with the device unregistration. [0]: BUG: KASAN: slab-use-after-free in ipvlan_get_iflink+0x84/0x88 drivers/net/ipvlan/ipvlan_main.c:353 Read of size 4 at addr ffff0000d768c0e0 by task kworker/u8:35/6944 CPU: 0 UID: 0 PID: 6944 Comm: kworker/u8:35 Not tainted 6.13.0-rc2-g9bc5c9515b48 #12 4c3cb9e8b4565456f6a355f312ff91f4f29b3c47 Hardware name: linux,dummy-virt (DT) Workqueue: events_unbound linkwatch_event Call trace: show_stack+0x38/0x50 arch/arm64/kernel/stacktrace.c:484 (C) __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0xb0/0x108 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x16c/0x6f0 mm/kasan/report.c:489 kasan_report+0xc0/0x120 mm/kasan/report.c:602 __asan_report_load4_noabort+0x20/0x30 mm/kasan/report_generic.c:380 ipvlan_get_iflink+0x84/0x88 drivers/net/ipvlan/ipvlan_main.c:353 dev_get_iflink+0x7c/0xd8 net/core/dev.c:674 default_operstate net/core/link_watch.c:45 [inline] rfc2863_policy+0x144/0x360 net/core/link_watch.c:72 linkwatch_do_dev+0x60/0x228 net/core/link_watch.c:175 __linkwatch_run_queue+0x2f4/0x5b8 net/core/link_watch.c:239 linkwatch_event+0x64/0xa8 net/core/link_watch.c:282 process_one_work+0x700/0x1398 kernel/workqueue.c:3229 process_scheduled_works kernel/workqueue.c:3310 [inline] worker_thread+0x8c4/0xe10 kernel/workqueue.c:3391 kthread+0x2b0/0x360 kernel/kthread.c:389 ret_from_fork+0x10/0x20 arch/arm64/kernel/entry.S:862 Allocated by task 9303: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x30/0x68 mm/kasan/common.c:68 kasan_save_alloc_info+0x44/0x58 mm/kasan/generic.c:568 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x84/0xa0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __do_kmalloc_node mm/slab.c:4283 [inline] __kmalloc_node_noprof+0x2a0/0x560 mm/slab.c:4289 __kvmalloc_node_noprof+0x9c/0x230 mm/util.c:650 alloc_netdev_mqs+0xb4/0x1118 net/core/dev.c:11209 rtnl_create_link+0x2b8/0xb60 net/core/rtnetlink.c:3595 rtnl_newlink_create+0x19c/0x868 net/core/rtnetlink.c:3771 __rtnl_newlink net/core/rtnetlink.c:3896 [inline] rtnl_newlink+0x122c/0x15c0 net/core/rtnetlink.c:4011 rtnetlink_rcv_msg+0x61c/0x918 net/core/rtnetlink.c:6901 netlink_rcv_skb+0x1dc/0x398 net/netlink/af_netlink.c:2542 rtnetlink_rcv+0x34/0x50 net/core/rtnetlink.c:6928 netlink_unicast_kernel net/netlink/af_netlink.c:1321 [inline] netlink_unicast+0x618/0x838 net/netlink/af_netlink.c:1347 netlink_sendmsg+0x5fc/0x8b0 net/netlink/af_netlink.c:1891 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg net/socket.c:726 [inline] __sys_sendto+0x2ec/0x438 net/socket.c:2197 __do_sys_sendto net/socket.c:2204 [inline] __se_sys_sendto net/socket.c:2200 [inline] __arm64_sys_sendto+0xe4/0x110 net/socket.c:2200 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x90/0x278 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x13c/0x250 arch/arm64/kernel/syscall.c:132 do_el0_svc+0x54/0x70 arch/arm64/kernel/syscall.c:151 el ---truncated---
CVE-2025-21653	In the Linux kernel, the following vulnerability has been resolved: net_sched: cls_flow: validate TCA_FLOW_RSHIFT attribute syzbot found that TCA_FLOW_RSHIFT attribute was not validated. Right shifting a 32bit integer is undefined for large shift values. UBSAN: shift-out-of-bounds in net/sched/cls_flow.c:329:23 shift exponent 9445 is too large for 32-bit type 'u32' (aka 'unsigned int') CPU: 1 UID: 0 PID: 54 Comm: kworker/u8:3 Not tainted 6.13.0-rc3-syzkaller-00180-g4f619d518db9 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: ipv6_addrconf addrconf_dad_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 ubsan_epilogue lib/ubsan.c:231 [inline] __ubsan_handle_shift_out_of_bounds+0x3c8/0x420 lib/ubsan.c:468 flow_classify+0x24d5/0x25b0 net/sched/cls_flow.c:329 tc_classify include/net/tc_wrapper.h:197 [inline] __tcf_classify net/sched/cls_api.c:1771 [inline] tcf_classify+0x420/0x1160 net/sched/cls_api.c:1867 sfb_classify net/sched/sch_sfb.c:260 [inline] sfb_enqueue+0x3ad/0x18b0 net/sched/sch_sfb.c:318 dev_qdisc_enqueue+0x4b/0x290 net/core/dev.c:3793 __dev_xmit_skb net/core/dev.c:3889 [inline] __dev_queue_xmit+0xf0e/0x3f50 net/core/dev.c:4400 dev_queue_xmit include/linux/netdevice.h:3168 [inline] neigh_hh_output include/net/neighbour.h:523 [inline] neigh_output include/net/neighbour.h:537 [inline] ip_finish_output2+0xd41/0x1390 net/ipv4/ip_output.c:236 iptunnel_xmit+0x55d/0x9b0 net/ipv4/ip_tunnel_core.c:82 udp_tunnel_xmit_skb+0x262/0x3b0 net/ipv4/udp_tunnel_core.c:173 geneve_xmit_skb drivers/net/geneve.c:916 [inline] geneve_xmit+0x21dc/0x2d00 drivers/net/geneve.c:1039 __netdev_start_xmit include/linux/netdevice.h:5002 [inline] netdev_start_xmit include/linux/netdevice.h:5011 [inline] xmit_one net/core/dev.c:3590 [inline] dev_hard_start_xmit+0x27a/0x7d0 net/core/dev.c:3606 __dev_queue_xmit+0x1b73/0x3f50 net/core/dev.c:4434

CVE ID	Description
CVE-2025-21654	In the Linux kernel, the following vulnerability has been resolved: ovl: support encoding fid from inode with no alias Dmitry Safonov reported that a WARN_ON() assertion can be triggered by userspace when calling inotify_show_fdinfo() for an overlaysfs watched inode, whose dentry aliases were discarded with drop_caches. The WARN_ON() assertion in inotify_show_fdinfo() was removed, because it is possible for encoding file handle to fail for other reason, but the impact of failing to encode an overlaysfs file handle goes beyond this assertion. As shown in the LTP test case mentioned in the link below, failure to encode an overlaysfs file handle from a non-aliased inode also leads to failure to report an fid with FAN_DELETE_SELF fanotify events. As Dmitry notes in his analysis of the problem, ovl_encode_fh() fails if it cannot find an alias for the inode, but this failure can be fixed. ovl_encode_fh() seldom uses the alias and in the case of non-decodable file handles, as is often the case with fanotify fid info, ovl_encode_fh() never needs to use the alias to encode a file handle. Defer finding an alias until it is actually needed so ovl_encode_fh() will not fail in the common case of FAN_DELETE_SELF fanotify events.
CVE-2025-21655	In the Linux kernel, the following vulnerability has been resolved: io_uring/eventfd: ensure io_eventfd_signal() defers another RCU period io_eventfd_do_signal() is invoked from an RCU callback, but when dropping the reference to the io_ev_fd, it calls io_eventfd_free() directly if the refcount drops to zero. This isn't correct, as any potential freeing of the io_ev_fd should be deferred another RCU grace period. Just call io_eventfd_put() rather than open-code the dec-and-test and free, which will correctly defer it another RCU grace period.
CVE-2025-21656	In the Linux kernel, the following vulnerability has been resolved: hwmon: (drivetemp) Fix driver producing garbage data when SCSI errors occur scsi_execute_cmd() function can return both negative (linux codes) and positive (scsi_cmnd result field) error codes. Currently the driver just passes error codes of scsi_execute_cmd() to hwmon core, which is incorrect because hwmon only checks for negative error codes. This leads to hwmon reporting uninitialized data to userspace in case of SCSI errors (for example if the disk drive was disconnected). This patch checks scsi_execute_cmd() output and returns -EIO if it's error code is positive. [groeck: Avoid inline variable declaration for portability]
CVE-2025-21658	In the Linux kernel, the following vulnerability has been resolved: btrfs: avoid NULL pointer dereference if no valid extent tree [BUG] Syzbot reported a crash with the following call trace: BTRFS info (device loop0): scrub: started on devid 1 BUG: kernel NULL pointer dereference, address: 0000000000000208 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 106e70067 P4D 106e70067 PUD 107143067 PMD 0 Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 1 UID: 0 PID: 689 Comm: repro Kdump: loaded Tainted: G O 6.13.0-rc4-custom+ #206 Tainted: [O]=OOT_MODULE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS unknown 02/02/2022 RIP: 0010:find_first_extent_item+0x26/0x1f0 [btrfs] Call Trace: <TASK> scrub_find_fill_first_stripe+0x13d/0x3b0 [btrfs] scrub_simple_mirror+0x175/0x260 [btrfs] scrub_stripe+0x5d4/0x6c0 [btrfs] scrub_chunk+0xbb/0x170 [btrfs] scrub_enumerate_chunks+0x2f4/0x5f0 [btrfs] btrfs_scrub_dev+0x240/0x600 [btrfs] btrfs_ioctl+0x1dc8/0x2fa0 [btrfs] ? do_sys_openat2+0xa5/0xf0 __x64_sys_ioctl+0x97/0xc0 do_syscall_64+0x4f/0x120 entry_SYSCALL_64_after_hwframe+0x76/0x7e <TASK> [CAUSE] The reproducer is using a corrupted image where extent tree root is corrupted, thus forcing to use "rescue=all,ro" mount option to mount the image. Then it triggered a scrub, but since scrub relies on extent tree to find where the data/metadata extents are, scrub_find_fill_first_stripe() relies on a non-empty extent root. But unfortunately scrub_find_fill_first_stripe() doesn't really expect a NULL pointer for extent root, it use extent_root to grab fs_info and triggered a NULL pointer dereference. [FIX] Add an extra check for a valid extent root at the beginning of scrub_find_fill_first_stripe(). The new error path is introduced by 42437a6386ff ("btrfs: introduce mount option rescue=ignorebadroots"), but that's pretty old, and later commit b979547513ff ("btrfs: scrub: introduce helper to find and fill sector info for a scrub_stripe") changed how we do scrub. So for kernels older than 6.6, the fix will need manual backport.
CVE-2025-21659	In the Linux kernel, the following vulnerability has been resolved: netdev: prevent accessing NAPI instances from another namespace The NAPI IDs were not fully exposed to user space prior to the netlink API, so they were never namespaced. The netlink API must ensure that at the very least NAPI instance belongs to the same netns as the owner of the genl sock. napi_by_id() can become static now, but it needs to move because of dev_get_by_napi_id().
CVE-2025-21660	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix unexpectedly changed path in ksmbd_vfs_kern_path_locked When `ksmbd_vfs_kern_path_locked` met an error and it is not the last entry, it will exit without restoring changed path buffer. But later this buffer may be used as the filename for creation.

CVE ID	Description
CVE-2025-21662	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix variable not being completed when function returns When cmd_alloc_index(), fails cmd_work_handler() needs to complete ent->slotted before returning early. Otherwise the task which issued the command may hang: mlx5_core 0000:01:00.0: cmd_work_handler:877:(pid 3880418): failed to allocate command entry INFO: task kworker/13:2:4055883 blocked for more than 120 seconds. Not tainted 4.19.90-25.44.v2101.ky10.aarch64 #1 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. kworker/13:2 D 0 4055883 2 0x00000228 Workqueue: events mlx5e_tx_dim_work [mlx5_core] Call trace: __switch_to+0xe8/0x150 __schedule+0x2a8/0x9b8 schedule+0x2c/0x88 schedule_timeout+0x204/0x478 wait_for_common+0x154/0x250 wait_for_completion+0x28/0x38 cmd_exec+0x7a0/0xa00 [mlx5_core] mlx5_cmd_exec+0x54/0x80 [mlx5_core] mlx5_core_modify_cq+0x6c/0x80 [mlx5_core] mlx5_core_modify_cq_moderation+0xa0/0xb8 [mlx5_core] mlx5e_tx_dim_work+0x54/0x68 [mlx5_core] process_one_work+0x1b0/0x448 worker_thread+0x54/0x468 kthread+0x134/0x138 ret_from_fork+0x10/0x18
CVE-2025-21663	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: dwmac-tegra: Read iommu stream id from device tree Nvidia's Tegra MGBE controllers require the IOMMU "Stream ID" (SID) to be written to the MGBE_WRAP_AXI_ASID0_CTRL register. The current driver is hard coded to use MGBE0's SID for all controllers. This causes softirq time outs and kernel panics when using controllers other than MGBE0. Example dmesg errors when an ethernet cable is connected to MGBE1: [116.133290] tegra-mgbe 6910000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx [121.851283] tegra-mgbe 6910000.ethernet eth1: NETDEV WATCHDOG: CPU: 5: transmit queue 0 timed out 5690 ms [121.851782] tegra-mgbe 6910000.ethernet eth1: Reset adapter. [121.892464] tegra-mgbe 6910000.ethernet eth1: Register MEM_TYPE_PAGE_POOL RxQ-0 [121.905920] tegra-mgbe 6910000.ethernet eth1: PHY [stmmac-1:00] driver [Aquantia AQR113] (irq=171) [121.907356] tegra-mgbe 6910000.ethernet eth1: Enabling Safety Features [121.907578] tegra-mgbe 6910000.ethernet eth1: IEEE 1588-2008 Advanced Timestamp supported [121.908399] tegra-mgbe 6910000.ethernet eth1: registered PTP clock [121.908582] tegra-mgbe 6910000.ethernet eth1: configuring for phy/10gbase-r link mode [125.961292] tegra-mgbe 6910000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx [181.921198] rcu: INFO: rcu_preempt detected stalls on CPUs/tasks: [181.921404] rcu: 7-....: (1 GPs behind) idle=540c/1/0x4000000000000002 softirq=1748/1749 fqs=2337 [181.921684] rcu: (detected by 4, t=6002 jiffies, g=1357, q=1254 ncpus=8) [181.921878] Sending NMI from CPU 4 to CPUs 7: [181.921886] NMI backtrace for cpu 7 [181.922131] CPU: 7 UID: 0 PID: 0 Comm: swapper/7 Kdump: loaded Not tainted 6.13.0-rc3+ #6 [181.922390] Hardware name: NVIDIA CTI Forge + Orin AGX/Jetson, BIOS 202402.1-Unknown 10/28/2024 [181.922658] pstate: 40400009 (nZcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [181.922847] pc : handle_softirqs+0x98/0x368 [181.922978] lr : __do_softirq+0x18/0x20 [181.923095] sp : ffff80008003bf50 [181.923189] x29: ffff80008003bf50 x28: 0000000000000008 x27: 0000000000000000 [181.923379] x26: ffffce78ea277000 x25: 0000000000000000 x24: 0000001c61befda0 [181.924486] x23: 0000000060400009 x22: ffffce78e99918bc x21: ffff80008018bd70 [181.925568] x20: ffffce78e8bb00d8 x19: ffff80008018bc20 x18: 0000000000000000 [181.926655] x17: ffff318ebe7d3000 x16: ffff800080038000 x15: 0000000000000000 [181.931455] x14: ffff000080816680 x13: ffff318ebe7d3000 x12: 000000003464d91d [181.938628] x11: 0000000000000040 x10: ffff000080165a70 x9 : ffffce78e8bb0160 [181.945804] x8 : ffff8000827b3160 x7 : f9157b241586f343 x6 : eeb6502a01c81c74 [181.953068] x5 : a4acfcdd2e8096bb x4 : ffffce78ea277340 x3 : 00000000ffffd1e1 [181.960329] x2 : 0000000000000101 x1 : ffffce78ea277340 x0 : ffff318ebe7d3000 [181.967591] Call trace: [181.970043] handle_softirqs+0x98/0x368 (P) [181.974240] __do_softirq+0x18/0x20 [181.977743] __do_softirq+0x14/0x28 [181.981415] call_on_irq_stack+0x24/0x30 [181.985180] do_softirq_own_stack+0x20/0x30 [181.989379] __irq_exit_rcu+0x114/0x140 [181.993142] irq_exit_rcu+0x14/0x28 [181.996816] el1_interrupt+0x44/0xb8 [182.000316] el1h_64_irq_handler+0x14/0x20 [182.004343] el1h_64_irq+0x80/0x88 [182.007755] cpuidle_enter_state+0xc4/0x4a8 (P) [182.012305] cpuidle_enter+0x3c/0x58 [182.015980] cpuidle_idle_call+0x128/0x1c0 [182.020005] do_idle+0xe0/0xf0 [182.023155] cpu_startup_entry+0x3c/0x48 [182.026917] secondary_start_kernel+0xdc/0x120 [182.031379] __secondary_switched+0x74/0x78 [212.971162] rcu: INFO: rcu_preempt detected expedited stalls on CPUs/tasks: { 7-.... } 6103 jiffies s: 417 root: 0x80/. [212.985935] rcu: blocking rcu_node structures (internal RCU debug): [212.992758] Sending NMI from CPU 0 to CPUs 7: [212.998539] NMI backtrace for cpu 7 [213.004304] CPU: 7 UID: 0 PI ---truncated---

CVE ID	Description
CVE-2025-21664	In the Linux kernel, the following vulnerability has been resolved: dm thin: make get_first_thin use rcu-safe list first function The documentation in rculist.h explains the absence of list_empty_rcu() and cautions programmers against relying on a list_empty() -> list_first() sequence in RCU safe code. This is because each of these functions performs its own READ_ONCE() of the list head. This can lead to a situation where the list_empty() sees a valid list entry, but the subsequent list_first() sees a different view of list head state after a modification. In the case of dm-thin, this author had a production box crash from a GP fault in the process_deferred_bios path. This function saw a valid list head in get_first_thin() but when it subsequently dereferenced that and turned it into a thin_c, it got the inside of the struct pool, since the list was now empty and referring to itself. The kernel on which this occurred printed both a warning about a refcount_t being saturated, and a UBSAN error for an out-of-bounds cpuid access in the queued spinlock, prior to the fault itself. When the resulting kdump was examined, it was possible to see another thread patiently waiting in thin_dtr's synchronize_rcu. The thin_dtr call managed to pull the thin_c out of the active thins list (and have it be the last entry in the active_thins list) at just the wrong moment which lead to this crash. Fortunately, the fix here is straight forward. Switch get_first_thin() function to use list_first_or_null_rcu() which performs just a single READ_ONCE() and returns NULL if the list is already empty. This was run against the devicemapper test suite's thin-provisioning suites for delete and suspend and no regressions were observed.
CVE-2025-21665	In the Linux kernel, the following vulnerability has been resolved: filemap: avoid truncating 64-bit offset to 32 bits On 32-bit kernels, folio_seek_hole_data() was inadvertently truncating a 64-bit value to 32 bits, leading to a possible infinite loop when writing to an xfs filesystem.
CVE-2025-21666	In the Linux kernel, the following vulnerability has been resolved: vsock: prevent null-ptr-deref in vsock_*[has_data has_space] Recent reports have shown how we sometimes call vsock_*_has_data() when a vsock socket has been de-assigned from a transport (see attached links), but we shouldn't. Previous commits should have solved the real problems, but we may have more in the future, so to avoid null-ptr-deref, we can return 0 (no space, no data available) but with a warning. This way the code should continue to run in a nearly consistent state and have a warning that allows us to debug future problems.
CVE-2025-21667	In the Linux kernel, the following vulnerability has been resolved: iomap: avoid avoid truncating 64-bit offset to 32 bits on 32-bit kernels, iomap_write_delalloc_scan() was inadvertently using a 32-bit position due to folio_next_index() returning an unsigned long. This could lead to an infinite loop when writing to an xfs filesystem.
CVE-2025-21668	In the Linux kernel, the following vulnerability has been resolved: pmdomain: imx8mp-blk-ctrl: add missing loop break condition Currently imx8mp_blk_ctrl_remove() will continue the for loop until an out-of-bounds exception occurs. pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPPE=--) pc : dev_pm_domain_detach+0x8/0x48 lr : imx8mp_blk_ctrl_shutdown+0x58/0x90 sp : ffffffff084f8bbf0 x29: ffffffff084f8bbf0 x28: ffffff80daf32ac0 x27: 0000000000000000 x26: ffffffff081658d78 x25: 0000000000000001 x24: ffffffff08201b028 x23: ffffff80d0db9490 x22: ffffffff082340a78 x21: 000000000000005b0 x20: ffffff80d19bc180 x19: 000000000000000a x18: ffffffff080a39e08 x16: ffffffff080a39c98 x15: 4f435f464f006c72 x14: 0000000000000004 x13: ffffff80d0172110 x12: 0000000000000000 x11: ffffff80d0537740 x10: ffffff80d05376c0 x9 : ffffffff0808ed2d8 x8 : ffffffff084f8bab0 x7 : 0000000000000000 x6 : 0000000000000000 x5 : ffffff80d19b9420 x4 : ffffffe03466e60 x3 : 0000000008080077 x2 : 0000000000000000 x1 : 0000000000000001 x0 : 0000000000000000 Call trace: dev_pm_domain_detach+0x8/0x48 platform_shutdown+0x2c/0x48 device_shutdown+0x158/0x268 kernel_restart_prepare+0x40/0x58 kernel_kexec+0x58/0xe8 __do_sys_reboot+0x198/0x258 __arm64_sys_reboot+0x2c/0x40 invoke_syscall+0x5c/0x138 el0_svc_common.constprop.0+0x48/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x38/0xc8 el0t_64_sync_handler+0x120/0x130 el0t_64_sync+0x190/0x198 Code: 8128c2d0 fffffffc0 aa1e03e9 d503201f
CVE-2025-21669	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: discard packets if the transport changes If the socket has been de-assigned or assigned to another transport, we must discard any packets received because they are not expected and would cause issues when we access vsk->transport. A possible scenario is described by Hyunwoo Kim in the attached link, where after a first connect() interrupted by a signal, and a second connect() failed, we can find `vsk->transport` at NULL, leading to a NULL pointer dereference.

CVE ID	Description
CVE-2025-21670	In the Linux kernel, the following vulnerability has been resolved: vsock/bpf: return early if transport is not assigned Some of the core functions can only be called if the transport has been assigned. As Michal reported, a socket might have the transport at NULL, for example after a failed connect(), causing the following trace: BUG: kernel NULL pointer dereference, address: 00000000000000a0 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 12faf8067 P4D 12faf8067 PUD 113670067 PMD 0 Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 15 UID: 0 PID: 1198 Comm: a.out Not tainted 6.13.0-rc2+ RIP: 0010:vsock_connectible_has_data+0x1f/0x40 Call Trace: vsock_bpf_recvmmsg+0xca/0x5e0 sock_recvmmsg+0xb9/0xc0 __sys_recvfrom+0xb3/0x130 __x64_sys_recvfrom+0x20/0x30 do_syscall_64+0x93/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e So we need to check the `vsk->transport` in vsock_bpf_recvmmsg(), especially for connected sockets (stream/seqpacket) as we already do in __vsock_connectible_recvmmsg().
CVE-2025-21671	In the Linux kernel, the following vulnerability has been resolved: zram: fix potential UAF of zram table If zram_meta_alloc failed early, it frees allocated zram->table without setting it NULL. Which will potentially cause zram_meta_free to access the table if user reset an failed and uninitialized device.
CVE-2025-21672	In the Linux kernel, the following vulnerability has been resolved: afs: Fix merge preference rule failure condition syzbot reported a lock held when returning to userspace[1]. This is because if argc is less than 0 and the function returns directly, the held inode lock is not released. Fix this by store the error in ret and jump to done to clean up instead of returning directly. [dh: Modified Lizhi Xu's original patch to make it honour the error code from afs_split_string()] [1] WARNING: lock held when returning to user space! 6.13.0-rc3-syzkaller-00209-g499551201b5f #0 Not tainted ----- syz-executor133/5823 is leaving the kernel with locks still held! 1 lock held by syz-executor133/5823: #0: ffff888071cffc00 (&sb->s_type->i_mutex_key#9) {++++}-{4:4}, at: inode_lock include/linux/fs.h:818 [inline] #0: ffff888071cffc00 (&sb->s_type->i_mutex_key#9){++++}-{4:4}, at: afs_proc_addr_prefs_write+0x2bb/0x14e0 fs/afs/addr_prefs.c:388
CVE-2025-21673	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix double free of TCP_Server_Info::hostname When shutting down the server in cifs_put_tcp_session(), cifsd thread might be reconnecting to multiple DFS targets before it realizes it should exit the loop, so @server->hostname can't be freed as long as cifsd thread isn't done. Otherwise the following can happen: RIP: 0010:__slab_free+0x223/0x3c0 Code: 5e 41 5f c3 cc cc cc 4c 89 de 4c 89 cf 44 89 44 24 08 4c 89 1c 24 e8 fb cf 8e 00 44 8b 44 24 08 4c 8b 1c 24 e9 5f fe ff <0f> 0b 41 f7 45 08 00 0d 21 00 0f 85 2d ff ff ff e9 1f ff ff ff 80 RSP: 0018:ffffb26180dbfd08 EFLAGS: 00010246 RAX: ffff8ea34728e510 RBX: ffff8ea34728e500 RCX: 0000000000800068 RDX: 0000000000800068 RSI: 0000000000000000 RDI: ffff8ea340042400 RBP: ffffe112041ca380 R08: 0000000000000001 R09: 0000000000000000 R10: 6170732e31303000 R11: 70726f632e786563 R12: ffff8ea34728e500 R13: ffff8ea340042400 R14: ffff8ea34728e500 R15: 0000000000800068 FS: 0000000000000000(0000) GS:ffff8ea66fd80000(0000) 000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007ffc25376080 CR3: 000000012a2ba001 CR4: PKRU: 55555554 Call Trace: <TASK> ? show_trace_log_lvl+0x1c4/0x2df ? show_trace_log_lvl+0x1c4/0x2df ? __reconnect_target_unlocked+0x3e/0x160 [cifs] ? __die_body.cold+0x8/0xd ? die+0x2b/0x50 ? do_trap+0xce/0x120 ? __slab_free+0x223/0x3c0 ? do_error_trap+0x65/0x80 ? __slab_free+0x223/0x3c0 ? exc_invalid_op+0x4e/0x70 ? __slab_free+0x223/0x3c0 ? asm_exc_invalid_op+0x16/0x20 ? __slab_free+0x223/0x3c0 ? extract_hostname+0x5c/0xa0 [cifs] ? extract_hostname+0x5c/0xa0 [cifs] ? __kmalloc+0x4b/0x140 __reconnect_target_unlocked+0x3e/0x160 [cifs] reconnect_dfs_server+0x145/0x430 [cifs] cifs_handle_standard+0x1ad/0x1d0 [cifs] cifs_demultiplex_thread+0x592/0x730 [cifs] ? __pfx_cifs_demultiplex_thread+0x10/0x10 [cifs] kthread+0xdd/0x100 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x29/0x50 </TASK>

CVE ID	Description
CVE-2025-21674	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix inversion dependency warning while enabling IPsec tunnel Attempt to enable IPsec packet offload in tunnel mode in debug kernel generates the following kernel panic, which is happening due to two issues: 1. In SA add section, the should be _bh() variant when marking SA mode. 2. There is not needed flush_workqueue in SA delete routine. It is not needed as at this stage as it is removed from SADB and the running work will be canceled later in SA free. ===== WARNING: SOFTIRQ-safe -> SOFTIRQ-unsafe lock order detected 6.12.0+ #4 Not tainted ----- charon/1337 [HC0[0];SC0[4];HE1:SE0] is trying to acquire: ffff88810f365020 (&xa->xa_lock#24){+.-.-}{3:3}, at: mlx5e_xfrm_del_state+0xca/0x1e0 [mlx5_core] and this task is already holding: ffff88813e0f0d48 (&x->lock){+.-.-}{3:3}, at: xfrm_state_delete+0x16/0x30 which would create a new lock dependency: (&x->lock){+.-.-}{3:3} -> (&xa->xa_lock#24){+.-.-}{3:3} but this new dependency connects a SOFTIRQ-irq-safe lock: (&x->lock){+.-.-}{3:3} ... which became SOFTIRQ-irq-safe at: lock_acquire+0x1be/0x520 _raw_spin_lock_bh+0x34/0x40 xfrm_timer_handler+0x91/0xd70 __hrtimer_run_queues+0x1dd/0xa60 hrtimer_run_softirq+0x146/0x2e0 handle_softirqs+0x266/0x860 irq_exit_rcu+0x115/0x1a0 sysvec_apic_timer_interrupt+0x6e/0x90 asm_sysvec_apic_timer_interrupt+0x16/0x20 default_idle+0x13/0x20 default_idle_call+0x67/0xa0 do_idle+0x2da/0x320 cpu_startup_entry+0x50/0x60 start_secondary+0x213/0x2a0 common_startup_64+0x129/0x138 to a SOFTIRQ-irq-unsafe lock: (&xa->xa_lock#24){+.-.-}{3:3} ... which became SOFTIRQ-irq-unsafe at: ... lock_acquire+0x1be/0x520 _raw_spin_lock+0x2c/0x40 xa_set_mark+0x70/0x110 mlx5e_xfrm_add_state+0xe48/0x2290 [mlx5_core] xfrm_dev_state_add+0x3bb/0xd70 xfrm_add_sa+0x2451/0x4a90 xfrm_user_rcv_msg+0x493/0x880 netlink_rcv_skb+0x12e/0x380 xfrm_netlink_rcv+0x6d/0x90 netlink_unicast+0x42f/0x740 netlink_sendmsg+0x745/0xbe0 __sock_sendmsg+0xc5/0x190 __sys_sendto+0x1fe/0x2c0 __x64_sys_sendto+0xdc/0x1b0 do_syscall_64+0x6d/0x140 entry_SYSCALL_64_after_hwframe+0x4b/0x53 other info that might help us debug this: Possible interrupt unsafe locking scenario: CPU0 CPU1 ---- lock(&xa->xa_lock#24); local_irq_disable(); lock(&x->lock); lock(&xa->xa_lock#24); <Interrupt> lock(&x->lock); *** DEADLOCK *** 2 locks held by charon/1337: #0: ffffffff87f8f858 (&net->xfrm.xfrm_cfg_mutex){+.-.-}{4:4}, at: xfrm_netlink_rcv+0x5e/0x90 #1: ffff88813e0f0d48 (&x->lock){+.-.-}{3:3}, at: xfrm_state_delete+0x16/0x30 the dependencies between SOFTIRQ-irq-safe lock and the holding lock: -> (&x->lock){+.-.-}{3:3} ops: 29 { HARDIRQ-ON-W at: lock_acquire+0x1be/0x520 _raw_spin_lock_bh+0x34/0x40 xfrm_alloc_spi+0xc0/0xe60 xfrm_alloc_userspi+0x5f6/0xbc0 xfrm_user_rcv_msg+0x493/0x880 netlink_rcv_skb+0x12e/0x380 xfrm_netlink_rcv+0x6d/0x90 netlink_unicast+0x42f/0x740 netlink_sendmsg+0x745/0xbe0 __sock_sendmsg+0xc5/0x190 __sys_sendto+0x1fe/0x2c0 __x64_sys_sendto+0xdc/0x1b0 do_syscall_64+0x6d/0x140 entry_SYSCALL_64_after_hwframe+0x4b/0x53 IN-SOFTIRQ-W at: lock_acquire+0x1be/0x520 _raw_spin_lock_bh+0x34/0x40 xfrm_timer_handler+0x91/0xd70 __hrtimer_run_queues+0x1dd/0xa60 ---truncated---

CVE ID	Description
CVE-2025-21675	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Clear port select structure when fail to create Clear the port select structure on error so no stale values left after definers are destroyed. That's because the mlx5_lag_destroy_definers() always try to destroy all lag definers in the tt_map, so in the flow below lag definers get double-destroyed and cause kernel crash: mlx5_lag_port_sel_create() mlx5_lag_create_definers() mlx5_lag_create_definer() <- Failed on tt 1 mlx5_lag_destroy_definers() <- definers[tt=0] gets destroyed mlx5_lag_port_sel_create() mlx5_lag_create_definers() mlx5_lag_create_definer() <- Failed on tt 0 mlx5_lag_destroy_definers() <- definers[tt=0] gets double-destroyed Unable to handle kernel NULL pointer dereference at virtual address 0000000000000008 Mem abort info: ESR = 0x0000000096000005 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x05: level 1 translation fault Data abort info: ISV = 0, ISS = 0x000000005, ISS2 = 0x000000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtable: 64k pages, 48-bit VAs, pgdp=0000000112ce2e00 [0000000000000008] pgd=0000000000000000, p4d=0000000000000000, pud=0000000000000000 Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP Modules linked in: iptable_raw bonding ip_gre ip6_gre gre ip6_tunnel tunnel6 geneve ip6_udp_tunnel udp_tunnel ipip tunnel4 ip_tunnel rdma_ucm(OE) rdma_cm(OE) iw_cm(OE) ib_ipoib(OE) ib_cm(OE) ib_umad(OE) mlx5_ib(OE) ib_uverbs(OE) mlx5_fwctl(OE) fwctl(OE) mlx5_core(OE) mlxdevm(OE) ib_core(OE) mlxfw(OE) memtrack(OE) mlx_compat(OE) openvswitch nsh nf_conntrack psample xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xfrm_user xfrm_algo xt_addrtype iptable_filter iptable_nat nf_nat nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 br_netfilter bridge stp llc netconsole overlay efi_pstore sch_fq_codel zram ip_tables crc10dif_ce qemu_fw_cfg fuse ipv6 crc_ccitt [last unloaded: mlx_compat(OE)] CPU: 3 UID: 0 PID: 217 Comm: kworker/u53:2 Tainted: G OE 6.11.0+ #2 Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 Workqueue: mlx5_lag mlx5_do_bond_work [mlx5_core] pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : mlx5_del_flow_rules+0x24/0x2c0 [mlx5_core] lr : mlx5_lag_destroy_definer+0x54/0x100 [mlx5_core] sp : ffff800085fafb00 x29: ffff800085fafb00 x28: ffff0000da0c8000 x27: 0000000000000000 x26: ffff0000da0c8000 x25: ffff0000da0c8000 x24: ffff0000da0c8000 x23: ffff0000c31f81a0 x22: 0400000000000000 x21: ffff0000da0c8000 x20: 0000000000000000 x19: 0000000000000001 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000ffff8b0c9350 x14: 0000000000000000 x13: ffff800081390d18 x12: ffff800081dc3cc0 x11: 0000000000000001 x10: 00000000000000b10 x9 : ffff80007ab7304c x8 : ffff0000d00711f0 x7 : 0000000000000004 x6 : 0000000000000190 x5 : ffff00027edb3010 x4 : 0000000000000000 x3 : 0000000000000000 x2 : ffff0000d39b8000 x1 : ffff0000d39b8000 x0 : 0400000000000000 Call trace: mlx5_del_flow_rules+0x24/0x2c0 [mlx5_core] mlx5_lag_destroy_definer+0x54/0x100 [mlx5_core] mlx5_lag_destroy_definers+0xa0/0x108 [mlx5_core] mlx5_lag_port_sel_create+0x2d4/0x6f8 [mlx5_core] mlx5_activate_lag+0x60c/0x6f8 [mlx5_core] mlx5_do_bond_work+0x284/0x5c8 [mlx5_core] process_one_work+0x170/0x3e0 worker_thread+0x2d8/0x3e0 kthread+0x11c/0x128 ret_from_fork+0x10/0x20 Code: a9025bf5 aa0003f6 a90363f7 f90023f9 (f9400400) ---[end trace 0000000000000000]---
CVE-2025-21676	In the Linux kernel, the following vulnerability has been resolved: net: fec: handle page_pool_dev_alloc_pages error The fec_enet_update_cbd function calls page_pool_dev_alloc_pages but did not handle the case when it returned NULL. There was a WARN_ON(!new_page) but it would still proceed to use the NULL pointer and then crash. This case does seem somewhat rare but when the system is under memory pressure it can happen. One case where I can duplicate this with some frequency is when writing over a smb share to a SATA HDD attached to an imx6q. Setting /proc/sys/vm/min_free_kbytes to higher values also seems to solve the problem for my test case. But it still seems wrong that the fec driver ignores the memory allocation error and can crash. This commit handles the allocation error by dropping the current packet.

CVE ID	Description
CVE-2025-21678	In the Linux kernel, the following vulnerability has been resolved: gtp: Destroy device along with udp socket's netns dismantle. gtp_newlink() links the device to a list in dev_net(dev) instead of src_net, where a udp tunnel socket is created. Even when src_net is removed, the device stays alive on dev_net(dev). Then, removing src_net triggers the splat below. [0] In this example, gtp0 is created in ns2, and the udp socket is created in ns1. ip netns add ns1 ip netns add ns2 ip -n ns1 link add netns ns2 name gtp0 type gtp role sgsn ip netns del ns1 Let's link the device to the socket's netns instead. Now, gtp_net_exit_batch_rtnl() needs another netdev iteration to remove all gtp devices in the netns. [0]: ref_tracker: net notrefcnt@000000003d6e7d05 has 1/2 users at sk_alloc (/include/net/net_namespace.h:345 net/core/sock.c:2236) inet_create (net/ipv4/af_inet.c:326 net/ipv4/af_inet.c:252) __sock_create (net/socket.c:1558) udp_sock_create4 (net/ipv4/udp_tunnel_core.c:18) gtp_create_sock (/include/net/udp_tunnel.h:59 drivers/net/gtp.c:1423) gtp_create_sockets (drivers/net/gtp.c:1447) gtp_newlink (drivers/net/gtp.c:1507) rtnl_newlink (net/core/rtnetlink.c:3786 net/core/rtnetlink.c:3897 net/core/rtnetlink.c:4012) rtnetlink_rcv_msg (net/core/rtnetlink.c:6922) netlink_rcv_skb (net/netlink/af_netlink.c:2542) netlink_unicast (net/netlink/af_netlink.c:1321 net/netlink/af_netlink.c:1347) netlink_sendmsg (net/netlink/af_netlink.c:1891) __sys_sendmsg (net/socket.c:711 net/socket.c:726 net/socket.c:2583) __sys_sendmsg (net/socket.c:2639) __sys_sendmsg (net/socket.c:2669) do_syscall_64 (arch/x86/entry/common.c:52 arch/x86/entry/common.c:83) WARNING: CPU: 1 PID: 60 at lib/ref_tracker.c:179 ref_tracker_dir_exit (lib/ref_tracker.c:179) Modules linked in: CPU: 1 UID: 0 PID: 60 Comm: kworker/u16:2 Not tainted 6.13.0-rc5-00147-g4c1224501e9d #5 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 Workqueue: netns cleanup_net RIP: 0010:ref_tracker_dir_exit (lib/ref_tracker.c:179) Code: 00 00 00 fc ff df 4d 8b 26 49 bd 00 01 00 00 00 ad de 4c 39 f5 0f 85 df 00 00 00 48 8b 74 24 08 48 89 df e8 a5 cc 12 02 90 <0f> 0b 90 48 8d 6b 44 be 04 00 00 00 48 89 ef e8 80 de 67 ff 48 89 RSP: 0018:ff11000009a07b60 EFLAGS: 00010286 RAX: 0000000000002bd3 RBX: ff1100000f4e1aa0 RCX: 1fffffff0e40ac6 RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffffffff8423ee3c RBP: ff1100000f4e1af0 R08: 0000000000000001 R09: fffffbfff0e395ae R10: 0000000000000001 R11: 0000000000036001 R12: ff1100000f4e1af0 R13: dead000000000100 R14: ff1100000f4e1af0 R15: dffffc0000000000 FS: 0000000000000000(0000) GS:ff1100006ce80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f9b2464bd98 CR3: 0000000005286005 CR4: 00000000000771ef DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe07f0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> ? __warn (kernel/panic.c:748) ? ref_tracker_dir_exit (lib/ref_tracker.c:179) ? report_bug (lib/bug.c:201 lib/bug.c:219) ? handle_bug (arch/x86/kernel/traps.c:285) ? exc_invalid_op (arch/x86/kernel/traps.c:309 (discriminator 1)) ? asm_exc_invalid_op (/arch/x86/include/asm/identry.h:621) ? _raw_spin_unlock_irqrestore (/arch/x86/include/asm/irqflags.h:42 ./arch/x86/include/asm/irqflags.h:97 ./arch/x86/include/asm/irqflags.h:155 ./include/linux/spinlock_api_smp.h:151 kernel/locking/spinlock.c:194) ? ref_tracker_dir_exit (lib/ref_tracker.c:179) ? __pfx_ref_tracker_dir_exit (lib/ref_tracker.c:158) ? kfree (mm/slub.c:4613 mm/slub.c:4761) net_free (net/core/net_namespace.c:476 net/core/net_namespace.c:467) cleanup_net (net/core/net_namespace.c:664 (discriminator 3)) process_one_work (kernel/workqueue.c:3229) worker_thread (kernel/workqueue.c:3304 kernel/workqueue.c:3391 ---truncated---
CVE-2025-21680	In the Linux kernel, the following vulnerability has been resolved: pktgen: Avoid out-of-bounds access in get_imix_entries Passing a sufficient amount of imix entries leads to invalid access to the pkt_dev->imix_entries array because of the incorrect boundary check. UBSAN: array-index-out-of-bounds in net/core/pktgen.c:874:24 index 20 is out of range for type 'imix_pkt [20]' CPU: 2 PID: 1210 Comm: bash Not tainted 6.10.0-rc1 #121 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) Call Trace: <TASK> dump_stack_lvl lib/dump_stack.c:117 __ubsan_handle_out_of_bounds lib/ubsan.c:429 get_imix_entries net/core/pktgen.c:874 pktgen_if_write net/core/pktgen.c:1063 pde_write fs/proc/inode.c:334 proc_reg_write fs/proc/inode.c:346 vfs_write fs/read_write.c:593 ksys_write fs/read_write.c:644 do_syscall_64 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe arch/x86/entry/entry_64.S:130 Found by Linux Verification Center (linuxtesting.org) with SVACE. [fp: allow to fill the array completely; minor changelog cleanup]

CVE ID	Description
CVE-2025-21681	In the Linux kernel, the following vulnerability has been resolved: openvswitch: fix lockup on tx to unregistering netdev with carrier Commit in a fixes tag attempted to fix the issue in the following sequence of calls: do_output -> ovs_vport_send -> dev_queue_xmit -> __dev_queue_xmit -> netdev_core_pick_tx -> skb_tx_hash When device is unregistering, the 'dev->real_num_tx_queues' goes to zero and the 'while (unlikely(hash >= qcount))' loop inside the 'skb_tx_hash' becomes infinite, locking up the core forever. But unfortunately, checking just the carrier status is not enough to fix the issue, because some devices may still be in unregistering state while reporting carrier status OK. One example of such device is a net/dummy. It sets carrier ON on start, but it doesn't implement .ndo_stop to set the carrier off. And it makes sense, because dummy doesn't really have a carrier. Therefore, while this device is unregistering, it's still easy to hit the infinite loop in the skb_tx_hash() from the OVS datapath. There might be other drivers that do the same, but dummy by itself is important for the OVS ecosystem, because it is frequently used as a packet sink for tcpdump while debugging OVS deployments. And when the issue is hit, the only way to recover is to reboot. Fix that by also checking if the device is running. The running state is handled by the net core during unregistering, so it covers unregistering case better, and we don't really need to send packets to devices that are not running anyway. While only checking the running state might be enough, the carrier check is preserved. The running and the carrier states seem disjointed throughout the code and different drivers. And other core functions like __dev_direct_xmit() check both before attempting to transmit a packet. So, it seems safer to check both flags in OVS as well.
CVE-2025-21682	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: always recalculate features after XDP clearing, fix null-deref Recalculate features when XDP is detached. Before: # ip li set dev eth0 xdp obj xdp_dummy.bpf.o sec xdp # ip li set dev eth0 xdp off # ethtool -k eth0 grep gro rx-gro-hw: off [requested on] After: # ip li set dev eth0 xdp obj xdp_dummy.bpf.o sec xdp # ip li set dev eth0 xdp off # ethtool -k eth0 grep gro rx-gro-hw: on The fact that HW-GRO doesn't get re-enabled automatically is just a minor annoyance. The real issue is that the features will randomly come back during another reconfiguration which just happens to invoke netdev_update_features(). The driver doesn't handle reconfiguring two things at a time very robustly. Starting with commit 98ba1d931f61 ("bnxt_en: Fix RSS logic in __bnxt_reserve_rings()") we only reconfigure the RSS hash table if the "effective" number of Rx rings has changed. If HW-GRO is enabled "effective" number of rings is 2x what user sees. So if we are in the bad state, with HW-GRO re-enablement "pending" after XDP off, and we lower the rings by / 2 - the HW-GRO rings doing 2x and the ethtool -L doing / 2 may cancel each other out, and the: if (old_rx_rings != bp->hw_rescv_rx_rings && condition in __bnxt_reserve_rings() will be false. The RSS map won't get updated, and we'll crash with: BUG: kernel NULL pointer dereference, address: 0000000000000168 RIP: 0010: __bnxt_hwrn_vnic_set_rss+0x13a/0x1a0 bnxt_hwrn_vnic_rss_cfg_p5+0x47/0x180 __bnxt_setup_vnic_p5+0x58/0x110 bnxt_init_nic+0xb72/0xf50 __bnxt_open_nic+0x40d/0xab0 bnxt_open_nic+0x2b/0x60 ethtool_set_channels+0x18c/0x1d0 As we try to access a freed ring. The issue is present since XDP support was added, really, but prior to commit 98ba1d931f61 ("bnxt_en: Fix RSS logic in __bnxt_reserve_rings()") it wasn't causing major issues.
CVE-2025-21683	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix bpf_sk_select_reuseport() memory leak As pointed out in the original comment, lookup in sockmap can return a TCP ESTABLISHED socket. Such TCP socket may have had SO_ATTACH_REUSEPORT_EBPF set before it was ESTABLISHED. In other words, a non-NULL sk_reuseport_cb does not imply a non-refcounted socket. Drop sk's reference in both error paths. unreferenced object 0xffff888101911800 (size 2048): comm "test_progs", pid 44109, jiffies 4297131437 hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace (crc 9336483b): __kmalloc_noprof+0x3bf/0x560 __reuseport_alloc+0x1d/0x40 reuseport_alloc+0xca/0x150 reuseport_attach_prog+0x87/0x140 sk_reuseport_attach_bpf+0xc8/0x100 sk_setsockopt+0x1181/0x1990 do_sock_setsockopt+0x12b/0x160 __sys_setsockopt+0x7b/0xc0 __x64_sys_setsockopt+0x1b/0x30 do_syscall_64+0x93/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e

CVE ID	Description
CVE-2025-21684	In the Linux kernel, the following vulnerability has been resolved: gpio: xilinx: Convert gpio_lock to raw spinlock irq_chip functions may be called in raw spinlock context. Therefore, we must also use a raw spinlock for our own internal locking. This fixes the following lockdep splat: [5.349336] ===== [5.353349] [BUG: Invalid wait context] [5.357361] 6.13.0-rc5+ #69 Tainted: G W [5.363031] ----- [5.367045] kworker/u17:1/44 is trying to lock: [5.371587] ffffff88018b02c0 (&chip->gpio_lock){....}-{3:3}, at: xgpio_irq_unmask (drivers/gpio/gpio-xilinx.c:433 (discriminator 8)) [5.380079] other info that might help us debug this: [5.385138] context-{5:5} [5.387762] 5 locks held by kworker/u17:1/44: [5.392123] #0: ffffff8800014958 ((wq_completion)events_unbound){+.+.}-{0:0}, at: process_one_work (kernel/workqueue.c:3204) [5.402260] #1: ffffff8082fcbdd8 (deferred_probe_work){+.+.}-{0:0}, at: process_one_work (kernel/workqueue.c:3205) [5.411528] #2: ffffff880172c900 (&dev->mutex){....}-{4:4}, at: __device_attach (drivers/base/dd.c:1006) [5.419929] #3: ffffff88039c8268 (request_class#2){+.+.}-{4:4}, at: __setup_irq (kernel/irq/internals.h:156 kernel/irq/manage.c:1596) [5.428331] #4: ffffff88039c80c8 (lock_class#2){....}-{2:2}, at: __setup_irq (kernel/irq/manage.c:1614) [5.436472] stack backtrace: [5.439359] CPU: 2 UID: 0 PID: 44 Comm: kworker/u17:1 Tainted: G W 6.13.0-rc5+ #69 [5.448690] Tainted: [W]=WARN [5.451656] Hardware name: xlnx,zynqmp (DT) [5.455845] Workqueue: events_unbound deferred_probe_work_func [5.461699] Call trace: [5.464147] show_stack+0x18/0x24 C [5.467821] dump_stack_lvl (lib/dump_stack.c:123) [5.471501] dump_stack (lib/dump_stack.c:130) [5.474824] __lock_acquire (kernel/locking/lockdep.c:4828 kernel/locking/lockdep.c:4898 kernel/locking/lockdep.c:5176) [5.478758] lock_acquire (arch/arm64/include/asm/percpu.h:40 kernel/locking/lockdep.c:467 kernel/locking/lockdep.c:5851 kernel/locking/lockdep.c:5814) [5.482429] _raw_spin_lock_irqsave (include/linux/spinlock_api_smp.h:111 kernel/locking/spinlock.c:162) [5.486797] xgpio_irq_unmask (drivers/gpio/gpio-xilinx.c:433 (discriminator 8)) [5.490737] irq_enable (kernel/irq/internals.h:236 kernel/irq/chip.c:170 kernel/irq/chip.c:439 kernel/irq/chip.c:432 kernel/irq/chip.c:345) [5.494060] __irq_startup (kernel/irq/internals.h:241 kernel/irq/chip.c:180 kernel/irq/chip.c:250) [5.497645] irq_startup (kernel/irq/chip.c:270) [5.501143] __setup_irq (kernel/irq/manage.c:1807) [5.504728] request_threaded_irq (kernel/irq/manage.c:2208)
CVE-2025-21687	In the Linux kernel, the following vulnerability has been resolved: vfio/platform: check the bounds of read/write syscalls count and offset are passed from user space and not checked, only offset is capped to 40 bits, which can be used to read/write out of bounds of the device.
CVE-2025-21689	In the Linux kernel, the following vulnerability has been resolved: USB: serial: quatech2: fix null-ptr-deref in qt2_process_read_urb() This patch addresses a null-ptr-deref in qt2_process_read_urb() due to an incorrect bounds check in the following: if (newport > serial->num_ports) { dev_err(&port->dev, "%s - port change to invalid port: %i\n", __func__, Newport); break; } The condition doesn't account for the valid range of the serial->port buffer, which is from 0 to serial->num_ports - 1. When Newport is equal to serial->num_ports, the assignment of "port" in the following code is out-of-bounds and NULL: serial_priv->current_port = Newport; port = serial->port[serial_priv->current_port]; The fix checks if Newport is greater than or equal to serial->num_ports indicating it is out-of-bounds.
CVE-2025-21690	In the Linux kernel, the following vulnerability has been resolved: scsi: storvsc: Ratelimit warning logs to prevent VM denial of service If there's a persistent error in the hypervisor, the SCSI warning for failed I/O can flood the kernel log and max out CPU utilization, preventing troubleshooting from the VM side. Ratelimit the warning so it doesn't DoS the VM.
CVE-2025-21691	In the Linux kernel, the following vulnerability has been resolved: cachestat: fix page cache statistics permission checking When the 'cachestat()' system call was added in commit cf264e1329fb ("cachestat: implement cachestat syscall"), it was meant to be a much more convenient (and performant) version of mincore() that didn't need mapping things into the user virtual address space in order to work. But it ended up missing the "check for writability or ownership" fix for mincore(), done in commit 134fca9063ad ("mm/mincore.c: make mincore() more conservative"). This just adds equivalent logic to 'cachestat()', modified for the file context (rather than vma).

CVE ID	Description
CVE-2025-21692	In the Linux kernel, the following vulnerability has been resolved: net: sched: fix ets qdisc OOB Indexing Haowei Yan <g1042620637@gmail.com> found that ets_class_from_arg() can index an Out-Of-Bound class in ets_class_from_arg() when passed clid of 0. The overflow may cause local privilege escalation. [18.852298] -----[cut here]----- [18.853271] UBSAN: array-index-out-of-bounds in net/sched/sch_ets.c:93:20 [18.853743] index 18446744073709551615 is out of range for type 'ets_class [16]' [18.854254] CPU: 0 UID: 0 PID: 1275 Comm: poc Not tainted 6.12.6-dirty #17 [18.854821] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 [18.856532] Call Trace: [18.857441] <TASK> [18.858227] dump_stack_lvl+0xc2/0xf0 [18.859607] dump_stack+0x10/0x20 [18.860908] __ubsan_handle_out_of_bounds+0xa7/0xf0 [18.864022] ets_class_change+0x3d6/0x3f0 [18.864322] tc_ctl_tclass+0x251/0x910 [18.864587] ? lock_acquire+0x5e/0x140 [18.865113] ? __mutex_lock+0x9c/0xe70 [18.866009] ? __mutex_lock+0xa34/0xe70 [18.866401] rtnetlink_rcv_msg+0x170/0x6f0 [18.866806] ? __lock_acquire+0x578/0xc10 [18.867184] ? __pfx_rtnetlink_rcv_msg+0x10/0x10 [18.867503] netlink_rcv_skb+0x59/0x110 [18.867776] rtnetlink_rcv+0x15/0x30 [18.868159] netlink_unicast+0x1c3/0x2b0 [18.868440] netlink_sendmsg+0x239/0x4b0 [18.868721] __sys_sendmsg+0x3e2/0x410 [18.869012] __sys_sendmsg+0x88/0xe0 [18.869276] ? rseq_ip_fixup+0x198/0x260 [18.869563] ? rseq_update_cpu_node_id+0x10a/0x190 [18.869900] ? trace_hardirqs_off+0x5a/0xd0 [18.870196] ? syscall_exit_to_user_mode+0xcc/0x220 [18.870547] ? do_syscall_64+0x93/0x150 [18.870821] ? __memcg_slab_free_hook+0x69/0x290 [18.871157] __sys_sendmsg+0x69/0xd0 [18.871416] __x64_sys_sendmsg+0x1d/0x30 [18.871699] x64_sys_call+0x9e2/0x2670 [18.871979] do_syscall_64+0x87/0x150 [18.873280] ? do_syscall_64+0x93/0x150 [18.874742] ? lock_release+0x7b/0x160 [18.876157] ? do_user_addr_fault+0x5ce/0x8f0 [18.877833] ? irqentry_exit_to_user_mode+0xc2/0x210 [18.879608] ? irqentry_exit+0x77/0xb0 [18.879808] ? clear_bhb_loop+0x15/0x70 [18.880023] ? clear_bhb_loop+0x15/0x70 [18.880223] ? clear_bhb_loop+0x15/0x70 [18.880426] entry_SYSCALL_64_after_hwframe+0x76/0xe [18.880683] RIP: 0033:0x44a957 [18.880851] Code: ff ff e8 fc 00 00 00 66 2e 0f 1f 84 00 00 00 00 66 90 f3 0f 1e fa 64 8b 04 25 18 00 00 00 85 c0 75 10 b8 2e 00 00 00 0f 05 <48> 3d 00 f0 ff 77 51 c3 48 83 ec 28 89 54 24 1c 48 8974 24 10 [18.881766] RSP: 002b:00007ffcd00fad8 EFLAGS: 00000246 ORIG_RAX: 000000000000002e [18.882149] RAX: ffffffff00000000 RBX: 00007ffcd0010db8 RCX: 000000000044a957 [18.882507] RDX: 0000000000000000 RSI: 00007ffcd00fb70 RDI: 0000000000000003 [18.885037] RBP: 00007ffcd0010bc0 R08: 0000000000703c770 R09: 0000000000703c7c0 [18.887203] R10: 0000000000000080 R11: 0000000000000246 R12: 0000000000000001 [18.888026] R13: 00007ffcd0010da8 R14: 00000000004ca7d0 R15: 0000000000000001 [18.888395] </TASK> [18.888610] ---[end trace]---
CVE-2025-21693	In the Linux kernel, the following vulnerability has been resolved: mm: zswap: properly synchronize freeing resources during CPU hotunplug In zswap_compress() and zswap_decompress(), the per-CPU acomp_ctx of the current CPU at the beginning of the operation is retrieved and used throughout. However, since neither preemption nor migration are disabled, it is possible that the operation continues on a different CPU. If the original CPU is hotunplugged while the acomp_ctx is still in use, we run into a UAF bug as some of the resources attached to the acomp_ctx are freed during hotunplug in zswap_cpu_comp_dead() (i.e. acomp_ctx.buffer, acomp_ctx.req, or acomp_ctx.acomp). The problem was introduced in commit 1ec3b5fe6ec ("mm/zswap: move to use crypto_acomp API for hardware acceleration") when the switch to the crypto_acomp API was made. Prior to that, the per-CPU crypto_comp was retrieved using get_cpu_ptr() which disables preemption and makes sure the CPU cannot go away from under us. Preemption cannot be disabled with the crypto_acomp API as a sleepable context is needed. Use the acomp_ctx.mutex to synchronize CPU hotplug callbacks allocating and freeing resources with compression/decompression paths. Make sure that acomp_ctx.req is NULL when the resources are freed. In the compression/decompression paths, check if acomp_ctx.req is NULL after acquiring the mutex (meaning the CPU was offlined) and retry on the new CPU. The initialization of acomp_ctx.mutex is moved from the CPU hotplug callback to the pool initialization where it belongs (where the mutex is allocated). In addition to adding clarity, this makes sure that CPU hotplug cannot reinitialize a mutex that is already locked by compression/decompression. Previously a fix was attempted by holding cpus_read_lock() [1]. This would have caused a potential deadlock as it is possible for code already holding the lock to fall into reclaim and enter zswap (causing a deadlock). A fix was also attempted using SRCU for synchronization, but Johannes pointed out that synchronize_srcu() cannot be used in CPU hotplug notifiers [2]. Alternative fixes that were considered/attempted and could have worked: - Refcounting the per-CPU acomp_ctx. This involves complexity in handling the race between the refcount dropping to zero in zswap_[de]compress() and the refcount being re-initialized when the CPU is online. - Disabling migration before getting the per-CPU acomp_ctx [3], but that's discouraged and is a much bigger hammer than needed, and could result in subtle performance issues. [1]https://lkml.kernel.org/20241219212437.2714151-1-yosryahmed@google.com/ [2]https://lkml.kernel.org/20250107074724.1756696-2-yosryahmed@google.com/ [3]https://lkml.kernel.org/20250107222236.2715883-2-yosryahmed@google.com/ [yosryahmed@google.com: remove comment]

CVE ID	Description
CVE-2025-21694	In the Linux kernel, the following vulnerability has been resolved: fs/proc: fix softlockup in __read_vcore (part 2) Since commit 5cbcb62dddf5 ("fs/proc: fix softlockup in __read_vcore") the number of softlockups in __read_vcore at kdump time have gone down, but they still happen sometimes. In a memory constrained environment like the kdump image, a softlockup is not just a harmless message, but it can interfere with things like RCU freeing memory, causing the crashdump to get stuck. The second loop in __read_vcore has a lot more opportunities for natural sleep points, like scheduling out while waiting for a data write to happen, but apparently that is not always enough. Add a cond_resched() to the second loop in __read_vcore to (hopefully) get rid of the softlockups.
CVE-2025-21696	In the Linux kernel, the following vulnerability has been resolved: mm: clear uffd-wp PTE/PMD state on mremap() When mremap()ing a memory region previously registered with userfaultfd as write-protected but without UFFD_FEATURE_EVENT_REMAP, an inconsistency in flag clearing leads to a mismatch between the vma flags (which have uffd-wp cleared) and the pte/pmd flags (which do not have uffd-wp cleared). This mismatch causes a subsequent mprotect(PROT_WRITE) to trigger a warning in page_table_check_pte_flags() due to setting the pte to writable while uffd-wp is still set. Fix this by always explicitly clearing the uffd-wp pte/pmd flags on any such mremap() so that the values are consistent with the existing clearing of VM_UFFD_WP. Be careful to clear the logical flag regardless of its physical form; a PTE bit, a swap PTE bit, or a PTE marker. Cover PTE, huge PMD and hugetlb paths.
CVE-2025-21697	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Ensure job pointer is set to NULL after job completion After a job completes, the corresponding pointer in the device must be set to NULL. Failing to do so triggers a warning when unloading the driver, as it appears the job is still active. To prevent this, assign the job pointer to NULL after completing the job, indicating the job has finished.
CVE-2025-21699	In the Linux kernel, the following vulnerability has been resolved: gfs2: Truncate address space when flipping GFS2_DIF_JDATA flag Truncate an inode's address space when flipping the GFS2_DIF_JDATA flag: depending on that flag, the pages in the address space will either use buffer heads or iomap_folio_state structs, and we cannot mix the two.
CVE-2025-21700	In the Linux kernel, the following vulnerability has been resolved: net: sched: Disallow replacing of child qdisc from one parent to another Lion Ackermann was able to create a UAF which can be abused for privilege escalation with the following script Step 1. create root qdisc tc qdisc add dev lo root handle 1:0 drr step2. a class for packet aggregation do demonstrate uaf tc class add dev lo classid 1:1 drr step3. a class for nesting tc class add dev lo classid 1:2 drr step4. a class to graft qdisc to tc class add dev lo classid 1:3 drr step5. tc qdisc add dev lo parent 1:1 handle 2:0 plug limit 1024 step6. tc qdisc add dev lo parent 1:2 handle 3:0 drr step7. tc class add dev lo classid 3:1 drr step 8. tc qdisc add dev lo parent 3:1 handle 4:0 pfifo step 9. Display the class/qdisc layout tc class ls dev lo class drr 1:1 root leaf 2: quantum 64Kb class drr 1:2 root leaf 3: quantum 64Kb class drr 3:1 root leaf 4: quantum 64Kb tc qdisc ls qdisc drr 1: dev lo root refcnt 2 qdisc plug 2: dev lo parent 1:1 qdisc pfifo 4: dev lo parent 3:1 limit 1000p qdisc drr 3: dev lo parent 1:2 step10. trigger the bug <=== prevented by this patch tc qdisc replace dev lo parent 1:3 handle 4:0 step 11. Redisplay again the qdiscs/classes tc class ls dev lo class drr 1:1 root leaf 2: quantum 64Kb class drr 1:2 root leaf 3: quantum 64Kb class drr 1:3 root leaf 4: quantum 64Kb class drr 3:1 root leaf 4: quantum 64Kb tc qdisc ls qdisc drr 1: dev lo root refcnt 2 qdisc plug 2: dev lo parent 1:1 qdisc pfifo 4: dev lo parent 3:1 refcnt 2 limit 1000p qdisc drr 3: dev lo parent 1:2 Observe that a) parent for 4:0 does not change despite the replace request. There can only be one parent. b) refcount has gone up by two for 4:0 and c) both class 1:3 and 3:1 are pointing to it. Step 12. send one packet to plug echo "" socat -u STDIN UDP4-DATAGRAM:127.0.0.1:8888,priority=\$((0x10001)) step13. send one packet to the grafted fifo echo "" socat -u STDIN UDP4-DATAGRAM:127.0.0.1:8888,priority=\$((0x10003)) step14. lets trigger the uaf tc class delete dev lo classid 1:3 tc class delete dev lo classid 1:1 The semantics of "replace" is for a del/add _on the same node_ and not a delete from one node(3:1) and add to another node (1:3) as in step10. While we could "fix" with a more complex approach there could be consequences to expectations so the patch takes the preventive approach of "disallow such config". Joint work with Lion Ackermann <nnamrec@gmail.com>

CVE ID	Description
CVE-2025-21701	In the Linux kernel, the following vulnerability has been resolved: net: avoid race between device unregistration and ethnl ops The following trace can be seen if a device is being unregistered while its number of channels are being modified. DEBUG_LOCKS_WARN_ON(lock->magic != lock) WARNING: CPU: 3 PID: 3754 at kernel/locking/mutex.c:564 __mutex_lock+0xc8a/0x1120 CPU: 3 UID: 0 PID: 3754 Comm: ethtool Not tainted 6.13.0-rc6+ #771 RIP: 0010: __mutex_lock+0xc8a/0x1120 Call Trace: <TASK> ethtool_check_max_channel+0x1ea/0x880 ethnl_set_channels+0x3c3/0xb10 ethnl_default_set_doit+0x306/0x650 genl_family_rcv_msg_doit+0x1e3/0x2c0 genl_rcv_msg+0x432/0x6f0 netlink_rcv_skb+0x13d/0x3b0 genl_rcv+0x28/0x40 netlink_unicast+0x42e/0x720 netlink_sendmsg+0x765/0xc20 __sys_sendto+0x3ac/0x420 __x64_sys_sendto+0xe0/0x1c0 do_syscall_64+0x95/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e This is because unregister_netdevice_many_notify might run before the rtnl lock section of ethnl operations, eg. set_channels in the above example. In this example the rss lock would be destroyed by the device unregistration path before being used again, but in general running ethnl operations while dismantle has started is not a good idea. Fix this by denying any operation on devices being unregistered. A check was already there in ethnl_ops_begin, but not wide enough. Note that the same issue cannot be seen on the ioctl version (__dev_ethtool) because the device reference is retrieved from within the rtnl lock section there. Once dismantle started, the net device is unlisted and no reference will be found.
CVE-2025-21702	In the Linux kernel, the following vulnerability has been resolved: pfifo_tail_enqueue: Drop new packet when sch->limit == 0 Expected behaviour: In case we reach scheduler's limit, pfifo_tail_enqueue() will drop a packet in scheduler's queue and decrease scheduler's qlen by one. Then, pfifo_tail_enqueue() enqueue new packet and increase scheduler's qlen by one. Finally, pfifo_tail_enqueue() return `NET_XMIT_CN` status code. Weird behaviour: In case we set `sch->limit == 0` and trigger pfifo_tail_enqueue() on a scheduler that has no packet, the 'drop a packet' step will do nothing. This means the scheduler's qlen still has value equal 0. Then, we continue to enqueue new packet and increase scheduler's qlen by one. In summary, we can leverage pfifo_tail_enqueue() to increase qlen by one and return `NET_XMIT_CN` status code. The problem is: Let's say we have two qdiscs: Qdisc_A and Qdisc_B. - Qdisc_A's type must have `->graft()` function to create parent/child relationship. Let's say Qdisc_A's type is `hfsc`. Enqueue packet to this qdisc will trigger `hfsc_enqueue`. - Qdisc_B's type is pfifo_head_drop. Enqueue packet to this qdisc will trigger `pfifo_tail_enqueue`. - Qdisc_B is configured to have `sch->limit == 0`. - Qdisc_A is configured to route the enqueued's packet to Qdisc_B. Enqueue packet through Qdisc_A will lead to: - hfsc_enqueue(Qdisc_A) -> pfifo_tail_enqueue(Qdisc_B) - Qdisc_B->q.qlen += 1 - pfifo_tail_enqueue() return `NET_XMIT_CN` - hfsc_enqueue() check for `NET_XMIT_SUCCESS` and see `NET_XMIT_CN` => hfsc_enqueue() don't increase qlen of Qdisc_A. The whole process lead to a situation where Qdisc_A->q.qlen == 0 and Qdisc_B->q.qlen == 1. Replace 'hfsc' with other type (for example: 'drr') still lead to the same problem. This violate the design where parent's qlen should equal to the sum of its childrens'qlen. Bug impact: This issue can be used for user->kernel privilege escalation when it is reachable.
CVE-2025-21703	In the Linux kernel, the following vulnerability has been resolved: netem: Update sch->q.qlen before qdisc_tree_reduce_backlog() qdisc_tree_reduce_backlog() notifies parent qdisc only if child qdisc becomes empty, therefore we need to reduce the backlog of the child qdisc before calling it. Otherwise it would miss the opportunity to call cops->qlen_notify(), in the case of DRR, it resulted in UAF since DRR uses ->qlen_notify() to maintain its active list.
CVE-2025-21704	In the Linux kernel, the following vulnerability has been resolved: usb: cdc-acm: Check control transfer buffer size before access If the first fragment is shorter than struct usb_cdc_notification, we can't calculate an expected_size. Log an error and discard the notification instead of reading lengths from memory outside the received data, which can lead to memory corruption when the expected_size decreases between fragments, causing `expected_size - acm->nb_index` to wrap. This issue has been present since the beginning of git history; however, it only leads to memory corruption since commit ea2583529cd1 ("cdc-acm: reassemble fragmented notifications"). A mitigating factor is that acm_ctrl_irq() can only execute after userspace has opened /dev/ttyACM*; but if ModemManager is running, ModemManager will do that automatically depending on the USB device's vendor/product IDs and its other interfaces.

CVE ID	Description
CVE-2025-21705	In the Linux kernel, the following vulnerability has been resolved: mptcp: handle fastopen disconnect correctly Syzbot was able to trigger a data stream corruption: WARNING: CPU: 0 PID: 9846 at net/mptcp/protocol.c:1024 __mptcp_clean_una+0xddb/0xff0 net/mptcp/protocol.c:1024 Modules linked in: CPU: 0 UID: 0 PID: 9846 Comm: syz-executor351 Not tainted 6.13.0-rc2-syzkaller-00059-g00a5acdbf398 #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 11/25/2024 RIP: 0010: __mptcp_clean_una+0xddb/0xff0 net/mptcp/protocol.c:1024 Code: fa ff ff 48 8b 4c 24 18 80 e1 07 fe c1 38 c1 0f 8c 8e fa ff ff 48 8b 7c 24 18 e8 e0 db 54 f6 e9 7f fa ff ff e8 e6 80 ee f5 90 <0f> 0b 90 4c 8b 6c 24 40 4d 89 f4 e9 04 f5 ff ff 44 89 f1 80 e1 07 RSP: 0018:ffffc9000c0cf400 EFLAGS: 00010293 RAX: ffffffff8bb0dd5a RBX: ffff88803f5d230 RCX: ffff888059ce8000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ffff9000c0cf518 R08: ffffffff8bb0d1dd R09: 1ffff10170c8928 R10: dffffc0000000000 R11: ffffed10170c8929 R12: 0000000000000000 R13: ffff88803f5d220 R14: dffffc0000000000 R15: ffff8880592b8000 FS: 00007f6e866496c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f6e86f491a0 CR3: 00000000310e6000 CR4: 0000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> __mptcp_clean_una_wakeup+0x7f/0x2d0 net/mptcp/protocol.c:1074 mptcp_release_cb+0x7cb/0xb30 net/mptcp/protocol.c:3493 release_sock+0x1aa/0x1f0 net/core/socket.c:3640 inet_wait_for_connect net/ipv4/af_inet.c:609 [inline] __inet_stream_connect+0x8bd/0xf30 net/ipv4/af_inet.c:703 mptcp_sendmsg_fastopen+0x2a2/0x530 net/mptcp/protocol.c:1755 mptcp_sendmsg+0x1884/0x1b10 net/mptcp/protocol.c:1830 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg +0x1a6/0x270 net/socket.c:726 ____sys_sendmsg+0x52a/0x7e0 net/socket.c:2583 __sys_sendmsg net/socket.c:2637 [inline] __sys_sendmsg+0x269/0x350 net/socket.c:2669 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f6e86ebfe69 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 b1 1f 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff f7 d8 64 89 01 48 RSP: 002b:00007f6e86649168 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffff8bffffda RBX: 00007f6e86f491b8 RCX: 00007f6e86ebfe69 RDX: 0000000030004001 RSI: 0000000020000080 RDI: 0000000000000003 RBP: 00007f6e86f491b0 R08: 00007f6e866496c0 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 00007f6e86f491bc R13: 0000000000000006 R14: 00007ffe445d9420 R15: 00007ffe445d9508 </TASK> The root cause is the bad handling of disconnect() generated internally by the MPTCP protocol in case of connect FASTOPEN errors. Address the issue increasing the socket disconnect counter even on such a case, to allow other threads waiting on the same socket lock to properly error out.

CVE ID	Description
CVE-2025-21706	In the Linux kernel, the following vulnerability has been resolved: mptcp: pm: only set fullmesh for subflow endp With the in-kernel path-manager, it is possible to change the 'fullmesh' flag. The code in mptcp_pm_nl_fullmesh() expects to change it only on 'subflow' endpoints, to recreate more or less subflows using the linked address. Unfortunately, the set_flags() hook was a bit more permissive, and allowed 'implicit' endpoints to get the 'fullmesh' flag while it is not allowed before. That's what syzbot found, triggering the following warning: WARNING: CPU: 0 PID: 6499 at net/mptcp/pm_netlink.c:1496 __mark_subflow_endp_available net/mptcp/pm_netlink.c:1496 [inline] WARNING: CPU: 0 PID: 6499 at net/mptcp/pm_netlink.c:1496 mptcp_pm_nl_fullmesh net/mptcp/pm_netlink.c:1980 [inline] WARNING: CPU: 0 PID: 6499 at net/mptcp/pm_netlink.c:1496 mptcp_nl_set_flags net/mptcp/pm_netlink.c:2003 [inline] WARNING: CPU: 0 PID: 6499 at net/mptcp/pm_netlink.c:1496 mptcp_pm_nl_set_flags+0x974/0xdc0 net/mptcp/pm_netlink.c:2064 Modules linked in: CPU: 0 UID: 0 PID: 6499 Comm: syz.1.413 Not tainted 6.13.0-rc5-syzkaller-00172-gd1bf27c4e176 #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 RIP: 0010: __mark_subflow_endp_available net/mptcp/pm_netlink.c:1496 [inline] RIP: 0010:mptcp_pm_nl_fullmesh net/mptcp/pm_netlink.c:1980 [inline] RIP: 0010:mptcp_nl_set_flags net/mptcp/pm_netlink.c:2003 [inline] RIP: 0010:mptcp_pm_nl_set_flags+0x974/0xdc0 net/mptcp/pm_netlink.c:2064 Code: 01 00 00 49 89 c5 e8 fb 45 e8 f5 e9 b8 fc ff ff e8 f1 45 e8 f5 4c 89 f7 be 03 00 00 00 e8 44 1d 0b f9 eb a0 e8 dd 45 e8 f5 90 <0f> 0b 90 e9 17 ff ff ff 89 d9 80 e1 07 38 c1 0f 8c c9 fc ff ff 48 RSP: 0018:ffffc9000d307240 EFLAGS: 00010293 RAX: ffffffff8bb72e03 RBX: 0000000000000000 RCX: ffff88807da88000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ffff88807da88000 R08: ffffffff8bb72cf0 R09: 1ffff1100b842a5e R10: dffffc0000000000 R11: ffffd100b842a5f R12: ffff88801e2e5ac0 R13: ffff88805c214800 R14: ffff88805c2152e8 R15: 1ffff1100b842a5d FS: 00005555619f6500(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020002840 CR3: 00000000247e6000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff00ff DR7: 0000000000000040 Call Trace: <TASK> genl_family_rcv_msg_doit net/netlink/genetlink.c:1115 [inline] genl_family_rcv_msg net/netlink/genetlink.c:1195 [inline] genl_rcv_msg+0xb14/0xec0 net/netlink/genetlink.c:1210 netlink_rcv_skb+0x1e3/0x430 net/netlink/af_netlink.c:2542 genl_rcv+0x28/0x40 net/netlink/genetlink.c:1219 netlink_unicast_kernel net/netlink/af_netlink.c:1321 [inline] netlink_unicast+0x7f6/0x990 net/netlink/af_netlink.c:1347 netlink_sendmsg+0x8e4/0xcb0 net/netlink/af_netlink.c:1891 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:726 ____sys_sendmsg+0x52a/0x7e0 net/socket.c:2583 __sys_sendmsg net/socket.c:2637 [inline] __sys_sendmsg+0x269/0x350 net/socket.c:2669 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f5fe8785d29 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fff571f5558 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffff8bda RBX: 00007f5fe8975fa0 RCX: 00007f5fe8785d29 RDX: 0000000000000000 RSI: 0000000020000480 RDI: 0000000000000007 RBP: 00007f5fe8801b08 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007f5fe8975fa0 R14: 00007f5fe8975fa0 R15: 000000 ---truncated---

CVE ID	Description
CVE-2025-21707	In the Linux kernel, the following vulnerability has been resolved: mptcp: consolidate suboption status MPTCP maintains the received sub-options status is the bitmask carrying the received suboptions and in several bitfields carrying per suboption additional info. Zeroing the bitmask before parsing is not enough to ensure a consistent status, and the MPTCP code has to additionally clear some bitfield depending on the actually parsed suboption. The above schema is fragile, and syzbot managed to trigger a path where a relevant bitfield is not cleared/initialized: BUG: KMSAN: uninit-value in __mptcp_expand_seq net/mptcp/options.c:1030 [inline] BUG: KMSAN: uninit-value in mptcp_expand_seq net/mptcp/protocol.h:864 [inline] BUG: KMSAN: uninit-value in ack_update_msk net/mptcp/options.c:1060 [inline] BUG: KMSAN: uninit-value in mptcp_incoming_options+0x2036/0x3d30 net/mptcp/options.c:1209 __mptcp_expand_seq net/mptcp/options.c:1030 [inline] mptcp_expand_seq net/mptcp/protocol.h:864 [inline] ack_update_msk net/mptcp/options.c:1060 [inline] mptcp_incoming_options+0x2036/0x3d30 net/mptcp/options.c:1209 tcp_data_queue+0xb4/0x7be0 net/ipv4/tcp_input.c:5233 tcp_rcv_established+0x1061/0x2510 net/ipv4/tcp_input.c:6264 tcp_v4_do_rcv+0x7f3/0x11a0 net/ipv4/tcp_ipv4.c:1916 tcp_v4_rcv+0x51df/0x5750 net/ipv4/tcp_ipv4.c:2351 ip_protocol_deliver_rcu+0x2a3/0x13d0 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x336/0x500 net/ipv4/ip_input.c:233 NF_HOOK include/linux/netfilter.h:314 [inline] ip_local_deliver+0x21f/0x490 net/ipv4/ip_input.c:254 dst_input include/net/dst.h:460 [inline] ip_rcv_finish+0x4a2/0x520 net/ipv4/ip_input.c:447 NF_HOOK include/linux/netfilter.h:314 [inline] ip_rcv+0xcd/0x380 net/ipv4/ip_input.c:567 __netif_receive_skb_one_core net/core/dev.c:5704 [inline] __netif_receive_skb+0x319/0xa00 net/core/dev.c:5817 process_backlog+0x4ad/0xa50 net/core/dev.c:6149 __napi_poll+0xe7/0x980 net/core/dev.c:6902 napi_poll net/core/dev.c:6971 [inline] net_rx_action+0xa5a/0x19b0 net/core/dev.c:7093 handle_softirqs+0x1a0/0x7c0 kernel/softirq.c:561 __do_softirq+0x14/0x1a kernel/softirq.c:595 do_softirq+0x9a/0x100 kernel/softirq.c:462 __local_bh_enable_ip+0x9f/0xb0 kernel/softirq.c:389 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:919 [inline] __dev_queue_xmit+0x2758/0x57d0 net/core/dev.c:4493 dev_queue_xmit include/linux/netdevice.h:3168 [inline] neigh_hh_output include/net/neighbour.h:523 [inline] neigh_output include/net/neighbour.h:537 [inline] ip_finish_output2+0x187c/0x1b70 net/ipv4/ip_output.c:236 __ip_finish_output+0x287/0x810 ip_finish_output+0x4b/0x600 net/ipv4/ip_output.c:324 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip_output+0x15f/0x3f0 net/ipv4/ip_output.c:434 dst_output include/net/dst.h:450 [inline] ip_local_out net/ipv4/ip_output.c:130 [inline] __ip_queue_xmit+0x1f2a/0x20d0 net/ipv4/ip_output.c:536 ip_queue_xmit+0x60/0x80 net/ipv4/ip_output.c:550 __tcp_transmit_skb+0x3cea/0x4900 net/ipv4/tcp_output.c:1468 tcp_transmit_skb net/ipv4/tcp_output.c:1486 [inline] tcp_write_xmit+0x3b90/0x9070 net/ipv4/tcp_output.c:2829 __tcp_push_pending_frames+0xc4/0x380 net/ipv4/tcp_output.c:3012 tcp_send_fin+0x9f6/0xf50 net/ipv4/tcp_output.c:3618 __tcp_close+0x140c/0x1550 net/ipv4/tcp.c:3130 __mptcp_close_ssk+0x74e/0x16f0 net/mptcp/protocol.c:2496 mptcp_close_ssk+0x26b/0x2c0 net/mptcp/protocol.c:2550 mptcp_pm_nl_rm_addr_or_subflow+0x635/0xd10 net/mptcp/pm_netlink.c:889 mptcp_pm_nl_rm_subflow_received net/mptcp/pm_netlink.c:924 [inline] mptcp_pm_flush_addrs_and_subflows net/mptcp/pm_netlink.c:1688 [inline] mptcp_nl_flush_addrs_list net/mptcp/pm_netlink.c:1709 [inline] mptcp_pm_nl_flush_addrs_doit+0xe10/0x1630 net/mptcp/pm_netlink.c:1750 genl_family_rcv_msg_doit net/netlink/genetlink.c:1115 [inline] ---truncated---

CVE ID	Description
CVE-2025-21708	In the Linux kernel, the following vulnerability has been resolved: net: usb: rtl8150: enable basic endpoint checking Syzkaller reports [1] encountering a common issue of utilizing a wrong usb endpoint type during URB submitting stage. This, in turn, triggers a warning shown below. For now, enable simple endpoint checking (specifically, bulk and interrupt eps, testing control one is not essential) to mitigate the issue with a view to do other related cosmetic changes later, if they are necessary. [1] Syzkaller report: usb 1-1: BOGUS urb xfer, pipe 3 != type 1 WARNING: CPU: 1 PID: 2586 at drivers/usb/core/urb.c:503 usb_submit_urb+0xe4b/0x1730 driv> Modules linked in: CPU: 1 UID: 0 PID: 2586 Comm: dhcpcd Not tainted 6.11.0-rc4-syzkaller-00069-gfc88bb11617> Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 08/06/2024 RIP: 0010:usb_submit_urb+0xe4b/0x1730 drivers/usb/core/urb.c:503 Code: 84 3c 02 00 00 e8 05 e4 fc fc 4c 89 ef e8 fd 25 d7 fe 45 89 e0 89 e9 4c 89 f2 48 8> RSP: 0018:ffffc9000441f740 EFLAGS: 00010282 RAX: 0000000000000000 RBX: ffff888112487a00 RCX: ffffffff811a99a9 RDX: ffff88810df6ba80 RSI: ffffffff811a99b6 RDI: 0000000000000001 RBP: 0000000000000003 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000001 R12: 0000000000000001 R13: ffff8881023bf0a8 R14: ffff888112452a20 R15: ffff888112487a7c FS: 00007fc04eea5740(0000) GS:ffff8881f6300000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f0a1de9f870 CR3: 000000010dbd0000 CR4: 00000000003506f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> rtl8150_open+0x300/0xe30 drivers/net/usb/rtl8150.c:733 __dev_open+0x2d4/0x4e0 net/core/dev.c:1474 __dev_change_flags+0x561/0x720 net/core/dev.c:8838 dev_change_flags+0x8f/0x160 net/core/dev.c:8910 devinet_ioctl+0x127a/0x1f10 net/ipv4/devinet.c:1177 inet_ioctl+0x3aa/0x3f0 net/ipv4/af_inet.c:1003 sock_do_ioctl+0x116/0x280 net/socket.c:1222 sock_ioctl+0x22e/0x6c0 net/socket.c:1341 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __x64_sys_ioctl+0x193/0x220 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fc04ef73d49 ... This change has not been tested on real hardware.
CVE-2025-21709	In the Linux kernel, the following vulnerability has been resolved: kernel: be more careful about dup_mmap() failures and uprobe registering If a memory allocation fails during dup_mmap(), the maple tree can be left in an unsafe state for other iterators besides the exit path. All the locks are dropped before the exit_mmap() call (in mm/mmap.c), but the incomplete mm_struct can be reached through (at least) the rmap finding the vmas which have a pointer back to the mm_struct. Up to this point, there have been no issues with being able to find an mm_struct that was only partially initialised. Syzbot was able to make the incomplete mm_struct fail with recent forking changes, so it has been proven unsafe to use the mm_struct that hasn't been initialised, as referenced in the link below. Although 8ac662f5da19f ("fork: avoid inappropriate uprobe access to invalid mm") fixed the uprobe access, it does not completely remove the race. This patch sets the MMF_OOM_SKIP to avoid the iteration of the vmas on the oom side (even though this is extremely unlikely to be selected as an oom victim in the race window), and sets MMF_UNSTABLE to avoid other potential users from using a partially initialised mm_struct. When registering vmas for uprobe, skip the vmas in an mm that is marked unstable. Modifying a vma in an unstable mm may cause issues if the mm isn't fully initialised.

CVE ID	Description
CVE-2025-21710	In the Linux kernel, the following vulnerability has been resolved: tcp: correct handling of extreme memory squeeze Testing with iperf3 using the "pasta" protocol splicer has revealed a problem in the way tcp handles window advertising in extreme memory squeeze situations. Under memory pressure, a socket endpoint may temporarily advertise a zero-sized window, but this is not stored as part of the socket data. The reasoning behind this is that it is considered a temporary setting which shouldn't influence any further calculations. However, if we happen to stall at an unfortunate value of the current window size, the algorithm selecting a new value will consistently fail to advertise a non-zero window once we have freed up enough memory. This means that this side's notion of the current window size is different from the one last advertised to the peer, causing the latter to not send any data to resolve the situation. The problem occurs on the iperf3 server side, and the socket in question is a completely regular socket with the default settings for the fedora40 kernel. We do not use SO_PEEK or SO_RCVBUF on the socket. The following excerpt of a logging session, with own comments added, shows more in detail what is happening: // tcp_v4_rcv(->) // tcp_rcv_established(->) [5201<->39222]: ===== Activating log @ net/ipv4/tcp_input.c/tcp_data_queue()/5257 ===== [5201<->39222]: tcp_data_queue(->) [5201<->39222]: DROPPING skb [265600160..265665640], reason: SKB_DROP_REASON_PROTO_MEM [rcv_nxt 265600160, rcv_wnd 262144, snt_ack 265469200, win_now 131184] [copied_seq 259909392->260034360 (124968), unread 5565800, qlen 85, ofoq 0] [OFO queue: gap: 65480, len: 0] [5201<->39222]: tcp_data_queue(->) [5201<->39222]: __tcp_transmit_skb(->) [tp->rcv_wup: 265469200, tp->rcv_wnd: 262144, tp->rcv_nxt 265600160] [5201<->39222]: tcp_select_window(->) [5201<->39222]: (inet_csk(sk)->icsk_ack.pending & ICSK_ACK_NOMEM) ? --> TRUE [tp->rcv_wup: 265469200, tp->rcv_wnd: 262144, tp->rcv_nxt 265600160] returning 0 [5201<->39222]: tcp_select_window(->) [5201<->39222]: ADVERTISING WIN 0, ACK_SEQ: 265600160 [5201<->39222]: __tcp_transmit_skb(->) [5201<->39222]: tcp_rcv_established(->) [5201<->39222]: tcp_v4_rcv(->) // Receive queue is at 85 buffers and we are out of memory. // We drop the incoming buffer, although it is in sequence, and decide // to send an advertisement with a window of zero. // We don't update tp->rcv_wnd and tp->rcv_wup accordingly, which means // we unconditionally shrink the window. [5201<->39222]: tcp_recvmmsg_locked(->) [5201<->39222]: __tcp_cleanup_rbuf(->) tp->rcv_wup: 265469200, tp->rcv_wnd: 262144, tp->rcv_nxt 265600160 [5201<->39222]: [new_win = 0, win_now = 131184, 2 * win_now = 262368] [5201<->39222]: [new_win >= (2 * win_now) ? --> time_to_ack = 0] [5201<->39222]: NOT calling tcp_send_ack() [tp->rcv_wup: 265469200, tp->rcv_wnd: 262144, tp->rcv_nxt 265600160] [5201<->39222]: __tcp_cleanup_rbuf(->) [rcv_nxt 265600160, rcv_wnd 262144, snt_ack 265469200, win_now 131184] [copied_seq 260040464->260040464 (0), unread 5559696, qlen 85, ofoq 0] returning 6104 bytes [5201<->39222]: tcp_recvmmsg_locked(->) // After each read, the algorithm for calculating the new receive // window in __tcp_cleanup_rbuf() finds it is too small to advertise // or to update tp->rcv_wnd. // Meanwhile, the peer thinks the window is zero, and will not send // any more data to trigger an update from the interrupt mode side. [5201<->39222]: tcp_recvmmsg_locked(->) [5201<->39222]: __tcp_cleanup_rbuf(->) tp->rcv_wup: 265469200, tp->rcv_wnd: 262144, tp->rcv_nxt 265600160 [5201<->39222]: [new_win = 262144, win_now = 131184, 2 * win_n ---truncated---
CVE-2025-21711	In the Linux kernel, the following vulnerability has been resolved: net/rose: prevent integer overflows in rose_setsockopt() In case of possible unpredictably large arguments passed to rose_setsockopt() and multiplied by extra values on top of that, integer overflows may occur. Do the safest minimum and fix these issues by checking the contents of 'opt' and returning -EINVAL if they are too large. Also, switch to unsigned int and remove useless check for negative 'opt' in ROSE_IDLE case.
CVE-2025-21712	In the Linux kernel, the following vulnerability has been resolved: md/md-bitmap: Synchronize bitmap_get_stats() with bitmap lifetime After commit ec6bb299c7c3 ("md/md-bitmap: add 'sync_size' into struct md_bitmap_stats"), following panic is reported: Oops: general protection fault, probably for non-canonical address RIP: 0010:bitmap_get_stats+0x2b/0xa0 Call Trace: <TASK> md_seq_show+0x2d2/0x5b0 seq_read_iter+0x2b9/0x470 seq_read+0x12f/0x180 proc_reg_read+0x57/0xb0 vfs_read+0xf6/0x380 ksys_read+0x6c/0xf0 do_syscall_64+0x82/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e Root cause is that bitmap_get_stats() can be called at anytime if mddev is still there, even if bitmap is destroyed, or not fully initialized. Deferencing bitmap in this case can crash the kernel. Meanwhile, the above commit start to deferencing bitmap->storage, make the problem easier to trigger. Fix the problem by protecting bitmap_get_stats() with bitmap_info.mutex.

CVE ID	Description
CVE-2025-21714	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix implicit ODP use after free Prevent double queueing of implicit ODP mr destroy work by using __xa_cmpxchg() to make sure this is the only time we are destroying this specific mr. Without this change, we could try to invalidate this mr twice, which in turn could result in queueing a MR work destroy twice, and eventually the second work could execute after the MR was freed due to the first work, causing a user after free and trace below. refcount_t: underflow; use-after-free. WARNING: CPU: 2 PID: 12178 at lib/refcount.c:28 refcount_warn_saturate+0x12b/0x130 Modules linked in: bonding ib_ipoib vfio_pci ip_gre geneve nf_tables ip6_gre gre ip6_tunnel tunnel6 ipip tunnel4 ib_umad rdma_ucm mlx5_vfio_pci vfio_pci_core vfio_iommu_type1 mlx5_ib vfio ib_uverbs mlx5_core iptable_raw openvswitch nsh rpcrdma ib_iser libiscsi scsi_transport_iscsi rdma_cm iw_cm ib_cm ib_core xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter rpcsec_gss_krb5 auth_rpcgss oid_registry overlay zram zsmalloc fuse [last unloaded: ib_uverbs] CPU: 2 PID: 12178 Comm: kworker/u20:5 Not tainted 6.5.0-rc1_net_next_mlx5_58c644e #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 Workqueue: events_unbound free_implicit_child_mr_work [mlx5_ib] RIP: 0010:refcount_warn_saturate+0x12b/0x130 Code: 48 c7 c7 38 95 2a 82 c6 05 bc c6 fe 00 01 e8 0c 66 aa ff 0f 0b 5b c3 48 c7 c7 e0 94 2a 82 c6 05 a7 c6 fe 00 01 e8 f5 65 aa ff <0f> 0b 5b c3 90 8b 07 3d 00 00 00 c0 74 12 83 f8 01 74 13 8d 50 ff RSP: 0018:ffff8881008e3e40 EFLAGS: 00010286 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000027 RDX: ffff88852c91b5c8 RSI: 0000000000000001 RDI: ffff88852c91b5c0 RBP: ffff8881dacd4e00 R08: 00000000ffffffff R09: 0000000000000019 R10: 0000000000000072e R11: 00000000063666572 R12: ffff88812bfd9e00 R13: ffff8881c792d200 R14: ffff88810011c005 R15: ffff8881002099c0 FS: 0000000000000000(0000) GS:ffff88852c900000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 00000000080050033 CR2: 00007f5694b5e000 CR3: 00000001153f6003 CR4: 0000000000370ea0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: <TASK> ? refcount_warn_saturate+0x12b/0x130 free_implicit_child_mr_work+0x180/0x1b0 [mlx5_ib] process_one_work+0x1cc/0x3c0 worker_thread+0x218/0x3c0 kthread+0xc6/0xf0 ret_from_fork+0x1f/0x30 </TASK>
CVE-2025-21715	In the Linux kernel, the following vulnerability has been resolved: net: davicom: fix UAF in dm9000_drv_remove dm is netdev private data and it cannot be used after free_netdev() call. Using dm after free_netdev() can cause UAF bug. Fix it by moving free_netdev() at the end of the function. This is similar to the issue fixed in commit ad297cd2db89 ("net: qcom/emac: fix UAF in emac_remove"). This bug is detected by our static analysis tool.
CVE-2025-21716	In the Linux kernel, the following vulnerability has been resolved: vxlan: Fix uninit-value in vxlan_vnifilter_dump() KMSAN reported an uninit-value access in vxlan_vnifilter_dump() [1]. If the length of the netlink message payload is less than sizeof(struct tunnel_msg), vxlan_vnifilter_dump() accesses bytes beyond the message. This can lead to uninit-value access. Fix this by returning an error in such situations. [1] BUG: KMSAN: uninit-value in vxlan_vnifilter_dump+0x328/0x920 drivers/net/vxlan/vxlan_vnifilter.c:422 vxlan_vnifilter_dump+0x328/0x920 drivers/net/vxlan/vxlan_vnifilter.c:422 rtnl_dumpit+0xd5/0x2f0 net/core/rtnetlink.c:6786 netlink_dump+0x93e/0x15f0 net/netlink/af_netlink.c:2317 __netlink_dump_start+0x716/0xd60 net/netlink/af_netlink.c:2432 netlink_dump_start include/linux/netlink.h:340 [inline] rtnetlink_dump_start net/core/rtnetlink.c:6815 [inline] rtnetlink_rcv_msg+0x1256/0x14a0 net/core/rtnetlink.c:6882 netlink_rcv_skb+0x467/0x660 net/netlink/af_netlink.c:2542 rtnetlink_rcv+0x35/0x40 net/core/rtnetlink.c:6944 netlink_unicast_kernel net/netlink/af_netlink.c:1321 [inline] netlink_unicast+0xed6/0x1290 net/netlink/af_netlink.c:1347 netlink_sendmsg+0x1092/0x1230 net/netlink/af_netlink.c:1891 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x330/0x3d0 net/socket.c:726 __sys_sendmsg+0x7f4/0xb50 net/socket.c:2583 __sys_sendmsg+0x271/0x3b0 net/socket.c:2637 __sys_sendmsg net/socket.c:2669 [inline] __do_sys_sendmsg net/socket.c:2674 [inline] __se_sys_sendmsg net/socket.c:2672 [inline] __x64_sys_sendmsg+0x211/0x3e0 net/socket.c:2672 x64_sys_call+0x3878/0x3d90 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xd9/0x1d0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slub.c:4110 [inline] slab_alloc_node mm/slub.c:4153 [inline] kmem_cache_alloc_node_noprof+0x800/0xe80 mm/slub.c:4205 kmemalloc_reserve+0x13b/0x4b0 net/core/skbuff.c:587 __alloc_skb+0x347/0x7d0 net/core/skbuff.c:678 alloc_skb include/linux/skbuff.h:1323 [inline] netlink_alloc_large_skb+0xa5/0x280 net/netlink/af_netlink.c:1196 netlink_sendmsg+0xac9/0x1230 net/netlink/af_netlink.c:1866 sock_sendmsg_nosec net/socket.c:711 [inline] __sock_sendmsg+0x330/0x3d0 net/socket.c:726 __sys_sendmsg+0x7f4/0xb50 net/socket.c:2583 __sys_sendmsg+0x271/0x3b0 net/socket.c:2637 __sys_sendmsg net/socket.c:2669 [inline] __do_sys_sendmsg net/socket.c:2674 [inline] __se_sys_sendmsg net/socket.c:2672 [inline] __x64_sys_sendmsg+0x211/0x3e0 net/socket.c:2672 x64_sys_call+0x3878/0x3d90 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xd9/0x1d0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f CPU: 0 UID: 0 PID: 30991 Comm: syz.4.10630 Not tainted 6.12.0-10694-gc44daa7e3c73 #29 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014

CVE ID	Description
CVE-2025-21718	In the Linux kernel, the following vulnerability has been resolved: net: rose: fix timer races against user threads Rose timers only acquire the socket spinlock, without checking if the socket is owned by one user thread. Add a check and rearm the timers if needed. BUG: KASAN: slab-use-after-free in rose_timer_expiry+0x31d/0x360 net/rose/rose_timer.c:174 Read of size 2 at addr ffff88802f09b82a by task swapper/0/0 CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.13.0-rc5-syzkaller-00172-gd1bf27c4e176 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <IRQ> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report+0x143/0x180 mm/kasan/report.c:602 rose_timer_expiry+0x31d/0x360 net/rose/rose_timer.c:174 call_timer_fn+0x187/0x650 kernel/time/timer.c:1793 expire_timers kernel/time/timer.c:1844 [inline] __run_timers kernel/time/timer.c:2418 [inline] __run_timer_base+0x66a/0x8e0 kernel/time/timer.c:2430 run_timer_base kernel/time/timer.c:2439 [inline] run_timer_softirq+0xb7/0x170 kernel/time/timer.c:2449 handle_softirqs+0x2d4/0x9b0 kernel/softirq.c:561 __do_softirq kernel/softirq.c:595 [inline] invoke_softirq kernel/softirq.c:435 [inline] __irq_exit_rcu+0xf7/0x220 kernel/softirq.c:662 irq_exit_rcu+0x9/0x30 kernel/softirq.c:678 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1049 [inline] sysvec_apic_timer_interrupt+0xa6/0xc0 arch/x86/kernel/apic/apic.c:1049 </IRQ>
CVE-2025-21719	In the Linux kernel, the following vulnerability has been resolved: ipmr: do not call mr_mfc_uses_dev() for unres entries syzbot found that calling mr_mfc_uses_dev() for unres entries would crash [1], because c->mfc_un.res.minvif / c->mfc_un.res.maxvif alias to "struct sk_buff_head unresolved", which contain two pointers. This code never worked, lets remove it. [1] Unable to handle kernel paging request at virtual address ffff5fff2d536613 KASAN: maybe wild-memory-access in range [0xfffffff96a9b3098-0xfffffff96a9b309f] Modules linked in: CPU: 1 UID: 0 PID: 7321 Comm: syz.0.16 Not tainted 6.13.0-rc7-syzkaller-g1950a0af2d55 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : mr_mfc_uses_dev net/ipv4/ipmr_base.c:290 [inline] pc : mr_table_dump+0x5a4/0x8b0 net/ipv4/ipmr_base.c:334 lr : mr_mfc_uses_dev net/ipv4/ipmr_base.c:289 [inline] lr : mr_table_dump+0x694/0x8b0 net/ipv4/ipmr_base.c:334 Call trace: mr_mfc_uses_dev net/ipv4/ipmr_base.c:290 [inline] (P) mr_table_dump+0x5a4/0x8b0 net/ipv4/ipmr_base.c:334 (P) mr_rtm_dumproute+0x254/0x454 net/ipv4/ipmr_base.c:382 ipmr_rtm_dumproute+0x248/0x4b4 net/ipv4/ipmr.c:2648 rtnl_dump_all+0x2e4/0x4e8 net/core/rtnetlink.c:4327 rtnl_dumpit+0x98/0x1d0 net/core/rtnetlink.c:6791 netlink_dump+0x4f0/0xabc0 net/netlink/af_netlink.c:2317 netlink_recvmmsg+0x56c/0xe64 net/netlink/af_netlink.c:1973 sock_recvmmsg_nosec net/socket.c:1033 [inline] sock_recvmmsg net/socket.c:1055 [inline] sock_read_iter+0x2d8/0x40c net/socket.c:1125 new_sync_read fs/read_write.c:484 [inline] vfs_read+0x740/0x970 fs/read_write.c:565 ksys_read+0x15c/0x26c fs/read_write.c:708

CVE ID	Description
CVE-2025-21720	In the Linux kernel, the following vulnerability has been resolved: xfrm: delete intermediate secpath entry in packet offload mode Packets handled by hardware have added secpath as a way to inform XFRM core code that this path was already handled. That secpath is not needed at all after policy is checked and it is removed later in the stack. However, in the case of IP forwarding is enabled (/proc/sys/net/ipv4/ip_forward), that secpath is not removed and packets which already were handled are reentered to the driver TX path with xfrm_offload set. The following kernel panic is observed in mlx5 in such case: mlx5_core 0000:04:00.0 enp4s0f0np0: Link up mlx5_core 0000:04:00.1 enp4s0f1np1: Link up Initializing XFRM netlink socket IPsec XFRM device driver BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor instruction fetch in kernel mode #PF: error_code(0x0010) - not-present page PGD 0 P4D 0 Oops: Oops: 0010 [#1] PREEMPT SMP CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.13.0-rc1-alex #3 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.13.0-1ubuntu1.1 04/01/2014 RIP: 0010:0x0 Code: Unable to access opcode bytes at 0xffffffffffffd6. RSP: 0018:ffff8b7380003800 EFLAGS: 00010206 RAX: ffff8df004e02600 RBX: ffff8b73800038d8 RCX: 00000000ffff98cf RDX: ffff8df00733e108 RSI: ffff8df00521fb80 RDI: ffff8df001661f00 RBP: ffff8b7380003850 R08: ffff8df013980000 R09: 0000000000000010 R10: 0000000000000002 R11: 0000000000000002 R12: ffff8df001661f00 R13: ffff8df00521fb80 R14: ffff8df00733e108 R15: ffff8df011faf04e FS: 0000000000000000(0000) GS:ffff8df46b800000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffffdf6 CR3: 0000000106384000 CR4: 0000000000350ef0 Call Trace: <IRQ> ? show_regs+0x63/0x70 ? __die_body+0x20/0x60 ? __die+0x2b/0x40 ? page_fault_oops+0x15c/0x550 ? do_user_addr_fault+0x3ed/0x870 ? exc_page_fault+0x7f/0x190 ? asm_exc_page_fault+0x27/0x30 mlx5e_ipsec_handle_tx_skb+0xe7/0x2f0 [mlx5_core] mlx5e_xmit+0x58e/0x1980 [mlx5_core] ? __fib_lookup+0x6a/0xb0 dev_hard_start_xmit+0x82/0x1d0 sch_direct_xmit+0xfe/0x390 __dev_queue_xmit+0x6d8/0xee0 ? __fib_lookup+0x6a/0xb0 ? internal_add_timer+0x48/0x70 ? mod_timer+0xe2/0x2b0 neigh_resolve_output+0x115/0x1b0 __neigh_update+0x26a/0xc50 neigh_update+0x14/0x20 arp_process+0x2cb/0x8e0 ? __napi_build_skb+0x5e/0x70 arp_rcv+0x11e/0x1c0 ? devgro_receive+0x574/0x820 __netif_receive_skb_list_core+0x1cf/0x1f0 netif_receive_skb_list_internal+0x183/0x2a0 napi_complete_done+0x76/0x1c0 mlx5e_napi_poll+0x234/0x7a0 [mlx5_core] __napi_poll+0x2d/0x1f0 net_rx_action+0x1a6/0x370 ? atomic_notifier_call_chain+0x3b/0x50 ? irq_int_handler+0x15/0x20 [mlx5_core] handle_softirqs+0xb9/0x2f0 ? handle_irq_event+0x44/0x60 irq_exit_rcu+0xdb/0x100 common_interrupt+0x98/0xc0 </IRQ> <TASK> asm_common_interrupt+0x27/0x40 RIP: 0010:pv_native_safe_halt+0xb/0x10 Code: 09 c3 66 66 2e 0f 1f 84 00 00 00 00 66 90 0f 22 0f 1f 84 00 00 00 00 00 90 eb 07 0f 00 2d 7f e9 36 00 fb 40 00 83 ff 07 77 21 89 ff ff 24 fd 8d 3d a1 bd 0f 21 f8 RSP: 0018:fffffb603de8 EFLAGS: 00000202 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 00000000f92f46680 RDX: 0000000000000037 RSI: 00000000ffffff RDI: 000000000000518d4 RBP: ffffffb603df0 R08: 000000cd42e4dffb R09: ffffffb603d70 R10: 0000004d80d62680 R11: 0000000000000001 R12: ffffffb60bf40 R13: 0000000000000000 R14: 0000000000000000 R15: ffffffb60aff8 ? default_idle+0x9/0x20 arch_cpu_idle+0x9/0x10 default_idle_call+0x29/0xf0 do_idle+0x1f2/0x240 cpu_startup_entry+0x2c/0x30 rest_init+0xe7/0x100 start_kernel+0x76b/0xb90 x86_64_start_reservations+0x18/0x30 x86_64_start_kernel+0xc0/0x110 ? setup_ghcb+0xe/0x130 common_startup_64+0x13e/0x141 </TASK> Modules linked in: esp4_offload esp4 xfrm_interface xfrm6_tunnel tunnel4 tunnel6 xfrm_user xfrm_algo binf ---truncated---
CVE-2025-21721	In the Linux kernel, the following vulnerability has been resolved: nilfs2: handle errors that nilfs_prepare_chunk() may return Patch series "nilfs2: fix issues with rename operations". This series fixes BUG_ON check failures reported by syzbot around rename operations, and a minor behavioral issue where the mtime of a child directory changes when it is renamed instead of moved. This patch (of 2): The directory manipulation routines nilfs_set_link() and nilfs_delete_entry() rewrite the directory entry in the folio/page previously read by nilfs_find_entry(), so error handling is omitted on the assumption that nilfs_prepare_chunk(), which prepares the buffer for rewriting, will always succeed for these. And if an error is returned, it triggers the legacy BUG_ON() checks in each routine. This assumption is wrong, as proven by syzbot: the buffer layer called by nilfs_prepare_chunk() may call nilfs_get_block() if necessary, which may fail due to metadata corruption or other reasons. This has been there all along, but improved sanity checks and error handling may have made it more reproducible in fuzzing tests. Fix this issue by adding missing error paths in nilfs_set_link(), nilfs_delete_entry(), and their caller nilfs_rename().

CVE ID	Description
CVE-2025-21722	In the Linux kernel, the following vulnerability has been resolved: nilfs2: do not force clear folio if buffer is referenced Patch series "nilfs2: protect busy buffer heads from being force-cleared". This series fixes the buffer head state inconsistency issues reported by syzbot that occurs when the filesystem is corrupted and falls back to read-only, and the associated buffer head use-after-free issue. This patch (of 2): Syzbot has reported that after nilfs2 detects filesystem corruption and falls back to read-only, inconsistencies in the buffer state may occur. One of the inconsistencies is that when nilfs2 calls mark_buffer_dirty() to set a data or metadata buffer as dirty, but it detects that the buffer is not in the uptodate state: WARNING: CPU: 0 PID: 6049 at fs/buffer.c:1177 mark_buffer_dirty+0x2e5/0x520 fs/buffer.c:1177 ... Call Trace: <TASK> nilfs_palloc_commit_alloc_entry+0x4b/0x160 fs/nilfs2/alloc.c:598 nilfs_ifile_create_inode+0x1dd/0x3a0 fs/nilfs2/ifile.c:73 nilfs_new_inode+0x254/0x830 fs/nilfs2/inode.c:344 nilfs_mkdir+0x10d/0x340 fs/nilfs2/namei.c:218 vfs_mkdir+0x2f9/0x4f0 fs/namei.c:4257 do_mkdirat+0x264/0x3a0 fs/namei.c:4280 __do_sys_mkdirat fs/namei.c:4295 [inline] __se_sys_mkdirat fs/namei.c:4293 [inline] __x64_sys_mkdirat+0x87/0xa0 fs/namei.c:4293 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f The other is when nilfs_btree_propagate(), which propagates the dirty state to the ancestor nodes of a b-tree that point to a dirty buffer, detects that the origin buffer is not dirty, even though it should be: WARNING: CPU: 0 PID: 5245 at fs/nilfs2/btree.c:2089 nilfs_btree_propagate+0xc79/0xdf0 fs/nilfs2/btree.c:2089 ... Call Trace: <TASK> nilfs_bmap_propagate+0x75/0x120 fs/nilfs2/bmap.c:345 nilfs_collect_file_data+0x4d/0xd0 fs/nilfs2/segment.c:587 nilfs_segctor_apply_buffers+0x184/0x340 fs/nilfs2/segment.c:1006 nilfs_segctor_scan_file+0x28c/0xa50 fs/nilfs2/segment.c:1045 nilfs_segctor_collect_blocks fs/nilfs2/segment.c:1216 [inline] nilfs_segctor_collect fs/nilfs2/segment.c:1540 [inline] nilfs_segctor_do_construct+0x1c28/0x6b90 fs/nilfs2/segment.c:2115 nilfs_segctor_construct+0x181/0x6b0 fs/nilfs2/segment.c:2479 nilfs_segctor_thread_construct fs/nilfs2/segment.c:2587 [inline] nilfs_segctor_thread+0x69e/0xe80 fs/nilfs2/segment.c:2701 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Both of these issues are caused by the callbacks that handle the page/folio write requests, forcibly clear various states, including the working state of the buffers they hold, at unexpected times when they detect read-only fallback. Fix these issues by checking if the buffer is referenced before clearing the page/folio state, and skipping the clear if it is.
CVE-2025-21723	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Fix possible crash when setting up bsg fails If bsg_setup_queue() fails, the bsg_queue is assigned a non-NULL value. Consequently, in mpi3mr_bsg_exit(), the condition "if(!mrioc->bsg_queue)" will not be satisfied, preventing execution from entering bsg_remove_queue(), which could lead to the following crash: BUG: kernel NULL pointer dereference, address: 000000000000041c Call Trace: <TASK> mpi3mr_bsg_exit+0x1f/0x50 [mpi3mr] mpi3mr_remove+0x6f/0x340 [mpi3mr] pci_device_remove+0x3f/0xb0 device_release_driver_internal+0x19d/0x220 unbind_store+0xa4/0xb0 kernfs_fop_write_iter+0x11f/0x200 vfs_write+0x1fc/0x3e0 ksys_write+0x67/0xe0 do_syscall_64+0x38/0x80 entry_SYSCALL_64_after_hwframe+0x78/0xe2
CVE-2025-21724	In the Linux kernel, the following vulnerability has been resolved: iommufd/iova_bitmap: Fix shift-out-of-bounds in iova_bitmap_offset_to_index() Resolve a UBSAN shift-out-of-bounds issue in iova_bitmap_offset_to_index() where shifting the constant "1" (of type int) by bitmap->mapped.pgshift (an unsigned long value) could result in undefined behavior. The constant "1" defaults to a 32-bit "int", and when "pgshift" exceeds 31 (e.g., pgshift = 63) the shift operation overflows, as the result cannot be represented in a 32-bit type. To resolve this, the constant is updated to "1UL", promoting it to an unsigned long type to match the operand's type.

CVE ID	Description
CVE-2025-21725	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix oops due to unset link speed It isn't guaranteed that NETWORK_INTERFACE_INFO::LinkSpeed will always be set by the server, so the client must handle any values and then prevent oopses like below from happening: Oops: divide error: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 0 UID: 0 PID: 1323 Comm: cat Not tainted 6.13.0-rc7 #2 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-3.fc41 04/01/2014 RIP: 0010:cifs_debug_data_proc_show+0xa45/0x1460 [cifs] Code: 00 00 48 89 df e8 3b cd 1b c1 41 f6 44 24 2c 04 0f 84 50 01 00 00 48 89 ef e8 e7 d0 1b c1 49 8b 44 24 18 31 d2 49 8d 7c 24 28 <48> f7 74 24 18 48 89 c3 e8 6e cf 1b c1 41 8b 6c 24 28 49 8d 7c 24 RSP: 0018:ffffc90001817be0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff88811230022c RCX: ffffffff041bd99 RDX: 0000000000000000 RSI: 0000000000000567 RDI: ffff888112300228 RBP: ffff888112300218 R08: ffff52000302f5f R09: ffffd1022fa58ac R10: ffff888117d2c566 R11: 00000000fffffffe R12: ffff888112300200 R13: 000000012a15343f R14: 0000000000000001 R15: ffff888113f2db58 FS: 00007fe27119e740(0000) GS:ffff888148600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fe2633c5000 CR3: 0000000124da0000 CR4: 0000000000750ef0 PKRU: 55555554 Call Trace: <TASK> ? __die_body.cold+0x19/0x27 ? die+0x2e/0x50 ? do_trap+0x159/0x1b0 ? cifs_debug_data_proc_show+0xa45/0x1460 [cifs] ? do_error_trap+0x90/0x130 ? cifs_debug_data_proc_show+0xa45/0x1460 [cifs] ? exc_divide_error+0x39/0x50 ? cifs_debug_data_proc_show+0xa45/0x1460 [cifs] ? asm_exc_divide_error+0x1a/0x20 ? cifs_debug_data_proc_show+0xa39/0x1460 [cifs] ? cifs_debug_data_proc_show+0xa45/0x1460 [cifs] ? seq_read_iter+0x42e/0x790 seq_read_iter+0x19a/0x790 proc_reg_read_iter+0xbe/0x110 ? __pfx_proc_reg_read_iter+0x10/0x10 vfs_read+0x469/0x570 ? do_user_addr_fault+0x398/0x760 ? __pfx_vfs_read+0x10/0x10 ? find_held_lock+0x8a/0xa0 ? __pfx_lock_release+0x10/0x10 ksys_read+0xd3/0x170 ? __pfx_ksys_read+0x10/0x10 ? __rcu_read_unlock+0x50/0x270 ? mark_held_locks+0x1a/0x90 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fe271288911 Code: 00 48 8b 15 01 25 10 00 f7 d8 64 89 02 b8 ff ff ff ff eb bd e8 20 ad 01 00 f3 0f 1e fa 80 3d b5 a7 10 00 00 74 13 31 c0 0f 05 <48> 3d 00 f0 ff ff 77 4f c3 66 0f 1f 44 00 00 55 48 89 e5 48 83 ec RSP: 002b:00007ffe87c079d8 EFLAGS: 00000246 ORIG_RAX: 0000000000000000 RAX: ffffffff00000000 RBX: 0000000000004000 RCX: 00007fe271288911 RDX: 0000000000004000 RSI: 00007fe2633c6000 RDI: 0000000000000003 RBP: 00007ffe87c07a00 R08: 0000000000000000 R09: 00007fe2713e6380 R10: 0000000000000022 R11: 0000000000000246 R12: 0000000000004000 R13: 00007fe2633c6000 R14: 0000000000000003 R15: 0000000000000000 <TASK> Fix this by setting cifs_server_iface::speed to a sane value (1Gbps) by default when link speed is unset.
CVE-2025-21726	In the Linux kernel, the following vulnerability has been resolved: padata: avoid UAF for reorder_work Although the previous patch can avoid ps and ps UAF for _do_serial, it can not avoid potential UAF issue for reorder_work. This issue can happen just as below: crypto_request crypto_request crypto_del_alg padata_do_serial ... padata_reorder // processes all remaining // requests then breaks while (1) { if (!padata) break; ... } padata_do_serial // new request added list_add // sees the new request queue_work(reorder_work) padata_reorder queue_work_on(squeue->work) ... <worker context> padata_serial_worker // completes new request, // no more outstanding // requests crypto_del_alg // free pd <worker context> invoke padata_reorder // UAF of pd To avoid UAF for 'reorder_work', get 'pd' ref before put 'reorder_work' into the 'serial_wq' and put 'pd' ref until the 'serial_wq' finish.
CVE-2025-21727	In the Linux kernel, the following vulnerability has been resolved: padata: fix UAF in padata_reorder A bug was found when run ltp test: BUG: KASAN: slab-use-after-free in padata_find_next +0x29/0x1a0 Read of size 4 at addr ffff88bbfe003524 by task kworker/u113:2/3039206 CPU: 0 PID: 3039206 Comm: kworker/u113:2 Kdump: loaded Not tainted 6.6.0+ Workqueue: pdecrypt_parallel padata_parallel_worker Call Trace: <TASK> dump_stack_lvl+0x32/0x50 print_address_description.constprop.0+0x6b/0x3d0 print_report+0xdd/0x2c0 kasan_report+0xa5/0xd0 padata_find_next+0x29/0x1a0 padata_reorder+0x131/0x220 padata_parallel_worker+0x3d/0xc0 process_one_work+0x2ec/0x5a0 If 'mdelay(10)' is added before calling 'padata_find_next' in the 'padata_reorder' function, this issue could be reproduced easily with ltp test (pccrypt_aead01). This can be explained as below: pccrypt_aead_encrypt ... padata_do_parallel refcount_inc(&pd->refcnt); // add refcnt ... padata_do_serial padata_reorder // pd while (1) { padata_find_next(pd, true); // using pd queue_work_on ... padata_serial_worker crypto_del_alg padata_put_pd_cnt // sub refcnt padata_free_shell padata_put_pd(ps->pd); // pd is freed // loop again, but pd is freed // call padata_find_next, UAF } In the padata_reorder function, when it loops in 'while', if the alg is deleted, the refcnt may be decreased to 0 before entering 'padata_find_next', which leads to UAF. As mentioned in [1], do_serial is supposed to be called with BHs disabled and always happen under RCU protection, to address this issue, add synchronize_rcu() in 'padata_free_shell' wait for all _do_serial calls to finish. [1] https://lore.kernel.org/all/20221028160401.cccypv4euxikusi@parnassus.localdomain/ [2] https://lore.kernel.org/linux-kernel/jfjz5d7zwbyztackem7ibzalm5lnxldi2e0feiczmqms2m7o6/fq426cwnjtkm/
CVE-2025-21728	In the Linux kernel, the following vulnerability has been resolved: bpf: Send signals asynchronously if ! preemptible BPF programs can execute in all kinds of contexts and when a program running in a non-preemptible context uses the bpf_send_signal() kfunc, it will cause issues because this kfunc can sleep. Change `irqs_disabled()` to `!preemptible()`.

CVE ID	Description
CVE-2025-21729	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix race between cancel_hw_scan and hw_scan completion The rtwdev->scanning flag isn't protected by mutex originally, so cancel_hw_scan can pass the condition, but suddenly hw_scan completion unset the flag and calls ieee80211_scan_completed() that will free local->hw_scan_req. Then, cancel_hw_scan raises null-ptr-deref and use-after-free. Fix it by moving the check condition to where protected by mutex. KASAN: null-ptr-deref in range [0x0000000000000088-0x000000000000008f] CPU: 2 PID: 6922 Comm: kworker/2:2 Tainted: G OE Hardware name: LENOVO 2356AD1/2356AD1, BIOS G7ETB6WW (2.76) 09/10/2019 Workqueue: events cfg80211_conn_work [cfg80211] RIP: 0010:rtw89_fw_h2c_scan_offload_be+0xc33/0x13c3 [rtw89_core] Code: 00 45 89 6c 24 1c 0f 85 23 01 00 00 48 8b 85 20 ff ff ff 48 8d RSP: 0018:ffff88811fd9f068 EFLAGS: 00010206 RAX: dffffc0000000000 RBX: ffff88811fd9f258 RCX: 0000000000000001 RDX: 0000000000000011 RSI: 0000000000000001 RDI: 0000000000000089 RBP: ffff88811fd9f170 R08: 0000000000000000 R09: 0000000000000000 R10: ffff88811fd9f108 R11: 0000000000000000 R12: ffff88810e47f960 R13: 0000000000000000 R14: 000000000000ffff R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff8881d6f00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007531dfca55b0 CR3: 00000001be296004 CR4: 00000000001706e0 Call Trace: <TASK> ? show_regs+0x61/0x73 ? __die_body+0x20/0x73 ? die_addr+0x4f/0x7b ? exc_general_protection+0x191/0x1db ? asm_exc_general_protection+0x27/0x30 ? rtw89_fw_h2c_scan_offload_be+0xc33/0x13c3 [rtw89_core] ? rtw89_fw_h2c_scan_offload_be+0x458/0x13c3 [rtw89_core] ? __pfx_rtw89_fw_h2c_scan_offload_be+0x10/0x10 [rtw89_core] ? do_raw_spin_lock+0x75/0xdb ? __pfx_do_raw_spin_lock+0x10/0x10 rtw89_hw_scan_offload+0xb5e/0xb7 [rtw89_core] ? _raw_spin_unlock+0xe/0x24 ? __mutex_lock.constprop.0+0x40c/0x471 ? __pfx_rtw89_hw_scan_offload+0x10/0x10 [rtw89_core] ? __mutex_lock_slowpath+0x13/0x1f ? mutex_lock+0xa2/0xdc ? __pfx_mutex_lock+0x10/0x10 rtw89_hw_scan_abort+0x58/0xb7 [rtw89_core] rtw89_ops_cancel_hw_scan+0x120/0x13b [rtw89_core] ieee80211_scan_cancel+0x468/0x4d0 [mac80211] ieee80211_prep_connection+0x858/0x899 [mac80211] ieee80211_mgd_auth+0xbea/0xdde [mac80211] ? __pfx_ieee80211_mgd_auth+0x10/0x10 [mac80211] ? cfg80211_find_elem+0x15/0x29 [cfg80211] ? is_bss+0x1b7/0x1d7 [cfg80211] ieee80211_auth+0x18/0x27 [mac80211] cfg80211_mlme_auth+0x3bb/0x3e7 [cfg80211] cfg80211_conn_do_work+0x410/0xb81 [cfg80211] ? __pfx_cfg80211_conn_do_work+0x10/0x10 [cfg80211] ? __kasan_check_read+0x11/0x1f ? psi_group_change+0x8bc/0x944 ? __kasan_check_write+0x14/0x22 ? mutex_lock+0x8e/0xdc ? __pfx_mutex_lock+0x10/0x10 ? __pfx__radix_tree_lookup+0x10/0x10 cfg80211_conn_work+0x245/0x34d [cfg80211] ? __pfx_cfg80211_conn_work+0x10/0x10 [cfg80211] ? update_cfs_rq_load_avg+0x3bc/0x3d7 ? sched_clock_noinstr+0x9/0x1a ? sched_clock+0x10/0x24 ? sched_clock_cpu+0x7e/0x42e ? newidle_balance+0x796/0x937 ? __pfx_sched_clock_cpu+0x10/0x10 ? __pfx_newidle_balance+0x10/0x10 ? __kasan_check_read+0x11/0x1f ? psi_group_change+0x8bc/0x944 ? _raw_spin_unlock+0xe/0x24 ? raw_spin_rq_unlock+0x47/0x54 ? raw_spin_rq_unlock_irq+0x9/0x1f ? finish_task_switch.isra.0+0x347/0x586 ? __schedule+0x27bf/0x2892 ? mutex_unlock+0x80/0xd0 ? do_raw_spin_lock+0x75/0xdb ? __pfx__schedule+0x10/0x10 process_scheduled_works+0x58c/0x821 worker_thread+0x4c7/0x586 ? __kasan_check_read+0x11/0x1f kthread+0x285/0x294 ? __pfx_worker_thread+0x10/0x10 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x29/0x6f ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1b/0x30 </TASK>
CVE-2025-2173	A vulnerability was found in libzvb up to 0.2.43. It has been classified as problematic. Affected is the function vbi_stmdui_conv_ucs2 of the file src/conv.c. The manipulation of the argument src_length leads to uninitialized pointer. It is possible to launch the attack remotely. The exploit has been disclosed to the public and may be used. Upgrading to version 0.2.44 is able to address this issue. The patch is identified as 8def647eea27f7fd7ad33ff79c2d6d3e39948dce. It is recommended to upgrade the affected component. The code maintainer was informed beforehand about the issues. She reacted very fast and highly professional.

CVE ID	Description
CVE-2025-21730	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid to init mgnt_entry list twice when WoWLAN failed If WoWLAN failed in resume flow, the rtw89_ops_add_interface() triggered without removing the interface first. Then the mgnt_entry list init again, causing the list_empty() check in rtw89_chanctx_ops_assign_vif() useless, and list_add_tail() again. Therefore, we have added a check to prevent double adding of the list. rtw89_8852ce 0000:01:00.0: failed to check wow status disabled rtw89_8852ce 0000:01:00.0: wow: failed to check disable fw ready rtw89_8852ce 0000:01:00.0: wow: failed to swap to normal fw rtw89_8852ce 0000:01:00.0: failed to disable wow rtw89_8852ce 0000:01:00.0: failed to resume for wow -110 rtw89_8852ce 0000:01:00.0: MAC has already powered on i2c_hid_acpi i2c-ILTK0001:00: PM: acpi_subsys_resume+0x0/0x60 returned 0 after 284705 usecs list_add corruption. prev->next should be next (ffff9d9719d82228), but was ffff9d9719f96030. (prev=ffff9d9719f96030). -----[cut here]----- kernel BUG at lib/list_debug.c:34! invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 2 PID: 6918 Comm: kworker/u8:19 Tainted: G U O Hardware name: Google Anraggar/Anraggar, BIOS Google_Anraggar.15217.514.0 03/25/2024 Workqueue: events_unbound async_run_entry_fn RIP: 0010:__list_add_valid_or_report+0x9f/0xb0 Code: e8 56 89 ff ff 0f 0b 48 c7 c7 3e fc e0 96 48 89 c6 e8 45 89 ff ... RSP: 0018:ffffa51b42bbbf0 EFLAGS: 00010246 RAX: 0000000000000075 RBX: ffff9d9719d82ab0 RCX: 13acb86e047a4400 RDX: 3fffffff RSI: 0000000000000000 RDI: 00000000ffffdfff RBP: ffffa51b42bbbf28 R08: ffffffff9768e250 R09: 00000000000001fff R10: ffffffff9765e250 R11: 00000000000005ffd R12: ffff9d9719f95c40 R13: ffff9d9719f95be8 R14: ffff9d97081bf78 R15: ffff9d9719d82060 FS: 0000000000000000(0000) GS:ffff9d9a6fb00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007e7d029a4060 CR3: 0000000345e38000 CR4: 000000000750ee0 PKRU: 55555554 Call Trace: <TASK> ? __die_body+0x68/0xb0 ? die+0xaa/0xd0 ? do_trap+0x9f/0x170 ? __list_add_valid_or_report+0x9f/0xb0 ? __list_add_valid_or_report+0x9f/0xb0 ? handle_invalid_op+0x69/0x90 ? __list_add_valid_or_report+0x9f/0xb0 ? exc_invalid_op+0x3c/0x50 ? asm_exc_invalid_op+0x16/0x20 ? __list_add_valid_or_report+0x9f/0xb0 rtw89_chanctx_ops_assign_vif+0x1f9/0x210 [rtw89_core cbb375c44bf28564ce479002bff66617a25d9ac1] ? __mutex_unlock_slowpath+0xa0/0xf0 rtw89_ops_assign_vif_chanctx+0x4b/0x90 [rtw89_core cbb375c44bf28564ce479002bff66617a25d9ac1] drv_assign_vif_chanctx+0xa7/0x1f0 [mac80211 6efaad16237eda0868b132d4f93ecf918a8b6] ieee80211_reconfig+0x9cb/0x17b0 [mac80211 6efaad16237eda0868b132d4f93ecf918a8b6] ? __pfx_wiphy_resume+0x10/0x10 [cfg80211 572d03acaaa933fe38251be7fce3b3675284b8ed] ? dev_printk_emit+0x51/0x70 ? _dev_info+0x6e/0x90 wiphy_resume+0x89/0x180 [cfg80211 572d03acaaa933fe38251be7fce3b3675284b8ed] ? __pfx_wiphy_resume+0x10/0x10 [cfg80211 572d03acaaa933fe38251be7fce3b3675284b8ed] dpm_run_callback+0x37/0x1e0 device_resume+0x26d/0x4b0 ? __pfx_dpm_watchdog_handler+0x10/0x10 async_resume+0x1d/0x30 async_run_entry_fn+0x29/0xd0 worker_thread+0x397/0x970 kthread+0xed/0x110 ? __pfx_worker_thread+0x10/0x10 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x38/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1b/0x30 </TASK>
CVE-2025-21731	In the Linux kernel, the following vulnerability has been resolved: nbd: don't allow reconnect after disconnect Following process can cause nbd_config UAF: 1) grab nbd_config temporarily; 2) nbd_genl_disconnect() flush all rcv_work() and release the initial reference: nbd_genl_disconnect nbd_disconnect_and_put nbd_disconnect flush_workqueue(nbd->rcv_workq) if (test_and_clear_bit(NBD_RT_HAS_CONFIG_REF, ...)) nbd_config_put -> due to step 1), reference is still not zero 3) nbd_genl_reconfigure() queue rcv_work() again; nbd_genl_reconfigure config = nbd_get_config_unlocked(nbd) if (!config) -> succeed if (!test_bit(NBD_RT_BOUND, ...)) -> succeed nbd_reconnect_socket queue_work(nbd->rcv_workq, &args->work) 4) step 1) release the reference; 5) Finally, rcv_work() will trigger UAF: rcv_work nbd_config_put(nbd) -> nbd_config is freed atomic_dec(&config->rcv_threads) -> UAF Fix the problem by clearing NBD_RT_BOUND in nbd_genl_disconnect(), so that nbd_genl_reconfigure() will fail.

CVE ID	Description
CVE-2025-21732	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix a race for an ODP MR which leads to CQE with error This patch addresses a race condition for an ODP MR that can result in a CQE with an error on the UMR QP. During the <code>__mlx5_ib_dereg_mr()</code> flow, the following sequence of calls occurs: <code>mlx5_revoke_mr()</code> <code>mlx5r_umr_revoke_mr()</code> <code>mlx5r_umr_post_send_wait()</code> At this point, the lkey is freed from the hardware's perspective. However, concurrently, <code>mlx5_ib_invalidate_range()</code> might be triggered by another task attempting to invalidate a range for the same freed lkey. This task will: - Acquire the <code>umem_odp->umem_mutex</code> lock. - Call <code>mlx5r_umr_update_xlt()</code> on the UMR QP. - Since the lkey has already been freed, this can lead to a CQE error, causing the UMR QP to enter an error state [1]. To resolve this race condition, the <code>umem_odp->umem_mutex</code> lock is now also acquired as part of the <code>mlx5_revoke_mr()</code> scope. Upon successful revoke, we set <code>umem_odp->private</code> which points to that MR to NULL, preventing any further invalidation attempts on its lkey. [1] From dmesg: infiniband rocep8s0f0: dump_cqe:277:(pid 0): WC error: 6, Message: memory bind operation error cqe_dump: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 cqe_dump: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 cqe_dump: 00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 cqe_dump: 00000030: 00 00 00 00 08 00 78 06 25 00 11 b9 00 0e dd d2 WARNING: CPU: 15 PID: 1506 at drivers/infiniband/hw/mlx5/umr.c:394 mlx5r_umr_post_send_wait+0x15a/0x2b0 [mlx5_ib] Modules linked in: ip6table_mangle ip6table_natip6table_filter ip6_tables iptable_mangle xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter rpcsec_gss_krb5 auth_rpcgss oid_registry overlay rpcrdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi rdma_cm iw_cm ib_umad ib_ipoib ib_cm mlx5_ib ib_uverbs ib_core fuse mlx5_core CPU: 15 UID: 0 PID: 1506 Comm: ibv_rc_pingpong Not tainted 6.12.0-rc7+ #1626 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:mlx5r_umr_post_send_wait+0x15a/0x2b0 [mlx5_ib] [...] Call Trace: <TASK> mlx5r_umr_update_xlt+0x23c/0x3e0 [mlx5_ib] mlx5_ib_invalidate_range+0x2e1/0x330 [mlx5_ib] __mmu_notifier_invalidate_range_start+0x1e1/0x240 zap_page_range_single+0xf1/0x1a0 madvise_vma_behavior+0x677/0x6e0 do_madvise+0x1a2/0x4b0 __x64_sys_madvise+0x25/0x30 do_syscall_64+0x6b/0x140 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2025-21733	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Fix resetting of tracepoints If a timerlat tracer is started with the osnoise option OSNOISE_WORKLOAD disabled, but then that option is enabled and timerlat is removed, the tracepoints that were enabled on timerlat registration do not get disabled. If the option is disabled again and timerlat is started, then it triggers a warning in the tracepoint code due to registering the tracepoint again without ever disabling it. Do not use the same user space defined options to know to disable the tracepoints when timerlat is removed. Instead, set a global flag when it is enabled and use that flag to know to disable the events. ~# echo NO_OSNOISE_WORKLOAD > /sys/kernel/tracing/osnoise/options ~# echo timerlat > /sys/kernel/tracing/current_tracer ~# echo OSNOISE_WORKLOAD > /sys/kernel/tracing/osnoise/options ~# echo nop > /sys/kernel/tracing/current_tracer ~# echo NO_OSNOISE_WORKLOAD > /sys/kernel/tracing/osnoise/options ~# echo timerlat > /sys/kernel/tracing/current_tracer Triggers: -----[cut here]----- WARNING: CPU: 6 PID: 1337 at kernel/tracepoint.c:294 tracepoint_add_func+0x3b6/0x3f0 Modules linked in: CPU: 6 UID: 0 PID: 1337 Comm: rta Not tainted 6.13.0-rc4-test-00018-ga867c441128e-dirty #73 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010:tracepoint_add_func+0x3b6/0x3f0 Code: 48 8b 53 28 48 8b 73 20 4c 89 04 24 e8 23 59 11 00 4c 8b 04 24 e9 36 fe ff ff 0f 0b b8 ea ff ff ff 45 84 e4 0f 84 68 fe ff ff <0> 0b e9 61 fe ff ff 48 8b 7b 18 48 85 ff 0f 84 4f ff ff ff 49 8b RSP: 0018:ffff9b003a87ca0 EFLAGS: 00010202 RAX: 00000000fffff RBX: ffffffff92f30860 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffff9bf59e91ccd0 RDI: ffffffff913b6410 RBP: 000000000000000a R08: 000000000000005c R09: 0000000000000002 R10: ffff9b003a87ce0 R11: 0000000000000002 R12: 0000000000000001 R13: ffff9b003a87ce0 R14: ffffffff92f30860 R15: 0000000000000008 FS: 00007fce81209240(0000) GS:ffff9bf6dd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000055e99b728000 CR3: 00000001277c0002 CR4: 0000000000172ef0 Call Trace: <TASK> ? __warn.cold+0xb7/0x14d ? tracepoint_add_func+0x3b6/0x3f0 ? report_bug+0xea/0x170 ? handle_bug+0x58/0x90 ? exc_invalid_op+0x17/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? __pfx_trace_sched_migrate_callback+0x10/0x10 ? tracepoint_add_func+0x3b6/0x3f0 ? __pfx_trace_sched_migrate_callback+0x10/0x10 ? __pfx_trace_sched_migrate_callback+0x10/0x10 tracepoint_probe_register+0x78/0xb0 ? __pfx_trace_sched_migrate_callback+0x10/0x10 osnoise_workload_start+0x2b5/0x370 timerlat_tracer_init+0x76/0x1b0 tracing_set_tracer+0x244/0x400 tracing_set_trace_write+0xa0/0xe0 vfs_write+0xfc/0x570 ? do_sys_openat2+0x9c/0xe0 ksys_write+0x72/0xf0 do_syscall_64+0x79/0x1c0 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2025-21734	In the Linux kernel, the following vulnerability has been resolved: misc: fastrpc: Fix copy buffer page size For non-registered buffer, fastrpc driver copies the buffer and pass it to the remote subsystem. There is a problem with current implementation of page size calculation which is not considering the offset in the calculation. This might lead to passing of improper and out-of-bounds page size which could result in memory issue. Calculate page start and page end using the offset adjusted address instead of absolute address.

CVE ID	Description
CVE-2025-21735	In the Linux kernel, the following vulnerability has been resolved: NFC: nci: Add bounds checking in nci_hci_create_pipe() The "pipe" variable is a u8 which comes from the network. If it's more than 127, then it results in memory corruption in the caller, nci_hci_connect_gate().
CVE-2025-21736	In the Linux kernel, the following vulnerability has been resolved: nilfs2: fix possible int overflows in nilfs_fiemap() Since nilfs_bmap_lookup_contig() in nilfs_fiemap() calculates its result by being prepared to go through potentially maxblocks == INT_MAX blocks, the value in n may experience an overflow caused by left shift of blkbits. While it is extremely unlikely to occur, play it safe and cast right hand expression to wider type to mitigate the issue. Found by Linux Verification Center (linuxtesting.org) with static analysis tool SVACE.
CVE-2025-21738	In the Linux kernel, the following vulnerability has been resolved: ata: libata-sff: Ensure that we cannot write outside the allocated buffer reveliofuzzing reported that a SCSI_IOCTL_SEND_COMMAND ioctl with out_len set to 0xd42, SCSI command set to ATA_16 PASS-THROUGH, ATA command set to ATA_NOP, and protocol set to ATA_PROT_PIO, can cause ata_pio_sector() to write outside the allocated buffer, overwriting random memory. While a ATA device is supposed to abort a ATA_NOP command, there does seem to be a bug either in libata-sff or QEMU, where either this status is not set, or the status is cleared before read by ata_sff_hsm_move(). Anyway, that is most likely a separate bug. Looking at __atapi_pio_bytes(), it already has a safety check to ensure that __atapi_pio_bytes() cannot write outside the allocated buffer. Add a similar check to ata_pio_sector(), such that also ata_pio_sector() cannot write outside the allocated buffer.
CVE-2025-21739	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Fix use-after free in init error and remove paths devm_blk_crypto_profile_init() registers a cleanup handler to run when the associated (platform-) device is being released. For UFS, the crypto private data and pointers are stored as part of the ufs_hba's data structure 'struct ufs_hba::crypto_profile'. This structure is allocated as part of the underlying ufshcd and therefore Scsi_host allocation. During driver release or during error handling in ufshcd_pltfrm_init(), this structure is released as part of ufshcd_dealloc_host() before the (platform-) device associated with the crypto call above is released. Once this device is released, the crypto cleanup code will run, using the just-released 'struct ufs_hba::crypto_profile'. This causes a use-after-free situation: Call trace: kfree+0x60/0x2d8 (P) kvfree+0x44/0x60 blk_crypto_profile_destroy_callback+0x28/0x70 devm_action_release+0x1c/0x30 release_nodes+0x6c/0x108 devres_release_all+0x98/0x100 device_unbind_cleanup+0x20/0x70 really_probe+0x218/0x2d0 In other words, the initialisation code flow is: platform-device probe ufshcd_pltfrm_init() ufshcd_alloc_host() scsi_host_alloc() allocation of struct ufs_hba creation of scsi-host devices devm_blk_crypto_profile_init() devm registration of cleanup handler using platform-device and during error handling of ufshcd_pltfrm_init() or during driver removal: ufshcd_dealloc_host() scsi_host_put() put_device(scsi-host) release of struct ufs_hba put_device(platform-device) crypto cleanup handler To fix this use-after free, change ufshcd_alloc_host() to register a devres action to automatically cleanup the underlying SCSI device on ufshcd destruction, without requiring explicit calls to ufshcd_dealloc_host(). This way: * the crypto profile and all other ufs_hba-owned resources are destroyed before SCSI (as they've been registered after) * a memleak is plugged in tc-dwc-g210-pci.c remove() as a side-effect * EXPORT_SYMBOL_GPL(ufshcd_dealloc_host) can be removed fully as it's not needed anymore * no future drivers using ufshcd_alloc_host() could ever forget adding the cleanup
CVE-2025-2174	A vulnerability was found in libzvbi up to 0.2.43. It has been declared as problematic. Affected by this vulnerability is the function vbi_stndup_iconv_ucs2 of the file src/conv.c. The manipulation of the argument src_length leads to integer overflow. The attack can be launched remotely. The exploit has been disclosed to the public and may be used. Upgrading to version 0.2.44 is able to address this issue. The patch is named ca1672134b3e2962cd392212c73f44f8f4cb489f. It is recommended to upgrade the affected component. The code maintainer was informed beforehand about the issues. She reacted very fast and highly professional.
CVE-2025-21741	In the Linux kernel, the following vulnerability has been resolved: usbnet: ipheth: fix DPE OoB read Fix an out-of-bounds DPE read, limit the number of processed DPEs to the amount that fits into the fixed-size NDP16 header.
CVE-2025-21742	In the Linux kernel, the following vulnerability has been resolved: usbnet: ipheth: use static NDP16 location in URB Original code allowed for the start of NDP16 to be anywhere within the URB based on the `wNdpIndex` value in NTH16. Only the start position of NDP16 was checked, so it was possible for even the fixed-length part of NDP16 to extend past the end of URB, leading to an out-of-bounds read. On iOS devices, the NDP16 header always directly follows NTH16. Rely on and check for this specific format. This, along with NCM-specific minimal URB length check that already exists, will ensure that the fixed-length part of NDP16 plus a set amount of DPEs fit within the URB. Note that this commit alone does not fully address the OoB read. The limit on the amount of DPEs needs to be enforced separately.

CVE ID	Description
CVE-2025-21743	In the Linux kernel, the following vulnerability has been resolved: usbnet: ipheth: fix possible overflow in DPE length check Originally, it was possible for the DPE length check to overflow if <code>wDatagramIndex + wDatagramLength > U16_MAX</code> . This could lead to an OoB read. Move the <code>wDatagramIndex</code> term to the other side of the inequality. An existing condition ensures that <code>wDatagramIndex < urb->actual_length</code> .
CVE-2025-21744	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: fix NULL pointer dereference in <code>brcmf_txfinalize()</code> On removal of the device or unloading of the kernel module a potential NULL pointer dereference occurs. The following sequence deletes the interface: <code>brcmf_detach()</code> <code>brcmf_remove_interface()</code> <code>brcmf_del_if()</code> Inside the <code>brcmf_del_if()</code> function the <code>drv->if2bss[ifidx]</code> is updated to <code>BRCMF_BSSIDX_INVALID (-1)</code> if the <code>bsscfgidx</code> matches. After <code>brcmf_remove_interface()</code> call the <code>brcmf_proto_detach()</code> function is called providing the following sequence: <code>brcmf_detach()</code> <code>brcmf_proto_detach()</code> <code>brcmf_proto_msgbuf_detach()</code> <code>brcmf_flowring_detach()</code> <code>brcmf_msgbuf_delete_flowring()</code> <code>brcmf_msgbuf_remove_flowring()</code> <code>brcmf_flowring_delete()</code> <code>brcmf_get_ifp()</code> <code>brcmf_txfinalize()</code> Since <code>brcmf_get_ip()</code> can and actually will return NULL in this case the call to <code>brcmf_txfinalize()</code> will result in a NULL pointer dereference inside <code>brcmf_txfinalize()</code> when trying to update <code>ifp->ndev->stats.tx_errors</code> . This will only happen if a flowring still has an skb. Although the NULL pointer dereference has only been seen when trying to update the tx statistic, all other uses of the <code>ifp</code> pointer have been guarded as well with an early return if <code>ifp</code> is NULL.
CVE-2025-21745	In the Linux kernel, the following vulnerability has been resolved: blk-cgroup: Fix class <code>@block_class</code> 's subsystem refcount leakage <code>blkcg_fill_root_iostats()</code> iterates over <code>@block_class</code> 's devices by <code>class_dev_iter_init(next)</code> , but does not end iterating with <code>class_dev_iter_exit()</code> , so causes the class's subsystem refcount leakage. Fix by ending the iterating with <code>class_dev_iter_exit()</code> .
CVE-2025-21746	In the Linux kernel, the following vulnerability has been resolved: Input: synaptics - fix crash when enabling pass-through port When enabling a pass-through port an interrupt might come before <code>psmouse</code> driver binds to the pass-through port. However synaptics sub-driver tries to access <code>psmouse</code> instance presumably associated with the pass-through port to figure out if only 1 byte of response or entire protocol packet needs to be forwarded to the pass-through port and may crash if <code>psmouse</code> instance has not been attached to the port yet. Fix the crash by introducing <code>open()</code> and <code>close()</code> methods for the port and check if the port is open before trying to access <code>psmouse</code> instance. Because <code>psmouse</code> calls <code>serio_open()</code> only after attaching <code>psmouse</code> instance to serio port instance this prevents the potential crash.
CVE-2025-21748	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix integer overflows on 32 bit systems On 32bit systems the addition operations in <code>ipc_msg_alloc()</code> can potentially overflow leading to memory corruption. Add bounds checking using <code>KSMBD_IPC_MAX_PAYLOAD</code> to avoid overflow.
CVE-2025-21749	In the Linux kernel, the following vulnerability has been resolved: net: rose: lock the socket in <code>rose_bind()</code> syzbot reported a soft lockup in <code>rose_loopback_timer()</code> , with a repro calling <code>bind()</code> from multiple threads. <code>rose_bind()</code> must lock the socket to avoid this issue.
CVE-2025-2175	A vulnerability was found in <code>libzvbi</code> up to 0.2.43. It has been rated as problematic. Affected by this issue is the function <code>_vbi_strndup_conv</code> . The manipulation leads to integer overflow. The attack may be launched remotely. The exploit has been disclosed to the public and may be used. Upgrading to version 0.2.44 is able to address this issue. It is recommended to upgrade the affected component. The code maintainer was informed beforehand about the issues. She reacted very fast and highly professional.

CVE ID	Description
CVE-2025-21750	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: Check the return value of of_property_read_string_index() Somewhen between 6.10 and 6.11 the driver started to crash on my MacBookPro14,3. The property doesn't exist and 'tmp' remains uninitialized, so we pass a random pointer to devm_kstrdup(). The crash I am getting looks like this: BUG: unable to handle page fault for address: 00007f033c669379 PF: supervisor read access in kernel mode PF: error_code(0x0001) - permissions violation PGD 8000000101341067 P4D 8000000101341067 PUD 101340067 PMD 1013bb067 PTE 800000010aee9025 Oops: Oops: 0001 [#1] SMP PTI CPU: 4 UID: 0 PID: 827 Comm: (udev-worker) Not tainted 6.11.8-gentoo #1 Hardware name: Apple Inc. MacBookPro14,3/Mac-551B86E5744E2388, BIOS 529.140.2.0.0 06/23/2024 RIP: 0010:strlen+0x4/0x30 Code: f7 75 ec 31 c0 c3 cc cc cc cc 48 89 f8 c3 cc cc cc cc 0f 1f 40 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa <80> 3f 00 74 14 48 89 f8 48 83 c0 01 80 38 00 75 f7 48 29 f8 c3 cc RSP: 0018:ffffb4aac0683ad8 EFLAGS: 00010202 RAX: 00000000fffffea RBX: 00007f033c669379 RCX: 0000000000000001 RDX: 00000000000000cc0 RSI: 00007f033c669379 RDI: 00007f033c669379 RBP: 00000000fffffea R08: 0000000000000000 R09: 00000000c0ba916a R10: ffffffff R11: ffffffff R12: ffff91f7815b50c8 R13: 00000000000000cc0 R14: ffff91fafef30 R15: ffff4aac0683b30 FS: 00007f033cbe8c0(0000) GS:ffff91faeed00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f033c669379 CR3: 0000000107b1e004 CR4: 0000000003706f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __die+0x23/0x70 ? page_fault_oops+0x149/0x4c0 ? raw_spin_rq_lock_nested+0xe/0x20 ? sched_balance_newidle+0x22b/0x3c0 ? update_load_avg+0x78/0x770 ? exc_page_fault+0x6f/0x150 ? asm_exc_page_fault+0x26/0x30 ? __pfx_pci_conf1_write+0x10/0x10 ? strlen+0x4/0x30 devm_kstrdup+0x25/0x70 brcmf_of_probe+0x273/0x350 [brcmfmac]
CVE-2025-21751	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: HWS, change error flow on matcher disconnect Currently, when firmware failure occurs during matcher disconnect flow, the error flow of the function reconnects the matcher back and returns an error, which continues running the calling function and eventually frees the matcher that is being disconnected. This leads to a case where we have a freed matcher on the matchers list, which in turn leads to use-after-free and eventual crash. This patch fixes that by not trying to reconnect the matcher back when some FW command fails during disconnect. Note that we're dealing here with FW error. We can't overcome this problem. This might lead to bad steering state (e.g. wrong connection between matchers), and will also lead to resource leakage, as it is the case with any other error handling during resource destruction. However, the goal here is to allow the driver to continue and not crash the machine with use-after-free error.

CVE ID	Description
CVE-2025-21752	In the Linux kernel, the following vulnerability has been resolved: btrfs: don't use btrfs_set_item_key_safe on RAID stripe-extents Don't use btrfs_set_item_key_safe() to modify the keys in the RAID stripe-tree, as this can lead to corruption of the tree, which is caught by the checks in btrfs_set_item_key_safe(): BTRFS info (device nvme1n1): leaf 49168384 gen 15 total ptrs 194 free space 8329 owner 12 BTRFS info (device nvme1n1): refs 2 lock_owner 1030 current 1030 [snip] item 105 key (354549760 230 20480) itemoff 14587 itemsize 16 stride 0 devid 5 physical 67502080 item 106 key (354631680 230 4096) itemoff 14571 itemsize 16 stride 0 devid 1 physical 88559616 item 107 key (354631680 230 32768) itemoff 14555 itemsize 16 stride 0 devid 1 physical 88555520 item 108 key (354717696 230 28672) itemoff 14539 itemsize 16 stride 0 devid 2 physical 67604480 [snip] BTRFS critical (device nvme1n1): slot 106 key (354631680 230 32768) new key (354635776 230 4096) -----[cut here]----- kernel BUG at fs/btrfs/ctree.c:2602! Oops: invalid opcode: 0000 [#1] PREEMPT SMP PTI CPU: 1 UID: 0 PID: 1055 Comm: fsstress Not tainted 6.13.0-rc1+ #1464 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.2-3-gd478f380-rebuilt.opensuse.org 04/01/2014 RIP: 0010:btrfs_set_item_key_safe+0xf7/0x270 Code: <snip> RSP: 0018:ffff90001337ab0 EFLAGS: 00010287 RAX: 0000000000000000 RBX: ffff8881115fd000 RCX: 0000000000000000 RDX: 0000000000000001 RSI: 0000000000000001 RDI: 00000000fffffffb RBP: ffff888110ed6f50 R08: 00000000fffffffb R09: ffffffff8244c500 R10: 00000000fffffffb R11: 00000000fffffffb R12: ffff888100586000 R13: 00000000000000c9 R14: ffff90001337b1f R15: ffff888110f23b58 FS: 00007f7d75c72740(0000) GS:ffff88813bd00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fa811652c60 CR3: 0000000111398001 CR4: 0000000000370eb0 Call Trace: <TASK> ? __die_body.cold+0x14/0x1a ? die+0x2e/0x50 ? do_trap+0xca/0x110 ? do_error_trap+0x65/0x80 ? btrfs_set_item_key_safe+0xf7/0x270 ? exc_invalid_op+0x50/0x70 ? btrfs_set_item_key_safe+0xf7/0x270 ? asm_exc_invalid_op+0x1a/0x20 ? btrfs_set_item_key_safe+0xf7/0x270 btrfs_partially_delete_raid_extent+0xc4/0xe0 btrfs_delete_raid_extent+0x227/0x240 __btrfs_free_extent.isra.0+0x57f/0x9c0 ? exc_coproc_segment_overrun+0x40/0x40 __btrfs_run_delayed_refs+0x2fa/0xe80 btrfs_run_delayed_refs+0x81/0xe0 btrfs_commit_transaction+0x2dd/0xbe0 ? preempt_count_add+0x52/0xb0 btrfs_sync_file+0x375/0x4c0 do_fsync+0x39/0x70 __x64_sys_fsync+0x13/0x20 do_syscall_64+0x54/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f7d7550ef90 Code: <snip> RSP: 002b:00007ffd70237248 EFLAGS: 00000202 ORIG_RAX: 000000000000004a RAX: ffffffff8da RBX: 0000000000000004 RCX: 00007f7d7550ef90 RDX: 000000000000013a RSI: 000000000040eb28 RDI: 0000000000000004 RBP: 000000000000001b R08: 0000000000000078 R09: 00007ffd7023725c R10: 00007f7d75400390 R11: 0000000000000202 R12: 028f5c28f5c28f5c R13: 8f5c28f5c28f5c29 R14: 000000000040b520 R15: 00007f7d75c726c8 </TASK> While the root cause of the tree order corruption isn't clear, using btrfs_duplicate_item() to copy the item and then adjusting both the key and the per-device physical addresses is a safe way to counter this problem.

CVE ID	Description
CVE-2025-21753	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix use-after-free when attempting to join an aborted transaction When we are trying to join the current transaction and if it's aborted, we read its 'aborted' field after unlocking fs_info->trans_lock and without holding any extra reference count on it. This means that a concurrent task that is aborting the transaction may free the transaction before we read its 'aborted' field, leading to a use-after-free. Fix this by reading the 'aborted' field while holding fs_info->trans_lock since any freeing task must first acquire that lock and set fs_info->running_transaction to NULL before freeing the transaction. This was reported by syzbot and Dmitry with the following stack traces from KASAN: <pre>===== BUG: KASAN: slab-use-after-free in join_transaction+0xd9b/0xda0 fs/btrfs/transaction.c:278 Read of size 4 at addr ffff888011839024 by task kworker/u4:9/1128 CPU: 0 UID: 0 PID: 1128 Comm: kworker/u4:9 Not tainted 6.13.0-rc7-syzkaller-00019-gc45323b7560e #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Workqueue: events_unbound btrfs_async_reclaim_data_space Call Trace: <TASK> __dump_stack lib/ dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report +0x143/0x180 mm/kasan/report.c:602 join_transaction+0xd9b/0xda0 fs/btrfs/transaction.c:278 start_transaction+0xaf8/0x1670 fs/btrfs/transaction.c:697 flush_space+0x448/0xcf0 fs/btrfs/space- info.c:803 btrfs_async_reclaim_data_space+0x159/0x510 fs/btrfs/space-info.c:1321 process_one_work kernel/workqueue.c:3236 [inline] process_scheduled_works+0xa66/0x1840 kernel/workqueue.c:3317 worker_thread+0x870/0xd30 kernel/workqueue.c:3398 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/ entry/entry_64.S:244 </TASK> Allocated by task 5315: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/ common.c:377 [inline] __kasan_kmalloc+0x98/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/ linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x243/0x390 mm/slub.c:4329 kmalloc_noprof include/linux/slab.h:901 [inline] join_transaction+0x144/0xda0 fs/btrfs/transaction.c:308 start_transaction+0xaf8/0x1670 fs/btrfs/transaction.c:697 btrfs_create_common+0x1b2/0x2e0 fs/btrfs/inode.c:6572 lookup_open fs/namei.c:3649 [inline] open_last_lookups fs/namei.c:3748 [inline] path_openat+0x1c03/0x3590 fs/namei.c:3984 do_filp_open+0x27f/0x4e0 fs/namei.c:4014 do_sys_openat2+0x13e/0x1d0 fs/open.c:1402 do_sys_open fs/open.c:1417 [inline] __do_sys_creat fs/open.c:1495 [inline] __se_sys_creat fs/open.c:1489 [inline] __x64_sys_creat+0x123/0x170 fs/ open.c:1489 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/ x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 5336: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/ common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:582 poison_slab_object mm/ kasan/common.c:247 [inline] __kasan_slab_free+0x59/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2353 [inline] slab_free mm/slub.c:4613 [inline] kfree+0x196/0x430 mm/slub.c:4761 cleanup_transaction fs/btrfs/transaction.c:2063 [inline] btrfs_commit_transaction+0x2c97/0x3720 fs/btrfs/transaction.c:2598 insert_balance_item +0x1284/0x20b0 fs/btrfs/volumes.c:3757 btrfs_balance+0x992/ ---truncated---</pre>

CVE ID	Description
CVE-2025-21754	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix assertion failure when splitting ordered extent after transaction abort If while we are doing a direct IO write a transaction abort happens, we mark all existing ordered extents with the BTRFS_ORDERED_IOERR flag (done at <code>btrfs_destroy_ordered_extents()</code>), and then after that if we enter <code>btrfs_split_ordered_extent()</code> and the ordered extent has bytes left (meaning we have a bio that doesn't cover the whole ordered extent, see details at <code>btrfs_extract_ordered_extent()</code>), we will fail on the following assertion at <code>btrfs_split_ordered_extent()</code>: <code>ASSERT(!(flags & ~BTRFS_ORDERED_TYPE_FLAGS))</code>; because the BTRFS_ORDERED_IOERR flag is set and the definition of BTRFS_ORDERED_TYPE_FLAGS is just the union of all flags that identify the type of write (regular, nocow, prealloc, compressed, direct IO, encoded). Fix this by returning an error from <code>btrfs_extract_ordered_extent()</code> if we find the BTRFS_ORDERED_IOERR flag in the ordered extent. The error will be the error that resulted in the transaction abort or -EIO if no transaction abort happened. This was recently reported by syzbot with the following trace: FAULT_INJECTION: forcing a failure. name failslab, interval 1, probability 0, space 0, times 1 CPU: 0 UID: 0 PID: 5321 Comm: syz.0.0 Not tainted 6.13.0-rc5-syzkaller #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 fail_dump lib/fault-inject.c:53 [inline] should_fail_ex+0x3b0/0x4e0 lib/fault-inject.c:154 should_failslab+0xac/0x100 mm/failslab.c:46 slab_pre_alloc_hook mm/slab.c:4072 [inline] slab_alloc_node mm/slab.c:4148 [inline] __do_kmalloc_node mm/slab.c:4297 [inline] __kmalloc_noprof+0xdd/0x4c0 mm/slab.c:4310 kmalloc_noprof include/linux/slab.h:905 [inline] kzalloc_noprof include/linux/slab.h:1037 [inline] btrfs_chunk_alloc_add_chunk_item+0x244/0x1100 fs/btrfs/volumes.c:5742 reserve_chunk_space+0x1ca/0x2c0 fs/btrfs/block-group.c:4292 check_system_chunk fs/btrfs/block-group.c:4319 [inline] do_chunk_alloc fs/btrfs/block-group.c:3891 [inline] btrfs_chunk_alloc+0x77b/0xf80 fs/btrfs/block-group.c:4187 find_free_extent_update_loop fs/btrfs/extent-tree.c:4166 [inline] find_free_extent+0x42d1/0x5810 fs/btrfs/extent-tree.c:4579 btrfs_reserve_extent+0x422/0x810 fs/btrfs/extent-tree.c:4672 btrfs_new_extent_direct fs/btrfs/direct-io.c:186 [inline] btrfs_get_blocks_direct_write+0x706/0xfa0 fs/btrfs/direct-io.c:321 btrfs_dio_iomap_begin+0xbb7/0x1180 fs/btrfs/direct-io.c:525 iomap_iter+0x697/0xf60 fs/iomap/iter.c:90 __iomap_dio_rw+0xeb9/0x25b0 fs/iomap/direct-io.c:702 btrfs_dio_write fs/btrfs/direct-io.c:775 [inline] btrfs_direct_write+0x610/0xa30 fs/btrfs/direct-io.c:880 btrfs_do_write_iter+0x2a0/0x760 fs/btrfs/file.c:1397 do_iter_readv_writev+0x600/0x880 vfs_writev+0x376/0xba0 fs/read_write.c:1050 do_pwritev fs/read_write.c:1146 [inline] __do_sys_pwritev2 fs/read_write.c:1204 [inline] __se_sys_pwritev2+0x196/0x2b0 fs/read_write.c:1195 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f1281f85d29 RSP: 002b:00007f12819fe038 EFLAGS: 00000246 ORIG_RAX: 0000000000000148 RAX: ffffffffda RBX: 00007f1282176080 RCX: 00007f1281f85d29 RDX: 0000000000000001 RSI: 0000000020000240 RDI: 0000000000000005 RBP: 00007f12819fe090 R08: 0000000000000000 R09: 0000000000000003 R10: 0000000000007000 R11: 0000000000000246 R12: 0000000000000002 R13: 0000000000000000 R14: 00007f1282176080 R15: 00007fcb9e23328 </TASK> BTRFS error (device loop0 state A): Transaction aborted (error -12) BTRFS: error (device loop0 state A ---truncated---

CVE ID	Description
CVE-2025-21756	In the Linux kernel, the following vulnerability has been resolved: vsock: Keep the binding until socket destruction Preserve sockets bindings; this includes both resulting from an explicit bind() and those implicitly bound through autobind during connect(). Prevents socket unbinding during a transport reassignment, which fixes a use-after-free: 1. vsock_create() (refcnt=1) calls vsock_insert_unbound() (refcnt=2) 2. transport->release() calls vsock_remove_bound() without checking if sk was bound and moved to bound list (refcnt=1) 3. vsock_bind() assumes sk is in unbound list and before __vsock_insert_bound(vsock_bound_sockets()) calls __vsock_remove_bound() which does: list_del_init(&vsk->bound_table); // nop sock_put(&vsk->sk); // refcnt=0 BUG: KASAN: slab-use-after-free in __vsock_bind+0x62e/0x730 Read of size 4 at addr ffff88816b46a74c by task a.out/2057 dump_stack_lvl+0x68/0x90 print_report+0x174/0x4f6 kasan_report+0xb9/0x190 __vsock_bind+0x62e/0x730 vsock_bind+0x97/0xe0 __sys_bind+0x154/0x1f0 __x64_sys_bind+0x6e/0xb0 do_syscall_64+0x93/0x1b0 entry_SYSCALL_64_after_hwframe+0x76/0x7e Allocated by task 2057: kasan_save_stack+0x1e/0x40 kasan_save_track+0x10/0x30 __kasan_slab_alloc+0x85/0x90 kmem_cache_alloc_noprof+0x131/0x450 sk_prot_alloc+0x5b/0x220 sk_alloc+0x2c/0x870 __vsock_create.constprop.0+0x2e/0xb60 vsock_create+0xe4/0x420 __sock_create+0x241/0x650 __sys_socket+0xf2/0x1a0 __x64_sys_socket+0x6e/0xb0 do_syscall_64+0x93/0x1b0 entry_SYSCALL_64_after_hwframe+0x76/0x7e Freed by task 2057: kasan_save_stack+0x1e/0x40 kasan_save_track+0x10/0x30 kasan_save_free_info+0x37/0x60 __kasan_slab_free+0x4b/0x70 kmem_cache_free+0x1a1/0x590 __sk_destruct+0x388/0x5a0 __vsock_bind+0x5e1/0x730 vsock_bind+0x97/0xe0 __sys_bind+0x154/0x1f0 __x64_sys_bind+0x6e/0xb0 do_syscall_64+0x93/0x1b0 entry_SYSCALL_64_after_hwframe+0x76/0x7e refcount_t: addition on 0; use-after-free. WARNING: CPU: 7 PID: 2057 at lib/refcount.c:25 refcount_warn_saturate+0xce/0x150 RIP: 0010:refcount_warn_saturate+0xce/0x150 __vsock_bind+0x66d/0x730 vsock_bind+0x97/0xe0 __sys_bind+0x154/0x1f0 __x64_sys_bind+0x6e/0xb0 do_syscall_64+0x93/0x1b0 entry_SYSCALL_64_after_hwframe+0x76/0x7e refcount_t: underflow; use-after-free. WARNING: CPU: 7 PID: 2057 at lib/refcount.c:28 refcount_warn_saturate+0xee/0x150 RIP: 0010:refcount_warn_saturate+0xee/0x150 vsock_remove_bound+0x187/0x1e0 __vsock_release+0x383/0x4a0 vsock_release+0x90/0x120 __sock_release+0xa3/0x250 sock_close+0x14/0x20 __fput+0x359/0xa80 task_work_run+0x107/0x1d0 do_exit+0x847/0x2560 do_group_exit+0xb8/0x250 __x64_sys_exit_group+0x3a/0x50 x64_sys_call+0xfec/0x14f0 do_syscall_64+0x93/0x1b0 entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2025-21758	In the Linux kernel, the following vulnerability has been resolved: ipv6: mcast: add RCU protection to mld_newpack() mld_newpack() can be called without RTNL or RCU being held. Note that we no longer can use sock_alloc_send_skb() because ipv6.igmp_sk uses GFP_KERNEL allocations which can sleep. Instead use alloc_skb() and charge the net->ipv6.igmp_sk socket under RCU protection.
CVE-2025-21759	In the Linux kernel, the following vulnerability has been resolved: ipv6: mcast: extend RCU protection in igmp6_send() igmp6_send() can be called without RTNL or RCU being held. Extend RCU protection so that we can safely fetch the net pointer and avoid a potential UAF. Note that we no longer can use sock_alloc_send_skb() because ipv6.igmp_sk uses GFP_KERNEL allocations which can sleep. Instead use alloc_skb() and charge the net->ipv6.igmp_sk socket under RCU protection.
CVE-2025-2176	A vulnerability classified as critical has been found in libzvb up to 0.2.43. This affects the function vbi_capture_sim_load_caption of the file src/io-sim.c. The manipulation leads to integer overflow. It is possible to initiate the attack remotely. The exploit has been disclosed to the public and may be used. Upgrading to version 0.2.44 is able to address this issue. The identifier of the patch is ca1672134b3e2962cd392212c73f44f8f4cb489f. It is recommended to upgrade the affected component. The code maintainer was informed beforehand about the issues. She reacted very fast and highly professional.
CVE-2025-21760	In the Linux kernel, the following vulnerability has been resolved: ndisc: extend RCU protection in ndisc_send_skb() ndisc_send_skb() can be called without RTNL or RCU held. Acquire rcu_read_lock() earlier, so that we can use dev_net_rcu() and avoid a potential UAF.
CVE-2025-21761	In the Linux kernel, the following vulnerability has been resolved: openvswitch: use RCU protection in ovs_vport_cmd_fill_info() ovs_vport_cmd_fill_info() can be called without RTNL or RCU. Use RCU protection and dev_net_rcu() to avoid potential UAF.
CVE-2025-21762	In the Linux kernel, the following vulnerability has been resolved: arp: use RCU protection in arp_xmit() arp_xmit() can be called without RTNL or RCU protection. Use RCU protection to avoid potential UAF.
CVE-2025-21763	In the Linux kernel, the following vulnerability has been resolved: neighbour: use RCU protection in __neigh_notify() __neigh_notify() can be called without RTNL or RCU protection. Use RCU protection to avoid potential UAF.
CVE-2025-21764	In the Linux kernel, the following vulnerability has been resolved: ndisc: use RCU protection in ndisc_alloc_skb() ndisc_alloc_skb() can be called without RTNL or RCU being held. Add RCU protection to avoid possible UAF.

CVE ID	Description
CVE-2025-21765	In the Linux kernel, the following vulnerability has been resolved: ipv6: use RCU protection in ip6_default_advmss() ip6_default_advmss() needs rcu protection to make sure the net structure it reads does not disappear.
CVE-2025-21766	In the Linux kernel, the following vulnerability has been resolved: ipv4: use RCU protection in __ip_rt_update_pmtu() __ip_rt_update_pmtu() must use RCU protection to make sure the net structure it reads does not disappear.
CVE-2025-21767	In the Linux kernel, the following vulnerability has been resolved: clocksource: Use migrate_disable() to avoid calling get_random_u32() in atomic context The following bug report happened with a PREEMPT_RT kernel: BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 2012, name: kwatchdog preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 get_random_u32+0x4f/0x110 clocksource_verify_choose_cpus+0xab/0x1a0 clocksource_verify_percpu.part.0+0x6b/0x330 clocksource_watchdog_kthread+0x193/0x1a0 It is due to the fact that clocksource_verify_choose_cpus() is invoked with preemption disabled. This function invokes get_random_u32() to obtain random numbers for choosing CPUs. The batched_entropy_32 local lock and/or the base_crng.lock spinlock in driver/char/random.c will be acquired during the call. In PREEMPT_RT kernel, they are both sleeping locks and so cannot be acquired in atomic context. Fix this problem by using migrate_disable() to allow smp_processor_id() to be reliably used without introducing atomic context. preempt_disable() is then called after clocksource_verify_choose_cpus() but before the clocksource measurement is being run to avoid introducing unexpected latency.
CVE-2025-21768	In the Linux kernel, the following vulnerability has been resolved: net: ipv6: fix dst ref loops in rpl, seg6 and ioam6 lwtunnels Some lwtunnels have a dst cache for post-transformation dst. If the packet destination did not change we may end up recording a reference to the lwtunnel in its own cache, and the lwtunnel state will never be freed. Discovered by the ioam6.sh test, kmemleak was recently fixed to catch per-cpu memory leaks. I'm not sure if rpl and seg6 can actually hit this, but in principle I don't see why not.
CVE-2025-2177	A vulnerability classified as critical was found in libzvb up to 0.2.43. This vulnerability affects the function vbi_search_new of the file src/search.c. The manipulation of the argument pat_len leads to integer overflow. The attack can be initiated remotely. The exploit has been disclosed to the public and may be used. Upgrading to version 0.2.44 is able to address this issue. The patch is identified as ca1672134b3e2962cd392212c73f44f8f4cb489f. It is recommended to upgrade the affected component. The code maintainer was informed beforehand about the issues. She reacted very fast and highly professional.
CVE-2025-21772	In the Linux kernel, the following vulnerability has been resolved: partitions: mac: fix handling of bogus partition table Fix several issues in partition probing: - The bailout for a bad partoffset must use put_dev_sector(), since the preceding read_part_sector() succeeded. - If the partition table claims a silly sector size like 0xffff bytes (which results in partition table entries straddling sector boundaries), bail out instead of accessing out-of-bounds memory. - We must not assume that the partition table contains proper NUL termination - use strlen() and strncmp() instead of strlen() and strcmp().
CVE-2025-21773	In the Linux kernel, the following vulnerability has been resolved: can: etas_es58x: fix potential NULL pointer dereference on udev->serial The driver assumed that es58x_dev->udev->serial could never be NULL. While this is true on commercially available devices, an attacker could spoof the device identity providing a NULL USB serial number. That would trigger a NULL pointer dereference. Add a check on es58x_dev->udev->serial before accessing it.
CVE-2025-21775	In the Linux kernel, the following vulnerability has been resolved: can: ctucanfd: handle skb allocation failure If skb allocation fails, the pointer to struct can_frame is NULL. This is actually handled everywhere inside ctucan_err_interrupt() except for the only place. Add the missed NULL check. Found by Linux Verification Center (linuxtesting.org) with SVACE static analysis tool.

CVE ID	Description
CVE-2025-21776	In the Linux kernel, the following vulnerability has been resolved: USB: hub: Ignore non-compliant devices with too many configs or interfaces Robert Morris created a test program which can cause <code>usb_hub_to_struct_hub()</code> to dereference a NULL or inappropriate pointer: Oops: general protection fault, probably for non-canonical address 0xffffffffcccccc: 0000 [#1] SMP DEBUG_PAGEALLOC PTI CPU: 7 UID: 0 PID: 117 Comm: kworker/7:1 Not tainted 6.13.0-rc3-00017-gf44d154d6e3d #14 Hardware name: FreeBSD BHYVE/BHYVE, BIOS 14.0 10/17/2021 Workqueue: usb_hub_wq hub_event RIP: 0010:usb_hub_adjust_devicemovable+0x78/0x110 ... Call Trace: <TASK> ? die_addr+0x31/0x80 ? exc_general_protection+0x1b4/0x3c0 ? asm_exc_general_protection+0x26/0x30 ? usb_hub_adjust_devicemovable+0x78/0x110 hub_probe+0x7c7/0xab0 usb_probe_interface+0x14b/0x350 really_probe+0xd0/0x2d0 ? __pfx__device_attach_driver+0x10/0x10 __driver_probe_device+0x6e/0x110 driver_probe_device+0x1a/0x90 __device_attach_driver+0x7e/0xc0 bus_for_each_drv+0x7f/0xd0 __device_attach+0xaa/0x1a0 bus_probe_device+0x8b/0xa0 device_add+0x62e/0x810 usb_set_configuration+0x65d/0x990 usb_generic_driver_probe+0x4b/0x70 usb_probe_device+0x36/0xd0 The cause of this error is that the device has two interfaces, and the hub driver binds to interface 1 instead of interface 0, which is where <code>usb_hub_to_struct_hub()</code> looks. We can prevent the problem from occurring by refusing to accept hub devices that violate the USB spec by having more than one configuration or interface.
CVE-2025-21779	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Reject Hyper-V's <code>SEND_IPI</code> hypercalls if local APIC isn't in-kernel Advertise support for Hyper-V's <code>SEND_IPI</code> and <code>SEND_IPI_EX</code> hypercalls if and only if the local API is emulated/virtualized by KVM, and explicitly reject said hypercalls if the local APIC is emulated in userspace, i.e. don't rely on userspace to opt-in to <code>KVM_CAP_HYPERV_ENFORCE_CPUID</code> . Rejecting <code>SEND_IPI</code> and <code>SEND_IPI_EX</code> fixes a NULL-pointer dereference if Hyper-V enlightenments are exposed to the guest without an in-kernel local APIC: <code>dump_stack+0xbe/0xfd __kasan_report.cold+0x34/0x84 kasan_report+0x3a/0x50 __apic_accept_irq+0x3a/0x5c0 kvm_hv_send_ipi.isra.0+0x34e/0x820 kvm_hv_hypercall+0x8d9/0x9d0 kvm_emulate_hypercall+0x506/0x7e0 __vmx_handle_exit+0x283/0xb60 vmx_handle_exit+0x1d/0xd0 vcpu_enter_guest+0x16b0/0x24c0 vcpu_run+0xc0/0x550 kvm_arch_vcpu_ioctl_run+0x170/0x6d0 kvm_vcpu_ioctl+0x413/0xb20 __se_sys_ioctl+0x111/0x160 do_syscall_64+0x30/0x40 entry_SYSCALL_64_after_hwframe+0x67/0xd1 Note, checking the sending vCPU is sufficient, as the per-VM <code>irqchip_mode</code> can't be modified after vCPUs are created, i.e. if one vCPU has an in-kernel local APIC, then all vCPUs have an in-kernel local APIC.</code>
CVE-2025-21780	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: avoid buffer overflow attach in <code>smu_sys_set_pp_table()</code> It malicious user provides a small pptable through <code>sysfs</code> and then a bigger pptable, it may cause buffer overflow attack in function <code>smu_sys_set_pp_table()</code> .
CVE-2025-21781	In the Linux kernel, the following vulnerability has been resolved: batman-adv: fix panic during interface removal Reference counting is used to ensure that <code>batadv_hardif_neigh_node</code> and <code>batadv_hard_iface</code> are not freed before/during <code>batadv_v_elp_throughput_metric_update</code> work is finished. But there isn't a guarantee that the hard if will remain associated with a soft interface up until the work is finished. This fixes a crash triggered by reboot that looks like this: Call trace: <code>batadv_v_mesh_free+0xd0/0x4dc [batman_adv] batadv_v_elp_throughput_metric_update+0x1c/0xa4 process_one_work+0x178/0x398 worker_thread+0x2e8/0x4d0 kthread+0xd8/0xdc ret_from_fork+0x10/0x20</code> (the <code>batadv_v_mesh_free</code> call is misleading, and does not actually happen) I was able to make the issue happen more reliably by changing <code>hardif_neigh->bat_v.metric_work</code> work to be delayed work. This allowed me to track down and confirm the fix. [sven@narfation.org: prevent entering <code>batadv_v_elp_get_throughput</code> without <code>soft_iface</code>]
CVE-2025-21782	In the Linux kernel, the following vulnerability has been resolved: orangefs: fix a oob in <code>orangefs_debug_write</code> I got a syzbot report: slab-out-of-bounds Read in <code>orangefs_debug_write</code> ... several people suggested fixes, I tested Al Viro's suggestion and made this patch.
CVE-2025-21783	In the Linux kernel, the following vulnerability has been resolved: gpiolib: Fix crash on error in <code>gpiochip_get_ngpios()</code> The <code>gpiochip_get_ngpios()</code> uses <code>chip_*</code> macros to print messages. However these macros rely on <code>gpiodev</code> to be initialised and set, which is not the case when called via <code>bgpio_init()</code> . In such a case the printing messages will crash on NULL pointer dereference. Replace <code>chip_*</code> macros by the respective <code>dev_*</code> ones to avoid such crash.
CVE-2025-21784	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: bail out when failed to load fw in <code>psp_init_cap_microcode()</code> In function <code>psp_init_cap_microcode()</code> , it should bail out when failed to load firmware, otherwise it may cause invalid memory access.
CVE-2025-21785	In the Linux kernel, the following vulnerability has been resolved: arm64: cacheinfo: Avoid out-of-bounds write to <code>cacheinfo</code> array The loop that detects/populates cache information already has a bounds check on the array size but does not account for cache levels with separate data/instructions cache. Fix this by incrementing the index for any populated leaf (instead of any populated level).

CVE ID	Description
CVE-2025-21786	In the Linux kernel, the following vulnerability has been resolved: workqueue: Put the pwq after detaching the rescuer from the pool The commit 68f83057b913("workqueue: Reap workers via kthread_stop() and remove detach_completion") adds code to reap the normal workers but mistakenly does not handle the rescuer and also removes the code waiting for the rescuer in put_unbound_pool(), which caused a use-after-free bug reported by Cheung Wall. To avoid the use-after-free bug, the pool's reference must be held until the detachment is complete. Therefore, move the code that puts the pwq after detaching the rescuer from the pool.
CVE-2025-21787	In the Linux kernel, the following vulnerability has been resolved: team: better TEAM_OPTION_TYPE_STRING validation syzbot reported following splat [1] Make sure user-provided data contains one nul byte. [1] BUG: KMSAN: uninit-value in string_nocheck lib/vsprintf.c:633 [inline] BUG: KMSAN: uninit-value in string+0x3ec/0x5f0 lib/vsprintf.c:714 string_nocheck lib/vsprintf.c:633 [inline] string+0x3ec/0x5f0 lib/vsprintf.c:714 vsnprintf+0xa5d/0x1960 lib/vsprintf.c:2843 __request_module+0x252/0x9f0 kernel/module/kmod.c:149 team_mode_get drivers/net/team/team_core.c:480 [inline] team_change_mode drivers/net/team/team_core.c:607 [inline] team_mode_option_set+0x437/0x970 drivers/net/team/team_core.c:1401 team_option_set drivers/net/team/team_core.c:375 [inline] team_nl_options_set_doit+0x1339/0x1f90 drivers/net/team/team_core.c:2662 genl_family_rcv_msg_doit net/netlink/genetlink.c:1115 [inline] genl_family_rcv_msg net/netlink/genetlink.c:1195 [inline] genl_rcv_msg+0x1214/0x12c0 net/netlink/genetlink.c:1210 netlink_rcv_skb+0x375/0x650 net/netlink/af_netlink.c:2543 genl_rcv+0x40/0x60 net/netlink/genetlink.c:1219 netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline] netlink_unicast+0xf52/0x1260 net/netlink/af_netlink.c:1348 netlink_sendmsg+0x10da/0x11e0 net/netlink/af_netlink.c:1892 sock_sendmsg_nosec net/socket.c:718 [inline] __sock_sendmsg+0x30f/0x380 net/socket.c:733 __sys_sendmsg+0x877/0xb60 net/socket.c:2573 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2627 __sys_sendmsg net/socket.c:2659 [inline] __do_sys_sendmsg net/socket.c:2664 [inline] __se_sys_sendmsg net/socket.c:2662 [inline] __x64_sys_sendmsg+0x212/0x3c0 net/socket.c:2662 x64_sys_call+0x2ed6/0x3c30 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2025-21790	In the Linux kernel, the following vulnerability has been resolved: vxlan: check vxlan_vnigroup_init() return value vxlan_init() must check vxlan_vnigroup_init() success otherwise a crash happens later, spotted by syzbot. Oops: general protection fault, probably for non-canonical address 0xdfc0000000002c: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000160-0x0000000000000167] CPU: 0 UID: 0 PID: 7313 Comm: syz-executor147 Not tainted 6.14.0-rc1-syzkaller-00276-g69b54314c975 #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian.1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:vxlan_vnigroup_uninit+0x89/0x500 drivers/net/vxlan/vxlan_vnifilter.c:912 Code: 00 48 8b 44 24 08 4c 8b b0 98 41 00 00 49 8d 86 60 01 00 00 48 89 c2 48 89 44 24 10 48 b8 00 00 00 00 fc ff df 48 c1 ea 03 <80> 3c 02 00 0f 85 4d 04 00 00 49 8b 86 60 01 00 00 48 ba 00 00 00 RSP: 0018:ffffc9000cc1eea8 EFLAGS: 00010202 RAX: dffffc0000000000 RBX: 0000000000000001 RCX: ffffffff8672effb RDX: 0000000000000002c RSI: ffffffff8672ecb9 RDI: ffff8880461b4f18 RBP: ffff8880461b4ef4 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000020000 R13: ffff8880461b0d80 R14: 0000000000000000 R15: dffffc0000000000 FS: 00007fecfa95d6c0(0000) GS:ffff88806a600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fecfa95cfb8 CR3: 000000004472c000 CR4: 0000000000352ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> vxlan_uninit+0x1ab/0x200 drivers/net/vxlan/vxlan_core.c:2942 unregister_netdevice_many_notify+0x12d6/0x1f30 net/core/dev.c:11824 unregister_netdevice_many net/core/dev.c:11866 [inline] unregister_netdevice_queue+0x307/0x3f0 net/core/dev.c:11736 register_netdevice+0x1829/0x1eb0 net/core/dev.c:10901 __vxlan_dev_create+0x7c6/0xa30 drivers/net/vxlan/vxlan_core.c:3981 vxlan_newlink+0xd1/0x130 drivers/net/vxlan/vxlan_core.c:4407 rtnl_newlink_create net/core/rtnetlink.c:3795 [inline] __rtnl_newlink net/core/rtnetlink.c:3906 [inline]
CVE-2025-21791	In the Linux kernel, the following vulnerability has been resolved: vrf: use RCU protection in l3mdev_l3_out() l3mdev_l3_out() can be called without RCU being held: raw_sendmsg() ip_push_pending_frames() ip_send_skb() ip_local_out() __ip_local_out() l3mdev_ip_out() Add rcu_read_lock() / rcu_read_unlock() pair to avoid a potential UAF.

CVE ID	Description
CVE-2025-21792	In the Linux kernel, the following vulnerability has been resolved: ax25: Fix refcount leak caused by setting SO_BINDTODEVICE sockopt If an AX25 device is bound to a socket by setting the SO_BINDTODEVICE socket option, a refcount leak will occur in ax25_release(). Commit 9fd75b66b8f6 ("ax25: Fix refcount leaks caused by ax25_cb_del()") added decrement of device refcounts in ax25_release(). In order for that to work correctly the refcounts must already be incremented when the device is bound to the socket. An AX25 device can be bound to a socket by either calling ax25_bind() or setting SO_BINDTODEVICE socket option. In both cases the refcounts should be incremented, but in fact it is done only in ax25_bind(). This bug leads to the following issue reported by Syzkaller: ===== <pre> refcount_t: decrement hit 0; leaking memory. WARNING: CPU: 1 PID: 5932 at lib/refcount.c:31 refcount_warn_saturate+0x1ed/0x210 lib/refcount.c:31 Modules linked in: CPU: 1 UID: 0 PID: 5932 Comm: syz-executor424 Not tainted 6.13.0-rc4-syzkaller-00110-g4099a71718b0 #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:refcount_warn_saturate+0x1ed/0x210 lib/refcount.c:31 Call Trace: <TASK> __refcount_dec include/linux/refcount.h:336 [inline] refcount_dec include/linux/ refcount.h:351 [inline] ref_tracker_free+0x710/0x820 lib/ref_tracker.c:236 netdev_tracker_free include/linux/netdevice.h:4156 [inline] netdev_put include/linux/netdevice.h:4173 [inline] netdev_put include/linux/netdevice.h:4169 [inline] ax25_release+0x33f/0xa10 net/ax25/ af_ax25.c:1069 __sock_release+0xb0/0x270 net/socket.c:640 sock_close+0x1c/0x30 net/ socket.c:1408 ... do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... <TASK> ===== Fix the implementation of ax25_setsockopt() by adding increment of refcounts for the new device bound, and decrement of refcounts for the old unbound device.</pre>
CVE-2025-21793	In the Linux kernel, the following vulnerability has been resolved: spi: sn-f-osp: Fix division by zero When there is no dummy cycle in the spi-nor commands, both dummy bus cycle bytes and width are zero. Because of the cpu's warning when divided by zero, the warning should be avoided. Return just zero to avoid such calculations.
CVE-2025-21795	In the Linux kernel, the following vulnerability has been resolved: NFSD: fix hang in nfsd4_shutdown_callback If nfs4_client is in courtesy state then there is no point to send the callback. This causes nfsd4_shutdown_callback to hang since cl_cb_inflight is not 0. This hang lasts about 15 minutes until TCP notifies NFSD that the connection was dropped. This patch modifies nfsd4_run_cb_work to skip the RPC call if nfs4_client is in courtesy state.
CVE-2025-21796	In the Linux kernel, the following vulnerability has been resolved: nfsd: clear acl_access/acl_default after releasing them If getting acl_default fails, acl_access and acl_default will be released simultaneously. However, acl_access will still retain a pointer pointing to the released posix_acl, which will trigger a WARNING in nfs3svc_release_getacl like this: -----[cut here]----- <pre> refcount_t: underflow; use-after-free. WARNING: CPU: 26 PID: 3199 at lib/refcount.c:28 refcount_warn_saturate+0xb5/0x170 Modules linked in: CPU: 26 UID: 0 PID: 3199 Comm: nfsd Not tainted 6.12.0-rc6-00079-g04ae226af01f-dirty #8 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 RIP: 0010:refcount_warn_saturate+0xb5/0x170 Code: cc cc 0f b6 1d b3 20 a5 03 80 fb 01 0f 87 65 48 d8 00 83 e3 01 75 e4 48 c7 c7 c0 3b 9b 85 c6 05 97 20 a5 03 01 e8 fb 3e 30 ff <0f> 0b eb cd 0f b6 1d 8a3 RSP: 0018:ffff90008637cd8 EFLAGS: 00010282 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffffff83904fde RDX: dffffc0000000000 RSI: 0000000000000008 RDI: ffff88871ed36380 RBP: ffff888158beeb40 R08: 0000000000000001 R09: fffff520010c6f56 R10: ffff90008637ab7 R11: 0000000000000001 R12: 0000000000000001 R13: ffff888140e77400 R14: ffff888140e77408 R15: ffffffff858b42c0 FS: 0000000000000000(0000) GS:ffff88871ed00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000562384d32158 CR3: 0000000055cc6a000 CR4: 0000000000000060 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? refcount_warn_saturate+0xb5/0x170 ? __warn+0xa5/0x140 ? refcount_warn_saturate+0xb5/0x170 ? report_bug+0x1b1/0x1e0 ? handle_bug+0x53/0xa0 ? exc_invalid_op+0x17/0x40 ? asm_exc_invalid_op +0x1a/0x20 ? tick_nohz_tick_stopped+0x1e/0x40 ? refcount_warn_saturate+0xb5/0x170 ? refcount_warn_saturate+0xb5/0x170 nfs3svc_release_getacl+0xc9/0xe0 svc_process_common +0x5db/0xb60 ? __pfx_svc_process_common+0x10/0x10 ? __rcu_read_unlock+0x69/0xa0 ? __pfx_nfsd_dispatch+0x10/0x10 ? svc_xprt_received+0xa1/0x120 ? xdr_init_decode+0x11d/0x190 svc_process+0x2a7/0x330 svc_handle_xprt+0x69d/0x940 svc_recv+0x180/0x2d0 nfsd+0x168/0x200 ? __pfx_nfsd+0x10/0x10 kthread+0x1a2/0x1e0 ? kthread+0xf4/0x1e0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x34/0x60 ? __pfx_kthread+0x10/0x10 ret_from_asm+0x1a/0x30 </TASK> Kernel panic - not syncing: kernel: panic_on_warn set ... Clear acl_access/acl_default after posix_acl_release is called to prevent UAF from being triggered.</pre>
CVE-2025-21798	In the Linux kernel, the following vulnerability has been resolved: firewire: test: Fix potential null dereference in firewire kunit test kunit_kzalloc() may return a NULL pointer, dereferencing it without NULL check may lead to NULL dereference. Add a NULL check for test_state.

CVE ID	Description
CVE-2025-21799	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: ti: am65-cpsw: fix freeing IRQ in am65_cpsw_nuss_remove_tx_chns() When getting the IRQ we use k3_udma_glue_tx_get_irq() which returns negative error value on error. So not NULL check is not sufficient to determine if IRQ is valid. Check that IRQ is greater than zero to ensure it is valid. There is no issue at probe time but at runtime user can invoke .set_channels which results in the following call chain. am65_cpsw_set_channels() am65_cpsw_nuss_update_tx_rx_chns() am65_cpsw_nuss_remove_tx_chns() am65_cpsw_nuss_init_tx_chns() At this point if am65_cpsw_nuss_init_tx_chns() fails due to k3_udma_glue_tx_get_irq() then tx_chn->irq will be set to a negative value. Then, at subsequent .set_channels with higher channel count we will attempt to free an invalid IRQ in am65_cpsw_nuss_remove_tx_chns() leading to a kernel warning. The issue is present in the original commit that introduced this driver, although there, am65_cpsw_nuss_update_tx_rx_chns() existed as am65_cpsw_nuss_update_tx_chns().
CVE-2025-21801	In the Linux kernel, the following vulnerability has been resolved: net: ravb: Fix missing rtnl lock in suspend/resume path Fix the suspend/resume path by ensuring the rtnl lock is held where required. Calls to ravb_open, ravb_close and wol operations must be performed under the rtnl lock to prevent conflicts with ongoing ndo operations. Without this fix, the following warning is triggered: [39.032969] ===== [39.032983] WARNING: suspicious RCU usage [39.033019] ----- [39.033033] drivers/net/phy/phy_device.c:2004 suspicious rcu_dereference_protected() usage! ... [39.033597] stack backtrace: [39.033613] CPU: 0 UID: 0 PID: 174 Comm: python3 Not tainted 6.13.0-rc7-next-20250116-arm64-renesas-00002-g35245dfdc62c #7 [39.033623] Hardware name: Renesas SMARC EVK version 2 based on r9a08g045s33 (DT) [39.033628] Call trace: [39.033633] show_stack+0x14/0x1c (C) [39.033652] dump_stack_lvl+0xb4/0xc4 [39.033664] dump_stack+0x14/0x1c [39.033671] lockdep_rcu_suspicious+0x16c/0x22c [39.033682] phy_detach+0x160/0x190 [39.033694] phy_disconnect+0x40/0x54 [39.033703] ravb_close+0x6c/0x1cc [39.033714] ravb_suspend+0x48/0x120 [39.033721] dpm_run_callback+0x4c/0x14c [39.033731] device_suspend+0x11c/0x4dc [39.033740] dpm_suspend+0xdc/0x214 [39.033748] dpm_suspend_start+0x48/0x60 [39.033758] suspend_devices_and_enter+0x124/0x574 [39.033769] pm_suspend+0x1ac/0x274 [39.033778] state_store+0x88/0x124 [39.033788] kobj_attr_store+0x14/0x24 [39.033798] sysfs_kf_write+0x48/0x6c [39.033808] kernfs_fop_write_iter+0x118/0x1a8 [39.033817] vfs_write+0x27c/0x378 [39.033825] ksys_write+0x64/0xf4 [39.033833] __arm64_sys_write+0x18/0x20 [39.033841] invoke_syscall+0x44/0x104 [39.033852] el0_svc_common.constprop.0+0xb4/0xd4 [39.033862] do_el0_svc+0x18/0x20 [39.033870] el0_svc+0x3c/0xf0 [39.033880] el0t_64_sync_handler+0xc0/0xc4 [39.033888] el0t_64_sync+0x154/0x158 [39.041274] ravb 11c30000.ethernet eth0: Link is Down
CVE-2025-21802	In the Linux kernel, the following vulnerability has been resolved: net: hns3: fix oops when unload drivers paralleling When unload hclge driver, it tries to disable sriov first for each ae_dev node from hnae3_ae_dev_list. If user unloads hns3 driver at the time, because it removes all the ae_dev nodes, and it may cause oops. But we can't simply use hnae3_common_lock for this. Because in the process flow of pci_disable_sriov(), it will trigger the remove flow of VF, which will also take hnae3_common_lock. To fix it, introduce a new mutex to protect the unload process.
CVE-2025-21804	In the Linux kernel, the following vulnerability has been resolved: PCI: rcar-ep: Fix incorrect variable used when calling devm_request_mem_region() The rcar_pcie_parse_outbound_ranges() uses the devm_request_mem_region() macro to request a needed resource. A string variable that lives on the stack is then used to store a dynamically computed resource name, which is then passed on as one of the macro arguments. This can lead to undefined behavior. Depending on the current contents of the memory, the manifestations of errors may vary. One possible output may be as follows: \$ cat /proc/iomem 30000000-37ffffff : 38000000-3ffffff : Sometimes, garbage may appear after the colon. In very rare cases, if no NULL-terminator is found in memory, the system might crash because the string iterator will overrun which can lead to access of unmapped memory above the stack. Thus, fix this by replacing outbound_name with the name of the previously requested resource. With the changes applied, the output will be as follows: \$ cat /proc/iomem 30000000-37ffffff : memory2 38000000-3ffffff : memory3 [kwilczynski: commit log]

CVE ID	Description
CVE-2025-21806	In the Linux kernel, the following vulnerability has been resolved: net: let net.core.dev_weight always be non-zero The following problem was encountered during stability test: (NULL net_device): NAPI poll function process_backlog+0x0/0x530 \ returned 1, exceeding its budget of 0. -----[cut here]----- list_add double add: new=ffff88905f746f48, prev=ffff88905f746f48, \ next=ffff88905f746e40. WARNING: CPU: 18 PID: 5462 at lib/list_debug.c:35 \ __list_add_valid_or_report+0xf3/0x130 CPU: 18 UID: 0 PID: 5462 Comm: ping Kdump: loaded Not tainted 6.13.0-rc7+ RIP: 0010: __list_add_valid_or_report+0xf3/0x130 Call Trace: ? __warn+0xcd/0x250 ? __list_add_valid_or_report+0xf3/0x130 enqueue_to_backlog+0x923/0x1070 netif_rx_internal+0x92/0x2b0 __netif_rx+0x15/0x170 loopback_xmit+0x2ef/0x450 dev_hard_start_xmit+0x103/0x490 __dev_queue_xmit+0xeac/0x1950 ip_finish_output2+0x6cc/0x1620 ip_output+0x161/0x270 ip_push_pending_frames+0x155/0x1a0 raw_sendmsg+0xe13/0x1550 __sys_sendto+0x3bf/0x4e0 __x64_sys_sendto+0xdc/0x1b0 do_syscall_64+0x5b/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e The reproduction command is as follows: sysctl -w net.core.dev_weight=0 ping 127.0.0.1 This is because when the napi's weight is set to 0, process_backlog() may return 0 and clear the NAPI_STATE_SCHEDULED bit of napi->state, causing this napi to be re-pollled in net_rx_action() until __do_softirq() times out. Since the NAPI_STATE_SCHEDULED bit has been cleared, napi_schedule_rps() can be retrigged in enqueue_to_backlog(), causing this issue. Making the napi's weight always non-zero solves this problem. Triggering this issue requires system-wide admin (setting is not namespaced).
CVE-2025-21808	In the Linux kernel, the following vulnerability has been resolved: net: xdp: Disallow attaching device-bound programs in generic mode Device-bound programs are used to support RX metadata kfuncs. These kfuncs are driver-specific and rely on the driver context to read the metadata. This means they can't work in generic XDP mode. However, there is no check to disallow such programs from being attached in generic mode, in which case the metadata kfuncs will be called in an invalid context, leading to crashes. Fix this by adding a check to disallow attaching device-bound programs in generic mode.
CVE-2025-21809	In the Linux kernel, the following vulnerability has been resolved: rxrpc, afs: Fix peer hash locking vs RCU callback In its address list, afs now retains pointers to and refs on one or more rxrpc_peer objects. The address list is freed under RCU and at this time, it puts the refs on those peers. Now, when an rxrpc_peer object runs out of refs, it gets removed from the peer hash table and, for that, rxrpc has to take a spinlock. However, it is now being called from afs's RCU cleanup, which takes place in BH context - but it is just taking an ordinary spinlock. The put may also be called from non-BH context, and so there exists the possibility of deadlock if the BH-based RCU cleanup happens whilst the hash spinlock is held. This led to the attached lockdep complaint. Fix this by changing spinlocks of rxnet->peer_hash_lock back to BH-disabling locks. ===== WARNING: inconsistent lock state 6.13.0-rc5-build2+ #1223 Tainted: G E ----- inconsistent {SOFTIRQ-ON-W} -> {IN-SOFTIRQ-W} usage. swapper/1/0 [HCO[0]:SC1[1]:HE1:SE0] takes: ffff88810babe228 (&rxnet->peer_hash_lock){+?.}-{3:3}, at: rxrpc_put_peer+0xcb/0x180 {SOFTIRQ-ON-W} state was registered at: mark_usage+0x164/0x180 __lock_acquire+0x544/0x990 lock_acquire.part.0+0x103/0x280 _raw_spin_lock+0x2f/0x40 rxrpc_peer_heartbeat_worker +0x144/0x440 process_one_work+0x486/0x7c0 process_scheduled_works+0x73/0x90 worker_thread +0x1c8/0x2a0 kthread+0x19b/0x1b0 ret_from_fork+0x24/0x40 ret_from_fork_asm+0x1a/0x30 irq event stamp: 972402 hardirqs last enabled at (972402): [<fffffff8244360e>] _raw_spin_unlock_irqrestore +0x2e/0x50 hardirqs last disabled at (972401): [<fffffff82443328>] _raw_spin_lock_irqsave +0x18/0x60 softirqs last enabled at (972300): [<fffffff810ffbbe>] handle_softirqs+0x3ee/0x430 softirqs last disabled at (972313): [<fffffff810ffc54>] __irq_exit_rcu+0x44/0x110 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(&rxnet->peer_hash_lock); <Interrupt> lock(&rxnet->peer_hash_lock); *** DEADLOCK *** 1 lock held by swapper/1/0: #0: ffffffff83576be0 (rcu_callback){....}-{0:0}, at: rcu_lock_acquire+0x7/0x30 stack backtrace: CPU: 1 UID: 0 PID: 0 Comm: swapper/1 Tainted: G E 6.13.0-rc5-build2+ #1223 Tainted: [E]=UNSIGNED_MODULE Hardware name: ASUS All Series/H97-PLUS, BIOS 2306 10/09/2014 Call Trace: <IRQ> dump_stack_lvl+0x57/0x80 print_usage_bug.part.0+0x227/0x240 valid_state+0x53/0x70 mark_lock_irq +0xa5/0x2f0 mark_lock+0xf7/0x170 mark_usage+0xe1/0x180 __lock_acquire+0x544/0x990 lock_acquire.part.0+0x103/0x280 _raw_spin_lock+0x2f/0x40 rxrpc_put_peer+0xcb/0x180 afs_free_addrlist+0x46/0x90 [kafs] rcu_do_batch+0x2d2/0x640 rcu_core+0x2f7/0x350 handle_softirqs +0x1ee/0x430 __irq_exit_rcu+0x44/0x110 irq_exit_rcu+0xa/0x30 sysvec_apic_timer_interrupt +0x7f/0xa0 </IRQ>

CVE ID	Description
CVE-2025-21810	In the Linux kernel, the following vulnerability has been resolved: driver core: class: Fix wild pointer dereferences in API class_dev_iter_next() There are a potential wild pointer dereferences issue regarding APIs class_dev_iter_init(next exit)(), as explained by below typical usage: // All members of @iter are wild pointers. struct class_dev_iter iter; // class_dev_iter_init(@iter, @class, ...) checks parameter @class for // potential class_to_subsys() error, and it returns void type and does // not initialize its output parameter @iter, so caller can not detect // the error and continues to invoke class_dev_iter_next(@iter) even if // @iter still contains wild pointers. class_dev_iter_init(&iter, ...); // Dereference these wild pointers in @iter here once suffer the error. while (dev = class_dev_iter_next(&iter)) { ... }; // Also dereference these wild pointers here. class_dev_iter_exit(&iter); Actually, all callers of these APIs have such usage pattern in kernel tree. Fix by: - Initialize output parameter @iter by memset() in class_dev_iter_init() and give callers prompt by pr_crit() for the error. - Check if @iter is valid in class_dev_iter_next().
CVE-2025-21811	In the Linux kernel, the following vulnerability has been resolved: nilfs2: protect access to buffers with no active references nilfs_lookup_dirty_data_buffers(), which iterates through the buffers attached to dirty data folios/pages, accesses the attached buffers without locking the folios/pages. For data cache, nilfs_clear_folio_dirty() may be called asynchronously when the file system degenerates to read only, so nilfs_lookup_dirty_data_buffers() still has the potential to cause use after free issues when buffers lose the protection of their dirty state midway due to this asynchronous clearing and are unintentionally freed by try_to_free_buffers(). Eliminate this race issue by adjusting the lock section in this function.
CVE-2025-21812	In the Linux kernel, the following vulnerability has been resolved: ax25: rcu protect dev->ax25_ptr syzbot found a lockdep issue [1]. We should remove ax25 RTNL dependency in ax25_setsockopt() This should also fix a variety of possible UAF in ax25. [1] WARNING: possible circular locking dependency detected 6.13.0-rc3-syzkaller-00762-g9268abe611b0 #0 Not tainted ----- syz.5.1818/12806 is trying to acquire lock: ffffffff8fcb3988 (rtnl_mutex){+.+.}-{4:4}, at: ax25_setsockopt+0xa55/0xe90 net/ax25/af_ax25.c:680 but task is already holding lock: ffff8880617ac258 (sk_lock-AF_AX25){+.+.}-{0:0}, at: lock_sock include/net/sock.h:1618 [inline] ffff8880617ac258 (sk_lock-AF_AX25){+.+.}-{0:0}, at: ax25_setsockopt+0x209/0xe90 net/ax25/af_ax25.c:574 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #1 (sk_lock-AF_AX25){+.+.}-{0:0}: lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5849 lock_sock_nested+0x48/0x100 net/core/sock.c:3642 lock_sock include/net/sock.h:1618 [inline] ax25_kill_by_device net/ax25/af_ax25.c:101 [inline] ax25_device_event+0x24d/0x580 net/ax25/af_ax25.c:146 notifier_call_chain+0x1a5/0x3f0 kernel/notifier.c:85 __dev_notify_flags+0x207/0x400 dev_change_flags+0xf0/0x1a0 net/core/dev.c:9026 dev_ifsioc+0x7c8/0xe70 net/core/dev_ioctl.c:563 dev_ioctl+0x719/0x1340 net/core/dev_ioctl.c:820 sock_do_ioctl+0x240/0x460 net/socket.c:1234 sock_ioctl+0x626/0x8e0 net/socket.c:1339 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl+0xf5/0x170 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f -> #0 (rtnl_mutex){+.+.}-{4:4}: check_prev_add kernel/locking/lockdep.c:3161 [inline] check_prevs_add kernel/locking/lockdep.c:3280 [inline] validate_chain+0x18ef/0x5920 kernel/locking/lockdep.c:3904 __lock_acquire+0x1397/0x2100 kernel/locking/lockdep.c:5226 lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5849 __mutex_lock_common kernel/locking/mutex.c:585 [inline] __mutex_lock+0x1ac/0xee0 kernel/locking/mutex.c:735 ax25_setsockopt+0xa55/0xe90 net/ax25/af_ax25.c:680 do_sock_setsockopt+0x3af/0x720 net/socket.c:2324 __sys_setsockopt net/socket.c:2349 [inline] __do_sys_setsockopt net/socket.c:2355 [inline] __se_sys_setsockopt net/socket.c:2352 [inline] __x64_sys_setsockopt+0x1ee/0x280 net/socket.c:2352 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f other info that might help us debug this: Possible unsafe locking scenario: CPU0 CPU1 ---- lock(sk_lock-AF_AX25); lock(rtnl_mutex); lock(sk_lock-AF_AX25); lock(rtnl_mutex); *** DEADLOCK *** 1 lock held by syz.5.1818/12806: #0: ffff8880617ac258 (sk_lock-AF_AX25){+.+.}-{0:0}, at: lock_sock include/net/sock.h:1618 [inline] #0: ffff8880617ac258 (sk_lock-AF_AX25){+.+.}-{0:0}, at: ax25_setsockopt+0x209/0xe90 net/ax25/af_ax25.c:574 stack backtrace: CPU: 1 UID: 0 PID: 12806 Comm: syz.5.1818 Not tainted 6.13.0-rc3-syzkaller-00762-g9268abe611b0 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_circular_bug+0x13a/0x1b0 kernel/locking/lockdep.c:2074 check_noncircular+0x36a/0x4a0 kernel/locking/lockdep.c:2206 check_prev_add kernel/locking/lockdep.c:3161 [inline] check_prevs_add kernel/lockin ---truncated---
CVE-2025-21814	In the Linux kernel, the following vulnerability has been resolved: ptp: Ensure info->enable callback is always set The ioctl and sysfs handlers unconditionally call the ->enable callback. Not all drivers implement that callback, leading to NULL dereferences. Example of affected drivers: ptp_s390.c, ptp_vclock.c and ptp_mock.c. Instead use a dummy callback if no better was specified by the driver.

CVE ID	Description
CVE-2025-21815	In the Linux kernel, the following vulnerability has been resolved: mm/compaction: fix UBSAN shift-out-of-bounds warning syzkaller reported a UBSAN shift-out-of-bounds warning of (1UL << order) in isolate_freepages_block(). The bogus compound_order can be any value because it is union with flags. Add back the MAX_PAGE_ORDER check to fix the warning.
CVE-2025-21816	In the Linux kernel, the following vulnerability has been resolved: hrtimers: Force migrate away hrtimers queued after CPUHP_AP_HRTIMERS_DYING hrtimers are migrated away from the dying CPU to any online target at the CPUHP_AP_HRTIMERS_DYING stage in order not to delay bandwidth timers handling tasks involved in the CPU hotplug forward progress. However wakeups can still be performed by the outgoing CPU after CPUHP_AP_HRTIMERS_DYING. Those can result again in bandwidth timers being armed. Depending on several considerations (crystal ball power management based election, earliest timer already enqueued, timer migration enabled or not), the target may eventually be the current CPU even if offline. If that happens, the timer is eventually ignored. The most notable example is RCU which had to deal with each and every of those wake-ups by deferring them to an online CPU, along with related workarounds: _ e787644caf76 (rcu: Defer RCU kthreads wakeup when CPU is dying) _ 9139f93209d1 (rcu/nocb: Fix RT throttling hrtimer armed from offline CPU) _ f7345ccc62a4 (rcu/nocb: Fix rcuog wake-up from offline softirq) The problem isn't confined to RCU though as the stop machine kthread (which runs CPUHP_AP_HRTIMERS_DYING) reports its completion at the end of its work through cpu_stop_signal_done() and performs a wake up that eventually arms the deadline server timer: WARNING: CPU: 94 PID: 588 at kernel/time/hrtimer.c:1086 hrtimer_start_range_ns+0x289/0x2d0 CPU: 94 UID: 0 PID: 588 Comm: migration/94 Not tainted Stopper: multi_cpu_stop+0x0/0x120 <- stop_machine_cpuslocked+0x66/0xc0 RIP: 0010:hrtimer_start_range_ns+0x289/0x2d0 Call Trace: <TASK> start_dl_timer enqueue_dl_entity dl_server_start enqueue_task_fair enqueue_task_ttwu_do_activate try_to_wake_up complete cpu_stopper_thread Instead of providing yet another bandaidd to work around the situation, fix it in the hrtimers infrastructure instead: always migrate away a timer to an online target whenever it is enqueued from an offline CPU. This will also allow to revert all the above RCU disgraceful hacks.
CVE-2025-21817	In the Linux kernel, the following vulnerability has been resolved: block: mark GFP_NOIO around sysfs ->store() sysfs ->store is called with queue freezed, meantime we have several ->store() callbacks(update_nr_requests, wbt, scheduler) to allocate memory with GFP_KERNEL which may run into direct reclaim code path, then potential deadlock can be caused. Fix the issue by marking NOIO around sysfs ->store()
CVE-2025-21819	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/amd/display: Use HW lock mgr for PSR1" This reverts commit a2b5a9956269 ("drm/amd/display: Use HW lock mgr for PSR1") Because it may cause system hang while connect with two edp panel.
CVE-2025-21820	In the Linux kernel, the following vulnerability has been resolved: tty: xilinx_uartps: split sysrq handling lockdep detects the following circular locking dependency: CPU 0 CPU 1 ===== cdns_uart_isr() printk() uart_port_lock(port) console_lock() cdns_uart_console_write() if (!port->sysrq) uart_port_lock(port) uart_handle_break() port->sysrq = ... uart_handle_sysrq_char() printk() console_lock() The fixed commit attempts to avoid this situation by only taking the port lock in cdns_uart_console_write if port->sysrq unset. However, if (as shown above) cdns_uart_console_write runs before port->sysrq is set, then it will try to take the port lock anyway. This may result in a deadlock. Fix this by splitting sysrq handling into two parts. We use the prepare helper under the port lock and defer handling until we release the lock.

CVE ID	Description
CVE-2025-21821	In the Linux kernel, the following vulnerability has been resolved: fbdev: omap: use threaded IRQ for LCD DMA When using touchscreen and framebuffer, Nokia 770 crashes easily with: BUG: scheduling while atomic: irq/144-ads7846/82/0x00010000 Modules linked in: usb_f_ecm g_ether usb_f_rndis u_ether libcomposite configfs omap_udc ohci_omap ohci_hcd CPU: 0 UID: 0 PID: 82 Comm: irq/144-ads7846 Not tainted 6.12.7-770 #2 Hardware name: Nokia 770 Call trace: unwind_backtrace from show_stack+0x10/0x14 show_stack from dump_stack_lvl+0x54/0x5c dump_stack_lvl from __schedule_bug+0x50/0x70 __schedule_bug from __schedule+0x4d4/0x5bc __schedule from schedule+0x34/0xa0 schedule from schedule_preempt_disabled+0xc/0x10 schedule_preempt_disabled from __mutex_lock.constprop.0+0x218/0x3b4 __mutex_lock.constprop.0 from clk_prepare_lock+0x38/0xe4 clk_prepare_lock from clk_set_rate+0x18/0x154 clk_set_rate from sossi_read_data+0x4c/0x168 sossi_read_data from hwa742_read_reg+0x5c/0x8c hwa742_read_reg from send_frame_handler+0xfc/0x300 send_frame_handler from process_pending_requests+0x74/0xd0 process_pending_requests from lcd_dma_irq_handler+0x50/0x74 lcd_dma_irq_handler from __handle_irq_event_percpu+0x44/0x130 __handle_irq_event_percpu from handle_irq_event+0x28/0x68 handle_irq_event from handle_level_irq+0x9c/0x170 handle_level_irq from generic_handle_domain_irq+0x2c/0x3c generic_handle_domain_irq from omap1_handle_irq+0x40/0x8c omap1_handle_irq from generic_handle_arch_irq+0x28/0x3c generic_handle_arch_irq from call_with_stack+0x1c/0x24 call_with_stack from __irq_svc+0x94/0xa8 Exception stack(0xc5255da0 to 0xc5255de8) 5da0: 00000001 c22fc620 00000000 00000000 c08384a8 c106fc00 00000000 c240c248 5dc0: c113a600 c3f6ec30 00000001 00000000 c22fc620 c5255df0 c22fc620 c0279a94 5de0: 60000013 ffffffff __irq_svc from clk_prepare_lock+0x4c/0xe4 clk_prepare_lock from clk_get_rate+0x10/0x74 clk_get_rate from uwire_setup_transfer+0x40/0x180 uwire_setup_transfer from spi_bitbang_transfer_one+0x2c/0x9c spi_bitbang_transfer_one from spi_transfer_one_message+0x2d0/0x664 spi_transfer_one_message from __spi_pump_transfer_message+0x29c/0x498 __spi_pump_transfer_message from __spi_sync+0x1f8/0x2e8 __spi_sync from spi_sync+0x24/0x40 spi_sync from ads7846_halfd_read_state+0x5c/0x1c0 ads7846_halfd_read_state from ads7846_irq+0x58/0x348 ads7846_irq from irq_thread_fn+0x1c/0x78 irq_thread_fn from irq_thread+0x120/0x228 irq_thread from kthread+0xc8/0xe8 kthread from ret_from_fork+0x14/0x28 As a quick fix, switch to a threaded IRQ which provides a stable system.
CVE-2025-21823	In the Linux kernel, the following vulnerability has been resolved: batman-adv: Drop unmanaged ELP metric worker The ELP worker needs to calculate new metric values for all neighbors "reachable" over an interface. Some of the used metric sources require locks which might need to sleep. This sleep is incompatible with the RCU list iterator used for the recorded neighbors. The initial approach to work around of this problem was to queue another work item per neighbor and then run this in a new context. Even when this solved the RCU vs might_sleep() conflict, it has a major problems: Nothing was stopping the work item in case it is not needed anymore - for example because one of the related interfaces was removed or the batman-adv module was unloaded - resulting in potential invalid memory accesses. Directly canceling the metric worker also has various problems: * cancel_work_sync for a to-be-deactivated interface is called with rtnl_lock held. But the code in the ELP metric worker also tries to use rtnl_lock() - which will never return in this case. This also means that cancel_work_sync would never return because it is waiting for the worker to finish. * iterating over the neighbor list for the to-be-deactivated interface is currently done using the RCU specific methods. Which means that it is possible to miss items when iterating over it without the associated spinlock - a behaviour which is acceptable for a periodic metric check but not for a cleanup routine (which must "stop" all still running workers) The better approach is to get rid of the per interface neighbor metric worker and handle everything in the interface worker. The original problems are solved by: * creating a list of neighbors which require new metric information inside the RCU protected context, gathering the metric according to the new list outside the RCU protected context * only use rcu_trylock inside metric gathering code to avoid a deadlock when the cancel_delayed_work_sync is called in the interface removal code (which is called with the rtnl_lock held)

CVE ID	Description
CVE-2025-21825	In the Linux kernel, the following vulnerability has been resolved: bpf: Cancel the running bpf_timer through kworker for PREEMPT_RT During the update procedure, when overwrite element in a pre-allocated htab, the freeing of old_element is protected by the bucket lock. The reason why the bucket lock is necessary is that the old_element has already been stashed in htab->extra_elems after alloc_htab_elem() returns. If freeing the old_element after the bucket lock is unlocked, the stashed element may be reused by concurrent update procedure and the freeing of old_element will run concurrently with the reuse of the old_element. However, the invocation of check_and_free_fields() may acquire a spin-lock which violates the lockdep rule because its caller has already held a raw-spin-lock (bucket lock). The following warning will be reported when such race happens: BUG: scheduling while atomic: test_progs/676/0x00000003 3 locks held by test_progs/676: #0: ffffffff864b0240 (rcu_read_lock_trace){....}-{0:0}, at: bpf_prog_test_run_syscall+0x2c0/0x830 #1: ffff88810e961188 (&htab->lockdep_key){....}-{2:2}, at: htab_map_update_elem+0x306/0x1500 #2: ffff8881f4eac1b8 (&base->softirq_expiry_lock){....}-{2:2}, at: hrtimer_cancel_wait_running+0xe9/0x1b0 Modules linked in: bpf_testmod(O) Preemption disabled at: [<fffffff817837a3>] htab_map_update_elem+0x293/0x1500 CPU: 0 UID: 0 PID: 676 Comm: test_progs Tainted: G ... 6.12.0+ #11 Tainted: [W]=WARN, [O]=OOT_MODULE Hardware name: QEMU Standard PC (i440FX + PIIX, 1996)... Call Trace: <TASK> dump_stack_lvl+0x57/0x70 dump_stack+0x10/0x20 __schedule_bug+0x120/0x170 __schedule+0x300c/0x4800 schedule_rtlock+0x37/0x60 rtlock_slowlock_locked+0x6d9/0x54c0 rt_spin_lock+0x168/0x230 hrtimer_cancel_wait_running+0xe9/0x1b0 hrtimer_cancel+0x24/0x30 bpf_timer_delete_work+0x1d/0x40 bpf_timer_cancel_and_free+0x5e/0x80 bpf_obj_free_fields+0x262/0x4a0 check_and_free_fields+0x1d0/0x280 htab_map_update_elem+0x7fc/0x1500 bpf_prog_9f90bc20768e0cb9_overwrite_cb+0x3f/0x43 bpf_prog_ea601c4649694dbd_overwrite_timer+0x5d/0x7e bpf_prog_test_run_syscall+0x322/0x830 __sys_bpf+0x135d/0x3ca0 __x64_sys_bpf+0x75/0xb0 __x64_sys_call+0x1b5/0xa10 do_syscall_64+0x3b/0xc0 entry_SYSCALL_64_after_hwframe+0x4b/0x53 ... </TASK> It seems feasible to break the reuse and refill of per-cpu extra_elems into two independent parts: reuse the per-cpu extra_elems with bucket lock being held and refill the old_element as per-cpu extra_elems after the bucket lock is unlocked. However, it will make the concurrent overwrite procedures on the same CPU return unexpected -E2BIG error when the map is full. Therefore, the patch fixes the lock problem by breaking the cancelling of bpf_timer into two steps for PREEMPT_RT: 1) use hrtimer_try_to_cancel() and check its return value 2) if the timer is running, use hrtimer_cancel() through a kworker to cancel it again Considering that the current implementation of hrtimer_cancel() will try to acquire a being held softirq_expiry_lock when the current timer is running, these steps above are reasonable. However, it also has downside. When the timer is running, the cancelling of the timer is delayed when releasing the last map uref. The delay is also fixable (e.g., break the cancelling of bpf timer into two parts: one part in locked scope, another one in unlocked scope), it can be revised later if necessary. It is a bit hard to decide the right fix tag. One reason is that the problem depends on PREEMPT_RT which is enabled in v6.12. Considering the softirq_expiry_lock lock exists since v5.4 and bpf_timer is introduced in v5.15, the bpf_timer commit is used in the fixes tag and an extra depends-on tag is added to state the dependency on PREEMPT_RT. Depends-on: v6.12+ with PREEMPT_RT enabled
CVE-2025-21826	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: reject mismatching sum of field_len with set key length The field length description provides the length of each separated key field in the concatenation, each field gets rounded up to 32-bits to calculate the pipapo rule width from pipapo_init(). The set key length provides the total size of the key aligned to 32-bits. Register-based arithmetics still allows for combining mismatching set key length and field length description, eg. set key length 10 and field description [5, 4] leading to pipapo width of 12.
CVE-2025-21828	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: don't flush non-uploaded STAs If STA state is pre-moved to AUTHORIZED (such as in IBSS scenarios) and insertion fails, the station is freed. In this case, the driver never knew about the station, so trying to flush it is unexpected and may crash. Check if the sta was uploaded to the driver before and fix this.
CVE-2025-21829	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix the warning "__rxe_cleanup+0x12c/0x170 [rdma_rxe]" The Call Trace is as below: " <TASK> ? show_regs.cold+0x1a/0x1f ? __rxe_cleanup+0x12c/0x170 [rdma_rxe] ? __warn+0x84/0xd0 ? __rxe_cleanup+0x12c/0x170 [rdma_rxe] ? report_bug+0x105/0x180 ? handle_bug+0x46/0x80 ? exc_invalid_op+0x19/0x70 ? asm_exc_invalid_op+0x1b/0x20 ? __rxe_cleanup+0x12c/0x170 [rdma_rxe] ? __rxe_cleanup+0x124/0x170 [rdma_rxe] rxe_destroy_qp.cold+0x24/0x29 [rdma_rxe] ib_destroy_qp_user+0x118/0x190 [ib_core] rdma_destroy_qp.cold+0x43/0x5e [rdma_cm] rtrs_cq_qp_destroy.cold+0x1d/0x2b [rtrs_core] rtrs_srv_close_work.cold+0x1b/0x31 [rtrs_server] process_one_work+0x21d/0x3f0 worker_thread+0x4a/0x3c0 ? process_one_work+0x3f0/0x3f0 kthread+0xf0/0x120 ? kthread_complete_and_exit+0x20/0x20 ret_from_fork+0x22/0x30 </TASK> " When too many rdma resources are allocated, rxe needs more time to handle these rdma resources. Sometimes with the current timeout, rxe can not release the rdma resources correctly. Compared with other rdma drivers, a bigger timeout is used.

CVE ID	Description
CVE-2025-21830	In the Linux kernel, the following vulnerability has been resolved: landlock: Handle weird files A corrupted filesystem (e.g. bcachefs) might return weird files. Instead of throwing a warning and allowing access to such file, treat them as regular files.
CVE-2025-21831	In the Linux kernel, the following vulnerability has been resolved: PCI: Avoid putting some root ports into D3 on TUXEDO Sirius Gen1 commit 9d26d3a8f1b0 ("PCI: Put PCIe ports into D3 during suspend") sets the policy that all PCIe ports are allowed to use D3. When the system is suspended if the port is not power manageable by the platform and won't be used for wakeup via a PME this sets up the policy for these ports to go into D3hot. This policy generally makes sense from an OSPM perspective but it leads to problems with wakeup from suspend on the TUXEDO Sirius 16 Gen 1 with a specific old BIOS. This manifests as a system hang. On the affected Device + BIOS combination, add a quirk for the root port of the problematic controller to ensure that these root ports are not put into D3hot at suspend. This patch is based on https://lore.kernel.org/linux-pci/20230708214457.1229-2-mario.limonciello@amd.com but with the added condition both in the documentation and in the code to apply only to the TUXEDO Sirius 16 Gen 1 with a specific old BIOS and only the affected root ports.
CVE-2025-21832	In the Linux kernel, the following vulnerability has been resolved: block: don't revert iter for -EIOCBQUEUED blkdev_read_iter() has a few odd checks, like gating the position and count adjustment on whether or not the result is bigger-than-or-equal to zero (where bigger than makes more sense), and not checking the return value of blkdev_direct_IO() before doing an iov_iter_revert(). The latter can lead to attempting to revert with a negative value, which when passed to iov_iter_revert() as an unsigned value will lead to throwing a WARN_ON() because unroll is bigger than MAX_RW_COUNT. Be sane and don't revert for -EIOCBQUEUED, like what is done in other spots.
CVE-2025-21833	In the Linux kernel, the following vulnerability has been resolved: iommu/vt-d: Avoid use of NULL after WARN_ON_ONCE There is a WARN_ON_ONCE to catch an unlikely situation when domain_remove_dev_pasid can't find the `pasid`. In case it nevertheless happens we must avoid using a NULL pointer.
CVE-2025-21835	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: f_midi: fix MIDI Streaming descriptor lengths While the MIDI jacks are configured correctly, and the MIDISTreaming endpoint descriptors are filled with the correct information, bNumEmbMIDIJack and bLength are set incorrectly in these descriptors. This does not matter when the numbers of in and out ports are equal, but when they differ the host will receive broken descriptors with uninitialized stack memory leaking into the descriptor for whichever value is smaller. The precise meaning of "in" and "out" in the port counts is not clearly defined and can be confusing. But elsewhere the driver consistently uses this to match the USB meaning of IN and OUT viewed from the host, so that "in" ports send data to the host and "out" ports receive data from it.
CVE-2025-21836	In the Linux kernel, the following vulnerability has been resolved: io_uring/kbuf: reallocate buf lists on upgrade IORING_REGISTER_PBUF_RING can reuse an old struct io_buffer_list if it was created for legacy selected buffer and has been emptied. It violates the requirement that most of the field should stay stable after publish. Always reallocate it instead.
CVE-2025-21838	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: core: flush gadget workqueue after device removal device_del() can lead to new work being scheduled in gadget->work workqueue. This is observed, for example, with the dwc3 driver with the following call stack: device_del() gadget_unbind_driver() usb_gadget_disconnect_locked() dwc3_gadget_pullup() dwc3_gadget_soft_disconnect() usb_gadget_set_state() schedule_work(&gadget->work) Move flush_work() after device_del() to ensure the workqueue is cleaned up.

CVE ID	Description
CVE-2025-21839	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Load DR6 with guest value only before entering .vcpu_run() loop Move the conditional loading of hardware DR6 with the guest's DR6 value out of the core .vcpu_run() loop to fix a bug where KVM can load hardware with a stale vcpu->arch.dr6. When the guest accesses a DR and host userspace isn't debugging the guest, KVM disables DR interception and loads the guest's values into hardware on VM-Enter and saves them on VM-Exit. This allows the guest to access DRs at will, e.g. so that a sequence of DR accesses to configure a breakpoint only generates one VM-Exit. For DR0-DR3, the logic/behavior is identical between VMX and SVM, and also identical between KVM_DEBUGREG_BP_ENABLED (userspace debugging the guest) and KVM_DEBUGREG_WONT_EXIT (guest using DRs), and so KVM handles loading DR0-DR3 in common code, _outside_ of the core kvm_x86_ops.vcpu_run() loop. But for DR6, the guest's value doesn't need to be loaded into hardware for KVM_DEBUGREG_BP_ENABLED, and SVM provides a dedicated VMCB field whereas VMX requires software to manually load the guest value, and so loading the guest's value into DR6 is handled by {svm,vmx}_vcpu_run(), i.e. is done _inside_ the core run loop. Unfortunately, saving the guest values on VM-Exit is initiated by common x86, again outside of the core run loop. If the guest modifies DR6 (in hardware, when DR interception is disabled), and then the next VM-Exit is a fastpath VM-Exit, KVM will reload hardware DR6 with vcpu->arch.dr6 and clobber the guest's actual value. The bug shows up primarily with nested VMX because KVM handles the VMX preemption timer in the fastpath, and the window between hardware DR6 being modified (in guest context) and DR6 being read by guest software is orders of magnitude larger in a nested setup. E.g. in non-nested, the VMX preemption timer would need to fire precisely between #DB injection and the #DB handler's read of DR6, whereas with a KVM-on-KVM setup, the window where hardware DR6 is "dirty" extends all the way from L1 writing DR6 to VMRESUME (in L1). L1's view: ===== <L1 disables DR interception> CPU 0/KVM-7289 [023] d.... 2925.640961: kvm_entry: vcpu 0 A: L1 Writes DR6 CPU 0/KVM-7289 [023] d.... 2925.640963: <hack>: Set DRs, DR6 = 0xffff0ff1 B: CPU 0/KVM-7289 [023] d.... 2925.640967: kvm_exit: vcpu 0 reason EXTERNAL_INTERRUPT intr_info 0x800000ec D: L1 reads DR6, arch.dr6 = 0 CPU 0/KVM-7289 [023] d.... 2925.640969: <hack>: Sync DRs, DR6 = 0xffff0ff0 CPU 0/KVM-7289 [023] d.... 2925.640976: kvm_entry: vcpu 0 L2 reads DR6, L1 disables DR interception CPU 0/KVM-7289 [023] d.... 2925.640980: kvm_exit: vcpu 0 reason DR_ACCESS info1 0x00000000000000216 CPU 0/KVM-7289 [023] d.... 2925.640983: kvm_entry: vcpu 0 CPU 0/KVM-7289 [023] d.... 2925.640983: <hack>: Set DRs, DR6 = 0xffff0ff0 L2 detects failure CPU 0/KVM-7289 [023] d.... 2925.640987: kvm_exit: vcpu 0 reason HLT L1 reads DR6 (confirms failure) CPU 0/KVM-7289 [023] d.... 2925.640990: <hack>: Sync DRs, DR6 = 0xffff0ff0 L0's view: ===== L2 reads DR6, arch.dr6 = 0 CPU 23/KVM-5046 [001] d.... 3410.005610: kvm_exit: vcpu 23 reason DR_ACCESS info1 0x00000000000000216 CPU 23/KVM-5046 [001] 3410.005610: kvm_nested_vmexit: vcpu 23 reason DR_ACCESS info1 0x00000000000000216 L2 => L1 nested VM-Exit CPU 23/KVM-5046 [001] 3410.005610: kvm_nested_vmexit_inject: reason: DR_ACCESS ext_inf1: 0x00000000000000216 CPU 23/KVM-5046 [001] d.... 3410.005610: kvm_entry: vcpu 23 CPU 23/KVM-5046 [001] d.... 3410.005611: kvm_exit: vcpu 23 reason VMREAD CPU 23/KVM-5046 [001] d.... 3410.005611: kvm_entry: vcpu 23 CPU 23/KVM-5046 [001] d.... 3410. ---truncated---
CVE-2025-21844	In the Linux kernel, the following vulnerability has been resolved: smb: client: Add check for next_buffer in receive_encrypted_standard() Add check for the return value of cifs_buf_get() and cifs_small_buf_get() in receive_encrypted_standard() to prevent null pointer dereference.
CVE-2025-21846	In the Linux kernel, the following vulnerability has been resolved: acct: perform last write from workqueue In [1] it was reported that the acct(2) system call can be used to trigger NULL deref in cases where it is set to write to a file that triggers an internal lookup. This can e.g., happen when pointing acc(2) to /sys/power/resume. At the point the where the write to this file happens the calling task has already exited and called exit_fs(). A lookup will thus trigger a NULL-deref when accessing current->fs. Reorganize the code so that the final write happens from the workqueue but with the caller's credentials. This preserves the (strange) permission model and has almost no regression risk. This api should stop to exist though.
CVE-2025-21847	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: stream-ipc: Check for cstream nullity in sof_ipc_msg_data() The nullity of sps->cstream should be checked similarly as it is done in sof_set_stream_data_offset() function. Assuming that it is not NULL if sps->stream is NULL is incorrect and can lead to NULL pointer dereference.
CVE-2025-21848	In the Linux kernel, the following vulnerability has been resolved: nfp: bpf: Add check for nfp_app_ctrl_msg_alloc() Add check for the return value of nfp_app_ctrl_msg_alloc() in nfp_bpf_cmsg_alloc() to prevent null pointer dereference.

CVE ID	Description
CVE-2025-21853	In the Linux kernel, the following vulnerability has been resolved: bpf: avoid holding freeze_mutex during mmap operation We use map->freeze_mutex to prevent races between map_freeze() and memory mapping BPF map contents with writable permissions. The way we naively do this means we'll hold freeze_mutex for entire duration of all the mm and VMA manipulations, which is completely unnecessary. This can potentially also lead to deadlocks, as reported by syzbot in [0]. So, instead, hold freeze_mutex only during writeability checks, bump (proactively) "write active" count for the map, unlock the mutex and proceed with mmap logic. And only if something went wrong during mmap logic, then undo that "write active" counter increment. [0] https://lore.kernel.org/bpf/678dcbc9.050a0220.303755.0066.GAE@google.com/
CVE-2025-21854	In the Linux kernel, the following vulnerability has been resolved: sockmap, vsock: For connectible sockets allow only connected sockmap expects all vsocks to have a transport assigned, which is expressed in vsock_proto::psock_update_sk_prot(). However, there is an edge case where an unconnected (connectible) socket may lose its previously assigned transport. This is handled with a NULL check in the vsock/BPF recv path. Another design detail is that listening vsocks are not supposed to have any transport assigned at all. Which implies they are not supported by the sockmap. But this is complicated by the fact that a socket, before switching to TCP_LISTEN, may have had some transport assigned during a failed connect() attempt. Hence, we may end up with a listening vsock in a sockmap, which blows up quickly: KASAN: null-ptr-deref in range [0x0000000000000120-0x0000000000000127] CPU: 7 UID: 0 PID: 56 Comm: kworker/7:0 Not tainted 6.14.0-rc1+ Workqueue: vsock-loopback vsock_loopback_work RIP: 0010:vsock_read_skb+0x4b/0x90 Call Trace: sk_psock_verdict_data_ready+0xa4/0x2e0 virtio_transport_recv_pkt+0x1ca8/0x2acc vsock_loopback_work+0x27d/0x3f0 process_one_work+0x846/0x1420 worker_thread+0x5b3/0xf80 kthread+0x35a/0x700 ret_from_fork+0x2d/0x70 ret_from_fork_asm+0x1a/0x30 For connectible sockets, instead of relying solely on the state of vsk->transport, tell sockmap to only allow those representing established connections. This aligns with the behaviour for AF_INET and AF_UNIX.
CVE-2025-21855	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Don't reference skb after sending to VIOS Previously, after successfully flushing the xmit buffer to VIOS, the tx_bytes stat was incremented by the length of the skb. It is invalid to access the skb memory after sending the buffer to the VIOS because, at any point after sending, the VIOS can trigger an interrupt to free this memory. A race between reading skb->len and freeing the skb is possible (especially during LPM) and will result in use-after-free: <pre> ===== BUG: KASAN: slab-use-after-free in ibmvnic_xmit+0x75c/0x1808 [ibmvnic] Read of size 4 at addr c00000024eb48a70 by task hxeom/14495 <...> Call Trace: [c000000118f66cf0] [c0000000018cba6c] dump_stack_lvl+0x84/0xe8 (unreliable) [c000000118f66d20] [c0000000006f0080] print_report+0x1a8/0x7f0 [c000000118f66df0] [c0000000006f08f0] kasan_report+0x128/0x1f8 [c000000118f66f00] [c0000000006f2868] _asan_load4+0xac/0xe0 [c000000118f66f20] [c0080000046eac84] ibmvnic_xmit+0x75c/0x1808 [ibmvnic] [c000000118f67340] [c00000000014be168] dev_hard_start_xmit+0x150/0x358 <...> Freed by task 0: kasan_save_stack+0x34/0x68 kasan_save_track+0x2c/0x50 kasan_save_free_info+0x64/0x108 __kasan_mempool_poison_object+0x148/0x2d4 napi_skb_cache_put+0x5c/0x194 net_tx_action +0x154/0x5b8 handle_softirqs+0x20c/0x60c do_softirq_own_stack+0x6c/0x88 <...> The buggy address belongs to the object at c00000024eb48a00 which belongs to the cache skbuff_head_cache of size 224 ===== </pre>
CVE-2025-21856	In the Linux kernel, the following vulnerability has been resolved: s390/ism: add release function for struct device According to device_release() in /drivers/base/core.c, a device without a release function is a broken device and must be fixed. The current code directly frees the device after calling device_add() without waiting for other kernel parts to release their references. Thus, a reference could still be held to a struct device, e.g., by sysfs, leading to potential use-after-free issues if a proper release function is not set.
CVE-2025-21857	In the Linux kernel, the following vulnerability has been resolved: net/sched: cls_api: fix error handling causing NULL dereference tcf_exts_miss_cookie_base_alloc() calls xa_alloc_cyclic() which can return 1 if the allocation succeeded after wrapping. This was treated as an error, with value 1 returned to caller tcf_exts_init_ex() which sets exts->actions to NULL and returns 1 to caller fl_change(). fl_change() treats err == 1 as success, calling tcf_exts_validate_ex() which calls tcf_action_init() with exts->actions as argument, where it is dereferenced. Example trace: BUG: kernel NULL pointer dereference, address: 0000000000000000 CPU: 114 PID: 16151 Comm: handler114 Kdump: loaded Not tainted 5.14.0-503.16.1.el9_5.x86_64 #1 RIP: 0010:tcf_action_init+0x1f8/0x2c0 Call Trace: tcf_action_init+0x1f8/0x2c0 tcf_exts_validate_ex+0x175/0x190 fl_change+0x537/0x1120 [cls_flower]

CVE ID	Description
CVE-2025-21858	In the Linux kernel, the following vulnerability has been resolved: geneve: Fix use-after-free in geneve_find_dev(). syzkaller reported a use-after-free in geneve_find_dev() [0] without repro. geneve_configure() links struct geneve_dev.next to net_generic(net, geneve_net_id)->geneve_list. The net here could differ from dev_net(dev) if IFLA_NET_NS_PID, IFLA_NET_NS_FD, or IFLA_TARGET_NETNSID is set. When dev_net(dev) is dismantled, geneve_exit_batch_rtnl() finally calls unregister_netdevice_queue() for each dev in the netns, and later the dev is freed. However, its geneve_dev.next is still linked to the backend UDP socket netns. Then, use-after-free will occur when another geneve dev is created in the netns. Let's call geneve_dellink() instead in geneve_destroy_tunnels(). [0]: BUG: KASAN: slab-use-after-free in geneve_find_dev drivers/net/geneve.c:1295 [inline] BUG: KASAN: slab-use-after-free in geneve_configure +0x234/0x858 drivers/net/geneve.c:1343 Read of size 2 at addr ffff000054d6ee24 by task syz.1.4029/13441 CPU: 1 UID: 0 PID: 13441 Comm: syz.1.4029 Not tainted 6.13.0-g0ad9617c78ac #24 dc35ca22c79fb82e8e7bc5c9c9adafae898b1e3d Hardware name: linux,dummy-virt (DT) Call trace: show_stack+0x38/0x50 arch/arm64/kernel/stacktrace.c:466 (C) __dump_stack lib/ dump_stack.c:94 [inline] dump_stack_lvl+0xbc/0x108 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x16c/0x6f0 mm/kasan/report.c:489 kasan_report +0xc0/0x120 mm/kasan/report.c:602 __asan_report_load2_noabort+0x20/0x30 mm/kasan/ report_generic.c:379 geneve_find_dev drivers/net/geneve.c:1295 [inline] geneve_configure +0x234/0x858 drivers/net/geneve.c:1343 geneve_newlink+0xb8/0x128 drivers/net/geneve.c:1634 rtnl_newlink_create+0x23c/0x868 net/core/rtnetlink.c:3795 __rtnl_newlink net/core/rtnetlink.c:3906 [inline] rtnl_newlink+0x1054/0x1630 net/core/rtnetlink.c:4021 rtnetlink_rcv_msg+0x61c/0x918 net/core/rtnetlink.c:6911 netlink_rcv_skb+0x1dc/0x398 net/netlink/af_netlink.c:2543 rtnetlink_rcv +0x34/0x50 net/core/rtnetlink.c:6938 netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline] netlink_unicast+0x618/0x838 net/netlink/af_netlink.c:1348 netlink_sendmsg+0x5fc/0x8b0 net/netlink/ af_netlink.c:1892 sock_sendmsg_nosec net/socket.c:713 [inline] __sock_sendmsg net/socket.c:728 [inline] ____sys_sendmsg+0x410/0x6f8 net/socket.c:2568 __sys_sendmsg+0x178/0x1d8 net/ socket.c:2622 __sys_sendmsg net/socket.c:2654 [inline] __do_sys_sendmsg net/socket.c:2659 [inline] __se_sys_sendmsg net/socket.c:2657 [inline] __arm64_sys_sendmsg+0x12c/0x1c8 net/socket.c:2657 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x90/0x278 arch/arm64/ kernel/syscall.c:49 el0_svc_common+0x13c/0x250 arch/arm64/kernel/syscall.c:132 do_el0_svc +0x54/0x70 arch/arm64/kernel/syscall.c:151 el0_svc+0x4c/0xa8 arch/arm64/kernel/entry-common.c:744 el0t_64_sync_handler+0x78/0x108 arch/arm64/kernel/entry-common.c:762 el0t_64_sync +0x198/0x1a0 arch/arm64/kernel/entry.S:600 Allocated by task 13247: kasan_save_stack mm/kasan/ common.c:47 [inline] kasan_save_track+0x30/0x68 mm/kasan/common.c:68 kasan_save_alloc_info +0x44/0x58 mm/kasan/generic.c:568 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x84/0xa0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __do_kmalloc_node mm/slab.c:4298 [inline] __kmalloc_node_noprof+0x2a0/0x560 mm/ slab.c:4304 __kvmalloc_node_noprof+0x9c/0x230 mm/util.c:645 alloc_netdev_mqs+0xb8/0x11a0 net/core/dev.c:11470 rtnl_create_link+0x2b8/0xb50 net/core/rtnetlink.c:3604 rtnl_newlink_create +0x19c/0x868 net/core/rtnetlink.c:3780 __rtnl_newlink net/core/rtnetlink.c:3906 [inline] rtnl_newlink +0x1054/0x1630 net/core/rtnetlink.c:4021 rtnetlink_rcv_msg+0x61c/0x918 net/core/rtnetlink.c:6911 netlink_rcv_skb+0x1dc/0x398 net/netlink/af_netlink.c:2543 rtnetlink_rcv+0x34/0x50 net/core/ rtnetlink.c:6938 netlink_unicast_kernel net/netlink/af_n ---truncated---
CVE-2025-21859	In the Linux kernel, the following vulnerability has been resolved: USB: gadget: f_midi: f_midi_complete to call queue_work When using USB MIDI, a lock is attempted to be acquired twice through a re-entrant call to f_midi_transmit, causing a deadlock. Fix it by using queue_work() to schedule the inner f_midi_transmit() via a high priority work queue from the completion handler.

CVE ID	Description
CVE-2025-21861	In the Linux kernel, the following vulnerability has been resolved: mm/migrate_device: don't add folio to be freed to LRU in migrate_device_finalize() If migration succeeded, we called folio_migrate_flags()->mem_cgroup_migrate() to migrate the memcg from the old to the new folio. This will set memcg_data of the old folio to 0. Similarly, if migration failed, memcg_data of the dst folio is left unset. If we call folio_putback_lru() on such folios (memcg_data == 0), we will add the folio to be freed to the LRU, making memcg code unhappy. Running the hmm selftests: # ./hmm-tests ... # RUN hmm.hmm_device_private.migrate ... [102.078007][T14893] page: refcount:1 mapcount:0 mapping:0000000000000000 index:0x7ff27d200 pfn:0x13cc00 [102.079974][T14893] anon flags: 0x17ff00000020018(uptodate dirty swapbacked node=0 zone=2 lastcpupid=0x7ff) [102.082037][T14893] raw: 017ff00000020018 dead000000000100 dead000000000122 ffff8881353896c9 [102.083687][T14893] raw: 00000007ff27d200 0000000000000000 00000001ffffff 0000000000000000 [102.085331][T14893] page dumped because: VM_WARN_ON_ONCE_FOLIO(! memcg && !mem_cgroup_disabled()) [102.087230][T14893] -----[cut here]----- [102.088279][T14893] WARNING: CPU: 0 PID: 14893 at ./include/linux/memcontrol.h:726 folio_lruvec_lock_irqsave+0x10e/0x170 [102.090478][T14893] Modules linked in: [102.091244][T14893] CPU: 0 UID: 0 PID: 14893 Comm: hmm-tests Not tainted 6.13.0-09623-g6c216bc522fd #151 [102.093089][T14893] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-2.fc40 04/01/2014 [102.094848][T14893] RIP: 0010:folio_lruvec_lock_irqsave+0x10e/0x170 [102.096104][T14893] Code: ... [102.099908][T14893] RSP: 0018:ffff900236c37b0 EFLAGS: 00010293 [102.101152][T14893] RAX: 0000000000000000 RBX: ffffea0004f30000 RCX: ffffffff8183f426 [102.102684][T14893] RDX: ffff8881063cb880 RSI: ffffffff81b8117f RDI: ffff8881063cb880 [102.104227][T14893] RBP: 0000000000000000 R08: 0000000000000005 R09: 0000000000000000 [102.105757][T14893] R10: 0000000000000001 R11: 0000000000000002 R12: ffff900236c37d8 [102.107296][T14893] R13: ffff888277a2bcb0 R14: 000000000000001f R15: 0000000000000000 [102.108830][T14893] FS: 00007ff27dbdd740(0000) GS: ffff888277a00000(0000) knlGS:0000000000000000 [102.110643][T14893] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [102.111924][T14893] CR2: 00007ff27d400000 CR3: 000000010866e000 CR4: 0000000000750ef0 [102.113478][T14893] PKRU: 55555554 [102.114172][T14893] Call Trace: [102.114805][T14893] <TASK> [102.115397][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170 [102.116547][T14893] ? __warn.cold+0x110/0x210 [102.117461][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170 [102.118667][T14893] ? report_bug+0x1b9/0x320 [102.119571][T14893] ? handle_bug+0x54/0x90 [102.120494][T14893] ? exc_invalid_op+0x17/0x50 [102.121433][T14893] ? asm_exc_invalid_op+0x1a/0x20 [102.122435][T14893] ? __wake_up_klogd.part.0+0x76/0xd0 [102.123506][T14893] ? dump_page+0x4f/0x60 [102.124352][T14893] ? folio_lruvec_lock_irqsave+0x10e/0x170 [102.125500][T14893] folio_batch_move_lru+0xd4/0x200 [102.126577][T14893] ? __pfx_lru_add+0x10/0x10 [102.127505][T14893] __folio_batch_add_and_move+0x391/0x720 [102.128633][T14893] ? __pfx_lru_add+0x10/0x10 [102.129550][T14893] folio_putback_lru+0x16/0x80 [102.130564][T14893] migrate_device_finalize+0x9b/0x530 [102.131640][T14893] dmirror_migrate_to_device.constprop.0+0x7c5/0xad0 [102.133047][T14893] dmirror_fops_unlocked_ioctl+0x89b/0xc80 Likely, nothing else goes wrong: putting the last folio reference will remove the folio from the LRU again. So besides memcg complaining, adding the folio to be freed to the LRU is just an unnecessary step. The new flow resembles what we have in migrate_folio_move(): add the dst to the lru, rem ---truncated---

CVE ID	Description
CVE-2025-21862	In the Linux kernel, the following vulnerability has been resolved: drop_monitor: fix incorrect initialization order Syzkaller reports the following bug: BUG: spinlock bad magic on CPU#1, syz-executor.0/7995 lock: 0xffff88805303f3e0, .magic: 00000000, .owner: <none>/-1, .owner_cpu: 0 CPU: 1 PID: 7995 Comm: syz-executor.0 Tainted: G E 5.10.209+ #1 Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 Call Trace: __dump_stack lib/dump_stack.c:77 [inline] dump_stack+0x119/0x179 lib/dump_stack.c:118 debug_spin_lock_before kernel/locking/spinlock_debug.c:83 [inline] do_raw_spin_lock+0x1f6/0x270 kernel/locking/spinlock_debug.c:112 __raw_spin_lock_irqsave include/linux/spinlock_api_smp.h:117 [inline] __raw_spin_lock_irqsave+0x50/0x70 kernel/locking/spinlock.c:159 reset_per_cpu_data+0xe6/0x240 [drop_monitor] net_dm_cmd_trace+0x43d/0x17a0 [drop_monitor] genl_family_rcv_msg_doit +0x22f/0x330 net/netlink/genetlink.c:739 genl_family_rcv_msg net/netlink/genetlink.c:783 [inline] genl_rcv_msg+0x341/0x5a0 net/netlink/genetlink.c:800 netlink_rcv_skb+0x14d/0x440 net/netlink/af_netlink.c:2497 genl_rcv+0x29/0x40 net/netlink/genetlink.c:811 netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline] netlink_unicast+0x54b/0x800 net/netlink/af_netlink.c:1348 netlink_sendmsg +0x914/0xe00 net/netlink/af_netlink.c:1916 sock_sendmsg_nosec net/socket.c:651 [inline] __sock_sendmsg+0x157/0x190 net/socket.c:663 __sys_sendmsg+0x712/0x870 net/socket.c:2378 __sys_sendmsg+0xf8/0x170 net/socket.c:2432 __sys_sendmsg+0xea/0x1b0 net/socket.c:2461 do_syscall_64+0x30/0x40 arch/x86/entry/common.c:46 entry_SYSCALL_64_after_hwframe +0x62/0xc7 RIP: 0033:0x7f3f9815aee9 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f3f972bf0c8 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffffdfda RBX: 00007f3f9826d050 RCX: 00007f3f9815aee9 RDX: 0000000020000000 RSI: 0000000020001300 RDI: 0000000000000007 RBP: 00007f3f981b63bd R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000006 R14: 00007f3f9826d050 R15: 00007ffe01ee6768 If drop_monitor is built as a kernel module, syzkaller may have time to send a netlink NET_DM_CMD_START message during the module loading. This will call the net_dm_monitor_start() function that uses a spinlock that has not yet been initialized. To fix this, let's place resource initialization above the registration of a generic netlink family. Found by InfoTeCS on behalf of Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-21863	In the Linux kernel, the following vulnerability has been resolved: io_uring: prevent opcode speculation sqe->opcode is used for different tables, make sure we sanitise it against speculations.
CVE-2025-21864	In the Linux kernel, the following vulnerability has been resolved: tcp: drop secpath at the same time as we currently drop dst Xiumei reported hitting the WARN in xfrm6_tunnel_net_exit while running tests that boil down to: - create a pair of netns - run a basic TCP test over ipcomp6 - delete the pair of netns The xfrm_state found on spi_byaddr was not deleted at the time we delete the netns, because we still have a reference on it. This lingering reference comes from a secpath (which holds a ref on the xfrm_state), which is still attached to an skb. This skb is not leaked, it ends up on sk_receive_queue and then gets defer-free'd by skb_attempt_defer_free. The problem happens when we defer freeing an skb (push it on one CPU's defer_list), and don't flush that list before the netns is deleted. In that case, we still have a reference on the xfrm_state that we don't expect at this point. We already drop the skb's dst in the TCP receive path when it's no longer needed, so let's also drop the secpath. At this point, tcp_filter has already called into the LSM hooks that may require the secpath, so it should not be needed anymore. However, in some of those places, the MPTCP extension has just been attached to the skb, so we cannot simply drop all extensions.

CVE ID	Description
CVE-2025-21865	In the Linux kernel, the following vulnerability has been resolved: gtp: Suppress list corruption splat in gtp_net_exit_batch_rtnl(). Brad Spengler reported the list_del() corruption splat in gtp_net_exit_batch_rtnl(). [0] Commit eb28fd76c0a0 ("gtp: Destroy device along with udp socket's netns dismantle.") added the for_each_netdev() loop in gtp_net_exit_batch_rtnl() to destroy devices in each netns as done in geneve and ip tunnels. However, this could trigger ->dellink() twice for the same device during ->exit_batch_rtnl(). Say we have two netns A & B and gtp device B that resides in netns B but whose UDP socket is in netns A. 1. cleanup_net() processes netns A and then B. 2. gtp_net_exit_batch_rtnl() finds the device B while iterating netns A's gn->gtp_dev_list and calls ->dellink(). [device B is not yet unlinked from netns B as unregister_netdevice_many() has not been called.] 3. gtp_net_exit_batch_rtnl() finds the device B while iterating netns B's for_each_netdev() and calls ->dellink(). gtp_dellink() cleans up the device's hash table, unlinks the dev from gn->gtp_dev_list, and calls unregister_netdevice_queue(). Basically, calling gtp_dellink() multiple times is fine unless CONFIG_DEBUG_LIST is enabled. Let's remove for_each_netdev() in gtp_net_exit_batch_rtnl() and delegate the destruction to default_device_exit_batch() as done in bareudp. [0]: list_del corruption, ffff8880aaa62c00->next (autoslab_size_M_dev_P_net_core_dev_11127_8_1328_8_S_4096_A_64_n_139+0xc00/0x1000 [slab object]) is LIST_POISON1 (ffffffffff02) (prev is 0xffffffffffff04) kernel BUG at lib/list_debug.c:58! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN CPU: 1 UID: 0 PID: 1804 Comm: kworker/u8:7 Tainted: G T 6.12.13-grsec-full-20250211091339 #1 Tainted: [T]=RANDSTRUCT Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Workqueue: netns cleanup_net RIP: 0010:[<fffff84947381>] __list_del_entry_valid_or_report+0x141/0x200 lib/list_debug.c:58 Code: c2 76 91 31 c0 e8 9f b1 f7 fc 0f 0b 4d 89 f0 48 c7 c1 02 ff ff ff 48 89 ea 48 89 ee 48 c7 c7 e0 c2 76 91 31 c0 e8 7f b1 f7 fc <0f> 0b 4d 89 e8 48 c7 c1 04 ff ff ff 48 89 ea 48 89 ee 48 c7 c7 60 RSP: 0018:ffffe8040b4fbd0 EFLAGS: 00010283 RAX: 00000000000000cc RBX: dffffc0000000000 RCX: fffffff818c4054 RDX: fffffff84947381 RSI: fffffff818d1512 RDI: 0000000000000000 RBP: ffff8880aaa62c00 R08: 0000000000000001 R09: fffffbd008169f32 R10: fffffe8040b4f997 R11: 0000000000000001 R12: a1988d84f24943e4 R13: fffffff818d1512 R14: fffffff818d1512 R15: ffff8880aaa62c00 RBX: kasan shadow of 0x0 RCX: __wake_up_klogd.part.0+0x74/0xe0 kernel/printk/printk.c:4554 RDX: __list_del_entry_valid_or_report+0x141/0x200 lib/list_debug.c:58 RSI: vprintk+0x72/0x100 kernel/printk/printk_safe.c:71 RBP: autoslab_size_M_dev_P_net_core_dev_11127_8_1328_8_S_4096_A_64_n_139+0xc00/0x1000 [slab object] RSP: process kstack fffffe8040b4fbd0+0x7bd0/0x8000 [kworker/u8:7+netns 1804] R09: kasan shadow of process kstack fffffe8040b4f990+0x7990/0x8000 [kworker/u8:7+netns 1804] R10: process kstack fffffe8040b4f997+0x7997/0x8000 [kworker/u8:7+netns 1804] R15: autoslab_size_M_dev_P_net_core_dev_11127_8_1328_8_S_4096_A_64_n_139+0xc08/0x1000 [slab object] FS: 0000000000000000(0000) GS:ffff888116000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000748f5372c000 CR3: 0000000015408000 CR4: 0000000003406f0 shadow CR4: 0000000003406f0 Stack: 0000000000000000 fffffff8a0c35e7 fffffff8a0c3603 ffff8880aaa62c00 ffff8880aaa62c00 0000000000000004 ffff88811145311c 0000000000000005 0000000000000001 ffff8880aaa62000 fffffe8040b4fd40 fffffff8a0c360d Call Trace: <TASK> [<ffffff8a0c360d>] __list_del_entry_valid include/linux/list.h:131 [inline] fffffe8040b4fc28 [<ffffff8a0c360d>] __list_del_entry include/linux/list.h:248 [inline] fffffe8040b4fc28 [<ffffff8a0c360d>] list_del include/linux/list.h:262 [inl ---truncated---

CVE ID	Description
CVE-2025-21866	In the Linux kernel, the following vulnerability has been resolved: powerpc/code-patching: Fix KASAN hit by not flagging text patching area as VM_ALLOC Erhard reported the following KASAN hit while booting his PowerMac G4 with a KASAN-enabled kernel 6.13-rc6: BUG: KASAN: vmalloc-out-of-bounds in copy_to_kernel_nofault+0xd8/0x1c8 Write of size 8 at addr f1000000 by task chrynyd/1293 CPU: 0 UID: 123 PID: 1293 Comm: chrynyd Tainted: G W 6.13.0-rc6-PMacG4 #2 Tainted: [W]=WARN Hardware name: PowerMac3,6 7455 0x80010303 PowerMac Call Trace: [c2437590] [c1631a84] dump_stack_lvl+0x70/0x8c (unreliable) [c24375b0] [c0504998] print_report+0xdc/0x504 [c2437610] [c050475c] kasan_report+0xf8/0x108 [c2437690] [c0505a3c] kasan_check_range +0x24/0x18c [c24376a0] [c03fb5e4] copy_to_kernel_nofault+0xd8/0x1c8 [c24376c0] [c004c014] patch_instructions+0x15c/0x16c [c2437710] [c00731a8] bpf_arch_text_copy+0x60/0x7c [c2437730] [c0281168] bpf_jit_binary_pack_finalize+0x50/0xac [c2437750] [c0073cf4] bpf_int_jit_compile +0xb30/0xdec [c2437880] [c0280394] bpf_prog_select_runtime+0x15c/0x478 [c24378d0] [c1263428] bpf_prepare_filter+0xbf8/0xc14 [c2437990] [c12677ec] bpf_prog_create_from_user+0x258/0x2b4 [c24379d0] [c027111c] do_seccomp+0x3dc/0x1890 [c2437ac0] [c001d8e0] system_call_exception +0x2dc/0x420 [c2437f30] [c00281ac] ret_from_syscall+0x0/0x2c --- interrupt: c00 at 0x5a1274 NIP: 005a1274 LR: 006a3b3c CTR: 005296c8 REGS: c2437f40 TRAP: 0c00 Tainted: G W (6.13.0-rc6-PMacG4) MSR: 0200f932 <VEC,EE,PR,FP,ME,IR,DR,RI> CR: 24004422 XER: 00000000 GPR00: 00000166 af8f3fa0 a7ee3540 00000001 00000000 013b6500 005a5858 0200f932 GPR08: 00000000 00001fe9 013d5fc8 005296c8 2822244c 00b2fcd8 00000000 af8f4b57 GPR16: 00000000 00000001 00000000 00000000 00000000 00000001 00000000 00000002 GPR24: 00afdbb0 00000000 00000000 00000000 006e0004 013ce060 006e7c1c 00000001 NIP [005a1274] 0x5a1274 LR [006a3b3c] 0x6a3b3c --- interrupt: c00 The buggy address belongs to the virtual mapping at [f1000000, f1002000] created by: text_area_cpu_up+0x20/0x190 The buggy address belongs to the physical page: page: refcount:1 mapcount:0 mapping:00000000 index:0x0 pfn:0x76e30 flags: 0x80000000(zone=2) raw: 80000000 00000000 00000122 00000000 00000000 00000000 ffffffff 00000001 raw: 00000000 page dumped because: kasan: bad access detected Memory state around the buggy address: f0ffff00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 f0ffff80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >f1000000: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 ^ f1000080: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f1000100: f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 f8 ===== f8 corresponds to KASAN_VMAALLOC_INVALID which means the area is not initialised hence not supposed to be used yet. Powerpc text patching infrastructure allocates a virtual memory area using get_vm_area() and flags it as VM_ALLOC. But that flag is meant to be used for vmalloc() and vmalloc() allocated memory is not supposed to be used before a call to __vmalloc_node_range() which is never called for that area. That went undetected until commit e4137f08816b ("mm, kasan, kmsan: instrument copy_from/to_kernel_nofault") The area allocated by text_area_cpu_up() is not vmalloc memory, it is mapped directly on demand when needed by map_kernel_page(). There is no VM flag corresponding to such usage, so just pass no flag. That way the area will be unpoisoned and usable immediately.
CVE-2025-21867	In the Linux kernel, the following vulnerability has been resolved: bpf, test_run: Fix use-after-free issue in eth_skb_pkt_type() KMSAN reported a use-after-free issue in eth_skb_pkt_type()[1]. The cause of the issue was that eth_skb_pkt_type() accessed skb's data that didn't contain an Ethernet header. This occurs when bpf_prog_test_run_xdp() passes an invalid value as the user_data argument to bpf_test_init(). Fix this by returning an error when user_data is less than ETH_HLEN in bpf_test_init(). Additionally, remove the check for "if (user_size > size)" as it is unnecessary. [1] BUG: KMSAN: use-after-free in eth_skb_pkt_type include/linux/etherdevice.h:627 [inline] BUG: KMSAN: use-after-free in eth_type_trans+0x4ee/0x980 net/ethernet/eth.c:165 eth_skb_pkt_type include/linux/etherdevice.h:627 [inline] eth_type_trans+0x4ee/0x980 net/ethernet/eth.c:165 __xdp_build_skb_from_frame +0x5a8/0xa50 net/core/xdp.c:635 xdp_rcv_frames net/bpf/test_run.c:272 [inline] xdp_test_run_batch net/bpf/test_run.c:361 [inline] bpf_test_run_xdp_live+0x2954/0x3330 net/bpf/test_run.c:390 bpf_prog_test_run_xdp+0x148e/0x1b10 net/bpf/test_run.c:1318 bpf_prog_test_run+0x5b7/0xa30 kernel/bpf/syscall.c:4371 __sys_bpf+0x6a6/0xe20 kernel/bpf/syscall.c:5777 __do_sys_bpf kernel/bpf/syscall.c:5866 [inline] __se_sys_bpf kernel/bpf/syscall.c:5864 [inline] __x64_sys_bpf +0xa4/0xf0 kernel/bpf/syscall.c:5864 x64_sys_call+0x2ea0/0x3d90 arch/x86/include/generated/asm/syscalls_64.h:322 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xd9/0x1d0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: free_pages_prepare mm/page_alloc.c:1056 [inline] free_unref_page+0x156/0x1320 mm/page_alloc.c:2657 __free_pages+0xa3/0x1b0 mm/page_alloc.c:4838 bpf_ringbuf_free kernel/bpf/ringbuf.c:226 [inline] ringbuf_map_free+0xff/0x1e0 kernel/bpf/ringbuf.c:235 bpf_map_free kernel/bpf/syscall.c:838 [inline] bpf_map_free_deferred+0x17c/0x310 kernel/bpf/syscall.c:862 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa2b/0x1b60 kernel/workqueue.c:3310 worker_thread+0xedf/0x1550 kernel/workqueue.c:3391 kthread+0x535/0x6b0 kernel/kthread.c:389 ret_from_fork+0x6e/0x90 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 CPU: 1 UID: 0 PID: 1276 Comm: syz.1.16450 Not tainted 6.12.0-05490-g9bb88c659673 #8 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014

CVE ID	Description
CVE-2025-21868	In the Linux kernel, the following vulnerability has been resolved: net: allow small head cache usage with large MAX_SKB_FRAGS values Sabrina reported the following splat: WARNING: CPU: 0 PID: 1 at net/core/dev.c:6935 netif_napi_add_weight_locked+0x8f2/0xba0 Modules linked in: CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.14.0-rc1-net-00092-g011b03359038 #996 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Arch Linux 1.16.3-1-1 04/01/2014 RIP: 0010:netif_napi_add_weight_locked+0x8f2/0xba0 Code: e8 c3 e6 6a fe 48 83 c4 28 5b 5d 41 5c 41 5d 41 5e 41 5f c3 cc cc cc cc c7 44 24 10 ff ff ff ff e9 8f fb ff ff e8 9e e6 6a fe <0f> 0b e9 d3 fe ff ff e8 92 e6 6a fe 48 8b 04 24 be ff ff ff ff 48 RSP: 0000:ffffc9000001fc60 EFLAGS: 00010293 RAX: 0000000000000000 RBX: ffff88806ce48128 RCX: 1ffff11001664b9e RDX: ffff888008f00040 RSI: ffffffff8317ca42 RDI: ffff88800b325cb6 RBP: ffff88800b325c40 R08: 0000000000000001 R09: ffffd100167502c R10: ffff88800b3a8163 R11: 0000000000000000 R12: ffff88800ac1c168 R13: ffff88800ac1c168 R14: ffff88800ac1c168 R15: 0000000000000007 FS: 0000000000000000(0000) GS:ffff88806ce00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffff888008201000 CR3: 0000000004c94001 CR4: 0000000000370ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> gro_cells_init+0x1ba/0x270 xfrm_input_init+0x4b/0x2a0 xfrm_init+0x38/0x50 ip_rt_init+0x2d7/0x350 ip_init+0xf/0x20 inet_init+0x406/0x590 do_one_initcall+0x9d/0x2e0 do_initcalls+0x23b/0x280 kernel_init_freeable+0x445/0x490 kernel_init+0x20/0x1d0 ret_from_fork+0x46/0x80 ret_from_fork_asm+0x1a/0x30 </TASK> irq event stamp: 584330 hardirqs last enabled at (584338): [<fffffff8168bf87>] __up_console_sem+0x77/0xb0 hardirqs last disabled at (584345): [<fffffff8168bf6c>] __up_console_sem+0x5c/0xb0 softirqs last enabled at (583242): [<fffffff833ee96d>] netlink_insert+0x14d/0x470 softirqs last disabled at (583754): [<fffffff8317c8cd>] netif_napi_add_weight_locked+0x77d/0xba0 on kernel built with MAX_SKB_FRAGS=45, where SKB_WITH_OVERHEAD(1024) is smaller than GRO_MAX_HEAD. Such built additionally contains the revert of the single page frag cache so that napi_get_frags() ends up using the page frag allocator, triggering the splat. Note that the underlying issue is independent from the mentioned revert; address it ensuring that the small head cache will fit either TCP and GRO allocation and updating napi_alloc_skb() and __netdev_alloc_skb() to select kmalloc() usage for any allocation fitting such cache.

CVE ID	Description
CVE-2025-21869	In the Linux kernel, the following vulnerability has been resolved: powerpc/code-patching: Disable KASAN report during patching via temporary mm Erhard reports the following KASAN hit on Talos II (power9) with kernel 6.13: [12.028126] <pre> ===== [12.028198] BUG: KASAN: user-memory-access in copy_to_kernel_nofault+0x8c/0x1a0 [12.028260] Write of size 8 at addr 0000187e458f2000 by task systemd/1 [12.028346] CPU: 87 UID: 0 PID: 1 Comm: systemd Tainted: G T 6.13.0-P9-dirty #3 [12.028408] Tainted: [T]=RANDSTRUCT [12.028446] Hardware name: T2P9D01 REV 1.01 POWER9 0x4e1202 opal:skiboot-bc106a0 PowerNV [12.028500] Call Trace: [12.028536] [c000000008dbf3b0] [c000000001656a48] dump_stack_lvl+0xbc/0x110 (unreliable) [12.028609] [c000000008dbf3f0] [c0000000006e2fc8] print_report+0x6b0/0x708 [12.028666] [c000000008dbf4e0] [c0000000006e2454] kasan_report +0x164/0x300 [12.028725] [c000000008dbf600] [c0000000006e54d4] kasan_check_range +0x314/0x370 [12.028784] [c000000008dbf640] [c0000000006e6310] __kasan_check_write +0x20/0x40 [12.028842] [c000000008dbf660] [c000000000578e8c] copy_to_kernel_nofault +0x8c/0x1a0 [12.028902] [c000000008dbf6a0] [c000000000acfe4] __patch_instructions +0x194/0x210 [12.028965] [c000000008dbf6e0] [c000000000ade80] patch_instructions +0x150/0x590 [12.029026] [c000000008dbf7c0] [c0000000001159bc] bpf_arch_text_copy +0x6c/0xe0 [12.029085] [c000000008dbf800] [c000000000424250] bpf_jit_binary_pack_finalize +0x40/0xc0 [12.029147] [c000000008dbf830] [c000000000115dec] bpf_int_jit_compile +0x3bc/0x930 [12.029206] [c000000008dbf990] [c000000000423720] bpf_prog_select_runtime +0x1f0/0x280 [12.029266] [c000000008dbfa00] [c000000000434b18] bpf_prog_load+0xbb8/0x1370 [12.029324] [c000000008dbfb70] [c000000000436ebc] __sys_bpf+0x5ac/0x2e00 [12.029379] [c000000008dbfd00] [c00000000043a228] sys_bpf+0x28/0x40 [12.029435] [c000000008dbfd20] [c000000000038eb4] system_call_exception+0x334/0x610 [12.029497] [c000000008dbfe50] [c00000000000c270] system_call_vectored_common+0xf0/0x280 [12.029561] --- interrupt: 3000 at 0x3fff82f5cfa8 [12.029608] NIP: 00003fff82f5cfa8 LR: 00003fff82f5cfa8 CTR: 0000000000000000 [12.029660] REGS: c000000008dbfe80 TRAP: 3000 Tainted: G T (6.13.0- P9-dirty) [12.029735] MSR: 900000000280f032 <SF,HV,VEC,VSX,EE,PR,FP,ME,IR,DR,RI> CR: 42004848 XER: 00000000 [12.029855] IRQMASK: 0 GPR00: 0000000000000169 00003fffdcf789a0 00003fff83067100 0000000000000005 GPR04: 00003fffdcf78a98 0000000000000090 0000000000000000 0000000000000008 GPR08: 0000000000000000 0000000000000000 0000000000000000 0000000000000000 GPR12: 0000000000000000 00003fff836ff7e0 c000000000010678 0000000000000000 GPR16: 0000000000000000 0000000000000000 00003fffdcf78f28 00003fffdcf78f90 GPR20: 0000000000000000 0000000000000000 0000000000000000 00003fffdcf78f80 GPR24: 00003fffdcf78f70 00003fffdcf78d10 00003fff835c7239 00003fffdcf78bd8 GPR28: 00003fffdcf78a98 0000000000000000 0000000000000000 000000011f547580 [12.030316] NIP [00003fff82f5cfa8] 0x3fff82f5cfa8 [12.030361] LR [00003fff82f5cfa8] 0x3fff82f5cfa8 [12.030405] --- interrupt: 3000 [12.030444] ===== Commit c28c15b6d28a ("powerpc/code-patching: Use temporary mm for Radix MMU") is inspired from x86 but unlike x86 is doesn't disable KASAN reports during patching. This wasn't a problem at the beginning because __patch_mem() is not instrumented. Commit 465cab97b42 ("powerpc/code-patching: introduce patch_instructions()") use copy_to_kernel_nofault() to copy several instructions at once. But when using temporary mm the destination is not regular kernel memory but a kind of kernel-like memory located in user address space. ---truncated---</pre>
CVE-2025-21870	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: ipc4-topology: Harden loops for looking up ALH copiers Other, non DAI copier widgets could have the same stream name (sname) as the ALH copier and in that case the copier->data is NULL, no alh_data is attached, which could lead to NULL pointer dereference. We could check for this NULL pointer in sof_ipc4_prepare_copier_module() and avoid the crash, but a similar loop in sof_ipc4_widget_setup_comp_dai() will miscalculate the ALH device count, causing broken audio. The correct fix is to harden the matching logic by making sure that the 1. widget is a DAI widget - so dai = w->private is valid 2. the dai (and thus the copier) is ALH copier
CVE-2025-21871	In the Linux kernel, the following vulnerability has been resolved: tee: optee: Fix supplicant wait loop OP-TEE supplicant is a user-space daemon and it's possible for it be hung or crashed or killed in the middle of processing an OP-TEE RPC call. It becomes more complicated when there is incorrect shutdown ordering of the supplicant process vs the OP-TEE client application which can eventually lead to system hang-up waiting for the closure of the client application. Allow the client process waiting in kernel for supplicant response to be killed rather than indefinitely waiting in an unkillable state. Also, a normal uninterruptible wait should not have resulted in the hung-task watchdog getting triggered, but the endless loop would. This fixes issues observed during system reboot/shutdown when supplicant got hung for some reason or gets crashed/killed which lead to client getting hung in an unkillable state. It in turn lead to system being in hung up state requiring hard power off/on to recover.

CVE ID	Description
CVE-2025-21872	In the Linux kernel, the following vulnerability has been resolved: efi: Don't map the entire mokvar table to determine its size Currently, when validating the mokvar table, we (re)map the entire table on each iteration of the loop, adding space as we discover new entries. If the table grows over a certain size, this fails due to limitations of early_memmap(), and we get a failure and traceback: ----- <pre>[cut here]----- WARNING: CPU: 0 PID: 0 at mm/early_ioremap.c:139 __early_ioremap +0xef/0x220 ... Call Trace: <TASK> ? __early_ioremap+0xef/0x220 ? __warn.cold+0x93/0xfa ? __early_ioremap+0xef/0x220 ? report_bug+0xff/0x140 ? early_fixup_exception+0x5d/0xb0 ? early_idt_handler_common+0x2f/0x3a ? __early_ioremap+0xef/0x220 ? efi_mokvar_table_init +0xce/0x1d0 ? setup_arch+0x864/0xc10 ? start_kernel+0x6b/0xa10 ? x86_64_start_reservations +0x24/0x30 ? x86_64_start_kernel+0xed/0xf0 ? common_startup_64+0x13e/0x141 </TASK> --- [end trace 0000000000000000]--- mokvar: Failed to map EFI MOKvar config table pa=0x7c4c3000, size=265187. Mapping the entire structure isn't actually necessary, as we don't ever need more than one entry header mapped at once. Changes efi_mokvar_table_init() to only map each entry header, not the entire table, when determining the table size. Since we're not mapping any data past the variable name, it also changes the code to enforce that each variable name is NUL terminated, rather than attempting to verify it in place. </pre>
CVE-2025-21873	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: bsg: Fix crash when arpm command fails If the device doesn't support arpm we'll crash due to copying user data in bsg_transport_sg_io_fn(). In the case where ufs_bsg_exec_advanced_rpmb_req() returns an error, do not set the job's reply_len. Memory crash backtrace: 3,1290,531166405,-;ufshcd 0000:00:12.5: ARPM OP failed: error code -22 4,1308,531166555,-;Call Trace: 4,1309,531166559,-; <TASK> 4,1310,531166565,-; ? show_regs+0x6d/0x80 4,1311,531166575,-; ? die+0x37/0xa0 4,1312,531166583,-; ? do_trap+0xd4/0xf0 4,1313,531166593,-; ? do_error_trap+0x71/0xb0 4,1314,531166601,-; ? usercopy_abort+0x6c/0x80 4,1315,531166610,-; ? exc_invalid_op+0x52/0x80 4,1316,531166622,-; ? usercopy_abort+0x6c/0x80 4,1317,531166630,-; ? asm_exc_invalid_op +0x1b/0x20 4,1318,531166643,-; ? usercopy_abort+0x6c/0x80 4,1319,531166652,-; __check_heap_object+0xe3/0x120 4,1320,531166661,-; check_heap_object+0x185/0x1d0 4,1321,531166670,-; __check_object_size.part.0+0x72/0x150 4,1322,531166679,-; __check_object_size +0x23/0x30 4,1323,531166688,-; bsg_transport_sg_io_fn+0x314/0x3b0

CVE ID	Description
CVE-2025-21875	In the Linux kernel, the following vulnerability has been resolved: mptcp: always handle address removal under msk socket lock Syzkaller reported a lockdep splat in the PM control path: WARNING: CPU: 0 PID: 6693 at ./include/net/sock.h:1711 sock_owned_by_me include/net/sock.h:1711 [inline] WARNING: CPU: 0 PID: 6693 at ./include/net/sock.h:1711 msk_owned_by_me net/mptcp/protocol.h:363 [inline] WARNING: CPU: 0 PID: 6693 at ./include/net/sock.h:1711 mptcp_pm_nl_addr_send_ack+0x57c/0x610 net/mptcp/pm_netlink.c:788 Modules linked in: CPU: 0 UID: 0 PID: 6693 Comm: syz.0.205 Not tainted 6.14.0-rc2-syzkaller-00303-gad1b832bf1cf #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 12/27/2024 RIP: 0010:sock_owned_by_me include/net/sock.h:1711 [inline] RIP: 0010:msk_owned_by_me net/mptcp/protocol.h:363 [inline] RIP: 0010:mptcp_pm_nl_addr_send_ack+0x57c/0x610 net/mptcp/pm_netlink.c:788 Code: 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc e8 ca 7b d3 f5 eb b9 e8 c3 7b d3 f5 90 0f 0b 90 e9 dd fb ff ff e8 b5 7b d3 f5 90 <0f> 0b 90 e9 3e fb ff ff 44 89 f1 80 e1 07 38 c1 0f 8c eb fb ff ff RSP: 0000:ffffc900034f6f60 EFLAGS: 00010283 RAX: ffffffff8bee3c2b RBX: 0000000000000001 RCX: 0000000000080000 RDX: fffffc90004d42000 RSI: 000000000000a407 RDI: 000000000000a408 RBP: fffffc900034f7030 R08: ffffffff8bee37f6 R09: 0100000000000000 R10: dffffc0000000000 R11: ffffd100bcc62e4 R12: ffff88805e6316e0 R13: ffff88805e630c00 R14: dffffc0000000000 R15: ffff88805e630c00 FS: 00007f7e9a7e96c0(0000) GS:ffff8880b8600000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 0000000000005033 CR2: 0000001b2fd18ff8 CR3: 0000000032c24000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 00000000000000400 Call Trace: <TASK> mptcp_pm_remove_addr +0x103/0x1d0 net/mptcp/pm.c:59 mptcp_pm_remove_anno_addr+0x1f4/0x2f0 net/mptcp/pm_netlink.c:1486 mptcp_nl_remove_subflow_and_signal_addr net/mptcp/pm_netlink.c:1518 [inline] mptcp_pm_nl_del_addr_doit+0x118d/0x1af0 net/mptcp/pm_netlink.c:1629 genl_family_rcv_msg_doit net/netlink/genetlink.c:1115 [inline] genl_family_rcv_msg net/netlink/genetlink.c:1195 [inline] genl_rcv_msg+0xb1f/0xec0 net/netlink/genetlink.c:1210 netlink_rcv_skb+0x206/0x480 net/netlink/af_netlink.c:2543 genl_rcv+0x28/0x40 net/netlink/genetlink.c:1219 netlink_unicast_kernel net/netlink/af_netlink.c:1322 [inline] netlink_unicast+0x7f6/0x990 net/netlink/af_netlink.c:1348 netlink_sendmsg+0x8de/0xcb0 net/netlink/af_netlink.c:1892 sock_sendmsg_nosec net/socket.c:718 [inline] __sock_sendmsg+0x221/0x270 net/socket.c:733 __sys_sendmsg+0x53a/0x860 net/socket.c:2573 __sys_sendmsg net/socket.c:2627 [inline] __sys_sendmsg+0x269/0x350 net/socket.c:2659 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f7e9998cde9 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f7e9a7e9038 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffff8bda RBX: 00007f7e99ba5fa0 RCX: 00007f7e9998cde9 RDX: 000000002000c094 RSI: 0000400000000000 RDI: 0000000000000007 RBP: 00007f7e99a0e2a0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000000 R14: 00007f7e99ba5fa0 R15: 00007fff49231088 Indeed the PM can try to send a RM_ADDR over a msk without acquiring first the msk socket lock. The bugged code-path comes from an early optimization: when there are no subflows, the PM should (usually) not send RM_ADDR notifications. The above statement is incorrect, as without locks another process could concur ---truncated---

CVE ID	Description
CVE-2025-21877	In the Linux kernel, the following vulnerability has been resolved: usbnet: gl620a: fix endpoint checking in genlink_bind() Syzbot reports [1] a warning in usb_submit_urb() triggered by inconsistencies between expected and actually present endpoints in gl620a driver. Since genlink_bind() does not properly verify whether specified eps are in fact provided by the device, in this case, an artificially manufactured one, one may get a mismatch. Fix the issue by resorting to a usbnet utility function usbnet_get_endpoints(), usually reserved for this very problem. Check for endpoints and return early before proceeding further if any are missing. [1] Syzbot report: usb 5-1: Manufacturer: syz usb 5-1: SerialNumber: syz usb 5-1: config 0 descriptor?? gl620a 5-1:0.23 usb0: register 'gl620a' at usb-dummy_hcd.0-1, ... -----[cut here]----- usb 5-1: BOGUS urb xfer, pipe 3 != type 1 WARNING: CPU: 2 PID: 1841 at drivers/usb/core/urb.c:503 usb_submit_urb+0xe4b/0x1730 drivers/usb/core/urb.c:503 Modules linked in: CPU: 2 UID: 0 PID: 1841 Comm: kworker/2:2 Not tainted 6.12.0-syzkaller-07834-g06afb0f36106 #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Workqueue: mld mld_ifc_work RIP: 0010:usb_submit_urb+0xe4b/0x1730 drivers/usb/core/urb.c:503 ... Call Trace: <TASK> usbnet_start_xmit+0x6be/0x2780 drivers/net/usb/usbnet.c:1467 __netdev_start_xmit include/linux/netdevice.h:5002 [inline] netdev_start_xmit include/linux/netdevice.h:5011 [inline] xmit_one net/core/dev.c:3590 [inline] dev_hard_start_xmit+0x9a/0x7b0 net/core/dev.c:3606 sch_direct_xmit+0x1ae/0xc30 net/sched/sch_generic.c:343 __dev_xmit_skb net/core/dev.c:3827 [inline] __dev_queue_xmit+0x13d4/0x43e0 net/core/dev.c:4400 dev_queue_xmit include/linux/netdevice.h:3168 [inline] neigh_resolve_output net/core/neighbour.c:1514 [inline] neigh_resolve_output+0x5bc/0x950 net/core/neighbour.c:1494 neigh_output include/net/neighbour.h:539 [inline] ip6_finish_output2+0xb1b/0x2070 net/ipv6/ip6_output.c:141 __ip6_finish_output net/ipv6/ip6_output.c:215 [inline] ip6_finish_output+0x3f9/0x1360 net/ipv6/ip6_output.c:226 NF_HOOK_COND include/linux/netfilter.h:303 [inline] ip6_output+0x1f8/0x540 net/ipv6/ip6_output.c:247 dst_output include/net/dst.h:450 [inline] NF_HOOK include/linux/netfilter.h:314 [inline] NF_HOOK include/linux/netfilter.h:308 [inline] mld_sendpack+0x9f0/0x11d0 net/ipv6/mcast.c:1819 mld_send_cr net/ipv6/mcast.c:2120 [inline] mld_ifc_work+0x740/0xca0 net/ipv6/mcast.c:2651 process_one_work+0x9c5/0x1ba0 kernel/workqueue.c:3229 process_scheduled_works kernel/workqueue.c:3310 [inline] worker_thread+0x6c8/0xf00 kernel/workqueue.c:3391 kthread+0x2c1/0x3a0 kernel/kthread.c:389 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK>
CVE-2025-21878	In the Linux kernel, the following vulnerability has been resolved: i2c: npcm: disable interrupt enable bit before devm_request_irq The customer reports that there is a soft lockup issue related to the i2c driver. After checking, the i2c module was doing a tx transfer and the bmc machine reboots in the middle of the i2c transaction, the i2c module keeps the status without being reset. Due to such an i2c module status, the i2c irq handler keeps getting triggered since the i2c irq handler is registered in the kernel booting process after the bmc machine is doing a warm rebooting. The continuous triggering is stopped by the soft lockup watchdog timer. Disable the interrupt enable bit in the i2c module before calling devm_request_irq to fix this issue since the i2c relative status bit is read-only. Here is the soft lockup log. [28.176395] watchdog: BUG: soft lockup - CPU#0 stuck for 26s! [swapper/0:1] [28.183351] Modules linked in: [28.186407] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.15.120-yocto-s-dirty-bbebc78 #1 [28.201174] pstate: 40000005 (nZcv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) [28.208128] pc : __do_softirq+0xb0/0x368 [28.212055] lr : __do_softirq+0x70/0x368 [28.215972] sp : ffffff8035ebca00 [28.219278] x29: ffffff8035ebca00 x28: 0000000000000002 x27: ffffff80071a3780 [28.226412] x26: ffffffc008bdc000 x25: ffffffc008bdc640 x24: ffffffc008be50c0 [28.233546] x23: ffffffc00800200c x22: 0000000000000000 x21: 000000000000001b [28.240679] x20: 0000000000000000 x19: ffffff80001c3200 x18: ffffffff00000000 [28.247812] x17: ffffffc02d2e0000 x16: ffffff8035eb8b40 x15: 00001e8480000000 [28.254945] x14: 02c3647e37dbfcb6 x13: 02c364f2ab14200c x12: 00000000002c364f2 [28.262078] x11: 00000000fa83b2da x10: 000000000000b67e x9 : ffffffc008010250 [28.269211] x8 : 000000009d983d00 x7 : 7fffffff00000000 x6 : 0000036d74732434 [28.276344] x5 : 00ffffff00000000 x4 : 0000000000000015 x3 : 0000000000000198 [28.283476] x2 : ffffffc02d2e0000 x1 : 00000000000000e0 x0 : ffffffc008bdc640 [28.290611] Call trace: [28.293052] __do_softirq+0xb0/0x368 [28.296625] __irq_exit_rcu+0xe0/0x100 [28.300374] irq_exit+0x14/0x20 [28.303513] handle_domain_irq+0x68/0x90 [28.307440] gic_handle_irq+0x78/0xb0 [28.311098] call_on_irq_stack+0x20/0x38 [28.315019] do_interrupt_handler+0x54/0x5c [28.319199] el1_interrupt+0x2c/0x4c [28.322777] el1h_64_irq_handler+0x14/0x20 [28.326872] el1h_64_irq+0x74/0x78 [28.330269] __setup_irq+0x454/0x780 [28.333841] request_threaded_irq+0xd0/0x1b4 [28.338107] devm_request_threaded_irq+0x84/0x100 [28.342809] npcm_i2c_probe_bus+0x188/0x3d0 [28.346990] platform_probe+0x6c/0xc4 [28.350653] really_probe+0xcc/0x45c [28.354227] __driver_probe_device+0x8c/0x160 [28.358578] driver_probe_device+0x44/0xe0 [28.362670] __driver_attach+0x124/0x1d0 [28.366589] bus_for_each_dev+0x7c/0xe0 [28.370426] driver_attach+0x28/0x30 [28.373997] bus_add_driver+0x124/0x240 [28.377830] driver_register+0x7c/0x124 [28.381662] __platform_driver_register+0x2c/0x34 [28.386362] npcm_i2c_init+0x3c/0x5c [28.389937] do_one_initcall+0x74/0x230 [28.393768] kernel_init_freeable+0x24c/0x2b4 [28.398126] kernel_init+0x28/0x130 [28.401614] ret_from_fork+0x10/0x20 [28.405189] Kernel panic - not syncing: softlockup: hung tasks [28.411011] SMP: stopping secondary CPUs [28.414933] Kernel Offset: disabled [28.418412] CPU features: 0x00000000,00000802 [28.427644] Rebooting in 20 seconds..

973

CVE ID	Description
CVE-2025-21883	In the Linux kernel, the following vulnerability has been resolved: ice: Fix deinitializing VF in error path If ice_ena_vfs() fails after calling ice_create_vf_entries(), it frees all VFs without removing them from snapshot PF-VF mailbox list, leading to list corruption. Reproducer: devlink dev eswitch set \$PF1_PCI mode switchdev ip l s \$PF1 up ip l s \$PF1 promisc on sleep 1 echo 1 > /sys/class/net/\$PF1/device/sriov_numvfs sleep 1 echo 1 > /sys/class/net/\$PF1/device/sriov_numvfs Trace (minimized): list_add corruption. next->prev should be prev (ffff8882e241c6f0), but was 0000000000000000. (next=ffff888455da1330). kernel BUG at lib/list_debug.c:29! RIP: 0010: __list_add_valid_or_report+0xa6/0x100 ice_mbx_init_vf_info+0xa7/0x180 [ice] ice_initialize_vf_entry+0x1fa/0x250 [ice] ice_sriov_configure+0x8d7/0x1520 [ice] ? __percpu_ref_switch_mode+0x1b1/0x5d0 ? __pfx_ice_sriov_configure+0x10/0x10 [ice] Sometimes a KASAN report can be seen instead with a similar stack trace: BUG: KASAN: use-after-free in __list_add_valid_or_report+0xf1/0x100 VFs are added to this list in ice_mbx_init_vf_info(), but only removed in ice_free_vfs(). Move the removing to ice_free_vf_entries(), which is also being called in other places where VFs are being removed (including ice_free_vfs() itself).
CVE-2025-21884	In the Linux kernel, the following vulnerability has been resolved: net: better track kernel sockets lifetime While kernel sockets are dismantled during pernet_operations->exit(), their freeing can be delayed by any tx packets still held in qdisc or device queues, due to skb_set_owner_w() prior calls. This then trigger the following warning from ref_tracker_dir_exit() [1] To fix this, make sure that kernel sockets own a reference on net->passive. Add sk_net_refcnt_upgrade() helper, used whenever a kernel socket is converted to a refcounted one. [1] [136.263918][T35] ref_tracker: net notrefcnt@ffff8880638f01e0 has 1/2 users at [136.263918][T35] sk_alloc+0x2b3/0x370 [136.263918][T35] inet6_create+0x6ce/0x10f0 [136.263918][T35] __sock_create+0x4c0/0xa30 [136.263918][T35] inet_ctl_sock_create+0xc2/0x250 [136.263918][T35] igmp6_net_init+0x39/0x390 [136.263918][T35] ops_init+0x31e/0x590 [136.263918][T35] setup_net+0x287/0x9e0 [136.263918][T35] copy_net_ns+0x33f/0x570 [136.263918][T35] create_new_namespaces+0x425/0x7b0 [136.263918][T35] unshare_nsproxy_namespaces+0x124/0x180 [136.263918][T35] ksys_unshare+0x57d/0xa70 [136.263918][T35] __x64_sys_unshare+0x38/0x40 [136.263918][T35] do_syscall_64+0xf3/0x230 [136.263918][T35] entry_SYSCALL_64_after_hwframe+0x77/0x7f [136.263918][T35] [136.343488][T35] ref_tracker: net notrefcnt@ffff8880638f01e0 has 1/2 users at [136.343488][T35] sk_alloc+0x2b3/0x370 [136.343488][T35] inet6_create+0x6ce/0x10f0 [136.343488][T35] __sock_create+0x4c0/0xa30 [136.343488][T35] inet_ctl_sock_create+0xc2/0x250 [136.343488][T35] ndisc_net_init+0xa7/0x2b0 [136.343488][T35] ops_init+0x31e/0x590 [136.343488][T35] setup_net+0x287/0x9e0 [136.343488][T35] copy_net_ns+0x33f/0x570 [136.343488][T35] create_new_namespaces+0x425/0x7b0 [136.343488][T35] unshare_nsproxy_namespaces+0x124/0x180 [136.343488][T35] ksys_unshare+0x57d/0xa70 [136.343488][T35] __x64_sys_unshare+0x38/0x40 [136.343488][T35] do_syscall_64+0xf3/0x230 [136.343488][T35] entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2025-21885	In the Linux kernel, the following vulnerability has been resolved: RDMA/bnxt_re: Fix the page details for the srq created by kernel consumers While using nvme target with use_srq on, below kernel panic is noticed. [549.698111] bnxt_en 0000:41:00.0 enp65s0np0: FEC autoneg off encoding: Clause 91 RS(544,514) [566.393619] Oops: divide error: 0000 [#1] PREEMPT SMP NOPTI .. [566.393799] <TASK> [566.393807] ? __die_body+0x1a/0x60 [566.393823] ? die+0x38/0x60 [566.393835] ? do_trap+0xe4/0x110 [566.393847] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re] [566.393867] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re] [566.393881] ? do_error_trap+0x7c/0x120 [566.393890] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re] [566.393911] ? exc_divide_error+0x34/0x50 [566.393923] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re] [566.393939] ? asm_exc_divide_error+0x16/0x20 [566.393966] ? bnxt_qplib_alloc_init_hwq+0x1d4/0x580 [bnxt_re] [566.393997] bnxt_qplib_create_srq+0xc9/0x340 [bnxt_re] [566.394040] bnxt_re_create_srq+0x335/0x3b0 [bnxt_re] [566.394057] ? srso_return_thunk+0x5/0x5f [566.394068] ? __init_swait_queue_head+0x4a/0x60 [566.394090] ib_create_srq_user+0xa7/0x150 [ib_core] [566.394147] nvmet_rdma_queue_connect+0x7d0/0xbe0 [nvmet_rdma] [566.394174] ? lock_release+0x22c/0x3f0 [566.394187] ? srso_return_thunk+0x5/0x5f Page size and shift info is set only for the user space SRQs. Set page size and page shift for kernel space SRQs also.

CVE ID	Description
CVE-2025-21886	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix implicit ODP hang on parent deregistration Fix the destroy_unused_implicit_child_mr() to prevent hanging during parent deregistration as of below [1]. Upon entering destroy_unused_implicit_child_mr(), the reference count for the implicit MR parent is incremented using: refcount_inc_not_zero(). A corresponding decrement must be performed if free_implicit_child_mr_work() is not called. The code has been updated to properly manage the reference count that was incremented. [1] INFO: task python3:2157 blocked for more than 120 seconds. Not tainted 6.12.0-rc7+ #1633 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:python3 state:D stack:0 pid:2157 tgid:2157 ppid:1685 flags:0x00000000 Call Trace: <TASK> __schedule+0x420/0xd30 schedule+0x47/0x130 __mlx5_ib_dereg_mr+0x379/0x5d0 [mlx5_ib] ? __pfx_autoremove_wake_function+0x10/0x10 ib_dereg_mr_user+0x5f/0x120 [ib_core] ? lock_release+0xc6/0x280 destroy_hw_idr_uobject+0x1d/0x60 [ib_uverbs] uverbs_destroy_uobject+0x58/0x1d0 [ib_uverbs] uobj_destroy+0x3f/0x70 [ib_uverbs] ib_uverbs_cmd_verbs+0x3e4/0xbb0 [ib_uverbs] ? __pfx_uverbs_destroy_def_handler+0x10/0x10 [ib_uverbs] ? lock_acquire+0xc1/0x2f0 ? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs] ? ib_uverbs_ioctl+0x116/0x170 [ib_uverbs] ? lock_release+0xc6/0x280 ib_uverbs_ioctl+0xe7/0x170 [ib_uverbs] ? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs] __x64_sys_ioctl+0x1b0/0xa70 ? kmem_cache_free+0x221/0x400 do_syscall_64+0x6b/0x140 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f20f21f017b RSP: 002b:00007ffc4a77c8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffffda RBX: 00007ffc4a78d8 RCX: 00007f20f21f017b RDX: 00007ffc4a78c0 RSI: 00000000c0181b01 RDI: 0000000000000003 RBP: 00007ffc4a78a0 R08: 000056147d125190 R09: 00007f20f1f14c60 R10: 0000000000000001 R11: 0000000000000246 R12: 00007ffc4a7890 R13: 000000000000001c R14: 000056147d100fc0 R15: 00007f20e365c9d0 </TASK>
CVE-2025-21887	In the Linux kernel, the following vulnerability has been resolved: ovl: fix UAF in ovl_dentry_update_reval by moving dput() in ovl_link_up The issue was caused by dput(upper) being called before ovl_dentry_update_reval(), while upper->d_flags was still accessed in ovl_dentry_remote(). Move dput(upper) after its last use to prevent use-after-free. BUG: KASAN: slab-use-after-free in ovl_dentry_remote fs/overlayfs/util.c:162 [inline] BUG: KASAN: slab-use-after-free in ovl_dentry_update_reval+0xd2/0xf0 fs/overlayfs/util.c:167 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:114 print_address_description mm/kasan/report.c:377 [inline] print_report+0xc3/0x620 mm/kasan/report.c:488 kasan_report+0xd9/0x110 mm/kasan/report.c:601 ovl_dentry_remote fs/overlayfs/util.c:162 [inline] ovl_dentry_update_reval+0xd2/0xf0 fs/overlayfs/util.c:167 ovl_link_up fs/overlayfs/copy_up.c:610 [inline] ovl_copy_up_one+0x2105/0x3490 fs/overlayfs/copy_up.c:1170 ovl_copy_up_flags+0x18d/0x200 fs/overlayfs/copy_up.c:1223 ovl_rename+0x39e/0x18c0 fs/overlayfs/dir.c:1136 vfs_rename+0xf84/0x20a0 fs/namei.c:4893 ... </TASK>

CVE ID	Description
CVE-2025-21888	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix a WARN during dereg_mr for DM type Memory regions (MR) of type DM (device memory) do not have an associated umem. In the __mlx5_ib_dereg_mr() -> mlx5_free_priv_descs() flow, the code incorrectly takes the wrong branch, attempting to call dma_unmap_single() on a DMA address that is not mapped. This results in a WARN [1], as shown below. The issue is resolved by properly accounting for the DM type and ensuring the correct branch is selected in mlx5_free_priv_descs(). [1] WARNING: CPU: 12 PID: 1346 at drivers/iommu/dma-iommu.c:1230 iommu_dma_unmap_page+0x79/0x90 Modules linked in: ip6table_mangle ip6table_nat ip6table_filter ip6_tables iptable_mangle xt_conntrack xt_MASQUERADE nf_conntrack_netlink nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter rpcsec_gss_krb5 auth_rpcgss oid_registry overlay rpcrdma rdma_ucm ib_iser libiscsi scsi_transport_iscsi ib_umad rdma_cm ib_ipoib iw_cm ib_cm mlx5_ib ib_uverbs ib_core fuse mlx5_core CPU: 12 UID: 0 PID: 1346 Comm: ibv_rc_pingpong Not tainted 6.12.0-rc7+ #1631 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:iommu_dma_unmap_page+0x79/0x90 Code: 2b 49 3b 29 72 26 49 3b 69 08 73 20 4d 89 f0 44 89 e9 4c 89 e2 48 89 ee 48 89 df 5b 5d 41 5c 41 5d 41 5e 41 5f e9 07 b8 88 ff <0f> 0b 5b 5d 41 5c 41 5d 41 5e 41 5f c3 cc cc cc cc 66 0f 1f 44 00 RSP: 0018:ffff90001913a10 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff88810194b0a8 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000001 RBP: ffff88810194b0a8 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000000 R13: 0000000000000001 R14: 0000000000000000 R15: 0000000000000000 FS: 00007f537abdd740(0000) GS:ffff88885fb00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f537aeb8000 CR3: 000000010c248001 CR4: 000000000372eb0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __warn+0x84/0x190 ? iommu_dma_unmap_page+0x79/0x90 ? report_bug+0xf8/0x1c0 ? handle_bug+0x55/0x90 ? exc_invalid_op+0x13/0x60 ? asm_exc_invalid_op+0x16/0x20 ? iommu_dma_unmap_page+0x79/0x90 dma_unmap_page_attr+0xe6/0x290 mlx5_free_priv_descs+0xb0/0xe0 [mlx5_ib] __mlx5_ib_dereg_mr+0x37e/0x520 [mlx5_ib] ? _raw_spin_unlock_irq+0x24/0x40 ? wait_for_completion+0xfe/0x130 ? rdma_restrack_put+0x63/0xe0 [ib_core] ib_dereg_mr_user+0x5f/0x120 [ib_core] ? lock_release+0xc6/0x280 destroy_hw_idr_uobject+0x1d/0x60 [ib_uverbs] uverbs_destroy_uobject+0x58/0x1d0 [ib_uverbs] uobj_destroy+0x3f/0x70 [ib_uverbs] ib_uverbs_cmd_verbs+0x3e4/0xbb0 [ib_uverbs] ? __pfx_uverbs_destroy_def_handler+0x10/0x10 [ib_uverbs] ? lock_acquire+0xc1/0x2f0 ? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs] ? ib_uverbs_ioctl+0x116/0x170 [ib_uverbs] ? lock_release+0xc6/0x280 ib_uverbs_ioctl+0xe7/0x170 [ib_uverbs] ? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs] __x64_sys_ioctl+0x1b0/0xa70 do_syscall_64+0x6b/0x140 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f537adaf17b Code: 0f 1e fa 48 8b 05 1d ad 0c 00 64 c7 00 26 00 00 00 48 c7 c0 ff ff ff c3 66 0f 1f 44 00 00 f3 0f 1e fa b8 10 00 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d ed ac 0c 00 f7 d8 64 89 01 48 RSP: 002b:00007ffff218f0b8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff8fda RBX: 00007ffff218f1d8 RCX: 00007f537adaf17b RDX: 00007ffff218f1c0 RSI: 00000000c0181b01 RDI: 0000000000000003 RBP: 00007ffff218f1a0 R08: 00007f537aa8d010 R09: 0000561ee2e4f270 R10: 00007f537aace3a8 R11: 00000000000000246 R12: 00007ffff218f190 R13: 000000000000001c R14: 0000561ee2e4d7c0 R15: 00007ffff218f450 </TASK>
CVE-2025-21889	In the Linux kernel, the following vulnerability has been resolved: perf/core: Add RCU read lock protection to perf_iterate_ctx() The perf_iterate_ctx() function performs RCU list traversal but currently lacks RCU read lock protection. This causes lockdep warnings when running perf probe with unshare(1) under CONFIG_PROVE_RCU_LIST=y: WARNING: suspicious RCU usage kernel/events/core.c:8168 RCU-list traversed in non-reader section!! Call Trace: lockdep_rcu_suspicious ? perf_event_addr_filters_apply perf_iterate_ctx perf_event_exec begin_new_exec ? load_elf_phdrs load_elf_binary ? lock_acquire ? find_held_lock ? bprm_execve bprm_execve do_execveat_common.isra.0 __x64_sys_execve do_syscall_64 entry_SYSCALL_64_after_hwframe This protection was previously present but was removed in commit bd2756811766 ("perf: Rewrite core context handling"). Add back the necessary rcu_read_lock()/rcu_read_unlock() pair around perf_iterate_ctx() call in perf_event_exec(). [mingo: Use scoped_guard() as suggested by Peter]

CVE ID	Description
CVE-2025-21890	In the Linux kernel, the following vulnerability has been resolved: idpf: fix checksums set in idpf_rx_rsc() idpf_rx_rsc() uses skb_transport_offset(skb) while the transport header is not set yet. This triggers the following warning for CONFIG_DEBUG_NET=y builds. DEBUG_NET_WARN_ON_ONCE(!skb_transport_header_was_set(skb)) [69.261620] WARNING: CPU: 7 PID: 0 at ./include/linux/skbuff.h:3020 idpf_vport_splitq_napi_poll (include/linux/skbuff.h:3020) idpf [69.261629] Modules linked in: vfat fat dummy bridge intel_uncore_frequency_tpmi intel_uncore_frequency_common intel_vsec_tpmi idpf intel_vsec cdc_ncm cdc_eem cdc_ether usbnet mii xhci_pci xhci_hcd ehci_pci ehci_hcd libeth [69.261644] CPU: 7 UID: 0 PID: 0 Comm: swapper/7 Tainted: G S W 6.14.0-smp-DEV #1697 [69.261648] Tainted: [S]=CPU_OUT_OF_SPEC, [W]=WARN [69.261650] RIP: 0010:idpf_vport_splitq_napi_poll (include/linux/skbuff.h:3020) idpf [69.261677] ? __warn (kernel/panic.c:242 kernel/panic.c:748) [69.261682] ? idpf_vport_splitq_napi_poll (include/linux/skbuff.h:3020) idpf [69.261687] ? report_bug (lib/bug.c:?) [69.261690] ? handle_bug (arch/x86/kernel/traps.c:285) [69.261694] ? exc_invalid_op (arch/x86/kernel/traps.c:309) [69.261697] ? asm_exc_invalid_op (arch/x86/include/asm/idtentry.h:621) [69.261700] ? pfx_idpf_vport_splitq_napi_poll (drivers/net/ethernet/intel/idpf/idpf_txrx.c:4011) idpf [69.261704] ? idpf_vport_splitq_napi_poll (include/linux/skbuff.h:3020) idpf [69.261708] ? idpf_vport_splitq_napi_poll (drivers/net/ethernet/intel/idpf/idpf_txrx.c:3072) idpf [69.261712] __napi_poll (net/core/dev.c:7194) [69.261716] net_rx_action (net/core/dev.c:7265) [69.261718] ? __qdisc_run (net/sched/sch_generic.c:293) [69.261721] ? sched_clock (arch/x86/include/asm/preempt.h:84 arch/x86/kernel/tsc.c:288) [69.261726] handle_softirqs (kernel/softirq.c:561)
CVE-2025-21891	In the Linux kernel, the following vulnerability has been resolved: ipvlan: ensure network headers are in skb linear part syzbot found that ipvlan_process_v6_outbound() was assuming the IPv6 network header is present in skb->head [1] Add the needed pskb_network_may_pull() calls for both IPv4 and IPv6 handlers. [1] BUG: KMSAN: uninit-value in __ipv6_addr_type+0xa2/0x490 net/ipv6/addrconf_core.c:47 __ipv6_addr_type+0xa2/0x490 net/ipv6/addrconf_core.c:47 ipv6_addr_type include/net/ipv6.h:555 [inline] ip6_route_output_flags_noref net/ipv6/route.c:2616 [inline] ip6_route_output_flags+0x51/0x720 net/ipv6/route.c:2651 ip6_route_output include/net/ipv6/route.h:93 [inline] ipvlan_route_v6_outbound+0x24e/0x520 drivers/net/ipvlan/ipvlan_core.c:476 ipvlan_process_v6_outbound drivers/net/ipvlan/ipvlan_core.c:491 [inline] ipvlan_process_outbound drivers/net/ipvlan/ipvlan_core.c:541 [inline] ipvlan_xmit_mode_l3 drivers/net/ipvlan/ipvlan_core.c:605 [inline] ipvlan_queue_xmit+0xd72/0x1780 drivers/net/ipvlan/ipvlan_core.c:671 ipvlan_start_xmit+0x5b/0x210 drivers/net/ipvlan/ipvlan_main.c:223 __netdev_start_xmit include/linux/netdevice.h:5150 [inline] netdev_start_xmit include/linux/netdevice.h:5159 [inline] xmit_one net/core/dev.c:3735 [inline] dev_hard_start_xmit+0x247/0xa20 net/core/dev.c:3751 sch_direct_xmit+0x399/0xd40 net/sched/sch_generic.c:343 qdisc_restart net/sched/sch_generic.c:408 [inline] __qdisc_run+0x14da/0x35d0 net/sched/sch_generic.c:416 qdisc_run+0x141/0x4d0 include/net/pkt_sched.h:127 net_tx_action+0x78b/0x940 net/core/dev.c:5484 handle_softirqs+0x1a0/0x7c0 kernel/softirq.c:561 __do_softirq+0x14/0x1a kernel/softirq.c:595 do_softirq+0x9a/0x100 kernel/softirq.c:462 __local_bh_enable_ip+0x9f/0xb0 kernel/softirq.c:389 local_bh_enable include/linux/bottom_half.h:33 [inline] rcu_read_unlock_bh include/linux/rcupdate.h:919 [inline] __dev_queue_xmit+0x2758/0x57d0 net/core/dev.c:4611 dev_queue_xmit include/linux/netdevice.h:3311 [inline] packet_xmit+0x9c/0x6c0 net/packet/af_packet.c:276 packet_snd net/packet/af_packet.c:3132 [inline] packet_sendmsg+0x93e0/0xa7e0 net/packet/af_packet.c:3164 sock_sendmsg_nosec net/socket.c:718 [inline]

CVE ID	Description
CVE-2025-21892	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix the recovery flow of the UMR QP This patch addresses an issue in the recovery flow of the UMR QP, ensuring tasks do not get stuck, as highlighted by the call trace [1]. During recovery, before transitioning the QP to the RESET state, the software must wait for all outstanding WRs to complete. Failing to do so can cause the firmware to skip sending some flushed CQEs with errors and simply discard them upon the RESET, as per the IB specification. This race condition can result in lost CQEs and tasks becoming stuck. To resolve this, the patch sends a final WR which serves only as a barrier before moving the QP state to RESET. Once a CQE is received for that final WR, it guarantees that no outstanding WRs remain, making it safe to transition the QP to RESET and subsequently back to RTS, restoring proper functionality. Note: For the barrier WR, we simply reuse the failed and ready WR. Since the QP is in an error state, it will only receive IB_WC_WR_FLUSH_ERR. However, as it serves only as a barrier we don't care about its status. [1] INFO: task rdma_resource_1:1922 blocked for more than 120 seconds. Tainted: G W 6.12.0-rc7+ #1626 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:rdma_resource_1 state:D stack:0 pid:1922 tgid:1922 ppid:1369 flags:0x00004004 Call Trace: <TASK> __schedule+0x420/0xd30 schedule+0x47/0x130 schedule_timeout+0x280/0x300 ? mark_held_locks+0x48/0x80 ? lockdep_hardirqs_on_prepare+0xe5/0x1a0 wait_for_completion+0x75/0x130 mlx5r_umr_post_send_wait+0x3c2/0x5b0 [mlx5_ib] ? __pfx_mlx5r_umr_done+0x10/0x10 [mlx5_ib] mlx5r_umr_revoke_mr+0x93/0xc0 [mlx5_ib] __mlx5_ib_dereg_mr+0x299/0x520 [mlx5_ib] ? _raw_spin_unlock_irq+0x24/0x40 ? wait_for_completion+0xfe/0x130 ? rdma_restrack_put+0x63/0xe0 [ib_core] ib_dereg_mr_user+0x5f/0x120 [ib_core] ? lock_release+0xc6/0x280 destroy_hw_idr_uobject+0x1d/0x60 [ib_uverbs] uverbs_destroy_uobject+0x58/0x1d0 [ib_uverbs] uobj_destroy+0x3f/0x70 [ib_uverbs] ib_uverbs_cmd_verbs+0x3e4/0xbb0 [ib_uverbs] ? __pfx_uverbs_destroy_def_handler+0x10/0x10 [ib_uverbs] ? __lock_acquire+0x64e/0x2080 ? mark_held_locks+0x48/0x80 ? find_held_lock+0x2d/0xa0 ? lock_acquire+0xc1/0x2f0 ? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs] ? __fget_files+0xc3/0x1b0 ib_uverbs_ioctl+0xe7/0x170 [ib_uverbs] ? ib_uverbs_ioctl+0xcb/0x170 [ib_uverbs] __x64_sys_ioctl+0x1b0/0xa70 do_syscall_64+0x6b/0x140 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f99c918b17b RSP: 002b:00007ffc766d0468 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffffda RBX: 00007ffc766d0578 RCX: 00007f99c918b17b RDX: 00007ffc766d0560 RSI: 00000000c0181b01 RDI: 0000000000000003 RBP: 00007ffc766d0540 R08: 00007f99c8f99010 R09: 0000000000000bd7e R10: 00007f99c94c1c70 R11: 0000000000000246 R12: 00007ffc766d0530 R13: 000000000000001c R14: 0000000040246a80 R15: 0000000000000000 </TASK>
CVE-2025-21894	In the Linux kernel, the following vulnerability has been resolved: net: enetc: VFs do not support HWTSTAMP_TX_ONESTEP_SYNC Actually ENETC VFs do not support HWTSTAMP_TX_ONESTEP_SYNC because only ENETC PF can access PMA_SINGLE_STEP registers. And there will be a crash if VFs are used to test one-step timestamp, the crash log as follows. [129.110909] Unable to handle kernel paging request at virtual address 000000000000080c0 [129.287769] Call trace: [129.290219] enetc_port_mac_wr+0x30/0xec (P) [129.294504] enetc_start_xmit+0xda4/0xe74 [129.298525] enetc_xmit+0x70/0xec [129.301848] dev_hard_start_xmit+0x98/0x118
CVE-2025-21895	In the Linux kernel, the following vulnerability has been resolved: perf/core: Order the PMU list to fix warning about unordered pmu_ctx_list Syskaller triggers a warning due to prev_epc->pmu != next_epc->pmu in perf_event_swap_task_ctx_data(). vmcore shows that two lists have the same perf_event_pmu_context, but not in the same order. The problem is that the order of pmu_ctx_list for the parent is impacted by the time when an event/PMU is added. While the order for a child is impacted by the event order in the pinned_groups and flexible_groups. So the order of pmu_ctx_list in the parent and child may be different. To fix this problem, insert the perf_event_pmu_context to its proper place after iteration of the pmu_ctx_list. The follow testcase can trigger above warning: # perf record -e cycles --call-graph lbr -- taskset -c 3 ./a.out & # perf stat -e cpu-clock,cs -p xxx // xxx is the pid of a.out test.c void main() { int count = 0; pid_t pid; printf("%d running\n", getpid()); sleep(30); printf("running\n"); pid = fork(); if (pid == -1) { printf("fork error\n"); return; } if (pid == 0) { while (1) { count++; } } else { while (1) { count++; } } } The testcase first opens an LBR event, so it will allocate task_ctx_data, and then open tracepoint and software events, so the parent context will have 3 different perf_event_pmu_contexts. On inheritance, child ctx will insert the perf_event_pmu_context in another order and the warning will trigger. [mingo: Tidied up the changelog.]
CVE-2025-21898	In the Linux kernel, the following vulnerability has been resolved: ftrace: Avoid potential division by zero in function_stat_show() Check whether denominator expression x * (x - 1) * 1000 mod {2^32, 2^64} produce zero and skip stddev computation in that case. For now don't care about rec->counter * rec->counter overflow because rec->time * rec->time overflow will likely happen earlier.

CVE ID	Description
CVE-2025-21899	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix bad hist from corrupting named_triggers list The following commands causes a crash: ~# cd /sys/kernel/tracing/events/rcu/rcu_callback ~# echo 'hist:name=bad:keys=common_pid:onmax(bogus).save(common_pid)' > trigger bash: echo: write error: Invalid argument ~# echo 'hist:name=bad:keys=common_pid' > trigger Because the following occurs: event_trigger_write() { trigger_process_regex() { event_hist_trigger_parse() { data = event_trigger_alloc(..); event_trigger_register(.., data) { cmd_ops->reg(.., data, ..) [hist_register_trigger()] { data->ops->init() [event_hist_trigger_init()] { save_named_trigger(name, data) { list_add(&data->named_list, &named_triggers); } } } ret = create_actions(); (return -EINVAL) if (ret) goto out_unreg; [...] ret = hist_trigger_enable(data, ...) { list_add_tail_rcu(&data->list, &file->triggers); <<<---- SKIPPED!!! (this is important!) [...] out_unreg: event_hist_unregister(.., data) { cmd_ops->unreg(.., data, ..) [hist_unregister_trigger()] { list_for_each_entry(iter, &file->triggers, list) { if (!hist_trigger_match(data, iter, named_data, false)) <- never matches continue; [...] test = iter; } if (test && test->ops->free) <<<-- test is NULL test->ops->free(test) [event_hist_trigger_free()] { [...] if (data->name) del_named_trigger(data) { list_del(&data->named_list); <<<-- NEVER gets removed! } } } [...] kfree(data); <<<-- frees item but it is still on list The next time a hist with name is registered, it causes an u-a-f bug and the kernel can crash. Move the code around such that if event_trigger_register() succeeds, the next thing called is hist_trigger_enable() which adds it to the list. A bunch of actions is called if get_named_trigger_data() returns false. But that doesn't need to be called after event_trigger_register(), so it can be moved up, allowing event_trigger_register() to be called just before hist_trigger_enable() keeping them together and allowing the file->triggers to be properly populated.
CVE-2025-21903	In the Linux kernel, the following vulnerability has been resolved: mctp i3c: handle NULL header address daddr can be NULL if there is no neighbour table entry present, in that case the tx packet should be dropped. saddr will usually be set by MCTP core, but check for NULL in case a packet is transmitted by a different protocol.
CVE-2025-21904	In the Linux kernel, the following vulnerability has been resolved: caif_virtio: fix wrong pointer check in cfv_probe() del_vqs() frees virtqueues, therefore cfv->vq_tx pointer should be checked for NULL before calling it, not cfv->vdev. Also the current implementation is redundant because the pointer cfv->vdev is dereferenced before it is checked for NULL. Fix this by checking cfv->vq_tx for NULL instead of cfv->vdev before calling del_vqs().
CVE-2025-21905	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: limit printed string from FW file There's no guarantee here that the file is always with a NUL-termination, so reading the string may read beyond the end of the TLV. If that's the last TLV in the file, it can perhaps even read beyond the end of the file buffer. Fix that by limiting the print format to the size of the buffer we have.
CVE-2025-21907	In the Linux kernel, the following vulnerability has been resolved: mm: memory-failure: update ttu flag inside unmap_poisoned_folio Patch series "mm: memory_failure: unmap poisoned folio during migrate properly", v3. Fix two bugs during folio migration if the folio is poisoned. This patch (of 3): Commit 6da6b1d4a7df ("mm/hwpoison: convert TTU_IGNORE_HWPOISON to TTU_HWPOISON") introduce TTU_HWPOISON to replace TTU_IGNORE_HWPOISON in order to stop send SIGBUS signal when accessing an error page after a memory error on a clean folio. However during page migration, anon folio must be set with TTU_HWPOISON during unmap_*. For pagecache we need some policy just like the one in hwpoison_user_mappings to set this flag. So move this policy from hwpoison_user_mappings to unmap_poisoned_folio to handle this warning properly. Warning will be produced during unamp poison folio with the following log: -----[cut here]----- WARNING: CPU: 1 PID: 365 at mm/rmap.c:1847 try_to_unmap_one+0x8fc/0xd3c Modules linked in: CPU: 1 UID: 0 PID: 365 Comm: bash Tainted: G W 6.13.0-rc1-00018-gacdb4bbda7ab #42 Tainted: [W]=WARN Hardware name: QEMU QEMU Virtual Machine, BIOS 0.0.0 02/06/2015 pstate: 20400005 (nzCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : try_to_unmap_one+0x8fc/0xd3c lr : try_to_unmap_one+0x3dc/0xd3c Call trace: try_to_unmap_one+0x8fc/0xd3c (P) try_to_unmap_one+0x3dc/0xd3c (L) rmap_walk_anon+0xdc/0x1f8 rmap_walk+0x3c/0x58 try_to_unmap+0x88/0x90 unmap_poisoned_folio+0x30/0xa8 do_migrate_range+0x4a0/0x568 offline_pages+0x5a4/0x670 memory_block_action+0x17c/0x374 memory_subsys_offline+0x3c/0x78 device_offline+0xa4/0xd0 state_store+0x8c/0xf0 dev_attr_store+0x18/0x2c sysfs_kf_write+0x44/0x54 kernfs_fop_write_iter+0x118/0x1a8 vfs_write+0x3a8/0x4bc ksys_write+0x6c/0xf8 __arm64_sys_write+0x1c/0x28 invoke_syscall+0x44/0x100 el0_svc_common.constprop.0+0x40/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x30/0xd0 el0t_64_sync_handler+0xc8/0xcc el0t_64_sync+0x198/0x19c ---[end trace 0000000000000000]--- [mawupeng1@huawei.com: unmap_poisoned_folio(): remove shadowed local 'mapping', per Miaohel]

CVE ID	Description
CVE-2025-21908	In the Linux kernel, the following vulnerability has been resolved: NFS: fix nfs_release_folio() to not deadlock via kcompactd writeback Add PF_KCOMPACTD flag and current_is_kcompactd() helper to check for it so nfs_release_folio() can skip calling nfs_wb_folio() from kcompactd. Otherwise NFS can deadlock waiting for kcompactd ended writeback which recurses back to NFS (which triggers writeback to NFSD via NFS loopback mount on the same host, NFSD blocks waiting for XFS's call to __filemap_get_folio): 6070.550357] INFO: task kcompactd0:58 blocked for more than 4435 seconds. {--- [58] "kcompactd0" [<0>] folio_wait_bit+0xe8/0x200 [<0>] folio_wait_writeback+0x2b/0x80 [<0>] nfs_wb_folio+0x80/0x1b0 [nfs] [<0>] nfs_release_folio+0x68/0x130 [nfs] [<0>] split_huge_page_to_list_to_order+0x362/0x840 [<0>] migrate_pages_batch+0x43d/0xb90 [<0>] migrate_pages_sync+0x9a/0x240 [<0>] migrate_pages+0x93c/0x9f0 [<0>] compact_zone+0x8e2/0x1030 [<0>] compact_node+0xdb/0x120 [<0>] kcompactd+0x121/0x2e0 [<0>] kthread+0xcfc/0x100 [<0>] ret_from_fork+0x31/0x40 [<0>] ret_from_fork_asm+0x1a/0x30 ---} [akpm@linux-foundation.org: fix build]
CVE-2025-21909	In the Linux kernel, the following vulnerability has been resolved: wifi: nl80211: reject cooked mode if it is set along with other flags It is possible to set both MONITOR_FLAG_COOK_FRAMES and MONITOR_FLAG_ACTIVE flags simultaneously on the same monitor interface from the userspace. This causes a sub-interface to be created with no IEEE80211_SDATA_IN_DRIVER bit set because the monitor interface is in the cooked state and it takes precedence over all other states. When the interface is then being deleted the kernel calls WARN_ONCE() from check_sdata_in_driver() because of missing that bit. Fix this by rejecting MONITOR_FLAG_COOK_FRAMES if it is set along with other flags. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-21910	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: regulatory: improve invalid hints checking Syzbot keeps reporting an issue [1] that occurs when erroneous symbols sent from userspace get through into user_alpha2[] via regulatory_hint_user() call. Such invalid regulatory hints should be rejected. While a sanity check from commit 47caf685a685 ("cfg80211: regulatory: reject invalid hints") looks to be enough to deter these very cases, there is a way to get around it due to 2 reasons. 1) The way isalpha() works, symbols other than latin lower and upper letters may be used to determine a country/domain. For instance, greek letters will also be considered upper/lower letters and for such characters isalpha() will return true as well. However, ISO-3166-1 alpha2 codes should only hold latin characters. 2) While processing a user regulatory request, between reg_process_hint_user() and regulatory_hint_user() there happens to be a call to queue_regulatory_request() which modifies letters in request->alpha2[] with toupper(). This works fine for latin symbols, less so for weird letter characters from the second part of _ctype[]. Syzbot triggers a warning in is_user_regdom_saved() by first sending over an unexpected non-latin letter that gets malformed by toupper() into a character that ends up failing isalpha() check. Prevent this by enhancing is_an_alpha2() to ensure that incoming symbols are latin letters and nothing else. [1] Syzbot report: -----[cut here]----- Unexpected user alpha2: A# WARNING: CPU: 1 PID: 964 at net/wireless/reg.c:442 is_user_regdom_saved net/wireless/reg.c:440 [inline] WARNING: CPU: 1 PID: 964 at net/wireless/reg.c:442 restore_alpha2 net/wireless/reg.c:3424 [inline] WARNING: CPU: 1 PID: 964 at net/wireless/reg.c:442 restore_regulatory_settings+0x3c0/0x1e50 net/wireless/reg.c:3516 Modules linked in: CPU: 1 UID: 0 PID: 964 Comm: kworker/1:2 Not tainted 6.12.0-rc5-syzkaller-00044-gc1e939a21eb1 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: events_power_efficient crda_timeout_work RIP: 0010:is_user_regdom_saved net/wireless/reg.c:440 [inline] RIP: 0010:restore_alpha2 net/wireless/reg.c:3424 [inline] RIP: 0010:restore_regulatory_settings+0x3c0/0x1e50 net/wireless/reg.c:3516 ... Call Trace: <TASK> crda_timeout_work+0x27/0x50 net/wireless/reg.c:542 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa65/0x1850 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f2/0x390 kernel/kthread.c:389 ret_from_fork+0x4d/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK>
CVE-2025-21911	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: avoid deadlock on fence release Do scheduler queue fence release processing on a workqueue, rather than in the release function itself. Fixes deadlock issues such as the following: [607.400437] ===== [607.405755] WARNING: possible recursive locking detected [607.415500] ----- [607.420817] weston:zfq0/24149 is trying to acquire lock: [607.426131] ffff000017d041a0 (reservation_ww_class_mutex){+.+.}-{3:3}, at: pvr_gem_object_vunmap+0x40/0xc0 [powervr] [607.436728] but task is already holding lock: [607.442554] ffff000017d105a0 (reservation_ww_class_mutex){+.+.}-{3:3}, at: dma_buf_iocd+0x250/0x554 [607.451727] other info that might help us debug this: [607.458245] Possible unsafe locking scenario: [607.464155] CPU0 [607.466601] ---- [607.469044] lock(reservation_ww_class_mutex); [607.473584] lock(reservation_ww_class_mutex); [607.478114] *** DEADLOCK ***

CVE ID	Description
CVE-2025-21912	In the Linux kernel, the following vulnerability has been resolved: gpio: rcar: Use raw_spinlock to protect register access Use raw_spinlock in order to fix spurious messages about invalid context when spinlock debugging is enabled. The lock is only used to serialize register access. [4.239592] ===== [4.239595] [BUG: Invalid wait context] [4.239599] 6.13.0-rc7-arm64-renesas-05496-gd088502a519f #35 Not tainted [4.239603] ----- [4.239606] kworker/u8:5/76 is trying to lock: [4.239609] ffff000001898a0 (&p->lock){....}-{3:3}, at: gpio_rcar_config_interrupt_input_mode+0x34/0x164 [4.239641] other info that might help us debug this: [4.239643] context-{5:5} [4.239646] 5 locks held by kworker/u8:5/76: [4.239651] #0: ffff00000080fb148 ((wq_completion)async){+.+.}-{0:0}, at: process_one_work+0x190/0x62c [4.250180] OF: /soc/sound@ec500000/ports/port@0/endpoint: Read of boolean property 'frame-master' with a value. [4.254094] #1: ffff80008299bd80 ((work_completion)(&entry->work)){+.+.}-{0:0}, at: process_one_work+0x1b8/0x62c [4.254109] #2: ffff00000920c8f8 [4.258345] OF: /soc/sound@ec500000/ports/port@1/endpoint: Read of boolean property 'bitclock-master' with a value. [4.264803] (&dev->mutex){....}-{4:4}, at: __device_attach_async_helper+0x3c/0xdc [4.264820] #3: ffff00000a50ca40 (request_class#2){+.+.}-{4:4}, at: __setup_irq+0xa0/0x690 [4.264840] #4: [4.268872] OF: /soc/sound@ec500000/ports/port@1/endpoint: Read of boolean property 'frame-master' with a value. [4.273275] ffff00000a50c8c8 (lock_class){....}-{2:2}, at: __setup_irq+0xc4/0x690 [4.296130] renesas_sdhi_internal_dmac ee100000.mmc: mmc1 base at 0x00000000ee100000, max clock rate 200 MHz [4.304082] stack backtrace: [4.304086] CPU: 1 UID: 0 PID: 76 Comm: kworker/u8:5 Not tainted 6.13.0-rc7-arm64-renesas-05496-gd088502a519f #35 [4.304092] Hardware name: Renesas Salvator-X 2nd version board based on r8a77965 (DT) [4.304097] Workqueue: async async_run_entry_fn [4.304106] Call trace: [4.304110] show_stack+0x14/0x20 (C) [4.304122] dump_stack_lvl+0x6c/0x90 [4.304131] dump_stack+0x14/0x1c [4.304138] __lock_acquire+0xdfc/0x1584 [4.426274] lock_acquire+0x1c4/0x33c [4.429942] _raw_spin_lock_irqsave+0x5c/0x80 [4.434307] gpio_rcar_config_interrupt_input_mode+0x34/0x164 [4.440061] gpio_rcar_irq_set_type+0xd4/0xd8 [4.444422] __irq_set_trigger+0x5c/0x178 [4.448435] __setup_irq+0x2e4/0x690 [4.452012] request_threaded_irq+0xc4/0x190 [4.456285] devm_request_threaded_irq+0x7c/0xf4 [4.459398] ata1: link resume succeeded after 1 retries [4.460902] mmc_gpiod_request_cd_irq+0x68/0xe0 [4.470660] mmc_start_host+0x50/0xac [4.474327] mmc_add_host+0x80/0xe4 [4.477817] tmio_mmc_host_probe+0x2b0/0x440 [4.482094] renesas_sdhi_probe+0x488/0x6f4 [4.486281] renesas_sdhi_internal_dmac_probe+0x60/0x78 [4.491509] platform_probe+0x64/0xd8 [4.495178] really_probe+0xb8/0x2a8 [4.498756] __driver_probe_device+0x74/0x118 [4.503116] driver_probe_device+0x3c/0x154 [4.507303] __device_attach_driver+0xd4/0x160 [4.511750] bus_for_each_drv+0x84/0xe0 [4.515588] __device_attach_async_helper+0xb0/0xdc [4.520470] async_run_entry_fn+0x30/0xd8 [4.524481] process_one_work+0x210/0x62c [4.528494] worker_thread+0x1ac/0x340 [4.532245] kthread+0x10c/0x110 [4.535476] ret_from_fork+0x10/0x20
CVE-2025-21913	In the Linux kernel, the following vulnerability has been resolved: x86/amd_nb: Use rdmsr_safe() in amd_get_mmconfig_range() Xen doesn't offer MSR_FAM10H_MMIO_CONF_BASE to all guests. This results in the following warning: unchecked MSR access error: RDMSR from 0xc0010058 at rIP: 0xfffffff8101d19f (xen_do_read_msr+0x7f/0xa0) Call Trace: xen_read_msr+0x1e/0x30 amd_get_mmconfig_range+0x2b/0x80 quirk_amd_mmconfig_area+0x28/0x100 pnp_fixup_device+0x39/0x50 __pnp_add_device+0xf/0x150 pnp_add_device+0x3d/0x100 pnpacpi_add_device_handler+0x1f9/0x280 acpi_ns_get_device_callback+0x104/0x1c0 acpi_ns_walk_namespace+0x1d0/0x260 acpi_get_devices+0x8a/0xb0 pnpacpi_init+0x50/0x80 do_one_initcall+0x46/0x2e0 kernel_init_freeable+0x1da/0x2f0 kernel_init+0x16/0x1b0 ret_from_fork+0x30/0x50 ret_from_fork_asm+0x1b/0x30 based on quirks for a "PNP0c01" device. Treating MMCFG as disabled is the right course of action, so no change is needed there. This was most likely exposed by fixing the Xen MSR accessors to not be silently-safe.
CVE-2025-21914	In the Linux kernel, the following vulnerability has been resolved: slimbus: messaging: Free transaction ID in delayed interrupt scenario In case of interrupt delay for any reason, slim_do_transfer() returns timeout error but the transaction ID (TID) is not freed. This results into invalid memory access inside qcom_slim_ngd_rx_msgq_cb() due to invalid TID. Fix the issue by freeing the TID in slim_do_transfer() before returning timeout error to avoid invalid memory access. Call trace: __memcpy_fromio+0x20/0x190 qcom_slim_ngd_rx_msgq_cb+0x130/0x290 [slim_qcom_ngd_ctrl] vchan_complete+0x2a0/0x4a0 tasklet_action_common+0x274/0x700 tasklet_action+0x28/0x3c _stext+0x188/0x620 run_ksoftirqd+0x34/0x74 smpboot_thread_fn+0x1d8/0x464 kthread+0x178/0x238 ret_from_fork+0x10/0x20 Code: aa0003e8 91000429 f100044a 3940002b (3800150b) ---[end trace 0fe00bec2b975c99]--- Kernel panic - not syncing: Oops: Fatal exception in interrupt.

CVE ID	Description
CVE-2025-21915	In the Linux kernel, the following vulnerability has been resolved: cdx: Fix possible UAF error in driver_override_show() Fixed a possible UAF problem in driver_override_show() in drivers/cdx/cdx.c This function driver_override_show() is part of DEVICE_ATTR_RW, which includes both driver_override_show() and driver_override_store(). These functions can be executed concurrently in sysfs. The driver_override_store() function uses driver_set_override() to update the driver_override value, and driver_set_override() internally locks the device (device_lock(dev)). If driver_override_show() reads cdx_dev->driver_override without locking, it could potentially access a freed pointer if driver_override_store() frees the string concurrently. This could lead to printing a kernel address, which is a security risk since DEVICE_ATTR can be read by all users. Additionally, a similar pattern is used in drivers/amba/bus.c, as well as many other bus drivers, where device_lock() is taken in the show function, and it has been working without issues. This potential bug was detected by our experimental static analysis tool, which analyzes locking APIs and paired functions to identify data races and atomicity violations.
CVE-2025-21916	In the Linux kernel, the following vulnerability has been resolved: usb: atm: cxacru: fix a flaw in existing endpoint checks Syzbot once again identified a flaw in usb endpoint checking, see [1]. This time the issue stems from a commit authored by me (2eabb655a968 ("usb: atm: cxacru: fix endpoint checking in cxacru_bind()")). While using usb_find_common_endpoints() may usually be enough to discard devices with wrong endpoints, in this case one needs more than just finding and identifying the sufficient number of endpoints of correct types - one needs to check the endpoint's address as well. Since cxacru_bind() fills URBs with CXACRU_EP_CMD address in mind, switch the endpoint verification approach to usb_check_XXX_endpoints() instead to fix incomplete ep testing. [1] Syzbot report: usb 5-1: BOGUS urb xfer, pipe 3 != type 1 WARNING: CPU: 0 PID: 1378 at drivers/usb/core/urb.c:504 usb_submit_urb+0xc4e/0x18c0 drivers/usb/core/urb.c:503 ... RIP: 0010:usb_submit_urb+0xc4e/0x18c0 drivers/usb/core/urb.c:503 ... Call Trace: <TASK> cxacru_cm+0x3c8/0xe50 drivers/usb/atm/cxacru.c:649 cxacru_card_status drivers/usb/atm/cxacru.c:760 [inline] cxacru_bind+0xcf9/0x1150 drivers/usb/atm/cxacru.c:1223 usbatm_usb_probe+0x314/0x1d30 drivers/usb/atm/usbatm.c:1058 cxacru_usb_probe+0x184/0x220 drivers/usb/atm/cxacru.c:1377 usb_probe_interface+0x641/0xbb0 drivers/usb/core/driver.c:396 really_probe+0x2b9/0xad0 drivers/base/dd.c:658 __driver_probe_device+0x1a2/0x390 drivers/base/dd.c:800 driver_probe_device+0x50/0x430 drivers/base/dd.c:830 ...
CVE-2025-21917	In the Linux kernel, the following vulnerability has been resolved: usb: renesas_usbhs: Flush the notify_hotplug_work When performing continuous unbind/bind operations on the USB drivers available on the Renesas RZ/G2L SoC, a kernel crash with the message "Unable to handle kernel NULL pointer dereference at virtual address" may occur. This issue points to the usbhsc_notify_hotplug() function. Flush the delayed work to avoid its execution when driver resources are unavailable.
CVE-2025-21918	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: Fix NULL pointer access Resources should be released only after all threads that utilize them have been destroyed. This commit ensures that resources are not released prematurely by waiting for the associated workqueue to complete before deallocating them.
CVE-2025-21919	In the Linux kernel, the following vulnerability has been resolved: sched/fair: Fix potential memory corruption in child_cfs_rq_on_list child_cfs_rq_on_list attempts to convert a 'prev' pointer to a cfs_rq. This 'prev' pointer can originate from struct rq's leaf_cfs_rq_list, making the conversion invalid and potentially leading to memory corruption. Depending on the relative positions of leaf_cfs_rq_list and the task group (tg) pointer within the struct, this can cause a memory fault or access garbage data. The issue arises in list_add_leaf_cfs_rq, where both cfs_rq->leaf_cfs_rq_list and rq->leaf_cfs_rq_list are added to the same leaf list. Also, rq->tmp_alone_branch can be set to rq->leaf_cfs_rq_list. This adds a check `if (prev == &rq->leaf_cfs_rq_list)` after the main conditional in child_cfs_rq_on_list. This ensures that the container_of operation will convert a correct cfs_rq struct. This check is sufficient because only cfs_rqs on the same CPU are added to the list, so verifying the 'prev' pointer against the current rq's list head is enough. Fixes a potential memory corruption issue that due to current struct layout might not be manifesting as a crash but could lead to unpredictable behavior when the layout changes.
CVE-2025-21920	In the Linux kernel, the following vulnerability has been resolved: vlan: enforce underlying device type Currently, VLAN devices can be created on top of non-ethernet devices. Besides the fact that it doesn't make much sense, this also causes a bug which leaks the address of a kernel function to usermode. When creating a VLAN device, we initialize GARP (garp_init_applicant) and MRP (mrp_init_applicant) for the underlying device. As part of the initialization process, we add the multicast address of each applicant to the underlying device, by calling dev_mc_add. __dev_mc_add uses dev->addr_len to determine the length of the new multicast address. This causes an out-of-bounds read if dev->addr_len is greater than 6, since the multicast addresses provided by GARP and MRP are only 6 bytes long. This behaviour can be reproduced using the following commands: ip tunnel add gretest mode ip6gre local ::1 remote ::2 dev lo ip l set up dev gretest ip link add link gretest name vlantest type vlan id 100 Then, the following command will display the address of garp_pdu_rcv: ip maddr show grep 01:80:c2:00:00:21 Fix the bug by enforcing the type of the underlying device during VLAN device initialization.

CVE ID	Description
CVE-2025-21922	In the Linux kernel, the following vulnerability has been resolved: ppp: Fix KMSAN uninit-value warning with bpf Syzbot caught an "KMSAN: uninit-value" warning [1], which is caused by the ppp driver not initializing a 2-byte header when using socket filter. The following code can generate a PPP filter BPF program: <code>"" struct bpf_program fp; pcap_t *handle; handle = pcap_open_dead(DLT_PPP_PPPOE, 65535); pcap_compile(handle, &fp, "ip and outbound", 0, 0); bpf_dump(&fp, 1); ""</code> Its output is: <code>"" (000) ldh [2] (001) jeq #0x21 jt 2 jf 5 (002) ldb [0] (003) jeq #0x1 jt 4 jf 5 (004) ret #65535 (005) ret #0 ""</code> Wen can find similar code at the following link: https://github.com/ppp-project/ppp/blob/master/pppd/options.c#L1680 The maintainer of this code repository is also the original maintainer of the ppp driver. As you can see the BPF program skips 2 bytes of data and then reads the 'Protocol' field to determine if it's an IP packet. Then it read the first byte of the first 2 bytes to determine the direction. The issue is that only the first byte indicating direction is initialized in current ppp driver code while the second byte is not initialized. For normal BPF programs generated by libpcap, uninitialized data won't be used, so it's not a problem. However, for carefully crafted BPF programs, such as those generated by syzkaller [2], which start reading from offset 0, the uninitialized data will be used and caught by KMSAN. [1] https://syzkaller.appspot.com/bug?extid=853242d9c9917165d791 [2] https://syzkaller.appspot.com/text?tag=ReproC&x=11994913980000
CVE-2025-21924	In the Linux kernel, the following vulnerability has been resolved: net: hns3: make sure ptp clock is unregister and freed if hclge_ptp_get_cycle returns an error During the initialization of ptp, hclge_ptp_get_cycle might return an error and returned directly without unregister clock and free it. To avoid that, call hclge_ptp_destroy_clock to unregister and free clock if hclge_ptp_get_cycle failed.
CVE-2025-21925	In the Linux kernel, the following vulnerability has been resolved: llc: do not use skb_get() before dev_queue_xmit() syzbot is able to crash hosts [1], using llc and devices not supporting IFF_TX_SKB_SHARING. In this case, e1000 driver calls eth_skb_pad(), while the skb is shared. Simply replace skb_get() by skb_clone() in net/llc/llc_s_ac.c Note that e1000 driver might have an issue with pktgen, because it does not clear IFF_TX_SKB_SHARING, this is an orthogonal change. We need to audit other skb_get() uses in net/llc. [1] kernel BUG at net/core/skbuff.c:2178 ! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 0 UID: 0 PID: 16371 Comm: syz.2.2764 Not tainted 6.14.0-rc4-syzkaller-00052-gac9c34d1e45a #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:pskb_expand_head+0x6ce/0x1240 net/core/skbuff.c:2178 Call Trace: <TASK> __skb_pad+0x18a/0x610 net/core/skbuff.c:2466 __skb_put_padto include/linux/skbuff.h:3843 [inline] skb_put_padto include/linux/skbuff.h:3862 [inline] eth_skb_pad include/linux/etherdevice.h:656 [inline] e1000_xmit_frame+0x2d99/0x5800 drivers/net/ethernet/intel/e1000/e1000_main.c:3128 __netdev_start_xmit include/linux/netdevice.h:5151 [inline] netdev_start_xmit include/linux/netdevice.h:5160 [inline] xmit_one net/core/dev.c:3806 [inline] dev_hard_start_xmit+0x9a/0x7b0 net/core/dev.c:3822 sch_direct_xmit+0x1ae/0xc30 net/sched/sch_generic.c:343 __dev_xmit_skb net/core/dev.c:4045 [inline] __dev_queue_xmit+0x13d4/0x43e0 net/core/dev.c:4621 dev_queue_xmit include/linux/netdevice.h:3313 [inline] llc_sap_action_send_test_c +0x268/0x320 net/llc/llc_s_ac.c:144 llc_exec_sap_trans_actions net/llc/llc_sap.c:153 [inline] llc_sap_next_state net/llc/llc_sap.c:182 [inline] llc_sap_state_process+0x239/0x510 net/llc/llc_sap.c:209 llc_ui_sendmsg+0xd0d/0x14e0 net/llc/af_llc.c:993 sock_sendmsg_nosec net/socket.c:718 [inline]
CVE-2025-21926	In the Linux kernel, the following vulnerability has been resolved: net: gso: fix ownership in __udp_gso_segment In __udp_gso_segment the skb destructor is removed before segmenting the skb but the socket reference is kept as-is. This is an issue if the original skb is later orphaned as we can hit the following bug: kernel BUG at ./include/linux/skbuff.h:3312! (skb_orphan) RIP: 0010:ip_rcv_core+0x8b2/0xca0 Call Trace: ip_rcv+0xab/0x6e0 __netif_receive_skb_one_core +0x168/0x1b0 process_backlog+0x384/0x1100 __napi_poll.constprop.0+0xa1/0x370 net_rx_action +0x925/0xe50 The above can happen following a sequence of events when using OpenVSwitch, when an OVS_ACTION_ATTR_USERSPACE action precedes an OVS_ACTION_ATTR_OUTPUT action: 1. OVS_ACTION_ATTR_USERSPACE is handled (in do_execute_actions): the skb goes through queue_gso_packets and then __udp_gso_segment, where its destructor is removed. 2. The segments' data are copied and sent to userspace. 3. OVS_ACTION_ATTR_OUTPUT is handled (in do_execute_actions) and the same original skb is sent to its path. 4. If it later hits skb_orphan, we hit the bug. Fix this by also removing the reference to the socket in __udp_gso_segment.
CVE-2025-21927	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: fix potential memory corruption in nvme_tcp_rcv_pdu() nvme_tcp_rcv_pdu() doesn't check the validity of the header length. When header digests are enabled, a target might send a packet with an invalid header length (e.g. 255), causing nvme_tcp_verify_hdgst() to access memory outside the allocated area and cause memory corruptions by overwriting it with the calculated digest. Fix this by rejecting packets with an unexpected header length.

CVE ID	Description
CVE-2025-21928	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: Fix use-after-free issue in ishtp_hid_remove() The system can experience a random crash a few minutes after the driver is removed. This issue occurs due to improper handling of memory freeing in the ishtp_hid_remove() function. The function currently frees the `driver_data` directly within the loop that destroys the HID devices, which can lead to accessing freed memory. Specifically, `hid_destroy_device()` uses `driver_data` when it calls `hid_ishtp_set_feature()` to power off the sensor, so freeing `driver_data` beforehand can result in accessing invalid memory. This patch resolves the issue by storing the `driver_data` in a temporary variable before calling `hid_destroy_device()`, and then freeing the `driver_data` after the device is destroyed.
CVE-2025-21929	In the Linux kernel, the following vulnerability has been resolved: HID: intel-ish-hid: Fix use-after-free issue in hid_ishtp_cl_remove() During the `rmmod` operation for the `intel_ishtp_hid` driver, a use-after-free issue can occur in the hid_ishtp_cl_remove() function. The function hid_ishtp_cl_deinit() is called before ishtp_hid_remove(), which can lead to accessing freed memory or resources during the removal process. Call Trace: ? ishtp_cl_send+0x168/0x220 [intel_ishtp] ? hid_output_report+0xe3/0x150 [hid] hid_ishtp_set_feature+0xb5/0x120 [intel_ishtp_hid] ishtp_hid_request+0x7b/0xb0 [intel_ishtp_hid] hid_hw_request+0x1f/0x40 [hid] sensor_hub_set_feature+0x11f/0x190 [hid_sensor_hub] _hid_sensor_power_state+0x147/0x1e0 [hid_sensor_trigger] hid_sensor_runtime_resume+0x22/0x30 [hid_sensor_trigger] sensor_hub_remove+0xa8/0xe0 [hid_sensor_hub] hid_device_remove+0x49/0xb0 [hid] hid_destroy_device+0x6f/0x90 [hid] ishtp_hid_remove+0x42/0x70 [intel_ishtp_hid] hid_ishtp_cl_remove+0x6b/0xb0 [intel_ishtp_hid] ishtp_cl_device_remove+0x4a/0x60 [intel_ishtp] ... Additionally, ishtp_hid_remove() is a HID level power off, which should occur before the ISHTP level disconnect. This patch resolves the issue by reordering the calls in hid_ishtp_cl_remove(). The function ishtp_hid_remove() is now called before hid_ishtp_cl_deinit().
CVE-2025-21930	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: mvm: don't try to talk to a dead firmware This fixes: bad state = 0 WARNING: CPU: 10 PID: 702 at drivers/net/wireless/intel/iwlwifi/iwl-trans.c:178 iwl_trans_send_cmd+0xba/0xe0 [iwlwifi] Call Trace: <TASK> ? __warn+0xca/0x1c0 ? iwl_trans_send_cmd+0xba/0xe0 [iwlwifi] 64fa9ad799a0e0d2ba53d4af93a53ad9a531f8d4] iwl_fw_dbg_clear_monitor_buf+0xd7/0x110 [iwlwifi] 64fa9ad799a0e0d2ba53d4af93a53ad9a531f8d4] _iwl_dbgfs_fw_dbg_clear_write+0xe2/0x120 [iwlwifi] 0e8adb18cea92d2c341766bcc10b18699290068a] Ask whether the firmware is alive before sending a command.
CVE-2025-21931	In the Linux kernel, the following vulnerability has been resolved: hwpoison, memory_hotplug: lock folio before unmap hwpoisoned folio Commit b15c87263a69 ("hwpoison, memory_hotplug: allow hwpoisoned pages to be offlined) add page poison checks in do_migrate_range in order to make offline hwpoisoned page possible by introducing isolate_lru_page and try_to_unmap for hwpoisoned page. However folio lock must be held before calling try_to_unmap. Add it to fix this problem. Warning will be produced if folio is not locked during unmap: -----[cut here]----- kernel BUG at ./include/linux/swaps.h:400! Internal error: Oops - BUG: 00000000f2000800 [#1] PREEMPT SMP Modules linked in: CPU: 4 UID: 0 PID: 411 Comm: bash Tainted: G W 6.13.0-rc1-00016-g3c434c7ee82a-dirty #41 Tainted: [W]=WARN Hardware name: QEMU QEMU Virtual Machine, BIOS 0.0.0 02/06/2015 pstate: 40400005 (nZcv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : try_to_unmap_one+0xb08/0xd3c lr : try_to_unmap_one+0x3dc/0xd3c Call trace: try_to_unmap_one+0xb08/0xd3c (P) try_to_unmap_one+0x3dc/0xd3c (L) rmap_walk_anon+0xdc/0x1f8 rmap_walk+0x3c/0x58 try_to_unmap+0x88/0x90 unmap_poisoned_folio+0x30/0xa8 do_migrate_range+0x4a0/0x568 offline_pages+0x5a4/0x670 memory_block_action+0x17c/0x374 memory_subsys_offline+0x3c/0x78 device_offline+0xa4/0xd0 state_store+0x8c/0xf0 dev_attr_store+0x18/0x2c sysfs_kf_write+0x44/0x54 kernfs_fop_write_iter+0x118/0x1a8 vfs_write+0x3a8/0x4bc ksys_write+0x6c/0xf8 __arm64_sys_write+0x1c/0x28 invoke_syscall+0x44/0x100 el0_svc_common.constprop.0+0x40/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x30/0xd0 el0t_64_sync_handler+0xc8/0xcc el0t_64_sync+0x198/0x19c Code: f9407be0 b5fff320 d4210000 17ffff97 (d4210000) ---[end trace 0000000000000000]---
CVE-2025-21934	In the Linux kernel, the following vulnerability has been resolved: rapidio: fix an API misuse when rio_add_net() fails rio_add_net() calls device_register() and fails when device_register() fails. Thus, put_device() should be used rather than kfree(). Add "mport->net = NULL;" to avoid a use after free issue.
CVE-2025-21935	In the Linux kernel, the following vulnerability has been resolved: rapidio: add check for rio_add_net() in rio_scan_alloc_net() The return value of rio_add_net() should be checked. If it fails, put_device() should be called to free the memory and give up the reference initialized in rio_add_net().
CVE-2025-21936	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Add check for mgmt_alloc_skb() in mgmt_device_connected() Add check for the return value of mgmt_alloc_skb() in mgmt_device_connected() to prevent null pointer dereference.

CVE ID	Description
CVE-2025-21937	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Add check for mgmt_alloc_skb() in mgmt_remote_name() Add check for the return value of mgmt_alloc_skb() in mgmt_remote_name() to prevent null pointer dereference.
CVE-2025-21938	In the Linux kernel, the following vulnerability has been resolved: mptcp: fix 'scheduling while atomic' in mptcp_pm_nl_append_new_local_addr If multiple connection requests attempt to create an implicit mptcp endpoint in parallel, more than one caller may end up in mptcp_pm_nl_append_new_local_addr because none found the address in local_addr_list during their call to mptcp_pm_nl_get_local_id. In this case, the concurrent new_local_addr calls may delete the address entry created by the previous caller. These deletes use synchronize_rcu, but this is not permitted in some of the contexts where this function may be called. During packet rcv, the caller may be in a rcu read critical section and have preemption disabled. An example stack: BUG: scheduling while atomic: swapper/2/0/0x00000302 Call Trace: <IRQ> dump_stack_lvl (lib/dump_stack.c:117 (discriminator 1)) dump_stack (lib/dump_stack.c:124) __schedule_bug (kernel/sched/core.c:5943) schedule_debug.constprop.0 (arch/x86/include/asm/preempt.h:33 kernel/sched/core.c:5970) __schedule (arch/x86/include/asm/jump_label.h:27 include/linux/jump_label.h:207 kernel/sched/features.h:29 kernel/sched/core.c:6621) schedule (arch/x86/include/asm/preempt.h:84 kernel/sched/core.c:6804 kernel/sched/core.c:6818) schedule_timeout (kernel/time/timer.c:2160) wait_for_completion (kernel/sched/completion.c:96 kernel/sched/completion.c:116 kernel/sched/completion.c:127 kernel/sched/completion.c:148) __wait_rcu_gp (include/linux/rcupdate.h:311 kernel/rcu/update.c:444) synchronize_rcu (kernel/rcu/tree.c:3609) mptcp_pm_nl_append_new_local_addr (net/mptcp/pm_netlink.c:966 net/mptcp/pm_netlink.c:1061) mptcp_pm_nl_get_local_id (net/mptcp/pm_netlink.c:1164) mptcp_pm_get_local_id (net/mptcp/pm.c:420) subflow_check_req (net/mptcp/subflow.c:98 net/mptcp/subflow.c:213) subflow_v4_route_req (net/mptcp/subflow.c:305) tcp_conn_request (net/ipv4/tcp_input.c:7216) subflow_v4_conn_request (net/mptcp/subflow.c:651) tcp_rcv_state_process (net/ipv4/tcp_input.c:6709) tcp_v4_do_rcv (net/ipv4/tcp_ipv4.c:1934) tcp_v4_rcv (net/ipv4/tcp_ipv4.c:2334) ip_protocol_deliver_rcu (net/ipv4/ip_input.c:205 (discriminator 1)) ip_local_deliver_finish (include/linux/rcupdate.h:813 net/ipv4/ip_input.c:234) ip_local_deliver (include/linux/netfilter.h:314 include/linux/netfilter.h:308 net/ipv4/ip_input.c:254) ip_sublist_rcv_finish (include/net/dst.h:461 net/ipv4/ip_input.c:580) ip_sublist_rcv (net/ipv4/ip_input.c:640) ip_list_rcv (net/ipv4/ip_input.c:675) __netif_receive_skb_list_core (net/core/dev.c:5583 net/core/dev.c:5631) netif_receive_skb_list_internal (net/core/dev.c:5685 net/core/dev.c:5774) napi_complete_done (include/linux/list.h:37 include/net/gro.h:449 include/net/gro.h:444 net/core/dev.c:6114) igb_poll (drivers/net/ethernet/intel/igb/igb_main.c:8244) igb __napi_poll (net/core/dev.c:6582) net_rx_action (net/core/dev.c:6653 net/core/dev.c:6787) handle_softirqs (kernel/softirq.c:553) __irq_exit_rcu (kernel/softirq.c:588 kernel/softirq.c:427 kernel/softirq.c:636) irq_exit_rcu (kernel/softirq.c:651) common_interrupt (arch/x86/kernel/irq.c:247 (discriminator 14)) </IRQ> This problem seems particularly prevalent if the user advertises an endpoint that has a different external vs internal address. In the case where the external address is advertised and multiple connections already exist, multiple subflow SYNs arrive in parallel which tends to trigger the race during creation of the first local_addr_list entries which have the internal address instead. Fix by skipping the replacement of an existing implicit local address if called via mptcp_pm_nl_get_local_id.
CVE-2025-21941	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix null check for pipe_ctx->plane_state in resource_build_scaling_params Null pointer dereference issue could occur when pipe_ctx->plane_state is null. The fix adds a check to ensure 'pipe_ctx->plane_state' is not null before accessing. This prevents a null pointer dereference. Found by code review. (cherry picked from commit 63e6a77ccf239337baa9b1e7787cde9fa0462092)

CVE ID	Description
CVE-2025-21943	In the Linux kernel, the following vulnerability has been resolved: gpio: aggregator: protect driver attr handlers against module unload Both new_device_store and delete_device_store touch module global resources (e.g. gpio_aggregator_lock). To prevent race conditions with module unload, a reference needs to be held. Add try_module_get() in these handlers. For new_device_store, this eliminates what appears to be the most dangerous scenario: if an id is allocated from gpio_aggregator_idr but platform_device_register has not yet been called or completed, a concurrent module unload could fail to unregister/delete the device, leaving behind a dangling platform device/GPIO forwarder. This can result in various issues. The following simple reproducer demonstrates these problems: <code>#!/bin/bash while ;; do # note: whether 'gpiochip0 0' exists or not does not matter. echo 'gpiochip0 0' > /sys/bus/platform/drivers/gpio-aggregator/new_device done & while ;; do modprobe gpio-aggregator modprobe -r gpio-aggregator done & wait Starting with the following warning, several kinds of warnings will appear and the system may become unstable: -----[cut here]----- list_del corruption, ffff888103e2e980->next is LIST_POISON1 (dead000000000100) WARNING: CPU: 1 PID: 1327 at lib/list_debug.c:56 __list_del_entry_valid_or_report+0xa3/0x120 [...] RIP: 0010: __list_del_entry_valid_or_report+0xa3/0x120 [...] Call Trace: <TASK> ? __list_del_entry_valid_or_report+0xa3/0x120 ? __warn.cold+0x93/0xf2 ? __list_del_entry_valid_or_report+0xa3/0x120 ? report_bug+0xe6/0x170 ? __irq_work_queue_local+0x39/0xe0 ? handle_bug+0x58/0x90 ? exc_invalid_op+0x13/0x60 ? asm_exc_invalid_op+0x16/0x20 ? __list_del_entry_valid_or_report+0xa3/0x120 gpiod_remove_lookup_table+0x22/0x60 new_device_store+0x315/0x350 [gpio_aggregator] kernfs_fop_write_iter+0x137/0x1f0 vfs_write+0x262/0x430 ksys_write+0x60/0xd0 do_syscall_64+0x6c/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e [...] </TASK> ---[end trace 0000000000000000]---</code>
CVE-2025-21944	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix bug on trap in smb2_lock If lock count is greater than 1, flags could be old value. It should be checked with flags of smb_lock, not flags. It will cause bug-on trap from locks_free_lock in error handling routine.
CVE-2025-21945	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in smb2_lock If smb_lock->zero_len has value, ->llist of smb_lock is not delete and flock is old one. It will cause use-after-free on error handling routine.
CVE-2025-21946	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix out-of-bounds in parse_sec_desc() If osidoffset, gsidoffset and dacloffset could be greater than smb_ntsd struct size. If it is smaller, It could cause slab-out-of-bounds. And when validating sid, It need to check it included subauth array size.
CVE-2025-21947	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix type confusion via race condition when using ipc_msg_send_request req->handle is allocated using ksmbd_acquire_id(&ipc_ida), based on ida_alloc. req->handle from ksmbd_ipc_login_request and FSCTL_PIPE_TRANSCEIVE ioctl can be same and it could lead to type confusion between messages, resulting in access to unexpected parts of memory after an incorrect delivery. ksmbd check type of ipc response but missing add continue to check next ipc reponse.

CVE ID	Description
CVE-2025-21948	In the Linux kernel, the following vulnerability has been resolved: HID: appleir: Fix potential NULL dereference at raw event handle Syzkaller reports a NULL pointer dereference issue in input_event(). BUG: KASAN: null-ptr-deref in instrument_atomic_read include/linux/instrumented.h:68 [inline] BUG: KASAN: null-ptr-deref in _test_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline] BUG: KASAN: null-ptr-deref in is_event_supported drivers/input/input.c:67 [inline] BUG: KASAN: null-ptr-deref in input_event+0x42/0xa0 drivers/input/input.c:395 Read of size 8 at addr 0000000000000028 by task syz-executor199/2949 CPU: 0 UID: 0 PID: 2949 Comm: syz-executor199 Not tainted 6.13.0-rc4-syzkaller-00076-gf097a36ef88d #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <IRQ> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 kasan_report+0xd9/0x110 mm/kasan/report.c:602 check_region_inline mm/kasan/generic.c:183 [inline] kasan_check_range+0xef/0x1a0 mm/kasan/generic.c:189 instrument_atomic_read include/linux/instrumented.h:68 [inline] _test_bit include/asm-generic/bitops/instrumented-non-atomic.h:141 [inline] is_event_supported drivers/input/input.c:67 [inline] input_event+0x42/0xa0 drivers/input/input.c:395 input_report_key include/linux/input.h:439 [inline] key_down drivers/hid/hid-appleir.c:159 [inline] appleir_raw_event+0x3e5/0x5e0 drivers/hid/hid-appleir.c:232 __hid_input_report.constprop.0+0x312/0x440 drivers/hid/hid-core.c:2111 hid_ctrl+0x49f/0x550 drivers/hid/usbhid/hid-core.c:484 __usb_hcd_giveback_urb+0x389/0x6e0 drivers/usb/core/hcd.c:1650 usb_hcd_giveback_urb+0x396/0x450 drivers/usb/core/hcd.c:1734 dummy_timer+0x17f7/0x3960 drivers/usb/gadget/udc/dummy_hcd.c:1993 __run_hrtimer kernel/time/hrtimer.c:1739 [inline] __hrtimer_run_queues+0x20a/0xae0 kernel/time/hrtimer.c:1803 hrtimer_run_softirq+0x17d/0x350 kernel/time/hrtimer.c:1820 handle_softirqs+0x206/0x8d0 kernel/softirq.c:561 __do_softirq kernel/softirq.c:595 [inline] invoke_softirq kernel/softirq.c:435 [inline] __irq_exit_rcu+0xfa/0x160 kernel/softirq.c:662 irq_exit_rcu+0x9/0x30 kernel/softirq.c:678 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1049 [inline] sysvec_apic_timer_interrupt+0x90/0xb0 arch/x86/kernel/apic/apic.c:1049 <IRQ> <TASK> asm_sysvec_apic_timer_interrupt+0x1a/0x20 arch/x86/include/asm/irqentry.h:702 _mod_timer+0x8f6/0xcd0 kernel/time/timer.c:1185 add_timer+0x62/0x90 kernel/time/timer.c:1295 schedule_timeout+0x11f/0x280 kernel/time/sleep_timeout.c:98 usbhid_wait_io+0x1c7/0x380 drivers/hid/usbhid/hid-core.c:645 usbhid_init_reports+0x19f/0x390 drivers/hid/usbhid/hid-core.c:784 hiddev_ioctl+0x1133/0x15b0 drivers/hid/usbhid/hiddev.c:794 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl fs/ioctl.c:892 [inline] __x64_sys_ioctl+0x190/0x200 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> This happens due to the malformed report items sent by the emulated device which results in a report, that has no fields, being added to the report list. Due to this appleir_input_configured() is never called, hidinput_connect() fails which results in the HID_CLAIMED_INPUT flag is not being set. However, it does not make appleir_probe() fail and lets the event callback to be called without the associated input device. Thus, add a check for the HID_CLAIMED_INPUT flag and leave the event hook early if the driver didn't claim any input_dev for some reason. Moreover, some other hid drivers accessing input_dev in their event callbacks do have similar checks, too. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-21949	In the Linux kernel, the following vulnerability has been resolved: LoongArch: Set hugetlb mmap base address aligned with pmd size With ltp test case "testcases/bin/hugefork02", there is a dmesg error report message such as: kernel BUG at mm/hugetlb.c:5550! Oops - BUG[#1]: CPU: 0 UID: 0 PID: 1517 Comm: hugefork02 Not tainted 6.14.0-rc2+ #241 Hardware name: QEMU QEMU Virtual Machine, BIOS unknown 2/2/2022 pc 90000000004eaf1c ra 9000000000485538 tp 900000010edbc000 sp 900000010edbf940 a0 900000010edbf00 a1 9000000108d20280 a2 00007fffe9474000 a3 00007ffff3474000 a4 0000000000000000 a5 0000000000000003 a6 000000000003cadd3 a7 0000000000000000 t0 0000000001ffffff t1 0000000001474000 t2 900000010ecd7900 t3 00007fffe9474000 t4 00007fffe9474000 t5 0000000000000040 t6 900000010edbf00 t7 0000000000000001 t8 0000000000000005 u0 90000000004849d0 s9 900000010edbf00 s0 9000000108d20280 s1 00007fffe9474000 s2 0000000002000000 s3 9000000108d20280 s4 9000000002b38b10 s5 900000010edbf00 s6 00007ffff3474000 s7 00000000000000406 s8 900000010edbf08 ra: 9000000000485538 unmap_vmas+0x130/0x218 ERA: 90000000004eaf1c __unmap_hugepage_range+0x6f4/0x7d0 PRMD: 00000004 (PPLV0 +PIE -PWE) EUEN: 00000007 (+FPE +SXE +ASXE -BTE) ECFG: 00071c1d (LIE=0,2-4,10-12 VS=7) ESTAT: 000c0000 [BRK] (IS= ECode=12 EsubCode=0) PRID: 0014c010 (Loongson-64bit, Loongson-3A5000) Process hugefork02 (pid: 1517, threadinfo=00000000a670eaf4, task=000000007a95fc64) Call Trace: [<90000000004eaf1c>] __unmap_hugepage_range+0x6f4/0x7d0 [<9000000000485534>] unmap_vmas+0x12c/0x218 [<9000000000494068>] exit_mmap+0xe0/0x308 [<900000000025fdc4>] mmput+0x74/0x180 [<900000000026a284>] do_exit+0x294/0x898 [<900000000026aa30>] do_group_exit+0x30/0x98 [<900000000027bed4>] get_signal+0x83c/0x868 [<90000000002457b4>] arch_do_signal_or_restart+0x54/0xf0a [<90000000015795e8>] irqentry_exit_to_user_mode+0xb8/0x138 [<90000000002572d0>] tlb_do_page_fault_1+0x114/0x1b4 The problem is that base address allocated from hugetlbfs is not aligned with pmd size. Here add a checking for hugetlbfs and align base address with pmd size. After this patch the test case "testcases/bin/hugefork02" passes to run. This is similar to the commit 7f24cbc9c4d42db8a3c8484d1 ("mm/mmap: teach generic_get_unmapped_area{ _topdown } to handle hugetlb mappings").

CVE ID	Description
CVE-2025-21950	In the Linux kernel, the following vulnerability has been resolved: drivers: virt: acrn: hsm: Use kzalloc to avoid info leak in pmcmd_ioctl In the "pmcmd_ioctl" function, three memory objects allocated by kmalloc are initialized by "hcall_get_cpu_state", which are then copied to user space. The initializer is indeed implemented in "acrn_hypervcall2" (arch/x86/include/asm/acrn.h). There is a risk of information leakage due to uninitialized bytes.
CVE-2025-21951	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: host: pci_generic: Use pci_try_reset_function() to avoid deadlock There are multiple places from where the recovery work gets scheduled asynchronously. Also, there are multiple places where the caller waits synchronously for the recovery to be completed. One such place is during the PM shutdown() callback. If the device is not alive during recovery_work, it will try to reset the device using pci_reset_function(). This function internally will take the device_lock() first before resetting the device. By this time, if the lock has already been acquired, then recovery_work will get stalled while waiting for the lock. And if the lock was already acquired by the caller which waits for the recovery_work to be completed, it will lead to deadlock. This is what happened on the X1E80100 CRD device when the device died before shutdown() callback. Driver core calls the driver's shutdown() callback while holding the device_lock() leading to deadlock. And this deadlock scenario can occur on other paths as well, like during the PM suspend() callback, where the driver core would hold the device_lock() before calling driver's suspend() callback. And if the recovery_work was already started, it could lead to deadlock. This is also observed on the X1E80100 CRD. So to fix both issues, use pci_try_reset_function() in recovery_work. This function first checks for the availability of the device_lock() before trying to reset the device. If the lock is available, it will acquire it and reset the device. Otherwise, it will return -EAGAIN. If that happens, recovery_work will fail with the error message "Recovery failed" as not much could be done.
CVE-2025-21955	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent connection release during oplock break notification ksmbd_work could be freed when after connection release. Increment r_count of ksmbd_conn to indicate that requests are not finished yet and to not release the connection.
CVE-2025-21956	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Assign normalized_pix_clk when color depth = 14 [WHY & HOW] A warning message "WARNING: CPU: 4 PID: 459 at ... /dc_resource.c:3397 calculate_phy_pix_clks+0xef/0x100 [amdgpu]" occurs because the display_color_depth == COLOR_DEPTH_141414 is not handled. This is observed in Radeon RX 6600 XT. It is fixed by assigning pix_clk * (14 * 3) / 24 - same as the rests. Also fixes the indentation in get_norm_pix_clk. (cherry picked from commit 274a87eb389f58eddc5659ab0b180b37e92775)
CVE-2025-21957	In the Linux kernel, the following vulnerability has been resolved: scsi: qla1280: Fix kernel oops when debug level > 2 A null dereference or oops exception will eventually occur when qla1280.c driver is compiled with DEBUG_QLA1280 enabled and ql_debug_level > 2. I think its clear from the code that the intention here is sg_dma_len(s) not length of sg_next(s) when printing the debug info.

CVE ID	Description
CVE-2025-21959	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_conncount: Fully initialize struct nf_conncount_tuple in insert_tree() Since commit b36e4523d4d5 ("netfilter: nf_conncount: fix garbage collection confirm race"), `cpu` and `jiffies32` were introduced to the struct nf_conncount_tuple. The commit made nf_conncount_add() initialize `conn->cpu` and `conn->jiffies32` when allocating the struct. In contrast, count_tree() was not changed to initialize them. By commit 34848d5c896e ("netfilter: nf_conncount: Split insert and traversal"), count_tree() was split and the relevant allocation code now resides in insert_tree(). Initialize `conn->cpu` and `conn->jiffies32` in insert_tree(). BUG: KMSAN: uninit-value in find_or_evict net/netfilter/nf_conncount.c:117 [inline] BUG: KMSAN: uninit-value in __nf_conncount_add+0xd9c/0x2850 net/netfilter/nf_conncount.c:143 find_or_evict net/netfilter/nf_conncount.c:117 [inline] __nf_conncount_add+0xd9c/0x2850 net/netfilter/nf_conncount.c:143 count_tree net/netfilter/nf_conncount.c:438 [inline] nf_conncount_count+0x82f/0x1e80 net/netfilter/nf_conncount.c:521 connlimit_mt+0x7f6/0xbd0 net/netfilter/xt_connlimit.c:72 __nft_match_eval net/netfilter/nft_compat.c:403 [inline] nft_match_eval+0x1a5/0x300 net/netfilter/nft_compat.c:433 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x426/0x2290 net/netfilter/nf_tables_core.c:288 nft_do_chain_ip4+0x1a5/0x230 net/netfilter/nft_chain_filter.c:23 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626 nf_hook_slow_list+0x24d/0x860 net/netfilter/core.c:663 NF_HOOK_LIST include/linux/netfilter.h:350 [inline] ip_sublist_rcv+0x17b7/0x17f0 net/ipv4/ip_input.c:633 ip_list_rcv+0x9ef/0xa40 net/ipv4/ip_input.c:669 __netif_receive_skb_list_ptype net/core/dev.c:5936 [inline] __netif_receive_skb_list_core+0x15c5/0x1670 net/core/dev.c:5983 __netif_receive_skb_list net/core/dev.c:6035 [inline] netif_receive_skb_list_internal+0x1085/0x1700 net/core/dev.c:6126 netif_receive_skb_list+0x5a/0x460 net/core/dev.c:6178 xdp_rcv_frames net/bpf/test_run.c:280 [inline] xdp_test_run_batch net/bpf/test_run.c:361 [inline] bpf_test_run_xdp_live+0x2e86/0x3480 net/bpf/test_run.c:390 bpf_prog_test_run_xdp+0xf1d/0x1ae0 net/bpf/test_run.c:1316 bpf_prog_test_run+0x5e5/0xa30 kernel/bpf/syscall.c:4407 __sys_bpf+0x6aa/0xd90 kernel/bpf/syscall.c:5813 __do_sys_bpf kernel/bpf/syscall.c:5902 [inline] __se_sys_bpf kernel/bpf/syscall.c:5900 [inline] __ia32_sys_bpf+0xa0/0xe0 kernel/bpf/syscall.c:5900 ia32_sys_call+0x394d/0x4180 arch/x86/include/generated/asm/syscalls_32.h:358 do_syscall_32_irqs_on arch/x86/entry/common.c:165 [inline] __do_fast_syscall_32+0xb0/0x110 arch/x86/entry/common.c:387 do_fast_syscall_32+0x38/0x80 arch/x86/entry/common.c:412 do_SYSENTER_32+0x1f/0x30 arch/x86/entry/common.c:450 entry_SYSENTER_compat_after_hwframe+0x84/0x8e Uninit was created at: slab_post_alloc_hook mm/slab.c:4121 [inline] slab_alloc_node mm/slab.c:4164 [inline] kmem_cache_alloc_noprof+0x915/0xe10 mm/slab.c:4171 insert_tree net/netfilter/nf_conncount.c:372 [inline] count_tree net/netfilter/nf_conncount.c:450 [inline] nf_conncount_count+0x1415/0x1e80 net/netfilter/nf_conncount.c:521 connlimit_mt+0x7f6/0xbd0 net/netfilter/xt_connlimit.c:72 __nft_match_eval net/netfilter/nft_compat.c:403 [inline] nft_match_eval+0x1a5/0x300 net/netfilter/nft_compat.c:433 expr_call_ops_eval net/netfilter/nf_tables_core.c:240 [inline] nft_do_chain+0x426/0x2290 net/netfilter/nf_tables_core.c:288 nft_do_chain_ip4+0x1a5/0x230 net/netfilter/nft_chain_filter.c:23 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf4/0x400 net/netfilter/core.c:626 nf_hook_slow_list+0x24d/0x860 net/netfilter/core.c:663 NF_HOOK_LIST include/linux/netfilter.h:350 [inline] ip_sublist_rcv+0x17b7/0x17f0 net/ipv4/ip_input.c:633 ip_list_rcv+0x9ef/0xa40 net/ip ---truncated---

CVE ID	Description
CVE-2025-21960	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: do not update checksum in bnxt_xdp_build_skb() The bnxt_rx_pkt() updates ip_summed value at the end if checksum offload is enabled. When the XDP-MB program is attached and it returns XDP_PASS, the bnxt_xdp_build_skb() is called to update skb_shared_info. The main purpose of bnxt_xdp_build_skb() is to update skb_shared_info, but it updates ip_summed value too if checksum offload is enabled. This is actually duplicate work. When the bnxt_rx_pkt() updates ip_summed value, it checks if ip_summed is CHECKSUM_NONE or not. It means that ip_summed should be CHECKSUM_NONE at this moment. But ip_summed may already be updated to CHECKSUM_UNNECESSARY in the XDP-MB-PASS path. So the by skb_checksum_none_assert() WARNS about it. This is duplicate work and updating ip_summed in the bnxt_xdp_build_skb() is not needed. Splat looks like: WARNING: CPU: 3 PID: 5782 at ./include/linux/skbuff.h:5155 bnxt_rx_pkt+0x479b/0x7610 [bnxt_en] Modules linked in: bnxt_re bnxt_en rdma_ucm rdma_cm iw_cm ib_cm ib_uverbs veth xt_nat xt_tcpudp xt_conntrack nft_chain_nat xt_MASQUERADE nf_] CPU: 3 UID: 0 PID: 5782 Comm: socat Tainted: G W 6.14.0-rc4+ #27 Tainted: [W]=WARN Hardware name: ASUS System Product Name/PRIME Z690-P D4, BIOS 0603 11/01/2021 RIP: 0010:bnxt_rx_pkt+0x479b/0x7610 [bnxt_en] Code: 54 24 0c 4c 89 f1 4c 89 ff c1 ea 1f ff d3 0f 1f 00 49 89 c6 48 85 c0 0f 84 4c e5 ff ff 48 89 c7 e8 ca 3d a0 c8 e9 8f f4 ff ff <0f> 0b f RSP: 0018:ffff88881ba09928 EFLAGS: 00010202 RAX: 0000000000000000 RBX: 00000000c7590303 RCX: 0000000000000000 RDX: 1ffff1104e7d1610 RSI: 0000000000000001 RDI: ffff8881c91300b8 RBP: ffff8881ba09b28 R08: ffff888273e8b0d0 R09: ffff888273e8b070 R10: ffff888273e8b010 R11: ffff888278b0f000 R12: ffff888273e8b080 R13: ffff8881c9130e00 R14: ffff8881505d3800 R15: ffff888273e8b000 FS: 00007f5a2e7be080(0000) GS:ffff8881ba00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fff2e708ff8 CR3: 000000013e3b0000 CR4: 00000000007506f0 PKRU: 55555554 Call Trace: <IRQ> ? __warn+0xcd/0x2f0 ? bnxt_rx_pkt+0x479b/0x7610 ? report_bug+0x326/0x3c0 ? handle_bug+0x53/0xa0 ? exc_invalid_op+0x14/0x50 ? asm_exc_invalid_op+0x16/0x20 ? bnxt_rx_pkt+0x479b/0x7610 ? bnxt_rx_pkt+0x3e41/0x7610 ? __pfx_bnxt_rx_pkt+0x10/0x10 ? napi_complete_done+0x2cf/0x7d0 __bnxt_poll_work+0x4e8/0x1220 ? __pfx__bnxt_poll_work+0x10/0x10 ? __pfx_mark_lock.part.0+0x10/0x10 bnxt_poll_p5+0x36a/0xfa0 ? __pfx_bnxt_poll_p5+0x10/0x10 __napi_poll.constprop.0+0xa0/0x440 net_rx_action+0x899/0xd00 ... Following ping.py patch adds xdp-mb-pass case. so ping.py is going to be able to reproduce this issue.
CVE-2025-21961	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: fix truesize for mb-xdp-pass case When mb-xdp is set and return is XDP_PASS, packet is converted from xdp_buff to sk_buff with xdp_update_skb_shared_info() in bnxt_xdp_build_skb(). bnxt_xdp_build_skb() passes incorrect truesize argument to xdp_update_skb_shared_info(). The truesize is calculated as BNXT_RX_PAGE_SIZE * sinfo->nr_frags but the skb_shared_info was wiped by napi_build_skb() before. So it stores sinfo->nr_frags before bnxt_xdp_build_skb() and use it instead of getting skb_shared_info from xdp_get_shared_info_from_buff(). Splat looks like: -----[cut here]----- WARNING: CPU: 2 PID: 0 at net/core/skbuff.c:6072 skb_try_coalesce+0x504/0x590 Modules linked in: xt_nat xt_tcpudp veth af_packet xt_conntrack nft_chain_nat xt_MASQUERADE nf_conntrack_netlink xfrm_user xt_addrtype nft_coms CPU: 2 UID: 0 PID: 0 Comm: swapper/2 Not tainted 6.14.0-rc2+ #3 RIP: 0010:skb_try_coalesce+0x504/0x590 Code: 4b fd ff ff 49 8b 34 24 40 80 e6 40 0f 84 3d fd ff ff 49 8b 74 24 48 40 f6 c6 01 0f 84 2e fd ff ff 48 8d 4e ff e9 25 fd ff ff <0f> 0b e9 RSP: 0018:ffffb62c4120caa8 EFLAGS: 00010287 RAX: 0000000000000003 RBX: ffff8b62c4120cb14 RCX: 00000000000000ec0 RDX: 0000000000001000 RSI: fffffa06e5d7dc000 RDI: 0000000000000003 RBP: fffffa06e5d7ddec0 R08: fffffa06e6120a800 R09: fffffa06e7a119900 R10: 00000000000002310 R11: fffffa06e5d7ddec0 R12: fffffa4360575f740 R13: fffffa43600000000 R14: 0000000000000002 R15: 0000000000000002 FS: 0000000000000000(0000) GS:ffffa0755f700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f147b76b0f8 CR3: 00000001615d4000 CR4: 00000000007506f0 PKRU: 55555554 Call Trace: <IRQ> ? __warn+0x84/0x130 ? skb_try_coalesce+0x504/0x590 ? report_bug+0x18a/0x1a0 ? handle_bug+0x53/0x90 ? exc_invalid_op+0x14/0x70 ? asm_exc_invalid_op+0x16/0x20 ? skb_try_coalesce+0x504/0x590 inet_frag_reasm_finish+0x11f/0x2e0 ip_defrag+0x37a/0x900 ip_local_deliver+0x51/0x120 ip_sublist_rcv_finish+0x64/0x70 ip_sublist_rcv+0x179/0x210 ip_list_rcv+0xf9/0x130 How to reproduce: <Node A> ip link set \$interface1 xdp obj xdp_pass.o ip link set \$interface1 mtu 9000 up ip a 10.0.0.1/24 dev \$interface1 <Node B> ip link set \$interface2 mtu 9000 up ip a 10.0.0.2/24 dev \$interface2 ping 10.0.0.1 -s 65000 Following ping.py patch adds xdp-mb-pass case. so ping.py is going to be able to reproduce this issue.
CVE-2025-21962	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix integer overflow while processing closetimeo mount option User-provided mount parameter closetimeo of type u32 is intended to have an upper limit, but before it is validated, the value is converted from seconds to jiffies which can lead to an integer overflow. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-21963	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix integer overflow while processing acdirmax mount option User-provided mount parameter acdirmax of type u32 is intended to have an upper limit, but before it is validated, the value is converted from seconds to jiffies which can lead to an integer overflow. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2025-21964	In the Linux kernel, the following vulnerability has been resolved: cifs: Fix integer overflow while processing acregmax mount option User-provided mount parameter acregmax of type u32 is intended to have an upper limit, but before it is validated, the value is converted from seconds to jiffies which can lead to an integer overflow. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-21966	In the Linux kernel, the following vulnerability has been resolved: dm-flakey: Fix memory corruption in optional corrupt_bio_byte feature Fix memory corruption due to incorrect parameter being passed to bio_init
CVE-2025-21967	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in ksmbd_free_work_struct ->interim_entry of ksmbd_work could be deleted after oplock is freed. We don't need to manage it with linked list. The interim request could be immediately sent whenever a oplock break wait is needed.
CVE-2025-21968	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix slab-use-after-free on hdcp_work [Why] A slab-use-after-free is reported when HDCP is destroyed but the property_validate_dwork queue is still running. [How] Cancel the delayed work when destroying workqueue. (cherry picked from commit 725a04ba5a95e89c89633d4322430cfbca7ce128)
CVE-2025-21969	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: L2CAP: Fix slab-use-after-free Read in l2cap_send_cmd After the hci sync command releases l2cap_conn, the hci receive data work queue references the released l2cap_conn when sending to the upper layer. Add hci dev lock to the hci receive data work queue to synchronize the two. [1] BUG: KASAN: slab-use-after-free in l2cap_send_cmd+0x187/0x8d0 net/bluetooth/l2cap_core.c:954 Read of size 8 at addr ffff8880271a4000 by task kworker/u9:2/5837 CPU: 0 UID: 0 PID: 5837 Comm: kworker/u9:2 Not tainted 6.13.0-rc5-syzkaller-00163-gab75170520d4 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: hci1 hci_rx_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report+0x143/0x180 mm/kasan/report.c:602 l2cap_build_cmd net/bluetooth/l2cap_core.c:2964 [inline] l2cap_send_cmd+0x187/0x8d0 net/bluetooth/l2cap_core.c:954 l2cap_sig_send_rej net/bluetooth/l2cap_core.c:5502 [inline] l2cap_sig_channel net/bluetooth/l2cap_core.c:5538 [inline] l2cap_rcv_frame+0x221f/0x10db0 net/bluetooth/l2cap_core.c:6817 hci_acldata_packet net/bluetooth/hci_core.c:3797 [inline] hci_rx_work+0x508/0xdb0 net/bluetooth/hci_core.c:4040 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa66/0x1840 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> Allocated by task 5837: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloccache_noprof+0x243/0x390 mm/slub.c:4329 kmallocc_noprof include/linux/slab.h:901 [inline] kzalloc_noprof include/linux/slab.h:1037 [inline] l2cap_conn_add+0xa9/0x8e0 net/bluetooth/l2cap_core.c:6860 l2cap_connect_cfm+0x115/0x1090 net/bluetooth/l2cap_core.c:7239 hci_connect_cfm include/net/bluetooth/hci_core.h:2057 [inline] hci_remote_features_evt+0x68e/0xac0 net/bluetooth/hci_event.c:3726 hci_event_func net/bluetooth/hci_event.c:7473 [inline] hci_event_packet+0xac2/0x1540 net/bluetooth/hci_event.c:7525 hci_rx_work+0x3f3/0xdb0 net/bluetooth/hci_core.c:4035 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa66/0x1840 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 Freed by task 54: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3f/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x40/0x50 mm/kasan/generic.c:582 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x59/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2353 [inline] slab_free mm/slub.c:4613 [inline] kfree+0x196/0x430 mm/slub.c:4761 l2cap_connect_cfm+0xcc/0x1090 net/bluetooth/l2cap_core.c:7235 hci_connect_cfm include/net/bluetooth/hci_core.h:2057 [inline] hci_conn_failed+0x287/0x400 net/bluetooth/hci_conn.c:1266 hci_abort_conn_sync+0x56c/0x11f0 net/bluetooth/hci_sync.c:5603 hci_cmd_sync_work+0x22b/0x400 net/bluetooth/hci_sync.c:332 process_one_work kernel/workqueue.c:3229 [inline] process_scheduled_works+0xa66/0x1840 kernel/workqueue.c:3310 worker_thread+0x870/0xd30 kernel/workqueue.c:3391 kthread+0x2f0/0x390 kernel/kthread.c:389 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:147 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entr ---truncated---

CVE ID	Description
CVE-2025-21970	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Bridge, fix the crash caused by LAG state check When removing LAG device from bridge, NETDEV_CHANGEUPPER event is triggered. Driver finds the lower devices (PFs) to flush all the offloaded entries. And mlx5_lag_is_shared_fdb is checked, it returns false if one of PF is unloaded. In such case, mlx5_esw_bridge_lag_rep_get() and its caller return NULL, instead of the alive PF, and the flush is skipped. Besides, the bridge fdb entry's lastuse is updated in mlx5 bridge event handler. But this SWITCHDEV_FDB_ADD_TO_BRIDGE event can be ignored in this case because the upper interface for bond is deleted, and the entry will never be aged because lastuse is never updated. To make things worse, as the entry is alive, mlx5 bridge workqueue keeps sending that event, which is then handled by kernel bridge notifier. It causes the following crash when accessing the passed bond netdev which is already destroyed. To fix this issue, remove such checks. LAG state is already checked in commit 15f8f168952f ("net/mlx5: Bridge, verify LAG state when adding bond to bridge"), driver still need to skip offload if LAG becomes invalid state after initialization. Oops: stack segment: 0000 [#1] SMP CPU: 3 UID: 0 PID: 23695 Comm: kworker/u40:3 Tainted: G OE 6.11.0_mlnx #1 Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 Workqueue: mlx5_bridge_wq mlx5_esw_bridge_update_work [mlx5_core] RIP: 0010:br_switchdev_event+0x2c/0x110 [bridge] Code: 44 00 00 48 8b 02 48 f7 00 00 02 00 00 74 69 41 54 55 53 48 83 ec 08 48 8b a8 08 01 00 00 48 85 ed 74 4a 48 83 fe 02 48 89 d3 <4c> 8b 65 00 74 23 76 49 48 83 fe 05 74 7e 48 83 fe 06 75 2f 0f b7 RSP: 0018:ffffc900092cfd0a EFLAGS: 00010297 RAX: ffff888123bfe000 RBX: ffff888123bfe000 RCX: 00000000ffffff RDX: ffff888123bfe000 RSI: 0000000000000001 RDI: ffff888123bfe000 RBP: 6669746f6e690a30 R08: 0000000000000000 R09: ffff888123ae92c8 R10: 0000000000000000 R11: fefefefefefeff R12: ffff888123ae9c60 R13: 0000000000000001 R14: ffff888123bfe000 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff88852c980000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000000000033 CR2: 00007f15914c8734 CR3: 0000000002830005 CR4: 000000000077ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000040 PKRU: 55555554 Call Trace: <TASK> ? __die_body+0x1a/0x60 ? die+0x38/0x60 ? do_trap+0x10b/0x120 ? do_error_trap+0x64/0xa0 ? exc_stack_segment+0x33/0x50 ? asm_exc_stack_segment+0x22/0x30 ? br_switchdev_event+0x2c/0x110 [bridge] ? sched_balance_newidle.isra.149+0x248/0x390 notifier_call_chain+0x4b/0xa0 atomic_notifier_call_chain+0x16/0x20 mlx5_esw_bridge_update+0xec/0x170 [mlx5_core] mlx5_esw_bridge_update_work+0x19/0x40 [mlx5_core] process_scheduled_works+0x81/0x390 worker_thread+0x106/0x250 ? bh_worker+0x110/0x110 kthread+0xb7/0xe0 ? kthread_park+0x80/0x80 ret_from_fork+0x2d/0x50 ? kthread_park+0x80/0x80 ret_from_fork_asm+0x11/0x20 </TASK>
CVE-2025-21971	In the Linux kernel, the following vulnerability has been resolved: net_sched: Prevent creation of classes with TC_H_ROOT The function qdisc_tree_reduce_backlog() uses TC_H_ROOT as a termination condition when traversing up the qdisc tree to update parent backlog counters. However, if a class is created with classid TC_H_ROOT, the traversal terminates prematurely at this class instead of reaching the actual root qdisc, causing parent statistics to be incorrectly maintained. In case of DRR, this could lead to a crash as reported by Mingi Cho. Prevent the creation of any Qdisc class with classid TC_H_ROOT (0xFFFFFFFF) across all qdisc types, as suggested by Jamal.
CVE-2025-21972	In the Linux kernel, the following vulnerability has been resolved: net: mctp: unshare packets when reassembling Ensure that the frag_list used for reassembly isn't shared with other packets. This avoids incorrect reassembly when packets are cloned, and prevents a memory leak due to circular references between fragments and their skb_shared_info. The upcoming MCTP-over-USB driver uses skb_clone which can trigger the problem - other MCTP drivers don't share SKBs. A kunit test is added to reproduce the issue.
CVE-2025-21975	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: handle errors in mlx5_chains_create_table() In mlx5_chains_create_table(), the return value of mlx5_get_fdb_sub_ns() and mlx5_get_flow_namespace() must be checked to prevent NULL pointer dereferences. If either function fails, the function should log error message with mlx5_core_warn() and return error pointer.

CVE ID	Description
CVE-2025-21976	In the Linux kernel, the following vulnerability has been resolved: fbdev: hyperv_fb: Allow graceful removal of framebuffer When a Hyper-V framebuffer device is unbind, hyperv_fb driver tries to release the framebuffer forcefully. If this framebuffer is in use it produce the following WARN and hence this framebuffer is never released. [44.111220] WARNING: CPU: 35 PID: 1882 at drivers/video/fbdev/core/fb_info.c:70 framebuffer_release+0x2c/0x40 <snip> [44.111289] Call Trace: [44.111290] <TASK> [44.111291] ? show_regs+0x6c/0x80 [44.111295] ? __warn+0x8d/0x150 [44.111298] ? framebuffer_release+0x2c/0x40 [44.111300] ? report_bug+0x182/0x1b0 [44.111303] ? handle_bug+0x6e/0xb0 [44.111306] ? exc_invalid_op+0x18/0x80 [44.111308] ? asm_exc_invalid_op+0x1b/0x20 [44.111311] ? framebuffer_release+0x2c/0x40 [44.111313] ? hvfb_remove+0x86/0xa0 [hyperv_fb] [44.111315] vmbus_remove+0x24/0x40 [hv_vmbus] [44.111323] device_remove+0x40/0x80 [44.111325] device_release_driver_internal+0x20b/0x270 [44.111327] ? bus_find_device+0xb3/0xf0 Fix this by moving the release of framebuffer and associated memory to fb_ops.fb_destroy function, so that framebuffer framework handles it gracefully. While we fix this, also replace manual registrations/unregistration of framebuffer with devm_register_framebuffer.
CVE-2025-21977	In the Linux kernel, the following vulnerability has been resolved: fbdev: hyperv_fb: Fix hang in kdump kernel when on Hyper-V Gen 2 VMs Gen 2 Hyper-V VMs boot via EFI and have a standard EFI framebuffer device. When the kdump kernel runs in such a VM, loading the efifb driver may hang because of accessing the framebuffer at the wrong memory address. The scenario occurs when the hyperv_fb driver in the original kernel moves the framebuffer to a different MMIO address because of conflicts with an already-running efifb or simplefb driver. The hyperv_fb driver then informs Hyper-V of the change, which is allowed by the Hyper-V FB VMBus device protocol. However, when the kexec command loads the kdump kernel into crash memory via the kexec_file_load() system call, the system call doesn't know the framebuffer has moved, and it sets up the kdump screen_info using the original framebuffer address. The transition to the kdump kernel does not go through the Hyper-V host, so Hyper-V does not reset the framebuffer address like it would do on a reboot. When efifb tries to run, it accesses a non-existent framebuffer address, which traps to the Hyper-V host. After many such accesses, the Hyper-V host thinks the guest is being malicious, and throttles the guest to the point that it runs very slowly or appears to have hung. When the kdump kernel is loaded into crash memory via the kexec_load() system call, the problem does not occur. In this case, the kexec command builds the screen_info table itself in user space from data returned by the FBIOGET_FSCREENINFO ioctl against /dev/fb0, which gives it the new framebuffer location. This problem was originally reported in 2020 [1], resulting in commit 3cb73bc3fa2a ("hyperv_fb: Update screen_info after removing old framebuffer"). This commit solved the problem by setting orig_video_isVGA to 0, so the kdump kernel was unaware of the EFI framebuffer. The efifb driver did not try to load, and no hang occurred. But in 2024, commit c25a19afb81c ("fbdev/hyperv_fb: Do not clear global screen_info") effectively reverted 3cb73bc3fa2a. Commit c25a19afb81c has no reference to 3cb73bc3fa2a, so perhaps it was done without knowing the implications that were reported with 3cb73bc3fa2a. In any case, as of commit c25a19afb81c, the original problem came back again. Interestingly, the hyperv_drm driver does not have this problem because it never moves the framebuffer. The difference is that the hyperv_drm driver removes any conflicting framebuffers *before* allocating an MMIO address, while the hyperv_fb drivers removes conflicting framebuffers *after* allocating an MMIO address. With the "after" ordering, hyperv_fb may encounter a conflict and move the framebuffer to a different MMIO address. But the conflict is essentially bogus because it is removed a few lines of code later. Rather than fix the problem with the approach from 2020 in commit 3cb73bc3fa2a, instead slightly reorder the steps in hyperv_fb so conflicting framebuffers are removed before allocating an MMIO address. Then the default framebuffer MMIO address should always be available, and there's never any confusion about which framebuffer address the kdump kernel should use -- it's always the original address provided by the Hyper-V host. This approach is already used by the hyperv_drm driver, and is consistent with the usage guidelines at the head of the module with the function aperture_remove_conflicting_devices(). This approach also solves a related minor problem when kexec_load() is used to load the kdump kernel. With current code, unbinding and rebinding the hyperv_fb driver could result in the framebuffer moving back to the default framebuffer address, because on the rebind there are no conflicts. If such a move is done after the kdump kernel is loaded with the new framebuffer address, at kdump time it could again have the wrong address. This problem and fix are described in terms of the kdump kernel, but it can also occur ---truncated---
CVE-2025-21978	In the Linux kernel, the following vulnerability has been resolved: drm/hyperv: Fix address space leak when Hyper-V DRM device is removed When a Hyper-V DRM device is probed, the driver allocates MMIO space for the vram, and maps it cacheable. If the device removed, or in the error path for device probing, the MMIO space is released but no unmap is done. Consequently the kernel address space for the mapping is leaked. Fix this by adding iounmap() calls in the device removal path, and in the error path during device probing.
CVE-2025-21979	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: cancel wiphy_work before freeing wiphy A wiphy_work can be queued from the moment the wiphy is allocated and initialized (i.e. wiphy_new_nm). When a wiphy_work is queued, the rdev::wiphy_work is getting queued. If wiphy_free is called before the rdev::wiphy_work had a chance to run, the wiphy memory will be freed, and then when it eventually gets to run it'll use invalid memory. Fix this by canceling the work before freeing the wiphy.

CVE ID	Description
CVE-2025-21980	In the Linux kernel, the following vulnerability has been resolved: sched: address a potential NULL pointer dereference in the GRED scheduler. If kzalloc in gred_init returns a NULL pointer, the code follows the error handling path, invoking gred_destroy. This, in turn, calls gred_offload, where memset could receive a NULL pointer as input, potentially leading to a kernel crash. When table->opt is NULL in gred_init(), gred_change_table_def() is not called yet, so it is not necessary to call ->ndo_setup_tc() in gred_offload().
CVE-2025-21981	In the Linux kernel, the following vulnerability has been resolved: ice: fix memory leak in aRFS after reset Fix aRFS (accelerated Receive Flow Steering) structures memory leak by adding a checker to verify if aRFS memory is already allocated while configuring VSI. aRFS objects are allocated in two cases: - as part of VSI initialization (at probe), and - as part of reset handling However, VSI reconfiguration executed during reset involves memory allocation one more time, without prior releasing already allocated resources. This led to the memory leak with the following signature: [root@os-delivery ~]# cat /sys/kernel/debug/kmemleak unreferenced object 0xff3c1ca7252e6000 (size 8192): comm "kworker/0:0", pid 8, jiffies 4296833052 hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace (crc 0): [<ffffffffff91ec485>] __kmalloc_cache_noprof+0x275/0x340 [<ffffffffffc0a6e06a>] ice_init_arfs+0x3a/0xe0 [ice] [<ffffffffffc09f1027>] ice_vsi_cfg_def+0x607/0x850 [ice] [<ffffffffffc09f244b>] ice_vsi_setup+0x5b/0x130 [ice] [<ffffffffffc09c2131>] ice_init+0x1c1/0x460 [ice] [<ffffffffffc09c64af>] ice_probe+0x2af/0x520 [ice] [<ffffffffff994fbc3>] local_pci_probe+0x43/0xa0 [<ffffffffff98f07103>] work_for_cpu_fn+0x13/0x20 [<ffffffffff98f0b6d9>] process_one_work+0x179/0x390 [<ffffffffff98f0c1e9>] worker_thread+0x239/0x340 [<ffffffffff98f14abc>] kthread+0xcc/0x100 [<ffffffffff98e45a6d>] ret_from_fork+0x2d/0x50 [<ffffffffff98e083ba>] ret_from_fork_asm+0x1a/0x30 ...
CVE-2025-21982	In the Linux kernel, the following vulnerability has been resolved: pinctrl: nuvoton: npcm8xx: Add NULL check in npcm8xx_gpio_fw devm_kasprintf() calls can return null pointers on failure. But the return values were not checked in npcm8xx_gpio_fw(). Add NULL check in npcm8xx_gpio_fw(), to handle kernel NULL pointer dereference error.

CVE ID	Description
CVE-2025-21984	In the Linux kernel, the following vulnerability has been resolved: mm: fix kernel BUG when userfaultfd_move encounters swapcache userfaultfd_move() checks whether the PTE entry is present or a swap entry. - If the PTE entry is present, move_present_pte() handles folio migration by setting: src_folio->index = linear_page_index(dst_vma, dst_addr); - If the PTE entry is a swap entry, move_swap_pte() simply copies the PTE to the new dst_addr. This approach is incorrect because, even if the PTE is a swap entry, it can still reference a folio that remains in the swap cache. This creates a race window between steps 2 and 4. 1. add_to_swap: The folio is added to the swapcache. 2. try_to_unmap: PTEs are converted to swap entries. 3. pageout: The folio is written back. 4. Swapcache is cleared. If userfaultfd_move() occurs in the window between steps 2 and 4, after the swap PTE has been moved to the destination, accessing the destination triggers do_swap_page(), which may locate the folio in the swapcache. However, since the folio's index has not been updated to match the destination VMA, do_swap_page() will detect a mismatch. This can result in two critical issues depending on the system configuration. If KSM is disabled, both small and large folios can trigger a BUG during the add_rmap operation due to: page_pgoff(folio, page) != linear_page_index(vma, address) [13.336953] page: refcount:6 mapcount:1 mapping:00000000f43db19c index:0xfffff150 pfn:0x4667c [13.337520] head: order:2 mapcount:1 entire_mapcount:0 nr_pages_mapped:1 pincount:0 [13.337716] memcg:ffff00000405f000 [13.337849] anon flags: 0x3fffc0000020459(locked uptodate dirty owner_priv_1 head swapbacked node=0 zone=0 lastcpupid=0xffff) [13.338630] raw: 03fffc0000020459 ffff80008507b538 ffff80008507b538 ffff000006260361 [13.338831] raw: 0000000ffffaf150 0000000000004000 0000000600000000 ffff00000405f000 [13.339031] head: 03fffc0000020459 ffff80008507b538 ffff80008507b538 ffff000006260361 [13.339204] head: 0000000ffffaf150 0000000000004000 0000000600000000 ffff00000405f000 [13.339375] head: 03fffc0000000202 fffffdfc0199f01 ffffffff00000000 0000000000000001 [13.339546] head: 0000000000000004 0000000000000000 00000000fffffff 0000000000000000 [13.339736] page dumped because: VM_BUG_ON_PAGE(page_pgoff(folio, page) != linear_page_index(vma, address)) [13.340190] -----[cut here]----- [13.340316] kernel BUG at mm/rmap.c:1380! [13.340683] Internal error: Oops - BUG: 0000000f2000800 [#1] PREEMPT SMP [13.340969] Modules linked in: [13.341257] CPU: 1 UID: 0 PID: 107 Comm: a.out Not tainted 6.14.0-rc3-gcf42737e247a-dirty #299 [13.341470] Hardware name: linux,dummy-virt (DT) [13.341671] pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) [13.341815] pc : __page_check_anon_rmap+0xa0/0xb0 [13.341920] lr : __page_check_anon_rmap+0xa0/0xb0 [13.342018] sp : ffff80008752bb20 [13.342093] x29: ffff80008752bb20 x28: fffffdfc0199f00 x27: 0000000000000001 [13.342404] x26: 0000000000000000 x25: 0000000000000001 x24: 0000000000000001 [13.342575] x23: 0000ffffaf0d0000 x22: 0000ffffaf0d0000 x21: fffffdfc0199f00 [13.342731] x20: fffffdfc0199f00 x19: ffff000006210700 x18: 00000000fffffff [13.342881] x17: 6c203d2120296567 x16: 6170202c6f696c6f x15: 662866666f67705f [13.343033] x14: 6567617028454741 x13: 2929737365726464 x12: ffff800083728ab0 [13.343183] x11: ffff800082996bf8 x10: 0000000000000fd7 x9: ffff80008011bc40 [13.343351] x8 : 0000000000017fe8 x7 : 00000000fffff000 x6 : ffff8000829eebf8 [13.343498] x5 : c0000000fffff000 x4 : 0000000000000000 x3 : 0000000000000000 [13.343645] x2 : 0000000000000000 x1 : ffff0000062db980 x0 : 000000000000005f [13.343876] Call trace: [13.344045] __page_check_anon_rmap+0xa0/0xb0 (P) [13.344234] folio_add_anon_rmap_ptes+0x22c/0x320 [13.344333] do_swap_page+0x1060/0x1400 [13.344417] __handle ---truncated---
CVE-2025-21985	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix out-of-bound accesses [WHAT & HOW] hpo_stream_to_link_encoder_mapping has size MAX_HPO_DP2_ENCODERS(=4), but location can have size up to 6. As a result, it is necessary to check location against MAX_HPO_DP2_ENCODERS. Similiarly, disp_cfg_stream_location can be used as an array index which should be 0..5, so the ASSERT's conditions should be less without equal.

CVE ID	Description
CVE-2025-21986	In the Linux kernel, the following vulnerability has been resolved: net: switchdev: Convert blocking notification chain to a raw one A blocking notification chain uses a read-write semaphore to protect the integrity of the chain. The semaphore is acquired for writing when adding / removing notifiers to / from the chain and acquired for reading when traversing the chain and informing notifiers about an event. In case of the blocking switchdev notification chain, recursive notifications are possible which leads to the semaphore being acquired twice for reading and to lockdep warnings being generated [1]. Specifically, this can happen when the bridge driver processes a SWITCHDEV_BRPORT_UNOFFLOADED event which causes it to emit notifications about deferred events when calling switchdev_deferred_process(). Fix this by converting the notification chain to a raw notification chain in a similar fashion to the netdev notification chain. Protect the chain using the RTNL mutex by acquiring it when modifying the chain. Events are always informed under the RTNL mutex, but add an assertion in call_switchdev_blocking_notifiers() to make sure this is not violated in the future. Maintain the "blocking" prefix as events are always emitted from process context and listeners are allowed to block. [1]: WARNING: possible recursive locking detected 6.14.0-rc4-custom-g079270089484 #1 Not tainted ----- ip/52731 is trying to acquire lock: ffffffff850918d8 ((switchdev_blocking_notif_chain).rwsem){++++}-{4:4}, at: blocking_notifier_call_chain+0x58/0xa0 but task is already holding lock: ffffffff850918d8 ((switchdev_blocking_notif_chain).rwsem){++++}-{4:4}, at: blocking_notifier_call_chain+0x58/0xa0 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock((switchdev_blocking_notif_chain).rwsem); lock((switchdev_blocking_notif_chain).rwsem); *** DEADLOCK *** May be due to missing lock nesting notation 3 locks held by ip/52731: #0: ffffffff84f795b0 (rtln_mutex){+.-.}-{4:4}, at: rtln_newlink+0x727/0x1dc0 #1: ffffffff8731f628 (&net->rtln_mutex){+.-.}-{4:4}, at: rtln_newlink+0x790/0x1dc0 #2: ffffffff850918d8 ((switchdev_blocking_notif_chain).rwsem){++++}-{4:4}, at: blocking_notifier_call_chain+0x58/0xa0 stack backtrace: ... ? __pfx_down_read+0x10/0x10 ? __pfx_mark_lock+0x10/0x10 ? __pfx_switchdev_port_attr_set_deferred+0x10/0x10 blocking_notifier_call_chain+0x58/0xa0 switchdev_port_attr_notify.constprop.0+0xb3/0x1b0 ? __pfx_switchdev_port_attr_notify.constprop.0+0x10/0x10 ? mark_held_locks+0x94/0xe0 ? switchdev_deferred_process+0x11a/0x340 switchdev_port_attr_set_deferred+0x27/0xd0 switchdev_deferred_process+0x164/0x340 br_switchdev_port_unoffload+0xc8/0x100 [bridge] br_switchdev_blocking_event+0x29f/0x580 [bridge] notifier_call_chain+0xa2/0x440 blocking_notifier_call_chain+0x6e/0xa0 switchdev_bridge_port_unoffload+0xde/0x1a0 ...
CVE-2025-21991	In the Linux kernel, the following vulnerability has been resolved: x86/microcode/AMD: Fix out-of-bounds on systems with CPU-less NUMA nodes Currently, load_microcode_amd() iterates over all NUMA nodes, retrieves their CPU masks and unconditionally accesses per-CPU data for the first CPU of each mask. According to Documentation/admin-guide/mm/numaperf.rst: "Some memory may share the same node as a CPU, and others are provided as memory only nodes." Therefore, some node CPU masks may be empty and wouldn't have a "first CPU". On a machine with far memory (and therefore CPU-less NUMA nodes): - cpumask_of_node(nid) is 0 - cpumask_first(0) is CONFIG_NR_CPUS - cpu_data(CONFIG_NR_CPUS) accesses the cpu_info per-CPU array at an index that is 1 out of bounds This does not have any security implications since flashing microcode is a privileged operation but I believe this has reliability implications by potentially corrupting memory while flashing a microcode update. When booting with CONFIG_UBSAN_BOUNDS=y on an AMD machine that flashes a microcode update. I get the following splat: UBSAN: array-index-out-of-bounds in arch/x86/kernel/cpu/microcode/amd.c:X:Y index 512 is out of range for type 'unsigned long[512]' [...] Call Trace: dump_stack __ubsan_handle_out_of_bounds load_microcode_amd request_microcode_amd reload_store kernfs_fop_write_iter vfs_write ksys_write do_syscall_64 entry_SYSCALL_64_after_hwframe Change the loop to go over only NUMA nodes which have CPUs before determining whether the first CPU on the respective node needs microcode update. [bp: Message commit message, fix typo.]
CVE-2025-21992	In the Linux kernel, the following vulnerability has been resolved: HID: ignore non-functional sensor in HP 5MP Camera The HP 5MP Camera (USB ID 0408:5473) reports a HID sensor interface that is not actually implemented. Attempting to access this non-functional sensor via iio_info causes system hangs as runtime PM tries to wake up an unresponsive sensor. [453] hid-sensor-hub 0003:0408:5473.0003: Report latency attributes: ffffffff:ffffffff [453] hid-sensor-hub 0003:0408:5473.0003: common attributes: 5:1, 2:1, 3:1 ffffffff:ffffffff Add this device to the HID ignore list since the sensor interface is non-functional by design and should not be exposed to userspace.
CVE-2025-21993	In the Linux kernel, the following vulnerability has been resolved: iscsi_ibft: Fix UBSAN shift-out-of-bounds warning in ibft_attr_show_nic() When performing an iSCSI boot using IPv6, iscsistart still reads the /sys/firmware/ibft/ethernetX/subnet-mask entry. Since the IPv6 prefix length is 64, this causes the shift exponent to become negative, triggering a UBSAN warning. As the concept of a subnet mask does not apply to IPv6, the value is set to ~0 to suppress the warning message.

CVE ID	Description
CVE-2025-21994	In the Linux kernel, the following vulnerability has been resolved: ksmdbd: fix incorrect validation for num_aces field of smb_acl parse_dcal() validate num_aces to allocate posix_ace_state_array. if (num_aces > ULONG_MAX / sizeof(struct smb_ace *)) It is an incorrect validation that we can create an array of size ULONG_MAX. smb_acl has ->size field to calculate actual number of aces in request buffer size. Use this to check invalid num_aces.
CVE-2025-21995	In the Linux kernel, the following vulnerability has been resolved: drm/sched: Fix fence reference count leak The last_scheduled fence leaks when an entity is being killed and adding the cleanup callback fails. Decrement the reference count of prev when dma_fence_add_callback() fails, ensuring proper balance. [phasta: add git tag info for stable kernel]
CVE-2025-21996	In the Linux kernel, the following vulnerability has been resolved: drm/radeon: fix uninitialized size issue in radeon_vce_cs_parse() On the off chance that command stream passed from userspace via ioctl() call to radeon_vce_cs_parse() is weirdly crafted and first command to execute is to encode (case 0x03000001), the function in question will attempt to call radeon_vce_cs_reloc() with size argument that has not been properly initialized. Specifically, 'size' will point to 'tmp' variable before the latter had a chance to be assigned any value. Play it safe and init 'tmp' with 0, thus ensuring that radeon_vce_cs_reloc() will catch an early error in cases like these. Found by Linux Verification Center (linuxtesting.org) with static analysis tool SVACE. (cherry picked from commit 2d52de55f9ee7aaee0e09ac443f77855989c6b68)
CVE-2025-21997	In the Linux kernel, the following vulnerability has been resolved: xsk: fix an integer overflow in xp_create_and_assign_umem() Since the i and pool->chunk_size variables are of type 'u32', their product can wrap around and then be cast to 'u64'. This can lead to two different XDP buffers pointing to the same memory area. Found by InfoTeCS on behalf of Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-21999	In the Linux kernel, the following vulnerability has been resolved: proc: fix UAF in proc_get_inode() Fix race between rmmmod and /proc/XXX's inode instantiation. The bug is that pde->proc_ops don't belong to /proc, it belongs to a module, therefore dereferencing it after /proc entry has been registered is a bug unless use_pde/unuse_pde() pair has been used. use_pde/unuse_pde can be avoided (2 atomic ops!) because pde->proc_ops never changes so information necessary for inode instantiation can be saved _before_ proc_register() in PDE itself and used later, avoiding pde->proc_ops->... dereference. rmmmod lookup sys_delete_module proc_lookup_de pde_get(de); proc_get_inode(dir->i_sb, de); mod->exit() proc_remove remove_proc_subtree proc_entry_rundown(de); free_module(mod); if (S_ISREG(inode->i_mode)) if (de->proc_ops->proc_read_iter) --> As module is already freed, will trigger UAF BUG: unable to handle page fault for address: ffffffbfff80a702b PGD 817fc4067 P4D 817fc4067 PUD 817fc0067 PMD 102ef4067 PTE 0 Oops: Oops: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 26 UID: 0 PID: 2667 Comm: ls Tainted: G Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) RIP: 0010:proc_get_inode+0x302/0x6e0 RSP: 0018:ffff88811c837998 EFLAGS: 00010a06 RAX: dffffc0000000000 RBX: ffffffff0538140 RCX: 0000000000000007 RDX: 1fffffff80a702b RSI: 0000000000000001 RDI: ffffffff0538158 RBP: ffff8881299a6000 R08: 0000000067bbe1e5 R09: 1ffff11023906f20 R10: ffffffff560ca07 R11: ffffffff2b43a58 R12: ffff888105bb78f0 R13: ffff888100518048 R14: ffff8881299a6004 R15: 0000000000000001 FS: 00007f95b9686840(0000) GS:ffff8883af100000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffbfff80a702b CR3: 0000000117dd2000 CR4: 0000000000000060 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> proc_lookup_de+0x11f/0x2e0 __lookup_slow+0x188/0x350 walk_component+0x2ab/0x4f0 path_lookupat+0x120/0x660 filename_lookup+0x1ce/0x560 vfs_statx+0xac/0x150 __do_sys_newstat+0x96/0x110 do_syscall_64+0x5f/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e [adobriyan@gmail.com: don't do 2 atomic ops on the common path]
CVE-2025-22001	In the Linux kernel, the following vulnerability has been resolved: accel/qaic: Fix integer overflow in qaic_validate_req() These are u64 variables that come from the user via qaic_attach_slice_bo_ioctl(). Use check_add_overflow() to ensure that the math doesn't have an integer wrapping bug.

CVE ID	Description
CVE-2025-22002	In the Linux kernel, the following vulnerability has been resolved: netfs: Call `invalidate_cache` only if implemented Many filesystems such as NFS and Ceph do not implement the `invalidate_cache` method. On those filesystems, if writing to the cache (`NETFS_WRITE_TO_CACHE`) fails for some reason, the kernel crashes like this: BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor instruction fetch in kernel mode #PF: error_code(0x0010) - not-present page PGD 0 P4D 0 Oops: Oops: 0010 [#1] SMP PTI CPU: 9 UID: 0 PID: 3380 Comm: kworker/u193:11 Not tainted 6.13.3-cm4all1-hp #437 Hardware name: HP ProLiant DL380 Gen9/ProLiant DL380 Gen9, BIOS P89 10/17/2018 Workqueue: events_unbound netfs_write_collection_worker RIP: 0010:0x0 Code: Unable to access opcode bytes at 0xfffffffffffd6. RSP: 0018:ffff9b86e2ca7dc0 EFLAGS: 00010202 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 7fffffff RDX: 0000000000000001 RSI: ffff89259d576a18 RDI: ffff89259d576900 RBP: ffff89259d5769b0 R08: ffff9b86e2ca7d28 R09: 0000000000000002 R10: ffff89258ceaca80 R11: 0000000000000001 R12: 0000000000000020 R13: ffff893d158b9338 R14: ffff89259d576900 R15: ffff89259d5769b0 FS: 0000000000000000(0000) GS:ffff893c9fa40000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffff RCR3: 000000054442e003 CR4: 0000000001706f0 Call Trace: <TASK> ? __die+0x1f/0x60 ? page_fault_oops+0x15c/0x460 ? try_to_wake_up+0x2d2/0x530 ? exc_page_fault+0x5e/0x100 ? asm_exc_page_fault+0x22/0x30 netfs_write_collection_worker +0xe9f/0x12b0 ? xs_poll_check_readable+0x3f/0x80 ? xs_stream_data_receive_workfn+0x8d/0x110 process_one_work+0x134/0x2d0 worker_thread+0x299/0x3a0 ? __pfx_worker_thread+0x10/0x10 kthread+0xba/0xe0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x30/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Modules linked in: CR2: 0000000000000000 This patch adds the missing `NULL` check.
CVE-2025-22003	In the Linux kernel, the following vulnerability has been resolved: can: ucan: fix out of bound read in strscpy() source Commit 7daf8966aae ("can: ucan: use strscpy() to instead of strncpy()") unintentionally introduced a one byte out of bound read on strscpy()'s source argument (which is kind of ironic knowing that strscpy() is meant to be a more secure alternative :)). Let's consider below buffers: dest[len + 1]; / * will be NUL terminated */ src[len]; /* may not be NUL terminated */ When doing: strscpy(dest, src, len); dest[len] = '\0'; strscpy() will read up to len bytes from src. On the other hand: strscpy(dest, src, len + 1); will read up to len + 1 bytes from src, that is to say, an out of bound read of one byte will occur on src if it is not NUL terminated. Note that the src[len] byte is never copied, but strscpy() still needs to read it to check whether a truncation occurred or not. This exact pattern happened in ucan. The root cause is that the source is not NUL terminated. Instead of doing a copy in a local buffer, directly NUL terminate it as soon as usb_control_msg() returns. With this, the local firmware_str[] variable can be removed. On top of this do a couple refactors: - ucan_ctl_payload->raw is only used for the firmware string, so rename it to ucan_ctl_payload->fw_str and change its type from u8 to char. - ucan_device_request_in() is only used to retrieve the firmware string, so rename it to ucan_get_fw_str() and refactor it to make it directly handle all the string termination logic.
CVE-2025-22004	In the Linux kernel, the following vulnerability has been resolved: net: atm: fix use after free in lec_send() The ->send() operation frees skb so save the length before calling ->send() to avoid a use after free.
CVE-2025-22005	In the Linux kernel, the following vulnerability has been resolved: ipv6: Fix memleak of nhc_pcpu_rth_output in fib_check_nh_v6_gw(). fib_check_nh_v6_gw() expects that fib6_nh_init() cleans up everything when it fails. Commit 7dd73168e273 ("ipv6: Always allocate pcpu memory in a fib6_nh") moved fib_nh_common_init() before alloc_percpu_gfp() within fib6_nh_init() but forgot to add cleanup for fib6_nh->nh_common.nhc_pcpu_rth_output in case it fails to allocate fib6_nh->rt6i_pcpu, resulting in memleak. Let's call fib_nh_common_release() and clear nhc_pcpu_rth_output in the error path. Note that we can remove the fib6_nh_release() call in nh_create_ipv6() later in net-next.git.
CVE-2025-22007	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix error code in chan_alloc_skb_cb() The chan_alloc_skb_cb() function is supposed to return error pointers on error. Returning NULL will lead to a NULL dereference.
CVE-2025-22008	In the Linux kernel, the following vulnerability has been resolved: regulator: check that dummy regulator has been probed before using it Due to asynchronous driver probing there is a chance that the dummy regulator hasn't already been probed when first accessing it.

CVE ID	Description
CVE-2025-22009	In the Linux kernel, the following vulnerability has been resolved: regulator: dummy: force synchronous probing Sometimes I get a NULL pointer dereference at boot time in kobject_get() with the following call stack: anatop_regulator_probe() devm_regulator_register() regulator_register() regulator_resolve_supply() kobject_get() By placing some extra BUG_ON() statements I could verify that this is raised because probing of the 'dummy' regulator driver is not completed ('dummy_regulator_rdev' is still NULL). In the JTAG debugger I can see that dummy_regulator_probe() and anatop_regulator_probe() can be run by different kernel threads (kworker/u4:*). I haven't further investigated whether this can be changed or if there are other possibilities to force synchronization between these two probe routines. On the other hand I don't expect much boot time penalty by probing the 'dummy' regulator synchronously.
CVE-2025-22010	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix soft lockup during bt pages loop Driver runs a for-loop when allocating bt pages and mapping them with buffer pages. When a large buffer (e.g. MR over 100GB) is being allocated, it may require a considerable loop count. This will lead to soft lockup: watchdog: BUG: soft lockup - CPU#27 stuck for 22s! ... Call trace: hem_list_alloc_mid_bt+0x124/0x394 [hns_roce_hw_v2] hns_roce_hem_list_request+0xf8/0x160 [hns_roce_hw_v2] hns_roce_mtr_create+0x2e4/0x360 [hns_roce_hw_v2] alloc_mr_pbl+0xd4/0x17c [hns_roce_hw_v2] hns_roce_reg_user_mr+0xf8/0x190 [hns_roce_hw_v2] ib_uverbs_reg_mr+0x118/0x290 watchdog: BUG: soft lockup - CPU#35 stuck for 23s! ... Call trace: hns_roce_hem_list_find_mtr+0x7c/0xb0 [hns_roce_hw_v2] mtr_map_bufs+0xc4/0x204 [hns_roce_hw_v2] hns_roce_mtr_create+0x31c/0x3c4 [hns_roce_hw_v2] alloc_mr_pbl+0xb0/0x160 [hns_roce_hw_v2] hns_roce_reg_user_mr+0x108/0x1c0 [hns_roce_hw_v2] ib_uverbs_reg_mr+0x120/0x2bc Add a cond_resched() to fix soft lockup during these loops. In order not to affect the allocation performance of normal-size buffer, set the loop count of a 100GB MR as the threshold to call cond_resched().
CVE-2025-22011	In the Linux kernel, the following vulnerability has been resolved: ARM: dts: bcm2711: Fix xHCI power-domain During s2idle tests on the Raspberry CM4 the VPU firmware always crashes on xHCI power-domain resume: root@raspberrypi:/sys/power# echo freeze > state [70.724347] xhci_suspend finished [70.727730] xhci_plat_suspend finished [70.755624] bcm2835-power bcm2835-power: Power grafx off [70.761127] USB: Set power to 0 [74.653040] USB: Failed to set power to 1 (-110) This seems to be caused because of the mixed usage of raspberrypi-power and bcm2835-power at the same time. So avoid the usage of the VPU firmware power-domain driver, which prevents the VPU crash.
CVE-2025-22013	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Unconditionally save +flush host FPSIMD/SVE/SME state There are several problems with the way hyp code lazily saves the host's FPSIMD/SVE state, including: * Host SVE being discarded unexpectedly due to inconsistent configuration of TIF_SVE and CPACR_ELx.ZEN. This has been seen to result in QEMU crashes where SVE is used by memmove(), as reported by Eric Auger: https://issues.redhat.com/browse/RHEL-68997 * Host SVE state is discarded *after* modification by ptrace, which was an unintentional ptrace ABI change introduced with lazy discarding of SVE state. * The host FPMR value can be discarded when running a non-protected VM, where FPMR support is not exposed to a VM, and that VM uses FPSIMD/SVE. In these cases the hyp code does not save the host's FPMR before unbinding the host's FPSIMD/SVE/SME state, leaving a stale value in memory. Avoid these by eagerly saving and "flushing" the host's FPSIMD/SVE/SME state when loading a vCPU such that KVM does not need to save any of the host's FPSIMD/SVE/SME state. For clarity, fpsimd_kvm_prepare() is removed and the necessary call to fpsimd_save_and_flush_cpu_state() is placed in kvm_arch_vcpu_load_fp(). As 'fpsimd_state' and 'fpmr_ptr' should not be used, they are set to NULL; all uses of these will be removed in subsequent patches. Historical problems go back at least as far as v5.17, e.g. erroneous assumptions about TIF_SVE being clear in commit: 8383741ab2e773a9 ("KVM: arm64: Get rid of host SVE tracking/saving") ... and so this eager save+flush probably needs to be backported to ALL stable trees.
CVE-2025-22014	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: pdr: Fix the potential deadlock When some client process A call pdr_add_lookup() to add the look up for the service and does schedule locator work, later a process B got a new server packet indicating locator is up and call pdr_locator_new_server() which eventually sets pdr->locator_init_complete to true which process A sees and takes list lock and queries domain list but it will timeout due to deadlock as the response will queued to the same qmi->wq and it is ordered workqueue and process B is not able to complete new server request work due to deadlock on list lock. Fix it by removing the unnecessary list iteration as the list iteration is already being done inside locator work, so avoid it here and just call schedule_work() here. Process A Process B process_scheduled_works() pdr_add_lookup() qmi_data_ready_work() process_scheduled_works() pdr_locator_new_server() pdr->locator_init_complete=true; pdr_locator_work() mutex_lock(&pdr->list_lock); pdr_locate_service() mutex_lock(&pdr->list_lock); pdr_get_domain_list() pr_err("PDR: %s get domain list txn wait failed: %d\n", req->service_name, ret); Timeout error log due to deadlock: " PDR: tms/servreg get domain list txn wait failed: -110 PDR: service lookup for msm/adsp/sensor_pd:tms/servreg failed: -110 " Thanks to Bjorn and Johan for letting me know that this commit also fixes an audio regression when using the in-kernel pd-mapper as that makes it easier to hit this race. [1]

CVE ID	Description
CVE-2025-22015	In the Linux kernel, the following vulnerability has been resolved: mm/migrate: fix shmem xarray update during migration A shmem folio can be either in page cache or in swap cache, but not at the same time. Namely, once it is in swap cache, folio->mapping should be NULL, and the folio is no longer in a shmem mapping. In __folio_migrate_mapping(), to determine the number of xarray entries to update, folio_test_swapbacked() is used, but that conflates shmem in page cache case and shmem in swap cache case. It leads to xarray multi-index entry corruption, since it turns a sibling entry to a normal entry during xas_store() (see [1] for a userspace reproduction). Fix it by only using folio_test_swapcache() to determine whether xarray is storing swap cache entries or not to choose the right number of xarray entries to update. [1] https://lore.kernel.org/linux-mm/Z8idPCkaJW1IChjT@casper.infradead.org/ Note: In __split_huge_page(), folio_test_anon() && folio_test_swapcache() is used to get swap_cache address space, but that ignores the shmem folio in swap cache case. It could lead to NULL pointer dereferencing when a in-swap-cache shmem folio is split at __xa_store(), since !folio_test_anon() is true and folio->mapping is NULL. But fortunately, its caller split_huge_page_to_list_to_order() bails out early with EBUSY when folio->mapping is NULL. So no need to take care of it here.
CVE-2025-22016	In the Linux kernel, the following vulnerability has been resolved: dpll: fix xa_alloc_cyclic() error handling In case of returning 1 from xa_alloc_cyclic() (wrapping) ERR_PTR(1) will be returned, which will cause IS_ERR() to be false. Which can lead to dereference not allocated pointer (pin). Fix it by checking if err is lower than zero. This wasn't found in real usecase, only noticed. Credit to Pierre.
CVE-2025-22017	In the Linux kernel, the following vulnerability has been resolved: devlink: fix xa_alloc_cyclic() error handling In case of returning 1 from xa_alloc_cyclic() (wrapping) ERR_PTR(1) will be returned, which will cause IS_ERR() to be false. Which can lead to dereference not allocated pointer (rel). Fix it by checking if err is lower than zero. This wasn't found in real usecase, only noticed. Credit to Pierre.
CVE-2025-22018	In the Linux kernel, the following vulnerability has been resolved: atm: Fix NULL pointer dereference When MPOA_cache_impos_rcvd() receives the msg, it can trigger Null Pointer Dereference Vulnerability if both entry and holding_time are NULL. Because there is only for the situation where entry is NULL and holding_time exists, it can be passed when both entry and holding_time are NULL. If these are NULL, the entry will be passed to eg_cache_put() as parameter and it is referenced by entry->use code in it. kasan log: [3.316691] Oops: general protection fault, probably for non-canonical address 0xdffffc0000000006:1 [3.317568] KASAN: null-ptr-deref in range [0x0000000000000030-0x0000000000000037] [3.318188] CPU: 3 UID: 0 PID: 79 Comm: ex Not tainted 6.14.0-rc2 #102 [3.318601] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 [3.319298] RIP: 0010:eg_cache_remove_entry+0xa5/0x470 [3.319677] Code: c1 f7 6e fd 48 c7 c7 00 7e 38 b2 e8 95 64 54 fd 48 c7 c7 04 7e 38 b2 48 89 ee e80 [3.321220] RSP: 0018:ffff88800583f8a8 EFLAGS: 00010006 [3.321596] RAX: 0000000000000006 RBX: ffff888005989000 RCX: ffffffff8aacc2d8e [3.322112] RDX: 0000000000000000 RSI: 0000000000000004 RDI: 0000000000000030 [3.322643] RBP: 0000000000000000 R08: 0000000000000000 R09: fffffbfff6558b88 [3.323181] R10: 0000000000000003 R11: 203a207972746e65 R12: 1ffff11000b07f15 [3.323707] R13: dffffc0000000000 R14: ffff888005989000 R15: ffff888005989068 [3.324185] FS: 000000001b6313c0(0000) GS:ffff88806d380000(0000) knlGS:0000000000000000 [3.325042] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [3.325545] CR2: 0000000004b4b40 CR3: 000000000248e000 CR4: 000000000000006f0 [3.326430] Call Trace: [3.326725] <TASK> [3.326927] ? die_addr+0x3c/0xa0 [3.327330] ? exc_general_protection+0x161/0x2a0 [3.327662] ? asm_exc_general_protection+0x26/0x30 [3.328214] ? vprintk_emit+0x15e/0x420 [3.328543] ? eg_cache_remove_entry+0xa5/0x470 [3.328910] ? eg_cache_remove_entry+0x9a/0x470 [3.329294] ? __pfx_eg_cache_remove_entry+0x10/0x10 [3.329664] ? console_unlock+0x107/0x1d0 [3.329946] ? __pfx_console_unlock+0x10/0x10 [3.330283] ? do_syscall_64+0xa6/0x1a0 [3.330584] ? entry_SYSCALL_64_after_hwframe+0x47/0x7f [3.331090] ? __pfx_prb_read_valid+0x10/0x10 [3.331395] ? down_trylock+0x52/0x80 [3.331703] ? vprintk_emit+0x15e/0x420 [3.331986] ? __pfx_vprintk_emit+0x10/0x10 [3.332279] ? down_trylock+0x52/0x80 [3.332527] ? _printk+0xbf/0x100 [3.332762] ? __pfx__printk+0x10/0x10 [3.333007] ? _raw_write_lock_irq+0x81/0xe0 [3.333284] ? __pfx__raw_write_lock_irq+0x10/0x10 [3.333614] msg_from_mpoad+0x1185/0x2750 [3.333893] ? __build_skb_around+0x27b/0x3a0 [3.334183] ? __pfx_msg_from_mpoad+0x10/0x10 [3.334501] ? __alloc_skb+0x1c0/0x310 [3.334809] ? __pfx__alloc_skb+0x10/0x10 [3.335283] ? _raw_spin_lock+0xe0/0xe0 [3.335632] ? finish_wait+0x8d/0x1e0 [3.335975] vcc_sendmsg+0x684/0xba0 [3.336250] ? __pfx_vcc_sendmsg+0x10/0x10 [3.336587] ? __pfx_autoremove_wake_function+0x10/0x10 [3.337056] ? fdget+0x176/0x3e0 [3.337348] __sys_sendto+0x4a2/0x510 [3.337663] ? __pfx__sys_sendto+0x10/0x10 [3.337969] ? ioctl_has_perm.constprop.0.isra.0+0x284/0x400 [3.338364] ? sock_ioctl+0x1bb/0x5a0 [3.338653] ? __rseq_handle_notify_resume+0x825/0xd20 [3.339017] ? __pfx_sock_ioctl+0x10/0x10 [3.339316] ? __pfx__rseq_handle_notify_resume+0x10/0x10 [3.339727] ? selinux_file_ioctl+0xa4/0x260 [3.340166] __x64_sys_sendto+0xe0/0x1c0 [3.340526] ? syscall_exit_to_user_mode+0x123/0x140 [3.340898] do_syscall_64+0xa6/0x1a0 [3.341170] entry_SYSCALL_64_after_hwframe+0x77/0x7f [3.341533] RIP: 0033:0x44a380 [3.341757] Code: 0f 1f 84 00 00 00 00 66 90 f3 0f 1e fa 41 89 ca 64 8b 04 25 18 00 00 00 85 c0 [---truncated---

CVE ID	Description
CVE-2025-22019	In the Linux kernel, the following vulnerability has been resolved: bcachefs: bch2_ioctl_subvolume_destroy() fixes bch2_evict_subvolume_inodes() was getting stuck - due to incorrectly pruning the dcache. Also, fix missing permissions checks.
CVE-2025-22020	In the Linux kernel, the following vulnerability has been resolved: memstick: rtsx_usb_ms: Fix slab-use-after-free in rtsx_usb_ms_drv_remove This fixes the following crash: <pre>===== BUG: KASAN: slab-use-after-free in rtsx_usb_ms_poll_card+0x159/0x200 [rttsx_usb_ms] Read of size 8 at addr ffff888136335380 by task kworker/6:0/140241 CPU: 6 UID: 0 PID: 140241 Comm: kworker/6:0 Kdump: loaded Tainted: G E 6.14.0-rc6+ #1 Tainted: [E]=UNSIGNED_MODULE Hardware name: LENOVO 30FNA1V7CW/1057, BIOS S0EKT54A 07/01/2024 Workqueue: events rtsx_usb_ms_poll_card [rttsx_usb_ms] Call Trace: <TASK> dump_stack_lvl+0x51/0x70 print_address_description.constprop.0+0x27/0x320 ? rtsx_usb_ms_poll_card+0x159/0x200 [rttsx_usb_ms] print_report+0x3e/0x70 kasan_report+0xab/0xe0 ? rtsx_usb_ms_poll_card+0x159/0x200 [rttsx_usb_ms] rtsx_usb_ms_poll_card+0x159/0x200 [rttsx_usb_ms] ? __pfx_rtsx_usb_ms_poll_card+0x10/0x10 [rttsx_usb_ms] ? __pfx__schedule+0x10/0x10 ? kick_pool+0x3b/0x270 process_one_work+0x357/0x660 worker_thread+0x390/0x4c0 ? __pfx_worker_thread+0x10/0x10 kthread+0x190/0x1d0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2d/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Allocated by task 161446: kasan_save_stack+0x20/0x40 kasan_save_track+0x10/0x30 __kasan_kmalloc+0x7b/0x90 __kmalloc_noprof+0x1a7/0x470 memstick_alloc_host+0x1f/0xe0 [memstick] rtsx_usb_ms_drv_probe+0x47/0x320 [rttsx_usb_ms] platform_probe+0x60/0xe0 call_driver_probe+0x35/0x120 really_probe+0x123/0x410 __driver_probe_device+0xc7/0x1e0 driver_probe_device+0x49/0xf0 __device_attach_driver+0xc6/0x160 bus_for_each_drv+0xe4/0x160 __device_attach+0x13a/0x2b0 bus_probe_device+0xbd/0xd0 device_add+0x4a5/0x760 platform_device_add+0x189/0x370 mfd_add_device+0x587/0x5e0 mfd_add_devices+0xb1/0x130 rtsx_usb_probe+0x28e/0x2e0 [rttsx_usb] usb_probe_interface+0x15c/0x460 call_driver_probe+0x35/0x120 really_probe+0x123/0x410 __driver_probe_device+0xc7/0x1e0 driver_probe_device+0x49/0xf0 __device_attach_driver+0xc6/0x160 bus_for_each_drv+0xe4/0x160 __device_attach+0x13a/0x2b0 rebind_marked_interfaces.isra.0+0xcc/0x110 usb_reset_device+0x352/0x410 usbdev_do_ioctl+0xe5c/0x1860 usbdev_ioctl+0xa/0x20 __x64_sys_ioctl+0xc5/0xf0 do_syscall_64+0x59/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e Freed by task 161506: kasan_save_stack+0x20/0x40 kasan_save_track+0x10/0x30 kasan_save_free_info+0x36/0x60 __kasan_slab_free+0x34/0x50 kfree+0x1fd/0x3b0 device_release+0x56/0xf0 kobject_cleanup+0x73/0x1c0 rtsx_usb_ms_drv_remove+0x13d/0x220 [rttsx_usb_ms] platform_remove+0x2f/0x50 device_release_driver_internal+0x24b/0x2e0 bus_remove_device+0x124/0x1d0 device_del+0x239/0x530 platform_device_del.part.0+0x19/0xe0 platform_device_unregister+0x1c/0x40 mfd_remove_devices_fn+0x167/0x170 device_for_each_child_reverse+0xc9/0x130 mfd_remove_devices+0x6e/0xa0 rtsx_usb_disconnect+0x2e/0xd0 [rttsx_usb] usb_unbind_interface+0xf3/0x3f0 device_release_driver_internal+0x24b/0x2e0 proc_disconnect_claim+0x13d/0x220 usbdev_do_ioctl+0xb5e/0x1860 usbdev_ioctl+0xa/0x20 __x64_sys_ioctl+0xc5/0xf0 do_syscall_64+0x59/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e Last potentially related work creation: kasan_save_stack+0x20/0x40 kasan_record_aux_stack+0x85/0x90 insert_work+0x29/0x100 __queue_work+0x34a/0x540 call_timer_fn+0x2a/0x160 expire_timers+0x5f/0x1f0 __run_timer_base.part.0+0x1b6/0x1e0 run_timer_softirq+0x8b/0xe0 handle_softirqs+0xf9/0x360 __irq_exit_rcu+0x114/0x130 sysvec_apic_timer_interrupt+0x72/0x90 asm_sysvec_apic_timer_interrupt+0x16/0x20 Second to last potentially related work creation: kasan_save_stack+0x20/0x40 kasan_record_aux_stack+0x85/0x90 insert_work+0x29/0x100 __queue_work+0x34a/0x540 call_timer_fn+0x2a/0x160 expire_timers+0x5f/0x1f0 __run_timer_base.part.0+0x1b6/0x1e0 run_timer_softirq+0x8b/0xe0 handle_softirqs+0xf9/0x ---truncated---</pre>
CVE-2025-22021	In the Linux kernel, the following vulnerability has been resolved: netfilter: socket: Lookup orig tuple for IPv6 SNAT nf_sk_lookup_slow_v4 does the conntrack lookup for IPv4 packets to restore the original 5-tuple in case of SNAT, to be able to find the right socket (if any). Then socket_match() can correctly check whether the socket was transparent. However, the IPv6 counterpart (nf_sk_lookup_slow_v6) lacks this conntrack lookup, making xt_socket fail to match on the socket when the packet was SNATed. Add the same logic to nf_sk_lookup_slow_v6. IPv6 SNAT is used in Kubernetes clusters for pod-to-world packets, as pods' addresses are in the fd00::/8 ULA subnet and need to be replaced with the node's external address. Cilium leverages Envoy to enforce L7 policies, and Envoy uses transparent sockets. Cilium inserts an iptables prerouting rule that matches on '-m socket --transparent' and redirects the packets to localhost, but it fails to match SNATed IPv6 packets due to that missing conntrack lookup.

CVE ID	Description
CVE-2025-22022	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Apply the link chain quirk on NEC isoc endpoints Two clearly different specimens of NEC uPD720200 (one with start/stop bug, one without) were seen to cause IOMMU faults after some Missed Service Errors. Faulting address is immediately after a transfer ring segment and patched dynamic debug messages revealed that the MSE was received when waiting for a TD near the end of that segment: [1.041954] xhci_hcd: Miss service interval error for slot 1 ep 2 expected TD DMA ffa08fe0 [1.042120] xhci_hcd: AMD-Vi: Event logged [IO_PAGE_FAULT domain=0x0005 address=0xffa09000 flags=0x0000] [1.042146] xhci_hcd: AMD-Vi: Event logged [IO_PAGE_FAULT domain=0x0005 address=0xffa09040 flags=0x0000] It gets even funnier if the next page is a ring segment accessible to the HC. Below, it reports MSE in segment at ff1e8000, plows through a zero-filled page at ff1e9000 and starts reporting events for TRBs in page at ff1ea000 every microframe, instead of jumping to seg ff1e6000. [7.041671] xhci_hcd: Miss service interval error for slot 1 ep 2 expected TD DMA ff1e8fe0 [7.041999] xhci_hcd: Miss service interval error for slot 1 ep 2 expected TD DMA ff1e8fe0 [7.042011] xhci_hcd: WARN: buffer overrun event for slot 1 ep 2 on endpoint [7.042028] xhci_hcd: All TDs skipped for slot 1 ep 2. Clear skip flag. [7.042134] xhci_hcd: WARN: buffer overrun event for slot 1 ep 2 on endpoint [7.042138] xhci_hcd: ERROR Transfer event TRB DMA ptr not part of current TD ep_index 2 comp_code 31 [7.042144] xhci_hcd: Looking for event-dma 00000000ff1ea040 trb-start 00000000ff1e6820 trb-end 00000000ff1e6820 [7.042259] xhci_hcd: WARN: buffer overrun event for slot 1 ep 2 on endpoint [7.042262] xhci_hcd: ERROR Transfer event TRB DMA ptr not part of current TD ep_index 2 comp_code 31 [7.042266] xhci_hcd: Looking for event-dma 00000000ff1ea050 trb-start 00000000ff1e6820 trb-end 00000000ff1e6820 At some point completion events change from Isoch Buffer Overrun to Short Packet and the HC finally finds cycle bit mismatch in ff1ec000. [7.098130] xhci_hcd: ERROR Transfer event TRB DMA ptr not part of current TD ep_index 2 comp_code 13 [7.098132] xhci_hcd: Looking for event-dma 00000000ff1ecc50 trb-start 00000000ff1e6820 trb-end 00000000ff1e6820 [7.098254] xhci_hcd: ERROR Transfer event TRB DMA ptr not part of current TD ep_index 2 comp_code 13 [7.098256] xhci_hcd: Looking for event-dma 00000000ff1ecc60 trb-start 00000000ff1e6820 trb-end 00000000ff1e6820 [7.098379] xhci_hcd: Overrun event on slot 1 ep 2 It's possible that data from the isochronous device were written to random buffers of pending TDs on other endpoints (either IN or OUT), other devices or even other HCs in the same IOMMU domain. Lastly, an error from a different USB device on another HC. Was it caused by the above? I don't know, but it may have been. The disk was working without any other issues and generated PCIe traffic to starve the NEC of upstream BW and trigger those MSEs. The two HCs shared one x1 slot by means of a commercial "PCIe splitter" board. [7.162604] usb 10-2: reset SuperSpeed USB device number 3 using xhci_hcd [7.178990] sd 9:0:0:0: [sdb] tag#0 UNKNOWN(0x2003) Result: hostbyte=0x07 driverbyte=DRIVER_OK cmd_age=0s [7.179001] sd 9:0:0:0: [sdb] tag#0 CDB: opcode=0x28 28 00 04 02 ae 00 00 02 00 00 [7.179004] I/O error, dev sdb, sector 67284480 op 0x0:(READ) flags 0x80700 phys_seg 5 prio class 0 Fortunately, it appears that this ridiculous bug is avoided by setting the chain bit of Link TRBs on isochronous rings. Other ancient HCs are known which also expect the bit to be set and they ignore Link TRBs if it's not. Reportedly, 0.95 spec guaranteed that the bit is set. The bandwidth-starved NEC HC running a 32KB/uframe UVC endpoint reports tens of MSEs per second and runs into the bug within seconds. Chaining Link TRBs allows the same workload to run for many minutes, many times. No ne ---truncated---
CVE-2025-22025	In the Linux kernel, the following vulnerability has been resolved: nfsd: put dl_stid if fail to queue dl_recall Before calling nfsd4_run_cb to queue dl_recall to the callback_wq, we increment the reference count of dl_stid. We expect that after the corresponding work_struct is processed, the reference count of dl_stid will be decremented through the callback function nfsd4_cb_recall_release. However, if the call to nfsd4_run_cb fails, the incremented reference count of dl_stid will not be decremented correspondingly, leading to the following nfs4_stid leak: unreferenced object 0xffff88812067b578 (size 344): comm "nfsd", pid 2761, jiffies 4295044002 (age 5541.241s) hex dump (first 32 bytes): 01 00 00 00 6b 6b 6b 6b b8 02 c0 e2 81 88 ff ff kkkk..... 00 6b 6b 6b 6b 6b 6b 6b 00 00 00 00 ad 4e ad de .kkkkkkk.....N.. backtrace: kmem_cache_alloc+0x4b9/0x700 nfsd4_process_open1+0x34/0x300 nfsd4_open+0x2d1/0x9d0 nfsd4_proc_compound+0x7a2/0xe30 nfsd_dispatch+0x241/0x3e0 svc_process_common+0x5d3/0xcc0 svc_process+0x2a3/0x320 nfsd +0x180/0x2e0 kthread+0x199/0x1d0 ret_from_fork+0x30/0x50 ret_from_fork_asm+0x1b/0x30 unreferenced object 0xffff8881499f4d28 (size 368): comm "nfsd", pid 2761, jiffies 4295044005 (age 5541.239s) hex dump (first 32 bytes): 01 00 00 00 00 00 00 00 30 4d 9f 49 81 88 ff ff 0M.L.... 30 4d 9f 49 81 88 ff ff 20 00 00 00 01 00 00 00 0M.L.... backtrace: kmem_cache_alloc+0x4b9/0x700 nfs4_alloc_stid+0x29/0x210 alloc_init_deleg+0x92/0x2e0 nfs4_set_delegation+0x284/0xc00 nfs4_open_delegation+0x216/0x3f0 nfsd4_process_open2+0x2b3/0xee0 nfsd4_open+0x770/0x9d0 nfsd4_proc_compound+0x7a2/0xe30 nfsd_dispatch+0x241/0x3e0 svc_process_common+0x5d3/0xcc0 svc_process+0x2a3/0x320 nfsd+0x180/0x2e0 kthread+0x199/0x1d0 ret_from_fork+0x30/0x50 ret_from_fork_asm+0x1b/0x30 Fix it by checking the result of nfsd4_run_cb and call nfs4_put_stid if fail to queue dl_recall.

CVE ID	Description
CVE-2025-22026	In the Linux kernel, the following vulnerability has been resolved: nfsd: don't ignore the return code of svc_proc_register() Currently, nfsd_proc_stat_init() ignores the return value of svc_proc_register(). If the procfile creation fails, then the kernel will WARN when it tries to remove the entry later. Fix nfsd_proc_stat_init() to return the same type of pointer as svc_proc_register(), and fix up nfsd_net_init() to check that and fail the nfsd_net construction if it occurs. svc_proc_register() can fail if the dentry can't be allocated, or if an identical dentry already exists. The second case is pretty unlikely in the nfsd_net construction codepath, so if this happens, return -ENOMEM.
CVE-2025-22027	In the Linux kernel, the following vulnerability has been resolved: media: streamzap: fix race between device disconnection and urb callback Syzkaller has reported a general protection fault at function ir_raw_event_store_with_filter(). This crash is caused by a NULL pointer dereference of dev->raw pointer, even though it is checked for NULL in the same function, which means there is a race condition. It occurs due to the incorrect order of actions in the streamzap_disconnect() function: rc_unregister_device() is called before usb_kill_urb(). The dev->raw pointer is freed and set to NULL in rc_unregister_device(), and only after that usb_kill_urb() waits for in-progress requests to finish. If rc_unregister_device() is called while streamzap_callback() handler is not finished, this can lead to accessing freed resources. Thus rc_unregister_device() should be called after usb_kill_urb(). Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-22028	In the Linux kernel, the following vulnerability has been resolved: media: vimc: skip .s_stream() for stopped entities Syzbot reported [1] a warning prompted by a check in call_s_stream() that checks whether .s_stream() operation is warranted for unstarted or stopped subdevs. Add a simple fix in vimc_streamer_pipeline_terminate() ensuring that entities skip a call to .s_stream() unless they have been previously properly started. [1] Syzbot report: -----[cut here]----- WARNING: CPU: 0 PID: 5933 at drivers/media/v4l2-core/v4l2-subdev.c:460 call_s_stream+0x2df/0x350 drivers/media/v4l2-core/v4l2-subdev.c:460 Modules linked in: CPU: 0 UID: 0 PID: 5933 Comm: syz-executor330 Not tainted 6.13.0-rc2-syzkaller-00362-g2d8308bf5b67 #0 ... Call Trace: <TASK> vimc_streamer_pipeline_terminate+0x218/0x320 drivers/media/test-drivers/vimc/vimc-streamer.c:62 vimc_streamer_pipeline_init drivers/media/test-drivers/vimc/vimc-streamer.c:101 [inline] vimc_streamer_s_stream+0x650/0x9a0 drivers/media/test-drivers/vimc/vimc-streamer.c:203 vimc_capture_start_streaming+0xa1/0x130 drivers/media/test-drivers/vimc/vimc-capture.c:256 vb2_start_streaming+0x15f/0x5a0 drivers/media/common/videobuf2/videobuf2-core.c:1789 vb2_core_streamon+0x2a7/0x450 drivers/media/common/videobuf2/videobuf2-core.c:2348 vb2_streamon drivers/media/common/videobuf2/videobuf2-v4l2.c:875 [inline] vb2_ioctl_streamon+0xf4/0x170 drivers/media/common/videobuf2/videobuf2-v4l2.c:1118 __video_do_ioctl+0xaf0/0xf00 drivers/media/v4l2-core/v4l2-ioctl.c:3122 video_usercopy+0x4d2/0x1620 drivers/media/v4l2-core/v4l2-ioctl.c:3463 v4l2_ioctl+0x1ba/0x250 drivers/media/v4l2-core/v4l2-dev.c:366 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl fs/ioctl.c:892 [inline] __x64_sys_ioctl+0x190/0x200 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f2b85c01b19 ...

CVE ID	Description
CVE-2025-22033	In the Linux kernel, the following vulnerability has been resolved: arm64: Don't call NULL in do_compat_alignment_fixup() do_alignment_t32_to_handler() only fixes up alignment faults for specific instructions; it returns NULL otherwise (e.g. LDREX). When that's the case, signal to the caller that it needs to proceed with the regular alignment fault handling (i.e. SIGBUS). Without this patch, the kernel panics: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x0000000008600006 EC = 0x21: IABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x06: level 2 translation fault user pgtable: 4k pages, 48-bit VAs, pgdp=00000800164aa000 [0000000000000000] pgd=0800081fdbd22003, p4d=0800081fdbd22003, pud=08000815d51c6003, pmd=0000000000000000 Internal error: Oops: 0000000008600006 [#1] SMP Modules linked in: cfg80211 rfkill xt_nat xt_tcpudp xt_conntrack nft_chain_nat xt_MASQUERADE nf_nat nf_conntrack_netlink nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 xfrm_user xfrm_algo xt_addrtype nft_compat br_netfilter veth nvme_fa> libcrc32c crc32c_generic raid0 multipath linear dm_mod dax raid1 md_mod xhci_pci nvme xhci_hcd nvme_core t10_pi usbcore igb crc64_rocksoft crc64 crc_t10dif crct10dif_generic crct10dif_ce crct10dif_common usb_common i2c_algo_bit i2c> CPU: 2 PID: 3932954 Comm: WPEWebProcess Not tainted 6.1.0-31-arm64 #1 Debian 6.1.128-1 Hardware name: GIGABYTE MP32-AR1-00/MP32-AR1-00, BIOS F18v (SCP: 1.08.20211002) 12/01/2021 pstate: 80400009 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : 0x0 lr : do_compat_alignment_fixup+0xd8/0x3dc sp : ffff80000f973dd0 x29: ffff80000f973dd0 x28: ffff081b42526180 x27: 0000000000000000 x26: 0000000000000000 x25: 0000000000000000 x24: 0000000000000000 x23: 0000000000000004 x22: 0000000000000000 x21: 0000000000000001 x20: 00000000e8551f00 x19: ffff80000f973eb0 x18: 0000000000000000 x17: 0000000000000000 x16: 0000000000000000 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000000 x10: 0000000000000000 x9: ffffaebc949bc488 x8 : 0000000000000000 x7 : 0000000000000000 x6 : 0000000000000000 x5 : 0000000000400000 x4 : 0000ffffffffff x3 : 0000000000000000 x2 : ffff80000f973eb0 x1 : 00000000e8551f00 x0 : 0000000000000001 Call trace: 0x0 do_alignment_fault+0x40/0x50 do_mem_abort+0x4c/0xa0 el0_da+0x48/0xf0 el0t_32_sync_handler+0x110/0x140 el0t_32_sync+0x190/0x194 Code: bad PC value --- [end trace 0000000000000000]---
CVE-2025-22035	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix use-after-free in print_graph_function_flags during tracer switching Kairui reported a UAF issue in print_graph_function_flags() during ftrace stress testing [1]. This issue can be reproduced if putting a 'mdelay(10)' after 'mutex_unlock(&trace_types_lock)' in s_start(), and executing the following script: \$ echo function_graph > current_tracer \$ cat trace > /dev/null & \$ sleep 5 # Ensure the 'cat' reaches the 'mdelay(10)' point \$ echo timerlat > current_tracer The root cause lies in the two calls to print_graph_function_flags within print_trace_line during each s_show(): * One through 'iter->trace->print_line()'; * Another through 'event->funcs->trace()', which is hidden in print_trace_fmt() before print_trace_line returns. Tracer switching only updates the former, while the latter continues to use the print_line function of the old tracer, which in the script above is print_graph_function_flags. Moreover, when switching from the 'function_graph' tracer to the 'timerlat' tracer, s_start only calls graph_trace_close of the 'function_graph' tracer to free 'iter->private', but does not set it to NULL. This provides an opportunity for 'event->funcs->trace()' to use an invalid 'iter->private'. To fix this issue, set 'iter->private' to NULL immediately after freeing it in graph_trace_close(), ensuring that an invalid pointer is not passed to other tracers. Additionally, clean up the unnecessary 'iter->private = NULL' during each 'cat trace' when using wakeup and irqsoff tracers. [1] https://lore.kernel.org/all/20231112150030.84609-1-ryncsn@gmail.com/
CVE-2025-22036	In the Linux kernel, the following vulnerability has been resolved: exfat: fix random stack corruption after get_block When get_block is called with a buffer_head allocated on the stack, such as do_mpage_readpage, stack corruption due to buffer_head UAF may occur in the following race condition situation. <CPU 0> <CPU 1> mpage_read_folio <<bh on stack>> do_mpage_readpage exfat_get_block bh_read __bh_read get_bh(bh) submit_bh wait_on_buffer ... end_buffer_read_sync __end_buffer_read_notouch unlock_buffer <<keep going>> <<bh is not valid out of mpage_read_folio>> .. another_function <<variable A on stack>> put_bh(bh) atomic_dec(bh->b_count) * stack corruption here * This patch returns -EAGAIN if a folio does not have buffers when bh_read needs to be called. By doing this, the caller can fallback to functions like block_read_full_folio(), create a buffer_head in the folio, and then call get_block again. Let's do not call bh_read() with on-stack buffer_head.
CVE-2025-22037	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix null pointer dereference in alloc_preauth_hash() The Client send malformed smb2 negotiate request. ksmbd return error response. Subsequently, the client can send smb2 session setup even though conn->preauth_info is not allocated. This patch add KSMDBD_SESS_NEED_SETUP status of connection to ignore session setup request if smb2 negotiate phase is not complete.
CVE-2025-22038	In the Linux kernel, the following vulnerability has been resolved: ksmbd: validate zero num_subauth before sub_auth is accessed Access psid->sub_auth[psid->num_subauth - 1] without checking if num_subauth is non-zero leads to an out-of-bounds read. This patch adds a validation step to ensure num_subauth != 0 before sub_auth is accessed.

CVE ID	Description
CVE-2025-22039	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix overflow in dacloffset bounds check The dacloffset field was originally typed as int and used in an unchecked addition, which could overflow and bypass the existing bounds check in both smb_check_perm_dacl() and smb_inherit_dacl(). This could result in out-of-bounds memory access and a kernel crash when dereferencing the DACL pointer. This patch converts dacloffset to unsigned int and uses check_add_overflow() to validate access to the DACL.
CVE-2025-22040	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix session use-after-free in multichannel connection There is a race condition between session setup and ksmbd_sessions_deregister. The session can be freed before the connection is added to channel list of session. This patch check reference count of session before freeing it.
CVE-2025-22041	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in ksmbd_sessions_deregister() In multichannel mode, UAF issue can occur in session_deregister when the second channel sets up a session through the connection of the first channel. session that is freed through the global session table can be accessed again through ->sessions of connection.
CVE-2025-22042	In the Linux kernel, the following vulnerability has been resolved: ksmbd: add bounds check for create lease context Add missing bounds check for create lease context.
CVE-2025-22043	In the Linux kernel, the following vulnerability has been resolved: ksmbd: add bounds check for durable handle context Add missing bounds check for durable handle context.
CVE-2025-22044	In the Linux kernel, the following vulnerability has been resolved: acpi: nfit: fix narrowing conversion in acpi_nfit_ctl Syzkaller has reported a warning in to_nfit_bus_uuid(): "only secondary bus families can be translated". This warning is emitted if the argument is equal to NVDIMM_BUS_FAMILY_NFIT == 0. Function acpi_nfit_ctl() first verifies that a user-provided value call_pkg->nd_family of type u64 is not equal to 0. Then the value is converted to int, and only after that is compared to NVDIMM_BUS_FAMILY_MAX. This can lead to passing an invalid argument to acpi_nfit_ctl(), if call_pkg->nd_family is non-zero, while the lower 32 bits are zero. Furthermore, it is best to return EINVAL immediately upon seeing the invalid user input. The WARNING is insufficient to prevent further undefined behavior based on other invalid user input. All checks of the input value should be applied to the original variable call_pkg->nd_family. [iweiny: update commit message]
CVE-2025-22045	In the Linux kernel, the following vulnerability has been resolved: x86/mm: Fix flush_tlb_range() when used for zapping normal PMDs On the following path, flush_tlb_range() can be used for zapping normal PMD entries (PMD entries that point to page tables) together with the PTE entries in the pointed-to page table: collapse_pte_mapped_thp pmdp_collapse_flush flush_tlb_range The arm64 version of flush_tlb_range() has a comment describing that it can be used for page table removal, and does not use any last-level invalidation optimizations. Fix the X86 version by making it behave the same way. Currently, X86 only uses this information for the following two purposes, which I think means the issue doesn't have much impact: - In native_flush_tlb_multi() for checking if lazy TLB CPUs need to be IPT'd to avoid issues with speculative page table walks. - In Hyper-V TLB paravirtualization, again for lazy TLB stuff. The patch "x86/mm: only invalidate final translations with INVLPGB" which is currently under review (see < https://lore.kernel.org/all/20241230175550.4046587-13-riel@surriel.com/ >) would probably be making the impact of this a lot worse.
CVE-2025-22047	In the Linux kernel, the following vulnerability has been resolved: x86/microcode/AMD: Fix __apply_microcode_amd()'s return value When verify_sha256_digest() fails, __apply_microcode_amd() should propagate the failure by returning false (and not -1 which is promoted to true).
CVE-2025-22050	In the Linux kernel, the following vulnerability has been resolved: usbnet:fix NPE during rx_complete Missing usbnet_going_away Check in Critical Path. The usb_submit_urb function lacks a usbnet_going_away validation, whereas __usbnet_queue_skb includes this check. This inconsistency creates a race condition where: A URB request may succeed, but the corresponding SKB data fails to be queued. Subsequent processes: (e.g., rx_complete # defer_bh # __skb_unlink(skb, list)) attempt to access skb->next, triggering a NULL pointer dereference (Kernel Panic).

CVE ID	Description
CVE-2025-22053	In the Linux kernel, the following vulnerability has been resolved: net: ibmveth: make veth_pool_store stop hanging v2: - Created a single error handling unlock and exit in veth_pool_store - Greatly expanded commit message with previous explanatory-only text Summary: Use rtnl_mutex to synchronize veth_pool_store with itself, ibmveth_close and ibmveth_open, preventing multiple calls in a row to napi_disable. Background: Two (or more) threads could call veth_pool_store through writing to /sys/devices/vio/30000002/pool*/*. You can do this easily with a little shell script. This causes a hang. I configured LOCKDEP, compiled ibmveth.c with DEBUG, and built a new kernel. I ran this test again and saw: Setting pool0/active to 0 Setting pool1/active to 1 [73.911067][T4365] ibmveth 30000002 eth0: close starting Setting pool1/active to 1 Setting pool1/active to 0 [73.911367] [T4366] ibmveth 30000002 eth0: close starting [73.916056][T4365] ibmveth 30000002 eth0: close complete [73.916064][T4365] ibmveth 30000002 eth0: open starting [110.808564][T712] systemd-journald[712]: Sent WATCHDOG=1 notification. [230.808495][T712] systemd-journald[712]: Sent WATCHDOG=1 notification. [243.683786][T123] INFO: task stress.sh:4365 blocked for more than 122 seconds. [243.683827][T123] Not tainted 6.14.0-01103-g2df0c02dab82-dirty #8 [243.683833] [T123] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. [243.683838] [T123] task:stress.sh state:D stack:28096 pid:4365 tgid:4365 ppid:4364 task_flags:0x400040 flags:0x00042000 [243.683852][T123] Call Trace: [243.683857][T123] [c00000000c38f690] [0000000000000001] 0x1 (unreliable) [243.683868][T123] [c00000000c38f840] [c00000000001f908] __switch_to+0x318/0x4e0 [243.683878][T123] [c00000000c38f8a0] [c000000001549a70] __schedule+0x500/0x12a0 [243.683888][T123] [c00000000c38f9a0] [c00000000154a878] schedule+0x68/0x210 [243.683896][T123] [c00000000c38f9d0] [c00000000154ac80] schedule_preempt_disabled+0x30/0x50 [243.683904][T123] [c00000000c38fa00] [c00000000154dbb0] __mutex_lock+0x730/0x10f0 [243.683913][T123] [c00000000c38fb10] [c000000001154d40] napi_enable+0x30/0x60 [243.683921][T123] [c00000000c38fb40] [c000000000f4ae94] ibmveth_open+0x68/0x5dc [243.683928][T123] [c00000000c38fbe0] [c000000000f4aa20] veth_pool_store+0x220/0x270 [243.683936][T123] [c00000000c38fc70] [c000000000826278] sysfs_kf_write+0x68/0xb0 [243.683944][T123] [c00000000c38fcb0] [c0000000008240b8] kernfs_fop_write_iter+0x198/0x2d0 [243.683951][T123] [c00000000c38fd00] [c00000000071b9ac] vfs_write+0x34c/0x650 [243.683958][T123] [c00000000c38fdc0] [c00000000071bea8] ksys_write+0x88/0x150 [243.683966] [T123] [c00000000c38fe10] [c000000000317f4] system_call_exception+0x124/0x340 [243.683973] [T123] [c00000000c38fe50] [c0000000000d05c] system_call_vectored_common+0x15c/0x2ec ... [243.684087][T123] Showing all locks held in the system: [243.684095][T123] 1 lock held by khungtaskd/123: [243.684099][T123] #0: c00000000278e370 (rcu_read_lock){....}-{1:2}, at: debug_show_all_locks+0x50/0x248 [243.684114][T123] 4 locks held by stress.sh/4365: [243.684119] [T123] #0: c00000003a4cd3f8 (sb_writers#3){.+.}-{0:0}, at: ksys_write+0x88/0x150 [243.684132] [T123] #1: c000000041aea888 (&of->mutex#2){.+.}-{3:3}, at: kernfs_fop_write_iter+0x154/0x2d0 [243.684143][T123] #2: c0000000366fb9a8 (kn->active#64){.+.}-{0:0}, at: kernfs_fop_write_iter+0x160/0x2d0 [243.684155][T123] #3: c000000035ff4cb8 (&dev->lock){.+.}-{3:3}, at: napi_enable+0x30/0x60 [243.684166][T123] 5 locks held by stress.sh/4366: [243.684170][T123] #0: c00000003a4cd3f8 (sb_writers#3){.+.}-{0:0}, at: ksys_write+0x88/0x150 [243. ---truncated---
CVE-2025-22054	In the Linux kernel, the following vulnerability has been resolved: arcnet: Add NULL check in com20020pci_probe() devm_kasprintf() returns NULL when memory allocation fails. Currently, com20020pci_probe() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue and ensure no resources are left allocated.

CVE ID	Description
CVE-2025-22055	In the Linux kernel, the following vulnerability has been resolved: net: fix geneve_opt length integer overflow struct geneve_opt uses 5 bit length for each single option, which means every vary size option should be smaller than 128 bytes. However, all current related Netlink policies cannot promise this length condition and the attacker can exploit a exact 128-byte size option to *fake* a zero length option and confuse the parsing logic, further achieve heap out-of-bounds read. One example crash log is like below: [3.905425] <pre>===== [3.905925] BUG: KASAN: slab-out-of-bounds in nla_put+0xa9/0xe0 [3.906255] Read of size 124 at addr ffff888005f291cc by task poc/177 [3.906646] [3.906775] CPU: 0 PID: 177 Comm: poc- oob-read Not tainted 6.1.132 #1 [3.907131] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 [3.907784] Call Trace: [3.907925] <TASK> [3.908048] dump_stack_lvl+0x44/0x5c [3.908258] print_report+0x184/0x4be [3.909151] kasan_report+0xc5/0x100 [3.909539] kasan_check_range+0xf3/0x1a0 [3.909794] memcpy+0x1f/0x60 [3.909968] nla_put+0xa9/0xe0 [3.910147] tunnel_key_dump+0x945/0xba0 [3.911536] tcf_action_dump_1+0x1c1/0x340 [3.912436] tcf_action_dump+0x101/0x180 [3.912689] tcf_exts_dump+0x164/0x1e0 [3.912905] fw_dump+0x18b/0x2d0 [3.913483] tcf_fill_node +0x2ee/0x460 [3.914778] tfilter_notify+0xf4/0x180 [3.915208] tc_new_tfilter+0xd51/0x10d0 [3.918615] rtnetlink_rcv_msg+0x4a2/0x560 [3.919118] netlink_rcv_skb+0xcd/0x200 [3.919787] netlink_unicast+0x395/0x530 [3.921032] netlink_sendmsg+0x3d0/0x6d0 [3.921987] __sock_sendmsg +0x99/0xa0 [3.922220] __sys_sendto+0x1b7/0x240 [3.922682] __x64_sys_sendto+0x72/0x90 [3.922906] do_syscall_64+0x5e/0x90 [3.923814] entry_SYSCALL_64_after_hwframe+0x6e/0xd8 [3.924122] RIP: 0033:0x7e83eab84407 [3.924331] Code: 48 89 fa 4c 89 df e8 38 aa 00 00 8b 93 08 03 00 00 59 5e 48 83 f8 fc 74 1a 5b c3 0f 1f 84 00 00 00 00 00 48 8b 44 24 10 0f 05 <5b> c3 0f 1f 80 00 00 00 00 83 e2 39 83 faf [3.925330] RSP: 002b:00007ffff505e370 EFLAGS: 00000202 ORIG_RAX: 000000000000002c [3.925752] RAX: ffffffffdfda RBX: 00007e83eafa740 RCX: 00007e83eab84407 [3.926173] RDX: 000000000000001a8 RSI: 00007ffff505e3c0 RDI: 0000000000000003 [3.926587] RBP: 00007ffff505f460 R08: 00007e83eace1000 R09: 0000000000000000 [3.926977] R10: 0000000000000000 R11: 0000000000000202 R12: 00007ffff505f3c0 [3.927367] R13: 00007ffff505f5c8 R14: 00007e83ead1b000 R15: 00005d4fbbe6dcb8 Fix these issues by enforcing correct length condition in related policies.</pre>
CVE-2025-22056	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_tunnel: fix geneve_opt type confusion addition When handling multiple NFTA_TUNNEL_KEY_OPTS_GENEVE attributes, the parsing logic should place every geneve_opt structure one by one compactly. Hence, when deciding the next geneve_opt position, the pointer addition should be in units of char *. However, the current implementation erroneously does type conversion before the addition, which will lead to heap out-of-bounds write. [6.989857] <pre>===== [6.990293] BUG: KASAN: slab-out-of-bounds in nft_tunnel_obj_init+0x977/0xa70 [6.990725] Write of size 124 at addr ffff888005f18974 by task poc/178 [6.991162] [6.991259] CPU: 0 PID: 178 Comm: poc-oob-write Not tainted 6.1.132 #1 [6.991655] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 [6.992281] Call Trace: [6.992423] <TASK> [6.992586] dump_stack_lvl+0x44/0x5c [6.992801] print_report +0x184/0x4be [6.993790] kasan_report+0xc5/0x100 [6.994252] kasan_check_range+0xf3/0x1a0 [6.994486] memcpy+0x38/0x60 [6.994692] nft_tunnel_obj_init+0x977/0xa70 [6.995677] nft_obj_init +0x10c/0x1b0 [6.995891] nf_tables_newobj+0x585/0x950 [6.996922] nfnetlink_rcv_batch +0xdf9/0x1020 [6.998997] nfnetlink_rcv+0x1df/0x220 [6.999537] netlink_unicast+0x395/0x530 [7.000771] netlink_sendmsg+0x3d0/0x6d0 [7.001462] __sock_sendmsg+0x99/0xa0 [7.001707] __sys_sendmsg+0x409/0x450 [7.002391] __sys_sendmsg+0xfd/0x170 [7.003145] __sys_sendmsg +0xea/0x170 [7.004359] do_syscall_64+0x5e/0x90 [7.005817] entry_SYSCALL_64_after_hwframe +0x6e/0xd8 [7.006127] RIP: 0033:0x7ec756d4e407 [7.006339] Code: 48 89 fa 4c 89 df e8 38 aa 00 00 8b 93 08 03 00 00 59 5e 48 83 f8 fc 74 1a 5b c3 0f 1f 84 00 00 00 00 00 48 8b 44 24 10 0f 05 <5b> c3 0f 1f 80 00 00 00 00 83 e2 39 83 faf [7.007364] RSP: 002b:00007ffed5d46760 EFLAGS: 00000202 ORIG_RAX: 000000000000002e [7.007827] RAX: ffffffffdfda RBX: 00007ec756cc4740 RCX: 00007ec756d4e407 [7.008223] RDX: 0000000000000000 RSI: 00007ffed5d467f0 RDI: 0000000000000003 [7.008620] RBP: 00007ffed5d468a0 R08: 0000000000000000 R09: 0000000000000000 [7.009039] R10: 0000000000000000 R11: 0000000000000202 R12: 0000000000000000 [7.009429] R13: 00007ffed5d478b0 R14: 00007ec756ee5000 R15: 00005cbd4e655cb8 Fix this bug with correct pointer addition and conversion in parse and dump code.</pre>

CVE ID	Description
CVE-2025-22057	In the Linux kernel, the following vulnerability has been resolved: net: decrease cached dst counters in dst_release Upstream fix ac888d58869b ("net: do not delay dst_entries_add() in dst_release()") moved decrementing the dst count from dst_destroy to dst_release to avoid accessing already freed data in case of netns dismantle. However in case CONFIG_DST_CACHE is enabled and OvS+tunnels are used, this fix is incomplete as the same issue will be seen for cached dsts: Unable to handle kernel paging request at virtual address ffff5aabf6b5c000 Call trace: percpu_counter_add_batch+0x3c/0x160 (P) dst_release+0xec/0x108 dst_cache_destroy+0x68/0xd8 dst_destroy+0x13c/0x168 dst_destroy_rcu+0x1c/0xb0 rcu_do_batch+0x18c/0x7d0 rcu_core+0x174/0x378 rcu_core_si+0x18/0x30 Fix this by invalidating the cache, and thus decrementing cached dst counters, in dst_release too.
CVE-2025-22058	In the Linux kernel, the following vulnerability has been resolved: udp: Fix memory accounting leak. Matt Dowling reported a weird UDP memory usage issue. Under normal operation, the UDP memory usage reported in /proc/net/sockstat remains close to zero. However, it occasionally spiked to 524,288 pages and never dropped. Moreover, the value doubled when the application was terminated. Finally, it caused intermittent packet drops. We can reproduce the issue with the script below [0]: 1. /proc/net/sockstat reports 0 pages # cat /proc/net/sockstat grep UDP: UDP: inuse 1 mem 0 2. Run the script till the report reaches 524,288 # python3 test.py & sleep 5 # cat /proc/net/sockstat grep UDP: UDP: inuse 3 mem 524288 <-- (INT_MAX + 1) >> PAGE_SHIFT 3. Kill the socket and confirm the number never drops # pkill python3 && sleep 5 # cat /proc/net/sockstat grep UDP: UDP: inuse 1 mem 524288 4. (necessary since v6.0) Trigger proto_memory_pcpu_drain() # python3 test.py & sleep 1 && pkill python3 5. The number doubles # cat /proc/net/sockstat grep UDP: UDP: inuse 1 mem 1048577 The application set INT_MAX to SO_RCVBUF, which triggered an integer overflow in udp_rmem_release(). When a socket is close(), udp_destruct_common() purges its receive queue and sums up skb->truesize in the queue. This total is calculated and stored in a local unsigned integer variable. The total size is then passed to udp_rmem_release() to adjust memory accounting. However, because the function takes a signed integer argument, the total size can wrap around, causing an overflow. Then, the released amount is calculated as follows: 1) Add size to sk->sk_forward_alloc. 2) Round down sk->sk_forward_alloc to the nearest lower multiple of PAGE_SIZE and assign it to amount. 3) Subtract amount from sk->sk_forward_alloc. 4) Pass amount >> PAGE_SHIFT to __sk_mem_reduce_allocated(). When the issue occurred, the total in udp_destruct_common() was 2147484480 (INT_MAX + 833), which was cast to -2147482816 in udp_rmem_release(). At 1) sk->sk_forward_alloc is changed from 3264 to -2147479552, and 2) sets -2147479552 to amount. 3) reverts the wraparound, so we don't see a warning in inet_sock_destruct(). However, udp_memory_allocated ends up doubling at 4). Since commit 3cd3399dd7a8 ("net: implement per-cpu reserves for memory_allocated"), memory usage no longer doubles immediately after a socket is close() because __sk_mem_reduce_allocated() caches the amount in udp_memory_per_cpu_fw_alloc. However, the next time a UDP socket receives a packet, the subtraction takes effect, causing UDP memory usage to double. This issue makes further memory allocation fail once the socket's sk->sk_rmem_alloc exceeds net.ipv4.udp_rmem_min, resulting in packet drops. To prevent this issue, let's use unsigned int for the calculation and call sk_forward_alloc_add() only once for the small delta. Note that first_packet_length() also potentially has the same problem. [0]: from socket import * SO_RCVBUFFORCE = 33 INT_MAX = (2 ** 31) - 1 s = socket(AF_INET, SOCK_DGRAM) s.bind(('', 0)) s.setsockopt(SOL_SOCKET, SO_RCVBUFFORCE, INT_MAX) c = socket(AF_INET, SOCK_DGRAM) c.connect(s.getsockname()) data = b'a' * 100 while True: c.send(data)
CVE-2025-22060	In the Linux kernel, the following vulnerability has been resolved: net: mvpp2: Prevent parser TCAM memory corruption Protect the parser TCAM/SRAM memory, and the cached (shadow) SRAM information, from concurrent modifications. Both the TCAM and SRAM tables are indirectly accessed by configuring an index register that selects the row to read or write to. This means that operations must be atomic in order to, e.g., avoid spreading writes across multiple rows. Since the shadow SRAM array is used to find free rows in the hardware table, it must also be protected in order to avoid TOCTOU errors where multiple cores allocate the same row. This issue was detected in a situation where `mvpp2_set_rx_mode()` ran concurrently on two CPUs. In this particular case the MVPP2_PE_MAC_UC_PROMISCUOUS entry was corrupted, causing the classifier unit to drop all incoming unicast - indicated by the `rx_classifier_drops` counter.

CVE ID	Description
CVE-2025-22062	In the Linux kernel, the following vulnerability has been resolved: sctp: add mutual exclusion in proc_sctp_do_udp_port() We must serialize calls to sctp_udp_sock_stop() and sctp_udp_sock_start() or risk a crash as syzbot reported: Oops: general protection fault, probably for non-canonical address 0xdffffc000000000d: 0000 [#1] SMP KASAN PTI KASAN: null-ptr-deref in range [0x0000000000000068-0x000000000000006f] CPU: 1 UID: 0 PID: 6551 Comm: syz.1.44 Not tainted 6.14.0-syzkaller-g7f2ff7b62617 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 RIP: 0010:kernel_sock_shutdown+0x47/0x70 net/socket.c:3653 Call Trace: <TASK> udp_tunnel_sock_release+0x68/0x80 net/ipv4/udp_tunnel_core.c:181 sctp_udp_sock_stop+0x71/0x160 net/sctp/protocol.c:930 proc_sctp_do_udp_port+0x264/0x450 net/sctp/sysctl.c:553 proc_sys_call_handler+0x3d0/0x5b0 fs/proc/proc_sysctl.c:601 iter_file_splice_write+0x91c/0x1150 fs/splice.c:738 do_splice_from fs/splice.c:935 [inline] direct_splice_actor+0x18f/0x6c0 fs/splice.c:1158 splice_direct_to_actor+0x342/0xa30 fs/splice.c:1102 do_splice_direct_actor fs/splice.c:1201 [inline] do_splice_direct+0x174/0x240 fs/splice.c:1227 do_sendfile+0xafd/0xe50 fs/read_write.c:1368 __do_sys_sendfile64 fs/read_write.c:1429 [inline] __se_sys_sendfile64 fs/read_write.c:1415 [inline] __x64_sys_sendfile64+0x1d8/0x220 fs/read_write.c:1415 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline]
CVE-2025-22063	In the Linux kernel, the following vulnerability has been resolved: netlabel: Fix NULL pointer exception caused by CALIPSO on IPv4 sockets When calling netlbl_conn_setattr(), addr->sa_family is used to determine the function behavior. If sk is an IPv4 socket, but the connect function is called with an IPv6 address, the function calipso_sock_setattr() is triggered. Inside this function, the following code is executed: sk_fullsock(__sk) ? inet_sk(__sk)->pinet6 : NULL; Since sk is an IPv4 socket, pinet6 is NULL, leading to a null pointer dereference. This patch fixes the issue by checking if inet6_sk(sk) returns a NULL pointer before accessing pinet6.
CVE-2025-22064	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_tables: don't unregister hook when table is dormant When nf_tables_updchain encounters an error, hook registration needs to be rolled back. This should only be done if the hook has been registered, which won't happen when the table is flagged as dormant (inactive). Just move the assignment into the registration block.
CVE-2025-22065	In the Linux kernel, the following vulnerability has been resolved: idpf: fix adapter NULL pointer dereference on reboot With SRIOV enabled, idpf ends up calling into idpf_remove() twice. First via idpf_shutdown() and then again when idpf_remove() calls into sriov_disable(), because the VF devices use the idpf driver, hence the same remove routine. When that happens, it is possible for the adapter to be NULL from the first call to idpf_remove(), leading to a NULL pointer dereference. echo 1 > /sys/class/net/<netif>/device/sriov_numvfs reboot BUG: kernel NULL pointer dereference, address: 0000000000000020 ... RIP: 0010:idpf_remove+0x22/0x1f0 [idpf] ... ? idpf_remove+0x22/0x1f0 [idpf] ? idpf_remove+0x1e4/0x1f0 [idpf] pci_device_remove+0x3f/0xb0 device_release_driver_internal+0x19f/0x200 pci_stop_bus_device+0x6d/0x90 pci_stop_and_remove_bus_device+0x12/0x20 pci_iiov_remove_virtfn+0xbe/0x120 sriov_disable+0x34/0xe0 idpf_sriov_configure+0x58/0x140 [idpf] idpf_remove+0x1b9/0x1f0 [idpf] idpf_shutdown+0x12/0x30 [idpf] pci_device_shutdown+0x35/0x60 device_shutdown+0x156/0x200 ... Replace the direct idpf_remove() call in idpf_shutdown() with idpf_vc_core_deinit() and idpf_deinit_dflt_mbx(), which perform the bulk of the cleanup, such as stopping the init task, freeing IRQs, destroying the vports and freeing the mailbox. This avoids the calls to sriov_disable() in addition to a small netdev cleanup, and destroying workqueues, which don't seem to be required on shutdown.
CVE-2025-22066	In the Linux kernel, the following vulnerability has been resolved: ASoC: imx-card: Add NULL check in imx_card_probe() devm_kasprintf() returns NULL when memory allocation fails. Currently, imx_card_probe() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue.
CVE-2025-22068	In the Linux kernel, the following vulnerability has been resolved: ublk: make sure ubq->canceling is set when queue is frozen Now ublk driver depends on `ubq->canceling` for deciding if the request can be dispatched via uring_cmd & io_uring_cmd_complete_in_task(). Once ubq->canceling is set, the uring_cmd can be done via ublk_cancel_cmd() and io_uring_cmd_done(). So set ubq->canceling when queue is frozen, this way makes sure that the flag can be observed from ublk_queue_rq() reliably, and avoids use-after-free on uring_cmd.

CVE ID	Description
CVE-2025-22070	In the Linux kernel, the following vulnerability has been resolved: fs/9p: fix NULL pointer dereference on mkdir When a 9p tree was mounted with option 'posixacl', parent directory had a default ACL set for its subdirectories, e.g.: setfacl -m default:group:simpsons:rxw parentdir then creating a subdirectory crashed 9p client, as v9fs_fid_add() call in function v9fs_vfs_mkdir_dotl() sets the passed 'fid' pointer to NULL (since dafbe689736) even though the subsequent v9fs_set_create_acl() call expects a valid non-NULL 'fid' pointer: [37.273191] BUG: kernel NULL pointer dereference, address: 0000000000000000 ... [37.322338] Call Trace: [37.323043] <TASK> [37.323621] ? __die (arch/x86/kernel/dumpstack.c:421 arch/x86/kernel/dumpstack.c:434) [37.324448] ? page_fault_oops (arch/x86/mm/fault.c:714) [37.325532] ? search_module_extables (kernel/module/main.c:3733) [37.326742] ? p9_client_walk (net/9p/client.c:1165) 9pnet [37.328006] ? search_bpf_extables (kernel/bpf/core.c:804) [37.329142] ? exc_page_fault (/arch/x86/include/asm/paravirt.h:686 arch/x86/mm/fault.c:1488 arch/x86/mm/fault.c:1538) [37.330196] ? asm_exc_page_fault (/arch/x86/include/asm/idtentry.h:574) [37.331330] ? p9_client_walk (net/9p/client.c:1165) 9pnet [37.332562] ? v9fs_fid_xattr_get (fs/9p/xattr.c:30) 9p [37.333824] v9fs_fid_xattr_set (fs/9p/fid.h:23 fs/9p/xattr.c:121) 9p [37.335077] v9fs_set_acl (fs/9p/acl.c:276) 9p [37.336112] v9fs_set_create_acl (fs/9p/acl.c:307) 9p [37.337326] v9fs_vfs_mkdir_dotl (fs/9p/vfs_inode_dotl.c:411) 9p [37.338590] v9fs_mkdir (fs/namei.c:4313) [37.339535] do_mkdirat (fs/namei.c:4336) [37.340465] __x64_sys_mkdir (fs/namei.c:4354) [37.341455] do_syscall_64 (arch/x86/entry/common.c:52 arch/x86/entry/common.c:83) [37.342447] entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) Fix this by simply swapping the sequence of these two calls in v9fs_vfs_mkdir_dotl(), i.e. calling v9fs_set_create_acl() before v9fs_fid_add().
CVE-2025-22071	In the Linux kernel, the following vulnerability has been resolved: spufs: fix a leak in spufs_create_context() Leak fixes back in 2008 missed one case - if we are trying to set affinity and spufs_mkdir() fails, we need to drop the reference to neighbor.
CVE-2025-22072	In the Linux kernel, the following vulnerability has been resolved: spufs: fix gang directory lifetimes prior to "[POWERPC] spufs: Fix gang destroy leaks" we used to have a problem with gang lifetimes - creation of a gang returns opened gang directory, which normally gets removed when that gets closed, but if somebody has created a context belonging to that gang and kept it alive until the gang got closed, removal failed and we ended up with a leak. Unfortunately, it had been fixed the wrong way. Dentry of gang directory was no longer pinned, and rmdir on close was gone. One problem was that failure of open kept calling simple_rmdir() as cleanup, which meant an unbalanced dput(). Another bug was in the success case - gang creation incremented link count on root directory, but that was no longer undone when gang got destroyed. Fix consists of * reverting the commit in question * adding a counter to gang, protected by ->i_rwsem of gang directory inode. * having it set to 1 at creation time, dropped in both spufs_dir_close() and spufs_gang_close() and bumped in spufs_create_context(), provided that it's not 0. * using simple_recursive_removal() to take the gang directory out when counter reaches zero.
CVE-2025-22073	In the Linux kernel, the following vulnerability has been resolved: spufs: fix a leak on spufs_new_file() failure It's called from spufs_fill_dir(), and caller of that will do spufs_rmdir() in case of failure. That does remove everything we'd managed to create, but... the problem dentry is still negative. IOW, it needs to be explicitly dropped.

CVE ID	Description
CVE-2025-22075	In the Linux kernel, the following vulnerability has been resolved: rtnetlink: Allocate vfinfo size for VF GUIDs when supported Commit 30aad41721e0 ("net/core: Add support for getting VF GUIDs") added support for getting VF port and node GUIDs in netlink ifinfo messages, but their size was not taken into consideration in the function that allocates the netlink message, causing the following warning when a netlink message is filled with many VF port and node GUIDs: # echo 64 > /sys/bus/pci/devices/0000:08:00:00/sriov_numvfs # ip link show dev ib0 RTNETLINK answers: Message too long Cannot send link get request: Message too long Kernel warning: -----[cut here]----- WARNING: CPU: 2 PID: 1930 at net/core/rtnetlink.c:4151 rtnl_getlink+0x586/0x5a0 Modules linked in: xt_conntrack xt_MASQUERADE nfnetlink xt_addrtype iptable_nat nf_nat br_netfilter overlay mlx5_ib macsec mlx5_core tls rpcrdma rdma_ucm ib_uverbs ib_iser libiscsi scsi_transport_iscsi ib_umad rdma_cm iw_cm ib_ipoib fuse ib_cm ib_core CPU: 2 UID: 0 PID: 1930 Comm: ip Not tainted 6.14.0-rc2+ #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:rtnl_getlink+0x586/0x5a0 Code: cb 82 e8 3d af 0a 00 4d 85 ff 0f 84 08 ff ff 4c 89 ff 41 be ea ff ff e8 66 63 5b ff 49 c7 07 80 4f cb 82 e9 36 fc ff ff <0f> 0b e9 16 fe ff ff e8 de a0 56 00 66 66 2e 0f 1f 84 00 00 00 00 RSP: 0018:ffff888113557348 EFLAGS: 00010246 RAX: 00000000fffffa6 RBX: ffff88817e87aa34 RCX: dffffc0000000000 RDX: 0000000000000003 RSI: 0000000000000000 RDI: ffff88817e87afb8 RBP: 0000000000000009 R08: ffffffff821f44aa R09: 0000000000000000 R10: ffff8881260f79a8 R11: ffff88817e87af00 R12: ffff88817e87aa00 R13: ffffffff8563d300 R14: 00000000fffffa6 R15: 00000000fffffff FS: 00007f63a5dbf280(0000) GS:ffff8881ee000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f63a5ba4493 CR3: 00000001700fe002 CR4: 0000000000772eb0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> ? __warn+0xa5/0x230 ? rtnl_getlink+0x586/0x5a0 ? report_bug+0x22d/0x240 ? handle_bug+0x53/0xa0 ? exc_invalid_op+0x14/0x50 ? asm_exc_invalid_op+0x16/0x20 ? skb_trim+0x6a/0x80 ? rtnl_getlink+0x586/0x5a0 ? __pfx_rtnl_getlink+0x10/0x10 ? rtnetlink_rcv_msg+0x1e5/0x860 ? __pfx__mutex_lock+0x10/0x10 ? rcu_is_watching+0x34/0x60 ? __pfx_lock_acquire+0x10/0x10 ? stack_trace_save+0x90/0xd0 ? filter_irq_stacks+0x1d/0x70 ? kasan_save_stack+0x30/0x40 ? kasan_save_stack+0x20/0x40 ? kasan_save_track+0x10/0x30 rtnetlink_rcv_msg+0x21c/0x860 ? entry_SYSCALL_64_after_hwframe+0x76/0x7e ? __pfx_rtnetlink_rcv_msg+0x10/0x10 ? arch_stack_walk+0x9e/0xf0 ? rcu_is_watching+0x34/0x60 ? lock_acquire+0xd5/0x410 ? rcu_is_watching+0x34/0x60 netlink_rcv_skb+0xe0/0x210 ? __pfx_rtnetlink_rcv_msg+0x10/0x10 ? __pfx_netlink_rcv_skb+0x10/0x10 ? rcu_is_watching+0x34/0x60 ? __pfx__netlink_lookup+0x10/0x10 ? lock_release+0x62/0x200 ? netlink_deliver_tap+0xfd/0x290 ? rcu_is_watching+0x34/0x60 ? lock_release+0x62/0x200 ? netlink_deliver_tap+0x95/0x290 netlink_unicast+0x31f/0x480 ? __pfx_netlink_unicast+0x10/0x10 ? rcu_is_watching+0x34/0x60 ? lock_acquire+0xd5/0x410 netlink_sendmsg+0x369/0x660 ? lock_release+0x62/0x200 ? __pfx_netlink_sendmsg+0x10/0x10 ? import_ubuf+0xb9/0xf0 ? __import_iovec+0x254/0x2b0 ? lock_release+0x62/0x200 ? __pfx_netlink_sendmsg+0x10/0x10 __sys_sendmsg+0x559/0x5a0 ? __pfx__sys_sendmsg+0x10/0x10 ? __pfx_copy_msghdr_from_user+0x10/0x10 ? rcu_is_watching+0x34/0x60 ? do_read_fault+0x213/0x4a0 ? rcu_is_watching+0x34/0x60 __sys_sendmsg+0xe4/0x150 ? __pfx__sys_sendmsg+0x10/0x10 ? do_fault+0x2cc/0x6f0 ? handle_pte_fault+0x2e3/0x3d0 ? __pfx_handle_pte_fault+0x10/0x10 ---truncated---

CVE ID	Description
CVE-2025-22077	In the Linux kernel, the following vulnerability has been resolved: Revert "smb: client: fix TCP timers deadlock after rmmmod" This reverts commit e9f2517a3e18a54a3943c098d2226b245d488801. Commit e9f2517a3e18 ("smb: client: fix TCP timers deadlock after rmmmod") is intended to fix a null-ptr-deref in LOCKDEP, which is mentioned as CVE-2024-54680, but is actually did not fix anything; The issue can be reproduced on top of it. [0] Also, it reverted the change by commit ef7134c7fc48 ("smb: client: Fix use-after-free of network namespace.") and introduced a real issue by reviving the kernel TCP socket. When a reconnect happens for a CIFS connection, the socket state transitions to FIN_WAIT_1. Then, inet_csk_clear_xmit_timers_sync() in tcp_close() stops all timers for the socket. If an incoming FIN packet is lost, the socket will stay at FIN_WAIT_1 forever, and such sockets could be leaked up to net.ipv4.tcp_max_orphans. Usually, FIN can be retransmitted by the peer, but if the peer aborts the connection, the issue comes into reality. I warned about this privately by pointing out the exact report [1], but the bogus fix was finally merged. So, we should not stop the timers to finally kill the connection on our side in that case, meaning we must not use a kernel socket for TCP whose sk->sk_net_refcnt is 0. The kernel socket does not have a reference to its netns to make it possible to tear down netns without cleaning up every resource in it. For example, tunnel devices use a UDP socket internally, but we can destroy netns without removing such devices and let it complete during exit. Otherwise, netns would be leaked when the last application died. However, this is problematic for TCP sockets because TCP has timers to close the connection gracefully even after the socket is close()d. The lifetime of the socket and its netns is different from the lifetime of the underlying connection. If the socket user does not maintain the netns lifetime, the timer could be fired after the socket is close()d and its netns is freed up, resulting in use-after-free. Actually, we have seen so many similar issues and converted such sockets to have a reference to netns. That's why I converted the CIFS client socket to have a reference to netns (sk->sk_net_refcnt == 1), which is somehow mentioned as out-of-scope of CIFS and technically wrong in e9f2517a3e18, but **is in-scope and right fix**. Regarding the LOCKDEP issue, we can prevent the module unload by bumping the module refcount when switching the LOCKDDEP key in sock_lock_init_class_and_name(). [2] For a while, let's revert the bogus fix. Note that now we can use sk_net_refcnt_upgrade() for the socket conversion, but I'll do so later separately to make backport easy.
CVE-2025-22079	In the Linux kernel, the following vulnerability has been resolved: ocfs2: validate l_tree_depth to avoid out-of-bounds access The l_tree_depth field is 16-bit (__le16), but the actual maximum depth is limited to OCFS2_MAX_PATH_DEPTH. Add a check to prevent out-of-bounds access if l_tree_depth has an invalid value, which may occur when reading from a corrupted mounted disk [1].
CVE-2025-22080	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Prevent integer overflow in hdr_first_de() The "de_off" and "used" variables come from the disk so they both need to check. The problem is that on 32bit systems if they're both greater than UINT_MAX - 16 then the check does work as intended because of an integer overflow.
CVE-2025-22081	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Fix a couple integer overflows on 32bit systems On 32bit systems the "off + sizeof(struct NTFS_DE)" addition can have an integer wrapping issue. Fix it by using size_add().

CVE ID	Description
CVE-2025-22083	In the Linux kernel, the following vulnerability has been resolved: vhost-scsi: Fix handling of multiple calls to vhost_scsi_set_endpoint If vhost_scsi_set_endpoint is called multiple times without a vhost_scsi_clear_endpoint between them, we can hit multiple bugs found by Haoran Zhang: 1. Use-after-free when no tpgs are found: This fixes a use after free that occurs when vhost_scsi_set_endpoint is called more than once and calls after the first call do not find any tpgs to add to the vs_tpg. When vhost_scsi_set_endpoint first finds tpgs to add to the vs_tpg array match=true, so we will do: vhost_vq_set_backend(vq, vs_tpg); ... kfree(vs->vs_tpg); vs->vs_tpg = vs_tpg; If vhost_scsi_set_endpoint is called again and no tpgs are found match=false so we skip the vhost_vq_set_backend call leaving the pointer to the vs_tpg we then free via: kfree(vs->vs_tpg); vs->vs_tpg = vs_tpg; If a scsi request is then sent we do: vhost_scsi_handle_vq-> vhost_scsi_get_req-> vhost_vq_get_backend which sees the vs_tpg we just did a kfree on. 2. Tpg dir removal hang: This patch fixes an issue where we cannot remove a LIO/target layer tpg (and structs above it like the target) dir due to the refcount dropping to -1. The problem is that if vhost_scsi_set_endpoint detects a tpg is already in the vs->vs_tpg array or if the tpg has been removed so target_depend_item fails, the undepend goto handler will do target_undepend_item on all tpgs in the vs_tpg array dropping their refcount to 0. At this time vs_tpg contains both the tpgs we have added in the current vhost_scsi_set_endpoint call as well as tpgs we added in previous calls which are also in vs->vs_tpg. Later, when vhost_scsi_clear_endpoint runs it will do target_undepend_item on all the tpgs in the vs->vs_tpg which will drop their refcount to -1. Userspace will then not be able to remove the tpg and will hang when it tries to do rmdir on the tpg dir. 3. Tpg leak: This fixes a bug where we can leak tpgs and cause them to be un-removable because the target name is overwritten when vhost_scsi_set_endpoint is called multiple times but with different target names. The bug occurs if a user has called VHOST_SCSI_SET_ENDPOINT and setup a vhost-scsi device to target/tpg mapping, then calls VHOST_SCSI_SET_ENDPOINT again with a new target name that has tpgs we haven't seen before (target1 has tpg1 but target2 has tpg2). When this happens we don't teardown the old target tpg mapping and just overwrite the target name and the vs->vs_tpg array. Later when we do vhost_scsi_clear_endpoint, we are passed in either target1 or target2's name and we will only match that target's tpgs when we loop over the vs->vs_tpg. We will then return from the function without doing target_undepend_item on the tpgs. Because of all these bugs, it looks like being able to call vhost_scsi_set_endpoint multiple times was never supported. The major user, QEMU, already has checks to prevent this use case. So to fix the issues, this patch prevents vhost_scsi_set_endpoint from being called if it's already successfully added tpgs. To add, remove or change the tpg config or target name, you must do a vhost_scsi_clear_endpoint first.
CVE-2025-22086	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix mlx5_poll_one() cur_qp update flow When cur_qp isn't NULL, in order to avoid fetching the QP from the radix tree again we check if the next cq QP is identical to the one we already have. The bug however is that we are checking if the QP is identical by checking the QP number inside the CQE against the QP number inside the mlx5_ib_qp, but that's wrong since the QP number from the CQE is from FW so it should be matched against mlx5_core_qp which is our FW QP number. Otherwise we could use the wrong QP when handling a CQE which could cause the kernel trace below. This issue is mainly noticeable over QPs 0 & 1, since for now they are the only QPs in our driver whereas the QP number inside mlx5_ib_qp doesn't match the QP number inside mlx5_core_qp. BUG: kernel NULL pointer dereference, address: 0000000000000012 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: Oops: 0000 [#1] SMP CPU: 0 UID: 0 PID: 7927 Comm: kworker/u62:1 Not tainted 6.14.0-rc3+ #189 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 Workqueue: ib-comp-unb-wq ib_cq_poll_work [ib_core] RIP: 0010:mlx5_ib_poll_cq+0x4c7/0xd90 [mlx5_ib] Code: 03 00 00 8d 58 ff 21 cb 66 39 d3 74 39 48 c7 c7 3c 89 6e a0 0f b7 db e8 b7 d2 b3 e0 49 8b 86 60 03 00 00 48 c7 c7 4a 89 6e a0 <0f> b7 5c 98 02 e8 9f d2 b3 e0 41 0f b7 86 78 03 00 00 83 e8 01 21 RSP: 0018:ffff88810511bd60 EFLAGS: 00010046 RAX: 0000000000000010 RBX: 0000000000000000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffff8885fa1b3c0 RDI: ffffffffa06e894a RBP: 00000000000000b0 R08: 0000000000000000 R09: ffff88810511bc10 R10: 0000000000000001 R11: 0000000000000001 R12: ffff88810d593000 R13: ffff88810e579108 R14: ffff888105146000 R15: 00000000000000b0 FS: 0000000000000000(0000) GS:ffff8885fa000000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000012 CR3: 00000001077e6001 CR4: 0000000000370eb0 Call Trace: <TASK> ? __die+0x20/0x60 ? page_fault_oops+0x150/0x3e0 ? exc_page_fault+0x74/0x130 ? asm_exc_page_fault+0x22/0x30 ? mlx5_ib_poll_cq+0x4c7/0xd90 [mlx5_ib] __ib_process_cq+0x5a/0x150 [ib_core] ib_cq_poll_work+0x31/0x90 [ib_core] process_one_work+0x169/0x320 worker_thread+0x288/0x3a0 ? work_busy+0xb0/0xb0 kthread+0xd7/0x1f0 ? kthreads_online_cpu+0x130/0x130 ? kthreads_online_cpu+0x130/0x130 ret_from_fork+0x2d/0x50 ? kthreads_online_cpu+0x130/0x130 ret_from_fork_asm+0x11/0x20 </TASK>
CVE-2025-22088	In the Linux kernel, the following vulnerability has been resolved: RDMA/erdma: Prevent use-after-free in erdma_accept_newconn() After the erdma_cqp_put(new_cqp) being called, new_cqp will be freed, and the following dereference will cause a UAF problem. Fix this issue.

CVE ID	Description
CVE-2025-22089	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Don't expose hw_counters outside of init net namespace Commit 467f432a521a ("RDMA/core: Split port and device counter sysfs attributes") accidentally almost exposed hw counters to non-init net namespaces. It didn't expose them fully, as an attempt to read any of those counters leads to a crash like this one: [42021.807566] BUG: kernel NULL pointer dereference, address: 0000000000000028 [42021.814463] #PF: supervisor read access in kernel mode [42021.819549] #PF: error_code(0x0000) - not-present page [42021.824636] PGD 0 P4D 0 [42021.827145] Oops: 0000 [#1] SMP PTI [42021.830598] CPU: 82 PID: 2843922 Comm: switchto-default Kdump: loaded Tainted: G S W I XXX [42021.841697] Hardware name: XXX [42021.849619] RIP: 0010:hw_stat_device_show+0x1e/0x40 [ib_core] [42021.855362] Code: 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa 0f 1f 44 00 00 49 89 d0 4c 8b 5e 20 48 8b 8f b8 04 00 00 48 81 c7 f0 fa ff ff <48> 8b 41 28 48 29 ce 48 83 c6 d0 48 c1 ee 04 69 d6 ab aa aa aa 48 [42021.873931] RSP: 0018:ffff97fe90f03da0 EFLAGS: 00010287 [42021.879108] RAX: ffff9406988a8c60 RBX: ffff940e1072d438 RCX: 0000000000000000 [42021.886169] RDX: ffff94085f1aa000 RSI: ffff93c6cbbdbcb0 RDI: ffff940c7517aef0 [42021.893230] RBP: ffff97fe90f03e70 R08: ffff94085f1aa000 R09: 0000000000000000 [42021.900294] R10: ffff94085f1aa000 R11: ffffffff0775680 R12: ffffffff87ca2530 [42021.907355] R13: ffff940651602840 R14: ffff93c6cbbdbcb0 R15: ffff94085f1aa000 [42021.914418] FS: 00007fda1a3b9700(0000) GS:ffff94453fb80000(0000) knlGS:0000000000000000 [42021.922423] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [42021.928130] CR2: 0000000000000028 CR3: 00000042dcfb8003 CR4: 00000000003726f0 [42021.935194] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [42021.942257] DR3: 0000000000000000 DR6: 00000000ffe0ff0 DR7: 00000000000000400 [42021.949324] Call Trace: [42021.951756] <TASK> [42021.953842] [<ffffffffff86c58674>] ? show_regs+0x64/0x70 [42021.959030] [<ffffffffff86c58468>] ? __die+0x78/0xc0 [42021.963874] [<ffffffffff86c9ef75>] ? page_fault_oops+0x2b5/0x3b0 [42021.969749] [<ffffffffff87674b92>] ? exc_page_fault+0x1a2/0x3c0 [42021.975549] [<ffffffffff87801326>] ? asm_exc_page_fault+0x26/0x30 [42021.981517] [<ffffffffffc0775680>] ? __pfx_show_hw_stats+0x10/0x10 [ib_core] [42021.988482] [<ffffffffffc077564e>] ? hw_stat_device_show+0x1e/0x40 [ib_core] [42021.995438] [<ffffffffff86ac7f8e>] dev_attr_show+0x1e/0x50 [42022.000803] [<ffffffffff86a3eeb1>] sysfs_kf_seq_show+0x81/0xe0 [42022.006508] [<ffffffffff86a11134>] seq_read_iter+0xf4/0x410 [42022.011954] [<ffffffffff869f4b2e>] vfs_read+0x16e/0x2f0 [42022.017058] [<ffffffffff869f50ee>] ksys_read+0x6e/0xe0 [42022.022073] [<ffffffffff8766f1ca>] do_syscall_64+0x6a/0xa0 [42022.027441] [<ffffffffff8780013b>] entry_SYSCALL_64_after_hwframe+0x78/0xe2 The problem can be reproduced using the following steps: ip netns add foo ip netns exec foo bash cat /sys/class/infiniband/mlx4_0/hw_counters/* The panic occurs because of casting the device pointer into an ib_device pointer using container_of() in hw_stat_device_show() is wrong and leads to a memory corruption. However the real problem is that hw counters should never been exposed outside of the non-init net namespace. Fix this by saving the index of the corresponding attribute group (it might be 1 or 2 depending on the presence of driver-specific attributes) and zeroing the pointer to hw_counters group for compat devices during the initialization. With this fix applied hw_counters are not available in a non-init net namespace: find /sys/class/infiniband/mlx4_0/ -name hw_counters /sys/class/infiniband/mlx4_0/ports/1/hw_counters /sys/class/infiniband/mlx4_0/ports/2/hw_counters /sys/class/infiniband/mlx4_0/hw_counters ip netns add foo ip netns exec foo bash find /sys/class/infiniband/mlx4_0/ -name hw_counters

CVE ID	Description
CVE-2025-22090	In the Linux kernel, the following vulnerability has been resolved: x86/mm/pat: Fix VM_PAT handling when fork() fails in copy_page_range() If track_pfn_copy() fails, we already added the dst VMA to the maple tree. As fork() fails, we'll cleanup the maple tree, and stumble over the dst VMA for which we neither performed any reservation nor copied any page tables. Consequently untrack_pfn() will see VM_PAT and try obtaining the PAT information from the page table -- which fails because the page table was not copied. The easiest fix would be to simply clear the VM_PAT flag of the dst VMA if track_pfn_copy() fails. However, the whole thing is about "simply" clearing the VM_PAT flag is shaky as well: if we passed track_pfn_copy() and performed a reservation, but copying the page tables fails, we'll simply clear the VM_PAT flag, not properly undoing the reservation ... which is also wrong. So let's fix it properly: set the VM_PAT flag only if the reservation succeeded (leaving it clear initially), and undo the reservation if anything goes wrong while copying the page tables: clearing the VM_PAT flag after undoing the reservation. Note that any copied page table entries will get zapped when the VMA will get removed later, after copy_page_range() succeeded; as VM_PAT is not set then, we won't try cleaning VM_PAT up once more and untrack_pfn() will be happy. Note that leaving these page tables in place without a reservation is not a problem, as we are aborting fork(); this process will never run. A reproducer can trigger this usually at the first try: https://gitlab.com/davidhildenbrand/scratchspace/-/raw/main/reproducers/pat_fork.c WARNING: CPU: 26 PID: 11650 at arch/x86/mm/pat/memtype.c:983 get_pat_info+0xf6/0x110 Modules linked in: ... CPU: 26 UID: 0 PID: 11650 Comm: repro3 Not tainted 6.12.0-rc5+ #92 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-2.fc40 04/01/2014 RIP: 0010:get_pat_info+0xf6/0x110 ... Call Trace: <TASK> ... untrack_pfn+0x52/0x110 unmap_single_vma+0xa6/0xe0 unmap_vmas+0x105/0x1f0 exit_mmap+0xf6/0x460 __mmap+0x4b/0x120 copy_process+0x1bf6/0x2aa0 kernel_clone+0xab/0x440 __do_sys_clone+0x66/0x90 do_syscall_64+0x95/0x180 Likely this case was missed in: d155df53f310 ("x86/mm/pat: clear VM_PAT if copy_p4d_range failed") ... and instead of undoing the reservation we simply cleared the VM_PAT flag. Keep the documentation of these functions in include/linux/pgtable.h, one place is more than sufficient -- we should clean that up for the other functions like track_pfn_remap/untrack_pfn separately.
CVE-2025-22095	In the Linux kernel, the following vulnerability has been resolved: PCI: brcmstb: Fix error path after a call to regulator_bulk_get() If the regulator_bulk_get() returns an error and no regulators are created, we need to set their number to zero. If we don't do this and the PCIe link up fails, a call to the regulator_bulk_free() will result in a kernel panic. While at it, print the error value, as we cannot return an error upwards as the kernel will WARN() on an error from add_bus(). [kwilczynski: commit log, use comma in the message to match style with other similar messages]
CVE-2025-22097	In the Linux kernel, the following vulnerability has been resolved: drm/vkms: Fix use after free and double free on init error If the driver initialization fails, the vkms_exit() function might access an uninitialized or freed default_config pointer and it might double free it. Fix both possible errors by initializing default_config only when the driver initialization succeeded.
CVE-2025-22101	In the Linux kernel, the following vulnerability has been resolved: net: libwx: fix Tx L4 checksum The hardware only supports L4 checksum offload for TCP/UDP/SCTP protocol. There was a bug to set Tx checksum flag for the other protocol that results in Tx ring hang. Fix to compute software checksum for these packets.

CVE ID	Description
CVE-2025-22102	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btqpuart: Fix kernel panic during FW release This fixes a kernel panic seen during release FW in a stress test scenario where WLAN and BT FW download occurs simultaneously, and due to a HW bug, chip sends out only 1 bootloader signatures. When driver receives the bootloader signature, it enters FW download mode, but since no consecutive bootloader signatures seen, FW file is not requested. After 60 seconds, when FW download times out, release_firmware causes a kernel panic. [2601.949184] Unable to handle kernel paging request at virtual address 0000312e6f006573 [2601.992076] user pgtable: 4k pages, 48-bit VAs, pgdp=0000000111802000 [2601.992080] [0000312e6f006573] pgd=0000000000000000, p4d=0000000000000000 [2601.992087] Internal error: Oops: 0000000096000021 [#1] PREEMPT SMP [2601.992091] Modules linked in: algif_hash algif_skcipher af_alg btqpuart(O) pciexxx(O) mlan(O) overlay fsl_jr_uio caam_jr caamkeyblob_desc caamhash_desc caamalg_desc crypto_engine authenc libdes crct10dif_ce polyval_ce snd_soc_fsl_easrc snd_soc_fsl_asoc_card imx8_media_dev(C) snd_soc_fsl_micfil polyval_generic snd_soc_fsl_xcvr snd_soc_fsl_sai snd_soc_imx_audmux snd_soc_fsl_asrc snd_soc_imx_card snd_soc_imx_hdmi snd_soc_fsl_aud2htx snd_soc_fsl_utils imx_pcm_dma dw_hdmi_cec flexcan can_dev [2602.001825] CPU: 2 PID: 20060 Comm: hciconfig Tainted: G C O 6.6.23-lts-next-06236-gb586a521770e #1 [2602.010182] Hardware name: NXP i.MX8MPlus EVK board (DT) [2602.010185] pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) [2602.010191] pc : _raw_spin_lock+0x34/0x68 [2602.010201] lr : free_fw_priv+0x20/0xfc [2602.020561] sp : ffff800089363b30 [2602.020563] x29: ffff800089363b30 x28: ffff0000d0eb5880 x27: 0000000000000000 [2602.020570] x26: 0000000000000000 x25: ffff0000d728b330 x24: 0000000000000000 [2602.020577] x23: ffff0000dc856f38 [2602.033797] x22: ffff800089363b70 x21: ffff0000dc856000 [2602.033802] x20: fff0312e6f006573 x19: ffff0000d0d9ea80 x18: 0000000000000000 [2602.033809] x17: 0000000000000000 x16: 0000000000000000 x15: 0000aaaad80dd480 [2602.083320] x14: 0000000000000000 x13: 00000000000001b9 x12: 0000000000000002 [2602.083326] x11: 0000000000000000 x10: 0000000000000a60 x9 : ffff800089363a30 [2602.083333] x8 : ffff0001793d75c0 x7 : ffff0000d6dbc400 x6 : 0000000000000000 [2602.083339] x5 : 00000000410fd030 x4 : 0000000000000000 x3 : 0000000000000001 [2602.083346] x2 : 0000000000000000 x1 : 0000000000000001 x0 : fff0312e6f006573 [2602.083354] Call trace: [2602.083356] _raw_spin_lock+0x34/0x68 [2602.083364] release_firmware+0x48/0x6c [2602.083370] nxp_setup+0x3c4/0x540 [btqpuart] [2602.083383] hci_dev_open_sync+0xf0/0xa34 [2602.083391] hci_dev_open+0xd8/0x178 [2602.083399] hci_sock_ioctl+0x3b0/0x590 [2602.083405] sock_do_ioctl+0x60/0x118 [2602.083413] sock_ioctl+0x2f4/0x374 [2602.091430] __arm64_sys_ioctl+0xac/0xf0 [2602.091437] invoke_syscall+0x48/0x110 [2602.091445] el0_svc_common.constprop.0+0xc0/0xe0 [2602.091452] do_el0_svc+0x1c/0x28 [2602.091457] el0_svc+0x40/0xe4 [2602.091465] el0t_64_sync_handler+0x120/0x12c [2602.091470] el0t_64_sync+0x190/0x194
CVE-2025-22103	In the Linux kernel, the following vulnerability has been resolved: net: fix NULL pointer dereference in l3mdev_l3_rcv When delete l3s ipvlan: ip link del link eth0 ipvlan1 type ipvlan mode l3s This may cause a null pointer dereference: Call trace: ip_rcv_finish+0x48/0xd0 ip_rcv+0x5c/0x100 __netif_receive_skb_one_core+0x64/0xb0 __netif_receive_skb+0x20/0x80 process_backlog+0xb4/0x204 napi_poll+0xe8/0x294 net_rx_action+0xd8/0x22c __do_softirq+0x12c/0x354 This is because l3mdev_l3_rcv() visit dev->l3mdev_ops after ipvlan_l3s_unregister() assign the dev->l3mdev_ops to NULL. The process like this: (CPU1) (CPU2) l3mdev_l3_rcv() check dev->priv_flags: master = skb->dev; ipvlan_l3s_unregister() set dev->priv_flags dev->l3mdev_ops = NULL; visit master->l3mdev_ops To avoid this by do not set dev->l3mdev_ops when unregister l3s ipvlan.
CVE-2025-22104	In the Linux kernel, the following vulnerability has been resolved: ibmvnic: Use kernel helpers for hex dumps Previously, when the driver was printing hex dumps, the buffer was cast to an 8 byte long and printed using string formatters. If the buffer size was not a multiple of 8 then a read buffer overflow was possible. Therefore, create a new ibmvnic function that loops over a buffer and calls hex_dump_to_buffer instead. This patch address KASAN reports like the one below: ibmvnic 30000003 env3: Login Buffer: ibmvnic 30000003 env3: 01000000af000000 <...> ibmvnic 30000003 env3: 2e6d62692e736261 ibmvnic 30000003 env3: 65050003006d6f63 ===== BUG: KASAN: slab-out-of-bounds in ibmvnic_login+0xacc/0xffc [ibmvnic] Read of size 8 at addr c0000001331a9aa8 by task ip/17681 <...> Allocated by task 17681: <...> ibmvnic_login+0x2f0/0xffc [ibmvnic] ibmvnic_open+0x148/0x308 [ibmvnic] __dev_open+0x1ac/0x304 <...> The buggy address is located 168 bytes inside of allocated 175-byte region [c0000001331a9a00, c0000001331a9aaf] <...> ===== ibmvnic 30000003 env3: 000000000033766e

CVE ID	Description
CVE-2025-22105	In the Linux kernel, the following vulnerability has been resolved: bonding: check xdp prog when set bond mode Following operations can trigger a warning[1]: ip netns add ns1 ip netns exec ns1 ip link add bond0 type bond mode balance-rr ip netns exec ns1 ip link set dev bond0 xdp obj af_xdp_kern.o sec xdp ip netns exec ns1 ip link set bond0 type bond mode broadcast ip netns del ns1 When delete the namespace, dev_xdp_uninstall() is called to remove xdp program on bond dev, and bond_xdp_set() will check the bond mode. If bond mode is changed after attaching xdp program, the warning may occur. Some bond modes (broadcast, etc.) do not support native xdp. Set bond mode with xdp program attached is not good. Add check for xdp program when set bond mode. [1] -----[cut here]----- WARNING: CPU: 0 PID: 11 at net/core/dev.c:9912 unregister_netdevice_many_notify+0x8d9/0x930 Modules linked in: CPU: 0 UID: 0 PID: 11 Comm: kworker/u4:0 Not tainted 6.14.0-rc4 #107 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.15.0-0-g2dd4b9b3f840-prebuilt.qemu.org 04/01/2014 Workqueue: netns cleanup_net RIP: 0010:unregister_netdevice_many_notify+0x8d9/0x930 Code: 00 00 48 c7 c6 6f e3 a2 82 48 c7 c7 d0 b3 96 82 e8 9c 10 3e ... RSP: 0018:ffffc90000063d80 EFLAGS: 00000282 RAX: 00000000fffffa1 RBX: ffff888004959000 RCX: 00000000ffffdfff RDX: 0000000000000000 RSI: 00000000fffffea RDI: ffff90000063b48 RBP: ffff90000063e28 R08: ffffffff82d39b28 R09: 00000000000009fb R10: 0000000000000175 R11: ffffffff82d09b40 R12: ffff8880049598e8 R13: 0000000000000001 R14: dead000000000100 R15: ffff900000450000 FS: 0000000000000000(0000) GS:ffff888007a00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000d406b60 CR3: 000000000483e000 CR4: 00000000000006f0 Call Trace: <TASK> ? __warn+0x83/0x130 ? unregister_netdevice_many_notify+0x8d9/0x930 ? report_bug+0x18e/0x1a0 ? handle_bug+0x54/0x90 ? exc_invalid_op+0x18/0x70 ? asm_exc_invalid_op+0x1a/0x20 ? unregister_netdevice_many_notify+0x8d9/0x930 ? bond_net_exit_batch_rtnl+0x5c/0x90 cleanup_net+0x237/0x3d0 process_one_work+0x163/0x390 worker_thread+0x293/0x3b0 ? __pfx_worker_thread+0x10/0x10 kthread+0xec/0x1e0 ? __pfx_kthread+0x10/0x10 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x2f/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> ---[end trace 0000000000000000]---
CVE-2025-22106	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: unregister xdp rxq info in the reset path vmxnet3 does not unregister xdp rxq info in the vmxnet3_reset_work() code path as vmxnet3_rq_destroy() is not invoked in this code path. So, we get below message with a backtrace. Missing unregister, handled but fix driver WARNING: CPU:48 PID: 500 at net/core/xdp.c:182 __xdp_rxq_info_reg+0x93/0xf0 This patch fixes the problem by moving the unregister code of XDP from vmxnet3_rq_destroy() to vmxnet3_rq_cleanup().
CVE-2025-22107	In the Linux kernel, the following vulnerability has been resolved: net: dsa: sja1105: fix kasan out-of-bounds warning in sja1105_table_delete_entry() There are actually 2 problems: - deleting the last element doesn't require the memmove of elements [i + 1, end) over it. Actually, element i+1 is out of bounds. - The memmove itself should move size - i - 1 elements, because the last element is out of bounds. The out-of-bounds element still remains out of bounds after being accessed, so the problem is only that we touch it, not that it becomes in active use. But I suppose it can lead to issues if the out-of-bounds element is part of an unmapped page.
CVE-2025-22108	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Mask the bd_cnt field in the TX BD properly The bd_cnt field in the TX BD specifies the total number of BDs for the TX packet. The bd_cnt field has 5 bits and the maximum number supported is 32 with the value 0. CONFIG_MAX_SKB_FRAGS can be modified and the total number of SKB fragments can approach or exceed the maximum supported by the chip. Add a macro to properly mask the bd_cnt field so that the value 32 will be properly masked and set to 0 in the bd_cnd field. Without this patch, the out-of-range bd_cnt value will corrupt the TX BD and may cause TX timeout. The next patch will check for values exceeding 32.

CVE ID	Description
CVE-2025-22109	In the Linux kernel, the following vulnerability has been resolved: ax25: Remove broken autobind Binding AX25 socket by using the autobind feature leads to memory leaks in ax25_connect() and also refcount leaks in ax25_release(). Memory leak was detected with kmemleak: <pre>===== unreferenced object 0xffff8880253cd680 (size 96): backtrace: __kmalloc_node_track_caller_noprof (/include/linux/ kmemleak.h:43) kmemdup_noprof (mm/util.c:136) ax25_rt_autobind (net/ax25/ax25_route.c:428) ax25_connect (net/ax25/af_ax25.c:1282) __sys_connect_file (net/socket.c:2045) __sys_connect (net/ socket.c:2064) __x64_sys_connect (net/socket.c:2067) do_syscall_64 (arch/x86/entry/common.c:52 arch/x86/entry/common.c:83) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) =====</pre> When socket is bound, refcounts must be incremented the way it is done in ax25_bind() and ax25_setsockopt() (SO_BINDTODEVICE). In case of autobind, the refcounts are not incremented. This bug leads to the following issue reported by Syzkaller: <pre>===== ax25_connect(): syz-executor318 uses autobind, please contact jreuter@yaina.de -----[cut here]----- refcount_t: decrement hit 0; leaking memory. WARNING: CPU: 0 PID: 5317 at lib/refcount.c:31 refcount_warn_saturate+0xfa/0x1d0 lib/refcount.c:31 Modules linked in: CPU: 0 UID: 0 PID: 5317 Comm: syz-executor318 Not tainted 6.14.0-rc4-syzkaller-00278-gece144f151ac #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:refcount_warn_saturate+0xfa/0x1d0 lib/refcount.c:31 ... Call Trace: <TASK> __refcount_dec include/linux/refcount.h:336 [inline] refcount_dec include/linux/refcount.h:351 [inline] ref_tracker_free+0x6af/0x7e0 lib/ref_tracker.c:236 netdev_tracker_free include/linux/ netdevice.h:4302 [inline] netdev_put include/linux/netdevice.h:4319 [inline] ax25_release +0x368/0x960 net/ax25/af_ax25.c:1080 __sock_release net/socket.c:647 [inline] sock_close +0xbc/0x240 net/socket.c:1398 __fput+0x3e9/0x9f0 fs/file_table.c:464 __do_sys_close fs/ open.c:1580 [inline] __se_sys_close fs/open.c:1565 [inline] __x64_sys_close+0x7f/0x110 fs/ open.c:1565 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... </TASK> =====</pre> Considering the issues above and the comments left in the code that say: "check if we can remove this feature. It is broken."; "autobinding in this may or may not work"; - it is better to completely remove this feature than to fix it because it is broken and leads to various kinds of memory bugs. Now calling connect() without first binding socket will result in an error (-EINVAL). Userspace software that relies on the autobind feature might get broken. However, this feature does not seem widely used with this specific driver as it was not reliable at any point of time, and it is already broken anyway. E.g. ax25-tools and ax25-apps packages for popular distributions do not use the autobind feature for AF_AX25. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.

CVE ID	Description
CVE-2025-22111	In the Linux kernel, the following vulnerability has been resolved: net: Remove RTNL dance for SIOCBRADDIF and SIOCBRDELIF. SIOCBRDELIF is passed to dev_ioctl() first and later forwarded to br_ioctl_call(), which causes unnecessary RTNL dance and the splat below [0] under RTNL pressure. Let's say Thread A is trying to detach a device from a bridge and Thread B is trying to remove the bridge. In dev_ioctl(), Thread A bumps the bridge device's refcnt by netdev_hold() and releases RTNL because the following br_ioctl_call() also re-acquires RTNL. In the race window, Thread B could acquire RTNL and try to remove the bridge device. Then, rtnl_unlock() by Thread B will release RTNL and wait for netdev_put() by Thread A. Thread A, however, must hold RTNL after the unlock in dev_ifsioc(), which may take long under RTNL pressure, resulting in the splat by Thread B. Thread A (SIOCBRDELIF) Thread B (SIOCBRDELBR) ----- sock_ioctl sock_ioctl `~ sock_do_ioctl `~ br_ioctl_call `~ dev_ioctl `~ br_ioctl_stub - rtnl_lock - dev_ifsioc `~  - dev = __dev_get_by_name(...)  - netdev_hold(dev, ...) . /  - rtnl_unlock -----.     - br_ioctl_call `~> - rtnl_lock Race `~ br_ioctl_stub  - br_del_bridge Window     - dev = __dev_get_by_name(...)    May take long   `~ br_dev_delete(dev, ...) under RTNL pressure `~ unregister_netdevice_queue(dev, ...)      `~ rtnl_unlock \ - rtnl_lock < `~ `~ netdev_run_todo ... `~ netdev_run_todo   `~ rtnl_unlock -__rtnl_unlock - netdev_wait_allrefs_any - netdev_put(dev, ...) <----- Wait refcnt decrement and log splat below To avoid blocking SIOCBRDELBR unnecessarily, let's not call dev_ioctl() for SIOCBRADDIF and SIOCBRDELIF. In the dev_ioctl() path, we do the following: 1. Copy struct ifreq by get_user_ifreq in sock_do_ioctl() 2. Check CAP_NET_ADMIN in dev_ioctl() 3. Call dev_load() in dev_ioctl() 4. Fetch the master dev from ifr.ifr_name in dev_ifsioc() 3. can be done by request_module() in br_ioctl_call(), so we move 1., 2., and 4. to br_ioctl_stub(). Note that 2. is also checked later in add_del_if(), but it's better performed before RTNL. SIOCBRADDIF and SIOCBRDELIF have been processed in dev_ioctl() since the pre-git era, and there seems to be no specific reason to process them there. [0]: unregister_netdevice: waiting for wpan3 to become free. Usage count = 2 ref_tracker: wpan3@ffff880662d8608 has 1/1 users at __netdev_tracker_alloc include/linux/netdevice.h:4282 [inline] netdev_hold include/linux/netdevice.h:4311 [inline] dev_ifsioc+0xc6a/0x1160 net/core/dev_ioctl.c:624 dev_ioctl+0x255/0x10c0 net/core/dev_ioctl.c:826 sock_do_ioctl+0x1ca/0x260 net/socket.c:1213 sock_ioctl+0x23a/0x6c0 net/socket.c:1318 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl fs/ioctl.c:892 [inline] __x64_sys_ioctl+0x1a4/0x210 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcb/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2025-22113	In the Linux kernel, the following vulnerability has been resolved: ext4: avoid journaling sb update on error if journal is destroying Presently we always BUG_ON if trying to start a transaction on a journal marked with JBD2_UNMOUNT, since this should never happen. However, while ltp running stress tests, it was observed that in case of some error handling paths, it is possible for update_super_work to start a transaction after the journal is destroyed eg: (umount) ext4_kill_sb kill_block_super generic_shutdown_super sync_filesystem /* commits all txns */ evict_inodes /* might start a new txn */ ext4_put_super flush_work(&sbi->s_sb_upd_work) /* flush the workqueue */ jbd2_journal_destroy journal_kill_thread journal->j_flags = JBD2_UNMOUNT; jbd2_journal_commit_transaction jbd2_journal_get_descriptor_buffer jbd2_journal_bmap ext4_journal_bmap ext4_map_blocks ... ext4_inode_error ext4_handle_error schedule_work(&sbi->s_sb_upd_work) /* work queue kicks in */ update_super_work jbd2_journal_start start_this_handle BUG_ON(journal->j_flags & JBD2_UNMOUNT) Hence, introduce a new mount flag to indicate journal is destroying and only do a journaled (and deferred) update of sb if this flag is not set. Otherwise, just fallback to an un-journaled commit. Further, in the journal destroy path, we have the following sequence: 1. Set mount flag indicating journal is destroying 2. force a commit and wait for it 3. flush pending sb updates This sequence is important as it ensures that, after this point, there is no sb update that might be journaled so it is safe to update the sb outside the journal. (To avoid race discussed in 2d01ddc86606) Also, we don't need a similar check in ext4_grp_locked_error since it is only called from mballo and AFAICT it would be always valid to schedule work here.

CVE ID	Description
CVE-2025-22115	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix block group refcount race in btrfs_create_pending_block_groups() Block group creation is done in two phases, which results in a slightly unintuitive property: a block group can be allocated/deallocated from after btrfs_make_block_group() adds it to the space_info with btrfs_add_bg_to_space_info(), but before creation is completely completed in btrfs_create_pending_block_groups(). As a result, it is possible for a block group to go unused and have 'btrfs_mark_bg_unused' called on it concurrently with 'btrfs_create_pending_block_groups'. This causes a number of issues, which were fixed with the block group flag 'BLOCK_GROUP_FLAG_NEW'. However, this fix is not quite complete. Since it does not use the unused_bg_lock, it is possible for the following race to occur: btrfs_create_pending_block_groups btrfs_mark_bg_unused if list_empty // false list_del_init clear_bit else if (test_bit) // true list_move_tail And we get into the exact same broken ref count and invalid new_bgs state for transaction cleanup that BLOCK_GROUP_FLAG_NEW was designed to prevent. The broken refcount aspect will result in a warning like: [1272.943527] refcount_t: underflow; use-after-free. [1272.943967] WARNING: CPU: 1 PID: 61 at lib/refcount.c:28 refcount_warn_saturate+0xba/0x110 [1272.944731] Modules linked in: btrfs virtio_net xor zstd_compress raid6_pq null_blk [last unloaded: btrfs] [1272.945550] CPU: 1 UID: 0 PID: 61 Comm: kworker/u32:1 Kdump: loaded Tainted: G W 6.14.0-rc5+ #108 [1272.946368] Tainted: [W]=WARN [1272.946585] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS Arch Linux 1.16.3-1-04/01/2014 [1272.947273] Workqueue: btrfs_discard btrfs_discard_workfn [btrfs] [1272.947788] RIP: 0010:refcount_warn_saturate+0xba/0x110 [1272.949532] RSP: 0018:ffffbf1200247df0 EFLAGS: 00010282 [1272.949901] RAX: 0000000000000000 RBX: fffff14b00dc2860 RCX: 0000000000000000 [1272.950437] RDX: 0000000000000000 RSI: fffff1200247c78 RDI: 00000000ffffdfff [1272.950986] RBP: fffff14b00dc2860 R08: 00000000ffffdfff R09: ffffffff90526268 [1272.951512] R10: ffffffff904762c0 R11: 0000000063666572 R12: fffff14b00dc28c0 [1272.952024] R13: 0000000000000000 R14: fffff14b00dc2868 R15: 000001285dcd12c0 [1272.952850] FS: 0000000000000000(0000) GS:ffffa14d33c40000(0000) knlGS:0000000000000000 [1272.953458] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [1272.953931] CR2: 00007f838cbda000 CR3: 000000010104e000 CR4: 000000000000006f0 [1272.954474] Call Trace: [1272.954655] <TASK> [1272.954812] ? refcount_warn_saturate+0xba/0x110 [1272.955173] ? __warn.cold+0x93/0xd7 [1272.955487] ? refcount_warn_saturate+0xba/0x110 [1272.955816] ? report_bug+0xe7/0x120 [1272.956103] ? handle_bug+0x53/0x90 [1272.956424] ? exc_invalid_op+0x13/0x60 [1272.956700] ? asm_exc_invalid_op+0x16/0x20 [1272.957011] ? refcount_warn_saturate+0xba/0x110 [1272.957399] btrfs_discard_cancel_work.cold+0x26/0x2b [btrfs] [1272.957853] btrfs_put_block_group.cold+0x5d/0x8e [btrfs] [1272.958289] btrfs_discard_workfn+0x194/0x380 [btrfs] [1272.958729] process_one_work+0x130/0x290 [1272.959026] worker_thread+0x2ea/0x420 [1272.959335] ? __pfx_worker_thread+0x10/0x10 [1272.959644] kthread+0xd7/0x1c0 [1272.959872] ? __pfx_kthread+0x10/0x10 [1272.960172] ret_from_fork+0x30/0x50 [1272.960474] ? __pfx_kthread+0x10/0x10 [1272.960745] ret_from_fork_asm+0x1a/0x30 [1272.961035] </TASK> [1272.961238] ---[end trace 0000000000000000]--- Though we have seen them in the async discard workfn as well. It is most likely to happen after a relocation finishes which cancels discard, tears down the block group, etc. Fix this fully by taking the lock aroun ---truncated---
CVE-2025-22116	In the Linux kernel, the following vulnerability has been resolved: idpf: check error for register_netdev() on init Current init logic ignores the error code from register_netdev(), which will cause WARN_ON() on attempt to unregister it, if there was one, and there is no info for the user that the creation of the netdev failed. WARNING: CPU: 89 PID: 6902 at net/core/dev.c:11512 unregister_netdevice_many_notify+0x211/0x1a10 ... [3707.563641] unregister_netdev+0x1c/0x30 [3707.563656] idpf_vport_dealloc+0x5cf/0xce0 [idpf] [3707.563684] idpf_deinit_task+0xef/0x160 [idpf] [3707.563712] idpf_vc_core_deinit+0x84/0x320 [idpf] [3707.563739] idpf_remove+0xbf/0x780 [idpf] [3707.563769] pci_device_remove+0xab/0x1e0 [3707.563786] device_release_driver_internal+0x371/0x530 [3707.563803] driver_detach+0xbf/0x180 [3707.563816] bus_remove_driver+0x11b/0x2a0 [3707.563829] pci_unregister_driver+0x2a/0x250 Introduce an error check and log the vport number and error code. On removal make sure to check VPORT_REG_NETDEV flag prior to calling unregister and free on the netdev. Add local variables for idx, vport_config and netdev for readability.
CVE-2025-22120	In the Linux kernel, the following vulnerability has been resolved: ext4: goto right label 'out_mmap_sem' in ext4_setattr() Otherwise, if ext4_inode_attach_jinode() fails, a hung task will happen because filemap_invalidate_unlock() isn't called to unlock mapping->invalidate_lock. Like this: EXT4-fs error (device sda) in ext4_setattr:5557: Out of memory INFO: task fsstress:374 blocked for more than 122 seconds. Not tainted 6.14.0-rc1-next-20250206-xfstests-dirty #726 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:fsstress state:D stack:0 pid:374 tgid:374 ppid:373 task_flags:0x440140 flags:0x00000000 Call Trace: <TASK> __schedule+0x2c9/0x7f0 schedule+0x27/0xa0 schedule_preempt_disabled+0x15/0x30 rwsem_down_read_slowpath+0x278/0x4c0 down_read+0x59/0xb0 page_cache_ra_unbounded+0x65/0x1b0 filemap_get_pages+0x124/0x3e0 filemap_read+0x114/0x3d0 vfs_read+0x297/0x360 ksys_read+0x6c/0xe0 do_syscall_64+0x4b/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e

CVE ID	Description
CVE-2025-22121	In the Linux kernel, the following vulnerability has been resolved: ext4: fix out-of-bound read in ext4_xattr_inode_dec_ref_all() There's issue as follows: BUG: KASAN: use-after-free in ext4_xattr_inode_dec_ref_all+0x6ff/0x790 Read of size 4 at addr ffff88807b003000 by task syz-executor.0/15172 CPU: 3 PID: 15172 Comm: syz-executor.0 Call Trace: __dump_stack lib/dump_stack.c:82 [inline] dump_stack+0xbe/0xfd lib/dump_stack.c:123 print_address_description.constprop.0+0x1e/0x280 mm/kasan/report.c:400 __kasan_report.cold+0x6c/0x84 mm/kasan/report.c:560 kasan_report+0x3a/0x50 mm/kasan/report.c:585 ext4_xattr_inode_dec_ref_all+0x6ff/0x790 fs/ext4/xattr.c:1137 ext4_xattr_delete_inode+0x4c7/0xda0 fs/ext4/xattr.c:2896 ext4_evict_inode+0xb3b/0x1670 fs/ext4/inode.c:323 evict+0x39f/0x880 fs/inode.c:622 iput_final fs/inode.c:1746 [inline] iput fs/inode.c:1772 [inline] iput+0x525/0x6c0 fs/inode.c:1758 ext4_orphan_cleanup fs/ext4/super.c:3298 [inline] ext4_fill_super+0x8c57/0xba40 fs/ext4/super.c:5300 mount_bdev+0x355/0x410 fs/super.c:1446 legacy_get_tree+0xfe/0x220 fs/fs_context.c:611 vfs_get_tree+0x8d/0x2f0 fs/super.c:1576 do_new_mount fs/namespace.c:2983 [inline] path_mount+0x119a/0x1ad0 fs/namespace.c:3316 do_mount+0xfc/0x110 fs/namespace.c:3329 __do_sys_mount fs/namespace.c:3540 [inline] __se_sys_mount+0x219/0x2e0 fs/namespace.c:3514 do_syscall_64+0x33/0x40 arch/x86/entry/common.c:46 entry_SYSCALL_64_after_hwframe+0x67/0xd1 Memory state around the buggy address: ffff88807b002f00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ffff88807b002f80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 >ffff88807b003000: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ffff88807b003080: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ffff88807b003100: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff Above issue happens as ext4_xattr_delete_inode() isn't check xattr is valid if xattr is in inode. To solve above issue call xattr_check_inode() check if xattr if valid in inode. In fact, we can directly verify in ext4_iget_extra_inode(), so that there is no divergent verification.
CVE-2025-22124	In the Linux kernel, the following vulnerability has been resolved: md/md-bitmap: fix wrong bitmap_limit for clustermd when write sb In clustermd, separate write-intent-bitmaps are used for each cluster node: 0 4k 8k 12k ----- idle md super bm super [0] + bits bm bits[0, contd] bm super[1] + bits bm bits[1, contd] bm super[2] + bits bm bits [2, contd] bm super[3] + bits bm bits [3, contd] So in node 1, pg_index in __write_sb_page() could equal to bitmap->storage.file_pages. Then bitmap_limit will be calculated to 0. md_super_write() will be called with 0 size. That means the first 4k sb area of node 1 will never be updated through filemap_write_page(). This bug causes hang of mdadm/clustermd_tests/01r1_Grow_resize. Here use (pg_index % bitmap->storage.file_pages) to make calculation of bitmap_limit correct.
CVE-2025-22125	In the Linux kernel, the following vulnerability has been resolved: md/raid1,raid10: don't ignore IO flags If blk-wbt is enabled by default, it's found that raid write performance is quite bad because all IO are throttled by wbt of underlying disks, due to flag REQ_IDLE is ignored. And turns out this behaviour exist since blk-wbt is introduced. Other than REQ_IDLE, other flags should not be ignored as well, for example REQ_META can be set for filesystems, clearing it can cause priority reverse problems; And REQ_NOWAIT should not be cleared as well, because io will wait instead of failing directly in underlying disks. Fix those problems by keep IO flags from master bio. Fises: f51d46d0e7cb ("md: add support for REQ_NOWAIT")
CVE-2025-22126	In the Linux kernel, the following vulnerability has been resolved: md: fix mddev uaf while iterating all_mddevs list While iterating all_mddevs list from md_notify_reboot() and md_exit(), list_for_each_entry_safe is used, and this can race with deletint the next mddev, causing UAF: t1: spin_lock //list_for_each_entry_safe(mddev, n, ...) mddev_get(mddev1) // assume mddev2 is the next entry spin_unlock t2: //remove mddev2 ... mddev_free spin_lock list_del spin_unlock kfree(mddev2) mddev_put(mddev1) spin_lock //continue dereference mddev2->all_mddevs The old helper for_each_mddev() actually grab the reference of mddev2 while holding the lock, to prevent from being freed. This problem can be fixed the same way, however, the code will be complex. Hence switch to use list_for_each_entry, in this case mddev_put() can free the mddev1 and it's not safe as well. Refer to md_seq_show(), also factor out a helper mddev_put_locked() to fix this problem.

CVE ID	Description
CVE-2025-22127	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix potential deadlock in prepare_compress_overwrite() Jan Prusakowski reported a kernel hang issue as below: When running xfstests on linux-next kernel (6.14.0-rc3, 6.12) I encountered a problem in generic/475 test where f2fsstress process gets blocked in __f2fs_write_data_pages() and the test hangs. The options I used are: MKFS_OPTIONS -- -O compression -O extra_attr -O project_quota -O quota /dev/vdc MOUNT_OPTIONS -- -o acl,user_xattr -o discard,compress_extension=* /dev/vdc /vdc INFO: task kworker/u8:0:11 blocked for more than 122 seconds. Not tainted 6.14.0-rc3-xfstests-lockdep #1 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:kworker/u8:0 state:D stack:0 pid:11 tgid:11 ppid:2 task_flags:0x4208160 flags:0x00004000 Workqueue: writeback wb_workfn (flush-253:0) Call Trace: <TASK> __schedule+0x309/0x8e0 schedule+0x3a/0x100 schedule_preempt_disabled+0x15/0x30 __mutex_lock+0x59a/0xdb0 __f2fs_write_data_pages+0x3ac/0x400 do_writepages+0xe8/0x290 __writeback_single_inode+0x5c/0x360 writeback_sb_inodes+0x22f/0x570 wb_writeback+0xb0/0x410 wb_do_writeback+0x47/0x2f0 wb_workfn+0x5a/0x1c0 process_one_work+0x223/0x5b0 worker_thread+0x1d5/0x3c0 kthread+0xf0/0x230 ret_from_fork+0x31/0x50 ret_from_fork_asm+0x1a/0x30 </TASK> The root cause is: once generic/475 starts to load error table to dm device, f2fs_prepare_compress_overwrite() will loop reading compressed cluster pages due to IO error, meanwhile it has held .writepages lock, it can block all other writeback tasks. Let's fix this issue w/ below changes: - add f2fs_handle_page_eio() in prepare_compress_overwrite() to detect IO error. - detect cp_error earlier in f2fs_read_multi_pages().
CVE-2025-22128	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Clear affinity hint before calling ath12k_pci_free_irq() in error path If a shared IRQ is used by the driver due to platform limitation, then the IRQ affinity hint is set right after the allocation of IRQ vectors in ath12k_pci_msi_alloc(). This does no harm unless one of the functions requesting the IRQ fails and attempt to free the IRQ. This may end up with a warning from the IRQ core that is expecting the affinity hint to be cleared before freeing the IRQ: kernel/irq/manage.c: /* make sure affinity_hint is cleaned up */ if (WARN_ON_ONCE(desc->affinity_hint)) desc->affinity_hint = NULL; So to fix this issue, clear the IRQ affinity hint before calling ath12k_pci_free_irq() in the error path. The affinity will be cleared once again further down the error path due to code organization, but that does no harm.
CVE-2025-22134	When switching to other buffers using the :all command and visual mode still being active, this may cause a heap-buffer overflow, because Vim does not properly end visual mode and therefore may try to access beyond the end of a line in a buffer. In Patch 9.1.1003 Vim will correctly reset the visual mode before opening other windows and buffers and therefore fix this bug. In addition it does verify that it won't try to access a position if the position is greater than the corresponding buffer line. Impact is medium since the user must have switched on visual mode when executing the :all ex command. The Vim project would like to thank github user gandalf4a for reporting this issue. The issue has been fixed as of Vim patch v9.1.1003
CVE-2025-22228	BCryptPasswordEncoder.matches(CharSequence,String) will incorrectly return true for passwords larger than 72 characters as long as the first 72 characters are the same.
CVE-2025-22233	CVE-2024-38820 ensured Locale-independent, lowercase conversion for both the configured disallowedFields patterns and for request parameter names. However, there are still cases where it is possible to bypass the disallowedFields checks. Affected Spring Products and Versions Spring Framework: * 6.2.0 - 6.2.6 * 6.1.0 - 6.1.19 * 6.0.0 - 6.0.27 * 5.3.0 - 5.3.42 * Older, unsupported versions are also affected Mitigation Users of affected versions should upgrade to the corresponding fixed version. Affected version(s) Fix Version Availability 6.2.x 6.2.7 OSS 6.1.x 6.1.20 OSS 6.0.x 6.0.28 Commercial https://enterprise.spring.io/ 5.3.x 5.3.43 Commercial https://enterprise.spring.io/ No further mitigation steps are necessary. Generally, we recommend using a dedicated model object with properties only for data binding, or using constructor binding since constructor arguments explicitly declare what to bind together with turning off setter binding through the declarativeBinding flag. See the Model Design section in the reference documentation. For setting binding, prefer the use of allowedFields (an explicit list) over disallowedFields. Credit This issue was responsibly reported by the TERASOLUNA Framework Development Team from NTT DATA Group Corporation.
CVE-2025-22235	EndpointRequest.to() creates a matcher for null/** if the actuator endpoint, for which the EndpointRequest has been created, is disabled or not exposed. Your application may be affected by this if all the following conditions are met: * You use Spring Security * EndpointRequest.to() has been used in a Spring Security chain configuration * The endpoint which EndpointRequest references is disabled or not exposed via web * Your application handles requests to /null and this path needs protection You are not affected if any of the following is true: * You don't use Spring Security * You don't use EndpointRequest.to() * The endpoint which EndpointRequest.to() refers to is enabled and is exposed * Your application does not handle requests to /null or this path does not need protection
CVE-2025-22866	Due to the usage of a variable time instruction in the assembly implementation of an internal function, a small number of bits of secret scalars are leaked on the ppc64le architecture. Due to the way this function is used, we do not believe this leakage is enough to allow recovery of the private key when P-256 is used in any well known protocols.

CVE ID	Description
CVE-2025-22868	An attacker can pass a malicious malformed token which causes unexpected memory to be consumed during parsing.
CVE-2025-22869	SSH servers which implement file transfer protocols are vulnerable to a denial of service attack from clients which complete the key exchange slowly, or not at all, causing pending content to be read into memory, but never transmitted.
CVE-2025-22870	Matching of hosts against proxy patterns can improperly treat an IPv6 zone ID as a hostname component. For example, when the NO_PROXY environment variable is set to "*.example.com", a request to "[::1%25.example.com]:80" will incorrectly match and not be proxied.
CVE-2025-22871	The net/http package improperly accepts a bare LF as a line terminator in chunked data chunk-size lines. This can permit request smuggling if a net/http server is used in conjunction with a server that incorrectly accepts a bare LF as part of a chunk-ext.
CVE-2025-22872	The tokenizer incorrectly interprets tags with unquoted attribute values that end with a solidus character (/) as self-closing. When directly using Tokenizer, this can result in such tags incorrectly being marked as self-closing, and when using the Parse functions, this can result in content following such tags as being placed in the wrong scope during DOM construction, but only when tags are in foreign content (e.g. <math>, <svg>, etc contexts).
CVE-2025-22874	Calling Verify with a VerifyOptions.KeyUsages that contains ExtKeyUsageAny unintentionally disabled policy validation. This only affected certificate chains which contain policy graphs, which are rather uncommon.
CVE-2025-22919	A reachable assertion in FFmpeg git-master commit N-113007-g8d24a28d06 allows attackers to cause a Denial of Service (DoS) via opening a crafted AAC file.
CVE-2025-22921	FFmpeg git-master, N-113007-g8d24a28d06 was discovered to contain a segmentation violation via the component /libavcodec/jpeg2000dec.c.
CVE-2025-2312	A flaw was found in cifs-utils. When trying to obtain Kerberos credentials, the cifs.upcall program from the cifs-utils package makes an upcall to the wrong namespace in containerized environments. This issue may lead to disclosing sensitive data from the host's Kerberos credentials cache.
CVE-2025-23129	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: Clear affinity hint before calling ath11k_pcie_free_irq() in error path If a shared IRQ is used by the driver due to platform limitation, then the IRQ affinity hint is set right after the allocation of IRQ vectors in ath11k_pcie_alloc_msi(). This does no harm unless one of the functions requesting the IRQ fails and attempt to free the IRQ. This results in the below warning: WARNING: CPU: 7 PID: 349 at kernel/irq/manage.c:1929 free_irq+0x278/0x29c Call trace: free_irq+0x278/0x29c ath11k_pcie_free_irq+0x70/0x10c [ath11k] ath11k_pcie_probe+0x800/0x820 [ath11k_pcie] local_pcie_probe+0x40/0x40 The warning is due to not clearing the affinity hint before freeing the IRQs. So to fix this issue, clear the IRQ affinity hint before calling ath11k_pcie_free_irq() in the error path. The affinity will be cleared once again further down the error path due to code organization, but that does no harm. Tested-on: QCA6390 hw2.0 PCI WLAN.HST.1.0.1-05266-QCAHSTSWPLZ_V2_TO_X86-1
CVE-2025-23130	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid panic once fallocation fails for pinfile syzbot reports a f2fs bug as below: -----[cut here]----- kernel BUG at fs/f2fs/segment.c:2746! CPU: 0 UID: 0 PID: 5323 Comm: syz.0.0 Not tainted 6.13.0-rc2-syzkaller-00018-g7cb1b4663150 #0 RIP: 0010:get_new_segment fs/f2fs/segment.c:2746 [inline] RIP: 0010:new_curseg+0x1f52/0x1f70 fs/f2fs/segment.c:2876 Call Trace: <TASK> __allocate_new_segment+0x1ce/0x940 fs/f2fs/segment.c:3210 f2fs_allocate_new_section fs/f2fs/segment.c:3224 [inline] f2fs_allocate_pinning_section+0xfa/0x4e0 fs/f2fs/segment.c:3238 f2fs_expand_inode_data+0x696/0xca0 fs/f2fs/file.c:1830 f2fs_fallocate+0x537/0xa10 fs/f2fs/file.c:1940 vfs_fallocate+0x569/0x6e0 fs/open.c:327 do_vfs_ioctl+0x258c/0x2e40 fs/ioctl.c:885 __do_sys_ioctl fs/ioctl.c:904 [inline] __se_sys_ioctl+0x80/0x170 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Concurrent pinfile allocation may run out of free section, result in panic in get_new_segment(), let's expand pin_sem lock coverage to include f2fs_gc(), so that we can make sure to reclaim enough free space for following allocation. In addition, do below changes to enhance error path handling: - call f2fs_bug_on() only in non-pinfile allocation path in get_new_segment(). - call reset_curseg_fields() to reset all fields of curseg in new_curseg()

CVE ID	Description
CVE-2025-23131	In the Linux kernel, the following vulnerability has been resolved: dlm: prevent NPD when writing a positive value to event_done do_uevent returns the value written to event_done. In case it is a positive value, new_lockspace would undo all the work, and lockspace would not be set. __dlm_new_lockspace, however, would treat that positive value as a success due to commit 8511a2728ab8 ("dlm: fix use count with multiple joins"). Down the line, device_create_lockspace would pass that NULL lockspace to dlm_find_lockspace_local, leading to a NULL pointer dereference. Treating such positive values as successes prevents the problem. Given this has been broken for so long, this is unlikely to break userspace expectations.
CVE-2025-23132	In the Linux kernel, the following vulnerability has been resolved: f2fs: quota: fix to avoid warning in dquot_writeback_dquots() F2FS-fs (dm-59): checkpoint=enable has some unwritten data. ----- [cut here]----- WARNING: CPU: 6 PID: 8013 at fs/quota/dquot.c:691 dquot_writeback_dquots+0x2fc/0x308 pc : dquot_writeback_dquots+0x2fc/0x308 lr : f2fs_quota_sync+0xcc/0x1c4 Call trace: dquot_writeback_dquots+0x2fc/0x308 f2fs_quota_sync+0xcc/0x1c4 f2fs_write_checkpoint+0x3d4/0x9b0 f2fs_issue_checkpoint+0x1bc/0x2c0 f2fs_sync_fs+0x54/0x150 f2fs_do_sync_file+0x2f8/0x814 __f2fs_ioclt+0x1960/0x3244 f2fs_ioclt+0x54/0xe0 __arm64_sys_ioclt+0xa8/0xe4 invoke_syscall+0x58/0x114 checkpoint and f2fs_remount may race as below, resulting triggering warning in dquot_writeback_dquots(). atomic write remount - do_remount - down_write(&sb->s_umount); - f2fs_remount - ioclt - f2fs_do_sync_file - f2fs_sync_fs - f2fs_write_checkpoint - block_operations - locked = down_read_trylock(&sbi->sb->s_umount) : fail to lock due to the write lock was held by remount - up_write(&sb->s_umount); - f2fs_quota_sync - dquot_writeback_dquots - WARN_ON_ONCE(!rwsem_is_locked(&sb->s_umount)) : trigger warning because s_umount lock was unlocked by remount If checkpoint comes from mount/umount/remount/freeze/quotactl, caller of checkpoint has already held s_umount lock, calling dquot_writeback_dquots() in the context should be safe. So let's record task to sbi->umount_lock_holder, so that checkpoint can know whether the lock has held in the context or not by checking current w/ it. In addition, in order to not misrepresent caller of checkpoint, we should not allow to trigger async checkpoint for those callers: mount/umount/remount/freeze/quotactl.
CVE-2025-23133	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: update channel list in reg_notifier instead reg_worker Currently when ath11k gets a new channel list, it will be processed according to the following steps: 1. update new channel list to cfg80211 and queue reg_work. 2. cfg80211 handles new channel list during reg_work. 3. update cfg80211's handled channel list to firmware by ath11k_reg_update_chan_list(). But ath11k will immediately execute step 3 after reg_work is just queued. Since step 2 is asynchronous, cfg80211 may not have completed handling the new channel list, which may leading to an out-of-bounds write error: BUG: KASAN: slab-out-of-bounds in ath11k_reg_update_chan_list Call Trace: ath11k_reg_update_chan_list+0xbfe/0xfe0 [ath11k] kfree+0x109/0x3a0 ath11k_regd_update+0x1cf/0x350 [ath11k] ath11k_regd_update_work+0x14/0x20 [ath11k] process_one_work+0xe35/0x14c0 Should ensure step 2 is completely done before executing step 3. Thus Wen raised patch[1]. When flag NL80211_REGDOM_SET_BY_DRIVER is set, cfg80211 will notify ath11k after step 2 is done. So enable the flag NL80211_REGDOM_SET_BY_DRIVER then cfg80211 will notify ath11k after step 2 is done. At this time, there will be no KASAN bug during the execution of the step 3. [1] https://patchwork.kernel.org/project/linux-wireless/patch/20230201065313.27203-1-quic_wgong@quicinc.com/ Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3

CVE ID	Description
CVE-2025-23135	In the Linux kernel, the following vulnerability has been resolved: RISC-V: KVM: Teardown riscv specific bits after kvm_exit During a module removal, kvm_exit invokes arch specific disable call which disables AIA. However, we invoke aia_exit before kvm_exit resulting in the following warning. KVM kernel module can't be inserted afterwards due to inconsistent state of IRQ. [25469.031389] percpu IRQ 31 still enabled on CPU0! [25469.031732] WARNING: CPU: 3 PID: 943 at kernel/irq/manage.c:2476 __free_percpu_irq+0xa2/0x150 [25469.031804] Modules linked in: kvm(-) [25469.031848] CPU: 3 UID: 0 PID: 943 Comm: rmmmod Not tainted 6.14.0-rc5-06947-g91c763118f47-dirty #2 [25469.031905] Hardware name: riscv-virtio,qemu (DT) [25469.031928] epc : __free_percpu_irq+0xa2/0x150 [25469.031976] ra : __free_percpu_irq+0xa2/0x150 [25469.032197] epc : ffffffff8007db1e ra : ffffffff8007db1e sp : ff2000000088bd50 [25469.032241] gp : ffffffff8131cef8 tp : ff60000080b96400 t0 : ff2000000088baf8 [25469.032285] t1 : ffffffff80c351c0 s2 : 5249207570637265 s0 : ff2000000088bd90 [25469.032329] s1 : ff60000098b21080 a0 : 037d527a15eb4f00 a1 : 037d527a15eb4f00 [25469.032372] a2 : 0000000000000023 a3 : 0000000000000001 a4 : ffffffff8122dbf8 [25469.032410] a5 : 00000000000000ff a6 : 0000000000000000 a7 : ffffffff8122dc10 [25469.032448] s2 : ff60000080c22eb0 s3 : 0000000200000022 s4 : 000000000000001f [25469.032488] s5 : ff60000080c22e00 s6 : ffffffff80c351c0 s7 : 0000000000000000 [25469.032582] s8 : 0000000000000003 s9 : 000055556b7fb490 s10: 00007ffff0e12fa0 [25469.032621] s11: 00007ffff0e13e9a t3 : ffffffff81354ac7 t4 : ffffffff81354ac7 [25469.032664] t5 : ffffffff81354ac8 t6 : ffffffff81354ac7 [25469.032698] status: 0000000200000100 badaddr: ffffffff8007db1e cause: 0000000000000003 [25469.032738] [<fffffff8007db1e>] __free_percpu_irq+0xa2/0x150 [25469.032797] [<fffffff8007dbfc>] free_percpu_irq+0x30/0x5e [25469.032856] [<fffffff013a57dc>] kvm_riscv_aia_exit+0x40/0x42 [kvm] [25469.033947] [<fffffff013b4e82>] cleanup_module+0x10/0x32 [kvm] [25469.035300] [<fffffff8009b150>] __riscv_sys_delete_module+0x18e/0x1fc [25469.035374] [<fffffff8000c1ca>] syscall_handler+0x3a/0x46 [25469.035456] [<fffffff809ec9a4>] do_trap_ecall_u+0x72/0x134 [25469.035536] [<fffffff809f5e18>] handle_exception+0x148/0x156 Invoke aia_exit and other arch specific cleanup functions after kvm_exit so that disable gets a chance to be called first before exit.
CVE-2025-23136	In the Linux kernel, the following vulnerability has been resolved: thermal: int340x: Add NULL check for adev Not all devices have an ACPI companion fwnode, so adev might be NULL. This is similar to the commit cd2fd6eab480 ("platform/x86: int3472: Check for adev == NULL"). Add a check for adev not being set and return -ENODEV in that case to avoid a possible NULL pointer deref in int3402_thermal_probe(). Note, under the same directory, int3400_thermal_probe() has such a check. [rjw: Subject edit, added Fixes:]
CVE-2025-23138	
CVE-2025-23140	
CVE-2025-23141	
CVE-2025-23142	
CVE-2025-23143	
CVE-2025-23144	
CVE-2025-23145	
CVE-2025-23146	
CVE-2025-23147	
CVE-2025-23148	
CVE-2025-23149	
CVE-2025-23150	
CVE-2025-23151	
CVE-2025-23155	
CVE-2025-23156	
CVE-2025-23157	
CVE-2025-23158	
CVE-2025-23159	
CVE-2025-23160	

CVE ID	Description
CVE-2025-23161	
CVE-2025-23162	
CVE-2025-23163	
CVE-2025-23184	
CVE-2025-24014	
CVE-2025-24023	
CVE-2025-24294	
CVE-2025-24357	
CVE-2025-24513	
CVE-2025-24514	
CVE-2025-24528	
CVE-2025-24789	
CVE-2025-24790	
CVE-2025-24793	
CVE-2025-24794	
CVE-2025-24795	
CVE-2025-24813	
CVE-2025-24970	
CVE-2025-25183	
CVE-2025-25184	
CVE-2025-25186	
CVE-2025-25193	
CVE-2025-25473	
CVE-2025-25724	
CVE-2025-26603	
CVE-2025-26699	
CVE-2025-27018	
CVE-2025-27111	
CVE-2025-27113	
CVE-2025-27144	
CVE-2025-27152	
CVE-2025-27219	
CVE-2025-27220	
CVE-2025-27221	
CVE-2025-27496	
CVE-2025-27516	
CVE-2025-27587	
CVE-2025-27610	
CVE-2025-27613	

CVE ID	Description
CVE-2025-27614	
CVE-2025-27789	
CVE-2025-27809	
CVE-2025-27810	
CVE-2025-29087	
CVE-2025-29088	
CVE-2025-2953	
CVE-2025-29768	
CVE-2025-29770	
CVE-2025-29786	
CVE-2025-30167	
CVE-2025-30202	
CVE-2025-30204	
CVE-2025-30258	
CVE-2025-30370	
CVE-2025-30473	
CVE-2025-31115	
CVE-2025-31650	
CVE-2025-31651	
CVE-2025-3198	
CVE-2025-32381	
CVE-2025-32386	
CVE-2025-32387	
CVE-2025-32414	
CVE-2025-32415	
CVE-2025-32434	
CVE-2025-32444	
CVE-2025-32462	
CVE-2025-32463	
CVE-2025-3262	
CVE-2025-3263	
CVE-2025-3264	
CVE-2025-32728	
CVE-2025-3277	
CVE-2025-32873	
CVE-2025-32962	
CVE-2025-32988	
CVE-2025-32989	
CVE-2025-32990	

CVE ID	Description
CVE-2025-3360	
CVE-2025-35036	Hibernate Validator before 6.2.0 and 7.0.0, by default and depending how it is used, may interpolate user-supplied input in a constraint violation message with Expression Language. This could allow an attacker to access sensitive information or execute arbitrary Java code. Hibernate Validator as of 6.2.0 and 7.0.0 no longer interpolates custom constraint violation messages with Expression Language and strongly recommends not allowing user-supplied input in constraint violation messages. CVE-2020-5245 and CVE-2025-4428 are examples of related, downstream vulnerabilities involving Expression Language interpolation of user-supplied data.
CVE-2025-3576	A vulnerability in the MIT Kerberos implementation allows GSSAPI-protected messages using RC4-HMAC-MD5 to be spoofed due to weaknesses in the MD5 checksum design. If RC4 is preferred over stronger encryption types, an attacker could exploit MD5 collisions to forge message integrity codes. This may lead to unauthorized message tampering.
CVE-2025-3588	A vulnerability, which was classified as problematic, has been found in joelittlejohn jsonschema2pojo 1.2.2. This issue affects the function apply of the file org/jsonschema2pojo/rules/SchemaRule.java of the component JSON File Handler. The manipulation leads to stack-based buffer overflow. Attacking locally is a requirement. The exploit has been disclosed to the public and may be used. The vendor was contacted early about this disclosure but did not respond in any way.
CVE-2025-37738	In the Linux kernel, the following vulnerability has been resolved: ext4: ignore xattrs past end Once inside 'ext4_xattr_inode_dec_ref_all' we should ignore xattrs entries past the 'end' entry. This fixes the following KASAN reported issue: ===== BUG: KASAN: slab-use-after-free in ext4_xattr_inode_dec_ref_all+0xb8c/0xe90 Read of size 4 at addr ffff888012c120c4 by task repro/2065 CPU: 1 UID: 0 PID: 2065 Comm: repro Not tainted 6.13.0-rc2+ #11 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x1fd/0x300 ? tcp_gro_dev_warn+0x260/0x260 ? _printk+0xc0/0x100 ? read_lock_is_recursive+0x10/0x10 ? irq_work_queue+0x72/0xf0 ? __virt_addr_valid+0x17b/0x4b0 print_address_description +0x78/0x390 print_report+0x107/0x1f0 ? __virt_addr_valid+0x17b/0x4b0 ? __virt_addr_valid +0x3ff/0x4b0 ? __phys_addr+0xb5/0x160 ? ext4_xattr_inode_dec_ref_all+0xb8c/0xe90 kasan_report +0xcc/0x100 ? ext4_xattr_inode_dec_ref_all+0xb8c/0xe90 ext4_xattr_inode_dec_ref_all +0xb8c/0xe90 ? ext4_xattr_delete_inode+0xd30/0xd30 ? __ext4_journal_ensure_credits +0x5f0/0x5f0 ? __ext4_journal_ensure_credits+0x2b/0x5f0 ? inode_update_timestamps +0x410/0x410 ext4_xattr_delete_inode+0xb64/0xd30 ? ext4_truncate+0xb70/0xdc0 ? ext4_expand_extra_isize_ea+0x1d20/0x1d20 ? __ext4_mark_inode_dirty+0x670/0x670 ? ext4_journal_check_start+0x16f/0x240 ? ext4_inode_is_fast_symlink+0x2f2/0x3a0 ext4_evict_inode +0xc8c/0xff0 ? ext4_inode_is_fast_symlink+0x3a0/0x3a0 ? do_raw_spin_unlock+0x53/0x8a0 ? ext4_inode_is_fast_symlink+0x3a0/0x3a0 evict+0x4ac/0x950 ? proc_nr_inodes+0x310/0x310 ? trace_ext4_drop_inode+0xa2/0x220 ? _raw_spin_unlock+0x1a/0x30 ? iput+0x4cb/0x7e0 do_unlinkat +0x495/0x7c0 ? try_break_deleg+0x120/0x120 ? 0xffffffff81000000 ? __check_object_size +0x15a/0x210 ? strncpy_from_user+0x13e/0x250 ? getname_flags+0x1dc/0x530 __x64_sys_unlinkat +0xc8/0xf0 do_syscall_64+0x65/0x110 entry_SYSCALL_64_after_hwframe+0x67/0x6f RIP: 0033:0x434ffd Code: 66 2e 0f 1f 84 00 00 00 00 0f 1f 0f 3f 0f 1e fa 89 fb 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 8 RSP: 002b:00007ffc50fa7b28 EFLAGS: 00000246 ORIG_RAX: 0000000000000107 RAX: ffffffff81ffffda RBX: 00007ffc50fa7e18 RCX: 0000000000434ffd RDX: 0000000000000000 RSI: 0000000020000240 RDI: 0000000000000005 RBP: 00007ffc50fa7be0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000001 R13: 00007ffc50fa7e08 R14: 00000000004abbf3 R15: 0000000000000001 </ TASK> The buggy address belongs to the object at ffff888012c12000 which belongs to the cache filp of size 360 The buggy address is located 196 bytes inside of freed 360-byte region [ffff888012c12000, ffff888012c12168) The buggy address belongs to the physical page: page: refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x12c12 head: order:1 mapcount:0 entire_mapcount:0 nr_pages_mapped:0 pincount:0 flags: 0x40(head node=0 zone=0) page_type: f5(slab) raw: 0000000000000004 ffff888000ad7640 ffffea0000497a00 dead000000000004 raw: 0000000000000000 0000000000100010 00000001f5000000 0000000000000000 head: 00000000000000040 ffff888000ad7640 ffffea0000497a00 dead000000000004 head: 0000000000000000 000000000100010 00000001f5000000 0000000000000000 head: 0000000000000001 ffffea00004b481 ffffffffbfffff 0000000000000000 head: 0000000000000002 0000000000000000 00000000ffffffbf 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff888012c11f80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ffff888012c12000: fa fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ^ ffff888012c12100: fb fb fb fb fb fb fb fb fb fb fc fc fc ffff888012c12180: fc fc fc fc fc fc fc fc fc fc ---truncated---

CVE ID	Description
CVE-2025-37739	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid out-of-bounds access in f2fs_truncate_inode_blocks() syzbot reports an UBSAN issue as below: <pre> -----[cut here]----- UBSAN: array-index-out-of-bounds in fs/f2fs/node.h:381:10 index 18446744073709550692 is out of range for type '___le32[5]' (aka 'unsigned int[5]') CPU: 0 UID: 0 PID: 5318 Comm: syz.0.0 Not tainted 6.14.0-rc3-syzkaller-00060-g6537cfb395f3 #0 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 ubsan_epilogue lib/ubsan.c:231 [inline] __ubsan_handle_out_of_bounds+0x121/0x150 lib/ubsan.c:429 get_nid fs/f2fs/node.h:381 [inline] f2fs_truncate_inode_blocks+0xa5e/0xf60 fs/f2fs/node.c:1181 f2fs_do_truncate_blocks+0x782/0x1030 fs/f2fs/file.c:808 f2fs_truncate_blocks+0x10d/0x300 fs/ f2fs/file.c:836 f2fs_truncate+0x417/0x720 fs/f2fs/file.c:886 f2fs_file_write_iter+0x1bdb/0x2550 fs/ f2fs/file.c:5093 aio_write+0x56b/0x7c0 fs/aio.c:1633 io_submit_one+0x8a7/0x18a0 fs/aio.c:2052 __do_sys_io_submit fs/aio.c:2111 [inline] __se_sys_io_submit+0x171/0x2e0 fs/aio.c:2081 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/ common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f238798cde9 index 18446744073709550692 (decimal, unsigned long long) = 0xffffffffffffc64 (hexadecimal, unsigned long long) = -924 (decimal, long long) In f2fs_truncate_inode_blocks(), UBSAN detects that get_nid() tries to access .i_nid[-924], it means both offset[0] and level should zero. The possible case should be in f2fs_do_truncate_blocks(), we try to truncate inode size to zero, however, dn.ofs_in_node is zero and dn.node_page is not an inode page, so it fails to truncate inode page, and then pass zeroed free_from to f2fs_truncate_inode_blocks(), result in this issue. if (dn.ofs_in_node IS_INODE(dn.node_page)) { f2fs_truncate_data_blocks_range(&dn, count); free_from += count; } I guess the reason why dn.node_page is not an inode page could be: there are multiple nat entries share the same node block address, once the node block address was reused, f2fs_get_node_page() may load a non-inode block. Let's add a sanity check for such condition to avoid out-of-bounds access issue.</pre>
CVE-2025-37740	In the Linux kernel, the following vulnerability has been resolved: jfs: add sanity check for agwidth in dbMount The width in dmapctl of the AG is zero, it trigger a divide error when calculating the control page level in dbAllocAG. To avoid this issue, add a check for agwidth in dbAllocAG.

CVE ID	Description
CVE-2025-37741	In the Linux kernel, the following vulnerability has been resolved: jfs: Prevent copying of nlink with value 0 from disk inode syzbot report a deadlock in diFree. [1] When calling "ioctl \$LOOP_SET_STATUS64", the offset value passed in is 4, which does not match the mounted loop device, causing the mapping of the mounted loop device to be invalidated. When creating the directory and creating the inode of iag in diReadSpecial(), read the page of fixed disk inode (AIT) in raw mode in read_metapage(), the metapage data it returns is corrupted, which causes the nlink value of 0 to be assigned to the iag inode when executing copy_from_dinode(), which ultimately causes a deadlock when entering diFree(). To avoid this, first check the nlink value of dinode before setting iag inode. [1] WARNING: possible recursive locking detected 6.12.0-rc7-syzkaller-00212-g4a5df3796467 #0 Not tainted ----- syz-executor301/5309 is trying to acquire lock: ffff888044548920 (&(imap->im_aglock[index])){+.+.}-{3:3}, at: diFree+0x37c/0x2fb0 fs/jfs/jfs_imap.c:889 but task is already holding lock: ffff888044548920 (&(imap->im_aglock[index])){+.+.}-{3:3}, at: diAlloc+0x1b6/0x1630 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(&(imap->im_aglock[index])); lock(&(imap->im_aglock[index])); *** DEADLOCK *** May be due to missing lock nesting notation 5 locks held by syz-executor301/5309: #0: ffff8880422a4420 (sb_writers#9){+.+.}-{0:0}, at: mnt_want_write+0x3f/0x90 fs/namespace.c:515 #1: ffff88804755b390 (&type->i_mutex_dir_key#6/1){+.+.}-{3:3}, at: inode_lock_nested include/linux/fs.h:850 [inline] #1: ffff88804755b390 (&type->i_mutex_dir_key#6/1){+.+.}-{3:3}, at: filename_create +0x260/0x540 fs/namei.c:4026 #2: ffff888044548920 (&(imap->im_aglock[index])){+.+.}-{3:3}, at: diAlloc+0x1b6/0x1630 #3: ffff888044548890 (&(imap->im_freelock)){+.+.}-{3:3}, at: diNewIAG fs/jfs/jfs_imap.c:2460 [inline] #3: ffff888044548890 (&(imap->im_freelock)){+.+.}-{3:3}, at: diAllocExt fs/jfs/jfs_imap.c:1905 [inline] #3: ffff888044548890 (&(imap->im_freelock)){+.+.}-{3:3}, at: diAllocAG +0x4b7/0x1e50 fs/jfs/jfs_imap.c:1669 #4: ffff88804755a618 (&jfs_ip->rdwrlock/1){++++}-{3:3}, at: diNewIAG fs/jfs/jfs_imap.c:2477 [inline] #4: ffff88804755a618 (&jfs_ip->rdwrlock/1){++++}-{3:3}, at: diAllocExt fs/jfs/jfs_imap.c:1905 [inline] #4: ffff88804755a618 (&jfs_ip->rdwrlock/1){++++}-{3:3}, at: diAllocAG+0x869/0x1e50 fs/jfs/jfs_imap.c:1669 stack backtrace: CPU: 0 UID: 0 PID: 5309 Comm: syz-executor301 Not tainted 6.12.0-rc7-syzkaller-00212-g4a5df3796467 #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_deadlock_bug+0x483/0x620 kernel/locking/lockdep.c:3037 check_deadlock kernel/locking/lockdep.c:3089 [inline] validate_chain+0x15e2/0x5920 kernel/locking/lockdep.c:3891 __lock_acquire+0x1384/0x2050 kernel/locking/lockdep.c:5202 lock_acquire+0x1ed/0x550 kernel/locking/lockdep.c:5825 __mutex_lock_common kernel/locking/mutex.c:608 [inline] __mutex_lock +0x136/0xd70 kernel/locking/mutex.c:752 diFree+0x37c/0x2fb0 fs/jfs/jfs_imap.c:889 jfs_evict_inode +0x32d/0x440 fs/jfs/inode.c:156 evict+0x4e8/0x9b0 fs/inode.c:725 diFreeSpecial fs/jfs/jfs_imap.c:552 [inline] duplicateXtree+0x3c6/0x550 fs/jfs/jfs_imap.c:3022 diNewIAG fs/jfs/jfs_imap.c:2597 [inline] diAllocExt fs/jfs/jfs_imap.c:1905 [inline] diAllocAG+0x17dc/0x1e50 fs/jfs/jfs_imap.c:1669 diAlloc+0x1d2/0x1630 fs/jfs/jfs_imap.c:1590 ialloc+0x8f/0x900 fs/jfs/jfs_inode.c:56 jfs_mkdir +0x1c5/0xba0 fs/jfs/namei.c:225 vfs_mkdir+0x2f9/0x4f0 fs/namei.c:4257 do_mkdirat+0x264/0x3a0 fs/namei.c:4280 __do_sys_mkdirat fs/namei.c:4295 [inline] __se_sys_mkdirat fs/namei.c:4293 [inline] __x64_sys_mkdirat+0x87/0xa0 fs/namei.c:4293 do_syscall_x64 arch/x86/en---truncated---
CVE-2025-37742	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix uninit-value access of imap allocated in the diMount() function syzbot reports that hex_dump_to_buffer is using uninit-value: ===== BUG: KMSAN: uninit-value in hex_dump_to_buffer+0x888/0x1100 lib/hexdump.c:171 hex_dump_to_buffer +0x888/0x1100 lib/hexdump.c:171 print_hex_dump+0x13d/0x3e0 lib/hexdump.c:276 diFree +0x5ba/0x4350 fs/jfs/jfs_imap.c:876 jfs_evict_inode+0x510/0x550 fs/jfs/inode.c:156 evict +0x723/0xd10 fs/inode.c:796 iput_final fs/inode.c:1946 [inline] iput+0x97b/0xdb0 fs/inode.c:1972 txUpdateMap+0xf3e/0x1150 fs/jfs/jfs_txnmgr.c:2367 txLazyCommit fs/jfs/jfs_txnmgr.c:2664 [inline] jfs_lazycommit+0x627/0x11d0 fs/jfs/jfs_txnmgr.c:2733 kthread+0x6b9/0xef0 kernel/kthread.c:464 ret_from_fork+0x6d/0x90 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 Uninit was created at: slab_post_alloc_hook mm/slab.c:4121 [inline] slab_alloc_node mm/slab.c:4164 [inline] __kmalloccache_noprof+0x8e3/0xdf0 mm/slab.c:4320 kmalloccache_noprof include/linux/slab.h:901 [inline] diMount+0x61/0x7f0 fs/jfs/jfs_imap.c:105 jfs_mount+0xa8e/0x11d0 fs/jfs/jfs_mount.c:176 jfs_fill_super+0xa47/0x17c0 fs/jfs/super.c:523 get_tree_bdev_flags+0x6ec/0x910 fs/super.c:1636 get_tree_bdev+0x37/0x50 fs/super.c:1659 jfs_get_tree+0x34/0x40 fs/jfs/super.c:635 vfs_get_tree+0xb1/0x5a0 fs/super.c:1814 do_new_mount +0x71f/0x1e0 fs/namespace.c:3560 path_mount+0x742/0x1f10 fs/namespace.c:3887 do_mount fs/namespace.c:3900 [inline] __do_sys_mount fs/namespace.c:4111 [inline] __se_sys_mount+0x71f/0x800 fs/namespace.c:4088 __x64_sys_mount+0xe4/0x150 fs/namespace.c:4088 x64_sys_call+0x39bf/0x3c30 arch/x86/include/generated/asm/syscalls_64.h:166 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe +0x77/0x7f ===== The reason is that imap is not properly initialized after memory allocation. It will cause the sprintf() function to write uninitialized data into linebuf within hex_dump_to_buffer(). Fix this by using kcalloc instead of kmalloc to clear its content at the beginning in diMount().

CVE ID	Description
CVE-2025-37743	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Avoid memory leak while enabling statistics Driver uses monitor destination rings for extended statistics mode and standalone monitor mode. In extended statistics mode, TLVs are parsed from the buffer received from the monitor destination ring and assigned to the ppdu_info structure to update per-packet statistics. In standalone monitor mode, along with per-packet statistics, the packet data (payload) is captured, and the driver updates per MSDU to mac80211. When the AP interface is enabled, only extended statistics mode is activated. As part of enabling monitor rings for collecting statistics, the driver subscribes to HAL_RX_MPDU_START TLV in the filter configuration. This TLV is received from the monitor destination ring, and kcalloc for the mon_mpdw object occurs, which is not freed, leading to a memory leak. The kcalloc for the mon_mpdw object is only required while enabling the standalone monitor interface. This causes a memory leak while enabling extended statistics mode in the driver. Fix this memory leak by removing the kcalloc for the mon_mpdw object in the HAL_RX_MPDU_START TLV handling. Additionally, remove the standalone monitor mode handlings in the HAL_MON_BUF_ADDR and HAL_RX_MSDU_END TLVs. These TLV tags will be handled properly when enabling standalone monitor mode in the future. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.3.1-00173-QCAHKSUPL_SILICONZ-1 Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.0.c5-00481-QCAHMTUPL_V1.0_V2.0_SILICONZ-3
CVE-2025-37744	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix memory leak in ath12k_pci_remove() Kmemleak reported this error: unreferenced object 0xffff1c165cec3060 (size 32): comm "insmod", pid 560, jiffies 4296964570 (age 235.596s) backtrace: [<000000005434db68>] __kmem_cache_alloc_node+0x1f4/0x2c0 [<000000001203b155>] kmalloc_trace+0x40/0x88 [<0000000028adc9c8>] _request_firmware+0xb8/0x608 [<00000000cad1aef7>] firmware_request_nowarn+0x50/0x80 [<000000005011a682>] local_pci_probe+0x48/0xd0 [<00000000077cd295>] pci_device_probe+0xb4/0x200 [<0000000087184c94>] really_probe+0x150/0x2c0 The firmware memory was allocated in ath12k_pci_probe(), but not freed in ath12k_pci_remove() in case ATH12K_FLAG_QMI_FAIL bit is set. So call ath12k_fw_unmap() to free the memory. Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.2.0-02280-QCAHMTUPL_V1.0_V2.0_SILICONZ-1

CVE ID	Description
CVE-2025-37745	In the Linux kernel, the following vulnerability has been resolved: PM: hibernate: Avoid deadlock in hibernate_compressor_param_set() syzbot reported a deadlock in lock_system_sleep() (see below). The write operation to "/sys/module/hibernate/parameters/compressor" conflicts with the registration of ieee80211 device, resulting in a deadlock when attempting to acquire system_transition_mutex under param_lock. To avoid this deadlock, change hibernate_compressor_param_set() to use mutex_trylock() for attempting to acquire system_transition_mutex and return -EBUSY when it fails. Task flags need not be saved or adjusted before calling mutex_trylock(&system_transition_mutex) because the caller is not going to end up waiting for this mutex and if it runs concurrently with system suspend in progress, it will be frozen properly when it returns to user space. syzbot report: syz-executor895/5833 is trying to acquire lock: ffffffff8e0828c8 (system_transition_mutex){+..}-{4:4}, at: lock_system_sleep +0x87/0xa0 kernel/power/main.c:56 but task is already holding lock: ffffffff8e07dc68 (param_lock) {+..}-{4:4}, at: kernel_param_lock kernel/params.c:607 [inline] ffffffff8e07dc68 (param_lock) {+..}-{4:4}, at: param_attr_store+0xe6/0x300 kernel/params.c:586 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #3 (param_lock){+..}-{4:4}: __mutex_lock_common kernel/locking/mutex.c:585 [inline] __mutex_lock+0x19b/0xb10 kernel/locking/mutex.c:730 ieee80211_rate_control_ops_get net/mac80211/rate.c:220 [inline] rate_control_alloc net/mac80211/rate.c:266 [inline] ieee80211_init_rate_ctrl_alg+0x18d/0x6b0 net/mac80211/rate.c:1015 ieee80211_register_hw+0x20cd/0x4060 net/mac80211/main.c:1531 mac80211_hwsim_new_radio+0x304e/0x54e0 drivers/net/wireless/virtual/mac80211_hwsim.c:5558 init_mac80211_hwsim+0x432/0x8c0 drivers/net/wireless/virtual/mac80211_hwsim.c:6910 do_one_initcall+0x128/0x700 init/main.c:1257 do_initcall_level init/main.c:1319 [inline] do_initcalls init/main.c:1335 [inline] do_basic_setup init/main.c:1354 [inline] kernel_init_freeable+0x5c7/0x900 init/main.c:1568 kernel_init+0x1c/0x2b0 init/main.c:1457 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 -> #2 (rtnl_mutex){+..}-{4:4}: __mutex_lock_common kernel/locking/mutex.c:585 [inline] __mutex_lock+0x19b/0xb10 kernel/locking/mutex.c:730 wg_pm_notification drivers/net/wireguard/device.c:80 [inline] wg_pm_notification +0x49/0x180 drivers/net/wireguard/device.c:64 notifier_call_chain+0xb7/0x410 kernel/notifier.c:85 notifier_call_chain_robust kernel/notifier.c:120 [inline] blocking_notifier_call_chain_robust kernel/notifier.c:345 [inline] blocking_notifier_call_chain_robust+0xc9/0x170 kernel/notifier.c:333 pm_notifier_call_chain_robust+0x27/0x60 kernel/power/main.c:102 snapshot_open+0x189/0x2b0 kernel/power/user.c:77 misc_open+0x35a/0x420 drivers/char/misc.c:179 chrdev_open+0x237/0x6a0 fs/char_dev.c:414 do_dentry_open+0x735/0x1c40 fs/open.c:956 vfs_open+0x82/0x3f0 fs/open.c:1086 do_open fs/namei.c:3830 [inline] path_openat+0x1e88/0x2d80 fs/namei.c:3989 do_filp_open +0x20c/0x470 fs/namei.c:4016 do_sys_openat2+0x17a/0x1e0 fs/open.c:1428 do_sys_open fs/open.c:1443 [inline] __do_sys_openat fs/open.c:1459 [inline] __se_sys_openat fs/open.c:1454 [inline] __x64_sys_openat+0x175/0x210 fs/open.c:1454 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe +0x77/0x7f -> #1 ((pm_chain_head),rwsem){++++}-{4:4}: down_read+0x9a/0x330 kernel/locking/rwsem.c:1524 blocking_notifier_call_chain_robust kerne ---truncated---
CVE-2025-37746	In the Linux kernel, the following vulnerability has been resolved: perf/dwc_pcie: fix duplicate pci_dev devices During platform_device_register, wrongly using struct device pci_dev as platform_data caused a kmempdup copy of pci_dev. Worse still, accessing the duplicated device leads to list corruption as its mutex content (e.g., list, magic) remains the same as the original.
CVE-2025-37747	In the Linux kernel, the following vulnerability has been resolved: perf: Fix hang while freeing sigtrap event Perf can hang while freeing a sigtrap event if a related deferred signal hadn't managed to be sent before the file got closed: perf_event_overflow() task_work_add(perf_pending_task) fput() task_work_add(__fput()) task_work_run() __fput() perf_release() perf_event_release_kernel() _free_event() perf_pending_task_sync() task_work_cancel() -> FAILED rcuwait_wait_event() Once task_work_run() is running, the list of pending callbacks is removed from the task_struct and from this point on task_work_cancel() can't remove any pending and not yet started work items, hence the task_work_cancel() failure and the hang on rcuwait_wait_event(). Task work could be changed to remove one work at a time, so a work running on the current task can always cancel a pending one, however the wait / wake design is still subject to inverted dependencies when remote targets are involved, as pictured by Oleg: T1 T2 fd = perf_event_open(pid => T2->pid); fd = perf_event_open(pid => T1->pid); close(fd) close(fd) <IRQ> <IRQ> perf_event_overflow() perf_event_overflow() task_work_add(perf_pending_task) task_work_add(perf_pending_task) <IRQ> <IRQ> fput() fput() task_work_add(__fput()) task_work_add(__fput()) task_work_run() task_work_run() __fput() __fput() perf_release() perf_release() perf_event_release_kernel() perf_event_release_kernel() _free_event() _free_event() perf_pending_task_sync() perf_pending_task_sync() rcuwait_wait_event() rcuwait_wait_event() Therefore the only option left is to acquire the event reference count upon queueing the perf task work and release it from the task work, just like it was done before 3a5465418f5f ("perf: Fix event leak upon exec and file release") but without the leaks it fixed. Some adjustments are necessary to make it work: * A child event might dereference its parent upon freeing. Care must be taken to release the parent last. * Some places assuming the event doesn't have any reference held and therefore can be freed right away must instead put the reference and let the reference counting to its job.

CVE ID	Description
CVE-2025-37748	In the Linux kernel, the following vulnerability has been resolved: iommu/mediatek: Fix NULL pointer dereference in mtk_iommu_device_group Currently, mtk_iommu calls during probe iommu_device_register before the hw_list from driver data is initialized. Since iommu probing issue fix, it leads to NULL pointer dereference in mtk_iommu_device_group when hw_list is accessed with list_first_entry (not null safe). So, change the call order to ensure iommu_device_register is called after the driver data are initialized.
CVE-2025-37749	In the Linux kernel, the following vulnerability has been resolved: net: ppp: Add bound checking for skb data on ppp_sync_txmung Ensure we have enough data in linear buffer from skb before accessing initial bytes. This prevents potential out-of-bounds accesses when processing short packets. When ppp_sync_txmung receives an incoming package with an empty payload: (remote) gef➤ p *(struct pppoe_hdr *) (skb->head + skb->network_header) \$18 = { type = 0x1, ver = 0x1, code = 0x0, sid = 0x2, length = 0x0, tag = 0xffff8880371cdb96 } from the skb struct (trimmed) tail = 0x16, end = 0x140, head = 0xffff88803346f400 "4", data = 0xffff88803346f416 "\377", truesize = 0x380, len = 0x0, data_len = 0x0, mac_len = 0xe, hdr_len = 0x0, it is not safe to access data[2]. [pabeni@redhat.com: fixed subj typo]
CVE-2025-37750	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix UAF in decryption with multichannel After commit f7025d861694 ("smb: client: allocate crypto only for primary server") and commit b0abcd65ec54 ("smb: client: fix UAF in async decryption"), the channels started reusing AEAD TFM from primary channel to perform synchronous decryption, but that can't done as there could be multiple cifs threads (one per channel) simultaneously accessing it to perform decryption. This fixes the following KASAN splat when running fstest generic/249 with 'vers=3.1.1,multichannel,max_channels=4,seal' against Windows Server 2022: BUG: KASAN: slab-use-after-free in gf128mul_4k_lle+0xba/0x110 Read of size 8 at addr ffff8881046c18a0 by task cifs/986 CPU: 3 UID: 0 PID: 986 Comm: cifs Not tainted 6.15.0-rc1 #1 PREEMPT(voluntary) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-3.fc41 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x5d/0x80 print_report+0x156/0x528 ? gf128mul_4k_lle+0xba/0x110 ? __virt_addr_valid+0x145/0x300 ? __phys_addr+0x46/0x90 ? gf128mul_4k_lle+0xba/0x110 kasan_report+0xdf/0x1a0 ? gf128mul_4k_lle+0xba/0x110 gf128mul_4k_lle+0xba/0x110 ghash_update+0x189/0x210 shash_ahash_update+0x295/0x370 ? __pfx_shash_ahash_update+0x10/0x10 ? __pfx_shash_ahash_update+0x10/0x10 ? __pfx_extract_iter_to_sg+0x10/0x10 ? __kmalloc_large_node+0x10e/0x180 ? __asan_memset+0x23/0x50 crypto_ahash_update+0x3c/0xc0 gcm_hash_assoc_remain_continue+0x93/0xc0 crypt_message+0xe09/0xec0 [cifs] ? __pfx_crypt_message+0x10/0x10 [cifs] ? __raw_spin_unlock+0x23/0x40 ? __pfx_cifs_readv_from_socket+0x10/0x10 [cifs] decrypt_raw_data+0x229/0x380 [cifs] ? __pfx_decrypt_raw_data+0x10/0x10 [cifs] ? __pfx_cifs_read_iter_from_socket+0x10/0x10 [cifs] smb3_receive_transform+0x837/0xc80 [cifs] ? __pfx_smb3_receive_transform+0x10/0x10 [cifs] ? __pfx__might_resched+0x10/0x10 ? __pfx_smb3_is_transform_hdr+0x10/0x10 [cifs] cifs_demultiplex_thread+0x692/0x1570 [cifs] ? __pfx_cifs_demultiplex_thread+0x10/0x10 [cifs] ? rcu_is_watching+0x20/0x50 ? rcu_lockdep_current_cpu_online+0x62/0xb0 ? find_held_lock+0x32/0x90 ? kvm_sched_clock_read+0x11/0x20 ? local_clock_noinstr+0xd/0xd0 ? trace_irq_enable.constprop.0+0xa8/0xe0 ? __pfx_cifs_demultiplex_thread+0x10/0x10 [cifs] kthread+0x1fe/0x380 ? kthread+0x10f/0x380 ? __pfx_kthread+0x10/0x10 ? local_clock_noinstr+0xd/0xd0 ? ret_from_fork+0x1b/0x60 ? local_clock+0x15/0x30 ? lock_release+0x29b/0x390 ? rcu_is_watching+0x20/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x31/0x60 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK>
CVE-2025-37752	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: move the limit validation It is not sufficient to directly validate the limit on the data that the user passes as it can be updated based on how the other parameters are changed. Move the check at the end of the configuration update process to also catch scenarios where the limit is indirectly updated, for example with the following configurations: tc qdisc add dev dummy0 handle 1: root sfq limit 2 flows 1 depth 1 tc qdisc add dev dummy0 handle 1: root sfq limit 2 flows 1 divisor 1 This fixes the following syzkaller reported crash: -----[cut here]----- UBSAN: array-index-out-of-bounds in net/sched/sch_sfq.c:203:6 index 65535 is out of range for type 'struct sfq_head[128]' CPU: 1 UID: 0 PID: 3037 Comm: syz.2.16 Not tainted 6.14.0-rc2-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 12/27/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x201/0x300 lib/dump_stack.c:120 ubsan_epilogue lib/ubsan.c:231 [inline] __ubsan_handle_out_of_bounds+0xf5/0x120 lib/ubsan.c:429 sfq_link net/sched/sch_sfq.c:203 [inline] sfq_dec+0x53c/0x610 net/sched/sch_sfq.c:231 sfq_dequeue+0x34e/0x8c0 net/sched/sch_sfq.c:493 sfq_reset+0x17/0x60 net/sched/sch_sfq.c:518 qdisc_reset+0x12e/0x600 net/sched/sch_generic.c:1035 tbf_reset+0x41/0x110 net/sched/sch_tbf.c:339 qdisc_reset+0x12e/0x600 net/sched/sch_generic.c:1035 dev_reset_queue+0x100/0x1b0 net/sched/sch_generic.c:1311 netdev_for_each_tx_queue include/linux/netdevice.h:2590 [inline] dev_deactivate_many+0x7e5/0xe70 net/sched/sch_generic.c:1375

CVE ID	Description
CVE-2025-37754	In the Linux kernel, the following vulnerability has been resolved: drm/i915/huc: Fix fence not released on early probe errors HuC delayed loading fence, introduced with commit 27536e03271da ("drm/i915/huc: track delayed HuC load with a fence"), is registered with object tracker early on driver probe but unregistered only from driver remove, which is not called on early probe errors. Since its memory is allocated under devres, then released anyway, it may happen to be allocated again to the fence and reused on future driver probes, resulting in kernel warnings that taint the kernel: <4> [309.731371] -----[cut here]----- <3> [309.731373] ODEBUG: init destroyed (active state 0) object: ffff88813d7dd2e0 object type: i915_sw_fence hint: sw_fence_dummy_notify+0x0/0x20 [i915] <4> [309.731575] WARNING: CPU: 2 PID: 3161 at lib/debugobjects.c:612 debug_print_object+0x93/0xf0 ... <4> [309.731693] CPU: 2 UID: 0 PID: 3161 Comm: i915_module_loa Tainted: G U 6.14.0-CL_DRM_16362-gf0fd77956987+ #1 ... <4> [309.731700] RIP: 0010:debug_print_object+0x93/0xf0 ... <4> [309.731728] Call Trace: <4> [309.731730] <TASK> ... <4> [309.731949] __debug_object_init+0x17b/0x1c0 <4> [309.731957] debug_object_init+0x34/0x50 <4> [309.732126] __i915_sw_fence_init+0x34/0x60 [i915] <4> [309.732256] intel_huc_init_early+0x4b/0x1d0 [i915] <4> [309.732468] intel_uc_init_early+0x61/0x680 [i915] <4> [309.732667] intel_gt_common_init_early+0x105/0x130 [i915] <4> [309.732804] intel_root_gt_init_early+0x63/0x80 [i915] <4> [309.732938] i915_driver_probe+0x1fa/0xeb0 [i915] <4> [309.733075] i915_pci_probe+0xe6/0x220 [i915] <4> [309.733198] local_pci_probe+0x44/0xb0 <4> [309.733203] pci_device_probe+0xf4/0x270 <4> [309.733209] really_probe+0xee/0x3c0 <4> [309.733215] __driver_probe_device+0x8c/0x180 <4> [309.733219] driver_probe_device+0x24/0xd0 <4> [309.733223] __driver_attach+0x10f/0x220 <4> [309.733230] bus_for_each_dev+0x7d/0xe0 <4> [309.733236] driver_attach+0x1e/0x30 <4> [309.733239] bus_add_driver+0x151/0x290 <4> [309.733244] driver_register+0x5e/0x130 <4> [309.733247] __pci_register_driver+0x7d/0x90 <4> [309.733251] i915_pci_register_driver+0x23/0x30 [i915] <4> [309.733413] i915_init+0x34/0x120 [i915] <4> [309.733655] do_one_initcall+0x62/0x3f0 <4> [309.733667] do_init_module+0x97/0x2a0 <4> [309.733671] load_module+0x25ff/0x2890 <4> [309.733688] init_module_from_file+0x97/0xe0 <4> [309.733701] idempotent_init_module+0x118/0x330 <4> [309.733711] __x64_sys_finit_module+0x77/0x100 <4> [309.733715] x64_sys_call+0x1f37/0x2650 <4> [309.733719] do_syscall_64+0x91/0x180 <4> [309.733763] entry_SYSCALL_64_after_hwframe+0x76/0x7e <4> [309.733792] </TASK> ... <4> [309.733806] ---[end trace 0000000000000000]--- That scenario is most easily reproducible with igt@i915_module_load@reload-with-fault-injection. Fix the issue by moving the cleanup step to driver release path. (cherry picked from commit 795dbde92fe5c6996a02a5b579481de73035e7bf)
CVE-2025-37755	In the Linux kernel, the following vulnerability has been resolved: net: libwx: handle page_pool_dev_alloc_pages error page_pool_dev_alloc_pages could return NULL. There was a WARN_ON(!page) but it would still proceed to use the NULL pointer and then crash. This is similar to commit 001ba0902046 ("net: fec: handle page_pool_dev_alloc_pages error"). This is found by our static analysis tool KNIghter.
CVE-2025-37756	In the Linux kernel, the following vulnerability has been resolved: net: tls: explicitly disallow disconnect syzbot discovered that it can disconnect a TLS socket and then run into all sort of unexpected corner cases. I have a vague recollection of Eric pointing this out to us a long time ago. Supporting disconnect is really hard, for one thing if offload is enabled we'd need to wait for all packets to be _acked_. Disconnect is not commonly used, disallow it. The immediate problem syzbot run into is the warning in the strp, but that's just the easiest bug to trigger: WARNING: CPU: 0 PID: 5834 at net/tls/tls_strp.c:486 tls_strp_msg_load+0x72e/0xa80 net/tls/tls_strp.c:486 RIP: 0010:tls_strp_msg_load+0x72e/0xa80 net/tls/tls_strp.c:486 Call Trace: <TASK> tls_rx_rec_wait+0x280/0xa60 net/tls/tls_sw.c:1363 tls_sw_recvmmsg+0x85c/0x1c30 net/tls/tls_sw.c:2043 inet6_recvmmsg+0x2c9/0x730 net/ipv6/af_inet6.c:678 sock_recvmmsg_nosec net/socket.c:1023 [inline] sock_recvmmsg+0x109/0x280 net/socket.c:1045 __sys_recvfrom+0x202/0x380 net/socket.c:2237
CVE-2025-37757	In the Linux kernel, the following vulnerability has been resolved: tipc: fix memory leak in tipc_link_xmit In case the backlog transmit queue for system-importance messages is overloaded, tipc_link_xmit() returns -ENOBUFFS but the skb list is not purged. This leads to memory leak and failure when a skb is allocated. This commit fixes this issue by purging the skb list before tipc_link_xmit() returns.
CVE-2025-37758	In the Linux kernel, the following vulnerability has been resolved: ata: pata_pxa: Fix potential NULL pointer dereference in pxa_ata_probe() devm_ioremap() returns NULL on error. Currently, pxa_ata_probe() does not check for this case, which can result in a NULL pointer dereference. Add NULL check after devm_ioremap() to prevent this issue.

CVE ID	Description
CVE-2025-37759	In the Linux kernel, the following vulnerability has been resolved: ublk: fix handling recovery & reissue in ublk_abort_queue() Commit 8284066946e6 ("ublk: grab request reference when the request is handled by userspace") doesn't grab request reference in case of recovery reissue. Then the request can be queued & re-dispatch & failed when canceling uring command. If it is one zc request, the request can be freed before io_uring returns the zc buffer back, then cause kernel panic: [126.773061] BUG: kernel NULL pointer dereference, address: 000000000000000c8 [126.773657] #PF: supervisor read access in kernel mode [126.774052] #PF: error_code(0x0000) - not-present page [126.774455] PGD 0 P4D 0 [126.774698] Oops: Oops: 0000 [#1] SMP NOPTI [126.775034] CPU: 13 UID: 0 PID: 1612 Comm: kworker/u64:55 Not tainted 6.14.0-blk+ #182 PREEMPT(full) [126.775676] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-1.fc39 04/01/2014 [126.776275] Workqueue: iou_exit io_ring_exit_work [126.776651] RIP: 0010:ublk_io_release+0x14/0x130 [ublk_drv] Fixes it by always grabbing request reference for aborting the request.
CVE-2025-37761	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix an out-of-bounds shift when invalidating TLB When the size of the range invalidated is larger than roundup_pow_of_two(ULONG_MAX), The function macro roundup_pow_of_two(length) will hit an out-of-bounds shift [1]. Use a full TLB invalidation for such cases. v2: - Use a define for the range size limit over which we use a full TLB invalidation. (Lucas) - Use a better calculation of the limit. [1]: [39.202421] -----[cut here]----- [39.202657] UBSAN: shift-out-of-bounds in ./include/linux/log2.h:57:13 [39.202673] shift exponent 64 is too large for 64-bit type 'long unsigned int' [39.202688] CPU: 8 UID: 0 PID: 3129 Comm: xe_exec_system_ Tainted: G U 6.14.0+ #10 [39.202690] Tainted: [U]=USER [39.202690] Hardware name: ASUS System Product Name/PRIME B560M-A AC, BIOS 2001 02/01/2023 [39.202691] Call Trace: [39.202692] <TASK> [39.202695] dump_stack_lvl+0x6e/0xa0 [39.202699] ubsan_epilogue+0x5/0x30 [39.202701] __ubsan_handle_shift_out_of_bounds.cold+0x61/0xe6 [39.202705] xe_gt_tlb_invalidation_range.cold+0x1d/0x3a [xe] [39.202800] ? find_held_lock+0x2b/0x80 [39.202803] ? mark_held_locks+0x40/0x70 [39.202806] xe_svm_invalidate+0x459/0x700 [xe] [39.202897] drm_gpusvm_notifier_invalidate+0x4d/0x70 [drm_gpusvm] [39.202900] __mmu_notifier_release+0x1f5/0x270 [39.202905] exit_mmap+0x40e/0x450 [39.202912] __mmapput+0x45/0x110 [39.202914] exit_mm+0xc5/0x130 [39.202916] do_exit+0x21c/0x500 [39.202918] ? lockdep_hardirqs_on_prepare+0xdb/0x190 [39.202920] do_group_exit+0x36/0xa0 [39.202922] get_signal+0x8f8/0x900 [39.202926] arch_do_signal_or_restart+0x35/0x100 [39.202930] syscall_exit_to_user_mode+0x1fc/0x290 [39.202932] do_syscall_64+0xa1/0x180 [39.202934] ? do_user_addr_fault+0x59f/0x8a0 [39.202937] ? lock_release+0xd2/0x2a0 [39.202939] ? do_user_addr_fault+0x5a9/0x8a0 [39.202942] ? trace_hardirqs_off+0x4b/0xc0 [39.202944] ? clear_bhb_loop+0x25/0x80 [39.202946] ? clear_bhb_loop+0x25/0x80 [39.202947] ? clear_bhb_loop+0x25/0x80 [39.202950] entry_SYSCALL_64_after_hwframe+0x76/0x7e [39.202952] RIP: 0033:0x7fa945e543e1 [39.202961] Code: Unable to access opcode bytes at 0x7fa945e543b7. [39.202962] RSP: 002b:00007ffca8fb4170 EFLAGS: 00000293 [39.202963] RAX: 000000000000003d RBX: 0000000000000000 RCX: 00007fa945e543e3 [39.202964] RDX: 0000000000000000 RSI: 00007ffca8fb41ac RDI: 00000000ffffffff [39.202964] RBP: 00007ffca8fb4190 R08: 0000000000000000 R09: 00007fa945f600a0 [39.202965] R10: 0000000000000000 R11: 0000000000000293 R12: 0000000000000000 [39.202966] R13: 00007fa9460dd310 R14: 00007ffca8fb41ac R15: 0000000000000000 [39.202970] </TASK> [39.202970] ---[end trace]--- (cherry picked from commit b88f48f86500bc0b44b4f73ac66d500a40d320ad)
CVE-2025-37763	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: take paired job reference For paired jobs, have the fragment job take a reference on the geometry job, so that the geometry job cannot be freed until the fragment job has finished with it. The geometry job structure is accessed when the fragment job is being prepared by the GPU scheduler. Taking the reference prevents the geometry job being freed until the fragment job no longer requires it. Fixes a use after free bug detected by KASAN: [124.256386] BUG: KASAN: slab-use-after-free in pvr_queue_prepare_job+0x108/0x868 [powervr] [124.264893] Read of size 1 at addr ffff0000084cb960 by task kworker/u16:4/63

CVE ID	Description
CVE-2025-37764	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: fix firmware memory leaks Free the memory used to hold the results of firmware image processing when the module is unloaded. Fix the related issue of the same memory being leaked if processing of the firmware image fails during module load. Ensure all firmware GEM objects are destroyed if firmware image processing fails. Fixes memory leaks on powervr module unload detected by Kmemleak: unreferenced object 0xffff000042e20000 (size 94208): comm "modprobe", pid 470, jiffies 4295277154 hex dump (first 32 bytes): 02 ae 7f ed bf 45 84 00 3c 5b 1f ed 9f 45 45 05 E..<[...EE. d5 4f 5d 14 6c 00 3d 23 30 d0 3a 4a 66 0e 48 c8 .O].l.=#0.:Jf.H. backtrace (crc dd329dec): kmemleak_alloc+0x30/0x40 ____kmalloclarge_node+0x140/0x188 ____kmalloclarge_node_noprof+0x2c/0x13c ____kmallocl_noprof+0x48/0x4c0 pvr_fw_init+0xaa4/0x1f50 [powervr] unreferenced object 0xffff000042d20000 (size 20480): comm "modprobe", pid 470, jiffies 4295277154 hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 09 00 00 00 0b 00 00 00 00 00 00 00 00 00 00 00 07 00 00 00 08 00 00 00 backtrace (crc 395b02e3): kmemleak_alloc+0x30/0x40 ____kmalloclarge_node+0x140/0x188 ____kmalloclarge_node_noprof+0x2c/0x13c ____kmallocl_noprof+0x48/0x4c0 pvr_fw_init+0xb0c/0x1f50 [powervr]
CVE-2025-37765	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: prime: fix ttm_bo_delayed_delete oops Fix an oops in ttm_bo_delayed_delete which results from dereferencing a dangling pointer: Oops: general protection fault, probably for non-canonical address 0x6b6b6b6b6b6b6b7b: 0000 [#1] PREEMPT SMP CPU: 4 UID: 0 PID: 1082 Comm: kworker/u65:2 Not tainted 6.14.0-rc4-00267-g505460b44513-dirty #216 Hardware name: LENOVO 82N6/LNVNB161216, BIOS GKCN65WW 01/16/2024 Workqueue: ttm ttm_bo_delayed_delete [ttm] RIP: 0010:dma_resv_iter_first_unlocked+0x55/0x290 Code: 31 f6 48 c7 c7 00 2b fa ae e8 97 bd 52 ff e8 a2 c1 53 00 5a 85 c0 74 48 e9 88 01 00 00 4c 89 63 20 4d 85 e4 0f 84 30 01 00 00 <41> 8b 44 24 10 c6 43 2c 01 48 89 df 89 43 28 e8 97 fd ff ff 4c 8b RSP: 0018:ffff9383473d60 EFLAGS: 00010202 RAX: 0000000000000001 RBX: fffff9383473d88 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: fffff9383473d78 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: 6b6b6b6b6b6b6b6b R13: fffffa003bbf78580 R14: fffffa003a6728040 R15: 000000000000383cc FS: 0000000000000000(0000) GS:ffffa00991c00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000758348024dd0 CR3: 000000012c259000 CR4: 0000000000f50ef0 PKRU: 55555554 Call Trace: <TASK> ? __die_body.cold+0x19/0x26 ? die_addr+0x3d/0x70 ? exc_general_protection+0x159/0x460 ? asm_exc_general_protection+0x27/0x30 ? dma_resv_iter_first_unlocked+0x55/0x290 dma_resv_wait_timeout+0x56/0x100 ttm_bo_delayed_delete+0x69/0xb0 [ttm] process_one_work+0x217/0x5c0 worker_thread+0x1c8/0x3d0 ? apply_wqattrs_cleanup.part.0+0xc0/0xc0 kthread+0x10b/0x240 ? kthreads_online_cpu+0x140/0x140 ret_from_fork+0x40/0x70 ? kthreads_online_cpu+0x140/0x140 ret_from_fork_asm+0x11/0x20 </TASK> The cause of this is: - drm_prime_gem_destroy calls dma_buf_put(dma_buf) which releases the reference to the shared dma_buf. The reference count is 0, so the dma_buf is destroyed, which in turn decrements the corresponding amdgpu_bo reference count to 0, and the amdgpu_bo is destroyed - calling drm_gem_object_release then dma_resv_fini (which destroys the reservation object), then finally freeing the amdgpu_bo. - nouveau_bo obj->bo.base.resv is now a dangling pointer to the memory formerly allocated to the amdgpu_bo. - nouveau_gem_object_del calls ttm_bo_put(&nvbo->bo) which calls ttm_bo_release, which schedules ttm_bo_delayed_delete. - ttm_bo_delayed_delete runs and dereferences the dangling resv pointer, resulting in a general protection fault. Fix this by moving the drm_prime_gem_destroy call from nouveau_gem_object_del to nouveau_bo_del_ttm. This ensures that it will be run after ttm_bo_delayed_delete.
CVE-2025-37766	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user can set any speed value. If speed is greater than UINT_MAX/8, division by zero is possible. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-37767	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user can set any speed value. If speed is greater than UINT_MAX/8, division by zero is possible. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-37768	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user can set any speed value. If speed is greater than UINT_MAX/8, division by zero is possible. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-37769	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm/smu11: Prevent division by zero The user can set any speed value. If speed is greater than UINT_MAX/8, division by zero is possible. Found by Linux Verification Center (linuxtesting.org) with SVACE. (cherry picked from commit da7dc714a8f8e1c9fc33c57cd63583779a3bef71)
CVE-2025-37770	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user can set any speed value. If speed is greater than UINT_MAX/8, division by zero is possible. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2025-37771	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: Prevent division by zero The user can set any speed value. If speed is greater than UINT_MAX/8, division by zero is possible. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-37772	In the Linux kernel, the following vulnerability has been resolved: RDMA/cma: Fix workqueue crash in cma_netevent_work_handler struct rdma_cm_id has member "struct work_struct net_work" that is reused for enqueueing cma_netevent_work_handler()s onto cma_wq. Below crash[1] can occur if more than one call to cma_netevent_callback() occurs in quick succession, which further enqueues cma_netevent_work_handler()s for the same rdma_cm_id, overwriting any previously queued work-item(s) that was just scheduled to run i.e. there is no guarantee the queued work item may run between two successive calls to cma_netevent_callback() and the 2nd INIT_WORK would overwrite the 1st work item (for the same rdma_cm_id), despite grabbing id_table_lock during enqueue. Also drgn analysis [2] indicates the work item was likely overwritten. Fix this by moving the INIT_WORK() to __rdma_create_id(), so that it doesn't race with any existing queue_work() or its worker thread. [1] Trimmed crash stack: ===== BUG: kernel NULL pointer dereference, address: 0000000000000008 kworker/u256:6 ... 6.12.0-0... Workqueue: cma_netevent_work_handler [rdma_cm] (rdma_cm) RIP: 0010:process_one_work+0xba/0x31a Call Trace: worker_thread+0x266/0x3a0 kthread+0xcf/0x100 ret_from_fork+0x31/0x50 ret_from_fork_asm+0x1a/0x30 ===== [2] drgn crash analysis: >>> trace = prog.crashed_thread().stack_trace() >>> trace (0) crash_setup_regs (/arch/x86/include/asm/kexec.h:111:15) (1) __crash_kexec (kernel/crash_core.c:122:4) (2) panic (kernel/panic.c:399:3) (3) oops_end (arch/x86/kernel/dumpstack.c:382:3) ... (8) process_one_work (kernel/workqueue.c:3168:2) (9) process_scheduled_works (kernel/workqueue.c:3310:3) (10) worker_thread (kernel/workqueue.c:3391:4) (11) kthread (kernel/kthread.c:389:9) Line workqueue.c:3168 for this kernel version is in process_one_work(): 3168 strcpy(worker->desc, pwq->wq->name, WORKER_DESC_LEN); >>> trace[8][\"work\"] *(struct work_struct *)0xffff92577d0a21d8 = { .data = (atomic_long_t){ .counter = (s64)536870912, <=== Note }, .entry = (struct list_head){ .next = (struct list_head *)0xffff924d075924c0, .prev = (struct list_head *)0xffff924d075924c0, }, .func = (work_func_t)cma_netevent_work_handler+0x0 = 0xffffffffc2cec280, } Suspicion is that pwq is NULL: >>> trace[8][\"pwq\"] (struct pool_workqueue *)<absent> In process_one_work(), pwq is assigned from: struct pool_workqueue *pwq = get_work_pwq(work); and get_work_pwq() is: static struct pool_workqueue *get_work_pwq(struct work_struct *work) { unsigned long data = atomic_long_read(&work->data); if (data & WORK_STRUCT_PWQ) return work_struct_pwq(data); else return NULL; } WORK_STRUCT_PWQ is 0x4: >>> print(repr(prog[\"WORK_STRUCT_PWQ\"])) Object(prog, 'enum work_flags', value=4) But work->data is 536870912 which is 0x20000000. So, get_work_pwq() returns NULL and we crash in process_one_work(): 3168 strcpy(worker->desc, pwq->wq->name, WORKER_DESC_LEN); =====
CVE-2025-37773	In the Linux kernel, the following vulnerability has been resolved: virtiofs: add filesystem context source name check In certain scenarios, for example, during fuzz testing, the source name may be NULL, which could lead to a kernel panic. Therefore, an extra check for the source name should be added.
CVE-2025-37775	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix the warning from __kernel_write_iter [2110.972290] -----[cut here]----- [2110.972301] WARNING: CPU: 3 PID: 735 at fs/read_write.c:599 __kernel_write_iter+0x21b/0x280 This patch doesn't allow writing to directory.
CVE-2025-37776	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in smb_break_all_levII_oplock() There is a room in smb_break_all_levII_oplock that can cause racy issues when unlocking in the middle of the loop. This patch use read lock to protect whole loop.
CVE-2025-37777	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in __smb2_lease_break_noti() Move tcp_transport free to ksmbd_conn_free. If ksmbd connection is referenced when ksmbd server thread terminates, It will not be freed, but conn->tcp_transport is freed. __smb2_lease_break_noti can be performed asynchronously when the connection is disconnected. __smb2_lease_break_noti calls ksmbd_conn_write, which can cause use-after-free when conn->ksmbd_transport is already freed.
CVE-2025-37778	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Fix dangling pointer in krb_authenticate krb_authenticate frees sess->user and does not set the pointer to NULL. It calls ksmbd_krb5_authenticate to reinitialise sess->user but that function may return without doing so. If that happens then smb2_sess_setup, which calls krb_authenticate, will be accessing free'd memory when it later uses sess->user.

CVE ID	Description
CVE-2025-37780	In the Linux kernel, the following vulnerability has been resolved: isofs: Prevent the use of too small fid syzbot reported a slab-out-of-bounds Read in isofs_fh_to_parent. [1] The handle_bytes value passed in by the reproducing program is equal to 12. In handle_to_path(), only 12 bytes of memory are allocated for the structure file_handle->f_handle member, which causes an out-of-bounds access when accessing the member parent_block of the structure isofs_fid in isofs, because accessing parent_block requires at least 16 bytes of f_handle. Here, fh_len is used to indirectly confirm that the value of handle_bytes is greater than 3 before accessing parent_block. [1] BUG: KASAN: slab-out-of-bounds in isofs_fh_to_parent+0x1b8/0x210 fs/isofs/export.c:183 Read of size 4 at addr ffff0000cc030d94 by task syz-executor215/6466 CPU: 1 UID: 0 PID: 6466 Comm: syz-executor215 Not tainted 6.14.0-rc7-syzkaller-ga2392f333575 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 Call trace: show_stack+0x2c/0x3c arch/arm64/kernel/stacktrace.c:466 (C) __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0xe4/0x150 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0x198/0x550 mm/kasan/report.c:521 kasan_report+0xd8/0x138 mm/kasan/report.c:634 __asan_report_load4_noabort+0x20/0x2c mm/kasan/report_generic.c:380 isofs_fh_to_parent+0x1b8/0x210 fs/isofs/export.c:183 exportfs_decode_fh_raw+0x2dc/0x608 fs/exportfs/expfs.c:523 do_handle_to_path+0xa0/0x198 fs/fhandle.c:257 handle_to_path fs/fhandle.c:385 [inline] do_handle_open+0x8cc/0xb8c fs/fhandle.c:403 __do_sys_open_by_handle_at fs/fhandle.c:443 [inline] __se_sys_open_by_handle_at fs/fhandle.c:434 [inline] __arm64_sys_open_by_handle_at+0x80/0x94 fs/fhandle.c:434 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc+0x54/0x168 arch/arm64/kernel/entry-common.c:744 el0t_64_sync_handler+0x84/0x108 arch/arm64/kernel/entry-common.c:762 el0t_64_sync+0x198/0x19c arch/arm64/kernel/entry.S:600 Allocated by task 6466: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x40/0x78 mm/kasan/common.c:68 kasan_save_alloc_info+0x40/0x50 mm/kasan/generic.c:562 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0xac/0xc4 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __do_kmalloc_node mm/slub.c:4294 [inline] __kmalloc_noprof+0x32c/0x54c mm/slub.c:4306 kmalloc_noprof include/linux/slab.h:905 [inline] handle_to_path fs/fhandle.c:357 [inline] do_handle_open+0x5a4/0xb8c fs/fhandle.c:403 __do_sys_open_by_handle_at fs/fhandle.c:443 [inline] __se_sys_open_by_handle_at fs/fhandle.c:434 [inline] __arm64_sys_open_by_handle_at+0x80/0x94 fs/fhandle.c:434 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc+0x54/0x168 arch/arm64/kernel/entry-common.c:744 el0t_64_sync_handler+0x84/0x108 arch/arm64/kernel/entry-common.c:762 el0t_64_sync+0x198/0x19c arch/arm64/kernel/entry.S:600

CVE ID	Description
CVE-2025-37781	In the Linux kernel, the following vulnerability has been resolved: i2c: cros-ec-tunnel: defer probe if parent EC is not present When i2c-cros-ec-tunnel and the EC driver are built-in, the EC parent device will not be found, leading to NULL pointer dereference. That can also be reproduced by unbinding the controller driver and then loading i2c-cros-ec-tunnel module (or binding the device). [271.991245] BUG: kernel NULL pointer dereference, address: 0000000000000058 [271.998215] #PF: supervisor read access in kernel mode [272.003351] #PF: error_code(0x0000) - not-present page [272.008485] PGD 0 P4D 0 [272.011022] Oops: Oops: 0000 [#1] SMP NOPTI [272.015207] CPU: 0 UID: 0 PID: 3859 Comm: insmod Tainted: G S 6.15.0-rc1-00004-g44722359ed83 #30 PREEMPT(full) 3c7fb39a552e7d949de2ad921a7d6588d3a4fdc5 [272.030312] Tainted: [S]=CPU_OUT_OF_SPEC [272.034233] Hardware name: HP Berknip/Berknip, BIOS Google_Berknip.13434.356.0 05/17/2021 [272.042400] RIP: 0010:ec_i2c_probe+0x2b/0x1c0 [i2c_cros_ec_tunnel] [272.048577] Code: 1f 44 00 00 41 57 41 56 41 55 41 54 53 48 83 ec 10 65 48 8b 05 06 a0 6c e7 48 89 44 24 08 4c 8d 7f 10 48 8b 47 50 4c 8b 60 78 <49> 83 7c 24 58 00 0f 84 2f 01 00 00 48 89 fb be 30 06 00 00 4c 9 [272.067317] RSP: 0018:ffffa32082a03940 EFLAGS: 00010282 [272.072541] RAX: ffff969580b6a810 RBX: ffff969580b68c10 RCX: 0000000000000000 [272.079672] RDX: 0000000000000000 RSI: 0000000000000282 RDI: ffff969580b68c00 [272.086804] RBP: 00000000fffffdfb R08: 0000000000000000 R09: 0000000000000000 [272.093936] R10: 0000000000000000 R11: ffffffff06000000 R12: 0000000000000000 [272.101067] R13: ffffffff6666bb8 R14: ffffffff05b5528 R15: ffff969580b68c10 [272.108198] FS: 00007b930906fc40(0000) GS:ffff969603149000(0000) knlGS:0000000000000000 [272.116282] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [272.122024] CR2: 0000000000000058 CR3: 000000012631c000 CR4: 000000000003506f0 [272.129155] Call Trace: [272.131606] <TASK> [272.133709] ? acpi_dev_pm_attach +0xdd/0x110 [272.137985] platform_probe+0x69/0xa0 [272.141652] really_probe+0x152/0x310 [272.145318] __driver_probe_device+0x77/0x110 [272.149678] driver_probe_device+0x1e/0x190 [272.153864] __driver_attach+0x10b/0x1e0 [272.157790] ? driver_attach+0x20/0x20 [272.161542] bus_for_each_dev+0x107/0x150 [272.165553] bus_add_driver+0x15d/0x270 [272.169392] driver_register+0x65/0x110 [272.173232] ? cleanup_module+0xa80/0xa80 [i2c_cros_ec_tunnel 3a00532f3f4af4a9eade753f86b0f8dd4e4e5698] [272.182617] do_one_initcall+0x110/0x350 [272.186543] ? security_kernfs_init_security+0x49/0xd0 [272.191682] ? __kernfs_new_node +0x1b9/0x240 [272.195954] ? security_kernfs_init_security+0x49/0xd0 [272.201093] ? __kernfs_new_node+0x1b9/0x240 [272.205365] ? kernfs_link_sibling+0x105/0x130 [272.209810] ? kernfs_next_descendant_post+0x1c/0xa0 [272.214773] ? kernfs_activate+0x57/0x70 [272.218699] ? kernfs_add_one+0x118/0x160 [272.222710] ? __kernfs_create_file+0x71/0xa0 [272.227069] ? sysfs_add_bin_file_mode_ns+0xd6/0x110 [272.232033] ? internal_create_group+0x453/0x4a0 [272.236651] ? __vunmap_range_noflush+0x214/0x2d0 [272.241355] ? __free_frozen_pages +0x1dc/0x420 [272.245799] ? free_vmap_area_noflush+0x10a/0x1c0 [272.250505] ? load_module+0x1509/0x16f0 [272.254431] do_init_module+0x60/0x230 [272.258181] __se_sys_finit_module+0x27a/0x370 [272.262627] do_syscall_64+0x6a/0xf0 [272.266206] ? do_syscall_64+0x76/0xf0 [272.269956] ? irqentry_exit_to_user_mode+0x79/0x90 [272.274836] entry_SYSCALL_64_after_hwframe+0x55/0x5d [272.279887] RIP: 0033:0x7b9309168d39 [272.283466] Code: 5b 41 5c 5d c3 66 2e 0f 1f 84 00 00 00 00 66 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d af 40 0c 00 f7 d8 64 89 01 8 [272.302210] RSP: 002b:00007fff50f1a288 EFLAGS: 00000246 ORIG_RAX: 000 --- truncated---
CVE-2025-37784	In the Linux kernel, the following vulnerability has been resolved: net: ti: icss-iep: Fix possible NULL pointer dereference for perout request The ICSS IEP driver tracks perout and pps enable state with flags. Currently when disabling pps and perout signals during icss_iep_exit(), results in NULL pointer dereference for perout. To fix the null pointer dereference issue, the icss_iep_perout_enable_hw function can be modified to directly clear the IEP CMP registers when disabling PPS or PEROUT, without referencing the ptp_perout_request structure, as its contents are irrelevant in this case.

CVE ID	Description
CVE-2025-37785	In the Linux kernel, the following vulnerability has been resolved: ext4: fix OOB read when checking dotdot dir Mounting a corrupted filesystem with directory which contains '.' dir entry with rec_len == block size results in out-of-bounds read (later on, when the corrupted directory is removed). ext4_empty_dir() assumes every ext4 directory contains at least '.' and '.' as directory entries in the first data block. It first loads the '.' dir entry, performs sanity checks by calling ext4_check_dir_entry() and then uses its rec_len member to compute the location of '.' dir entry (in ext4_next_entry). It assumes the '.' dir entry fits into the same data block. If the rec_len of '.' is precisely one block (4KB), it slips through the sanity checks (it is considered the last directory entry in the data block) and leaves "struct ext4_dir_entry_2 *de" point exactly past the memory slot allocated to the data block. The following call to ext4_check_dir_entry() on new value of de then dereferences this pointer which results in out-of-bounds mem access. Fix this by extending __ext4_check_dir_entry() to check for '.' dir entries that reach the end of data block. Make sure to ignore the phony dir entries for checksum (by checking name_len for non-zero). Note: This is reported by KASAN as use-after-free in case another structure was recently freed from the slot past the bound, but it is really an OOB read. This issue was found by syzkaller tool. Call Trace: [38.594108] BUG: KASAN: slab-use-after-free in __ext4_check_dir_entry+0x67e/0x710 [38.594649] Read of size 2 at addr ffff88802b41a004 by task syz-executor/5375 [38.595158] [38.595288] CPU: 0 UID: 0 PID: 5375 Comm: syz-executor Not tainted 6.14.0-rc7 #1 [38.595298] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 [38.595304] Call Trace: [38.595308] <TASK> [38.595311] dump_stack_lvl+0xa7/0xd0 [38.595325] print_address_description.constprop.0+0x2c/0x3f0 [38.595339] ? __ext4_check_dir_entry+0x67e/0x710 [38.595349] print_report+0xaa/0x250 [38.595359] ? __ext4_check_dir_entry+0x67e/0x710 [38.595368] ? kasan_addr_to_slab+0x9/0x90 [38.595378] kasan_report+0xab/0xe0 [38.595389] ? __ext4_check_dir_entry+0x67e/0x710 [38.595400] __ext4_check_dir_entry+0x67e/0x710 [38.595410] ext4_empty_dir+0x465/0x990 [38.595421] ? __pfx_ext4_empty_dir+0x10/0x10 [38.595432] ext4_rmdir.part.0+0x29a/0xd10 [38.595441] ? __dquot_initialize+0x2a7/0xbf0 [38.595455] ? __pfx_ext4_rmdir.part.0+0x10/0x10 [38.595464] ? __pfx__dquot_initialize+0x10/0x10 [38.595478] ? down_write+0xdb/0x140 [38.595487] ? __pfx_down_write+0x10/0x10 [38.595497] ext4_rmdir+0xee/0x140 [38.595506] vfs_rmdir+0x209/0x670 [38.595517] ? lookup_one_qstr_excl+0x3b/0x190 [38.595529] do_rmdir+0x363/0x3c0 [38.595537] ? __pfx_do_rmdir+0x10/0x10 [38.595544] ? strncpy_from_user+0x1ff/0x2e0 [38.595561] __x64_sys_unlinkat+0xf0/0x130 [38.595570] do_syscall_64+0x5b/0x180 [38.595583] entry_SYSCALL_64_after_hwframe+0x76/0x7e
CVE-2025-37786	In the Linux kernel, the following vulnerability has been resolved: net: dsa: free routing table on probe failure If complete = true in dsa_tree_setup(), it means that we are the last switch of the tree which is successfully probing, and we should be setting up all switches from our probe path. After "complete" becomes true, dsa_tree_setup_cpu_ports() or any subsequent function may fail. If that happens, the entire tree setup is in limbo: the first N-1 switches have successfully finished probing (doing nothing but having allocated persistent memory in the tree's dst->ports, and maybe dst->rtable), and switch N failed to probe, ending the tree setup process before anything is tangible from the user's PoV. If switch N fails to probe, its memory (ports) will be freed and removed from dst->ports. However, the dst->rtable elements pointing to its ports, as created by dsa_link_touch(), will remain there, and will lead to use-after-free if dereferenced. If dsa_tree_setup_switches() returns -EPROBE_DEFER, which is entirely possible because that is where ds->ops->setup() is, we get a kasan report like this: ===== BUG: KASAN: slab-use-after-free in mv88e6xxx_setup_upstream_port+0x240/0x568 Read of size 8 at addr ffff000004f56020 by task kworker/u8:3/42 Call trace: __asan_report_load8_noabort+0x20/0x30 mv88e6xxx_setup_upstream_port+0x240/0x568 mv88e6xxx_setup+0xebc/0x1eb0 dsa_register_switch+0x1af4/0x2ae0 mv88e6xxx_register_switch+0x1b8/0x2a8 mv88e6xxx_probe+0xc4c/0xf60 mdio_probe+0x78/0xb8 really_probe+0x2b8/0x5a8 __driver_probe_device+0x164/0x298 driver_probe_device+0x78/0x258 __device_attach_driver+0x274/0x350 Allocated by task 42: __kasan_kmalloc+0x84/0xa0 __kmalloc_cache_noprof+0x298/0x490 dsa_switch_touch_ports+0x174/0x3d8 dsa_register_switch+0x800/0x2ae0 mv88e6xxx_register_switch+0x1b8/0x2a8 mv88e6xxx_probe+0xc4c/0xf60 mdio_probe+0x78/0xb8 really_probe+0x2b8/0x5a8 __driver_probe_device+0x164/0x298 driver_probe_device+0x78/0x258 __device_attach_driver+0x274/0x350 Freed by task 42: __kasan_slab_free+0x48/0x68 kfree+0x138/0x418 dsa_register_switch+0x2694/0x2ae0 mv88e6xxx_register_switch+0x1b8/0x2a8 mv88e6xxx_probe+0xc4c/0xf60 mdio_probe+0x78/0xb8 really_probe+0x2b8/0x5a8 __driver_probe_device+0x164/0x298 driver_probe_device+0x78/0x258 __device_attach_driver+0x274/0x350 The simplest way to fix the bug is to delete the routing table in its entirety. dsa_tree_setup_routing_table() has no problem in regenerating it even if we deleted links between ports other than those of switch N, because dsa_link_touch() first checks whether the port pair already exists in dst->rtable, allocating if not. The deletion of the routing table in its entirety already exists in dsa_tree_teardown(), so refactor that into a function that can also be called from the tree setup error path. In my analysis of the commit to blame, it is the one which added dsa_link elements to dst->rtable. Prior to that, each switch had its own ds->rtable which is freed when the switch fails to probe. But the tree is potentially persistent memory.

CVE ID	Description
CVE-2025-37787	In the Linux kernel, the following vulnerability has been resolved: net: dsa: mv88e6xxx: avoid unregistering devlink regions which were never registered Russell King reports that a system with mv88e6xxx dereferences a NULL pointer when unbinding this driver: https://lore.kernel.org/netdev/Z_JRkMITJ1KQ0kVX@shell.armlinux.org.uk/ The crash seems to be in devlink_region_destroy(), which is not NULL tolerant but is given a NULL devlink global region pointer. At least on some chips, some devlink regions are conditionally registered since the blamed commit, see mv88e6xxx_setup_devlink_regions_global(): if (cond && !cond(chip)) continue; These are MV88E6XXX_REGION_STU and MV88E6XXX_REGION_PVT. If the chip does not have an STU or PVT, it should crash like this. To fix the issue, avoid unregistering those regions which are NULL, i.e. were skipped at mv88e6xxx_setup_devlink_regions_global() time.
CVE-2025-37788	In the Linux kernel, the following vulnerability has been resolved: cxgb4: fix memory leak in cxgb4_init_ethtool_filters() error path In the for loop used to allocate the loc_array and bmap for each port, a memory leak is possible when the allocation for loc_array succeeds, but the allocation for bmap fails. This is because when the control flow goes to the label free_eth_finfo, only the allocations starting from (i-1)th iteration are freed. Fix that by freeing the loc_array in the bmap allocation error path.
CVE-2025-37789	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: fix nested key length validation in the set() action It's not safe to access nla_len(ovs_key) if the data is smaller than the netlink header. Check that the attribute is OK first.
CVE-2025-37790	In the Linux kernel, the following vulnerability has been resolved: net: mctp: Set SOCK_RCU_FREE Bind lookup runs under RCU, so ensure that a socket doesn't go away in the middle of a lookup.
CVE-2025-37792	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btrtl: Prevent potential NULL dereference The btrtl_initialize() function checks that rtl_load_file() either had an error or it loaded a zero length file. However, if it loaded a zero length file then the error code is not set correctly. It results in an error pointer vs NULL bug, followed by a NULL pointer dereference. This was detected by Smatch: drivers/bluetooth/btrtl.c:592 btrtl_initialize() warn: passing zero to 'ERR_PTR'
CVE-2025-37793	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: avs: Fix null-ptr-deref in avs_component_probe() devm_kasprintf() returns NULL when memory allocation fails. Currently, avs_component_probe() does not check for this case, which results in a NULL pointer dereference.

CVE ID	Description
CVE-2025-37794	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Purge vif txq in ieee80211_do_stop() After ieee80211_do_stop() SKB from vif's txq could still be processed. Indeed another concurrent vif schedule_and_wake_txq call could cause those packets to be dequeued (see ieee80211_handle_wake_tx_queue()) without checking the sdata current state. Because vif.drv_priv is now cleared in this function, this could lead to driver crash. For example in ath12k, ahvif is store in vif.drv_priv. Thus if ath12k_mac_op_tx() is called after ieee80211_do_stop(), ahvif->ah can be NULL, leading the ath12k_warn(ahvif->ah,...) call in this function to trigger the NULL deref below. Unable to handle kernel paging request at virtual address dffffc0000000001 KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] batman_adv: bat0: Interface deactivated: brbh1337 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [dffffc0000000001] address between user and kernel address ranges Internal error: Oops: 0000000096000004 [#1] SMP CPU: 1 UID: 0 PID: 978 Comm: lbd Not tainted 6.13.0-g633f875b8f1e #114 Hardware name: HW (DT) pstate: 10000005 (nzcV daif -PAN -UAO -TCO -DIT -SSBS BTYP=--): pc : ath12k_mac_op_tx +0x6cc/0x29b8 [ath12k] lr : ath12k_mac_op_tx+0x174/0x29b8 [ath12k] sp : fffffc086ace450 x29: fffffc086ace450 x28: 0000000000000000 x27: 1fffff81d05f7c0 x26: ffffff801d05f7c0 x25: 0000000000000000 x24: 000000004000001e x23: ffffff8009ce4926 x22: ffffff801f9c0800 x21: ffffff801d05f7f0 x20: ffffff8034a19f40 x19: 0000000000000000 x18: ffffff801f9c0958 x17: ffffff800bc0a504 x16: dffffc0000000000 x15: fffffc086ace4f8 x14: ffffff801d05f83c x13: 0000000000000000 x12: fffffb003a0bf03 x11: 0000000000000000 x10: fffffb003a0bf02 x9 : ffffff8034a19f40 x8 : ffffff801d05f818 x7 : 1fffff0069433dc x6 : ffffff8034a19ee0 x5 : ffffff801d05f7f0 x4 : 0000000000000000 x3 : 0000000000000001 x2 : 0000000000000000 x1 : dffffc0000000000 x0 : 0000000000000008 Call trace: ath12k_mac_op_tx+0x6cc/0x29b8 [ath12k] (P) ieee80211_handle_wake_tx_queue+0x16c/0x260 ieee80211_queue_skb+0x0ec/0x1d20 ieee80211_tx+0x200/0x2c8 ieee80211_xmit+0x22c/0x338 ___ieee80211_subif_start_xmit+0x7e8/0xc60 ieee80211_subif_start_xmit+0xc4/0xee0 ___ieee80211_subif_start_xmit_8023.isra.0+0x854/0x17a0 ieee80211_subif_start_xmit_8023+0x124/0x488 dev_hard_start_xmit+0x160/0x5a8 ___dev_queue_xmit+0x6f8/0x3120 br_dev_queue_push_xmit+0x120/0x4a8 ___br_forward+0xe4/0x2b0 deliver_clone+0x5c/0xd0 br_flood+0x398/0x580 br_dev_xmit+0x454/0x9f8 dev_hard_start_xmit+0x160/0x5a8 ___dev_queue_xmit+0x6f8/0x3120 ip6_finish_output2+0xc28/0x1b60 ___ip6_finish_output+0x38c/0x638 ip6_output+0x1b4/0x338 ip6_local_out+0x7c/0xa8 ip6_send_skb+0x7c/0x1b0 ip6_push_pending_frames+0x94/0xd0 rawv6_sendmsg+0x1a98/0x2898 inet_sendmsg+0x94/0xe0 ___sys_sendto+0x1e4/0x308 ___arm64_sys_sendto+0xc4/0x140 do_el0_svc+0x110/0x280 el0_svc+0x20/0x60 el0t_64_sync_handler+0x104/0x138 el0t_64_sync+0x154/0x158 To avoid that, empty vif's txq at ieee80211_do_stop() so no packet could be dequeued after ieee80211_do_stop() (new packets cannot be queued because SDATA_STATE_RUNNING is cleared at this point).
CVE-2025-37796	In the Linux kernel, the following vulnerability has been resolved: wifi: at76c50x: fix use after free access in at76_disconnect The memory pointed to by priv is freed at the end of at76_delete_device function (using ieee80211_free_hw). But the code then accesses the udev field of the freed object to put the USB device. This may also lead to a memory leak of the usb device. Fix this by using udev from interface.
CVE-2025-37797	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Fix a UAF vulnerability in class handling This patch fixes a Use-After-Free vulnerability in the HFSC qdisc class handling. The issue occurs due to a time-of-check/time-of-use condition in hfsc_change_class() when working with certain child qdiscs like netem or code1. The vulnerability works as follows: 1. hfsc_change_class() checks if a class has packets (q.qlen != 0) 2. It then calls qdisc_peek_len(), which for certain qdiscs (e.g., code1, netem) might drop packets and empty the queue 3. The code continues assuming the queue is still non-empty, adding the class to vtree 4. This breaks HFSC scheduler assumptions that only non-empty classes are in vtree 5. Later, when the class is destroyed, this can lead to a Use-After-Free The fix adds a second queue length check after qdisc_peek_len() to verify the queue wasn't emptied.
CVE-2025-37798	In the Linux kernel, the following vulnerability has been resolved: code1: remove sch->q.qlen check before qdisc_tree_reduce_backlog() After making all ->q.qlen_notify() callbacks idempotent, now it is safe to remove the check of qlen!=0 from both fq_code1_dequeue() and code1_qdisc_dequeue().

CVE ID	Description
CVE-2025-37799	In the Linux kernel, the following vulnerability has been resolved: vmxnet3: Fix malformed packet sizing in vmxnet3_process_xdp vmxnet3 driver's XDP handling is buggy for packet sizes using ring0 (that is, packet sizes between 128 - 3k bytes). We noticed MTU-related connectivity issues with Cilium's service load- balancing in case of vmxnet3 as NIC underneath. A simple curl to a HTTP backend service where the XDP LB was doing IPID encaps led to overly large packet sizes but only for *some* of the packets (e.g. HTTP GET request) while others (e.g. the prior TCP 3WHS) looked completely fine on the wire. In fact, the pcap recording on the backend node actually revealed that the node with the XDP LB was leaking uninitialized kernel data onto the wire for the affected packets, for example, while the packets should have been 152 bytes their actual size was 1482 bytes, so the remainder after 152 bytes was padded with whatever other data was in that page at the time (e.g. we saw user/payload data from prior processed packets). We only noticed this through an MTU issue, e.g. when the XDP LB node and the backend node both had the same MTU (e.g. 1500) then the curl request got dropped on the backend node's NIC given the packet was too large even though the IPID-encapped packet normally would never even come close to the MTU limit. Lowering the MTU on the XDP LB (e.g. 1480) allowed to let the curl request succeed (which also indicates that the kernel ignored the padding, and thus the issue wasn't very user-visible). Commit e127ce7699c1 ("vmxnet3: Fix missing reserved tailroom") was too eager to also switch xdp_prepare_buff() from rcd->len to rbi->len. It really needs to stick to rcd->len which is the actual packet length from the descriptor. The latter we also feed into vmxnet3_process_xdp_small(), by the way, and it indicates the correct length needed to initialize the xdp->{data,data_end} parts. For e127ce7699c1 ("vmxnet3: Fix missing reserved tailroom") the relevant part was adapting xdp_init_buff() to address the warning given the xdp_data_hard_end() depends on xdp->frame_sz. With that fixed, traffic on the wire looks good again.
CVE-2025-37800	In the Linux kernel, the following vulnerability has been resolved: driver core: fix potential NULL pointer dereference in dev_uevent() If userspace reads "uevent" device attribute at the same time as another threads unbinds the device from its driver, change to dev->driver from a valid pointer to NULL may result in crash. Fix this by using READ_ONCE() when fetching the pointer, and take bus' drivers klist lock to make sure driver instance will not disappear while we access it. Use WRITE_ONCE() when setting the driver pointer to ensure there is no tearing.
CVE-2025-37801	In the Linux kernel, the following vulnerability has been resolved: spi: spi-imx: Add check for spi_imx_setupxfer() Add check for the return value of spi_imx_setupxfer(). spi_imx->rx and spi_imx->tx function pointer can be NULL when spi_imx_setupxfer() return error, and make NULL pointer dereference. Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Call trace: 0x0 spi_imx_pio_transfer+0x50/0xd8 spi_imx_transfer_one +0x18c/0x858 spi_transfer_one_message+0x43c/0x790 __spi_pump_transfer_message+0x238/0x5d4 __spi_sync+0x2b0/0x454 spi_write_then_read+0x11c/0x200
CVE-2025-37802	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix WARNING "do not call blocking ops when !TASK_RUNNING" wait_event_timeout() will set the state of the current task to TASK_UNINTERRUPTIBLE, before doing the condition check. This means that ksmbd_durable_scavenger_alive() will try to acquire the mutex while already in a sleeping state. The scheduler warns us by giving the following warning: do not call blocking ops when !TASK_RUNNING; state=2 set at [<0000000061515a6f>] prepare_to_wait_event+0x9f/0x6c0 WARNING: CPU: 2 PID: 4147 at kernel/sched/core.c:10099 __might_sleep+0x12f/0x160 mutex lock is not needed in ksmbd_durable_scavenger_alive().
CVE-2025-37803	In the Linux kernel, the following vulnerability has been resolved: udmabuf: fix a buf size overflow issue during udmabuf creation by casting size_limit_mb to u64 when calculate pglimit.
CVE-2025-37805	In the Linux kernel, the following vulnerability has been resolved: sound/virtio: Fix cancel_sync warnings on uninitialized work_structs Betty reported hitting the following warning: [8.709131] [T221] WARNING: CPU: 2 PID: 221 at kernel/workqueue.c:4182 ... [8.713282][T221] Call trace: [8.713365][T221] __flush_work+0x8d0/0x914 [8.713468][T221] __cancel_work_sync+0xac/0xfc [8.713570][T221] cancel_work_sync+0x24/0x34 [8.713667][T221] virtsnd_remove+0xa8/0xf8 [virtio_snd ab15f34d0dd772f6d11327e08a81d46dc9c36276] [8.713868][T221] virtsnd_probe +0x48c/0x664 [virtio_snd ab15f34d0dd772f6d11327e08a81d46dc9c36276] [8.714035][T221] virtio_dev_probe+0x28c/0x390 [8.714139][T221] really_probe+0x1bc/0x4c8 ... It seems we're hitting the error path in virtsnd_probe(), which triggers a virtsnd_remove() which iterates over the substreams calling cancel_work_sync() on the elapsed_period work_struct. Looking at the code, from earlier in: virtsnd_probe()->virtsnd_build_devs()->virtsnd_pcm_parse_cfg() We set snd->nsubstreams, allocate the snd->substreams, and if we then hit an error on the info allocation or something in virtsnd_ctl_query_info() fails, we will exit without having initialized the elapsed_period work_struct. When that error path unwinds we then call virtsnd_remove() which as long as the substreams array is allocated, will iterate through calling cancel_work_sync() on the uninitialized work struct hitting this warning. Takashi Iwai suggested this fix, which initializes the substreams structure right after allocation, so that if we hit the error paths we avoid trying to cleanup uninitialized data. Note: I have not yet managed to reproduce the issue myself, so this patch has had limited testing. Feedback or thoughts would be appreciated!

CVE ID	Description
CVE-2025-37806	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Keep write operations atomic syzbot reported a NULL pointer dereference in __generic_file_write_iter. [1] Before the write operation is completed, the user executes ioctl[2] to clear the compress flag of the file, which causes the is_compressed() judgment to return 0, further causing the program to enter the wrong process and call the wrong ops ntfs_aops_cmpr, which triggers the null pointer dereference of write_begin. Use inode lock to synchronize ioctl and write to avoid this case. [1] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x0000000086000006 EC = 0x21: IABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x06: level 2 translation fault user pgtbl: 4k pages, 48-bit VAs, pgdp=000000011896d000 [0000000000000000] pgd=0800000118b44403, p4d=0800000118b44403, pud=0800000117517403, pmd=0000000000000000 Internal error: Oops: 0000000086000006 [#1] PREEMPT SMP Modules linked in: CPU: 0 UID: 0 PID: 6427 Comm: syz-executor347 Not tainted 6.13.0-rc3-syzkaller-g573067a5a685 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=) pc : 0x0 lr : generic_perform_write+0x29c/0x868 mm/filemap.c:4055 sp : ffff80009d4978a0 x29: ffff80009d4979c0 x28: dfff800000000000 x27: ffff80009d497bc8 x26: 0000000000000000 x25: ffff80009d497960 x24: ffff80008ba71c68 x23: 0000000000000000 x22: ffff0000c655dac0 x21: 00000000000001000 x20: 0000000000000000c x19: 1ffff0013a92f2c x18: ffff0000e183aa1c x17: 00040600000000014 x16: ffff800083275834 x15: 0000000000000001 x14: 0000000000000000 x13: 0000000000000001 x12: ffff0000c655dac0 x11: 0000000000ff0100 x10: 0000000000ff0100 x9 : 0000000000000000 x8 : 0000000000000000 x7 : 0000000000000000 x6 : 0000000000000000 x5 : ffff80009d497980 x4 : ffff80009d497960 x3 : 00000000000001000 x2 : 0000000000000000 x1 : ffff0000e183a928 x0 : ffff0000d60b0fc0 Call trace: 0x0 (P) __generic_file_write_iter+0x4c/0x204 mm/filemap.c:4156 ntfs_file_write_iter+0x54c/0x630 fs/ntfs3/file.c:1267 new_sync_write fs/read_write.c:586 [inline] vfs_write+0x920/0xc4 fs/read_write.c:679 ksys_write+0x15c/0x26c fs/read_write.c:731 __do_sys_write fs/read_write.c:742 [inline] __se_sys_write fs/read_write.c:739 [inline] __arm64_sys_write+0x7c/0x90 fs/read_write.c:739 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc+0x54/0x168 arch/arm64/kernel/entry-common.c:744 el0t_64_sync_handler+0x84/0x108 arch/arm64/kernel/entry-common.c:762 [2] ioctl\$FS_IOC_SETFLAGS(r0, 0x40086602, &(0x7f00000000c0)=0x20)
CVE-2025-37807	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix kmemleak warning for percpu hashmap Vlad Poenaru reported the following kmemleak issue: unreferenced object 0x606fd7c44ac8 (size 32): backtrace (crc 0): pcpu_alloc_noprof+0x730/0xeb0 bpf_map_alloc_percpu+0x69/0xc0 prealloc_init+0x9d/0x1b0 htab_map_alloc+0x363/0x510 map_create+0x215/0x3a0 __sys_bpf+0x16b/0x3e0 __x64_sys_bpf+0x18/0x20 do_syscall_64+0x7b/0x150 entry_SYSCALL_64_after_hwframe+0x4b/0x53 Further investigation shows the reason is due to not 8-byte aligned store of percpu pointer in htab_elem_set_ptr(): *(void __percpu **)(l->key + key_size) = pptr; Note that the whole htab_elem alignment is 8 (for x86_64). If the key_size is 4, that means pptr is stored in a location which is 4 byte aligned but not 8 byte aligned. In mm/kmemleak.c, scan_block() scans the memory based on 8 byte stride, so it won't detect above pptr, hence reporting the memory leak. In htab_map_alloc(), we already have htab->elem_size = sizeof(struct htab_elem) + round_up(htab->map.key_size, 8); if (percpu) htab->elem_size += sizeof(void *); else htab->elem_size += round_up(htab->map.value_size, 8); So storing pptr with 8-byte alignment won't cause any problem and can fix kmemleak too. The issue can be reproduced with bpf selftest as well: 1. Enable CONFIG_DEBUG_KMEMLEAK config 2. Add a getchar() before skel destroy in test_hash_map() in prog_tests/for_each.c. The purpose is to keep map available so kmemleak can be detected. 3. run './test_progs -t for_each/hash_map &' and a kmemleak should be reported.
CVE-2025-37808	In the Linux kernel, the following vulnerability has been resolved: crypto: null - Use spin lock instead of mutex As the null algorithm may be freed in softirq context through af_alg, use spin locks instead of mutexes to protect the default null algorithm.
CVE-2025-37809	In the Linux kernel, the following vulnerability has been resolved: usb: typec: class: Fix NULL pointer access Concurrent calls to typec_partner_unlink_device can lead to a NULL pointer dereference. This patch adds a mutex to protect USB device pointers and prevent this issue. The same mutex protects both the device pointers and the partner device registration.
CVE-2025-37810	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: gadget: check that event count does not exceed event buffer length The event count is read from register DWC3_GEVNTPCOUNT. There is a check for the count being zero, but not for exceeding the event buffer length. Check that event count does not exceed event buffer length, avoiding an out-of-bounds access when memcpy'ing the event. Crash log: Unable to handle kernel paging request at virtual address ffffffc0129be000 pc : __memcpy+0x114/0x180 lr : dwc3_check_event_buf+0xec/0x348 x3 : 0000000000000030 x2 : 0000000000000dfc4 x1 : ffffffc0129be000 x0 : ffffff87aad60080 Call trace: __memcpy+0x114/0x180 dwc3_interrupt+0x24/0x34

CVE ID	Description
CVE-2025-37811	In the Linux kernel, the following vulnerability has been resolved: usb: chipidea: ci_hdrc_imx: fix usbmisc handling usbmisc is an optional device property so it is totally valid for the corresponding data->usbmisc_data to have a NULL value. Check that before dereferencing the pointer. Found by Linux Verification Center (linuxtesting.org) with Svmc static analysis tool.
CVE-2025-37812	In the Linux kernel, the following vulnerability has been resolved: usb: cdns3: Fix deadlock when using NCM gadget The cdns3 driver has the same NCM deadlock as fixed in cdnsp by commit 58f2fcb3a845 ("usb: cdnsp: Fix deadlock issue during using NCM gadget"). Under PREEMPT_RT the deadlock can be readily triggered by heavy network traffic, for example using "iperf --bidir" over NCM ethernet link. The deadlock occurs because the threaded interrupt handler gets preempted by a softirq, but both are protected by the same spinlock. Prevent deadlock by disabling softirq during threaded irq handler.
CVE-2025-37813	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Fix invalid pointer dereference in Etron workaround This check is performed before prepare_transfer() and prepare_ring(), so enqueue can already point at the final link TRB of a segment. And indeed it will, some 0.4% of times this code is called. Then enqueue + 1 is an invalid pointer. It will crash the kernel right away or load some junk which may look like a link TRB and cause the real link TRB to be replaced with a NOOP. This wouldn't end well. Use a functionally equivalent test which doesn't dereference the pointer and always gives correct result. Something has crashed my machine twice in recent days while playing with an Etron HC, and a control transfer stress test ran for confirmation has just crashed it again. The same test passes with this patch applied.
CVE-2025-37815	In the Linux kernel, the following vulnerability has been resolved: misc: microchip: pci1xxxx: Fix Kernel panic during IRQ handler registration Resolve kernel panic while accessing IRQ handler associated with the generated IRQ. This is done by acquiring the spinlock and storing the current interrupt state before handling the interrupt request using generic_handle_irq. A previous fix patch was submitted where 'generic_handle_irq' was replaced with 'handle_nested_irq'. However, this change also causes the kernel panic where after determining which GPIO triggered the interrupt and attempting to call handle_nested_irq with the mapped IRQ number, leads to a failure in locating the registered handler.
CVE-2025-37816	In the Linux kernel, the following vulnerability has been resolved: mei: vsc: Fix fortify-panic caused by invalid counted_by() use gcc 15 honors the __counted_by(len) attribute on vsc_tp_packet.buf[] and the vsc-tp.c code is using this in a wrong way. len does not contain the available size in the buffer, it contains the actual packet length *without* the crc. So as soon as vsc_tp_xfer() tries to add the crc to buf[] the fortify-panic handler gets triggered: [80.842193] memcpy: detected buffer overflow: 4 byte write of buffer size 0 [80.842243] WARNING: CPU: 4 PID: 272 at lib/string_helpers.c:1032 __fortify_report+0x45/0x50 ... [80.843175] __fortify_panic+0x9/0xb [80.843186] vsc_tp_xfer.cold +0x67/0x67 [mei_vsc_hw] [80.843210] ? seqcount_lockdep_reader_access.constprop.0+0x82/0x90 [80.843229] ? lockdep_hardirqs_on+0x7c/0x110 [80.843250] mei_vsc_hw_start+0x98/0x120 [mei_vsc] [80.843270] mei_reset+0x11d/0x420 [mei] The easiest fix would be to just drop the counted-by but with the exception of the ack buffer in vsc_tp_xfer_helper() which only contains enough room for the packet-header, all other uses of vsc_tp_packet always use a buffer of VSC_TP_MAX_XFER_SIZE bytes for the packet. Instead of just dropping the counted-by, split the vsc_tp_packet struct definition into a header and a full-packet definition and use a fixed size buf[] in the packet definition, this way fortify-source buffer overrun checking still works when enabled.
CVE-2025-37817	In the Linux kernel, the following vulnerability has been resolved: mcb: fix a double free bug in chameleon_parse_gdd() In chameleon_parse_gdd(), if mcb_device_register() fails, 'mdev' would be released in mcb_device_register() via put_device(). Thus, goto 'err' label and free 'mdev' again causes a double free. Just return if mcb_device_register() fails.
CVE-2025-37819	In the Linux kernel, the following vulnerability has been resolved: irqchip/gic-v2m: Prevent use after free of gicv2m_get_fwnode() With ACPI in place, gicv2m_get_fwnode() is registered with the pci subsystem as pci_msi_get_fwnode_cb(), which may get invoked at runtime during a PCI host bridge probe. But, the call back is wrongly marked as __init, causing it to be freed, while being registered with the PCI subsystem and could trigger: Unable to handle kernel paging request at virtual address ffff8000816c0400 gicv2m_get_fwnode+0x0/0x58 (P) pci_set_bus_msi_domain+0x74/0x88 pci_register_host_bridge+0x194/0x548 This is easily reproducible on a Juno board with ACPI boot. Retain the function for later use.
CVE-2025-37820	In the Linux kernel, the following vulnerability has been resolved: xen-netfront: handle NULL returned by xdp_convert_buff_to_frame() The function xdp_convert_buff_to_frame() may return NULL if it fails to correctly convert the XDP buffer into an XDP frame due to memory constraints, internal errors, or invalid data. Failing to check for NULL may lead to a NULL pointer dereference if the result is used later in processing, potentially causing crashes, data corruption, or undefined behavior. On XDP redirect failure, the associated page must be released explicitly if it was previously retained via get_page(). Failing to do so may result in a memory leak, as the pages reference count is not decremented.

CVE ID	Description
CVE-2025-37822	In the Linux kernel, the following vulnerability has been resolved: riscv: uprobes: Add missing fence.i after building the XOL buffer The XOL (execute out-of-line) buffer is used to single-step the replaced instruction(s) for uprobes. The RISC-V port was missing a proper fence.i (i\$ flushing) after constructing the XOL buffer, which can result in incorrect execution of stale/broken instructions. This was found running the BPF selftests "test_progs: uprobe_autoattach, attach_probe" on the Spacemint K1/X60, where the uprobes tests randomly blew up.
CVE-2025-37823	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Fix a potential UAF in hfsc_dequeue() too Similarly to the previous patch, we need to safe guard hfsc_dequeue() too. But for this one, we don't have a reliable reproducer.
CVE-2025-37824	In the Linux kernel, the following vulnerability has been resolved: tipc: fix NULL pointer dereference in tipc_mon_reinit_self() syzbot reported: tipc: Node number set to 1055423674 Oops: general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000007] CPU: 3 UID: 0 PID: 6017 Comm: kworker/3:5 Not tainted 6.15.0-rc1-syzkaller-00246-g900241a5cc15 #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Workqueue: events tipc_net_finalize_work RIP: 0010:tipc_mon_reinit_self +0x11c/0x210 net/tipc/monitor.c:719 ... RSP: 0018:ffff9000356fb68 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 000000003ee87cba RDX: 0000000000000000 RSI: ffffffff8dbc56a7 RDI: ffff88804c2cc010 RBP: dffffc0000000000 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000007 R13: ffffffbff2111097 R14: ffff88804ead8000 R15: ffff88804ead9010 FS: 0000000000000000(0000) GS:ffff888097ab9000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000f720eb00 CR3: 000000000e182000 CR4: 000000000352ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> tipc_net_finalize +0x10b/0x180 net/tipc/net.c:140 process_one_work+0x9cc/0x1b70 kernel/workqueue.c:3238 process_scheduled_works kernel/workqueue.c:3319 [inline] worker_thread+0x6c8/0xf10 kernel/workqueue.c:3400 kthread+0x3c2/0x780 kernel/kthread.c:464 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:153 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 </TASK> ... RIP: 0010:tipc_mon_reinit_self+0x11c/0x210 net/tipc/monitor.c:719 ... RSP: 0018:ffff9000356fb68 EFLAGS: 00010246 RAX: 0000000000000000 RBX: 0000000000000000 RCX: 000000003ee87cba RDX: 0000000000000000 RSI: ffffffff8dbc56a7 RDI: ffff88804c2cc010 RBP: dffffc0000000000 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000007 R13: ffffffbff2111097 R14: ffff88804ead8000 R15: ffff88804ead9010 FS: 0000000000000000(0000) GS:ffff888097ab9000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000f720eb00 CR3: 000000000e182000 CR4: 000000000352ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 There is a racing condition between workqueue created when enabling bearer and another thread created when disabling bearer right after that as follow: enabling_bearer disabling_bearer ----- ----- tipc_disc_timeout() { bearer_disable() ... { schedule_work(&tn->work); tipc_mon_delete() ... { } ... write_lock_bh(&mon->lock); mon->self = NULL; write_unlock_bh(&mon->lock); ... } tipc_net_finalize_work() } { ... tipc_net_finalize() { ... tipc_mon_reinit_self() { ... write_lock_bh(&mon->lock); mon->self->addr = tipc_own_addr(net); write_unlock_bh(&mon->lock); ... ---truncated---
CVE-2025-37826	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: core: Add NULL check in ufshcd_mcq_compl_pending_transfer() Add a NULL check for the returned hwq pointer by ufshcd_mcq_req_to_hwq(). This is similar to the fix in commit 74736103fb41 ("scsi: ufs: core: Fix ufshcd_abort_one racing issue").

CVE ID	Description
CVE-2025-37827	In the Linux kernel, the following vulnerability has been resolved: btrfs: zoned: return EIO on RAID1 block group write pointer mismatch There was a bug report about a NULL pointer dereference in <code>__btrfs_add_free_space_zoned()</code> that ultimately happens because a conversion from the default metadata profile DUP to a RAID1 profile on two disks. The stack trace has the following signature: BTRFS error (device sdc): zoned: write pointer offset mismatch of zones in raid1 profile BUG: kernel NULL pointer dereference, address: 0000000000000058 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI RIP: 0010: __btrfs_add_free_space_zoned.isra.0+0x61/0x1a0 RSP: 0018:ffffa236b6f3f6d0 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff96c8132f3400 RCX: 0000000000000001 RDX: 0000000010000000 RSI: 0000000000000000 RDI: ffff96c8132f3410 RBP: 0000000010000000 R08: 0000000000000003 R09: 0000000000000000 R10: 0000000000000000 R11: 00000000ffffff R12: 0000000000000000 R13: ffff96c758f65a40 R14: 0000000000000001 R15: 000011aac0000000 FS: 00007fdab1cb2900(0000) GS:ffff96e60ca00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000058 CR3: 00000001a05ae000 CR4: 000000000350ef0 Call Trace: <TASK> ? __die_body.cold+0x19/0x27 ? page_fault_oops+0x15c/0x2f0 ? exc_page_fault+0x7e/0x180 ? asm_exc_page_fault+0x26/0x30 ? __btrfs_add_free_space_zoned.isra.0+0x61/0x1a0 btrfs_add_free_space_async_trimmed+0x34/0x40 btrfs_add_new_free_space+0x107/0x120 btrfs_make_block_group+0x104/0x2b0 btrfs_create_chunk+0x977/0xf20 btrfs_chunk_alloc+0x174/0x510 ? srso_return_thunk+0x5/0x5f btrfs_inc_block_group_ro+0x1b1/0x230 btrfs_relocate_block_group+0x9e/0x410 btrfs_relocate_chunk+0x3f/0x130 btrfs_balance+0x8ac/0x12b0 ? srso_return_thunk+0x5/0x5f ? srso_return_thunk+0x5/0x5f ? __kmalloccache_noprof+0x14c/0x3e0 btrfs_ioctl+0x2686/0x2a80 ? srso_return_thunk+0x5/0x5f ? ioctl_has_perm.constprop.0.isra.0+0xd2/0x120 __x64_sys_ioctl+0x97/0xc0 do_syscall_64+0x82/0x160 ? srso_return_thunk+0x5/0x5f ? __memcg_slab_free_hook+0x11a/0x170 ? srso_return_thunk+0x5/0x5f ? kmem_cache_free+0x3f0/0x450 ? srso_return_thunk+0x5/0x5f ? srso_return_thunk+0x5/0x5f ? syscall_exit_to_user_mode+0x10/0x210 ? srso_return_thunk+0x5/0x5f ? do_syscall_64+0x8e/0x160 ? sysfs_emit+0xaf/0xc0 ? srso_return_thunk+0x5/0x5f ? srso_return_thunk+0x5/0x5f ? seq_read_iter+0x207/0x460 ? srso_return_thunk+0x5/0x5f ? vfs_read+0x29c/0x370 ? srso_return_thunk+0x5/0x5f ? srso_return_thunk+0x5/0x5f ? syscall_exit_to_user_mode+0x10/0x210 ? srso_return_thunk+0x5/0x5f ? do_syscall_64+0x8e/0x160 ? srso_return_thunk+0x5/0x5f ? exc_page_fault+0x7e/0x180 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7fdab1e0ca6d RSP: 002b:00007ffeb2b60c80 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff96c8132f34 RBX: 0000000000000003 RCX: 00007fdab1e0ca6d RDX: 00007ffeb2b60d80 RSI: 00000000c4009420 RDI: 0000000000000003 RBP: 00007ffeb2b60cd0 R08: 0000000000000000 R09: 0000000000000013 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 00007ffeb2b6343b R14: 00007ffeb2b60d80 R15: 0000000000000001 </TASK> CR2: 0000000000000058 ---[end trace 0000000000000000]--- The 1st line is the most interesting here: BTRFS error (device sdc): zoned: write pointer offset mismatch of zones in raid1 profile When a RAID1 block-group is created and a write pointer mismatch between the disks in the RAID set is detected, btrfs sets the <code>alloc_offset</code> to the length of the block group marking it as full. Afterwards the code expects that a balance operation will evacuate the data in this block-group and repair the problems. But before this is possible, the new space of this block-group will be accounted in the free space cache. But in <code>__btrfs_</code> ---truncated---
CVE-2025-37828	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: mcq: Add NULL check in <code>ufshcd_mcq_abort()</code> A race can occur between the MCQ completion path and the abort handler: once a request completes, <code>__blk_mq_free_request()</code> sets <code>rq->mq_hctx</code> to NULL, meaning the subsequent <code>ufshcd_mcq_req_to_hwq()</code> call in <code>ufshcd_mcq_abort()</code> can return a NULL pointer. If this NULL pointer is dereferenced, the kernel will crash. Add a NULL check for the returned hwq pointer. If hwq is NULL, log an error and return FAILED, preventing a potential NULL-pointer dereference. As suggested by Bart, the <code>ufshcd_cmd_inflight()</code> check is removed. This is similar to the fix in commit 74736103fb41 ("scsi: ufs: core: Fix ufshcd_abort_one racing issue"). This is found by our static analysis tool KNightier.
CVE-2025-37829	In the Linux kernel, the following vulnerability has been resolved: cpufreq: scpi: Fix null-ptr-deref in <code>scpi_cpufreq_get_rate()</code> <code>cpufreq_cpu_get_raw()</code> can return NULL when the target CPU is not present in the <code>policy->cpus</code> mask. <code>scpi_cpufreq_get_rate()</code> does not check for this case, which results in a NULL pointer dereference.
CVE-2025-37830	In the Linux kernel, the following vulnerability has been resolved: cpufreq: scmi: Fix null-ptr-deref in <code>scmi_cpufreq_get_rate()</code> <code>cpufreq_cpu_get_raw()</code> can return NULL when the target CPU is not present in the <code>policy->cpus</code> mask. <code>scmi_cpufreq_get_rate()</code> does not check for this case, which results in a NULL pointer dereference. Add NULL check after <code>cpufreq_cpu_get_raw()</code> to prevent this issue.
CVE-2025-37831	In the Linux kernel, the following vulnerability has been resolved: cpufreq: apple-soc: Fix null-ptr-deref in <code>apple_soc_cpufreq_get_rate()</code> <code>cpufreq_cpu_get_raw()</code> can return NULL when the target CPU is not present in the <code>policy->cpus</code> mask. <code>apple_soc_cpufreq_get_rate()</code> does not check for this case, which results in a NULL pointer dereference.

CVE ID	Description
CVE-2025-37833	In the Linux kernel, the following vulnerability has been resolved: net/niu: Niu requires MSIX ENTRY_DATA fields touch before entry reads Fix niu_try_msix() to not cause a fatal trap on sparc systems. Set PCI_DEV_FLAGS_MSIX_TOUCH_ENTRY_DATA_FIRST on the struct pci_dev to work around a bug in the hardware or firmware. For each vector entry in the msix table, niu chips will cause a fatal trap if any registers in that entry are read before that entries' ENTRY_DATA register is written to. Testing indicates writes to other registers are not sufficient to prevent the fatal trap, however the value does not appear to matter. This only needs to happen once after power up, so simply rebooting into a kernel lacking this fix will NOT cause the trap. NON-RESUMABLE ERROR: Reporting on cpu 64 NON-RESUMABLE ERROR: TPC [0x000000000005f6900] <msix_prepare_msi_desc+0x90/0xa0> NON-RESUMABLE ERROR: RAW [4010000000000016:00000e37f93e32ff:0000000202000080:ffffffffffffffff NON-RESUMABLE ERROR: 0000008000000000:0000000000000000:0000000000000000:0000000000000000] NON-RESUMABLE ERROR: handle [0x4010000000000016] stick [0x00000e37f93e32ff] NON-RESUMABLE ERROR: type [precise nonresumable] NON-RESUMABLE ERROR: attrs [0x02000080] < ASI sp-faulted priv > NON-RESUMABLE ERROR: raddr [0xffffffffffffffff] NON-RESUMABLE ERROR: insn effective address [0x000000c50020000c] NON-RESUMABLE ERROR: size [0x8] NON-RESUMABLE ERROR: asi [0x00] CPU: 64 UID: 0 PID: 745 Comm: kworker/64:1 Not tainted 6.11.5 #63 Workqueue: events work_for_cpu_fn TSTATE: 0000000011001602 TPC: 000000000005f6900 TNPC: 000000000005f6904 Y: 00000000 Not tainted TPC: <msix_prepare_msi_desc+0x90/0xa0> g0: 00000000000002e9 g1: 000000000000000c g2: 000000c50020000c g3: 0000000000000100 g4: ffff8000470307c0 g5: ffff800fec5be000 g6: ffff800047a08000 g7: 0000000000000000 o0: ffff800014feb000 o1: ffff800047a0b620 o2: 0000000000000011 o3: ffff800047a0b620 o4: 0000000000000080 o5: 0000000000000011 sp: ffff800047a0ad51 ret_pc: 000000000005f7128 RPC: <__pci_enable_msix_range+0x3cc/0x460> l0: 000000000000000d l1: 0000000000000c01f l2: ffff800014feb0a8 l3: 0000000000000020 l4: 000000000000c000 l5: 0000000000000001 l6: 0000000002000000 l7: ffff800047a0b734 i0: ffff800014feb000 i1: ffff800047a0b730 i2: 0000000000000001 i3: 000000000000000d i4: 0000000000000000 i5: 0000000000000000 i6: ffff800047a0ae81 i7: 00000000101888b0 i7: <niu_try_msix.constprop.0+0xc0/0x130 [niu]> Call Trace: [<00000000101888b0>] niu_try_msix.constprop.0+0xc0/0x130 [niu] [<000000001018f840>] niu_get_invariants+0x183c/0x207c [niu] [<00000000101902fc>] niu_pci_init_one+0x27c/0x2fc [niu] [<000000000005ef3e4>] local_pci_probe+0x28/0x74 [<0000000000469240>] work_for_cpu_fn+0x8/0x1c [<000000000046b008>] process_scheduled_works+0x144/0x210 [<000000000046b518>] worker_thread+0x13c/0x1c0 [<00000000004710e0>] kthread+0xb8/0xc8 [<00000000004060c8>] ret_from_fork+0x1c/0x2c [<0000000000000000>] 0x0 Kernel panic - not syncing: Non-resumable error.
CVE-2025-37834	In the Linux kernel, the following vulnerability has been resolved: mm/vmscan: don't try to reclaim hwpoison folio Syzkaller reports a bug as follows: Injecting memory failure for pfn 0x18b00e at process virtual address 0x20ffd000 Memory failure: 0x18b00e: dirty swapcache page still referenced by 2 users Memory failure: 0x18b00e: recovery action for dirty swapcache page: Failed page: refcount:2 mapcount:0 mapping:0000000000000000 index:0x20ffd pfn:0x18b00e memcg:ffff0000dd6d9000 anon flags: 0x5ffffe00482011(locked dirty arch_1 swapbacked hwpoison node=0 zone=2 lastcpupid=0xfffff) raw: 005ffffe00482011 dead000000000100 dead000000000122 ffff0000e232a7c9 raw: 00000000000020ffd 0000000000000000 00000002ffffff ffff0000dd6d9000 page dumped because: VM_BUG_ON_FOLIO(!folio_test_uptodate(folio)) -----[cut here]----- kernel BUG at mm/swap_state.c:184! Internal error: Oops - BUG: 00000000f2000800 [#1] SMP Modules linked in: CPU: 0 PID: 60 Comm: kswapd0 Not tainted 6.6.0-gcb097e7de84e #3 Hardware name: linux,dummy-virt (DT) pstate: 80400005 (Nzcv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : add_to_swap+0xbc/0x158 lr : add_to_swap+0xbc/0x158 sp : ffff800087f37340 x29: ffff800087f37340 x28: fffffc00052c0380 x27: ffff800087f37780 x26: ffff800087f37490 x25: ffff800087f37c78 x24: ffff800087f377a0 x23: ffff800087f37c50 x22: 0000000000000000 x21: fffffc00052c03b4 x20: 0000000000000000 x19: fffffc00052c0380 x18: 0000000000000000 x17: 296f696c6f662865 x16: 7461646f7470755f x15: 747365745f6f696c x14: 6f6621284f494c4f x13: 0000000000000001 x12: ffff600036d8b97b x11: 1fffe00036d8b97a x10: ffff600036d8b97a x9 : dfff800000000000 x8 : 00009fffc9274686 x7 : ffff0001b6c5cbd3 x6 : 0000000000000001 x5 : ffff0000c25896c0 x4 : 0000000000000000 x3 : 0000000000000000 x2 : 0000000000000000 x1 : ffff0000c25896c0 x0 : 0000000000000000 Call trace: add_to_swap+0xbc/0x158 shrink_folio_list+0x12ac/0x2648 shrink_inactive_list+0x318/0x948 shrink_lruvec+0x450/0x720 shrink_node_memcgs+0x280/0x4a8 shrink_node+0x128/0x978 balance_pgdat+0x4f0/0xb20 kswapd0+0x228/0x438 kthread+0x214/0x230 ret_from_fork+0x10/0x20 I can reproduce this issue with the following steps: 1) When a dirty swapcache page is isolated by reclaim process and the page isn't locked, inject memory failure for the page. me_swapcache_dirty() clears uptodate flag and tries to delete from lru, but fails. Reclaim process will put the hwpoisoned page back to lru. 2) The process that maps the hwpoisoned page exits, the page is deleted the page will never be freed and will be in the lru forever. 3) If we trigger a reclaim again and tries to reclaim the page, add_to_swap() will trigger VM_BUG_ON_FOLIO due to the uptodate flag is cleared. To fix it, skip the hwpoisoned page in shrink_folio_list(). Besides, the hwpoison folio may not be unmapped by hwpoison_user_mappings() yet, unmap it in shrink_folio_list(), otherwise the folio will fail to be unmapped by hwpoison_user_mappings() since the folio isn't in lru list.

CVE ID	Description
CVE-2025-37836	In the Linux kernel, the following vulnerability has been resolved: PCI: Fix reference leak in pci_register_host_bridge() If device_register() fails, call put_device() to give up the reference to avoid a memory leak, per the comment at device_register(). Found by code review. [bhelgaas: squash Dan Carpenter's double free fix from https://lore.kernel.org/r/db806a6c-a91b-4e5a-a84b-6b7e01bdac85@stanley.mountain]
CVE-2025-37839	In the Linux kernel, the following vulnerability has been resolved: jbd2: remove wrong sb->s_sequence check Journal emptiness is not determined by sb->s_sequence == 0 but rather by sb->s_start == 0 (which is set a few lines above). Furthermore 0 is a valid transaction ID so the check can spuriously trigger. Remove the invalid WARN_ON.
CVE-2025-37840	In the Linux kernel, the following vulnerability has been resolved: mtd: rawnand: brcmnand: fix PM resume warning Fixed warning on PM resume as shown below caused due to uninitialized struct nand_operation that checks chip select field : WARN_ON(op->cs >= nanddev_ntargets(&chip->base) [14.588522] -----[cut here]----- [14.588529] WARNING: CPU: 0 PID: 1392 at drivers/mtd/nand/raw/internals.h:139 nand_reset_op+0x1e0/0x1f8 [14.588553] Modules linked in: bdc udc_core [14.588579] CPU: 0 UID: 0 PID: 1392 Comm: rtcwake Tainted: G W 6.14.0-rc4-g5394eea10651 #16 [14.588590] Tainted: [W]=WARN [14.588593] Hardware name: Broadcom STB (Flattened Device Tree) [14.588598] Call trace: [14.588604] dump_backtrace from show_stack+0x18/0x1c [14.588622] r7:00000009 r6:0000008b r5:60000153 r4:c0fa558c [14.588625] show_stack from dump_stack_lvl+0x70/0x7c [14.588639] dump_stack_lvl from dump_stack+0x18/0x1c [14.588653] r5:c08d40b0 r4:c1003cb0 [14.588656] dump_stack from __warn+0x84/0xe4 [14.588668] __warn from warn_slowpath_fmt+0x18c/0x194 [14.588678] r7:c08d40b0 r6:c1003cb0 r5:00000000 r4:00000000 [14.588681] warn_slowpath_fmt from nand_reset_op+0x1e0/0x1f8 [14.588695] r8:70c40dff r7:89705f41 r6:36b4a597 r5:c26c9444 r4:c26b0048 [14.588697] nand_reset_op from brcmnand_resume+0x13c/0x150 [14.588714] r9:00000000 r8:00000000 r7:c24f8010 r6:c228a3f8 r5:c26c94bc r4:c26b0040 [14.588717] brcmnand_resume from platform_pm_resume+0x34/0x54 [14.588735] r5:00000010 r4:c0840a50 [14.588738] platform_pm_resume from dpm_run_callback+0x5c/0x14c [14.588757] dpm_run_callback from device_resume+0xc0/0x324 [14.588776] r9:c24f8054 r8:c24f80a0 r7:00000000 r6:00000000 r5:00000010 r4:c24f8010 [14.588779] device_resume from dpm_resume+0x130/0x160 [14.588799] r9:c22539e4 r8:00000010 r7:c22bebb0 r6:c24f8010 r5:c22539dc r4:c22539b0 [14.588802] dpm_resume from dpm_resume_end+0x14/0x20 [14.588822] r10:c2204e40 r9:00000000 r8:c228a3fc r7:00000000 r6:00000003 r5:c228a414 [14.588826] r4:00000010 [14.588828] dpm_resume_end from suspend_devices_and_enter+0x274/0x6f8 [14.588848] r5:c228a414 r4:00000000 [14.588851] suspend_devices_and_enter from pm_suspend+0x228/0x2bc [14.588868] r10:c3502910 r9:c3501f40 r8:00000004 r7:c228a438 r6:c0f95e18 r5:00000000 [14.588871] r4:00000003 [14.588874] pm_suspend from state_store+0x74/0xd0 [14.588889] r7:c228a438 r6:c0f934c8 r5:00000003 r4:00000003 [14.588892] state_store from kobj_attr_store+0x1c/0x28 [14.588913] r9:00000000 r8:00000000 r7:f09f9f08 r6:00000004 r5:c3502900 r4:c0283250 [14.588916] kobj_attr_store from sysfs_kf_write+0x40/0x4c [14.588936] r5:c3502900 r4:c0d92a48 [14.588939] sysfs_kf_write from kernfs_fop_write_iter+0x104/0x1f0 [14.588956] r5:c3502900 r4:c3501f40 [14.588960] kernfs_fop_write_iter from vfs_write+0x250/0x420 [14.588980] r10:c0e14b48 r9:00000000 r8:c25f5780 r7:00443398 r6:f09f9f68 r5:c34f7f00 [14.588983] r4:c042a88c [14.588987] vfs_write from ksys_write+0x74/0xe4 [14.589005] r10:00000004 r9:c25f5780 r8:c02002fA0 r7:00000000 r6:00000000 r5:c34f7f00 [14.589008] r4:c34f7f00 [14.589011] ksys_write from sys_write+0x10/0x14 [14.589029] r7:00000004 r6:004421c0 r5:00443398 r4:00000004 [14.589032] sys_write from ret_fast_syscall+0x0/0x5c [14.589044] Exception stack(0xf09f9fa8 to 0xf09f9ff0) [14.589050] 9fa0: 00000004 00443398 00000004 00443398 00000004 00000001 [14.589056] 9fc0: 00000004 00443398 004421c0 00000004 b6ecbd58 00000008 bebfbc38 0043eb78 [14.589062] 9fe0: 00440eb0 bebfba8 b6de18a0 b6e579e8 [14.589065] ---[end trace 0000000000000000]--- The fix uses the higher level nand_reset(chip, chipnr); where chipnr = 0, when doing PM resume operation in compliance with the controller support for single die nand chip. Switching from nand_reset_op() to nan ---truncated---
CVE-2025-37841	In the Linux kernel, the following vulnerability has been resolved: pm: cpupower: bench: Prevent NULL dereference on malloc failure If malloc returns NULL due to low memory, 'config' pointer can be NULL. Add a check to prevent NULL dereference.
CVE-2025-37842	In the Linux kernel, the following vulnerability has been resolved: spi: fsl-qspi: use devm function instead of driver remove Driver use devm APIs to manage clk/irq/resources and register the spi controller, but the legacy remove function will be called first during device detach and trigger kernel panic. Drop the remove function and use devm_add_action_or_reset() for driver cleanup to ensure the release sequence. Trigger kernel panic on i.MX8MQ by echo 30bb0000.spi >/sys/bus/platform/drivers/fsl-quadspi/unbind
CVE-2025-37844	In the Linux kernel, the following vulnerability has been resolved: cifs: avoid NULL pointer dereference in dbg call cifs_server_dbg() implies server to be non-NULL so move call under condition to avoid NULL pointer dereference. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2025-37846	In the Linux kernel, the following vulnerability has been resolved: arm64: mops: Do not dereference src reg for a set operation The source register is not used for SET* and reading it can result in a UBSAN out-of-bounds array access error, specifically when the MOPS exception is taken from a SET* sequence with XZR (reg 31) as the source. Architecturally this is the only case where a src/dst/size field in the ESR can be reported as 31. Prior to 2de451a329cf662b the code in do_el0_mops() was benign as the use of pt_regs_read_reg() prevented the out-of-bounds access.
CVE-2025-37849	In the Linux kernel, the following vulnerability has been resolved: KVM: arm64: Tear down vGIC on failed vCPU creation If kvm_arch_vcpu_create() fails to share the vCPU page with the hypervisor, we propagate the error back to the ioctl but leave the vGIC vCPU data initialised. Note only does this leak the corresponding memory when the vCPU is destroyed but it can also lead to use-after-free if the redistributor device handling tries to walk into the vCPU. Add the missing cleanup to kvm_arch_vcpu_create(), ensuring that the vGIC vCPU structures are destroyed on error.
CVE-2025-37850	In the Linux kernel, the following vulnerability has been resolved: pwm: mediatek: Prevent divide-by-zero in pwm_mediatek_config() With CONFIG_COMPILE_TEST && !CONFIG_HAVE_CLK, pwm_mediatek_config() has a divide-by-zero in the following line: do_div(resolution, clk_get_rate(pc->clk_pwms[pwm->hwpwm])); due to the fact that the !CONFIG_HAVE_CLK version of clk_get_rate() returns zero. This is presumably just a theoretical problem: CONFIG_COMPILE_TEST overrides the dependency on RALINK which would select COMMON_CLK. Regardless it's a good idea to check for the error explicitly to avoid divide-by-zero. Fixes the following warning: drivers/pwm/pwm-mediatek.o: warning: objtool: .text: unexpected end of section [ukleinek: s/CONFIG_CLK/CONFIG_HAVE_CLK/]
CVE-2025-37851	In the Linux kernel, the following vulnerability has been resolved: fbdev: omapfb: Add 'plane' value check Function dispc_ovl_setup is not intended to work with the value OMAP_DSS_WB of the enum parameter plane. The value of this parameter is initialized in dss_init_overlays and in the current state of the code it cannot take this value so it's not a real problem. For the purposes of defensive coding it wouldn't be superfluous to check the parameter value, because some functions down the call stack process this value correctly and some not. For example, in dispc_ovl_setup_global_alpha it may lead to buffer overflow. Add check for this value. Found by Linux Verification Center (linuxtesting.org) with SVACE static analysis tool.
CVE-2025-37852	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: handle amdgpu_cgs_create_device() errors in amd_powerplay_create() Add error handling to propagate amdgpu_cgs_create_device() failures to the caller. When amdgpu_cgs_create_device() fails, release hwmgr and return -ENOMEM to prevent null pointer dereference. [v1]->[v2]: Change error code from -EINVAL to -ENOMEM. Free hwmgr.
CVE-2025-37853	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: debugfs hang_hws skip GPU with MES debugfs hang_hws is used by GPU reset test with HWS, for MES this crash the kernel with NULL pointer access because dqm->packet_mgr is not setup for MES path. Skip GPU with MES for now, MES hang_hws debugfs interface will be supported later.
CVE-2025-37854	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Fix mode1 reset crash issue If HW scheduler hangs and mode1 reset is used to recover GPU, KFD signal user space to abort the processes. After process abort exit, user queues still use the GPU to access system memory before h/w is reset while KFD cleanup worker free system memory and free VRAM. There is use-after-free race bug that KFD allocate and reuse the freed system memory, and user queue write to the same system memory to corrupt the data structure and cause driver crash. To fix this race, KFD cleanup worker terminate user queues, then flush reset_domain wq to wait for any GPU ongoing reset complete, and then free outstanding BOs.
CVE-2025-37855	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Guard Possible Null Pointer Dereference [WHY] In some situations, dc->res_pool may be null. [HOW] Check if pointer is null before dereference.
CVE-2025-37856	In the Linux kernel, the following vulnerability has been resolved: btrfs: harden block_group::bg_list against list_del() races As far as I can tell, these calls of list_del_init() on bg_list cannot run concurrently with btrfs_mark_bg_unused() or btrfs_mark_bg_to_reclaim(), as they are in transaction error paths and situations where the block group is readonly. However, if there is any chance at all of racing with mark_bg_unused(), or a different future user of bg_list, better to be safe than sorry. Otherwise we risk the following interleaving (bg_list refcount in parens) T1 (some random op) T2 (btrfs_mark_bg_unused) !list_empty(&bg->bg_list); (1) list_del_init(&bg->bg_list); (1) list_move_tail (1) btrfs_put_block_group (0) btrfs_delete_unused_bgs bg = list_first_entry list_del_init(&bg->bg_list); btrfs_put_block_group(bg); (-1) Ultimately, this results in a broken ref count that hits zero one deref early and the real final deref underflows the refcount, resulting in a WARNING.
CVE-2025-37857	In the Linux kernel, the following vulnerability has been resolved: scsi: st: Fix array overflow in st_setup() Change the array size to follow parms size instead of a fixed value.

CVE ID	Description
CVE-2025-37858	In the Linux kernel, the following vulnerability has been resolved: fs/jfs: Prevent integer overflow in AG size calculation The JFS filesystem calculates allocation group (AG) size using <code>1 << l2agsize</code> in <code>dbExtendFS()</code> . When <code>l2agsize</code> exceeds 31 (possible with >2TB aggregates on 32-bit systems), this 32-bit shift operation causes undefined behavior and improper AG sizing. On 32-bit architectures: - Left-shifting 1 by 32+ bits results in 0 due to integer overflow - This creates invalid AG sizes (0 or garbage values) in <code>sbi->bmap->db_agsize</code> - Subsequent block allocations would reference invalid AG structures - Could lead to: - Filesystem corruption during extend operations - Kernel crashes due to invalid memory accesses - Security vulnerabilities via malformed on-disk structures Fix by casting to <code>s64</code> before shifting: <code>bmap->db_agsize = (s64)1 << l2agsize</code> ; This ensures 64-bit arithmetic even on 32-bit architectures. The cast matches the data type of <code>db_agsize</code> (<code>s64</code>) and follows similar patterns in JFS block calculation code. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-37859	In the Linux kernel, the following vulnerability has been resolved: page_pool: avoid infinite loop to schedule delayed worker We noticed the kworker in <code>page_pool_release_retry()</code> was waken up repeatedly and infinitely in production because of the buggy driver causing the inflight less than 0 and warning us in <code>page_pool_inflight()[1]</code> . Since the inflight value goes negative, it means we should not expect the whole page_pool to get back to work normally. This patch mitigates the adverse effect by not rescheduling the kworker when detecting the inflight negative in <code>page_pool_release_retry()</code> . [1] [Mon Feb 10 20:36:11 2025] -----[cut here]----- [Mon Feb 10 20:36:11 2025] Negative(-51446) inflight packet-pages ... [Mon Feb 10 20:36:11 2025] Call Trace: [Mon Feb 10 20:36:11 2025] page_pool_release_retry+0x23/0x70 [Mon Feb 10 20:36:11 2025] process_one_work+0x1b1/0x370 [Mon Feb 10 20:36:11 2025] worker_thread+0x37/0x3a0 [Mon Feb 10 20:36:11 2025] kthread+0x11a/0x140 [Mon Feb 10 20:36:11 2025] ? process_one_work+0x370/0x370 [Mon Feb 10 20:36:11 2025] ? _kthread_cancel_work+0x40/0x40 [Mon Feb 10 20:36:11 2025] ret_from_fork+0x35/0x40 [Mon Feb 10 20:36:11 2025] --- [end trace ebffe800f33e7e34]--- Note: before this patch, the above calltrace would flood the dmesg due to repeated reschedule of <code>release_dw</code> kworker.
CVE-2025-37860	In the Linux kernel, the following vulnerability has been resolved: sfc: fix NULL dereferences in <code>ef100_process_design_param()</code> Since cited commit, <code>ef100_probe_main()</code> and hence also <code>ef100_check_design_params()</code> run before <code>efx->net_dev</code> is created; consequently, we cannot <code>netif_set_tso_max_size()</code> or <code>_segs()</code> at this point. Move those <code>netif</code> calls to <code>ef100_probe_netdev()</code> , and also replace <code>netif_err</code> within the design params code with <code>pci_err</code> .
CVE-2025-37861	In the Linux kernel, the following vulnerability has been resolved: scsi: mpi3mr: Synchronous access b/w reset and tm thread for reply queue When the task management thread processes reply queues while the reset thread resets them, the task management thread accesses an invalid queue ID (0xFFFF), set by the reset thread, which points to unallocated memory, causing a crash. Add flag 'io_admin_reset_sync' to synchronize access between the reset, I/O, and admin threads. Before a reset, the reset handler sets this flag to block I/O and admin processing threads. If any thread bypasses the initial check, the reset thread waits up to 10 seconds for processing to finish. If the wait exceeds 10 seconds, the controller is marked as unrecoverable.
CVE-2025-37862	In the Linux kernel, the following vulnerability has been resolved: HID: pidff: Fix null pointer dereference in <code>pidff_find_fields</code> This function triggered a null pointer dereference if used to search for a report that isn't implemented on the device. This happened both for optional and required reports alike. The same logic was applied to <code>pidff_find_special_field</code> and although <code>pidff_init_fields</code> should return an error earlier if one of the required reports is missing, future modifications could change this logic and resurface this possible null pointer dereference again. LKML bug report: https://lore.kernel.org/all/CAL-gK7f5=R0nrrQdPtaZZr1fd-cdAMbDMuZ_NLA8vM0SX+nGSw@mail.gmail.com
CVE-2025-37863	In the Linux kernel, the following vulnerability has been resolved: ovl: don't allow datadir only In theory overlays could support upper layer directly referring to a data layer, but there's no current use case for this. Originally, when data-only layers were introduced, this wasn't allowed, only introduced by the "datadir+" feature, but without actually handling this case, resulting in an Oops. Fix by disallowing datadir without lowerdir.

CVE ID	Description
CVE-2025-37864	In the Linux kernel, the following vulnerability has been resolved: net: dsa: clean up FDB, MDB, VLAN entries on unbind As explained in many places such as commit b117e1e8a86d ("net: dsa: delete dsa_legacy_fdb_add and dsa_legacy_fdb_del"), DSA is written given the assumption that higher layers have balanced additions/deletions. As such, it only makes sense to be extremely vocal when those assumptions are violated and the driver unbinds with entries still present. But Ido Schimmel points out a very simple situation where that is wrong: https://lore.kernel.org/netdev/ZDazSM5UsPPjQuKr@shredder/ (also briefly discussed by me in the aforementioned commit). Basically, while the bridge bypass operations are not something that DSA explicitly documents, and for the majority of DSA drivers this API simply causes them to go to promiscuous mode, that isn't the case for all drivers. Some have the necessary requirements for bridge bypass operations to do something useful - see dsa_switch_supports_uc_filtering(). Although in tools/testing/selftests/net/forwarding/local_termination.sh, we made an effort to popularize better mechanisms to manage address filters on DSA interfaces from user space - namely macvlan for unicast, and setsockopt(IP_ADD_MEMBERSHIP) - through mtools - for multicast, the fact is that 'bridge fdb add ... self static local' also exists as kernel UAPI, and might be useful to someone, even if only for a quick hack. It seems counter-productive to block that path by implementing shim .ndo_fdb_add and .ndo_fdb_del operations which just return -EOPNOTSUPP in order to prevent the ndo_dflt_fdb_add() and ndo_dflt_fdb_del() from running, although we could do that. Accepting that cleanup is necessary seems to be the only option. Especially since we appear to be coming back at this from a different angle as well. Russell King is noticing that the WARN_ON() triggers even for VLANs: https://lore.kernel.org/netdev/Z_li8Bj8bD4-BYKQ@shell.armlinux.org.uk/ What happens in the bug report above is that dsa_port_do_vlan_del() fails, then the VLAN entry lingers on, and then we warn on unbind and leak it. This is not a straight revert of the blamed commit, but we now add an informational print to the kernel log (to still have a way to see that bugs exist), and some extra comments gathered from past years' experience, to justify the logic.
CVE-2025-37865	In the Linux kernel, the following vulnerability has been resolved: net: dsa: mv88e6xxx: fix - ENOENT when deleting VLANs and MST is unsupported Russell King reports that on the ZII dev rev B, deleting a bridge VLAN from a user port fails with -ENOENT: https://lore.kernel.org/netdev/Z_IQXNP0s5-IiJzd@shell.armlinux.org.uk/ This comes from mv88e6xxx_port_vlan_leave() -> mv88e6xxx_mst_put(), which tries to find an MST entry in &chip->msts associated with the SID, but fails and returns -ENOENT as such. But we know that this chip does not support MST at all, so that is not surprising. The question is why does the guard in mv88e6xxx_mst_put() not exit early: if (!sid) return 0; And the answer seems to be simple: the sid comes from vlan.sid which supposedly was previously populated by mv88e6xxx_vtu_get(). But some chip->info->ops->vtu_getnext() implementations do not populate vlan.sid, for example see mv88e6185_g1_vtu_getnext(). In that case, later in mv88e6xxx_port_vlan_leave() we are using a garbage sid which is just residual stack memory. Testing for sid == 0 covers all cases of a non-bridge VLAN or a bridge VLAN mapped to the default MSTI. For some chips, SID 0 is valid and installed by mv88e6xxx_stu_setup(). A chip which does not support the STU would implicitly only support mapping all VLANs to the default MSTI, so although SID 0 is not valid, it would be sufficient, if we were to zero-initialize the vlan structure, to fix the bug, due to the coincidence that a test for vlan.sid == 0 already exists and leads to the same (correct) behavior. Another option which would be sufficient would be to add a test for mv88e6xxx_has_stu() inside mv88e6xxx_mst_put(), symmetric to the one which already exists in mv88e6xxx_mst_get(). But that placement means the caller will have to dereference vlan.sid, which means it will access uninitialized memory, which is not nice even if it ignores it later. So we end up making both modifications, in order to not rely just on the sid == 0 coincidence, but also to avoid having uninitialized structure fields which might get temporarily accessed.
CVE-2025-37867	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Silence oversized kvmalloc() warning syzkaller triggered an oversized kvmalloc() warning. Silence it by adding __GFP_NOWARN. syzkaller log: WARNING: CPU: 7 PID: 518 at mm/util.c:665 __kvmalloc_node_noprof+0x175/0x180 CPU: 7 UID: 0 PID: 518 Comm: c_repro Not tainted 6.11.0-rc6+ #6 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4aeb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:__kvmalloc_node_noprof+0x175/0x180 RSP: 0018:ffffc90001e67c10 EFLAGS: 00010246 RAX: 0000000000000100 RBX: 0000000000000400 RCX: ffffffff8149d46b RDX: 0000000000000000 RSI: ffff8881030fae80 RDI: 0000000000000002 RBP: 000000712c800000 R08: 0000000000000100 R09: 0000000000000000 R10: fffffc90001e67c10 R11: 0030ae0601000000 R12: 0000000000000000 R13: 0000000000000000 R14: 00000000ffffff R15: 0000000000000000 FS: 00007fde79159740(0000) GS:ffff88813bdc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000180 CR3: 0000000105eb4005 CR4: 0000000003706b0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ib_umem_odp_get+0x1f6/0x390 mlx5_ib_reg_user_mr+0x1e8/0x450 ib_uverbs_reg_mr+0x28b/0x440 ib_uverbs_write+0x7d3/0xa30 vfs_write+0x1ac/0x6c0 ksys_write+0x134/0x170 ? __sanitizer_cov_trace_pc+0x1c/0x50 do_syscall_64+0x50/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e

CVE ID	Description
CVE-2025-37869	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Use local fence in error path of xe_migrate_clear The intent of the error path in xe_migrate_clear is to wait on locally generated fence and then return. The code is waiting on m->fence which could be the local fence but this is only stable under the job mutex leading to a possible UAF. Fix code to wait on local fence. (cherry picked from commit 762b7e95362170b3e13a8704f38d5e47eca4ba74)
CVE-2025-37870	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: prevent hang on link training fail [Why] When link training fails, the phy clock will be disabled. However, in enable_streams, it is assumed that link training succeeded and the mux selects the phy clock, causing a hang when a register write is made. [How] When enable_stream is hit, check if link training failed. If it did, fall back to the ref clock to avoid a hang and keep the system in a recoverable state.
CVE-2025-37872	In the Linux kernel, the following vulnerability has been resolved: net: txgbe: fix memory leak in txgbe_probe() error path When txgbe_sw_init() is called, memory is allocated for wx->rss_key in wx_init_rss_key(). However, in txgbe_probe() function, the subsequent error paths after txgbe_sw_init() don't free the rss_key. Fix that by freeing it in error path along with wx->mac_table. Also change the label to which execution jumps when txgbe_sw_init() fails, because otherwise, it could lead to a double free for rss_key, when the mac_table allocation fails in wx_sw_init().
CVE-2025-37873	In the Linux kernel, the following vulnerability has been resolved: eth: bnxt: fix missing ring index trim on error path Commit under Fixes converted tx_prod to be free running but missed masking it on the Tx error path. This crashes on error conditions, for example when DMA mapping fails.
CVE-2025-37874	In the Linux kernel, the following vulnerability has been resolved: net: ngbe: fix memory leak in ngbe_probe() error path When ngbe_sw_init() is called, memory is allocated for wx->rss_key in wx_init_rss_key(). However, in ngbe_probe() function, the subsequent error paths after ngbe_sw_init() don't free the rss_key. Fix that by freeing it in error path along with wx->mac_table. Also change the label to which execution jumps when ngbe_sw_init() fails, because otherwise, it could lead to a double free for rss_key, when the mac_table allocation fails in wx_sw_init().
CVE-2025-37875	In the Linux kernel, the following vulnerability has been resolved: igc: fix PTM cycle trigger logic Writing to clear the PTM status 'valid' bit while the PTM cycle is triggered results in unreliable PTM operation. To fix this, clear the PTM 'trigger' and status after each PTM transaction. The issue can be reproduced with the following: \$ sudo phc2sys -R 1000 -O 0 -i tsn0 -m Note: 1000 Hz (-R 1000) is unrealistically large, but provides a way to quickly reproduce the issue. PHC2SYS exits with: "ioctl PTP_OFFSET_PRECISE: Connection timed out" when the PTM transaction fails This patch also fixes a hang in igc_probe() when loading the igc driver in the kdump kernel on systems supporting PTM. The igc driver running in the base kernel enables PTM trigger in igc_probe(). Therefore the driver is always in PTM trigger mode, except in brief periods when manually triggering a PTM cycle. When a crash occurs, the NIC is reset while PTM trigger is enabled. Due to a hardware problem, the NIC is subsequently in a bad busmaster state and doesn't handle register reads/writes. When running igc_probe() in the kdump kernel, the first register access to a NIC register hangs driver probing and ultimately breaks kdump. With this patch, igc has PTM trigger disabled most of the time, and the trigger is only enabled for very brief (10 - 100 us) periods when manually triggering a PTM cycle. Chances that a crash occurs during a PTM trigger are not 0, but extremely reduced.

CVE ID	Description
CVE-2025-37876	In the Linux kernel, the following vulnerability has been resolved: netfs: Only create /proc/fs/netfs with CONFIG_PROC_FS When testing a special config: CONFIG_NETFS_SUPPORTS=y CONFIG_PROC_FS=n The system crashes with something like: [3.766197] -----[cut here]----- [3.766484] kernel BUG at mm/mempool.c:560! [3.766789] Oops: invalid opcode: 0000 [#1] SMP NOPTI [3.767123] CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Tainted: G W [3.767777] Tainted: [W]=WARN [3.767968] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), [3.768523] RIP: 0010:mempool_alloc_slab.cold+0x17/0x19 [3.768847] Code: 50 fe ff 58 5b 5d 41 5c 41 5d 41 5e 41 5f e9 93 95 13 00 [3.769977] RSP: 0018:ffff90000013998 EFLAGS: 00010286 [3.770315] RAX: 000000000000002f RBX: ffff888100ba8640 RCX: 0000000000000000 [3.770749] RDX: 0000000000000000 RSI: 0000000000000003 RDI: 00000000ffffffffff [3.771217] RBP: 0000000000009288 R08: 0000000000000000 R09: fffff90000013828 [3.771664] R10: 0000000000000001 R11: 00000000fffffee R12: 0000000000092cc0 [3.772117] R13: 0000000000000400 R14: ffff8881004b1620 R15: ffffea0004ef7e40 [3.772554] FS: 0000000000000000(0000) GS:ffff8881b5f3c000(0000) knlGS:0000000000000000 [3.773061] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [3.773443] CR2: ffffffff830901b4 CR3: 0000000004296001 CR4: 00000000000770ef0 [3.773884] PKRU: 55555554 [3.774058] Call Trace: [3.774232] <TASK> [3.774371] mempool_alloc_noprof+0x6a/0x190 [3.774649] ? _printk+0x57/0x80 [3.774862] netfs_alloc_request+0x85/0x2ce [3.775147] netfs_readahead+0x28/0x170 [3.775395] read_pages+0x6c/0x350 [3.775623] ? srso_alias_return_thunk+0x5/0xfbef5 [3.775928] page_cache_ra_unbounded+0x1bd/0x2a0 [3.776247] filemap_get_pages+0x139/0x970 [3.776510] ? srso_alias_return_thunk+0x5/0xfbef5 [3.776820] filemap_read+0xf9/0x580 [3.777054] ? srso_alias_return_thunk+0x5/0xfbef5 [3.777368] ? srso_alias_return_thunk+0x5/0xfbef5 [3.777674] ? find_held_lock+0x32/0x90 [3.777929] ? netfs_start_io_read+0x19/0x70 [3.778221] ? netfs_start_io_read+0x19/0x70 [3.778489] ? srso_alias_return_thunk+0x5/0xfbef5 [3.778800] ? lock_acquired+0x1e6/0x450 [3.779054] ? srso_alias_return_thunk+0x5/0xfbef5 [3.779379] netfs_buffered_read_iter+0x57/0x80 [3.779670] __kernel_read+0x158/0x2c0 [3.779927] bprm_execve+0x300/0x7a0 [3.780185] kernel_execve+0x10c/0x140 [3.780423] ? __pfx_kernel_init+0x10/0x10 [3.780690] kernel_init+0xd5/0x150 [3.780910] ret_from_fork+0x2d/0x50 [3.781156] ? __pfx_kernel_init+0x10/0x10 [3.781414] ret_from_fork_asm+0x1a/0x30 [3.781677] </TASK> [3.781823] Modules linked in: [3.782065] ---[end trace 0000000000000000]--- This is caused by the following error path in netfs_init(): if (!proc_mkdir("fs/netfs", NULL)) goto error_proc; Fix this by adding ifdef in netfs_main(), so that /proc/fs/netfs is only created with CONFIG_PROC_FS.
CVE-2025-37877	In the Linux kernel, the following vulnerability has been resolved: iommu: Clear iommu-dma ops on cleanup If iommu_device_register() encounters an error, it can end up tearing down already-configured groups and default domains, however this currently still leaves devices hooked up to iommu-dma (and even historically the behaviour in this area was at best inconsistent across architectures/drivers...) Although in the case that an IOMMU is present whose driver has failed to probe, users cannot necessarily expect DMA to work anyway, it's still arguable that we should do our best to put things back as if the IOMMU driver was never there at all, and certainly the potential for crashing in iommu-dma itself is undesirable. Make sure we clean up the dev->dma_iommu flag along with everything else.
CVE-2025-37878	In the Linux kernel, the following vulnerability has been resolved: perf/core: Fix WARN_ON(!ctx) in __free_event() for partial init Move the get_ctx(child_ctx) call and the child_event->ctx assignment to occur immediately after the child event is allocated. Ensure that child_event->ctx is non-NULL before any subsequent error path within inherit_event calls free_event(), satisfying the assumptions of the cleanup code. Details: There's no clear Fixes tag, because this bug is a side-effect of multiple interacting commits over time (up to 15 years old), not a single regression. The code initially incremented refcount then assigned context immediately after the child_event was created. Later, an early validity check for child_event was added before the refcount/assignment. Even later, a WARN_ON_ONCE() cleanup check was added, assuming event->ctx is valid if the pmu_ctx is valid. The problem is that the WARN_ON_ONCE() could trigger after the initial check passed but before child_event->ctx was assigned, violating its precondition. The solution is to assign child_event->ctx right after its initial validation. This ensures the context exists for any subsequent checks or cleanup routines, resolving the WARN_ON_ONCE(). To resolve it, defer the refcount update and child_event->ctx assignment directly after child_event->pmu_ctx is set but before checking if the parent event is orphaned. The cleanup routine depends on event->pmu_ctx being non-NULL before it verifies event->ctx is non-NULL. This also maintains the author's original intent of passing in child_ctx to find_get_pmu_context before its refcount/assignment. [mingo: Expanded the changelog from another email by Gabriel Shahrouzi.]
CVE-2025-37879	In the Linux kernel, the following vulnerability has been resolved: 9p/net: fix improper handling of bogus negative read/write replies In p9_client_write() and p9_client_read_once(), if the server incorrectly replies with success but a negative write/read count then we would consider written (negative) <= rsize (positive) because both variables were signed. Make variables unsigned to avoid this problem. The reproducer linked below now fails with the following error instead of a null pointer deref: 9pnet: bogus RWRITE count (4294967295 > 3)

CVE ID	Description
CVE-2025-37880	In the Linux kernel, the following vulnerability has been resolved: um: work around sched_yield not yielding in time-travel mode sched_yield by a userspace may not actually cause scheduling in time-travel mode as no time has passed. In the case seen it appears to be a badly implemented userspace spinlock in ASAN. Unfortunately, with time-travel it causes an extreme slowdown or even deadlock depending on the kernel configuration (CONFIG_UML_MAX_USERSPACE_ITERATIONS). Work around it by accounting time to the process whenever it executes a sched_yield syscall.
CVE-2025-37881	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: aspeed: Add NULL pointer check in ast_vhub_init_dev() The variable d->name, returned by devm_kasprintf(), could be NULL. A pointer check is added to prevent potential NULL pointer dereference. This is similar to the fix in commit 3027e7b15b02 ("ice: Fix some null pointer dereference issues in ice_ptp.c"). This issue is found by our static analysis tool
CVE-2025-37882	In the Linux kernel, the following vulnerability has been resolved: usb: xhci: Fix isochronous Ring Underrun/Overrun event handling The TRB pointer of these events points at enqueue at the time of error occurrence on xHCI 1.1+ HCs or it's NULL on older ones. By the time we are handling the event, a new TD may be queued at this ring position. I can trigger this race by rising interrupt moderation to increase IRQ handling delay. Similar delay may occur naturally due to system load. If this ever happens after a Missed Service Error, missed TDs will be skipped and the new TD processed as if it matched the event. It could be given back prematurely, risking data loss or buffer UAF by the xHC. Don't complete TDs on xrun events and don't warn if queued TDs don't match the event's TRB pointer, which can be NULL or a link/no-op TRB. Don't warn if there are no queued TDs at all. Now that it's safe, also handle xrun events if the skip flag is clear. This ensures completion of any TD stuck in 'error mid TD' state right before the xrun event, which could happen if a driver submits a finite number of URBs to a buggy HC and then an error occurs on the last TD.
CVE-2025-37883	In the Linux kernel, the following vulnerability has been resolved: s390/sclp: Add check for get_zeroed_page() Add check for the return value of get_zeroed_page() in sclp_console_init() to prevent null pointer dereference. Furthermore, to solve the memory leak caused by the loop allocation, add a free helper to do the free job.
CVE-2025-37884	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix deadlock between rcu_tasks_trace and event_mutex. Fix the following deadlock: CPU A _free_event() perf_kprobe_destroy() mutex_lock(&event_mutex) perf_trace_event_unreg() synchronize_rcu_tasks_trace() There are several paths where _free_event() grabs event_mutex and calls sync_rcu_tasks_trace. Above is one such case. CPU B bpf_prog_test_run_syscall() rcu_read_lock_trace() bpf_prog_run_pin_on_cpu() bpf_prog_load() bpf_tracing_func_proto() trace_set_clr_event() mutex_lock(&event_mutex) Delegate trace_set_clr_event() to workqueue to avoid such lock dependency.
CVE-2025-37885	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: Reset IRTE to host control if *new* route isn't postable Restore an IRTE back to host control (remapped or posted MSI mode) if the *new* GSI route prevents posting the IRQ directly to a vCPU, regardless of the GSI routing type. Updating the IRTE if and only if the new GSI is an MSI results in KVM leaving an IRTE posting to a vCPU. The dangling IRTE can result in interrupts being incorrectly delivered to the guest, and in the worst case scenario can result in use-after-free, e.g. if the VM is torn down, but the underlying host IRQ isn't freed.
CVE-2025-37886	In the Linux kernel, the following vulnerability has been resolved: pds_core: make wait_context part of q_info Make the wait_context a full part of the q_info struct rather than a stack variable that goes away after pdsc_adminq_post() is done so that the context is still available after the wait loop has given up. There was a case where a slow development firmware caused the adminq request to time out, but then later the FW finally finished the request and sent the interrupt. The handler tried to complete_all() the completion context that had been created on the stack in pdsc_adminq_post() but no longer existed. This caused bad pointer usage, kernel crashes, and much wailing and gnashing of teeth.
CVE-2025-37887	In the Linux kernel, the following vulnerability has been resolved: pds_core: handle unsupported PDS_CORE_CMD_FW_CONTROL result If the FW doesn't support the PDS_CORE_CMD_FW_CONTROL command the driver might at the least print garbage and at the worst crash when the user runs the "devlink dev info" devlink command. This happens because the stack variable fw_list is not 0 initialized which results in fw_list.num_fw_slots being a garbage value from the stack. Then the driver tries to access fw_list.fw_names[i] with i >= ARRAY_SIZE and runs off the end of the array. Fix this by initializing the fw_list and by not failing completely if the devcmd fails because other useful information is printed via devlink dev info even if the devcmd fails.

CVE ID	Description
CVE-2025-37889	In the Linux kernel, the following vulnerability has been resolved: ASoC: ops: Consistently treat platform_max as control value This reverts commit 9bdd10d57a88 ("ASoC: ops: Shift tested values in snd_soc_put_volsw() by +min"), and makes some additional related updates. There are two ways the platform_max could be interpreted; the maximum register value, or the maximum value the control can be set to. The patch moved from treating the value as a control value to a register one. When the patch was applied it was technically correct as snd_soc_limit_volume() also used the register interpretation. However, even then most of the other usages treated platform_max as a control value, and snd_soc_limit_volume() has since been updated to also do so in commit fb9ad24485087 ("ASoC: ops: add correct range check for limiting volume"). That patch however, missed updating snd_soc_put_volsw() back to the control interpretation, and fixing snd_soc_info_volsw_range(). The control interpretation makes more sense as limiting is typically done from the machine driver, so it is appropriate to use the customer facing representation rather than the internal codec representation. Update all the code to consistently use this interpretation of platform_max. Finally, also add some comments to the soc_mixer_control struct to hopefully avoid further patches switching between the two approaches.
CVE-2025-37890	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Fix a UAF vulnerability in class with netem as child qdisc As described in Gerrard's report [1], we have a UAF case when an hfsc class has a netem child qdisc. The crux of the issue is that hfsc is assuming that checking for cl->qdisc->q.len == 0 guarantees that it hasn't inserted the class in the vtree or eltree (which is not true for the netem duplicate case). This patch checks the n_active class variable to make sure that the code won't insert the class in the vtree or eltree twice, catering for the reentrant case. [1] https://lore.kernel.org/netdev/CAHcdcOm+03OD2j6R0=YHKqmy=VgJ8xEOKuP6c7mSgnp-TEJJbw@mail.gmail.com/
CVE-2025-37891	In the Linux kernel, the following vulnerability has been resolved: ALSA: ump: Fix buffer overflow at UMP SysEx message conversion The conversion function from MIDI 1.0 to UMP packet contains an internal buffer to keep the incoming MIDI bytes, and its size is 4, as it was supposed to be the max size for a MIDI1 UMP packet data. However, the implementation overlooked that SysEx is handled in a different format, and it can be up to 6 bytes, as found in do_convert_to_ump(). It leads eventually to a buffer overflow, and may corrupt the memory when a longer SysEx message is received. The fix is simply to extend the buffer size to 6 to fit with the SysEx UMP message.
CVE-2025-37892	In the Linux kernel, the following vulnerability has been resolved: mtd: infatcore: Add error check for infat_read_oob() In INFAT_findwriteunit(), the return value of infat_read_oob() need to be checked. A proper implementation can be found in INFAT_deleteblock(). The status will be set as SECTOR_IGNORE to break from the while-loop correctly if the infat_read_oob() fails.

CVE ID	Description
CVE-2025-37897	In the Linux kernel, the following vulnerability has been resolved: wifi: plfxc: Remove erroneous assert in plfxc_mac_release plfxc_mac_release() asserts that mac->lock is held. This assertion is incorrect, because even if it was possible, it would not be the valid behaviour. The function is used when probe fails or after the device is disconnected. In both cases mac->lock can not be held as the driver is not working with the device at the moment. All functions that use mac->lock unlock it just after it was held. There is also no need to hold mac->lock for plfxc_mac_release() itself, as mac data is not affected, except for mac->flags, which is modified atomically. This bug leads to the following warning: ===== <pre>WARNING: CPU: 0 PID: 127 at drivers/net/wireless/purelifi/plfxc/mac.c:106 plfxc_mac_release +0x7d/0xa0 Modules linked in: CPU: 0 PID: 127 Comm: kworker/0:2 Not tainted 6.1.124-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Workqueue: usb_hub_wq hub_event RIP: 0010:plfxc_mac_release+0x7d/0xa0 drivers/net/wireless/ purelifi/plfxc/mac.c:106 Call Trace: <TASK> probe+0x941/0xbd0 drivers/net/wireless/purelifi/plfxc/ usb.c:694 usb_probe_interface+0x5c0/0xaf0 drivers/usb/core/driver.c:396 really_probe+0x2ab/0xcb0 drivers/base/dd.c:639 __driver_probe_device+0x1a2/0x3d0 drivers/base/dd.c:785 driver_probe_device +0x50/0x420 drivers/base/dd.c:815 __device_attach_driver+0x2cf/0x510 drivers/base/dd.c:943 bus_for_each_drv+0x183/0x200 drivers/base/bus.c:429 __device_attach+0x359/0x570 drivers/ base/dd.c:1015 bus_probe_device+0xba/0x1e0 drivers/base/bus.c:489 device_add+0xb48/0xfd0 drivers/base/core.c:3696 usb_set_configuration+0x19dd/0x2020 drivers/usb/core/message.c:2165 usb_generic_driver_probe+0x84/0x140 drivers/usb/core/generic.c:238 usb_probe_device+0x130/0x260 drivers/usb/core/driver.c:293 really_probe+0x2ab/0xcb0 drivers/base/dd.c:639 __driver_probe_device +0x1a2/0x3d0 drivers/base/dd.c:785 driver_probe_device+0x50/0x420 drivers/base/dd.c:815 __device_attach_driver+0x2cf/0x510 drivers/base/dd.c:943 bus_for_each_drv+0x183/0x200 drivers/base/bus.c:429 __device_attach+0x359/0x570 drivers/base/dd.c:1015 bus_probe_device +0xba/0x1e0 drivers/base/bus.c:489 device_add+0xb48/0xfd0 drivers/base/core.c:3696 usb_new_device +0xbdd/0x18f0 drivers/usb/core/hub.c:2620 hub_port_connect drivers/usb/core/hub.c:5477 [inline] hub_port_connect_change drivers/usb/core/hub.c:5617 [inline] port_event drivers/usb/core/ hub.c:5773 [inline] hub_event+0x2efe/0x5730 drivers/usb/core/hub.c:5855 process_one_work +0x8a9/0x11d0 kernel/workqueue.c:2292 worker_thread+0xa47/0x1200 kernel/workqueue.c:2439 kthread+0x28d/0x320 kernel/kthread.c:376 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:295 </TASK> ===== Found by Linux Verification Center (linuxtesting.org) with Syzkaller.</pre>
CVE-2025-37899	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in session logoff The sess->user object can currently be in use by another thread, for example if another connection has sent a session setup request to bind to the session being free'd. The handler for that connection could be in the smb2_sess_setup function which makes use of sess->user.
CVE-2025-37900	In the Linux kernel, the following vulnerability has been resolved: iommu: Fix two issues in iommu_copy_struct_from_user() In the review for iommu_copy_struct_to_user() helper, Matt pointed out that a NULL pointer should be rejected prior to dereferencing it: https://lore.kernel.org/all/86881827-8E2D-461C-BDA3-FA8FD14C343C@nvidia.com And Alok pointed out a typo at the same time: https://lore.kernel.org/all/480536af-6830-43ce-a327-adbd13dc3f1d@oracle.com Since both issues were copied from iommu_copy_struct_from_user(), fix them first in the current header.
CVE-2025-37901	In the Linux kernel, the following vulnerability has been resolved: irqchip/qcom-mpm: Prevent crash when trying to handle non-wake GPIOs On Qualcomm chipsets not all GPIOs are wakeup capable. Those GPIOs do not have a corresponding MPM pin and should not be handled inside the MPM driver. The IRQ domain hierarchy is always applied, so it's required to explicitly disconnect the hierarchy for those. The pinctrl-msm driver marks these with GPIO_NO_WAKE_IRQ. qcom-pdc has a check for this, but irq-qcom-mpm is currently missing the check. This is causing crashes when setting up interrupts for non-wake GPIOs: root@rb1:~# gpiomon -c gpiochip1 10 irq: IRQ159: trimming hierarchy from :soc@0:interrupt-controller@f200000-1 Unable to handle kernel paging request at virtual address ffff8000a1dc3820 Hardware name: Qualcomm Technologies, Inc. Robotics RB1 (DT) pc : mpm_set_type+0x80/0xcc lr : mpm_set_type+0x5c/0xcc Call trace: mpm_set_type+0x80/0xcc (P) qcom_mpm_set_type+0x64/0x158 irq_chip_set_type_parent+0x20/0x38 msm_gpio_irq_set_type+0x50/0x530 __irq_set_trigger+0x60/0x184 __setup_irq+0x304/0x6bc request_threaded_irq+0xc8/0x19c edge_detector_setup+0x260/0x364 linereq_create+0x420/0x5a8 gpio_ioclt+0x2d4/0x6c0 Fix this by copying the check for GPIO_NO_WAKE_IRQ from qcom-pdc.c, so that MPM is removed entirely from the hierarchy for non-wake GPIOs.

CVE ID	Description
CVE-2025-37903	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Fix slab-use-after-free in hdcp The HDCP code in amdgpu_dm_hdcp.c copies pointers to amdgpu_dm_connector objects without incrementing the kref reference counts. When using a USB-C dock, and the dock is unplugged, the corresponding amdgpu_dm_connector objects are freed, creating dangling pointers in the HDCP code. When the dock is plugged back, the dangling pointers are dereferenced, resulting in a slab-use-after-free: [66.775837] BUG: KASAN: slab-use-after-free in event_property_validate+0x42f/0x6c0 [amdgpu] [66.776171] Read of size 4 at addr ffff888127804120 by task kworker/0:1/10 [66.776179] CPU: 0 UID: 0 PID: 10 Comm: kworker/0:1 Not tainted 6.14.0-rc7-00180-g54505f727a38-dirty #233 [66.776183] Hardware name: HP HP Pavilion Aero Laptop 13-be0xxx/8916, BIOS F.17 12/18/2024 [66.776186] Workqueue: events event_property_validate [amdgpu] [66.776494] Call Trace: [66.776496] <TASK> [66.776497] dump_stack_lvl+0x70/0xa0 [66.776504] print_report+0x175/0x555 [66.776507] ? __virt_addr_valid+0x243/0x450 [66.776510] ? kasan_complete_mode_report_info+0x66/0x1c0 [66.776515] kasan_report+0xeb/0x1c0 [66.776518] ? event_property_validate+0x42f/0x6c0 [amdgpu] [66.776819] ? event_property_validate+0x42f/0x6c0 [amdgpu] [66.777121] __asan_report_load4_noabort+0x14/0x20 [66.777124] event_property_validate+0x42f/0x6c0 [amdgpu] [66.777342] ? __lock_acquire+0x6b40/0x6b40 [66.777347] ? enable_assr+0x250/0x250 [amdgpu] [66.777571] process_one_work+0x86b/0x1510 [66.777575] ? pwq_dec_nr_in_flight+0xcf0/0xcf0 [66.777578] ? assign_work+0x16b/0x280 [66.777580] ? lock_is_held_type+0xa3/0x130 [66.777583] worker_thread+0x5c0/0xfa0 [66.777587] ? process_one_work+0x1510/0x1510 [66.777588] kthread+0x3a2/0x840 [66.777591] ? kthread_is_per_cpu+0xd0/0xd0 [66.777594] ? trace_hardirqs_on+0x4f/0x60 [66.777597] ? _raw_spin_unlock_irq+0x27/0x60 [66.777599] ? calculate_sigpending+0x77/0xa0 [66.777602] ? kthread_is_per_cpu+0xd0/0xd0 [66.777605] ret_from_fork+0x40/0x90 [66.777607] ? kthread_is_per_cpu+0xd0/0xd0 [66.777609] ret_from_fork_asm+0x11/0x20 [66.777614] </TASK> [66.777643] Allocated by task 10: [66.777646] kasan_save_stack+0x39/0x60 [66.777649] kasan_save_track+0x14/0x40 [66.777652] kasan_save_alloc_info+0x37/0x50 [66.777655] __kasan_kmalloc+0xbb/0xc0 [66.777658] __kmalloc_cache_noprof+0x1c8/0x4b0 [66.777661] dm_dp_add_mst_connector+0xdd/0x5c0 [amdgpu] [66.777880] drm_dp_mst_port_add_connector+0x47e/0x770 [drm_display_helper] [66.777892] drm_dp_send_link_address+0x1554/0x2bf0 [drm_display_helper] [66.777901] drm_dp_check_and_send_link_address+0x187/0x1f0 [drm_display_helper] [66.777909] drm_dp_mst_link_probe_work+0x2b8/0x410 [drm_display_helper] [66.777917] process_one_work+0x86b/0x1510 [66.777919] worker_thread+0x5c0/0xfa0 [66.777922] kthread+0x3a2/0x840 [66.777925] ret_from_fork+0x40/0x90 [66.777927] ret_from_fork_asm+0x11/0x20 [66.777932] Freed by task 1713: [66.777935] kasan_save_stack+0x39/0x60 [66.777938] kasan_save_track+0x14/0x40 [66.777940] kasan_save_free_info+0x3b/0x60 [66.777944] __kasan_slab_free+0x52/0x70 [66.777946] kfree+0x13f/0x4b0 [66.777949] dm_dp_mst_connector_destroy+0xfa/0x150 [amdgpu] [66.778179] drm_connector_free+0x7d/0xb0 [66.778184] drm_mode_object_put.part.0+0xee/0x160 [66.778188] drm_mode_object_put+0x37/0x50 [66.778191] drm_atomic_state_default_clear+0x220/0xd60 [66.778194] __drm_atomic_state_free+0x16e/0x2a0 [66.778197] drm_mode_atomic_ioctl+0x15ed/0x2ba0 [66.778200] drm_ioctl_kernel+0x17a/0x310 [66.778203] drm_ioctl+0x584/0xd10 [66.778206] amdgpu_dm_ioctl+0xd2/0x1c0 [amdgpu] [66.778375] __x64_sys_ioctl+0x139/0x1a0 [66.778378] x64_sys_call+0xee7/0xfbo [66.778381] ---truncated---
CVE-2025-37905	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_scmi: Balance device refcount when destroying devices Using device_find_child() to lookup the proper SCMI device to destroy causes an unbalance in device refcount, since device_find_child() calls an implicit get_device(): this, in turns, inhibits the call of the provided release methods upon devices destruction. As a consequence, one of the structures that is not freed properly upon destruction is the internal struct device_private dev->p populated by the drivers subsystem core. KMemleak detects this situation since loading/unloading some SCMI driver causes related devices to be created/destroyed without calling any device_release method. unreferenced object 0xffff0000f583800 (size 512): comm "insmod", pid 227, jiffies 4294912190 hex dump (first 32 bytes): 00 00 00 00 ad 4e ad de ff ff ff ff 00 00 00 00 N..... ff ff ff ff ff ff ff 60 36 1d 8a 00 80 ff ff`6..... backtrace (crc 114e2eed): kmemleak_alloc+0xbc/0xd8 __kmalloc_cache_noprof+0x2dc/0x398 device_add+0x954/0x12d0 device_register+0x28/0x40 __scmi_device_create.part.0+0x1bc/0x380 scmi_device_create+0x2d0/0x390 scmi_create_protocol_devices+0x74/0xf8 scmi_device_request_notifier+0x1f8/0x2a8 notifier_call_chain+0x110/0x3b0 blocking_notifier_call_chain+0x70/0xb0 scmi_driver_register+0x350/0x7f0 0xffff80000a3b3038 do_one_initcall+0x12c/0x730 do_init_module+0x1dc/0x640 load_module+0x4b20/0x5b70 init_module_from_file+0xec/0x158 \$./scripts/faddr2line ./vmlinux device_add+0x954/0x12d0 device_add+0x954/0x12d0: kmalloc_noprof at include/linux/slab.h:901 (inlined by) kcalloc_noprof at include/linux/slab.h:1037 (inlined by) device_private_init at drivers/base/core.c:3510 (inlined by) device_add at drivers/base/core.c:3561 Balance device refcount by issuing a put_device() on devices found via device_find_child().

CVE ID	Description
CVE-2025-37906	In the Linux kernel, the following vulnerability has been resolved: ublk: fix race between <code>io_uring_cmd_complete_in_task</code> and <code>ublk_cancel_cmd</code> <code>ublk_cancel_cmd()</code> calls <code>io_uring_cmd_done()</code> to complete <code>uring_cmd</code> , but we may have scheduled task work via <code>io_uring_cmd_complete_in_task()</code> for dispatching request, then kernel crash can be triggered. Fix it by not trying to canceling the command if <code>ublk</code> block request is started.
CVE-2025-37907	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Fix locking order in <code>ivpu_job_submit</code> Fix deadlock in job submission and abort handling. When a thread aborts currently executing jobs due to a fault, it first locks the global lock protecting submitted_jobs (#1). After the last job is destroyed, it proceeds to release the related context and locks <code>file_priv</code> (#2). Meanwhile, in the job submission thread, the <code>file_priv</code> lock (#2) is taken first, and then the submitted_jobs lock (#1) is obtained when a job is added to the submitted jobs list. CPU0 CPU1 ---- (for example due to a fault) (jobs submissions keep coming) lock(&vdev->submitted_jobs_lock) #1 ivpu_jobs_abort_all() job_destroy() lock(&file_priv->lock) #2 lock(&vdev->submitted_jobs_lock) #1 file_priv_release() lock(&vdev->context_list_lock) lock(&file_priv->lock) #2 This order of locking causes a deadlock. To resolve this issue, change the order of locking in <code>ivpu_job_submit()</code> .
CVE-2025-37909	In the Linux kernel, the following vulnerability has been resolved: net: lan743x: Fix memleak issue when GSO enabled Always map the <code>skb</code> to the LS descriptor. Previously <code>skb</code> was mapped to EXT descriptor when the number of fragments is zero with GSO enabled. Mapping the <code>skb</code> to EXT descriptor prevents it from being freed, leading to a memory leak
CVE-2025-37911	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix out-of-bound <code>memcpy()</code> during <code>ethtool -w</code> When retrieving the FW core dump using <code>ethtool</code> , it can sometimes cause memory corruption: BUG: Kfence: memory corruption in <code>__bnxt_get_coredump+0x3ef/0x670</code> [bnxt_en] Corrupted memory at 0x000000008f0f30e8 [! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !] (in <code>kfence-#45</code>): <code>__bnxt_get_coredump+0x3ef/0x670</code> [bnxt_en] <code>ethtool_get_dump_data+0xdc/0x1a0</code> <code>__dev_ethtool+0xa1e/0x1af0</code> <code>dev_ethtool+0xa8/0x170</code> <code>dev_ioctl+0x1b5/0x580</code> <code>sock_do_ioctl+0xab/0xf0</code> <code>sock_ioctl+0x1ce/0x2e0</code> <code>__x64_sys_ioctl+0x87/0xc0</code> <code>do_syscall_64+0x5c/0xf0</code> <code>entry_SYSCALL_64_after_hwframe+0x78/0x80</code> ... This happens when copying the core dump segment list in <code>bnxt_hwrmb_dbg_dma_data()</code> with the <code>HWRM_DBG_COREDUMP_LIST</code> FW command. The <code>info->dest_buf</code> buffer is allocated based on the number of core dump segments returned by the FW. The segment list is then DMA'ed by the FW and the length of the DMA is returned by FW. The driver then copies this DMA'ed segment list to <code>info->dest_buf</code> . In some cases, this DMA length may exceed the <code>info->dest_buf</code> length and cause the above BUG condition. Fix it by capping the copy length to not exceed the length of <code>info->dest_buf</code> . The extra DMA data contains no useful information. This code path is shared for the <code>HWRM_DBG_COREDUMP_LIST</code> and the <code>HWRM_DBG_COREDUMP_RETRIEVE</code> FW commands. The buffering is different for these 2 FW commands. To simplify the logic, we need to move the line to adjust the buffer length for <code>HWRM_DBG_COREDUMP_RETRIEVE</code> up, so that the new check to cap the copy length will work for both commands.
CVE-2025-37912	In the Linux kernel, the following vulnerability has been resolved: ice: Check VF VSI Pointer Value in <code>ice_vc_add_fdir_fltr()</code> As mentioned in the commit <code>baeb705fd6a7</code> ("ice: always check VF VSI pointer values"), we need to perform a null pointer check on the return value of <code>ice_get_vf_vsi()</code> before using it.
CVE-2025-37913	In the Linux kernel, the following vulnerability has been resolved: net_sched: qfq: Fix double list add in class with netem as child qdisc As described in Gerrard's report [1], there are use cases where a netem child qdisc will make the parent qdisc's enqueue callback reentrant. In the case of qfq, there won't be a UAF, but the code will add the same classifier to the list twice, which will cause memory corruption. This patch checks whether the class was already added to the <code>agg->active_list</code> (<code>cl_is_active</code>) before doing the addition to cater for the reentrant case. [1] https://lore.kernel.org/netdev/CAHcdcOm+03OD2j6R0=YHKqmy=VgJ8xEOKuP6c7mSgnp-TEJJbw@mail.gmail.com/
CVE-2025-37914	In the Linux kernel, the following vulnerability has been resolved: net_sched: ets: Fix double list add in class with netem as child qdisc As described in Gerrard's report [1], there are use cases where a netem child qdisc will make the parent qdisc's enqueue callback reentrant. In the case of ets, there won't be a UAF, but the code will add the same classifier to the list twice, which will cause memory corruption. In addition to checking for <code>qlen</code> being zero, this patch checks whether the class was already added to the <code>active_list</code> (<code>cl_is_active</code>) before doing the addition to cater for the reentrant case. [1] https://lore.kernel.org/netdev/CAHcdcOm+03OD2j6R0=YHKqmy=VgJ8xEOKuP6c7mSgnp-TEJJbw@mail.gmail.com/
CVE-2025-37915	In the Linux kernel, the following vulnerability has been resolved: net_sched: drr: Fix double list add in class with netem as child qdisc As described in Gerrard's report [1], there are use cases where a netem child qdisc will make the parent qdisc's enqueue callback reentrant. In the case of drr, there won't be a UAF, but the code will add the same classifier to the list twice, which will cause memory corruption. In addition to checking for <code>qlen</code> being zero, this patch checks whether the class was already added to the <code>active_list</code> (<code>cl_is_active</code>) before adding to the list to cover for the reentrant case. [1] https://lore.kernel.org/netdev/CAHcdcOm+03OD2j6R0=YHKqmy=VgJ8xEOKuP6c7mSgnp-TEJJbw@mail.gmail.com/

CVE ID	Description
CVE-2025-37916	In the Linux kernel, the following vulnerability has been resolved: pds_core: remove write-after-free of client_id A use-after-free error popped up in stress testing: [Mon Apr 21 21:21:33 2025] BUG: Kfence: use-after-free write in pdsc_auxbus_dev_del+0xef/0x160 [pds_core] [Mon Apr 21 21:21:33 2025] Use-after-free write at 0x000000007013ecd1 (in kfence-#47): [Mon Apr 21 21:21:33 2025] pdsc_auxbus_dev_del+0xef/0x160 [pds_core] [Mon Apr 21 21:21:33 2025] pdsc_remove+0xc0/0x1b0 [pds_core] [Mon Apr 21 21:21:33 2025] pci_device_remove+0x24/0x70 [Mon Apr 21 21:21:33 2025] device_release_driver_internal+0x11f/0x180 [Mon Apr 21 21:21:33 2025] driver_detach+0x45/0x80 [Mon Apr 21 21:21:33 2025] bus_remove_driver+0x83/0xe0 [Mon Apr 21 21:21:33 2025] pci_unregister_driver+0x1a/0x80 The actual device uninit usually happens on a separate thread scheduled after this code runs, but there is no guarantee of order of thread execution, so this could be a problem. There's no actual need to clear the client_id at this point, so simply remove the offending code.
CVE-2025-37917	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk-star-emas: fix spinlock recursion issues on rx/tx poll Use spin_lock_irqsave and spin_unlock_irqrestore instead of spin_lock and spin_unlock in mtk_star_emas driver to avoid spinlock recursion occurrence that can happen when enabling the DMA interrupts again in rx/tx poll. ```` BUG: spinlock recursion on CPU#0, swapper/0/0 lock: 0xffff0000db9cf20, .magic: dead4ead, .owner: swapper/0/0, .owner_cpu: 0 CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.15.0-rc2-next-20250417-00001-gf6a27738686c-dirty #28 PREEMPT Hardware name: MediaTek MT8365 Open Platform EVK (DT) Call trace: show_stack+0x18/0x24 (C) dump_stack_lvl+0x60/0x80 dump_stack+0x18/0x24 spin_dump+0x78/0x88 do_raw_spin_lock+0x11c/0x120 _raw_spin_lock+0x20/0x2c mtk_star_handle_irq+0xc0/0x22c [mtk_star_emas] __handle_irq_event_percpu+0x48/0x140 handle_irq_event+0x4c/0xb0 handle_fastevent_irq+0xa0/0x1bc handle_irq_desc+0x34/0x58 generic_handle_domain_irq+0x1c/0x28 gic_handle_irq+0x4c/0x120 do_interrupt_handler+0x50/0x84 el1_interrupt+0x34/0x68 el1h_64_irq_handler+0x18/0x24 el1h_64_irq+0x6c/0x70 regmap_mmio_read32le+0xc/0x20 (P) _regmap_bus_reg_read+0x6c/0xac _regmap_read+0x60/0xdc regmap_read+0x4c/0x80 mtk_star_rx_poll+0x2f4/0x39c [mtk_star_emas] __napi_poll+0x38/0x188 net_rx_action+0x164/0x2c0 handle_softirqs+0x100/0x244 __do_softirq+0x14/0x20 ____do_softirq+0x10/0x20 call_on_irq_stack+0x24/0x64 do_softirq_own_stack+0x1c/0x40 __irq_exit_rcu+0xd4/0x10c irq_exit_rcu+0x10/0x1c el1_interrupt+0x38/0x68 el1h_64_irq_handler+0x18/0x24 el1h_64_irq+0x6c/0x70 cpuidle_enter_state+0xac/0x320 (P) cpuidle_enter+0x38/0x50 do_idle+0x1e4/0x260 cpu_startup_entry+0x34/0x3c rest_init+0xdc/0xe0 console_on_rootfs+0x0/0x6c __primary_switched+0x88/0x90 ````
CVE-2025-37918	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: btusb: avoid NULL pointer dereference in skb_dequeue() A NULL pointer dereference can occur in skb_dequeue() when processing a QCA firmware crash dump on WCN7851 (0489:e0f3). [93.672166] Bluetooth: hci0: ACL memdump size(589824) [93.672475] BUG: kernel NULL pointer dereference, address: 0000000000000008 [93.672517] Workqueue: hci0 hci_devcd_rx [bluetooth] [93.672598] RIP: 0010:skb_dequeue+0x50/0x80 The issue stems from handle_dump_pkt_qca() returning 0 even when a dump packet is successfully processed. This is because it incorrectly forwards the return value of hci_devcd_init() (which returns 0 on success). As a result, the caller (btusb_recv_acl_qca() or btusb_recv_evt_qca()) assumes the packet was not handled and passes it to hci_recv_frame(), leading to premature kfree() of the skb. Later, hci_devcd_rx() attempts to dequeue the same skb from the dump queue, resulting in a NULL pointer dereference. Fix this by: 1. Making handle_dump_pkt_qca() return 0 on success and negative errno on failure, consistent with kernel conventions. 2. Splitting dump packet detection into separate functions for ACL and event packets for better structure and readability. This ensures dump packets are properly identified and consumed, avoiding double handling and preventing NULL pointer access.
CVE-2025-37920	In the Linux kernel, the following vulnerability has been resolved: xsk: Fix race condition in AF_XDP generic RX path Move rx_lock from xsk_socket to xsk_buff_pool. Fix synchronization for shared umem mode in generic RX path where multiple sockets share single xsk_buff_pool. RX queue is exclusive to xsk_socket, while FILL queue can be shared between multiple sockets. This could result in race condition where two CPU cores access RX path of two different sockets sharing the same umem. Protect both queues by acquiring spinlock in shared xsk_buff_pool. Lock contention may be minimized in the future by some per-thread FQ buffering. It's safe and necessary to move spin_lock_bh(rx_lock) after xsk_rcv_check(): * xs->pool and spinlock_init is synchronized by xsk_bind() -> xsk_is_bound() memory barriers. * xsk_rcv_check() may return true at the moment of xsk_release() or xsk_unbind_dev(), however this will not cause any data races or race conditions. xsk_unbind_dev() removes xdp socket from all maps and waits for completion of all outstanding rx operations. Packets in RX path will either complete safely or drop.

CVE ID	Description
CVE-2025-37921	In the Linux kernel, the following vulnerability has been resolved: vxlan: vni filter: Fix unlocked deletion of default FDB entry When a VNI is deleted from a VXLAN device in 'vni filter' mode, the FDB entry associated with the default remote (assuming one was configured) is deleted without holding the hash lock. This is wrong and will result in a warning [1] being generated by the lockdep annotation that was added by commit ebe642067455 ("vxlan: Create wrappers for FDB lookup"). Reproducer: # ip link add vx0 up type vxlan dstport 4789 external vni filter local 192.0.2.1 # bridge vni add vni 10010 remote 198.51.100.1 dev vx0 # bridge vni del vni 10010 dev vx0 Fix by acquiring the hash lock before the deletion and releasing it afterwards. Blame the original commit that introduced the issue rather than the one that exposed it. [1] WARNING: CPU: 3 PID: 392 at drivers/net/vxlan/vxlan_core.c:417 vxlan_find_mac+0x17f/0x1a0 [...] RIP: 0010:vxlan_find_mac+0x17f/0x1a0 [...] Call Trace: <TASK> __vxlan_fdb_delete+0xbe/0x560 vxlan_vni_delete_group+0x2ba/0x940 vxlan_vni_del.isra.0+0x15f/0x580 vxlan_process_vni_filter+0x38b/0x7b0 vxlan_vni filter_process+0x3bb/0x510 rtnetlink_rcv_msg+0x2f7/0xb70 netlink_rcv_skb+0x131/0x360 netlink_unicast+0x426/0x710 netlink_sendmsg+0x75a/0xc20 __sock_sendmsg+0xc1/0x150 ____sys_sendmsg+0x5aa/0x7b0 __sys_sendmsg+0xfc/0x180 __sys_sendmsg+0x121/0x1b0 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x4b/0x53
CVE-2025-37922	In the Linux kernel, the following vulnerability has been resolved: book3s64/radix : Align section vmemmap start address to PAGE_SIZE A vmemmap altmap is a device-provided region used to provide backing storage for struct pages. For each namespace, the altmap should belong to that same namespace. If the namespaces are created unaligned, there is a chance that the section vmemmap start address could also be unaligned. If the section vmemmap start address is unaligned, the altmap page allocated from the current namespace might be used by the previous namespace also. During the free operation, since the altmap is shared between two namespaces, the previous namespace may detect that the page does not belong to its altmap and incorrectly assume that the page is a normal page. It then attempts to free the normal page, which leads to a kernel crash. Kernel attempted to read user page (18) - exploit attempt? (uid: 0) BUG: Kernel NULL pointer dereference on read at 0x00000018 Faulting instruction address: 0xc000000000530c7c Oops: Kernel access of bad area, sig: 11 [#1] LE PAGE_SIZE=64K MMU=Radix SMP NR_CPUS=2048 NUMA pSeries CPU: 32 PID: 2104 Comm: ndctl Kdump: loaded Tainted: G W NIP: c000000000530c7c LR: c000000000530e00 CTR: 00000000000007fe REGS: c000000015e57040 TRAP: 0300 Tainted: G W MSR: 800000000280b033 <SF,VEC,VSX,EE,FP,ME,IR,DR,RI,LE> CR: 84482404 CFAR: c000000000530dfc DAR: 0000000000000018 DSISR: 40000000 IRQMASK: 0 GPR00: c000000000530e00 c000000015e572e0 c000000002c5cb00 c00c000101008040 GPR04: 0000000000000000 0000000000000007 0000000000000001 000000000000001f GPR08: 0000000000000005 0000000000000000 0000000000000018 0000000000002000 GPR12: c000000001d2fb0 c0000060de6b0080 0000000000000000 c0000060dbf90020 GPR16: c00c000101008000 0000000000000001 0000000000000000 c000000125b20f00 GPR20: 0000000000000001 0000000000000000 ffffffff ffffffff c00c000101007fff GPR24: 0000000000000001 0000000000000000 0000000000000000 0000000000404201 0000000000000001 0000000000000000 c00c000101008040 NIP [c000000000530c7c] get_pfnblock_flags_mask+0x7c/0xd0 LR [c000000000530e00] free_unref_page_prepare+0x130/0x4f0 Call Trace: free_unref_page+0x50/0x1e0 free_reserved_page+0x40/0x68 free_vmemmap_pages+0x98/0xe0 remove_pte_table+0x164/0x1e8 remove_pmd_table+0x204/0x2c8 remove_pud_table+0x1c4/0x288 remove_pagetable+0x1c8/0x310 vmemmap_free+0x24/0x50 section_deactivate+0x28c/0x2a0 __remove_pages+0x84/0x110 arch_remove_memory+0x38/0x60 munmap_pages+0x18c/0x3d0 devm_action_release+0x30/0x50 release_nodes+0x68/0x140 devres_release_group+0x100/0x190 dax_pmem_compat_release+0x44/0x80 [dax_pmem_compat] device_for_each_child+0x8c/0x100 [dax_pmem_compat_remove+0x2c/0x50 [dax_pmem_compat] nvdimmm_bus_remove+0x78/0x140 [libnvdimm] device_remove+0x70/0xd0 Another issue is that if there is no altmap, a PMD-sized vmemmap page will be allocated from RAM, regardless of the alignment of the section start address. If the section start address is not aligned to the PMD size, a VM_BUG_ON will be triggered when setting the PMD-sized page to page table. In this patch, we are aligning the section vmemmap start address to PAGE_SIZE. After alignment, the start address will not be part of the current namespace, and a normal page will be allocated for the vmemmap mapping of the current section. For the remaining sections, altmaps will be allocated. During the free operation, the normal page will be correctly freed. In the same way, a PMD_SIZE vmemmap page will be allocated only if the section start address is PMD_SIZE-aligned; otherwise, it will fall back to a PAGE-sized vmemmap allocation. Without this patch ===== NS1 start NS2 start _____ NS1 NS2 ----- Altmap Altmap Altmap Altmap NS1 NS1 ---truncated---

CVE ID	Description
CVE-2025-37923	In the Linux kernel, the following vulnerability has been resolved: tracing: Fix oob write in trace_seq_to_buffer() syzbot reported this bug: ===== BUG: KASAN: slab-out-of-bounds in trace_seq_to_buffer kernel/trace/trace.c:1830 [inline] BUG: KASAN: slab-out-of-bounds in tracing_splice_read_pipe+0x6be/0xdd0 kernel/trace/trace.c:6822 Write of size 4507 at addr ffff888032b6b000 by task syz.2.320/7260 CPU: 1 UID: 0 PID: 7260 Comm: syz.2.320 Not tainted 6.15.0-rc1-syzkaller-00301-g3bde70a2c827 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xc3/0x670 mm/kasan/report.c:521 kasan_report+0xe0/0x110 mm/kasan/report.c:634 check_region_inline mm/kasan/generic.c:183 [inline] kasan_check_range+0xef/0x1a0 mm/kasan/generic.c:189 __asan_memcpy+0x3c/0x60 mm/kasan/shadow.c:106 trace_seq_to_buffer kernel/trace/trace.c:1830 [inline] tracing_splice_read_pipe+0x6be/0xdd0 kernel/trace/trace.c:6822 ===== It has been reported that trace_seq_to_buffer() tries to copy more data than PAGE_SIZE to buf. Therefore, to prevent this, we should use the smaller of trace_seq_used(&iter->seq) and PAGE_SIZE as an argument.
CVE-2025-37924	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in kerberos authentication Setting sess->user = NULL was introduced to fix the dangling pointer created by ksmbd_free_user. However, it is possible another thread could be operating on the session and make use of sess->user after it has been passed to ksmbd_free_user but before sess->user is set to NULL.
CVE-2025-37925	In the Linux kernel, the following vulnerability has been resolved: jfs: reject on-disk inodes of an unsupported type Syzbot has reported the following BUG: kernel BUG at fs/inode.c:668! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN PTI CPU: 3 UID: 0 PID: 139 Comm: jfsCommit Not tainted 6.12.0-rc4-syzkaller-00085-g4e46774408d9 #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014 RIP: 0010:clear_inode+0x168/0x190 Code: 4c 89 f7 e8 ba fe e5 ff e9 61 ff ff ff 44 89 f1 80 e1 07 80 c1 03 38 c1 7c c1 4c 89 f7 e8 90 ff e5 ff eb b7 0b e8 01 5d 7f ff 90 0f 0b e8 f9 5c 7f ff 90 0f 0b e8 f1 5c 7f RSP: 0018:ffffc900027dfe8 EFLAGS: 00010093 RAX: ffffffff82157a87 RBX: 0000000000000001 RCX: ffff888104d4b980 RDX: 0000000000000000 RSI: 0000000000000001 RDI: 0000000000000000 RBP: fffff900027dfc90 R08: ffffffff82157977 R09: fffff520004fbf38 R10: dffffc0000000000 R11: fffff520004fbf38 R12: dffffc0000000000 R13: ffff88811315bc00 R14: ffff88811315bda8 R15: ffff88811315bb80 FS: 0000000000000000(0000) GS:ffff888135f00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005565222e0578 CR3: 00000000026ef000 CR4: 000000000000006f Call Trace: <TASK> ? __die_body+0x5f/0xb0 ? die+0x9e/0xc0 ? do_trap+0x15a/0x3a0 ? clear_inode+0x168/0x190 ? do_error_trap+0x1dc/0x2c0 ? clear_inode+0x168/0x190 ? __pfx_do_error_trap+0x10/0x10 ? report_bug+0x3cd/0x500 ? handle_invalid_op+0x34/0x40 ? clear_inode+0x168/0x190 ? exc_invalid_op+0x38/0x50 ? asm_exc_invalid_op+0x1a/0x20 ? clear_inode+0x57/0x190 ? clear_inode+0x167/0x190 ? clear_inode+0x168/0x190 ? clear_inode+0x167/0x190 jfs_evict_inode+0xb5/0x440 ? __pfx_jfs_evict_inode+0x10/0x10 evict+0x4ea/0x9b0 ? __pfx_evict+0x10/0x10 ? iput+0x713/0xa50 txUpdateMap+0x931/0xb10 ? __pfx_txUpdateMap+0x10/0x10 jfs_lazycommit+0x49a/0xb80 ? __raw_spin_unlock_irqrestore+0x8f/0x140 ? lockdep_hardirqs_on+0x99/0x150 ? __pfx_jfs_lazycommit+0x10/0x10 ? __pfx_default_wake_function+0x10/0x10 ? __kthread_parkme+0x169/0x1d0 ? __pfx_jfs_lazycommit+0x10/0x10 kthread+0x2f2/0x390 ? __pfx_jfs_lazycommit+0x10/0x10 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x4d/0x80 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> This happens when 'clear_inode()' makes an attempt to finalize an underlying JFS inode of unknown type. According to JFS layout description from https://jfs.sourceforge.net/project/pub/jfslayout.pdf, inode types from 5 to 15 are reserved for future extensions and should not be encountered on a valid filesystem. So add an extra check for valid inode type in 'copy_from_dinode()'.
CVE-2025-37926	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix use-after-free in ksmbd_session_rpc_open A UAF issue can occur due to a race condition between ksmbd_session_rpc_open() and __session_rpc_close(). Add rpc_lock to the session to protect it.
CVE-2025-37927	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Fix potential buffer overflow in parse_ivrs_acpihid There is a string parsing logic error which can lead to an overflow of hid or uid buffers. Comparing ACPIID_LEN against a total string length doesn't take into account the lengths of individual hid and uid buffers so the check is insufficient in some cases. For example if the length of hid string is 4 and the length of the uid string is 260, the length of str will be equal to ACPIID_LEN + 1 but uid string will overflow uid buffer which size is 256. The same applies to the hid string with length 13 and uid string with length 250. Check the length of hid and uid strings separately to prevent buffer overflow. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2025-37928	In the Linux kernel, the following vulnerability has been resolved: dm-bufio: don't schedule in atomic context A BUG was reported as below when CONFIG_DEBUG_ATOMIC_SLEEP and try_verify_in_tasklet are enabled. [129.444685][T934] BUG: sleeping function called from invalid context at drivers/md/dm-bufio.c:2421 [129.444723][T934] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 934, name: kworker/1:4 [129.444740][T934] preempt_count: 201, expected: 0 [129.444756][T934] RCU nest depth: 0, expected: 0 [129.444781][T934] Preemption disabled at: [129.444789][T934] [<fffffd816231900>] shrink_work+0x21c/0x248 [129.445167][T934] kernel BUG at kernel/sched/walt/walt_debug.c:16! [129.445183][T934] Internal error: Oops - BUG: 00000000f2000800 [#1] PREEMPT SMP [129.445204][T934] Skip md ftrace buffer dump for: 0x1609e0 [129.447348][T934] CPU: 1 PID: 934 Comm: kworker/1:4 Tainted: G W OE 6.6.56-android15-8-o-g6f82312b30b9-debug #1 1400000003000000474e5500b3187743670464e8 [129.447362][T934] Hardware name: Qualcomm Technologies, Inc. Parrot QRD, Alpha-M (DT) [129.447373][T934] Workqueue: dm_bufio_cache shrink_work [129.447394][T934] pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [129.447406] [T934] pc : android_rvh_schedule_bug+0x0/0x8 [sched_walt_debug] [129.447435][T934] lr : __traceiter_android_rvh_schedule_bug+0x44/0x6c [129.447451][T934] sp : fffffffc0843dbc90 [129.447459][T934] x29: fffffffc0843dbc90 x28: fffffffc0843dbc90 x27: 00000000000000c8b [129.447479] [T934] x26: 00000000000000040 x25: fffffff804b3d6260 x24: fffffffd816232b68 [129.447497] [T934] x23: fffffff805171c5b4 x22: 00000000000000000 x21: fffffffd816231900 [129.447517] [T934] x20: fffffff80306ba898 x19: 00000000000000000 x18: fffffffc084159030 [129.447535] [T934] x17: 00000000d2b5dd1f x16: 00000000d2b5dd1f x15: fffffff816720358 [129.447554] [T934] x14: 00000000000000004 x13: fffffff89ef978000 x12: 00000000000000003 [129.447572] [T934] x11: fffffffd817a23c4 x10: 00000000000000202 x9: 7e779c5735de9400 [129.447591] [T934] x8: fffffffd81560d004 x7: 205b5d3938373434 x6: fffffffd8167397c8 [129.447610] [T934] x5: 00000000000000000 x4: 00000000000000001 x3: fffffffc0843db9e0 [129.447629] [T934] x2: 00000000000002f15 x1: 00000000000000000 x0: 00000000000000000 [129.447647] [T934] Call trace: [129.447655][T934] android_rvh_schedule_bug+0x0/0x8 [sched_walt_debug 1400000003000000474e550080cce8a8a78606b6] [129.447681][T934] __might_resched+0x190/0x1a8 [129.447694][T934] shrink_work+0x180/0x248 [129.447706][T934] process_one_work +0x260/0x624 [129.447718][T934] worker_thread+0x28c/0x454 [129.447729][T934] kthread+0x118/0x158 [129.447742][T934] ret_from_fork+0x10/0x20 [129.447761][T934] Code: ???????? ???????? ???????? d2b5dd1f (d4210000) [129.447772][T934] ---[end trace 00000000000000000]--- dm_bufio_lock will call spin_lock_bh when try_verify_in_tasklet is enabled, and __scan will be called in atomic context.
CVE-2025-37930	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: Fix WARN_ON in nouveau_fence_context_kill() Nouveau is mostly designed in a way that it's expected that fences only ever get signaled through nouveau_fence_signal(). However, in at least one other place, nouveau_fence_done(), can signal fences, too. If that happens (race) a signaled fence remains in the pending list for a while, until it gets removed by nouveau_fence_update(). Should nouveau_fence_context_kill() run in the meantime, this would be a bug because the function would attempt to set an error code on an already signaled fence. Have nouveau_fence_context_kill() check for a fence being signaled.
CVE-2025-37931	In the Linux kernel, the following vulnerability has been resolved: btrfs: adjust subpage bit start based on sectorsize When running machines with 64k page size and a 16k nodesize we started seeing tree log corruption in production. This turned out to be because we were not writing out dirty blocks sometimes, so this in fact affects all metadata writes. When writing out a subpage EB we scan the subpage bitmap for a dirty range. If the range isn't dirty we do bit_start++; to move onto the next bit. The problem is the bitmap is based on the number of sectors that an EB has. So in this case, we have a 64k pagesize, 16k nodesize, but a 4k sectorsize. This means our bitmap is 4 bits for every node. With a 64k page size we end up with 4 nodes per page. To make this easier this is how everything looks [0 16k 32k 48k] logical address [0 4 8 12] radix tree offset [64k page] folio [16k eb][16k eb][16k eb][16k eb] extent buffers [] bitmap Now we use all of our addressing based on fs_info->sectorsize_bits, so as you can see the above our 16k eb->start turns into radix entry 4. When we find a dirty range for our eb, we correctly do bit_start += sectors_per_node, because if we start at bit 0, the next bit for the next eb is 4, to correspond to eb->start 16k. However if our range is clean, we will do bit_start++, which will now put us offset from our radix tree entries. In our case, assume that the first time we check the bitmap the block is not dirty, we increment bit_start so now it == 1, and then we loop around and check again. This time it is dirty, and we go to find that start using the following equation start = folio_start + bit_start * fs_info->sectorsize; so in the case above, eb->start 0 is now dirty, and we calculate start as 0 + 1 * fs_info->sectorsize = 4096 4096 >> 12 = 1 Now we're looking up the radix tree for 1, and we won't find an eb. What's worse is now we're using bit_start == 1, so we do bit_start += sectors_per_node, which is now 5. If that eb is dirty we will run into the same thing, we will look at an offset that is not populated in the radix tree, and now we're skipping the writeout of dirty extent buffers. The best fix for this is to not use sectorsize_bits to address nodes, but that's a larger change. Since this is a fs corruption problem fix it simply by always using sectors_per_node to increment the start bit.

CVE ID	Description
CVE-2025-37932	In the Linux kernel, the following vulnerability has been resolved: sch_htb: make htb_qlen_notify() idempotent htb_qlen_notify() always deactivates the HTB class and in fact could trigger a warning if it is already deactivated. Therefore, it is not idempotent and not friendly to its callers, like fq_codel_dequeue(). Let's make it idempotent to ease qdisc_tree_reduce_backlog() callers' life.
CVE-2025-37933	In the Linux kernel, the following vulnerability has been resolved: octeon_ep: Fix host hang issue during device reboot When the host loses heartbeat messages from the device, the driver calls the device-specific ndo_stop function, which frees the resources. If the driver is unloaded in this scenario, it calls ndo_stop again, attempting to free resources that have already been freed, leading to a host hang issue. To resolve this, dev_close should be called instead of the device-specific stop function.dev_close internally calls ndo_stop to stop the network interface and performs additional cleanup tasks. During the driver unload process, if the device is already down, ndo_stop is not called.
CVE-2025-37935	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: mtk_eth_soc: fix SER panic with 4GB+ RAM If the mtk_poll_rx() function detects the MTK_RESETTING flag, it will jump to release_desc and refill the high word of the SDP on the 4GB RFB. Subsequently, mtk_rx_clean will process an incorrect SDP, leading to a panic. Add patch from MediaTek's SDK to resolve this.
CVE-2025-37936	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: KVM: Mask PEBS_ENABLE loaded for guest with vCPU's value. When generating the MSR_IA32_PEBS_ENABLE value that will be loaded on VM-Entry to a KVM guest, mask the value with the vCPU's desired PEBS_ENABLE value. Consulting only the host kernel's host vs. guest masks results in running the guest with PEBS enabled even when the guest doesn't want to use PEBS. Because KVM uses perf events to proxy the guest virtual PMU, simply looking at exclude_host can't differentiate between events created by host userspace, and events created by KVM on behalf of the guest. Running the guest with PEBS unexpectedly enabled typically manifests as crashes due to a near-infinite stream of #PFs. E.g. if the guest hasn't written MSR_IA32_DS_AREA, the CPU will hit page faults on address '0' when trying to record PEBS events. The issue is most easily reproduced by running `perf kvm top` from before commit 7b100989b4f6 ("perf evlist: Remove __evlist__add_default") (after which, `perf kvm top` effectively stopped using PEBS). The userspace side of perf creates a guest-only PEBS event, which intel_guest_get_msrs() misconstrues a guest-*owned* PEBS event. Arguably, this is a userspace bug, as enabling PEBS on guest-only events simply cannot work, and userspace can kill VMs in many other ways (there is no danger to the host). However, even if this is considered to be bad userspace behavior, there's zero downside to perf/KVM restricting PEBS to guest-owned events. Note, commit 854250329c02 ("KVM: x86/pmu: Disable guest PEBS temporarily in two rare situations") fixed the case where host userspace is profiling KVM *and* userspace, but missed the case where userspace is profiling only KVM.
CVE-2025-37937	In the Linux kernel, the following vulnerability has been resolved: objtool, media: dib8000: Prevent divide-by-zero in dib8000_set_ddss() If dib8000_set_ddss()'s call to dib8000_read32() returns zero, the result is a divide-by-zero. Prevent that from happening. Fixes the following warning with an UBSAN kernel: drivers/media/dvb-frontends/dib8000.o: warning: objtool: dib8000_tune() falls through to next function dib8096p_cfg_DibRx()
CVE-2025-37938	In the Linux kernel, the following vulnerability has been resolved: tracing: Verify event formats that have "%*p.." The trace event verifier checks the formats of trace events to make sure that they do not point at memory that is not in the trace event itself or in data that will never be freed. If an event references data that was allocated when the event triggered and that same data is freed before the event is read, then the kernel can crash by reading freed memory. The verifier runs at boot up (or module load) and scans the print formats of the events and checks their arguments to make sure that dereferenced pointers are safe. If the format uses "%*p.." the verifier will ignore it, and that could be dangerous. Cover this case as well. Also add to the sample code a use case of "%*pbl".
CVE-2025-37940	In the Linux kernel, the following vulnerability has been resolved: ftrace: Add cond_resched() to ftrace_graph_set_hash() When the kernel contains a large number of functions that can be traced, the loop in ftrace_graph_set_hash() may take a lot of time to execute. This may trigger the softlockup watchdog. Add cond_resched() within the loop to allow the kernel to remain responsive even when processing a large number of functions. This matches the cond_resched() that is used in other locations of the code that iterates over all functions that can be traced.
CVE-2025-37942	In the Linux kernel, the following vulnerability has been resolved: HID: pidff: Make sure to fetch pool before checking SIMULTANEOUS_MAX As noted by Anssi some 20 years ago, pool report is sometimes messed up. This worked fine on many devices but caused oops on VRS DirectForce PRO. Here, we're making sure pool report is refetched before trying to access any of it's fields. While loop was replaced with a for loop + exit conditions were moved around to decrease the possibility of creating an infinite loop scenario.

CVE ID	Description
CVE-2025-37943	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix invalid data access in ath12k_dp_rx_h_undecap_nwifi In certain cases, hardware might provide packets with a length greater than the maximum native Wi-Fi header length. This can lead to accessing and modifying fields in the header within the ath12k_dp_rx_h_undecap_nwifi function for DP_RX_DECAP_TYPE_NATIVE_WIFI decap type and potentially resulting in invalid data access and memory corruption. Add a sanity check before processing the SKB to prevent invalid data access in the undecap native Wi-Fi function for the DP_RX_DECAP_TYPE_NATIVE_WIFI decap type. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.3.1-00173-QCAHKS WPL_SILICONZ-1
CVE-2025-37944	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix invalid entry fetch in ath12k_dp_mon_srng_process Currently, ath12k_dp_mon_srng_process uses ath12k_hal_srng_src_get_next_entry to fetch the next entry from the destination ring. This is incorrect because ath12k_hal_srng_src_get_next_entry is intended for source rings, not destination rings. This leads to invalid entry fetches, causing potential data corruption or crashes due to accessing incorrect memory locations. This happens because the source ring and destination ring have different handling mechanisms and using the wrong function results in incorrect pointer arithmetic and ring management. To fix this issue, replace the call to ath12k_hal_srng_src_get_next_entry with ath12k_hal_srng_dst_get_next_entry in ath12k_dp_mon_srng_process. This ensures that the correct function is used for fetching entries from the destination ring, preventing invalid memory accesses. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.3.1-00173-QCAHKS WPL_SILICONZ-1 Tested-on: WCN7850 hw2.0 WLAN.HMT.1.0.c5-00481-QCAHMTSWPL_V1.0_V2.0_SILICONZ-3
CVE-2025-37945	In the Linux kernel, the following vulnerability has been resolved: net: phy: allow MDIO bus PM ops to start/stop state machine for phylink-controlled PHY DSA has 2 kinds of drivers: 1. Those who call dsa_switch_suspend() and dsa_switch_resume() from their device PM ops: qca8k-8xxx, bcm_sf2, microchip ksz 2. Those who don't: all others. The above methods should be optional. For type 1, dsa_switch_suspend() calls dsa_user_suspend() -> phylink_stop(), and dsa_switch_resume() calls dsa_user_resume() -> phylink_start(). These seem good candidates for setting mac_managed_pm = true because that is essentially its definition [1], but that does not seem to be the biggest problem for now, and is not what this change focuses on. Talking strictly about the 2nd category of DSA drivers here (which do not have MAC managed PM, meaning that for their attached PHYs, mdio_bus_phy_suspend() and mdio_bus_phy_resume() should run in full), I have noticed that the following warning from mdio_bus_phy_resume() is triggered: WARN_ON(phydev->state != PHY_HALTED && phydev->state != PHY_READY && phydev->state != PHY_UP); because the PHY state machine is running. It's running as a result of a previous dsa_user_open() -> ... -> phylink_start() -> phy_start() having been initiated by the user. The previous mdio_bus_phy_suspend() was supposed to have called phy_stop_machine(), but it didn't. So this is why the PHY is in state PHY_NOLINK by the time mdio_bus_phy_resume() runs. mdio_bus_phy_suspend() did not call phy_stop_machine() because for phylink, the phydev->adjust_link function pointer is NULL. This seems a technicality introduced by commit fddd91016d16 ("phylib: fix PAL state machine restart on resume"). That commit was written before phylink existed, and was intended to avoid crashing with consumer drivers which don't use the PHY state machine - phylink always does, when using a PHY. But phylink itself has historically not been developed with suspend/resume in mind, and apparently not tested too much in that scenario, allowing this bug to exist unnoticed for so long. Plus, prior to the WARN_ON(), it would have likely been invisible. This issue is not in fact restricted to type 2 DSA drivers (according to the above ad-hoc classification), but can be extrapolated to any MAC driver with phylink and MDIO-bus-managed PHY PM ops. DSA is just where the issue was reported. Assuming mac_managed_pm is set correctly, a quick search indicates the following other drivers might be affected: \$ grep -Zlr PHYLINK_NETDEV drivers/ xargs -0 grep -L mac_managed_pm drivers/net/ethernet/atheros/ag71xx.c drivers/net/ethernet/microchip/sparx5/sparx5_main.c drivers/net/ethernet/microchip/lan966x/lan966x_main.c drivers/net/ethernet/freescale/dpaa2/dpaa2-mac.c drivers/net/ethernet/freescale/fs_enet/fs_enet-main.c drivers/net/ethernet/freescale/dpaa/dpaa_eth.c drivers/net/ethernet/freescale/ucc_ghet.c drivers/net/ethernet/freescale/enetc/enetc_pf_common.c drivers/net/ethernet/marvell/mvpp2/mvpp2_main.c drivers/net/ethernet/marvell/mvnet.c drivers/net/ethernet/marvell/prestera/prestera_main.c drivers/net/ethernet/mediatek/mtk_eth_soc.c drivers/net/ethernet/altera/altera_tse_main.c drivers/net/ethernet/wangxun/txgbe/txgbe_phy.c drivers/net/ethernet/meta/fbnic/fbnic_phylink.c drivers/net/ethernet/tehuti/tn40_phy.c drivers/net/ethernet/mscc/ocelot_net.c Make the existing conditions dependent on the PHY device having a phydev->phy_link_change() implementation equal to the default phy_link_change() provided by phylib. Otherwise, we implicitly know that the phydev has the phylink-provided phylink_phy_change() callback, and when phylink is used, the PHY state machine always needs to be stopped/ started on the suspend/resume path. The code is structured as such that if phydev->phy_link_change() is absent, it is a matter of time until the kernel will crash - no need to further complicate the test. Thus, for the situation where the PM is not managed b ---truncated---
CVE-2025-37947	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent out-of-bounds stream writes by validating *pos ksmbd_vfs_stream_write() did not validate whether the write offset (*pos) was within the bounds of the existing stream data length (v_len). If *pos was greater than or equal to v_len, this could lead to an out-of-bounds memory write. This patch adds a check to ensure *pos is less than v_len before proceeding. If the condition fails, -EINVAL is returned.

CVE ID	Description
CVE-2025-37948	In the Linux kernel, the following vulnerability has been resolved: arm64: bpf: Add BHB mitigation to the epilogue for cBPF programs A malicious BPF program may manipulate the branch history to influence what the hardware speculates will happen next. On exit from a BPF program, emit the BHB mitigation sequence. This is only applied for 'classic' cBPF programs that are loaded by seccomp.
CVE-2025-37949	In the Linux kernel, the following vulnerability has been resolved: xenbus: Use kref to track req lifetime Marek reported seeing a NULL pointer fault in the xenbus_thread callstack: BUG: kernel NULL pointer dereference, address: 0000000000000000 RIP: e030: __wake_up_common+0x4c/0x180 Call Trace: <TASK> __wake_up_common_lock+0x82/0xd0 process_msg+0x18e/0x2f0 xenbus_thread+0x165/0x1c0 process_msg+0x18e is req->cb(req). req->cb is set to xs_wake_up(), a thin wrapper around wake_up(), or xenbus_dev_queue_reply(). It seems like it was xs_wake_up() in this case. It seems like req may have woken up the xs_wait_for_reply(), which kfree()ed the req. When xenbus_thread resumes, it faults on the zero-ed data. Linux Device Drivers 2nd edition states: "Normally, a wake_up call can cause an immediate reschedule to happen, meaning that other processes might run before wake_up returns." ... which would match the behaviour observed. Change to keeping two krefs on each request. One for the caller, and one for xenbus_thread. Each will kref_put() when finished, and the last will free it. This use of kref matches the description in Documentation/core-api/kref.rst
CVE-2025-37951	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Add job to pending list if the reset was skipped When a CL/CSD job times out, we check if the GPU has made any progress since the last timeout. If so, instead of resetting the hardware, we skip the reset and let the timer get rearmed. This gives long-running jobs a chance to complete. However, when `timedout_job()` is called, the job in question is removed from the pending list, which means it won't be automatically freed through `free_job()`. Consequently, when we skip the reset and keep the job running, the job won't be freed when it finally completes. This situation leads to a memory leak, as exposed in [1] and [2]. Similarly to commit 704d3d60fec4 ("drm/etnaviv: don't block scheduler when GPU is still active"), this patch ensures the job is put back on the pending list when extending the timeout.
CVE-2025-37952	In the Linux kernel, the following vulnerability has been resolved: ksmbd: Fix UAF in __close_file_table_ids A use-after-free is possible if one thread destroys the file via __ksmbd_close_fd while another thread holds a reference to it. The existing checks on fp->refcount are not sufficient to prevent this. The fix takes ft->lock around the section which removes the file from the file table. This prevents two threads acquiring the same file pointer via __close_file_table_ids, as well as the other functions which retrieve a file from the IDR and which already use this same lock.
CVE-2025-37954	In the Linux kernel, the following vulnerability has been resolved: smb: client: Avoid race in open_cached_dir with lease breaks A pre-existing valid cfid returned from find_or_create_cached_dir might race with a lease break, meaning open_cached_dir doesn't consider it valid, and thinks it's newly-constructed. This leaks a dentry reference if the allocation occurs before the queued lease break work runs. Avoid the race by extending holding the cfid_list_lock across find_or_create_cached_dir and when the result is checked.
CVE-2025-37956	In the Linux kernel, the following vulnerability has been resolved: ksmbd: prevent rename with empty string Client can send empty newname string to ksmbd server. It will cause a kernel oops from d_alloc. This patch return the error when attempting to rename a file or directory with an empty new name string.

CVE ID	Description
CVE-2025-37957	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Forcibly leave SMM mode on SHUTDOWN interception Previously, commit ed129ec9057f ("KVM: x86: forcibly leave nested mode on vCPU reset") addressed an issue where a triple fault occurring in nested mode could lead to use-after-free scenarios. However, the commit did not handle the analogous situation for System Management Mode (SMM). This omission results in triggering a WARN when KVM forces a vCPU INIT after SHUTDOWN interception while the vCPU is in SMM. This situation was reproduced using Syzkaller by: 1) Creating a KVM VM and vCPU 2) Sending a KVM_SMI ioctl to explicitly enter SMM 3) Executing invalid instructions causing consecutive exceptions and eventually a triple fault The issue manifests as follows: WARNING: CPU: 0 PID: 25506 at arch/x86/kvm/x86.c:12112 kvm_vcpu_reset+0x1d2/0x1530 arch/x86/kvm/x86.c:12112 Modules linked in: CPU: 0 PID: 25506 Comm: syz-executor.0 Not tainted 6.1.130-syzkaller-00157-g164fe5dde9b6 #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 RIP: 0010:kvm_vcpu_reset+0x1d2/0x1530 arch/x86/kvm/x86.c:12112 Call Trace: <TASK> shutdown_interception+0x66/0xb0 arch/x86/kvm/svm/svm.c:2136 svm_invoke_exit_handler+0x110/0x530 arch/x86/kvm/svm/svm.c:3395 svm_handle_exit+0x424/0x920 arch/x86/kvm/svm/svm.c:3457 vcpu_enter_guest arch/x86/kvm/x86.c:10959 [inline] vcpu_run+0x2c43/0x5a90 arch/x86/kvm/x86.c:11062 kvm_arch_vcpu_ioctl_run+0x50f/0x1cf0 arch/x86/kvm/x86.c:11283 kvm_vcpu_ioctl+0x570/0xf00 arch/x86/kvm/../../../../virt/kvm/kvm_main.c:4122 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:870 [inline] __se_sys_ioctl fs/ioctl.c:856 [inline] __x64_sys_ioctl+0x19a/0x210 fs/ioctl.c:856 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x35/0x80 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x6e/0xd8 Architecturally, INIT is blocked when the CPU is in SMM, hence KVM's WARN() in kvm_vcpu_reset() to guard against KVM bugs, e.g. to detect improper emulation of INIT. SHUTDOWN on SVM is a weird edge case where KVM needs to do <code>_something_sane</code> with the VMCB, since it's technically undefined, and INIT is the least awful choice given KVM's ABI. So, double down on stuffing INIT on SHUTDOWN, and force the vCPU out of SMM to avoid any weirdness (and the WARN). Found by Linux Verification Center (linuxtesting.org) with Syzkaller. [sean: massage changelog, make it clear this isn't architectural behavior]
CVE-2025-37958	In the Linux kernel, the following vulnerability has been resolved: mm/huge_memory: fix dereferencing invalid pmd migration entry When migrating a THP, concurrent access to the PMD migration entry during a deferred split scan can lead to an invalid address access, as illustrated below. To prevent this invalid access, it is necessary to check the PMD migration entry and return early. In this context, there is no need to use <code>pmd_to_swp_entry</code> and <code>pfn_swap_entry_to_page</code> to verify the equality of the target folio. Since the PMD migration entry is locked, it cannot be served as the target. Mailing list discussion and explanation from Hugh Dickins: "An anon_vma lookup points to a location which may contain the folio of interest, but might instead contain another folio: and weeding out those other folios is precisely what the <code>"folio != pmd_folio(*pmd)"</code> check (and the "risk of replacing the wrong folio" comment a few lines above it) is for." BUG: unable to handle page fault for address: ffffea60001db008 CPU: 0 UID: 0 PID: 2199114 Comm: tee Not tainted 6.14.0+ #4 NONE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010:split_huge_pmd_locked+0x3b5/0x2b60 Call Trace: <TASK> try_to_migrate_one+0x28c/0x3730 rmap_walk_anon+0x4f6/0x770 unmap_folio+0x196/0x1f0 split_huge_page_to_list_to_order+0x9f6/0x1560 deferred_split_scan+0xac5/0x12a0 shrinker_debugfs_scan_write+0x376/0x470 full_proxy_write+0x15c/0x220 vfs_write+0x2fc/0xcb0 ksys_write+0x146/0x250 do_syscall_64+0x6a/0x120 entry_SYSCALL_64_after_hwframe+0x76/0x7e The bug is found by syzkaller on an internal kernel, then confirmed on upstream.

CVE ID	Description
CVE-2025-37959	In the Linux kernel, the following vulnerability has been resolved: bpf: Scrub packet on bpf_redirect_peer When bpf_redirect_peer is used to redirect packets to a device in another network namespace, the skb isn't scrubbed. That can lead skb information from one namespace to be "misused" in another namespace. As one example, this is causing Cilium to drop traffic when using bpf_redirect_peer to redirect packets that just went through IPsec decryption to a container namespace. The following pwrtrace shows (1) the packet path from the host's XFRM layer to the container's XFRM layer where it's dropped and (2) the number of active skb extensions at each function. NETNS MARK IFACE TUPLE FUNC 4026533547 d00 eth0 10.244.3.124:35473->10.244.2.158:53 xfrm_rcv_cb .active_extensions = (__u8)2, 4026533547 d00 eth0 10.244.3.124:35473->10.244.2.158:53 xfrm4_rcv_cb .active_extensions = (__u8)2, 4026533547 d00 eth0 10.244.3.124:35473->10.244.2.158:53 gro_cells_receive .active_extensions = (__u8)2, [...] 4026533547 0 eth0 10.244.3.124:35473->10.244.2.158:53 skb_do_redirect .active_extensions = (__u8)2, 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 ip_rcv .active_extensions = (__u8)2, 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 ip_rcv_core .active_extensions = (__u8)2, [...] 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 udp_queue_rcv_one_skb .active_extensions = (__u8)2, 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 __xfrm_policy_check .active_extensions = (__u8)2, 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 __xfrm_decode_session .active_extensions = (__u8)2, 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 security_xfrm_decode_session .active_extensions = (__u8)2, 4026534999 0 eth0 10.244.3.124:35473->10.244.2.158:53 kfree_skb_reason(SKB_DROP_REASON_XFRM_POLICY) .active_extensions = (__u8)2, In this case, there are no XFRM policies in the container's network namespace so the drop is unexpected. When we decrypt the IPsec packet, the XFRM state used for decryption is set in the skb extensions. This information is preserved across the netns switch. When we reach the XFRM policy check in the container's netns, __xfrm_policy_check drops the packet with LINUX_MIB_XFRMINNOPOLS because a (container-side) XFRM policy can't be found that matches the (host-side) XFRM state used for decryption. This patch fixes this by scrubbing the packet when using bpf_redirect_peer, as is done on typical netns switches via veth devices except skb->mark and skb->timestamp are not zeroed.
CVE-2025-37960	In the Linux kernel, the following vulnerability has been resolved: memblock: Accept allocated memory before use in memblock_double_array() When increasing the array size in memblock_double_array() and the slab is not yet available, a call to memblock_find_in_range() is used to reserve/allocate memory. However, the range returned may not have been accepted, which can result in a crash when booting an SNP guest: RIP: 0010:memcpy_orig+0x68/0x130 Code: ... RSP: 0000:ffff9cc03ce8 EFLAGS: 00010006 RAX: ff11001ff83e5000 RBX: 0000000000000000 RCX: ffffffff000 RDX: 00000000000000bc0 RSI: ffffffff9dba8860 RDI: ff11001ff83e5c00 RBP: 0000000000002000 R08: 0000000000000000 R09: 0000000000002000 R10: 00000207fffe000 R11: 0000040000000000 R12: ffffffff9d06ef78 R13: ff11001ff83e5000 R14: ffffffff9dba7c60 R15: 0000000000000c00 memblock_double_array+0xff/0x310 memblock_add_range+0x1fb/0x2f0 memblock_reserve+0x4f/0xa0 memblock_alloc_range_nid+0xac/0x130 memblock_alloc_internal+0x53/0xc0 memblock_alloc_try_nid+0x3d/0xa0 swiotlb_init_remap+0x149/0x2f0 mem_init+0xb/0xb0 mm_core_init+0x8f/0x350 start_kernel+0x17e/0x5d0 x86_64_start_reservations+0x14/0x30 x86_64_start_kernel+0x92/0xa0 secondary_startup_64_no_verify+0x194/0x19b Mitigate this by calling accept_memory() on the memory range returned before the slab is available. Prior to v6.12, the accept_memory() interface used a 'start' and 'end' parameter instead of 'start' and 'size', therefore the accept_memory() call must be adjusted to specify 'start + size' for 'end' when applying to kernels prior to v6.12.

CVE ID	Description
CVE-2025-37961	In the Linux kernel, the following vulnerability has been resolved: ipvs: fix uninit-value for saddr in do_output_route4 syzbot reports for uninit-value for the saddr argument [1]. commit 4754957f04f5 ("ipvs: do not use random local source address for tunnels") already implies that the input value of saddr should be ignored but the code is still reading it which can prevent to connect the route. Fix it by changing the argument to ret_saddr. [1] BUG: KMSAN: uninit-value in do_output_route4+0x42c/0x4d0 net/netfilter/ipvs/ip_vs_xmit.c:147 do_output_route4+0x42c/0x4d0 net/netfilter/ipvs/ip_vs_xmit.c:147 __ip_vs_get_out_rt+0x403/0x21d0 net/netfilter/ipvs/ip_vs_xmit.c:330 ip_vs_tunnel_xmit+0x205/0x2380 net/netfilter/ipvs/ip_vs_xmit.c:1136 ip_vs_in_hook+0x1aa5/0x35b0 net/netfilter/ipvs/ip_vs_core.c:2063 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf7/0x400 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] __ip_local_out+0x758/0x7e0 net/ipv4/ip_output.c:118 ip_local_out net/ipv4/ip_output.c:127 [inline] ip_send_skb+0x6a/0x3c0 net/ipv4/ip_output.c:1501 udp_send_skb+0xfda/0x1b70 net/ipv4/udp.c:1195 udp_sendmsg+0x2fe3/0x33c0 net/ipv4/udp.c:1483 inet_sendmsg+0x1fc/0x280 net/ipv4/af_inet.c:851 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg+0x267/0x380 net/socket.c:727 __sys_sendmsg+0x91b/0xda0 net/socket.c:2566 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2620 __sys_sendmmsg+0x41d/0x880 net/socket.c:2702 __compat_sys_sendmmsg net/compat.c:360 [inline] __do_compat_sys_sendmmsg net/compat.c:367 [inline] __se_compat_sys_sendmmsg net/compat.c:364 [inline] __ia32_compat_sys_sendmmsg+0xc8/0x140 net/compat.c:364 ia32_sys_call+0x3ffa/0x41f0 arch/x86/include/generated/asm/syscalls_32.h:346 do_syscall_32_irqs_on arch/x86/entry/syscall_32.c:83 [inline] __do_fast_syscall_32+0xb0/0x110 arch/x86/entry/syscall_32.c:306 do_fast_syscall_32+0x38/0x80 arch/x86/entry/syscall_32.c:331 do_SYSENTER_32+0x1f/0x30 arch/x86/entry/syscall_32.c:369 entry_SYSENTER_compat_after_hwframe+0x84/0x8e Uninit was created at: slab_post_alloc_hook mm/slub.c:4167 [inline] slab_alloc_node mm/slub.c:4210 [inline] __kmalloc_cache_noprof+0x8fa/0xe00 mm/slub.c:4367 kmalloc_noprof include/linux/slab.h:905 [inline] ip_vs_dest_dst_alloc net/netfilter/ipvs/ip_vs_xmit.c:61 [inline] __ip_vs_get_out_rt+0x35d/0x21d0 net/netfilter/ipvs/ip_vs_xmit.c:323 ip_vs_tunnel_xmit+0x205/0x2380 net/netfilter/ipvs/ip_vs_xmit.c:1136 ip_vs_in_hook+0x1aa5/0x35b0 net/netfilter/ipvs/ip_vs_core.c:2063 nf_hook_entry_hookfn include/linux/netfilter.h:154 [inline] nf_hook_slow+0xf7/0x400 net/netfilter/core.c:626 nf_hook include/linux/netfilter.h:269 [inline] __ip_local_out+0x758/0x7e0 net/ipv4/ip_output.c:118 ip_local_out net/ipv4/ip_output.c:127 [inline] ip_send_skb+0x6a/0x3c0 net/ipv4/ip_output.c:1501 udp_send_skb+0xfda/0x1b70 net/ipv4/udp.c:1195 udp_sendmsg+0x2fe3/0x33c0 net/ipv4/udp.c:1483 inet_sendmsg+0x1fc/0x280 net/ipv4/af_inet.c:851 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg+0x267/0x380 net/socket.c:727 __sys_sendmsg+0x91b/0xda0 net/socket.c:2566 __sys_sendmsg+0x28d/0x3c0 net/socket.c:2620 __sys_sendmmsg+0x41d/0x880 net/socket.c:2702 __compat_sys_sendmmsg net/compat.c:360 [inline] __do_compat_sys_sendmmsg net/compat.c:367 [inline] __se_compat_sys_sendmmsg net/compat.c:364 [inline] __ia32_compat_sys_sendmmsg+0xc8/0x140 net/compat.c:364 ia32_sys_call+0x3ffa/0x41f0 arch/x86/include/generated/asm/syscalls_32.h:346 do_syscall_32_irqs_on arch/x86/entry/syscall_32.c:83 [inline] __do_fast_syscall_32+0xb0/0x110 arch/x86/entry/syscall_32.c:306 do_fast_syscall_32+0x38/0x80 arch/x86/entry/syscall_32.c:331 do_SYSENTER_32+0x1f/0x30 arch/x86/entry/syscall_32.c:369 entry_SYSENTER_compat_after_hwframe+0x84/0x8e CPU: 0 UID: 0 PID: 22408 Comm: syz.4.5165 Not tainted 6.15.0-rc3-syzkaller-00019-gbc3372351d0c #0 PREEMPT(undef) Hardware name: Google Google Compute Engi ---truncated---
CVE-2025-37963	In the Linux kernel, the following vulnerability has been resolved: arm64: bpf: Only mitigate cBPF programs loaded by unprivileged users Support for eBPF programs loaded by unprivileged users is typically disabled. This means only cBPF programs need to be mitigated for BHB. In addition, only mitigate cBPF programs that were loaded by an unprivileged user. Privileged users can also load the same program via eBPF, making the mitigation pointless.
CVE-2025-37967	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: displayport: Fix deadlock This patch introduces the ucsi_con_mutex_lock / ucsi_con_mutex_unlock functions to the UCSI driver. ucsi_con_mutex_lock ensures the connector mutex is only locked if a connection is established and the partner pointer is valid. This resolves a deadlock scenario where ucsi_displayport_remove_partner holds con->mutex waiting for dp_altmode_work to complete while dp_altmode_work attempts to acquire it.
CVE-2025-37968	In the Linux kernel, the following vulnerability has been resolved: iiio: light: opt3001: fix deadlock due to concurrent flag access The threaded IRQ function in this driver is reading the flag twice: once to lock a mutex and once to unlock it. Even though the code setting the flag is designed to prevent it, there are subtle cases where the flag could be true at the mutex_lock stage and false at the mutex_unlock stage. This results in the mutex not being unlocked, resulting in a deadlock. Fix it by making the opt3001_irq() code generally more robust, reading the flag into a variable and using the variable value at both stages.
CVE-2025-37969	In the Linux kernel, the following vulnerability has been resolved: iiio: imu: st_lsm6dsx: fix possible lockup in st_lsm6dsx_read_tagged_fifo Prevent st_lsm6dsx_read_tagged_fifo from falling in an infinite loop in case pattern_len is equal to zero and the device FIFO is not empty.

CVE ID	Description
CVE-2025-37970	In the Linux kernel, the following vulnerability has been resolved: iio: imu: st_lsm6dsx: fix possible lockup in st_lsm6dsx_read_fifo Prevent st_lsm6dsx_read_fifo from falling in an infinite loop in case pattern_len is equal to zero and the device FIFO is not empty.
CVE-2025-37972	In the Linux kernel, the following vulnerability has been resolved: Input: mtk-pmic-keys - fix possible null pointer dereference In mtk_pmic_keys_probe, the regs parameter is only set if the button is parsed in the device tree. However, on hardware where the button is left floating, that node will most likely be removed not to enable that input. In that case the code will try to dereference a null pointer. Let's use the regs struct instead as it is defined for all supported platforms. Note that it is ok setting the key reg even if that latter is disabled as the interrupt won't be enabled anyway.
CVE-2025-37973	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: fix out-of-bounds access during multi-link element defragmentation Currently during the multi-link element defragmentation process, the multi-link element length added to the total IEs length when calculating the length of remaining IEs after the multi-link element in cfg80211_defrag_mle(). This could lead to out-of-bounds access if the multi-link element or its corresponding fragment elements are the last elements in the IEs buffer. To address this issue, correctly calculate the remaining IEs length by deducting the multi-link element end offset from total IEs end offset.
CVE-2025-37974	In the Linux kernel, the following vulnerability has been resolved: s390/pci: Fix missing check for zpci_create_device() error return The zpci_create_device() function returns an error pointer that needs to be checked before dereferencing it as a struct zpci_dev pointer. Add the missing check in __clp_add() where it was missed when adding the scan_list in the fixed commit. Simply not adding the device to the scan list results in the previous behavior.
CVE-2025-37975	In the Linux kernel, the following vulnerability has been resolved: riscv: module: Fix out-of-bounds relocation access The current code allows rel[j] to access one element past the end of the relocation section. Simplify to num_relocations which is equivalent to the existing size expression.
CVE-2025-37977	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: exynos: Disable ioccc if dma-coherent property isn't set If dma-coherent property isn't set then descriptors are non-cacheable and the ioccc shareability bits should be disabled. Without this UFS can end up in an incompatible configuration and suffer from random cache related stability issues.
CVE-2025-37978	In the Linux kernel, the following vulnerability has been resolved: block: integrity: Do not call set_page_dirty_lock() Placing multiple protection information buffers inside the same page can lead to oopses because set_page_dirty_lock() can't be called from interrupt context. Since a protection information buffer is not backed by a file there is no point in setting its page dirty, there is nothing to synchronize. Drop the call to set_page_dirty_lock() and remove the last argument to bio_integrity_unpin_bvec().
CVE-2025-37979	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: Fix sc7280 lpass potential buffer overflow Case values introduced in commit 5f78e1fb7a3e ("ASoC: qcom: Add driver support for audioreach solution") cause out of bounds access in arrays of sc7280 driver data (e.g. in case of RX_CODEC_DMA_RX_0 in sc7280_snd_hw_params()). Redefine LPASS_MAX_PORTS to consider the maximum possible port id for q6dsp as sc7280 driver utilizes some of those values. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-37980	In the Linux kernel, the following vulnerability has been resolved: block: fix resource leak in blk_register_queue() error path When registering a queue fails after blk_mq_sysfs_register() is successful but the function later encounters an error, we need to clean up the blk_mq_sysfs resources. Add the missing blk_mq_sysfs_unregister() call in the error path to properly clean up these resources and prevent a memory leak.
CVE-2025-37982	In the Linux kernel, the following vulnerability has been resolved: wifi: wl1251: fix memory leak in wl1251_tx_work The skb dequeued from tx_queue is lost when wl1251_ps_elp_wakeup fails with a -ETIMEDOUT error. Fix that by queueing the skb back to tx_queue.
CVE-2025-37983	In the Linux kernel, the following vulnerability has been resolved: qibfs: fix _another_ leak failure to allocate inode => leaked dentry... this one had been there since the initial merge; to be fair, if we are that far OOM, the odds of failing at that particular allocation are low...
CVE-2025-37984	In the Linux kernel, the following vulnerability has been resolved: crypto: ecdsa - Harden against integer overflows in DIV_ROUND_UP() Herbert notes that DIV_ROUND_UP() may overflow unnecessarily if an ecDSA implementation's ->key_size() callback returns an unusually large value. Herbert instead suggests (for a division by 8): X / 8 + !! (X & 7) Based on this formula, introduce a generic DIV_ROUND_UP_POW2() macro and use it in lieu of DIV_ROUND_UP() for ->key_size() return values. Additionally, use the macro in ecc_digits_from_bytes(), whose "nbytes" parameter is a ->key_size() return value in some instances, or a user-specified ASN.1 length in the case of ecDSA_get_signature_rs().

CVE ID	Description
CVE-2025-37985	In the Linux kernel, the following vulnerability has been resolved: USB: wdm: close race between wdm_open and wdm_wwan_port_stop Clearing WDM_WWAN_IN_USE must be the last action or we can open a chardev whose URBs are still poisoned
CVE-2025-37986	In the Linux kernel, the following vulnerability has been resolved: usb: typec: class: Invalidate USB device pointers on partner unregistration To avoid using invalid USB device pointers after a Type-C partner disconnects, this patch clears the pointers upon partner unregistration. This ensures a clean state for future connections.
CVE-2025-37987	In the Linux kernel, the following vulnerability has been resolved: pds_core: Prevent possible adminq overflow/stuck condition The pds_core's adminq is protected by the adminq_lock, which prevents more than 1 command to be posted onto it at any one time. This makes it so the client drivers cannot simultaneously post adminq commands. However, the completions happen in a different context, which means multiple adminq commands can be posted sequentially and all waiting on completion. On the FW side, the backing adminq request queue is only 16 entries long and the retry mechanism and/or overflow/stuck prevention is lacking. This can cause the adminq to get stuck, so commands are no longer processed and completions are no longer sent by the FW. As an initial fix, prevent more than 16 outstanding adminq commands so there's no way to cause the adminq from getting stuck. This works because the backing adminq request queue will never have more than 16 pending adminq commands, so it will never overflow. This is done by reducing the adminq depth to 16.
CVE-2025-37988	In the Linux kernel, the following vulnerability has been resolved: fix a couple of races in MNT_TREE_BENEATH handling by do_move_mount() Normally do_lock_mount(path, _) is locking a mountpoint pinned by *path and at the time when matching unlock_mount() unlocks that location it is still pinned by the same thing. Unfortunately, for 'beneath' case it's no longer that simple - the object being locked is not the one *path points to. It's the mountpoint of path->mnt. The thing is, without sufficient locking ->mnt_parent may change under us and none of the locks are held at that point. The rules are * mount_lock stabilizes m->mnt_parent for any mount m. * namespace_sem stabilizes m->mnt_parent, provided that m is mounted. * if either of the above holds and refcount of m is positive, we are guaranteed the same for refcount of m->mnt_parent. namespace_sem nests inside inode_lock(), so do_lock_mount() has to take inode_lock() before grabbing namespace_sem. It does recheck that path->mnt is still mounted in the same place after getting namespace_sem, and it does take care to pin the dentry. It is needed, since otherwise we might end up with racing mount --move (or umount) happening while we were getting locks; in that case dentry would no longer be a mountpoint and could've been evicted on memory pressure along with its inode - not something you want when grabbing lock on that inode. However, pinning a dentry is not enough - the matching mount is also pinned only by the fact that path->mnt is mounted on top it and at that point we are not holding any locks whatsoever, so the same kind of races could end up with all references to that mount gone just as we are about to enter inode_lock(). If that happens, we are left with filesystem being shut down while we are holding a dentry reference on it; results are not pretty. What we need to do is grab both dentry and mount at the same time; that makes inode_lock() safe *and* avoids the problem with fs getting shut down under us. After taking namespace_sem we verify that path->mnt is still mounted (which stabilizes its ->mnt_parent) and check that it's still mounted at the same place. From that point on to the matching namespace_unlock() we are guaranteed that mount/dentry pair we'd grabbed are also pinned by being the mountpoint of path->mnt, so we can quietly drop both the dentry reference (as the current code does) and mnt one - it's OK to do under namespace_sem, since we are not dropping the final refs. That solves the problem on do_lock_mount() side; unlock_mount() also has one, since dentry is guaranteed to stay pinned only until the namespace_unlock(). That's easy to fix - just have inode_unlock() done earlier, while it's still pinned by mp->m_dentry.
CVE-2025-37989	In the Linux kernel, the following vulnerability has been resolved: net: phy: leds: fix memory leak A network restart test on a router led to an out-of-memory condition, which was traced to a memory leak in the PHY LED trigger code. The root cause is misuse of the devm API. The registration function (phy_led_triggers_register) is called from phy_attach_direct, not phy_probe, and the unregister function (phy_led_triggers_unregister) is called from phy_detach, not phy_remove. This means the register and unregister functions can be called multiple times for the same PHY device, but devm-allocated memory is not freed until the driver is unbound. This also prevents kmemleak from detecting the leak, as the devm API internally stores the allocated pointer. Fix this by replacing devm_kzalloc/devm_kcalloc with standard kzalloc/kcalloc, and add the corresponding kfree calls in the unregister path.
CVE-2025-37990	In the Linux kernel, the following vulnerability has been resolved: wifi: brcm80211: fmac: Add error handling for brcmf_usb_dl_writeimage() The function brcmf_usb_dl_writeimage() calls the function brcmf_usb_dl_cmd() but does not check its return value. The 'state.state' and the 'state.bytes' are uninitialized if the function brcmf_usb_dl_cmd() fails. It is dangerous to use uninitialized variables in the conditions. Add error handling for brcmf_usb_dl_cmd() to jump to error handling path if the brcmf_usb_dl_cmd() fails and the 'state.state' and the 'state.bytes' are uninitialized. Improve the error message to report more detailed error information.

CVE ID	Description
CVE-2025-37991	In the Linux kernel, the following vulnerability has been resolved: parisc: Fix double SIGFPE crash Camm noticed that on parisc a SIGFPE exception will crash an application with a second SIGFPE in the signal handler. Dave analyzed it, and it happens because glibc uses a double-word floating-point store to atomically update function descriptors. As a result of lazy binding, we hit a floating-point store in fpe_func almost immediately. When the T bit is set, an assist exception trap occurs when the co-processor encounters *any* floating-point instruction except for a double store of register %fr0. The latter cancels all pending traps. Let's fix this by clearing the Trap (T) bit in the FP status register before returning to the signal handler in userspace. The issue can be reproduced with this test program: root@parisc:~# cat fpe.c static void fpe_func(int sig, siginfo_t *i, void *v) { sigset_t set; sigemptyset(&set); sigaddset(&set, SIGFPE); sigprocmask(SIG_UNBLOCK, &set, NULL); printf("GOT signal %d with si_code %ld\n", sig, i->si_code); } int main() { struct sigaction action = { .sa_sigaction = fpe_func, .sa_flags = SA_RESTART SA_SIGINFO }; sigaction(SIGFPE, &action, 0); feenableexcept(FE_OVERFLOW); return printf("%lf\n", 1.7976931348623158E308*1.7976931348623158E308); } root@parisc:~# gcc fpe.c -lm root@parisc:~# ./a.out Floating point exception root@parisc:~# strace -f ./a.out execve("./a.out", ["/a.out"], 0xf9ac7034 /* 20 vars */) = 0 getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM_INFINITY}) = 0 ... rt_sigaction(SIGFPE, {sa_handler=0x1110a, sa_mask=[], sa_flags=SA_RESTART SA_SIGINFO}, NULL, 8) = 0 --- SIGFPE {si_signo=SIGFPE, si_code=FPE_FLTOVF, si_addr=0x1078f} --- --- SIGFPE {si_signo=SIGFPE, si_code=FPE_FLTOVF, si_addr=0xf8f21237} --- +++ killed by SIGFPE +++ Floating point exception
CVE-2025-37992	In the Linux kernel, the following vulnerability has been resolved: net_sched: Flush gso_skb list too during ->change() Previously, when reducing a qdisc's limit via the ->change() operation, only the main skb queue was trimmed, potentially leaving packets in the gso_skb list. This could result in NULL pointer dereference when we only check sch->limit against sch->q.qlen. This patch introduces a new helper, qdisc_dequeue_internal(), which ensures both the gso_skb list and the main queue are properly flushed when trimming excess packets. All relevant qdiscs (codel, fq, fq_codel, fq_pie, hhf, pie) are updated to use this helper in their ->change() routines.
CVE-2025-37994	In the Linux kernel, the following vulnerability has been resolved: usb: typec: ucsi: displayport: Fix NULL pointer access This patch ensures that the UCSI driver waits for all pending tasks in the ucsi_displayport_work workqueue to finish executing before proceeding with the partner removal.
CVE-2025-37995	In the Linux kernel, the following vulnerability has been resolved: module: ensure that kobject_put() is safe for module type kobjects In 'lookup_or_create_module_kobject()', an internal kobject is created using 'module_ktype'. So call to 'kobject_put()' on error handling path causes an attempt to use an uninitialized completion pointer in 'module_kobject_release()'. In this scenario, we just want to release kobject without an extra synchronization required for a regular module unloading process, so adding an extra check whether 'complete()' is actually required makes 'kobject_put()' safe.
CVE-2025-37997	In the Linux kernel, the following vulnerability has been resolved: netfilter: ipset: fix region locking in hash types Region locking introduced in v5.6-rc4 contained three macros to handle the region locks: ahash_bucket_start(), ahash_bucket_end() which gave back the start and end hash bucket values belonging to a given region lock and ahash_region() which should give back the region lock belonging to a given hash bucket. The latter was incorrect which can lead to a race condition between the garbage collector and adding new elements when a hash type of set is defined with timeouts.
CVE-2025-37998	In the Linux kernel, the following vulnerability has been resolved: openvswitch: Fix unsafe attribute parsing in output_userspace() This patch replaces the manual Netlink attribute iteration in output_userspace() with nla_for_each_nested(), which ensures that only well-formed attributes are processed.
CVE-2025-38000	In the Linux kernel, the following vulnerability has been resolved: sch_hfsc: Fix qlen accounting bug when using peek in hfsc_enqueue() When enqueueing the first packet to an HFSC class, hfsc_enqueue() calls the child qdisc's peek() operation before incrementing sch->q.qlen and sch->qstats.backlog. If the child qdisc uses qdisc_peek_dequeued(), this may trigger an immediate dequeue and potential packet drop. In such cases, qdisc_tree_reduce_backlog() is called, but the HFSC qdisc's qlen and backlog have not yet been updated, leading to inconsistent queue accounting. This can leave an empty HFSC class in the active list, causing further consequences like use-after-free. This patch fixes the bug by moving the increment of sch->q.qlen and sch->qstats.backlog before the call to the child qdisc's peek() operation. This ensures that queue length and backlog are always accurate when packet drops or dequeues are triggered during the peek.

CVE ID	Description
CVE-2025-38001	In the Linux kernel, the following vulnerability has been resolved: net_sched: hfsc: Address reentrant enqueue adding class to eltree twice Savino says: "We are writing to report that this recent patch (141d34391abbb315d68556b7c67ad97885407547) [1] can be bypassed, and a UAF can still occur when HFSC is utilized with NETEM. The patch only checks the cl->cl_nactive field to determine whether it is the first insertion or not [2], but this field is only incremented by init_vf [3]. By using HFSC_RSC (which uses init_ed) [4], it is possible to bypass the check and insert the class twice in the eltree. Under normal conditions, this would lead to an infinite loop in hfsc_dequeue for the reasons we already explained in this report [5]. However, if TBF is added as root qdisc and it is configured with a very low rate, it can be utilized to prevent packets from being dequeued. This behavior can be exploited to perform subsequent insertions in the HFSC eltree and cause a UAF." To fix both the UAF and the infinite loop, with netem as an hfsc child, check explicitly in hfsc_enqueue whether the class is already in the eltree whenever the HFSC_RSC flag is set. [1] https://web.git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=141d34391abbb315d68556b7c67ad97885407547 [2] https://elixir.bootlin.com/linux/v6.15-rc5/source/net/sched/sch_hfsc.c#L1572 [3] https://elixir.bootlin.com/linux/v6.15-rc5/source/net/sched/sch_hfsc.c#L677 [4] https://elixir.bootlin.com/linux/v6.15-rc5/source/net/sched/sch_hfsc.c#L1574 [5] https://lore.kernel.org/netdev/8DuRWwfqjoRDLDmBMlIfbrsZg9Gx50DHJc1ilxsEBNe2D6NMoigR_eIRIG0LOjMc3r10nUUZtArXx4oZBldUfZT/#u
CVE-2025-38003	In the Linux kernel, the following vulnerability has been resolved: can: bcm: add missing rcu read protection for procfs content When the procfs content is generated for a bcm_op which is in the process to be removed the procfs output might show unreliable data (UAF). As the removal of bcm_op's is already implemented with rcu handling this patch adds the missing rcu_read_lock() and makes sure the list entries are properly removed under rcu protection.
CVE-2025-38004	In the Linux kernel, the following vulnerability has been resolved: can: bcm: add locking for bcm_op runtime updates The CAN broadcast manager (CAN BCM) can send a sequence of CAN frames via hrtimer. The content and also the length of the sequence can be changed resp reduced at runtime where the 'currframe' counter is then set to zero. Although this appeared to be a safe operation the updates of 'currframe' can be triggered from user space and hrtimer context in bcm_can_tx(). Anderson Nascimento created a proof of concept that triggered a KASAN slab-out-of-bounds read access which can be prevented with a spin_lock_bh. At the rework of bcm_can_tx() the 'count' variable has been moved into the protected section as this variable can be modified from both contexts too.
CVE-2025-38005	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: k3-udma: Add missing locking Recent kernels complain about a missing lock in k3-udma.c when the lock validator is enabled: [4.128073] WARNING: CPU: 0 PID: 746 at drivers/dma/ti/./virt-dma.h:169 udma_start.isra.0+0x34/0x238 [4.137352] CPU: 0 UID: 0 PID: 746 Comm: kworker/0:3 Not tainted 6.12.9-arm64 #28 [4.144867] Hardware name: pp-v12 (DT) [4.148648] Workqueue: events udma_check_tx_completion [4.153841] pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=) [4.160834] pc : udma_start.isra.0+0x34/0x238 [4.165227] lr : udma_start.isra.0+0x30/0x238 [4.169618] sp : fffffffc083cabcf0 [4.172963] x29: fffffffc083cabcf0 x28: 0000000000000000 x27: fffffff800001b005 [4.180167] x26: fffffff8012f0000 x25: 0000000000000000 x24: 0000000000000000 [4.187370] x23: 0000000000000001 x22: 00000000e21eabe9 x21: fffffff8000fa0670 [4.194571] x20: fffffff8001b6bf00 x19: fffffff8000fa0430 x18: fffffffc083b95030 [4.201773] x17: 0000000000000000 x16: 00000000f0000000 x15: 0000000000000048 [4.208976] x14: 0000000000000048 x13: 0000000000000000 x12: 0000000000000001 [4.216179] x11: fffffffc08151a240 x10: 00000000000003ea1 x9 : fffffffc08046ab68 [4.223381] x8 : fffffffc083cabac0 x7 : fffffffc081df3718 x6 : 0000000000029fc8 [4.230583] x5 : fffffffc0817ee6d8 x4 : 0000000000000bc0 x3 : 0000000000000000 [4.237784] x2 : 0000000000000000 x1 : 00000000001fffff x0 : 0000000000000000 [4.244986] Call trace: [4.247463] udma_start.isra.0+0x34/0x238 [4.251509] udma_check_tx_completion+0xd0/0xdc [4.256076] process_one_work+0x244/0x3fc [4.260129] process_scheduled_works+0x6c/0x74 [4.264610] worker_thread+0x150/0x1dc [4.268398] kthread+0xd8/0xe8 [4.271492] ret_from_fork+0x10/0x20 [4.275107] irq event stamp: 220 [4.278363] hardirqs last enabled at (219): [<ffffffc080a27c7c>] _raw_spin_unlock_irq+0x38/0x50 [4.287183] hardirqs last disabled at (220): [<ffffffc080a1c154>] el1_dbg+0x24/0x50 [4.294879] softirqs last enabled at (182): [<ffffffc080037e68>] handle_softirqs+0x1c0/0x3cc [4.303437] softirqs last disabled at (177): [<ffffffc080010170>] __do_softirq+0x1c/0x28 [4.311559] ---[end trace 0000000000000000]--- This commit adds the missing locking.
CVE-2025-38006	In the Linux kernel, the following vulnerability has been resolved: net: mctp: Don't access ifa_index when missing In mctp_dump_addrinfo, ifa_index can be used to filter interfaces, but only when the struct ifaddrmsg is provided. Otherwise it will be comparing to uninitialised memory - reproducible in the syzkaller case from dhcpd, or busybox "ip addr show". The kernel MCTP implementation has always filtered by ifa_index, so existing userspace programs expecting to dump MCTP addresses must already be passing a valid ifa_index value (either 0 or a real index). BUG: KMSAN: uninit-value in mctp_dump_addrinfo+0x208/0xac0 net/mctp/device.c:128 mctp_dump_addrinfo+0x208/0xac0 net/mctp/device.c:128 rtnl_dump_all+0x3ec/0x5b0 net/core/rtnetlink.c:4380 rtnl_dumpit+0xd5/0x2f0 net/core/rtnetlink.c:6824 netlink_dump+0x97b/0x1690 net/netlink/af_netlink.c:2309

CVE ID	Description
CVE-2025-38007	In the Linux kernel, the following vulnerability has been resolved: HID: uclogic: Add NULL check in uclogic_input_configured() devm_kasprintf() returns NULL when memory allocation fails. Currently, uclogic_input_configured() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue.
CVE-2025-38008	In the Linux kernel, the following vulnerability has been resolved: mm/page_alloc: fix race condition in unaccepted memory handling The page allocator tracks the number of zones that have unaccepted memory using static_branch_enc/dec() and uses that static branch in hot paths to determine if it needs to deal with unaccepted memory. Borislav and Thomas pointed out that the tracking is racy: operations on static_branch are not serialized against adding/removing unaccepted pages to/from the zone. Sanity checks inside static_branch machinery detects it: WARNING: CPU: 0 PID: 10 at kernel/jump_label.c:276 __static_key_slow_dec_cpuslocked+0x8e/0xa0 The comment around the WARN() explains the problem: /* * Warn about the '-1' case though; since that means a * decrement is concurrent with a first (0->1) increment. IOW * people are trying to disable something that wasn't yet fully * enabled. This suggests an ordering problem on the user side. */ The effect of this static_branch optimization is only visible on microbenchmark. Instead of adding more complexity around it, remove it altogether.
CVE-2025-38009	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: disable napi on driver removal A warning on driver removal started occurring after commit 9dd05df8403b ("net: warn if NAPI instance wasn't shut down"). Disable tx napi before deleting it in mt76_dma_cleanup(). WARNING: CPU: 4 PID: 18828 at net/core/dev.c:7288 __netif_napi_del_locked+0xf0/0x100 CPU: 4 UID: 0 PID: 18828 Comm: modprobe Not tainted 6.15.0-rc4 #4 PREEMPT(lazy) Hardware name: ASUS System Product Name/PRIME X670E-PRO WIFI, BIOS 3035 09/05/2024 RIP: 0010: __netif_napi_del_locked+0xf0/0x100 Call Trace: <TASK> mt76_dma_cleanup +0x54/0x2f0 [mt76] mt7921_pci_remove+0xd5/0x190 [mt7921e] pci_device_remove+0x47/0xc0 device_release_driver_internal+0x19e/0x200 driver_detach+0x48/0x90 bus_remove_driver +0x6d/0xf0 pci_unregister_driver+0x2e/0xb0 __do_sys_delete_module.isra.0+0x197/0x2e0 do_syscall_64+0x7b/0x160 entry_SYSCALL_64_after_hwframe+0x76/0x7e Tested with mt7921e but the same pattern can be actually applied to other mt76 drivers calling mt76_dma_cleanup() during removal. Tx napi is enabled in their *_dma_init() functions and only toggled off and on again inside their suspend/resume/reset paths. So it should be okay to disable tx napi in such a generic way. Found by Linux Verification Center (linuxtesting.org).
CVE-2025-38010	In the Linux kernel, the following vulnerability has been resolved: phy: tegra: xusb: Use a bitmask for UTMI pad power state tracking The current implementation uses bias_pad_enable as a reference count to manage the shared bias pad for all UTMI PHYs. However, during system suspension with connected USB devices, multiple power-down requests for the UTMI pad result in a mismatch in the reference count, which in turn produces warnings such as: [237.762967] WARNING: CPU: 10 PID: 1618 at tegra186_utmi_pad_power_down+0x160/0x170 [237.763103] Call trace: [237.763104] tegra186_utmi_pad_power_down+0x160/0x170 [237.763107] tegra186_utmi_phy_power_off +0x10/0x30 [237.763110] phy_power_off+0x48/0x100 [237.763113] tegra_xusb_enter_elpg +0x204/0x500 [237.763119] tegra_xusb_suspend+0x48/0x140 [237.763122] platform_pm_suspend +0x2c/0xb0 [237.763125] dpm_run_callback.isra.0+0x20/0xa0 [237.763127] __device_suspend +0x118/0x330 [237.763129] dpm_suspend+0x10c/0x1f0 [237.763130] dpm_suspend_start+0x88/0xb0 [237.763132] suspend_devices_and_enter+0x120/0x500 [237.763135] pm_suspend+0x1ec/0x270 The root cause was traced back to the dynamic power-down changes introduced in commit a30951d31b25 ("xhci: tegra: USB2 pad power controls"), where the UTMI pad was being powered down without verifying its current state. This unbalanced behavior led to discrepancies in the reference count. To rectify this issue, this patch replaces the single reference counter with a bitmask, renamed to utmi_pad_enabled. Each bit in the mask corresponds to one of the four USB2 PHYs, allowing us to track each pad's enablement status individually. With this change: - The bias pad is powered on only when the mask is clear. - Each UTMI pad is powered on or down based on its corresponding bit in the mask, preventing redundant operations. - The overall power state of the shared bias pad is maintained correctly during suspend/resume cycles. The mutex used to prevent race conditions during UTMI pad enable/disable operations has been moved from the tegra186_utmi_bias_pad_power_on/off functions to the parent functions tegra186_utmi_pad_power_on/down. This change ensures that there are no race conditions when updating the bitmask.

CVE ID	Description
CVE-2025-38011	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: csa unmap use uninterruptible lock After process exit to unmap csa and free GPU vm, if signal is accepted and then waiting to take vm lock is interrupted and return, it causes memory leaking and below warning backtrace. Change to use uninterruptible wait lock fix the issue. WARNING: CPU: 69 PID: 167800 at amd/amdgpu/amdgpu_kms.c:1525 amdgpu_driver_postclose_kms+0x294/0x2a0 [amdgpu] Call Trace: <TASK> drm_file_free.part.0+0x1da/0x230 [drm] drm_close_helper.isra.0+0x65/0x70 [drm] drm_release+0x6a/0x120 [drm] amdgpu_drm_release+0x51/0x60 [amdgpu] __fput+0x9f/0x280 ____fput+0xe/0x20 task_work_run+0x67/0xa0 do_exit+0x217/0x3c0 do_group_exit+0x3b/0xb0 get_signal+0x14a/0x8d0 arch_do_signal_or_restart+0xde/0x100 exit_to_user_mode_loop+0xc1/0x1a0 exit_to_user_mode_prepare+0xf4/0x100 syscall_exit_to_user_mode+0x17/0x40 do_syscall_64+0x69/0xc0 (cherry picked from commit 7dbbf3c171a6f63b01165958629c9c26abf38ab)
CVE-2025-38013	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: Set n_channels after allocating struct cfg80211_scan_request Make sure that n_channels is set after allocating the struct cfg80211_registered_device::int_scan_req member. Seen with syzkaller: UBSAN: array-index-out-of-bounds in net/mac80211/scan.c:1208:5 index 0 is out of range for type 'struct ieee80211_channel *[] __counted_by(n_channels)' (aka 'struct ieee80211_channel *[]') This was missed in the initial conversions because I failed to locate the allocation likely due to the "sizeof(void *)" not matching the "channels" array type.
CVE-2025-38014	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Refactor remove call with idxd_cleanup() helper The idxd_cleanup() helper cleans up perfmon, interrupts, internals and so on. Refactor remove call with the idxd_cleanup() helper to avoid code duplication. Note, this also fixes the missing put_device() for idxd groups, engines and wqs.
CVE-2025-38015	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: fix memory leak in error handling path of idxd_alloc Memory allocated for idxd is not freed if an error occurs during idxd_alloc(). To fix it, free the allocated memory in the reverse order of allocation before exiting the function in case of an error.
CVE-2025-38017	In the Linux kernel, the following vulnerability has been resolved: fs/eventpoll: fix endless busy loop after timeout has expired After commit 0a65bc27bd64 ("eventpoll: Set epoll timeout if it's in the future"), the following program would immediately enter a busy loop in the kernel: ```` int main() { int e = epoll_create1(0); struct epoll_event event = {.events = EPOLLIN}; epoll_ctl(e, EPOLL_CTL_ADD, 0, &event); const struct timespec timeout = {.tv_nsec = 1}; epoll_pwait2(e, &event, 1, &timeout, 0); } ```` This happens because the given (non-zero) timeout of 1 nanosecond usually expires before ep_poll() is entered and then ep_schedule_timeout() returns false, but `timed_out` is never set because the code line that sets it is skipped. This quickly turns into a soft lockup, RCU stalls and deadlocks, inflicting severe headaches to the whole system. When the timeout has expired, we don't need to schedule a hrtimer, but we should set the `timed_out` variable. Therefore, I suggest moving the ep_schedule_timeout() check into the `timed_out` expression instead of skipping it. brauner: Note that there was an earlier fix by Joe Damato in response to my bug report in [1].
CVE-2025-38018	In the Linux kernel, the following vulnerability has been resolved: net/tls: fix kernel panic when alloc_page failed We cannot set frag_list to NULL pointer when alloc_page failed. It will be used in tls_strp_check_queue_ok when the next time tls_strp_read_sock is called. This is because we don't reset full_len in tls_strp_flush_anchor_copy() so the rcv path will try to continue handling the partial record on the next call but we detached the rcvq from the frag list. Alternative fix would be to reset full_len. Unable to handle kernel NULL pointer dereference at virtual address 0000000000000028 Call trace: tls_strp_check_rcv+0x128/0x27c tls_strp_data_ready+0x34/0x44 tls_data_ready+0x3c/0x1f0 tcp_data_ready+0x9c/0xe4 tcp_data_queue+0xf6c/0x12d0 tcp_rcv_established+0x52c/0x798

CVE ID	Description
CVE-2025-38019	In the Linux kernel, the following vulnerability has been resolved: mlxsw: spectrum_router: Fix use-after-free when deleting GRE net devices The driver only offloads neighbors that are constructed on top of net devices registered by it or their uppers (which are all Ethernet). The device supports GRE encapsulation and decapsulation of forwarded traffic, but the driver will not offload dummy neighbors constructed on top of GRE net devices as they are not uppers of its net devices: # ip link add name gre1 up type gre tos inherit local 192.0.2.1 remote 198.51.100.1 # ip neigh add 0.0.0.0 lladdr 0.0.0.0 nud noarp dev gre1 \$ ip neigh show dev gre1 nud noarp 0.0.0.0 lladdr 0.0.0.0 NOARP (Note that the neighbor is not marked with 'offload') When the driver is reloaded and the existing configuration is replayed, the driver does not perform the same check regarding existing neighbors and offloads the previously added one: # devlink dev reload pci/0000:01:00.0 \$ ip neigh show dev gre1 nud noarp 0.0.0.0 lladdr 0.0.0.0 offload NOARP If the neighbor is later deleted, the driver will ignore the notification (given the GRE net device is not its upper) and will therefore keep referencing freed memory, resulting in a use-after-free [1] when the net device is deleted: # ip neigh del 0.0.0.0 lladdr 0.0.0.0 dev gre1 # ip link del dev gre1 Fix by skipping neighbor replay if the net device for which the replay is performed is not our upper. [1] BUG: KASAN: slab-use-after-free in mlxsw_sp_neigh_entry_update+0x1ea/0x200 Read of size 8 at addr ffff888155b0e420 by task ip/2282 [...] Call Trace: <TASK> dump_stack_lvl+0x6f/0xa0 print_address_description.constprop.0+0x6f/0x350 print_report+0x108/0x205 kasan_report+0xdf/0x110 mlxsw_sp_neigh_entry_update+0x1ea/0x200 mlxsw_sp_router_rif_gone_sync+0x2a8/0x440 mlxsw_sp_rif_destroy+0x1e9/0x750 mlxsw_sp_netdevice_ipip_ol_event+0x3c9/0xdc0 mlxsw_sp_router_netdevice_event+0x3ac/0x15e0 notifier_call_chain+0xca/0x150 call_netdevice_notifiers_info+0x7f/0x100 unregister_netdevice_many_notify+0xc8c/0x1d90 rtnl_dellink+0x34e/0xa50 rtnetlink_rcv_msg+0x6fb/0xb70 netlink_rcv_skb+0x131/0x360 netlink_unicast+0x426/0x710 netlink_sendmsg+0x75a/0xc20 __sock_sendmsg+0xc1/0x150 ____sys_sendmsg+0x5aa/0x7b0 ___sys_sendmsg+0xfc/0x180 __sys_sendmsg+0x121/0x1b0 do_syscall_64+0xbb/0x1d0 entry_SYSCALL_64_after_hwframe+0x4b/0x53

CVE ID	Description
CVE-2025-38020	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Disable MACsec offload for uplink representor profile MACsec offload is not supported in switchdev mode for uplink representors. When switching to the uplink representor profile, the MACsec offload feature must be cleared from the netdevice's features. If left enabled, attempts to add offloads result in a null pointer dereference, as the uplink representor does not support MACsec offload even though the feature bit remains set. Clear NETIF_F_HW_MACSEC in mlx5e_fix_uplink_rep_features(). Kernel log: Oops: general protection fault, probably for non-canonical address 0xdffffc000000000f: 0000 [#1] SMP KASAN KASAN: null-ptr-deref in range [0x0000000000000078-0x000000000000007f] CPU: 29 UID: 0 PID: 4714 Comm: ip Not tainted 6.14.0-rc4_for_upstream_debug_2025_03_02_17_35 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 RIP: 0010: __mutex_lock+0x128/0x1dd0 Code: d0 7c 08 84 d2 0f 85 ad 15 00 00 8b 35 91 5c fe 03 85 f6 75 29 49 8d 7e 60 48 b8 00 00 00 00 00 fc ff df 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 a6 15 00 00 4d 3b 76 60 0f 85 fd 0b 00 00 65 ff RSP: 0018:ffff888147a4f160 EFLAGS: 00010206 RAX: dffffc0000000000 RBX: 0000000000000000 RCX: 0000000000000001 RDX: 000000000000000f RSI: 0000000000000000 RDI: 0000000000000078 RBP: ffff888147a4f2e0 R08: ffffffff05d2c19 R09: 0000000000000000 DR6: 0000000000000000 R10: 0000000000000001 R11: 0000000000000000 R12: 0000000000000000 R13: dffffc0000000000 R14: 0000000000000018 R15: ffff888152de0000 FS: 00007f85e27d800(0000) GS:ffff88881ee80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00000000004e5768 CR3: 000000013ae7c005 CR4: 000000000372eb0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 0000000000000000 DR7: 00000000000000400 Call Trace: <TASK> ? die_addr+0x3d/0xa0 ? exc_general_protection+0x144/0x220 ? asm_exc_general_protection+0x22/0x30 ? mlx5e_macsec_add_secy+0xf9/0x700 [mlx5_core] ? __mutex_lock+0x128/0x1dd0 ? lockdep_set_lock_cmp_fn+0x190/0x190 ? mlx5e_macsec_add_secy+0xf9/0x700 [mlx5_core] ? mutex_lock_io_nested+0x1ae0/0x1ae0 ? lock_acquire+0x1c2/0x530 ? macsec_upd_offload+0x145/0x380 ? lockdep_hardirqs_on_prepare+0x400/0x400 ? kasan_save_stack+0x30/0x40 ? kasan_save_stack+0x20/0x40 ? kasan_save_track+0x10/0x30 ? __kasan_kmalloc+0x77/0x90 ? __kmalloc_noprof+0x249/0x6b0 ? genl_family_rcv_msg_attrs_parse.constprop.0+0xb5/0x240 ? mlx5e_macsec_add_secy+0xf9/0x700 [mlx5_core] mlx5e_macsec_add_secy+0xf9/0x700 [mlx5_core] ? mlx5e_macsec_add_rxsa+0x11a0/0x11a0 [mlx5_core] macsec_update_offload+0x26c/0x820 ? macsec_set_mac_address+0x4b0/0x4b0 ? lockdep_hardirqs_on_prepare+0x284/0x400 ? _raw_spin_unlock_irqrestore+0x47/0x50 macsec_upd_offload+0x2c8/0x380 ? macsec_update_offload+0x820/0x820 ? __nla_parse+0x22/0x30 ? genl_family_rcv_msg_attrs_parse.constprop.0+0x15e/0x240 genl_family_rcv_msg_doit+0x1cc/0x2a0 ? genl_family_rcv_msg_attrs_parse.constprop.0+0x240/0x240 ? cap_capable+0xd4/0x330 genl_rcv_msg+0x3ea/0x670 ? genl_family_rcv_msg_dumpit+0x2a0/0x2a0 ? lockdep_set_lock_cmp_fn+0x190/0x190 ? macsec_update_offload+0x820/0x820 netlink_rcv_skb+0x12b/0x390 ? genl_family_rcv_msg_dumpit+0x2a0/0x2a0 ? netlink_ack+0xd80/0xd80 ? rwsem_down_read_slowpath+0xf90/0xf90 ? netlink_deliver_tap+0xcd/0xac0 ? netlink_deliver_tap+0x155/0xac0 ? _copy_from_iter+0x1bb/0x12c0 genl_rcv+0x24/0x40 netlink_unicast+0x440/0x700 ? netlink_attachskb+0x760/0x760 ? lock_acquire+0x1c2/0x530 ? __might_fault+0xbb/0x170 netlink_sendmsg+0x749/0xc10 ? netlink_unicast+0x700/0x700 ? __might_fault+0xbb/0x170 ? netlink_unicast+0x700/0x700 __sock_sendmsg+0xc5/0x190 __sys_sendmsg+0x53f/0x760 ? import_iovec+0x7/0x10 ? kernel_sendmsg+0x30/0x30 ? __copy_msghdr+0x3c0/0x3c0 ? filter_irq_stacks+0x90/0x90 ? stack_depot_save_flags+0x28/0xa30 __sys_sendmsg+0x53f/0x760 ---truncated---
CVE-2025-38022	In the Linux kernel, the following vulnerability has been resolved: RDMA/core: Fix "KASAN: slab-use-after-free Read in ib_register_device" problem Call Trace: __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xc3/0x670 mm/kasan/report.c:521 kasan_report+0xe0/0x110 mm/kasan/report.c:634 strlen+0x93/0xa0 lib/string.c:420 __fortify_strlen include/linux/fortify-string.h:268 [inline] get_kobj_path_length lib/kobject.c:118 [inline] kobject_get_path+0x3f/0x2a0 lib/kobject.c:158 kobject_uevent_env+0x289/0x1870 lib/kobject_uevent.c:545 ib_register_device drivers/infiniband/core/device.c:1472 [inline] ib_register_device+0x8cf/0xe00 drivers/infiniband/core/device.c:1393 rxe_register_device+0x275/0x320 drivers/infiniband/sw/rxe/rxe_verbs.c:1552 rxe_net_add+0x8e/0xe0 drivers/infiniband/sw/rxe/rxe_net.c:550 rxe_newlink+0x70/0x190 drivers/infiniband/sw/rxe/rxe.c:225 nldev_newlink+0x3a3/0x680 drivers/infiniband/core/nldev.c:1796 rdma_nl_rcv_msg+0x387/0x6e0 drivers/infiniband/core/netlink.c:195 rdma_nl_rcv_skb.constprop.0.isra.0+0x2e5/0x450 netlink_unicast_kernel net/netlink/af_netlink.c:1313 [inline] netlink_unicast+0x53a/0x7f0 net/netlink/af_netlink.c:1339 netlink_sendmsg+0x8d1/0xdd0 net/netlink/af_netlink.c:1883 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg net/socket.c:727 [inline] __sys_sendmsg+0xa95/0xc70 net/socket.c:2566 __sys_sendmsg+0x134/0x1d0 net/socket.c:2620 __sys_sendmsg+0x16d/0x220 net/socket.c:2652 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x260 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f This problem is similar to the problem that the commit 1d6a9e7449e2 ("RDMA/core: Fix use-after-free when rename device name") fixes. The root cause is: the function ib_device_rename() renames the name with lock. But in the function kobject_uevent(), this name is accessed without lock protection at the same time. The solution is to add the lock protection when this name is accessed in the function kobject_uevent().

CVE ID	Description
CVE-2025-38023	In the Linux kernel, the following vulnerability has been resolved: nfs: handle failure of nfs_get_lock_context in unlock path When memory is insufficient, the allocation of nfs_lock_context in nfs_get_lock_context() fails and returns -ENOMEM. If we mistakenly treat an nfs4_unlockdata structure (whose l_ctx member has been set to -ENOMEM) as valid and proceed to execute rpc_run_task(), this will trigger a NULL pointer dereference in nfs4_locku_prepare. For example: BUG: kernel NULL pointer dereference, address: 000000000000000c PGD 0 P4D 0 Oops: Oops: 0000 [#1] SMP PTI CPU: 15 UID: 0 PID: 12 Comm: kworker/u64:0 Not tainted 6.15.0-rc2-dirty #60 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-2.fc40 Workqueue: rpciod rpc_async_schedule RIP: 0010:nfs4_locku_prepare+0x35/0xc2 Code: 89 f2 48 89 fd 48 c7 68 69 ef b5 53 48 8b 8e 90 00 00 00 48 89 f3 RSP: 0018:ffffbafcf006bdb8 EFLAGS: 00010246 RAX: 000000000000004b RBX: ffff9b964fc1fa00 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffffffff9bafcf00 RDI: ffff9ba53fddbf40 RBP: ffff9ba539934000 R08: 0000000000000000 R09: ffff9bafcf006bc38 R10: ffffffff9b6b689c8 R11: 0000000000000003 R12: ffff9ba539934030 R13: 0000000000000001 R14: 0000000004248060 R15: ffffffff9b56d1c30 FS: 0000000000000000(0000) GS:ffff9ba5881f0000(0000) knlGS:00000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000000000c CR3: 0000000093f244000 CR4: 000000000000006f0 Call Trace: <TASK> __rpc_execute+0x5bc/0x480 rpc_async_schedule+0x2f/0x40 process_one_work+0x232/0x5d0 worker_thread+0x1da/0x3d0 ? __pfx_worker_thread+0x10/0x10 kthread+0x10d/0x240 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x34/0x50 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Modules linked in: CR2: 000000000000000c ---[end trace 0000000000000000]--- Free the allocated nfs4_unlockdata when nfs_get_lock_context() fails and return NULL to terminate subsequent rpc_run_task, preventing NULL pointer dereference.
CVE-2025-38024	In the Linux kernel, the following vulnerability has been resolved: RDMA/rxe: Fix slab-use-after-free Read in rxe_queue_cleanup bug Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x7d/0xa0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xcf/0x610 mm/kasan/report.c:489 kasan_report+0xb5/0xe0 mm/kasan/report.c:602 rxe_queue_cleanup+0xd0/0xe0 drivers/infiniband/sw/rxe/rxe_queue.c:195 rxe_cq_cleanup+0x3f/0x50 drivers/infiniband/sw/rxe/rxe_cq.c:132 __rxe_cleanup+0x168/0x300 drivers/infiniband/sw/rxe/rxe_pool.c:232 rxe_create_cq+0x22e/0x3a0 drivers/infiniband/sw/rxe/rxe_verbs.c:1109 create_cq+0x658/0xb90 drivers/infiniband/core/uverbs_cmd.c:1052 ib_uverbs_create_cq+0xc7/0x120 drivers/infiniband/core/uverbs_cmd.c:1095 ib_uverbs_write+0x969/0xc90 drivers/infiniband/core/uverbs_main.c:679 vfs_write fs/read_write.c:677 [inline] vfs_write+0x26a/0xcc0 fs/read_write.c:659 ksys_write+0x1b8/0x200 fs/read_write.c:731 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xaa/0x1b0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f In the function rxe_create_cq, when rxe_cq_from_init fails, the function rxe_cleanup will be called to handle the allocated resources. In fact, some memory resources have already been freed in the function rxe_cq_from_init. Thus, this problem will occur. The solution is to let rxe_cleanup do all the work.
CVE-2025-38027	In the Linux kernel, the following vulnerability has been resolved: regulator: max20086: fix invalid memory access max20086_parse_regulators_dt() calls of_regulator_match() using an array of struct of_regulator_match allocated on the stack for the matches argument. of_regulator_match() calls devm_of_regulator_put_matches(), which calls devres_alloc() to allocate a struct devm_of_regulator_matches which will be de-allocated using devm_of_regulator_put_matches(). struct devm_of_regulator_matches is populated with the stack allocated matches array. If the device fails to probe, devm_of_regulator_put_matches() will be called and will try to call of_node_put() on that stack pointer, generating the following dmesg entries: max20086 6-0028: Failed to read DEVICE_ID reg: -121 kobject: '\xc0\$xa5\x03' (000000002cebcb7a): is not initialized, yet kobject_put() is being called. Followed by a stack trace matching the call flow described above. Switch to allocating the matches array using devm_kcalloc() to avoid accessing the stack pointer long after it's out of scope. This also has the advantage of allowing multiple max20086 to probe without overriding the data stored inside the global of_regulator_match.

CVE ID	Description
CVE-2025-38029	In the Linux kernel, the following vulnerability has been resolved: kasan: avoid sleepable page allocation from atomic context apply_to_pte_range() enters the lazy MMU mode and then invokes kasan_populate_vmalloc_pte() callback on each page table walk iteration. However, the callback can go into sleep when trying to allocate a single page, e.g. if an architecture disables preemption on lazy MMU mode enter. On s390 if make arch_enter_lazy_mmu_mode() -> preempt_enable() and arch_leave_lazy_mmu_mode() -> preempt_disable(), such crash occurs: [0.663336] BUG: sleeping function called from invalid context at ./include/linux/sched/mm.h:321 [0.663348] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 2, name: kthreadd [0.663358] preempt_count: 1, expected: 0 [0.663366] RCU nest depth: 0, expected: 0 [0.663375] no locks held by kthreadd/2. [0.663383] Preemption disabled at: [0.663386] [<0002f3284cbb4eda>] apply_to_pte_range+0xfa/0x4a0 [0.663405] CPU: 0 UID: 0 PID: 2 Comm: kthreadd Not tainted 6.15.0-rc5-gcc-kasan-00043-gd76bb1ebb558-dirty #162 PREEMPT [0.663408] Hardware name: IBM 3931 A01 701 (KVM/Linux) [0.663409] Call Trace: [0.663410] [<0002f3284c385f58>] dump_stack_lvl+0xe8/0x140 [0.663413] [<0002f3284c507b9e>] __might_resched+0x66e/0x700 [0.663415] [<0002f3284cc4f6c0>] __alloc_frozen_pages_noprof+0x370/0x4b0 [0.663419] [<0002f3284ccc73c0>] alloc_pages_mpol+0x1a0/0x4a0 [0.663421] [<0002f3284ccc8518>] alloc_frozen_pages_noprof+0x88/0xc0 [0.663424] [<0002f3284ccc8572>] alloc_pages_noprof+0x22/0x120 [0.663427] [<0002f3284cc341ac>] get_free_pages_noprof+0x2c/0xc0 [0.663429] [<0002f3284cceb70>] kasan_populate_vmalloc_pte+0x50/0x120 [0.663433] [<0002f3284cbb4ef8>] apply_to_pte_range+0x118/0x4a0 [0.663435] [<0002f3284cbc7c14>] apply_to_pmd_range+0x194/0x3e0 [0.663437] [<0002f3284cbc99be>] __apply_to_page_range+0x2fe/0x7a0 [0.663440] [<0002f3284cbc9e88>] apply_to_page_range+0x28/0x40 [0.663442] [<0002f3284cceb12>] kasan_populate_vmalloc+0x82/0xa0 [0.663445] [<0002f3284cc1578c>] alloc_vmap_area+0x34c/0xc10 [0.663448] [<0002f3284cc1c2a6>] __get_vm_area_node+0x186/0x2a0 [0.663451] [<0002f3284cc1e696>] __vmalloc_node_range_noprof+0x116/0x310 [0.663454] [<0002f3284cc1d950>] __vmalloc_node_noprof+0xd0/0x110 [0.663457] [<0002f3284c454b88>] alloc_thread_stack_node+0xf8/0x330 [0.663460] [<0002f3284c458d56>] dup_task_struct+0x66/0x4d0 [0.663463] [<0002f3284c45be90>] copy_process+0x280/0x4b90 [0.663465] [<0002f3284c460940>] kernel_clone+0xd0/0x4b0 [0.663467] [<0002f3284c46115e>] kernel_thread+0xbe/0xe0 [0.663469] [<0002f3284c4e440e>] kthreadd+0x50e/0x7f0 [0.663472] [<0002f3284c38c04a>] __ret_from_fork+0x8a/0xf0 [0.663475] [<0002f3284ed57ff2>] ret_from_fork+0xa/0x38 Instead of allocating single pages per-PTE, bulk-allocate the shadow memory prior to applying kasan_populate_vmalloc_pte() callback on a page range.
CVE-2025-38031	In the Linux kernel, the following vulnerability has been resolved: padata: do not leak refcount in reorder_work A recent patch that addressed a UAF introduced a reference count leak: the parallel_data refcount is incremented unconditionally, regardless of the return value of queue_work(). If the work item is already queued, the incremented refcount is never decremented. Fix this by checking the return value of queue_work() and decrementing the refcount when necessary. Resolves: Unreferenced object 0xffff9d9f421e3d80 (size 192): comm "cryptomgr_probe", pid 157, jiffies 4294694003 hex dump (first 32 bytes): 80 8b cf 41 9f 9d ff ff b8 97 e0 89 ff ff ff ff ...A..... d0 97 e0 89 ff ff ff 19 00 00 00 1f 88 23 00#. backtrace (crc 838fb36): __kmalloc_cache_noprof+0x284/0x320 padata_alloc_pd+0x20/0x1e0 padata_alloc_shell+0x3b/0xa0 0xffffffffc040a54d cryptomgr_probe+0x43/0xc0 kthread+0xf6/0x1f0 ret_from_fork+0x2f/0x50 ret_from_fork_asm+0x1a/0x30

CVE ID	Description
CVE-2025-38033	In the Linux kernel, the following vulnerability has been resolved: x86/Kconfig: make CFI_AUTO_DEFAULT depend on !RUST or Rust >= 1.88 Calling core::fmt::write() from rust code while FineIBT is enabled results in a kernel panic: [4614.199779] kernel BUG at arch/x86/kernel/cet.c:132! [4614.205343] Oops: invalid opcode: 0000 [#1] PREEMPT SMP NOPTI [4614.211781] CPU: 2 UID: 0 PID: 6057 Comm: dmabuf_dump Tainted: G U O 6.12.17-android16-0-g6ab38c534a43 #1 9da040f27673ec3945e23b998a0f8bd64c846599 [4614.227832] Tainted: [U]=USER, [O]=OOT_MODULE [4614.241247] RIP: 0010:do_kernel_cp_fault+0xea/0xf0 ... [4614.398144] RIP: 0010:_RNvXs5_NtNtCs3o2tGsuHyou_4core3fmt3num3impyNtB9_7Display3fmt+0x0/0x20 [4614.407792] Code: 48 f7 df 48 0f 48 f9 48 89 f2 89 c6 5d e9 18 fd ff ff 0f 1f 84 00 00 00 00 00 f3 0f 1e fa 41 81 ea 14 61 af 2c 74 03 0f 0b 90 <66> 0f 1f 00 55 48 89 e5 48 89 f2 48 8b 3f be 01 00 00 00 5d e9 e7 [4614.428775] RSP: 0018:ffffb95acfa4ba68 EFLAGS: 00010246 [4614.434609] RAX: 0000000000000000 RBX: 0000000000000010 RCX: 0000000000000000 [4614.442587] RDX: 0000000000000007 RSI: fffffb95acfa4ba70 RDI: fffffb95acfa4bc88 [4614.450557] RBP: fffffb95acfa4bae0 R08: ffff0a00ffffff05 R09: 0000000000000070 [4614.458527] R10: 0000000000000000 R11: ffffffffab67eaf0 R12: fffffb95acfa4bcc8 [4614.466493] R13: ffffffffac5d50f0 R14: 0000000000000000 R15: 0000000000000000 [4614.474473] ? __cfi__RNvXs5_NtNtCs3o2tGsuHyou_4core3fmt3num3impyNtB9_7Display3fmt+0x10/0x10 [4614.484118] ? _RNvNtCs3o2tGsuHyou_4core3fmt5write+0x1d2/0x250 This happens because core::fmt::write() calls core::rt::Argument::fmt(), which currently has CFI disabled: library/core/src/fmt/rt.rs: 171 // FIXME: Transmuting formatter in new and indirectly branching to/calling 172 // it here is an explicit CFI violation. 173 #[allow(inline_no_sanitization)] 174 #[no_sanitization(cfi, kcfi)] 175 #[inline] 176 pub(super) unsafe fn fmt(&self, f: &mut Formatter<'_>) -> Result { This causes a Control Protection exception, because FineIBT has sealed off the original function's endbr64. This makes rust currently incompatible with FineIBT. Add a Kconfig dependency that prevents FineIBT from getting turned on by default if rust is enabled. [Rust 1.88.0 (scheduled for 2025-06-26) should have this fixed [1], and thus we relaxed the condition with Rust >= 1.88. When `objtool` lands checking for this with e.g. [2], the plan is to ideally run that in upstream Rust's CI to prevent regressions early [3], since we do not control `core`'s source code. Alice tested the Rust PR backported to an older compiler. Peter would like that Rust provides a stable `core` which can be pulled into the kernel: "Relying on that much out of tree code is 'unfortunate'". - Miguel] [Reduced splat. - Miguel]
CVE-2025-38034	In the Linux kernel, the following vulnerability has been resolved: btrfs: correct the order of prelim_ref arguments in btrfs__prelim_ref btrfs_prelim_ref() calls the old and new reference variables in the incorrect order. This causes a NULL pointer dereference because oldref is passed as NULL to trace_btrfs_prelim_ref_insert(). Note, trace_btrfs_prelim_ref_insert() is being called with newref as oldref (and oldref as NULL) on purpose in order to print out the values of newref. To reproduce: echo 1 > /sys/kernel/debug/tracing/events/btrfs/btrfs_prelim_ref_insert/enable Perform some writeback operations. Backtrace: BUG: kernel NULL pointer dereference, address: 0000000000000018 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 115949067 P4D 115949067 PUD 11594a067 PMD 0 Oops: Oops: 0000 [#1] SMP NOPTI CPU: 1 UID: 0 PID: 1188 Comm: fsstress Not tainted 6.15.0-rc2-tester+ #47 PREEMPT(voluntary) 7ca2cef72d5e9c600f0c7718adb6462de8149622 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.3-2-gc13ff2cd-prebuilt.qemu.org 04/01/2014 RIP: 0010:trace_event_raw_event_btrfs__prelim_ref+0x72/0x130 Code: e8 43 81 9f ff 48 85 c0 74 78 4d 85 e4 0f 84 8f 00 00 00 49 8b 94 24 c0 06 00 00 48 8b 0a 48 89 48 08 48 8b 52 08 48 89 50 10 <49> 8b 55 18 48 89 50 18 49 8b 55 20 48 89 50 20 41 0f b6 55 28 88 RSP: 0018:ffffce44820077a0 EFLAGS: 00010286 RAX: ffff8c6b403f9014 RBX: ffff8c6b55825730 RCX: 304994edf9cf506b RDX: d8b11eb7f0fdb699 RSI: ffff8c6b403f9010 RDI: ffff8c6b403f9010 RBP: 0000000000000001 R08: 0000000000000001 R09: 0000000000000010 R10: 00000000ffffff R11: 0000000000000000 R12: ffff8c6b4e8fb000 R13: 0000000000000000 R14: ffffce44820077a8 R15: ffff8c6b4abd1540 FS: 00007f4dc6813740(0000) GS:ffff8c6c1d378000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000018 CR3: 000000010eb42000 CR4: 0000000000750ef0 PKRU: 55555554 Call Trace: <TASK> prelim_ref_insert+0x1c1/0x270 find_parent_nodes+0x12a6/0x1ee0 ? __entry_text_end+0x101f06/0x101f09 ? srso_alias_return_thunk+0x5/0xfbef5 ? srso_alias_return_thunk+0x5/0xfbef5 ? srso_alias_return_thunk+0x5/0xfbef5 ? srso_alias_return_thunk+0x5/0xfbef5 btrfs_is_data_extents_shared+0x167/0x640 ? fiemap_process_hole+0xd0/0x2c0 extent_fiemap+0xa5c/0xabc0 ? __entry_text_end+0x101f05/0x101f09 btrfs_fiemap+0x7e/0xd0 do_vfs_ioctl+0x425/0x9d0 __x64_sys_ioctl+0x75/0xc0

CVE ID	Description
CVE-2025-38035	In the Linux kernel, the following vulnerability has been resolved: nvmet-tcp: don't restore null sk_state_change queue->state_change is set as part of nvmet_tcp_set_queue_sock(), but if the TCP connection isn't established when nvmet_tcp_set_queue_sock() is called then queue->state_change isn't set and sock->sk->sk_state_change isn't replaced. As such we don't need to restore sock->sk->sk_state_change if queue->state_change is NULL. This avoids NULL pointer dereferences such as this: [286.462026][C0] BUG: kernel NULL pointer dereference, address: 0000000000000000 [286.462814][C0] #PF: supervisor instruction fetch in kernel mode [286.463796][C0] #PF: error_code(0x0010) - not-present page [286.464392][C0] PGD 8000000140620067 P4D 8000000140620067 PUD 114201067 PMD 0 [286.465086][C0] Oops: Oops: 0010 [#1] SMP KASAN PTI [286.465559][C0] CPU: 0 UID: 0 PID: 1628 Comm: nvme Not tainted 6.15.0-rc2+ #11 PREEMPT(voluntary) [286.466393][C0] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014 [286.467147][C0] RIP: 0010:0x0 [286.467420][C0] Code: Unable to access opcode bytes at 0xffffffffffffd6, [286.467977][C0] RSP: 0018:ffff8883ae008580 EFLAGS: 00010246 [286.468425][C0] RAX: 0000000000000000 RBX: ffff88813fd34100 RCX: ffffffff386cc43 [286.469019][C0] RDX: 1ffff11027fa68b6 RSI: 0000000000000008 RDI: ffff88813fd34100 [286.469545][C0] RBP: ffff88813fd34160 R08: 0000000000000000 R09: ffffed1027fa682c [286.470072][C0] R10: ffff88813fd34167 R11: 0000000000000000 R12: ffff88813fd344c3 [286.470585][C0] R13: ffff88813fd34112 R14: ffff88813fd34aec R15: ffff888132cdd268 [286.471070][C0] FS: 00007fe3c04c7d80(0000) GS:ffff88840743f000(0000) knlGS:0000000000000000 [286.471644][C0] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [286.472543][C0] CR2: ffffffff8fdd6 CR3: 000000012daca000 CR4: 000000000000006f0 [286.473500][C0] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 [286.474467][C0] DR3: 0000000000000000 DR6: 00000000ffff07f0 DR7: 00000000000000400 [286.475453][C0] Call Trace: [286.476102][C0] <IRQ> [286.476719][C0] tcp_fin+0x2bb/0x440 [286.477429][C0] tcp_data_queue+0x190f/0x4e60 [286.478174][C0] ? __build_skb_around+0x234/0x330 [286.478940][C0] ? rcu_is_watching+0x11/0xb0 [286.479659][C0] ? __pfx_tcp_data_queue+0x10/0x10 [286.480431][C0] ? tcp_try_undo_loss+0x640/0x6c0 [286.481196][C0] ? seqcount_lockdep_reader_access.constprop.0+0x82/0x90 [286.482046][C0] ? kvm_clock_get_cycles+0x14/0x30 [286.482769][C0] ? ktime_get+0x66/0x150 [286.483433][C0] ? rcu_is_watching+0x11/0xb0 [286.484146][C0] tcp_rcv_established+0x6e4/0x2050 [286.484857][C0] ? rcu_is_watching+0x11/0xb0 [286.485523][C0] ? ipv4_dst_check+0x160/0x2b0 [286.486203][C0] ? __pfx_tcp_rcv_established+0x10/0x10 [286.486917][C0] ? lock_release+0x217/0x2c0 [286.487595][C0] tcp_v4_do_rcv+0x4d6/0x9b0 [286.488279][C0] tcp_v4_rcv+0x2af8/0x3e30 [286.488904][C0] ? raw_local_deliver+0x51b/0xad0 [286.489551][C0] ? rcu_is_watching+0x11/0xb0 [286.490198][C0] ? __pfx_tcp_v4_rcv+0x10/0x10 [286.490813][C0] ? __pfx_raw_local_deliver+0x10/0x10 [286.491487][C0] ? __pfx_nf_confirm+0x10/0x10 [nf_conntrack] [286.492275][C0] ? rcu_is_watching+0x11/0xb0 [286.492900][C0] ip_protocol_deliver_rcu+0x8f/0x370 [286.493579][C0] ip_local_deliver_finish+0x297/0x420 [286.494268][C0] ip_local_deliver+0x168/0x430 [286.494867][C0] ? __pfx_ip_local_deliver+0x10/0x10 [286.495498][C0] ? __pfx_ip_local_deliver_finish+0x10/0x10 [286.496204][C0] ? ip_rcv_finish_core+0x19a/0x1f20 [286.496806][C0] ? lock_release+0x217/0x2c0 [286.497414][C0] ip_rcv+0x455/0x6e0 [286.497945][C0] ? __pfx_ip_rcv+0x10/0x10 [---truncated---
CVE-2025-38036	In the Linux kernel, the following vulnerability has been resolved: drm/xe/vf: Perform early GT MMIO initialization to read GMDID VFs need to communicate with the GuC to obtain the GMDID value and existing GuC functions used for that assume that the GT has its MMIO members already setup. However, due to recent refactoring the gt->mmio is initialized later, and any attempt by the VF to use xe_mmio_read/write() from GuC functions will lead to NPD crash due to unset MMIO register address: [] xe 0000:00:02.1: [drm] Running in SR-IOV VF mode [] xe 0000:00:02.1: [drm] GT0: sending H2G MMIO 0x5507 [] BUG: unable to handle page fault for address: 0000000000190240 Since we are already tweaking the id and type of the primary GT to mimic it's a Media GT before initializing the GuC communication, we can also call xe_gt_mmio_init() to perform early setup of the gt->mmio which will make those GuC functions work again.

CVE ID	Description
CVE-2025-38037	In the Linux kernel, the following vulnerability has been resolved: vxlan: Annotate FDB data races The 'used' and 'updated' fields in the FDB entry structure can be accessed concurrently by multiple threads, leading to reports such as [1]. Can be reproduced using [2]. Suppress these reports by annotating these accesses using READ_ONCE() / WRITE_ONCE(). [1] BUG: KCSAN: data-race in vxlan_xmit / vxlan_xmit write to 0xffff942604d263a8 of 8 bytes by task 286 on cpu 0: vxlan_xmit+0xb29/0x2380 dev_hard_start_xmit+0x84/0x2f0 __dev_queue_xmit+0x45a/0x1650 packet_xmit+0x100/0x150 packet_sendmsg+0x2114/0x2ac0 __sys_sendto+0x318/0x330 __x64_sys_sendto+0x76/0x90 x64_sys_call+0x14e8/0x1c00 do_syscall_64+0x9e/0x1a0 entry_SYSCALL_64_after_hwframe+0x77/0x7f read to 0xffff942604d263a8 of 8 bytes by task 287 on cpu 2: vxlan_xmit+0xadf/0x2380 dev_hard_start_xmit+0x84/0x2f0 __dev_queue_xmit+0x45a/0x1650 packet_xmit+0x100/0x150 packet_sendmsg+0x2114/0x2ac0 __sys_sendto+0x318/0x330 __x64_sys_sendto+0x76/0x90 x64_sys_call+0x14e8/0x1c00 do_syscall_64+0x9e/0x1a0 entry_SYSCALL_64_after_hwframe+0x77/0x7f value changed: 0x00000000fffbac6e -> 0x00000000fffbac6f Reported by Kernel Concurrency Sanitizer on: CPU: 2 UID: 0 PID: 287 Comm: mausezahn Not tainted 6.13.0-rc7-01544-gb4b270f11a02 #5 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014 [2] #!/bin/bash set +H echo whitelist > /sys/kernel/debug/kcsan echo !vxlan_xmit > /sys/kernel/debug/kcsan ip link add name vx0 up type vxlan id 10010 dstport 4789 local 192.0.2.1 bridge fdb add 00:11:22:33:44:55 dev vx0 self static dst 198.51.100.1 taskset -c 0 mausezahn vx0 -a own -b 00:11:22:33:44:55 -c 0 -q & taskset -c 2 mausezahn vx0 -a own -b 00:11:22:33:44:55 -c 0 -q &
CVE-2025-38038	In the Linux kernel, the following vulnerability has been resolved: cpufreq: amd-pstate: Remove unnecessary driver_lock in set_boost set_boost is a per-policy function call, hence a driver wide lock is unnecessary. Also this mutex_acquire can collide with the mutex_acquire from the mode-switch path in status_store(), which can lead to a deadlock. So, remove it.
CVE-2025-38039	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Avoid WARN_ON when configuring MQPRIO with HTB offload enabled When attempting to enable MQPRIO while HTB offload is already configured, the driver currently returns '-EINVAL' and triggers a 'WARN_ON', leading to an unnecessary call trace. Update the code to handle this case more gracefully by returning '-EOPNOTSUPP' instead, while also providing a helpful user message.
CVE-2025-38040	In the Linux kernel, the following vulnerability has been resolved: serial: mctrl_gpio: split disable_ms into sync and no_sync APIs The following splat has been observed on a SAMA5D27 platform using atmel_serial: BUG: sleeping function called from invalid context at kernel/irq/manage.c:738 in_atomic(): 1, irqs_disabled(): 128, non_block: 0, pid: 27, name: kworker/u5:0 preempt_count: 1, expected: 0 INFO: lockdep is turned off. irq event stamp: 0 hardirqs last enabled at (0): [<00000000>] 0x0 hardirqs last disabled at (0): [<c01588f0>] copy_process+0x1ca0/0x7bec softirqs last enabled at (0): [<c0158944>] copy_process+0x1ca0/0x7bec softirqs last disabled at (0): [<00000000>] 0x0 CPU: 0 UID: 0 PID: 27 Comm: kworker/u5:0 Not tainted 6.13.0-rc7+ #74 Hardware name: Atmel SAMA5 Workqueue: hci0 hci_power_on [bluetooth] Call trace: unwind_backtrace from show_stack+0x18/0x1c show_stack from dump_stack_lvl+0x44/0x70 dump_stack_lvl from __might_resched+0x38c/0x598 __might_resched from disable_irq+0x1c/0x48 disable_irq from mctrl_gpio_disable_ms+0x74/0xc0 mctrl_gpio_disable_ms from atmel_disable_ms.part.0+0x80/0x1f4 atmel_disable_ms.part.0 from atmel_set_termios+0x764/0x11e8 atmel_set_termios from uart_change_line_settings+0x15c/0x994 uart_change_line_settings from uart_set_termios+0x2b0/0x668 uart_set_termios from tty_set_termios+0x600/0x8ec tty_set_termios from ttyport_set_flow_control+0x188/0x1e0 ttyport_set_flow_control from wilc_setup+0xd0/0x524 [hci_wilc] wilc_setup [hci_wilc] from hci_dev_open_sync+0x330/0x203c [bluetooth] hci_dev_open_sync [bluetooth] from hci_dev_do_open+0x40/0xb0 [bluetooth] hci_dev_do_open [bluetooth] from hci_power_on+0x12c/0x664 [bluetooth] hci_power_on [bluetooth] from process_one_work+0x998/0x1a38 process_one_work from worker_thread+0x6e0/0xfb4 worker_thread from kthread+0x3d4/0x484 kthread from ret_from_fork+0x14/0x28 This warning is emitted when trying to toggle, at the highest level, some flow control (with serdev_device_set_flow_control) in a device driver. At the lowest level, the atmel_serial driver is using serial_mctrl_gpio lib to enable/disable the corresponding IRQs accordingly. The warning emitted by CONFIG_DEBUG_ATOMIC_SLEEP is due to disable_irq (called in mctrl_gpio_disable_ms) being possibly called in some atomic context (some tty drivers perform modem lines configuration in regions protected by port lock). Split mctrl_gpio_disable_ms into two different APIs, a non-blocking one and a blocking one. Replace mctrl_gpio_disable_ms calls with the relevant version depending on whether the call is protected by some port lock.
CVE-2025-38041	In the Linux kernel, the following vulnerability has been resolved: clk: sunxi-ng: h616: Reparent GPU clock during frequency changes The H616 manual does not state that the GPU PLL supports dynamic frequency configuration, so we must take extra care when changing the frequency. Currently any attempt to do device DVFS on the GPU lead to panfrost various oops, and GPU hangs. The manual describes the algorithm for changing the PLL frequency, which the CPU PLL notifier code already support, so we reuse that to reparent the GPU clock to GPU1 clock during frequency changes.

CVE ID	Description
CVE-2025-38042	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: k3-udma-glue: Drop skip_fdq argument from k3_udma_glue_reset_rx_chn The user of k3_udma_glue_reset_rx_chn() e.g. ti_am65_cpsw_nuss can run on multiple platforms having different DMA architectures. On some platforms there can be one FDQ for all flows in the RX channel while for others there is a separate FDQ for each flow in the RX channel. So far we have been relying on the skip_fdq argument of k3_udma_glue_reset_rx_chn(). Instead of relying on the user to provide this information, infer it based on DMA architecture during k3_udma_glue_request_rx_chn() and save it in an internal flag 'single_fdq'. Use that flag at k3_udma_glue_reset_rx_chn() to decide if the FDQ needs to be cleared for every flow or just for flow 0. Fixes the below issue on ti_am65_cpsw_nuss driver on AM62-SK. > ip link set eth1 down > ip link set eth0 down > ethtool -L eth0 rx 8 > ip link set eth0 up > modprobe -r ti_am65_cpsw_nuss [103.045726] -----[cut here]----- [103.050505] k3_knav_desc_pool size 512000 != avail 64000 [103.050703] WARNING: CPU: 1 PID: 450 at drivers/net/ethernet/ti/k3-ccpi-desc-pool.c:33 k3_ccpi_desc_pool_destroy+0xa0/0xa8 [k3_ccpi_desc_pool] [103.068810] Modules linked in: ti_am65_cpsw_nuss(-) k3_ccpi_desc_pool snd_soc_hdmi_codec crct10dif_ce snd_soc_simple_card snd_soc_simple_card_utils display_connector rtc_ti_k3 k3_j72xx_bandgap tidss drm_client_lib snd_soc_davinci_mcasp drm_dma_helper tps6598x phylink snd_soc_ti_udma rti_wdt drm_display_helper snd_soc_tlv320aic3x_i2c typec_at24 phy_gmii_sel snd_soc_ti_edma snd_soc_tlv320aic3x sii902x snd_soc_ti_sdma sa2ul omap_mailbox drm_kms_helper authenc cfg80211 rfkill fuse drm drm_panel_orientation_quirks backlight ip_tables x_tables ipv6 [last unloaded: k3_ccpi_desc_pool] [103.119950] CPU: 1 UID: 0 PID: 450 Comm: modprobe Not tainted 6.13.0-rc7-00001-g9c5e3435fa66 #1011 [103.119968] Hardware name: Texas Instruments AM625 SK (DT) [103.119974] pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) [103.119983] pc : k3_ccpi_desc_pool_destroy+0xa0/0xa8 [k3_ccpi_desc_pool] [103.148007] lr : k3_ccpi_desc_pool_destroy+0xa0/0xa8 [k3_ccpi_desc_pool] [103.154709] sp : ffff8000826ebbc0 [103.158015] x29: ffff8000826ebbc0 x28: ffff000090b6300 x27: 0000000000000000 [103.165145] x26: 0000000000000000 x25: 0000000000000000 x24: ffff0000019df6b0 [103.172271] x23: ffff0000019df6b8 x22: ffff0000019df410 x21: ffff8000826ebc88 [103.179397] x20: 0000000000007d000 x19: ffff00000a3b3000 x18: 0000000000000000 [103.186522] x17: 0000000000000000 x16: 0000000000000000 x15: 000001e8c35e1cde [103.193647] x14: 00000000000000396 x13: 0000000000000035c x12: 0000000000000000 [103.200772] x11: 0000000000000003a x10: 000000000000009c0 x9: ffff8000826eba20 [103.207897] x8: ffff0000090b6d20 x7: ffff00007728c180 x6: ffff00007728c100 [103.215022] x5: 00000000000000001 x4: ffff000000508a50 x3: ffff7ffff6146000 [103.222147] x2: 00000000000000000 x1: e300b4173ee6b200 x0: 0000000000000000 [103.229274] Call trace: [103.231714] k3_ccpi_desc_pool_destroy+0xa0/0xa8 [k3_ccpi_desc_pool] (P) [103.238408] am65_cpsw_nuss_free_rx_chns+0x28/0x4c [ti_am65_cpsw_nuss] [103.244942] devm_action_release+0x14/0x20 [103.249040] release_nodes+0x3c/0x68 [103.252610] devres_release_all+0x8c/0xdc [103.256614] device_unbind_cleanup+0x18/0x60 [103.260876] device_release_driver_internal+0xf8/0x178 [103.266004] driver_detach+0x50/0x9c [103.269571] bus_remove_driver+0x6c/0xbc [103.273485] driver_unregister+0x30/0x60 [103.277401] platform_driver_unregister+0x14/0x20 [103.282096] am65_cpsw_nuss_driver_exit+0x18/0xff4 [ti_am65_cpsw_nuss] [103.288620] __arm64_sys_delete_module+0x17c/0x25c [103.293404] invoke_syscall+0x44/0x100 [103.297149] el0_svc_common.constprop.0+0xc0/0xe0 [103.301845] do_el0_svc+0x1c/0x28 [103.305155] el0_svc+0x28/0x98 ---truncated---
CVE-2025-38043	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_ffa: Set dma_mask for ffa devices Set dma_mask for FFA devices, otherwise DMA allocation using the device pointer lead to following warning: WARNING: CPU: 1 PID: 1 at kernel/dma/mapping.c:597 dma_alloc_attrs+0xe0/0x124
CVE-2025-38044	In the Linux kernel, the following vulnerability has been resolved: media: cx231xx: set device_caps for 417 The video_device for the MPEG encoder did not set device_caps. Add this, otherwise the video device can't be registered (you get a WARN_ON instead). Not seen before since currently 417 support is disabled, but I found this while experimenting with it.
CVE-2025-38045	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: fix debug actions order The order of actions taken for debug was implemented incorrectly. Now we implemented the dump split and do the FW reset only in the middle of the dump (rather than the FW killing itself on error.) As a result, some of the actions taken when applying the config will now crash the device, so we need to fix the order.

CVE ID	Description
CVE-2025-38047	In the Linux kernel, the following vulnerability has been resolved: x86/fred: Fix system hang during S4 resume with FRED enabled Upon a wakeup from S4, the restore kernel starts and initializes the FRED MSRs as needed from its perspective. It then loads a hibernation image, including the image kernel, and attempts to load image pages directly into their original page frames used before hibernation unless those frames are currently in use. Once all pages are moved to their original locations, it jumps to a "trampoline" page in the image kernel. At this point, the image kernel takes control, but the FRED MSRs still contain values set by the restore kernel, which may differ from those set by the image kernel before hibernation. Therefore, the image kernel must ensure the FRED MSRs have the same values as before hibernation. Since these values depend only on the location of the kernel text and data, they can be recomputed from scratch.
CVE-2025-38048	In the Linux kernel, the following vulnerability has been resolved: virtio_ring: Fix data race by tagging event_triggered as racy for KCSAN syzbot reports a data-race when accessing the event_triggered, here is the simplified stack when the issue occurred: <pre>===== BUG: KCSAN: data-race in virtqueue_disable_cb / virtqueue_enable_cb_delayed write to 0xffff8881025bc452 of 1 bytes by task 3288 on cpu 0: virtqueue_enable_cb_delayed+0x42/0x3c0 drivers/virtio/virtio_ring.c:2653 start_xmit+0x230/0x1310 drivers/net/virtio_net.c:3264 __netdev_start_xmit include/linux/netdevice.h:5151 [inline] netdev_start_xmit include/linux/netdevice.h:5160 [inline] xmit_one net/core/dev.c:3800 [inline] read to 0xffff8881025bc452 of 1 bytes by interrupt on cpu 1: virtqueue_disable_cb_split drivers/virtio/virtio_ring.c:880 [inline] virtqueue_disable_cb+0x92/0x180 drivers/virtio/virtio_ring.c:2566 skb_xmit_done+0x5f/0x140 drivers/net/virtio_net.c:777 vring_interrupt+0x161/0x190 drivers/virtio/virtio_ring.c:2715 __handle_irq_event_percpu+0x95/0x490 kernel/irq/handle.c:158 handle_irq_event_percpu kernel/irq/handle.c:193 [inline] value changed: 0x01 -> 0x00 =====</pre> When the data race occurs, the function virtqueue_enable_cb_delayed() sets event_triggered to false, and virtqueue_disable_cb_split/packed() reads it as false due to the race condition. Since event_triggered is an unreliable hint used for optimization, this should only cause the driver temporarily suggest that the device not send an interrupt notification when the event index is used. Fix this KCSAN reported data-race issue by explicitly tagging the access as data_racy.

CVE ID	Description
CVE-2025-38051	In the Linux kernel, the following vulnerability has been resolved: smb: client: Fix use-after-free in cifs_fill_dirent There is a race condition in the readdir concurrency process, which may access the rsp buffer after it has been released, triggering the following KASAN warning. <pre> ===== BUG: KASAN: slab-use-after-free in cifs_fill_dirent+0xb03/0xb60 [cifs] Read of size 4 at addr ffff8880099b819c by task a.out/342975 CPU: 2 UID: 0 PID: 342975 Comm: a.out Not tainted 6.15.0-rc6+ #240 PREEMPT(full) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 Call Trace: <TASK> dump_stack_lvl+0x53/0x70 print_report+0xce/0x640 kasan_report+0xb8/0xf0 cifs_fill_dirent+0xb03/0xb60 [cifs] cifs_readdir+0x12cb/0x3190 [cifs] iterate_dir+0x1a1/0x520 __x64_sys_getdents+0x134/0x220 do_syscall_64+0x4b/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f996f64b9f9 Code: ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 0d f7 c3 0c 00 f7 d8 64 89 8 RSP: 002b:00007f996f53de78 EFLAGS: 00000207 ORIG_RAX: 000000000000004e RAX: ffffffffffffffda RBX: 00007f996f53ecd RCX: 00007f996f64b9f9 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000003 RBP: 00007f996f53dea0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000207 R12: ffffffffffff88 R13: 0000000000000000 R14: 00007ffc8cd9a500 R15: 00007f996f51e000 </TASK> Allocated by task 408: kasan_save_stack+0x20/0x40 kasan_save_track+0x14/0x30 __kasan_slab_alloc+0x6e/0x70 kmem_cache_alloc_noprof+0x117/0x3d0 mempool_alloc_noprof+0xf2/0x2c0 cifs_buf_get+0x36/0x80 [cifs] allocate_buffers+0x1d2/0x330 [cifs] cifs_demultiplex_thread+0x22b/0x2690 [cifs] kthread +0x394/0x720 ret_from_fork+0x34/0x70 ret_from_fork_asm+0x1a/0x30 Freed by task 342979: kasan_save_stack+0x20/0x40 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3b/0x60 __kasan_slab_free+0x37/0x50 kmem_cache_free+0x2b8/0x500 cifs_buf_release+0x3c/0x70 [cifs] cifs_readdir+0x1c97/0x3190 [cifs] iterate_dir+0x1a1/0x520 __x64_sys_getdents64+0x134/0x220 do_syscall_64+0x4b/0x110 entry_SYSCALL_64_after_hwframe+0x76/0x7e The buggy address belongs to the object at ffff8880099b8000 which belongs to the cache cifs_request of size 16588 The buggy address is located 412 bytes inside of freed 16588-byte region [ffff8880099b8000, ffff8880099bc0cc) The buggy address belongs to the physical page: page: refcount:0 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x99b8 head: order:3 mapcount:0 entire_mapcount:0 nr_pages_mapped:0 pincount:0 anon flags: 0x8000000000000040(head node=0 zone=1) page_type: f5(slab) raw: 0080000000000000 ffff888001e03400 0000000000000000 dead000000000001 raw: 0000000000000000 0000000000010001 00000000f5000000 0000000000000000 head: 0080000000000000 ffff888001e03400 0000000000000000 dead000000000001 head: 0000000000000000 0000000000010001 00000000f5000000 0000000000000000 head: 00800000000000003 ffffea0000266e01 00000000ffffffff 00000000ffffffff head: ffffffff 0000000000000000 00000000ffffffff 0000000000000008 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff8880099b8080: fb fb fb fb fb fb fb fb fb fb fb fb ffff8880099b8100: fb fb fb fb fb fb fb fb fb fb fb fb >ffff8880099b8180: fb fb fb fb fb fb fb fb fb fb fb fb ^ ffff8880099b8200: fb fb fb fb fb fb fb fb fb fb fb ffff8880099b8280: fb fb fb fb fb fb fb fb fb fb fb fb fb ===== POC is available in the link [1]. The problem triggering process is as follows: Process 1 Process 2 -----truncated----- </pre>

CVE ID	Description
CVE-2025-38052	In the Linux kernel, the following vulnerability has been resolved: net/tipc: fix slab-use-after-free Read in tipc_aead_encrypt_done Syzbot reported a slab-use-after-free with the following call trace: ===== BUG: KASAN: slab-use-after-free in tipc_aead_encrypt_done+0x4bd/0x510 net/tipc/crypto.c:840 Read of size 8 at addr ffff88807a733000 by task kworker/1:0/25 Call Trace: kasan_report +0xd9/0x110 mm/kasan/report.c:601 tipc_aead_encrypt_done+0x4bd/0x510 net/tipc/crypto.c:840 crypto_request_complete include/crypto/algapi.h:266 aead_request_complete include/crypto/internal/aead.h:85 cryptd_aead_crypt+0x3b8/0x750 crypto/cryptd.c:772 crypto_request_complete include/crypto/algapi.h:266 cryptd_queue_worker+0x131/0x200 crypto/cryptd.c:181 process_one_work +0x9fb/0x1b60 kernel/workqueue.c:3231 Allocated by task 8355: kzalloc_noprof include/linux/slab.h:778 tipc_crypto_start+0xcc/0x9e0 net/tipc/crypto.c:1466 tipc_init_net+0x2dd/0x430 net/tipc/core.c:72 ops_init+0xb9/0x650 net/core/net_namespace.c:139 setup_net+0x435/0xb40 net/core/net_namespace.c:343 copy_net_ns+0x2f0/0x670 net/core/net_namespace.c:508 create_new_namespaces +0x3ea/0xb10 kernel/nsproxy.c:110 unshare_nsproxy_namespaces+0xc0/0x1f0 kernel/nsproxy.c:228 ksys_unshare+0x419/0x970 kernel/fork.c:3323 __do_sys_unshare kernel/fork.c:3394 Freed by task 63: kfree+0x12a/0x3b0 mm/slub.c:4557 tipc_crypto_stop+0x23c/0x500 net/tipc/crypto.c:1539 tipc_exit_net+0x8c/0x110 net/tipc/core.c:119 ops_exit_list+0xb0/0x180 net/core/net_namespace.c:173 cleanup_net+0x5b7/0xbf0 net/core/net_namespace.c:640 process_one_work+0x9fb/0x1b60 kernel/workqueue.c:3231 After freed the tipc_crypto tx by delete namespace, tipc_aead_encrypt_done may still visit it in cryptd_queue_worker workqueue. I reproduce this issue by: ip netns add ns1 ip link add veth1 type veth peer name veth2 ip link set veth1 netns ns1 ip netns exec ns1 tipc bearer enable media eth dev veth1 ip netns exec ns1 tipc node set key this_is_a_master_key master ip netns exec ns1 tipc bearer disable media eth dev veth1 ip netns del ns1 The key of reproduction is that, simd_aead_encrypt is interrupted, leading to crypto_simd_usable() return false. Thus, the cryptd_queue_worker is triggered, and the tipc_crypto tx will be visited. tipc_disc_timeout tipc_bearer_xmit_skb tipc_crypto_xmit tipc_aead_encrypt crypto_aead_encrypt // encrypt() simd_aead_encrypt // crypto_simd_usable() is false child = &ctx->cryptd_tfm->base; simd_aead_encrypt crypto_aead_encrypt // encrypt() cryptd_aead_encrypt_enqueue cryptd_aead_enqueue cryptd_enqueue_request // trigger cryptd_queue_worker queue_work_on(smp_processor_id(), cryptd_wq, &cpu_queue->work) Fix this by holding net reference count before encrypt.
CVE-2025-38053	In the Linux kernel, the following vulnerability has been resolved: idpf: fix null-ptr-deref in idpf_features_check idpf_features_check is used to validate the TX packet. skb header length is compared with the hardware supported value received from the device control plane. The value is stored in the adapter structure and to access it, vport pointer is used. During reset all the vports are released and the vport pointer that the netdev private structure points to is NULL. To avoid null-ptr-deref, store the max header length value in netdev private structure. This also helps to cache the value and avoid accessing adapter pointer in hot path. BUG: kernel NULL pointer dereference, address: 0000000000000068 ... RIP: 0010:idpf_features_check+0x6d/0xe0 [idpf] Call Trace: <TASK> ? __die +0x23/0x70 ? page_fault_oops+0x154/0x520 ? exc_page_fault+0x76/0x190 ? asm_exc_page_fault +0x26/0x30 ? idpf_features_check+0x6d/0xe0 [idpf] netif_skb_features+0x88/0x310 validate_xmit_skb +0x2a/0x2b0 validate_xmit_skb_list+0x4c/0x70 sch_direct_xmit+0x19d/0x3a0 __dev_queue_xmit +0xb74/0xe70 ...
CVE-2025-38055	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: Fix segfault with PEBS-via-PT with sample_freq Currently, using PEBS-via-PT with a sample frequency instead of a sample period, causes a segfault. For example: BUG: kernel NULL pointer dereference, address: 0000000000000195 <NMI> ? __die_body.cold+0x19/0x27 ? page_fault_oops+0xca/0x290 ? exc_page_fault+0x7e/0x1b0 ? asm_exc_page_fault+0x26/0x30 ? intel_pmu_pebs_event_update_no_drain+0x40/0x60 ? intel_pmu_pebs_event_update_no_drain +0x32/0x60 intel_pmu_drain_pebs_icl+0x333/0x350 handle_pmi_common+0x272/0x3c0 intel_pmu_handle_irq+0x10a/0x2e0 perf_event_nmi_handler+0x2a/0x50 That happens because intel_pmu_pebs_event_update_no_drain() assumes all the pebs_enabled bits represent counter indexes, which is not always the case. In this particular case, bits 60 and 61 are set for PEBS-via-PT purposes. The behaviour of PEBS-via-PT with sample frequency is questionable because although a PMI is generated (PEBS_PMI_AFTER_EACH_RECORD), the period is not adjusted anyway. Putting that aside, fix intel_pmu_pebs_event_update_no_drain() by passing the mask of counter bits instead of 'size'. Note, prior to the Fixes commit, 'size' would be limited to the maximum counter index, so the issue was not hit.
CVE-2025-38057	In the Linux kernel, the following vulnerability has been resolved: espintcp: fix skb leaks A few error paths are missing a kfree_skb.

CVE ID	Description
CVE-2025-38058	In the Linux kernel, the following vulnerability has been resolved: <code>__legitimize_mnt()</code> : check for <code>MNT_SYNC_UMOUNT</code> should be under <code>mount_lock</code> ... or we risk stealing final <code>mntput</code> from <code>sync_umount</code> - raising <code>mnt_count</code> after <code>umount(2)</code> has verified that victim is not busy, but before it has set <code>MNT_SYNC_UMOUNT</code> ; in that case <code>__legitimize_mnt()</code> doesn't see that it's safe to quietly undo <code>mnt_count</code> increment and leaves dropping the reference to caller, where it'll be a full-blown <code>mntput()</code> . Check under <code>mount_lock</code> is needed; leaving the current one done before taking that makes no sense - it's nowhere near common enough to bother with.
CVE-2025-38059	In the Linux kernel, the following vulnerability has been resolved: <code>btrfs</code> : avoid NULL pointer dereference if no valid csum tree [BUG] When trying read-only scrub on a <code>btrfs</code> with <code>rescue=idatacsums</code> mount option, it will crash with the following call trace: BUG: kernel NULL pointer dereference, address: 0000000000000208 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page CPU: 1 UID: 0 PID: 835 Comm: <code>btrfs</code> Tainted: G O 6.15.0-rc3-custom+ #236 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS unknown 02/02/2022 RIP: 0010:btrfs_lookup_csums_bitmap+0x49/0x480 [btrfs] Call Trace: <TASK> scrub_find_fill_first_stripe+0x35b/0x3d0 [btrfs] scrub_simple_mirror+0x175/0x290 [btrfs] scrub_stripe+0x5f7/0x6f0 [btrfs] scrub_chunk+0x9a/0x150 [btrfs] scrub_enumerate_chunks+0x333/0x660 [btrfs] btrfs_scrub_dev+0x23e/0x600 [btrfs] btrfs_ioctl+0x1dcf/0x2f80 [btrfs] __x64_sys_ioctl+0x97/0xc0 do_syscall_64+0x4f/0x120 entry_SYSCALL_64_after_hwframe+0x76/0x7e [CAUSE] Mount option "rescue=idatacsums" will completely skip loading the csum tree, so that any data read will not find any data csum thus we will ignore data checksum verification. Normally call sites utilizing csum tree will check the fs state flag <code>NO_DATA_CSUMS</code> bit, but unfortunately scrub does not check that bit at all. This results in scrub to call <code>btrfs_search_slot()</code> on a NULL pointer and triggered above crash. [FIX] Check both extent and csum tree root before doing any tree search.
CVE-2025-38060	In the Linux kernel, the following vulnerability has been resolved: <code>bpf</code> : <code>copy_verifier_state()</code> should copy 'loop_entry' field The <code>bpf_verifier_state.loop_entry</code> state should be copied by <code>copy_verifier_state()</code> . Otherwise, <code>.loop_entry</code> values from unrelated states would poison <code>env->cur_state</code> . Additionally, <code>env->stack</code> should not contain any states with <code>.loop_entry != NULL</code> . The states in <code>env->stack</code> are yet to be verified, while <code>.loop_entry</code> is set for states that reached an equivalent state. This means that <code>env->cur_state->loop_entry</code> should always be NULL after <code>pop_stack()</code> . See the selftest in the next commit for an example of the program that is not safe yet is accepted by verifier w/o this fix. This change has some verification performance impact for selftests: File Program Insns (A) Insns (B) Insns (DIFF) States (A) States (B) States (DIFF) ----- arena_htab.bpf.o arena_htab_llvm 717 426 -291 (-40.59%) 57 37 -20 (-35.09%) arena_htab_asm.bpf.o arena_htab_asm 597 445 -152 (-25.46%) 47 37 -10 (-21.28%) arena_list.bpf.o arena_list_del 309 279 -30 (-9.71%) 23 14 -9 (-39.13%) iters.bpf.o iter_subprog_check_stacksafe 155 141 -14 (-9.03%) 15 14 -1 (-6.67%) iters.bpf.o iter_subprog_iters 1094 1003 -91 (-8.32%) 88 83 -5 (-5.68%) iters.bpf.o loop_state_deps2 479 725 +246 (+51.36%) 46 63 +17 (+36.96%) kmem_cache_iter.bpf.o open_coded_iter 63 59 -4 (-6.35%) 7 6 -1 (-14.29%) verifier_bits_iter.bpf.o max_words 92 84 -8 (-8.70%) 8 7 -1 (-12.50%) verifier_iterating_callbacks.bpf.o cond_break2 113 107 -6 (-5.31%) 12 12 +0 (+0.00%) And significant negative impact for sched_ext: File Program Insns (A) Insns (B) Insns (DIFF) States (A) States (B) States (DIFF) ----- bpf.bpf.o lavd_init 7039 14723 +7684 (+109.16%) 490 1139 +649 (+132.45%) bpf.bpf.o layered_dispatch 11485 10548 -937 (-8.16%) 848 762 -86 (-10.14%) bpf.bpf.o layered_dump 7422 1000001 +992579 (+13373.47%) 681 31178 +30497 (+4478.27%) bpf.bpf.o layered_enqueue 16854 71127 +54273 (+322.02%) 1611 6450 +4839 (+300.37%) bpf.bpf.o p2dq_dispatch 665 791 +126 (+18.95%) 68 78 +10 (+14.71%) bpf.bpf.o p2dq_init 2343 2980 +637 (+27.19%) 201 237 +36 (+17.91%) bpf.bpf.o refresh_layer_cpumasks 16487 674760 +658273 (+3992.68%) 1770 65370 +63600 (+3593.22%) bpf.bpf.o rusty_select_cpu 1937 40872 +38935 (+2010.07%) 177 3210 +3033 (+1713.56%) scx_central.bpf.o central_dispatch 636 2687 +2051 (+322.48%) 63 227 +164 (+260.32%) scx_nest.bpf.o nest_init 636 815 +179 (+28.14%) 60 73 +13 (+21.67%) scx_qmap.bpf.o qmap_dispatch ---truncated---
CVE-2025-38061	In the Linux kernel, the following vulnerability has been resolved: <code>net</code> : <code>pktgen</code> : fix access outside of user given buffer in <code>pktgen_thread_write()</code> Honour the user given buffer size for the <code>strn_len()</code> calls (otherwise <code>strn_len()</code> will access memory outside of the user given buffer).

CVE ID	Description
CVE-2025-38062	In the Linux kernel, the following vulnerability has been resolved: genirq/msi: Store the IOMMU IOVA directly in msi_desc instead of iommu_cookie The IOMMU translation for MSI message addresses has been a 2-step process, separated in time: 1) iommu_dma_prepare_msi(): A cookie pointer containing the IOVA address is stored in the MSI descriptor when an MSI interrupt is allocated. 2) iommu_dma_compose_msi_msg(): this cookie pointer is used to compute a translated message address. This has an inherent lifetime problem for the pointer stored in the cookie that must remain valid between the two steps. However, there is no locking at the irq layer that helps protect the lifetime. Today, this works under the assumption that the iommu domain is not changed while MSI interrupts being programmed. This is true for normal DMA API users within the kernel, as the iommu domain is attached before the driver is probed and cannot be changed while a driver is attached. Classic VFIO type1 also prevented changing the iommu domain while VFIO was running as it does not support changing the "container" after starting up. However, iommufd has improved this so that the iommu domain can be changed during VFIO operation. This potentially allows userspace to directly race VFIO_DEVICE_ATTACH_IOMMUFD_PT (which calls iommu_attach_group()) and VFIO_DEVICE_SET_IRQS (which calls into iommu_dma_compose_msi_msg()). This potentially causes both the cookie pointer and the unlocked call to iommu_get_domain_for_dev() on the MSI translation path to become UAFs. Fix the MSI cookie UAF by removing the cookie pointer. The translated IOVA address is already known during iommu_dma_prepare_msi() and cannot change. Thus, it can simply be stored as an integer in the MSI descriptor. The other UAF related to iommu_get_domain_for_dev() will be addressed in patch "iommu: Make iommu_dma_prepare_msi() into a generic operation" by using the IOMMU group mutex.
CVE-2025-38063	In the Linux kernel, the following vulnerability has been resolved: dm: fix unconditional IO throttle caused by REQ_PREFLUSH When a bio with REQ_PREFLUSH is submitted to dm, __send_empty_flush() generates a flush_bio with REQ_OP_WRITE REQ_PREFLUSH REQ_SYNC, which causes the flush_bio to be throttled by wbt_wait(). An example from v5.4, similar problem also exists in upstream: crash> bt 2091206 PID: 2091206 TASK: ffff2050df92a300 CPU: 109 COMMAND: "kworker/u260:0" #0 [ffff800084a2f7f0] __switch_to at ffff80004008aeb8 #1 [ffff800084a2f820] __schedule at ffff800040bfa0c4 #2 [ffff800084a2f880] schedule at ffff800040bfa4b4 #3 [ffff800084a2f8a0] io_schedule at ffff800040bfa9c4 #4 [ffff800084a2f8c0] rq_qos_wait at ffff8000405925bc #5 [ffff800084a2f940] wbt_wait at ffff8000405bb3a0 #6 [ffff800084a2f9a0] __rq_qos_throttle at ffff800040592254 #7 [ffff800084a2f9c0] blk_mq_make_request at ffff80004057cf38 #8 [ffff800084a2fa60] generic_make_request at ffff800040570138 #9 [ffff800084a2fae0] submit_bio at ffff8000405703b4 #10 [ffff800084a2fb50] xlog_write_iclog at ffff800001280834 [xfs] #11 [ffff800084a2fbb0] xlog_sync at ffff800001280c3c [xfs] #12 [ffff800084a2fbf0] xlog_state_release_iclog at ffff800001280df4 [xfs] #13 [ffff800084a2fc10] xlog_write at ffff80000128203c [xfs] #14 [ffff800084a2fcd0] xlog_cil_push at ffff8000012846dc [xfs] #15 [ffff800084a2fda0] xlog_cil_push_work at ffff800001284a2c [xfs] #16 [ffff800084a2fdb0] process_one_work at ffff800040111d08 #17 [ffff800084a2fe00] worker_thread at ffff8000401121cc #18 [ffff800084a2fe70] kthread at ffff800040118de4 After commit 2def2845cc33 ("xfs: don't allow log IO to be throttled"), the metadata submitted by xlog_write_iclog() should not be throttled. But due to the existence of the dm layer, throttling flush_bio indirectly causes the metadata bio to be throttled. Fix this by conditionally adding REQ_IDLE to flush_bio.bi_opf, which makes wbt_should_throttle() return false to avoid wbt_wait().
CVE-2025-38064	In the Linux kernel, the following vulnerability has been resolved: virtio: break and reset virtio devices on device_shutdown() Hongyu reported a hang on kexec in a VM. QEMU reported invalid memory accesses during the hang. Invalid read at addr 0x102877002, size 2, region '(null)', reason: rejected Invalid write at addr 0x102877A44, size 2, region '(null)', reason: rejected ... It was traced down to virtio-console. Kexec works fine if virtio-console is not in use. The issue is that virtio-console continues to write to the MMIO even after underlying virtio-pci device is reset. Additionally, Eric noticed that IOMMUs are reset before devices, if devices are not reset on shutdown they continue to poke at guest memory and get errors from the IOMMU. Some devices get wedged then. The problem can be solved by breaking all virtio devices on virtio bus shutdown, then resetting them.
CVE-2025-38065	In the Linux kernel, the following vulnerability has been resolved: orangesfs: Do not truncate file size 'len' is used to store the result of i_size_read(), so making 'len' a size_t results in truncation to 4GiB on 32-bit systems.

CVE ID	Description
CVE-2025-38066	In the Linux kernel, the following vulnerability has been resolved: dm cache: prevent BUG_ON by blocking retries on failed device resumes A cache device failing to resume due to mapping errors should not be retried, as the failure leaves a partially initialized policy object. Repeating the resume operation risks triggering BUG_ON when reloading cache mappings into the incomplete policy object. Reproduce steps: 1. create a cache metadata consisting of 512 or more cache blocks, with some mappings stored in the first array block of the mapping array. Here we use cache_restore v1.0 to build the metadata. cat <<EOF >> cmeta.xml <superblock uuid="" block_size="128" nr_cache_blocks="512" \ policy="smq" hint_width="4"> <mappings> <mapping cache_block="0" origin_block="0" dirty="false"/> </mappings> </superblock> EOF dmsetup create cmeta --table "0 8192 linear /dev/sdc 0" cache_restore -i cmeta.xml -o /dev/mapper/cmeta --metadata-version=2 dmsetup remove cmeta 2. wipe the second array block of the mapping array to simulate data degradations. mapping_root=\$(dd if=/dev/sdc bs=1c count=8 skip=192 \ 2>/dev/null hexdump -e '1/8 "%u\n"') ablock=\$(dd if=/dev/sdc bs=1c count=8 skip=\$((4096*mapping_root+2056)) \ 2>/dev/null hexdump -e '1/8 "%u\n"') dd if=/dev/zero of=/dev/sdc bs=4k count=1 seek=\$ablock 3. try bringing up the cache device. The resume is expected to fail due to the broken array block. dmsetup create cmeta --table "0 8192 linear /dev/sdc 0" dmsetup create cdata --table "0 65536 linear /dev/sdc 8192" dmsetup create corig --table "0 524288 linear /dev/sdc 262144" dmsetup create cache --notable dmsetup load cache --table "0 524288 cache /dev/mapper/cmeta \ /dev/mapper/cdata /dev/mapper/corig 128 2 metadata2 writethrough smq 0" dmsetup resume cache 4. try resuming the cache again. An unexpected BUG_ON is triggered while loading cache mappings. dmsetup resume cache Kernel logs: (snip) -----[cut here]----- kernel BUG at drivers/md/dm-cache-policy-smq.c:752! Oops: invalid opcode: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 0 UID: 0 PID: 332 Comm: dmsetup Not tainted 6.13.4 #3 RIP: 0010:smq_load_mapping+0x3e5/0x570 Fix by disallowing resume operations for devices that failed the initial attempt.
CVE-2025-38067	In the Linux kernel, the following vulnerability has been resolved: rseq: Fix segfault on registration when rseq_cs is non-zero The rseq_cs field is documented as being set to 0 by user-space prior to registration, however this is not currently enforced by the kernel. This can result in a segfault on return to user-space if the value stored in the rseq_cs field doesn't point to a valid struct rseq_cs. The correct solution to this would be to fail the rseq registration when the rseq_cs field is non-zero. However, some older versions of glibc will reuse the rseq area of previous threads without clearing the rseq_cs field and will also terminate the process if the rseq registration fails in a secondary thread. This wasn't caught in testing because in this case the leftover rseq_cs does point to a valid struct rseq_cs. What we can do is clear the rseq_cs field on registration when it's non-zero which will prevent segfaults on registration and won't break the glibc versions that reuse rseq areas on thread creation.
CVE-2025-38068	In the Linux kernel, the following vulnerability has been resolved: crypto: lzo - Fix compression buffer overrun Unlike the decompression code, the compression code in LZO never checked for output overruns. It instead assumes that the caller always provides enough buffer space, disregarding the buffer length provided by the caller. Add a safe compression interface that checks for the end of buffer before each write. Use the safe interface in crypto/lzo.
CVE-2025-38069	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: pci-epf-test: Fix double free that causes kernel to oops Fix a kernel oops found while testing the stm32_pcie Endpoint driver with handling of PERST# deassertion: During EP initialization, pci_epf_test_alloc_space() allocates all BARs, which are further freed if epc_set_bar() fails (for instance, due to no free inbound window). However, when pci_epc_set_bar() fails, the error path: pci_epc_set_bar() -> pci_epf_free_space() does not clear the previous assignment to epf_test->reg[bar]. Then, if the host reboots, the PERST# deassertion restarts the BAR allocation sequence with the same allocation failure (no free inbound window), creating a double free situation since epf_test->reg[bar] was deallocated and is still non-NULL. Thus, make sure that pci_epf_alloc_space() and pci_epf_free_space() invocations are symmetric, and as such, set epf_test->reg[bar] to NULL when memory is freed. [kwilczynski: commit log]
CVE-2025-38070	In the Linux kernel, the following vulnerability has been resolved: ASoC: sma1307: Add NULL check in sma1307_setting_loaded() All varibale allocated by kzalloc and devm_kzalloc could be NULL. Multiple pointer checks and their cleanup are added. This issue is found by our static analysis tool
CVE-2025-38071	In the Linux kernel, the following vulnerability has been resolved: x86/mm: Check return value from memblock_phys_alloc_range() At least with CONFIG_PHYSICAL_START=0x100000, if there is < 4 MiB of contiguous free memory available at this point, the kernel will crash and burn because memblock_phys_alloc_range() returns 0 on failure, which leads memblock_phys_free() to throw the first 4 MiB of physical memory to the wolves. At a minimum it should fail gracefully with a meaningful diagnostic, but in fact everything seems to work fine without the weird reserve allocation.

CVE ID	Description
CVE-2025-38072	In the Linux kernel, the following vulnerability has been resolved: libnvdimm/labels: Fix divide error in nd_label_data_init() If a faulty CXL memory device returns a broken zero LSA size in its memory device information (Identify Memory Device (Opcode 4000h), CXL spec. 3.1, 8.2.9.9.1.1), a divide error occurs in the libnvdimm driver: Oops: divide error: 0000 [#1] PREEMPT SMP NOPTI RIP: 0010:nd_label_data_init+0x10e/0x800 [libnvdimm] Code and flow: 1) CXL Command 4000h returns LSA size = 0 2) config_size is assigned to zero LSA size (CXL pmem driver): drivers/cxl/pmem.c: .config_size = mds->lsa_size, 3) max_xfer is set to zero (nvdimm driver): drivers/nvdimm/label.c: max_xfer = min_t(size_t, ndd->nsarea.max_xfer, config_size); 4) A subsequent DIV_ROUND_UP() causes a division by zero: drivers/nvdimm/label.c: /* Make our initial read size a multiple of max_xfer size */ drivers/nvdimm/label.c: read_size = min(DIV_ROUND_UP(read_size, max_xfer) * max_xfer, drivers/nvdimm/label.c- config_size); Fix this by checking the config size parameter by extending an existing check.
CVE-2025-38073	In the Linux kernel, the following vulnerability has been resolved: block: fix race between set_blocksize and read paths With the new large sector size support, it's now the case that set_blocksize can change i_blksize and the folio order in a manner that conflicts with a concurrent reader and causes a kernel crash. Specifically, let's say that udev-worker calls libblkid to detect the labels on a block device. The read call can create an order-0 folio to read the first 4096 bytes from the disk. But then udev is preempted. Next, someone tries to mount an 8k-sectorsize filesystem from the same block device. The filesystem calls set_blksize, which sets i_blksize to 8192 and the minimum folio order to 1. Now udev resumes, still holding the order-0 folio it allocated. It then tries to schedule a read bio and do_mpage_readahead tries to create bufferheads for the folio. Unfortunately, blocks_per_folio == 0 because the page size is 4096 but the blocksize is 8192 so no bufferheads are attached and the bh walk never sets bdev. We then submit the bio with a NULL block device and crash. Therefore, truncate the page cache after flushing but before updating i_blksize. However, that's not enough -- we also need to lock out file IO and page faults during the update. Take both the i_rwsem and the invalidate_lock in exclusive mode for invalidations, and in shared mode for read/write operations. I don't know if this is the correct fix, but xfs/259 found it.
CVE-2025-38074	In the Linux kernel, the following vulnerability has been resolved: vhost-scsi: protect vq->log_used with vq->mutex The vhost-scsi completion path may access vq->log_base when vq->log_used is already set to false. vhost-thread QEMU-thread vhost_scsi_complete_cmd_work() -> vhost_add_used() -> vhost_add_used_n() if (unlikely(vq->log_used)) QEMU disables vq->log_used via VHOST_SET_VRING_ADDR. mutex_lock(&vq->mutex); vq->log_used = false now! mutex_unlock(&vq->mutex); QEMU gfree(vq->log_base) log_used() -> log_write(vq->log_base) Assuming the VMM is QEMU. The vq->log_base is from QEMU userspace and can be reclaimed via gfree(). As a result, this causes invalid memory writes to QEMU userspace. The control queue path has the same issue.
CVE-2025-38075	In the Linux kernel, the following vulnerability has been resolved: scsi: target: iscsi: Fix timeout on deleted connection NOPIN response timer may expire on a deleted connection and crash with such logs: Did not receive response to NOPIN on CID: 0, failing connection for I_T Nexus (null),i,0x00023d000125,iqn.2017-01.com.iscsi.target,t,0x3d BUG: Kernel NULL pointer dereference on read at 0x00000000 NIP strlcpy+0x8/0xb0 LR iscsit_fill_cxn_timeout_err_stats+0x5c/0xc0 [iscsi_target_mod] Call Trace: iscsit_handle_nopin_response_timeout+0xfc/0x120 [iscsi_target_mod] call_timer_fn+0x58/0x1f0 run_timer_softirq+0x740/0x860 __do_softirq+0x16c/0x420 irq_exit+0x188/0x1c0 timer_interrupt+0x184/0x410 That is because nopin response timer may be re-started on nopin timer expiration. Stop nopin timer before stopping the nopin response timer to be sure that no one of them will be re-started.
CVE-2025-38077	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-wmi-sysman: Avoid buffer overflow in current_password_store() If the 'buf' array received from the user contains an empty string, the 'length' variable will be zero. Accessing the 'buf' array element with index 'length - 1' will result in a buffer overflow. Add a check for an empty string. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-38078	In the Linux kernel, the following vulnerability has been resolved: ALSA: pcm: Fix race of buffer access at PCM OSS layer The PCM OSS layer tries to clear the buffer with the silence data at initialization (or reconfiguration) of a stream with the explicit call of snd_pcm_format_set_silence() with runtime->dma_area. But this may lead to a UAF because the accessed runtime->dma_area might be freed concurrently, as it's performed outside the PCM ops. For avoiding it, move the code into the PCM core and perform it inside the buffer access lock, so that it won't be changed during the operation.
CVE-2025-38079	In the Linux kernel, the following vulnerability has been resolved: crypto: algif_hash - fix double free in hash_accept If accept(2) is called on socket type algif_hash with MSG_MORE flag set and crypto_ahash_import fails, sk2 is freed. However, it is also freed in af_alg_release, leading to slab-use-after-free error.

CVE ID	Description
CVE-2025-38080	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Increase block_sequence array size [Why] It's possible to generate more than 50 steps in hwss_build_fast_sequence, for example with a 6-pipe asic where all pipes are in one MPC chain. This overflows the block_sequence buffer and corrupts block_sequence_steps, causing a crash. [How] Expand block_sequence to 100 items. A naive upper bound on the possible number of steps for a 6-pipe asic, ignoring the potential for steps to be mutually exclusive, is 91 with current code, therefore 100 is sufficient.
CVE-2025-38081	In the Linux kernel, the following vulnerability has been resolved: spi-rockchip: Fix register out of bounds access Do not write native chip select stuff for GPIO chip selects. GPIOs can be numbered much higher than native CS. Also, it makes no sense.
CVE-2025-38082	In the Linux kernel, the following vulnerability has been resolved: gpio: virtuser: fix potential out-of-bound write If the caller wrote more characters, count is truncated to the max available space in "simple_write_to_buffer". Check that the input size does not exceed the buffer size. Write a zero termination afterwards.
CVE-2025-38083	In the Linux kernel, the following vulnerability has been resolved: net_sched: prio: fix a race in prio_tune() Gerrard Tai reported a race condition in PRIO, whenever SFQ perturb timer fires at the wrong time. The race is as follows: CPU 0 CPU 1 [1]: lock root [2]: qdisc_tree_flush_backlog() [3]: unlock root [4]: [5]: lock root [6]: rehash [7]: qdisc_tree_reduce_backlog() [4]: qdisc_put() This can be abused to underflow a parent's qlen. Calling qdisc_purge_queue() instead of qdisc_tree_flush_backlog() should fix the race, because all packets will be purged from the qdisc before releasing the lock.
CVE-2025-38084	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: unshare page tables during VMA split, not before Currently, __split_vma() triggers hugetlb page table unsharing through vm_ops->may_split(). This happens before the VMA lock and rmap locks are taken - which is too early, it allows racing VMA-locked page faults in our process and racing rmap walks from other processes to cause page tables to be shared again before we actually perform the split. Fix it by explicitly calling into the hugetlb unshare logic from __split_vma() in the same place where THP splitting also happens. At that point, both the VMA and the rmap(s) are write-locked. An annoying detail is that we can now call into the helper hugetlb_unshare_pmds() from two different locking contexts: 1. from hugetlb_split(), holding: - mmap lock (exclusively) - VMA lock - file rmap lock (exclusively) 2. hugetlb_unshare_all_pmds(), which I think is designed to be able to call us with only the mmap lock held (in shared mode), but currently only runs while holding mmap lock (exclusively) and VMA lock Backporting note: This commit fixes a racy protection that was introduced in commit b30c14cd6102 ("hugetlb: unshare some PMDs when splitting VMAs"); that commit claimed to fix an issue introduced in 5.13, but it should actually also go all the way back. [jannh@google.com: v2]
CVE-2025-38085	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: fix huge_pmd_unshare() vs GUP-fast race huge_pmd_unshare() drops a reference on a page table that may have previously been shared across processes, potentially turning it into a normal page table used in another process in which unrelated VMAs can afterwards be installed. If this happens in the middle of a concurrent gup_fast(), gup_fast() could end up walking the page tables of another process. While I don't see any way in which that immediately leads to kernel memory corruption, it is really weird and unexpected. Fix it with an explicit broadcast IPI through tlb_remove_table_sync_one(), just like we do in khugepaged when removing page tables for a THP collapse.
CVE-2025-38086	In the Linux kernel, the following vulnerability has been resolved: net: ch9200: fix uninitialised access during mii_nway_restart In mii_nway_restart() the code attempts to call mii->mdio_read which is ch9200_mdio_read(). ch9200_mdio_read() utilises a local buffer called "buff", which is initialised with control_read(). However "buff" is conditionally initialised inside control_read(): if (err == size) { memcpy(data, buf, size); } If the condition of "err == size" is not met, then "buff" remains uninitialised. Once this happens the uninitialised "buff" is accessed and returned during ch9200_mdio_read(): return (buff[0] buff[1] << 8); The problem stems from the fact that ch9200_mdio_read() ignores the return value of control_read(), leading to uninit-access of "buff". To fix this we should check the return value of control_read() and return early on error.
CVE-2025-38087	In the Linux kernel, the following vulnerability has been resolved: net/sched: fix use-after-free in taprio_dev_notifier Since taprio's taprio_dev_notifier() isn't protected by an RCU read-side critical section, a race with advance_sched() can lead to a use-after-free. Adding rcu_read_lock() inside taprio_dev_notifier() prevents this.
CVE-2025-38088	In the Linux kernel, the following vulnerability has been resolved: powerpc/powernv/memtrace: Fix out of bounds issue in memtrace mmap memtrace mmap issue has an out of bounds issue. This patch fixes the by checking that the requested mapping region size should stay within the allocated region size.

CVE ID	Description
CVE-2025-38089	In the Linux kernel, the following vulnerability has been resolved: sunrpc: handle SVC_GARBAGE during svc auth processing as auth error tianshuo han reported a remotely-triggerable crash if the client sends a kernel RPC server a specially crafted packet. If decoding the RPC reply fails in such a way that SVC_GARBAGE is returned without setting the rq_accept_statp pointer, then that pointer can be dereferenced and a value stored there. If it's the first time the thread has processed an RPC, then that pointer will be set to NULL and the kernel will crash. In other cases, it could create a memory scribble. The server sunrpc code treats a SVC_GARBAGE return from svc_authenticate or pg_authenticate as if it should send a GARBAGE_ARGS reply. RFC 5531 says that if authentication fails that the RPC should be rejected instead with a status of AUTH_ERR. Handle a SVC_GARBAGE return as an AUTH_ERROR, with a reason of AUTH_BADCRED instead of returning GARBAGE_ARGS in that case. This sidesteps the whole problem of touching the rpc_accept_statp pointer in this situation and avoids the crash.
CVE-2025-38090	In the Linux kernel, the following vulnerability has been resolved: drivers/rapidio/rio_cm.c: prevent possible heap overwrite In rio_cm_cdev_ioctl(RIO_CM_CHAN_SEND) -> cm_chan_msg_send() -> rio_cm_ch_send() cm_chan_msg_send() checks that userspace didn't send too much data but rio_cm_ch_send() failed to check that userspace sent sufficient data. The result is that rio_cm_ch_send() can write to fields in the rio_ch_chan_hdr which were outside the bounds of the space which cm_chan_msg_send() allocated. Address this by teaching rio_cm_ch_send() to check that the entire rio_ch_chan_hdr was copied in from userspace.
CVE-2025-38091	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: check stream id dml21 wrapper to get plane_id [Why & How] Fix a false positive warning which occurs due to lack of correct checks when querying plane_id in DML21. This fixes the warning when performing a mode1 reset (cat /sys/kernel/debug/dri/1/amdgpu_gpu_recover): [35.751250] WARNING: CPU: 11 PID: 326 at /tmp/amd.PHPyAl7v/amd/amdgpu/./display/dc/dml2/dml2_dc_resource_mgmt.c:91 dml2_map_dc_pipes+0x243d/0x3f40 [amdgpu] [35.751434] Modules linked in: amdgpu(OE) amddrm_ttm_helper(OE) amdtm(OE) amddrm_buddy(OE) amdxcp(OE) amddrm_exec(OE) amd_sched(OE) amdkcl(OE) drm_suballoc_helper drm_ttm_helper ttm drm_display_helper cec rc_core i2c_algo_bit rfcomm qtr cmac algif_hash algif_skcipher af_alg bnep amd_atl intel_rapl_msr intel_rapl_common snd_hda_codec_hdmi snd_hda_intel edac_mce_amd snd_intel_dspcfg snd_intel_sdw_acpi snd_hda_codec kvm_amd snd_hda_core snd_hwdep snd_pcm kvm snd_seq_midi snd_seq_midi_event snd_rawmidi crct10dif_pclmul polyval_clmulni polyval_generic btusb ghash_clmulni_intel sha256_ssse3 btrtl sha1_ssse3 snd_seq_bintell aesni_intel btbcm btmtk snd_seq_device crypto_simd sunrpc cryptd bluetooth snd_timer ccp binfmt_misc rapl snd_i2c_piix4 wmi_bmof gigabyte_wmi k10temp i2c_smbus soundcore gpio_amdpt mac_hid sch_fq_codel msr parport_pc ppdev lp parport efi_pstore nfnetlink dmi_sysfs ip_tables x_tables autofs4 hid_generic usbhid hid crc32_pclmul igc ahci xhci_pci libahci xhci_pci_renesas video wmi [35.751501] CPU: 11 UID: 0 PID: 326 Comm: kworker/u64:9 Tainted: G OE 6.11.0-21-generic #21~24.04.1-Ubuntu [35.751504] Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE [35.751505] Hardware name: Gigabyte Technology Co., Ltd. X670E AORUS PRO X/X670E AORUS PRO X, BIOS F30 05/22/2024 [35.751506] Workqueue: amdgpu-reset-dev amdgpu_debugfs_reset_work [amdgpu] [35.751638] RIP: 0010:dml2_map_dc_pipes+0x243d/0x3f40 [amdgpu] [35.751794] Code: 6d 0c 00 00 8b 84 24 88 00 00 00 41 3b 44 9c 20 0f 84 fc 07 00 00 48 83 c3 01 48 83 fb 06 75 b3 4c 8b 64 24 68 4c 8b 6c 24 40 <0f> 0b b8 06 00 00 00 49 8b 94 24 a0 49 00 00 89 c3 83 f8 07 0f 87 [35.751796] RSP: 0018:ffffbfa3805d7680 EFLAGS: 00010246 [35.751798] RAX: 00000000000010000 RBX: 0000000000000006 RCX: 0000000000000000 [35.751799] RDX: 0000000000000000 RSI: 0000000000000005 RDI: 0000000000000000 [35.751800] RBP: fffffbfa3805d78f0 R08: 0000000000000000 R09: 0000000000000000 [35.751801] R10: 0000000000000000 R11: 0000000000000000 R12: fffffbfa383249000 [35.751802] R13: fffffa0e68f280000 R14: fffffbfa383249658 R15: 0000000000000000 [35.751803] FS: 0000000000000000(0000) GS:ffffa0edbe580000(0000) knlGS:0000000000000000 [35.751804] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [35.751805] CR2: 00005d847ef96c58 CR3: 000000041de3e000 CR4: 0000000000f50ef0 [35.751806] PKRU: 55555554 [35.751807] Call Trace: [35.751810] <TASK> [35.751816] ? show_regs+0x6c/0x80 [35.751820] ? __warn+0x88/0x140 [35.751822] ? dml2_map_dc_pipes+0x243d/0x3f40 [amdgpu] [35.751964] ? report_bug+0x182/0x1b0 [35.751969] ? handle_bug+0x6e/0xb0 [35.751972] ? exc_invalid_op+0x18/0x80 [35.751974] ? asm_exc_invalid_op+0x1b/0x20 [35.751978] ? dml2_map_dc_pipes+0x243d/0x3f40 [amdgpu] [35.752117] ? math_pow+0x48/0xa0 [amdgpu] [35.752256] ? srso_alias_return_thunk+0x5/0xfbef5 [35.752260] ? math_pow+0x48/0xa0 [amdgpu] [35.752400] ? srso_alias_return_thunk+0x5/0xfbef5 [35.752403] ? math_pow+0x11/0xa0 [amdgpu] [35.752524] ? srso_alias_return_thunk+0x5/0xfbef5 [35.752526] ? core_dcn4_mode_programming+0xe4d/0x20d0 [amdgpu] [35.752663] ? srso_alias_return_thunk+0x5/0xfbef5 [35.752669] dml21_validate+0x3d4/0x980 [amdgpu] (cherry picked from commit f8ad62c0a93e5dd94243e10f1b742232e4d6411e)
CVE-2025-38092	In the Linux kernel, the following vulnerability has been resolved: ksmbd: use list_first_entry_or_null for opinfo_get_list() The list_first_entry() macro never returns NULL. If the list is empty then it returns an invalid pointer. Use list_first_entry_or_null() to check if the list is empty.

CVE ID	Description
CVE-2025-38094	In the Linux kernel, the following vulnerability has been resolved: net: cadence: macb: Fix a possible deadlock in macb_halt_tx. There is a situation where after THALT is set high, TGO stays high as well. Because jiffies are never updated, as we are in a context with interrupts disabled, we never exit that loop and have a deadlock. That deadlock was noticed on a sama5d4 device that stayed locked for days. Use retries instead of jiffies so that the timeout really works and we do not have a deadlock anymore.
CVE-2025-38095	In the Linux kernel, the following vulnerability has been resolved: dma-buf: insert memory barrier before updating num_fences smp_store_mb() inserts memory barrier after storing operation. It is different with what the comment is originally aiming so Null pointer dereference can be happened if memory update is reordered.
CVE-2025-38096	In the Linux kernel, the following vulnerability has been resolved: wifi: iwlwifi: don't warn when if there is a FW error iwl_trans_reclaim is warning if it is called when the FW is not alive. But if it is called when there is a pending restart, i.e. after a FW error, there is no need to warn, instead - return silently.
CVE-2025-38097	In the Linux kernel, the following vulnerability has been resolved: espintcp: remove encap socket caching to avoid reference leak The current scheme for caching the encap socket can lead to reference leaks when we try to delete the netns. The reference chain is: xfrm_state -> enacp_sk -> netns Since the encap socket is a userspace socket, it holds a reference on the netns. If we delete the espintcp state (through flush or individual delete) before removing the netns, the reference on the socket is dropped and the netns is correctly deleted. Otherwise, the netns may not be reachable anymore (if all processes within the ns have terminated), so we cannot delete the xfrm state to drop its reference on the socket. This patch results in a small (~2% in my tests) performance regression. A GC-type mechanism could be added for the socket cache, to clear references if the state hasn't been used "recently", but it's a lot more complex than just not caching the socket.
CVE-2025-38098	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Don't treat wb connector as physical in create_validate_stream_for_sink Don't try to operate on a drm_wb_connector as an amdgpu_dm_connector. While dereferencing aconnector->base will "work" it's wrong and might lead to unknown bad things. Just... don't.
CVE-2025-38099	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Disable SCO support if READ_VOICE_SETTING is unsupported/broken A SCO connection without the proper voice_setting can cause the controller to lock up.
CVE-2025-38100	In the Linux kernel, the following vulnerability has been resolved: x86/iopl: Cure TIF_IO_BITMAP inconsistencies io_bitmap_exit() is invoked from exit_thread() when a task exists or when a fork fails. In the latter case the exit_thread() cleans up resources which were allocated during fork(). io_bitmap_exit() invokes task_update_io_bitmap(), which in turn ends up in tss_update_io_bitmap(). tss_update_io_bitmap() operates on the current task. If current has TIF_IO_BITMAP set, but no bitmap installed, tss_update_io_bitmap() crashes with a NULL pointer dereference. There are two issues, which lead to that problem: 1) io_bitmap_exit() should not invoke task_update_io_bitmap() when the task, which is cleaned up, is not the current task. That's a clear indicator for a cleanup after a failed fork(). 2) A task should not have TIF_IO_BITMAP set and neither a bitmap installed nor IOPL emulation level 3 activated. This happens when a kernel thread is created in the context of a user space thread, which has TIF_IO_BITMAP set as the thread flags are copied and the IO bitmap pointer is cleared. Other than in the failed fork() case this has no impact because kernel threads including IO workers never return to user space and therefore never invoke tss_update_io_bitmap(). Cure this by adding the missing cleanups and checks: 1) Prevent io_bitmap_exit() to invoke task_update_io_bitmap() if the to be cleaned up task is not the current task. 2) Clear TIF_IO_BITMAP in copy_thread() unconditionally. For user space forks it is set later, when the IO bitmap is inherited in io_bitmap_share(). For paranoia sake, add a warning into tss_update_io_bitmap() to catch the case, when that code is invoked with inconsistent state.
CVE-2025-38101	In the Linux kernel, the following vulnerability has been resolved: ring-buffer: Fix buffer locking in ring_buffer_subbuf_order_set() Enlarge the critical section in ring_buffer_subbuf_order_set() to ensure that error handling takes place with per-buffer mutex held, thus preventing list corruption and other concurrency-related issues.

CVE ID	Description
CVE-2025-38102	In the Linux kernel, the following vulnerability has been resolved: VMCI: fix race between vmci_host_setup_notify and vmci_ctx_unset_notify During our test, it is found that a warning can be trigger in try_grab_folio as follow: -----[cut here]----- WARNING: CPU: 0 PID: 1678 at mm/gup.c:147 try_grab_folio+0x106/0x130 Modules linked in: CPU: 0 UID: 0 PID: 1678 Comm: syz.3.31 Not tainted 6.15.0-rc5 #163 PREEMPT(undef) RIP: 0010:try_grab_folio+0x106/0x130 Call Trace: <TASK> follow_huge_pmd+0x240/0x8e0 follow_pmd_mask.constprop.0.isra.0+0x40b/0x5c0 follow_pud_mask.constprop.0.isra.0+0x14a/0x170 follow_page_mask+0x1c2/0x1f0 __get_user_pages+0x176/0x950 __gup_longterm_locked+0x15b/0x1060 ? gup_fast+0x120/0x1f0 gup_fast_fallback+0x17e/0x230 get_user_pages_fast+0x5f/0x80 vmci_host_unlocked_ioctl+0x21c/0xf80 RIP: 0033:0x54d2cd ---[end trace 0000000000000000]--- Digging into the source, context->notify_page may init by get_user_pages_fast and can be seen in vmci_ctx_unset_notify which will try to put_page. However get_user_pages_fast is not finished here and lead to following try_grab_folio warning. The race condition is shown as follow: cpu0 cpu1 vmci_host_do_set_notify vmci_host_setup_notify get_user_pages_fast(uva, 1, FOLL_WRITE, &context->notify_page); lockless_pages_from_mm gup_pgd_range gup_huge_pmd // update &context->notify_page vmci_host_do_set_notify vmci_ctx_unset_notify notify_page = context->notify_page; if (notify_page) put_page(notify_page); // page is freed __gup_longterm_locked __get_user_pages follow_trans_huge_pmd try_grab_folio // warn here To slove this, use local variable page to make notify_page can be seen after finish get_user_pages_fast.
CVE-2025-38103	In the Linux kernel, the following vulnerability has been resolved: HID: usbhid: Eliminate recurrent out-of-bounds bug in usbhid_parse() Update struct hid_descriptor to better reflect the mandatory and optional parts of the HID Descriptor as per USB HID 1.11 specification. Note: the kernel currently does not parse any optional HID class descriptors, only the mandatory report descriptor. Update all references to member element desc[0] to rpt_desc. Add test to verify bLength and bNumDescriptors values are valid. Replace the for loop with direct access to the mandatory HID class descriptor member for the report descriptor. This eliminates the possibility of getting an out-of-bounds fault. Add a warning message if the HID descriptor contains any unsupported optional HID class descriptors.

CVE ID	Description
CVE-2025-38104	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Replace Mutex with Spinlock for RLCG register access to avoid Priority Inversion in SRIOV RLCG Register Access is a way for virtual functions to safely access GPU registers in a virtualized environment., including TLB flushes and register reads. When multiple threads or VFs try to access the same registers simultaneously, it can lead to race conditions. By using the RLCG interface, the driver can serialize access to the registers. This means that only one thread can access the registers at a time, preventing conflicts and ensuring that operations are performed correctly. Additionally, when a low-priority task holds a mutex that a high-priority task needs, ie., If a thread holding a spinlock tries to acquire a mutex, it can lead to priority inversion. register access in amdgpu_virt_rlcg_reg_rw especially in a fast code path is critical. The call stack shows that the function amdgpu_virt_rlcg_reg_rw is being called, which attempts to acquire the mutex. This function is invoked from amdgpu_sriov_wreg, which in turn is called from gmc_v11_0_flush_gpu_tlb. The [BUG: Invalid wait context] indicates that a thread is trying to acquire a mutex while it is in a context that does not allow it to sleep (like holding a spinlock). Fixes the below: [253.013423] ===== [253.013434] [BUG: Invalid wait context] [253.013446] 6.12.0-amdstaging-drm-next-lol-050225 #14 Tainted: G U OE [253.013464] ----- [253.013475] kworker/0:1/10 is trying to lock: [253.013487] ffff9f30542e3cf8 (&adev->virt.rlcg_reg_lock){+.+.}-{3:3}, at: amdgpu_virt_rlcg_reg_rw+0xf6/0x330 [amdgpu] [253.013815] other info that might help us debug this: [253.013827] context-{4:4} [253.013835] 3 locks held by kworker/0:1/10: [253.013847] #0: ffff9f3040050f58 ((wq_completion)events){+.+.}-{0:0}, at: process_one_work+0x3f5/0x680 [253.013877] #1: ffff789c008be40 ((work_completion)(&wfc.work)){+.+.}-{0:0}, at: process_one_work+0x1d6/0x680 [253.013905] #2: ffff9f3054281838 (&adev->gmc.invalidate_lock){+.+.}-{2:2}, at: gmc_v11_0_flush_gpu_tlb+0x198/0x4f0 [amdgpu] [253.014154] stack backtrace: [253.014164] CPU: 0 UID: 0 PID: 10 Comm: kworker/0:1 Tainted: G U OE 6.12.0-amdstaging-drm-next-lol-050225 #14 [253.014189] Tainted: [U]=USER, [O]=OOT_MODULE, [E]=UNSIGNED_MODULE [253.014203] Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS Hyper-V UEFI Release v4.1 11/18/2024 [253.014224] Workqueue: events work_for_cpu_fn [253.014241] Call Trace: [253.014250] <TASK> [253.014260] dump_stack_lvl+0x9b/0xf0 [253.014275] dump_stack+0x10/0x20 [253.014287] __lock_acquire+0xa47/0x2810 [253.014303] ? srso_alias_return_thunk+0x5/0xfbef5 [253.014321] lock_acquire+0xd1/0x300 [253.014333] ? amdgpu_virt_rlcg_reg_rw+0xf6/0x330 [amdgpu] [253.014562] ? __lock_acquire+0xa6b/0x2810 [253.014578] __mutex_lock+0x85/0xe20 [253.014591] ? amdgpu_virt_rlcg_reg_rw+0xf6/0x330 [amdgpu] [253.014782] ? sched_clock_noinstr+0x9/0x10 [253.014795] ? srso_alias_return_thunk+0x5/0xfbef5 [253.014808] ? local_clock_noinstr+0xe/0xc0 [253.014822] ? amdgpu_virt_rlcg_reg_rw+0xf6/0x330 [amdgpu] [253.015012] ? srso_alias_return_thunk+0x5/0xfbef5 [253.015029] mutex_lock_nested+0x1b/0x30 [253.015044] ? mutex_lock_nested+0x1b/0x30 [253.015057] amdgpu_virt_rlcg_reg_rw+0xf6/0x330 [amdgpu] [253.015249] amdgpu_sriov_wreg+0xc5/0xd0 [amdgpu] [253.015435] gmc_v11_0_flush_gpu_tlb+0x44b/0x4f0 [amdgpu] [253.015667] gfx_v11_0_hw_init+0x499/0x29c0 [amdgpu] [253.015901] ? __pfx_smu_v13_0_update_pcie_parameters+0x10/0x10 [amdgpu] [253.016159] ? srso_alias_return_thunk+0x5/0xfbef5 [253.016173] ? smu_hw_init+0x18d/0x300 [amdgpu] [253.016403] amdgpu_device_init+0x29ad/0x36a0 [amdgpu] [253.016614] amdgpu_driver_load_kms+0x1a/0xc0 [amdgpu] [253.0170 ---truncated---
CVE-2025-38105	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Kill timer properly at removal The USB-audio MIDI code initializes the timer, but in a rare case, the driver might be freed without the disconnect call. This leaves the timer in an active state while the assigned object is released via snd_usbmidi_free(), which ends up with a kernel warning when the debug configuration is enabled, as spotted by fuzzer. For avoiding the problem, put timer_shutdown_sync() at snd_usbmidi_free(), so that the timer can be killed properly. While we're at it, replace the existing timer_delete_sync() at the disconnect callback with timer_shutdown_sync(), too.
CVE-2025-38107	In the Linux kernel, the following vulnerability has been resolved: net_sched: ets: fix a race in ets_qdisc_change() Gerrard Tai reported a race condition in ETS, whenever SFQ perturb timer fires at the wrong time. The race is as follows: CPU 0 CPU 1 [1]: lock root [2]: qdisc_tree_flush_backlog() [3]: unlock root [5]: lock root [6]: rehash [7]: qdisc_tree_reduce_backlog() [4]: qdisc_put() This can be abused to underflow a parent's qlen. Calling qdisc_purge_queue() instead of qdisc_tree_flush_backlog() should fix the race, because all packets will be purged from the qdisc before releasing the lock.
CVE-2025-38108	In the Linux kernel, the following vulnerability has been resolved: net_sched: red: fix a race in __red_change() Gerrard Tai reported a race condition in RED, whenever SFQ perturb timer fires at the wrong time. The race is as follows: CPU 0 CPU 1 [1]: lock root [2]: qdisc_tree_flush_backlog() [3]: unlock root [5]: lock root [6]: rehash [7]: qdisc_tree_reduce_backlog() [4]: qdisc_put() This can be abused to underflow a parent's qlen. Calling qdisc_purge_queue() instead of qdisc_tree_flush_backlog() should fix the race, because all packets will be purged from the qdisc before releasing the lock.

CVE ID	Description
CVE-2025-38109	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix ECVF vports unload on shutdown flow Fix shutdown flow UAF when a virtual function is created on the embedded chip (ECVF) of a BlueField device. In such case the vport acl ingress table is not properly destroyed. ECVF functionality is independent of ecvp_vport_exists capability and thus functions mlx5_eswitch_(enable disable)_pf_vf_vports() should not test it when enabling/disabling ECVF vports. kernel log: [] refcount_t: underflow; use-after-free. [] WARNING: CPU: 3 PID: 1 at lib/refcount.c:28 refcount_warn_saturate+0x124/0x220 ----- [] Call trace: [] refcount_warn_saturate+0x124/0x220 [] tree_put_node+0x164/0x1e0 [mlx5_core] [] mlx5_destroy_flow_table+0x98/0x2c0 [mlx5_core] [] esw_acl_ingress_table_destroy+0x28/0x40 [mlx5_core] [] esw_acl_ingress_lgcy_cleanup+0x80/0xf4 [mlx5_core] [] esw_legacy_vport_acl_cleanup+0x44/0x60 [mlx5_core] [] esw_vport_cleanup+0x64/0x90 [mlx5_core] [] mlx5_esw_vport_disable+0xc0/0x1d0 [mlx5_core] [] mlx5_eswitch_unload_ec_vf_vports+0xc0/0x150 [mlx5_core] [] mlx5_eswitch_disable_sriov+0x198/0x2a0 [mlx5_core] [] mlx5_device_disable_sriov+0xb8/0x1e0 [mlx5_core] [] mlx5_sriov_detach+0x40/0x50 [mlx5_core] [] mlx5_unload+0x40/0xc4 [mlx5_core] [] mlx5_unload_one_devl_locked+0x6c/0xe4 [mlx5_core] [] mlx5_unload_one+0x3c/0x60 [mlx5_core] [] shutdown+0x7c/0xa4 [mlx5_core] [] pci_device_shutdown+0x3c/0xa0 [] device_shutdown+0x170/0x340 [] __do_sys_reboot+0x1f4/0x2a0 [] __arm64_sys_reboot+0x2c/0x40 [] invoke_syscall+0x78/0x100 [] el0_svc_common.constprop.0+0x54/0x184 [] do_el0_svc+0x30/0xac [] el0_svc+0x48/0x160 [] el0t_64_sync_handler+0xa4/0x12c [] el0t_64_sync+0x1a4/0x1a8 [] --[end trace 9c4601d68c70030e]---
CVE-2025-38110	In the Linux kernel, the following vulnerability has been resolved: net/mdiobus: Fix potential out-of-bounds clause 45 read/write access When using publicly available tools like 'mdio-tools' to read/write data from/to network interface and its PHY via C45 (clause 45) mdiobus, there is no verification of parameters passed to the ioctl and it accepts any mdio address. Currently there is support for 32 addresses in kernel via PHY_MAX_ADDR define, but it is possible to pass higher value than that via ioctl. While read/write operation should generally fail in this case, mdiobus provides stats array, where wrong address may allow out-of-bounds read/write. Fix that by adding address verification before C45 read/write operation. While this excludes this access from any statistics, it improves security of read/write operation.
CVE-2025-38111	In the Linux kernel, the following vulnerability has been resolved: net/mdiobus: Fix potential out-of-bounds read/write access When using publicly available tools like 'mdio-tools' to read/write data from/to network interface and its PHY via mdiobus, there is no verification of parameters passed to the ioctl and it accepts any mdio address. Currently there is support for 32 addresses in kernel via PHY_MAX_ADDR define, but it is possible to pass higher value than that via ioctl. While read/write operation should generally fail in this case, mdiobus provides stats array, where wrong address may allow out-of-bounds read/write. Fix that by adding address verification before read/write operation. While this excludes this access from any statistics, it improves security of read/write operation.
CVE-2025-38112	In the Linux kernel, the following vulnerability has been resolved: net: Fix TOCTOU issue in sk_is_readable() sk->sk_prot->sock_is_readable is a valid function pointer when sk resides in a sockmap. After the last sk_psock_put() (which usually happens when socket is removed from sockmap), sk->sk_prot gets restored and sk->sk_prot->sock_is_readable becomes NULL. This makes sk_is_readable() racy, if the value of sk->sk_prot is reloaded after the initial check. Which in turn may lead to a null pointer dereference. Ensure the function pointer does not turn NULL after the check.
CVE-2025-38113	In the Linux kernel, the following vulnerability has been resolved: ACPI: CPPC: Fix NULL pointer dereference when nosmp is used With nosmp in cmdline, other CPUs are not brought up, leaving their cpc_desc_ptr NULL. CPU0's iteration via for_each_possible_cpu() dereferences these NULL pointers, causing panic. Panic backtrace: [0.401123] Unable to handle kernel NULL pointer dereference at virtual address 00000000000000b8 ... [0.403255] [<ffff8009a5818>] cppc_allow_fast_switch+0x6a/0xd4 ... Kernel panic - not syncing: Attempted to kill init! [rjw: New subject]
CVE-2025-38115	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: fix a potential crash on gso_skb handling SFQ has an assumption of always being able to queue at least one packet. However, after the blamed commit, sch->q.len can be inflated by packets in sch->gso_skb, and an enqueue() on an empty SFQ qdisc can be followed by an immediate drop. Fix sfq_drop() to properly clear q->tail in this situation. ip netns add lb ip link add dev to-lb type veth peer name in-lb netns lb ethtool -K to-lb tso off # force qdisc to requeue gso_skb ip netns exec lb ethtool -K in-lb gro on # enable NAPI ip link set dev to-lb up ip -netns lb link set dev in-lb up ip addr add dev to-lb 192.168.20.1/24 ip -netns lb addr add dev in-lb 192.168.20.2/24 tc qdisc replace dev to-lb root sfq limit 100 ip netns exec lb netserver netperf -H 192.168.20.2 -l 100 & netperf -H 192.168.20.2 -l 100 & netperf -H 192.168.20.2 -l 100 & netperf -H 192.168.20.2 -l 100 &

CVE ID	Description
CVE-2025-38117	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Protect mgmt_pending list with its own lock This uses a mutex to protect from concurrent access of mgmt_pending list which can cause crashes like: <pre>===== BUG: KASAN: slab-use-after-free in hci_sock_get_channel+0x60/0x68 net/bluetooth/hci_sock.c:91 Read of size 2 at addr ffff0000c48885b2 by task syz.4.334/7318 CPU: 0 UID: 0 PID: 7318 Comm: syz.4.334 Not tainted 6.15.0-rc7-syzkaller-g187899f4124a #0 PREEMPT Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 Call trace: show_stack +0x2c/0x3c arch/arm64/kernel/stacktrace.c:466 (C) __dump_stack+0x30/0x40 lib/dump_stack.c:94 dump_stack_lvl+0xd8/0x12c lib/dump_stack.c:120 print_address_description+0xa8/0x254 mm/ kasan/report.c:408 print_report+0x68/0x84 mm/kasan/report.c:521 kasan_report+0xb0/0x110 mm/kasan/report.c:634 __asan_report_load2_noabort+0x20/0x2c mm/kasan/report_generic.c:379 hci_sock_get_channel+0x60/0x68 net/bluetooth/hci_sock.c:91 mgmt_pending_find+0x7c/0x140 net/ bluetooth/mgmt_util.c:223 pending_find net/bluetooth/mgmt.c:947 [inline] remove_adv_monitor +0x44/0x1a4 net/bluetooth/mgmt.c:5445 hci_mgmt_cmd+0x780/0xc00 net/bluetooth/hci_sock.c:1712 hci_sock_sendmsg+0x544/0xbb0 net/bluetooth/hci_sock.c:1832 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg net/socket.c:727 [inline] sock_write_iter+0x25c/0x378 net/socket.c:1131 new_sync_write fs/read_write.c:591 [inline] vfs_write+0x62c/0x97c fs/read_write.c:684 ksys_write +0x120/0x210 fs/read_write.c:736 __do_sys_write fs/read_write.c:747 [inline] __se_sys_write fs/ read_write.c:744 [inline] __arm64_sys_write+0x7c/0x90 fs/read_write.c:744 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/ kernel/syscall.c:151 el0_svc+0x58/0x17c arch/arm64/kernel/entry-common.c:767 el0t_64_sync_handler +0x78/0x108 arch/arm64/kernel/entry-common.c:786 el0t_64_sync+0x198/0x19c arch/arm64/ kernel/entry.S:600 Allocated by task 7037: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x40/0x78 mm/kasan/common.c:68 kasan_save_alloc_info+0x44/0x54 mm/ kasan/generic.c:562 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc +0x9c/0xb4 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __do_kmalloc_node mm/slub.c:4327 [inline] __kmalloc_noprof+0x2fc/0x4c8 mm/slub.c:4339 kmalloc_noprof include/linux/slab.h:909 [inline] sk_prot_alloc+0xc4/0x1f0 net/core/sock.c:2198 sk_alloc+0x44/0x3ac net/core/sock.c:2254 bt_sock_alloc+0x4c/0x300 net/bluetooth/af_bluetooth.c:148 hci_sock_create+0xa8/0x194 net/bluetooth/hci_sock.c:2202 bt_sock_create+0x14c/0x24c net/ bluetooth/af_bluetooth.c:132 __sock_create+0x43c/0x91c net/socket.c:1541 sock_create net/ socket.c:1599 [inline] __sys_socket_create net/socket.c:1636 [inline] __sys_socket+0xd4/0x1c0 net/ socket.c:1683 __do_sys_socket net/socket.c:1697 [inline] __se_sys_socket net/socket.c:1695 [inline] __arm64_sys_socket+0x7c/0x94 net/socket.c:1695 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc +0x58/0x17c arch/arm64/kernel/entry-common.c:767 el0t_64_sync_handler+0x78/0x108 arch/arm64/ kernel/entry-common.c:786 el0t_64_sync+0x198/0x19c arch/arm64/kernel/entry.S:600 Freed by task 6607: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x40/0x78 mm/kasan/ common.c:68 kasan_save_free_info+0x58/0x70 mm/kasan/generic.c:576 poison_slab_object mm/ kasan/common.c:247 [inline] __kasan_slab_free+0x68/0x88 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] ---truncated---</pre>

CVE ID	Description
CVE-2025-38118	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix UAF on mgmt_remove_adv_monitor_complete This reworks MGMT_OP_REMOVE_ADV_MONITOR to not use mgmt_pending_add to avoid crashes like bellow: ===== BUG: KASAN: slab-use-after-free in mgmt_remove_adv_monitor_complete+0xe5/0x540 net/bluetooth/mgmt.c:5406 Read of size 8 at addr ffff88801c53f318 by task kworker/u5:5/5341 CPU: 0 UID: 0 PID: 5341 Comm: kworker/u5:5 Not tainted 6.15.0-syzkaller-10402-g4cb6c8af8591 #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> dump_stack_lvl+0x189/0x250 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xd2/0x2b0 mm/kasan/report.c:521 kasan_report+0x118/0x150 mm/kasan/report.c:634 mgmt_remove_adv_monitor_complete+0xe5/0x540 net/bluetooth/mgmt.c:5406 hci_cmd_sync_work+0x261/0x3a0 net/bluetooth/hci_sync.c:334 process_one_work kernel/workqueue.c:3238 [inline] process_scheduled_works+0xade/0x17b0 kernel/workqueue.c:3321 worker_thread+0x8a0/0xda0 kernel/workqueue.c:3402 kthread+0x711/0x8a0 kernel/kthread.c:464 ret_from_fork+0x3fc/0x770 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 </TASK> Allocated by task 5987: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x93/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x230/0x3d0 mm/slub.c:4358 kmalloc_noprof include/linux/slab.h:905 [inline] kzalloc_noprof include/linux/slab.h:1039 [inline] mgmt_pending_new+0x65/0x240 net/bluetooth/mgmt_util.c:252 mgmt_pending_add+0x34/0x120 net/bluetooth/mgmt_util.c:279 remove_adv_monitor+0x103/0x1b0 net/bluetooth/mgmt.c:5454 hci_mgmt_cmd+0x9c9/0xef0 net/bluetooth/hci_sock.c:1719 hci_sock_sendmsg+0x6ca/0xef0 net/bluetooth/hci_sock.c:1839 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg +0x219/0x270 net/socket.c:727 sock_write_iter+0x258/0x330 net/socket.c:1131 new_sync_write fs/read_write.c:593 [inline] vfs_write+0x548/0xa90 fs/read_write.c:686 ksys_write+0x145/0x250 fs/read_write.c:738 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0xf Freed by task 5989: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x46/0x50 mm/kasan/generic.c:576 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x62/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2380 [inline] slab_free mm/slub.c:4642 [inline] kfree+0x18e/0x440 mm/slub.c:4841 mgmt_pending_foreach+0xc9/0x120 net/bluetooth/mgmt_util.c:242 mgmt_index_removed+0x10d/0x2f0 net/bluetooth/mgmt.c:9366 hci_sock_bind +0xbe9/0x1000 net/bluetooth/hci_sock.c:1314 __sys_bind_socket net/socket.c:1810 [inline] __sys_bind+0x2c3/0x3e0 net/socket.c:1841 __do_sys_bind net/socket.c:1846 [inline] __se_sys_bind net/socket.c:1844 [inline] __x64_sys_bind+0x7a/0x90 net/socket.c:1844 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0xf
CVE-2025-38119	In the Linux kernel, the following vulnerability has been resolved: scsi: core: ufs: Fix a hang in the error handler ufshcd_err_handling_prepare() calls ufshcd_rpm_get_sync(). The latter function can only succeed if UFSHCD_EH_IN_PROGRESS is not set because resuming involves submitting a SCSI command and ufshcd_queuecommand() returns SCSI_MLQUEUE_HOST_BUSY if UFSHCD_EH_IN_PROGRESS is set. Fix this hang by setting UFSHCD_EH_IN_PROGRESS after ufshcd_rpm_get_sync() has been called instead of before. Backtrace: __switch_to +0x174/0x338 __schedule+0x600/0x9e4 schedule+0x7c/0xe8 schedule_timeout+0xa4/0x1c8 io_schedule_timeout+0x48/0x70 wait_for_common_io+0xa8/0x160 //waiting on START_STOP wait_for_completion_io_timeout+0x10/0x20 blk_execute_rq+0xe4/0x1e4 scsi_execute_cmd +0x108/0x244 ufshcd_set_dev_pwr_mode+0xe8/0x250 ufshcd_wl_resume+0x94/0x354 ufshcd_wl_runtime_resume+0x3c/0x174 scsi_runtime_resume+0x64/0xa4 rpm_resume+0x15c/0xa1c __pm_runtime_resume+0x4c/0x90 // Runtime resume ongoing ufshcd_err_handler+0x1a0/0xd08 process_one_work+0x174/0x808 worker_thread+0x15c/0x490 kthread+0xf4/0x1ec ret_from_fork +0x10/0x20 [bvanassche: rewrote patch description]
CVE-2025-38120	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_set_pipapo_avx2: fix initial map fill If the first field doesn't cover the entire start map, then we must zero out the remainder, else we leak those bits into the next match round map. The early fix was incomplete and did only fix up the generic C implementation. A followup patch adds a test case to nft_concat_range.sh.
CVE-2025-38122	In the Linux kernel, the following vulnerability has been resolved: gve: add missing NULL check for gve_alloc_pending_packet() in TX DQO gve_alloc_pending_packet() can return NULL, but gve_tx_add_skb_dqo() did not check for this case before dereferencing the returned pointer. Add a missing NULL check to prevent a potential NULL pointer dereference when allocation fails. This improves robustness in low-memory scenarios.

CVE ID	Description
CVE-2025-38123	In the Linux kernel, the following vulnerability has been resolved: net: wwan: t7xx: Fix napi rx poll issue When driver handles the napi rx polling requests, the netdev might have been released by the dellink logic triggered by the disconnect operation on user plane. However, in the logic of processing skb in polling, an invalid netdev is still being used, which causes a panic. BUG: kernel NULL pointer dereference, address: 00000000000000f1 Oops: 0000 [#1] PREEMPT SMP NOPTI RIP: 0010:dev_gro_receive+0x3a/0x620 [...] Call Trace: <IRQ> ? __die_body+0x68/0xb0 ? page_fault_oops+0x379/0x3e0 ? exc_page_fault+0x4f/0xa0 ? asm_exc_page_fault+0x22/0x30 ? __pfx_t7xx_ccmni_rcv_skb+0x10/0x10 [mtk_t7xx (HASH:1400 7)] ? dev_gro_receive+0x3a/0x620 napi_gro_receive+0xad/0x170 t7xx_ccmni_rcv_skb+0x48/0x70 [mtk_t7xx (HASH:1400 7)] t7xx_dpmaif_napi_rx_poll+0x590/0x800 [mtk_t7xx (HASH:1400 7)] net_rx_action+0x103/0x470 irq_exit_rcu+0x13a/0x310 sysvec_apic_timer_interrupt+0x56/0x90 <IRQ>
CVE-2025-38124	In the Linux kernel, the following vulnerability has been resolved: net: fix udp gso skb_segment after pull from frag_list Commit a1e40ac5b5e9 ("net: gso: fix udp gso fraglist segmentation after pull from frag_list") detected invalid geometry in frag_list skbs and redirects them from skb_segment_list to more robust skb_segment. But some packets with modified geometry can also hit bugs in that code. We don't know how many such cases exist. Addressing each one by one also requires touching the complex skb_segment code, which risks introducing bugs for other types of skbs. Instead, linearize all these packets that fail the basic invariants on gso fraglist skbs. That is more robust. If only part of the fraglist payload is pulled into head_skb, it will always cause exception when splitting skbs by skb_segment. For detailed call stack information, see below. Valid SKB_GSO_FRAGLIST skbs - consist of two or more segments - the head_skb holds the protocol headers plus first gso_size - one or more frag_list skbs hold exactly one segment - all but the last must be gso_size Optional datapath hooks such as NAT and BPF (bpf_skb_pull_data) can modify fraglist skbs, breaking these invariants. In extreme cases they pull one part of data into skb linear. For UDP, this causes three payloads with lengths of (11,11,10) bytes were pulled tail to become (12,10,10) bytes. The skbs no longer meets the above SKB_GSO_FRAGLIST conditions because payload was pulled into head_skb, it needs to be linearized before pass to regular skb_segment. skb_segment+0xcd0/0xd14 __udp_gso_segment+0x334/0x5f4 udp4_ufo_fragment+0x118/0x15c inet_gso_segment+0x164/0x338 skb_mac_gso_segment+0xc4/0x13c __skb_gso_segment+0xc4/0x124 validate_xmit_skb+0x9c/0x2c0 validate_xmit_skb_list+0x4c/0x80 sch_direct_xmit+0x70/0x404 __dev_queue_xmit+0x64c/0xe5c neigh_resolve_output+0x178/0x1c4 ip_finish_output2+0x37c/0x47c __ip_finish_output+0x194/0x240 ip_finish_output+0x20/0xf4 ip_output+0x100/0x1a0 NF_HOOK+0xc4/0x16c ip_forward+0x314/0x32c ip_rcv+0x90/0x118 __netif_receive_skb+0x74/0x124 process_backlog+0xe8/0x1a4 __napi_poll+0x5c/0x1f8 net_rx_action+0x154/0x314 handle_softirqs+0x154/0x4b8 [118.376811] [C201134] rxq0_pus: [name:bug&]kernel BUG at net/core/skbuff.c:4278! [118.376829] [C201134] rxq0_pus: [name:traps&]Internal error: Oops - BUG: 00000000f2000800 [#1] PREEMPT SMP [118.470774] [C201134] rxq0_pus: [name:mrdump&]Kernel Offset: 0x178cc00000 from 0xfffffc0080000000 [118.470810] [C201134] rxq0_pus: [name:mrdump&]PHYS_OFFSET: 0x40000000 [118.470827] [C201134] rxq0_pus: [name:mrdump&]pstate: 60400005 (nZCv daif +PAN -UAO) [118.470848] [C201134] rxq0_pus: [name:mrdump&]pc : [0xfffffd79598aefc] skb_segment+0xcd0/0xd14 [118.470900] [C201134] rxq0_pus: [name:mrdump&]lr : [0xfffffd79598a5e8] skb_segment+0x3bc/0xd14 [118.470928] [C201134] rxq0_pus: [name:mrdump&]sp : fffffc008013770
CVE-2025-38125	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: make sure that ptp_rate is not 0 before configuring EST If the ptp_rate recorded earlier in the driver happens to be 0, this bogus value will propagate up to EST configuration, where it will trigger a division by 0. Prevent this division by 0 by adding the corresponding check and error code.
CVE-2025-38126	In the Linux kernel, the following vulnerability has been resolved: net: stmmac: make sure that ptp_rate is not 0 before configuring timestamping The stmmac platform drivers that do not open-code the clk_ptp_rate value after having retrieved the default one from the device-tree can end up with 0 in clk_ptp_rate (as clk_get_rate can return 0). It will eventually propagate up to PTP initialization when bringing up the interface, leading to a divide by 0: Division by zero in kernel. CPU: 1 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.12.30-00001-g48313bd5768a #22 Hardware name: STM32 (Device Tree Support) Call trace: unwind_backtrace from show_stack+0x18/0x1c show_stack from dump_stack_lvl+0x6c/0x8c dump_stack_lvl from Ldiv0_64+0x8/0x18 Ldiv0_64 from stmmac_init_tstamp_counter+0x190/0x1a4 stmmac_init_tstamp_counter from stmmac_hw_setup+0xc1c/0x111c stmmac_hw_setup from __stmmac_open+0x18c/0x434 __stmmac_open from stmmac_open+0x3c/0xbc stmmac_open from __dev_open+0xf4/0x1ac __dev_open from __dev_change_flags+0x1cc/0x224 __dev_change_flags from dev_change_flags+0x24/0x60 dev_change_flags from ip_auto_config+0x2e8/0x11a0 ip_auto_config from do_one_initcall+0x84/0x33c do_one_initcall from kernel_init_freeable+0x1b8/0x214 kernel_init_freeable from kernel_init+0x24/0x140 kernel_init from ret_from_fork+0x14/0x28 Exception stack(0xe0815fb0 to 0xe0815ff8) Prevent this division by 0 by adding an explicit check and error log about the actual issue. While at it, remove the same check from stmmac_ptp_register, which then becomes duplicate

CVE ID	Description
CVE-2025-38127	In the Linux kernel, the following vulnerability has been resolved: ice: fix Tx scheduler error handling in XDP callback When the XDP program is loaded, the XDP callback adds new Tx queues. This means that the callback must update the Tx scheduler with the new queue number. In the event of a Tx scheduler failure, the XDP callback should also fail and roll back any changes previously made for XDP preparation. The previous implementation had a bug that not all changes made by the XDP callback were rolled back. This caused the crash with the following call trace: [+9.549584] ice 0000:ca:00:0: Failed VSI LAN queue config for XDP, error: -5 [+0.382335] Oops: general protection fault, probably for non-canonical address 0x50a2250a90495525: 0000 [#1] SMP NOPTI [+0.010710] CPU: 103 UID: 0 PID: 0 Comm: swapper/103 Not tainted 6.14.0-net-next-mar-31+ #14 PREEMPT(voluntary) [+0.010175] Hardware name: Intel Corporation M50CYP2SBSTD/M50CYP2SBSTD, BIOS SE5C620.86B.01.01.0005.2202160810 02/16/2022 [+0.010946] RIP: 0010: __ice_update_sample +0x39/0xe0 [ice] [...] [+0.002715] Call Trace: [+0.002452] <IRQ> [+0.002021] ? __die_body.cold +0x19/0x29 [+0.003922] ? die_addr+0x3c/0x60 [+0.003319] ? exc_general_protection+0x17c/0x400 [+0.004707] ? asm_exc_general_protection+0x26/0x30 [+0.004879] ? __ice_update_sample +0x39/0xe0 [ice] [+0.004835] ice_napi_poll+0x665/0x680 [ice] [+0.004320] __napi_poll+0x28/0x190 [+0.003500] net_rx_action+0x198/0x360 [+0.003752] ? update_rq_clock+0x39/0x220 [+0.004013] handle_softirqs+0xf1/0x340 [+0.003840] ? sched_clock_cpu+0xf/0x1f0 [+0.003925] __irq_exit_rcu +0xc2/0xe0 [+0.003665] common_interrupt+0x85/0xa0 [+0.003839] </IRQ> [+0.002098] <TASK> [+0.002106] asm_common_interrupt+0x26/0x40 [+0.004184] RIP: 0010:cpuidle_enter_state +0xd3/0x690 Fix this by performing the missing unmapping of XDP queues from q_vectors and setting the XDP rings pointer back to NULL after all those queues are released. Also, add an immediate exit from the XDP callback in case of ring preparation failure.
CVE-2025-38129	In the Linux kernel, the following vulnerability has been resolved: page_pool: Fix use-after-free in page_pool_recycle_in_ring syzbot reported a uaf in page_pool_recycle_in_ring: BUG: KASAN: slab-use-after-free in lock_release+0x151/0xa30 kernel/locking/lockdep.c:5862 Read of size 8 at addr ffff8880286045a0 by task syz.0.284/6943 CPU: 0 UID: 0 PID: 6943 Comm: syz.0.284 Not tainted 6.13.0-rc3-syzkaller-gdffa94ce54f41 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 09/13/2024 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report +0x143/0x180 mm/kasan/report.c:602 lock_release+0x151/0xa30 kernel/locking/lockdep.c:5862 __raw_spin_unlock_bh include/linux/spinlock_api_smp.h:165 [inline] _raw_spin_unlock_bh+0x1b/0x40 kernel/locking/spinlock.c:210 spin_unlock_bh include/linux/spinlock.h:396 [inline] ptr_ring_produce_bh include/linux/ptr_ring.h:164 [inline] page_pool_recycle_in_ring net/core/page_pool.c:707 [inline] page_pool_put_unrefed_netmem+0x748/0xb00 net/core/page_pool.c:826 page_pool_put_netmem include/net/page_pool/helpers.h:323 [inline] page_pool_put_full_netmem include/net/page_pool/helpers.h:353 [inline] napi_pp_put_page+0x149/0x2b0 net/core/skbuff.c:1036 skb_pp_recycle net/core/skbuff.c:1047 [inline] skb_free_head net/core/skbuff.c:1094 [inline] skb_release_data +0x6c4/0x8a0 net/core/skbuff.c:1125 skb_release_all net/core/skbuff.c:1190 [inline] __kfree_skb net/core/skbuff.c:1204 [inline] sk_skb_reason_drop+0x1c9/0x380 net/core/skbuff.c:1242 kfree_skb_reason include/linux/skbuff.h:1263 [inline] __skb_queue_purge_reason include/linux/skbuff.h:3343 [inline] root cause is: page_pool_recycle_in_ring ptr_ring_produce spin_lock(&r->producer_lock); WRITE_ONCE(r->queue[r->producer++], ptr) //recycle last page to pool page_pool_release page_pool_scrub page_pool_empty_ring ptr_ring_consume page_pool_return_page //release all page __page_pool_destroy free_percpu(pool->recycle_stats); free(pool) //free spin_unlock(&r->producer_lock); //pool->ring uaf read recycle_stat_inc(pool, ring); page_pool can be free while page pool recycle the last page in ring. Add producer-lock barrier to page_pool_release to prevent the page pool from being free before all pages have been recycled. recycle_stat_inc() is empty when CONFIG_PAGE_POOL_STATS is not enabled, which will trigger Wempty-body build warning. Add definition for pool stat macro to fix warning.
CVE-2025-38131	In the Linux kernel, the following vulnerability has been resolved: coresight: prevent deactivate active config while enabling the config While enable active config via cscfg_csdev_enable_active_config(), active config could be deactivated via configs' sysfs interface. This could make UAF issue in below scenario: CPU0 CPU1 (sysfs enable) load module cscfg_load_config_sets() activate config. // sysfs (sys_active_cnt == 1) ... cscfg_csdev_enable_active_config() lock(csdev->cscfg_csdev_lock) // here load config activate by CPU1 unlock(csdev->cscfg_csdev_lock) deactivate config // sysfs (sys_active_cnt == 0) cscfg_unload_config_sets() unload module // access to config_desc which freed // while unloading module. cscfg_csdev_enable_config To address this, use cscfg_config_desc's active_cnt as a reference count which will be holded when - activate the config. - enable the activated config. and put the module reference when config_active_cnt == 0.

CVE ID	Description
CVE-2025-38132	In the Linux kernel, the following vulnerability has been resolved: coresight: holding cscfg_csdev_lock while removing cscfg from csdev There'll be possible race scenario for coresight config: CPU0 CPU1 (perf enable) load module cscfg_load_config_sets() activate config. // sysfs (sys_active_cnt == 1) ... cscfg_csdev_enable_active_config() lock(csdev->cscfg_csdev_lock) deactivate config // sysfs (sys_active_cnt == 0) cscfg_unload_config_sets() <iterating config_csdev_list> cscfg_remove_owned_csdev_configs() // here load config activate by CPU1 unlock(csdev->cscfg_csdev_lock) iterating config_csdev_list could be raced with config_csdev_list's entry delete. To resolve this race , hold csdev->cscfg_csdev_lock() while cscfg_remove_owned_csdev_configs()
CVE-2025-38135	In the Linux kernel, the following vulnerability has been resolved: serial: Fix potential null-ptr-deref in mlb_usio_probe() devm_ioremap() can return NULL on error. Currently, mlb_usio_probe() does not check for this case, which could result in a NULL pointer dereference. Add NULL check after devm_ioremap() to prevent this issue.
CVE-2025-38136	In the Linux kernel, the following vulnerability has been resolved: usb: renesas_usbhs: Reorder clock handling and power management in probe Reorder the initialization sequence in `usbhs_probe()` to enable runtime PM before accessing registers, preventing potential crashes due to uninitialized clocks. Currently, in the probe path, registers are accessed before enabling the clocks, leading to a synchronous external abort on the RZ/V2H SoC. The problematic call flow is as follows: usbhs_probe() usbhs_sys_clock_ctrl() usbhs_bset() usbhs_write() iowrite16() <-- Register access before enabling clocks Since `iowrite16()` is performed without ensuring the required clocks are enabled, this can lead to access errors. To fix this, enable PM runtime early in the probe function and ensure clocks are acquired before register access, preventing crashes like the following on RZ/V2H: [13.272640] Internal error: synchronous external abort: 0000000096000010 [#1] PREEMPT SMP [13.280814] Modules linked in: cec renesas_usbhs(+) drm_kms_helper fuse drm backlight ipv6 [13.289088] CPU: 1 UID: 0 PID: 195 Comm: (udev-worker) Not tainted 6.14.0-rc7+ #98 [13.296640] Hardware name: Renesas RZ/V2H EVK Board based on r9a09g057h44 (DT) [13.303834] pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) [13.310770] pc : usbhs_bset+0x14/0x4c [renesas_usbhs] [13.315831] lr : usbhs_probe+0x2e4/0x5ac [renesas_usbhs] [13.321138] sp : ffff8000827e3850 [13.324438] x29: ffff8000827e3860 x28: 0000000000000000 x27: ffff8000827e3ca0 [13.331554] x26: ffff8000827e3ba0 x25: ffff800081729668 x24: 0000000000000025 [13.338670] x23: ffff0000c0f08000 x22: 0000000000000000 x21: ffff0000c0f08010 [13.345783] x20: 0000000000000000 x19: ffff0000c3b52080 x18: 00000000ffffff [13.352895] x17: 0000000000000000 x16: 0000000000000000 x15: ffff8000827e36ce [13.360009] x14: 000000000000003d7 x13: 000000000000003d7 x12: 0000000000000000 [13.367122] x11: 0000000000000000 x10: 00000000000000aa0 x9: ffff8000827e3750 [13.374235] x8: ffff0000c1850b00 x7: 0000000003826060 x6: 000000000000001c [13.381347] x5: 000000030d5fcc00 x4: ffff8000825c0000 x3: 0000000000000000 [13.388459] x2: 00000000000000400 x1: 0000000000000000 x0: ffff0000c3b52080 [13.395574] Call trace: [13.398013] usbhs_bset+0x14/0x4c [renesas_usbhs] (P) [13.403076] platform_probe+0x68/0xdc [13.406738] really_probe+0xbc/0x2c0 [13.410306] __driver_probe_device+0x78/0x120 [13.414653] driver_probe_device+0x3c/0x154 [13.418825] __driver_attach+0x90/0x1a0 [13.422647] bus_for_each_dev+0x7c/0xe0 [13.426470] driver_attach+0x24/0x30 [13.430032] bus_add_driver+0xe4/0x208 [13.433766] driver_register+0x68/0x130 [13.437587] __platform_driver_register+0x24/0x30 [13.442273] renesas_usbhs_driver_init+0x20/0x1000 [renesas_usbhs] [13.448450] do_one_initcall+0x60/0x1d4 [13.452276] do_init_module+0x54/0x1f8 [13.456014] load_module+0x1754/0x1c98 [13.459750] init_module_from_file+0x88/0xcc [13.464004] __arm64_sys_finit_module+0x1c4/0x328 [13.468689] invoke_syscall+0x48/0x104 [13.472426] el0_svc_common.constprop.0+0xc0/0xe0 [13.477113] do_el0_svc+0x1c/0x28 [13.480415] el0_svc+0x30/0xcc [13.483460] el0t_64_sync_handler+0x10c/0x138 [13.487800] el0t_64_sync+0x198/0x19c [13.491453] Code: 2a0103e1 12003c42 12003c63 8b010084 (79400084) [13.497522] --- [end trace 0000000000000000]---
CVE-2025-38138	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: Add NULL check in udma_probe() devm_kasprintf() returns NULL when memory allocation fails. Currently, udma_probe() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue.

CVE ID	Description
CVE-2025-38140	In the Linux kernel, the following vulnerability has been resolved: dm: limit swapping tables for devices with zone write plugs dm_revalidate_zones() only allowed new or previously unzoned devices to call blk_revalidate_disk_zones(). If the device was already zoned, disk->nr_zones would always equal md->nr_zones, so dm_revalidate_zones() returned without doing any work. This would make the zoned settings for the device not match the new table. If the device had zone write plug resources, it could run into errors like bdev_zone_is_seq() reading invalid memory because disk->conv_zones_bitmap was the wrong size. If the device doesn't have any zone write plug resources, calling blk_revalidate_disk_zones() will always correctly update device. If blk_revalidate_disk_zones() fails, it can still overwrite or clear the current disk->nr_zones value. In this case, DM must restore the previous value of disk->nr_zones, so that the zoned settings will continue to match the previous value that it fell back to. If the device already has zone write plug resources, blk_revalidate_disk_zones() will not correctly update them, if it is called for arbitrary zoned device changes. Since there is not much need for this ability, the easiest solution is to disallow any table reloads that change the zoned settings, for devices that already have zone plug resources. Specifically, if a device already has zone plug resources allocated, it can only switch to another zoned table that also emulates zone append. Also, it cannot change the device size or the zone size. A device can switch to an error target.
CVE-2025-38142	In the Linux kernel, the following vulnerability has been resolved: hwmon: (asus-ec-sensors) check sensor index in read_string() Prevent a potential invalid memory access when the requested sensor is not found. find_ec_sensor_index() may return a negative value (e.g. -ENOENT), but its result was used without checking, which could lead to undefined behavior when passed to get_sensor_info(). Add a proper check to return -EINVAL if sensor_index is negative. Found by Linux Verification Center (linuxtesting.org) with SVACE. [groeck: Return error code returned from find_ec_sensor_index]
CVE-2025-38143	In the Linux kernel, the following vulnerability has been resolved: backlight: pm8941: Add NULL check in wled_configure() devm_kasprintf() returns NULL when memory allocation fails. Currently, wled_configure() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue.
CVE-2025-38145	In the Linux kernel, the following vulnerability has been resolved: soc: aspeed: Add NULL check in aspeed_lpc_enable_snoop() devm_kasprintf() returns NULL when memory allocation fails. Currently, aspeed_lpc_enable_snoop() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue. [arj: Fix Fixes: tag to use subject from 3772e5da4454]
CVE-2025-38146	In the Linux kernel, the following vulnerability has been resolved: net: openvswitch: Fix the dead loop of MPLS parse The unexpected MPLS packet may not end with the bottom label stack. When there are many stacks, The label count value has wrapped around. A dead loop occurs, soft lockup/CPU stuck finally. stack backtrace: UBSAN: array-index-out-of-bounds in /build/linux-0Pa0xK/linux-5.15.0/net/openvswitch/flow.c:662:26 index -1 is out of range for type '__be32 [3]' CPU: 34 PID: 0 Comm: swapper/34 Kdump: loaded Tainted: G OE 5.15.0-121-generic #131-Ubuntu Hardware name: Dell Inc. PowerEdge C6420/OJP9TF, BIOS 2.12.2 07/14/2021 Call Trace: <IRQ> show_stack+0x52/0x5c dump_stack_lvl+0x4a/0x63 dump_stack+0x10/0x16 ubsan_epilogue+0x9/0x36 __ubsan_handle_out_of_bounds.cold+0x44/0x49 key_extract_l3l4+0x82a/0x840 [openvswitch] ? kfree_skbmem+0x52/0xa0 key_extract+0x9c/0x2b0 [openvswitch] ovs_flow_key_extract+0x124/0x350 [openvswitch] ovs_vport_receive+0x61/0xd0 [openvswitch] ? kernel_init_free_pages.part.0+0x4a/0x70 ? get_page_from_freelist+0x353/0x540 netdev_port_receive+0xc4/0x180 [openvswitch] ? netdev_port_receive+0x180/0x180 [openvswitch] netdev_frame_hook+0x1f/0x40 [openvswitch] __netif_receive_skb_core.constprop.0+0x23a/0xf00 __netif_receive_skb_list_core+0xfa/0x240 netif_receive_skb_list_internal+0x18e/0x2a0 napi_complete_done+0x7a/0x1c0 bnxt_poll+0x155/0x1c0 [bnxt_en] __napi_poll+0x30/0x180 net_rx_action+0x126/0x280 ? bnxt_msix+0x67/0x80 [bnxt_en] handle_softirqs+0xda/0x2d0 irq_exit_rcu+0x96/0xc0 common_interrupt+0x8e/0xa0 </IRQ>

CVE ID	Description
CVE-2025-38147	In the Linux kernel, the following vulnerability has been resolved: calipso: Don't call calipso functions for AF_INET sk. syzkaller reported a null-ptr-deref in txopt_get(). [0] The offset 0x70 was of struct ipv6_txoptions in struct ipv6_pininfo, so struct ipv6_pininfo was NULL there. However, this never happens for IPv6 sockets as inet_sk(sk)->pinet6 is always set in inet6_create(), meaning the socket was not IPv6 one. The root cause is missing validation in netlbl_conn_setattr(). netlbl_conn_setattr() switches branches based on struct sockaddr.sa_family, which is passed from userspace. However, netlbl_conn_setattr() does not check if the address family matches the socket. The syzkaller must have called connect() for an IPv6 address on an IPv4 socket. We have a proper validation in tcp_v[46]_connect(), but security_socket_connect() is called in the earlier stage. Let's copy the validation to netlbl_conn_setattr(). [0]: Oops: general protection fault, probably for non-canonical address 0xdffffc000000000e: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000070-0x0000000000000077] CPU: 2 UID: 0 PID: 12928 Comm: syz.9.1677 Not tainted 6.12.0 #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 RIP: 0010:txopt_get include/net/ipv6.h:390 [inline] RIP: 0010: Code: 02 00 00 49 8b ac 24 f8 02 00 00 e8 84 69 2a fd e8 ff 00 16 fd 48 8d 7d 70 48 b8 00 00 00 00 fc ff df 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 53 02 00 00 48 8b 6d 70 48 85 ed 0f 84 ab 01 00 RSP: 0018:ffff88811b8afc48 EFLAGS: 00010212 RAX: dffffc0000000000 RBX: 1ffff11023715f8a RCX: ffffffff841ab00c RDX: 000000000000000e RSI: fffff90007d9e000 RDI: 0000000000000070 RBP: 0000000000000000 R08: ffffd1023715f9d R09: ffffd1023715f9e R10: ffffd1023715f9d R11: 0000000000000003 R12: ffff888123075f00 R13: ffff88810245bd80 R14: ffff888113646780 R15: ffff888100578a80 FS: 00007f9019bd7640(0000) GS:ffff8882d2d00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f901b927bac CR3: 0000000104788003 CR4: 0000000000770ef0 PKRU: 80000000 Call Trace: <TASK> calipso_sock_setattr+0x56/0x80 net/netlabel/netlabel_calipso.c:557 netlbl_conn_setattr+0x10c/0x280 net/netlabel/netlabel_kapi.c:1177 selinux_netlbl_socket_connect_helper+0xd3/0x1b0 security/selinux/netlabel.c:569 selinux_netlbl_socket_connect_locked security/selinux/netlabel.c:597 [inline] selinux_netlbl_socket_connect+0xb6/0x100 security/selinux/netlabel.c:615 selinux_socket_connect+0x5f/0x80 security/selinux/hooks.c:4931 security_socket_connect+0x50/0xa0 security/security.c:4598 __sys_connect_file+0xa4/0x190 net/socket.c:2067 __sys_connect+0x12c/0x170 net/socket.c:2088 __do_sys_connect net/socket.c:2098 [inline] __se_sys_connect net/socket.c:2095 [inline] __x64_sys_connect+0x73/0xb0 net/socket.c:2095 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xaa/0x1b0 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f901b61a12d Code: 02 b8 ff ff ff c3 66 0f 1f 44 00 00 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f9019bd6fa8 EFLAGS: 00000246 ORIG_RAX: 000000000000002a RAX: ffffffff8b1b1b1b RBX: 00007f901b925fa0 RCX: 00007f901b61a12d RDX: 0000000000000001c RSI: 0000200000000140 RDI: 0000000000000003 RBP: 00007f901b701505 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000024 R12: 0000000000000000 R13: 0000000000000000 R14: 00007f901b5b62a0 R15: 00007f9019b7000 </TASK> Modules linked in:
CVE-2025-38148	In the Linux kernel, the following vulnerability has been resolved: net: phy: msc: Fix memory leak when using one step timestamping Fix memory leak when running one-step timestamping. When running one-step sync timestamping, the HW is configured to insert the TX time into the frame, so there is no reason to keep the skb anymore. As in this case the HW will never generate an interrupt to say that the frame was timestamped, then the frame will never released. Fix this by freeing the frame in case of one-step timestamping.
CVE-2025-38149	In the Linux kernel, the following vulnerability has been resolved: net: phy: clear phydev->devlink when the link is deleted There is a potential crash issue when disabling and re-enabling the network port. When disabling the network port, phy_detach() calls device_link_del() to remove the device link, but it does not clear phydev->devlink, so phydev->devlink is not a NULL pointer. Then the network port is re-enabled, but if phy_attach_direct() fails before calling device_link_add(), the code jumps to the "error" label and calls phy_detach(). Since phydev->devlink retains the old value from the previous attach/detach cycle, device_link_del() uses the old value, which accesses a NULL pointer and causes a crash. The simplified crash log is as follows. [24.702421] Call trace: [24.704856] device_link_put_kref+0x20/0x120 [24.709124] device_link_del+0x30/0x48 [24.712864] phy_detach+0x24/0x168 [24.716261] phy_attach_direct+0x168/0x3a4 [24.720352] phylink_fwnode_phy_connect+0xc8/0x14c [24.725140] phylink_of_phy_connect+0x1c/0x34 Therefore, phydev->devlink needs to be cleared when the device link is deleted.

CVE ID	Description
CVE-2025-38152	In the Linux kernel, the following vulnerability has been resolved: remoteproc: core: Clear table_sz when rproc_shutdown There is case as below could trigger kernel dump: Use U-Boot to start remote processor(rproc) with resource table published to a fixed address by rproc. After Kernel boots up, stop the rproc, load a new firmware which doesn't have resource table ,and start rproc. When starting rproc with a firmware not have resource table, `memcpy(loaded_table, rproc->cached_table, rproc->table_sz)` will trigger dump, because rproc->cache_table is set to NULL during the last stop operation, but rproc->table_sz is still valid. This issue is found on i.MX8MP and i.MX9. Dump as below: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtbl: 4k pages, 48-bit VAs, pgdp=000000010af63000 [0000000000000000] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP Modules linked in: CPU: 2 UID: 0 PID: 1060 Comm: sh Not tainted 6.14.0-rc7-next-20250317-dirty #38 Hardware name: NXP i.MX8MPlus EVK board (DT) pstate: a0000005 (NzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPPE=--) pc : __pi_memcpy_generic+0x110/0x22c lr : rproc_start+0x88/0x1e0 Call trace: __pi_memcpy_generic+0x110/0x22c (P) rproc_boot+0x198/0x57c state_store+0x40/0x104 dev_attr_store+0x18/0x2c sysfs_kf_write+0x7c/0x94 kernfs_fop_write_iter+0x120/0x1cc vfs_write+0x240/0x378 ksys_write+0x70/0x108 __arm64_sys_write+0x1c/0x28 invoke_syscall+0x48/0x10c el0_svc_common.constprop.0+0xc0/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x30/0xccc el0t_64_sync_handler+0x10c/0x138 el0t_64_sync+0x198/0x19c Clear rproc->table_sz to address the issue.
CVE-2025-38153	In the Linux kernel, the following vulnerability has been resolved: net: usb: aqc111: fix error handling of usbnet read calls Syzkaller, courtesy of syzbot, identified an error (see report [1]) in aqc111 driver, caused by incomplete sanitation of usb read calls' results. This problem is quite similar to the one fixed in commit 920a9fa27e78 ("net: asix: add proper error handling of usb read errors"). For instance, usbnet_read_cmd() may read fewer than 'size' bytes, even if the caller expected the full amount, and aqc111_read_cmd() will not check its result properly. As [1] shows, this may lead to MAC address in aqc111_bind() being only partly initialized, triggering KMSAN warnings. Fix the issue by verifying that the number of bytes read is as expected and not less. [1] Partial syzbot report: BUG: KMSAN: uninit-value in is_valid_ether_addr include/linux/etherdevice.h:208 [inline] BUG: KMSAN: uninit-value in usbnet_probe+0x2e57/0x4390 drivers/net/usb/usbnet.c:1830 is_valid_ether_addr include/linux/etherdevice.h:208 [inline] usbnet_probe+0x2e57/0x4390 drivers/net/usb/usbnet.c:1830 usb_probe_interface+0xd01/0x1310 drivers/usb/core/driver.c:396 call_driver_probe drivers/base/dd.c:-1 [inline] really_probe+0x4d1/0xd90 drivers/base/dd.c:658 __driver_probe_device+0x268/0x380 drivers/base/dd.c:800 ... Uninit was stored to memory at: dev_addr_mod+0xb0/0x550 net/core/dev_addr_lists.c:582 __dev_addr_set include/linux/netdevice.h:4874 [inline] eth_hw_addr_set include/linux/etherdevice.h:325 [inline] aqc111_bind+0x35f/0x1150 drivers/net/usb/aqc111.c:717 usbnet_probe+0xbe6/0x4390 drivers/net/usb/usbnet.c:1772 usb_probe_interface+0xd01/0x1310 drivers/usb/core/driver.c:396 ... Uninit was stored to memory at: ether_addr_copy include/linux/etherdevice.h:305 [inline] aqc111_read_perm_mac drivers/net/usb/aqc111.c:663 [inline] aqc111_bind+0x794/0x1150 drivers/net/usb/aqc111.c:713 usbnet_probe+0xbe6/0x4390 drivers/net/usb/usbnet.c:1772 usb_probe_interface+0xd01/0x1310 drivers/usb/core/driver.c:396 call_driver_probe drivers/base/dd.c:-1 [inline] ... Local variable buf.i created at: aqc111_read_perm_mac drivers/net/usb/aqc111.c:656 [inline] aqc111_bind+0x221/0x1150 drivers/net/usb/aqc111.c:713 usbnet_probe+0xbe6/0x4390 drivers/net/usb/usbnet.c:1772

CVE ID	Description
CVE-2025-38154	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Avoid using sk_socket after free when sending The sk->sk_socket is not locked or referenced in backlog thread, and during the call to skb_send_sock(), there is a race condition with the release of sk_socket. All types of sockets(tcp/udp/unix/vsock) will be affected. Race conditions: "" CPU0 CPU1 backlog::skb_send_sock sendmsg_unlocked sock_sendmsg sock_sendmsg_nosec close(fd): ... ops->release() -> sock_map_close() sk_socket->ops = NULL free(socket) sock->ops->sendmsg ^ panic here "" The ref of psock become 0 after sock_map_close() executed. "" void sock_map_close() { ... if (likely(psock)) { ... // !! here we remove psock and the ref of psock become 0 sock_map_remove_links(sk, psock) psock = sk_psock_get(sk); if (unlikely(!psock)) goto no_psock; <=== Control jumps here via goto ... cancel_delayed_work_sync(&psock->work); <=== not executed sk_psock_put(sk, psock); ... } "" Based on the fact that we already wait for the workqueue to finish in sock_map_close() if psock is held, we simply increase the psock reference count to avoid race conditions. With this patch, if the backlog thread is running, sock_map_close() will wait for the backlog thread to complete and cancel all pending work. If no backlog running, any pending work that hasn't started by then will fail when invoked by sk_psock_get(), as the psock reference count have been zeroed, and sk_psock_drop() will cancel all jobs via cancel_delayed_work_sync(). In summary, we require synchronization to coordinate the backlog thread and close() thread. The panic I caught: "" Workqueue: events sk_psock_backlog RIP: 0010:sock_sendmsg+0x21d/0x440 RAX: 0000000000000000 RBX: fffff9000521fad8 RCX: 0000000000000001 ... Call Trace: <TASK> ? die_addr+0x40/0xa0 ? exc_general_protection+0x14c/0x230 ? asm_exc_general_protection+0x26/0x30 ? sock_sendmsg+0x21d/0x440 ? sock_sendmsg+0x3e0/0x440 ? __pfx_sock_sendmsg+0x10/0x10 __skb_send_sock+0x543/0xb70 sk_psock_backlog+0x247/0xb80 ... ""
CVE-2025-38155	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: Fix null-ptr-deref in mt7915_mmio_wed_init() devm_ioremap() returns NULL on error. Currently, mt7915_mmio_wed_init() does not check for this case, which results in a NULL pointer dereference. Prevent null pointer dereference in mt7915_mmio_wed_init().
CVE-2025-38156	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: Fix null-ptr-deref in mt7996_mmio_wed_init() devm_ioremap() returns NULL on error. Currently, mt7996_mmio_wed_init() does not check for this case, which results in a NULL pointer dereference. Prevent null pointer dereference in mt7996_mmio_wed_init()
CVE-2025-38157	In the Linux kernel, the following vulnerability has been resolved: wifi: ath9k_htc: Abort software beacon handling if disabled A malicious USB device can send a WMI_SWBA_EVENTID event from an ath9k_htc-managed device before beaconing has been enabled. This causes a device-by-zero error in the driver, leading to either a crash or an out of bounds read. Prevent this by aborting the handling in ath9k_htc_swba() if beacons are not enabled.
CVE-2025-38158	In the Linux kernel, the following vulnerability has been resolved: hisi_acc_vfio_pci: fix XQE dma address error The dma addresses of EQE and AEQE are wrong after migration and results in guest kernel-mode encryption services failure. Comparing the definition of hardware registers, we found that there was an error when the data read from the register was combined into an address. Therefore, the address combination sequence needs to be corrected. Even after fixing the above problem, we still have an issue where the Guest from an old kernel can get migrated to new kernel and may result in wrong data. In order to ensure that the address is correct after migration, if an old magic number is detected, the dma address needs to be updated.
CVE-2025-38159	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw88: fix the 'para' buffer size to avoid reading out of bounds Set the size to 6 instead of 2, since 'para' array is passed to 'rtw_fw_bt_wifi_control(rtwdev, para[0], &para[1])', which reads 5 bytes: void rtw_fw_bt_wifi_control(struct rtw_dev *rtwdev, u8 op_code, u8 *data) { ... SET_BT_WIFI_CONTROL_DATA1(h2c_pkt, *data); SET_BT_WIFI_CONTROL_DATA2(h2c_pkt, *(data + 1)); ... SET_BT_WIFI_CONTROL_DATA5(h2c_pkt, *(data + 4)); Detected using the static analysis tool - Svace.
CVE-2025-38160	In the Linux kernel, the following vulnerability has been resolved: clk: bcm: rpi: Add NULL check in raspberrypi_clk_register() devm_kasprintf() returns NULL when memory allocation fails. Currently, raspberrypi_clk_register() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue.

CVE ID	Description
CVE-2025-38161	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix error flow upon firmware failure for RQ destruction Upon RQ destruction if the firmware command fails which is the last resource to be destroyed some SW resources were already cleaned regardless of the failure. Now properly rollback the object to its original state upon such failure. In order to avoid a use-after free in case someone tries to destroy the object again, which results in the following kernel trace: refcount_t: underflow; use-after-free. WARNING: CPU: 0 PID: 37589 at lib/refcount.c:28 refcount_warn_saturate+0xf4/0x148 Modules linked in: rdma_ucm(OE) rdma_cm(OE) iw_cm(OE) ib_ipoib(OE) ib_cm(OE) ib_umad(OE) mlx5_ib(OE) rfkill mlx5_core(OE) mlxdevm(OE) ib_uverbs(OE) ib_core(OE) psample mlxfw(OE) mlx_compat(OE) macsec tls pci_hyperv_intf sunrpc vfat fat virtio_net net_failover failover fuse loop nfnetlink vsock_loopback vmw_vsock_virtio_transport_common vmw_vsock_vmci_transport vmw_vmci vsock xfs crct10dif_ce ghash_ce sha2_ce sha256_arm64 sha1_ce virtio_console virtio_gpu virtio_blk virtio_dma_buf virtio_mmio dm_mirror dm_region_hash dm_log dm_mod xpmem(OE) CPU: 0 UID: 0 PID: 37589 Comm: python3 Kdump: loaded Tainted: G OE ----- 6.12.0-54.el10.aarch64 #1 Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU KVM Virtual Machine, BIOS 0.0.0 02/06/2015 pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : refcount_warn_saturate+0xf4/0x148 lr : refcount_warn_saturate+0xf4/0x148 sp : ffff80008b81b7e0 x29: ffff80008b81b7e0 x28: ffff000133d51600 x27: 0000000000000001 x26: 0000000000000000 x25: 00000000fffffea x24: ffff00010ae80f00 x23: ffff00010ae80f80 x22: ffff0000c66e5d08 x21: 0000000000000000 x20: ffff0000c66e0000 x19: ffff00010ae80340 x18: 0000000000000006 x17: 0000000000000000 x16: 0000000000000020 x15: ffff80008b81b37f x14: 0000000000000000 x13: 2e656572662d7265 x12: ffff80008283ef78 x11: ffff80008257efd0 x10: ffff80008283efd0 x9: ffff80008021ed90 x8: 0000000000000001 x7: 000000000000bffe x6: c0000000ffff7fff x5: ffff0001fb8e3408 x4: 0000000000000000 x3: ffff800179993000 x2: 0000000000000000 x1: 0000000000000000 x0: ffff000133d51600 Call trace: refcount_warn_saturate+0xf4/0x148 mlx5_core_put_rsc+0x88/0xa0 [mlx5_ib] mlx5_core_destroy_rq_tracked+0x64/0x98 [mlx5_ib] mlx5_ib_destroy_wq+0x34/0x80 [mlx5_ib] ib_destroy_wq_user+0x30/0xc0 [ib_core] uverbs_free_wq+0x28/0x58 [ib_uverbs] destroy_hw_idr_uobject+0x34/0x78 [ib_uverbs] uverbs_destroy_uobject+0x48/0x240 [ib_uverbs] __uverbs_cleanup_ufile+0xd4/0x1a8 [ib_uverbs] uverbs_destroy_ufile_hw+0x48/0x120 [ib_uverbs] ib_uverbs_close+0x2c/0x100 [ib_uverbs] __fput+0xd8/0x2f0 __fput_sync+0x50/0x70 __arm64_sys_close+0x40/0x90 invoke_syscall.constprop.0+0x74/0xd0 do_el0_svc+0x48/0xe8 el0_svc+0x44/0x1d0 el0t_64_sync_handler+0x120/0x130 el0t_64_sync+0x1a4/0x1a8
CVE-2025-38162	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: prevent overflow in lookup table allocation When calculating the lookup table size, ensure the following multiplication does not overflow: - desc->field_len[] maximum value is U8_MAX multiplied by NFT_PIPAPO_GROUPS_PER_BYTE(f) that can be 2, worst case. - NFT_PIPAPO_BUCKETS(f->bb) is 2^8, worst case. - sizeof(unsigned long), from sizeof(*f->lt), lt in struct nft_pipapo_field. Then, use check_mul_overflow() to multiply by bucket size and then use check_add_overflow() to the alignment for avx2 (if needed). Finally, add lt_size_check_overflow() helper and use it to consolidate this. While at it, replace leftover allocation using the GFP_KERNEL to GFP_KERNEL_ACCOUNT for consistency, in pipapo_resize().
CVE-2025-38163	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on sbi->total_valid_block_count syzbot reported a f2fs bug as below: -----[cut here]----- kernel BUG at fs/f2fs/f2fs.h:2521! RIP: 0010:dec_valid_block_count+0x3b2/0x3c0 fs/f2fs/f2fs.h:2521 Call Trace: f2fs_truncate_data_blocks_range+0xc8c/0x11a0 fs/f2fs/file.c:695 truncate_dnode+0x417/0x740 fs/f2fs/node.c:973 truncate_nodes+0x3ec/0xf50 fs/f2fs/node.c:1014 f2fs_truncate_inode_blocks+0x8e3/0x1370 fs/f2fs/node.c:1197 f2fs_do_truncate_blocks+0x840/0x12b0 fs/f2fs/file.c:810 f2fs_truncate_blocks+0x10d/0x300 fs/f2fs/file.c:838 f2fs_truncate+0x417/0x720 fs/f2fs/file.c:888 f2fs_setattr+0xc4f/0x12f0 fs/f2fs/file.c:1112 notify_change+0xbca/0xe90 fs/attr.c:552 do_truncate+0x222/0x310 fs/open.c:65 handle_truncate fs/namei.c:3466 [inline] do_open fs/namei.c:3849 [inline] path_openat+0x2e4f/0x35d0 fs/namei.c:4004 do_filp_open+0x284/0x4e0 fs/namei.c:4031 do_sys_openat2+0x12b/0x1d0 fs/open.c:1429 do_sys_open fs/open.c:1444 [inline] __do_sys_create fs/open.c:1522 [inline] __se_sys_create fs/open.c:1516 [inline] __x64_sys_create+0x124/0x170 fs/open.c:1516 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/syscall_64.c:94 The reason is: in fuzzed image, sbi->total_valid_block_count is inconsistent w/ mapped blocks indexed by inode, so, we should not trigger panic for such case, instead, let's print log and set fsck flag.

CVE ID	Description
CVE-2025-38164	In the Linux kernel, the following vulnerability has been resolved: f2fs: zone: fix to avoid inconsistence in between SIT and SSA w/ below testcase, it will cause inconsistence in between SIT and SSA. <pre>create_null_blk 512 2 1024 1024 mkfs.f2fs -m /dev/nullb0 mount /dev/nullb0 /mnt/f2fs/ touch /mnt/f2fs/file f2fs_io pinfile set /mnt/f2fs/file falloccate -l 4GiB /mnt/f2fs/file F2FS-fs (nullb0): Inconsistent segment (0) type [1, 0] in SSA and SIT CPU: 5 UID: 0 PID: 2398 Comm: falloccate Tainted: G O 6.13.0-rc1 #84 Tainted: [O]=OOT_MODULE Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006 Call Trace: <TASK> dump_stack_lvl+0xb3/0xd0 dump_stack+0x14/0x20 f2fs_handle_critical_error+0x18c/0x220 [f2fs] f2fs_stop_checkpoint+0x38/0x50 [f2fs] do_garbage_collect+0x674/0x6e0 [f2fs] f2fs_gc_range+0x12b/0x230 [f2fs] f2fs_allocate_pinning_section+0x5c/0x150 [f2fs] f2fs_expand_inode_data+0x1cc/0x3c0 [f2fs] f2fs_falloccate+0x3c3/0x410 [f2fs] vfs_falloccate+0x15f/0x4b0 __x64_sys_falloccate+0x4a/0x80 x64_sys_call+0x15e8/0x1b80 do_syscall_64+0x68/0x130 entry_SYSCALL_64_after_hwframe+0x67/0x6f RIP: 0033:0x7f9dba5197ca F2FS-fs (nullb0): Stopped filesystem due to reason: 4 The reason is f2fs_gc_range() may try to migrate block in curseg, however, its SSA block is not uptodate due to the last summary block data is still in cache of curseg. In this patch, we add a condition in f2fs_gc_range() to check whether section is opened or not, and skip block migration for opened section.</pre>
CVE-2025-38165	In the Linux kernel, the following vulnerability has been resolved: bpf, sockmap: Fix panic when calling skb_linearize The panic can be reproduced by executing the command: ./bench sockmap -c 2 -p 1 -a --rx-verdict-ingress --rx-strp 100000 Then a kernel panic was captured: "" [657.460555] kernel BUG at net/core/skbuf.c:2178! [657.462680] Tainted: [W]=WARN [657.463287] Workqueue: events sk_psock_backlog ... [657.469610] <TASK> [657.469738] ? die+0x36/0x90 [657.469916] ? do_trap+0x1d0/0x270 [657.470118] ? pskb_expand_head+0x612/0xf40 [657.470376] ? pskb_expand_head+0x612/0xf40 [657.470620] ? do_error_trap+0xa3/0x170 [657.470846] ? pskb_expand_head+0x612/0xf40 [657.471092] ? handle_invalid_op+0x2c/0x40 [657.471335] ? pskb_expand_head+0x612/0xf40 [657.471579] ? exc_invalid_op+0x2d/0x40 [657.471805] ? asm_exc_invalid_op+0x1a/0x20 [657.472052] ? pskb_expand_head+0xd1/0xf40 [657.472292] ? pskb_expand_head+0x612/0xf40 [657.472540] ? lock_acquire+0x18f/0x4e0 [657.472766] ? find_held_lock+0x2d/0x110 [657.472999] ? __pfx_pskb_expand_head+0x10/0x10 [657.473263] ? __kmalloc_cache_noprof+0x5b/0x470 [657.473537] ? __pfx__lock_release.isra.0+0x10/0x10 [657.473826] __pskb_pull_tail+0xfd/0x1d20 [657.474062] ? __kasan_slab_alloc+0x4e/0x90 [657.474707] sk_psock_skb_ingress_enqueue+0x3bf/0x510 [657.475392] ? __kasan_kmalloc+0xaa/0xb0 [657.476010] sk_psock_backlog+0x5cf/0xd70 [657.476637] process_one_work+0x858/0x1a20 "" The panic originates from the assertion BUG_ON(skb_shared(skb)) in skb_linearize(). A previous commit(see Fixes tag) introduced skb_get() to avoid race conditions between skb operations in the backlog and skb release in the recvmmsg path. However, this caused the panic to always occur when skb_linearize is executed. The "--rx-strp 100000" parameter forces the RX path to use the strparser module which aggregates data until it reaches 100KB before calling sockmap logic. The 100KB payload exceeds MAX_MSG_FRAGS, triggering skb_linearize. To fix this issue, just move skb_get into sk_psock_skb_ingress_enqueue. "" sk_psock_backlog: sk_psock_handle_skb skb_get(skb) <== we move it into 'sk_psock_skb_ingress_enqueue' sk_psock_skb_ingress_____# # sk_psock_skb_ingress_self sk_psock_skb_ingress_enqueue sk_psock_verdict_apply_____# skb_linearize "" Note that for verdict_apply path, the skb_get operation is unnecessary so we add 'take_ref' param to control it's behavior.

CVE ID	Description
CVE-2025-38166	In the Linux kernel, the following vulnerability has been resolved: bpf: fix ktl's panic with sockmap [2172.936997] -----[cut here]----- [2172.936999] kernel BUG at lib/iov_iter.c:629! [2172.944996] PKRU: 55555554 [2172.945155] Call Trace: [2172.945299] <TASK> [2172.945428] ? die+0x36/0x90 [2172.945601] ? do_trap+0xdd/0x100 [2172.945795] ? iov_iter_revert+0x178/0x180 [2172.946031] ? iov_iter_revert+0x178/0x180 [2172.946267] ? do_error_trap+0x7d/0x110 [2172.946499] ? iov_iter_revert+0x178/0x180 [2172.946736] ? exc_invalid_op+0x50/0x70 [2172.946961] ? iov_iter_revert+0x178/0x180 [2172.947197] ? asm_exc_invalid_op+0x1a/0x20 [2172.947446] ? iov_iter_revert+0x178/0x180 [2172.947683] ? iov_iter_revert+0x5c/0x180 [2172.947913] tls_sw_sendmsg_locked.isra.0+0x794/0x840 [2172.948206] tls_sw_sendmsg+0x52/0x80 [2172.948420] ? inet_sendmsg+0x1f/0x70 [2172.948634] __sys_sendto+0x1cd/0x200 [2172.948848] ? find_held_lock+0x2b/0x80 [2172.949072] ? syscall_trace_enter+0x140/0x270 [2172.949330] ? __lock_release.isra.0+0x5e/0x170 [2172.949595] ? find_held_lock+0x2b/0x80 [2172.949817] ? syscall_trace_enter+0x140/0x270 [2172.950211] ? lockdep_hardirqs_on_prepare+0xda/0x190 [2172.950632] ? ktime_get_coarse_real_ts64+0xc2/0xd0 [2172.951036] __x64_sys_sendto+0x24/0x30 [2172.951382] do_syscall_64+0x90/0x170 After calling bpf_exec_tx_verdict(), the size of msg_pl->sg may increase, e.g., when the BPF program executes bpf_msg_push_data(). If the BPF program sets cork_bytes and sg.size is smaller than cork_bytes, it will return -ENOSPC and attempt to roll back to the non-zero copy logic. However, during rollback, msg->msg_iter is reset, but since msg_pl->sg.size has been increased, subsequent executions will exceed the actual size of msg_iter. "" iov_iter_revert(&msg->msg_iter, msg_pl->sg.size - orig_size); "" The changes in this commit are based on the following considerations: 1. When cork_bytes is set, rolling back to non-zero copy logic is pointless and can directly go to zero-copy logic. 2. We can not calculate the correct number of bytes to revert msg_iter. Assume the original data is "abcdefgh" (8 bytes), and after 3 pushes by the BPF program, it becomes 11-byte data: "abc?de?fgh?". Then, we set cork_bytes to 6, which means the first 6 bytes have been processed, and the remaining 5 bytes "?fgh?" will be cached until the length meets the cork_bytes requirement. However, some data in "?fgh?" is not within 'sg->msg_iter' (but in msg_pl instead), especially the data "?" we pushed. So it doesn't seem as simple as just reverting through an offset of msg_iter. 3. For non-TLS sockets in tcp_bpf_sendmsg, when a "cork" situation occurs, the user-space send() doesn't return an error, and the returned length is the same as the input length parameter, even if some data is cached. Additionally, I saw that the current non-zero-copy logic for handling corking is written as: "" line 1177 else if (ret != -EAGAIN) { if (ret == -ENOSPC) ret = 0; goto send_end; "" So it's ok to just return 'copied' without error when a "cork" situation occurs.
CVE-2025-38167	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: handle hdr_first_de() return value The hdr_first_de() function returns a pointer to a struct NTFS_DE. This pointer may be NULL. To handle the NULL error effectively, it is important to implement an error handler. This will help manage potential errors consistently. Additionally, error handling for the return value already exists at other points where this function is called. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-38168	In the Linux kernel, the following vulnerability has been resolved: perf: arm-ni: Unregister PMUs on probe failure When a resource allocation fails in one clock domain of an NI device, we need to properly roll back all previously registered perf PMUs in other clock domains of the same device. Otherwise, it can lead to kernel panics. Calling arm_ni_init+0x0/0xff8 [arm_ni] @ 2374 arm-ni ARMHCB70:00: Failed to request PMU region 0x1f3c13000 arm-ni ARMHCB70:00: probe with driver arm-ni failed with error -16 list_add corruption: next->prev should be prev (ffffd01e9698a18), but was 0000000000000000. (next=ffff10001a0decc8). pstate: 6340009 (nZCv daif +PAN -UAO +TCO +DIT -SSBS BTYPE=--) pc : list_add_valid_or_report+0x7c/0xb8 lr : list_add_valid_or_report+0x7c/0xb8 Call trace: __list_add_valid_or_report+0x7c/0xb8 perf_pmu_register+0x22c/0x3a0 arm_ni_probe+0x554/0x70c [arm_ni] platform_probe+0x70/0xe8 really_probe+0xc6/0x4d8 driver_probe_device+0x48/0x170 __driver_attach+0x8e/0x1c0 bus_for_each_dev+0x64/0xf0 driver_add+0x138/0x260 bus_add_driver+0x68/0x138 __platform_driver_register+0x2c/0x40 arm_ni_init+0x14/0x2a [arm_ni] do_init_module+0x36/0x298 ---[end trace 0000000000000000]--- Kernel panic - not syncing: Oops - BUG: Fatal exception SMP: stopping secondary CPUs
CVE-2025-38169	In the Linux kernel, the following vulnerability has been resolved: arm64/fpsimd: Avoid clobbering kernel FPSIMD state with SMSTOP On system with SME, a thread's kernel FPSIMD state may be erroneously clobbered during a context switch immediately after that state is restored. Systems without SME are unaffected. If the CPU happens to be in streaming SVE mode before a context switch to a thread with kernel FPSIMD state, fpsimd_thread_switch() will restore the kernel FPSIMD state using fpsimd_load_kernel_state() while the CPU is still in streaming SVE mode. When fpsimd_thread_switch() subsequently calls fpsimd_flush_cpu_state(), this will execute an SMSTOP, causing an exit from streaming SVE mode. The exit from streaming SVE mode will cause the hardware to reset a number of FPSIMD/SVE/SME registers, clobbering the FPSIMD state. Fix this by calling fpsimd_flush_cpu_state() before restoring the kernel FPSIMD state.

CVE ID	Description
CVE-2025-38170	In the Linux kernel, the following vulnerability has been resolved: arm64/fpsimd: Discard stale CPU state when handling SME traps The logic for handling SME traps manipulates saved FPSIMD/SVE/SME state incorrectly, and a race with preemption can result in a task having TIF_SME set and TIF_FOREIGN_FPSTATE clear even though the live CPU state is stale (e.g. with SME traps enabled). This can result in warnings from do_sme_acc() where SME traps are not expected while TIF_SME is set: /* With TIF_SME userspace shouldn't generate any traps */ if (test_and_set_thread_flag(TIF_SME)) WARN_ON(1); This is very similar to the SVE issue we fixed in commit: 751ecf6afd6568ad ("arm64/sve: Discard stale CPU state when handling SVE traps") The race can occur when the SME trap handler is preempted before and after manipulating the saved FPSIMD/SVE/SME state, starting and ending on the same CPU, e.g. void do_sme_acc(unsigned long esr, struct pt_regs *regs) { // Trap on CPU 0 with TIF_SME clear, SME traps enabled // task->fpsimd_cpu is 0. // per_cpu_ptr(&fpsimd_last_state, 0) is task. ... // Preempted; migrated from CPU 0 to CPU 1. // TIF_FOREIGN_FPSTATE is set. get_cpu_fpsimd_context(); /* With TIF_SME userspace shouldn't generate any traps */ if (test_and_set_thread_flag(TIF_SME)) WARN_ON(1); if (!test_thread_flag(TIF_FOREIGN_FPSTATE)) { unsigned long vq_minus_one = sve_vq_from_vl(task_get_sme_vl(current)) - 1; sme_set_vq(vq_minus_one); fpsimd_bind_task_to_cpu(); } put_cpu_fpsimd_context(); // Preempted; migrated from CPU 1 to CPU 0. // task->fpsimd_cpu is still 0 // If per_cpu_ptr(&fpsimd_last_state, 0) is still task then: // - Stale HW state is reused (with SME traps enabled) // - TIF_FOREIGN_FPSTATE is cleared // - A return to userspace skips HW state restore } Fix the case where the state is not live and TIF_FOREIGN_FPSTATE is set by calling fpsimd_flush_task_state() to detach from the saved CPU state. This ensures that a subsequent context switch will not reuse the stale CPU state, and will instead set TIF_FOREIGN_FPSTATE, forcing the new state to be reloaded from memory prior to a return to userspace. Note: this was originally posted as [1]. [Rutland: rewrite commit message]
CVE-2025-38173	In the Linux kernel, the following vulnerability has been resolved: crypto: marvell/cesa - Handle zero-length skcipher requests Do not access random memory for zero-length skcipher requests. Just return 0.
CVE-2025-38174	In the Linux kernel, the following vulnerability has been resolved: thunderbolt: Do not double dequeue a configuration request Some of our devices crash in tb_cfg_request_dequeue(): general protection fault, probably for non-canonical address 0xdead000000000122 CPU: 6 PID: 91007 Comm: kworker/6:2 Tainted: G U W 6.6.65 RIP: 0010:tb_cfg_request_dequeue+0x2d/0xa0 Call Trace: <TASK> ? tb_cfg_request_dequeue+0x2d/0xa0 tb_cfg_request_work+0x33/0x80 worker_thread+0x386/0x8f0 kthread+0xed/0x110 ret_from_fork+0x38/0x50 ret_from_fork_asm+0x1b/0x30 The circumstances are unclear, however, the theory is that tb_cfg_request_work() can be scheduled twice for a request: first time via frame.callback from ring_work() and second time from tb_cfg_request(). Both times kworkers will execute tb_cfg_request_dequeue(), which results in double list_del() from the ctl->request_queue (the list poison deference hints at it: 0xdead000000000122). Do not dequeue requests that don't have TB_CFG_REQUEST_ACTIVE bit set.
CVE-2025-38177	In the Linux kernel, the following vulnerability has been resolved: sch_hfsc: make hfsc_qlen_notify() idempotent hfsc_qlen_notify() is not idempotent either and not friendly to its callers, like fq_codelet_dequeue(). Let's make it idempotent to ease qdisc_tree_reduce_backlog() callers' life: 1. update_vf() decreases cl->cl_nactive, so we can check whether it is non-zero before calling it. 2. eltree_remove() always removes RB node cl->el_node, but we can use RB_EMPTY_NODE() + RB_CLEAR_NODE() to make it safe.
CVE-2025-38180	In the Linux kernel, the following vulnerability has been resolved: net: atm: fix /proc/net/atm/lec handling /proc/net/atm/lec must ensure safety against dev_lec[] changes. It appears it had dev_put() calls without prior dev_hold(), leading to imbalance and UAF.

CVE ID	Description
CVE-2025-38181	In the Linux kernel, the following vulnerability has been resolved: calipso: Fix null-ptr-deref in calipso_req_{set,del}attr(). syzkaller reported a null-ptr-deref in sock_omalloc() while allocating a CALIPSO option. [0] The NULL is of struct sock, which was fetched by sk_to_full_sk() in calipso_req_setattr(). Since commit a1a5344db8 ("tcp: avoid two atomic ops for syncookies"), reqsk->rsk_listener could be NULL when SYN Cookie is returned to its client, as hinted by the leading SYN Cookie log. Here are 3 options to fix the bug: 1) Return 0 in calipso_req_setattr() 2) Return an error in calipso_req_setattr() 3) Always set rsk_listener 1) is no go as it bypasses LSM, but 2) effectively disables SYN Cookie for CALIPSO. 3) is also no go as there have been many efforts to reduce atomic ops and make TCP robust against DDoS. See also commit 3b24d854cb35 ("tcp/dccp: do not touch listener sk_refcnt under synflood"). As of the blamed commit, SYN Cookie already did not need refcounting, and no one has stumbled on the bug for 9 years, so no CALIPSO user will care about SYN Cookie. Let's return an error in calipso_req_setattr() and calipso_req_delattr() in the SYN Cookie case. This can be reproduced by [1] on Fedora and now connect() of nc times out. [0]: TCP: request_sock_TCPv6: Possible SYN flooding on port [::]:20002. Sending cookies. Oops: general protection fault, probably for non-canonical address 0xdffffc0000000006: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000030-0x0000000000000037] CPU: 3 UID: 0 PID: 12262 Comm: syz.1.2611 Not tainted 6.14.0 #2 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 RIP: 0010:read_pnet include/net/net_namespace.h:406 [inline] RIP: 0010:sock_net include/net/sock.h:655 [inline] RIP: 0010:sock_kmalloc+0x35/0x170 net/core/sock.c:2806 Code: 89 d5 41 54 55 89 f5 53 48 89 fb e8 25 e3 c6 fd e8 f0 91 e3 00 48 8d 7b 30 48 b8 00 00 00 00 ff df 48 89 fa 48 c1 ea 03 <80> 3c 02 00 0f 85 26 01 00 00 48 b8 00 00 00 00 fc ff df 4c 8b RSP: 0018:ffff88811af89038 EFLAGS: 00010216 RAX: dffffc0000000000 RBX: 0000000000000000 RCX: ffff888105266400 RDX: 0000000000000006 RSI: ffff88800c890000 RDI: 0000000000000030 RBP: 0000000000000050 R08: 0000000000000000 R09: ffff88810526640e R10: ffffd1020a4cc81 R11: ffff88810526640f R12: 0000000000000000 R13: 00000000000000820 R14: ffff888105266400 R15: 00000000000000050 FS: 00007f0653a07640(0000) GS:ffff88811af80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f863ba096f4 CR3: 00000000163c0005 CR4: 0000000000770ef0 PKRU: 80000000 Call Trace: <IRQ> ipv6_renew_options+0x279/0x950 net/ipv6/exthdrs.c:1288 calipso_req_setattr+0x181/0x340 net/ipv6/calipso.c:1204 calipso_req_setattr+0x56/0x80 net/netlabel/netlabel_calipso.c:597 netlbl_req_setattr+0x18a/0x440 net/netlabel/netlabel_kapi.c:1249 selinux_netlbl_inet_conn_request+0x1fb/0x320 security/selinux/netlabel.c:342 selinux_inet_conn_request+0x1eb/0x2c0 security/selinux/hooks.c:5551 security_inet_conn_request+0x50/0xa0 security/security.c:4945 tcp_v6_route_req+0x22c/0x550 net/ipv6/tcp_ipv6.c:825 tcp_conn_request+0xec8/0x2b70 net/ipv4/tcp_input.c:7275 tcp_v6_conn_request+0x1e3/0x440 net/ipv6/tcp_ipv6.c:1328 tcp_rcv_state_process+0xaf/0x52b0 net/ipv4/tcp_input.c:6781 tcp_v6_do_rcv+0x8a6/0x1a40 net/ipv6/tcp_ipv6.c:1667 tcp_v6_rcv+0x505e/0x5b50 net/ipv6/tcp_ipv6.c:1904 ip6_protocol_deliver_rcu+0x17c/0x1da0 net/ipv6/ip6_input.c:436 ip6_input_finish+0x103/0x180 net/ipv6/ip6_input.c:480 NF_HOOK include/linux/netfilter.h:314 [inline] NF_HOOK include/linux/netfilter.h:308 [inline] ip6_input+0x13c/0x6b0 net/ipv6/ip6_input.c:491 dst_input include/net/dst.h:469 [inline] ip6_rcv_finish net/ipv6/ip6_input.c:79 [inline] ip6_rcv_finish+0xb6/0x490 net/ipv6/ip6_input.c:69 NF_HOOK include/linux/netfilter.h:314 [inline] NF_HOOK include/linux/netf --- truncated---
CVE-2025-38182	In the Linux kernel, the following vulnerability has been resolved: ublk: sanitize the arguments from userspace when adding a device Sanity check the values for queue depth and number of queues we get from userspace when adding a device.
CVE-2025-38183	In the Linux kernel, the following vulnerability has been resolved: net: lan743x: fix potential out-of-bounds write in lan743x_ptp_io_event_clock_get() Before calling lan743x_ptp_io_event_clock_get(), the 'channel' value is checked against the maximum value of PCI11X1X_PTP_IO_MAX_CHANNELS(8). This seems correct and aligns with the PTP interrupt status register (PTP_INT_STS) specifications. However, lan743x_ptp_io_event_clock_get() writes to ptp->extts[] with only LAN743X_PTP_N_EXTTS(4) elements, using channel as an index: lan743x_ptp_io_event_clock_get(..., u8 channel,...) { ... /* Update Local timestamp */ extts = &ptp->extts[channel]; extts->ts.tv_sec = sec; ... } To avoid an out-of-bounds write and utilize all the supported GPIO inputs, set LAN743X_PTP_N_EXTTS to 8. Detected using the static analysis tool - Svace.
CVE-2025-38184	In the Linux kernel, the following vulnerability has been resolved: tipc: fix null-ptr-deref when acquiring remote ip of ethernet bearer The reproduction steps: 1. create a tun interface 2. enable l2 bearer 3. TIPC_NL_UDP_GET_REMOTEIP with media name set to tun tipc: Started in network mode tipc: Node identity 8af312d38a21, cluster identity 4711 tipc: Enabled bearer <eth:syz_tun>, priority 1 Oops: general protection fault KASAN: null-ptr-deref in range CPU: 1 UID: 1000 PID: 559 Comm: poc Not tainted 6.16.0-rc1+ #117 PREEMPT Hardware name: QEMU Ubuntu 24.04 PC RIP: 0010:tipc_udp_nl_dump_remoteip+0x4a4/0x8f0 the ub was in fact a struct dev. when bid != 0 && skip_cnt != 0, bearer_list[bid] may be NULL or other media when other thread changes it. fix this by checking media_id.

CVE ID	Description
CVE-2025-38185	In the Linux kernel, the following vulnerability has been resolved: atm: atmtcp: Free invalid length skb in atmtcp_c_send(). syzbot reported the splat below. [0] vcc_sendmsg() copies data passed from userspace to skb and passes it to vcc->dev->ops->send(). atmtcp_c_send() accesses skb->data as struct atmtcp_hdr after checking if skb->len is 0, but it's not enough. Also, when skb->len == 0, skb and sk (vcc) were leaked because dev_kfree_skb() is not called and sk_wmem_alloc adjustment is missing to revert atm_account_tx() in vcc_sendmsg(), which is expected to be done in atm_pop_raw(). Let's properly free skb with an invalid length in atmtcp_c_send(). [0]: BUG: KMSAN: uninit-value in atmtcp_c_send+0x255/0xed0 drivers/atm/atmtcp.c:294 atmtcp_c_send+0x255/0xed0 drivers/atm/atmtcp.c:294 vcc_sendmsg+0xd7c/0xff0 net/atm/common.c:644 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg+0x330/0x3d0 net/socket.c:727 ____sys_sendmsg+0x7e0/0xd80 net/socket.c:2566 __sys_sendmsg+0x271/0x3b0 net/socket.c:2620 __sys_sendmsg net/socket.c:2652 [inline] __do_sys_sendmsg net/socket.c:2657 [inline] __se_sys_sendmsg net/socket.c:2655 [inline] __x64_sys_sendmsg+0x211/0x3e0 net/socket.c:2655 x64_sys_call+0x32fb/0x3db0 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xd9/0x210 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f Uninit was created at: slab_post_alloc_hook mm/slub.c:4154 [inline] slab_alloc_node mm/slub.c:4197 [inline] kmem_cache_alloc_node_noprof +0x818/0xf00 mm/slub.c:4249 kmalloc_reserve+0x13c/0x4b0 net/core/skbuff.c:579 __alloc_skb +0x347/0x7d0 net/core/skbuff.c:670 alloc_skb include/linux/skbuff.h:1336 [inline] vcc_sendmsg +0xb40/0xff0 net/atm/common.c:628 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg +0x330/0x3d0 net/socket.c:727 ____sys_sendmsg+0x7e0/0xd80 net/socket.c:2566 __sys_sendmsg +0x271/0x3b0 net/socket.c:2620 __sys_sendmsg net/socket.c:2652 [inline] __do_sys_sendmsg net/socket.c:2657 [inline] __se_sys_sendmsg net/socket.c:2655 [inline] __x64_sys_sendmsg+0x211/0x3e0 net/socket.c:2655 x64_sys_call+0x32fb/0x3db0 arch/x86/include/generated/asm/syscalls_64.h:47 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xd9/0x210 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f CPU: 1 UID: 0 PID: 5798 Comm: syz-executor192 Not tainted 6.16.0-rc1-syzkaller-00010-g2c4a1f3fe03e #0 PREEMPT(undef) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025
CVE-2025-38187	In the Linux kernel, the following vulnerability has been resolved: drm/nouveau: fix a use-after-free in r535_gsp_rpc_push() The RPC container is released after being passed to r535_gsp_rpc_send(). When sending the initial fragment of a large RPC and passing the caller's RPC container, the container will be freed prematurely. Subsequent attempts to send remaining fragments will therefore result in a use-after-free. Allocate a temporary RPC container for holding the initial fragment of a large RPC when sending. Free the caller's container when all fragments are successfully sent. [Rebase onto Blackwell changes. - Danilo]
CVE-2025-38188	In the Linux kernel, the following vulnerability has been resolved: drm/msm/a7xx: Call CP_RESET_CONTEXT_STATE Calling this packet is necessary when we switch contexts because there are various pieces of state used by userspace to synchronize between BR and BV that are persistent across submits and we need to make sure that they are in a "safe" state when switching contexts. Otherwise a userspace submission in one context could cause another context to function incorrectly and hang, effectively a denial of service (although without leaking data). This was missed during initial a7xx bringup. Patchwork: https://patchwork.freedesktop.org/patch/654924/

CVE ID	Description
CVE-2025-38189	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Avoid NULL pointer dereference in `v3d_job_update_stats()` The following kernel Oops was recently reported by Mesa CI: [800.139824] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000588 [800.148619] Mem abort info: [800.151402] ESR = 0x0000000096000005 [800.155141] EC = 0x25: DABT (current EL), IL = 32 bits [800.160444] SET = 0, FnV = 0 [800.163488] EA = 0, S1PTW = 0 [800.166619] FSC = 0x05: level 1 translation fault [800.171487] Data abort info: [800.174357] ISV = 0, ISS = 0x00000005, ISS2 = 0x00000000 [800.179832] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [800.184873] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [800.190176] user pgtable: 4k pages, 39-bit VAs, pgdp=00000001014c2000 [800.196607] [0000000000000588] pgd=0000000000000000, p4d=0000000000000000, pud=0000000000000000 [800.205305] Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP [800.211564] Modules linked in: vc4 snd_soc_hdmi_codec drm_display_helper v3d cec gpu_sched drm_dma_helper drm_shmem_helper drm_kms_helper drm drm_panel_orientation_quirks snd_soc_core snd_compress snd_pcm_dmaengine snd_pcm i2c_brcmstb snd_timer snd backlight [800.234448] CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.25+rpt-rpi-v8 #1 Debian 1:6.12.25-1+rpt1 [800.244182] Hardware name: Raspberry Pi 4 Model B Rev 1.4 (DT) [800.250005] pstate: 60000c5 (nZCv daIF -PAN -UAO -TCO -DIT -SSBS BTYP=) [800.256959] pc : v3d_job_update_stats+0x60/0x130 [v3d] [800.262112] lr : v3d_job_update_stats+0x48/0x130 [v3d] [800.267251] sp : ffffffffc080003e60 [800.270555] x29: ffffffffc080003e60 x28: ffffffffd842784980 x27: 0224012000000000 [800.277687] x26: ffffffffd84277f630 x25: ffffffffd81012fd800 x24: 0000000000000020 [800.284818] x23: ffffffffd8040238b08 x22: 00000000000000570 x21: 00000000000000158 [800.291948] x20: 0000000000000000 x19: ffffffffd8040238000 x18: 0000000000000000 [800.299078] x17: ffffffffa8c1bd2000 x16: ffffffffc080000000 x15: 0000000000000000 [800.306208] x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 [800.313338] x11: 0000000000000040 x10: 0000000000001a40 x9 : ffffffffd83b39757c [800.320468] x8 : ffffffffd842786420 x7 : 7fffffff00000000 x6 : 0000000000ef32b0 [800.327598] x5 : 00ffffff00000000 x4 : 0000000000000015 x3 : ffffffffd842784980 [800.334728] x2 : 0000000000000004 x1 : 0000000000010002 x0 : 000000ba4c0ca382 [800.341859] Call trace: [800.344294] v3d_job_update_stats+0x60/0x130 [v3d] [800.349086] v3d_irq+0x124/0x2e0 [v3d] [800.352835] __handle_irq_event_percpu+0x58/0x218 [800.357539] handle_irq_event+0x54/0xb8 [800.361369] handle_fastoi_irq+0xac/0x240 [800.365458] handle_irq_desc+0x48/0x68 [800.369200] generic_handle_domain_irq+0x24/0x38 [800.373810] gic_handle_irq+0x48/0xd8 [800.377464] call_on_irq_stack+0x24/0x58 [800.381379] do_interrupt_handler+0x88/0x98 [800.385554] el1_interrupt+0x34/0x68 [800.389123] el1h_64_irq_handler+0x18/0x28 [800.393211] el1h_64_irq+0x64/0x68 [800.396603] default_idle_call+0x3c/0x168 [800.400606] do_idle+0x1fc/0x230 [800.403827] cpu_startup_entry+0x40/0x50 [800.407742] rest_init+0xe4/0xf0 [800.410962] start_kernel+0x5e8/0x790 [800.414616] __primary_switched+0x80/0x90 [800.418622] Code: 8b170277 8b160296 11000421 b9000861 (b9401ac1) [800.424707] ---[end trace 0000000000000000]--- [800.457313] ---[end Kernel panic - not syncing: Oops: Fatal exception in interrupt]--- This issue happens when the file descriptor is closed before the jobs submitted by it are completed. When the job completes, we update the global GPU stats and the per-fd GPU stats, which are exposed through fdinfo. If the file descriptor was closed, then the struct `v3d_file_priv` and its stats were already freed and we can't update the per-fd stats. Therefore, if the file descriptor was already closed, don't u ---truncated---
CVE-2025-38190	In the Linux kernel, the following vulnerability has been resolved: atm: Revert atm_account_tx() if copy_from_iter_full() fails. In vcc_sendmsg(), we account skb->truesize to sk->sk_wmem_alloc by atm_account_tx(). It is expected to be reverted by atm_pop_raw() later called by vcc->dev->ops->send(vcc, skb). However, vcc_sendmsg() misses the same revert when copy_from_iter_full() fails, and then we will leak a socket. Let's factorise the revert part as atm_return_tx() and call it in the failure path. Note that the corresponding sk_wmem_alloc operation can be found in alloc_tx() as of the blamed commit. \$ git blame -L:alloc_tx net/atm/common.c c55fa3ccbc2c~
CVE-2025-38191	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix null pointer dereference in destroy_previous_session If client set ->PreviousSessionId on kerberos session setup stage, NULL pointer dereference error will happen. Since sess->user is not set yet, It can pass the user argument as NULL to destroy_previous_session. sess->user will be set in ksmbd_krb5_authenticate(). So this patch move calling destroy_previous_session() after ksmbd_krb5_authenticate().

CVE ID	Description
CVE-2025-38192	In the Linux kernel, the following vulnerability has been resolved: net: clear the dst when changing skb protocol A not-so-careful NAT46 BPF program can crash the kernel if it indiscriminately flips ingress packets from v4 to v6: BUG: kernel NULL pointer dereference, address: 0000000000000000 ip6_rcv_core (net/ipv6/ip6_input.c:190:20) ipv6_rcv (net/ipv6/ip6_input.c:306:8) process_backlog (net/core/dev.c:6186:4) napi_poll (net/core/dev.c:6906:9) net_rx_action (net/core/dev.c:7028:13) do_softirq (kernel/softirq.c:462:3) netif_rx (net/core/dev.c:5326:3) dev_loopback_xmit (net/core/dev.c:4015:2) ip_mc_finish_output (net/ipv4/ip_output.c:363:8) NF_HOOK (/include/linux/netfilter.h:314:9) ip_mc_output (net/ipv4/ip_output.c:400:5) dst_output (/include/net/dst.h:459:9) ip_local_out (net/ipv4/ip_output.c:130:9) ip_send_skb (net/ipv4/ip_output.c:1496:8) udp_send_skb (net/ipv4/udp.c:1040:8) udp_sendmsg (net/ipv4/udp.c:1328:10) The output interface has a 4->6 program attached at ingress. We try to loop the multicast skb back to the sending socket. Ingress BPF runs as part of netif_rx(), pushes a valid v6 hdr and changes skb->protocol to v6. We enter ip6_rcv_core which tries to use skb_dst(). But the dst is still an IPv4 one left after IPv4 mcast output. Clear the dst in all BPF helpers which change the protocol. Try to preserve metadata dsts, those may carry non-routing metadata.
CVE-2025-38193	In the Linux kernel, the following vulnerability has been resolved: net_sched: sch_sfq: reject invalid perturb period Gerrard Tai reported that SFQ perturb_period has no range check yet, and this can be used to trigger a race condition fixed in a separate patch. We want to make sure ctl->perturb_period * HZ will not overflow and is positive. tc qd add dev lo root sfq perturb -10 # negative value : error Error: sch_sfq: invalid perturb period. tc qd add dev lo root sfq perturb 1000000000 # too big : error Error: sch_sfq: invalid perturb period. tc qd add dev lo root sfq perturb 2000000 # acceptable value tc -s -d qd sh dev lo qdisc sfq 8005: root refcnt 2 limit 127p quantum 64Kb depth 127 flows 128 divisor 1024 perturb 2000000sec Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0) backlog 0b 0p requeues 0
CVE-2025-38194	In the Linux kernel, the following vulnerability has been resolved: jffs2: check that raw node were preallocated before writing summary Syzkaller detected a kernel bug in jffs2_link_node_ref, caused by fault injection in jffs2_prealloc_raw_node_refs. jffs2_sum_write_sumnode doesn't check return value of jffs2_prealloc_raw_node_refs and simply lets any error propagate into jffs2_sum_write_data, which eventually calls jffs2_link_node_ref in order to link the summary to an expectedly allocated node. kernel BUG at fs/jffs2/nodelist.c:592! invalid opcode: 0000 [#1] PREEMPT SMP KASAN NOPTI CPU: 1 PID: 31277 Comm: syz-executor.7 Not tainted 6.1.128-syzkaller-00139-ge10f83ca10a1 #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 RIP: 0010:jffs2_link_node_ref+0x570/0x690 fs/jffs2/nodelist.c:592 Call Trace: <TASK> jffs2_sum_write_data fs/jffs2/summary.c:841 [inline] jffs2_sum_write_sumnode+0xd1a/0x1da0 fs/jffs2/summary.c:874 jffs2_do_reserve_space+0xa18/0xd60 fs/jffs2/nodemgmt.c:388 jffs2_reserve_space+0x55f/0xaa0 fs/jffs2/nodemgmt.c:197 jffs2_write_inode_range+0x246/0xb50 fs/jffs2/write.c:362 jffs2_write_end+0x726/0x15d0 fs/jffs2/file.c:301 generic_perform_write+0x314/0x5d0 mm/filemap.c:3856 __generic_file_write_iter+0x2ae/0x4d0 mm/filemap.c:3973 generic_file_write_iter+0xe3/0x350 mm/filemap.c:4005 call_write_iter include/linux/fs.h:2265 [inline] do_iter_readv_writev+0x20f/0x3c0 fs/read_write.c:735 do_iter_write+0x186/0x710 fs/read_write.c:861 vfs_iter_write+0x70/0xa0 fs/read_write.c:902 iter_file_splice_write+0x73b/0xc90 fs/splice.c:685 do_splice_from fs/splice.c:763 [inline] direct_splice_actor+0x10c/0x170 fs/splice.c:950 splice_direct_to_actor+0x337/0xa10 fs/splice.c:896 do_splice_direct+0x1a9/0x280 fs/splice.c:1002 do_sendfile+0xb13/0x12c0 fs/read_write.c:1255 __do_sys_sendfile64 fs/read_write.c:1323 [inline] __se_sys_sendfile64 fs/read_write.c:1309 [inline] __x64_sys_sendfile64+0x1cf/0x210 fs/read_write.c:1309 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x35/0x80 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x6e/0xd8 Fix this issue by checking return value of jffs2_prealloc_raw_node_refs before calling jffs2_sum_write_data. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-38197	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell_rbu: Fix list usage Pass the correct list head to list_for_each_entry*() when looping through the packet list. Without this patch, reading the packet data via sysfs will show the data incorrectly (because it starts at the wrong packet), and clearing the packet list will result in a NULL pointer dereference.
CVE-2025-38198	In the Linux kernel, the following vulnerability has been resolved: fbcon: Make sure modelist not set on unregistered console It looks like attempting to write to the "store_modes" sysfs node will run afoul of unregistered consoles: UBSAN: array-index-out-of-bounds in drivers/video/fbdev/core/fbcon.c:122:28 index -1 is out of range for type 'fb_info *[32]' ... fbcon_info_from_console+0x192/0x1a0 drivers/video/fbdev/core/fbcon.c:122 fbcon_new_modelist+0xbf/0x2d0 drivers/video/fbdev/core/fbcon.c:3048 fb_new_modelist+0x328/0x440 drivers/video/fbdev/core/fbmem.c:673 store_modes+0x1c9/0x3e0 drivers/video/fbdev/core/fbcon.c:113 dev_attr_store+0x55/0x80 drivers/base/core.c:2439 static struct fb_info *fbcon_registered_fb[FB_MAX]; ... static signed char con2fb_map[MAX_NR_CONSOLES]; ... static struct fb_info *fbcon_info_from_console(int console) ... return fbcon_registered_fb[con2fb_map[console]]; If con2fb_map contains a -1 things go wrong here. Instead, return NULL, as callers of fbcon_info_from_console() are trying to compare against existing "info" pointers, so error handling should kick in correctly.

CVE ID	Description
CVE-2025-38199	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Fix memory leak due to multiple rx_stats allocation rx_stats for each arsta is allocated when adding a station. arsta->rx_stats will be freed when a station is removed. Redundant allocations are occurring when the same station is added multiple times. This causes ath12k_mac_station_add() to be called multiple times, and rx_stats is allocated each time. As a result there is memory leaks. Prevent multiple allocations of rx_stats when ath12k_mac_station_add() is called repeatedly by checking if rx_stats is already allocated before allocating again. Allocate arsta->rx_stats if arsta->rx_stats is NULL respectively. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.3.1-00173-QCAHKSUWPL_SILICONZ-1 Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.0.c5-00481-QCAHMTSWPL_V1.0_V2.0_SILICONZ-3
CVE-2025-38200	In the Linux kernel, the following vulnerability has been resolved: i40e: fix MMIO write access to an invalid page in i40e_clear_hw When the device sends a specific input, an integer underflow can occur, leading to MMIO write access to an invalid page. Prevent the integer underflow by changing the type of related variables.
CVE-2025-38201	In the Linux kernel, the following vulnerability has been resolved: netfilter: nft_set_pipapo: clamp maximum map bucket size to INT_MAX Otherwise, it is possible to hit WARN_ON_ONCE in __kvmalloc_node_noprof() when resizing hashtable because __GFP_NOWARN is unset. Similar to: b541ba7d1f5a ("netfilter: conntrack: clamp maximum hashtable size to INT_MAX")
CVE-2025-38202	In the Linux kernel, the following vulnerability has been resolved: bpf: Check rcu_read_lock_trace_held() in bpf_map_lookup_percpu_elem() bpf_map_lookup_percpu_elem() helper is also available for sleepable bpf program. When BPF JIT is disabled or under 32-bit host, bpf_map_lookup_percpu_elem() will not be inlined. Using it in a sleepable bpf program will trigger the warning in bpf_map_lookup_percpu_elem(), because the bpf program only holds rcu_read_lock_trace lock. Therefore, add the missed check.

CVE ID	Description
CVE-2025-38203	In the Linux kernel, the following vulnerability has been resolved: jfs: Fix null-ptr-deref in jfs_ioc_trim [Syzkaller Report] Oops: general protection fault, probably for non-canonical address 0xdffffc00000000087: 0000 [#1 KASAN: null-ptr-deref in range [0x0000000000000438-0x000000000000043f] CPU: 2 UID: 0 PID: 10614 Comm: syz-executor.0 Not tainted 6.13.0-rc6-gfbfd64d25c7a-dirty #1 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 Sched_ext: serialise (enabled+all), task: runnable_at=-30ms RIP: 0010:jfs_ioc_trim+0x34b/0x8f0 Code: e7 e8 59 a4 87 fe 4d 8b 24 24 4d 8d bc 24 38 04 00 00 48 8d 93 90 82 fe ff 4c 89 ff 31 f6 RSP: 0018:ffffc900055f7cd0 EFLAGS: 00010206 RAX: 0000000000000087 RBX: 00005866a9e67ff8 RCX: 000000000000000a RDX: 0000000000000001 RSI: 0000000000000004 RDI: 0000000000000001 RBP: dffffc0000000000 R08: ffff88807c180003 R09: 1ffff1100f830000 R10: dffffc0000000000 R11: ffffd100f830001 R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000001 R15: 0000000000000438 FS: 00007fe520225640(0000) GS:ffff8880b7e80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005593c91b2c88 CR3: 000000014927c000 CR4: 00000000000006f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ? __die_body+0x61/0xb0 ? die_addr+0xb1/0xe0 ? exc_general_protection+0x333/0x510 ? asm_exc_general_protection+0x26/0x30 ? jfs_ioc_trim+0x34b/0x8f0 jfs_ioctl+0x3c8/0x4f0 ? __pfx_jfs_ioctl+0x10/0x10 ? __pfx_jfs_ioctl+0x10/0x10 __se_sys_ioctl+0x269/0x350 ? __pfx__se_sys_ioctl+0x10/0x10 ? do_syscall_64+0xfb/0x210 do_syscall_64+0xee/0x210 ? syscall_exit_to_user_mode+0x1e0/0x330 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fe51f4903ad Code: c3 e8 a7 2b 00 00 0f 1f 80 00 00 00 00 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d RSP: 002b:00007fe5202250c8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff80000000 RBX: 00007fe51f5cbf80 RCX: 00007fe51f4903ad RDX: 0000000020000680 RSI: 00000000c0185879 RDI: 0000000000000005 RBP: 0000000000000000 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 00007fe520225640 R13: 000000000000000e R14: 00007fe51f44fca0 R15: 00007fe52021d000 </TASK> Modules linked in: ---[end trace 0000000000000000]--- RIP: 0010:jfs_ioc_trim+0x34b/0x8f0 Code: e7 e8 59 a4 87 fe 4d 8b 24 24 4d 8d bc 24 38 04 00 00 48 8d 93 90 82 fe ff 4c 89 ff 31 f6 RSP: 0018:ffffc900055f7cd0 EFLAGS: 00010206 RAX: 0000000000000087 RBX: 00005866a9e67ff8 RCX: 000000000000000a RDX: 0000000000000001 RSI: 0000000000000004 RDI: 0000000000000001 RBP: dffffc0000000000 R08: ffff88807c180003 R09: 1ffff1100f830000 R10: dffffc0000000000 R11: ffffd100f830001 R12: 0000000000000000 R13: 0000000000000000 R14: 0000000000000001 R15: 0000000000000438 FS: 00007fe520225640(0000) GS:ffff8880b7e80000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005593c91b2c88 CR3: 000000014927c000 CR4: 00000000000006f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Kernel panic - not syncing: Fatal exception [Analysis] We believe that we have found a concurrency bug in the `fs/jfs` module that results in a null pointer dereference. There is a closely related issue which has been fixed: https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=d6c1b3599b2feb5c7291f5ac3a36e5fa7cedb234 ... but, unfortunately, the accepted patch appears to still be susceptible to a null pointer dereference under some interleavings. To trigger the bug, we think that `JFS_SBI(ipbmap->i_sb)->bmap` is set to NULL in `dbFreeBits` and then dereferenced in `jfs_ioc_trim`. This bug manifests quite rarely under normal circumstances, but is triggerable from a syz-program.
CVE-2025-38204	In the Linux kernel, the following vulnerability has been resolved: jfs: fix array-index-out-of-bounds read in add_missing_indices stbl is s8 but it must contain offsets into slot which can go from 0 to 127. Added a bound check for that error and return -EIO if the check fails. Also make jfs_readdir return with error if add_missing_indices returns with an error.
CVE-2025-38205	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Avoid divide by zero by initializing dummy pitch to 1 [Why] If the dummy values in `populate_dummy_dml_surface_cfg()` aren't updated then they can lead to a divide by zero in downstream callers like CalculateVMAndRowBytes() [How] Initialize dummy value to a value to avoid divide by zero.
CVE-2025-38206	In the Linux kernel, the following vulnerability has been resolved: exfat: fix double free in delayed_free The double free could happen in the following path. exfat_create_upcase_table() exfat_create_upcase_table() : return error exfat_free_upcase_table() : free ->vol_utbl exfat_load_default_upcase_table : return error exfat_kill_sb() delayed_free() exfat_free_upcase_table() <----- double free This patch set ->vol_util as NULL after freeing it.

CVE ID	Description
CVE-2025-38207	In the Linux kernel, the following vulnerability has been resolved: mm: fix uprobe pte be overwritten when expanding vma Patch series "Fix uprobe pte be overwritten when expanding vma". This patch (of 4): We encountered a BUG alert triggered by Syzkaller as follows: BUG: Bad rss-counter state mm:00000000b4a60fca type:MM_ANONPAGES val:1 And we can reproduce it with the following steps: 1. register uprobe on file at zero offset 2. mmap the file at zero offset: addr1 = mmap(NULL, 2 * 4096, PROT_NONE, MAP_PRIVATE, fd, 0); 3. mremap part of vma1 to new vma2: addr2 = mremap(addr1, 4096, 2 * 4096, MREMAP_MAYMOVE); 4. mremap back to orig addr1: mremap(addr2, 4096, 4096, MREMAP_MAYMOVE MREMAP_FIXED, addr1); In step 3, the vma1 range [addr1, addr1 + 4096] will be remap to new vma2 with range [addr2, addr2 + 8192], and remap uprobe anon page from the vma1 to vma2, then unmap the vma1 range [addr1, addr1 + 4096]. In step 4, the vma2 range [addr2, addr2 + 4096] will be remap back to the addr range [addr1, addr1 + 4096]. Since the addr range [addr1 + 4096, addr1 + 8192] still maps the file, it will take vma_merge_new_range to expand the range, and then do uprobe_mmap in vma_complete. Since the merged vma pgoff is also zero offset, it will install uprobe anon page to the merged vma. However, the upcoming move_page_tables step, which use set_pte_at to remap the vma2 uprobe pte to the merged vma, will overwrite the newly uprobe pte in the merged vma, and lead that pte to be orphan. Since the uprobe pte will be remapped to the merged vma, we can remove the unnecessary uprobe_mmap upon merged vma. This problem was first found in linux-6.6.y and also exists in the community syzkaller: https://lore.kernel.org/all/000000000000ada39605a5e71711@google.com/T/
CVE-2025-38208	In the Linux kernel, the following vulnerability has been resolved: smb: client: add NULL check in automount_fullpath page is checked for null in __build_path_from_dentry_optional_prefix when tcon->origin_fullpath is not set. However, the check is missing when it is set. Add a check to prevent a potential NULL pointer dereference.
CVE-2025-38210	In the Linux kernel, the following vulnerability has been resolved: configs-tsm-report: Fix NULL dereference of tsm_ops Unlike sysfs, the lifetime of configs objects is controlled by userspace. There is no mechanism for the kernel to find and delete all created config-items. Instead, the configs-tsm-report mechanism has an expectation that tsm_unregister() can happen at any time and cause established config-item access to start failing. That expectation is not fully satisfied. While tsm_report_read(), tsm_report_{is,is_bin}_visible(), and tsm_report_make_item() safely fail if tsm_ops have been unregistered, tsm_report_privlevel_store() tsm_report_provider_show() fail to check for ops registration. Add the missing checks for tsm_ops having been removed. Now, in supporting the ability for tsm_unregister() to always succeed, it leaves the problem of what to do with lingering config-items. The expectation is that the admin that arranges for the ->remove() (unbind) of the \${tsm_arch}-guest driver is also responsible for deletion of all open config-items. Until that deletion happens, ->probe() (reload / bind) of the \${tsm_arch}-guest driver fails. This allows for emergency shutdown / revocation of attestation interfaces, and requires coordinated restart.

CVE ID	Description
CVE-2025-38211	In the Linux kernel, the following vulnerability has been resolved: RDMA/iwcm: Fix use-after-free of work objects after cm_id destruction The commit 59c68ac31e15 ("iw_cm: free cm_id resources on the last deref") simplified cm_id resource management by freeing cm_id once all references to the cm_id were removed. The references are removed either upon completion of iw_cm event handlers or when the application destroys the cm_id. This commit introduced the use-after-free condition where cm_id_private object could still be in use by event handler works during the destruction of cm_id. The commit aee2424246f9 ("RDMA/iwcm: Fix a use-after-free related to destroying CM IDs") addressed this use-after-free by flushing all pending works at the cm_id destruction. However, still another use-after-free possibility remained. It happens with the work objects allocated for each cm_id_priv within alloc_work_entries() during cm_id creation, and subsequently freed in dealloc_work_entries() once all references to the cm_id are removed. If the cm_id's last reference is decremented in the event handler work, the work object for the work itself gets removed, and causes the use-after-free BUG below: BUG: KASAN: slab-use-after-free in __pwq_activate_work+0x1ff/0x250 Read of size 8 at addr ffff88811f9cf800 by task kworker/u16:1/147091 CPU: 2 UID: 0 PID: 147091 Comm: kworker/u16:1 Not tainted 6.15.0-rc2+ #27 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-3.fc41 04/01/2014 Workqueue: 0x0 (iw_cm_wq) Call Trace: <TASK> dump_stack_lvl+0x6a/0x90 print_report+0x174/0x554 ? __virt_addr_valid+0x208/0x430 ? __pwq_activate_work+0x1ff/0x250 kasan_report+0xae/0x170 ? __pwq_activate_work+0x1ff/0x250 __pwq_activate_work+0x1ff/0x250 pwq_dec_nr_in_flight+0x8c5/0xf0 process_one_work+0xc11/0x1460 ? __pfx_process_one_work+0x10/0x10 ? assign_work+0x16c/0x240 worker_thread+0x5ef/0xf0 ? __pfx_worker_thread+0x10/0x10 kthread+0x3b0/0x770 ? __pfx_kthread+0x10/0x10 ? rcu_is_watching+0x11/0xb0 ? __raw_spin_unlock_irq+0x24/0x50 ? rcu_is_watching+0x11/0xb0 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x30/0x70 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Allocated by task 147416: kasan_save_stack+0x2c/0x50 kasan_save_track+0x10/0x30 __kasan_kmalloc+0xa6/0xb0 alloc_work_entries+0xa9/0x260 [iw_cm] iw_cm_connect+0x23/0x4a0 [iw_cm] rdma_connect_locked+0xbfd/0x1920 [rdma_cm] nvme_rdma_cm_handler+0x8e5/0x1b60 [nvme_rdma] cma_cm_event_handler+0xae/0x320 [rdma_cm] cma_work_handler+0x106/0x1b0 [rdma_cm] process_one_work+0x84f/0x1460 worker_thread+0x5ef/0xf0 kthread+0x3b0/0x770 ret_from_fork+0x30/0x70 ret_from_fork_asm+0x1a/0x30 Freed by task 147091: kasan_save_stack+0x2c/0x50 kasan_save_track+0x10/0x30 kasan_save_free_info+0x37/0x60 __kasan_slab_free+0x4b/0x70 kfree+0x13a/0x4b0 dealloc_work_entries+0x125/0x1f0 [iw_cm] iwcm_deref_id+0x6f/0xa0 [iw_cm] cm_work_handler+0x136/0x1ba0 [iw_cm] process_one_work+0x84f/0x1460 worker_thread+0x5ef/0xf0 kthread+0x3b0/0x770 ret_from_fork+0x30/0x70 ret_from_fork_asm+0x1a/0x30 Last potentially related work creation: kasan_save_stack+0x2c/0x50 kasan_record_aux_stack+0xa3/0xb0 __queue_work+0x2ff/0x1390 queue_work_on+0x67/0xc0 cm_event_handler+0x46a/0x820 [iw_cm] siw_cm_upcall+0x330/0x650 [siw] siw_cm_work_handler+0x6b9/0x2b20 [siw] process_one_work+0x84f/0x1460 worker_thread+0x5ef/0xf0 kthread+0x3b0/0x770 ret_from_fork+0x30/0x70 ret_from_fork_asm+0x1a/0x30 This BUG is reproducible by repeating the blktests test case nvme/061 for the rdma transport and the siw driver. To avoid the use-after-free of cm_id_private work objects, ensure that the last reference to the cm_id is decremented not in the event handler works, but in the cm_id destruction context. For that purpose, mo ---truncated---
CVE-2025-38212	In the Linux kernel, the following vulnerability has been resolved: ipc: fix to protect IPCS lookups using RCU syzbot reported that it discovered a use-after-free vulnerability, [0] [0]: https://lore.kernel.org/all/67af13f8.050a0220.21dd3.0038.GAE@google.com/ idr_for_each() is protected by rwsem, but this is not enough. If it is not protected by RCU read-critical region, when idr_for_each() calls radix_tree_node_free() through call_rcu() to free the radix_tree_node structure, the node will be freed immediately, and when reading the next node in radix_tree_for_each_slot(), the already freed memory may be read. Therefore, we need to add code to make sure that idr_for_each() is protected within the RCU read-critical region when we call it in shm_destroy_orphaned().

CVE ID	Description
CVE-2025-38214	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix fb_set_var to prevent null-ptr-deref in fb_videomode_to_var If fb_add_videomode() in fb_set_var() fails to allocate memory for fb_videomode, later it may lead to a null-ptr dereference in fb_videomode_to_var(), as the fb_info is registered while not having the mode in modelist that is expected to be there, i.e. the one that is described in fb_info->var. <pre>===== general protection fault, probably for non-canonical address 0xdffffc0000000001: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] CPU: 1 PID: 30371 Comm: syz-executor.1 Not tainted 5.10.226-syzkaller #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 RIP: 0010:fb_videomode_to_var+0x24/0x610 drivers/video/fbdev/core/modedb.c:901 Call Trace: display_to_var+0x3a/0x7c0 drivers/video/fbdev/core/fbcon.c:929 fbcon_resize+0x3e2/0x8f0 drivers/video/fbdev/core/fbcon.c:2071 resize_screen drivers/tty/vt.c:1176 [inline] vc_do_resize +0x53a/0x1170 drivers/tty/vt.c:1263 fbcon_modechanged+0x3ac/0x6e0 drivers/video/ fbdev/core/fbcon.c:2720 fbcon_update_vcs+0x43/0x60 drivers/video/fbdev/core/fbcon.c:2776 do_fb_ioctl+0x6d2/0x740 drivers/video/fbdev/core/fbmem.c:1128 fb_ioctl+0xe7/0x150 drivers/ video/fbdev/core/fbmem.c:1203 vfs_ioctl fs/ioctl.c:48 [inline] __do_sys_ioctl fs/ioctl.c:753 [inline] __se_sys_ioctl fs/ioctl.c:739 [inline] __x64_sys_ioctl+0x19a/0x210 fs/ioctl.c:739 do_syscall_64+0x33/0x40 arch/x86/entry/common.c:46 entry_SYSCALL_64_after_hwframe +0x67/0xd1 =====</pre> The reason is that fb_info->var is being modified in fb_set_var(), and then fb_videomode_to_var() is called. If it fails to add the mode to fb_info->modelist, fb_set_var() returns error, but does not restore the old value of fb_info->var. Restore fb_info->var on failure the same way it is done earlier in the function. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-38215	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix do_register_framebuffer to prevent null-ptr-deref in fb_videomode_to_var If fb_add_videomode() in do_register_framebuffer() fails to allocate memory for fb_videomode, it will later lead to a null-ptr dereference in fb_videomode_to_var(), as the fb_info is registered while not having the mode in modelist that is expected to be there, i.e. the one that is described in fb_info->var. <pre>===== general protection fault, probably for non-canonical address 0xdffffc0000000001: 0000 [#1] PREEMPT SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] CPU: 1 PID: 30371 Comm: syz-executor.1 Not tainted 5.10.226-syzkaller #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 RIP: 0010:fb_videomode_to_var+0x24/0x610 drivers/video/fbdev/core/modedb.c:901 Call Trace: display_to_var+0x3a/0x7c0 drivers/video/fbdev/core/fbcon.c:929 fbcon_resize+0x3e2/0x8f0 drivers/video/fbdev/core/fbcon.c:2071 resize_screen drivers/tty/vt.c:1176 [inline] vc_do_resize +0x53a/0x1170 drivers/tty/vt.c:1263 fbcon_modechanged+0x3ac/0x6e0 drivers/video/ fbdev/core/fbcon.c:2720 fbcon_update_vcs+0x43/0x60 drivers/video/fbdev/core/fbcon.c:2776 do_fb_ioctl+0x6d2/0x740 drivers/video/fbdev/core/fbmem.c:1128 fb_ioctl+0xe7/0x150 drivers/ video/fbdev/core/fbmem.c:1203 vfs_ioctl fs/ioctl.c:48 [inline] __do_sys_ioctl fs/ioctl.c:753 [inline] __se_sys_ioctl fs/ioctl.c:739 [inline] __x64_sys_ioctl+0x19a/0x210 fs/ioctl.c:739 do_syscall_64+0x33/0x40 arch/x86/entry/common.c:46 entry_SYSCALL_64_after_hwframe +0x67/0xd1 =====</pre> Even though fbcon_init() checks beforehand if fb_match_mode() in var_to_display() fails, it can not prevent the panic because fbcon_init() does not return error code. Considering this and the comment in the code about fb_match_mode() returning NULL - "This should not happen" - it is better to prevent registering the fb_info if its mode was not set successfully. Also move fb_add_videomode() closer to the beginning of do_register_framebuffer() to avoid having to do the cleanup on fail. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-38217	In the Linux kernel, the following vulnerability has been resolved: hwmon: (ftsutates) Fix TOCTOU race in fts_read() In the fts_read() function, when handling hwmon_pwm_auto_channels_temp, the code accesses the shared variable data->fan_source[channel] twice without holding any locks. It is first checked against FTS_FAN_SOURCE_INVALID, and if the check passes, it is read again when used as an argument to the BIT() macro. This creates a Time-of-Check to Time-of-Use (TOCTOU) race condition. Another thread executing fts_update_device() can modify the value of data->fan_source[channel] between the check and its use. If the value is changed to FTS_FAN_SOURCE_INVALID (0xff) during this window, the BIT() macro will be called with a large shift value (BIT(255)). A bit shift by a value greater than or equal to the type width is undefined behavior and can lead to a crash or incorrect values being returned to userspace. Fix this by reading data->fan_source[channel] into a local variable once, eliminating the race condition. Additionally, add a bounds check to ensure the value is less than BITS_PER_LONG before passing it to the BIT() macro, making the code more robust against undefined behavior. This possible bug was found by an experimental static analysis tool developed by our team.

CVE ID	Description
CVE-2025-38218	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on sit_bitmap_size w/ below testcase, resize will generate a corrupted image which contains inconsistent metadata, so when mounting such image, it will trigger kernel panic: touch img truncate -s \$((512*1024*1024*1024)) img mkfs.f2fs -f img \$((256*1024*1024)) resize.f2fs -s -i img -t \$((1024*1024*1024)) mount img /mnt/f2fs -----[cut here]----- kernel BUG at fs/f2fs/segment.h:863! Oops: invalid opcode: 0000 [#1] SMP PTI CPU: 11 UID: 0 PID: 3922 Comm: mount Not tainted 6.15.0-rc1+ #191 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010:f2fs_ra_meta_pages+0x47c/0x490 Call Trace: f2fs_build_segment_manager+0x11c3/0x2600 f2fs_fill_super+0xe97/0x2840 mount_bdev+0xf4/0x140 legacy_get_tree+0x2b/0x50 vfs_get_tree+0x29/0xd0 path_mount+0x487/0xaf0 __x64_sys_mount+0x116/0x150 do_syscall_64+0x82/0x190 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7fdbfde1bcfe The reason is: sit_i->bitmap_size is 192, so size of sit_bitmap is 192*8=1536, at maximum there are 1536 sit blocks, however MAIN_SEGS is 261893, so that sit_blk_cnt is 4762, build_sit_entries() -> current_sit_addr() tries to access out-of-boundary in sit_bitmap at offset from [1536, 4762), once sit_bitmap and sit_bitmap_mirror is not the same, it will trigger f2fs_bug_on(). Let's add sanity check in f2fs_sanity_check_ckpt() to avoid panic.
CVE-2025-38219	In the Linux kernel, the following vulnerability has been resolved: f2fs: prevent kernel warning due to negative i_nlink from corrupted image WARNING: CPU: 1 PID: 9426 at fs/inode.c:417 drop_nlink+0xac/0xd0 home/cc/linux/fs/inode.c:417 Modules linked in: CPU: 1 UID: 9426 Comm: syz-executor568 Not tainted 6.14.0-12627-g94d471a4f428 #2 PREEMPT(full) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014 RIP: 0010:drop_nlink+0xac/0xd0 home/cc/linux/fs/inode.c:417 Code: 48 8b 5d 28 be 08 00 00 00 48 8d bb 70 07 00 00 e8 f9 67 e6 ff f0 4d ff 83 70 07 00 00 5b 5d e9 9a 12 82 ff e8 95 12 82 ff 90 <0f> 0b 90 c7 45 48 ff ff ff ff 5b 5d e9 83 12 82 ff e8 fe 5f e6 ff RSP: 0018:ffffc900026b7c28 EFLAGS: 00010293 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffffff8239710f RDX: ffff888041345a00 RSI: ffffffff8239717b RDI: 0000000000000005 RBP: ffff888054509ad0 R08: 0000000000000005 R09: 0000000000000000 R10: 0000000000000000 R11: ffffffff9ab36f08 R12: ffff88804bb40000 R13: ffff8880545091e0 R14: 0000000000008000 R15: ffff8880545091e0 FS: 00005555d0c5880(0000) GS:ffff8880eb3e3000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f915c55b178 CR3: 0000000050d20000 CR4: 000000000352ef0 Call Trace: <task> f2fs_i_links_write home/cc/linux/fs/f2fs/f2fs.h:3194 [inline] f2fs_drop_nlink+0xd1/0x3c0 home/cc/linux/fs/f2fs/dir.c:845 f2fs_delete_entry+0x542/0x1450 home/cc/linux/fs/f2fs/dir.c:909 f2fs_unlink+0x45c/0x890 home/cc/linux/fs/f2fs/namei.c:581 vfs_unlink+0x2fb/0x9b0 home/cc/linux/fs/namei.c:4544 do_unlinkat+0x4c5/0x6a0 home/cc/linux/fs/namei.c:4608 __do_sys_unlink home/cc/linux/fs/namei.c:4654 [inline] __se_sys_unlink home/cc/linux/fs/namei.c:4652 [inline] __x64_sys_unlink+0xc5/0x110 home/cc/linux/fs/namei.c:4652 do_syscall_x64 home/cc/linux/arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xc7/0x250 home/cc/linux/arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fb3d092324b Code: 73 01 c3 48 c7 c1 c0 ff ff ff f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa b8 57 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 c0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007ffdc232d938 EFLAGS: 00000206 ORIG_RAX: 0000000000000057 RAX: ffffffff8239710f RBX: 0000000000000000 RCX: 00007fb3d092324b RDX: 00007ffdc232d960 RSI: 00007ffdc232d960 RDI: 00007ffdc232d9f0 RBP: 00007ffdc232d9f0 R08: 0000000000000001 R09: 00007ffdc232d7c0 R10: 00000000fffffffd R11: 0000000000000206 R12: 00007ffdc232eaf0 R13: 00005555d0c5880 R14: 00007ffdc232d958 R15: 0000000000000001 </task>
CVE-2025-38220	In the Linux kernel, the following vulnerability has been resolved: ext4: only dirty folios when data journaling regular files fstest generic/388 occasionally reproduces a crash that looks as follows: BUG: kernel NULL pointer dereference, address: 0000000000000000 ... Call Trace: <TASK> ext4_block_zero_page_range+0x30c/0x380 [ext4] ext4_truncate+0x436/0x440 [ext4] ext4_process_orphan+0x5d/0x110 [ext4] ext4_orphan_cleanup+0x124/0x4f0 [ext4] ext4_fill_super+0x262d/0x3110 [ext4] get_tree_bdev_flags+0x132/0x1d0 vfs_get_tree+0x26/0xd0 vfs_cmd_create+0x59/0xe0 __do_sys_fsconfig+0x4ed/0x6b0 do_syscall_64+0x82/0x170 ... This occurs when processing a symlink inode from the orphan list. The partial block zeroing code in the truncate path calls ext4_dirty_journalled_data() -> folio_mark_dirty(). The latter calls mapping->a_ops->dirty_folio(), but symlink inodes are not assigned an a_ops vector in ext4, hence the crash. To avoid this problem, update the ext4_dirty_journalled_data() helper to only mark the folio dirty on regular files (for which a_ops is assigned). This also matches the journaling logic in the ext4_symlink() creation path, where ext4_handle_dirty_metadata() is called directly.

CVE ID	Description
CVE-2025-38222	In the Linux kernel, the following vulnerability has been resolved: ext4: inline: fix len overflow in ext4_prepare_inline_data When running the following code on an ext4 filesystem with inline_data feature enabled, it will lead to the bug below. fd = open("file1", O_RDWR O_CREAT O_TRUNC, 0666); ftruncate(fd, 30); pwrite(fd, "a", 1, (1UL << 40) + 5UL); That happens because write_begin will succeed as when ext4_generic_write_inline_data calls ext4_prepare_inline_data, pos + len will be truncated, leading to ext4_prepare_inline_data parameter to be 6 instead of 0x10000000006. Then, later when write_end is called, we hit: BUG_ON(pos + len > EXT4_I(inode)->i_inline_size); at ext4_write_inline_data. Fix it by using a loff_t type for the len parameter in ext4_prepare_inline_data instead of an unsigned int. [44.545164] -----[cut here]----- [44.545530] kernel BUG at fs/ext4/inline.c:240! [44.545834] Oops: invalid opcode: 0000 [#1] SMP NOPTI [44.546172] CPU: 3 UID: 0 PID: 343 Comm: test Not tainted 6.15.0-rc2-00003-g9080916f4863 #45 PREEMPT(full) 112853fcebfbdb93254270a7959841d2c6aa2c8bb [44.546523] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [44.546523] RIP: 0010:ext4_write_inline_data+0xfe/0x100 [44.546523] Code: 3c 0e 48 83 c7 48 89 de 5b 41 5c 41 5d 41 5e 41 5f 5d e9 e4 fa 43 01 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc cc 0f 0b <0f> 0b 0f 1f 44 00 00 55 41 57 41 56 41 55 41 54 53 48 83 ec 20 49 [44.546523] RSP: 0018:ffffb342008b79a8 EFLAGS: 00010216 [44.546523] RAX: 0000000000000001 RBX: ffff9329c579c000 RCX: 0000010000000006 [44.546523] RDX: 000000000000003c RSI: ffff342008b79f0 RDI: ffff9329c158e738 [44.546523] RBP: 0000000000000001 R08: 0000000000000001 R09: 0000000000000000 [44.546523] R10: 00007fffff000 R11: ffffffff9bd0d910 R12: 0000006210000000 [44.546523] R13: fffffc7e4015e700 R14: 0000010000000005 R15: ffff9329c158e738 [44.546523] FS: 00007f4299934740(0000) GS:ffff932a60179000(0000) knlGS:0000000000000000 [44.546523] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [44.546523] CR2: 00007f4299a1ec90 CR3: 0000000002886002 CR4: 000000000770eb0 [44.546523] PKRU: 55555554 [44.546523] Call Trace: [44.546523] <TASK> [44.546523] ext4_write_inline_data_end+0x126/0x2d0 [44.546523] generic_perform_write+0x17e/0x270 [44.546523] ext4_buffered_write_iter+0xc8/0x170 [44.546523] vfs_write+0x2be/0x3e0 [44.546523] __x64_sys_pwrite64+0x6d/0xc0 [44.546523] do_syscall_64+0x6a/0xf0 [44.546523] ? __wake_up+0x89/0xb0 [44.546523] ? xas_find+0x72/0x1c0 [44.546523] ? next_uptodate_folio+0x317/0x330 [44.546523] ? set_pte_range+0x1a6/0x270 [44.546523] ? filemap_map_pages+0x6ee/0x840 [44.546523] ? ext4_setattr+0x2fa/0x750 [44.546523] ? do_pte_missing+0x128/0xf70 [44.546523] ? security_inode_post_setattr+0x3e/0xd0 [44.546523] ? __pte_offset_map+0x19/0x100 [44.546523] ? handle_mm_fault+0x721/0xa10 [44.546523] ? do_user_addr_fault+0x197/0x730 [44.546523] ? do_syscall_64+0x76/0xf0 [44.546523] ? arch_exit_to_user_mode_prepare+0x1e/0x60 [44.546523] ? irqentry_exit_to_user_mode+0x79/0x90 [44.546523] entry_SYSCALL_64_after_hwframe+0x55/0x5d [44.546523] RIP: 0033:0x7f42999c6687 [44.546523] Code: 48 89 fa 4c 89 df e8 58 b3 00 00 8b 93 08 03 00 00 59 5e 48 83 f8 fc 74 1a 5b c3 0f 1f 84 00 00 00 00 00 48 8b 44 24 10 0f 05 <5b> c3 0f 1f 80 00 00 00 00 83 e2 39 83 fa 08 75 de e8 23 ff ff ff [44.546523] RSP: 002b:00007f42999c6687 EFLAGS: 00000202 ORIG_RAX: 0000000000000012 [44.546523] RAX: ffffffff9329c158 RBX: 00007f4299934740 RCX: 00007f42999c6687 [44.546523] RDX: 0000000000000001 RSI: 000055ea6149200f RDI: 0000000000000003 [44.546523] RBP: 00007f42999c6687 R08: 0000000000000000 R09: 0000000000000000 [44.546523] R10: 0000010000000005 R11: 0000000000000202 R12: 0000 ---truncated---
CVE-2025-38225	In the Linux kernel, the following vulnerability has been resolved: media: imx-jpeg: Cleanup after an allocation error When allocation failures are not cleaned up by the driver, further allocation errors will be false-positives, which will cause buffers to remain uninitialized and cause NULL pointer dereferences. Ensure proper cleanup of failed allocations to prevent these issues.
CVE-2025-38226	In the Linux kernel, the following vulnerability has been resolved: media: vivid: Change the size of the composing syzkaller found a bug: BUG: KASAN: vmalloc-out-of-bounds in tpg_fill_plane_pattern drivers/media/common/v4l2-tpg/v4l2-tpg-core.c:2608 [inline] BUG: KASAN: vmalloc-out-of-bounds in tpg_fill_plane_buffer+0x1a9c/0x5af0 drivers/media/common/v4l2-tpg/v4l2-tpg-core.c:2705 Write of size 1440 at addr fffff9000d0ffda0 by task vivid-000-vid-c/5304 CPU: 0 UID: 0 PID: 5304 Comm: vivid-000-vid-c Not tainted 6.14.0-rc2-syzkaller-00039-g09fbf3d50205 #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x169/0x550 mm/kasan/report.c:489 kasan_report+0x143/0x180 mm/kasan/report.c:602 kasan_check_range+0x282/0x290 mm/kasan/generic.c:189 __asan_memcpy+0x40/0x70 mm/kasan/shadow.c:106 tpg_fill_plane_pattern drivers/media/common/v4l2-tpg/v4l2-tpg-core.c:2608 [inline] tpg_fill_plane_buffer+0x1a9c/0x5af0 drivers/media/common/v4l2-tpg/v4l2-tpg-core.c:2705 vivid_fillbuff drivers/media/test-drivers/vivid/vivid-kthread-cap.c:470 [inline] vivid_thread_vid_cap_tick+0xf8e/0x60d0 drivers/media/test-drivers/vivid/vivid-kthread-cap.c:629 vivid_thread_vid_cap+0x8aa/0xf30 drivers/media/test-drivers/vivid/vivid-kthread-cap.c:767 kthread+0x7a9/0x920 kernel/kthread.c:464 ret_from_fork+0x4b/0x80 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:244 </TASK> The composition size cannot be larger than the size of fmt_cap_rect. So execute v4l2_rect_map_inside() even if has_compose_cap == 0.

CVE ID	Description
CVE-2025-38229	In the Linux kernel, the following vulnerability has been resolved: media: cxusb: no longer judge rbuf when the write fails syzbot reported a uninit-value in cxusb_i2c_xfer. [1] Only when the write operation of usb_bulk_msg() in dvb_usb_generic_rw() succeeds and rlen is greater than 0, the read operation of usb_bulk_msg() will be executed to read rlen bytes of data from the dvb device into the rbuf. In this case, although rlen is 1, the write operation failed which resulted in the dvb read operation not being executed, and ultimately variable i was not initialized. [1] BUG: KMSAN: uninit-value in cxusb_gpio_tuner drivers/media/usb/dvb-usb/cxusb.c:124 [inline] BUG: KMSAN: uninit-value in cxusb_i2c_xfer+0x153a/0x1a60 drivers/media/usb/dvb-usb/cxusb.c:196 cxusb_gpio_tuner drivers/media/usb/dvb-usb/cxusb.c:124 [inline] cxusb_i2c_xfer+0x153a/0x1a60 drivers/media/usb/dvb-usb/cxusb.c:196 __i2c_transfer+0xe25/0x3150 drivers/i2c/i2c-core-base.c:-1 i2c_transfer+0x317/0x4a0 drivers/i2c/i2c-core-base.c:2315 i2c_transfer_buffer_flags+0x125/0x1e0 drivers/i2c/i2c-core-base.c:2343 i2c_master_send include/linux/i2c.h:109 [inline] i2cdev_write+0x210/0x280 drivers/i2c/i2c-dev.c:183 do_loop_readv_writev fs/read_write.c:848 [inline] vfs_writev+0x963/0x14e0 fs/read_write.c:1057 do_writev+0x247/0x5c0 fs/read_write.c:1101 __do_sys_writev fs/read_write.c:1169 [inline] __se_sys_writev fs/read_write.c:1166 [inline] __x64_sys_writev+0x98/0xe0 fs/read_write.c:1166 x64_sys_call+0x2229/0x3c80 arch/x86/include/generated/asm/syscalls_64.h:21 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x1e0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2025-38230	In the Linux kernel, the following vulnerability has been resolved: jfs: validate AG parameters in dbMount() to prevent crashes Validate db_agheight, db_agwidth, and db_agstart in dbMount to catch corrupted metadata early and avoid undefined behavior in dbAllocAG. Limits are derived from L2LPERCTL, LPERCTL/MAXAG, and CTLTREESIZE: - agheight: 0 to L2LPERCTL/2 (0 to 5) ensures shift (L2LPERCTL - 2*agheight) >= 0. - agwidth: 1 to min(LPERCTL/MAXAG, 2^(L2LPERCTL - 2*agheight)) ensures agperlev >= 1. - Ranges: 1-8 (agheight 0-3), 1-4 (agheight 4), 1 (agheight 5). - LPERCTL/MAXAG = 1024/128 = 8 limits leaves per AG; 2^(10 - 2*agheight) prevents division to 0. - agstart: 0 to CTLTREESIZE-1 - agwidth*(MAXAG-1) keeps ti within stree (size 1365). - Ranges: 0-1237 (agwidth 1), 0-348 (agwidth 8). UBSAN: shift-out-of-bounds in fs/jfs/jfs_dmap.c:1400:9 shift exponent -335544310 is negative CPU: 0 UID: 0 PID: 5822 Comm: syz-executor130 Not tainted 6.14.0-rc5-syzkaller #0 Hardware name: Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120 ubsan_epilogue lib/ubsan.c:231 [inline] __ubsan_handle_shift_out_of_bounds+0x3c8/0x420 lib/ubsan.c:468 dbAllocAG+0x1087/0x10b0 fs/jfs/jfs_dmap.c:1400 dbDiscardAG+0x352/0xa20 fs/jfs/jfs_dmap.c:1613 jfs_ioc_trim+0x45a/0x6b0 fs/jfs/jfs_discard.c:105 jfs_ioctl+0x2cd/0x3e0 fs/jfs/ioctl.c:131 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl+0xf5/0x170 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-38231	In the Linux kernel, the following vulnerability has been resolved: nfsd: Initialize ssc before laundromat_work to prevent NULL dereference In nfs4_state_start_net(), laundromat_work may access nfsd_ssc through nfs4_laundromat -> nfsd4_ssc_expire_umount. If nfsd_ssc isn't initialized, this can cause NULL pointer dereference. Normally the delayed start of laundromat_work allows sufficient time for nfsd_ssc initialization to complete. However, when the kernel waits too long for userspace responses (e.g. in nfs4_state_start_net -> nfsd4_end_grace -> nfsd4_record_grace_done -> nfsd4_cld_grace_done -> cld_pipe_upcall -> __cld_pipe_upcall -> wait_for_completion path), the delayed work may start before nfsd_ssc initialization finishes. Fix this by moving nfsd_ssc initialization before starting laundromat_work.

CVE ID	Description
CVE-2025-38232	In the Linux kernel, the following vulnerability has been resolved: NFSD: fix race between nfsd registration and exports_proc As of now nfsd calls create_proc_exports_entry() at start of init_nfsd and cleanup by remove_proc_entry() at last of exit_nfsd. Which causes kernel OOPs if there is race between below 2 operations: (i) exportfs -r (ii) mount -t nfsd none /proc/fs/nfsd for 5.4 kernel ARM64: CPU 1: el1_irq+0xbc/0x180 arch_counter_get_cntvct+0x14/0x18 running_clock+0xc/0x18 preempt_count_add+0x88/0x110 prep_new_page+0xb0/0x220 get_page_from_freelist+0x2d8/0x1778 __alloc_pages_nodemask+0x15c/0xef0 __vmalloc_node_range+0x28c/0x478 __vmalloc_node_flags_caller+0x8c/0xb0 kvmalloc_node+0x88/0xe0 nfsd_init_net+0x6c/0x108 [nfsd] ops_init+0x44/0x170 register_pernet_operations+0x114/0x270 register_pernet_subsys+0x34/0x50 init_nfsd+0xa8/0x718 [nfsd] do_one_initcall+0x54/0x2e0 CPU 2 : Unable to handle kernel NULL pointer dereference at virtual address 0000000000000010 PC is at : exports_net_open+0x50/0x68 [nfsd] Call trace: exports_net_open+0x50/0x68 [nfsd] exports_proc_open+0x2c/0x38 [nfsd] proc_reg_open+0xb8/0x198 do_dentry_open+0x1c4/0x418 vfs_open+0x38/0x48 path_openat+0x28c/0xf18 do_filp_open+0x70/0xe8 do_sys_open+0x154/0x248 Sometimes it crashes at exports_net_open() and sometimes cache_seq_next_rcu(). and same is happening on latest 6.14 kernel as well: [0.000000] Linux version 6.14.0-rc5-next-20250304-dirty ... [285.455918] Unable to handle kernel paging request at virtual address 00001f4800001f48 ... [285.464902] pc : cache_seq_next_rcu+0x78/0xa4 ... [285.469695] Call trace: [285.470083] cache_seq_next_rcu+0x78/0xa4 (P) [285.470488] seq_read+0xe0/0x11c [285.470675] proc_reg_read+0x9c/0xf0 [285.470874] vfs_read+0xc4/0x2fc [285.471057] ksys_read+0x6c/0xf4 [285.471231] __arm64_sys_read+0x1c/0x28 [285.471428] invoke_syscall+0x44/0x100 [285.471633] el0_svc_common.constprop.0+0x40/0xe0 [285.471870] do_el0_svc_compat+0x1c/0x34 [285.472073] el0_svc_compat+0x2c/0x80 [285.472265] el0t_32_sync_handler+0x90/0x140 [285.472473] el0t_32_sync+0x19c/0x1a0 [285.472887] Code: f9400885 93407c23 937d7c27 11000421 (f86378a3) [285.473422] ---[end trace 0000000000000000]--- It reproduced simply with below script: while [1] do /exportfs -r done & while [1] do insmod /nfsd.ko mount -t nfsd none /proc/fs/nfsd umount /proc/fs/nfsd rmmod nfsd done & So exporting interfaces to user space shall be done at last and cleanup at first place. With change there is no Kernel OOPs.

CVE ID	Description
CVE-2025-38234	In the Linux kernel, the following vulnerability has been resolved: sched/rt: Fix race in push_rt_task Overview ===== When a CPU chooses to call push_rt_task and picks a task to push to another CPU's runqueue then it will call find_lock_lowest_rq method which would take a double lock on both CPUs' runqueues. If one of the locks aren't readily available, it may lead to dropping the current runqueue lock and reacquiring both the locks at once. During this window it is possible that the task is already migrated and is running on some other CPU. These cases are already handled. However, if the task is migrated and has already been executed and another CPU is now trying to wake it up (ttwu) such that it is queued again on the runqueue (on_rq is 1) and also if the task was run by the same CPU, then the current checks will pass even though the task was migrated out and is no longer in the pushable tasks list. Crashes ===== This bug resulted in quite a few flavors of crashes triggering kernel panics with various crash signatures such as assert failures, page faults, null pointer dereferences, and queue corruption errors all coming from scheduler itself. Some of the crashes: -> kernel BUG at kernel/sched/rt.c:1616! BUG_ON(idx >= MAX_RT_PRIO) Call Trace: ? __die_body+0x1a/0x60 ? die+0x2a/0x50 ? do_trap+0x85/0x100 ? pick_next_task_rt+0x6e/0x1d0 ? do_error_trap+0x64/0xa0 ? pick_next_task_rt+0x6e/0x1d0 ? exc_invalid_op+0x4c/0x60 ? pick_next_task_rt+0x6e/0x1d0 ? asm_exc_invalid_op+0x12/0x20 ? pick_next_task_rt+0x6e/0x1d0 __schedule+0x5cb/0x790 ? update_ts_time_stats+0x55/0x70 schedule_idle+0x1e/0x40 do_idle+0x15e/0x200 cpu_startup_entry+0x19/0x20 start_secondary+0x117/0x160 secondary_startup_64_no_verify+0xb0/0xbb -> BUG: kernel NULL pointer dereference, address: 00000000000000c0 Call Trace: ? __die_body+0x1a/0x60 ? no_context+0x183/0x350 ? __warn+0x8a/0xe0 ? exc_page_fault+0x3d6/0x520 ? asm_exc_page_fault+0x1e/0x30 ? pick_next_task_rt+0xb5/0x1d0 ? pick_next_task_rt+0x8c/0x1d0 __schedule+0x583/0x7e0 ? update_ts_time_stats+0x55/0x70 schedule_idle+0x1e/0x40 do_idle+0x15e/0x200 cpu_startup_entry+0x19/0x20 start_secondary+0x117/0x160 secondary_startup_64_no_verify+0xb0/0xbb -> BUG: unable to handle page fault for address: ffff9464daea5900 kernel BUG at kernel/sched/rt.c:1861! BUG_ON(rq->cpu != task_cpu(p)) -> kernel BUG at kernel/sched/rt.c:1055! BUG_ON(!rq->nr_running) Call Trace: ? __die_body+0x1a/0x60 ? die+0x2a/0x50 ? do_trap+0x85/0x100 ? dequeue_top_rt_rq+0xa2/0xb0 ? do_error_trap+0x64/0xa0 ? dequeue_top_rt_rq+0xa2/0xb0 ? exc_invalid_op+0x4c/0x60 ? dequeue_top_rt_rq+0xa2/0xb0 ? asm_exc_invalid_op+0x12/0x20 ? dequeue_top_rt_rq+0xa2/0xb0 dequeue_rt_entity+0x1f/0x70 dequeue_task_rt+0x2d/0x70 __schedule+0x1a8/0x7e0 ? blk_finish_plug+0x25/0x40 schedule+0x3c/0xb0 futex_wait_queue_me+0xb6/0x120 futex_wait+0xd9/0x240 do_futex+0x344/0xa90 ? get_mm_exe_file+0x30/0x60 ? audit_exe_compare+0x58/0x70 ? audit_filter_rules.constprop.26+0x65e/0x1220 __x64_sys_futex+0x148/0x1f0 do_syscall_64+0x30/0x80 entry_SYSCALL_64_after_hwframe+0x62/0xc7 -> BUG: unable to handle page fault for address: ffff8cf3608bc2c0 Call Trace: ? __die_body+0x1a/0x60 ? no_context+0x183/0x350 ? spurious_kernel_fault+0x171/0x1c0 ? exc_page_fault+0x3b6/0x520 ? plist_check_list+0x15/0x40 ? plist_check_list+0x2e/0x40 ? asm_exc_page_fault+0x1e/0x30 ? _cond_resched+0x15/0x30 ? futex_wait_queue_me+0xc8/0x120 ? futex_wait+0xd9/0x240 ? try_to_wake_up+0x1b8/0x490 ? futex_wake+0x78/0x160 ? do_futex+0xcd/0xa90 ? plist_check_list+0x15/0x40 ? plist_check_list+0x2e/0x40 ? plist_del+0x6a/0xd0 ? plist_check_list+0x15/0x40 ? plist_check_list+0x2e/0x40 ? dequeue_pushable_task+0x20/0x70 ? __schedule+0x382/0x7e0 ? asm_sysvec_resched_i ---truncated---

CVE ID	Description
CVE-2025-38236	In the Linux kernel, the following vulnerability has been resolved: af_unix: Don't leave consecutive consumed OOB skbs. Jann Horn reported a use-after-free in <code>unix_stream_read_generic()</code>. The following sequences reproduce the issue: <code>\$ python3 from socket import * s1, s2 = socketpair(AF_UNIX, SOCK_STREAM) s1.send(b'x', MSG_OOB) s2.recv(1, MSG_OOB) # leave a consumed OOB skb s1.send(b'y', MSG_OOB) s2.recv(1, MSG_OOB) # leave a consumed OOB skb s1.send(b'z', MSG_OOB) s2.recv(1) # recv 'z' illegally s2.recv(1, MSG_OOB) # access 'z' skb (use-after-free)</code> Even though a user reads OOB data, the skb holding the data stays on the recv queue to mark the OOB boundary and break the next <code>recv()</code>. After the last <code>send()</code> in the scenario above, the <code>sk2</code>'s <code>recv</code> queue has 2 leading consumed OOB skbs and 1 real OOB skb. Then, the following happens during the next <code>recv()</code> without <code>MSG_OOB</code> 1. <code>unix_stream_read_generic()</code> peeks the first consumed OOB skb 2. <code>manage_oob()</code> returns the next consumed OOB skb 3. <code>unix_stream_read_generic()</code> fetches the next not-yet-consumed OOB skb 4. <code>unix_stream_read_generic()</code> reads and frees the OOB skb, and the last <code>recv(MSG_OOB)</code> triggers KASAN splat. The 3. above occurs because of the <code>SO_PEEK_OFF</code> code, which does not expect <code>unix_skb_len(skb)</code> to be 0, but this is true for such consumed OOB skbs. while (skip >= unix_skb_len(skb)) { skip -= unix_skb_len(skb); skb = skb_peek_next(skb, &sk->sk_receive_queue); ... } In addition to this use-after-free, there is another issue that <code>ioctl(SIOCATMARK)</code> does not function properly with consecutive consumed OOB skbs. So, nothing good comes out of such a situation. Instead of complicating <code>manage_oob()</code>, <code>ioctl()</code> handling, and the next <code>ECONNRESET</code> fix by introducing a loop for consecutive consumed OOB skbs, let's not leave such consecutive OOB unnecessarily. Now, while receiving an OOB skb in <code>unix_stream_recvmsg()</code>, if its previous skb is a consumed OOB skb, it is freed. [0]: BUG: KASAN: slab-use-after-free in <code>unix_stream_read_actor</code> (net/unix/af_unix.c:3027) Read of size 4 at addr ffff888106ef2904 by task python3/315 CPU: 2 UID: 0 PID: 315 Comm: python3 Not tainted 6.16.0-rc1-00407-gec315832f6f9 #8 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-4.fc42 04/01/2014 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:122) print_report (mm/kasan/report.c:409 mm/kasan/report.c:521) kasan_report (mm/kasan/report.c:636) unix_stream_read_actor (net/unix/af_unix.c:3027) unix_stream_read_generic (net/unix/af_unix.c:2708 net/unix/af_unix.c:2847) unix_stream_recvmsg (net/unix/af_unix.c:3048) sock_recvmsg (net/socket.c:1063 (discriminator 20) net/socket.c:1085 (discriminator 20)) __sys_recvfrom (net/socket.c:2278) __x64_sys_recvfrom (net/socket.c:2291 (discriminator 1) net/socket.c:2287 (discriminator 1) net/socket.c:2287 (discriminator 1)) do_syscall_64 (arch/x86/entry/syscall_64.c:63 (discriminator 1) arch/x86/entry/syscall_64.c:94 (discriminator 1)) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) RIP: 0033:0x7f8911fcea06 Code: 5d e8 41 8b 93 08 03 00 00 59 5e 48 83 f8 fc 75 19 83 e2 39 83 fa 08 75 11 e8 26 ff ff 66 0f 1f 44 00 00 48 8b 45 10 0f 05 <48> 8b 5d f8 c9 c3 0f 1f 40 00 f3 0f 1e fa 55 48 89 e5 48 83 ec 08 RSP: 002b:00007ffdb0dccb0 EFLAGS: 00000202 ORIG_RAX: 000000000000002d RAX: ffffffff8bda RBX: 00007ffdb0dcdc8 RCX: 00007f8911fcea06 RDX: 0000000000000001 RSI: 00007f8911a5e060 RDI: 0000000000000006 RBP: 00007ffdb0dccb0 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000020 R12: 00007f89119a7d20 R13: ffffffff4653600 R14: 0000000000000000 R15: 0000000000000000 </TASK> Allocated by task 315: kasan_save_stack (mm/kasan/common.c:48) kasan_save_track (mm/kasan/common.c:60 (discriminator 1) mm/kasan/common.c:69 (discriminator 1)) __kasan_slab_alloc (mm/kasan/common.c:348) kmem_cache_alloc ---truncated---
CVE-2025-38237	In the Linux kernel, the following vulnerability has been resolved: media: platform: exynos4-is: Add hardware sync wait to <code>fimc_is_hw_change_mode()</code> In <code>fimc_is_hw_change_mode()</code>, the function changes camera modes without waiting for hardware completion, risking corrupted data or system hangs if subsequent operations proceed before the hardware is ready. Add <code>fimc_is_hw_wait_intmsr0_intmsd0()</code> after mode configuration, ensuring hardware state synchronization and stable interrupt handling.
CVE-2025-38239	In the Linux kernel, the following vulnerability has been resolved: scsi: megaraid_sas: Fix invalid node index On a system with DRAM interleave enabled, out-of-bound access is detected: megaraid_sas 0000:3f:00.0: requested/available msix 128/128 poll_queue 0 -----[cut here]----- UBSAN: array-index-out-of-bounds in ./arch/x86/include/asm/topology.h:72:28 index -1 is out of range for type 'cpumask *[1024]' dump_stack_lvl+0x5d/0x80 ubsan_epilogue+0x5/0x2b __ubsan_handle_out_of_bounds.cold+0x46/0x4b megasas_alloc_irq_vectors+0x149/0x190 [megaraid_sas] megasas_probe_one.cold+0xa4d/0x189c [megaraid_sas] local_pci_probe+0x42/0x90 pci_device_probe+0xdc/0x290 really_probe+0xdb/0x340 __driver_probe_device+0x78/0x110 driver_probe_device+0x1f/0xa0 __driver_attach+0xba/0x1c0 bus_for_each_dev+0x8b/0xe0 bus_add_driver+0x142/0x220 driver_register+0x72/0xd0 megasas_init+0xdf/0xff0 [megaraid_sas] do_one_initcall+0x57/0x310 do_init_module+0x90/0x250 init_module_from_file+0x85/0xc0 idempotent_init_module+0x114/0x310 __x64_sys_finit_module+0x65/0xc0 do_syscall_64+0x82/0x170 entry_SYSCALL_64_after_hwframe+0x76/0x7e Fix it accordingly.

CVE ID	Description
CVE-2025-38240	In the Linux kernel, the following vulnerability has been resolved: drm/mediatek: dp: drm_err => dev_err in HPD path to avoid NULL ptr The function mtk_dp_wait_hpd_asserted() may be called before the `mtk_dp->drm_dev` pointer is assigned in mtk_dp_bridge_attach(). Specifically it can be called via this callpath: - mtk_edp_wait_hpd_asserted - [panel probe] - dp_aux_ep_probe Using "drm" level prints anywhere in this callpath causes a NULL pointer dereference. Change the error message directly in mtk_dp_wait_hpd_asserted() to dev_err() to avoid this. Also change the error messages in mtk_dp_parse_capabilities(), which is called by mtk_dp_wait_hpd_asserted(). While touching these prints, also add the error code to them to make future debugging easier.
CVE-2025-38242	In the Linux kernel, the following vulnerability has been resolved: mm: userfaultfd: fix race of userfaultfd_move and swap cache This commit fixes two kinds of races, they may have different results: Barry reported a BUG_ON in commit c50f8e6053b0, we may see the same BUG_ON if the filemap lookup returned NULL and folio is added to swap cache after that. If another kind of race is triggered (folio changed after lookup) we may see RSS counter is corrupted: [406.893936] BUG: Bad rss-counter state mm:ffff0000c5a9ddc0 type:MM_ANONPAGES val:-1 [406.894071] BUG: Bad rss-counter state mm:ffff0000c5a9ddc0 type:MM_SHMEMPAGES val:1 Because the folio is being accounted to the wrong VMA. I'm not sure if there will be any data corruption though, seems no. The issues above are critical already. On seeing a swap entry PTE, userfaultfd_move does a lockless swap cache lookup, and tries to move the found folio to the faulting vma. Currently, it relies on checking the PTE value to ensure that the moved folio still belongs to the src swap entry and that no new folio has been added to the swap cache, which turns out to be unreliable. While working and reviewing the swap table series with Barry, following existing races are observed and reproduced [1]: In the example below, move_pages_pte is moving src_pte to dst_pte, where src_pte is a swap entry PTE holding swap entry S1, and S1 is not in the swap cache: CPU1 CPU2 userfaultfd_move move_pages_pte() entry = pte_to_swp_entry(orig_src_pte); // Here it got entry = S1 ... < interrupted> ... <swapon src_pte, alloc and use folio A> // folio A is a new allocated folio // and get installed into src_pte <frees swap entry S1> // src_pte now points to folio A, S1 // has swap count == 0, it can be freed // by folio_swap_swap or swap // allocator's reclaim. <try to swap out another folio B> // folio B is a folio in another VMA. <put folio B to swap cache using S1 > // S1 is freed, folio B can use it // for swap out with no problem. ... folio = filemap_get_folio(S1) // Got folio B here !!! ... < interrupted again> ... <swapon folio B and free S1> // Now S1 is free to be used again. <swapon src_pte & folio A using S1> // Now src_pte is a swap entry PTE // holding S1 again. folio_trylock(folio) move_swap_pte double_pt_lock is_pte_pages_stable // Check passed because src_pte == S1 folio_move_anon_rmap(...) // Moved invalid folio B here !!! The race window is very short and requires multiple collisions of multiple rare events, so it's very unlikely to happen, but with a deliberately constructed reproducer and increased time window, it can be reproduced easily. This can be fixed by checking if the folio returned by filemap is the valid swap cache folio after acquiring the folio lock. Another similar race is possible: filemap_get_folio may return NULL, but folio (A) could be swapped in and then swapped out again using the same swap entry after the lookup. In such a case, folio (A) may remain in the swap cache, so it must be moved too: CPU1 CPU2 userfaultfd_move move_pages_pte() entry = pte_to_swp_entry(orig_src_pte); // Here it got entry = S1, and S1 is not in swap cache folio = filemap_get ---truncated---

CVE ID	Description
CVE-2025-38244	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix potential deadlock when reconnecting channels Fix cifs_signal_cifs_for_reconnect() to take the correct lock order and prevent the following deadlock from happening ===== WARNING: possible circular locking dependency detected 6.16.0-rc3-build2+ #1301 Tainted: G S W ----- cifs/6055 is trying to acquire lock: ffff88810ad56038 (&tcp_ses->srv_lock){+.+.}-{3:3}, at: cifs_signal_cifs_for_reconnect+0x134/0x200 but task is already holding lock: ffff888119c64330 (&ret_buf->chan_lock){+.+.}-{3:3}, at: cifs_signal_cifs_for_reconnect+0xc0/0x200 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #2 (&ret_buf->chan_lock){+.+.}-{3:3}: validate_chain+0x1cf/0x270 __lock_acquire+0x60e/0x780 lock_acquire.part.0+0xb4/0x1f0 _raw_spin_lock+0x2f/0x40 cifs_setup_session+0x81/0x4b0 cifs_get_smb_ses+0x771/0x900 cifs_mount_get_session+0x7e/0x170 cifs_mount+0x92/0x2d0 cifs_smb3_do_mount+0x161/0x460 smb3_get_tree+0x55/0x90 vfs_get_tree+0x46/0x180 do_new_mount+0x1b0/0x2e0 path_mount+0x6ee/0x740 do_mount+0x98/0xe0 __do_sys_mount+0x148/0x180 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x76/0x7e -> #1 (&ret_buf->ses_lock){+.+.}-{3:3}: validate_chain+0x1cf/0x270 __lock_acquire+0x60e/0x780 lock_acquire.part.0+0xb4/0x1f0 _raw_spin_lock+0x2f/0x40 cifs_match_super+0x101/0x320 sget+0xab/0x270 cifs_smb3_do_mount+0x1e0/0x460 smb3_get_tree+0x55/0x90 vfs_get_tree+0x46/0x180 do_new_mount+0x1b0/0x2e0 path_mount+0x6ee/0x740 do_mount+0x98/0xe0 __do_sys_mount+0x148/0x180 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x76/0x7e -> #0 (&tcp_ses->srv_lock){+.+.}-{3:3}: check_noncircular+0x95/0xc0 check_prev_add+0x115/0x2f0 validate_chain+0x1cf/0x270 __lock_acquire+0x60e/0x780 lock_acquire.part.0+0xb4/0x1f0 _raw_spin_lock+0x2f/0x40 cifs_signal_cifs_for_reconnect+0x134/0x200 __cifs_reconnect+0x8f/0x500 cifs_handle_standard+0x112/0x280 cifs_demultiplex_thread+0x64d/0xbc0 kthread+0x2f7/0x310 ret_from_fork+0x2a/0x230 ret_from_fork_asm+0x1a/0x30 other info that might help us debug this: Chain exists of: &tcp_ses->srv_lock --> &ret_buf->ses_lock --> &ret_buf->chan_lock Possible unsafe locking scenario: CPU0 CPU1 ----- lock(&ret_buf->chan_lock); lock(&ret_buf->ses_lock); lock(&ret_buf->chan_lock); lock(&tcp_ses->srv_lock); *** DEADLOCK *** 3 locks held by cifs/6055: #0: ffffffff857de398 (&cifs_tcp_ses_lock){+.+.}-{3:3}, at: cifs_signal_cifs_for_reconnect+0x7b/0x200 #1: ffff888119c64060 (&ret_buf->ses_lock){+.+.}-{3:3}, at: cifs_signal_cifs_for_reconnect+0x9c/0x200 #2: ffff888119c64330 (&ret_buf->chan_lock){+.+.}-{3:3}, at: cifs_signal_cifs_for_reconnect+0xc0/0x200

CVE ID	Description
CVE-2025-38245	In the Linux kernel, the following vulnerability has been resolved: atm: Release atm_dev_mutex after removing procfs in atm_dev_deregister(). syzbot reported a warning below during atm_dev_register(). [0] Before creating a new device and procfs/sysfs for it, atm_dev_register() looks up a duplicated device by __atm_dev_lookup(). These operations are done under atm_dev_mutex. However, when removing a device in atm_dev_deregister(), it releases the mutex just after removing the device from the list that __atm_dev_lookup() iterates over. So, there will be a small race window where the device does not exist on the device list but procfs/sysfs are still not removed, triggering the splat. Let's hold the mutex until procfs/sysfs are removed in atm_dev_deregister(). [0]: proc_dir_entry 'atm/atmtcp:0' already registered WARNING: CPU: 0 PID: 5919 at fs/proc/generic.c:377 proc_register+0x455/0x5f0 fs/proc/generic.c:377 Modules linked in: CPU: 0 UID: 0 PID: 5919 Comm: syz-executor284 Not tainted 6.16.0-rc2-syzkaller-00047-g52da431bf03b #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 RIP: 0010:proc_register+0x455/0x5f0 fs/proc/generic.c:377 Code: 48 89 f9 48 c1 e9 03 80 3c 01 00 0f 85 a2 01 00 00 48 8b 44 24 10 48 c7 c7 20 c0 c2 8b 48 8b b0 d8 00 00 00 e8 0c 02 1c ff 90 <0f> 0b 90 90 48 c7 c7 80 f2 82 8e e8 0b de 23 09 48 8b 4c 24 28 48 RSP: 0018:ffff9000466fa30 EFLAGS: 00010282 RAX: 0000000000000000 RBX: 0000000000000000 RCX: ffffffff817ae248 RDX: ffff888026280000 RSI: ffffffff817ae255 RDI: 0000000000000001 RBP: ffff8880232bed48 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000001 R12: ffff888076ed2140 R13: dffffc0000000000 R14: ffff888078a61340 R15: ffffd100edda444 FS: 00007f38b3b0c6c0(0000) GS:ffff888124753000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f38b3bdf953 CR3: 0000000076d58000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> proc_create_data+0xbe/0x110 fs/proc/generic.c:585 atm_proc_dev_register+0x112/0x1e0 net/atm/proc.c:361 atm_dev_register+0x46d/0x890 net/atm/resources.c:113 atmtcp_create+0x77/0x210 drivers/atm/atmtcp.c:369 atmtcp_attach drivers/atm/atmtcp.c:403 [inline] atmtcp_ioctl+0x2f9/0xd60 drivers/atm/atmtcp.c:464 do_vcc_ioctl+0x12c/0x930 net/atm/ioctl.c:159 sock_do_ioctl+0x115/0x280 net/socket.c:1190 sock_ioctl+0x227/0x6b0 net/socket.c:1311 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __x64_sys_ioctl+0x18b/0x210 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x4c0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f38b3b74459 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 51 18 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f38b3b0c198 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff817ae248 RBX: 00007f38b3bfe318 RCX: 00007f38b3b74459 RDX: 0000000000000000 RSI: 0000000000000618 RDI: 0000000000000005 RBP: 00007f38b3bfe310 R08: 65732f636f72702f R09: 65732f636f72702f R10: 65732f636f72702f R11: 0000000000000246 R12: 00007f38b3bcb0ac R13: 00007f38b3b0c1a0 R14: 0000200000000200 R15: 00007f38b3bcb03b </TASK>

CVE ID	Description
CVE-2025-38246	In the Linux kernel, the following vulnerability has been resolved: bnxt: properly flush XDP redirect lists We encountered following crash when testing a XDP_REDIRECT feature in production: [56251.579676] list_add corruption. next->prev should be prev (ffff93120dd40f30), but was ffff9301ef3a6740. (next=ffff93120dd 40f30). [56251.601413] -----[cut here]----- [56251.611357] kernel BUG at lib/list_debug.c:29! [56251.621082] Oops: invalid opcode: 0000 [#1] PREEMPT SMP NOPTI [56251.632073] CPU: 111 UID: 0 PID: 0 Comm: swapper/111 Kdump: loaded Tainted: P O 6.12.33-cloudflare-2025.6. 3 #1 [56251.653155] Tainted: [P]=PROPRIETARY_MODULE, [O]=OOT_MODULE [56251.663877] Hardware name: MiTAC GC68B-B8032-G11P6-GPU/S8032GM-HE-CFR, BIOS V7.020.B10-sig 01/22/2025 [56251.682626] RIP: 0010: __list_add_valid_or_report+0x4b/0xa0 [56251.693203] Code: 0e 48 c7 c7 68 e7 d9 97 e8 42 16 fe ff 0f 0b 48 8b 52 08 48 39 c2 74 14 48 89 f1 48 c7 c7 90 e7 d9 97 48 89 c6 e8 25 16 fe ff <0f> 0b 4c 8b 02 49 39 f0 74 14 48 89 d1 48 c7 c7 e8 e7 d9 97 4c 89 [56251.725811] RSP: 0018:ffff93120dd40b80 EFLAGS: 00010246 [56251.736094] RAX: 0000000000000075 RBX: ffff9301e6bba9d8 RCX: 0000000000000000 [56251.748260] RDX: 0000000000000000 RSI: ffff9149afda0b80 RDI: ffff9149afda0b80 [56251.760349] RBP: ffff9131e49c8000 R08: 0000000000000000 R09: ffff93120dd40a18 [56251.772382] R10: ffff9159cf2ce1a8 R11: 0000000000000003 R12: ffff911a80850000 [56251.784364] R13: ffff93120fbc7000 R14: 0000000000000010 R15: ffff9139e7510e40 [56251.796278] FS: 0000000000000000(0000) GS: ffff9149afd80000(0000) knlGS:0000000000000000 [56251.809133] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [56251.819561] CR2: 00007f5e85e6f300 CR3: 00000038b85e2006 CR4: 00000000000770ef0 [56251.831365] PKRU: 55555554 [56251.838653] Call Trace: [56251.845560] <IRQ> [56251.851943] cpu_map_enqueue.cold+0x5/0xa [56251.860243] xdp_do_redirect +0x2d9/0x480 [56251.868388] bnxt_rx_xdp+0x1d8/0x4c0 [bnxt_en] [56251.877028] bnxt_rx_pkt +0x5f7/0x19b0 [bnxt_en] [56251.885665] ? cpu_max_write+0x1e/0x100 [56251.893510] ? srso_alias_return_thunk+0x5/0xfbef5 [56251.902276] __bnxt_poll_work+0x190/0x340 [bnxt_en] [56251.911058] bnxt_poll+0xab/0x1b0 [bnxt_en] [56251.919041] ? srso_alias_return_thunk +0x5/0xfbef5 [56251.927568] ? srso_alias_return_thunk+0x5/0xfbef5 [56251.935958] ? srso_alias_return_thunk+0x5/0xfbef5 [56251.944250] __napi_poll+0x2b/0x160 [56251.951155] bpf_trampoline_6442548651+0x79/0x123 [56251.959262] __napi_poll+0x5/0x160 [56251.966037] net_rx_action+0x3d2/0x880 [56251.973133] ? srso_alias_return_thunk+0x5/0xfbef5 [56251.981265] ? srso_alias_return_thunk+0x5/0xfbef5 [56251.989262] ? __hrtimer_run_queues+0x162/0x2a0 [56251.996967] ? srso_alias_return_thunk+0x5/0xfbef5 [56252.004875] ? srso_alias_return_thunk +0x5/0xfbef5 [56252.012673] ? bnxt_msix+0x62/0x70 [bnxt_en] [56252.019903] handle_softirqs +0xcf/0x270 [56252.026650] irq_exit_rcu+0x67/0x90 [56252.032933] common_interrupt+0x85/0xa0 [56252.039498] <IRQ> [56252.044246] <TASK> [56252.048935] asm_common_interrupt+0x26/0x40 [56252.055727] RIP: 0010:cpuidle_enter_state+0xb8/0x420 [56252.063305] Code: dc 01 00 00 e8 f9 79 3b ff e8 64 f7 ff ff 49 89 c5 0f 1f 44 00 00 31 ff e8 a5 32 3a ff 45 84 ff 0f 85 ae 01 00 00 fb 45 85 f6 <0f> 88 88 01 00 00 48 8b 04 24 49 63 ce 4c 89 ea 48 6b f1 68 48 29 [56252.088911] RSP: 0018:ffff93120c97fe98 EFLAGS: 00000202 [56252.096912] RAX: ffff9149afd80000 RBX: ffff9141d3a72800 RCX: 0000000000000000 [56252.106844] RDX: 00003329176c6b98 RSI: fffffe36db3fdc7 RDI: 0000000000000000 [56252.116733] RBP: 0000000000000002 R08: 0000000000000002 R09: 000000000000004e [56252.126652] R10: ffff9149afdb30c4 R11: 071c71c71c71c71c R12: ffffffff985ff860 [56252.136637] R13: 00003329176c6b98 R14: 0000000000000002 R15: 0000000000000000 [56252.146667] ? cpuidle_enter_state+0xab/0x420 [56252.153909] cpuidle_enter+0x2d/0x40 [56252.160360] do_idle+0x176/0x1c0 [56252.166456] --- truncated---

CVE ID	Description
CVE-2025-38248	In the Linux kernel, the following vulnerability has been resolved: bridge: mcast: Fix use-after-free during router port configuration The bridge maintains a global list of ports behind which a multicast router resides. The list is consulted during forwarding to ensure multicast packets are forwarded to these ports even if the ports are not member in the matching MDB entry. When per-VLAN multicast snooping is enabled, the per-port multicast context is disabled on each port and the port is removed from the global router port list: # ip link add name br1 up type bridge vlan_filtering 1 mcast_snooping 1 # ip link add name dummy1 up master br1 type dummy # ip link set dev dummy1 type bridge_slave mcast_router 2 \$ bridge -d mdb show grep router router ports on br1: dummy1 # ip link set dev br1 type bridge mcast_vlan_snooping 1 \$ bridge -d mdb show grep router However, the port can be re-added to the global list even when per-VLAN multicast snooping is enabled: # ip link set dev dummy1 type bridge_slave mcast_router 0 # ip link set dev dummy1 type bridge_slave mcast_router 2 \$ bridge -d mdb show grep router router ports on br1: dummy1 Since commit 4b30ae9adb04 ("net: bridge: mcast: re-implement br_multicast_{enable, disable}_port functions"), when per-VLAN multicast snooping is enabled, multicast disablement on a port will disable the per-{port, VLAN} multicast contexts and not the per-port one. As a result, a port will remain in the global router port list even after it is deleted. This will lead to a use-after-free [1] when the list is traversed (when adding a new port to the list, for example): # ip link del dev dummy1 # ip link add name dummy2 up master br1 type dummy # ip link set dev dummy2 type bridge_slave mcast_router 2 Similarly, stale entries can also be found in the per-VLAN router port list. When per-VLAN multicast snooping is disabled, the per-{port, VLAN} contexts are disabled on each port and the port is removed from the per-VLAN router port list: # ip link add name br1 up type bridge vlan_filtering 1 mcast_snooping 1 mcast_vlan_snooping 1 # ip link add name dummy1 up master br1 type dummy # bridge vlan add vid 2 dev dummy1 # bridge vlan global set vid 2 dev br1 mcast_snooping 1 # bridge vlan set vid 2 dev dummy1 mcast_router 2 \$ bridge vlan global show dev br1 vid 2 grep router router ports: dummy1 # ip link set dev br1 type bridge mcast_vlan_snooping 0 \$ bridge vlan global show dev br1 vid 2 grep router However, the port can be re-added to the per-VLAN list even when per-VLAN multicast snooping is disabled: # bridge vlan set vid 2 dev dummy1 mcast_router 0 # bridge vlan set vid 2 dev dummy1 mcast_router 2 \$ bridge vlan global show dev br1 vid 2 grep router router ports: dummy1 When the VLAN is deleted from the port, the per-{port, VLAN} multicast context will not be disabled since multicast snooping is not enabled on the VLAN. As a result, the port will remain in the per-VLAN router port list even after it is no longer member in the VLAN. This will lead to a use-after-free [2] when the list is traversed (when adding a new port to the list, for example): # ip link add name dummy2 up master br1 type dummy # bridge vlan add vid 2 dev dummy2 # bridge vlan del vid 2 dev dummy1 # bridge vlan set vid 2 dev dummy2 mcast_router 2 Fix these issues by removing the port from the relevant (global or per-VLAN) router port list in br_multicast_port_ctx_deinit(). The function is invoked during port deletion with the per-port multicast context and during VLAN deletion with the per-{port, VLAN} multicast context. Note that deleting the multicast router timer is not enough as it only takes care of the temporary multicast router states (1 or 3) and not the permanent one (2). [1] BUG: KASAN: slab-out-of-bounds in br_multicast_add_router.part.0+0x3f1/0x560 Write of size 8 at addr ffff888004a67328 by task ip/384 [...] Call Trace: <TASK> dump_stack ---truncated---
CVE-2025-38249	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Fix out-of-bounds read in snd_usb_get_audioformat_uac3() In snd_usb_get_audioformat_uac3(), the length value returned from snd_usb_ctl_msg() is used directly for memory allocation without validation. This length is controlled by the USB device. The allocated buffer is cast to a uac3_cluster_header_descriptor and its fields are accessed without verifying that the buffer is large enough. If the device returns a smaller than expected length, this leads to an out-of-bounds read. Add a length check to ensure the buffer is large enough for uac3_cluster_header_descriptor.

CVE ID	Description
CVE-2025-38250	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_core: Fix use-after-free in vhci_flush() syzbot reported use-after-free in vhci_flush() without repro. [0] From the splat, a thread close()d a vhci file descriptor while its device was being used by ioctl() on another thread. Once the last fd refcnt is released, vhci_release() calls hci_unregister_dev(), hci_free_dev(), and kfree() for struct vhci_data, which is set to hci_dev->dev->driver_data. The problem is that there is no synchronisation after unlinking hdev from hci_dev_list in hci_unregister_dev(). There might be another thread still accessing the hdev which was fetched before the unlink operation. We can use SRCU for such synchronisation. Let's run hci_dev_reset() under SRCU and wait for its completion in hci_unregister_dev(). Another option would be to restore hci_dev->destruct(), which was removed in commit 587ae086f6e4 ("Bluetooth: Remove unused hci-destruct cb"). However, this would not be a good solution, as we should not run hci_unregister_dev() while there are in-flight ioctl() requests, which could lead to another data-race KCSAN splat. Note that other drivers seem to have the same problem, for example, virtbt_remove(). [0]: BUG: KASAN: slab-use-after-free in skb_queue_empty_lockless include/linux/skbuff.h:1891 [inline] BUG: KASAN: slab-use-after-free in skb_queue_purge_reason+0x99/0x360 net/core/skbuff.c:3937 Read of size 8 at addr ffff88807cb8d858 by task syz.1.219/6718 CPU: 1 UID: 0 PID: 6718 Comm: syz.1.219 Not tainted 6.16.0-rc1-syzkaller-00196-g08207f42d3ff #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 Call Trace: <TASK> dump_stack_lvl+0x189/0x250 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xd2/0x2b0 mm/kasan/report.c:521 kasan_report+0x118/0x150 mm/kasan/report.c:634 skb_queue_empty_lockless include/linux/skbuff.h:1891 [inline] skb_queue_purge_reason+0x99/0x360 net/core/skbuff.c:3937 skb_queue_purge include/linux/skbuff.h:3368 [inline] vhci_flush+0x44/0x50 drivers/bluetooth/hci_vhci.c:69 hci_dev_do_reset net/bluetooth/hci_core.c:552 [inline] hci_dev_reset+0x420/0x5c0 net/bluetooth/hci_core.c:592 sock_do_ioctl+0xd9/0x300 net/socket.c:1190 sock_ioctl+0x576/0x790 net/socket.c:1311 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xf9/0x170 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fcf5b98e929 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fcf5c7b9038 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffffda RBX: 00007fcf5bbb6160 RCX: 00007fcf5b98e929 RDX: 0000000000000000 RSI: 00000000400448cb RDI: 0000000000000009 RBP: 00007fcf5ba10b39 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000000 R14: 00007fcf5bbb6160 R15: 00007ffd6353d528 </TASK> Allocated by task 6535: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x93/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x230/0x3d0 mm/slub.c:4359 kmalloc_noprof include/linux/slab.h:905 [inline] kzalloc_noprof include/linux/slab.h:1039 [inline] vhci_open+0x57/0x360 drivers/bluetooth/hci_vhci.c:635 misc_open+0x2bc/0x330 drivers/char/misc.c:161 chrdev_open+0x4c9/0x5e0 fs/char_dev.c:414 do_dentry_open+0xdf0/0x1970 fs/open.c:964 vfs_open+0x3b/0x340 fs/open.c:1094 do_open fs/namei.c:3887 [inline] path_openat+0x2ee5/0x3830 fs/namei.c:---truncated---
CVE-2025-38251	In the Linux kernel, the following vulnerability has been resolved: atm: clip: prevent NULL deref in clip_push() Blamed commit missed that vcc_destroy_socket() calls clip_push() with a NULL skb. If clip_devs is NULL, clip_push() then crashes when reading skb->truesize.
CVE-2025-38253	In the Linux kernel, the following vulnerability has been resolved: HID: wacom: fix crash in wacom_aes_battery_handler() Commit fd2a9b29dc9c ("HID: wacom: Remove AES power_supply after extended inactivity") introduced wacom_aes_battery_handler() which is scheduled as a delayed work (aes_battery_work). In wacom_remove(), aes_battery_work is not canceled. Consequently, if the device is removed while aes_battery_work is still pending, then hard crashes or "Oops: general protection fault..." are experienced when wacom_aes_battery_handler() is finally called. E.g., this happens with built-in USB devices after resume from hibernate when aes_battery_work was still pending at the time of hibernation. So, take care to cancel aes_battery_work in wacom_remove().

CVE ID	Description
CVE-2025-38255	In the Linux kernel, the following vulnerability has been resolved: lib/group_cpus: fix NULL pointer dereference from group_cpus_evenly() While testing null_blk with configfs, echo 0 > poll_queues will trigger following panic: BUG: kernel NULL pointer dereference, address: 0000000000000010 Oops: Oops: 0000 [#1] SMP NOPTI CPU: 27 UID: 0 PID: 920 Comm: bash Not tainted 6.15.0-02023-gadbb95c8696-dirty #1238 PREEMPT(undef) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 RIP: 0010: __bitmap_or+0x48/0x70 Call Trace: <TASK> __group_cpus_evenly+0x822/0x8c0 group_cpus_evenly+0x2d9/0x490 blk_mq_map_queues+0x1e/0x110 null_map_queues+0xc9/0x170 [null_blk] blk_mq_update_queue_map+0xdb/0x160 blk_mq_update_nr_hw_queues+0x22b/0x560 nullb_update_nr_hw_queues+0x71/0xf0 [null_blk] nullb_device_poll_queues_store+0xa4/0x130 [null_blk] configfs_write_iter+0x109/0x1d0 vfs_write+0x26e/0x6f0 ksys_write+0x79/0x180 __x64_sys_write+0x1d/0x30 x64_sys_call+0x45c4/0x45f0 do_syscall_64+0xa5/0x240 entry_SYSCALL_64_after_hwframe+0x76/0x7e Root cause is that numgrps is set to 0, and ZERO_SIZE_PTR is returned from kcalloc(), and later ZERO_SIZE_PTR will be dereferenced. Fix the problem by checking numgrps first in group_cpus_evenly(), and return NULL directly if numgrps is zero. [yukuai3@huawei.com: also fix the non-SMP version]
CVE-2025-38257	In the Linux kernel, the following vulnerability has been resolved: s390/pkey: Prevent overflow in size calculation for memdup_user() Number of apqn target list entries contained in 'nr_apqns' variable is determined by userspace via an ioctl call so the result of the product in calculation of size passed to memdup_user() may overflow. In this case the actual size of the allocated area and the value describing it won't be in sync leading to various types of unpredictable behaviour later. Use a proper memdup_array_user() helper which returns an error if an overflow is detected. Note that it is different from when nr_apqns is initially zero - that case is considered valid and should be handled in subsequent pkey_handler implementations. Found by Linux Verification Center (linuxtesting.org).
CVE-2025-38258	In the Linux kernel, the following vulnerability has been resolved: mm/damon/sysfs-schemes: free old damon_sysfs_scheme_filter->memcg_path on write memcg_path_store() assigns a newly allocated memory buffer to filter->memcg_path, without deallocating the previously allocated and assigned memory buffer. As a result, users can leak kernel memory by continuously writing a data to memcg_path DAMOS sysfs file. Fix the leak by deallocating the previously set memory buffer.
CVE-2025-38259	In the Linux kernel, the following vulnerability has been resolved: ASoC: codecs: wcd9335: Fix missing free of regulator supplies Driver gets and enables all regulator supplies in probe path (wcd9335_parse_dt() and wcd9335_power_on_reset()), but does not cleanup in final error paths and in unbind (missing remove() callback). This leads to leaked memory and unbalanced regulator enable count during probe errors or unbind. Fix this by converting entire code into devm_regulator_bulk_get_enable() which also greatly simplifies the code.

CVE ID	Description
CVE-2025-38260	In the Linux kernel, the following vulnerability has been resolved: btrfs: handle csum tree error with rescue=ibadroots correctly [BUG] There is syzbot based reproducer that can crash the kernel, with the following call trace: (With some debug output added) DEBUG: rescue=ibadroots parsed BTRFS: device fsid 14d642db-7b15-43e4-81e6-4b8fac6a25f8 devid 1 transid 8 /dev/loop0 (7:0) scanned by repro (1010) BTRFS info (device loop0): first mount of filesystem 14d642db-7b15-43e4-81e6-4b8fac6a25f8 BTRFS info (device loop0): using blake2b (blake2b-256-generic) checksum algorithm BTRFS info (device loop0): using free-space-tree BTRFS warning (device loop0): checksum verify failed on logical 5312512 mirror 1 wanted 0xb043382657aede36608fd3386d6b001692ff406164733d94e2d9a180412c6003 found 0x810ceb2bacb7f0f9eb2bf3b2b15c02af867cb35ad450898169f3b1f0bd818651 level 0 DEBUG: read tree root path failed for tree csum, ret=-5 BTRFS warning (device loop0): checksum verify failed on logical 5328896 mirror 1 wanted 0x51be4e8b303da58e6340226815b70e3a93592dac3f30dd510c7517454de8567a found 0x51be4e8b303da58e634022a315b70e3a93592dac3f30dd510c7517454de8567a level 0 BTRFS warning (device loop0): checksum verify failed on logical 5292032 mirror 1 wanted 0x1924ccd683be9efc2fa98582ef58760e3848e9043db8649ee382681e220cdee4 found 0x0cb6184f6e8799d9f8cb335dcd1d1832da1071d12290dab3b85b587ecacca6e level 0 process 'repro' launched './file2' with NULL argv: empty string added DEBUG: no csum root, idatacsums=0 ibadroots=134217728 Oops: general protection fault, probably for non-canonical address 0xdffffc0000000041: 0000 [#1] SMP KASAN NOPTI KASAN: null-ptr-deref in range [0x0000000000000208-0x000000000000020f] CPU: 5 UID: 0 PID: 1010 Comm: repro Tainted: G OE 6.15.0-custom+ #249 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS unknown 02/02/2022 RIP: 0010:btrfs_lookup_csum+0x93/0x3d0 [btrfs] Call Trace: <TASK> btrfs_lookup_bio_sums+0x47a/0xdf0 [btrfs] btrfs_submit_bbio+0x43e/0x1a80 [btrfs] submit_one_bio+0xde/0x160 [btrfs] btrfs_readahead+0x498/0x6a0 [btrfs] read_pages+0x1c3/0xb20 page_cache_ra_order+0x4b5/0xc20 filemap_get_pages+0x2d3/0x19e0 filemap_read+0x314/0xde0 __kernel_read+0x35b/0x900 bprm_execve+0x62e/0x1140 do_execveat_common.isra.0+0x3fc/0x520 __x64_sys_execveat+0xdc/0x130 do_syscall_64+0x54/0x1d0 entry_SYSCALL_64_after_hwframe+0x76/0x7e ---[end trace 0000000000000000]--- [CAUSE] Firstly the fs has a corrupted csum tree root, thus to mount the fs we have to go "ro,rescue=ibadroots" mount option. Normally with that mount option, a bad csum tree root should set BTRFS_FS_STATE_NO_DATA_CSUMS flag, so that any future data read will ignore csum search. But in this particular case, we have the following call trace that caused NULL csum root, but not setting BTRFS_FS_STATE_NO_DATA_CSUMS: load_global_roots_objectid(): ret = btrfs_search_slot(); /* Succeeded */ btrfs_item_key_to_cpu() found = true; /* We found the root item for csum tree. */ root = read_tree_root_path(); if (IS_ERR(root)) { if (!btrfs_test_opt(fs_info, IGNOREBADROOTS)) /* * Since we have rescue=ibadroots mount option, * @ret is still 0. */ break; if (!found ret) { /* @found is true, @ret is 0, error handling for csum * tree is skipped. */ } This means we completely skipped to set BTRFS_FS_STATE_NO_DATA_CSUMS if the csum tree is corrupted, which results unexpected later csum lookup. [FIX] If read_tree_root_path() failed, always populate @ret to the error number. As at the end of the function, we need @ret to determine if we need to do the extra error handling for csum tree.

CVE ID	Description
CVE-2025-38261	In the Linux kernel, the following vulnerability has been resolved: riscv: save the SR_SUM status over switches When threads/tasks are switched we need to ensure the old execution's SR_SUM state is saved and the new thread has the old SR_SUM state restored. The issue was seen under heavy load especially with the syz-stress tool running, with crashes as follows in schedule_tail: Unable to handle kernel access to user memory without uaccess routines at virtual address 000000002749f0d0 Oops [#1] Modules linked in: CPU: 1 PID: 4875 Comm: syz-executor.0 Not tainted 5.12.0-rc2-syzkaller-00467-g0d7588ab9ef9 #0 Hardware name: riscv-virtio,qemu (DT) epc : schedule_tail+0x72/0xb2 kernel/sched/core.c:4264 ra : task_pid_vnr include/linux/sched.h:1421 [inline] ra : schedule_tail+0x70/0xb2 kernel/sched/core.c:4264 epc : fffffffe00008c8b ra : fffffffe00008c8a sp : fffffffe025d17ec gp : fffffffe005d25378 tp : fffffffe00f0d0000 t0 : 0000000000000000 t1 : 0000000000000001 t2 : 000000000000f4240 s0 : fffffffe025d17ee s1 : 000000002749f0d0 a0 : 000000000000002a a1 : 0000000000000003 a2 : 1ffffffc0cfac500 a3 : fffffffe0000c80c a4 : 5ae9db91c19bbe00 a5 : 0000000000000000 a6 : 0000000000f00000 a7 : fffffffe000082eba s2 : 0000000000040000 s3 : fffffffe00eef96c0 s4 : fffffffe022c77fe0 s5 : 0000000000004000 s6 : fffffffe067d74e00 s7 : fffffffe067d74850 s8 : fffffffe067d73e18 s9 : fffffffe067d74e00 s10 : fffffffe00eef96e8 s11: 000000ae6cdf8368 t3 : 5ae9db91c19bbe00 t4 : fffffffc4043cafb2 t5 : fffffffc4043cafb2 t6 : 0000000000040000 status: 000000000000120 badaddr: 000000002749f0d0 cause: 000000000000000f Call Trace: [<ffffffe00008c8b0>] schedule_tail+0x72/0xb2 kernel/sched/core.c:4264 [<ffffffe000005570>] ret_from_exception+0x0/0x14 Dumping ftrace buffer: (ftrace buffer empty) ---[end trace b5f8f9231dc87dda]--- The issue comes from the put_user() in schedule_tail (kernel/sched/core.c) doing the following: asm linkage __visible void schedule_tail(struct task_struct *prev) { ... if (current->set_child_tid) put_user(task_pid_vnr(current), current->set_child_tid); ... } the put_user() macro causes the code sequence to come out as follows: 1: __enable_user_access() 2: reg = task_pid_vnr(current); 3: *current->set_child_tid = reg; 4: __disable_user_access() The problem is that we may have a sleeping function as argument which could clear SR_SUM causing the panic above. This was fixed by evaluating the argument of the put_user() macro outside the user-enabled section in commit 285a76bb2cf5 ("riscv: evaluate put_user() arg before enabling user access")" In order for riscv to take advantage of unsafe_get/put_XXX() macros and to avoid the same issue we had with put_user() and sleeping functions we must ensure code flow can go through switch_to() from within a region of code with SR_SUM enabled and come back with SR_SUM still enabled. This patch addresses the problem allowing future work to enable full use of unsafe_get/put_XXX() macros without needing to take a CSR bit flip cost on every access. Make switch_to() save and restore SR_SUM.
CVE-2025-38262	In the Linux kernel, the following vulnerability has been resolved: tty: serial: uartlite: register uart driver in init When two instances of uart devices are probing, a concurrency race can occur. If one thread calls uart_register_driver function, which first allocates and assigns memory to 'uart_state' member of uart_driver structure, the other instance can bypass uart driver registration and call ulite_assign. This calls uart_add_one_port, which expects the uart driver to be fully initialized. This leads to a kernel panic due to a null pointer dereference: [8.143581] BUG: kernel NULL pointer dereference, address: 00000000000002b8 [8.156982] #PF: supervisor write access in kernel mode [8.156984] #PF: error_code(0x0002) - not-present page [8.156986] PGD 0 P4D 0 ... [8.180668] RIP: 0010:mutex_lock+0x19/0x30 [8.188624] Call Trace: [8.188629] ? __die_body.cold+0x1a/0x1f [8.195260] ? page_fault_oops+0x15c/0x290 [8.209183] ? __irq_resolve_mapping+0x47/0x80 [8.209187] ? exc_page_fault+0x64/0x140 [8.209190] ? asm_exc_page_fault+0x22/0x30 [8.209196] ? mutex_lock+0x19/0x30 [8.223116] uart_add_one_port+0x60/0x440 [8.223122] ? proc_tty_register_driver+0x43/0x50 [8.223126] ? tty_register_driver+0x1ca/0x1e0 [8.246250] ulite_probe+0x357/0x4b0 [uartlite] To prevent it, move uart driver registration in to init function. This will ensure that uart_driver is always registered when probe function is called.

CVE ID	Description
CVE-2025-38263	In the Linux kernel, the following vulnerability has been resolved: bcache: fix NULL pointer in cache_set_flush() 1. LINE#1794 - LINE#1887 is some codes about function of bch_cache_set_alloc(). 2. LINE#2078 - LINE#2142 is some codes about function of register_cache_set(). 3. register_cache_set() will call bch_cache_set_alloc() in LINE#2098. 1794 struct cache_set *bch_cache_set_alloc(struct cache_sb *sb) 1795 { ... 1860 if (!c->devices = kcalloc(c->nr_uuids, sizeof(void *), GFP_KERNEL)) 1861 mempool_init_slab_pool(&c->search, 32, bch_search_cache) 1862 mempool_init_kmalloc_pool(&c->bio_meta, 2, 1863 sizeof(struct bbio) + sizeof(struct bio_vec) * 1864 bucket_pages(c)) 1865 mempool_init_kmalloc_pool(&c->fill_iter, 1, iter_size) 1866 bioset_init(&c->bio_split, 4, offsetof(struct bbio, bio), 1867 BIOSSET_NEED_BVECS BIOSSET_NEED_RESCUER) 1868 !(c->uuids = alloc_bucket_pages(GFP_KERNEL, c)) 1869 ! (c->moving_gc_wq = alloc_workqueue("bcache_gc", 1870 WQ_MEM_RECLAIM, 0)) 1871 bch_journal_alloc(c) 1872 bch_btree_cache_alloc(c) 1873 bch_open_buckets_alloc(c) 1874 bch_bset_sort_state_init(&c->sort, ilog2(c->btree_pages))) 1875 goto err; ^^^^^^ 1876 ... 1883 return c; 1884 err: 1885 bch_cache_set_unregister(c); ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ 1886 return NULL; 1887 } ... 2078 static const char *register_cache_set(struct cache *ca) 2079 { ... 2098 c = bch_cache_set_alloc(&ca->sb); 2099 if (!c) 2100 return err; ^^^^^^^^^^ ... 2128 ca->set = c; 2129 ca->set->cache[ca->sb.nr_this_dev] = ca; ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ ... 2138 return NULL; 2139 err: 2140 bch_cache_set_unregister(c); 2141 return err; 2142 } (1) If LINE#1860 - LINE#1874 is true, then do 'goto err'(LINE#1875) and call bch_cache_set_unregister()(LINE#1885). (2) As (1) return NULL(LINE#1886), LINE#2098 - LINE#2100 would return. (3) As (2) has returned, LINE#2128 - LINE#2129 would do *not* give the value to c->cache[], it means that c->cache[] is NULL. LINE#1624 - LINE#1665 is some codes about function of cache_set_flush(). As (1), in LINE#1885 call bch_cache_set_unregister() --> bch_cache_set_stop() --> closure_queue() --> cache_set_flush() (as below LINE#1624) 1624 static void cache_set_flush(struct closure *cl) 1625 { ... 1654 for_each_cache(ca, c, i) 1655 if (ca->alloc_thread) ^^ 1656 kthread_stop(ca->alloc_thread); ... 1665 } (4) In LINE#1655 ca is NULL(see (3)) in cache_set_flush() then the kernel crash occurred as below: [846.712887] bcache: register_cache() error drbd6: cannot allocate memory [846.713242] bcache: register_bcache() error : failed to register device [846.713336] bcache: cache_set_free() Cache set 2f84bdc1-498a-4f2f-98a7-01946bf54287 unregistered [846.713768] BUG: unable to handle kernel NULL pointer dereference at 000000000000009f8 [846.714790] PGD 0 P4D 0 [846.715129] Oops: 0000 [#1] SMP PTI [846.715472] CPU: 19 PID: 5057 Comm: kworker/19:16 Kdump: loaded Tainted: G OE ----- - - 4.18.0-147.5.1.el8_1.Ses.3.x86_64 #1 [846.716082] Hardware name: ESPAN GI-25212/X11DPL-i, BIOS 2.1 06/15/2018 [846.716451] Workqueue: events cache_set_flush [bcache] [846.716808] RIP: 0010:cache_set_flush+0xc9/0x1b0 [bcache] [846.717155] Code: 00 4c 89 a5 b0 03 00 00 48 8b 85 68 f6 ff ff a8 08 0f 84 88 00 00 00 31 db 66 83 bd 3c f7 ff ff 00 48 8b 85 48 ff ff 74 28 <48> 8b b8 f8 09 00 0 ---truncated---
CVE-2025-38264	In the Linux kernel, the following vulnerability has been resolved: nvme-tcp: sanitize request list handling Validate the request in nvme_tcp_handle_r2t() to ensure it's not part of any list, otherwise a malicious R2T PDU might inject a loop in request list processing.
CVE-2025-38265	In the Linux kernel, the following vulnerability has been resolved: serial: jsm: fix NPE during jsm_uart_port_init No device was set which caused serial_base_ctrl_add to crash. BUG: kernel NULL pointer dereference, address: 0000000000000050 Oops: Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 16 UID: 0 PID: 368 Comm: (udev-worker) Not tainted 6.12.25-amd64 #1 Debian 6.12.25-1 RIP: 0010:serial_base_ctrl_add+0x96/0x120 Call Trace: <TASK> serial_core_register_port+0x1a0/0x580 ? __setup_irq+0x39c/0x660 ? __kmalloc_cache_noprof+0x111/0x310 jsm_uart_port_init+0xe8/0x180 [jsm] jsm_probe_one+0x1f4/0x410 [jsm] local_pci_probe+0x42/0x90 pci_device_probe+0x22f/0x270 really_probe+0xdb/0x340 ? pm_runtime_barrier+0x54/0x90 ? __pfx_driver_attach+0x10/0x10 __driver_probe_device+0x78/0x110 driver_probe_device+0x1f/0xa0 __driver_attach+0xba/0x1c0 bus_for_each_dev+0x8c/0xe0 bus_add_driver+0x112/0x1f0 driver_register+0x72/0xd0 jsm_init_module+0x36/0xff0 [jsm] ? __pfx_jsm_init_module+0x10/0x10 [jsm] do_one_initcall+0x58/0x310 do_init_module+0x60/0x230 Tested with Digi Neo PCIe 8 port card.
CVE-2025-38269	In the Linux kernel, the following vulnerability has been resolved: btrfs: exit after state insertion failure at btrfs_convert_extent_bit() If insert_state() state failed it returns an error pointer and we call extent_io_tree_panic() which will trigger a BUG() call. However if CONFIG_BUG is disabled, which is an uncommon and exotic scenario, then we fallthrough and call cache_state() which will dereference the error pointer, resulting in an invalid memory access. So jump to the 'out' label after calling extent_io_tree_panic(), it also makes the code more clear besides dealing with the exotic scenario where CONFIG_BUG is disabled.
CVE-2025-38272	In the Linux kernel, the following vulnerability has been resolved: net: dsa: b53: do not enable EEE on bcm63xx BCM63xx internal switches do not support EEE, but provide multiple RGMII ports where external PHYs may be connected. If one of these PHYs are EEE capable, we may try to enable EEE for the MACs, which then hangs the system on access of the (non-existent) EEE registers. Fix this by checking if the switch actually supports EEE before attempting to configure it.

CVE ID	Description
CVE-2025-38274	In the Linux kernel, the following vulnerability has been resolved: fpga: fix potential null pointer deref in fpga_mgr_test_img_load_sgt() fpga_mgr_test_img_load_sgt() allocates memory for sgt using kunit_kzalloc() however it does not check if the allocation failed. It then passes sgt to sg_alloc_table(), which passes it to __sg_alloc_table(). This function calls memset() on sgt in an attempt to zero it out. If the allocation fails then sgt will be NULL and the memset will trigger a NULL pointer dereference. Fix this by checking the allocation with KUNIT_ASSERT_NOT_ERR_OR_NULL().
CVE-2025-38275	In the Linux kernel, the following vulnerability has been resolved: phy: qcom-qmp-usb: Fix an NULL vs IS_ERR() bug The qmp_usb_iomap() helper function currently returns the raw result of devm_ioremap() for non-exclusive mappings. Since devm_ioremap() may return a NULL pointer and the caller only checks error pointers with IS_ERR(), NULL could bypass the check and lead to an invalid dereference. Fix the issue by checking if devm_ioremap() returns NULL. When it does, qmp_usb_iomap() now returns an error pointer via IOMEM_ERR_PTR(-ENOMEM), ensuring safe and consistent error handling.
CVE-2025-38277	In the Linux kernel, the following vulnerability has been resolved: mtd: nand: ecc-mxic: Fix use of uninitialized variable ret If ctx->steps is zero, the loop processing ECC steps is skipped, and the variable ret remains uninitialized. It is later checked and returned, which leads to undefined behavior and may cause unpredictable results in user space or kernel crashes. This scenario can be triggered in edge cases such as misconfigured geometry, ECC engine misuse, or if ctx->steps is not validated after initialization. Initialize ret to zero before the loop to ensure correct and safe behavior regardless of the ctx->steps value. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-38278	In the Linux kernel, the following vulnerability has been resolved: octeontx2-pf: QOS: Refactor TC_HTB_LEAF_DEL_LAST callback This patch addresses below issues, 1. Active traffic on the leaf node must be stopped before its send queue is reassigned to the parent. This patch resolves the issue by marking the node as 'Inner'. 2. During a system reboot, the interface receives TC_HTB_LEAF_DEL and TC_HTB_LEAF_DEL_LAST callbacks to delete its HTB queues. In the case of TC_HTB_LEAF_DEL_LAST, although the same send queue is reassigned to the parent, the current logic still attempts to update the real number of queues, leading to below warnings New queues can't be registered after device unregistration. WARNING: CPU: 0 PID: 6475 at net/core/net-sysfs.c:1714 netdev_queue_update_kobjects+0x1e4/0x200

CVE ID	Description
CVE-2025-38279	In the Linux kernel, the following vulnerability has been resolved: bpf: Do not include stack ptr register in precision backtracking bookkeeping Yi Lai reported an issue ([1]) where the following warning appears in kernel dmesg: [60.643604] verifier backtracking bug [60.643635] WARNING: CPU: 10 PID: 2315 at kernel/bpf/verifier.c:4302 __mark_chain_precision+0x3a6c/0x3e10 [60.648428] Modules linked in: bpf_testmod(OE) [60.650471] CPU: 10 UID: 0 PID: 2315 Comm: test_progs Tainted: G OE 6.15.0-rc4-gef11287f8289-dirty #327 PREEMPT(full) [60.654385] Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE [60.656682] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org 04/01/2014 [60.660475] RIP: 0010: __mark_chain_precision+0x3a6c/0x3e10 [60.662814] Code: 5a 30 84 89 ea e8 c4 d9 01 00 80 3d 3e 7d d8 04 00 0f 85 60 fa ff ff c6 05 31 7d d8 04 01 48 c7 c7 00 58 30 84 e8 c4 06 a5 ff <0f> 0b e9 46 fa ff ff 48 ... [60.668720] RSP: 0018:ffff888116cc7298 EFLAGS: 00010246 [60.671075] RAX: 54d70e82dfd31900 RBX: ffff888115b65e20 RCX: 0000000000000000 [60.673659] RDX: 0000000000000001 RSI: 0000000000000004 RDI: 00000000ffffffff [60.676241] RBP: 0000000000000400 R08: ffff8881f6f23bd3 R09: 1ffff1103ede477a [60.678787] R10: dffffc0000000000 R11: ffffd103ede477b R12: ffff888115b60ae8 [60.681420] R13: 1ffff11022b6cbc4 R14: 00000000ffffffff R15: 0000000000000001 [60.684030] FS: 00007fc2aed80c0(0000) GS:ffff88826fa8a000(0000) knlGS:0000000000000000 [60.686837] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000000005003 [60.689027] CR2: 000056325369e000 CR3: 000000011088b002 CR4: 0000000000370ef0 [60.691623] Call Trace: [60.692821] <TASK> [60.693960] ? __pfx_verbose+0x10/0x10 [60.695656] ? __pfx_disasm_kfunc_name+0x10/0x10 [60.697495] check_cond_jump_op+0x16f7/0x39b0 [60.699237] do_check+0x58fa/0xab10 ... Further analysis shows the warning is at line 4302 as below: 4294 /* static subprog call instruction, which 4295 * means that we are exiting current subprog, 4296 * so only r1-r5 could be still requested as 4297 * precise, r0 and r6-r10 or any stack slot in 4298 * the current frame should be zero by now 4299 */ 4300 if (bt_reg_mask(bt) & ~BPF_REGMASK_ARGS) { 4301 verbose(env, "BUG regs %x\n", bt_reg_mask(bt)); 4302 WARN_ONCE(1, "verifier backtracking bug"); 4303 return -EFAULT; 4304 } With the below test (also in the next patch): __used __naked static void __bpf_jump_r10(void) { asm volatile ("r2 = 2314885393468386424 ll;" "goto +0;" "if r2 <= r10 goto +3;" "if r1 >= -1835016 goto +0;" "if r2 <= 8 goto +0;" "if r3 <= 0 goto +0;" "exit;" ::: __clobber_all); } SEC("?raw_tp") __naked void bpf_jump_r10(void) { asm volatile ("r3 = 0 ll;" "call __bpf_jump_r10;" "r0 = 0;" "exit;" ::: __clobber_all); } The following is the verifier failure log: 0: (18) r3 = 0x0; R3_w=0 2: (85) call pc+2 caller: R10=fp0 callee: frame1: R1=ctx() R3_w=0 R10=fp0 5: frame1: R1=ctx() R3_w=0 R10=fp0; asm volatile (" \ @ verifier_precision.c:184 5: (18) r2 = 0x20202000256c6c78; frame1: R2_w=0x20202000256c6c78 7: (05) goto pc+0 8: (bd) if r2 <= r10 goto pc+3; frame1: R2_w=0x20202000256c6c78 R10=fp0 9: (35) if r1 >= 0xffe3fff8 goto pc+0; frame1: R1=ctx() 10: (b5) if r2 <= 0x8 goto pc+0 mark_precise: frame1: last_idx 10 first_idx 0 subseq_idx -1 mark_precise: frame1: regs=r2 stack= before 9: (35) if r1 >= 0xffe3fff8 goto pc+0 mark_precise: frame1: regs=r2 stack= before 8: (bd) if r2 <= r10 goto pc+3 mark_preci ---truncated---
CVE-2025-38280	In the Linux kernel, the following vulnerability has been resolved: bpf: Avoid __bpf_prog_ret0_warn when jit fails syzkaller reported an issue: WARNING: CPU: 3 PID: 217 at kernel/bpf/core.c:2357 __bpf_prog_ret0_warn+0xa/0x20 kernel/bpf/core.c:2357 Modules linked in: CPU: 3 UID: 0 PID: 217 Comm: kworker/u32:6 Not tainted 6.15.0-rc4-syzkaller-00040-g8bac8898fe39 RIP: 0010: __bpf_prog_ret0_warn+0xa/0x20 kernel/bpf/core.c:2357 Call Trace: <TASK> bpf_dispatcher_nop_func include/linux/bpf.h:1316 [inline] __bpf_prog_run include/linux/filter.h:718 [inline] bpf_prog_run include/linux/filter.h:725 [inline] cls_bpf_classify+0x74a/0x1110 net/sched/cls_bpf.c:105 ... When creating bpf program, 'fp->jit_requested' depends on bpf_jit_enable. This issue is triggered because of CONFIG_BPF_JIT_ALWAYS_ON is not set and bpf_jit_enable is set to 1, causing the arch to attempt JIT the prog, but jit failed due to FAULT_INJECTION. As a result, incorrectly treats the program as valid, when the program runs it calls `__bpf_prog_ret0_warn` and triggers the WARN_ON_ONCE(1).
CVE-2025-38282	In the Linux kernel, the following vulnerability has been resolved: kernfs: Relax constraint in draining guard The active reference lifecycle provides the break/unbreak mechanism but the active reference is not truly active after unbreak -- callers don't use it afterwards but it's important for proper pairing of kn->active counting. Assuming this mechanism is in place, the WARN check in kernfs_should_drain_open_files() is too sensitive -- it may transiently catch those (rightful) callers between kernfs_unbreak_active_protection() and kernfs_put_active() as found out by Chen Ridong: kernfs_remove_by_name_ns kernfs_get_active // active=1 __kernfs_remove // active=0x80000002 kernfs_drain ... wait_event //waiting (active == 0x80000001) kernfs_break_active_protection // active = 0x80000001 // continue kernfs_unbreak_active_protection // active = 0x80000002 ... kernfs_should_drain_open_files // warning occurs kernfs_put_active To avoid the false positives (mind panic_on_warn) remove the check altogether. (This is meant as quick fix, I think active reference break/unbreak may be simplified with larger rework.)

CVE ID	Description
CVE-2025-38283	In the Linux kernel, the following vulnerability has been resolved: hisi_acc_vfio_pci: bugfix live migration function without VF device driver If the VF device driver is not loaded in the Guest OS and we attempt to perform device data migration, the address of the migrated data will be NULL. The live migration recovery operation on the destination side will access a null address value, which will cause access errors. Therefore, live migration of VMs without added VF device drivers does not require device data migration. In addition, when the queue address data obtained by the destination is empty, device queue recovery processing will not be performed.
CVE-2025-38285	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix WARN() in get_bpf_raw_tp_regs syzkaller reported an issue: WARNING: CPU: 3 PID: 5971 at kernel/trace/bpf_trace.c:1861 get_bpf_raw_tp_regs+0xa4/0x100 kernel/trace/bpf_trace.c:1861 Modules linked in: CPU: 3 UID: 0 PID: 5971 Comm: syz-executor205 Not tainted 6.15.0-rc5-syzkaller-00038-g707df3375124 #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:get_bpf_raw_tp_regs+0xa4/0x100 kernel/trace/bpf_trace.c:1861 RSP: 0018:ffffc90003636fa8 EFLAGS: 00010293 RAX: 0000000000000000 RBX: 0000000000000003 RCX: ffffffff81c6bc4c RDX: ffff888032efc880 RSI: ffffffff81c6bc83 RDI: 0000000000000005 RBP: ffff88806a730860 R08: 0000000000000005 R09: 0000000000000003 R10: 0000000000000004 R11: 0000000000000000 R12: 0000000000000004 R13: 0000000000000001 R14: ffff90003637008 R15: 0000000000000090 FS: 0000000000000000(0000) GS:ffff8880d6cdf000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f7baee09130 CR3: 0000000029f5a000 CR4: 000000000352ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ____bpf_get_stack_raw_tp kernel/trace/bpf_trace.c:1934 [inline] bpf_get_stack_raw_tp+0x24/0x160 kernel/trace/bpf_trace.c:1931 bpf_prog_ec3b2eefa702d8d3+0x43/0x47 bpf_dispatcher_nop_func include/linux/bpf.h:1316 [inline] __bpf_prog_run include/linux/filter.h:718 [inline] bpf_prog_run include/linux/filter.h:725 [inline] __bpf_trace_run kernel/trace/bpf_trace.c:2363 [inline] bpf_trace_run3+0x23f/0x5a0 kernel/trace/bpf_trace.c:2405 __bpf_trace_mmap_lock_acquire_returned+0xfc/0x140 include/trace/events/mmap_lock.h:47 __traceiter_mmap_lock_acquire_returned+0x79/0xc0 include/trace/events/mmap_lock.h:47 __do_trace_mmap_lock_acquire_returned include/trace/events/mmap_lock.h:47 [inline] trace_mmap_lock_acquire_returned include/trace/events/mmap_lock.h:47 [inline] __mmap_lock_do_trace_acquire_returned+0x138/0x1f0 mm/mmap_lock.c:35 __mmap_lock_trace_acquire_returned include/linux/mmap_lock.h:36 [inline] mmap_read_trylock include/linux/mmap_lock.h:204 [inline] stackmap_get_build_id_offset+0x535/0x6f0 kernel/bpf/stackmap.c:157 __bpf_get_stack+0x307/0xa10 kernel/bpf/stackmap.c:483 ____bpf_get_stack kernel/bpf/stackmap.c:499 [inline] bpf_get_stack+0x32/0x40 kernel/bpf/stackmap.c:496 ____bpf_get_stack_raw_tp kernel/trace/bpf_trace.c:1941 [inline] bpf_get_stack_raw_tp+0x124/0x160 kernel/trace/bpf_trace.c:1931 bpf_prog_ec3b2eefa702d8d3+0x43/0x47 Tracepoint like trace_mmap_lock_acquire_returned may cause nested call as the corner case show above, which will be resolved with more general method in the future. As a result, WARN_ON_ONCE will be triggered. As Alexei suggested, remove the WARN_ON_ONCE first.
CVE-2025-38286	In the Linux kernel, the following vulnerability has been resolved: pinctrl: at91: Fix possible out-of-boundary access at91_gpio_probe() doesn't check that given OF alias is not available or something went wrong when trying to get it. This might have consequences when accessing gpio_chips array with that value as an index. Note, that BUG() can be compiled out and hence won't actually perform the required checks.
CVE-2025-38289	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Avoid potential ndlp use-after-free in dev_loss_tmo_callback Smatch detected a potential use-after-free of an ndlp object in dev_loss_tmo_callback during driver unload or fatal error handling. Fix by reordering code to avoid potential use-after-free if initial nodelist reference has been previously removed.
CVE-2025-38290	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix node corruption in ar->arvifs list In current WLAN recovery code flow, ath12k_core_halt() only reinitializes the "arvifs" list head. This will cause the list node immediately following the list head to become an invalid list node. Because the prev of that node still points to the list head "arvifs", but the next of the list head "arvifs" no longer points to that list node. When a WLAN recovery occurs during the execution of a vif removal, and it happens before the spin_lock_bh(&ar->data_lock) in ath12k_mac_vdev_delete(), list_del() will detect the previously mentioned situation, thereby triggering a kernel panic. The fix is to remove and reinitialize all vif list nodes from the list head "arvifs" during WLAN halt. The reinitialization is to make the list nodes valid, ensuring that the list_del() in ath12k_mac_vdev_delete() can execute normally. Call trace: __list_del_entry_valid_or_report+0xd4/0x100 (P) ath12k_mac_remove_link_interface.isra.0+0xf8/0x2e4 [ath12k] ath12k_scan_vdev_clean_work+0x40/0x164 [ath12k] cfg80211_wiphy_work+0xfc/0x100 process_one_work+0x164/0x2d0 worker_thread+0x254/0x380 kthread+0xfc/0x100 ret_from_fork+0x10/0x20 The change is mostly copied from the ath11k patch: https://lore.kernel.org/all/20250320053145.3445187-1-quic_stonez@quicinc.com/ Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.4.1-00199-QCAHKSUWPL_SILICONZ-1

CVE ID	Description
CVE-2025-38292	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix invalid access to memory In ath12k_dp_rx_msdu_coalesce(), rxcb is fetched from skb and boolean is_continuation is part of rxcb. Currently, after freeing the skb, the rxcb->is_continuation accessed again which is wrong since the memory is already freed. This might lead use-after-free error. Hence, fix by locally defining bool is_continuation from rxcb, so that after freeing skb, is_continuation can be used. Compile tested only.
CVE-2025-38293	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix node corruption in ar->arvifs list In current WLAN recovery code flow, ath11k_core_halt() only reinitializes the "arvifs" list head. This will cause the list node immediately following the list head to become an invalid list node. Because the prev of that node still points to the list head "arvifs", but the next of the list head "arvifs" no longer points to that list node. When a WLAN recovery occurs during the execution of a vif removal, and it happens before the spin_lock_bh(&ar->data_lock) in ath11k_mac_op_remove_interface(), list_del() will detect the previously mentioned situation, thereby triggering a kernel panic. The fix is to remove and reinitialize all vif list nodes from the list head "arvifs" during WLAN halt. The reinitialization is to make the list nodes valid, ensuring that the list_del() in ath11k_mac_op_remove_interface() can execute normally. Call trace: __list_del_entry_valid_or_report+0xb8/0xd0 ath11k_mac_op_remove_interface+0xb0/0x27c [ath11k] drv_remove_interface+0x48/0x194 [mac80211] ieee80211_do_stop+0x6e0/0x844 [mac80211] ieee80211_stop+0x44/0x17c [mac80211] __dev_close_many+0xac/0x150 __dev_change_flags+0x194/0x234 dev_change_flags+0x24/0x6c devinet_ioctl+0x3a0/0x670 inet_ioctl+0x200/0x248 sock_do_ioctl+0x60/0x118 sock_ioctl+0x274/0x35c __arm64_sys_ioctl+0xac/0xf0 invoke_syscall+0x48/0x114 ... Tested-on: QCA6698AQ hw2.1 PCI WLAN.HSP.1.1-04591-QCAHSPSWPL_V1_V2_SILICONZ_IOE-1
CVE-2025-38295	In the Linux kernel, the following vulnerability has been resolved: perf/amlogic: Replace smp_processor_id() with raw_smp_processor_id() in meson_ddr_pmu_create() The Amlogic DDR PMU driver meson_ddr_pmu_create() function incorrectly uses smp_processor_id(), which assumes disabled preemption. This leads to kernel warnings during module loading because meson_ddr_pmu_create() can be called in a preemptible context. Following kernel warning and stack trace: [31.745138] [T2289] BUG: using smp_processor_id() in preemptible [00000000] code: (udev-worker)/2289 [31.745154] [T2289] caller is debug_smp_processor_id+0x28/0x38 [31.745172] [T2289] CPU: 4 UID: 0 PID: 2289 Comm: (udev-worker) Tainted: GW 6.14.0-0-MANJARO-ARM #1 59519addcbca6ba8de735e151fd7b9e97aac7ff0 [31.745181] [T2289] Tainted: [W]=WARN [31.745183] [T2289] Hardware name: Hardkernel ODROID-N2Plus (DT) [31.745188] [T2289] Call trace: [31.745191] [T2289] show_stack+0x28/0x40 (C) [31.745199] [T2289] dump_stack_lvl+0x4c/0x198 [31.745205] [T2289] dump_stack+0x20/0x50 [31.745209] [T2289] check_preemption_disabled+0xec/0xf0 [31.745213] [T2289] debug_smp_processor_id+0x28/0x38 [31.745216] [T2289] meson_ddr_pmu_create+0x200/0x560 [meson_ddr_pmu_g12 8095101c49676ad138d9961e3eddaee10acca7bd] [31.745237] [T2289] g12_ddr_pmu_probe+0x20/0x38 [meson_ddr_pmu_g12 8095101c49676ad138d9961e3eddaee10acca7bd] [31.745246] [T2289] platform_probe+0x98/0xe0 [31.745254] [T2289] really_probe+0x144/0x3f8 [31.745258] [T2289] __driver_probe_device+0xb8/0x180 [31.745261] [T2289] driver_probe_device+0x54/0x268 [31.745264] [T2289] __driver_attach+0x11c/0x288 [31.745267] [T2289] bus_for_each_dev+0xfc/0x160 [31.745274] [T2289] driver_attach+0x34/0x50 [31.745277] [T2289] bus_add_driver+0x160/0x2b0 [31.745281] [T2289] driver_register+0x78/0x120 [31.745285] [T2289] __platform_driver_register+0x30/0x48 [31.745288] [T2289] init_module+0x30/0xfe0 [meson_ddr_pmu_g12 8095101c49676ad138d9961e3eddaee10acca7bd] [31.745298] [T2289] do_one_initcall+0x11c/0x438 [31.745303] [T2289] do_init_module+0x68/0x228 [31.745311] [T2289] load_module+0x118c/0x13a8 [31.745315] [T2289] __arm64_sys_finit_module+0x274/0x390 [31.745320] [T2289] invoke_syscall+0x74/0x108 [31.745326] [T2289] el0_svc_common+0x90/0xf8 [31.745330] [T2289] do_el0_svc+0x2c/0x48 [31.745333] [T2289] el0_svc+0x60/0x150 [31.745337] [T2289] el0t_64_sync_handler+0x80/0x118 [31.745341] [T2289] el0t_64_sync+0x1b8/0x1c0 Changes replaces smp_processor_id() with raw_smp_processor_id() to ensure safe CPU ID retrieval in preemptible contexts.
CVE-2025-38298	In the Linux kernel, the following vulnerability has been resolved: EDAC/skx_common: Fix general protection fault After loading i10nm_edac (which automatically loads skx_edac_common), if unload only i10nm_edac, then reload it and perform error injection testing, a general protection fault may occur: mce: [Hardware Error]: Machine check events logged Oops: general protection fault Workqueue: events mce_gen_pool_process RIP: 0010:string+0x53/0xe0 ... Call Trace: <TASK> ? die_addr+0x37/0x90 ? exc_general_protection+0x1e7/0x3f0 ? asm_exc_general_protection+0x26/0x30 ? string+0x53/0xe0 vsnprintf+0x23e/0x4c0 snprintf+0x4d/0x70 skx_adxl_decode+0x16a/0x330 [skx_edac_common] skx_mce_check_error.part.0+0xf8/0x220 [skx_edac_common] skx_mce_check_error+0x17/0x20 [skx_edac_common] ... The issue arose because the variable 'adxl_component_count' (inside skx_edac_common), which counts the ADXL components, was not reset. During the reloading of i10nm_edac, the count was incremented by the actual number of ADXL components again, resulting in a count that was double the real number of ADXL components. This led to an out-of-bounds reference to the ADXL component array, causing the general protection fault above. Fix this issue by resetting the 'adxl_component_count' in adxl_put(), which is called during the unloading of {skx,i10nm}_edac.

CVE ID	Description
CVE-2025-38299	In the Linux kernel, the following vulnerability has been resolved: ASoC: mediatek: mt8195: Set ETDM1/2 IN/OUT to COMP_DUMMY() ETDM2_IN_BE and ETDM1_OUT_BE are defined as COMP_EMPTY(), in the case the codec dai_name will be null. Avoid a crash if the device tree is not assigning a codec to these links. [1.179936] Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 [1.181065] Mem abort info: [1.181420] ESR = 0x0000000096000004 [1.181892] EC = 0x25: DABT (current EL), IL = 32 bits [1.182576] SET = 0, FnV = 0 [1.182964] EA = 0, S1PTW = 0 [1.183367] FSC = 0x04: level 0 translation fault [1.183983] Data abort info: [1.184406] ISV = 0, ISS = 0x000000004, ISS2 = 0x00000000 [1.185097] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [1.185766] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [1.186439] [0000000000000000] user address but active_mm is swapper [1.187239] Internal error: Oops: 0000000096000004 [#1] PREEMPT SMP [1.188029] Modules linked in: [1.188420] CPU: 7 UID: 0 PID: 70 Comm: kworker/u32:1 Not tainted 6.14.0-rc4-next-20250226+ #85 [1.189515] Hardware name: Radxa NIO 12L (DT) [1.190065] Workqueue: events_unbound deferred_probe_work_func [1.190808] pstate: 40400009 (nZcv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) [1.191683] pc : __pi_strerror+0x24/0x140 [1.192170] lr : mt8195_mt6359_soc_card_probe+0x224/0x7b0 [1.192854] sp : ffff800083473970 [1.193271] x29: ffff800083473a10 x28: 0000000000001008 x27: 0000000000000002 [1.194168] x26: ffff800082408960 x25: ffff800082417db0 x24: ffff800082417d88 [1.195065] x23: 000000000000001e x22: ffff800082dbf480 x21: ffff800082dc07b8 [1.195961] x20: 0000000000000000 x19: 0000000000000013 x18: 00000000ffffff [1.196858] x17: 000000040044ffff x16: 005000f2b5503510 x15: 0000000000000006 [1.197755] x14: ffff800082407af0 x13: 6e6f69737265766e x12: 692d6b636f6c6374 [1.198651] x11: 0000000000000002 x10: ffff80008240b920 x9: 0000000000000018 [1.199547] x8: 0101010101010101 x7: 0000000000000000 x6: 0000000000000000 [1.200443] x5: 0000000000000000 x4: 8080808080000000 x3: 303933383978616d [1.201339] x2: 0000000000000000 x1: ffff80008240b920 x0: 0000000000000000 [1.202236] Call trace: [1.202545] __pi_strerror+0x24/0x140 (P) [1.203029] mtk_soundcard_common_probe+0x3bc/0x5b8 [1.203644] platform_probe+0x70/0xe8 [1.204106] really_probe+0xc8/0x3a0 [1.204556] __driver_probe_device+0x84/0x160 [1.205104] driver_probe_device+0x44/0x130 [1.205630] __device_attach_driver+0xc4/0x170 [1.206189] bus_for_each_drv+0x8c/0xf8 [1.206672] __device_attach+0xa8/0x1c8 [1.207155] device_initial_probe+0x1c/0x30 [1.207681] bus_probe_device+0xb0/0xc0 [1.208165] deferred_probe_work_func+0xa4/0x100 [1.208747] process_one_work+0x158/0x3e0 [1.209254] worker_thread+0x2c4/0x3e8 [1.209727] kthread+0x134/0x1f0 [1.210136] ret_from_fork+0x10/0x20 [1.210589] Code: 54000401 b50002c6 d503201f f86a6803 (f8408402) [1.211355] ---[end trace 0000000000000000]---
CVE-2025-38300	In the Linux kernel, the following vulnerability has been resolved: crypto: sun8i-ce-cipher - fix error handling in sun8i_ce_cipher_prepare() Fix two DMA cleanup issues on the error path in sun8i_ce_cipher_prepare(): 1) If dma_map_sg() fails for areq->dst, the device driver would try to free DMA memory it has not allocated in the first place. To fix this, on the "theend_sgs" error path, call dma_unmap only if the corresponding dma map was successful. 2) If the dma_map_single() call for the IV fails, the device driver would try to free an invalid DMA memory address on the "theend_iv" path: -----[cut here]----- DMA-API: sun8i-ce 1904000.crypto: device driver tries to free an invalid DMA memory address WARNING: CPU: 2 PID: 69 at kernel/dma/debug.c:968 check_unmap+0x123c/0x1b90 Modules linked in: skcipher_example(O+) CPU: 2 UID: 0 PID: 69 Comm: 1904000.crypto- Tainted: G O 6.15.0-rc3+ #24 PREEMPT Tainted: [O]=OOT_MODULE Hardware name: OrangePi Zero2 (DT) pc : check_unmap+0x123c/0x1b90 lr : check_unmap+0x123c/0x1b90 ... Call trace: check_unmap+0x123c/0x1b90 (P) debug_dma_unmap_page+0xac/0xc0 dma_unmap_page_attr+0x1f4/0x5fc sun8i_ce_cipher_do_one+0x1bd4/0x1f40 crypto_pump_work+0x334/0x6e0 kthread_worker_fn+0x21c/0x438 kthread+0x374/0x664 ret_from_fork+0x10/0x20 ---[end trace 0000000000000000]--- To fix this, check for !dma_mapping_error() before calling dma_unmap_single() on the "theend_iv" path.
CVE-2025-38303	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: eir: Fix possible crashes on eir_create_adv_data eir_create_adv_data may attempt to add EIR_FLAGS and EIR_TX_POWER without checking if that would fit.
CVE-2025-38304	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix NULL pointer deference on eir_get_service_data The len parameter is considered optional so it can be NULL so it cannot be used for skipping to next entry of EIR_SERVICE_DATA.

CVE ID	Description
CVE-2025-38305	In the Linux kernel, the following vulnerability has been resolved: ptp: remove ptp->n_vclocks check logic in ptp_vclock_in_use() There is no disagreement that we should check both ptp->is_virtual_clock and ptp->n_vclocks to check if the ptp virtual clock is in use. However, when we acquire ptp->n_vclocks_mux to read ptp->n_vclocks in ptp_vclock_in_use(), we observe a recursive lock in the call trace starting from n_vclocks_store(). ===== WARNING: possible recursive locking detected 6.15.0-rc6 #1 Not tainted ----- syz.0.1540/13807 is trying to acquire lock: ffff888035a24868 (&ptp->n_vclocks_mux){+.+.}-{4:4}, at: ptp_vclock_in_use drivers/ptp/ptp_private.h:103 [inline] ffff888035a24868 (&ptp->n_vclocks_mux){+.+.}-{4:4}, at: ptp_clock_unregister+0x21/0x250 drivers/ptp/ptp_clock.c:415 but task is already holding lock: ffff888030704868 (&ptp->n_vclocks_mux){+.+.}-{4:4}, at: n_vclocks_store+0xf1/0x6d0 drivers/ptp/ptp_sysfs.c:215 other info that might help us debug this: Possible unsafe locking scenario: CPU0 ---- lock(&ptp->n_vclocks_mux); lock(&ptp->n_vclocks_mux); *** DEADLOCK *** ===== The best way to solve this is to remove the logic that checks ptp->n_vclocks in ptp_vclock_in_use(). The reason why this is appropriate is that any path that uses ptp->n_vclocks must unconditionally check if ptp->n_vclocks is greater than 0 before unregistering vclocks, and all functions are already written this way. And in the function that uses ptp->n_vclocks, we already get ptp->n_vclocks_mux before unregistering vclocks. Therefore, we need to remove the redundant check for ptp->n_vclocks in ptp_vclock_in_use() to prevent recursive locking.
CVE-2025-38307	In the Linux kernel, the following vulnerability has been resolved: ASoC: Intel: avs: Verify content returned by parse_int_array() The first element of the returned array stores its length. If it is 0, any manipulation beyond the element at index 0 ends with null-ptr-deref.
CVE-2025-38310	In the Linux kernel, the following vulnerability has been resolved: seg6: Fix validation of nexthop addresses The kernel currently validates that the length of the provided nexthop address does not exceed the specified length. This can lead to the kernel reading uninitialized memory if user space provided a shorter length than the specified one. Fix by validating that the provided length exactly matches the specified one.
CVE-2025-38311	In the Linux kernel, the following vulnerability has been resolved: iavf: get rid of the crit lock Get rid of the crit lock. That frees us from the error prone logic of try_locks. Thanks to netdev_lock() by Jakub it is now easy, and in most cases we were protected by it already - replace crit lock by netdev lock when it was not the case. Lockdep reports that we should cancel the work under crit_lock [splat1], and that was the scheme we have mostly followed since [1] by Slawomir. But when that is done we still got into deadlocks [splat2]. So instead we should look at the bigger problem, namely "weird locking/scheduling" of the iavf. The first step to fix that is to remove the crit lock. I will followup with a -next series that simplifies scheduling/tasks. Cancel the work without netdev lock (weird unlock +lock scheme), to fix the [splat2] (which would be totally ugly if we would kept the crit lock). Extend protected part of iavf_watchdog_task() to include scheduling more work. Note that the removed comment in iavf_reset_task() was misplaced, it belonged to inside of the removed if condition, so it's gone now. [splat1] - w/o this patch - The deadlock during VF removal: WARNING: possible circular locking dependency detected sh/3825 is trying to acquire lock: ((work_completion)((&adapter->watchdog_task)->work)){+.+.}-{0:0}, at: start_flush_work+0x1a1/0x470 but task is already holding lock: (&adapter->crit_lock){+.+.}-{4:4}, at: iavf_remove+0xd1/0x690 [iavf] which lock already depends on the new lock. [splat2] - when cancelling work under crit lock, w/o this series, see [2] for the band aid attempt WARNING: possible circular locking dependency detected sh/3550 is trying to acquire lock: ((wq_completion)iavf){+.+.}-{0:0}, at: touch_wq_lockdep_map+0x26/0x90 but task is already holding lock: (&dev->lock){+.+.}-{4:4}, at: iavf_remove+0xa6/0x6e0 [iavf] which lock already depends on the new lock. [1] fc2e6b3b132a ("iavf: Rework mutexes for better synchronisation") [2] https://github.com/pkiszczel/linux/commit/52ddbfc2bb60294083f5711a158a
CVE-2025-38312	In the Linux kernel, the following vulnerability has been resolved: fbdev: core: fbcv: avoid division by 0 in fb_cvt_hperiod() In fb_find_mode_cvt(), iff mode->refresh somehow happens to be 0x80000000, cvt.f_refresh will become 0 when multiplying it by 2 due to overflow. It's then passed to fb_cvt_hperiod(), where it's used as a divider -- division by 0 will result in kernel oops. Add a sanity check for cvt.f_refresh to avoid such overflow... Found by Linux Verification Center (linuxtesting.org) with the Svace static analysis tool.
CVE-2025-38313	In the Linux kernel, the following vulnerability has been resolved: bus: fsl-mc: fix double-free on mc_dev The blamed commit tried to simplify how the deallocations are done but, in the process, introduced a double-free on the mc_dev variable. In case the MC device is a DPRC, a new mc_bus is allocated and the mc_dev variable is just a reference to one of its fields. In this circumstance, on the error path only the mc_bus should be freed. This commit introduces back the following checkpatch warning which is a false-positive. WARNING: kfree(NULL) is safe and this check is probably not required + if (mc_bus) + kfree(mc_bus);

CVE ID	Description
CVE-2025-38319	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pp: Fix potential NULL pointer dereference in atomctrl_initialize_mc_reg_table The function atomctrl_initialize_mc_reg_table() and atomctrl_initialize_mc_reg_table_v2_2() does not check the return value of smu_atom_get_data_table(). If smu_atom_get_data_table() fails to retrieve vram_info, it returns NULL which is later dereferenced.
CVE-2025-38320	In the Linux kernel, the following vulnerability has been resolved: arm64/ptrace: Fix stack-out-of-bounds read in regs_get_kernel_stack_nth() KASAN reports a stack-out-of-bounds read in regs_get_kernel_stack_nth(). Call Trace: [97.283505] BUG: KASAN: stack-out-of-bounds in regs_get_kernel_stack_nth+0xa8/0xc8 [97.284677] Read of size 8 at addr ffff800089277c10 by task 1.sh/2550 [97.285732] [97.286067] CPU: 7 PID: 2550 Comm: 1.sh Not tainted 6.6.0+ #11 [97.287032] Hardware name: linux,dummy-virt (DT) [97.287815] Call trace: [97.288279] dump_backtrace+0xa0/0x128 [97.288946] show_stack+0x20/0x38 [97.289551] dump_stack_lvl+0x78/0xc8 [97.290203] print_address_description.constprop.0+0x84/0x3c8 [97.291159] print_report+0xb0/0x280 [97.291792] kasan_report+0x84/0xd0 [97.292421] __asan_load8+0x9c/0xc0 [97.293042] regs_get_kernel_stack_nth+0xa8/0xc8 [97.293835] process_fetch_insn+0x770/0xa30 [97.294562] kprobe_trace_func+0x254/0x3b0 [97.295271] kprobe_dispatcher+0x98/0xe0 [97.295955] kprobe_breakpoint_handler+0x1b0/0x210 [97.296774] call_break_hook+0xc4/0x100 [97.297451] brk_handler+0x24/0x78 [97.298073] do_debug_exception+0xac/0x178 [97.298785] el1_dbg+0x70/0x90 [97.299344] el1h_64_sync_handler+0xcc/0xe8 [97.300066] el1h_64_sync+0x78/0x80 [97.300699] kernel_clone+0x0/0x500 [97.301331] __arm64_sys_clone+0x70/0x90 [97.302084] invoke_syscall+0x68/0x198 [97.302746] el0_svc_common.constprop.0+0x11c/0x150 [97.303569] do_el0_svc+0x38/0x50 [97.304164] el0_svc+0x44/0x1d8 [97.304749] el0t_64_sync_handler+0x100/0x130 [97.305500] el0t_64_sync+0x188/0x190 [97.306151] [97.306475] The buggy address belongs to stack of task 1.sh/2550 [97.307461] and is located at offset 0 in frame: [97.308257] __se_sys_clone+0x0/0x138 [97.308910] [97.309241] This frame has 1 object: [97.309873] [48, 184] 'args' [97.309876] [97.310749] The buggy address belongs to the virtual mapping at [97.310749] [ffff800089270000, ffff800089279000] created by: [97.310749] dup_task_struct+0xc0/0x2e8 [97.313347] [97.313674] The buggy address belongs to the physical page: [97.314604] page: refcount:1 mapcount:0 mapping:0000000000000000 index:0x0 pfn:0x14f69a [97.315885] flags: 0x15ffffe00000000 (node=1 zone=2 lastcpuid=0xffff) [97.316957] raw: 015ffffe00000000 0000000000000000 dead000000000122 0000000000000000 [97.318207] raw: 0000000000000000 0000000000000000 00000001ffffff 0000000000000000 [97.319445] page dumped because: kasan: bad access detected [97.320371] [97.320694] Memory state around the buggy address: [97.321511] ffff800089277b00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [97.322681] ffff800089277b80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [97.323846] >ffff800089277c00: 00 00 f1 f1 f1 f1 f1 00 00 00 00 00 00 00 00 [97.325023] ^ [97.325683] ffff800089277c80: 00 00 00 00 00 00 00 00 f3 f3 f3 f3 f3 f3 [97.326856] ffff800089277d00: f3 f3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 This issue seems to be related to the behavior of some gcc compilers and was also fixed on the s390 architecture before: commit d93a855c31b7 ("s390/ptrace: Avoid KASAN false positives in regs_get_kernel_stack_nth()") As described in that commit, regs_get_kernel_stack_nth() has confirmed that 'addr' is on the stack, so reading the value at '*addr' should be allowed. Use READ_ONCE_NOCHECK() helper to silence the KASAN check for this case. [will: Use '*addr' as the argument to READ_ONCE_NOCHECK()]
CVE-2025-38321	In the Linux kernel, the following vulnerability has been resolved: smb: Log an error when close_all_cached_dirs fails Under low-memory conditions, close_all_cached_dirs() can't move the dentries to a separate list to dput() them once the locks are dropped. This will result in a "Dentry still in use" error, so add an error message that makes it clear this is what happened: [495.281119] CIFS: VFS: \otters.example.com\share Out of memory while dropping dentries [495.281595] -----[cut here]----- [495.281887] BUG: Dentry ffff888115531138{i=78,n=/} still in use (2) [unmount of cifs cifs] [495.282391] WARNING: CPU: 1 PID: 2329 at fs/dcache.c:1536 umount_check+0xc8/0xf0 Also, bail out of looping through all tcons as soon as a single allocation fails, since we're already in trouble, and kmalloc() attempts for subsequent tcons are likely to fail just like the first one did.
CVE-2025-38322	In the Linux kernel, the following vulnerability has been resolved: perf/x86/intel: Fix crash in icl_update_topdown_event() The perf_fuzzer found a hard-lockup crash on a RaptorLake machine: Oops: general protection fault, maybe for address 0xffff89aeceab400: 0000 CPU: 23 UID: 0 PID: 0 Comm: swapper/23 Tainted: [W]=WARN Hardware name: Dell Inc. Precision 9660/0VJ762 RIP: 0010:native_read_pmc+0x7/0x40 Code: cc e8 8d a9 01 00 48 89 03 5b cd cc cc cc 0f 1f ... RSP: 000:ffffb03100273de8 EFLAGS: 00010046 Call Trace: <TASK> icl_update_topdown_event+0x165/0x190 ? ktime_get+0x38/0xd0 intel_pmu_read_event+0xf9/0x210 __perf_event_read+0xf9/0x210 CPUs 16-23 are E-core CPUs that don't support the perf metrics feature. The icl_update_topdown_event() should not be invoked on these CPUs. It's a regression of commit: f9bdf1f95339 ("perf/x86/intel: Avoid disable PMU if !cpuc->enabled in sample read") The bug introduced by that commit is that the is_topdown_event() function is mistakenly used to replace the is_topdown_count() call to check if the topdown functions for the perf metrics feature should be invoked. Fix it.

CVE ID	Description
CVE-2025-38323	In the Linux kernel, the following vulnerability has been resolved: net: atm: add lec_mutex syzbot found its way in net/atm/lec.c, and found an error path in lecd_attach() could leave a dangling pointer in dev_lecp[]. Add a mutex to protect dev_lecp[] uses from lecd_attach(), lec_vcc_attach() and lec_mcast_attach(). Following patch will use this mutex for /proc/net/atm/lec. BUG: KASAN: slab-use-after-free in lecd_attach net/atm/lec.c:751 [inline] BUG: KASAN: slab-use-after-free in lane_ioctl+0x2224/0x23e0 net/atm/lec.c:1008 Read of size 8 at addr ffff88807c7b8e68 by task syz.1.17/6142 CPU: 1 UID: 0 PID: 6142 Comm: syz.1.17 Not tainted 6.16.0-rc1-syzkaller-00239-g08215f5486ec #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xcd/0x680 mm/kasan/report.c:521 kasan_report+0xe0/0x110 mm/kasan/report.c:634 lecd_attach net/atm/lec.c:751 [inline] lane_ioctl+0x2224/0x23e0 net/atm/lec.c:1008 do_vcc_ioctl+0x12c/0x930 net/atm/ioctl.c:159 sock_do_ioctl+0x118/0x280 net/socket.c:1190 sock_ioctl+0x227/0x6b0 net/socket.c:1311 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __x64_sys_ioctl+0x18e/0x210 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x4c0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> Allocated by task 6132: kasan_save_stack+0x33/0x60 mm/kasan/common.c:47 kasan_save_track+0x14/0x30 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0xaa/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __do_kmalloc_node mm/slub.c:4328 [inline] __kvmalloc_node_noprof+0x27b/0x620 mm/slub.c:5015 alloc_netdev_mqs+0xd2/0x1570 net/core/dev.c:11711 lecd_attach net/atm/lec.c:737 [inline] lane_ioctl+0x17db/0x23e0 net/atm/lec.c:1008 do_vcc_ioctl+0x12c/0x930 net/atm/ioctl.c:159 sock_do_ioctl+0x118/0x280 net/socket.c:1190 sock_ioctl+0x227/0x6b0 net/socket.c:1311 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __x64_sys_ioctl+0x18e/0x210 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x4c0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 6132: kasan_save_stack+0x33/0x60 mm/kasan/common.c:47 kasan_save_track+0x14/0x30 mm/kasan/common.c:68 kasan_save_free_info+0x3b/0x60 mm/kasan/generic.c:576 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x51/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2381 [inline] slab_free mm/slub.c:4643 [inline] kfree+0x2b4/0x4d0 mm/slub.c:4842 free_netdev+0x6c5/0x910 net/core/dev.c:11892 lecd_attach net/atm/lec.c:744 [inline] lane_ioctl+0x1ce8/0x23e0 net/atm/lec.c:1008 do_vcc_ioctl+0x12c/0x930 net/atm/ioctl.c:159 sock_do_ioctl+0x118/0x280 net/socket.c:1190 sock_ioctl+0x227/0x6b0 net/socket.c:1311 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl fs/ioctl.c:893 [inline] __x64_sys_ioctl+0x18e/0x210 fs/ioctl.c:893
CVE-2025-38324	In the Linux kernel, the following vulnerability has been resolved: mpls: Use rcu_dereference_rtnl() in mpls_route_input_rcu(). As syzbot reported [0], mpls_route_input_rcu() can be called from mpls_getroute(), where is under RTNL. net->mpls.platform_label is only updated under RTNL. Let's use rcu_dereference_rtnl() in mpls_route_input_rcu() to silence the splat. [0]: WARNING: suspicious RCU usage 6.15.0-rc7-syzkaller-00082-g5cdb2c77c4c3 #0 Not tainted ----- net/mpls/af_mpls.c:84 suspicious rcu_dereference_check() usage! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 1 lock held by syz.2.4451/17730: #0: ffffffff9012a3e8 (rtnl_mutex){+.,.}-{4:4}, at: rtnl_lock net/core/rtnetlink.c:80 [inline] #0: ffffffff9012a3e8 (rtnl_mutex){+.,.}-{4:4}, at: rtnetlink_rcv_msg+0x371/0xe90 net/core/rtnetlink.c:6961 stack backtrace: CPU: 1 UID: 0 PID: 17730 Comm: syz.2.4451 Not tainted 6.15.0-rc7-syzkaller-00082-g5cdb2c77c4c3 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x16c/0x1f0 lib/dump_stack.c:120 lockdep_rcu_suspicious+0x166/0x260 kernel/locking/lockdep.c:6865 mpls_route_input_rcu+0x1d4/0x200 net/mpls/af_mpls.c:84 mpls_getroute+0x621/0x1ea0 net/mpls/af_mpls.c:2381 rtnetlink_rcv_msg+0x3c9/0xe90 net/core/rtnetlink.c:6964 netlink_rcv_skb+0x16d/0x440 net/netlink/af_netlink.c:2534 netlink_unicast_kernel net/netlink/af_netlink.c:1313 [inline] netlink_unicast+0x53a/0x7f0 net/netlink/af_netlink.c:1339 netlink_sendmsg+0x8d1/0xdd0 net/netlink/af_netlink.c:1883 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg net/socket.c:727 [inline] ____sys_sendmsg+0xa98/0xc70 net/socket.c:2566 __sys_sendmsg+0x134/0x1d0 net/socket.c:2620 __sys_sendmmsg+0x200/0x420 net/socket.c:2709 __do_sys_sendmmsg net/socket.c:2736 [inline] __se_sys_sendmmsg net/socket.c:2733 [inline] __x64_sys_sendmmsg+0x9c/0x100 net/socket.c:2733 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x230 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f0a2818e969 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f0a28f52038 EFLAGS: 00000246 ORIG_RAX: 0000000000000133 RAX: ffffffff9012a3e8 RBX: 00007f0a283b5fa0 RCX: 00007f0a2818e969 RDX: 0000000000000003 RSI: 0000200000000000 RDI: 0000000000000003 RBP: 00007f0a28210ab1 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000000 R13: 0000000000000000 R14: 00007f0a283b5fa0 R15: 00007f0a283b5e9f268 </TASK>

CVE ID	Description
CVE-2025-38326	In the Linux kernel, the following vulnerability has been resolved: aoe: clean device rq_list in aoe_dev_downdev() An aoe device's rq_list contains accepted block requests that are waiting to be transmitted to the aoe target. This queue was added as part of the conversion to blk_mq. However, the queue was not cleaned out when an aoe device is downed which caused blk_mq_freeze_queue() to sleep indefinitely waiting for those requests to complete, causing a hang. This fix cleans out the queue before calling blk_mq_freeze_queue().
CVE-2025-38328	In the Linux kernel, the following vulnerability has been resolved: jffs2: check jffs2_prealloc_raw_node_refs() result in few other places Fuzzing hit another invalid pointer dereference due to the lack of checking whether jffs2_prealloc_raw_node_refs() completed successfully. Subsequent logic implies that the node refs have been allocated. Handle that. The code is ready for propagating the error upwards. KASAN: null-ptr-deref in range [0x0000000000000008-0x000000000000000f] CPU: 1 PID: 5835 Comm: syz-executor145 Not tainted 5.10.234-syzkaller #0 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.12.0-1 04/01/2014 RIP: 0010:jffs2_link_node_ref+0xac/0x690 fs/jffs2/nodelist.c:600 Call Trace: jffs2_mark_erased_block fs/jffs2/erase.c:460 [inline] jffs2_erase_pending_blocks+0x688/0x1860 fs/jffs2/erase.c:118 jffs2_garbage_collect_pass+0x638/0x1a00 fs/jffs2/gc.c:253 jffs2_reserve_space+0x3f4/0xad0 fs/jffs2/nodemgmt.c:167 jffs2_write_inode_range+0x246/0xb50 fs/jffs2/write.c:362 jffs2_write_end+0x712/0x1110 fs/jffs2/file.c:302 generic_perform_write+0x2c2/0x500 mm/filemap.c:3347 __generic_file_write_iter+0x252/0x610 mm/filemap.c:3465 generic_file_write_iter+0xdb/0x230 mm/filemap.c:3497 call_write_iter include/linux/fs.h:2039 [inline] do_iter_readv_writev+0x46d/0x750 fs/read_write.c:740 do_iter_write+0x18c/0x710 fs/read_write.c:866 vfs_writev+0x1db/0x6a0 fs/read_write.c:939 do_pwritev fs/read_write.c:1036 [inline] __do_sys_pwritev fs/read_write.c:1083 [inline] __se_sys_pwritev fs/read_write.c:1078 [inline] __x64_sys_pwritev+0x235/0x310 fs/read_write.c:1078 do_syscall_64+0x30/0x40 arch/x86/entry/common.c:46 entry_SYSCALL_64_after_hwframe+0x67/0xd1 Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-38329	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix OOB memory read access in KUnit test (wmfw info) KASAN reported out of bounds access - cs_dsp_mock_wmfw_add_info(), because the source string length was rounded up to the allocation size.
CVE-2025-38330	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix OOB memory read access in KUnit test (ctl cache) KASAN reported out of bounds access - cs_dsp_ctl_cache_init_multiple_offsets(). The code uses mock_coeff_template.length_bytes (4 bytes) for register value allocations. But later, this length is set to 8 bytes which causes test code failures. As fix, just remove the length override, keeping the original value 4 for all operations.
CVE-2025-38331	In the Linux kernel, the following vulnerability has been resolved: net: ethernet: cortina: Use TOE/TSO on all TCP It is desirable to push the hardware accelerator to also process non-segmented TCP frames: we pass the skb->len to the "TOE/TSO" offloader and it will handle them. Without this quirk the driver becomes unstable and lock up and crash. I do not know exactly why, but it is probably due to the TOE (TCP offload engine) feature that is coupled with the segmentation feature - it is not possible to turn one part off and not the other, either both TOE and TSO are active, or neither of them. Not having the TOE part active seems detrimental, as if that hardware feature is not really supposed to be turned off. The datasheet says: "Based on packet parsing and TCP connection/NAT table lookup results, the NetEngine puts the packets belonging to the same TCP connection to the same queue for the software to process. The NetEngine puts incoming packets to the buffer or series of buffers for a jumbo packet. With this hardware acceleration, IP/TCP header parsing, checksum validation and connection lookup are offloaded from the software processing." After numerous tests with the hardware locking up after something between minutes and hours depending on load using iperf3 I have concluded this is necessary to stabilize the hardware.
CVE-2025-38332	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Use memcpy() for BIOS version The strlcat() with FORTIFY support is triggering a panic because it thinks the target buffer will overflow although the correct target buffer size is passed in. Anyway, instead of memset() with 0 followed by a strlcat(), just use memcpy() and ensure that the resulting buffer is NULL terminated. BIOSVersion is only used for the lpfc_printf_log() which expects a properly terminated string.
CVE-2025-38333	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to bail out in get_new_segment() -----[cut here]----- WARNING: CPU: 3 PID: 579 at fs/f2fs/segment.c:2832 new_curseg+0x5e8/0x6dc pc : new_curseg+0x5e8/0x6dc Call trace: new_curseg+0x5e8/0x6dc f2fs_allocate_data_block+0xa54/0xe28 do_write_page+0x6c/0x194 f2fs_do_write_node_page+0x38/0x78 __write_node_page+0x248/0x6d4 f2fs_sync_node_pages+0x524/0x72c f2fs_write_checkpoint+0x4bc/0x9b0 __checkpoint_and_complete_reqs+0x80/0x244 issue_checkpoint_thread+0x8c/0xec kthread+0x114/0x1bc ret_from_fork+0x10/0x20 get_new_segment() detects inconsistent status in between free_segmap and free_secmap, let's record such error into super block, and bail out get_new_segment() instead of continue using the segment.

CVE ID	Description
CVE-2025-38334	In the Linux kernel, the following vulnerability has been resolved: x86/sgx: Prevent attempts to reclaim poisoned pages TL;DR: SGX page reclaim touches the page to copy its contents to secondary storage. SGX instructions do not gracefully handle machine checks. Despite this, the existing SGX code will try to reclaim pages that it _knows_ are poisoned. Avoid even trying to reclaim poisoned pages. The longer story: Pages used by an enclave only get epc_page->poison set in arch_memory_failure() but they currently stay on sgx_active_page_list until sgx_encl_release(), with the SGX_EPC_PAGE_RECLAIMER_TRACKED flag untouched. epc_page->poison is not checked in the reclaimer logic meaning that, if other conditions are met, an attempt will be made to reclaim an EPC page that was poisoned. This is bad because 1. we don't want that page to end up added to another enclave and 2. it is likely to cause one core to shut down and the kernel to panic. Specifically, reclaiming uses microcode operations including "EWB" which accesses the EPC page contents to encrypt and write them out to non-SGX memory. Those operations cannot handle MCEs in their accesses other than by putting the executing core into a special shutdown state (affecting both threads with HT.) The kernel will subsequently panic on the remaining cores seeing the core didn't enter MCE handler(s) in time. Call sgx_unmark_page_reclaimable() to remove the affected EPC page from sgx_active_page_list on memory error to stop it being considered for reclaiming. Testing epc_page->poison in sgx_reclaim_pages() would also work but I assume it's better to add code in the less likely paths. The affected EPC page is not added to &node->sgx_poison_page_list until later in sgx_encl_release()->sgx_free_epc_page() when it is EREMOVED. Membership on other lists doesn't change to avoid changing any of the lists' semantics except for sgx_active_page_list. There's a "TBD" comment in arch_memory_failure() about pre-emptive actions, the goal here is not to address everything that it may imply. This also doesn't completely close the time window when a memory error notification will be fatal (for a not previously poisoned EPC page) -- the MCE can happen after sgx_reclaim_pages() has selected its candidates or even *inside* a microcode operation (actually easy to trigger due to the amount of time spent in them.) The spinlock in sgx_unmark_page_reclaimable() is safe because memory_failure() runs in process context and no spinlocks are held, explicitly noted in a mm/memory-failure.c comment.
CVE-2025-38335	In the Linux kernel, the following vulnerability has been resolved: Input: gpio-keys - fix a sleep while atomic with PREEMPT_RT When enabling PREEMPT_RT, the gpio_keys_irq_timer() callback runs in hard irq context, but the input_event() takes a spin_lock, which isn't allowed there as it is converted to a rt_spin_lock(). [4054.289999] BUG: sleeping function called from invalid context at kernel/locking/spinlock_rt.c:48 [4054.290028] in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 0, name: swapper/0 ... [4054.290195] __might_resched+0x13c/0x1f4 [4054.290209] rt_spin_lock+0x54/0x11c [4054.290219] input_event+0x48/0x80 [4054.290230] gpio_keys_irq_timer+0x4c/0x78 [4054.290243] __hrtimer_run_queues+0x1a4/0x438 [4054.290257] hrtimer_interrupt+0xe4/0x240 [4054.290269] arch_timer_handler_phys+0x2c/0x44 [4054.290283] handle_percpu_devid_irq+0x8c/0x14c [4054.290297] handle_irq_desc+0x40/0x58 [4054.290307] generic_handle_domain_irq+0x1c/0x28 [4054.290316] gic_handle_irq+0x44/0xcc Considering the gpio_keys_irq_isr() can run in any context, e.g. it can be threaded, it seems there's no point in requesting the timer isr to run in hard irq context. Relax the hrtimer not to use the hard context.
CVE-2025-38336	In the Linux kernel, the following vulnerability has been resolved: ata: pata_via: Force PIO for ATAPI devices on VT6415/VT6330 The controller has a hardware bug that can hard hang the system when doing ATAPI DMAs without any trace of what happened. Depending on the device attached, it can also prevent the system from booting. In this case, the system hangs when reading the ATIP from optical media with cdrecord -vvv -atip on an _NEC DVD_RW ND-4571A 1-01 and an Optiarc DVD RW AD-7200A 1.06 attached to an ASRock 990FX Extreme 4, running at UDMA/33. The issue can be reproduced by running the same command with a cygwin build of cdrecord on WinXP, although it requires more attempts to cause it. The hang in that case is also resolved by forcing PIO. It doesn't appear that VIA has produced any drivers for that OS, thus no known workaround exists. HDDs attached to the controller do not suffer from any DMA issues.

CVE ID	Description
CVE-2025-38337	In the Linux kernel, the following vulnerability has been resolved: jbd2: fix data-race and null-ptr-deref in jbd2_journal_dirty_metadata() Since handle->h_transaction may be a NULL pointer, so we should change it to call is_handle_aborted(handle) first before dereferencing it. And the following data-race was reported in my fuzzer: <pre>===== BUG: KCSAN: data-race in jbd2_journal_dirty_metadata / jbd2_journal_dirty_metadata write to 0xffff888011024104 of 4 bytes by task 10881 on cpu 1: jbd2_journal_dirty_metadata +0x2a5/0x770 fs/jbd2/transaction.c:1556 __ext4_handle_dirty_metadata+0xe7/0x4b0 fs/ext4/ ext4_jbd2.c:358 ext4_do_update_inode fs/ext4/inode.c:5220 [inline] ext4_mark_iloc_dirty +0x32c/0xd50 fs/ext4/inode.c:5869 __ext4_mark_inode_dirty+0xe1/0x450 fs/ext4/ inode.c:6074 ext4_dirty_inode+0x98/0xc0 fs/ext4/inode.c:6103 read to 0xffff888011024104 of 4 bytes by task 10880 on cpu 0: jbd2_journal_dirty_metadata+0xf2/0x770 fs/jbd2/ transaction.c:1512 __ext4_handle_dirty_metadata+0xe7/0x4b0 fs/ext4/ext4_jbd2.c:358 ext4_do_update_inode fs/ext4/inode.c:5220 [inline] ext4_mark_iloc_dirty+0x32c/0xd50 fs/ext4/inode.c:5869 __ext4_mark_inode_dirty+0xe1/0x450 fs/ext4/inode.c:6074 ext4_dirty_inode+0x98/0xc0 fs/ext4/inode.c:6103 value changed: 0x00000000 -> 0x00000001 ===== This issue is caused by missing data-race annotation for jh->b_modified. Therefore, the missing annotation needs to be added.</pre>
CVE-2025-38338	In the Linux kernel, the following vulnerability has been resolved: fs/nfs/read: fix double-unlock bug in nfs_return_empty_folio() Sometimes, when a file was read while it was being truncated by another NFS client, the kernel could deadlock because folio_unlock() was called twice, and the second call would XOR back the 'PG_locked' flag. Most of the time (depending on the timing of the truncation), nobody notices the problem because folio_unlock() gets called three times, which flips 'PG_locked' back off: 1. vfs_read, nfs_read_folio, ... nfs_read_add_folio, nfs_return_empty_folio 2. vfs_read, nfs_read_folio, ... netfs_read_collection, netfs_unlock_abandoned_read_pages 3. vfs_read, ... nfs_do_read_folio, nfs_read_add_folio, nfs_return_empty_folio The problem is that nfs_read_add_folio() is not supposed to unlock the folio if fscache is enabled, and a nfs_netfs_folio_unlock() check is missing in nfs_return_empty_folio(). Rarely this leads to a warning in netfs_read_collection(): ----- [cut here]----- R=0000031c: folio 10 is not locked WARNING: CPU: 0 PID: 29 at fs/netfs/read_collect.c:133 netfs_read_collection+0x7c0/0xf00 [...] Workqueue: events_unbound netfs_read_collection_worker RIP: 0010:netfs_read_collection+0x7c0/0xf00 [...] Call Trace: <TASK> netfs_read_collection_worker+0x67/0x80 process_one_work+0x12e/0x2c0 worker_thread+0x295/0x3a0 Most of the time, however, processes just get stuck forever in folio_wait_bit_common(), waiting for 'PG_locked' to disappear, which never happens because nobody is really holding the folio lock.
CVE-2025-38340	In the Linux kernel, the following vulnerability has been resolved: firmware: cs_dsp: Fix OOB memory read access in KUnit test KASAN reported out of bounds access - cs_dsp_mock_bin_add_name_or_info(), because the source string length was rounded up to the allocation size.
CVE-2025-38342	In the Linux kernel, the following vulnerability has been resolved: software node: Correct a OOB check in software_node_get_reference_args() software_node_get_reference_args() wants to get @index-th element, so the property value requires at least '(index + 1) * sizeof(*ref)' bytes but that can not be guaranteed by current OOB check, and may cause OOB for malformed property. Fix by using as OOB check '((index + 1) * sizeof(*ref) > prop->length)'.
CVE-2025-38343	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7996: drop fragments with multicast or broadcast RA IEEE 802.11 fragmentation can only be applied to unicast frames. Therefore, drop fragments with multicast or broadcast RA. This patch addresses vulnerabilities such as CVE-2020-26145.

CVE ID	Description
CVE-2025-38344	In the Linux kernel, the following vulnerability has been resolved: ACPICA: fix acpi parse and parseext cache leaks ACPICA commit 8829e70e1360c81e7a5a901b5d4f48330e021ea5 I'm Seunghun Han, and I work for National Security Research Institute of South Korea. I have been doing a research on ACPI and found an ACPI cache leak in ACPI early abort cases. Boot log of ACPI cache leak is as follows: [0.352414] ACPI: Added _OSI(Module Device) [0.353182] ACPI: Added _OSI(Processor Device) [0.353182] ACPI: Added _OSI(3.0 _SCP Extensions) [0.353182] ACPI: Added _OSI(Processor Aggregator Device) [0.356028] ACPI: Unable to start the ACPI Interpreter [0.356799] ACPI Error: Could not remove SCI handler (20170303/evmisc-281) [0.360215] kmem_cache_destroy Acpi-State: Slab cache still has objects [0.360648] CPU: 0 PID: 1 Comm: swapper/0 Tainted: G W 4.12.0-rc4-next-20170608+ #10 [0.361273] Hardware name: innotek gmb_h virtual_box/virtual_box, BIOS virtual_box 12/01/2006 [0.361873] Call Trace: [0.362243] ? dump_stack+0x5c/0x81 [0.362591] ? kmem_cache_destroy+0x1aa/0x1c0 [0.362944] ? acpi_sleep_proc_init+0x27/0x27 [0.363296] ? acpi_os_delete_cache+0xa/0x10 [0.363646] ? acpi_ut_delete_caches+0x6d/0x7b [0.364000] ? acpi_terminate+0xa/0x14 [0.364000] ? acpi_init+0x2af/0x34f [0.364000] ? __class_create+0x4c/0x80 [0.364000] ? video_setup+0x7f/0x7f [0.364000] ? acpi_sleep_proc_init+0x27/0x27 [0.364000] ? do_one_initcall+0x4e/0x1a0 [0.364000] ? kernel_init_freeable+0x189/0x20a [0.364000] ? rest_init+0xc0/0xc0 [0.364000] ? kernel_init+0xa/0x100 [0.364000] ? ret_from_fork+0x25/0x30 I analyzed this memory leak in detail. I found that "Acpi-State" cache and "Acpi-Parser" cache were merged because the size of cache objects was same slab cache size. I finally found "Acpi-Parser" cache and "Acpi-parse_ext" cache were leaked using SLAB_NEVER_MERGE flag in kmem_cache_create() function. Real ACPI cache leak point is as follows: [0.360101] ACPI: Added _OSI(Module Device) [0.360101] ACPI: Added _OSI(Processor Device) [0.360101] ACPI: Added _OSI(3.0 _SCP Extensions) [0.361043] ACPI: Added _OSI(Processor Aggregator Device) [0.364016] ACPI: Unable to start the ACPI Interpreter [0.365061] ACPI Error: Could not remove SCI handler (20170303/evmisc-281) [0.368174] kmem_cache_destroy Acpi-Parser: Slab cache still has objects [0.369332] CPU: 1 PID: 1 Comm: swapper/0 Tainted: G W 4.12.0-rc4-next-20170608+ #8 [0.371256] Hardware name: innotek gmb_h virtual_box/virtual_box, BIOS virtual_box 12/01/2006 [0.372000] Call Trace: [0.372000] ? dump_stack+0x5c/0x81 [0.372000] ? kmem_cache_destroy+0x1aa/0x1c0 [0.372000] ? acpi_sleep_proc_init+0x27/0x27 [0.372000] ? acpi_os_delete_cache+0xa/0x10 [0.372000] ? acpi_ut_delete_caches+0x56/0x7b [0.372000] ? acpi_terminate+0xa/0x14 [0.372000] ? acpi_init+0x2af/0x34f [0.372000] ? __class_create+0x4c/0x80 [0.372000] ? video_setup+0x7f/0x7f [0.372000] ? acpi_sleep_proc_init+0x27/0x27 [0.372000] ? do_one_initcall+0x4e/0x1a0 [0.372000] ? kernel_init_freeable+0x189/0x20a [0.372000] ? rest_init+0xc0/0xc0 [0.372000] ? kernel_init+0xa/0x100 [0.372000] ? ret_from_fork+0x25/0x30 [0.388039] kmem_cache_destroy Acpi-parse_ext: Slab cache still has objects [0.389063] CPU: 1 PID: 1 Comm: swapper/0 Tainted: G W 4.12.0-rc4-next-20170608+ #8 [0.390557] Hardware name: innotek gmb_h virtual_box/virtual_box, BIOS virtual_box 12/01/2006 [0.392000] Call Trace: [0.392000] ? dump_stack+0x5c/0x81 [0.392000] ? kmem_cache_destroy+0x1aa/0x1c0 [0.392000] ? acpi_sleep_proc_init+0x27/0x27 [0.392000] ? acpi_os_delete_cache+0xa/0x10 [0.392000] ? acpi_ut_delete_caches+0x6d/0x7b [0.392000] ? acpi_terminate+0xa/0x14 [0.392000] ? acpi_init+0x2af/0x34f ---truncated---

CVE ID	Description
CVE-2025-38345	In the Linux kernel, the following vulnerability has been resolved: ACPICA: fix acpi operand cache leak in dswstate.c ACPICA commit 987a3b5cf7175916e2a4b6ea5b8e70f830dfe732 I found an ACPI cache leak in ACPI early termination and boot continuing case. When early termination occurs due to malicious ACPI table, Linux kernel terminates ACPI function and continues to boot process. While kernel terminates ACPI function, kmem_cache_destroy() reports Acpi-Operand cache leak. Boot log of ACPI operand cache leak is as follows: >[0.585957] ACPI: Added _OSI(Module Device) >[0.587218] ACPI: Added _OSI(Processor Device) >[0.588530] ACPI: Added _OSI(3.0 _SCP Extensions) >[0.589790] ACPI: Added _OSI(Processor Aggregator Device) >[0.591534] ACPI Error: Illegal I/O port address/length above 64K: C806E00000004002/0x2 (20170303/hwvalid-155) >[0.594351] ACPI Exception: AE_LIMIT, Unable to initialize fixed events (20170303/evevent-88) >[0.597858] ACPI: Unable to start the ACPI Interpreter >[0.599162] ACPI Error: Could not remove SCI handler (20170303/evmisc-281) >[0.601836] kmem_cache_destroy Acpi-Operand: Slab cache still has objects >[0.603556] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 4.12.0-rc5 #26 >[0.605159] Hardware name: innotek gmb_h virtual_box/virtual_box, BIOS virtual_box 12/01/2006 >[0.609177] Call Trace: >[0.610063] ? dump_stack+0x5c/0x81 >[0.611118] ? kmem_cache_destroy+0x1aa/0x1c0 >[0.612632] ? acpi_sleep_proc_init+0x27/0x27 >[0.613906] ? acpi_os_delete_cache+0xa/0x10 >[0.617986] ? acpi_ut_delete_caches+0x3f/0x7b >[0.619293] ? acpi_terminate+0xa/0x14 >[0.620394] ? acpi_init+0x2af/0x34f >[0.621616] ? __class_create+0x4c/0x80 >[0.623412] ? video_setup+0x7f/0x7f >[0.624585] ? acpi_sleep_proc_init+0x27/0x27 >[0.625861] ? do_one_initcall+0x4e/0x1a0 >[0.627513] ? kernel_init_freeable+0x19e/0x21f >[0.628972] ? rest_init+0x80/0x80 >[0.630043] ? kernel_init+0xa/0x100 >[0.631084] ? ret_from_fork+0x25/0x30 >[0.633343] vgaarb: loaded >[0.635036] EDAC MC: Ver: 3.0.0 >[0.638601] PCI: Probing PCI hardware >[0.639833] PCI host bridge to bus 0000:00 >[0.641031] pci_bus 0000:00: root bus resource [io 0x0000-0xffff] > ... Continue to boot and log is omitted ... I analyzed this memory leak in detail and found acpi_ds_obj_stack_pop_and_delete() function miscalculated the top of the stack. acpi_ds_obj_stack_push() function uses walk_state->operand_index for start position of the top, but acpi_ds_obj_stack_pop_and_delete() function considers index 0 for it. Therefore, this causes acpi operand memory leak. This cache leak causes a security threat because an old kernel (<= 4.9) shows memory locations of kernel functions in stack dump. Some malicious users could use this information to neutralize kernel ASLR. I made a patch to fix ACPI operand cache leak.
CVE-2025-38346	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix UAF when lookup kallsym after ftrace disabled The following issue happens with a buggy module: BUG: unable to handle page fault for address: ffffffff05d0218 PGD 1bd66f067 P4D 1bd66f067 PUD 1bd671067 PMD 101808067 PTE 0 Oops: Oops: 0000 [#1] SMP KASAN PTI Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS RIP: 0010:sized_strncpy+0x81/0x2f0 RSP: 0018:ffff88812d76fa08 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffffffff0601010 RCX: dffffc0000000000 RDX: 0000000000000038 RSI: dffffc0000000000 RDI: ffff88812608da2d RBP: 8080808080808080 R08: ffff88812608da2d R09: ffff88812608da68 R10: ffff88812608d82d R11: ffff88812608d810 R12: 0000000000000038 R13: ffff88812608da2d R14: ffffffff05d0218 R15: fefefefefefeff FS: 00007fef552de740(0000) GS: ffff8884251c7000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CRO: 0000000080050033 CR2: ffffffff05d0218 CR3: 00000001146f0000 CR4: 000000000000006f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> ftrace_mod_get_kallsym+0x1ac/0x590 update_iter_mod+0x239/0x5b0 s_next+0x5b/0xa0 seq_read_iter+0x8c9/0x1070 seq_read+0x249/0x3b0 proc_reg_read+0x1b0/0x280 vfs_read+0x17f/0x920 ksys_read+0xf3/0x1c0 do_syscall_64+0x5f/0x2e0 entry_SYSCALL_64_after_hwframe+0x76/0x7e The above issue may happen as follows: (1) Add kprobe tracepoint; (2) insmod test.ko; (3) Module triggers ftrace disabled; (4) rmmmod test.ko; (5) cat /proc/kallsyms; --> Will trigger UAF as test.ko already removed; ftrace_mod_get_kallsym() ... strncpy(module_name, mod_map->mod->name, MODULE_NAME_LEN); ... The problem is when a module triggers an issue with ftrace and sets ftrace_disable. The ftrace_disable is set when an anomaly is discovered and to prevent any more damage, ftrace stops all text modification. The issue that happened was that the ftrace_disable stops more than just the text modification. When a module is loaded, its init functions can also be traced. Because kallsyms deletes the init functions after a module has loaded, ftrace saves them when the module is loaded and function tracing is enabled. This allows the output of the function trace to show the init function names instead of just their raw memory addresses. When a module is removed, ftrace_release_mod() is called, and if ftrace_disable is set, it just returns without doing anything more. The problem here is that it leaves the mod_list still around and if kallsyms is called, it will call into this code and access the module memory that has already been freed as it will return: strncpy(module_name, mod_map->mod->name, MODULE_NAME_LEN); Where the "mod" no longer exists and triggers a UAF bug.

CVE ID	Description
CVE-2025-38347	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to do sanity check on ino and xnid syzbot reported a f2fs bug as below: INFO: task syz-executor140:5308 blocked for more than 143 seconds. Not tainted 6.14.0-rc7-syzkaller-00069-g81e4f8d68c66 #0 "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message. task:syz-executor140 state:D stack:24016 pid:5308 tgid:5308 ppid:5306 task_flags:0x400140 flags:0x00000006 Call Trace: <TASK> context_switch kernel/sched/core.c:5378 [inline] __schedule+0x190e/0x4c90 kernel/sched/core.c:6765 __schedule_loop kernel/sched/core.c:6842 [inline] schedule+0x14b/0x320 kernel/sched/core.c:6857 io_schedule+0x8d/0x110 kernel/sched/core.c:7690 folio_wait_bit_common+0x839/0xee0 mm/filemap.c:1317 __folio_lock mm/filemap.c:1664 [inline] folio_lock include/linux/pagemap.h:1163 [inline] __filemap_get_folio+0x147/0xb40 mm/filemap.c:1917 pagecache_get_page+0x2c/0x130 mm/folio-compat.c:87 find_get_page_flags include/linux/pagemap.h:842 [inline] f2fs_grab_cache_page+0x2b/0x320 fs/f2fs/f2fs.h:2776 __get_node_page+0x131/0x11b0 fs/f2fs/node.c:1463 read_xattr_block+0xfb/0x190 fs/f2fs/xattr.c:306 lookup_all_xattrs fs/f2fs/xattr.c:355 [inline] f2fs_getxattr+0x676/0xf70 fs/f2fs/xattr.c:533 __f2fs_get_acl+0x52/0x870 fs/f2fs/acl.c:179 f2fs_acl_create fs/f2fs/acl.c:375 [inline] f2fs_init_acl+0xd7/0x9b0 fs/f2fs/acl.c:418 f2fs_init_inode_metadata+0xa0f/0x1050 fs/f2fs/dir.c:539 f2fs_add_inline_entry+0x448/0x860 fs/f2fs/inline.c:666 f2fs_add_dentry+0xba/0x1e0 fs/f2fs/dir.c:765 f2fs_do_add_link+0x28c/0x3a0 fs/f2fs/dir.c:808 f2fs_add_link fs/f2fs/f2fs.h:3616 [inline] f2fs_mknod+0x2e8/0x5b0 fs/f2fs/namei.c:766 vfs_mknod+0x36d/0x3b0 fs/namei.c:4191 unix_bind_bsd net/unix/af_unix.c:1286 [inline] unix_bind+0x563/0xe30 net/unix/af_unix.c:1379 __sys_bind_socket net/socket.c:1817 [inline] __sys_bind+0x1e4/0x290 net/socket.c:1848 __do_sys_bind net/socket.c:1853 [inline] __se_sys_bind net/socket.c:1851 [inline] __x64_sys_bind+0x7a/0x90 net/socket.c:1851 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f Let's dump and check metadata of corrupted inode, it shows its xattr_nid is the same to its i_ino. dump.f2fs -i 3 chaseyu.img.raw i_xattr_nid [0x 3 : 3] So that, during mknod in the corrupted directory, it tries to get and lock inode page twice, result in deadlock. - f2fs_mknod - f2fs_add_inline_entry - f2fs_get_inode_page --- lock dir's inode page - f2fs_init_acl - f2fs_acl_create(dir,..) - __f2fs_get_acl - f2fs_getxattr - lookup_all_xattrs - __get_node_page --- try to lock dir's inode page In order to fix this, let's add sanity check on ino and xnid.
CVE-2025-38348	In the Linux kernel, the following vulnerability has been resolved: wifi: p54: prevent buffer-overflow in p54_rx_eeprom_readback() Robert Morris reported: If a malicious USB device pretends to be an Intersil p54 wifi interface and generates an eeprom_readback message with a large eeprom->v1.len, p54_rx_eeprom_readback() will copy data from the message beyond the end of priv->eeprom. static void p54_rx_eeprom_readback(struct p54_common *priv, struct sk_buff *skb) { struct p54_hdr *hdr = (struct p54_hdr *) skb->data; struct p54_eeprom_lm86 *eeprom = (struct p54_eeprom_lm86 *) hdr->data; if (priv->fw_var >= 0x509) { memcpy(priv->eeprom, eeprom->v2.data, le16_to_cpu(eeprom->v2.len)); } else { memcpy(priv->eeprom, eeprom->v1.data, le16_to_cpu(eeprom->v1.len)); } [...] The eeprom->v[1,2].len is set by the driver in p54_download_eeprom(). The device is supposed to provide the same length back to the driver. But yes, it's possible (like shown in the report) to alter the value to something that causes a crash/panic due to overrun. This patch addresses the issue by adding the size to the common device context, so p54_rx_eeprom_readback no longer relies on possibly tampered values... That said, it also checks if the "firmware" altered the value and no longer copies them. The one, small saving grace is: Before the driver tries to read the eeprom, it needs to upload >a< firmware. the vendor firmware has a proprietary license and as a reason, it is not present on most distributions by default.
CVE-2025-38349	In the Linux kernel, the following vulnerability has been resolved: eventpoll: don't decrement ep refcount while still holding the ep mutex Jann Horn points out that epoll is decrementing the ep refcount and then doing a mutex_unlock(&ep->mtx); afterwards. That's very wrong, because it can lead to a use-after-free. That pattern is actually fine for the very last reference, because the code in question will delay the actual call to "ep_free(ep)" until after it has unlocked the mutex. But it's wrong for the much subtler "next to last" case when somebody *else* may also be dropping their reference and free the ep while we're still using the mutex. Note that this is true even if that other user is also using the same ep mutex: mutexes, unlike spinlocks, can not be used for object ownership, even if they guarantee mutual exclusion. A mutex "unlock" operation is not atomic, and as one user is still accessing the mutex as part of unlocking it, another user can come in and get the now released mutex and free the data structure while the first user is still cleaning up. See our mutex documentation in Documentation/locking/mutex-design.rst, in particular the section [1] about semantics: "mutex_unlock() may access the mutex structure even after it has internally released the lock already - so it's not safe for another context to acquire the mutex and assume that the mutex_unlock() context is not using the structure anymore" So if we drop our ep ref before the mutex unlock, but we weren't the last one, we may then unlock the mutex, another user comes in, drops _their_ reference and releases the 'ep' as it now has no users - all while the mutex_unlock() is still accessing it. Fix this by simply moving the ep refcount dropping to outside the mutex: the refcount itself is atomic, and doesn't need mutex protection (that's the whole _point_ of refcounts: unlike mutexes, they are inherently about object lifetimes).

CVE ID	Description
CVE-2025-38350	In the Linux kernel, the following vulnerability has been resolved: net/sched: Always pass notifications when child class becomes empty Certain classful qdiscs may invoke their classes' dequeue handler on an enqueue operation. This may unexpectedly empty the child qdisc and thus make an in-flight class passive via qlen_notify(). Most qdiscs do not expect such behaviour at this point in time and may re-activate the class eventually anyways which will lead to a use-after-free. The referenced fix commit attempted to fix this behavior for the HFSC case by moving the backlog accounting around, though this turned out to be incomplete since the parent's parent may run into the issue too. The following reproducer demonstrates this use-after-free: tc qdisc add dev lo root handle 1: drr tc filter add dev lo parent 1: basic classid 1:1 tc class add dev lo parent 1: classid 1:1 drr tc qdisc add dev lo parent 1:1 handle 2: hfsc def 1 tc class add dev lo parent 2: classid 2:1 hfsc rt m1 8 d 1 m2 0 tc qdisc add dev lo parent 2:1 handle 3: netem tc qdisc add dev lo parent 3:1 handle 4: blackhole echo 1 socat -u STDIN UDP4-DATAGRAM:127.0.0.1:8888 tc class delete dev lo classid 1:1 echo 1 socat -u STDIN UDP4-DATAGRAM:127.0.0.1:8888 Since backlog accounting issues leading to a use-after-frees on stale class pointers is a recurring pattern at this point, this patch takes a different approach. Instead of trying to fix the accounting, the patch ensures that qdisc_tree_reduce_backlog always calls qlen_notify when the child qdisc is empty. This solves the problem because deletion of qdiscs always involves a call to qdisc_reset() and / or qdisc_purge_queue() which ultimately resets its qlen to 0 thus causing the following qdisc_tree_reduce_backlog() to report to the parent. Note that this may call qlen_notify on passive classes multiple times. This is not a problem after the recent patch series that made all the classful qdiscs qlen_notify() handlers idempotent.
CVE-2025-38351	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/hyper-v: Skip non-canonical addresses during PV TLB flush In KVM guests with Hyper-V hypercalls enabled, the hypercalls HVCALL_FLUSH_VIRTUAL_ADDRESS_LIST and HVCALL_FLUSH_VIRTUAL_ADDRESS_LIST_EX allow a guest to request invalidation of portions of a virtual TLB. For this, the hypercall parameter includes a list of GVAs that are supposed to be invalidated. However, when non-canonical GVAs are passed, there is currently no filtering in place and they are eventually passed to checked invocations of INVVPID on Intel / INVLPGA on AMD. While AMD's INVLPGA silently ignores non-canonical addresses (effectively a no-op), Intel's INVVPID explicitly signals VM-Fail and ultimately triggers the WARN_ONCE in invvpid_error(): invvpid failed: ext=0x0 vpid=1 gva=0xaaaaaaaaaaaa000 WARNING: CPU: 6 PID: 326 at arch/x86/kvm/vmx/vmx.c:482 invvpid_error+0x91/0xa0 [kvm_intel] Modules linked in: kvm_intel kvm 9pnet_virtio irqbypass fuse CPU: 6 UID: 0 PID: 326 Comm: kvm-vm Not tainted 6.15.0 #14 PREEMPT(voluntary) RIP: 0010:invvpid_error+0x91/0xa0 [kvm_intel] Call Trace: vmx_flush_tlb_gva+0x320/0x490 [kvm_intel] kvm_hv_vcpu_flush_tlb+0x24f/0x4f0 [kvm] kvm_arch_vcpu_ioctl_run+0x3013/0x5810 [kvm] Hyper-V documents that invalid GVAs (those that are beyond a partition's GVA space) are to be ignored. While not completely clear whether this ruling also applies to non-canonical GVAs, it is likely fine to make that assumption, and manual testing on Azure confirms "real" Hyper-V interprets the specification in the same way. Skip non-canonical GVAs when processing the list of address to avoid tripping the INVVPID failure. Alternatively, KVM could filter out "bad" GVAs before inserting into the FIFO, but practically speaking the only downside of pushing validation to the final processing is that doing so is suboptimal for the guest, and no well-behaved guest will request TLB flushes for non-canonical addresses.
CVE-2025-38353	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Fix taking invalid lock on wedge If device wedges on e.g. GuC upload, the submission is not yet enabled and the state is not even initialized. Protect the wedge call so it does nothing in this case. It fixes the following splat: [] xe 0000:bf:00.0: [drm] device wedged, needs recovery [] -----[cut here]----- [] DEBUG_LOCKS_WARN_ON(lock->magic != lock) [] WARNING: CPU: 48 PID: 312 at kernel/locking/mutex.c:564 __mutex_lock+0x8a1/0xe60 ... [] RIP: 0010:__mutex_lock+0x8a1/0xe60 [] mutex_lock_nested+0x1b/0x30 [] xe_guc_submit_wedge+0x80/0x2b0 [xe]

CVE ID	Description
CVE-2025-38354	In the Linux kernel, the following vulnerability has been resolved: drm/msm/gpu: Fix crash when throttling GPU immediately during boot There is a small chance that the GPU is already hot during boot. In that case, the call to of_devfreq_cooling_register() will immediately try to apply devfreq cooling, as seen in the following crash: Unable to handle kernel paging request at virtual address 0000000000014110 pc : a6xx_gpu_busy+0x1c/0x58 [msm] lr : msm_devfreq_get_dev_status+0x9c/0x140 [msm] Call trace: a6xx_gpu_busy+0x1c/0x58 [msm] (P) devfreq_simple_ondemand_func+0x3c/0x150 devfreq_update_target+0x44/0xd8 qos_max_notifier_call+0x30/0x84 blocking_notifier_call_chain+0x6c/0xa0 pm_qos_update_target+0xd0/0x110 freq_qos_apply+0x3c/0x74 apply_constraint+0x88/0x148 __dev_pm_qos_update_request+0x7c/0xcc dev_pm_qos_update_request+0x38/0x5c devfreq_cooling_set_cur_state+0x98/0xf0 __thermal_cdev_update+0x64/0xb4 thermal_cdev_update+0x4c/0x58 step_wise_manage+0x1f0/0x318 __thermal_zone_device_update+0x278/0x424 __thermal_cooling_device_register+0x2bc/0x308 thermal_of_cooling_device_register+0x10/0x1c of_devfreq_cooling_register_power+0x240/0x2bc of_devfreq_cooling_register+0x14/0x20 msm_devfreq_init+0xc4/0x1a0 [msm] msm_gpu_init+0x304/0x574 [msm] adreno_gpu_init+0x1c4/0x2e0 [msm] a6xx_gpu_init+0x5c8/0x9c8 [msm] adreno_bind+0x2a8/0x33c [msm] ... At this point we haven't initialized the GMU at all yet, so we cannot read the GMU registers inside a6xx_gpu_busy(). A similar issue was fixed before in commit 6694482a70e9 ("drm/msm: Avoid unlocked GMU register access in 6xx gpu_busy"): msm_devfreq_init() does call devfreq_suspend_device(), but unlike msm_devfreq_suspend(), it doesn't set the df->suspended flag accordingly. This means the df->suspended flag does not match the actual devfreq state after initialization and msm_devfreq_get_dev_status() will end up accessing GMU registers, causing the crash. Fix this by setting df->suspended correctly during initialization. Patchwork: https://patchwork.freedesktop.org/patch/650772/
CVE-2025-38359	In the Linux kernel, the following vulnerability has been resolved: s390/mm: Fix in_atomic() handling in do_secure_storage_access() Kernel user spaces accesses to not exported pages in atomic context incorrectly try to resolve the page fault. With debug options enabled call traces like this can be seen: BUG: sleeping function called from invalid context at kernel/locking/rwsem.c:1523 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 419074, name: qemu-system-s39 preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 INFO: lockdep is turned off. Preemption disabled at: [<00000383ea47cfa2>] copy_page_from_iter_atomic+0xa2/0x8a0 CPU: 12 UID: 0 PID: 419074 Comm: qemu-system-s39 Tainted: G W 6.16.0-20250531.rc0.git0.69b3a602feac.63.fc42.s390x +debug #1 PREEMPT Tainted: [W]=WARN Hardware name: IBM 3931 A01 703 (LPAR) Call Trace: [<00000383e990d282>] dump_stack_lvl+0xa2/0xe8 [<00000383e99bf152>] __might_resched+0x292/0x2d0 [<00000383eaa7c374>] down_read+0x34/0x2d0 [<00000383e99432f8>] do_secure_storage_access+0x108/0x360 [<00000383eaa724b0>] __do_pgm_check+0x130/0x220 [<00000383eaa842e4>] pgm_check_handler+0x114/0x160 [<00000383ea47d028>] copy_page_from_iter_atomic+0x128/0x8a0 [<00000383ea47d016>] copy_page_from_iter_atomic+0x116/0x8a0 [<00000383e9c45eae>] generic_perform_write+0x16e/0x310 [<00000383e9eb87f4>] ext4_buffered_write_iter+0x84/0x160 [<00000383e9da0de4>] vfs_write+0x1c4/0x460 [<00000383e9da123c>] ksys_write+0x7c/0x100 [<00000383eaa7284e>] __do_syscall+0x15e/0x280 [<00000383eaa8417e>] system_call+0x6e/0x90 INFO: lockdep is turned off. It is not allowed to take the mmap_lock while in atomic context. Therefore handle such a secure storage access fault as if the accessed page is not mapped: the uaccess function will return -EFAULT, and the caller has to deal with this. Usually this means that the access is retried in process context, which allows to resolve the page fault (or in this case export the page).
CVE-2025-38360	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add more checks for DSC / HUBP ONO guarantees [WHY] For non-zero DSC instances it's possible that the HUBP domain required to drive it for sequential ONO ASICs isn't met, potentially causing the logic to the tile to enter an undefined state leading to a system hang. [HOW] Add more checks to ensure that the HUBP domain matching the DSC instance is appropriately powered. (cherry picked from commit da63df07112e5a9857a8d2aaa04255c4206754ec)
CVE-2025-38361	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Check dce_hwseq before dereferencing it [WHAT] hws was checked for null earlier in dce110_blank_stream, indicating hws can be null, and should be checked whenever it is used. (cherry picked from commit 79db43611ff61280b6de58ce1305e0b2ecf675ad)
CVE-2025-38362	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null pointer check for get_first_active_display() The function mod_hdcp_hdcp1_enable_encryption() calls the function get_first_active_display(), but does not check its return value. The return value is a null pointer if the display list is empty. This will lead to a null pointer dereference in mod_hdcp_hdcp2_enable_encryption(). Add a null pointer check for get_first_active_display() and return MOD_HDCP_STATUS_DISPLAY_NOT_FOUND if the function return null.

CVE ID	Description
CVE-2025-38363	In the Linux kernel, the following vulnerability has been resolved: drm/tegra: Fix a possible null pointer dereference In <code>tegra_crtc_reset()</code> , new memory is allocated with <code>kzalloc()</code> , but no check is performed. Before calling <code>__drm_atomic_helper_crtc_reset</code> , state should be checked to prevent possible null pointer dereference.
CVE-2025-38364	In the Linux kernel, the following vulnerability has been resolved: maple_tree: fix MA_STATE_PREALLOC flag in <code>mas_preallocate()</code> Temporarily clear the preallocation flag when explicitly requesting allocations. Pre-existing allocations are already counted against the request through <code>mas_node_count_gfp()</code> , but the allocations will not happen if the MA_STATE_PREALLOC flag is set. This flag is meant to avoid re-allocating in bulk allocation mode, and to detect issues with preallocation calculations. The MA_STATE_PREALLOC flag should also always be set on zero allocations so that detection of underflow allocations will print a <code>WARN_ON()</code> during consumption. User visible effect of this flaw is a <code>WARN_ON()</code> followed by a null pointer dereference when subsequent requests for larger number of nodes is ignored, such as the vma merge retry in <code>mmap_region()</code> caused by drivers altering the vma flags (which happens in v6.6, at least)
CVE-2025-38365	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix a race between renames and directory logging We have a race between a rename and directory inode logging that if it happens and we crash/power fail before the rename completes, the next time the filesystem is mounted, the log replay code will end up deleting the file that was being renamed. This is best explained following a step by step analysis of an interleaving of steps that lead into this situation. Consider the initial conditions: 1) We are at transaction N; 2) We have directories A and B created in a past transaction (< N); 3) We have inode X corresponding to a file that has 2 hardlinks, one in directory A and the other in directory B, so we'll name them as "A/foo_link1" and "B/foo_link2". Both hard links were persisted in a past transaction (< N); 4) We have inode Y corresponding to a file that as a single hard link and is located in directory A, we'll name it as "A/bar". This file was also persisted in a past transaction (< N). The steps leading to a file loss are the following and for all of them we are under transaction N: 1) Link "A/foo_link1" is removed, so inode's X last_unlink_trans field is updated to N, through <code>btrfs_unlink()</code> -> <code>btrfs_record_unlink_dir()</code> ; 2) Task A starts a rename for inode Y, with the goal of renaming from "A/bar" to "A/baz", so we enter <code>btrfs_rename()</code> ; 3) Task A inserts the new BTRFS_INODE_REF_KEY for inode Y by calling <code>btrfs_insert_inode_ref()</code> ; 4) Because the rename happens in the same directory, we don't set the last_unlink_trans field of directory A's inode to the current transaction id, that is, we don't call <code>btrfs_record_unlink_dir()</code> ; 5) Task A then removes the entries from directory A (BTRFS_DIR_ITEM_KEY and BTRFS_DIR_INDEX_KEY items) when calling <code>__btrfs_unlink_inode()</code> (actually the dir index item is added as a delayed item, but the effect is the same); 6) Now before task A adds the new entry "A/baz" to directory A by calling <code>btrfs_add_link()</code> , another task, task B is logging inode X; 7) Task B starts a fsync of inode X and after logging inode X, at <code>btrfs_log_inode_parent()</code> it calls <code>btrfs_log_all_parents()</code> , since inode X has a last_unlink_trans value of N, set at in step 1; 8) At <code>btrfs_log_all_parents()</code> we search for all parent directories of inode X using the commit root, so we find directories A and B and log them. But when logging directory A, we don't have a dir index item for inode Y anymore, neither the old name "A/bar" nor for the new name "A/baz" since the rename has deleted the old name but has not yet inserted the new name - task A hasn't called yet <code>btrfs_add_link()</code> to do that. Note that logging directory A doesn't fallback to a transaction commit because its last_unlink_trans has a lower value than the current transaction's id (see step 4); 9) Task B finishes logging directories A and B and gets back to <code>btrfs_sync_file()</code> where it calls <code>btrfs_sync_log()</code> to persist the log tree; 10) Task B successfully persisted the log tree, <code>btrfs_sync_log()</code> completed with success, and a power failure happened. We have a log tree without any directory entry for inode Y, so the log replay code deletes the entry for inode Y, name "A/bar", from the subvolume tree since it doesn't exist in the log tree and the log tree is authoritative for its index (we logged a BTRFS_DIR_LOG_INDEX_KEY item that covers the index range for the entry that corresponds to "A/bar"). Since there's no other hard link for inode Y and the log replay code deletes the name "A/bar", the file is lost. The issue wouldn't happen if task B synced the log only after task A called <code>btrfs_log_new_name()</code> , which would update the log with the new name for inode Y ("A/bar"). Fix this by pinning the log root during renames before removing the old directory entry, and unpinning after ---truncated---
CVE-2025-38368	In the Linux kernel, the following vulnerability has been resolved: misc: tps6594-pfsm: Add NULL pointer check in <code>tps6594_pfsm_probe()</code> The returned value, <code>pfsm->miscdev.name</code> , from <code>devm_kasprintf()</code> could be NULL. A pointer check is added to prevent potential NULL pointer dereference. This is similar to the fix in commit 3027e7b15b02 ("ice: Fix some null pointer dereference issues in ice_ptp.c"). This issue is found by our static analysis tool.
CVE-2025-38369	In the Linux kernel, the following vulnerability has been resolved: dmaengine: idxd: Check availability of workqueue allocated by idxd wq driver before using Running IDXD workloads in a container with the /dev directory mounted can trigger a call trace or even a kernel panic when the parent process of the container is terminated. This issue occurs because, under certain configurations, Docker does not properly propagate the mount replica back to the original mount point. In this case, when the user driver detaches, the WQ is destroyed but it still calls <code>destroy_workqueue()</code> attempting to completes all pending work. It's necessary to check <code>wq->wq</code> and skip the drain if it no longer exists.

CVE ID	Description
CVE-2025-38371	In the Linux kernel, the following vulnerability has been resolved: drm/v3d: Disable interrupts before resetting the GPU. Currently, an interrupt can be triggered during a GPU reset, which can lead to GPU hangs and NULL pointer dereference in an interrupt context as shown in the following trace: <pre>[314.035040] Unable to handle kernel NULL pointer dereference at virtual address 00000000000000c0 [314.043822] Mem abort info: [314.046606] ESR = 0x0000000096000005 [314.050347] EC = 0x25: DABT (current EL), IL = 32 bits [314.055651] SET = 0, FnV = 0 [314.058695] EA = 0, S1PTW = 0 [314.061826] FSC = 0x05: level 1 translation fault [314.066694] Data abort info: [314.069564] ISV = 0, ISS = 0x00000005, ISS2 = 0x00000000 [314.075039] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [314.080080] GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 [314.085382] user pgtable: 4k pages, 39-bit VAs, pgdp=0000000102728000 [314.091814] [00000000000000c0] pgd=0000000000000000, p4d=0000000000000000, pud=0000000000000000 [314.100511] Internal error: Oops: 0000000096000005 [#1] PREEMPT SMP [314.106770] Modules linked in: v3d i2c_brcmstb vc4 snd_soc_hdmi_codec gpu_sched drm_shmem_helper drm_display_helper cec drm_dma_helper drm_kms_helper drm drm_panel_orientation_quirks snd_soc_core snd_compress snd_pcm_dmaengine snd_pcm snd_timer snd backlight [314.129654] CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.12.25+rpt-rpi-v8 #1 Debian 1:6.12.25-1+rpt1 [314.139388] Hardware name: Raspberry Pi 4 Model B Rev 1.4 (DT) [314.145211] pstate: 600000c5 (nZCv daIF -PAN -UAO -TCO -DIT -SSBS BTYPPE=) [314.152165] pc : v3d_irq+0xec/0x2e0 [v3d] [314.156187] lr : v3d_irq+0xe0/0x2e0 [v3d] [314.160198] sp : ffffffc080003ea0 [314.163502] x29: ffffffc080003ea0 x28: ffffffec1f184980 x27: 021202b000000000 [314.170633] x26: fffffec1f17f630 x25: ffffff8101372000 x24: ffffffec1f17d9f0 [314.177764] x23: 000000000000002a x22: 000000000000002a x21: ffffff8103252000 [314.184895] x20: 0000000000000001 x19: 00000000deadbeef x18: 0000000000000000 [314.192026] x17: ffffff94e51d2000 x16: ffffffec1dac3cb0 x15: c306000000000000 [314.199156] x14: 0000000000000000 x13: b2fc982e03cc5168 x12: 0000000000000001 [314.206286] x11: ffffff8103f8bcc0 x10: ffffffec1f196868 x9 : fffffec1dac3874 [314.213416] x8 : 0000000000000000 x7 : 00000000000042a3a x6 : ffffff810017a180 [314.220547] x5 : ffffffec1ebad400 x4 : ffffffec1ebad320 x3 : 000000000000bebe [314.227677] x2 : 0000000000000000 x1 : 0000000000000000 x0 : 0000000000000000 [314.234807] Call trace: [314.237243] v3d_irq+0xec/0x2e0 [v3d] [314.240906] __handle_irq_event_percpu +0x58/0x218 [314.245609] handle_irq_event+0x54/0xb8 [314.249439] handle_fastestoi_irq +0xac/0x240 [314.253527] handle_irq_desc+0x48/0x68 [314.257269] generic_handle_domain_irq +0x24/0x38 [314.261879] gic_handle_irq+0x48/0xd8 [314.265533] call_on_irq_stack+0x24/0x58 [314.269448] do_interrupt_handler+0x88/0x98 [314.273624] el1_interrupt+0x34/0x68 [314.277193] el1h_64_irq_handler+0x18/0x28 [314.281281] el1h_64_irq+0x64/0x68 [314.284673] default_idle_call +0x3c/0x168 [314.288675] do_idle+0x1fc/0x230 [314.291895] cpu_startup_entry+0x3c/0x50 [314.295810] rest_init+0xe4/0xf0 [314.299030] start_kernel+0x5e8/0x790 [314.302684] __primary_switched+0x80/0x90 [314.306691] Code: 940029eb 360ffc13 f9442ea0 52800001 (f9406017) [314.312775] ---[end trace 0000000000000000]--- [314.317384] Kernel panic - not syncing: Oops: Fatal exception in interrupt [314.324249] SMP: stopping secondary CPUs [314.328167] Kernel Offset: 0x2b9da00000 from 0xfffffc0800000000 [314.334076] PHYS_OFFSET: 0x0 [314.336946] CPU features: 0x08,00002013,c0200000,0200421b [314.342337] Memory Limit: none [314.345382] ---[end Kernel panic - not syncing: Oops: Fatal exception in interrupt]--- Before resetting the G ---truncated--- </pre>
CVE-2025-38372	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Fix unsafe xarray access in implicit ODP handling __xa_store() and __xa_erase() were used without holding the proper lock, which led to a lockdep warning due to unsafe RCU usage. This patch replaces them with xa_store() and xa_erase(), which perform the necessary locking internally. ===== <pre>WARNING: suspicious RCU usage 6.14.0-rc7_for_upstream_debug_2025_03_18_15_01 #1 Not tainted ----- ./include/linux/xarray.h:1211 suspicious rcu_dereference_protected() usage! other info that might help us debug this: rcu_scheduler_active = 2, debug_locks = 1 3 locks held by kworker/u136:0/219: at: process_one_work+0xbe4/0x15f0 process_one_work+0x75c/0x15f0 pagefault_mr+0x9a5/0x1390 [mlx5_ib] stack backtrace: CPU: 14 UID: 0 PID: 219 Comm: kworker/ u136:0 Not tainted 6.14.0-rc7_for_upstream_debug_2025_03_18_15_01 #1 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.16.0-0-gd239552ce722-prebuilt.qemu.org 04/01/2014 Workqueue: mlx5_ib_page_fault mlx5_ib_eqe_pf_action [mlx5_ib] Call Trace: dump_stack_lvl +0xa8/0xc0 lockdep_rcu_suspicious+0x1e6/0x260 xas_create+0xb8a/0xee0 xas_store+0x73/0x14c0 __xa_store+0x13c/0x220 ? xa_store_range+0x390/0x390 ? spin_bug+0x1d0/0x1d0 pagefault_mr +0xcb5/0x1390 [mlx5_ib] ? _raw_spin_unlock+0x1f/0x30 mlx5_ib_eqe_pf_action+0x3be/0x2620 [mlx5_ib] ? lockdep_hardirqs_on_prepare+0x400/0x400 ? mlx5_ib_invalidate_range+0xcb0/0xcb0 [mlx5_ib] process_one_work+0x7db/0x15f0 ? pwq_dec_nr_in_flight+0xda0/0xda0 ? assign_work +0x168/0x240 worker_thread+0x57d/0xcd0 ? rescuer_thread+0xc40/0xc40 kthread+0x3b3/0x800 ? kthread_is_per_cpu+0xb0/0xb0 ? lock_downgrade+0x680/0x680 ? do_raw_spin_lock+0x12d/0x270 ? spin_bug+0x1d0/0x1d0 ? finish_task_switch.isra.0+0x284/0x9e0 ? lockdep_hardirqs_on_prepare +0x284/0x400 ? kthread_is_per_cpu+0xb0/0xb0 ret_from_fork+0x2d/0x70 ? kthread_is_per_cpu +0xb0/0xb0 ret_from_fork_asm+0x11/0x20</pre>

CVE ID	Description
CVE-2025-38373	In the Linux kernel, the following vulnerability has been resolved: IB/mlx5: Fix potential deadlock in MR deregistration The issue arises when kcalloc() is invoked while holding umem_mutex or any other lock acquired under umem_mutex. This is problematic because kcalloc() can trigger fs_reclaim_acquire(), which may, in turn, invoke mmu_notifier_invalidate_range_start(). This function can lead to mlx5_ib_invalidate_range(), which attempts to acquire umem_mutex again, resulting in a deadlock. The problematic flow: CPU0 CPU1 <pre> ----- ----- # mlx5_ib_dereg_mr() # revoke_mr() # mutex_lock(&umem_odp->umem_mutex) # mlx5_mkey_cache_init() # mutex_lock(&dev->cache.rb_lock) # mlx5r_cache_create_ent_locked() # kcalloc(GFP_KERNEL) # fs_reclaim() # mmu_notifier_invalidate_range_start() # mlx5_ib_invalidate_range() # mutex_lock(&umem_odp->umem_mutex) # cache_ent_find_and_store() # mutex_lock(&dev- >cache.rb_lock) Additionally, when kcalloc() is called from within cache_ent_find_and_store(), we encounter the same deadlock due to re-acquisition of umem_mutex. Solve by releasing umem_mutex in dereg_mr() after umr_revoke_mr() and before acquiring rb_lock. This ensures that we don't hold umem_mutex while performing memory allocations that could trigger the reclaim path. This change prevents the deadlock by ensuring proper lock ordering and avoiding holding locks during memory allocation operations that could trigger the reclaim path. The following lockdep warning demonstrates the deadlock: python3/20557 is trying to acquire lock: ffff888387542128 (&umem_odp- >umem_mutex){+.+.}-{4:4}, at: mlx5_ib_invalidate_range+0x5b/0x550 [mlx5_ib] but task is already holding lock: ffffffff82f6b840 (mmu_notifier_invalidate_range_start){+.+.}-{0:0}, at: unmap_vmas+0x7b/0x1a0 which lock already depends on the new lock. the existing dependency chain (in reverse order) is: -> #3 (mmu_notifier_invalidate_range_start){+.+.}-{0:0}: fs_reclaim_acquire +0x60/0xd0 mem_cgroup_css_alloc+0x6f/0x9b0 cgroup_init_subsys+0xa4/0x240 cgroup_init +0x1c8/0x510 start_kernel+0x747/0x760 x86_64_start_reservations+0x25/0x30 x86_64_start_kernel +0x73/0x80 common_startup_64+0x129/0x138 -> #2 (fs_reclaim){+.+.}-{0:0}: fs_reclaim_acquire +0x91/0xd0 __kmmalloc_cache_noprof+0x4d/0x4c0 mlx5r_cache_create_ent_locked+0x75/0x620 [mlx5_ib] mlx5_mkey_cache_init+0x186/0x360 [mlx5_ib] mlx5_ib_stage_post_ib_reg_umr_init +0x3c/0x60 [mlx5_ib] __mlx5_ib_add+0x4b/0x190 [mlx5_ib] mlx5r_probe+0xd9/0x320 [mlx5_ib] auxiliary_bus_probe+0x42/0x70 really_probe+0xdb/0x360 __driver_probe_device+0x8f/0x130 driver_probe_device+0x1f/0xb0 __driver_attach+0xd4/0x1f0 bus_for_each_dev+0x79/0xd0 bus_add_driver+0xf0/0x200 driver_register+0x6e/0xc0 __auxiliary_driver_register+0x6a/0xc0 do_one_initcall+0x5e/0x390 do_init_module+0x88/0x240 init_module_from_file+0x85/0xc0 idempotent_init_module+0x104/0x300 __x64_sys_finit_module+0x68/0xc0 do_syscall_64+0x6d/0x140 entry_SYSCALL_64_after_hwframe+0x4b/0x53 -> #1 (&dev->cache.rb_lock){+.+.}-{4:4}: __mutex_lock+0x98/0xf10 __mlx5_ib_dereg_mr+0x6f2/0x890 [mlx5_ib] mlx5_ib_dereg_mr +0x21/0x110 [mlx5_ib] ib_dereg_mr_user+0x85/0x1f0 [ib_core] ---truncated---</pre>
CVE-2025-38374	In the Linux kernel, the following vulnerability has been resolved: optee: ffa: fix sleep in atomic context The OP-TEE driver registers the function notif_callback() for FF-A notifications. However, this function is called in an atomic context leading to errors like this when processing asynchronous notifications: <pre> BUG: sleeping function called from invalid context at kernel/locking/mutex.c:258 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 9, name: kworker/0:0 preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 CPU: 0 UID: 0 PID: 9 Comm: kworker/0:0 Not tainted 6.14.0-00019- g657536be0aa #13 Hardware name: linux,dummy-virt (DT) Workqueue: ffa_pcpu_irq_notification notif_pcpu_irq_work_fn Call trace: show_stack+0x18/0x24 (C) dump_stack_lvl+0x78/0x90 dump_stack+0x18/0x24 __might_resched+0x114/0x170 __might_sleep+0x48/0x98 mutex_lock+0x24/0x80 optee_get_msg_arg+0x7c/0x21c simple_call_with_arg+0x50/0xc0 optee_do_bottom_half+0x14/0x20 notif_callback+0x3c/0x48 handle_notif_callbacks+0x9c/0xe0 notif_get_and_handle+0x40/0x88 generic_exec_single+0x80/0xc0 smp_call_function_single +0xfc/0x1a0 notif_pcpu_irq_work_fn+0x2c/0x38 process_one_work+0x14c/0x2b4 worker_thread +0x2e4/0x3e0 kthread+0x13c/0x210 ret_from_fork+0x10/0x20 Fix this by adding work queue to process the notification in a non-atomic context.</pre>
CVE-2025-38375	In the Linux kernel, the following vulnerability has been resolved: virtio-net: ensure the received length does not exceed allocated size In xdp_linearize_page, when reading the following buffers from the ring, we forget to check the received length with the true allocate size. This can lead to an out-of-bound read. This commit adds that missing check.

CVE ID	Description
CVE-2025-38376	In the Linux kernel, the following vulnerability has been resolved: usb: chipidea: udc: disconnect/reconnect from host when do suspend/resume Shawn and John reported a hang issue during system suspend as below: - USB gadget is enabled as Ethernet - There is data transfer over USB Ethernet (scp a big file between host and device) - Device is going in/out suspend (echo mem > /sys/power/state) The root cause is the USB device controller is suspended but the USB bus is still active which caused the USB host continues to transfer data with device and the device continues to queue USB requests (in this case, a delayed TCP ACK packet trigger the issue) after controller is suspended, however the USB controller clock is already gated off. Then if udc driver access registers after that point, the system will hang. The correct way to avoid such issue is to disconnect device from host when the USB bus is not at suspend state. Then the host will receive disconnect event and stop data transfer in time. To continue make USB gadget device work after system resume, this will reconnect device automatically. To make usb wakeup work if USB bus is already at suspend state, this will keep connection for it only when USB device controller has enabled wakeup capability.
CVE-2025-38377	In the Linux kernel, the following vulnerability has been resolved: rose: fix dangling neighbour pointers in rose_rt_device_down() There are two bugs in rose_rt_device_down() that can cause use-after-free: 1. The loop bound `t->count` is modified within the loop, which can cause the loop to terminate early and miss some entries. 2. When removing an entry from the neighbour array, the subsequent entries are moved up to fill the gap, but the loop index `i` is still incremented, causing the next entry to be skipped. For example, if a node has three neighbours (A, A, B) with count=3 and A is being removed, the second A is not checked. i=0: (A, A, B) -> (A, B) with count=2 ^ checked i=1: (A, B) -> (A, B) with count=2 ^ checked (B, not A!) i=2: (doesn't occur because i < count is false) This leaves the second A in the array with count=2, but the rose_neigh structure has been freed. Code that accesses these entries assumes that the first `count` entries are valid pointers, causing a use-after-free when it accesses the dangling pointer. Fix both issues by iterating over the array in reverse order with a fixed loop bound. This ensures that all entries are examined and that the removal of an entry doesn't affect subsequent iterations.
CVE-2025-38382	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix iteration of extrefs during log replay At __inode_add_ref() when processing extrefs, if we jump into the next label we have an undefined value of victim_name.len, since we haven't initialized it before we did the goto. This results in an invalid memory access in the next iteration of the loop since victim_name.len was not initialized to the length of the name of the current extref. Fix this by initializing victim_name.len with the current extref's name length.
CVE-2025-38384	In the Linux kernel, the following vulnerability has been resolved: mtd: spinand: fix memory leak of ECC engine conf Memory allocated for the ECC engine conf is not released during spinand cleanup. Below kmemleak trace is seen for this memory leak: unreferenced object 0xfffff80064f00e0 (size 8): comm "swapper/0", pid 1, jiffies 4294937458 hex dump (first 8 bytes): 00 00 00 00 00 00 00 00 backtrace (crc 0): kmemleak_alloc+0x30/0x40 __kmalloccache_noprof+0x208/0x3c0 spinand_ondie_ecc_init_ctx+0x114/0x200 nand_ecc_init_ctx+0x70/0xa8 nanddev_ecc_engine_init+0xec/0x27c spinand_probe+0xa2c/0x1620 spi_mem_probe+0x130/0x21c spi_probe+0xf0/0x170 really_probe+0x17c/0x6e8 __driver_probe_device+0x17c/0x21c driver_probe_device+0x58/0x180 __device_attach_driver+0x15c/0x1f8 bus_for_each_drv+0xec/0x150 __device_attach+0x188/0x24c device_initial_probe+0x10/0x20 bus_probe_device+0x11c/0x160 Fix the leak by calling nanddev_ecc_engine_cleanup() inside spinand_cleanup().

CVE ID	Description
CVE-2025-38385	In the Linux kernel, the following vulnerability has been resolved: net: usb: lan78xx: fix WARN in __netif_napi_del_locked on disconnect Remove redundant netif_napi_del() call from disconnect path. A WARN may be triggered in __netif_napi_del_locked() during USB device disconnect: WARNING: CPU: 0 PID: 11 at net/core/dev.c:7417 __netif_napi_del_locked+0x2b4/0x350 This happens because netif_napi_del() is called in the disconnect path while NAPI is still enabled. However, it is not necessary to call netif_napi_del() explicitly, since unregister_netdev() will handle NAPI teardown automatically and safely. Removing the redundant call avoids triggering the warning. Full trace: lan78xx 1-1:1.0 enu1: Failed to read register index 0x000000c4. ret = -ENODEV lan78xx 1-1:1.0 enu1: Failed to set MAC down with error -ENODEV lan78xx 1-1:1.0 enu1: Link is Down lan78xx 1-1:1.0 enu1: Failed to read register index 0x00000120. ret = -ENODEV -----[cut here]----- WARNING: CPU: 0 PID: 11 at net/core/dev.c:7417 __netif_napi_del_locked+0x2b4/0x350 Modules linked in: flexcan can_dev fuse CPU: 0 UID: 0 PID: 11 Comm: kworker/0:1 Not tainted 6.16.0-rc2-00624-ge926949dab03 #9 PREEMPT Hardware name: SKOV IMX8MP CPU revC - bd500 (DT) Workqueue: usb_hub_wq hub_event pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYP=) pc : __netif_napi_del_locked+0x2b4/0x350 lr : __netif_napi_del_locked+0x7c/0x350 sp : fffffffc085b673c0 x29: fffffffc085b673c0 x28: fffffff800b7f2000 x27: fffffff800b7f20d8 x26: fffffff80110bcf58 x25: fffffff80110bd978 x24: 1ffffff0022179eb x23: fffffff80110bc000 x22: fffffff800b7f5000 x21: fffffff80110bc000 x20: fffffff80110bcf38 x19: fffffff80110bcf28 x18: dffffffc000000000 x17: fffffffc081578940 x16: fffffffc08284cee0 x15: 0000000000000028 x14: 0000000000000006 x13: 0000000000040000 x12: fffffffb0022179e8 x11: 1ffffff0022179e7 x10: fffffffb0022179e7 x9 : dffffffc000000000 x8 : 0000004ffde8619 x7 : fffffff80110bcf3f x6 : 0000000000000001 x5 : fffffff80110bcf38 x4 : fffffff80110bcf38 x3 : 0000000000000000 x2 : 0000000000000000 x1 : 1ffffff0022179e7 x0 : 0000000000000000 Call trace: __netif_napi_del_locked+0x2b4/0x350 (P) lan78xx_disconnect+0xf4/0x360 usb_unbind_interface+0x158/0x718 device_remove+0x100/0x150 device_release_driver_internal+0x308/0x478 device_release_driver+0x1c/0x30 bus_remove_device+0x1a8/0x368 device_del+0x2e0/0x7b0 usb_disable_device+0x244/0x540 usb_disconnect+0x220/0x758 hub_event+0x105c/0x35e0 process_one_work+0x760/0x17b0 worker_thread+0x768/0xce8 kthread+0x3bc/0x690 ret_from_fork+0x10/0x20 irq event stamp: 211604 hardirqs last enabled at (211603): [<ffffffc0828cc9ec>] _raw_spin_unlock_irqrestore+0x84/0x98 hardirqs last disabled at (211604): [<ffffffc0828a9a84>] el1_dbg+0x24/0x80 softirqs last enabled at (211296): [<ffffffc080095f10>] handle_softirqs+0x820/0xbc8 softirqs last disabled at (210993): [<ffffffc080010288>] __do_softirq+0x18/0x20 ---[end trace 0000000000000000]--- lan78xx 1-1:1.0 enu1: failed to kill vid 0081/0
CVE-2025-38386	In the Linux kernel, the following vulnerability has been resolved: ACPICA: Refuse to evaluate a method if arguments are missing As reported in [1], a platform firmware update that increased the number of method parameters and forgot to update a least one of its callers, caused ACPICA to crash due to use-after-free. Since this a result of a clear AML issue that arguably cannot be fixed up by the interpreter (it cannot produce missing data out of thin air), address it by making ACPICA refuse to evaluate a method if the caller attempts to pass fewer arguments than expected to it.

CVE ID	Description
CVE-2025-38387	In the Linux kernel, the following vulnerability has been resolved: RDMA/mlx5: Initialize obj_event->obj_sub_list before xa_insert The obj_event may be loaded immediately after inserted, then if the list_head is not initialized then we may get a poisonous pointer. This fixes the crash below: <pre> mlx5_core 0000:03:00.0: MLX5E: StrdRq(1) RqSz(8) StrdSz(2048) RxCqeCmprss(0 enhanced) mlx5_core.sf mlx5_core.sf.4: firmware version: 32.38.3056 mlx5_core 0000:03:00.0 en3f0pf0sf2002: renamed from eth0 mlx5_core.sf mlx5_core.sf.4: Rate limit: 127 rates are supported, range: 0Mbps to 195312Mbps IPv6: ADDRCONF(NETDEV_CHANGE): en3f0pf0sf2002: link becomes ready Unable to handle kernel NULL pointer dereference at virtual address 0000000000000060 Mem abort info: ESR = 0x96000006 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 Data abort info: ISV = 0, ISS = 0x00000006 CM = 0, WnR = 0 user pgtable: 4k pages, 48-bit VAs, pgdp=00000007760fb000 [0000000000000060] pgd=000000076fd7003, p4d=000000076fd7003, pud=0000000777841003, pmd=0000000000000000 Internal error: Oops: 96000006 [#1] SMP Modules linked in: ipmb_host(OE) act_mirred(E) cls_flow(E) sch_ingress(E) mptcp_diag(E) udp_diag(E) raw_diag(E) unix_diag(E) tcp_diag(E) inet_diag(E) binfmt_misc(E) bonding(OE) rdma_ucm(OE) rdma_cm(OE) iw_cm(OE) ib_ipoib(OE) ib_cm(OE) isofs(E) cdrom(OE) mst_pciconf(OE) ib_umad(OE) mlx5_ib(OE) ipmb_dev_int(OE) mlx5_core(OE) kpatch_15237886(OEK) mlxdevm(OE) auxiliary(OE) ib_uverbs(OE) ib_core(OE) psample(E) mlxfw(OE) tls(E) sunrpc(E) vfat(E) fat(E) crct10dif_ce(E) ghash_ce(E) sha1_ce(E) sbas_gwdt(E) virtio_console(E) ext4(E) mbcache(E) jbd2(E) xfs(E) libcrc32c(E) mmc_block(E) virtio_net(E) net_failover(E) failover(E) sha2_ce(E) sha256_arm64(E) nvme(OE) nvme_core(OE) gpio_mxlbf3(OE) mlx_compat(OE) mxlbf_pmc(OE) i2c_mxlbf(OE) sdhci_of_dwcmshc(OE) pinctrl_mxlbf3(OE) mxlbf_pka(OE) gpio_generic(E) i2c_core(E) mmc_core(E) mxlbf_gige(OE) vitesse(E) pwr_mxlbf(OE) mxlbf_tmfifo(OE) micrel(E) mxlbf_bootctl(OE) virtio_ring(E) virtio(E) ipmi_devintf(E) ipmi_msghandler(E) [last unloaded: mst_pci] CPU: 11 PID: 20913 Comm: rte-worker-11 Kdump: loaded Tainted: G OE K 5.10.134-13.1.an8.aarch64 #1 Hardware name: https://www.mellanox.com BlueField-3 SmartNIC Main Card/BlueField-3 SmartNIC Main Card, BIOS 4.2.2.12968 Oct 26 2023 pstate: a0400089 (NzCv daIf +PAN -UAO -TCO BTYP=--) pc : dispatch_event_fd+0x68/0x300 [mlx5_ib] lr : devx_event_notifier+0xcc/0x228 [mlx5_ib] sp : ffff80001005bcf0 x29: ffff80001005bcf0 x28: 0000000000000001 x27: ffff244e0740a1d8 x26: ffff244e0740a1d0 x25: ffffd56beff5ae0 x24: ffffd56bf911618 x23: ffff244e0596a480 x22: ffff244e0596a480 x21: ffff244d8312ad90 x20: ffff244e0596a480 x19: ffffffff00000000 x18: 0000000000000000 x17: 0000000000000000 x16: fffd56be66d620 x15: 0000000000000000 x14: 0000000000000000 x13: 0000000000000000 x12: 0000000000000000 x11: 0000000000000040 x10: ffffd56bfcafb50 x9: ffffd5655c25f2c x8 : 0000000000000010 x7 : 0000000000000000 x6 : ffff24545a2e24b8 x5 : 0000000000000003 x4 : ffff80001005bd28 x3 : 0000000000000000 x2 : 0000000000000000 x1 : ffff244e0596a480 x0 : ffff244d8312ad90 Call trace: dispatch_event_fd+0x68/0x300 [mlx5_ib] devx_event_notifier +0xcc/0x228 [mlx5_ib] atomic_notifier_call_chain+0x58/0x80 mlx5_eq_async_int+0x148/0x2b0 [mlx5_core] atomic_notifier_call_chain+0x58/0x80 irq_int_handler+0x20/0x30 [mlx5_core] __handle_irq_event_percpu+0x60/0x220 handle_irq_event_percpu+0x3c/0x90 handle_irq_event +0x58/0x158 handle_fasteoi_irq+0xfc/0x188 generic_handle_irq+0x34/0x48 ... </pre>
CVE-2025-38388	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_ffa: Replace mutex with rwlock to avoid sleep in atomic context The current use of a mutex to protect the notifier hashtable accesses can lead to issues in the atomic context. It results in the below kernel warnings: <pre> BUG: sleeping function called from invalid context at kernel/locking/mutex.c:258 in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 9, name: kworker/0:0 preempt_count: 1, expected: 0 RCU nest depth: 0, expected: 0 CPU: 0 UID: 0 PID: 9 Comm: kworker/0:0 Not tainted 6.14.0 #4 Workqueue: ffa_pcpu_irq_notification notif_pcpu_irq_work_fn Call trace: show_stack +0x18/0x24 (C) dump_stack_lvl+0x78/0x90 dump_stack+0x18/0x24 __might_resched+0x114/0x170 __might_sleep+0x48/0x98 mutex_lock+0x24/0x80 handle_notif_callbacks+0x54/0xe0 notif_get_and_handle+0x40/0x88 generic_exec_single+0x80/0xc0 smp_call_function_single +0xfc/0x1a0 notif_pcpu_irq_work_fn+0x2c/0x38 process_one_work+0x14c/0x2b4 worker_thread +0x2e4/0x3e0 kthread+0x13c/0x210 ret_from_fork+0x10/0x20 To address this, replace the mutex with an rwlock to protect the notifier hashtable accesses. This ensures that read-side locking does not sleep and multiple readers can acquire the lock concurrently, avoiding unnecessary contention and potential deadlocks. Writer access remains exclusive, preserving correctness. This change resolves warnings from lockdep about potential sleep in atomic context. </pre>

CVE ID	Description
CVE-2025-38389	In the Linux kernel, the following vulnerability has been resolved: drm/i915/gt: Fix timeline left held on VMA alloc error The following error has been reported sporadically by CI when a test unbinds the i915 driver on a ring submission platform: <4> [239.330153] ----- [cut here]----- <4> [239.330166] i915 0000:00:02.0: [drm] drm_WARN_ON(dev_priv->mm.shrink_count) <4> [239.330196] WARNING: CPU: 1 PID: 18570 at drivers/gpu/drm/i915/i915_gem.c:1309 i915_gem_cleanup_early+0x13e/0x150 [i915] ... <4> [239.330640] RIP: 0010:i915_gem_cleanup_early+0x13e/0x150 [i915] ... <4> [239.330942] Call Trace: <4> [239.330944] <TASK> <4> [239.330949] i915_driver_late_release+0x2b/0xa0 [i915] <4> [239.331202] i915_driver_release+0x86/0xa0 [i915] <4> [239.331482] devm_drm_dev_init_release+0x61/0x90 <4> [239.331494] devm_action_release+0x15/0x30 <4> [239.331504] release_nodes+0x3d/0x120 <4> [239.331517] devres_release_all+0x96/0xd0 <4> [239.331533] device_unbind_cleanup+0x12/0x80 <4> [239.331543] device_release_driver_internal+0x23a/0x280 <4> [239.331550] ? bus_find_device+0xa5/0xe0 <4> [239.331563] device_driver_detach+0x14/0x20 ... <4> [357.719679] ---[end trace 0000000000000000]--- If the test also unloads the i915 module then that's followed with: <3> [357.787478] ===== <3> [357.788006] BUG i915_vma (Tainted: G U W N): Objects remaining on __kmem_cache_shutdown() <3> [357.788031] ----- <3> [357.788204] Object 0xffff888109e7f480 @offset=29824 <3> [357.788670] Allocated in i915_vma_instance+0xee/0xc10 [i915] age=292729 cpu=4 pid=2244 <4> [357.788994] i915_vma_instance+0xee/0xc10 [i915] <4> [357.789290] init_status_page+0x7b/0x420 [i915] <4> [357.789532] intel_engines_init+0x1d8/0x980 [i915] <4> [357.789772] intel_gt_init+0x175/0x450 [i915] <4> [357.790014] i915_gem_init+0x113/0x340 [i915] <4> [357.790281] i915_driver_probe+0x847/0xed0 [i915] <4> [357.790504] i915_pci_probe+0xe6/0x220 [i915] ... Closer analysis of CI results history has revealed a dependency of the error on a few IGT tests, namely: - igt@api_intel_allocator@fork-simple-stress-signal, - igt@api_intel_allocator@two-level-inception-interruptible, - igt@gem_linear_blits@interruptible, - igt@prime_mmap_coherency@ioctl-errors, which invisibly trigger the issue, then exhibited with first driver unbind attempt. All of the above tests perform actions which are actively interrupted with signals. Further debugging has allowed to narrow that scope down to DRM_IOCTL_I915_GEM_EXECBUFFER2, and ring_context_alloc(), specific to ring submission, in particular. If successful then that function, or its execlists or GuC submission equivalent, is supposed to be called only once per GEM context engine, followed by raise of a flag that prevents the function from being called again. The function is expected to unwind its internal errors itself, so it may be safely called once more after it returns an error. In case of ring submission, the function first gets a reference to the engine's legacy timeline and then allocates a VMA. If the VMA allocation fails, e.g. when i915_vma_instance() called from inside is interrupted with a signal, then ring_context_alloc() fails, leaving the timeline held referenced. On next I915_GEM_EXECBUFFER2 IOCTL, another reference to the timeline is got, and only that last one is put on successful completion. As a consequence, the legacy timeline, with its underlying engine status page's VMA object, is still held and not released on driver unbind. Get the legacy timeline only after successful allocation of the context engine's VMA. v2: Add a note on other submission methods (Krzysztof Karas): Both execlists and GuC submission use lrc_alloc() which seems free from a similar issue. (cherry picked from commit cc43422b3cc79eacff4c5a8ba0d224688ca9dd4f)
CVE-2025-38390	In the Linux kernel, the following vulnerability has been resolved: firmware: arm_ffa: Fix memory leak by freeing notifier callback node Commit e0573444edbf ("firmware: arm_ffa: Add interfaces to request notification callbacks") adds support for notifier callbacks by allocating and inserting a callback node into a hashtable during registration of notifiers. However, during unregistration, the code only removes the node from the hashtable without freeing the associated memory, resulting in a memory leak. Resolve the memory leak issue by ensuring the allocated notifier callback node is properly freed after it is removed from the hashtable entry.
CVE-2025-38391	In the Linux kernel, the following vulnerability has been resolved: usb: typec: altmodes/displayport: do not index invalid pin_assignments A poorly implemented DisplayPort Alt Mode port partner can indicate that its pin assignment capabilities are greater than the maximum value, DP_PIN_ASSIGN_F. In this case, calls to pin_assignment_show will cause a BRK exception due to an out of bounds array access. Prevent for loop in pin_assignment_show from accessing invalid values in pin_assignments by adding DP_PIN_ASSIGN_MAX value in typec_dp.h and using i < DP_PIN_ASSIGN_MAX as a loop condition.

CVE ID	Description
CVE-2025-38392	In the Linux kernel, the following vulnerability has been resolved: idpf: convert control queue mutex to a spinlock With VIRTCHNL2_CAP_MACFILTER enabled, the following warning is generated on module load: [324.701677] BUG: sleeping function called from invalid context at kernel/locking/mutex.c:578 [324.701684] in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 1582, name: NetworkManager [324.701689] preempt_count: 201, expected: 0 [324.701693] RCU nest depth: 0, expected: 0 [324.701697] 2 locks held by NetworkManager/1582: [324.701702] #0: ffffffff9f7be770 (rtnl_mutex){....}-{3:3}, at: rtnl_newlink+0x791/0x21e0 [324.701730] #1: ff1100216c380368 (_xmit_ETHER){....}-{2:2}, at: __dev_open+0x3f0/0x870 [324.701749] Preemption disabled at: [324.701752] [<fffffff9cd23b9d>] __dev_open+0x3dd/0x870 [324.701765] CPU: 30 UID: 0 PID: 1582 Comm: NetworkManager Not tainted 6.15.0-rc5+ #2 PREEMPT(voluntary) [324.701771] Hardware name: Intel Corporation M50FCP2SBSTD/M50FCP2SBSTD, BIOS SE5C741.86B.01.01.0001.2211140926 11/14/2022 [324.701774] Call Trace: [324.701777] <TASK> [324.701779] dump_stack_lvl+0x5d/0x80 [324.701788] ? __dev_open+0x3dd/0x870 [324.701793] __might_resched.cold+0x1ef/0x23d <..> [324.701818] __mutex_lock+0x113/0x1b80 <..> [324.701917] idpf_ctlq_clean_sq+0xad/0x4b0 [idpf] [324.701935] ? kasan_save_track+0x14/0x30 [324.701941] idpf_mb_clean+0x143/0x380 [idpf] <..> [324.701991] idpf_send_mb_msg+0x111/0x720 [idpf] [324.702009] idpf_vc_xn_exec+0x4cc/0x990 [idpf] [324.702021] ? rcu_is_watching+0x12/0xc0 [324.702035] idpf_add_del_mac_filters+0x3ed/0xb50 [idpf] <..> [324.702122] __hw_addr_sync_dev+0x1cf/0x300 [324.702126] ? find_held_lock+0x32/0x90 [324.702134] idpf_set_rx_mode+0x317/0x390 [idpf] [324.702152] __dev_open+0x3f8/0x870 [324.702159] ? __pfx__dev_open+0x10/0x10 [324.702174] __dev_change_flags+0x443/0x650 <..> [324.702208] netif_change_flags+0x80/0x160 [324.702218] do_setlink.isra.0+0x16a0/0x3960 <..> [324.702349] rtnl_newlink+0x12fd/0x21e0 The sequence is as follows: rtnl_newlink()-> __dev_change_flags()-> __dev_open()-> dev_set_rx_mode() -> # disables BH and grabs "dev->addr_list_lock" idpf_set_rx_mode()-> # proceed only if VIRTCHNL2_CAP_MACFILTER is ON __dev_uc_sync()-> idpf_add_mac_filter-> idpf_add_del_mac_filters-> idpf_send_mb_msg()-> idpf_mb_clean()-> idpf_ctlq_clean_sq() # mutex_lock(cq_lock) Fix by converting cq_lock to a spinlock. All operations under the new lock are safe except freeing the DMA memory, which may use vunmap(). Fix by requesting a contiguous physical memory for the DMA mapping.
CVE-2025-38393	In the Linux kernel, the following vulnerability has been resolved: NFSv4/pNFS: Fix a race to wake on NFS_LAYOUT_DRAIN We found a few different systems hung up in writeback waiting on the same page lock, and one task waiting on the NFS_LAYOUT_DRAIN bit in pnfs_update_layout(), however the pnfs_layout_hdr's plh_outstanding count was zero. It seems most likely that this is another race between the waiter and waker similar to commit ed0172af5d6f ("SUNRPC: Fix a race to wake a sync task"). Fix it up by applying the advised barrier.
CVE-2025-38395	In the Linux kernel, the following vulnerability has been resolved: regulator: gpio: Fix the out-of-bounds access to drvdata::gpiods drvdata::gpiods is supposed to hold an array of 'gpio_desc' pointers. But the memory is allocated for only one pointer. This will lead to out-of-bounds access later in the code if 'config::ngpios' is > 1. So fix the code to allocate enough memory to hold 'config::ngpios' of GPIO descriptors. While at it, also move the check for memory allocation failure to be below the allocation to make it more readable.
CVE-2025-38396	In the Linux kernel, the following vulnerability has been resolved: fs: export anon_inode_make_secure_inode() and fix secretmem LSM bypass Export anon_inode_make_secure_inode() to allow KVM guest_memfd to create anonymous inodes with proper security context. This replaces the current pattern of calling alloc_anon_inode() followed by inode_init_security_anon() for creating security context manually. This change also fixes a security regression in secretmem where the S_PRIVATE flag was not cleared after alloc_anon_inode(), causing LSM/SELinux checks to be bypassed for secretmem file descriptors. As guest_memfd currently resides in the KVM module, we need to export this symbol for use outside the core kernel. In the future, guest_memfd might be moved to core-mm, at which point the symbols no longer would have to be exported. When/if that happens is still unclear.
CVE-2025-38399	In the Linux kernel, the following vulnerability has been resolved: scsi: target: Fix NULL pointer dereference in core_scsi3_decode_spec_i_port() The function core_scsi3_decode_spec_i_port(), in its error code path, unconditionally calls core_scsi3_lunacl_undepend_item() passing the dest_se_deve pointer, which may be NULL. This can lead to a NULL pointer dereference if dest_se_deve remains unset. SPC-3 PR SPEC_I_PT: Unable to locate dest_tpg Unable to handle kernel paging request at virtual address dfff800000000012 Call trace: core_scsi3_lunacl_undepend_item+0x2c/0xf0 [target_core_mod] (P) core_scsi3_decode_spec_i_port+0x120c/0x1c30 [target_core_mod] core_scsi3_emulate_pro_register+0x6b8/0xcd8 [target_core_mod] target_scsi3_emulate_pr_out+0x56c/0x840 [target_core_mod] Fix this by adding a NULL check before calling core_scsi3_lunacl_undepend_item()

CVE ID	Description
CVE-2025-38400	In the Linux kernel, the following vulnerability has been resolved: nfs: Clean up /proc/net/rpc/nfs when nfs_fs_proc_net_init() fails. syzbot reported a warning below [1] following a fault injection in nfs_fs_proc_net_init(). [0] When nfs_fs_proc_net_init() fails, /proc/net/rpc/nfs is not removed. Later, rpc_proc_exit() tries to remove /proc/net/rpc, and the warning is logged as the directory is not empty. Let's handle the error of nfs_fs_proc_net_init() properly. [0]: FAULT_INJECTION: forcing a failure. name failslab, interval 1, probability 0, space 0, times 0 CPU: 1 UID: 0 PID: 6120 Comm: syz.2.27 Not tainted 6.16.0-rc1-syzkaller-00010-g2c4a1f3fe03e #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 Call Trace: <TASK> dump_stack_lvl (lib/dump_stack.c:123) should_fail_ex (lib/fault-inject.c:73 lib/fault-inject.c:174) should_failslab (mm/failslab.c:46) kmem_cache_alloc_noprof (mm/slub.c:4178 mm/slub.c:4204) __proc_create (fs/proc/generic.c:427) proc_create_reg (fs/proc/generic.c:554) proc_create_net_data (fs/proc/proc_net.c:120) nfs_fs_proc_net_init (fs/nfs/client.c:1409) nfs_net_init (fs/nfs/inode.c:2600) ops_init (net/core/net_namespace.c:138) setup_net (net/core/net_namespace.c:443) copy_net_ns (net/core/net_namespace.c:576) create_new_namespaces (kernel/nsproxy.c:110) unshare_nsproxy_namespaces (kernel/nsproxy.c:218 (discriminator 4)) ksys_unshare (kernel/fork.c:3123) __x64_sys_unshare (kernel/fork.c:3190) do_syscall_64 (arch/x86/entry/syscall_64.c:63 arch/x86/entry/syscall_64.c:94) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) </TASK> [1]: remove_proc_entry: removing non-empty directory 'net/rpc', leaking at least 'nfs' WARNING: CPU: 1 PID: 6120 at fs/proc/generic.c:727 remove_proc_entry+0x45e/0x530 fs/proc/generic.c:727 Modules linked in: CPU: 1 UID: 0 PID: 6120 Comm: syz.2.27 Not tainted 6.16.0-rc1-syzkaller-00010-g2c4a1f3fe03e #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 RIP: 0010:remove_proc_entry+0x45e/0x530 fs/proc/generic.c:727 Code: 3c 02 00 0f 85 85 00 00 00 48 8b 93 d8 00 00 00 4d 89 f0 4c 89 e9 48 c7 c6 40 ba a2 8b 48 c7 c7 60 b9 a2 8b e8 33 81 1d ff 90 <0f> 0b 90 90 e9 5f fe ff ff e8 04 69 5e ff 90 48 b8 00 00 00 00 00 00 RSP: 0018:ffff90003637b08 EFLAGS: 0010282 RAX: 0000000000000000 RBX: ffff88805f534140 RCX: fffffff817a92c8 RDX: ffff88807da99e00 RSI: fffffff817a92d5 RDI: 0000000000000001 RBP: ffff888033431ac0 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000001 R12: ffff888033431a00 R13: ffff888033431ae4 R14: ffff888033184724 R15: dffffc0000000000 FS: 0000555580328500(0000) GS:ffff888124a62000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000008005003 CR2: 00007f1733743e0 CR3: 000000007f618000 CR4: 00000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> sunrpc_exit_net+0x46/0x90 net/sunrpc/sunrpc_syms.c:76 ops_exit_list net/core/net_namespace.c:200 [inline] ops_undo_list+0x2eb/0xab0 net/core/net_namespace.c:253 setup_net+0x2e1/0x510 net/core/net_namespace.c:457 copy_net_ns+0x2a6/0x5f0 net/core/net_namespace.c:574 create_new_namespaces+0x3ea/0xa90 kernel/nsproxy.c:110 unshare_nsproxy_namespaces+0xc0/0x1f0 kernel/nsproxy.c:218 ksys_unshare+0x45b/0xa40 kernel/fork.c:3121 __do_sys_unshare kernel/fork.c:3192 [inline] __se_sys_unshare kernel/fork.c:3190 [inline] __x64_sys_unshare+0x31/0x40 kernel/fork.c:3190 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x490 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fa1a6b8e929 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c ---truncated---
CVE-2025-38401	In the Linux kernel, the following vulnerability has been resolved: mtk-sd: Prevent memory corruption from DMA map failure If msdc_prepare_data() fails to map the DMA region, the request is not prepared for data receiving, but msdc_start_data() proceeds the DMA with previous setting. Since this will lead a memory corruption, we have to stop the request operation soon after the msdc_prepare_data() fails to prepare it.

CVE ID	Description
CVE-2025-38402	In the Linux kernel, the following vulnerability has been resolved: idpf: return 0 size for RSS key if not supported Returning -EOPNOTSUPP from function returning u32 is leading to cast and invalid size value as a result. -EOPNOTSUPP as a size probably will lead to allocation fail. Command: ethtool -x eth0 It is visible on all devices that don't have RSS caps set. [136.615917] Call Trace: [136.615921] <TASK> [136.615927] ? __warn+0x89/0x130 [136.615942] ? __alloc_frozen_pages_noprof+0x322/0x330 [136.615953] ? report_bug+0x164/0x190 [136.615968] ? handle_bug+0x58/0x90 [136.615979] ? exc_invalid_op+0x17/0x70 [136.615987] ? asm_exc_invalid_op+0x1a/0x20 [136.616001] ? rss_prepare_get.constprop.0+0xb9/0x170 [136.616016] ? __alloc_frozen_pages_noprof+0x322/0x330 [136.616028] __alloc_pages_noprof+0xe/0x20 [136.616038] __kmalloc_large_node+0x80/0x110 [136.616072] __kmalloc_large_node_noprof+0x1d/0xa0 [136.616081] __kmalloc_noprof+0x32c/0x4c0 [136.616098] ? rss_prepare_get.constprop.0+0xb9/0x170 [136.616105] rss_prepare_get.constprop.0+0xb9/0x170 [136.616114] ethnl_default_doit+0x107/0x3d0 [136.616131] genl_family_rcv_msg_doit+0x100/0x160 [136.616147] genl_rcv_msg+0x1b8/0x2c0 [136.616156] ? __pfx_ethnl_default_doit+0x10/0x10 [136.616168] ? __pfx_genl_rcv_msg+0x10/0x10 [136.616176] netlink_rcv_skb+0x58/0x110 [136.616186] genl_rcv+0x28/0x40 [136.616195] netlink_unicast+0x19b/0x290 [136.616206] netlink_sendmsg+0x222/0x490 [136.616215] __sys_sendto+0x1fd/0x210 [136.616233] __x64_sys_sendto+0x24/0x30 [136.616242] do_syscall_64+0x82/0x160 [136.616252] ? __sys_recvmsg+0x83/0xe0 [136.616265] ? syscall_exit_to_user_mode+0x10/0x210 [136.616275] ? do_syscall_64+0x8e/0x160 [136.616282] ? __count_memcg_events+0xa1/0x130 [136.616295] ? count_memcg_events.constprop.0+0x1a/0x30 [136.616306] ? handle_mm_fault+0xae/0x2d0 [136.616319] ? do_user_addr_fault+0x379/0x670 [136.616328] ? clear_bhb_loop+0x45/0xa0 [136.616340] ? clear_bhb_loop+0x45/0xa0 [136.616349] ? clear_bhb_loop+0x45/0xa0 [136.616359] entry_SYSCALL_64_after_hwframe+0x76/0x7e [136.616369] RIP: 0033:0x7fd30ba7b047 [136.616376] Code: 0c 00 f7 d8 64 89 02 48 c7 c0 ff ff ff eb b8 0f 1f 00 f3 0f 1e fa 80 3d bd d5 0c 00 00 41 89 ca 74 10 b8 2c 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 71 c3 55 48 83 ec 30 44 89 4c 24 2c 4c 89 44 [136.616381] RSP: 002b:00007ffde1796d68 EFLAGS: 00000202 ORIG_RAX: 000000000000002c [136.616388] RAX: ffffffffda RBX: 000055d7bd89f2a0 RCX: 00007fd30ba7b047 [136.616392] RDX: 0000000000000028 RSI: 000055d7bd89f3b0 RDI: 0000000000000003 [136.616396] RBP: 00007ffde1796e10 R08: 00007fd30bb4e200 R09: 000000000000000c [136.616399] R10: 0000000000000000 R11: 0000000000000202 R12: 000055d7bd89f340 [136.616403] R13: 000055d7bd89f3b0 R14: 000055d78943f200 R15: 0000000000000000
CVE-2025-38403	In the Linux kernel, the following vulnerability has been resolved: vsock/vmci: Clear the vmci transport packet properly when initializing it In vmci_transport_packet_init memset the vmci_transport_packet before populating the fields to avoid any uninitialised data being left in the structure.
CVE-2025-38405	In the Linux kernel, the following vulnerability has been resolved: nvmet: fix memory leak of bio integrity If nvmet receives commands with metadata there is a continuous memory leak of kmalloc-128 slab or more precisely bio->bi_integrity. Since commit bf4c89fc8797 ("block: don't call bio_uninit from bio_endio") each user of bio_init has to use bio_uninit as well. Otherwise the bio integrity is not getting free. Nvmet uses bio_init for inline bios. Uninit the inline bio to complete deallocation of integrity in bio.
CVE-2025-38406	In the Linux kernel, the following vulnerability has been resolved: wifi: ath6kl: remove WARN on bad firmware input If the firmware gives bad input, that's nothing to do with the driver's stack at this point etc., so the WARN_ON() doesn't add any value. Additionally, this is one of the top syzbot reports now. Just print a message, and as an added bonus, print the sizes too.

CVE ID	Description
CVE-2025-38407	In the Linux kernel, the following vulnerability has been resolved: riscv: cpu_ops_sbi: Use static array for boot_data Since commit 6b9f29b81b15 ("riscv: Enable pcpu page first chunk allocator"), if NUMA is enabled, the page percpu allocator may be used on very sparse configurations, or when requested on boot with percpu_alloc=page. In that case, percpu data gets put in the vmalloc area. However, sbi_hsm_hart_start() needs the physical address of a sbi_hart_boot_data, and simply assumes that __pa() would work. This causes the just started hart to immediately access an invalid address and hang. Fortunately, struct sbi_hart_boot_data is not too large, so we can simply allocate an array for boot_data statically, putting it in the kernel image. This fixes NUMA=y SMP boot on Sophgo SG2042. To reproduce on QEMU: Set CONFIG_NUMA=y and CONFIG_DEBUG_VIRTUAL=y, then run with: qemu-system-riscv64 -M virt -smp 2 -nographic \ -kernel arch/riscv/boot/Image \ -append "percpu_alloc=page" Kernel output: [0.000000] Booting Linux on hartid 0 [0.000000] Linux version 6.16.0-rc1 (dram@sakuya) (riscv64-unknown-linux-gnu-gcc (GCC) 14.2.1 20250322, GNU ld (GNU Binutils) 2.44) #11 SMP Tue Jun 24 14:56:22 CST 2025 ... [0.000000] percpu: 28 4K pages/cpu s85784 r8192 d20712 ... [0.083192] smp: Bringing up secondary CPUs ... [0.086722] -----[cut here]----- [0.086849] virt_to_phys used for non-linear address: (__ptrval__) (0xff2000000001d080) [0.088001] WARNING: CPU: 0 PID: 1 at arch/riscv/mm/physaddr.c:14 __virt_to_phys+0xae/0xe8 [0.088376] Modules linked in: [0.088656] CPU: 0 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.16.0-rc1 #11 NONE [0.088833] Hardware name: riscv-virtio,qemu (DT) [0.088948] epc : __virt_to_phys+0xae/0xe8 [0.089001] ra : __virt_to_phys+0xae/0xe8 [0.089037] epc : ffffffff80021eaa ra : ffffffff80021eaa sp : ff2000000004bbc0 [0.089057] gp : ffffffff817f49c0 tp : ff60000001d60000 t0 : 5f6f745f74726976 [0.089076] t1 : 0000000000000076 t2 : 705f6f745f747269 s0 : ff2000000004bbe0 [0.089095] s1 : ff2000000001d080 a0 : 0000000000000000 a1 : 0000000000000000 [0.089113] a2 : 0000000000000000 a3 : 0000000000000000 a4 : 0000000000000000 [0.089131] a5 : 0000000000000000 a6 : 0000000000000000 a7 : 0000000000000000 [0.089155] s2 : ffffffff8130dc00 s3 : 0000000000000001 s4 : 0000000000000001 [0.089174] s5 : ffffffff8185eff8 s6 : ff2000007f1eb000 s7 : ffffffff8002a2ec [0.089193] s8 : 0000000000000001 s9 : 0000000000000001 s10: 0000000000000000 [0.089211] s11: 0000000000000000 t3 : ffffffff8180a9f7 t4 : ffffffff8180a9f7 [0.089960] t5 : ffffffff8180a9f8 t6 : ff2000000004b9d8 [0.089984] status: 0000000200000120 badaddr: ffffffff80021eaa cause: 0000000000000003 [0.090101] [<ffffffffff80021eaa>] __virt_to_phys+0xae/0xe8 [0.090228] [<ffffffffff8001d796>] sbi_cpu_start+0x6e/0xe8 [0.090247] [<ffffffffff8001a5da>] __cpu_up+0x1e/0x8c [0.090260] [<ffffffffff8002a32e>] bringup_cpu+0x42/0x258 [0.090277] [<ffffffffff8002914c>] cpuhp_invoke_callback+0xe0/0x40c [0.090292] [<ffffffffff800294e0>] __cpuhp_invoke_callback_range+0x68/0xfc [0.090320] [<ffffffffff8002a96a>] __cpu_up+0x11a/0x244 [0.090334] [<ffffffffff8002aae6>] cpu_up+0x52/0x90 [0.090384] [<ffffffffff80c09350>] bringup_nonboot_cpus+0x78/0x118 [0.090411] [<ffffffffff80c11060>] smp_init+0x34/0xb8 [0.090425] [<ffffffffff80c01220>] kernel_init_freeable+0x148/0x2e4 [0.090442] [<ffffffffff80b83802>] kernel_init+0x1e/0x14c [0.090455] [<ffffffffff800124ca>] ret_from_fork_kernel+0xe/0xf0 [0.090471] [<ffffffffff80b8d9c2>] ret_from_fork_kernel_asm+0x16/0x18 [0.090560] ---[end trace 0000000000000000]--- [1.179875] CPU1: failed to come online [1.190324] smp: Brought up 1 node, 1 CPU
CVE-2025-38408	In the Linux kernel, the following vulnerability has been resolved: genirq/irq_sim: Initialize work context pointers properly Initialize `ops` member's pointers properly by using kzalloc() instead of kmalloc() when allocating the simulation work context. Otherwise the pointers contain random content leading to invalid dereferencing.
CVE-2025-38409	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Fix another leak in the submit error path put_unused_fd() doesn't free the installed file, if we've already done fd_install(). So we need to also free the sync_file. Patchwork: https://patchwork.freedesktop.org/patch/653583/
CVE-2025-38410	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Fix a fence leak in submit error path In error paths, we could unref the submit without calling drm_sched_entity_push_job(), so msm_job_free() will never get called. Since drm_sched_job_cleanup() will NULL out the s_fence, we can use that to detect this case. Patchwork: https://patchwork.freedesktop.org/patch/653584/
CVE-2025-38412	In the Linux kernel, the following vulnerability has been resolved: platform/x86: dell-wmi-sysman: Fix WMI data block retrieval in sysfs callbacks After retrieving WMI data blocks in sysfs callbacks, check for the validity of them before dereferencing their content.
CVE-2025-38414	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix GCC_GCC_PCIE_HOT_RST definition for WCN7850 GCC_GCC_PCIE_HOT_RST is wrongly defined for WCN7850, causing kernel crash on some specific platforms. Since this register is divergent for WCN7850 and QCN9274, move it to register table to allow different definitions. Then correct the register address for WCN7850 to fix this issue. Note IPQ5332 is not affected as it is not PCIe based device. Tested-on: WCN7850 hw2.0 PCI WLAN.HMT.1.0.c5-00481-QCAHMTSWPL_V1.0_V2.0_SILICONZ-3

CVE ID	Description
CVE-2025-38415	In the Linux kernel, the following vulnerability has been resolved: Squashfs: check return result of sb_min_blocksize Syzkaller reports an "UBSAN: shift-out-of-bounds in squashfs_bio_read" bug. Syzkaller forks multiple processes which after mounting the Squashfs filesystem, issues an ioctl("/dev/loop0", LOOP_SET_BLOCK_SIZE, 0x8000). Now if this ioctl occurs at the same time another process is in the process of mounting a Squashfs filesystem on /dev/loop0, the failure occurs. When this happens the following code in squashfs_fill_super() fails. ---- mslbk->devblksize = sb_min_blocksize(sb, SQUASHFS_DEVBLOCK_SIZE); mslbk->devblksize_log2 = ffz(~msblk->devblksize); ---- sb_min_blocksize() returns 0, which means mslbk->devblksize is set to 0. As a result, ffz(~msblk->devblksize) returns 64, and mslbk->devblksize_log2 is set to 64. This subsequently causes the UBSAN: shift-out-of-bounds in fs/squashfs/block.c:195:36 shift exponent 64 is too large for 64-bit type 'u64' (aka 'unsigned long') This commit adds a check for a 0 return by sb_min_blocksize().
CVE-2025-38416	In the Linux kernel, the following vulnerability has been resolved: NFC: nci: uart: Set tty->disc_data only in success path Setting tty->disc_data before opening the NCI device means we need to clean it up on error paths. This also opens some short window if device starts sending data, even before NCIUARTSETDRIVER IOCTL succeeded (broken hardware?). Close the window by exposing tty->disc_data only on the success path, when opening of the NCI device and try_module_get() succeeds. The code differs in error path in one aspect: tty->disc_data won't be ever assigned thus NULL-ified. This however should not be relevant difference, because of "tty->disc_data=NULL" in nci_uart_tty_open().
CVE-2025-38418	In the Linux kernel, the following vulnerability has been resolved: remoteproc: core: Release rproc->clean_table after rproc_attach() fails When rproc->state = RPROC_DETACHED is attached to remote processor through rproc_attach(), if rproc_handle_resources() returns failure, then the clean table should be released, otherwise the following memory leak will occur. unreferenced object 0xffff000086a99800 (size 1024): comm "kworker/u12:3", pid 59, jiffies 4294893670 (age 121.140s) hex dump (first 32 bytes): 00 00 00 00 00 80 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00 00 00 backtrace: [<000000008bbe4ca8>] slab_post_alloc_hook+0x98/0x3fc [<000000003b8a272b>] __kmem_cache_alloc_node+0x13c/0x230 [<000000007a507c51>] __kmalloc_node_track_caller+0x5c/0x260 [<0000000037818dae>] kmemdup+0x34/0x60 [<00000000610f7f57>] rproc_boot+0x35c/0x56c [<0000000065f8871a>] rproc_add+0x124/0x17c [<00000000497416ee>] imx_rproc_probe+0x4ec/0x5d4 [<000000003bcaa37d>] platform_probe+0x68/0xd8 [<00000000771577f9>] really_probe+0x110/0x27c [<00000000531fea59>] __driver_probe_device+0x78/0x12c [<0000000080036a04>] driver_probe_device+0x3c/0x118 [<000000007e0bddcb>] __device_attach_driver+0xb8/0xf8 [<000000000cf1fa33>] bus_for_each_drv+0x84/0xe4 [<000000001a53b53e>] __device_attach+0xfc/0x18c [<00000000d1a2a32c>] device_initial_probe+0x14/0x20 [<00000000d8f8b7ae>] bus_probe_device+0xb0/0xb4 unreferenced object 0xffff0000864c9690 (size 16):
CVE-2025-38419	In the Linux kernel, the following vulnerability has been resolved: remoteproc: core: Cleanup acquired resources when rproc_handle_resources() fails in rproc_attach() When rproc->state = RPROC_DETACHED and rproc_attach() is used to attach to the remote processor, if rproc_handle_resources() returns a failure, the resources allocated by imx_rproc_prepare() should be released, otherwise the following memory leak will occur. Since almost the same thing is done in imx_rproc_prepare() and rproc_resource_cleanup(), Function rproc_resource_cleanup() is able to deal with empty lists so it is better to fix the "goto" statements in rproc_attach(). replace the "unprepare_device" goto statement with "clean_up_resources" and get rid of the "unprepare_device" label. unreferenced object 0xffff0000861c5d00 (size 128): comm "kworker/u12:3", pid 59, jiffies 4294893509 (age 149.220s) hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 88 00 00 00 00 00 00 10 00 00 00 00 00 backtrace: [<00000000f949fe18>] slab_post_alloc_hook+0x98/0x37c [<000000000adbfb3e7>] __kmem_cache_alloc_node+0x138/0x2e0 [<00000000521c0345>] kmalloc_trace+0x40/0x158 [<0000000004e330a49>] rproc_mem_entry_init+0x60/0xf8 [<0000000002815755e>] imx_rproc_prepare+0xe0/0x180 [<0000000003f61b4e>] rproc_boot+0x2ec/0x528 [<00000000e7e994ac>] rproc_add+0x124/0x17c [<0000000048594076>] imx_rproc_probe+0x4ec/0x5d4 [<00000000efc298a1>] platform_probe+0x68/0xd8 [<00000000110be6fe>] really_probe+0x110/0x27c [<00000000e245c0ae>] __driver_probe_device+0x78/0x12c [<00000000f61f6f5e>] driver_probe_device+0x3c/0x118 [<000000000a7874938>] __device_attach_driver+0xb8/0xf8 [<00000000065319e69>] bus_for_each_drv+0x84/0xe4 [<000000000db3eb243>] __device_attach+0xfc/0x18c [<0000000072e4e1a4>] device_initial_probe+0x14/0x20
CVE-2025-38420	In the Linux kernel, the following vulnerability has been resolved: wifi: carl9170: do not ping device which has failed to load firmware Syzkaller reports [1, 2] crashes caused by an attempts to ping the device which has failed to load firmware. Since such a device doesn't pass 'ieee80211_register_hw()', an internal workqueue managed by 'ieee80211_queue_work()' is not yet created and an attempt to queue work on it causes null-ptr-deref. [1] https://syzkaller.appspot.com/bug?extid=9a4aec827829942045ff [2] https://syzkaller.appspot.com/bug?extid=0d8afb53e8fb2633217

CVE ID	Description
CVE-2025-38421	In the Linux kernel, the following vulnerability has been resolved: platform/x86/amd: pmf: Use device managed allocations If setting up smart PC fails for any reason then this can lead to a double free when unloading amd-pmf. This is because dev->buf was freed but never set to NULL and is again freed in amd_pmf_remove(). To avoid subtle allocation bugs in failures leading to a double free change all allocations into device managed allocations.
CVE-2025-38422	In the Linux kernel, the following vulnerability has been resolved: net: lan743x: Modify the EEPROM and OTP size for PCI1xxxx devices Maximum OTP and EEPROM size for hearthstone PCI1xxxx devices are 8 Kb and 64 Kb respectively. Adjust max size definitions and return correct EEPROM length based on device. Also prevent out-of-bound read/write.
CVE-2025-38424	In the Linux kernel, the following vulnerability has been resolved: perf: Fix sample vs do_exit() Baisheng Gao reported an ARM64 crash, which Mark decoded as being a synchronous external abort -- most likely due to trying to access MMIO in bad ways. The crash further shows perf trying to do a user stack sample while in exit_mmap()'s tlb_finish_mmu() -- i.e. while tearing down the address space it is trying to access. It turns out that we stop perf after we tear down the userspace mm; a receipt for disaster, since perf likes to access userspace for various reasons. Flip this order by moving up where we stop perf in do_exit(). Additionally, harden PERF_SAMPLE_CALLCHAIN and PERF_SAMPLE_STACK_USER to abort when the current task does not have an mm (exit_mm()) makes sure to set current->mm = NULL; before commencing with the actual teardown). Such that CPU wide events don't trip on this same problem.
CVE-2025-38425	In the Linux kernel, the following vulnerability has been resolved: i2c: tegra: check msg length in SMBUS block read For SMBUS block read, do not continue to read if the message length passed from the device is '0' or greater than the maximum allowed bytes.
CVE-2025-38426	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: Add basic validation for RAS header If RAS header read from EEPROM is corrupted, it could result in trying to allocate huge memory for reading the records. Add some validation to header fields.
CVE-2025-38427	In the Linux kernel, the following vulnerability has been resolved: video: screen_info: Relocate framebuffer behind PCI bridges Apply PCI host-bridge window offsets to screen_info framebuffers. Fixes invalid access to I/O memory. Resources behind a PCI host bridge can be relocated by a certain offset in the kernel's CPU address range used for I/O. The framebuffer memory range stored in screen_info refers to the CPU addresses as seen during boot (where the offset is 0). During boot up, firmware may assign a different memory offset to the PCI host bridge and thereby relocating the framebuffer address of the PCI graphics device as seen by the kernel. The information in screen_info must be updated as well. The helper pcibios_bus_to_resource() performs the relocation of the screen_info's framebuffer resource (given in PCI bus addresses). The result matches the I/O-memory resource of the PCI graphics device (given in CPU addresses). As before, we store away the information necessary to later update the information in screen_info itself. Commit 78aa89d1dfba ("firmware/sysfb: Update screen_info for relocated EFI framebuffers") added the code for updating screen_info. It is based on similar functionality that pre-existed in efifb. Efifb uses a pointer to the PCI resource, while the newer code does a memcpy of the region. Hence efifb sees any updates to the PCI resource and avoids the issue. v3: - Only use struct pci_bus_region for PCI bus addresses (Bjorn) - Clarify address semantics in commit messages and comments (Bjorn) v2: - Fixed tags (Takashi, Ivan) - Updated information on efifb
CVE-2025-38428	In the Linux kernel, the following vulnerability has been resolved: Input: ims-pcu - check record size in ims_pcu_flash_firmware() The "len" variable comes from the firmware and we generally do trust firmware, but it's always better to double check. If the "len" is too large it could result in memory corruption when we do "memcpy(fragment->data, rec->data, len);"
CVE-2025-38429	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: ep: Update read pointer only after buffer is written Inside mhi_ep_ring_add_element, the read pointer (rd_offset) is updated before the buffer is written, potentially causing race conditions where the host sees an updated read pointer before the buffer is actually written. Updating rd_offset prematurely can lead to the host accessing an uninitialized or incomplete element, resulting in data corruption. Invoke the buffer write before updating rd_offset to ensure the element is fully written before signaling its availability.
CVE-2025-38430	In the Linux kernel, the following vulnerability has been resolved: nfsd: nfsd4_spo_must_allow() must check this is a v4 compound request If the request being processed is not a v4 compound request, then examining the cstate can have undefined results. This patch adds a check that the rpc procedure being executed (rq_procinfo) is the NFSPROC4_COMPOUND procedure.

CVE ID	Description
CVE-2025-38436	In the Linux kernel, the following vulnerability has been resolved: drm/scheduler: signal scheduled fence when kill job When an entity from application B is killed, drm_sched_entity_kill() removes all jobs belonging to that entity through drm_sched_entity_kill_jobs_work(). If application A's job depends on a scheduled fence from application B's job, and that fence is not properly signaled during the killing process, application A's dependency cannot be cleared. This leads to application A hanging indefinitely while waiting for a dependency that will never be resolved. Fix this issue by ensuring that scheduled fences are properly signaled when an entity is killed, allowing dependent applications to continue execution.
CVE-2025-38437	In the Linux kernel, the following vulnerability has been resolved: ksmdb: fix potential use-after-free in oplock/lease break ack If ksmdb_iov_pin_rsp return error, use-after-free can happen by accessing opinfo->state and opinfo_put and ksmdb_fd_put could called twice.
CVE-2025-38438	In the Linux kernel, the following vulnerability has been resolved: ASoC: SOF: Intel: hda: Use devm_kstrdup() to avoid memleak. sof_pdata->tplg_filename can have address allocated by kstrdup() and can be overwritten. Memory leak was detected with kmemleak: unreferenced object 0xffff88812391ff60 (size 16): comm "kworker/4:1", pid 161, jiffies 4294802931 hex dump (first 16 bytes): 73 6f 66 2d 68 64 61 2d 67 65 6e 65 72 69 63 00 sof-hda-generic. backtrace (crc 4bf1675c): __kmalloct_node_track_caller_noprof+0x49c/0x6b0 kstrdup+0x46/0xc0 hda_machine_select.cold+0x1de/0x12cf [snd_sof_intel_hda_generic] sof_init_environment+0x16f/0xb50 [snd_sof] sof_probe_continue+0x45/0x7c0 [snd_sof] sof_probe_work+0x1e/0x40 [snd_sof] process_one_work+0x894/0x14b0 worker_thread+0x5e5/0xf0 kthread+0x39d/0x760 ret_from_fork+0x31/0x70 ret_from_fork_asm+0x1a/0x30
CVE-2025-38439	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Set DMA unmap len correctly for XDP_REDIRECT When transmitting an XDP_REDIRECT packet, call dma_unmap_len_set() with the proper length instead of 0. This bug triggers this warning on a system with IOMMU enabled: WARNING: CPU: 36 PID: 0 at drivers/iommu/dma-iommu.c:842 __iommu_dma_unmap+0x159/0x170 RIP: 0010: __iommu_dma_unmap+0x159/0x170 Code: a8 00 00 00 00 48 c7 45 b0 00 00 00 00 48 c7 45 c8 00 00 00 00 48 c7 45 a0 ff ff ff ff 4c 89 45 b8 4c 89 45 c0 e9 77 ff ff ff <0f> 0b e9 60 ff ff ff e8 8b bf 6a 00 66 66 2e 0f 1f 84 00 00 00 00 RSP: 0018:ff22d31181150c88 EFLAGS: 00010206 RAX: 0000000000002000 RBX: 00000000e13a0000 RCX: 0000000000000000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ff22d31181150cf0 R08: ff22d31181150ca8 R09: 0000000000000000 R10: 0000000000000000 R11: ff22d311d36c9d80 R12: 00000000000001000 R13: ff13544d10645010 R14: ff22d31181150c90 R15: ff13544d0b2bac00 FS: 0000000000000000(0000) GS:ff13550908a00000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 00000000080050033 CR2: 00005be909dacff8 CR3: 0008000173408003 CR4: 0000000000f71ef0 PKRU: 55555554 Call Trace: <IRQ> ? show_regs+0x6d/0x80 ? __warn+0x89/0x160 ? __iommu_dma_unmap+0x159/0x170 ? report_bug+0x17e/0x1b0 ? handle_bug+0x46/0x90 ? exc_invalid_op+0x18/0x80 ? asm_exc_invalid_op+0x1b/0x20 ? __iommu_dma_unmap+0x159/0x170 ? __iommu_dma_unmap+0xb3/0x170 iommu_dma_unmap_page+0x4f/0x100 dma_unmap_page_attr+0x52/0x220 ? srso_alias_return_thunk+0x5/0xfbf5 ? xdp_return_frame+0x2e/0xd0 bnxt_tx_int_xdp+0xdf/0x440 [bnxt_en] __bnxt_poll_work_done+0x81/0x1e0 [bnxt_en] bnxt_poll+0xd3/0x1e0 [bnxt_en]
CVE-2025-38440	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Fix race between DIM disable and net_dim() There's a race between disabling DIM and NAPI callbacks using the dim pointer on the RQ or SQ. If NAPI checks the DIM state bit and sees it still set, it assumes `rq->dim` or `sq->dim` is valid. But if DIM gets disabled right after that check, the pointer might already be set to NULL, leading to a NULL pointer dereference in net_dim(). Fix this by calling `synchronize_net()` before freeing the DIM context. This ensures all in-progress NAPI callbacks are finished before the pointer is cleared. Kernel log: BUG: kernel NULL pointer dereference, address: 0000000000000000 ... RIP: 0010:net_dim+0x23/0x190 ... Call Trace: <TASK> ? __die+0x20/0x60 ? page_fault_oops+0x150/0x3e0 ? common_interrupt+0xf/0xa0 ? sysvec_call_function_single+0xb/0x90 ? exc_page_fault+0x74/0x130 ? asm_exc_page_fault+0x22/0x30 ? net_dim+0x23/0x190 ? mlx5e_poll_ico_cq+0x41/0x6f0 [mlx5_core] ? sysvec_apic_timer_interrupt+0xb/0x90 mlx5e_handle_rx_dim+0x92/0xd0 [mlx5_core] mlx5e_napi_poll+0x2cd/0xac0 [mlx5_core] ? mlx5e_poll_ico_cq+0xe5/0x6f0 [mlx5_core] busy_poll_stop+0xa2/0x200 ? mlx5e_napi_poll+0x1d9/0xac0 [mlx5_core] ? mlx5e_trigger_irq+0x130/0x130 [mlx5_core] __napi_busy_loop+0x345/0x3b0 ? sysvec_call_function_single+0xb/0x90 ? asm_sysvec_call_function_single+0x16/0x20 ? sysvec_apic_timer_interrupt+0xb/0x90 ? pcpu_free_area+0x1e4/0x2e0 napi_busy_loop+0x11/0x20 xsk_recvmg+0x10c/0x130 sock_recvmg+0x44/0x70 __sys_recvfrom+0xbc/0x130 ? __schedule+0x398/0x890 __x64_sys_recvfrom+0x20/0x30 do_syscall_64+0x4c/0x100 entry_SYSCALL_64_after_hwframe+0x4b/0x53 ... ---[end trace 0000000000000000]--- ... ---[end Kernel panic - not syncing: Fatal exception in interrupt]---

CVE ID	Description
CVE-2025-38441	In the Linux kernel, the following vulnerability has been resolved: netfilter: flowtable: account for Ethernet header in nf_flow_pppoe_proto() syzbot found a potential access to uninit-value in nf_flow_pppoe_proto() Blamed commit forgot the Ethernet header. BUG: KMSAN: uninit-value in nf_flow_offload_inet_hook+0x7e4/0x940 net/netfilter/nf_flow_table_inet.c:27 nf_flow_offload_inet_hook+0x7e4/0x940 net/netfilter/nf_flow_table_inet.c:27 nf_hook_entry_hookfn include/linux/netfilter.h:157 [inline] nf_hook_slow+0xe1/0x3d0 net/netfilter/core.c:623 nf_hook_ingress include/linux/netfilter_netdev.h:34 [inline] nf_ingress net/core/dev.c:5742 [inline] __netif_receive_skb_core+0x4aff/0x70c0 net/core/dev.c:5837 __netif_receive_skb_one_core net/core/dev.c:5975 [inline] __netif_receive_skb+0xcc/0xac0 net/core/dev.c:6090 netif_receive_skb_internal net/core/dev.c:6176 [inline] netif_receive_skb+0x57/0x630 net/core/dev.c:6235 tun_rx_batched +0x1df/0x980 drivers/net/tun.c:1485 tun_get_user+0x4ee0/0x6b40 drivers/net/tun.c:1938 tun_chr_write_iter+0x3e9/0x5c0 drivers/net/tun.c:1984 new_sync_write fs/read_write.c:593 [inline] vfs_write+0xb4b/0x1580 fs/read_write.c:686 ksys_write fs/read_write.c:738 [inline] __do_sys_write fs/read_write.c:749 [inline]
CVE-2025-38443	In the Linux kernel, the following vulnerability has been resolved: nbd: fix uaf in nbd_genl_connect() error path There is a use-after-free issue in nbd: block nbd6: Receive control failed (result -104) block nbd6: shutting down sockets ===== BUG: KASAN: slab-use-after-free in recv_work+0x694/0xa80 drivers/block/nbd.c:1022 Write of size 4 at addr ffff8880295de478 by task kworker/u33:0/67 CPU: 2 UID: 0 PID: 67 Comm: kworker/u33:0 Not tainted 6.15.0-rc5-syzkaller-00123-g2c89c1b655c0 #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Workqueue: nbd6-recv recv_work Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl +0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xc3/0x670 mm/kasan/report.c:521 kasan_report+0xe0/0x110 mm/kasan/report.c:634 check_region_inline mm/kasan/generic.c:183 [inline] kasan_check_range+0xef/0x1a0 mm/kasan/generic.c:189 instrument_atomic_read_write include/linux/instrumented.h:96 [inline] atomic_dec include/linux/atomic/atomic-instrumented.h:592 [inline] recv_work+0x694/0xa80 drivers/block/nbd.c:1022 process_one_work+0x9cc/0x1b70 kernel/workqueue.c:3238 process_scheduled_works kernel/workqueue.c:3319 [inline] worker_thread+0x6c8/0xf10 kernel/workqueue.c:3400 kthread +0x3c2/0x780 kernel/kthread.c:464 ret_from_fork+0x45/0x80 arch/x86/kernel/process.c:153 ret_from_fork_asm+0x1a/0x30 arch/x86/entry/entry_64.S:245 </TASK> nbd_genl_connect() does not properly stop the device on certain error paths after nbd_start_device() has been called. This causes the error path to put nbd->config while recv_work continue to use the config after putting it, leading to use-after-free in recv_work. This patch moves nbd_start_device() after the backend file creation.
CVE-2025-38444	In the Linux kernel, the following vulnerability has been resolved: raid10: cleanup memleak at raid10_make_request If raid10_read_request or raid10_write_request registers a new request and the REQ_NOWAIT flag is set, the code does not free the malloc from the mempool. unreferenced object 0xffff8884802c3200 (size 192): comm "fio", pid 9197, jiffies 4298078271 hex dump (first 32 bytes): 00 00 00 00 00 00 00 00 88 41 02 00 00 00 00 00 A..... 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace (crc c1a049a2): __kmalloc+0x2bb/0x450 mempool_alloc+0x11b/0x320 raid10_make_request+0x19e/0x650 [raid10] md_handle_request+0x3b3/0x9e0 __submit_bio +0x394/0x560 __submit_bio_noacct+0x145/0x530 submit_bio_noacct_noccheck+0x682/0x830 __blkdev_direct_IO_async+0x4dc/0x6b0 blkdev_read_iter+0x1e5/0x3b0 __io_read+0x230/0x1110 io_read+0x13/0x30 io_issue_sqe+0x134/0x1180 io_submit_sqes+0x48c/0xe90 __do_sys_io_uring_enter +0x574/0x8b0 do_syscall_64+0x5c/0xe0 entry_SYSCALL_64_after_hwframe+0x76/0x7e V4: changing backing tree to see if CKI tests will pass. The patch code has not changed between any versions.
CVE-2025-38445	In the Linux kernel, the following vulnerability has been resolved: md/raid1: Fix stack memory use after return in raid1_reshape In the raid1_reshape function, newpool is allocated on the stack and assigned to conf->r1bio_pool. This results in conf->r1bio_pool.wait.head pointing to a stack address. Accessing this address later can lead to a kernel panic. Example access path: raid1_reshape() { // newpool is on the stack mempool_t newpool, oldpool; // initialize newpool.wait.head to stack address mempool_init(&newpool, ...); conf->r1bio_pool = newpool; } raid1_read_request() or raid1_write_request() { alloc_r1bio() { mempool_alloc() { // if pool->alloc fails remove_element() { -- pool->curr_nr; } } } mempool_free() { if (pool->curr_nr < pool->min_nr) { // pool->wait.head is a stack address // wake_up() will try to access this invalid address // which leads to a kernel panic return; wake_up(&pool->wait); } } Fix: reinit conf->r1bio_pool.wait after assigning newpool.
CVE-2025-38448	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: u_serial: Fix race condition in TTY wakeup A race condition occurs when gs_start_io() calls either gs_start_rx() or gs_start_tx(), as those functions briefly drop the port_lock for usb_ep_queue(). This allows gs_close() and gserial_disconnect() to clear port.tty and port_usb, respectively. Use the null-safe TTY Port helper function to wake up TTY. Example CPU1: CPU2: gserial_connect() // lock gs_close() // await lock gs_start_rx() // unlock usb_ep_queue() gs_close() // lock, reset port.tty and unlock gs_start_rx() // lock tty_wakeup() // NPE

CVE ID	Description
CVE-2025-38449	In the Linux kernel, the following vulnerability has been resolved: drm/gem: Acquire references on GEM handles for framebuffers A GEM handle can be released while the GEM buffer object is attached to a DRM framebuffer. This leads to the release of the dma-buf backing the buffer object, if any. [1] Trying to use the framebuffer in further mode-setting operations leads to a segmentation fault. Most easily happens with driver that use shadow planes for vmap-ing the dma-buf during a page flip. An example is shown below. [156.791968] -----[cut here]----- [156.796830] WARNING: CPU: 2 PID: 2255 at drivers/dma-buf/dma-buf.c:1527 dma_buf_vmap+0x224/0x430 [...] [156.942028] RIP: 0010:dma_buf_vmap+0x224/0x430 [157.043420] Call Trace: [157.045898] <TASK> [157.048030] ? show_trace_log_lvl+0x1af/0x2c0 [157.052436] ? show_trace_log_lvl+0x1af/0x2c0 [157.056836] ? show_trace_log_lvl+0x1af/0x2c0 [157.061253] ? drm_gem_shmem_vmap+0x74/0x710 [157.065567] ? dma_buf_vmap+0x224/0x430 [157.069446] ? __warn.cold+0x58/0xe4 [157.073061] ? dma_buf_vmap+0x224/0x430 [157.077111] ? report_bug+0x1dd/0x390 [157.080842] ? handle_bug+0x5e/0xa0 [157.084389] ? exc_invalid_op+0x14/0x50 [157.088291] ? asm_exc_invalid_op+0x16/0x20 [157.092548] ? dma_buf_vmap+0x224/0x430 [157.096663] ? dma_resv_get_singleton+0x6d/0x230 [157.101341] ? __pfx_dma_buf_vmap+0x10/0x10 [157.105588] ? __pfx_dma_resv_get_singleton+0x10/0x10 [157.110697] drm_gem_shmem_vmap+0x74/0x710 [157.114866] drm_gem_vmap+0xa9/0x1b0 [157.118763] drm_gem_vmap_unlocked+0x46/0xa0 [157.123086] drm_gem_fb_vmap+0xab/0x300 [157.126979] drm_atomic_helper_prepare_planes.part.0+0x487/0xb10 [157.133032] ? lockdep_init_map_type+0x19d/0x880 [157.137701] drm_atomic_helper_commit+0x13d/0x2e0 [157.142671] ? drm_atomic_nonblocking_commit+0xa0/0x180 [157.147988] drm_mode_atomic_ioctl+0x766/0xe40 [...] [157.346424] ---[end trace 0000000000000000]--- Acquiring GEM handles for the framebuffer's GEM buffer objects prevents this from happening. The framebuffer's cleanup later puts the handle references. Commit 1a148af06000 ("drm/gem-shmem: Use dma_buf from GEM object instance") triggers the segmentation fault easily by using the dma-buf field more widely. The underlying issue with reference counting has been present before. v2: - acquire the handle instead of the BO (Christian) - fix comment style (Christian) - drop the Fixes tag (Christian) - rename err_gotos - add missing Link tag

CVE ID	Description
CVE-2025-38455	In the Linux kernel, the following vulnerability has been resolved: KVM: SVM: Reject SEV{-ES} intra host migration if vCPU creation is in-flight Reject migration of SEV{-ES} state if either the source or destination VM is actively creating a vCPU, i.e. if <code>kvm_vm_ioctl_create_vcpu()</code> is in the section between incrementing <code>created_vcpus</code> and <code>online_vcpus</code>. The bulk of vCPU creation runs <code>_outside_of_kvm->lock</code> to allow creating multiple vCPUs in parallel, and so <code>sev_info.es_active</code> can get toggled from <code>false=>true</code> in the destination VM after (or during) <code>svm_vcpu_create()</code>, resulting in an SEV{-ES} VM effectively having a non-SEV{-ES} vCPU. The issue manifests most visibly as a crash when trying to free a vCPU's NULL VMSA page in an SEV-ES VM, but any number of things can go wrong. BUG: unable to handle page fault for address: <code>ffffebde00000000</code> #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] SMP KASAN NOPTI CPU: 227 UID: 0 PID: 64063 Comm: syz.5.60023 Tainted: G U O 6.15.0-smp-DEV #2 NONE Tainted: [U]=USER, [O]=OOT_MODULE Hardware name: Google, Inc. Arcadia_IT_80/Arcadia_IT_80, BIOS 12.52.0-0 10/28/2024 RIP: 0010:constant_test_bit arch/x86/include/asm/bitops.h:206 [inline] RIP: 0010:arch_test_bit arch/x86/include/asm/bitops.h:238 [inline] RIP: 0010:_test_bit include/asm-generic/bitops/instrumented-non-atomic.h:142 [inline] RIP: 0010:PageHead include/linux/page-flags.h:866 [inline] RIP: 0010:___free_pages+0x3e/0x120 mm/page_alloc.c:5067 Code: <49> f7 06 40 00 00 00 75 05 45 31 ff eb 0c 66 90 4c 89 f0 4c 39 f0 RSP: 0018:ffff8984551978d0 EFLAGS: 00010246 RAX: 0000777f80000001 RBX: 0000000000000000 RCX: ffffffff91aeb98 RDX: 0000000000000000 RSI: 0000000000000008 RDI: fffffebde0000000 RBP: 0000000000000000 R08: fffffebde0000000 R09: 1ffffd7bc0000000 R10: dffffc0000000000 R11: fffff97bc0000001 R12: dffffc0000000000 R13: fffff8983e19751a8 R14: fffffebde00000000 R15: 1ffffd7bc0000000 FS: 0000000000000000(0000) GS:ffff89ee661d3000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: fffffebde0000000 CR3: 000000793ceaa000 CR4: 0000000000350ef0 DR0: 0000000000000000 DR1: 0000000000000b5f DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400 Call Trace: <TASK> sev_free_vcpu+0x413/0x630 arch/x86/kvm/svm/sev.c:3169 svm_vcpu_free+0x13a/0x2a0 arch/x86/kvm/svm/svm.c:1515 kvm_arch_vcpu_destroy+0x6a/0x1d0 arch/x86/kvm/x86.c:12396 kvm_vcpu_destroy virt/kvm/kvm_main.c:470 [inline] kvm_destroy_vcpus+0xd1/0x300 virt/kvm/kvm_main.c:490 kvm_arch_destroy_vm+0x636/0x820 arch/x86/kvm/x86.c:12895 kvm_put_kvm+0xb8e/0xfb0 virt/kvm/kvm_main.c:1310 kvm_vm_release+0x48/0x60 virt/kvm/kvm_main.c:1369 __fput+0x3e4/0x9e0 fs/file_table.c:465 task_work_run+0x1a9/0x220 kernel/task_work.c:227 exit_task_work include/linux/task_work.h:40 [inline] do_exit+0x7f0/0x25b0 kernel/exit.c:953 do_group_exit+0x203/0x2d0 kernel/exit.c:1102 get_signal+0x1357/0x1480 kernel/signal.c:3034 arch_do_signal_or_restart+0x40/0x690 arch/x86/kernel/signal.c:337 exit_to_user_mode_loop kernel/entry/common.c:111 [inline] exit_to_user_mode_prepare include/linux/entry-common.h:329 [inline] __syscall_exit_to_user_mode_work kernel/entry/common.c:207 [inline] syscall_exit_to_user_mode+0x67/0xb0 kernel/entry/common.c:218 do_syscall_64+0x7c/0x150 arch/x86/entry/syscall_64.c:100 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f87a898e969 </TASK> Modules linked in: gq(O) gsmi: Log Shutdown Reason 0x03 CR2: fffffebde0000000 ---[end trace 0000000000000000]--- Deliberately don't check for a NULL VMSA when freeing the vCPU, as crashing the host is likely desirable due to the VMSA being consumed by hardware. E.g. if KVM manages to allow VMRUN on the vCPU, hardware may read/write a bogus VMSA page. Accessing P --- truncated---
CVE-2025-38456	In the Linux kernel, the following vulnerability has been resolved: ipmi:msghandler: Fix potential memory corruption in <code>ipmi_create_user()</code> The "intf" list iterator is an invalid pointer if the correct "intf->intf_num" is not found. Calling <code>atomic_dec(&intf->nr_users)</code> on an invalid pointer will lead to memory corruption. We don't really need to call <code>atomic_dec()</code> if we haven't called <code>atomic_add_return()</code> so update the <code>if (intf->in_shutdown)</code> path as well.
CVE-2025-38457	In the Linux kernel, the following vulnerability has been resolved: net/sched: Abort <code>__tc_modify_qdisc</code> if parent class does not exist Lion's patch [1] revealed an ancient bug in the qdisc API. Whenever a user creates/modifies a qdisc specifying as a parent another qdisc, the qdisc API will, during grafting, detect that the user is not trying to attach to a class and reject. However grafting is performed after <code>qdisc_create</code> (and thus the qdiscs' init callback) is executed. In qdiscs that eventually call <code>qdisc_tree_reduce_backlog</code> during init or change (such as fq, hfh, choke, etc), an issue arises. For example, executing the following commands: <code>sudo tc qdisc add dev lo root handle a: htb default 2</code> <code>sudo tc qdisc add dev lo parent a: handle beef fq</code> Qdiscs such as fq, hfh, choke, etc unconditionally invoke <code>qdisc_tree_reduce_backlog()</code> in their control path <code>init()</code> or <code>change()</code> which then causes a failure to find the child class; however, that does not stop the unconditional invocation of the assumed child qdisc's <code>qlen_notify</code> with a null class. All these qdiscs make the assumption that class is non-null. The solution is ensure that <code>qdisc_leaf()</code> which looks up the parent class, and is invoked prior to <code>qdisc_create()</code>, should return failure on not finding the class. In this patch, we leverage <code>qdisc_leaf</code> to return <code>ERR_PTRs</code> whenever the parentid doesn't correspond to a class, so that we can detect it earlier on and abort before <code>qdisc_create</code> is called. [1] https://lore.kernel.org/netdev/d912cbd7-193b-4269-9857-525bee8bbb6a@gmail.com/

CVE ID	Description
CVE-2025-38458	In the Linux kernel, the following vulnerability has been resolved: atm: clip: Fix NULL pointer dereference in vcc_sendmsg() atmarpd_dev_ops does not implement the send method, which may cause crash as bellow. BUG: kernel NULL pointer dereference, address: 0000000000000000 PGD 0 P4D 0 Oops: Oops: 0010 [#1] SMP KASAN NOPTI CPU: 0 UID: 0 PID: 5324 Comm: syz.0.0 Not tainted 6.15.0-rc6-syzkaller-00346-g5723cc3450bc #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:0x0 Code: Unable to access opcode bytes at 0xffffffffffffd6. RSP: 0018:ffffc9000d3cf778 EFLAGS: 00010246 RAX: 1fffffff1910dd1 RBX: 00000000000000c0 RCX: dffffc0000000000 RDX: fffffc9000dc82000 RSI: ffff88803e4c4640 RDI: ffff888052cd0000 RBP: fffffc9000d3cf8d0 R08: ffff888052c9143f R09: 1ffff1100a592287 R10: dffffc0000000000 R11: 0000000000000000 R12: 1ffff92001a79f00 R13: ffff888052cd0000 R14: ffff88803e4c4640 R15: ffffff8c886e88 FS: 00007fbc762566c0(0000) GS:ffff88808d6c2000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffff8d6 CR3: 0000000041f1b000 CR4: 0000000000352ef0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> vcc_sendmsg+0xa10/0xc50 net/atm/common.c:644 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg+0x219/0x270 net/socket.c:727 __sys_sendmsg+0x52d/0x830 net/socket.c:2566 __sys_sendmsg+0x21f/0x2a0 net/socket.c:2620 __sys_sendmsg+0x227/0x430 net/socket.c:2709 __do_sys_sendmsg net/socket.c:2736 [inline] __se_sys_sendmsg net/socket.c:2733 [inline] __x64_sys_sendmsg+0xa0/0xc0 net/socket.c:2733 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xf6/0x210 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f
CVE-2025-38459	In the Linux kernel, the following vulnerability has been resolved: atm: clip: Fix infinite recursive call of clip_push(). syzbot reported the splat below. [0] This happens if we call ioctl(ATMARP_MKIP) more than once. During the first call, clip_mkkip() sets clip_push() to vcc->push(), and the second call copies it to clip_vcc->old_push(). Later, when the socket is close(), vcc_destroy_socket() passes NULL skb to clip_push(), which calls clip_vcc->old_push(), triggering the infinite recursion. Let's prevent the second ioctl(ATMARP_MKIP) by checking vcc->user_back, which is allocated by the first call as clip_vcc. Note also that we use lock_sock() to prevent racy calls. [0]: BUG: TASK stack guard page was hit at fffffc9000d66fff8 (stack is fffffc9000d670000..ffffc9000d678000) Oops: stack guard page: 0000 [#1] SMP KASAN NOPTI CPU: 0 UID: 0 PID: 5322 Comm: syz.0.0 Not tainted 6.16.0-rc4-syzkaller #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 RIP: 0010:clip_push+0x5/0x720 net/atm/clip.c:191 Code: e0 8f aa 8c e8 1c ad 5b fa eb ae 66 2e 0f 1f 84 00 00 00 00 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa 55 <41> 57 41 56 41 55 41 54 53 48 83 ec 20 48 89 f3 49 89 fd 48 bd 00 RSP: 0018:ffffc9000d670000 EFLAGS: 00010246 RAX: 1ffff1100235a4a5 RBX: ffff888011ad2508 RCX: ffff8880003c0000 RDX: 0000000000000000 RSI: 0000000000000000 RDI: ffff888037f01000 RBP: dffffc0000000000 R08: ffffffff8fa104f7 R09: 1fffff1f4209e R10: dffffc0000000000 R11: ffffffff8a99b300 R12: ffff888037f01000 R14: ffff888011ad2500 R15: ffff888037f01578 FS: 000055557ab6d500(0000) GS:ffff88808d250000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: fffffc9000d66fff8 CR3: 0000000043172000 CR4: 0000000000352ef0 Call Trace: <TASK> clip_push+0x6dc/0x720 net/atm/clip.c:200 clip_push+0x6dc/0x720 net/atm/clip.c:200 clip_push+0x6dc/0x720 net/atm/clip.c:200 ... clip_push+0x6dc/0x720 net/atm/clip.c:200 clip_push+0x6dc/0x720 net/atm/clip.c:200 clip_push+0x6dc/0x720 net/atm/clip.c:200 clip_push+0x6dc/0x720 net/atm/clip.c:200 vcc_destroy_socket net/atm/common.c:183 [inline] vcc_release+0x157/0x460 net/atm/common.c:205 __sock_release net/socket.c:647 [inline] sock_close+0xc0/0x240 net/socket.c:1391 __fput+0x449/0xa70 fs/file_table.c:465 task_work_run+0x1d1/0x260 kernel/task_work.c:227 resume_user_mode_work include/linux/resume_user_mode.h:50 [inline] exit_to_user_mode_loop+0xec/0x110 kernel/entry/common.c:114 exit_to_user_mode_prepare include/linux/entry-common.h:330 [inline] syscall_exit_to_user_mode_work include/linux/entry-common.h:414 [inline] syscall_exit_to_user_mode include/linux/entry-common.h:449 [inline] do_syscall_64+0x2bd/0x3b0 arch/x86/entry/syscall_64.c:100 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7ff31c98e929 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca d4 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007fffb5aa1f78 EFLAGS: 00000246 ORIG_RAX: 00000000000001b4 RAX: 0000000000000000 RBX: 0000000000012747 RCX: 00007ff31c98e929 RDX: 0000000000000000 RSI: 000000000000001e RDI: 0000000000000003 RBP: 00007ff31cbb7ba0 R08: 0000000000000001 R09: 00000000db5aa226f R10: 00007ff31c7ff030 R11: 0000000000000246 R12: 00007ff31cbb608c R13: 00007ff31cbb6080 R14: ffffffff8d6 R15: 00007fffb5aa2090 </TASK> Modules linked in:
CVE-2025-38460	In the Linux kernel, the following vulnerability has been resolved: atm: clip: Fix potential null-ptr-deref in to_atmarpd(). atmarpd is protected by RTNL since commit f3a0592b37b8 ("[ATM]: clip causes unregister hang"). However, it is not enough because to_atmarpd() is called without RTNL, especially clip_neigh_solicit() / neigh_ops->solicit() is unsleepable. Also, there is no RTNL dependency around atmarpd. Let's use a private mutex and RCU to protect access to atmarpd in to_atmarpd().

CVE ID	Description
CVE-2025-38461	In the Linux kernel, the following vulnerability has been resolved: vsock: Fix transport_* TOCTOU Transport assignment may race with module unload. Protect new_transport from becoming a stale pointer. This also takes care of an insecure call in vsock_use_local_transport(); add a lockdep assert. BUG: unable to handle page fault for address: fffffbfff8056000 Oops: Oops: 0000 [#1] SMP KASAN RIP: 0010:vsock_assign_transport+0x366/0x600 Call Trace: vsock_connect+0x59c/0xc40 __sys_connect+0xe8/0x100 __x64_sys_connect+0x6e/0xc0 do_syscall_64+0x92/0x1c0 entry_SYSCALL_64_after_hwframe+0x4b/0x53
CVE-2025-38462	In the Linux kernel, the following vulnerability has been resolved: vsock: Fix transport_{g2h,h2g} TOCTOU vsock_find_cid() and vsock_dev_do_ioctl() may race with module unload. transport_{g2h,h2g} may become NULL after the NULL check. Introduce vsock_transport_local_cid() to protect from a potential null-ptr-deref. KASAN: null-ptr-deref in range [0x0000000000000118-0x000000000000011f] RIP: 0010:vsock_find_cid+0x47/0x90 Call Trace: __vsock_bind+0x4b2/0x720 vsock_bind+0x90/0xe0 __sys_bind+0x14d/0x1e0 __x64_sys_bind+0x6e/0xc0 do_syscall_64+0x92/0x1c0 entry_SYSCALL_64_after_hwframe+0x4b/0x53 KASAN: null-ptr-deref in range [0x0000000000000118-0x000000000000011f] RIP: 0010:vsock_dev_do_ioctl.isra.0+0x58/0xf0 Call Trace: __x64_sys_ioctl+0x12d/0x190 do_syscall_64+0x92/0x1c0 entry_SYSCALL_64_after_hwframe+0x4b/0x53
CVE-2025-38463	In the Linux kernel, the following vulnerability has been resolved: tcp: Correct signedness in skb remaining space calculation Syzkaller reported a bug [1] where sk->sk_forward_alloc can overflow. When we send data, if an skb exists at the tail of the write queue, the kernel will attempt to append the new data to that skb. However, the code that checks for available space in the skb is flawed: "" copy = size_goal - skb->len "" The types of the variables involved are: "" copy: ssize_t (s64 on 64-bit systems) size_goal: int skb->len: unsigned int "" Due to C's type promotion rules, the signed size_goal is converted to an unsigned int to match skb->len before the subtraction. The result is an unsigned int. When this unsigned int result is then assigned to the s64 copy variable, it is zero-extended, preserving its non-negative value. Consequently, copy is always >= 0. Assume we are sending 2GB of data and size_goal has been adjusted to a value smaller than skb->len. The subtraction will result in copy holding a very large positive integer. In the subsequent logic, this large value is used to update sk->sk_forward_alloc, which can easily cause it to overflow. The syzkaller reproducer uses TCP_REPAIR to reliably create this condition. However, this can also occur in real-world scenarios. The tcp_bound_to_half_wnd() function can also reduce size_goal to a small value. This would cause the subsequent tcp_wmem_schedule() to set sk->sk_forward_alloc to a value close to INT_MAX. Further memory allocation requests would then cause sk_forward_alloc to wrap around and become negative. [1]: https://syzkaller.appspot.com/bug?extid=de6565462ab540f50e47

CVE ID	Description
CVE-2025-38464	In the Linux kernel, the following vulnerability has been resolved: tipc: Fix use-after-free in tipc_conn_close(). syzbot reported a null-ptr-deref in tipc_conn_close() during netns dismantle. [0] tipc_topsrv_stop() iterates tipc_net(net)->topsrv->conn_idr and calls tipc_conn_close() for each tipc_conn. The problem is that tipc_conn_close() is called after releasing the IDR lock. At the same time, there might be tipc_conn_rcv_work() running and it could call tipc_conn_close() for the same tipc_conn and release its last ->kref. Once we release the IDR lock in tipc_topsrv_stop(), there is no guarantee that the tipc_conn is alive. Let's hold the ref before releasing the lock and put the ref after tipc_conn_close() in tipc_topsrv_stop(). [0]: BUG: KASAN: use-after-free in tipc_conn_close+0x122/0x140 net/tipc/topsrv.c:165 Read of size 8 at addr ffff888099305a08 by task kworker/u4:3/435 CPU: 0 PID: 435 Comm: kworker/u4:3 Not tainted 4.19.204-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011 Workqueue: netns cleanup_net Call Trace: __dump_stack lib/dump_stack.c:77 [inline] dump_stack+0x1fc/0x2ef lib/dump_stack.c:118 print_address_description.cold+0x54/0x219 mm/kasan/report.c:256 kasan_report_error.cold+0x8a/0x1b9 mm/kasan/report.c:354 kasan_report mm/kasan/report.c:412 [inline] __asan_report_load8_noabort+0x88/0x90 mm/kasan/report.c:433 tipc_conn_close+0x122/0x140 net/tipc/topsrv.c:165 tipc_topsrv_stop net/tipc/topsrv.c:701 [inline] tipc_topsrv_exit_net+0x27b/0x5c0 net/tipc/topsrv.c:722 ops_exit_list+0xa5/0x150 net/core/net_namespace.c:153 cleanup_net+0x3b4/0x8b0 net/core/net_namespace.c:553 process_one_work+0x864/0x1570 kernel/workqueue.c:2153 worker_thread+0x64c/0x1130 kernel/workqueue.c:2296 kthread+0x33f/0x460 kernel/kthread.c:259 ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:415 Allocated by task 23: kmem_cache_alloc_trace+0x12f/0x380 mm/slab.c:3625 kmalloc include/linux/slab.h:515 [inline] kzalloc include/linux/slab.h:709 [inline] tipc_conn_alloc+0x43/0x4f0 net/tipc/topsrv.c:192 tipc_topsrv_accept+0x1b5/0x280 net/tipc/topsrv.c:470 process_one_work+0x864/0x1570 kernel/workqueue.c:2153 worker_thread+0x64c/0x1130 kernel/workqueue.c:2296 kthread+0x33f/0x460 kernel/kthread.c:259 ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:415 Freed by task 23: __cache_free mm/slab.c:3503 [inline] kfree+0xcc/0x210 mm/slab.c:3822 tipc_conn_kref_release net/tipc/topsrv.c:150 [inline] kref_put include/linux/kref.h:70 [inline] conn_put+0x2cd/0x3a0 net/tipc/topsrv.c:155 process_one_work+0x864/0x1570 kernel/workqueue.c:2153 worker_thread+0x64c/0x1130 kernel/workqueue.c:2296 kthread+0x33f/0x460 kernel/kthread.c:259 ret_from_fork+0x24/0x30 arch/x86/entry/entry_64.S:415 The buggy address belongs to the object at ffff888099305a00 which belongs to the cache kmalloc-512 of size 512 The buggy address is located 8 bytes inside of 512-byte region [ffff888099305a00, ffff888099305c00) The buggy address belongs to the page: page:ffffea000264c140 count:1 mapcount:0 mapping:ffff88813bfff0940 index:0x0 flags: 0xffff00000000100(slab) raw: 00ffff00000000100 ffffea00028b6b88 ffffea0002cd2b08 ffff88813bfff0940 raw: 000000000000000000 ffff888099305000 000000001000000006 0000000000000000 page dumped because: kasan: bad access detected Memory state around the buggy address: ffff888099305900: fb fb fb fb fb fb fb fb fb fb fb fb ffff888099305980: fc fc fc fc fc fc fc fc fc fc fc fc >ffff888099305a00: fb fb fb fb fb fb fb fb fb fb fb fb ffff888099305b00: fb fb fb fb fb fb fb fb fb fb fb fb
CVE-2025-38465	In the Linux kernel, the following vulnerability has been resolved: netlink: Fix wraparounds of sk->sk_rmem_alloc. Netlink has this pattern in some places if (atomic_read(&sk->sk_rmem_alloc) > sk->sk_rcvbuf) atomic_add(skb->truesize, &sk->sk_rmem_alloc); , which has the same problem fixed by commit 5a465a0da13e ("udp: Fix multiple wraparounds of sk->sk_rmem_alloc."). For example, if we set INT_MAX to SO_RCVBUFFORCE, the condition is always false as the two operands are of int. Then, a single socket can eat as many skb as possible until OOM happens, and we can see multiple wraparounds of sk->sk_rmem_alloc. Let's fix it by using atomic_add_return() and comparing the two variables as unsigned int. Before: [root@fedora ~]# ss -f netlink Recv-Q Send-Q Local Address:Port Peer Address:Port -1668710080 0 rtnl:nl_wraparound/293 * After: [root@fedora ~]# ss -f netlink Recv-Q Send-Q Local Address:Port Peer Address:Port 2147483072 0 rtnl:nl_wraparound/290 * ^ `--- INT_MAX - 576
CVE-2025-38466	In the Linux kernel, the following vulnerability has been resolved: perf: Revert to requiring CAP_SYS_ADMIN for uprobes Jann reports that uprobes can be used destructively when used in the middle of an instruction. The kernel only verifies there is a valid instruction at the requested offset, but due to variable instruction length cannot determine if this is an instruction as seen by the intended execution stream. Additionally, Mark Rutland notes that on architectures that mix data in the text segment (like arm64), a similar things can be done if the data word is 'mistaken' for an instruction. As such, require CAP_SYS_ADMIN for uprobes.
CVE-2025-38467	In the Linux kernel, the following vulnerability has been resolved: drm/exynos: exynos7_drm_decon: add vblank check in IRQ handling If there's support for another console device (such as a TTY serial), the kernel occasionally panics during boot. The panic message and a relevant snippet of the call stack is as follows: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000000 Call trace: drm_crtc_handle_vblank+0x10/0x30 (P) decon_irq_handler+0x88/0xb4 [...] Otherwise, the panics don't happen. This indicates that it's some sort of race condition. Add a check to validate if the drm device can handle vblanks before calling drm_crtc_handle_vblank() to avoid this.

CVE ID	Description
CVE-2025-38468	In the Linux kernel, the following vulnerability has been resolved: net/sched: Return NULL when htb_lookup_leaf encounters an empty rbtree htb_lookup_leaf has a BUG_ON that can trigger with the following: tc qdisc del dev lo root tc qdisc add dev lo root handle 1: htb default 1 tc class add dev lo parent 1: classid 1:1 htb rate 64bit tc qdisc add dev lo parent 1:1 handle 2: netem tc qdisc add dev lo parent 2:1 handle 3: blackhole ping -I lo -c1 -W0.001 127.0.0.1 The root cause is the following: 1. htb_dequeue calls htb_dequeue_tree which calls the dequeue handler on the selected leaf qdisc 2. netem_dequeue calls enqueue on the child qdisc 3. blackhole_enqueue drops the packet and returns a value that is not just NET_XMIT_SUCCESS 4. Because of this, netem_dequeue calls qdisc_tree_reduce_backlog, and since qlen is now 0, it calls htb_qlen_notify -> htb_deactivate -> htb_deactivate_prios -> htb_remove_class_from_row -> htb_safe_rb_erase 5. As this is the only class in the selected hprio rbtree, __rb_change_child in __rb_erase_augmented sets the rb_root pointer to NULL 6. Because blackhole_dequeue returns NULL, netem_dequeue returns NULL, which causes htb_dequeue_tree to call htb_lookup_leaf with the same hprio rbtree, and fail the BUG_ON The function graph for this scenario is shown here: 0 htb_enqueue() { 0 } + 13.635 us netem_enqueue(); 0 4.719 us htb_activate_prios(); 0 # 2249.199 us } 0 htb_dequeue() { 0 } 2.355 us htb_lookup_leaf(); 0 netem_dequeue() { 0 } + 11.061 us blackhole_enqueue(); 0 qdisc_tree_reduce_backlog() { 0 } qdisc_lookup_rcu() { 0 } 1.873 us qdisc_match_from_root(); 0 6.292 us } 0 1.894 us htb_search(); 0 htb_qlen_notify() { 0 } 2.655 us htb_deactivate_prios(); 0 6.933 us } 0 + 25.227 us } 0 1.983 us blackhole_dequeue(); 0 + 86.553 us } 0 # 2932.761 us qdisc_warn_nonwc(); 0 htb_lookup_leaf() { 0 } BUG_ON(); ----- The full original bug report can be seen here [1]. We can fix this just by returning NULL instead of the BUG_ON, as htb_dequeue_tree returns NULL when htb_lookup_leaf returns NULL. [1] https://lore.kernel.org/netdev/pF5XOOIim0IuEfhI-SOxTgRvNoDwuux7UHKnE_Y5-zVd4wmGvNk2ceHjKb8ORnzW0cGwfmVu42g9dL7XyJLf1NEzaztboTWcm0Ogxuojoeo=@willsroot.io/
CVE-2025-38469	In the Linux kernel, the following vulnerability has been resolved: KVM: x86/xen: Fix cleanup logic in emulation of Xen schedop poll hypercalls kvm_xen_schedop_poll does a kmallocc_array() when a VM polls the host for more than one event channel potr (nr_ports > 1). After the kmallocc_array(), the error paths need to go through the "out" label, but the call to kvm_read_guest_virt() does not. [Adjusted commit message. - Paolo]

CVE ID	Description
CVE-2025-38470	In the Linux kernel, the following vulnerability has been resolved: net: vlan: fix VLAN 0 refcount imbalance of toggling filtering during runtime Assuming the "rx-vlan-filter" feature is enabled on a net device, the 8021q module will automatically add or remove VLAN 0 when the net device is put administratively up or down, respectively. There are a couple of problems with the above scheme. The first problem is a memory leak that can happen if the "rx-vlan-filter" feature is disabled while the device is running: # ip link add bond1 up type bond mode 0 # ethtool -K bond1 rx-vlan-filter off # ip link del dev bond1 When the device is put administratively down the "rx-vlan-filter" feature is disabled, so the 8021q module will not remove VLAN 0 and the memory will be leaked [1]. Another problem that can happen is that the kernel can automatically delete VLAN 0 when the device is put administratively down despite not adding it when the device was put administratively up since during that time the "rx-vlan-filter" feature was disabled. null-ptr-unref or bug_on[2] will be triggered by unregister_vlan_dev() for refcount imbalance if toggling filtering during runtime: \$ ip link add bond0 type bond mode 0 \$ ip link add link bond0 name vlan0 type vlan id 0 protocol 802.1q \$ ethtool -K bond0 rx-vlan-filter off \$ ifconfig bond0 up \$ ethtool -K bond0 rx-vlan-filter on \$ ifconfig bond0 down \$ ip link del vlan0 Root cause is as below: step1: add vlan0 for real_dev, such as bond, team. register_vlan_dev vlan_vid_add(real_dev,htons(ETH_P_8021Q),0) //refcnt=1 step2: disable vlan filter feature and enable real_dev step3: change filter from 0 to 1 vlan_device_event vlan_filter_push_vids ndo_vlan_rx_add_vid //No refcnt added to real_dev vlan0 step4: real_dev down vlan_device_event vlan_vid_del(dev, htons(ETH_P_8021Q), 0); //refcnt=0 vlan_info_rcu_free // free vlan0 step5: delete vlan0 unregister_vlan_dev BUG_ON(!vlan_info); //vlan_info is null Fix both problems by noting in the VLAN info whether VLAN 0 was automatically added upon NETDEV_UP and based on that decide whether it should be deleted upon NETDEV_DOWN, regardless of the state of the "rx-vlan-filter" feature. [1] unreferenced object 0xffff8880068e3100 (size 256): comm "ip", pid 384, jiffies 4296130254 hex dump (first 32 bytes): 00 20 30 0d 80 88 ff ff 00 00 00 00 00 00 00 00 . 0..... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 backtrace (crc 81ce31fa): __kmalloc_cache_noprof+0x2b5/0x340 vlan_vid_add+0x434/0x940 vlan_device_event.cold+0x75/0xa8 notifier_call_chain+0xca/0x150 __dev_notify_flags+0xe3/0x250 rtnl_configure_link+0x193/0x260 rtnl_newlink_create+0x383/0x8e0 __rtnl_newlink+0x22c/0xa40 rtnl_newlink+0x627/0xb00 rtnetlink_rcv_msg+0x6fb/0xb70 netlink_rcv_skb+0x11f/0x350 netlink_unicast+0x426/0x710 netlink_sendmsg+0x75a/0xc20 __sock_sendmsg+0xc1/0x150 __sys_sendmsg+0x5aa/0x7b0 __sys_sendmsg+0xfc/0x180 [2] kernel BUG at net/8021q/vlan.c:99! Oops: invalid opcode: 0000 [#1] SMP KASAN PTI CPU: 0 UID: 0 PID: 382 Comm: ip Not tainted 6.16.0-rc3 #61 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010:unregister_vlan_dev (net/8021q/vlan.c:99 (discriminator 1)) RSP: 0018:ffff88810badf310 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff88810da84000 RCX: ffffffff47ceb9a RDX: dffffc0000000000 RSI: 0000000000000008 RDI: ffff88810e8b43c8 RBP: 0000000000000000 R08: 0000000000000000 R09: ffffffffb6cfe80 R10: ffffffffb677f407 R11: ffff88810badf3c0 R12: ffff88810e8b4000 R13: 0000000000000000 R14: ffff88810642a5c0 R15: 0000000000000017e FS: 00007f1ff68c20c0(0000) GS:ffff888163a24000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f1ff5dad240 CR3: 0000000107e56000 CR4: 00000000000006f0 Call Trace: <TASK ---truncated---
CVE-2025-38471	In the Linux kernel, the following vulnerability has been resolved: tls: always refresh the queue when reading sock After recent changes in net-next TCP compacts skbs much more aggressively. This unearthed a bug in TLS where we may try to operate on an old skb when checking if all skbs in the queue have matching decrypt state and geometry. BUG: KASAN: slab-use-after-free in tls_strp_check_rcv+0x898/0x9a0 [tls] (net/tls/tls_strp.c:436 net/tls/tls_strp.c:530 net/tls/tls_strp.c:544) Read of size 4 at addr ffff888013085750 by task tls/13529 CPU: 2 UID: 0 PID: 13529 Comm: tls Not tainted 6.16.0-rc5-virtme Call Trace: kasan_report+0xca/0x100 tls_strp_check_rcv+0x898/0x9a0 [tls] tls_rx_rec_wait+0x2c9/0x8d0 [tls] tls_sw_recvmmsg+0x40f/0x1aa0 [tls] inet_recvmmsg+0x1c3/0x1f0 Always reload the queue, fast path is to have the record in the queue when we wake, anyway (IOW the path going down "if !strp->stm.full_len").

CVE ID	Description
CVE-2025-38472	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_contrack: fix crash due to removal of uninitialised entry A crash in conntrack was reported while trying to unlink the conntrack entry from the hash bucket list: [exception RIP: __nf_ct_delete_from_lists+172] [...] #7 [ff539b5a2b043aa0] nf_ct_delete at ffffffff124d421 [nf_contrack] #8 [ff539b5a2b043ad0] nf_ct_gc_expired at ffffffff124d999 [nf_contrack] #9 [ff539b5a2b043ae0] __nf_contrack_find_get at ffffffff124efbc [nf_contrack] [...] The nf_conn struct is marked as allocated from slab but appears to be in a partially initialised state: ct hlist pointer is garbage; looks like the ct hash value (hence crash). ct->status is equal to IPS_CONFIRMED IPS_DYING, which is expected ct->timeout is 30000 (=30s), which is unexpected. Everything else looks like normal udp conntrack entry. If we ignore ct->status and pretend its 0, the entry matches those that are newly allocated but not yet inserted into the hash: - ct hlist pointers are overloaded and store/cache the raw tuple hash - ct->timeout matches the relative time expected for a new udp flow rather than the absolute 'jiffies' value. If it were not for the presence of IPS_CONFIRMED, __nf_contrack_find_get() would have skipped the entry. Theory is that we did hit following race: cpu x cpu y cpu z found entry E found entry E E is expired <preemption> nf_ct_delete() return E to rcu slab init_conntrack E is re-init'd, ct->status set to 0 reply tuplehash hnode.pprev stores hash value. cpu y found E right before it was deleted on cpu x. E is now re-init'd on cpu z. cpu y was preempted before checking for expiry and/or confirm bit. ->refcnt set to 1 E now owned by skb ->timeout set to 30000 If cpu y were to resume now, it would observe E as expired but would skip E due to missing CONFIRMED bit. nf_contrack_confirm gets called sets: ct->status = CONFIRMED This is wrong: E is not yet added to hashtable. cpu y resumes, it observes E as expired but CONFIRMED: <resumes> nf_ct_expired() -> yes (ct->timeout is 30s) confirmed bit set. cpu y will try to delete E from the hashtable: nf_ct_delete() -> set DYING bit __nf_ct_delete_from_lists Even this scenario doesn't guarantee a crash: cpu z still holds the table bucket lock(s) so y blocks: wait for spinlock held by z CONFIRMED is set but there is no guarantee ct will be added to hash: "chaintoolong" or "clash resolution" logic both skip the insert step. reply hnode.pprev still stores the hash value. unlocks spinlock return NF_DROP <unblocks, then crashes on hlist_nulls_del_rcu pprev> In case CPU z does insert the entry into the hashtable, cpu y will unlink E again right away but no crash occurs. Without 'cpu y' race, 'garbage' hlist is of no consequence: ct refcnt remains at 1, eventually skb will be free'd and E gets destroyed via: nf_contrack_put -> nf_contrack_destroy -> nf_ct_destroy. To resolve this, move the IPS_CONFIRMED assignment after the table insertion but before the unlock. Pablo points out that the confirm-bit-store could be reordered to happen before hlist add resp. the timeout fixup, so switch to set_bit and before_atomic memory barrier to prevent this. It doesn't matter if other CPUs can observe a newly inserted entry right before the CONFIRMED bit was set: Such event cannot be distinguished from above "E is the old incarnation" case: the entry will be skipped. Also change nf_ct_should_gc() to first check the confirmed bit. The gc sequence is: 1. Check if entry has expired, if not skip to next entry 2. Obtain a reference to the expired entry. 3. Call nf_ct_should_gc() to double-check step 1. nf_ct_should_gc() is thus called only for entries that already failed an expiry check. After this patch, once the confirmed bit check pas ---truncated---
CVE-2025-38473	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix null-ptr-deref in l2cap_sock_resume_cb() syzbot reported null-ptr-deref in l2cap_sock_resume_cb(). [0] l2cap_sock_resume_cb() has a similar problem that was fixed by commit 1bff51ea59a9 ("Bluetooth: fix use-after-free error in lock_sock_nested()"). Since both l2cap_sock_kill() and l2cap_sock_resume_cb() are executed under l2cap_sock_resume_cb(), we can avoid the issue simply by checking if chan->data is NULL. Let's not access to the killed socket in l2cap_sock_resume_cb(). [0]: BUG: KASAN: null-ptr-deref in instrument_atomic_write include/linux/instrumented.h:82 [inline] BUG: KASAN: null-ptr-deref in clear_bit include/asm-generic/bitops/instrumented-atomic.h:41 [inline] BUG: KASAN: null-ptr-deref in l2cap_sock_resume_cb+0xb4/0x17c net/bluetooth/l2cap_sock.c:1711 Write of size 8 at addr 0000000000000570 by task kworker/u9:0/52 CPU: 1 UID: 0 PID: 52 Comm: kworker/u9:0 Not tainted 6.16.0-rc4-syzkaller-g7482bb149b9f #0 PREEMPT Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 Workqueue: hci0 hci_rx_work Call trace: show_stack+0x2c/0x3c arch/arm64/kernel/stacktrace.c:501 (C) __dump_stack+0x30/0x40 lib/dump_stack.c:94 dump_stack_lvl+0xd8/0x12c lib/dump_stack.c:120 print_report+0x58/0x84 mm/kasan/report.c:524 kasan_report+0xb0/0x110 mm/kasan/report.c:634 check_region_inline mm/kasan/generic.c:-1 [inline] kasan_check_range+0x264/0x2a4 mm/kasan/generic.c:189 __kasan_check_write+0x20/0x30 mm/kasan/shadow.c:37 instrument_atomic_write include/linux/instrumented.h:82 [inline] clear_bit include/asm-generic/bitops/instrumented-atomic.h:41 [inline] l2cap_sock_resume_cb+0xb4/0x17c net/bluetooth/l2cap_sock.c:1711 l2cap_security_cfm+0x524/0xea0 net/bluetooth/l2cap_core.c:7357 hci_auth_cfm include/net/bluetooth/hci_core.h:2092 [inline] hci_auth_complete_evt+0x2e8/0xa4c net/bluetooth/hci_event.c:3514 hci_event_func net/bluetooth/hci_event.c:7511 [inline] hci_event_packet+0x650/0xe9c net/bluetooth/hci_event.c:7565 hci_rx_work+0x320/0xb18 net/bluetooth/hci_core.c:4070 process_one_work+0x7e8/0x155c kernel/workqueue.c:3238 process_scheduled_works kernel/workqueue.c:3321 [inline] worker_thread+0x958/0xed8 kernel/workqueue.c:3402 kthread+0x5fc/0x75c kernel/kthread.c:464 ret_from_fork+0x10/0x20 arch/arm64/kernel/entry.S:847

CVE ID	Description
CVE-2025-38474	In the Linux kernel, the following vulnerability has been resolved: usb: net: sierra: check for no status endpoint The driver checks for having three endpoints and having bulk in and out endpoints, but not that the third endpoint is interrupt input. Rectify the omission.
CVE-2025-38476	In the Linux kernel, the following vulnerability has been resolved: rpl: Fix use-after-free in rpl_do_srh_inline(). Running lwt_dst_cache_ref_loop.sh in selftest with KASAN triggers the splat below [0]. rpl_do_srh_inline() fetches ipv6_hdr(skb) and accesses it after skb_cow_head(), which is illegal as the header could be freed then. Let's fix it by making oldhdr to a local struct instead of a pointer. [0]: [root@fedora net]# ./lwt_dst_cache_ref_loop.sh ... TEST: rpl (input) [57.631529] ===== BUG: KASAN: slab-use-after-free in rpl_do_srh_inline.isra.0 (net/ipv6/rpl_ip tunnel.c:174) Read of size 40 at addr ffff888122bf96d8 by task ping6/1543 CPU: 50 UID: 0 PID: 1543 Comm: ping6 Not tainted 6.16.0-rc5-01302-gfadd1e6231b1 #23 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 Call Trace: <IRQ> dump_stack_lvl (lib/dump_stack.c:122) print_report (mm/kasan/report.c:409 mm/kasan/report.c:521) kasan_report (mm/kasan/report.c:221 mm/kasan/report.c:636) kasan_check_range (mm/kasan/generic.c:175 (discriminator 1) mm/kasan/generic.c:189 (discriminator 1)) __asan_memmove (mm/kasan/shadow.c:94 (discriminator 2)) rpl_do_srh_inline.isra.0 (net/ipv6/rpl_ip tunnel.c:174) rpl_input (net/ipv6/rpl_ip tunnel.c:201 net/ipv6/rpl_ip tunnel.c:282) lwtunnel_input (net/core/lwtunnel.c:459) ipv6_rcv (/include/net/dst.h:471 (discriminator 1) ./include/net/dst.h:469 (discriminator 1) net/ipv6/ip6_input.c:79 (discriminator 1) ./include/linux/netfilter.h:317 (discriminator 1) ./include/linux/netfilter.h:311 (discriminator 1) net/ipv6/ip6_input.c:311 (discriminator 1)) __netif_receive_skb_one_core (net/core/dev.c:5967) process_backlog (/include/linux/rcupdate.h:869 net/core/dev.c:6440) __napi_poll.constprop.0 (net/core/dev.c:7452) net_rx_action (net/core/dev.c:7518 net/core/dev.c:7643) handle_softirqs (kernel/softirq.c:579) do_softirq (kernel/softirq.c:480 (discriminator 20)) </IRQ> <TASK> __local_bh_enable_ip (kernel/softirq.c:407) __dev_queue_xmit (net/core/dev.c:4740) ip6_finish_output2 (/include/linux/netdevice.h:3358 ./include/net/ neighbour.h:526 ./include/net/ neighbour.h:540 net/ipv6/ip6_output.c:141) ip6_finish_output (net/ipv6/ip6_output.c:215 net/ipv6/ip6_output.c:226) ip6_output (/include/linux/netfilter.h:306 net/ipv6/ip6_output.c:248) ip6_send_skb (net/ipv6/ip6_output.c:1983) rawv6_sendmsg (net/ipv6/raw.c:588 net/ipv6/raw.c:918) __sys_sendto (net/socket.c:714 (discriminator 1) net/socket.c:729 (discriminator 1) net/socket.c:2228 (discriminator 1)) __x64_sys_sendto (net/socket.c:2231) do_syscall_64 (arch/x86/entry/syscall_64.c:63 (discriminator 1) arch/x86/entry/syscall_64.c:94 (discriminator 1)) entry_SYSCALL_64_after_hwframe (arch/x86/entry/entry_64.S:130) RIP: 0033:0x7f68c9fb2a06 Code: 5d e8 41 8b 93 08 03 00 00 59 5e 48 83 f8 fc 75 19 83 e2 39 83 fa 08 75 11 e8 26 ff ff ff 66 0f 1f 44 00 00 48 8b 45 10 0f 05 <48> 8b 5d f8 c9 c3 0f 1f 40 00 f3 0f 1e fa 55 48 89 e5 48 83 ec 08 RSP: 002b:00007ffefb7c53d0 EFLAGS: 00000202 ORIG_RAX: 000000000000002c RAX: ffffffff80000000 RBX: 0000564cd69f10a0 RCX: 00007f68c9fb2a06 RDX: 0000000000000040 RSI: 0000564cd69f10a4 RDI: 0000000000000003 RBP: 00007ffefb7c53f0 R08: 0000564cd6a032ac R09: 000000000000001c R10: 0000000000000000 R11: 0000000000000202 R12: 0000564cd69f10a4 R13: 0000000000000040 R14: 00007ffefb7c66e0 R15: 0000564cd69f10a0 </TASK> Allocated by task 1543: kasan_save_stack (mm/kasan/common.c:48) kasan_save_track (mm/kasan/common.c:60 (discriminator 1) mm/kasan/common.c:69 (discriminator 1)) __kasan_slab_alloc (mm/kasan/common.c:319 mm/kasan/common.c:345) kmem_cache_alloc_node_noprof (/include/linux/kasan.h:250 mm/slub.c:4148 mm/slub.c:4197 mm/slub.c:4249) kmallocc_reserve (net/core/skbuff.c:581 (discriminator 88)) __alloc_skb (net/core/skbuff.c:669) __ip6_append_data (net/ipv6/ip6_output.c:1672 (discriminator 1)) ip6_ --- truncated---
CVE-2025-38477	In the Linux kernel, the following vulnerability has been resolved: net/sched: sch_qfq: Fix race condition on qfq_aggregate A race condition can occur when 'agg' is modified in qfq_change_agg (called during qfq_enqueue) while other threads access it concurrently. For example, qfq_dump_class may trigger a NULL dereference, and qfq_delete_class may cause a use-after-free. This patch addresses the issue by: 1. Moved qfq_destroy_class into the critical section. 2. Added sch_tree_lock protection to qfq_dump_class and qfq_dump_class_stats.
CVE-2025-38478	In the Linux kernel, the following vulnerability has been resolved: comedi: Fix initialization of data for instructions that write to subdevice Some Comedi subdevice instruction handlers are known to access instruction data elements beyond the first 'insn->n' elements in some cases. The 'do_insn_ioctl()' and 'do_insnlist_ioctl()' functions allocate at least 'MIN_SAMPLES' (16) data elements to deal with this, but they do not initialize all of that. For Comedi instruction codes that write to the subdevice, the first 'insn->n' data elements are copied from user-space, but the remaining elements are left uninitialized. That could be a problem if the subdevice instruction handler reads the uninitialized data. Ensure that the first 'MIN_SAMPLES' elements are initialized before calling these instruction handlers, filling the uncopied elements with 0. For 'do_insnlist_ioctl()', the same data buffer elements are used for handling a list of instructions, so ensure the first 'MIN_SAMPLES' elements are initialized for each instruction that writes to the subdevice.

CVE ID	Description
CVE-2025-38480	In the Linux kernel, the following vulnerability has been resolved: comedi: Fix use of uninitialized data in <code>insn_rw_emulate_bits()</code> For Comedi <code>INSN_READ</code> and <code>INSN_WRITE</code> instructions on "digital" subdevices (subdevice types <code>COMEDI_SUBD_DI</code> , <code>COMEDI_SUBD_DO</code> , and <code>COMEDI_SUBD_DIO</code>), it is common for the subdevice driver not to have <code>insn_read</code> and <code>insn_write</code> handler functions, but to have an <code>insn_bits</code> handler function for handling Comedi <code>INSN_BITS</code> instructions. In that case, the subdevice's <code>insn_read</code> and/or <code>insn_write</code> function handler pointers are set to point to the <code>insn_rw_emulate_bits()</code> function by <code>__comedi_device_postconfig()</code> . For <code>INSN_WRITE</code> , <code>insn_rw_emulate_bits()</code> currently assumes that the supplied <code>data[0]</code> value is a valid copy from user memory. It will at least exist because <code>do_insnlist_ioctl()</code> and <code>do_insn_ioctl()</code> in <code>comedi_fops.c</code> ensure at least <code>MIN_SAMPLES</code> (16) elements are allocated. However, if <code>insn->n</code> is 0 (which is allowable for <code>INSN_READ</code> and <code>INSN_WRITE</code> instructions, then <code>data[0]</code> may contain uninitialized data, and certainly contains invalid data, possibly from a different instruction in the array of instructions handled by <code>do_insnlist_ioctl()</code> . This will result in an incorrect value being written to the digital output channel (or to the digital input/output channel if configured as an output), and may be reflected in the internal saved state of the channel. Fix it by returning 0 early if <code>insn->n</code> is 0, before reaching the code that accesses <code>data[0]</code> . Previously, the function always returned 1 on success, but it is supposed to be the number of data samples actually read or written up to <code>insn->n</code> , which is 0 in this case.
CVE-2025-38481	In the Linux kernel, the following vulnerability has been resolved: comedi: Fail COMEDI_INSNLIST ioctl if n_insns is too large The handling of the <code>COMEDI_INSNLIST</code> ioctl allocates a kernel buffer to hold the array of <code>struct comedi_insn</code> , getting the length from the <code>n_insns</code> member of the <code>struct comedi_insnlist</code> supplied by the user. The allocation will fail with a WARNING and a stack dump if it is too large. Avoid that by failing with an <code>-EINVAL</code> error if the supplied <code>n_insns</code> value is unreasonable. Define the limit on the <code>n_insns</code> value in the <code>MAX_INSNS</code> macro. Set this to the same value as <code>MAX_SAMPLES</code> (65536), which is the maximum allowed sum of the values of the member <code>n</code> in the array of <code>struct comedi_insn</code> , and sensible comedi instructions will have an <code>n</code> of at least 1.
CVE-2025-38482	In the Linux kernel, the following vulnerability has been resolved: comedi: das6402: Fix bit shift out of bounds When checking for a supported IRQ number, the following test is used: <code>/* IRQs 2,3,5,6,7, 10,11,15 are valid for "enhanced" mode */ if ((1 << it->options[1]) & 0x8cec) {</code> However, <code>it->options[i]</code> is an unchecked <code>int</code> value from userspace, so the shift amount could be negative or out of bounds. Fix the test by requiring <code>it->options[1]</code> to be within bounds before proceeding with the original test. Valid <code>it->options[1]</code> values that select the IRQ will be in the range [1,15]. The value 0 explicitly disables the use of interrupts.
CVE-2025-38483	In the Linux kernel, the following vulnerability has been resolved: comedi: das16m1: Fix bit shift out of bounds When checking for a supported IRQ number, the following test is used: <code>/* only irq's 2, 3, 4, 5, 6, 7, 10, 11, 12, 14, and 15 are valid */ if ((1 << it->options[1]) & 0xdcfc) {</code> However, <code>it->options[i]</code> is an unchecked <code>int</code> value from userspace, so the shift amount could be negative or out of bounds. Fix the test by requiring <code>it->options[1]</code> to be within bounds before proceeding with the original test.
CVE-2025-38485	In the Linux kernel, the following vulnerability has been resolved: iio: accel: fxls8962af: Fix use after free in <code>fxls8962af_fifo_flush</code> <code>fxls8962af_fifo_flush()</code> uses <code>indio_dev->active_scan_mask</code> (with <code>iio_for_each_active_channel()</code>) without making sure the <code>indio_dev</code> stays in buffer mode. There is a race if <code>indio_dev</code> exits buffer mode in the middle of the interrupt that flushes the fifo. Fix this by calling <code>synchronize_irq()</code> to ensure that no interrupt is currently running when disabling buffer mode. Unable to handle kernel NULL pointer dereference at virtual address 00000000 when read [...] <code>_find_first_bit_le</code> from <code>fxls8962af_fifo_flush+0x17c/0x290</code> <code>fxls8962af_fifo_flush</code> from <code>fxls8962af_interrupt+0x80/0x178</code> <code>fxls8962af_interrupt</code> from <code>irq_thread_fn+0x1c/0x7c</code> <code>irq_thread_fn</code> from <code>irq_thread+0x110/0x1f4</code> <code>irq_thread</code> from <code>kthread+0xe0/0xfc</code> <code>kthread</code> from <code>ret_from_fork+0x14/0x2c</code>

CVE ID	Description
CVE-2025-38486	In the Linux kernel, the following vulnerability has been resolved: soundwire: Revert "soundwire: qcom: Add set_channel_map api support" This reverts commit 7796c97df6b1b2206681a07f3c80f6023a6593d5. This patch broke Dragonboard 845c (sdm845). I see: Unexpected kernel BRK exception at EL1 Internal error: BRK handler: 00000000f20003e8 [#1] SMP pc : qcom_swrn_set_channel_map +0x7c/0x80 [soundwire_qcom] lr : snd_soc_dai_set_channel_map+0x34/0x78 Call trace: qcom_swrn_set_channel_map+0x7c/0x80 [soundwire_qcom] (P) sdm845_dai_init +0x18c/0x2e0 [snd_soc_sdm845] snd_soc_link_init+0x28/0x6c snd_soc_bind_card +0x5f4/0xb0c snd_soc_register_card+0x148/0x1a4 devm_snd_soc_register_card+0x50/0xb0 sdm845_snd_platform_probe+0x124/0x148 [snd_soc_sdm845] platform_probe+0x6c/0xd0 really_probe+0xc0/0x2a4 __driver_probe_device+0x7c/0x130 driver_probe_device+0x40/0x118 __device_attach_driver+0xc4/0x108 bus_for_each_drv+0x8c/0xf0 __device_attach+0xa4/0x198 device_initial_probe+0x18/0x28 bus_probe_device+0xb8/0xbc deferred_probe_work_func +0xac/0xfc process_one_work+0x244/0x658 worker_thread+0x1b4/0x360 kthread+0x148/0x228 ret_from_fork+0x10/0x20 Kernel panic - not syncing: BRK handler: Fatal exception Dan has also reported following issues with the original patch https://lore.kernel.org/all/33fe8fe7-719a-405a-9ed2-d9f816ce1d57@sabinyo.mountain/ Bug #1: The zeroeth element of ctrl->pconfig[] is supposed to be unused. We start counting at 1. However this code sets ctrl->pconfig[0].ch_mask = 128. Bug #2: There are SLIM_MAX_TX_PORTS (16) elements in tx_ch[] array but only QCOM_SDW_MAX_PORTS + 1 (15) in the ctrl->pconfig[] array so it corrupts memory like Yongqin Liu pointed out. Bug 3: Like Jie Gan pointed out, it erases all the tx information with the rx information.
CVE-2025-38487	In the Linux kernel, the following vulnerability has been resolved: soc: aspeed: lpc-snoop: Don't disable channels that aren't enabled Mitigate e.g. the following: # echo 1e789080.lpc-snoop > /sys/bus/platform/drivers/aspeed-lpc-snoop/unbind ... [120.363594] Unable to handle kernel NULL pointer dereference at virtual address 00000004 when write [120.373866] [000000004] *pgd=00000000 [120.377910] Internal error: Oops: 805 [#1] SMP ARM [120.383306] CPU: 1 UID: 0 PID: 315 Comm: sh Not tainted 6.15.0-rc1-00009-g926217bc7d7d-dirty #20 NONE ... [120.679543] Call trace: [120.679559] misc_deregister from aspeed_lpc_snoop_remove+0x84/0xac [120.692462] aspeed_lpc_snoop_remove from platform_remove+0x28/0x38 [120.700996] platform_remove from device_release_driver_internal +0x188/0x200 ...
CVE-2025-38488	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix use-after-free in crypt_message when using async crypto The CVE-2024-50047 fix removed asynchronous crypto handling from crypt_message(), assuming all crypto operations are synchronous. However, when hardware crypto accelerators are used, this can cause use-after-free crashes: crypt_message() // Allocate the creq buffer containing the req creq = smb2_get_aead_req(..., &req); // Async encryption returns -EINPROGRESS immediately rc = enc ? crypto_aead_encrypt(req) : crypto_aead_decrypt(req); // Free creq while async operation is still in progress kvfree_sensitive(creq, ...); Hardware crypto modules often implement async AEAD operations for performance. When crypto_aead_encrypt/decrypt() returns -EINPROGRESS, the operation completes asynchronously. Without crypto_wait_req(), the function immediately frees the request buffer, leading to crashes when the driver later accesses the freed memory. This results in a use-after-free condition when the hardware crypto driver later accesses the freed request structure, leading to kernel crashes with NULL pointer dereferences. The issue occurs because crypto_alloc_aead() with mask=0 doesn't guarantee synchronous operation. Even without CRYPTO_ALG_ASYNC in the mask, async implementations can be selected. Fix by restoring the async crypto handling: - DECLARE_CRYPT_WAIT(wait) for completion tracking - aead_request_set_callback() for async completion notification - crypto_wait_req() to wait for operation completion This ensures the request buffer isn't freed until the crypto operation completes, whether synchronous or asynchronous, while preserving the CVE-2024-50047 fix.
CVE-2025-38489	In the Linux kernel, the following vulnerability has been resolved: s390/bpf: Fix bpf_arch_text_poke() with new_addr == NULL again Commit 7ded842b356d ("s390/bpf: Fix bpf_plt pointer arithmetic") has accidentally removed the critical piece of commit c730fce7c70c ("s390/bpf: Fix bpf_arch_text_poke() with new_addr == NULL"), causing intermittent kernel panics in e.g. perf's on_switch() prog to reappear. Restore the fix and add a comment.

CVE ID	Description
CVE-2025-38490	In the Linux kernel, the following vulnerability has been resolved: net: libwx: remove duplicate page_pool_put_full_page() page_pool_put_full_page() should only be invoked when freeing Rx buffers or building a skb if the size is too short. At other times, the pages need to be reused. So remove the redundant page put. In the original code, double free pages cause kernel panic: <pre>[876.949834] __irq_exit_rcu+0xc7/0x130 [876.949836] common_interrupt+0xb8/0xd0 [876.949838] </IRQ> [876.949838] <TASK> [876.949840] asm_common_interrupt+0x22/0x40 [876.949841] RIP: 0010:cpuidle_enter_state+0xc2/0x420 [876.949843] Code: 00 00 e8 d1 1d 5e ff e8 ac f0 ff ff 49 89 c5 0f 1f 44 00 00 31 ff e8 cd fc 5c ff 45 84 ff 0f 85 40 02 00 00 fb 0f 1f 44 00 00 <45> 85 f6 0f 88 84 01 00 00 49 63 d6 48 8d 04 52 48 8d 04 82 49 8d [876.949844] RSP: 0018:ffffaa7340267e78 EFLAGS: 00000246 [876.949845] RAX: ffff9e3f135be000 RBX: 0000000000000002 RCX: 0000000000000000 [876.949846] RDX: 000000cc2dc4cb7c RSI: ffffffffff89ee49ae RDI: fffffffffff89ef9f9e [876.949847] RBP: ffff9e378f940800 R08: 0000000000000002 R09: 000000000000000ed [876.949848] R10: 0000000000000af8 R11: ffff9e3e9e5a9b6c R12: ffffff8a6d8580 [876.949849] R13: 000000cc2dc4cb7c R14: 0000000000000002 R15: 0000000000000000 [876.949852] ? cpuidle_enter_state+0xb3/0x420 [876.949855] cpuidle_enter +0x29/0x40 [876.949857] cpuidle_idle_call+0xfd/0x170 [876.949859] do_idle+0x7a/0xc0 [876.949861] cpu_startup_entry+0x25/0x30 [876.949862] start_secondary+0x117/0x140 [876.949864] common_startup_64+0x13e/0x148 [876.949867] </TASK> [876.949868] ---[end trace 0000000000000000]--- [876.949869] -----[cut here]----- [876.949870] list_del corruption, ffffead40445a348->next is NULL [876.949873] WARNING: CPU: 14 PID: 0 at lib/ list_debug.c:52 __list_del_entry_valid_or_report+0x67/0x120 [876.949875] Modules linked in: snd_hrtimer(E) bnep(E) binfmt_misc(E) amdgpu(E) squashfs(E) vfat(E) loop(E) fat(E) amd_atl(E) snd_hda_codec_realtek(E) intel_rapl_msr(E) snd_hda_codec_generic(E) intel_rapl_common(E) snd_hda_scodec_component(E) snd_hda_codec_hdmi(E) snd_hda_intel(E) edac_mce_amd(E) snd_intel_dspcfg(E) snd_hda_codec(E) snd_hda_core(E) amdxcip(E) kvm_amd(E) snd_hwdep(E) gpu_sched(E) drm_panel_backlight_quirks(E) cec(E) snd_pcm(E) drm_buddy(E) snd_seq_dummy(E) drm_ttm_helper(E) btusb(E) kvm(E) snd_seq_oss(E) btrtl(E) ttm(E) btintel(E) snd_seq_midi(E) btbcm(E) drm_exec(E) snd_seq_midi_event(E) i2c_algo_bit(E) snd_rawmidi(E) bluetooth(E) drm_suballoc_helper(E) irqbypass(E) snd_seq(E) ghash_clmulni_intel(E) sha512_ssse3(E) drm_display_helper(E) aesni_intel(E) snd_seq_device(E) rfkill(E) snd_timer(E) gf128mul(E) drm_client_lib(E) drm_kms_helper(E) snd(E) i2c_piix4(E) joydev(E) soundcore(E) wmi_bmof(E) ccp(E) k10temp(E) i2c_smbus(E) gpio_amdpt(E) i2c_designware_platform(E) gpio_generic(E) sg(E) [876.949914] i2c_designware_core(E) sch_fq_codel(E) parport_pc(E) drm(E) ppdev(E) lp(E) parport(E) fuse(E) nfnetlink(E) ip_tables(E) ext4 crc16 mbcache jbd2 sd_mod sfp mdio_i2c i2c_core txgbe ahci ngbe pcs_xpcs libahci libwx r8169 phylink libata realtek ptp pps_core video wmi [876.949933] CPU: 14 UID: 0 PID: 0 Comm: swapper/14 Kdump: loaded Tainted: G W E 6.16.0- rc2+ #20 PREEMPT(voluntary) [876.949935] Tainted: [W]=WARN, [E]=UNSIGNED_MODULE [876.949936] Hardware name: Micro-Star International Co., Ltd. MS-7E16/X670E GAMING PLUS WIFI (MS-7E16), BIOS 1.90 12/31/2024 [876.949936] RIP: 0010: __list_del_entry_valid_or_report +0x67/0x120 [876.949938] Code: 00 00 00 48 39 7d 08 0f 85 a6 00 00 00 5b b8 01 00 00 00 5d 41 5c e9 73 0d 93 ff 48 89 fe 48 c7 c7 a0 31 e8 89 e8 59 7c b3 ff <0f> 0b 31 c0 5b 5d 41 5c e9 57 0d 93 ff 48 89 fe 48 c7 c7 c8 31 e8 [876.949940] RSP: 0018:ffffaa73405d0c60 EFLAGS: 00010282 [876.949941] RAX: 0000000000000000 RBX: ffffead40445a348 RCX: 0000000000000000 [876.949942] RDX: 0000000000000105 RSI: 00000 ---truncated---</pre>

CVE ID	Description
CVE-2025-38491	In the Linux kernel, the following vulnerability has been resolved: mptcp: make fallback action and fallback decision atomic Syzkaller reported the following splat: WARNING: CPU: 1 PID: 7704 at net/mptcp/protocol.h:1223 __mptcp_do_fallback net/mptcp/protocol.h:1223 [inline] WARNING: CPU: 1 PID: 7704 at net/mptcp/protocol.h:1223 mptcp_do_fallback net/mptcp/protocol.h:1244 [inline] WARNING: CPU: 1 PID: 7704 at net/mptcp/protocol.h:1223 check_fully_established net/mptcp/options.c:982 [inline] WARNING: CPU: 1 PID: 7704 at net/mptcp/protocol.h:1223 mptcp_incoming_options+0x21a8/0x2510 net/mptcp/options.c:1153 Modules linked in: CPU: 1 UID: 0 PID: 7704 Comm: syz.3.1419 Not tainted 6.16.0-rc3-gbd5ce2324dba #20 PREEMPT(voluntary) Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010: __mptcp_do_fallback net/mptcp/protocol.h:1223 [inline] RIP: 0010: mptcp_do_fallback net/mptcp/protocol.h:1244 [inline] RIP: 0010: check_fully_established net/mptcp/options.c:982 [inline] RIP: 0010: mptcp_incoming_options+0x21a8/0x2510 net/mptcp/options.c:1153 Code: 24 18 e8 bb 2a 00 fd e9 1b df ff ff e8 b1 21 0f 00 e8 ec 5f c4 fc 44 0f b7 ac 24 b0 00 00 00 e9 54 f1 ff ff e8 d9 5f c4 fc 90 <0f> 0b 90 e9 b8 f4 ff ff e8 8b 2a 00 fd e9 8d e6 ff ff e8 81 2a 00 RSP: 0018:ffff8880a3f08448 EFLAGS: 00010246 RAX: 0000000000000000 RBX: ffff8880180a8000 RCX: ffffffff84afc45 RDX: ffff888090223700 RSI: ffffffff84afdaa7 RDI: 0000000000000001 RBP: ffff888017955780 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000 R13: ffff8880180a8910 R14: ffff8880a3e9d058 R15: 0000000000000000 FS: 0000555791b8500(0000) GS:ffff88811c495000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000110c2800b7 CR3: 00000000058e44000 CR4: 0000000000350ef0 Call Trace: <IRQ> tcp_reset+0x26f/0x2b0 net/ipv4/tcp_input.c:4432 tcp_validate_incoming+0x1057/0x1b60 net/ipv4/tcp_input.c:5975 tcp_rcv_established+0x5b5/0x21f0 net/ipv4/tcp_input.c:6166 tcp_v4_do_rcv+0x5dc/0xa70 net/ipv4/tcp_ipv4.c:1925 tcp_v4_rcv+0x3473/0x44a0 net/ipv4/tcp_ipv4.c:2363 ip_protocol_deliver_rcu+0xba/0x480 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x2f1/0x500 net/ipv4/ip_input.c:233 NF_HOOK include/linux/netfilter.h:317 [inline] NF_HOOK include/linux/netfilter.h:311 [inline] ip_local_deliver+0x1be/0x560 net/ipv4/ip_input.c:254 dst_input include/net/dst.h:469 [inline] ip_rcv_finish net/ipv4/ip_input.c:447 [inline] NF_HOOK include/linux/netfilter.h:317 [inline] NF_HOOK include/linux/netfilter.h:311 [inline] ip_rcv+0x514/0x810 net/ipv4/ip_input.c:567 __netif_receive_skb_one_core+0x197/0x1e0 net/core/dev.c:5975 __netif_receive_skb+0x1f/0x120 net/core/dev.c:6088 process_backlog+0x301/0x1360 net/core/dev.c:6440 __napi_poll.constprop.0+0xba/0x550 net/core/dev.c:7453 napi_poll net/core/dev.c:7517 [inline] net_rx_action+0xb44/0x1010 net/core/dev.c:7644 handle_softirqs+0x1d0/0x770 kernel/softirq.c:579 do_softirq+0x3f/0x90 kernel/softirq.c:480 <IRQ> <TASK> __local_bh_enable_ip+0xed/0x110 kernel/softirq.c:407 local_bh_enable include/linux/bottom_half.h:33 [inline] inet_csk_listen_stop+0x2c5/0x1070 net/ipv4/inet_connection_sock.c:1524 mptcp_check_listen_stop.part.0+0x1cc/0x220 net/mptcp/protocol.c:2985 mptcp_check_listen_stop net/mptcp/mib.h:118 [inline] __mptcp_close+0x9b9/0xbd0 net/mptcp/protocol.c:3000 mptcp_close+0x2f/0x140 net/mptcp/protocol.c:3066 inet_release+0xed/0x200 net/ipv4/af_inet.c:435 inet6_release+0x4f/0x70 net/ipv6/af_inet6.c:487 __sock_release+0xb3/0x270 net/socket.c:649 sock_close+0x1c/0x30 net/socket.c:1439 __fput+0x402/0xb70 fs/file_table.c:465 task_work_run+0x150/0x240 kernel/task_work.c:227 resume_user_mode_work include/linux/resume_user_mode.h:50 [inline] exit_to_user_mode_loop+0xd4 ---truncated---
CVE-2025-38493	In the Linux kernel, the following vulnerability has been resolved: tracing/osnoise: Fix crash in timerlat_dump_stack() We have observed kernel panics when using timerlat with stack saving, with the following dmesg output: memcopy: detected buffer overflow: 88 byte write of buffer size 0 WARNING: CPU: 2 PID: 8153 at lib/string_helpers.c:1032 __fortify_report+0x55/0xa0 CPU: 2 UID: 0 PID: 8153 Comm: timerlatu/2 Kdump: loaded Not tainted 6.15.3-200.fc42.x86_64 #1 PREEMPT(lazy) Call Trace: <TASK> ? trace_buffer_lock_reserve+0x2a/0x60 __fortify_panic+0xd/0xf __timerlat_dump_stack.cold+0xd/0xd timerlat_dump_stack.part.0+0x47/0x80 timerlat_fd_read+0x36d/0x390 vfs_read+0xe2/0x390 ? syscall_exit_to_user_mode+0x1d5/0x210 ksys_read+0x73/0xe0 do_syscall_64+0x7b/0x160 ? exc_page_fault+0x7e/0x1a0 entry_SYSCALL_64_after_hwframe+0x76/0x7e __timerlat_dump_stack() constructs the ftrace stack entry like this: struct stack_entry *entry; ... memcopy(&entry->caller, fstack->calls, size); entry->size = fstack->nr_entries; Since commit e7186af7b26 ("tracing: Add back FORTIFY_SOURCE logic to kernel_stack event structure"), struct stack_entry marks its caller field with __counted_by(size). At the time of the memcopy, entry->size contains garbage from the ringbuffer, which under some circumstances is zero, triggering a kernel panic by buffer overflow. Populate the size field before the memcopy so that the out-of-bounds check knows the correct size. This is analogous to __ftrace_trace_stack().
CVE-2025-38494	In the Linux kernel, the following vulnerability has been resolved: HID: core: do not bypass hid_hw_raw_request hid_hw_raw_request() is actually useful to ensure the provided buffer and length are valid. Directly calling in the low level transport driver function bypassed those checks and allowed invalid paramto be used.

CVE ID	Description
CVE-2025-38495	In the Linux kernel, the following vulnerability has been resolved: HID: core: ensure the allocated report buffer can contain the reserved report ID When the report ID is not used, the low level transport drivers expect the first byte to be 0. However, currently the allocated buffer not account for that extra byte, meaning that instead of having 8 guaranteed bytes for implement to be working, we only have 7.
CVE-2025-38496	In the Linux kernel, the following vulnerability has been resolved: dm-bufio: fix sched in atomic context If "try_verify_in_tasklet" is set for dm-verity, DM_BUFIO_CLIENT_NO_SLEEP is enabled for dm-bufio. However, when bufio tries to evict buffers, there is a chance to trigger scheduling in spin_lock_bh, the following warning is hit: BUG: sleeping function called from invalid context at drivers/md/dm-bufio.c:2745 in_atomic(): 1, irqs_disabled(): 0, non_block: 0, pid: 123, name: kworker/2:2 preempt_count: 201, expected: 0 RCU nest depth: 0, expected: 0 4 locks held by kworker/2:2/123: #0: ffff88800a2d1548 ((wq_completion)dm_bufio_cache){....}-{0:0}, at: process_one_work+0xe46/0x1970 #1: fffffc90000d97d20 ((work_completion)(dm_bufio_replacement_work)){....}-{0:0}, at: process_one_work+0x763/0x1970 #2: ffffffff8555b528 (dm_bufio_clients_lock){....}-{3:3}, at: do_global_cleanup+0x1ce/0x710 #3: ffff88801d5820b8 (&c->spinlock){....}-{2:2}, at: do_global_cleanup+0x2a5/0x710 Preemption disabled at: [0000000000000000] 0x0 CPU: 2 UID: 0 PID: 123 Comm: kworker/2:2 Not tainted 6.16.0-rc3-g90548c634bd0 #305 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 Workqueue: dm_bufio_cache do_global_cleanup Call Trace: <TASK> dump_stack_lvl+0x53/0x70 __might_resched+0x360/0x4e0 do_global_cleanup+0x2f5/0x710 process_one_work+0x7db/0x1970 worker_thread+0x518/0xea0 kthread+0x359/0x690 ret_from_fork+0xf3/0x1b0 ret_from_fork_asm+0x1a/0x30 </TASK> That can be reproduced by: veritysetup format --data-block-size=4096 --hash-block-size=4096 /dev/vda /dev/vdb SIZE=\$(blockdev --getsz /dev/vda) dmsetup create myverity -r --table "0 \$SIZE verity 1 /dev/vda /dev/vdb 4096 4096 <data_blocks> 1 sha256 <root_hash> <salt> 1 try_verify_in_tasklet" mount /dev/dm-0 /mnt -o ro echo 102400 > /sys/module/dm_bufio/parameters/max_cache_size_bytes [read files in /mnt]
CVE-2025-38497	In the Linux kernel, the following vulnerability has been resolved: usb: gadget: configs: Fix OOB read on empty string write When writing an empty string to either 'qw_sign' or 'landingPage' sysfs attributes, the store functions attempt to access page[1 - 1] before validating that the length 'l' is greater than zero. This patch fixes the vulnerability by adding a check at the beginning of os_desc_qw_sign_store() and webusb_landingPage_store() to handle the zero-length input case gracefully by returning immediately.
CVE-2025-38498	In the Linux kernel, the following vulnerability has been resolved: do_change_type(): refuse to operate on unmounted/not ours mounts Ensure that propagation settings can only be changed for mounts located in the caller's mount namespace. This change aligns permission checking with the rest of mount(2).
CVE-2025-38499	In the Linux kernel, the following vulnerability has been resolved: clone_private_mnt(): make sure that caller has CAP_SYS_ADMIN in the right usersns What we want is to verify there is that clone won't expose something hidden by a mount we wouldn't be able to undo. "Wouldn't be able to undo" may be a result of MNT_LOCKED on a child, but it may also come from lacking admin rights in the usersns of the namespace mount belongs to. clone_private_mnt() checks the former, but not the latter. There's a number of rather confusing CAP_SYS_ADMIN checks in various usersns during the mount, especially with the new mount API; they serve different purposes and in case of clone_private_mnt() they usually, but not always end up covering the missing check mentioned above.

CVE ID	Description
CVE-2025-38500	In the Linux kernel, the following vulnerability has been resolved: xfrm: interface: fix use-after-free after changing collect_md xfrm interface collect_md property on xfrm interfaces can only be set on device creation, thus xfrm_i_changelink() should fail when called on such interfaces. The check to enforce this was done only in the case where the xi was returned from xfrm_i_locate() which doesn't look for the collect_md interface, and thus the validation was never reached. Calling changelink would thus erroneously place the special interface xi in the xfrm_i_net->xfrm_i_hash, but since it also exists in the xfrm_i_net->collect_md_xfrm_i pointer it would lead to a double free when the net namespace was taken down [1]. Change the check to use the xi from netdev_priv which is available earlier in the function to prevent changes in xfrm collect_md interfaces. [1] resulting oops: [8.516540] kernel BUG at net/core/dev.c:12029! [8.516552] Oops: invalid opcode: 0000 [#1] SMP NOPTI [8.516559] CPU: 0 UID: 0 PID: 12 Comm: kworker/u80:0 Not tainted 6.15.0-virtme #5 PREEMPT(voluntary) [8.516565] Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [8.516569] Workqueue: netns cleanup_net [8.516579] RIP: 0010:unregister_netdevice_many_notify+0x101/0xab0 [8.516590] Code: 90 0f 0b 90 48 8b b0 78 01 00 00 48 8b 90 80 01 00 00 48 89 56 08 48 89 32 4c 89 80 78 01 00 00 48 89 b8 80 01 00 00 eb ac 90 <0f> 0b 48 8b 45 00 4c 8d a0 88 fe ff ff 48 39 c5 74 5c 41 80 bc 24 [8.516593] RSP: 0018:ffffa93b8006bd30 EFLAGS: 00010206 [8.516598] RAX: ffff98fe4226e000 RBX: fffffa93b8006bd58 RCX: fffffa93b8006bc60 [8.516601] RDX: 0000000000000004 RSI: 0000000000000000 RDI: dead000000000122 [8.516603] RBP: fffffa93b8006bdd8 R08: dead000000000100 R09: ffff98fe4133c100 [8.516605] R10: 0000000000000000 R11: 000000000000003d2 R12: fffffa93b8006be00 [8.516608] R13: ffffffff96c1a510 R14: ffffffff96c1a510 R15: fffffa93b8006be00 [8.516615] FS: 0000000000000000(0000) GS:ffff98fee73b7000(0000) knlGS:0000000000000000 [8.516619] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [8.516622] CR2: 00007fcd2abd0700 CR3: 000000003aa40000 CR4: 0000000000752ef0 [8.516625] PKRU: 55555554 [8.516627] Call Trace: [8.516632] <TASK> [8.516635] ? rtnl_is_locked+0x15/0x20 [8.516641] ? unregister_netdevice_queue+0x29/0xf0 [8.516650] ops_undo_list+0x1f2/0x220 [8.516659] cleanup_net+0x1ad/0x2e0 [8.516664] process_one_work+0x160/0x380 [8.516673] worker_thread+0x2aa/0x3c0 [8.516679] ? __pfx_worker_thread+0x10/0x10 [8.516686] kthread+0xfb/0x200 [8.516690] ? __pfx_kthread+0x10/0x10 [8.516693] ? __pfx_kthread+0x10/0x10 [8.516697] ret_from_fork+0x82/0xf0 [8.516705] ? __pfx_kthread+0x10/0x10 [8.516709] ret_from_fork_asm+0x1a/0x30 [8.516718] </TASK>
CVE-2025-38501	In the Linux kernel, the following vulnerability has been resolved: ksmbd: limit repeated connections from clients with the same IP Repeated connections from clients with the same IP address may exhaust the max connections and prevent other normal client connections. This patch limit repeated connections from clients with the same IP.
CVE-2025-38502	In the Linux kernel, the following vulnerability has been resolved: bpf: Fix oob access in cgroup local storage Lonia reported that an out-of-bounds access in cgroup local storage can be crafted via tail calls. Given two programs each utilizing a cgroup local storage with a different value size, and one program doing a tail call into the other. The verifier will validate each of the individual programs just fine. However, in the runtime context the bpf_cg_run_ctx holds an bpf_prog_array_item which contains the BPF program as well as any cgroup local storage flavor the program uses. Helpers such as bpf_get_local_storage() pick this up from the runtime context: ctx = container_of(current->bpf_ctx, struct bpf_cg_run_ctx, run_ctx); storage = ctx->prog_item->cgroup_storage[stype]; if (stype == BPF_CGROUP_STORAGE_SHARED) ptr = &READ_ONCE(storage->buf)->data[0]; else ptr = this_cpu_ptr(storage->percpu_buf); For the second program which was called from the originally attached one, this means bpf_get_local_storage() will pick up the former program's map, not its own. With mismatching sizes, this can result in an unintended out-of-bounds access. To fix this issue, we need to extend bpf_map_owner with an array of storage_cookie[] to match on i) the exact maps from the original program if the second program was using bpf_get_local_storage(), or ii) allow the tail call combination if the second program was not using any of the cgroup local storage maps.

CVE ID	Description
CVE-2025-38503	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix assertion when building free space tree When building the free space tree with the block group tree feature enabled, we can hit an assertion failure like this: BTRFS info (device loop0 state M): rebuilding free space tree assertion failed: ret == 0, in fs/btrfs/free-space-tree.c:1102 -----[cut here]----- kernel BUG at fs/btrfs/free-space-tree.c:1102! Internal error: Oops - BUG: 0000000f2000800 [#1] SMP Modules linked in: CPU: 1 UID: 0 PID: 6592 Comm: syz-executor322 Not tainted 6.15.0-rc7-syzkaller-gd7fa1af5b33e #0 PREEMPT Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=) pc : populate_free_space_tree+0x514/0x518 fs/btrfs/free-space-tree.c:1102 lr : populate_free_space_tree+0x514/0x518 fs/btrfs/free-space-tree.c:1102 sp : ffff8000a4ce7600 x29: ffff8000a4ce76e0 x28: ffff0000c9bc6000 x27: ffff0000ddfff3d8 x26: ffff0000ddfff378 x25: dfff800000000000 x24: 0000000000000001 x23: ffff8000a4ce7660 x22: ffff70001499cecc x21: ffff0000e1d8c160 x20: ffff0000e1cb7800 x19: ffff0000e1d8c0b0 x18: 00000000ffffffff x17: ffff800092f39000 x16: ffff80008ad27e48 x15: ffff700011e740c0 x14: 1ffff00011e740c0 x13: 0000000000000004 x12: ffffffff00000000 x11: ffff700011e740c0 x10: 000000000ff0100 x9 : 94ef24f55d2dbc00 x8 : 94ef24f55d2dbc00 x7 : 0000000000000001 x6 : 0000000000000001 x5 : ffff8000a4ce6f98 x4 : ffff80008f415ba0 x3 : ffff800080548ef0 x2 : 0000000000000000 x1 : 0000000100000000 x0 : 000000000000003e Call trace: populate_free_space_tree+0x514/0x518 fs/btrfs/free-space-tree.c:1102 (P) btrfs_rebuild_free_space_tree+0x14c/0x54c fs/btrfs/free-space-tree.c:1337 btrfs_start_pre_rw_mount+0xa78/0xe10 fs/btrfs/disk-io.c:3074 btrfs_remount_rw fs/btrfs/super.c:1319 [inline] btrfs_reconfigure+0x828/0x2418 fs/btrfs/super.c:1543 reconfigure_super+0x1d4/0x6f0 fs/super.c:1083 do_remount fs/namespace.c:3365 [inline] path_mount+0xb34/0xde0 fs/namespace.c:4200 do_mount fs/namespace.c:4221 [inline] __do_sys_mount fs/namespace.c:4432 [inline] __se_sys_mount fs/namespace.c:4409 [inline] __arm64_sys_mount+0x3e8/0x468 fs/namespace.c:4409 __invoke_syscall arch/arm64/kernel/syscall.c:35 [inline] invoke_syscall+0x98/0x2b8 arch/arm64/kernel/syscall.c:49 el0_svc_common+0x130/0x23c arch/arm64/kernel/syscall.c:132 do_el0_svc+0x48/0x58 arch/arm64/kernel/syscall.c:151 el0_svc+0x58/0x17c arch/arm64/kernel/entry-common.c:767 el0t_64_sync_handler+0x78/0x108 arch/arm64/kernel/entry-common.c:786 el0t_64_sync+0x198/0x19c arch/arm64/kernel/entry.S:600 Code: f0047182 91178042 528089c3 9771d47b (d4210000) ---[end trace 0000000000000000]--- This happens because we are processing an empty block group, which has no extents allocated from it, there are no items for this block group, including the block group item since block group items are stored in a dedicated tree when using the block group tree feature. It also means this is the block group with the highest start offset, so there are no higher keys in the extent root, hence btrfs_search_slot_for_read() returns 1 (no higher key found). Fix this by asserting 'ret' is 0 only if the block group tree feature is not enabled, in which case we should find a block group item for the block group since it's stored in the extent root and block group item keys are greater than extent item keys (the value for BTRFS_BLOCK_GROUP_ITEM_KEY is 192 and for BTRFS_EXTENT_ITEM_KEY and BTRFS_METADATA_ITEM_KEY the values are 168 and 169 respectively). In case 'ret' is 1, we just need to add a record to the free space tree which spans the whole block group, and we can achieve this by making 'ret == 0' as the while loop's condition.

CVE ID	Description
CVE-2025-38506	In the Linux kernel, the following vulnerability has been resolved: KVM: Allow CPU to reschedule while setting per-page memory attributes When running an SEV-SNP guest with a sufficiently large amount of memory (1TB+), the host can experience CPU soft lockups when running an operation in <code>kvm_vm_set_mem_attributes()</code> to set memory attributes on the whole range of guest memory. watchdog: BUG: soft lockup - CPU#8 stuck for 26s! [qemu-kvm:6372] CPU: 8 UID: 0 PID: 6372 Comm: qemu-kvm Kdump: loaded Not tainted 6.15.0-rc7.20250520.el9uek.rc1.x86_64 #1 PREEMPT(voluntary) Hardware name: Oracle Corporation ORACLE SERVER E4-2c/Asm,MB Tray,2U,E4-2c, BIOS 78016600 11/13/2024 RIP: 0010:xas_create+0x78/0x1f0 Code: 00 00 00 41 80 fc 01 0f 84 82 00 00 00 ba 06 00 00 00 bd 06 00 00 00 49 8b 45 08 4d 8d 65 08 41 39 d6 73 20 83 ed 06 48 85 c0 <74> 67 48 89 c2 83 e2 03 48 83 fa 02 75 0c 48 3d 00 10 00 00 0f 87 RSP: 0018:ffffad890a34b940 EFLAGS: 00000286 RAX: ffff96f30b261daa RBX: ffffad890a34b9c8 RCX: 0000000000000000 RDX: 0000000000000001e RSI: 0000000000000000 RDI: 0000000000000000 RBP: 0000000000000018 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000000 R12: ffffad890a356868 R13: ffffad890a356860 R14: 0000000000000000 R15: ffffad890a356868 FS: 00007f5578a2a400(0000) GS:ffff97ed317e1000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f015c70fb18 CR3: 00000001109fd006 CR4: 000000000f70ef0 PKRU: 55555554 Call Trace: <TASK> xas_store+0x58/0x630 __xa_store+0xa5/0x130 xa_store+0x2c/0x50 kvm_vm_set_mem_attributes+0x343/0x710 [kvm] kvm_vm_ioctl+0x796/0xab0 [kvm] __x64_sys_ioctl+0xa3/0xd0 do_syscall_64+0x8c/0x7a0 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f5578d031bb Code: ff ff ff 85 c0 79 9b 49 c7 c4 ff ff ff ff 5b 5d 4c 89 e0 41 5c c3 66 0f 1f 84 00 00 00 00 00 f3 0f 1e fa b8 10 00 00 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d 2d 4c 0f 00 f7 d8 64 89 01 48 RSP: 002b:00007ffe0a742b88 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff00000000 RBX: 000000004020aed2 RCX: 00007f5578d031bb RDX: 00007ffe0a742c80 RSI: 000000004020aed2 RDI: 000000000000000b RBP: 0000010000000000 R08: 0000010000000000 R09: 0000017680000000 R10: 0000000000000080 R11: 0000000000000246 R12: 00005575e5f95120 R13: 00007ffe0a742c80 R14: 0000000000000008 R15: 00005575e5f961e0 While looping through the range of memory setting the attributes, call <code>cond_resched()</code> to give the scheduler a chance to run a higher priority task on the runqueue if necessary and avoid staying in kernel mode long enough to trigger the lockup.
CVE-2025-38507	In the Linux kernel, the following vulnerability has been resolved: HID: nintendo: avoid bluetooth suspend/resume stalls Ensure we don't stall or panic the kernel when using bluetooth-connected controllers. This was reported as an issue on android devices using kernel 6.6 due to the resume hook which had been added for usb joycons. First, set a new state value to <code>JOYCON_CTLR_STATE_SUSPENDED</code> in a newly-added <code>nintendo_hid_suspend</code> . This makes sure we will not stall out the kernel waiting for input reports during led classdev suspend. The stalls could happen if connectivity is unreliable or lost to the controller prior to suspend. Second, since we lose connectivity during suspend, do not try <code>joycon_init()</code> for bluetooth controllers in the <code>nintendo_hid_resume</code> path. Tested via multiple suspend/resume flows when using the controller both in USB and bluetooth modes.
CVE-2025-38510	In the Linux kernel, the following vulnerability has been resolved: kasan: remove <code>kasan_find_vm_area()</code> to prevent possible deadlock <code>find_vm_area()</code> couldn't be called in atomic_context. If <code>find_vm_area()</code> is called to reports vm area information, kasan can trigger deadlock like: CPU0 CPU1 <code>vmalloc(); alloc_vmap_area(); spin_lock(&vn->busy.lock) spin_lock_bh(&some_lock); <interrupt occurs> <in softirq> spin_lock(&some_lock); <access invalid address> kasan_report(); print_report(); print_address_description(); kasan_find_vm_area(); find_vm_area(); spin_lock(&vn->busy.lock) // deadlock! To prevent possible deadlock while kasan reports, remove <code>kasan_find_vm_area()</code>.</code>
CVE-2025-38511	In the Linux kernel, the following vulnerability has been resolved: drm/xe/pf: Clear all LMTT pages on alloc Our LMEM buffer objects are not cleared by default on alloc and during VF provisioning we only setup LMTT PTEs for the actually provisioned LMEM range. But beyond that valid range we might leave some stale data that could either point to some other VFs allocations or even to the PF pages. Explicitly clear all new LMTT page to avoid the risk that a malicious VF would try to exploit that gap. While around add asserts to catch any undesired PTE overwrites and low-level debug traces to track LMTT PT life-cycle. (cherry picked from commit 3fae6918a3e27cce20ded2551f863fb05d4bef8d)

CVE ID	Description
CVE-2025-38512	In the Linux kernel, the following vulnerability has been resolved: wifi: prevent A-MSDU attacks in mesh networks This patch is a mitigation to prevent the A-MSDU spoofing vulnerability for mesh networks. The initial update to the IEEE 802.11 standard, in response to the FragAttacks, missed this case (CVE-2025-27558). It can be considered a variant of CVE-2020-24588 but for mesh networks. This patch tries to detect if a standard MSDU was turned into an A-MSDU by an adversary. This is done by parsing a received A-MSDU as a standard MSDU, calculating the length of the Mesh Control header, and seeing if the 6 bytes after this header equal the start of an rfc1042 header. If equal, this is a strong indication of an ongoing attack attempt. This defense was tested with mac80211_hwsim against a mesh network that uses an empty Mesh Address Extension field, i.e., when four addresses are used, and when using a 12-byte Mesh Address Extension field, i.e., when six addresses are used. Functionality of normal MSDUs and A-MSDUs was also tested, and confirmed working, when using both an empty and 12-byte Mesh Address Extension field. It was also tested with mac80211_hwsim that A-MSDU attacks in non-mesh networks keep being detected and prevented. Note that the vulnerability being patched, and the defense being implemented, was also discussed in the following paper and in the following IEEE 802.11 presentation: https://papers.mathyvanhoef.com/wisec2025.pdf https://mentor.ieee.org/802.11/dcn/25/11-25-0949-00-000m-a-msdu-mesh-spoof-protection.docx
CVE-2025-38513	In the Linux kernel, the following vulnerability has been resolved: wifi: zd1211rw: Fix potential NULL pointer dereference in zd_mac_tx_to_dev() There is a potential NULL pointer dereference in zd_mac_tx_to_dev(). For example, the following is possible: T0 T1 zd_mac_tx_to_dev() / * len == skb_queue_len(q) */ while (len > ZD_MAC_MAX_ACK_WAITERS) { filter_ack() spin_lock_irqsave(&q->lock, flags); /* position == skb_queue_len(q) */ for (i=1; i<position; i++) skb = __skb_dequeue(q) if (mac->type == NL80211_IFTYPE_AP) skb = __skb_dequeue(q); spin_unlock_irqrestore(&q->lock, flags); skb_dequeue() -> NULL Since there is a small gap between checking skb queue length and skb being unconditionally dequeued in zd_mac_tx_to_dev(), skb_dequeue() can return NULL. Then the pointer is passed to zd_mac_tx_status() where it is dereferenced. In order to avoid potential NULL pointer dereference due to situations like above, check if skb is not NULL before passing it to zd_mac_tx_status(). Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-38514	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix oops due to non-existence of prealloc backlog struct If an AF_RXRPC service socket is opened and bound, but calls are preallocated, then rxrpc_alloc_incoming_call() will oops because the rxrpc_backlog struct doesn't get allocated until the first preallocation is made. Fix this by returning NULL from rxrpc_alloc_incoming_call() if there is no backlog struct. This will cause the incoming call to be aborted.
CVE-2025-38515	In the Linux kernel, the following vulnerability has been resolved: drm/sched: Increment job count before swapping tail spsc queue A small race exists between spsc_queue_push and the run-job worker, in which spsc_queue_push may return not-first while the run-job worker has already idled due to the job count being zero. If this race occurs, job scheduling stops, leading to hangs while waiting on the job's DMA fences. Seal this race by incrementing the job count before appending to the SPSC queue. This race was observed on a drm-tip 6.16-rc1 build with the Xe driver in an SVM test case.
CVE-2025-38516	In the Linux kernel, the following vulnerability has been resolved: pinctrl: qcom: msm: mark certain pins as invalid for interrupts On some platforms, the UFS-reset pin has no interrupt logic in TLMM but is nevertheless registered as a GPIO in the kernel. This enables the user-space to trigger a BUG() in the pinctrl-msm driver by running, for example: `gpiomon -c 0 113` on RB2. The exact culprit is requesting pins whose intr_detection_width setting is not 1 or 2 for interrupts. This hits a BUG() in msm_gpio_irq_set_type(). Potentially crashing the kernel due to an invalid request from user-space is not optimal, so let's go through the pins and mark those that would fail the check as invalid for the irq chip as we should not even register them as available irqs. This function can be extended if we determine that there are more corner-cases like this.

CVE ID	Description
CVE-2025-38520	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Don't call mmpu from MMU notifier callback If the process is exiting, the mmpu inside mmu notifier callback from compactd or fork or numa balancing could release the last reference of mm struct to call exit_mmap and free_pgtable, this triggers deadlock with below backtrace. The deadlock will leak kfd process as mmu notifier release is not called and cause VRAM leaking. The fix is to take mm reference mmget_non_zero when adding prange to the deferred list to pair with mmpu in deferred list work. If prange split and add into pchild list, the pchild work_item.mm is not used, so remove the mm parameter from svm_range_unmap_split and svm_range_add_child. The backtrace of hung task: INFO: task python:348105 blocked for more than 64512 seconds. Call Trace: __schedule+0x1c3/0x550 schedule+0x46/0xb0 rwsem_down_write_slowpath+0x24b/0x4c0 unlink_anon_vmas+0xb1/0x1c0 free_pgtables+0xa9/0x130 exit_mmap+0xbc/0x1a0 mmpu+0x5a/0x140 svm_range_cpu_invalidate_pagetables+0x2b/0x40 [amdGPU] mn_itree_invalidate+0x72/0xc0 __mmu_notifier_invalidate_range_start+0x48/0x60 try_to_unmap_one+0x10fa/0x1400 rmap_walk_anon+0x196/0x460 try_to_unmap+0xbb/0x210 migrate_page_unmap+0x54d/0x7e0 migrate_pages_batch+0x1c3/0xae0 migrate_pages_sync+0x98/0x240 migrate_pages+0x25c/0x520 compact_zone+0x29d/0x590 compact_zone_order+0xb6/0xf0 try_to_compact_pages+0xbe/0x220 __alloc_pages_direct_compact+0x96/0x1a0 __alloc_pages_slowpath+0x410/0x930 __alloc_pages_nodemask+0x3a9/0x3e0 do_huge_pmd_anonymous_page+0xd7/0x3e0 __handle_mm_fault+0x5e3/0x5f0 handle_mm_fault+0xf7/0x2e0 hmm_vma_fault.isra.0+0x4d/0xa0 walk_pmd_range.isra.0+0xa8/0x310 walk_pud_range+0x167/0x240 walk_pgd_range+0x55/0x100 __walk_page_range+0x87/0x90 walk_page_range+0xf6/0x160 hmm_range_fault+0x4f/0x90 amdGPU_hmm_range_get_pages+0x123/0x230 [amdGPU] amdGPU_ttm_tt_get_user_pages+0xb1/0x150 [amdGPU] init_user_pages+0xb1/0x2a0 [amdGPU] amdGPU_amdkfd_gpuvm_alloc_memory_of_gpu+0x543/0x7d0 [amdGPU] kfd_ioctl_alloc_memory_of_gpu+0x24c/0x4e0 [amdGPU] kfd_ioctl+0x29d/0x500 [amdGPU] (cherry picked from commit a29e067bd38946f752b0ef855f3dfff87e77bec7)
CVE-2025-38521	In the Linux kernel, the following vulnerability has been resolved: drm/imagination: Fix kernel crash when hard resetting the GPU The GPU hard reset sequence calls pm_runtime_force_suspend() and pm_runtime_force_resume(), which according to their documentation should only be used during system-wide PM transitions to sleep states. The main issue though is that depending on some internal runtime PM state as seen by pm_runtime_force_suspend() (whether the usage count is <= 1), pm_runtime_force_resume() might not resume the device unless needed. If that happens, the runtime PM resume callback pvr_power_device_resume() is not called, the GPU clocks are not re-enabled, and the kernel crashes on the next attempt to access GPU registers as part of the power-on sequence. Replace calls to pm_runtime_force_suspend() and pm_runtime_force_resume() with direct calls to the driver's runtime PM callbacks, pvr_power_device_suspend() and pvr_power_device_resume(), to ensure clocks are re-enabled and avoid the kernel crash.
CVE-2025-38524	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix recv-recv race of completed call If a call receives an event (such as incoming data), the call gets placed on the socket's queue and a thread in recvmmsg can be awakened to go and process it. Once the thread has picked up the call off of the queue, further events will cause it to be requeued, and once the socket lock is dropped (recvmmsg uses call->user_mutex to allow the socket to be used in parallel), a second thread can come in and its recvmmsg can pop the call off the socket queue again. In such a case, the first thread will be receiving stuff from the call and the second thread will be blocked on call->user_mutex. The first thread can, at this point, process both the event that it picked call for and the event that the second thread picked the call for and may see the call terminate - in which case the call will be "released", decoupling the call from the user call ID assigned to it (RXRPC_USER_CALL_ID in the control message). The first thread will return okay, but then the second thread will wake up holding the user_mutex and, if it sees that the call has been released by the first thread, it will BUG thusly: kernel BUG at net/rxrpc/recvmmsg.c:474! Fix this by just dequeuing the call and ignoring it if it is seen to be already released. We can't tell userspace about it anyway as the user call ID has become stale.
CVE-2025-38526	In the Linux kernel, the following vulnerability has been resolved: ice: add NULL check in eswitch lag check The function ice_lag_is_switchdev_running() is being called from outside of the LAG event handler code. This results in the lag->upper_netdev being NULL sometimes. To avoid a NULL-pointer dereference, there needs to be a check before it is dereferenced.

CVE ID	Description
CVE-2025-38527	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix use-after-free in cifs_oplock_break A race condition can occur in cifs_oplock_break() leading to a use-after-free of the cinode structure when unmounting: cifs_oplock_break() _cifsFileInfo_put(cfile) cifsFileInfo_put_final() cifs_sb_deactive() [last ref, start releasing sb] kill_sb() kill_anon_super() generic_shutdown_super() evict_inodes() dispose_list() evict() destroy_inode() call_rcu(&inode->i_rcu, i_callback) spin_lock(&cinode->open_file_lock) <- OK [later] i_callback() cifs_free_inode() kmem_cache_free(cinode) spin_unlock(&cinode->open_file_lock) <- UAF cifs_done_oplock_break(cinode) <- UAF The issue occurs when umount has already released its reference to the superblock. When _cifsFileInfo_put() calls cifs_sb_deactive(), this releases the last reference, triggering the immediate cleanup of all inodes under RCU. However, cifs_oplock_break() continues to access the cinode after this point, resulting in use-after-free. Fix this by holding an extra reference to the superblock during the entire oplock break operation. This ensures that the superblock and its inodes remain valid until the oplock break completes.
CVE-2025-38528	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject %p% format string in bprintf-like helpers static const char fmt[] = "%p%"; bpf_trace_printk(fmt, sizeof(fmt)); The above BPF program isn't rejected and causes a kernel warning at runtime: Please remove unsupported %\x00 in format string WARNING: CPU: 1 PID: 7244 at lib/vsprintf.c:2680 format_decode+0x49c/0x5d0 This happens because bpf_bprintf_prepare skips over the second %, detected as punctuation, while processing %p. This patch fixes it by not skipping over punctuation. %\x00 is then processed in the next iteration and rejected.
CVE-2025-38529	In the Linux kernel, the following vulnerability has been resolved: comedi: aio_iio_16: Fix bit shift out of bounds When checking for a supported IRQ number, the following test is used: if ((1 << it->options[1]) & 0xdcfc) { However, `it->options[i]` is an unchecked `int` value from userspace, so the shift amount could be negative or out of bounds. Fix the test by requiring `it->options[1]` to be within bounds before proceeding with the original test. Valid `it->options[1]` values that select the IRQ will be in the range [1,15]. The value 0 explicitly disables the use of interrupts.
CVE-2025-38530	In the Linux kernel, the following vulnerability has been resolved: comedi: pcl812: Fix bit shift out of bounds When checking for a supported IRQ number, the following test is used: if ((1 << it->options[1]) & board->irq_bits) { However, `it->options[i]` is an unchecked `int` value from userspace, so the shift amount could be negative or out of bounds. Fix the test by requiring `it->options[1]` to be within bounds before proceeding with the original test. Valid `it->options[1]` values that select the IRQ will be in the range [1,15]. The value 0 explicitly disables the use of interrupts.
CVE-2025-38531	In the Linux kernel, the following vulnerability has been resolved: iio: common: st_sensors: Fix use of uninitialized device structs Throughout the various probe functions &indio_dev->dev is used before it is initialized. This caused a kernel panic in st_sensors_power_enable() when the call to devm_regulator_bulk_get_enable() fails and then calls dev_err_probe() with the uninitialized device. This seems to only cause a panic with dev_err_probe(), dev_err(), dev_warn() and dev_info() don't seem to cause a panic, but are fixed as well. The issue is reported and traced here: [1]

CVE ID	Description
CVE-2025-38532	In the Linux kernel, the following vulnerability has been resolved: net: libwx: properly reset Rx ring descriptor When device reset is triggered by feature changes such as toggling Rx VLAN offload, wx->do_reset() is called to reinitialize Rx rings. The hardware descriptor ring may retain stale values from previous sessions. And only set the length to 0 in rx_desc[0] would result in building malformed SKBs. Fix it to ensure a clean slate after device reset. [549.186435] [C16] -----[cut here]----- [549.186457] [C16] kernel BUG at net/core/skbuff.c:2814! [549.186468] [C16] Oops: invalid opcode: 0000 [#1] SMP NOPTI [549.186472] [C16] CPU: 16 UID: 0 PID: 0 Comm: swapper/16 Kdump: loaded Not tainted 6.16.0-rc4+ #23 PREEMPT(voluntary) [549.186476] [C16] Hardware name: Micro-Star International Co., Ltd. MS-7E16/X670E GAMING PLUS WIFI (MS-7E16), BIOS 1.90 12/31/2024 [549.186478] [C16] RIP: 0010:__pskb_pull_tail+0x3ff/0x510 [549.186484] [C16] Code: 06 f0 ff 4f 34 74 7b 4d 8b 8c 24 c8 00 00 00 45 8b 84 24 c0 00 00 00 e9 c8 fd ff ff 48 c7 44 24 08 00 00 00 e9 5e fe ff ff <0f> 0b 31 c0 e9 23 90 5b ff 41 f7 c6 ff 0f 00 00 75 bf 49 8b 06 a8 [549.186487] [C16] RSP: 0018:ffffb391c0640d70 EFLAGS: 00010282 [549.186490] [C16] RAX: 00000000ffffffff2 RBX: ffff8fe7e4d40200 RCX: 00000000ffffffff2 [549.186492] [C16] RDX: ffff8fe7c3a4bf8e RSI: 0000000000000180 RDI: ffff8fe7c3a4bf40 [549.186494] [C16] RBP: ffff8b391c0640da8 R08: ffff8fe7c3a4c0c0 R09: 000000000000000e [549.186496] [C16] R10: ffff8b391c0640d88 R11: 000000000000000e R12: ffff8fe7e4d40200 [549.186497] [C16] R13: 00000000ffffffff2 R14: ffff8fe7fa01a000 R15: 00000000ffffffff2 [549.186499] [C16] FS: 0000000000000000(0000) GS: ffff8fef5ae40000(0000) knlGS:0000000000000000 [549.186502] [C16] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [549.186503] [C16] CR2: 00007f77d81d6000 CR3: 0000000051a03200 CR4: 0000000000750ef0 [549.186505] [C16] PKRU: 55555554 [549.186507] [C16] Call Trace: [549.186510] [C16] <IRQ> [549.186513] [C16] ? srso_alias_return_thunk+0x5/0xfbf5 [549.186517] [C16] __skb_pad+0xc7/0xf0 [549.186523] [C16] wx_clean_rx_irq+0x355/0x3b0 [libwx] [549.186533] [C16] wx_poll+0x92/0x120 [libwx] [549.186540] [C16] __napi_poll+0x28/0x190 [549.186544] [C16] net_rx_action+0x301/0x3f0 [549.186548] [C16] ? srso_alias_return_thunk+0x5/0xfbf5 [549.186551] [C16] ? __raw_spin_lock_irqsave+0x1e/0x50 [549.186554] [C16] ? srso_alias_return_thunk+0x5/0xfbf5 [549.186557] [C16] ? wake_up_nohz_cpu+0x35/0x160 [549.186559] [C16] ? srso_alias_return_thunk+0x5/0xfbf5 [549.186563] [C16] handle_softirqs+0xf9/0x2c0 [549.186568] [C16] __irq_exit_rcu+0xc7/0x130 [549.186572] [C16] common_interrupt+0xb8/0xd0 [549.186576] [C16] </IRQ> [549.186577] [C16] <TASK> [549.186579] [C16] asm_common_interrupt+0x22/0x40 [549.186582] [C16] RIP: 0010:cpuidle_enter_state+0xc2/0x420 [549.186585] [C16] Code: 00 00 e8 11 0e 5e ff e8 ac f0 ff ff 49 89 c5 0f 1f 44 00 00 31 ff e8 0d ed 5c ff 45 84 ff 0f 85 40 02 00 00 fb 0f 1f 44 00 00 <45> 85 f6 0f 88 84 01 00 00 49 63 d6 48 8d 04 52 48 8d 04 82 49 8d [549.186587] [C16] RSP: 0018:ffffb391c0277e78 EFLAGS: 00000246 [549.186590] [C16] RAX: ffff8fef5ae40000 RBX: 0000000000000003 RCX: 0000000000000000 [549.186591] [C16] RDX: 0000007fde0faac5 RSI: ffffffff826e53f6 RDI: ffffffff826fa9b3 [549.186593] [C16] RBP: ffff8fe7c3a20800 R08: 0000000000000002 R09: 0000000000000000 [549.186595] [C16] R10: 0000000000000000 R11: 000000000000ffff R12: ffffffff82ed7a40 [549.186596] [C16] R13: 0000007fde0faac5 R14: 0000000000000003 R15: 0000000000000000 [549.186601] [C16] ? cpuidle_enter_state+0xb3/0x420 [549.186605] [C16] cpuidle_en ---truncated---
CVE-2025-38533	In the Linux kernel, the following vulnerability has been resolved: net: libwx: fix the using of Rx buffer DMA The wx_rx_buffer structure contained two DMA address fields: 'dma' and 'page_dma'. However, only 'page_dma' was actually initialized and used to program the Rx descriptor. But 'dma' was uninitialized and used in some paths. This could lead to undefined behavior, including DMA errors or use-after-free, if the uninitialized 'dma' was used. Although such error has not yet occurred, it is worth fixing in the code.
CVE-2025-38535	In the Linux kernel, the following vulnerability has been resolved: phy: tegra: xusb: Fix unbalanced regulator disable in UTMI PHY mode When transitioning from USB_ROLE_DEVICE to USB_ROLE_NONE, the code assumed that the regulator should be disabled. However, if the regulator is marked as always-on, regulator_is_enabled() continues to return true, leading to an incorrect attempt to disable a regulator which is not enabled. This can result in warnings such as: [250.155624] WARNING: CPU: 1 PID: 7326 at drivers/regulator/core.c:3004 _regulator_disable+0xe4/0x1a0 [250.155652] unbalanced disables for VIN_SYS_5V0 To fix this, we move the regulator control logic into tegra186_xusb_padctl_id_override() function since it's directly related to the ID override state. The regulator is now only disabled when the role transitions from USB_ROLE_HOST to USB_ROLE_NONE, by checking the VBUS_ID register. This ensures that regulator enable/disable operations are properly balanced and only occur when actually transitioning to/from host mode.

CVE ID	Description
CVE-2025-38537	In the Linux kernel, the following vulnerability has been resolved: net: phy: Don't register LEDs for genphy If a PHY has no driver, the genphy driver is probed/removed directly in phy_attach/detach. If the PHY's ofnode has an "leds" subnode, then the LEDs will be (un)registered when probing/removing the genphy driver. This could occur if the leds are for a non-generic driver that isn't loaded for whatever reason. Synchronously removing the PHY device in phy_detach leads to the following deadlock: rtnl_lock() ndo_close() ... phy_detach() phy_remove() phy_leds_unregister() led_classdev_unregister() led_trigger_set() netdev_trigger_deactivate() unregister_netdevice_notifier() rtnl_lock() There is a corresponding deadlock on the open/register side of things (and that one is reported by lockdep), but it requires a race while this one is deterministic. Generic PHYs do not support LEDs anyway, so don't bother registering them.
CVE-2025-38538	In the Linux kernel, the following vulnerability has been resolved: dmaengine: nbpfaxi: Fix memory corruption in probe() The nbpf->chan[] array is allocated earlier in the nbpf_probe() function and it has "num_channels" elements. These three loops iterate one element farther than they should and corrupt memory. The changes to the second loop are more involved. In this case, we're copying data from the irqbuf[] array into the nbpf->chan[] array. If the data in irqbuf[i] is the error IRQ then we skip it, so the iterators are not in sync. I added a check to ensure that we don't go beyond the end of the irqbuf[] array. I'm pretty sure this can't happen, but it seemed harmless to add a check. On the other hand, after the loop has ended there is a check to ensure that the "chan" iterator is where we expect it to be. In the original code we went one element beyond the end of the array so the iterator wasn't in the correct place and it would always return -EINVAL. However, now it will always be in the correct place. I deleted the check since we know the result.
CVE-2025-38539	In the Linux kernel, the following vulnerability has been resolved: tracing: Add down_write(trace_event_sem) when adding trace event When a module is loaded, it adds trace events defined by the module. It may also need to modify the modules trace printk formats to replace enum names with their values. If two modules are loaded at the same time, the adding of the event to the ftrace_events list can corrupt the walking of the list in the code that is modifying the printk format strings and crash the kernel. The addition of the event should take the trace_event_sem for write while it adds the new event. Also add a lockdep_assert_held() on that semaphore in __trace_add_event_dirs() as it iterates the list.
CVE-2025-38540	In the Linux kernel, the following vulnerability has been resolved: HID: quirks: Add quirk for 2 Chicony Electronics HP 5MP Cameras The Chicony Electronics HP 5MP Cameras (USB ID 04F2:B824 & 04F2:B82C) report a HID sensor interface that is not actually implemented. Attempting to access this non-functional sensor via iio_info causes system hangs as runtime PM tries to wake up an unresponsive sensor. Add these 2 devices to the HID ignore list since the sensor interface is non-functional by design and should not be exposed to userspace.
CVE-2025-38542	In the Linux kernel, the following vulnerability has been resolved: net: appletalk: Fix device refcount leak in attr_create() When updating an existing route entry in attr_create(), the old device reference was not being released before assigning the new device, leading to a device refcount leak. Fix this by calling dev_put() to release the old device reference before holding the new one.
CVE-2025-38543	In the Linux kernel, the following vulnerability has been resolved: drm/tegra: nvdec: Fix dma_alloc_coherent error check Check for NULL return value with dma_alloc_coherent, in line with Robin's fix for vic.c in 'drm/tegra: vic: Fix DMA API misuse'.
CVE-2025-38544	In the Linux kernel, the following vulnerability has been resolved: rxrpc: Fix bug due to prealloc collision When userspace is using AF_RXRPC to provide a server, it has to preallocate incoming calls and assign to them call IDs that will be used to thread related recvmsg() and sendmsg() together. The preallocated call IDs will automatically be attached to calls as they come in until the pool is empty. To the kernel, the call IDs are just arbitrary numbers, but userspace can use the call ID to hold a pointer to prepared structs. In any case, the user isn't permitted to create two calls with the same call ID (call IDs become available again when the call ends) and EBADSLT should result from sendmsg() if an attempt is made to preallocate a call with an in-use call ID. However, the cleanup in the error handling will trigger both assertions in rxrpc_cleanup_call() because the call isn't marked complete and isn't marked as having been released. Fix this by setting the call state in rxrpc_service_prealloc_one() and then marking it as being released before calling the cleanup function.
CVE-2025-38546	In the Linux kernel, the following vulnerability has been resolved: atm: clip: Fix memory leak of struct clip_vcc. ioctl(ATMARP_MKIP) allocates struct clip_vcc and set it to vcc->user_back. The code assumes that vcc_destroy_socket() passes NULL skb to vcc->push() when the socket is close(d), and then clip_push() frees clip_vcc. However, ioctl(ATMARPD_CTRL) sets NULL to vcc->push() in atm_init_atmarp(), resulting in memory leak. Let's serialise two ioctl() by lock_sock() and check vcc->push() in atm_init_atmarp() to prevent memleak.
CVE-2025-38548	In the Linux kernel, the following vulnerability has been resolved: hwmon: (corsair-cpro) Validate the size of the received input buffer Add buffer_rcv_size to store the size of the received bytes. Validate buffer_rcv_size in send_usb_cmd().

CVE ID	Description
CVE-2025-38549	In the Linux kernel, the following vulnerability has been resolved: efivarfs: Fix memory leak of efivarfs_fs_info in fs_context error paths When processing mount options, efivarfs allocates efivarfs_fs_info (sfi) early in fs_context initialization. However, sfi is associated with the superblock and typically freed when the superblock is destroyed. If the fs_context is released (final put) before fill_super is called—such as on error paths or during reconfiguration—the sfi structure would leak, as ownership never transfers to the superblock. Implement the .free callback in efivarfs_context_ops to ensure any allocated sfi is properly freed if the fs_context is torn down before fill_super, preventing this memory leak.
CVE-2025-38550	In the Linux kernel, the following vulnerability has been resolved: ipv6: mcast: Delay put pmc->idev in mld_del_delrec() pmc->idev is still used in ip6_mc_clear_src(), so as mld_clear_delrec() does, the reference should be put after ip6_mc_clear_src() return.
CVE-2025-38551	In the Linux kernel, the following vulnerability has been resolved: virtio-net: fix recursived rtnl_lock() during probe() The deadlock appears in a stack trace like: virtnet_probe() rtnl_lock() virtio_config_changed_work() netdev_notify_peers() rtnl_lock() It happens if the VMM sends a VIRTIO_NET_S_ANNOUNCE request while the virtio-net driver is still probing. The config_work in probe() will get scheduled until virtnet_open() enables the config change notification via virtio_config_driver_enable().
CVE-2025-38552	In the Linux kernel, the following vulnerability has been resolved: mptcp: plug races between subflow fail and subflow creation We have races similar to the one addressed by the previous patch between subflow failing and additional subflow creation. They are just harder to trigger. The solution is similar. Use a separate flag to track the condition 'socket state prevent any additional subflow creation' protected by the fallback lock. The socket fallback makes such flag true, and also receiving or sending an MP_FAIL option. The field 'allow_infinite_fallback' is now always touched under the relevant lock, we can drop the ONCE annotation on write.
CVE-2025-38553	In the Linux kernel, the following vulnerability has been resolved: net/sched: Restrict conditions for adding duplicating netems to qdisc tree netem_enqueue's duplication prevention logic breaks when a netem resides in a qdisc tree with other netems - this can lead to a soft lockup and OOM loop in netem_dequeue, as seen in [1]. Ensure that a duplicating netem cannot exist in a tree with other netems. Previous approaches suggested in discussions in chronological order: 1) Track duplication status or ttl in the sk_buff struct. Considered too specific a use case to extend such a struct, though this would be a resilient fix and address other previous and potential future DOS bugs like the one described in loopy fun [2]. 2) Restrict netem_enqueue recursion depth like in act_mirred with a per cpu variable. However, netem_dequeue can call enqueue on its child, and the depth restriction could be bypassed if the child is a netem. 3) Use the same approach as in 2, but add metadata in netem_skb_cb to handle the netem_dequeue case and track a packet's involvement in duplication. This is an overly complex approach, and Jamal notes that the skb cb can be overwritten to circumvent this safeguard. 4) Prevent the addition of a netem to a qdisc tree if its ancestral path contains a netem. However, filters and actions can cause a packet to change paths when re-enqueued to the root from netem duplication, leading us to the current solution: prevent a duplicating netem from inhabiting the same tree as other netems. [1] https://lore.kernel.org/netdev/8DuRWwfqjoRDLDmBMIIfbZg9Gx50DHJc1ilxsEBNe2D6NMoigr_eIRIG0LOJmC3r10nUUZtArXx4oZBIdUfZ [2] https://lwn.net/Articles/719297/
CVE-2025-38555	In the Linux kernel, the following vulnerability has been resolved: usb: gadget : fix use-after-free in composite_dev_cleanup() 1. In func configfs_composite_bind() -> composite_os_desc_req_prepare(): if kmalloc fails, the pointer cdev->os_desc_req will be freed but not set to NULL. Then it will return a failure to the upper-level function. 2. in func configfs_composite_bind() -> composite_dev_cleanup(): it will checks whether cdev->os_desc_req is NULL. If it is not NULL, it will attempt to use it. This will lead to a use-after-free issue. BUG: KASAN: use-after-free in composite_dev_cleanup+0xf4/0x2c0 Read of size 8 at addr 0000004827837a00 by task init/1 CPU: 10 PID: 1 Comm: init Tainted: G O 5.10.97-oh #1 kasan_report+0x188/0x1cc __asan_load8+0xb4/0xbc composite_dev_cleanup+0xf4/0x2c0 configfs_composite_bind+0x210/0x7ac udc_bind_to_driver+0xb4/0x1ec usb_gadget_probe_driver+0xec/0x21c gadget_dev_desc_UDC_store+0x264/0x27c
CVE-2025-38556	In the Linux kernel, the following vulnerability has been resolved: HID: core: Harden s32ton() against conversion to 0 bits Testing by the syzbot fuzzer showed that the HID core gets a shift-out-of-bounds exception when it tries to convert a 32-bit quantity to a 0-bit quantity. Ideally this should never occur, but there are buggy devices and some might have a report field with size set to zero; we shouldn't reject the report or the device just because of that. Instead, harden the s32ton() routine so that it returns a reasonable result instead of crashing when it is called with the number of bits set to 0 -- the same as what snto32() does.

CVE ID	Description
CVE-2025-38560	In the Linux kernel, the following vulnerability has been resolved: x86/sev: Evict cache lines during SNP memory validation An SNP cache coherency vulnerability requires a cache line eviction mitigation when validating memory after a page state change to private. The specific mitigation is to touch the first and last byte of each 4K page that is being validated. There is no need to perform the mitigation when performing a page state change to shared and rescinding validation. CPUID bit Fn8000001F_EBX[31] defines the COHERENCY_SFW_NO CPUID bit that, when set, indicates that the software mitigation for this vulnerability is not needed. Implement the mitigation and invoke it when validating memory (making it private) and the COHERENCY_SFW_NO bit is not set, indicating the SNP guest is vulnerable.
CVE-2025-38561	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix Preauth_HashValue race condition If client send multiple session setup requests to ksmbd, Preauth_HashValue race condition could happen. There is no need to free sess->Preauth_HashValue at session setup phase. It can be freed together with session at connection termination phase.
CVE-2025-38562	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix null pointer dereference error in generate_encryptionkey If client send two session setups with krb5 authenticate to ksmbd, null pointer dereference error in generate_encryptionkey could happen. sess->Preauth_HashValue is set to NULL if session is valid. So this patch skip generate encryption key if session is valid.
CVE-2025-38563	In the Linux kernel, the following vulnerability has been resolved: perf/core: Prevent VMA split of buffer mappings The perf mmap code is careful about mmap()'ing the user page with the ringbuffer and additionally the auxiliary buffer, when the event supports it. Once the first mapping is established, subsequent mapping have to use the same offset and the same size in both cases. The reference counting for the ringbuffer and the auxiliary buffer depends on this being correct. Though perf does not prevent that a related mapping is split via mmap(2), munmap(2) or mremap(2). A split of a VMA results in perf_mmap_open() calls, which take reference counts, but then the subsequent perf_mmap_close() calls are not longer fulfilling the offset and size checks. This leads to reference count leaks. As perf already has the requirement for subsequent mappings to match the initial mapping, the obvious consequence is that VMA splits, caused by resizing of a mapping or partial unmapping, have to be prevented. Implement the vm_operations_struct::may_split() callback and return unconditionally -EINVAL. That ensures that the mapping offsets and sizes cannot be changed after the fact. Remapping to a different fixed address with the same size is still possible as it takes the references for the new mapping and drops those of the old mapping.
CVE-2025-38565	In the Linux kernel, the following vulnerability has been resolved: perf/core: Exit early on perf_mmap() fail When perf_mmap() fails to allocate a buffer, it still invokes the event_mapped() callback of the related event. On X86 this might increase the perf_rdpmc_allowed reference counter. But nothing undoes this as perf_mmap_close() is never called in this case, which causes another reference count leak. Return early on failure to prevent that.
CVE-2025-38566	In the Linux kernel, the following vulnerability has been resolved: sunrpc: fix handling of server side tls alerts Scott Mayhew discovered a security exploit in NFS over TLS in tls_alert_recv() due to its assumption it can read data from the msg iterator's kvec.. kTLS implementation splits TLS non-data record payload between the control message buffer (which includes the type such as TLS alert or TLS cipher change) and the rest of the payload (say TLS alert's level/description) which goes into the msg payload buffer. This patch proposes to rework how control messages are setup and used by sock_recvmsg(). If no control message structure is setup, kTLS layer will read and process TLS data record types. As soon as it encounters a TLS control message, it would return an error. At that point, NFS can setup a kvec backed msg buffer and read in the control message such as a TLS alert. Msg iterator can advance the kvec pointer as a part of the copy process thus we need to revert the iterator before calling into the tls_alert_recv.
CVE-2025-38568	In the Linux kernel, the following vulnerability has been resolved: net/sched: mqprio: fix stack out-of-bounds write in tc entry parsing TCA_MQPRIO_TC_ENTRY_INDEX is validated using NLA_POLICY_MAX(NLA_U32, TC_QOPT_MAX_QUEUE), which allows the value TC_QOPT_MAX_QUEUE (16). This leads to a 4-byte out-of-bounds stack write in the fp[] array, which only has room for 16 elements (0-15). Fix this by changing the policy to allow only up to TC_QOPT_MAX_QUEUE - 1.
CVE-2025-38569	In the Linux kernel, the following vulnerability has been resolved: benet: fix BUG when creating VFs benet crashes as soon as SRIOV VFs are created: kernel BUG at mm/vmalloc.c:3457! Oops: invalid opcode: 0000 [#1] SMP KASAN NOPTI CPU: 4 UID: 0 PID: 7408 Comm: test.sh Kdump: loaded Not tainted 6.16.0+ #1 PREEMPT(voluntary) [...] RIP: 0010:vunmap+0x5f/0x70 [...] Call Trace: <TASK> __iommu_dma_free+0xe8/0x1c0 be_cmd_set_mac_list+0x3fe/0x640 [be2net] be_cmd_set_mac+0xaf/0x110 [be2net] be_vf_eth_addr_config+0x19f/0x330 [be2net] be_vf_setup+0x4f7/0x990 [be2net] be_pci_sriov_configure+0x3a1/0x470 [be2net] sriov_numvfs_store+0x20b/0x380 kernfs_fop_write_iter+0x354/0x530 vfs_write+0x9b9/0xf60 ksys_write+0xf3/0x1d0 do_syscall_64+0x8c/0x3d0 be_cmd_set_mac_list() calls dma_free_coherent() under a spin_lock_bh. Fix it by freeing only after the lock has been released.

CVE ID	Description
CVE-2025-38571	In the Linux kernel, the following vulnerability has been resolved: sunrpc: fix client side handling of tls alerts A security exploit was discovered in NFS over TLS in <code>tls_alert_rcv</code> due to its assumption that there is valid data in the <code>msghdr</code> 's iterator's <code>kvec</code> . Instead, this patch proposes the rework how control messages are setup and used by <code>sock_recvmsg()</code> . If no control message structure is setup, kTLS layer will read and process TLS data record types. As soon as it encounters a TLS control message, it would return an error. At that point, NFS can setup a <code>kvec</code> backed control buffer and read in the control message such as a TLS alert. Scott found that a <code>msg</code> iterator can advance the <code>kvec</code> pointer as a part of the copy process thus we need to revert the iterator before calling into the <code>tls_alert_rcv</code> .
CVE-2025-38572	In the Linux kernel, the following vulnerability has been resolved: ipv6: reject malicious packets in <code>ipv6_gso_segment()</code> syzbot was able to craft a packet with very long IPv6 extension headers leading to an overflow of <code>skb->transport_header</code> . This 16bit field has a limited range. Add <code>skb_reset_transport_header_careful()</code> helper and use it from <code>ipv6_gso_segment()</code> WARNING: CPU: 0 PID: 5871 at ./include/linux/skbuff.h:3032 <code>skb_reset_transport_header</code> include/linux/skbuff.h:3032 [inline] WARNING: CPU: 0 PID: 5871 at ./include/linux/skbuff.h:3032 <code>ipv6_gso_segment</code> +0x15e2/0x21e0 <code>net/ipv6/ip6_offload.c:151</code> Modules linked in: CPU: 0 UID: 0 PID: 5871 Comm: syz-executor211 Not tainted 6.16.0-rc6-syzkaller-g7abc678e3084 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 07/12/2025 RIP: 0010:skb_reset_transport_header include/linux/skbuff.h:3032 [inline] RIP: 0010:ipv6_gso_segment+0x15e2/0x21e0 <code>net/ipv6/ip6_offload.c:151</code> Call Trace: <TASK> <code>skb_mac_gso_segment+0x31c/0x640</code> <code>net/core/gso.c:53</code> <code>nsh_gso_segment+0x54a/0xe10</code> <code>net/nsh/nsh.c:110</code> <code>skb_mac_gso_segment+0x31c/0x640</code> <code>net/core/gso.c:53</code> <code>__skb_gso_segment+0x342/0x510</code> <code>net/core/gso.c:124</code> <code>skb_gso_segment</code> include/net/gso.h:83 [inline] <code>validate_xmit_skb+0x857/0x11b0</code> <code>net/core/dev.c:3950</code> <code>validate_xmit_skb_list+0x84/0x120</code> <code>net/core/dev.c:4000</code> <code>sch_direct_xmit+0xd3/0x4b0</code> <code>net/sched/sch_generic.c:329</code> <code>__dev_xmit_skb</code> <code>net/core/dev.c:4102</code> [inline] <code>__dev_queue_xmit</code> +0x17b6/0x3a70 <code>net/core/dev.c:4679</code>
CVE-2025-38574	In the Linux kernel, the following vulnerability has been resolved: ptp: ensure minimal <code>skb</code> length in <code>ptp_xmit()</code> Commit aabc6596ffb3 ("net: ptp: Add bound checking for <code>skb</code> data on <code>ppp_sync_txmunge()</code> ") fixed <code>ppp_sync_txmunge()</code> We need a similar fix in <code>ptp_xmit()</code> , otherwise we might read uninit data as reported by syzbot. BUG: KMSAN: uninit-value in <code>ptp_xmit+0xc34/0x2720</code> <code>drivers/net/ppp/pptp.c:193</code> <code>ptp_xmit+0xc34/0x2720</code> <code>drivers/net/ppp/pptp.c:193</code> <code>ppp_channel_bridge_input</code> <code>drivers/net/ppp/ppp_generic.c:2290</code> [inline] <code>ppp_input+0x1d6/0xe60</code> <code>drivers/net/ppp/ppp_generic.c:2314</code> <code>pppoe_rcv_core+0x1e8/0x760</code> <code>drivers/net/ppp/pppoe.c:379</code> <code>sk_backlog_rcv+0x142/0x420</code> include/net/sock.h:1148 <code>__release_sock+0x1d3/0x330</code> <code>net/core/sock.c:3213</code> <code>release_sock+0x6b/0x270</code> <code>net/core/sock.c:3767</code> <code>pppoe_sendmsg+0x15d/0xcb0</code> <code>drivers/net/ppp/pppoe.c:904</code> <code>sock_sendmsg_nosec</code> <code>net/socket.c:712</code> [inline] <code>__sock_sendmsg+0x330/0x3d0</code> <code>net/socket.c:727</code> <code>__sys_sendmsg+0x893/0xd80</code> <code>net/socket.c:2566</code> <code>__sys_sendmsg+0x271/0x3b0</code> <code>net/socket.c:2620</code> <code>__sys_sendmmsg+0x2d9/0x7c0</code> <code>net/socket.c:2709</code>
CVE-2025-38575	In the Linux kernel, the following vulnerability has been resolved: ksmbd: use <code>aead_request_free</code> to match <code>aead_request_alloc</code> Use <code>aead_request_free()</code> instead of <code>kfree()</code> to properly free memory allocated by <code>aead_request_alloc()</code> . This ensures sensitive crypto data is zeroed before being freed.
CVE-2025-38576	In the Linux kernel, the following vulnerability has been resolved: powerpc/eeh: Make EEH driver device hotplug safe Multiple race conditions existed between the PCIe hotplug driver and the EEH driver, leading to a variety of kernel oopses of the same general nature: <pcie device unplug> <eeh driver trigger> <hotplug removal trigger> <pcie tree reconfiguration> <eeh recovery next step> <oops in EEH driver bus iteration loop> A second class of oops is also seen when the underlying bus disappears during device recovery. Refactor the EEH module to be PCI rescan and remove safe. Also clean up a few minor formatting / readability issues.

CVE ID	Description
CVE-2025-38577	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid panic in f2fs_evict_inode As syzbot [1] reported as below: R10: 0000000000000100 R11: 0000000000000206 R12: 00007ffe17473450 R13: 00007f28b1c10854 R14: 0000000000000dae5 R15: 00007ffe17474520 </TASK> ---[end trace 0000000000000000]--- ===== BUG: KASAN: use-after-free in __list_del_entry_valid+0xa6/0x130 lib/list_debug.c:62 Read of size 8 at addr ffff88812d962278 by task syz-executor/564 CPU: 1 PID: 564 Comm: syz-executor Tainted: G W 6.1.129-syzkaller #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 Call Trace: <TASK> __dump_stack+0x21/0x24 lib/dump_stack.c:88 dump_stack_lvl+0xee/0x158 lib/dump_stack.c:106 print_address_description+0x71/0x210 mm/kasan/report.c:316 print_report+0x4a/0x60 mm/kasan/report.c:427 kasan_report+0x122/0x150 mm/kasan/report.c:531 __asan_report_load8_noabort+0x14/0x20 mm/kasan/report_generic.c:351 __list_del_entry_valid+0xa6/0x130 lib/list_debug.c:62 __list_del_entry include/linux/list.h:134 [inline] list_del_init include/linux/list.h:206 [inline] f2fs_inode_synced+0xf7/0x2e0 fs/f2fs/super.c:1531 f2fs_update_inode+0x74/0x1c40 fs/f2fs/inode.c:585 f2fs_update_inode_page+0x137/0x170 fs/f2fs/inode.c:703 f2fs_write_inode+0x4ec/0x770 fs/f2fs/inode.c:731 write_inode fs/fs-writeback.c:1460 [inline] __writeback_single_inode+0x4a0/0xab0 fs/fs-writeback.c:1677 writeback_single_inode+0x221/0x8b0 fs/fs-writeback.c:1733 sync_inode_metadata+0xb6/0x110 fs/fs-writeback.c:2789 f2fs_sync_inode_meta+0x16d/0x2a0 fs/f2fs/checkpoint.c:1159 block_operations fs/f2fs/checkpoint.c:1269 [inline] f2fs_write_checkpoint+0xca3/0x2100 fs/f2fs/checkpoint.c:1658 kill_f2fs_super+0x231/0x390 fs/f2fs/super.c:4668 deactivate_locked_super+0x98/0x100 fs/super.c:332 deactivate_super+0xaf/0xe0 fs/super.c:363 cleanup_mnt+0x45f/0x4e0 fs/namespace.c:1186 __cleanup_mnt+0x19/0x20 fs/namespace.c:1193 task_work_run+0x1c6/0x230 kernel/task_work.c:203 exit_task_work include/linux/task_work.h:39 [inline] do_exit+0x9fb/0x2410 kernel/exit.c:871 do_group_exit+0x210/0x2d0 kernel/exit.c:1021 __do_sys_exit_group kernel/exit.c:1032 [inline] __se_sys_exit_group kernel/exit.c:1030 [inline] __x64_sys_exit_group+0x3f/0x40 kernel/exit.c:1030 x64_sys_call+0x7b4/0x9a0 arch/x86/include/generated/asm/syscalls_64.h:232 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x4c/0xa0 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x68/0xd2 RIP: 0033:0x7f28b1b8e169 Code: Unable to access opcode bytes at 0x7f28b1b8e13f. RSP: 002b:00007ffe174710a8 EFLAGS: 00000246 ORIG_RAX: 00000000000000e7 RAX: ffffffff80000000 RBX: 00007f28b1c10879 RCX: 00007f28b1b8e169 RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000001 RBP: 0000000000000002 R08: 00007ffe1746ee47 R09: 00007ffe17472360 R10: 0000000000000009 R11: 0000000000000246 R12: 00007ffe17472360 R13: 00007f28b1c10854 R14: 0000000000000dae5 R15: 00007ffe17474520 </TASK> Allocated by task 569: kasan_save_stack mm/kasan/common.c:45 [inline] kasan_set_track+0x4b/0x70 mm/kasan/common.c:52 kasan_save_alloc_info+0x25/0x30 mm/kasan/generic.c:505 __kasan_slab_alloc+0x72/0x80 mm/kasan/common.c:328 kasan_slab_alloc include/linux/kasan.h:201 [inline] slab_post_alloc_hook+0x4f/0x2c0 mm/slab.h:737 slab_alloc_node mm/slub.c:3398 [inline] slab_alloc mm/slub.c:3406 [inline] __kmem_cache_alloc_lru mm/slub.c:3413 [inline] kmem_cache_alloc_lru+0x104/0x220 mm/slub.c:3429 alloc_inode_sb include/linux/fs.h:3245 [inline] f2fs_alloc_inode+0x2d/0x340 fs/f2fs/super.c:1419 alloc_inode fs/inode.c:261 [inline] iget_locked+0x186/0x880 fs/inode.c:1373 f2fs_iget+0x55/0x4c60 fs/f2fs/inode.c:483 f2fs_lookup+0x366/0xab0 fs/f2fs/namei.c:487 __lookup_slow+0x2a3/0x3d0 fs/namei.c:1690 lookup_slow+0x57/0x70 fs/namei.c:1707 walk_component+0x2e6/0x410 fs/namei ---truncated---

CVE ID	Description
CVE-2025-38578	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid UAF in f2fs_sync_inode_meta() syzbot reported an UAF issue as below: [1] [2] [1] https://syzkaller.appspot.com/text?tag=CrashReport&x=16594c60580000 ===== BUG: KASAN: use-after-free in __list_del_entry_valid+0xa6/0x130 lib/list_debug.c:62 Read of size 8 at addr ffff888100567dc8 by task kworker/u4:0/8 CPU: 1 PID: 8 Comm: kworker/u4:0 Tainted: G W 6.1.129-syzkaller-00017-g642656a36791 #0 Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 02/12/2025 Workqueue: writeback wb_workfn (flush-7:0) Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0x151/0x1b7 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:316 [inline] print_report+0x158/0x4e0 mm/kasan/report.c:427 kasan_report+0x13c/0x170 mm/kasan/report.c:531 __asan_report_load8_noabort+0x14/0x20 mm/kasan/report_generic.c:351 __list_del_entry_valid+0xa6/0x130 lib/list_debug.c:62 __list_del_entry include/linux/list.h:134 [inline] list_del_init include/linux/list.h:206 [inline] f2fs_inode_synced+0x100/0x2e0 fs/f2fs/super.c:1553 f2fs_update_inode+0x72/0x1c40 fs/f2fs/inode.c:588 f2fs_update_inode_page+0x135/0x170 fs/f2fs/inode.c:706 f2fs_write_inode+0x416/0x790 fs/f2fs/inode.c:734 write_inode fs/fs-writeback.c:1460 [inline] __writeback_single_inode+0x4cf/0xb80 fs/fs-writeback.c:1677 writeback_sb_inodes+0xb32/0x1910 fs/fs-writeback.c:1903 __writeback_inodes_wb+0x118/0x3f0 fs/fs-writeback.c:1974 wb_writeback+0x3da/0xa00 fs/fs-writeback.c:2081 wb_check_background_flush fs/fs-writeback.c:2151 [inline] wb_do_writeback fs/fs-writeback.c:2239 [inline] wb_workfn+0xbba/0x1030 fs/fs-writeback.c:2266 process_one_work+0x73d/0xcb0 kernel/workqueue.c:2299 worker_thread+0xa60/0x1260 kernel/workqueue.c:2446 kthread+0x26d/0x300 kernel/kthread.c:386 ret_from_fork+0x1f/0x30 arch/x86/entry/entry_64.S:295 </TASK> Allocated by task 298: kasan_save_stack mm/kasan/common.c:45 [inline] kasan_set_track+0x4b/0x70 mm/kasan/common.c:52 kasan_save_alloc_info+0x1f/0x30 mm/kasan/generic.c:505 __kasan_slab_alloc+0x6c/0x80 mm/kasan/common.c:333 kasan_slab_alloc include/linux/kasan.h:202 [inline] slab_post_alloc_hook+0x53/0x2c0 mm/slab.h:768 slab_alloc_node mm/slab.c:3421 [inline] slab_alloc mm/slab.c:3431 [inline] __kmem_cache_alloc_lru mm/slab.c:3438 [inline] kmem_cache_alloc_lru+0x102/0x270 mm/slab.c:3454 alloc_inode_sb include/linux/fs.h:3255 [inline] f2fs_alloc_inode+0x2d/0x350 fs/f2fs/super.c:1437 alloc_inode fs/inode.c:261 [inline] iget_locked+0x18c/0x7e0 fs/inode.c:1373 f2fs_iget+0x55/0x4ca0 fs/f2fs/inode.c:486 f2fs_lookup+0x3c1/0xb50 fs/f2fs/namei.c:484 __lookup_slow+0x2b9/0x3e0 fs/namei.c:1689 lookup_slow+0x5a/0x80 fs/namei.c:1706 walk_component+0x2e7/0x410 fs/namei.c:1997 lookup_last fs/namei.c:2454 [inline] path_lookupat+0x16d/0x450 fs/namei.c:2478 filename_lookup+0x251/0x600 fs/namei.c:2507 vfs_statx+0x107/0x4b0 fs/stat.c:229 vfs_fstatat fs/stat.c:267 [inline] vfs_lstat include/linux/fs.h:3434 [inline] __do_sys_newlstat fs/stat.c:423 [inline] __se_sys_newlstat+0xda/0x7c0 fs/stat.c:417 __x64_sys_newlstat+0x5b/0x70 fs/stat.c:417 x64_sys_call+0x52/0x9a0 arch/x86/include/generated/asm/syscalls_64.h:7 do_syscall_x64 arch/x86/entry/common.c:51 [inline] do_syscall_64+0x3b/0x80 arch/x86/entry/common.c:81 entry_SYSCALL_64_after_hwframe+0x68/0xd2 Freed by task 0: kasan_save_stack mm/kasan/common.c:45 [inline] kasan_set_track+0x4b/0x70 mm/kasan/common.c:52 kasan_save_free_info+0x2b/0x40 mm/kasan/generic.c:516 ____kasan_slab_free+0x131/0x180 mm/kasan/common.c:241 __kasan_slab_free+0x11/0x20 mm/kasan/common.c:249 kasan_slab_free include/linux/kasan.h:178 [inline] slab_free_hook mm/slab.c:1745 [inline] slab_free_freelist_hook mm/slab.c:1771 [inline] slab_free mm/slab.c:3686 [inline] kmem_cache_free+0x ---truncated---
CVE-2025-38579	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix KMSAN uninit-value in extent_info usage KMSAN reported a use of uninitialized value in `__is_extent_mergeable()` and `__is_back_mergeable()` via the read extent tree path. The root cause is that `get_read_extent_info()` only initializes three fields (`fofs`, `blk`, `len`) of `struct extent_info`, leaving the remaining fields uninitialized. This leads to undefined behavior when those fields are accessed later, especially during extent merging. Fix it by zero-initializing the `extent_info` struct before population.

CVE ID	Description
CVE-2025-38581	In the Linux kernel, the following vulnerability has been resolved: crypto: ccp - Fix crash when rebind ccp device for ccp.ko When CONFIG_CRYPTODEV_CCP_DEBUGFS is enabled, rebinding the ccp device causes the following crash: \$ echo '0000:0a:00.2' > /sys/bus/pci/drivers/ccp/unbind \$ echo '0000:0a:00.2' > /sys/bus/pci/drivers/ccp/bind [204.976930] BUG: kernel NULL pointer dereference, address: 0000000000000098 [204.978026] #PF: supervisor write access in kernel mode [204.979126] #PF: error_code(0x0002) - not-present page [204.980226] PGD 0 P4D 0 [204.981317] Oops: Oops: 0002 [#1] SMP NOPTI ... [204.997852] Call Trace: [204.999074] <TASK> [205.000297] start_creating+0x9f/0x1c0 [205.001533] debugfs_create_dir+0x1f/0x170 [205.002769] ? srso_return_thunk+0x5/0x5f [205.004000] ccp5_debugfs_setup+0x87/0x170 [ccp] [205.005241] ccp5_init+0x8b2/0x960 [ccp] [205.006469] ccp_dev_init+0xd4/0x150 [ccp] [205.007709] sp_init+0x5f/0x80 [ccp] [205.008942] sp_pci_probe+0x283/0x2e0 [ccp] [205.010165] ? srso_return_thunk+0x5/0x5f [205.011376] local_pci_probe+0x4f/0xb0 [205.012584] pci_device_probe+0xdb/0x230 [205.013810] really_probe+0xed/0x380 [205.015024] __driver_probe_device+0x7e/0x160 [205.016240] device_driver_attach+0x2f/0x60 [205.017457] bind_store+0x7c/0xb0 [205.018663] drv_attr_store+0x28/0x40 [205.019868] sysfs_kf_write+0x5f/0x70 [205.021065] kernfs_fop_write_iter+0x145/0x1d0 [205.022267] vfs_write+0x308/0x440 [205.023453] ksys_write+0x6d/0xe0 [205.024616] __x64_sys_write+0x1e/0x30 [205.025778] x64_sys_call+0x16ba/0x2150 [205.026942] do_syscall_64+0x56/0x1e0 [205.028108] entry_SYSCALL_64_after_hwframe+0x76/0x7e [205.029276] RIP: 0033:0x7fbc36f10104 [205.030420] Code: 89 02 48 c7 c0 ff ff ff c3 66 2e 0f 1f 84 00 00 00 00 00 66 90 48 8d 05 e1 08 2e 00 8b 00 85 c0 75 13 b8 01 00 00 0f 05 <48> 3d 00 f0 ff ff 77 54 f3 c3 66 90 41 54 55 49 89 d4 53 48 89 f5 This patch sets ccp_debugfs_dir to NULL after destroying it in ccp5_debugfs_destroy, allowing the directory dentry to be recreated when rebinding the ccp device. Tested on AMD Ryzen 7 1700X.
CVE-2025-38582	In the Linux kernel, the following vulnerability has been resolved: RDMA/hns: Fix double destruction of rsv_qp rsv_qp may be double destroyed in error flow, first in free_mr_init(), and then in hns_roce_exit(). Fix it by moving the free_mr_init() call into hns_roce_v2_init(). list_del corruption, ffff589732eb9b50->next is LIST_POISON1 (dead000000000100) WARNING: CPU: 8 PID: 1047115 at lib/list_debug.c:53 __list_del_entry_valid+0x148/0x240 ... Call trace: __list_del_entry_valid+0x148/0x240 hns_roce_qp_remove+0x4c/0x3f0 [hns_roce_hw_v2] hns_roce_v2_destroy_qp_common+0x1dc/0x5f4 [hns_roce_hw_v2] hns_roce_v2_destroy_qp+0x22c/0x46c [hns_roce_hw_v2] free_mr_exit+0x6c/0x120 [hns_roce_hw_v2] hns_roce_v2_exit+0x170/0x200 [hns_roce_hw_v2] hns_roce_exit+0x118/0x350 [hns_roce_hw_v2] __hns_roce_hw_v2_init_instance+0x1c8/0x304 [hns_roce_hw_v2] hns_roce_hw_v2_reset_notify_init+0x170/0x21c [hns_roce_hw_v2] hns_roce_hw_v2_reset_notify+0x6c/0x190 [hns_roce_hw_v2] hclge_notify_roce_client+0x6c/0x160 [hclge] hclge_reset_rebuild+0x150/0x5c0 [hclge] hclge_reset+0x10c/0x140 [hclge] hclge_reset_subtask+0x80/0x104 [hclge] hclge_reset_service_task+0x168/0x3ac [hclge] hclge_service_task+0x50/0x100 [hclge] process_one_work+0x250/0x9a0 worker_thread+0x324/0x990 kthread+0x190/0x210 ret_from_fork+0x10/0x18
CVE-2025-38583	In the Linux kernel, the following vulnerability has been resolved: clk: xilinx: vcu: unregister pll_post only if registered correctly If registration of pll_post is failed, it will be set to NULL or ERR, unregistering same will fail with following call trace: Unable to handle kernel NULL pointer dereference at virtual address 008 pc : clk_hw_unregister+0xc/0x20 lr : clk_hw_unregister_fixed_factor+0x18/0x30 sp : ffff800011923850 ... Call trace: clk_hw_unregister+0xc/0x20 clk_hw_unregister_fixed_factor+0x18/0x30 xvcu_unregister_clock_provider+0xcc/0xf4 [xlnx_vcu] xvcu_probe+0x2bc/0x53c [xlnx_vcu]
CVE-2025-38584	In the Linux kernel, the following vulnerability has been resolved: padata: Fix pd UAF once and for all There is a race condition/UAF in padata_reorder that goes back to the initial commit. A reference count is taken at the start of the process in padata_do_parallel, and released at the end in padata_serial_worker. This reference count is (and only is) required for padata_replace to function correctly. If padata_replace is never called then there is no issue. In the function padata_reorder which serves as the core of padata, as soon as padata is added to queue->serial.list, and the associated spin lock released, that padata may be processed and the reference count on pd would go away. Fix this by getting the next padata before the queue->serial lock is released. In order to make this possible, simplify padata_reorder by only calling it once the next padata arrives.

CVE ID	Description
CVE-2025-38585	In the Linux kernel, the following vulnerability has been resolved: staging: media: atomisp: Fix stack buffer overflow in gmin_get_var_int() When gmin_get_config_var() calls efi.get_variable() and the EFI variable is larger than the expected buffer size, two behaviors combine to create a stack buffer overflow: 1. gmin_get_config_var() does not return the proper error code when efi.get_variable() fails. It returns the stale 'ret' value from earlier operations instead of indicating the EFI failure. 2. When efi.get_variable() returns EFI_BUFFER_TOO_SMALL, it updates *out_len to the required buffer size but writes no data to the output buffer. However, due to bug #1, gmin_get_var_int() believes the call succeeded. The caller gmin_get_var_int() then performs: - Allocates val[CFG_VAR_NAME_MAX + 1] (65 bytes) on stack - Calls gmin_get_config_var(dev, is_gmin, var, val, &len) with len=64 - If EFI variable is >64 bytes, efi.get_variable() sets len=required_size - Due to bug #1, thinks call succeeded with len=required_size - Executes val[len] = 0, writing past end of 65-byte stack buffer This creates a stack buffer overflow when EFI variables are larger than 64 bytes. Since EFI variables can be controlled by firmware or system configuration, this could potentially be exploited for code execution. Fix the bug by returning proper error codes from gmin_get_config_var() based on EFI status instead of stale 'ret' value. The gmin_get_var_int() function is called during device initialization for camera sensor configuration on Intel Bay Trail and Cherry Trail platforms using the atomisp camera stack.
CVE-2025-38587	In the Linux kernel, the following vulnerability has been resolved: ipv6: fix possible infinite loop in fib6_info Uses dev() fib6_info Uses dev() seems to rely on RCU without an explicit protection. Like the prior fix in rt6_nmsg_size(), we need to make sure fib6_del_route() or fib6_add_rt2node() have not removed the anchor from the list, or we risk an infinite loop.
CVE-2025-38588	In the Linux kernel, the following vulnerability has been resolved: ipv6: prevent infinite loop in rt6_nmsg_size() While testing prior patch, I was able to trigger an infinite loop in rt6_nmsg_size() in the following place: list_for_each_entry_rcu(sibling, &f6i->fib6_siblings, fib6_siblings) { rt6_nh_nmsg_size(sibling->fib6_nh, &next_hop_len); } This is because fib6_del_route() and fib6_add_rt2node() uses list_del_rcu(), which can confuse rcu readers, because they might no longer see the head of the list. Restart the loop if f6i->fib6_nsiblings is zero.
CVE-2025-38590	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Remove skb secpath if xfrm state is not found Hardware returns a unique identifier for a decrypted packet's xfrm state, this state is looked up in an xarray. However, the state might have been freed by the time of this lookup. Currently, if the state is not found, only a counter is incremented. The secpath (sp) extension on the skb is not removed, resulting in sp->len becoming 0. Subsequently, functions like __xfrm_policy_check() attempt to access fields such as xfrm_input_state(skb)->xso.type (which dereferences sp->xvec[sp->len - 1]) without first validating sp->len. This leads to a crash when dereferencing an invalid state pointer. This patch prevents the crash by explicitly removing the secpath extension from the skb if the xfrm state is not found after hardware decryption. This ensures downstream functions do not operate on a zero-length secpath. BUG: unable to handle page fault for address: ffffffff000002c8 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 282e067 P4D 282e067 PUD 0 Oops: Oops: 0000 [#1] SMP CPU: 12 UID: 0 PID: 0 Comm: swapper/12 Not tainted 6.15.0-rc7_for_upstream_min_debug_2025_05_27_22_44 #1 NONE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS rel-1.13.0-0-gf21b5a4eb02-prebuilt.qemu.org 04/01/2014 RIP: 0010: __xfrm_policy_check+0x61a/0xa30 Code: b6 77 7f 83 e6 02 74 14 4d 8b af d8 00 00 00 41 0f b6 45 05 c1 e0 03 48 98 49 01 c5 41 8b 45 00 83 e8 01 48 98 49 8b 44 c5 10 <0f> b6 80 c8 02 00 00 83 e0 0c 3c 04 0f 84 0c 02 00 00 31 ff 80 fa RSP: 0018:ffff88885fb04918 EFLAGS: 00010297 RAX: ffffffff00000000 RBX: 0000000000000002 RCX: 0000000000000000 RDX: 0000000000000002 RSI: 0000000000000002 RDI: 0000000000000000 RBP: ffffffff8311af80 R08: 0000000000000020 R09: 00000000c2eda353 R10: ffff88812be2bbc8 R11: 000000001faab533 R12: ffff88885fb049c8 R13: ffff88812be2bbc8 R14: 0000000000000000 R15: ffff88811896ae00 FS: 0000000000000000(0000) GS: ffff8888dca82000(0000) knlGS: 0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: ffffffff000002c8 CR3: 00000000243050002 CR4: 0000000000372eb0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <IRQ> ? try_to_wake_up+0x108/0x4c0 ? udp4_lib_lookup2+0x8e/0x150 ? udp_lib_lport_inuse+0x100/0x100 ? __udp4_lib_lookup+0x2b0/0x410 __xfrm_policy_check2.constprop.0+0x11e/0x130 udp_queue_rcv_one_skb+0x1d/0x530 udp_unicast_rcv_skb+0x76/0x90 __udp4_lib_rcv+0xa64/0xe90 ip_protocol_deliver_rcu+0x20/0x130 ip_local_deliver_finish+0x75/0xa0 ip_local_deliver+0xc1/0xd0 ? ip_protocol_deliver_rcu+0x130/0x130 ip_sublist_rcv+0x1f9/0x240 ? ip_rcv_finish_core+0x430/0x430 ip_list_rcv+0xfc/0x130 __netif_receive_skb_list_core+0x181/0x1e0 netif_receive_skb_list_internal+0x200/0x360 ? mlx5e_build_rx_skb+0x1bc/0xda0 [mlx5_core] gro_receive_skb+0xfd/0x210 mlx5e_handle_rx_cqe_mpwrrq+0x141/0x280 [mlx5_core] mlx5e_poll_rx_cq+0xcc/0x8e0 [mlx5_core] ? mlx5e_handle_rx_dim+0x91/0xd0 [mlx5_core] mlx5e_napi_poll+0x114/0xab0 [mlx5_core] __napi_poll+0x25/0x170 net_rx_action+0x32d/0x3a0 ? mlx5_eq_comp_int+0x8d/0x280 [mlx5_core] ? notifier_call_chain+0x33/0xa0 handle_softirqs+0xda/0x250 irq_exit_rcu+0x6d/0xc0 common_interrupt+0x81/0xa0 </IRQ>

CVE ID	Description
CVE-2025-38591	In the Linux kernel, the following vulnerability has been resolved: bpf: Reject narrower access to pointer ctx fields The following BPF program, simplified from a syzkaller repro, causes a kernel warning: r0 = *(u8 *) (r1 + 169); exit; With pointer field sk being at offset 168 in __sk_buff. This access is detected as a narrower read in bpf_skb_is_valid_access because it doesn't match offsetof(struct __sk_buff, sk). It is therefore allowed and later proceeds to bpf_convert_ctx_access. Note that for the "is_narrower_load" case in the convert_ctx_accesses(), the insn->off is aligned, so the cnt may not be 0 because it matches the offsetof(struct __sk_buff, sk) in the bpf_convert_ctx_access. However, the target_size stays 0 and the verifier errors with a kernel warning: verifier bug: error during ctx access conversion(1) This patch fixes that to return a proper "invalid bpf_context access off=X size=Y" error on the load instruction. The same issue affects multiple other fields in context structures that allow narrow access. Some other non-affected fields (for sk_msg, sk_lookup, and sockopt) were also changed to use bpf_ctx_range_ptr for consistency. Note this syzkaller crash was reported in the "Closes" link below, which used to be about a different bug, fixed in commit fce7bd8e385a ("bpf/verifier: Handle BPF_LOAD_ACQ instructions in insn_def_regno()"). Because syzbot somehow confused the two bugs, the new crash and repro didn't get reported to the mailing list.
CVE-2025-38593	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_sync: fix double free in 'hci_discovery_filter_clear()' Function 'hci_discovery_filter_clear()' frees 'uuids' array and then sets it to NULL. There is a tiny chance of the following race: 'hci_cmd_sync_work()' 'update_passive_scan_sync()' 'hci_update_passive_scan_sync()' 'hci_discovery_filter_clear()' kfree(uuids); <-----preempted-----> 'start_service_discovery()' 'hci_discovery_filter_clear()' kfree(uuids); // DOUBLE FREE <-----preempted-----> uuids = NULL; To fix it let's add locking around 'kfree()' call and NULL pointer assignment. Otherwise the following backtrace fires: [] -----[cut here]----- [] kernel BUG at mm/slub.c:547! [] Internal error: Oops - BUG: 00000000f2000800 [#1] PREEMPT SMP [] CPU: 3 UID: 0 PID: 246 Comm: bluetoothd Tainted: G O 6.12.19-kernel #1 [] Tainted: [O]=OOT_MODULE [] pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYPE=--) [] pc : __slab_free+0xf8/0x348 [] lr : __slab_free+0x48/0x348 ... [] Call trace: [] __slab_free+0xf8/0x348 [] kfree+0x164/0x27c [] start_service_discovery+0x1d0/0x2c0 [] hci_sock_sendmsg+0x518/0x924 [] __sock_sendmsg+0x54/0x60 [] sock_write_iter+0x98/0xf8 [] do_iter_readv_writev+0xe4/0x1c8 [] vfs_writev+0x128/0x2b0 [] do_writev+0xfc/0x118 [] __arm64_sys_writev+0x20/0x2c [] invoke_syscall+0x68/0xf0 [] el0_svc_common.constprop.0+0x40/0xe0 [] do_el0_svc+0x1c/0x28 [] el0_svc+0x30/0xd0 [] el0t_64_sync_handler+0x100/0x12c [] el0t_64_sync+0x194/0x198 [] Code: 8b0002e6 eb17031f 54ffbe1 d503201f (d4210000) [] ---[end trace 0000000000000000]---
CVE-2025-38595	In the Linux kernel, the following vulnerability has been resolved: xen: fix UAF in dmabuf_exp_from_pages() [dma_buf_fd()] fixes; no preferences regarding the tree it goes through - up to xen folks] As soon as we'd inserted a file reference into descriptor table, another thread could close it. That's fine for the case when all we are doing is returning that descriptor to userland (it's a race, but it's a userland race and there's nothing the kernel can do about it). However, if we follow fd_install() with any kind of access to objects that would be destroyed on close (be it the struct file itself or anything destroyed by its ->release()), we have a UAF. dma_buf_fd() is a combination of reserving a descriptor and fd_install(). gntdev dmabuf_exp_from_pages() calls it and then proceeds to access the objects destroyed on close - starting with gntdev_dmabuf itself. Fix that by doing reserving descriptor before anything else and do fd_install() only when everything had been set up.
CVE-2025-38597	In the Linux kernel, the following vulnerability has been resolved: drm/rockchip: vop2: fail cleanly if missing a primary plane for a video-port Each window of a vop2 is usable by a specific set of video ports, so while binding the vop2, we look through the list of available windows trying to find one designated as primary-plane and usable by that specific port. The code later wants to use drm_crtc_init_with_planes with that found primary plane, but nothing has checked so far if a primary plane was actually found. For whatever reason, the rk3576 vp2 does not have a usable primary window (if vp0 is also in use) which brought the issue to light and ended in a null-pointer dereference further down. As we expect a primary-plane to exist for a video-port, add a check at the end of the window-iteration and fail probing if none was found.

CVE ID	Description
CVE-2025-38601	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: clear initialized flag for deinit-ed srng lists In a number of cases we see kernel panics on resume due to ath11k kernel page fault, which happens under the following circumstances: 1) First ath11k_hal_dump_srng_stats() call Last interrupt received for each group: ath11k_pci 0000:01:00.0: group_id 0 22511ms before ath11k_pci 0000:01:00.0: group_id 1 14440788ms before [...] ath11k_pci 0000:01:00.0: failed to receive control response completion, polling.. ath11k_pci 0000:01:00.0: Service connect timeout ath11k_pci 0000:01:00.0: failed to connect to HTTP: -110 ath11k_pci 0000:01:00.0: failed to start core: -110 ath11k_pci 0000:01:00.0: firmware crashed: MHI_CB_EE_RDDM ath11k_pci 0000:01:00.0: already resetting count 2 ath11k_pci 0000:01:00.0: failed to wait wlan mode request (mode 4): -110 ath11k_pci 0000:01:00.0: qmi failed to send wlan mode off: -110 ath11k_pci 0000:01:00.0: failed to reconfigure driver on crash recovery [...] 2) At this point reconfiguration fails (we have 2 resets) and ath11k_core_reconfigure_on_crash() calls ath11k_hal_srng_deinit() which destroys srng lists. However, it does not reset per-list ->initialized flag. 3) Second ath11k_hal_dump_srng_stats() call sees stale ->initialized flag and attempts to dump srng stats: Last interrupt received for each group: ath11k_pci 0000:01:00.0: group_id 0 66785ms before ath11k_pci 0000:01:00.0: group_id 1 14485062ms before ath11k_pci 0000:01:00.0: group_id 2 14485062ms before ath11k_pci 0000:01:00.0: group_id 3 14485062ms before ath11k_pci 0000:01:00.0: group_id 4 14780845ms before ath11k_pci 0000:01:00.0: group_id 5 14780845ms before ath11k_pci 0000:01:00.0: group_id 6 14485062ms before ath11k_pci 0000:01:00.0: group_id 7 66814ms before ath11k_pci 0000:01:00.0: group_id 8 68997ms before ath11k_pci 0000:01:00.0: group_id 9 67588ms before ath11k_pci 0000:01:00.0: group_id 10 69511ms before BUG: unable to handle page fault for address: fffff007404eb010 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 100000067 P4D 100000067 PUD 10022d067 PMD 100b01067 PTE 0 Oops: 0000 [#1] PREEMPT SMP NOPTI RIP: 0010:ath11k_hal_dump_srng_stats+0x2b4/0x3b0 [ath11k] Call Trace: <TASK> ? __die_body+0xae/0xb0 ? page_fault_oops+0x381/0x3e0 ? exc_page_fault+0x69/0xa0 ? asm_exc_page_fault+0x22/0x30 ? ath11k_hal_dump_srng_stats+0x2b4/0x3b0 [ath11k (HASH:6cea 4)] ath11k_qmi_driver_event_work+0xbd/0x1050 [ath11k (HASH:6cea 4)] worker_thread+0x389/0x930 kthread+0x149/0x170 Clear per-list ->initialized flag in ath11k_hal_srng_deinit().
CVE-2025-38602	In the Linux kernel, the following vulnerability has been resolved: iwlfwifi: Add missing check for alloc_ordered_workqueue Add check for the return value of alloc_ordered_workqueue since it may return NULL pointer.
CVE-2025-38604	In the Linux kernel, the following vulnerability has been resolved: wifi: rtl818x: Kill URBs before clearing tx status queue In rtl8187_stop() move the call of usb_kill_anchored_urbs() before clearing b_tx_status.queue. This change prevents callbacks from using already freed skb due to anchor was not killed before freeing such skb. BUG: kernel NULL pointer dereference, address: 0000000000000080 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: Oops: 0000 [#1] SMP NOPTI CPU: 7 UID: 0 PID: 0 Comm: swapper/7 Not tainted 6.15.0 #8 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 0.0.0 02/06/2015 RIP: 0010:ieee80211_tx_status_irqsafe+0x21/0xc0 [mac80211] Call Trace: <IRQ> rtl8187_tx_cb+0x116/0x150 [rtl8187] __usb_hcd_giveback_urb+0x9d/0x120 usb_giveback_urb_bh+0xbb/0x140 process_one_work+0x19b/0x3c0 bh_worker+0x1a7/0x210 tasklet_action+0x10/0x30 handle_softirqs+0xf0/0x340 __irq_exit_rcu+0xcd/0xf0 common_interrupt+0x85/0xa0 <IRQ> Tested on RTL8187BvE device. Found by Linux Verification Center (linuxtesting.org) with SVACE.
CVE-2025-38605	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Pass ab pointer directly to ath12k_dp_tx_get_encap_type() In ath12k_dp_tx_get_encap_type(), the arvif parameter is only used to retrieve the ab pointer. In vdev delete sequence the arvif->ar could become NULL and that would trigger kernel panic. Since the caller ath12k_dp_tx() already has a valid ab pointer, pass it directly to avoid panic and unnecessary dereferencing. PC points to "ath12k_dp_tx+0x228/0x988 [ath12k]" LR points to "ath12k_dp_tx+0xc8/0x988 [ath12k]". The Backtrace obtained is as follows: ath12k_dp_tx+0x228/0x988 [ath12k] ath12k_mac_tx_check_max_limit+0x608/0x920 [ath12k] ieee80211_process_measurement_req+0x320/0x348 [mac80211] ieee80211_tx_dequeue+0x9ac/0x1518 [mac80211] ieee80211_tx_dequeue+0xb14/0x1518 [mac80211] ieee80211_tx_prepare_skb+0x224/0x254 [mac80211] ieee80211_xmit+0x9c/0x100 [mac80211] __ieee80211_subif_start_xmit+0xc50/0xf40 [mac80211] ieee80211_subif_start_xmit+0x2e8/0x308 [mac80211] netdev_start_xmit+0x150/0x18c dev_hard_start_xmit+0x74/0xc0 Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.3.1-00173-QCAHKSUPL_SILICONZ-1
CVE-2025-38608	In the Linux kernel, the following vulnerability has been resolved: bpf, ktls: Fix data corruption when using bpf_msg_pop_data() in ktls When sending plaintext data, we initially calculated the corresponding ciphertext length. However, if we later reduced the plaintext data length via socket policy, we failed to recalculate the ciphertext length. This results in transmitting buffers containing uninitialized data during ciphertext transmission. This causes uninitialized bytes to be appended after a complete "Application Data" packet, leading to errors on the receiving end when parsing TLS record.

CVE ID	Description
CVE-2025-38609	In the Linux kernel, the following vulnerability has been resolved: PM / devfreq: Check governor before using governor->name Commit 96ffcdf239de ("PM / devfreq: Remove redundant governor_name from struct devfreq") removes governor_name and uses governor->name to replace it. But devfreq->governor may be NULL and directly using devfreq->governor->name may cause null pointer exception. Move the check of governor to before using governor->name.
CVE-2025-38610	In the Linux kernel, the following vulnerability has been resolved: powercap: dtpm_cpu: Fix NULL pointer dereference in get_pd_power_uw() The get_pd_power_uw() function can crash with a NULL pointer dereference when em_cpu_get() returns NULL. This occurs when a CPU becomes impossible during runtime, causing get_cpu_device() to return NULL, which propagates through em_cpu_get() and leads to a crash when em_span_cpus() dereferences the NULL pointer. Add a NULL check after em_cpu_get() and return 0 if unavailable, matching the existing fallback behavior in __dtpm_cpu_setup(). [rjw: Drop an excess empty code line]
CVE-2025-38612	In the Linux kernel, the following vulnerability has been resolved: staging: fbttf: fix potential memory leak in fbttf_framebuffer_alloc() In the error paths after fb_info structure is successfully allocated, the memory allocated in fb_deferred_io_init() for info->pagerefs is not freed. Fix that by adding the cleanup function on the error path.
CVE-2025-38614	In the Linux kernel, the following vulnerability has been resolved: eventpoll: Fix semi-unbounded recursion Ensure that epoll instances can never form a graph deeper than EP_MAX_NESTS+1 links. Currently, ep_loop_check_proc() ensures that the graph is loop-free and does some recursion depth checks, but those recursion depth checks don't limit the depth of the resulting tree for two reasons: - They don't look upwards in the tree. - If there are multiple downwards paths of different lengths, only one of the paths is actually considered for the depth check since commit 28d82dc1c4ed ("epoll: limit paths"). Essentially, the current recursion depth check in ep_loop_check_proc() just serves to prevent it from recursing too deeply while checking for loops. A more thorough check is done in reverse_path_check() after the new graph edge has already been created; this checks, among other things, that no paths going upwards from any non-epoll file with a length of more than 5 edges exist. However, this check does not apply to non-epoll files. As a result, it is possible to recurse to a depth of at least roughly 500, tested on v6.15. (I am unsure if deeper recursion is possible; and this may have changed with commit 8c44dac8add7 ("eventpoll: Fix priority inversion problem").) To fix it: 1. In ep_loop_check_proc(), note the subtree depth of each visited node, and use subtree depths for the total depth calculation even when a subtree has already been visited. 2. Add ep_get_upwards_depth_proc() for similarly determining the maximum depth of an upwards walk. 3. In ep_loop_check(), use these values to limit the total path length between epoll nodes to EP_MAX_NESTS edges.
CVE-2025-38615	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: cangle set bad inode after removing name fails The reproducer uses a file0 on a ntfs3 file system with a corrupted i_link. When renaming, the file0's inode is marked as a bad inode because the file name cannot be deleted. The underlying bug is that make_bad_inode() is called on a live inode. In some cases it's "icache lookup finds a normal inode, d_splice_alias() is called to attach it to dentry, while another thread decides to call make_bad_inode() on it - that would evict it from icache, but we'd already found it there earlier". In some it's outright "we have an inode attached to dentry - that's how we got it in the first place; let's call make_bad_inode() on it just for shits and giggles".
CVE-2025-38617	In the Linux kernel, the following vulnerability has been resolved: net/packet: fix a race in packet_set_ring() and packet_notifier() When packet_set_ring() releases po->bind_lock, another thread can run packet_notifier() and process an NETDEV_UP event. This race and the fix are both similar to that of commit 15fe076dea7 ("net/packet: fix a race in packet_bind() and packet_notifier()"). There too the packet_notifier NETDEV_UP event managed to run while a po->bind_lock critical section had to be temporarily released. And the fix was similarly to temporarily set po->num to zero to keep the socket unhooked until the lock is retaken. The po->bind_lock in packet_set_ring and packet_notifier precede the introduction of git history.
CVE-2025-38618	In the Linux kernel, the following vulnerability has been resolved: vsock: Do not allow binding to VMADDR_PORT_ANY It is possible for a vsock to autobind to VMADDR_PORT_ANY. This can cause a use-after-free when a connection is made to the bound socket. The socket returned by accept() also has port VMADDR_PORT_ANY but is not on the list of unbound sockets. Binding it will result in an extra refcount decrement similar to the one fixed in fcdd2242c023 (vsock: Keep the binding until socket destruction). Modify the check in __vsock_bind_connectible() to also prevent binding to VMADDR_PORT_ANY.

CVE ID	Description
CVE-2025-38619	In the Linux kernel, the following vulnerability has been resolved: media: ti: j721e-csi2rx: fix list_del corruption If ti_csi2rx_start_dma() fails in ti_csi2rx_dma_callback(), the buffer is marked done with VB2_BUF_STATE_ERROR but is not removed from the DMA queue. This causes the same buffer to be retried in the next iteration, resulting in a double list_del() and eventual list corruption. Fix this by removing the buffer from the queue before calling vb2_buffer_done() on error. This resolves a crash due to list_del corruption: [37.811243] j721e-csi2rx 30102000.ticsi2rx: Failed to queue the next buffer for DMA [37.832187] slab kmallo-2k start ffff00000255b000 pointer offset 1064 size 2048 [37.839761] list_del corruption. next->prev should be ffff00000255bc28, but was ffff00000255d428. (next=ffff00000255b428) [37.850799] -----[cut here]----- [37.855424] kernel BUG at lib/list_debug.c:65! [37.859876] Internal error: Oops - BUG: 00000000f2000800 [#1] SMP [37.866061] Modules linked in: i2c_dev usb_f_rndis u_ether libcomposite dwc3 udc_core usb_common aes_ce_blk aes_ce_cipher ghash_ce gf128mul sha1_ce cpufreq_dt dwc3_am62 phy_gmii_sel sa2ul [37.882830] CPU: 0 UID: 0 PID: 0 Comm: swapper/0 Not tainted 6.16.0-rc3+ #28 VOLUNTARY [37.890851] Hardware name: Bosch STLA-GSRV2-B0 (DT) [37.895737] pstate: 600000c5 (nZCv daIF -PAN -UAO -TCO -DIT -SSBS BTYPE=) [37.902703] pc : __list_del_entry_valid_or_report+0x0dc/0x114 [37.908390] lr : __list_del_entry_valid_or_report+0x0dc/0x114 [37.914059] sp : ffff800080003db0 [37.917375] x29: ffff800080003db0 x28: 0000000000000007 x27: ffff800080e50000 [37.924521] x26: 0000000000000000 x25: ffff0000016abb50 x24: dead000000000122 [37.931666] x23: ffff0000016abb78 x22: ffff0000016ab080 x21: ffff800080003de0 [37.938810] x20: ffff00000255bc00 x19: ffff00000255b800 x18: 000000000000000a [37.945956] x17: 20747562202c3832 x16: 6362353532303030 x15: 0720072007200720 [37.953101] x14: 0720072007200720 x13: 0720072007200720 x12: 00000000fffffea [37.960248] x11: ffff800080003b18 x10: 00000000ffffefff x9 : ffff800080f5b568 [37.967396] x8 : ffff800080f5b5c0 x7 : 0000000000017fe8 x6 : c0000000ffffefff [37.974542] x5 : ffff00000fea6688 x4 : 0000000000000000 x3 : 0000000000000000 [37.981686] x2 : 0000000000000000 x1 : ffff800080ef2b40 x0 : 000000000000006d [37.988832] Call trace: [37.991281] __list_del_entry_valid_or_report+0x0dc/0x114 (P) [37.996959] ti_csi2rx_dma_callback+0x84/0x1c4 [38.001419] udma_vchan_complete+0x1e0/0x344 [38.005705] tasklet_action_common+0x118/0x310 [38.010163] tasklet_action+0x30/0x3c [38.013832] handle_softirqs+0x10c/0x2e0 [38.017761] __do_softirq+0x14/0x20 [38.021256] ____do_softirq+0x10/0x20 [38.024931] call_on_irq_stack+0x24/0x60 [38.028873] do_softirq_own_stack+0x1c/0x40 [38.033064] __irq_exit_rcu+0x130/0x15c [38.036909] irq_exit_rcu+0x10/0x20 [38.040403] el1_interrupt+0x38/0x60 [38.043987] el1h_64_irq_handler+0x18/0x24 [38.048091] el1h_64_irq+0x6c/0x70 [38.051501] default_idle_call+0x34/0xe0 (P) [38.055783] do_idle+0x1f8/0x250 [38.059021] cpu_startup_entry+0x34/0x3c [38.062951] rest_init+0xb4/0xc0 [38.066186] console_on_roofs+0x0/0x6c [38.070031] __primary_switched+0x88/0x90 [38.074059] Code: b00037e0 91378000 f9400462 97e9bf49 (d4210000) [38.080168] ---[end trace 0000000000000000]--- [38.084795] Kernel panic - not syncing: Oops - BUG: Fatal exception in interrupt [38.092197] SMP: stopping secondary CPUs [38.096139] Kernel Offset: disabled [38.099631] CPU features: 0x0000,00002000,02000801,0400420b [38.105202] Memory Limit: none [38.108260] ---[end Kernel panic - not syncing: Oops - BUG: Fatal exception in interrupt]---
CVE-2025-38621	In the Linux kernel, the following vulnerability has been resolved: md: make rdev_addable usable for rcu mode Our testcase trigger panic: BUG: kernel NULL pointer dereference, address: 00000000000000e0 ... Oops: Oops: 0000 [#1] SMP NOPTI CPU: 2 UID: 0 PID: 85 Comm: kworker/2:1 Not tainted 6.16.0+ #94 PREEMPT(none) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.1-2.fc37 04/01/2014 Workqueue: md_misc md_start_sync RIP: 0010:rdev_addable+0x4d/0xf0 ... Call Trace: <TASK> md_start_sync+0x329/0x480 process_one_work+0x226/0x6d0 worker_thread+0x19e/0x340 kthread+0x10f/0x250 ret_from_fork+0x14d/0x180 ret_from_fork_asm+0x1a/0x30 </TASK> Modules linked in: raid10 CR2: 00000000000000e0 --- [end trace 0000000000000000]--- RIP: 0010:rdev_addable+0x4d/0xf0 md_spares_need_change in md_start_sync will call rdev_addable which protected by rcu_read_lock/rcu_read_unlock. This rcu context will help protect rdev won't be released, but rdev->mdev will be set to NULL before we call synchronize_rcu in md_kick_rdev_from_array. Fix this by using READ_ONCE and check does rdev->mdev still alive.

CVE ID	Description
CVE-2025-38622	In the Linux kernel, the following vulnerability has been resolved: net: drop UFO packets in udp_rcv_segment() When sending a packet with virtio_net_hdr to tun device, if the gso_type in virtio_net_hdr is SKB_GSO_UDP and the gso_size is less than udphdr size, below crash may happen. -----[cut here]----- kernel BUG at net/core/skbuff.c:4572! Oops: invalid opcode: 0000 [#1] SMP NOPTI CPU: 0 UID: 0 PID: 62 Comm: mytest Not tainted 6.16.0-rc7 #203 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.15.0-1 04/01/2014 RIP: 0010:skb_pull_rcsum+0x8e/0xa0 Code: 00 00 5b c3 cc cc cc cc 8b 93 88 00 00 00 f7 da e8 37 44 38 00 f7 d8 89 83 88 00 00 00 48 8b 83 c8 00 00 00 5b c3 cc cc cc cc <0f> 0b 0f 0b 66 66 2e 0f 1f 84 00 000 RSP: 0018:ffff900001fba38 EFLAGS: 00000297 RAX: 0000000000000004 RBX: ffff8880040c1000 RCX: ffff900001fb948 RDX: ffff888003e6d700 RSI: 0000000000000008 RDI: ffff88800411a062 RBP: ffff8880040c1000 R08: 0000000000000000 R09: 0000000000000001 R10: ffff888003606c00 R11: 0000000000000001 R12: 0000000000000000 R13: ffff888004060900 R14: ffff888004050000 R15: ffff888004060900 FS: 000000002406d3c0(0000) GS:ffff888084a19000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000020000040 CR3: 000000004007000 CR4: 00000000000006f0 Call Trace: <TASK> udp_queue_rcv_one_skb+0x176/0x240 net/ipv4/udp.c:2445 udp_queue_rcv_skb+0x155/0x1f0 net/ipv4/udp.c:2475 udp_unicast_rcv_skb+0x71/0x90 net/ipv4/udp.c:2626 __udp4_lib_rcv+0x433/0xb00 net/ipv4/udp.c:2690 ip_protocol_deliver_rcu+0xa6/0x160 net/ipv4/ip_input.c:205 ip_local_deliver_finish+0x72/0x90 net/ipv4/ip_input.c:233 ip_sublist_rcv_finish+0x5f/0x70 net/ipv4/ip_input.c:579 ip_sublist_rcv+0x122/0x1b0 net/ipv4/ip_input.c:636 ip_list_rcv+0xf7/0x130 net/ipv4/ip_input.c:670 __netif_receive_skb_list_core+0x21d/0x240 net/core/dev.c:6067 netif_receive_skb_list_internal+0x186/0x2b0 net/core/dev.c:6210 napi_complete_done+0x78/0x180 net/core/dev.c:6580 tun_get_user+0xa63/0x1120 drivers/net/tun.c:1909 tun_chr_write_iter+0x65/0xb0 drivers/net/tun.c:1984 vfs_write+0x300/0x420 fs/read_write.c:593 ksys_write+0x60/0xd0 fs/read_write.c:686 do_syscall_64+0x50/0x1c0 arch/x86/entry/syscall_64.c:63 </TASK> To trigger gso segment in udp_queue_rcv_skb(), we should also set option UDP_ENCAP_ESPINUDP to enable udp_sk(sk)->encap_rcv. When the encap_rcv hook return 1 in udp_queue_rcv_one_skb(), udp_csum_pull_header() will try to pull udphdr, but the skb size has been segmented to gso size, which leads to this crash. Previous commit cf329aa42b66 ("udp: cope with UDP GRO packet misdirection") introduces segmentation in UDP receive path only for GRO, which was never intended to be used for UFO, so drop UFO packets in udp_rcv_segment().
CVE-2025-38623	In the Linux kernel, the following vulnerability has been resolved: PCI: pnv_php: Fix surprise plug detection and recovery The existing PowerNV hotplug code did not handle surprise plug events correctly, leading to a complete failure of the hotplug system after device removal and a required reboot to detect new devices. This comes down to two issues: 1) When a device is surprise removed, often the bridge upstream port will cause a PE freeze on the PHB. If this freeze is not cleared, the MSI interrupts from the bridge hotplug notification logic will not be received by the kernel, stalling all plug events on all slots associated with the PE. 2) When a device is removed from a slot, regardless of surprise or programmatic removal, the associated PHB/PE is left frozen. If this freeze is not cleared via a fundamental reset, skiboot is unable to clear the freeze and cannot retrain / rescan the slot. This also requires a reboot to clear the freeze and redetect the device in the slot. Issue the appropriate unfreeze and rescan commands on hotplug events, and don't oops on hotplug if pci_bus_to_OF_node() returns NULL. [bhelgaas: tidy comments]
CVE-2025-38624	In the Linux kernel, the following vulnerability has been resolved: PCI: pnv_php: Clean up allocated IRQs on unplug When the root of a nested PCIe bridge configuration is unplugged, the pnv_php driver leaked the allocated IRQ resources for the child bridges' hotplug event notifications, resulting in a panic. Fix this by walking all child buses and deallocating all its IRQ resources before calling pci_hp_remove_devices(). Also modify the lifetime of the workqueue at struct pnv_php_slot::wq so that it is only destroyed in pnv_php_free_slot(), instead of pnv_php_disable_irq(). This is required since pnv_php_disable_irq() will now be called by workers triggered by hot unplug interrupts, so the workqueue needs to stay allocated. The abridged kernel panic that occurs without this patch is as follows: WARNING: CPU: 0 PID: 687 at kernel/irq/msi.c:292 msi_device_data_release+0x6c/0x9c CPU: 0 UID: 0 PID: 687 Comm: bash Not tainted 6.14.0-rc5+ #2 Call Trace: msi_device_data_release+0x34/0x9c (unreliable) release_nodes+0x64/0x13c devres_release_all+0xc0/0x140 device_del+0x2d4/0x46c pci_destroy_dev+0x5c/0x194 pci_hp_remove_devices+0x90/0x128 pci_hp_remove_devices+0x44/0x128 pnv_php_disable_slot+0x54/0xd4 power_write_file+0xf8/0x18c pci_slot_attr_store+0x40/0x5c sysfs_kf_write+0x64/0x78 kernfs_fop_write_iter+0x1b0/0x290 vfs_write+0x3bc/0x50c ksys_write+0x84/0x140 system_call_exception+0x124/0x230 system_call_vectored_common+0x15c/0x2ec [bhelgaas: tidy comments]
CVE-2025-38625	In the Linux kernel, the following vulnerability has been resolved: vfio/pds: Fix missing detach_ioas op When CONFIG_IOMMUFD is enabled and a device is bound to the pds_vfio_pci driver, the following WARN_ON() trace is seen and probe fails: WARNING: CPU: 0 PID: 5040 at drivers/vfio/vfio_main.c:317 __vfio_register_dev+0x130/0x140 [vfio] <...> pds_vfio_pci 0000:08:00.1: probe with driver pds_vfio_pci failed with error -22 This is because the driver's vfio_device_ops.detach_ioas isn't set. Fix this by using the generic vfio_iommufd_physical_detach_ioas function.

CVE ID	Description
CVE-2025-38626	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to trigger foreground gc during f2fs_map_blocks() in lfs mode w/ "mode=lfs" mount option, generic/299 will cause system panic as below: -----[cut here]----- kernel BUG at fs/f2fs/segment.c:2835! Call Trace: <TASK> f2fs_allocate_data_block+0x6f4/0xc50 f2fs_map_blocks+0x970/0x1550 f2fs_iomap_begin+0xb2/0x1e0 iomap_iter+0x1d6/0x430 __iomap_dio_rw+0x208/0x9a0 f2fs_file_write_iter+0x6b3/0xfa0 aio_write+0x15d/0x2e0 io_submit_one+0x55e/0xab0 __x64_sys_io_submit+0xa5/0x230 do_syscall_64+0x84/0x2f0 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0010:new_curseg+0x70f/0x720 The root cause of we run out-of-space is: in f2fs_map_blocks(), f2fs may trigger foreground gc only if it allocates any physical block, it will be a little bit later when there is multiple threads writing data w/ aio/dio/bufio method in parallel, since we always use OPU in lfs mode, so f2fs_map_blocks() does block allocations aggressively. In order to fix this issue, let's give a chance to trigger foreground gc in prior to block allocation in f2fs_map_blocks().
CVE-2025-38627	In the Linux kernel, the following vulnerability has been resolved: f2fs: compress: fix UAF of f2fs_inode_info in f2fs_free_dic The decompress_io_ctx may be released asynchronously after I/O completion. If this file is deleted immediately after read, and the kworker of processing post_read_wq has not been executed yet due to high workloads, It is possible that the inode(f2fs_inode_info) is evicted and freed before it is used f2fs_free_dic. The UAF case as below: Thread A Thread B - f2fs_decompress_end_io - f2fs_put_dic - queue_work add free_dic work to post_read_wq - do_unlink - iput - evict - call_rcu This file is deleted after read. Thread C kworker to process post_read_wq - rcu_do_batch - f2fs_free_inode - kmem_cache_free inode is freed by rcu - process_scheduled_works - f2fs_late_free_dic - f2fs_free_dic - f2fs_release_decomp_mem read (dic->inode)->i_compress_algorithm This patch store compress_algorithm and sbi in dic to avoid inode UAF. In addition, the previous solution is deprecated in [1] may cause system hang. [1] https://lore.kernel.org/all/c36ab955-c8db-4a8b-a9d0-f07b5f426c3f@kernel.org
CVE-2025-38630	In the Linux kernel, the following vulnerability has been resolved: fbdev: imxfb: Check fb_add_videomode to prevent null-ptr-deref fb_add_videomode() can fail with -ENOMEM when its internal kmalloc() cannot allocate a struct fb_modelist. If that happens, the modelist stays empty but the driver continues to register. Add a check for its return value to prevent potential null-ptr-deref, which is similar to the commit 17186f1f90d3 ("fbdev: Fix do_register_framebuffer to prevent null-ptr-deref in fb_videomode_to_var").
CVE-2025-38632	In the Linux kernel, the following vulnerability has been resolved: pinmux: fix race causing mux_owner NULL with active mux_usecount commit 5a3e85c3c397 ("pinmux: Use sequential access to access desc->pinmux data") tried to address the issue when two client of the same gpio calls pinctrl_select_state() for the same functionality, was resulting in NULL pointer issue while accessing desc->mux_owner. However, issue was not completely fixed due to the way it was handled and it can still result in the same NULL pointer. The issue occurs due to the following interleaving: cpu0 (process A) cpu1 (process B) pin_request() { pin_free() { mutex_lock() desc->mux_usecount--; //becomes 0 .. mutex_unlock() mutex_lock(desc->mux) desc->mux_usecount++; // becomes 1 desc->mux_owner = owner; mutex_unlock(desc->mux) mutex_lock(desc->mux) desc->mux_owner = NULL; mutex_unlock(desc->mux) This sequence leads to a state where the pin appears to be in use ('mux_usecount == 1') but has no owner ('mux_owner == NULL'), which can cause NULL pointer on next pin_request on the same pin. Ensure that updates to mux_usecount and mux_owner are performed atomically under the same lock. Only clear mux_owner when mux_usecount reaches zero and no new owner has been assigned.
CVE-2025-38634	In the Linux kernel, the following vulnerability has been resolved: power: supply: cpcap-charger: Fix null check for power_supply_get_by_name In the cpcap_usb_detect() function, the power_supply_get_by_name() function may return 'NULL' instead of an error pointer. To prevent potential null pointer dereferences, Added a null check.
CVE-2025-38635	In the Linux kernel, the following vulnerability has been resolved: clk: davinci: Add NULL check in davinci_lpsc_clk_register() devm_kasprintf() returns NULL when memory allocation fails. Currently, davinci_lpsc_clk_register() does not check for this case, which results in a NULL pointer dereference. Add NULL check after devm_kasprintf() to prevent this issue and ensuring no resources are left allocated.

CVE ID	Description
CVE-2025-38636	In the Linux kernel, the following vulnerability has been resolved: rv: Use strings in da monitors tracepoints Using DA monitors tracepoints with KASAN enabled triggers the following warning: BUG: KASAN: global-out-of-bounds in do_trace_event_raw_event_event_da_monitor+0xd6/0x1a0 Read of size 32 at addr ffffffffada8980 by task ... Call Trace: <TASK> [...] do_trace_event_raw_event_event_da_monitor+0xd6/0x1a0 ? __pfx_do_trace_event_raw_event_event_da_monitor+0x10/0x10 ? trace_event_sncid+0x83/0x200 trace_event_sncid+0x163/0x200 [...] The buggy address belongs to the variable: automaton_sneep+0x4e0/0x5e0 This is caused by the tracepoints reading 32 bytes __array instead of __string from the automata definition. Such strings are literals and reading 32 bytes ends up in out of bound memory accesses (e.g. the next automaton's data in this case). The error is harmless as, while printing the string, we stop at the null terminator, but it should still be fixed. Use the __string facilities while defining the tracepoints to avoid reading out of bound memory.
CVE-2025-38637	In the Linux kernel, the following vulnerability has been resolved: net_sched: skbprio: Remove overly strict queue assertions In the current implementation, skbprio enqueue/dequeue contains an assertion that fails under certain conditions when SKBPRIO is used as a child qdisc under TBF with specific parameters. The failure occurs because TBF sometimes peeks at packets in the child qdisc without actually dequeuing them when tokens are unavailable. This peek operation creates a discrepancy between the parent and child qdisc queue length counters. When TBF later receives a high-priority packet, SKBPRIO's queue length may show a different value than what's reflected in its internal priority queue tracking, triggering the assertion. The fix removes this overly strict assertions in SKBPRIO, they are not necessary at all.
CVE-2025-38639	In the Linux kernel, the following vulnerability has been resolved: netfilter: xt_nfacct: don't assume acct name is null-terminated BUG: KASAN: slab-out-of-bounds in .. lib/vsprintf.c:721 Read of size 1 at addr ffff88801eac95c8 by task syz-executor183/5851 [...] string+0x231/0x2b0 lib/vsprintf.c:721 vsnprintf+0x739/0xf00 lib/vsprintf.c:2874 [...] nfacct_mt_checkentry+0xd2/0xe0 net/netfilter/xt_nfacct.c:41 xt_check_match+0x3d1/0xab0 net/netfilter/x_tables.c:523 nfnl_acct_find_get() handles non-null input, but the error printk relied on its presence.
CVE-2025-38640	In the Linux kernel, the following vulnerability has been resolved: bpf: Disable migration in nf_hook_run_bpf(). syzbot reported that the netfilter bpf prog can be called without migration disabled in xmit path. Then the assertion in __bpf_prog_run() fails, triggering the splat below. [0] Let's use bpf_prog_run_pin_on_cpu() in nf_hook_run_bpf(). [0]: BUG: assuming non migratable context at ./include/linux/filter.h:703 in_atomic(): 0, irqs_disabled(): 0, migration_disabled() 0 pid: 5829, name: sshd-session 3 locks held by sshd-session/5829: #0: ffff88807b4e4218 (sk_lock-AF_INET){+.-.-}{0:0}, at: lock_sock include/net/sock.h:1667 [inline] #0: ffff88807b4e4218 (sk_lock-AF_INET){+.-.-}{0:0}, at: tcp_sendmsg+0x20/0x50 net/ipv4/tcp.c:1395 #1: ffffffff8e5c4e00 (rcu_read_lock){....}-{1:3}, at: rcu_lock_acquire include/linux/rcupdate.h:331 [inline] #1: ffffffff8e5c4e00 (rcu_read_lock){....}-{1:3}, at: rcu_read_lock include/linux/rcupdate.h:841 [inline] #1: ffffffff8e5c4e00 (rcu_read_lock){....}-{1:3}, at: __ip_queue_xmit+0x69/0x26c0 net/ipv4/ip_output.c:470 #2: ffffffff8e5c4e00 (rcu_read_lock){....}-{1:3}, at: rcu_lock_acquire include/linux/rcupdate.h:331 [inline] #2: ffffffff8e5c4e00 (rcu_read_lock){....}-{1:3}, at: rcu_read_lock include/linux/rcupdate.h:841 [inline] #2: ffffffff8e5c4e00 (rcu_read_lock){....}-{1:3}, at: nf_hook+0xb2/0x680 include/linux/netfilter.h:241 CPU: 0 UID: 0 PID: 5829 Comm: sshd-session Not tainted 6.16.0-rc6-syzkaller-00002-g155a3c003e55 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 05/07/2025 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x16c/0x1f0 lib/dump_stack.c:120 __cant_migrate kernel/sched/core.c:8860 [inline] __cant_migrate+0x1c7/0x250 kernel/sched/core.c:8834 __bpf_prog_run include/linux/filter.h:703 [inline] bpf_prog_run include/linux/filter.h:725 [inline] nf_hook_run_bpf+0x83/0x1e0 net/netfilter/nf_bpf_link.c:20 nf_hook_entry_hookfn include/linux/netfilter.h:157 [inline] nf_hook_slow+0xbb/0x200 net/netfilter/core.c:623 nf_hook+0x370/0x680 include/linux/netfilter.h:272 NF_HOOK_COND include/linux/netfilter.h:305 [inline] ip_output+0x1bc/0x2a0 net/ipv4/ip_output.c:433 dst_output include/net/dst.h:459 [inline] ip_local_out net/ipv4/ip_output.c:129 [inline] __ip_queue_xmit+0x1d7d/0x26c0 net/ipv4/ip_output.c:527 __tcp_transmit_skb+0x2686/0x3e90 net/ipv4/tcp_output.c:1479 tcp_transmit_skb net/ipv4/tcp_output.c:1497 [inline] tcp_write_xmit+0x1274/0x84e0 net/ipv4/tcp_output.c:2838 __tcp_push_pending_frames+0xaf/0x390 net/ipv4/tcp_output.c:3021 tcp_push+0x225/0x700 net/ipv4/tcp.c:759 tcp_sendmsg_locked+0x1870/0x42b0 net/ipv4/tcp.c:1359 tcp_sendmsg+0x2e/0x50 net/ipv4/tcp.c:1396 inet_sendmsg+0xb9/0x140 net/ipv4/af_inet.c:851 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg net/socket.c:727 [inline] sock_write_iter+0x4aa/0x5b0 net/socket.c:1131 new_sync_write fs/read_write.c:593 [inline] vfs_write+0x6c7/0x1150 fs/read_write.c:686 ksys_write+0x1f8/0x250 fs/read_write.c:738 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcd/0x4c0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7fe7d365d407 Code: 48 89 fa 4c 89 df e8 38 aa 00 00 8b 93 08 03 00 00 59 5e 48 83 f8 fc 74 1a 5b c3 0f 1f 84 00 00 00 00 00 48 8b 44 24 10 0f 05 <5b> c3 0f 1f 80 00 00 00 00 83 e2 39 83 fa 08 75 de e8 23 ff ff RSP:

CVE ID	Description
CVE-2025-38643	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: Add missing lock in cfg80211_check_and_end_cac() Callers of wdev_chandef() must hold the wiphy mutex. But the worker cfg80211_propagate_cac_done_wk() never takes the lock. Which triggers the warning below with the mesh_peer_connected_dfs test from hostapd and not (yet) released mac80211 code changes: WARNING: CPU: 0 PID: 495 at net/wireless/chan.c:1552 wdev_chandef+0x60/0x165 Modules linked in: CPU: 0 UID: 0 PID: 495 Comm: kworker/u4:2 Not tainted 6.14.0-rc5-wt-g03960e6f9d47 #33 13c287eeabfe1efea01c0bcc863723ab082e17cf Workqueue: cfg80211_propagate_cac_done_wk Stack: 00000000 00000001 ffffffff 6093267c 00000000 6002ec30 6d577c50 60037608 00000000 67e8d108 6063717b 00000000 Call Trace: [<6002ec30>] ? _printk+0x0/0x98 [<6003c2b3>] show_stack+0x10e/0x11a [<6002ec30>] ? _printk+0x0/0x98 [<60037608>] dump_stack_lvl+0x71/0xb8 [<6063717b>] ? wdev_chandef+0x60/0x165 [<6003766d>] dump_stack+0x1e/0x20 [<6005d1b7>] __warn+0x101/0x20f [<6005d3a8>] warn_slowpath_fmt+0xe3/0x15d [<600b0c5c>] ? mark_lock.part.0+0x0/0x4ec [<60751191>] ? __this_cpu_preempt_check+0x0/0x16 [<600b11a2>] ? mark_held_locks+0x5a/0x6e [<6005d2c5>] ? warn_slowpath_fmt+0x0/0x15d [<60052e53>] ? unblock_signals+0x3a/0xe7 [<60052f2d>] ? um_set_signals+0x2d/0x43 [<60751191>] ? __this_cpu_preempt_check+0x0/0x16 [<607508b2>] ? lock_is_held_type+0x207/0x21f [<6063717b>] wdev_chandef+0x60/0x165 [<605f89b4>] regulatory_propagate_dfs_state+0x247/0x43f [<60052f00>] ? um_set_signals+0x0/0x43 [<605e6bfd>] cfg80211_propagate_cac_done_wk+0x3a/0x4a [<6007e460>] process_scheduled_works+0x3bc/0x60e [<6007d0ec>] ? move_linked_works+0x4d/0x81 [<6007d120>] ? assign_work+0x0/0xaa [<6007f81f>] worker_thread+0x220/0x2dc [<600786ef>] ? set_pf_worker+0x0/0x57 [<60087c96>] ? to_kthread+0x0/0x43 [<6008ab3c>] kthread+0x2d3/0x2e2 [<6007f5ff>] ? worker_thread+0x0/0x2dc [<6006c05b>] ? calculate_sigpending+0x0/0x56 [<6003b37d>] new_thread_handler+0x4a/0x64 irq event stamp: 614611 hardirqs last enabled at (614621): [<00000000600bc96b>] __up_console_sem+0x82/0xaf hardirqs last disabled at (614630): [<00000000600bc92c>] __up_console_sem+0x43/0xaf softirqs last enabled at (614268): [<00000000606c55c6>] __ieee80211_wake_queue+0x933/0x985 softirqs last disabled at (614266): [<00000000606c52d6>] __ieee80211_wake_queue+0x643/0x985
CVE-2025-38644	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: reject TDLS operations when station is not associated syzbot triggered a WARN in ieee80211_tdls_oper() by sending NL80211_TDLS_ENABLE_LINK immediately after NL80211_CMD_CONNECT, before association completed and without prior TDLS setup. This left internal state like sdata->u.mgd.tdls_peer uninitialized, leading to a WARN_ON() in code paths that assumed it was valid. Reject the operation early if not in station mode or not associated.
CVE-2025-38645	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Check device memory pointer before usage Add a NULL check before accessing device memory to prevent a crash if dev->dm allocation in mlx5_init_once() fails.
CVE-2025-38646	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: avoid NULL dereference when RX problematic packet on unsupported 6 GHz band With a quite rare chance, RX report might be problematic to make SW think a packet is received on 6 GHz band even if the chip does not support 6 GHz band actually. Since SW won't initialize stuffs for unsupported bands, NULL dereference will happen then in the sequence, rtw89_vif_rx_stats_iter() -> rtw89_core_cancel_6ghz_probe_tx(). So, add a check to avoid it. The following is a crash log for this case. BUG: kernel NULL pointer dereference, address: 0000000000000032 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: 0000 [#1] PREEMPT SMP NOPTI CPU: 1 PID: 1907 Comm: irq/131-rtw89_p Tainted: G U 6.6.56-05896-g89f5fb0eb30b #1 (HASH:1400 4) Hardware name: Google Telith/Telith, BIOS Google_Telith.15217.747.0 11/12/2024 RIP: 0010:rtw89_vif_rx_stats_iter+0xd2/0x310 [rtw89_core] Code: 4c 89 7d c8 48 89 55 c0 49 8d 44 24 02 48 89 45 b8 45 31 ff eb 11 41 c6 45 3a 01 41 b7 01 4d 8b 6d 00 4d 39 f5 74 42 8b 43 10 <41> 33 45 32 0f b7 4b 14 66 41 33 4d 36 0f b7 c9 09 c1 74 d8 4d 85 RSP: 0018:ffff9f3080138ca0 EFLAGS: 00010246 RAX: 00000000b8bf5770 RBX: ffff91b5e8c639c0 RCX: 0000000000000011 RDX: ffff91b582de1be8 RSI: 0000000000000000 RDI: ffff91b5e8c639e6 RBP: ffff9f3080138d00 R08: 0000000000000000 R09: 0000000000000000 R10: ffff91b59de70000 R11: ffffffff069be50 R12: ffff91b5e8c639e4 R13: 0000000000000000 R14: ffff91b5828020b8 R15: 0000000000000000 FS: 0000000000000000(0000) GS:ffff91b8efa40000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000032 CR3: 00000002bf838000 CR4: 0000000000750ee0 PKRU: 55555554 Call Trace: <IRQ> ? __die_body+0x68/0xb0 ? page_fault_oops+0x379/0x3e0 ? exc_page_fault+0x4f/0xa0 ? asm_exc_page_fault+0x22/0x30 ? __pfx_rtw89_vif_rx_stats_iter+0x10/0x10 [rtw89_core (HASH:1400 5)] ? rtw89_vif_rx_stats_iter+0xd2/0x310 [rtw89_core (HASH:1400 5)] __iterate_interfaces+0x59/0x110 [mac80211 (HASH:1400 6)] ? __pfx_rtw89_vif_rx_stats_iter+0x10/0x10 [rtw89_core (HASH:1400 5)] ? __pfx_rtw89_vif_rx_stats_iter+0x10/0x10 [rtw89_core (HASH:1400 6)] ieee80211_iterate_active_interfaces_atomic+0x36/0x50 [mac80211 (HASH:1400 6)] rtw89_core_rx_to_mac80211+0xfd/0x1b0 [rtw89_core (HASH:1400 5)] rtw89_core_rx+0x43a/0x980 [rtw89_core (HASH:1400 5)]

CVE ID	Description
CVE-2025-38648	In the Linux kernel, the following vulnerability has been resolved: spi: stm32: Check for cfg availability in stm32_spi_probe The stm32_spi_probe function now includes a check to ensure that the pointer returned by of_device_get_match_data is not NULL before accessing its members. This resolves a warning where a potential NULL pointer dereference could occur when accessing cfg->has_device_mode. Before accessing the 'has_device_mode' member, we verify that 'cfg' is not NULL. If 'cfg' is NULL, an error message is logged. This change ensures that the driver does not attempt to access configuration data if it is not available, thus preventing a potential system crash due to a NULL pointer dereference.
CVE-2025-38650	In the Linux kernel, the following vulnerability has been resolved: hfsplus: remove mutex_lock check in hfsplus_free_extents Syzbot reported an issue in hfsplus filesystem: -----[cut here]----- WARNING: CPU: 0 PID: 4400 at fs/hfsplus/extents.c:346 hfsplus_free_extents+0x700/0xad0 Call Trace: <TASK> hfsplus_file_truncate+0x768/0xbb0 fs/hfsplus/extents.c:606 hfsplus_write_begin+0xc2/0xd0 fs/hfsplus/inode.c:56 cont_expand_zero fs/buffer.c:2383 [inline] cont_write_begin+0x2cf/0x860 fs/buffer.c:2446 hfsplus_write_begin+0x86/0xd0 fs/hfsplus/inode.c:52 generic_cont_expand_simple+0x151/0x250 fs/buffer.c:2347 hfsplus_setattr+0x168/0x280 fs/hfsplus/inode.c:263 notify_change+0xe38/0x10f0 fs/attr.c:420 do_truncate+0x1fb/0x2e0 fs/open.c:65 do_sys_ftruncate+0x2eb/0x380 fs/open.c:193 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x3d/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe +0x63/0xcd To avoid deadlock, Commit 31651c607151 ("hfsplus: avoid deadlock on file truncation") unlock extree before hfsplus_free_extents(), and add check wheather extree is locked in hfsplus_free_extents(). However, when operations such as hfsplus_file_release, hfsplus_setattr, hfsplus_unlink, and hfsplus_get_block are executed concurrently in different files, it is very likely to trigger the WARN_ON, which will lead syzbot and xfstest to consider it as an abnormality. The comment above this warning also describes one of the easy triggering situations, which can easily trigger and cause xfstest&syzbot to report errors. [task A] [task B] ->hfsplus_file_release ->hfsplus_file_truncate ->hfs_find_init ->mutex_lock ->mutex_unlock ->hfsplus_write_begin ->hfsplus_get_block ->hfsplus_file_extend ->hfsplus_ext_read_extent ->hfs_find_init ->mutex_lock ->hfsplus_free_extents WARN_ON(mutex_is_locked) !!! Several threads could try to lock the shared extents tree. And warning can be triggered in one thread when another thread has locked the tree. This is the wrong behavior of the code and we need to remove the warning.
CVE-2025-38652	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid out-of-boundary access in devs.path - touch /mnt/f2fs/01234567890123456789012345678901234567890123 - truncate -s \$(1024*1024*1024)) \ /mnt/f2fs/0123456789012345678901234567890123456789012345678901234567890123 - touch /mnt/f2fs/file - truncate -s \$(1024*1024*1024)) /mnt/f2fs/file - mkfs.f2fs /mnt/f2fs/012345678901234567890123456789012345678901234567890123 \ -c /mnt/f2fs/file - mount /mnt/f2fs/0123456789012345678901234567890123456789012345678901234567890123 \ /mnt/f2fs/loop [16937.192225] F2FS-fs (loop0): Mount Device [0]: /mnt/f2fs/012345678901234567890123456789012345678901234567890123\xffx01, 511, 0 - 3ffff [16937.192268] F2FS-fs (loop0): Failed to find devices If device path length equals to MAX_PATH_LEN, sbi->devs.path[] may not end up w/ null character due to path array is fully filled, So accidentally, fields locate after path[] may be treated as part of device path, result in parsing wrong device path. struct f2fs_dev_info { ... char path[MAX_PATH_LEN]; ... }; Let's add one byte space for sbi->devs.path[] to store null character of device path string.
CVE-2025-38653	In the Linux kernel, the following vulnerability has been resolved: proc: use the same treatment to check proc_lseek as ones for proc_read_iter et.al Check pde->proc_ops->proc_lseek directly may cause UAF in rmmmod scenario. It's a gap in proc_reg_open() after commit 654b33ada4ab("proc: fix UAF in proc_get_inode()"). Followed by AI Viro's suggestion, fix it in same manner.
CVE-2025-38659	In the Linux kernel, the following vulnerability has been resolved: gfs2: No more self recovery When a node withdraws and it turns out that it is the only node that has the filesystem mounted, gfs2 currently tries to replay the local journal to bring the filesystem back into a consistent state. Not only is that a very bad idea, it has also never worked because gfs2_recover_func() will refuse to do anything during a withdraw. However, before even getting to this point, gfs2_recover_func() dereferences sdp->sd_jdesc->jd_inode. This was a use-after-free before commit 04133b607a78 ("gfs2: Prevent double iput for journal on error") and is a NULL pointer dereference since then. Simply get rid of self recovery to fix that.
CVE-2025-38660	In the Linux kernel, the following vulnerability has been resolved: [ceph] parse_longname(): strrchr() expects NUL-terminated string ... and parse_longname() is not guaranteed that. That's the reason why it uses kmemdup_nul() to build the argument for kstrtou64(); the problem is, kstrtou64() is not the only thing that need it. Just get a NUL-terminated copy of the entire thing and be done with that...

CVE ID	Description
CVE-2025-38663	In the Linux kernel, the following vulnerability has been resolved: nilfs2: reject invalid file types when reading inodes To prevent inodes with invalid file types from tripping through the vfs and causing malfunctions or assertion failures, add a missing sanity check when reading an inode from a block device. If the file type is not valid, treat it as a filesystem error.
CVE-2025-38664	In the Linux kernel, the following vulnerability has been resolved: ice: Fix a null pointer dereference in ice_copy_and_init_pkg() Add check for the return value of devm_kmemdup() to prevent potential null pointer dereference.
CVE-2025-38665	In the Linux kernel, the following vulnerability has been resolved: can: netlink: can_changelink(): fix NULL pointer deref of struct can_priv::do_set_mode Andrei Lalaev reported a NULL pointer deref when a CAN device is restarted from Bus Off and the driver does not implement the struct can_priv::do_set_mode callback. There are 2 code path that call struct can_priv::do_set_mode: - directly by a manual restart from the user space, via can_changelink() - delayed automatic restart after bus off (deactivated by default) To prevent the NULL pointer deference, refuse a manual restart or configure the automatic restart delay in can_changelink() and report the error via extack to user space. As an additional safety measure let can_restart() return an error if can_priv::do_set_mode is not set instead of dereferencing it unchecked.
CVE-2025-38666	In the Linux kernel, the following vulnerability has been resolved: net: appletalk: Fix use-after-free in AARP proxy probe The AARP proxy#probe routine (aarp_proxy_probe_network) sends a probe, releases the aarp_lock, sleeps, then re-acquires the lock. During that window an expire timer thread (__aarp_expire_timer) can remove and kfree() the same entry, leading to a use-after-free. race condition: cpu 0 cpu 1 atalk_sendmsg() atif_proxy_probe_device() aarp_send_ddp() aarp_proxy_probe_network() mod_timer() lock(aarp_lock) // LOCK!! timeout around 200ms alloc(aarp_entry) and then call proxies[hash] = aarp_entry aarp_expire_timeout() aarp_send_probe() unlock(aarp_lock) // UNLOCK!! lock(aarp_lock) // LOCK!! msleep(100); __aarp_expire_timer(&proxies[ct]) free(aarp_entry) unlock(aarp_lock) // UNLOCK!! lock(aarp_lock) // LOCK!! UAF aarp_entry !! ===== BUG: KASAN: slab-use-after-free in aarp_proxy_probe_network+0x560/0x630 net/appletalk/aarp.c:493 Read of size 4 at addr ffff8880123aa360 by task repro/13278 CPU: 3 UID: 0 PID: 13278 Comm: repro Not tainted 6.15.2 #3 PREEMPT(full) Call Trace: <TASK> _dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1b0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:408 [inline] print_report+0xc1/0x630 mm/kasan/report.c:521 kasan_report +0xca/0x100 mm/kasan/report.c:634 aarp_proxy_probe_network+0x560/0x630 net/appletalk/aarp.c:493 atif_proxy_probe_device net/appletalk/ddp.c:332 [inline] atif_ioctl+0xb58/0x16c0 net/appletalk/ddp.c:857 atalk_ioctl+0x198/0x2f0 net/appletalk/ddp.c:1818 sock_do_ioctl+0xdc/0x260 net/socket.c:1190 sock_ioctl+0x239/0x6a0 net/socket.c:1311 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:906 [inline] __se_sys_ioctl fs/ioctl.c:892 [inline] __x64_sys_ioctl+0x194/0x200 fs/ioctl.c:892 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xcb/0x250 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> Allocated: aarp_alloc net/appletalk/aarp.c:382 [inline] aarp_proxy_probe_network+0xd8/0x630 net/appletalk/aarp.c:468 atif_proxy_probe_device net/appletalk/ddp.c:332 [inline] atif_ioctl+0xb58/0x16c0 net/appletalk/ddp.c:857 atalk_ioctl+0x198/0x2f0 net/appletalk/ddp.c:1818 Freed: kfree+0x148/0x4d0 mm/slub.c:4841 __aarp_expire net/appletalk/aarp.c:90 [inline] __aarp_expire_timer net/appletalk/aarp.c:261 [inline] aarp_expire_timeout+0x480/0x6e0 net/appletalk/aarp.c:317 The buggy address belongs to the object at ffff8880123aa300 which belongs to the cache kmalloc-192 of size 192 The buggy address is located 96 bytes inside of freed 192-byte region [ffff8880123aa300, ffff8880123aa3c0) Memory state around the buggy address: ffff8880123aa200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ffff8880123aa280: 00 00 00 00 fc fc fc fc fc fc fc fc fc fc fc fc >ffff8880123aa300: fa fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb ^ ffff8880123aa380: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc ffff8880123aa400: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 =====
CVE-2025-38668	In the Linux kernel, the following vulnerability has been resolved: regulator: core: fix NULL dereference on unbind due to stale coupling data Failing to reset coupling_desc.n_coupled after freeing coupled_rdevs can lead to NULL pointer dereference when regulators are accessed post-unbind. This can happen during runtime PM or other regulator operations that rely on coupling metadata. For example, on ridesx4, unbinding the 'reg-dummy' platform device triggers a panic in regulator_lock_recursive() due to stale coupling state. Ensure n_coupled is set to 0 to prevent access to invalid pointers.

CVE ID	Description
CVE-2025-38669	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/gem-shmem: Use dma_buf from GEM object instance" This reverts commit 1a148af06000e545e714fe3210af3d77ff903c11. The dma_buf field in struct drm_gem_object is not stable over the object instance's lifetime. The field becomes NULL when user space releases the final GEM handle on the buffer object. This resulted in a NULL-pointer deref. Workarounds in commit 5307dce878d4 ("drm/gem: Acquire references on GEM handles for framebuffers") and commit f6bfc9afc751 ("drm/framebuffer: Acquire internal references on GEM handles") only solved the problem partially. They especially don't work for buffer objects without a DRM framebuffer associated. Hence, this revert to going back to using .import_attach->dmabuf. v3: - cc stable
CVE-2025-38670	In the Linux kernel, the following vulnerability has been resolved: arm64/entry: Mask DAIF in cpu_switch_to(), call_on_irq_stack() `cpu_switch_to()` and `call_on_irq_stack()` manipulate SP to change to different stacks along with the Shadow Call Stack if it is enabled. Those two stack changes cannot be done atomically and both functions can be interrupted by SErrors or Debug Exceptions which, though unlikely, is very much broken : if interrupted, we can end up with mismatched stacks and Shadow Call Stack leading to clobbered stacks. In `cpu_switch_to()`, it can happen when SP_ELO points to the new task, but x18 stills points to the old task's SCS. When the interrupt handler tries to save the task's SCS pointer, it will save the old task SCS pointer (x18) into the new task struct (pointed to by SP_ELO), clobbering it. In `call_on_irq_stack()`, it can happen when switching from the task stack to the IRQ stack and when switching back. In both cases, we can be interrupted when the SCS pointer points to the IRQ SCS, but SP points to the task stack. The nested interrupt handler pushes its return addresses on the IRQ SCS. It then detects that SP points to the task stack, calls `call_on_irq_stack()` and clobbers the task SCS pointer with the IRQ SCS pointer, which it will also use ! This leads to tasks returning to addresses on the wrong SCS, or even on the IRQ SCS, triggering kernel panics via CONFIG_VMAP_STACK or FPAC if enabled. This is possible on a default config, but unlikely. However, when enabling CONFIG_ARM64_PSEUDO_NMI, DAIF is unmasked and instead the GIC is responsible for filtering what interrupts the CPU should receive based on priority. Given the goal of emulating NMIs, pseudo-NMIs can be received by the CPU even in `cpu_switch_to()` and `call_on_irq_stack()`, possibly *very* frequently depending on the system configuration and workload, leading to unpredictable kernel panics. Completely mask DAIF in `cpu_switch_to()` and restore it when returning. Do the same in `call_on_irq_stack()`, but restore and mask around the branch. Mask DAIF even if CONFIG_SHADOW_CALL_STACK is not enabled for consistency of behaviour between all configurations. Introduce and use an assembly macro for saving and masking DAIF, as the existing one saves but only masks IF.
CVE-2025-38671	In the Linux kernel, the following vulnerability has been resolved: i2c: qup: jump out of the loop in case of timeout Original logic only sets the return value but doesn't jump out of the loop if the bus is kept active by a client. This is not expected. A malicious or buggy i2c client can hang the kernel in this case and should be avoided. This is observed during a long time test with a PCA953x GPIO extender. Fix it by changing the logic to not only sets the return value, but also jumps out of the loop and return to the caller with -ETIMEDOUT.
CVE-2025-38672	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/gem-dma: Use dma_buf from GEM object instance" This reverts commit e8afa1557f4f963c9a511bd2c6074a941c308685. The dma_buf field in struct drm_gem_object is not stable over the object instance's lifetime. The field becomes NULL when user space releases the final GEM handle on the buffer object. This resulted in a NULL-pointer deref. Workarounds in commit 5307dce878d4 ("drm/gem: Acquire references on GEM handles for framebuffers") and commit f6bfc9afc751 ("drm/framebuffer: Acquire internal references on GEM handles") only solved the problem partially. They especially don't work for buffer objects without a DRM framebuffer associated. Hence, this revert to going back to using .import_attach->dmabuf. v3: - cc stable
CVE-2025-38673	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/gem-framebuffer: Use dma_buf from GEM object instance" This reverts commit cce16fcd7446dcff7480cd9d2b6417075ed81065. The dma_buf field in struct drm_gem_object is not stable over the object instance's lifetime. The field becomes NULL when user space releases the final GEM handle on the buffer object. This resulted in a NULL-pointer deref. Workarounds in commit 5307dce878d4 ("drm/gem: Acquire references on GEM handles for framebuffers") and commit f6bfc9afc751 ("drm/framebuffer: Acquire internal references on GEM handles") only solved the problem partially. They especially don't work for buffer objects without a DRM framebuffer associated. Hence, this revert to going back to using .import_attach->dmabuf. v3: - cc stable

CVE ID	Description
CVE-2025-38674	In the Linux kernel, the following vulnerability has been resolved: Revert "drm/prime: Use dma_buf from GEM object instance" This reverts commit f83a9b8c7fd0557b0c50784bfdc1bbe9140c9bf8. The dma_buf field in struct drm_gem_object is not stable over the object instance's lifetime. The field becomes NULL when user space releases the final GEM handle on the buffer object. This resulted in a NULL-pointer deref. Workarounds in commit 5307dce878d4 ("drm/gem: Acquire references on GEM handles for framebuffers") and commit f6bfc9afc751 ("drm/framebuffer: Acquire internal references on GEM handles") only solved the problem partially. They especially don't work for buffer objects without a DRM framebuffer associated. Hence, this revert to going back to using .import_attach->dmabuf. v3: - cc stable
CVE-2025-38675	In the Linux kernel, the following vulnerability has been resolved: xfrm: state: initialize state_ptrs earlier in xfrm_state_find In case of preemption, xfrm_state_look_at will find a different pcpu_id and look up states for that other CPU. If we matched a state for CPU2 in the state_cache while the lookup started on CPU1, we will jump to "found", but the "best" state that we got will be ignored and we will enter the "acquire" block. This block uses state_ptrs, which isn't initialized at this point. Let's initialize state_ptrs just after taking rcu_read_lock. This will also prevent a possible misuse in the future, if someone adjusts this function.
CVE-2025-38676	In the Linux kernel, the following vulnerability has been resolved: iommu/amd: Avoid stack buffer overflow from kernel cmdline While the kernel command line is considered trusted in most environments, avoid writing 1 byte past the end of "acpiid" if the "str" argument is maximum length.
CVE-2025-38677	In the Linux kernel, the following vulnerability has been resolved: f2fs: fix to avoid out-of-boundary access in dnode page As Jiaming Zhang reported: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x1c1/0x2a0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0x17e/0x800 mm/kasan/report.c:480 kasan_report+0x147/0x180 mm/kasan/report.c:593 data_blkaddr fs/f2fs/f2fs.h:3053 [inline] f2fs_data_blkaddr fs/f2fs/f2fs.h:3058 [inline] f2fs_get_dnode_of_data+0x1a09/0x1c40 fs/f2fs/node.c:855 f2fs_reserve_block+0x53/0x310 fs/f2fs/data.c:1195 prepare_write_begin fs/f2fs/data.c:3395 [inline] f2fs_write_begin+0xf39/0x2190 fs/f2fs/data.c:3594 generic_perform_write+0x2c7/0x910 mm/filemap.c:4112 f2fs_buffered_write_iter fs/f2fs/file.c:4988 [inline] f2fs_file_write_iter+0x1ec8/0x2410 fs/f2fs/file.c:5216 new_sync_write fs/read_write.c:593 [inline] vfs_write+0x546/0xa90 fs/read_write.c:686 ksys_write+0x149/0x250 fs/read_write.c:738 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xf3/0x3d0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f The root cause is in the corrupted image, there is a dnode has the same node id w/ its inode, so during f2fs_get_dnode_of_data(), it tries to access block address in dnode at offset 934, however it parses the dnode as inode node, so that get_dnode_addr() returns 360, then it tries to access page address from 360 + 934 * 4 = 4096 w/ 4 bytes. To fix this issue, let's add sanity check for node id of all direct nodes during f2fs_get_dnode_of_data().
CVE-2025-38679	In the Linux kernel, the following vulnerability has been resolved: media: venus: Fix OOB read due to missing payload bound check Currently, The event_seq_changed() handler processes a variable number of properties sent by the firmware. The number of properties is indicated by the firmware and used to iterate over the payload. However, the payload size is not being validated against the actual message length. This can lead to out-of-bounds memory access if the firmware provides a property count that exceeds the data available in the payload. Such a condition can result in kernel crashes or potential information leaks if memory beyond the buffer is accessed. Fix this by properly validating the remaining size of the payload before each property access and updating bounds accordingly as properties are parsed. This ensures that property parsing is safely bounded within the received message buffer and protects against malformed or malicious firmware behavior.
CVE-2025-38680	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Fix 1-byte out-of-bounds read in uvc_parse_format() The buffer length check before calling uvc_parse_format() only ensured that the buffer has at least 3 bytes (buflen > 2), but the function accesses buffer[3], requiring at least 4 bytes. This can lead to an out-of-bounds read if the buffer has exactly 3 bytes. Fix it by checking that the buffer has at least 4 bytes in uvc_parse_format().

CVE ID	Description
CVE-2025-38681	In the Linux kernel, the following vulnerability has been resolved: mm/ptdump: take the memory hotplug lock inside ptdump_walk_pgd() Memory hot remove unmmaps and tears down various kernel page table regions as required. The ptdump code can race with concurrent modifications of the kernel page tables. When leaf entries are modified concurrently, the dump code may log stale or inconsistent information for a VA range, but this is otherwise not harmful. But when intermediate levels of kernel page table are freed, the dump code will continue to use memory that has been freed and potentially reallocated for another purpose. In such cases, the ptdump code may dereference bogus addresses, leading to a number of potential problems. To avoid the above mentioned race condition, platforms such as arm64, riscv and s390 take memory hotplug lock, while dumping kernel page table via the sysfs interface /sys/kernel/debug/kernel_page_tables. Similar race condition exists while checking for pages that might have been marked W+X via /sys/kernel/debug/kernel_page_tables/check_wx_pages which in turn calls ptdump_check_wx(). Instead of solving this race condition again, let's just move the memory hotplug lock inside generic ptdump_check_wx() which will benefit both the scenarios. Drop get_online_mems() and put_online_mems() combination from all existing platform ptdump code paths.
CVE-2025-38683	In the Linux kernel, the following vulnerability has been resolved: hv_netvsc: Fix panic during namespace deletion with VF The existing code move the VF NIC to new namespace when NETDEV_REGISTER is received on netvsc NIC. During deletion of the namespace, default_device_exit_batch() >> default_device_exit_net() is called. When netvsc NIC is moved back and registered to the default namespace, it automatically brings VF NIC back to the default namespace. This will cause the default_device_exit_net() >> for_each_netdev_safe loop unable to detect the list end, and hit NULL ptr: [231.449420] mana 7870:00:00:00 enP30832s1: Moved VF to namespace with: eth0 [231.449656] BUG: kernel NULL pointer dereference, address: 0000000000000010 [231.450246] #PF: supervisor read access in kernel mode [231.450579] #PF: error_code(0x0000) - not-present page [231.450916] PGD 17b8a8067 P4D 0 [231.451163] Oops: Oops: 0000 [#1] SMP NOPTI [231.451450] CPU: 82 UID: 0 PID: 1394 Comm: kworker/u768:1 Not tainted 6.16.0-rc4+ #3 VOLUNTARY [231.452042] Hardware name: Microsoft Corporation Virtual Machine/Virtual Machine, BIOS Hyper-V UEFI Release v4.1 11/21/2024 [231.452692] Workqueue: netns cleanup_net [231.452947] RIP: 0010:default_device_exit_batch+0x16c/0x3f0 [231.453326] Code: c0 0c f5 b3 e8 d5 db fe ff 48 85 c0 74 15 48 c7 c2 f8 fd ca b2 be 10 00 00 00 48 8d 7d c0 e8 7b 77 25 00 49 8b 86 28 01 00 00 <48> 8b 50 10 4c 8b 2a 4c 8d 62 f0 49 83 ed 10 4c 39 e0 0f 84 d6 00 [231.454294] RSP: 0018:ff75fc7c9bf9fd00 EFLAGS: 00010246 [231.454610] RAX: 0000000000000000 RBX: 0000000000000002 RCX: 61c8864680b583eb [231.455094] RDX: ff1fa9f71462d800 RSI: ff75fc7c9bf9fd38 RDI: 0000000030766564 [231.455686] RBP: ff75fc7c9bf9fd78 R08: 0000000000000000 R09: 0000000000000000 [231.456126] R10: 0000000000000001 R11: 0000000000000004 R12: ff1fa9f70088e340 [231.456621] R13: ff1fa9f70088e340 R14: ffffffff3f50c20 R15: ff1fa9f7103e6340 [231.457161] FS: 0000000000000000(0000) GS:ff1faa6783a08000(0000) knlGS:0000000000000000 [231.457707] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [231.458031] CR2: 0000000000000010 CR3: 0000000179ab2006 CR4: 0000000000b73ef0 [231.458434] Call Trace: [231.458600] <TASK> [231.458777] ops_undo_list+0x100/0x220 [231.459015] cleanup_net+0x1b8/0x300 [231.459285] process_one_work+0x184/0x340 To fix it, move the ns change to a workqueue, and take rtnl_lock to avoid changing the netdev list when default_device_exit_net() is using it.

CVE ID	Description
CVE-2025-38684	In the Linux kernel, the following vulnerability has been resolved: net/sched: ets: use old 'nbands' while purging unused classes Shuang reported sch_ets test-case [1] crashing in ets_class_qlen_notify() after recent changes from Lion [2]. The problem is: in ets_qdisc_change() we purge unused DWRR queues; the value of 'q->nbands' is the new one, and the cleanup should be done with the old one. The problem is here since my first attempts to fix ets_qdisc_change(), but it surfaced again after the recent qdisc len accounting fixes. Fix it purging idle DWRR queues before assigning a new value of 'q->nbands', so that all purge operations find a consistent configuration: - old 'q->nbands' because it's needed by ets_class_find() - old 'q->nstrict' because it's needed by ets_class_is_strict() BUG: kernel NULL pointer dereference, address: 0000000000000000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 0 P4D 0 Oops: Oops: 0000 [#1] SMP NOPTI CPU: 62 UID: 0 PID: 39457 Comm: tc Kdump: loaded Not tainted 6.12.0-116.el10.x86_64 #1 PREEMPT(voluntary) Hardware name: Dell Inc. PowerEdge R640/06DKY5, BIOS 2.12.2 07/09/2021 RIP: 0010: __list_del_entry_valid_or_report+0x4/0x80 Code: ff 4c 39 c7 0f 84 39 19 8e ff b8 01 00 00 00 c3 cc cc cc cc 66 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e fa <48> 8b 17 48 8b 4f 08 48 85 d2 0f 84 56 19 8e ff 48 85 c9 0f 84 ab RSP: 0018:ffffba186009f400 EFLAGS: 00010202 RAX: 00000000000000d6 RBX: 0000000000000000 RCX: 0000000000000004 RDX: ffff9f0fa29b69c0 RSI: 0000000000000000 RDI: 0000000000000000 RBP: ffffffff12c2400 R08: 0000000000000008 R09: 0000000000000004 R10: ffffffff00000000 R11: 0000000000000004 R12: 0000000000000000 R13: ffff9f0f8cfe0000 R14: 0000000000100005 R15: 0000000000000000 FS: 00007f2154f37480(0000) GS:ffff9f269c1c0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000000000000000 CR3: 00000001530be001 CR4: 00000000007726f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 PKRU: 55555554 Call Trace: <TASK> ets_class_qlen_notify+0x65/0x90 [sch_ets] qdisc_tree_reduce_backlog+0x74/0x110 ets_qdisc_change+0x630/0xa40 [sch_ets] __tc_modify_qdisc.constprop.0+0x216/0xf70 tc_modify_qdisc+0x7c/0x120 rtnetlink_rcv_msg+0x145/0x3f0 netlink_rcv_skb+0x53/0x100 netlink_unicast+0x245/0x390 netlink_sendmsg+0x21b/0x470 ____sys_sendmsg+0x39d/0x3d0 ____sys_sendmsg+0x9a/0xe0 __sys_sendmsg+0x7a/0xd0 do_syscall_64+0x7d/0x160 entry_SYSCALL_64_after_hwframe+0x76/0x7e RIP: 0033:0x7f2155114084 Code: 89 02 b8 ff ff ff ff eb bb 66 2e 0f 1f 84 00 00 00 00 0f 1f 00 f3 0f 1e fa 80 3d 25 f0 0c 00 00 74 13 b8 2e 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 54 c3 0f 1f 00 48 83 ec 28 89 54 24 1c 48 89 RSP: 002b:00007fff1fd7a988 EFLAGS: 00000202 ORIG_RAX: 000000000000002e RAX: ffffffff00000000 RBX: 0000560ec063e5e0 RCX: 00007f2155114084 RDX: 0000000000000000 RSI: 00007fff1fd7a9f0 RDI: 0000000000000003 RBP: 00007fff1fd7aa60 R08: 0000000000000010 R09: 000000000000003f R10: 0000560ee9b3a010 R11: 0000000000000202 R12: 00007fff1fd7aae0 R13: 000000006891ccde R14: 0000560ec063e5e0 R15: 00007fff1fd7aad0 <TASK> [1] https://lore.kernel.org/netdev/e08c7f4a6882f260011909a868311c6e9b54f3e4.1639153474.git.dcaratti@redhat.com/ [2] https://lore.kernel.org/netdev/d912cbd7-193b-4269-9857-525bee8bbb6a@gmail.com/
CVE-2025-38685	In the Linux kernel, the following vulnerability has been resolved: fbdev: Fix vmalloc out-of-bounds write in fast_imageblit This issue triggers when a userspace program does an ioctl FBIOPUT_CON2FBMAP by passing console number and frame buffer number. Ideally this maps console to frame buffer and updates the screen if console is visible. As part of mapping it has to do resize of console according to frame buffer info. if this resize fails and returns from vc_do_resize() and continues further. At this point console and new frame buffer are mapped and sets display vars. Despite failure still it continue to proceed updating the screen at later stages where vc_data is related to previous frame buffer and frame buffer info and display vars are mapped to new frame buffer and eventually leading to out-of-bounds write in fast_imageblit(). This bheaviour is excepted only when fg_console is equal to requested console which is a visible console and updates screen with invalid struct references in fbcon_putcs().
CVE-2025-38686	In the Linux kernel, the following vulnerability has been resolved: userfaultfd: fix a crash in UFFDIO_MOVE when PMD is a migration entry When UFFDIO_MOVE encounters a migration PMD entry, it proceeds with obtaining a folio and accessing it even though the entry is swp_entry_t. Add the missing check and let split_huge_pmd() handle migration entries. While at it also remove unnecessary folio check. [surenb@google.com: remove extra folio check, per David]

CVE ID	Description
CVE-2025-38687	In the Linux kernel, the following vulnerability has been resolved: comedi: fix race between polling and detaching syzbot reports a use-after-free in comedi in the below link, which is due to comedi gladly removing the allocated async area even though poll requests are still active on the wait_queue_head inside of it. This can cause a use-after-free when the poll entries are later triggered or removed, as the memory for the wait_queue_head has been freed. We need to check there are no tasks queued on any of the subdevices' wait queues before allowing the device to be detached by the `COMEDI_DEVCONFIG` ioctl. Tasks will read-lock `dev->attach_lock` before adding themselves to the subdevice wait queue, so fix the problem in the `COMEDI_DEVCONFIG` ioctl handler by write-locking `dev->attach_lock` before checking that all of the subdevices are safe to be deleted. This includes testing for any sleepers on the subdevices' wait queues. It remains locked until the device has been detached. This requires the `comedi_device_detach()` function to be refactored slightly, moving the bulk of it into new function `comedi_device_detach_locked()`. Note that the refactor of `comedi_device_detach()` results in `comedi_device_cancel_all()` now being called while `dev->attach_lock` is write-locked, which wasn't the case previously, but that does not matter. Thanks to Jens Axboe for diagnosing the problem and co-developing this patch.
CVE-2025-38688	In the Linux kernel, the following vulnerability has been resolved: iommufd: Prevent ALIGN() overflow When allocating IOVA the candidate range gets aligned to the target alignment. If the range is close to ULONG_MAX then the ALIGN() can wrap resulting in a corrupted iova. Open code the ALIGN() using get_add_overflow() to prevent this. This simplifies the checks as we don't need to check for length earlier either. Consolidate the two copies of this code under a single helper. This bug would allow userspace to create a mapping that overlaps with some other mapping or a reserved range.
CVE-2025-38689	In the Linux kernel, the following vulnerability has been resolved: x86/fpu: Fix NULL dereference in avx512_status() Problem ----- With CONFIG_X86_DEBUG_FPU enabled, reading /proc/[kthread]/arch_status causes a warning and a NULL pointer dereference. This is because the AVX-512 timestamp code uses x86_task_fpu() but doesn't check it for NULL. CONFIG_X86_DEBUG_FPU adds that function for kernel threads (PF_KTHREAD specifically), making it return NULL. The point of the warning was to ensure that kernel threads only access task->fpu after going through kernel_fpu_begin()/_end(). Note: all kernel tasks exposed in /proc have a valid task->fpu. Solution ----- One option is to silence the warning and check for NULL from x86_task_fpu(). However, that warning is fairly fresh and seems like a defense against misuse of the FPU state in kernel threads. Instead, stop outputting AVX-512_elapsed_ms for kernel threads altogether. The data was garbage anyway because avx512_timestamp is only updated for user threads, not kernel threads. If anyone ever wants to track kernel thread AVX-512 use, they can come back later and do it properly, separate from this bug fix. [dhansen: mostly rewrite changelog]
CVE-2025-38691	In the Linux kernel, the following vulnerability has been resolved: pNFS: Fix uninit ptr deref in block/scsi layout The error occurs on the third attempt to encode extents. When function ext_tree_prepare_commit() reallocates a larger buffer to retry encoding extents, the "layoutupdate_pages" page array is initialized only after the retry loop. But ext_tree_free_commitdata() is called on every iteration and tries to put pages in the array, thus dereferencing uninitialized pointers. An additional problem is that there is no limit on the maximum possible buffer_size. When there are too many extents, the client may create a layoutcommit that is larger than the maximum possible RPC size accepted by the server. During testing, we observed two typical scenarios. First, one memory page for extents is enough when we work with small files, append data to the end of the file, or preallocate extents before writing. But when we fill a new large file without preallocating, the number of extents can be huge, and counting the number of written extents in ext_tree_encode_commit() does not help much. Since this number increases even more between unlocking and locking of ext_tree, the reallocated buffer may not be large enough again and again.
CVE-2025-38692	In the Linux kernel, the following vulnerability has been resolved: exfat: add cluster chain loop check for dir An infinite loop may occur if the following conditions occur due to file system corruption. (1) Condition for exfat_count_dir_entries() to loop infinitely. - The cluster chain includes a loop. - There is no UNUSED entry in the cluster chain. (2) Condition for exfat_create_upcase_table() to loop infinitely. - The cluster chain of the root directory includes a loop. - There are no UNUSED entry and up-case table entry in the cluster chain of the root directory. (3) Condition for exfat_load_bitmap() to loop infinitely. - The cluster chain of the root directory includes a loop. - There are no UNUSED entry and bitmap entry in the cluster chain of the root directory. (4) Condition for exfat_find_dir_entry() to loop infinitely. - The cluster chain includes a loop. - The unused directory entries were exhausted by some operation. (5) Condition for exfat_check_dir_empty() to loop infinitely. - The cluster chain includes a loop. - The unused directory entries were exhausted by some operation. - All files and sub-directories under the directory are deleted. This commit adds checks to break the above infinite loop.

CVE ID	Description
CVE-2025-38693	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: w7090p: fix null-ptr-deref in w7090p_tuner_write_serpar and w7090p_tuner_read_serpar In w7090p_tuner_write_serpar, msg is controlled by user. When msg[0].buf is null and msg[0].len is zero, former checks on msg[0].buf would be passed. If accessing msg[0].buf[2] without sanity check, null pointer deref would happen. We add check on msg[0].len to prevent crash. Similar commit: commit 0ed554fd769a ("media: dvb-usb: az6027: fix null-ptr-deref in az6027_i2c_xfer()")
CVE-2025-38694	In the Linux kernel, the following vulnerability has been resolved: media: dvb-frontends: dib7090p: fix null-ptr-deref in dib7090p_rw_on_apb() In dib7090p_rw_on_apb, msg is controlled by user. When msg[0].buf is null and msg[0].len is zero, former checks on msg[0].buf would be passed. If accessing msg[0].buf[2] without sanity check, null pointer deref would happen. We add check on msg[0].len to prevent crash. Similar issue occurs when access msg[1].buf[0] and msg[1].buf[1]. Similar commit: commit 0ed554fd769a ("media: dvb-usb: az6027: fix null-ptr-deref in az6027_i2c_xfer()")
CVE-2025-38695	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Check for hdwq null ptr when cleaning up lpfc_vport structure If a call to lpfc_sli4_read_rev() from lpfc_sli4_hba_setup() fails, the resultant cleanup routine lpfc_sli4_vport_delete_fcp_xri_aborted() may occur before sli4_hba.hdwqs are allocated. This may result in a null pointer dereference when attempting to take the abts_io_buf_list_lock for the first hardware queue. Fix by adding a null ptr check on phba->sli4_hba.hdwq and early return because this situation means there must have been an error during port initialization.
CVE-2025-38696	In the Linux kernel, the following vulnerability has been resolved: MIPS: Don't crash in stack_top() for tasks without ABI or vDSO Not all tasks have an ABI associated or vDSO mapped, for example kthreads never do. If such a task ever ends up calling stack_top(), it will dereference the NULL ABI pointer and crash. This can for example happen when using kunit: mips_stack_top+0x28/0xc0 arch_pick_mmap_layout+0x190/0x220 kunit_vm_mmap_init+0xf8/0x138 __kunit_add_resource+0x40/0xa8 kunit_vm_mmap+0x88/0xd8 usercopy_test_init+0xb8/0x240 kunit_try_run_case+0x5c/0x1a8 kunit_generic_run_threadfn_adapter+0x28/0x50 kthread+0x118/0x240 ret_from_kernel_thread+0x14/0x1c Only dereference the ABI point if it is set. The GIC page is also included as it is specific to the vDSO. Also move the randomization adjustment into the same conditional.
CVE-2025-38697	In the Linux kernel, the following vulnerability has been resolved: jfs: upper bound check of tree index in dbAllocAG When computing the tree index in dbAllocAG, we never check if we are out of bounds relative to the size of the stree. This could happen in a scenario where the filesystem metadata are corrupted.
CVE-2025-38698	In the Linux kernel, the following vulnerability has been resolved: jfs: Regular file corruption check The reproducer builds a corrupted file on disk with a negative i_size value. Add a check when opening this file to avoid subsequent operation failures.
CVE-2025-38699	In the Linux kernel, the following vulnerability has been resolved: scsi: bfa: Double-free fix When the bfad_im_probe() function fails during initialization, the memory pointed to by bfad->im is freed without setting bfad->im to NULL. Subsequently, during driver uninstallation, when the state machine enters the bfad_sm_stopping state and calls the bfad_im_probe_undo() function, it attempts to free the memory pointed to by bfad->im again, thereby triggering a double-free vulnerability. Set bfad->im to NULL if probing fails.
CVE-2025-38700	In the Linux kernel, the following vulnerability has been resolved: scsi: libiscsi: Initialize iscsi_conn->dd_data only if memory is allocated In case of an ib_fast_reg_mr allocation failure during iSER setup, the machine hits a panic because iscsi_conn->dd_data is initialized unconditionally, even when no memory is allocated (dd_size == 0). This leads invalid pointer dereference during connection teardown. Fix by setting iscsi_conn->dd_data only if memory is actually allocated. Panic trace: ----- iser: iser_create_fastreg_desc: Failed to allocate ib_fast_reg_mr err=-12 iser: iser_alloc_rx_descriptors: failed allocating rx descriptors / data buffers BUG: unable to handle page fault for address: ffffffff80000000 RIP: 0010:swake_up_locked.part.5+0xa/0x40 Call Trace: complete+0x31/0x40 iscsi_iscsi_conn_stop+0x88/0xb0 [ib_iscsi] iscsi_stop_conn+0x66/0xc0 [scsi_transport_iscsi] iscsi_if_stop_conn+0x14a/0x150 [scsi_transport_iscsi] iscsi_if_rx+0x1135/0x1834 [scsi_transport_iscsi] ? netlink_lookup+0x12f/0x1b0 ? netlink_deliver_tap+0x2c/0x200 netlink_unicast+0x1ab/0x280 netlink_sendmsg+0x257/0x4f0 ? _copy_from_user+0x29/0x60 sock_sendmsg+0x5f/0x70
CVE-2025-38701	In the Linux kernel, the following vulnerability has been resolved: ext4: do not BUG when INLINE_DATA_FL lacks system.data xattr A syzbot fuzzed image triggered a BUG_ON in ext4_update_inline_data() when an inode had the INLINE_DATA_FL flag set but was missing the system.data extended attribute. Since this can happen due to a maliciously fuzzed file system, we shouldn't BUG, but rather, report it as a corrupted file system. Add similar replacements of BUG_ON with EXT4_ERROR_INODE() in ext4_create_inline_data() and ext4_inline_data_truncate().

CVE ID	Description
CVE-2025-38702	In the Linux kernel, the following vulnerability has been resolved: fbdev: fix potential buffer overflow in do_register_framebuffer() The current implementation may lead to buffer overflow when: 1. Unregistration creates NULL gaps in registered_fb[] 2. All array slots become occupied despite num_registered_fb < FB_MAX 3. The registration loop exceeds array bounds Add boundary check to prevent registered_fb[FB_MAX] access.
CVE-2025-38703	In the Linux kernel, the following vulnerability has been resolved: drm/xe: Make dma-fences compliant with the safe access rules Xe can free some of the data pointed to by the dma-fences it exports. Most notably the timeline name can get freed if userspace closes the associated submit queue. At the same time the fence could have been exported to a third party (for example a sync_fence fd) which will then cause an use- after-free on subsequent access. To make this safe we need to make the driver compliant with the newly documented dma-fence rules. Driver has to ensure a RCU grace period between signalling a fence and freeing any data pointed to by said fence. For the timeline name we simply make the queue be freed via kfree_rcu and for the shared lock associated with multiple queues we add a RCU grace period before freeing the per GT structure holding the lock.
CVE-2025-38704	In the Linux kernel, the following vulnerability has been resolved: rcu/nocb: Fix possible invalid rdp's->nocb_cb_kthread pointer access In the preparation stage of CPU online, if the corresponding the rdp's->nocb_cb_kthread does not exist, will be created, there is a situation where the rdp's rcuop kthreads creation fails, and then de-offload this CPU's rdp, does not assign this CPU's rdp->nocb_cb_kthread pointer, but this rdp's->nocb_gp_rdp and rdp's->rdp_gp->nocb_gp_kthread is still valid. This will cause the subsequent re-offload operation of this offline CPU, which will pass the conditional check and the kthread_unpark() will access invalid rdp's->nocb_cb_kthread pointer. This commit therefore use rdp's->nocb_gp_kthread instead of rdp_gp's->nocb_gp_kthread for safety check.
CVE-2025-38705	In the Linux kernel, the following vulnerability has been resolved: drm/amd/pm: fix null pointer access Writing a string without delimiters (' ', '\n', '\0') to the under gpu_od/fan_ctrl sysfs or pp_power_profile_mode for the CUSTOM profile will result in a null pointer dereference.
CVE-2025-38706	In the Linux kernel, the following vulnerability has been resolved: ASoC: core: Check for rtd == NULL in snd_soc_remove_pcm_runtime() snd_soc_remove_pcm_runtime() might be called with rtd == NULL which will leads to null pointer dereference. This was reproduced with topology loading and marking a link as ignore due to missing hardware component on the system. On module removal the soc_tplg_remove_link() would call snd_soc_remove_pcm_runtime() with rtd == NULL since the link was ignored, no runtime was created.
CVE-2025-38707	In the Linux kernel, the following vulnerability has been resolved: fs/ntfs3: Add sanity check for file name The length of the file name should be smaller than the directory entry size.
CVE-2025-38708	In the Linux kernel, the following vulnerability has been resolved: drbd: add missing kref_get in handle_write_conflicts With `two-primaries` enabled, DRBD tries to detect "concurrent" writes and handle write conflicts, so that even if you write to the same sector simultaneously on both nodes, they end up with the identical data once the writes are completed. In handling "superseeded" writes, we forgot a kref_get, resulting in a premature drbd_destroy_device and use after free, and further to kernel crashes with symptoms. Relevance: No one should use DRBD as a random data generator, and apparently all users of "two-primaries" handle concurrent writes correctly on layer up. That is cluster file systems use some distributed lock manager, and live migration in virtualization environments stops writes on one node before starting writes on the other node. Which means that other than for "test cases", this code path is never taken in real life. FYI, in DRBD 9, things are handled differently nowadays. We still detect "write conflicts", but no longer try to be smart about them. We decided to disconnect hard instead: upper layers must not submit concurrent writes. If they do, that's their fault.
CVE-2025-38709	In the Linux kernel, the following vulnerability has been resolved: loop: Avoid updating block size under exclusive owner Syzbot came up with a reproducer where a loop device block size is changed underneath a mounted filesystem. This causes a mismatch between the block device block size and the block size stored in the superbblock causing confusion in various places such as fs/buffer.c. The particular issue triggered by syzbot was a warning in __getblk_slow() due to requested buffer size not matching block device block size. Fix the problem by getting exclusive hold of the loop device to change its block size. This fails if somebody (such as filesystem) has already an exclusive ownership of the block device and thus prevents modifying the loop device under some exclusive owner which doesn't expect it.
CVE-2025-38710	In the Linux kernel, the following vulnerability has been resolved: gfs2: Validate i_depth for exhash directories A fuzzer test introduced corruption that ends up with a depth of 0 in dir_e_read(), causing an undefined shift by 32 at: index = hash >> (32 - dip->i_depth); As calculated in an open-coded way in dir_make_exhash(), the minimum depth for an exhash directory is ilog2(sdp->sd_hash_ptrs) and 0 is invalid as sdp->sd_hash_ptrs is fixed as sdp->bsize / 16 at mount time. So we can avoid the undefined behaviour by checking for depth values lower than the minimum in gfs2_dinode_in(). Values greater than the maximum are already being checked for there. Also switch the calculation in dir_make_exhash() to use ilog2() to clarify how the depth is calculated. Tested with the syzkaller repro.c and xfstests '-g quick'.

CVE ID	Description
CVE-2025-38711	In the Linux kernel, the following vulnerability has been resolved: smb/server: avoid deadlock when linking with ReplaceIfExists If smb2_create_link() is called with ReplaceIfExists set and the name does exist then a deadlock will happen. ksmbd_vfs_kern_path_locked() will return with success and the parent directory will be locked. ksmbd_vfs_remove_file() will then remove the file. ksmbd_vfs_link() will then be called while the parent is still locked. It will try to lock the same parent and will deadlock. This patch moves the ksmbd_vfs_kern_path_unlock() call to *before* ksmbd_vfs_link() and then simplifies the code, removing the file_present flag variable.
CVE-2025-38712	In the Linux kernel, the following vulnerability has been resolved: hfsplus: don't use BUG_ON() in hfsplus_create_attributes_file() When the volume header contains erroneous values that do not reflect the actual state of the filesystem, hfsplus_fill_super() assumes that the attributes file is not yet created, which later results in hitting BUG_ON() when hfsplus_create_attributes_file() is called. Replace this BUG_ON() with -EIO error with a message to suggest running fsck tool.
CVE-2025-38713	In the Linux kernel, the following vulnerability has been resolved: hfsplus: fix slab-out-of-bounds read in hfsplus_uni2asc() The hfsplus_readdir() method is capable to crash by calling hfsplus_uni2asc(): [667.121659][T9805] ===== [667.122651][T9805] BUG: KASAN: slab-out-of-bounds in hfsplus_uni2asc+0x902/0xa10 [667.123627][T9805] Read of size 2 at addr ffff88802592f40c by task repro/9805 [667.124578] [T9805] [667.124876][T9805] CPU: 3 UID: 0 PID: 9805 Comm: repro Not tainted 6.16.0-rc3 #1 PREEMPT(full) [667.124886][T9805] Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [667.124890][T9805] Call Trace: [667.124893][T9805] <TASK> [667.124896][T9805] dump_stack_lvl+0x10e/0x1f0 [667.124911][T9805] print_report+0xd0/0x660 [667.124920][T9805] ? __virt_addr_valid+0x81/0x610 [667.124928][T9805] ? __phys_addr+0xe8/0x180 [667.124934][T9805] ? hfsplus_uni2asc+0x902/0xa10 [667.124942][T9805] kasan_report+0xc6/0x100 [667.124950] [T9805] ? hfsplus_uni2asc+0x902/0xa10 [667.124959][T9805] hfsplus_uni2asc+0x902/0xa10 [667.124966][T9805] ? hfsplus_bnode_read+0x14b/0x360 [667.124974][T9805] hfsplus_readdir+0x845/0xfc0 [667.124984][T9805] ? __pfx_hfsplus_readdir+0x10/0x10 [667.124994][T9805] ? stack_trace_save+0x8e/0xc0 [667.125008][T9805] ? iterate_dir+0x18b/0xb20 [667.125015][T9805] ? trace_lock_acquire+0x85/0xd0 [667.125022][T9805] ? lock_acquire+0x30/0x80 [667.125029] [T9805] ? iterate_dir+0x18b/0xb20 [667.125037][T9805] ? down_read_killable+0x1ed/0x4c0 [667.125044][T9805] ? putname+0x154/0x1a0 [667.125051][T9805] ? __pfx_down_read_killable+0x10/0x10 [667.125058][T9805] ? apparmor_file_permission+0x239/0x3e0 [667.125069] [T9805] iterate_dir+0x296/0xb20 [667.125076][T9805] __x64_sys_getdents64+0x13c/0x2c0 [667.125084][T9805] ? __pfx__x64_sys_getdents64+0x10/0x10 [667.125091][T9805] ? __x64_sys_openat+0x141/0x200 [667.125126][T9805] ? __pfx_filldir64+0x10/0x10 [667.125134] [T9805] ? do_user_addr_fault+0x7fe/0x12f0 [667.125143][T9805] do_syscall_64+0xc9/0x480 [667.125151][T9805] entry_SYSCALL_64_after_hwframe+0x77/0x7f [667.125158][T9805] RIP: 0033:0x7fa8753b2fc9 [667.125164][T9805] Code: 00 c3 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 44 00 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 48 [667.125172][T9805] RSP: 002b:00007ffe96f8e0f8 EFLAGS: 00000217 ORIG_RAX: 00000000000000d9 [667.125181][T9805] RAX: ffffffff88802592f40c RBX: 0000000000000000 RCX: 00007fa8753b2fc9 [667.125185][T9805] RDX: 0000000000000040 RSI: 000020000000063c RDI: 0000000000000004 [667.125190][T9805] RBP: 00007ffe96f8e110 R08: 00007ffe96f8e110 R09: 00007ffe96f8e110 [667.125195][T9805] R10: 0000000000000000 R11: 0000000000000217 R12: 0000556b1e3b4260 [667.125199][T9805] R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 [667.125207][T9805] </TASK> [667.125210][T9805] [667.145632][T9805] Allocated by task 9805: [667.145991] [T9805] kasan_save_stack+0x20/0x40 [667.146352][T9805] kasan_save_track+0x14/0x30 [667.146717][T9805] __kasan_kmalloc+0xaa/0xb0 [667.147065][T9805] __kmalloc_noprof+0x205/0x550 [667.147448][T9805] hfsplus_find_init+0x95/0x1f0 [667.147813][T9805] hfsplus_readdir+0x220/0xfc0 [667.148174][T9805] iterate_dir+0x296/0xb20 [667.148549][T9805] __x64_sys_getdents64+0x13c/0x2c0 [667.148937][T9805] do_syscall_64+0xc9/0x480 [667.149291] [T9805] entry_SYSCALL_64_after_hwframe+0x77/0x7f [667.149809][T9805] [667.150030] [T9805] The buggy address belongs to the object at ffff88802592f000 [667.150030][T9805] which belongs to the cache kmalloc-2k of size 2048 [667.151282][T9805] The buggy address is located 0 bytes to the right of [667.151282][T9805] allocated 1036-byte region [ffff88802592f000, ffff88802592f40c) [667.1 ---truncated---

CVE ID	Description
CVE-2025-38714	In the Linux kernel, the following vulnerability has been resolved: hfsplus: fix slab-out-of-bounds in hfsplus_bnode_read() The hfsplus_bnode_read() method can trigger the issue: [174.852007] [T9784] ===== [174.852709][T9784] BUG: KASAN: slab-out-of-bounds in hfsplus_bnode_read+0x2f4/0x360 [174.853412][T9784] Read of size 8 at addr ffff88810b5fc6c0 by task repro/9784 [174.854059] [T9784] [174.854272][T9784] CPU: 1 UID: 0 PID: 9784 Comm: repro Not tainted 6.16.0-rc3 #7 PREEMPT(full) [174.854281][T9784] Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [174.854286][T9784] Call Trace: [174.854289] [T9784] <TASK> [174.854292][T9784] dump_stack_lvl+0x10e/0x1f0 [174.854305][T9784] print_report+0xd0/0x660 [174.854315][T9784] ? __virt_addr_valid+0x81/0x610 [174.854323] [T9784] ? __phys_addr+0xe8/0x180 [174.854330][T9784] ? hfsplus_bnode_read+0x2f4/0x360 [174.854337][T9784] kasan_report+0xc6/0x100 [174.854346][T9784] ? hfsplus_bnode_read+0x2f4/0x360 [174.854354][T9784] hfsplus_bnode_read+0x2f4/0x360 [174.854362][T9784] hfsplus_bnode_dump+0x2ec/0x380 [174.854370][T9784] ? __pfx_hfsplus_bnode_dump+0x10/0x10 [174.854377][T9784] ? hfsplus_bnode_write_u16+0x83/0xb0 [174.854385][T9784] ? srcu_gp_start+0xd0/0x310 [174.854393][T9784] ? __mark_inode_dirty+0x29e/0xe40 [174.854402] [T9784] hfsplus_brec_remove+0x3d2/0x4e0 [174.854411][T9784] __hfsplus_delete_attr+0x290/0x3a0 [174.854419][T9784] ? __pfx_hfs_find_1st_rec_by_cnid+0x10/0x10 [174.854427] [T9784] ? __pfx__hfsplus_delete_attr+0x10/0x10 [174.854436][T9784] ? __asan_memset+0x23/0x50 [174.854450][T9784] hfsplus_delete_all_attrs+0x262/0x320 [174.854459][T9784] ? __pfx_hfsplus_delete_all_attrs+0x10/0x10 [174.854469][T9784] ? rcu_is_watching+0x12/0xc0 [174.854476][T9784] ? __mark_inode_dirty+0x29e/0xe40 [174.854483][T9784] hfsplus_delete_cat+0x845/0xde0 [174.854493][T9784] ? __pfx_hfsplus_delete_cat+0x10/0x10 [174.854507] [T9784] hfsplus_unlink+0x1ca/0x7c0 [174.854516][T9784] ? __pfx_hfsplus_unlink+0x10/0x10 [174.854525][T9784] ? down_write+0x148/0x200 [174.854532][T9784] ? __pfx_down_write+0x10/0x10 [174.854540][T9784] vfs_unlink+0x2fe/0x9b0 [174.854549][T9784] do_unlinkat+0x490/0x670 [174.854557][T9784] ? __pfx_do_unlinkat+0x10/0x10 [174.854565][T9784] ? __might_fault+0x9bc/0x130 [174.854576][T9784] ? getname_flags.part.0+0x1c5/0x550 [174.854584] [T9784] __x64_sys_unlink+0xc5/0x110 [174.854592][T9784] do_syscall_64+0xc9/0x480 [174.854600][T9784] entry_SYSCALL_64_after_hwframe+0x77/0x7f [174.854608][T9784] RIP: 0033:0x7f6fdf4c3167 [174.854614][T9784] Code: f0 ff ff 73 01 c3 48 8b 0d 26 0d 0e 00 f7 d8 64 89 01 48 83 c8 ff c3 66 2e 0f 1f 84 00 00 00 00 00 [174.854622][T9784] RSP: 002b:00007ffcb948bca8 EFLAGS: 00000206 ORIG_RAX: 0000000000000057 [174.854630][T9784] RAX: ffffffffdfda RBX: 0000000000000000 RCX: 00007f6fdf4c3167 [174.854636][T9784] RDX: 00007ffcb948bcc0 RSI: 00007ffcb948bcc0 RDI: 00007ffcb948bd50 [174.854641][T9784] RBP: 00007ffcb948cd90 R08: 0000000000000001 R09: 00007ffcb948bb40 [174.854645][T9784] R10: 00007f6fdf564fc0 R11: 0000000000000206 R12: 0000561e1bc9c2d0 [174.854650][T9784] R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000 [174.854658][T9784] </TASK> [174.854661][T9784] [174.879281][T9784] Allocated by task 9784: [174.879664][T9784] kasan_save_stack+0x20/0x40 [174.880082][T9784] kasan_save_track+0x14/0x30 [174.880500][T9784] __kasan_kmalloc+0xaa/0xb0 [174.880908][T9784] __kmalloc_noprof+0x205/0x550 [174.881337][T9784] __hfs_bnode_create+0x107/0x890 [174.881779][T9784] hfsplus_bnode_find+0x2d0/0xd10 [174.882222][T9784] hfsplus_brec_find+0x2b0/0x520 [174.882659][T9784] hfsplus_delete_all_attrs+0x23b/0x3 ---truncated---
CVE-2025-38715	In the Linux kernel, the following vulnerability has been resolved: hfs: fix slab-out-of-bounds in hfs_bnode_read() This patch introduces is_bnode_offset_valid() method that checks the requested offset value. Also, it introduces check_and_correct_requested_length() method that checks and correct the requested length (if it is necessary). These methods are used in hfs_bnode_read(), hfs_bnode_write(), hfs_bnode_clear(), hfs_bnode_copy(), and hfs_bnode_move() with the goal to prevent the access out of allocated memory and triggering the crash.

CVE ID	Description
CVE-2025-38716	In the Linux kernel, the following vulnerability has been resolved: hfs: fix general protection fault in hfs_find_init() The hfs_find_init() method can trigger the crash if tree pointer is NULL: [45.746290][T9787] Oops: general protection fault, probably for non-canonical address 0xdffffc0000000008: 0000 [#1] SMP KAI [45.747287][T9787] KASAN: null-ptr-deref in range [0x0000000000000040-0x0000000000000047] [45.748716][T9787] CPU: 2 UID: 0 PID: 9787 Comm: repro Not tainted 6.16.0-rc3 #10 PREEMPT(full) [45.750250][T9787] Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [45.751983][T9787] RIP: 0010:hfs_find_init+0x86/0x230 [45.752834][T9787] Code: c1 ea 03 80 3c 02 00 0f 85 9a 01 00 00 4c 8d 6b 40 48 c7 45 18 00 00 00 00 48 b8 00 00 00 00 fc [45.755574][T9787] RSP: 0018:ffff90015157668 EFLAGS: 00010202 [45.756432][T9787] RAX: dffffc0000000000 RBX: 0000000000000000 RCX: ffffffff819a4d09 [45.757457][T9787] RDX: 0000000000000008 RSI: ffffffff819acd3a RDI: ffff900151576e8 [45.758282][T9787] RBP: ffff900151576d0 R08: 0000000000000005 R09: 0000000000000000 [45.758943][T9787] R10: 0000000080000000 R11: 0000000000000001 R12: 0000000000000004 [45.759619] [T9787] R13: 0000000000000040 R14: ffff88802c50814a R15: 0000000000000000 [45.760293] [T9787] FS: 00007ffb72734540(0000) GS:ffff8880cec64000(0000) knlGS:0000000000000000 [45.761050][T9787] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [45.761606] [T9787] CR2: 00007f9bd8225000 CR3: 000000010979a000 CR4: 0000000000000060 [45.762286] [T9787] Call Trace: [45.762570][T9787] <TASK> [45.762824][T9787] hfs_ext_read_extent+0x190/0x9d0 [45.763269][T9787] ? submit_bio_noacct_nocheck+0x2dd/0xce0 [45.763766] [T9787] ? __pfx_hfs_ext_read_extent+0x10/0x10 [45.764250][T9787] hfs_get_block+0x55f/0x830 [45.764646][T9787] block_read_full_folio+0x36d/0x850 [45.765105][T9787] ? __pfx_hfs_get_block+0x10/0x10 [45.765541][T9787] ? const_folio_flags+0x5b/0x100 [45.765972][T9787] ? __pfx_hfs_read_folio+0x10/0x10 [45.766415][T9787] filemap_read_folio+0xbe/0x290 [45.766840] [T9787] ? __pfx_filemap_read_folio+0x10/0x10 [45.767325][T9787] ? __filemap_get_folio+0x32b/0xbf0 [45.767780][T9787] do_read_cache_folio+0x263/0x5c0 [45.768223][T9787] ? __pfx_hfs_read_folio+0x10/0x10 [45.768666][T9787] read_cache_page+0x5b/0x160 [45.769070] [T9787] hfs_btree_open+0x491/0x1740 [45.769481][T9787] hfs_mdb_get+0x15e2/0x1fb0 [45.769877][T9787] ? __pfx_hfs_mdb_get+0x10/0x10 [45.770316][T9787] ? find_held_lock+0x2b/0x80 [45.770731][T9787] ? lockdep_init_map_type+0x5c/0x280 [45.771200][T9787] ? lockdep_init_map_type+0x5c/0x280 [45.771674][T9787] hfs_fill_super+0x38e/0x720 [45.772092] [T9787] ? __pfx_hfs_fill_super+0x10/0x10 [45.772549][T9787] ? snprintf+0xbe/0x100 [45.772931] [T9787] ? __pfx_snprintf+0x10/0x10 [45.773350][T9787] ? do_raw_spin_lock+0x129/0x2b0 [45.773796][T9787] ? find_held_lock+0x2b/0x80 [45.774215][T9787] ? set_blocksize+0x40a/0x510 [45.774636][T9787] ? sb_set_blocksize+0x176/0x1d0 [45.775087][T9787] ? setup_bdev_super+0x369/0x730 [45.775533][T9787] get_tree_bdev_flags+0x384/0x620 [45.775985][T9787] ? __pfx_hfs_fill_super+0x10/0x10 [45.776453][T9787] ? __pfx_get_tree_bdev_flags+0x10/0x10 [45.776950][T9787] ? bpf_lsm_capable+0x9/0x10 [45.777365][T9787] ? security_capable+0x80/0x260 [45.777803][T9787] vfs_get_tree+0x8e/0x340 [45.778203][T9787] path_mount+0x13de/0x2010 [45.778604][T9787] ? kmem_cache_free+0x2b0/0x4c0 [45.779052][T9787] ? __pfx_path_mount+0x10/0x10 [45.779480][T9787] ? getname_flags.part.0+0x1c5/0x550 [45.779954] [T9787] ? putname+0x154/0x1a0 [45.780335][T9787] __x64_sys_mount+0x27b/0x300 [45.780758] [T9787] ? __pfx__x64_sys_mount+0x10/0x10 [45.781232][T9787] ---truncated---
CVE-2025-38717	In the Linux kernel, the following vulnerability has been resolved: net: kcm: Fix race condition in kcm_unattach() syzbot found a race condition when kcm_unattach(psock) and kcm_release(kcm) are executed at the same time. kcm_unattach() is missing a check of the flag kcm->tx_stopped before calling queue_work(). If the kcm has a reserved psock, kcm_unattach() might get executed between cancel_work_sync() and unreserve_psock() in kcm_release(), requeuing kcm->tx_work right before kcm gets freed in kcm_done(). Remove kcm->tx_stopped and replace it by the less error-prone disable_work_sync().

CVE ID	Description
CVE-2025-38718	In the Linux kernel, the following vulnerability has been resolved: sctp: linearize cloned gso packets in sctp_rcv A cloned head skb still shares these frag skbs in fraglist with the original head skb. It's not safe to access these frag skbs. syzbot reported two use-of-uninitialized-memory bugs caused by this: BUG: KMSAN: uninit-value in sctp_inq_pop+0x15b7/0x1920 net/sctp/inqueue.c:211 sctp_inq_pop+0x15b7/0x1920 net/sctp/inqueue.c:211 sctp_assoc_bh_rcv+0x1a7/0xc50 net/sctp/associola.c:998 sctp_inq_push+0x2ef/0x380 net/sctp/inqueue.c:88 sctp_backlog_rcv+0x397/0xdb0 net/sctp/input.c:331 sk_backlog_rcv+0x13b/0x420 include/net/sock.h:1122 __release_sock+0x1da/0x330 net/core/sock.c:3106 release_sock+0x6b/0x250 net/core/sock.c:3660 sctp_wait_for_connect+0x487/0x820 net/sctp/socket.c:9360 sctp_sendmsg_to_asoc+0x1ec1/0x1f00 net/sctp/socket.c:1885 sctp_sendmsg+0x32b9/0x4a80 net/sctp/socket.c:2031 inet_sendmsg+0x25a/0x280 net/ipv4/af_inet.c:851 sock_sendmsg_nosec net/socket.c:718 [inline] and BUG: KMSAN: uninit-value in sctp_assoc_bh_rcv+0x34e/0xbc0 net/sctp/associola.c:987 sctp_assoc_bh_rcv+0x34e/0xbc0 net/sctp/associola.c:987 sctp_inq_push+0x2a3/0x350 net/sctp/inqueue.c:88 sctp_backlog_rcv+0x3c7/0xda0 net/sctp/input.c:331 sk_backlog_rcv+0x142/0x420 include/net/sock.h:1148 __release_sock+0x1d3/0x330 net/core/sock.c:3213 release_sock+0x6b/0x270 net/core/sock.c:3767 sctp_wait_for_connect+0x458/0x820 net/sctp/socket.c:9367 sctp_sendmsg_to_asoc+0x223a/0x2260 net/sctp/socket.c:1886 sctp_sendmsg+0x3910/0x49f0 net/sctp/socket.c:2032 inet_sendmsg+0x269/0x2a0 net/ipv4/af_inet.c:851 sock_sendmsg_nosec net/socket.c:712 [inline] This patch fixes it by linearizing cloned gso packets in sctp_rcv().
CVE-2025-38721	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: fix refcount leak on table dump There is a reference count leak in ctnetlink_dump_table(): if (res < 0) { nf_conntrack_get(&ct->ct_general); // HERE cb->args[1] = (unsigned long)ct; ... While its very unlikely, its possible that ct == last. If this happens, then the refcount of ct was already incremented. This 2nd increment is never undone. This prevents the conntrack object from being released, which in turn keeps prevents cnet->count from dropping back to 0. This will then block the netns dismantle (or conntrack rmmmod) as nf_conntrack_cleanup_net_list() will wait forever. This can be reproduced by running conntrack_resize.sh selftest in a loop. It takes ~20 minutes for me on a preemptible kernel on average before I see a runaway kworker spinning in nf_conntrack_cleanup_net_list. One fix would to change this to: if (res < 0) { if (ct != last) nf_conntrack_get(&ct->ct_general); But this reference counting isn't needed in the first place. We can just store a cookie value instead. A followup patch will do the same for ctnetlink_exp_dump_table, it looks to me as if this has the same problem and like ctnetlink_dump_table, we only need a 'skip hint', not the actual object so we can apply the same cookie strategy there as well.
CVE-2025-38722	In the Linux kernel, the following vulnerability has been resolved: habanalabs: fix UAF in export_dmabuf() As soon as we'd inserted a file reference into descriptor table, another thread could close it. That's fine for the case when all we are doing is returning that descriptor to userland (it's a race, but it's a userland race and there's nothing the kernel can do about it). However, if we follow fd_install() with any kind of access to objects that would be destroyed on close (be it the struct file itself or anything destroyed by its ->release()), we have a UAF. dma_buf_fd() is a combination of reserving a descriptor and fd_install(). habanalabs export_dmabuf() calls it and then proceeds to access the objects destroyed on close. In particular, it grabs an extra reference to another struct file that will be dropped as part of ->release() for ours; that "will be" is actually "might have already been". Fix that by reserving descriptor before anything else and do fd_install() only when everything had been set up. As a side benefit, we no longer have the failure exit with file already created, but reference to underlying file (as well as ->dmabuf_export_cnt, etc.) not grabbed yet; unlike dma_buf_fd(), fd_install() can't fail.
CVE-2025-38723	In the Linux kernel, the following vulnerability has been resolved: LoongArch: BPF: Fix jump offset calculation in tailcall The extra pass of bpf_int_jit_compile() skips JIT context initialization which essentially skips offset calculation leaving out_offset = -1, so the jmp_offset in emit_bpf_tail_call is calculated by "#define jmp_offset (out_offset - (cur_offset))" is a negative number, which is wrong. The final generated assembly are as follow. 54: bgeu \$a2, \$t1, -8 # 0x00000004c 58: addi.d \$a6, \$s5, -1 5c: bltz \$a6, -16 # 0x00000004c 60: alsld \$t2, \$a2, \$a1, 0x3 64: ld.d \$t2, \$t2, 264 68: beq \$t2, \$zero, -28 # 0x00000004c Before apply this patch, the follow test case will reveal soft lock issues. cd tools/testing/selftests/bpf/ ./test_progs --allow=tailcalls/tailcall_bpf2bpf_1 dmesg: watchdog: BUG: soft lockup - CPU#2 stuck for 26s! [test_progs:25056]
CVE-2025-38724	In the Linux kernel, the following vulnerability has been resolved: nfsd: handle get_client_locked() failure in nfsd4_setclientid_confirm() Lei Lu recently reported that nfsd4_setclientid_confirm() did not check the return value from get_client_locked(). a SETCLIENTID_CONFIRM could race with a confirmed client expiring and fail to get a reference. That could later lead to a UAF. Fix this by getting a reference early in the case where there is an extant confirmed client. If that fails then treat it as if there were no confirmed client found at all. In the case where the unconfirmed client is expiring, just fail and return the result from get_client_locked().

CVE ID	Description
CVE-2025-38725	In the Linux kernel, the following vulnerability has been resolved: net: usb: asix_devices: add phy_mask for ax88772 mdio bus Without setting phy_mask for ax88772 mdio bus, current driver may create at most 32 mdio phy devices with phy address range from 0x00 ~ 0x1f. DLink DUB-E100 H/W Ver B1 is such a device. However, only one main phy device will bind to net phy driver. This is creating issue during system suspend/resume since phy_polling_mode() in phy_state_machine() will directly deference member of phydev->drv for non-main phy devices. Then NULL pointer dereference issue will occur. Due to only external phy or internal phy is necessary, add phy_mask for ax88772 mdio bus to workarround the issue.
CVE-2025-38728	In the Linux kernel, the following vulnerability has been resolved: smb3: fix for slab out of bounds on mount to ksmbd With KASAN enabled, it is possible to get a slab out of bounds during mount to ksmbd due to missing check in parse_server_interfaces() (see below): BUG: KASAN: slab-out-of-bounds in parse_server_interfaces+0x14ee/0x1880 [cifs] Read of size 4 at addr ffff8881433dba98 by task mount/9827 CPU: 5 UID: 0 PID: 9827 Comm: mount Tainted: G OE 6.16.0-rc2-kasan #2 PREEMPT(voluntary) Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: Dell Inc. Precision Tower 3620/0MWYPT, BIOS 2.13.1 06/14/2019 Call Trace: <TASK> dump_stack_lvl+0x9f/0xf0 print_report+0xd1/0x670 __virt_addr_valid+0x22c/0x430 ? parse_server_interfaces+0x14ee/0x1880 [cifs] ? kasan_complete_mode_report_info+0x2a/0x1f0 ? parse_server_interfaces+0x14ee/0x1880 [cifs] kasan_report+0xd6/0x110 parse_server_interfaces+0x14ee/0x1880 [cifs] __asan_report_load_n_noabort+0x13/0x20 parse_server_interfaces+0x14ee/0x1880 [cifs] ? __pfx_parse_server_interfaces+0x10/0x10 [cifs] ? trace_hardirqs_on+0x51/0x60 SMB3_request_interfaces+0x1ad/0x3f0 [cifs] ? __pfx_SMB3_request_interfaces+0x10/0x10 [cifs] ? SMB2_tcon+0x23c/0x15d0 [cifs] smb3_qfs_tcon+0x173/0x2b0 [cifs] ? __pfx_smb3_qfs_tcon+0x10/0x10 [cifs] ? cifs_get_tcon+0x105d/0x2120 [cifs] ? do_raw_spin_unlock+0x5d/0x200 ? cifs_get_tcon+0x105d/0x2120 [cifs] ? __pfx_smb3_qfs_tcon+0x10/0x10 [cifs] cifs_mount_get_tcon+0x369/0xb90 [cifs] ? dfs_cache_find+0xe7/0x150 [cifs] dfs_mount_share+0x985/0x2970 [cifs] ? check_path.constprop.0+0x28/0x50 ? save_trace+0x54/0x370 ? __pfx_dfs_mount_share+0x10/0x10 [cifs] ? __lock_acquire+0xb82/0x2ba0 ? __kasan_check_write+0x18/0x20 cifs_mount+0x9bc/0x9e0 [cifs] ? __pfx_cifs_mount+0x10/0x10 [cifs] ? do_raw_spin_unlock+0x5d/0x200 ? cifs_setup_cifs_sb+0x29d/0x810 [cifs] cifs_smb3_do_mount+0x263/0x1990 [cifs]
CVE-2025-38729	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Validate UAC3 power domain descriptors, too UAC3 power domain descriptors need to be verified with its variable bLength for avoiding the unexpected OOB accesses by malicious firmware, too.
CVE-2025-38730	In the Linux kernel, the following vulnerability has been resolved: io_uring/net: commit partial buffers on retry Ring provided buffers are potentially only valid within the single execution context in which they were acquired. io_uring deals with this and invalidates them on retry. But on the networking side, if MSG_WAITALL is set, or if the socket is of the streaming type and too little was processed, then it will hang on to the buffer rather than recycle or commit it. This is problematic for two reasons: 1) If someone unregisters the provided buffer ring before a later retry, then the req->buf_list will no longer be valid. 2) If multiple sockers are using the same buffer group, then multiple receives can consume the same memory. This can cause data corruption in the application, as either receive could land in the same userspace buffer. Fix this by disallowing partial retries from pinning a provided buffer across multiple executions, if ring provided buffers are used.
CVE-2025-38732	In the Linux kernel, the following vulnerability has been resolved: netfilter: nf_reject: don't leak dst refcount for loopback packets recent patches to add a WARN() when replacing skb dst entry found an old bug: WARNING: include/linux/skbuff.h:1165 skb_dst_check_unset include/linux/skbuff.h:1164 [inline] WARNING: include/linux/skbuff.h:1165 skb_dst_set include/linux/skbuff.h:1210 [inline] WARNING: include/linux/skbuff.h:1165 nf_reject_fill_skb_dst+0x2a4/0x330 net/ipv4/netfilter/nf_reject_ipv4.c:234 [...] Call Trace: nf_send_unreach+0x17b/0x6e0 net/ipv4/netfilter/nf_reject_ipv4.c:325 nft_reject_inet_eval+0x4bc/0x690 net/netfilter/nft_reject_inet.c:27 expr_call_ops_eval net/netfilter/nf_tables_core.c:237 [inline] .. This is because blamed commit forgot about loopback packets. Such packets already have a dst_entry attached, even at PRE_ROUTING stage. Instead of checking hook just check if the skb already has a route attached to it.

CVE ID	Description
CVE-2025-38734	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix UAF on smcsock after smc_listen_out() BPF CI testing report a UAF issue: [16.446633] BUG: kernel NULL pointer dereference, address: 0000000000000003 0 [16.447134] #PF: supervisor read access in kernel mode [16.447516] #PF: error_code(0x0000) - not-present page [16.447878] PGD 0 P4D 0 [16.448063] Oops: Oops: 0000 [#1] PREEMPT SMP NOPT I [16.448409] CPU: 0 UID: 0 PID: 9 Comm: kworker/0:1 Tainted: G OE 6.13.0-rc3-g89e8a75fda73-dirty #4 2 [16.449124] Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE [16.449502] Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/201 4 [16.450201] Workqueue: smc_hs_wq smc_listen_work [16.450531] RIP: 0010:smc_listen_work+0xc02/0x159 0 [16.452158] RSP: 0018:ffff5ab40053d98 EFLAGS: 0001024 6 [16.452526] RAX: 0000000000000001 RBX: 0000000000000002 RCX: 0000000000000030 0 [16.452994] RDX: 0000000000000280 RSI: 00003513840053f0 RDI: 0000000000000000 0 [16.453492] RBP: fffffa097808e3800 R08: fffffa09782dba1e0 R09: 0000000000000000 5 [16.453987] R10: 0000000000000000 R11: 0000000000000000 R12: fffffa0978274640 0 [16.454497] R13: 0000000000000000 R14: 0000000000000000 R15: fffffa09782d4092 0 [16.454996] FS: 0000000000000000(0000) GS:ffffa097bb000000(0000) knlGS:0000000000000000 0 [16.455557] CS: 0010 DS: 0000 ES: 0000 CR0: 000000008005003 3 [16.455961] CR2: 0000000000000030 CR3: 0000000102788004 CR4: 0000000000770ef 0 [16.456459] PKRU: 5555555 4 [16.456654] Call Trace : [16.456832] <TASK > [16.456989] ? __die+0x23/0x7 0 [16.457215] ? page_fault_oops+0x180/0x4c 0 [16.457508] ? __lock_acquire+0x3e6/0x249 0 [16.457801] ? exc_page_fault+0x68/0x20 0 [16.458080] ? asm_exc_page_fault+0x26/0x3 0 [16.458389] ? smc_listen_work+0xc02/0x159 0 [16.458689] ? smc_listen_work+0xc02/0x159 0 [16.458987] ? lock_is_held_type+0x8f/0x10 0 [16.459284] process_one_work+0x1ea/0x6d 0 [16.459570] worker_thread+0x1c3/0x38 0 [16.459839] ? __pfx_worker_thread+0x10/0x1 0 [16.460144] kthread+0xe0/0x11 0 [16.460372] ? __pfx_kthread+0x10/0x1 0 [16.460640] ret_from_fork+0x31/0x5 0 [16.460896] ? __pfx_kthread+0x10/0x1 0 [16.461166] ret_from_fork_asm+0x1a/0x3 0 [16.461453] </TASK > [16.461616] Modules linked in: bpf_testmod(OE) [last unloaded: bpf_testmod(OE)] [16.462134] CR2: 0000000000000003 0 [16.462380] ---[end trace 0000000000000000]--- [16.462710] RIP: 0010:smc_listen_work+0xc02/0x1590 The direct cause of this issue is that after smc_listen_out_connected(), newclsock->sk may be NULL since it will releases the smcsock. Therefore, if the application closes the socket immediately after accept, newclsock->sk can be NULL. A possible execution order could be as follows: smc_listen_work userspace ----- lock_sock(sk) smc_listen_out_connected() \- smc_listen_out \- release_sock \- sk->sk_data_ready() fd = accept(); close(fd); \- socket->sk = NULL; /* newclsock->sk is NULL now */ SMC_STAT_SERV_SUCC_INC(socket_net(newclsock->sk)) Since smc_listen_out_connected() will not fail, simply swapping the order of the code can easily fix this issue.
CVE-2025-38735	In the Linux kernel, the following vulnerability has been resolved: gve: prevent ethtool ops after shutdown A crash can occur if an ethtool operation is invoked after shutdown() is called. shutdown() is invoked during system shutdown to stop DMA operations without performing expensive deallocations. It is discouraged to unregister the netdev in this path, so the device may still be visible to userspace and kernel helpers. In gve, shutdown() tears down most internal data structures. If an ethtool operation is dispatched after shutdown(), it will dereference freed or NULL pointers, leading to a kernel panic. While graceful shutdown normally quiesces userspace before invoking the reboot syscall, forced shutdowns (as observed on GCP VMs) can still trigger this path. Fix by calling netif_device_detach() in shutdown(). This marks the device as detached so the ethtool ioctl handler will skip dispatching operations to the driver.
CVE-2025-39673	In the Linux kernel, the following vulnerability has been resolved: ppp: fix race conditions in ppp_fill_forward_path ppp_fill_forward_path() has two race conditions: 1. The ppp->channels list can change between list_empty() and list_first_entry(), as ppp_lock() is not held. If the only channel is deleted in ppp_disconnect_channel(), list_first_entry() may access an empty head or a freed entry, and trigger a panic. 2. pch->chan can be NULL. When ppp_unregister_channel() is called, pch->chan is set to NULL before pch is removed from ppp->channels. Fix these by using a lockless RCU approach: - Use list_first_or_null_rcu() to safely test and access the first list entry. - Convert list modifications on ppp->channels to their RCU variants and add synchronize_net() after removal. - Check for a NULL pch->chan before dereferencing it.
CVE-2025-39675	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Add null pointer check in mod_hdcp_hdcp1_create_session() The function mod_hdcp_hdcp1_create_session() calls the function get_first_active_display(), but does not check its return value. The return value is a null pointer if the display list is empty. This will lead to a null pointer dereference. Add a null pointer check for get_first_active_display() and return MOD_HDCP_STATUS_DISPLAY_NOT_FOUND if the function return null. This is similar to the commit c3e9826a2202 ("drm/amd/display: Add null pointer check for get_first_active_display()"). (cherry picked from commit 5e43eb3cd731649c4f8b9134f857be62a416c893)

CVE ID	Description
CVE-2025-39676	In the Linux kernel, the following vulnerability has been resolved: <code>scsi: qla4xxx</code> : Prevent a potential error pointer dereference The <code>qla4xxx_get_ep_fwdb()</code> function is supposed to return NULL on error, but <code>qla4xxx_ep_connect()</code> returns error pointers. Propagating the error pointers will lead to an Oops in the caller, so change the error pointers to NULL.
CVE-2025-39677	In the Linux kernel, the following vulnerability has been resolved: <code>net/sched</code> : Fix backlog accounting in <code>qdisc_dequeue_internal</code> This issue applies for the following qdiscs: <code>hhf</code> , <code>fq</code> , <code>fq_codel</code> , and <code>fq_pie</code> , and occurs in their change handlers when adjusting to the new limit. The problem is the following in the values passed to the subsequent <code>qdisc_tree_reduce_backlog</code> call given a <code>tbft</code> parent: When the <code>tbft</code> parent runs out of tokens, skbs of these qdiscs will be placed in <code>gso_skb</code> . Their peek handlers are <code>qdisc_peek_dequeued</code> , which accounts for both <code>qlen</code> and backlog. However, in the case of <code>qdisc_dequeue_internal</code> , ONLY <code>qlen</code> is accounted for when pulling from <code>gso_skb</code> . This means that these qdiscs are missing a <code>qdisc_qstats_backlog_dec</code> when dropping packets to satisfy the new limit in their change handlers. One can observe this issue with the following (with <code>tc</code> patched to support a limit of 0): <code>export TARGET=fq tc qdisc del dev lo root tc qdisc add dev lo root handle 1: tbft rate 8bit burst 100b latency 1ms tc qdisc replace dev lo handle 3: parent 1:1 \$TARGET limit 1000 echo "; echo 'add child'; tc -s -d qdisc show dev lo ping -l 1 -f -c2 -s32 -W0.001 127.0.0.1 2>&1 >/dev/null echo "; echo 'after ping'; tc -s -d qdisc show dev lo tc qdisc change dev lo handle 3: parent 1:1 \$TARGET limit 0 echo "; echo 'after limit drop'; tc -s -d qdisc show dev lo tc qdisc replace dev lo handle 2: parent 1:1 sfq echo "; echo 'post graft'; tc -s -d qdisc show dev lo</code> The second to last show command shows 0 packets but a positive number (74) of backlog bytes. The problem becomes clearer in the last show command, where <code>qdisc_purge_queue</code> triggers <code>qdisc_tree_reduce_backlog</code> with the positive backlog and causes an underflow in the <code>tbft</code> parent's backlog (4096 Mb instead of 0). To fix this issue, the codepath for all clients of <code>qdisc_dequeue_internal</code> has been simplified: <code>codel</code> , <code>pie</code> , <code>hhf</code> , <code>fq</code> , <code>fq_pie</code> , and <code>fq_codel</code> . <code>qdisc_dequeue_internal</code> handles the backlog adjustments for all cases that do not directly use the dequeue handler. The old <code>fq_codel_change</code> limit adjustment loop accumulated the arguments to the subsequent <code>qdisc_tree_reduce_backlog</code> call through the <code>cstats</code> field. However, this is confusing and error prone as <code>fq_codel_dequeue</code> could also potentially mutate this field (which <code>qdisc_dequeue_internal</code> calls in the non <code>gso_skb</code> case), so we have unified the code here with other qdiscs.
CVE-2025-39678	In the Linux kernel, the following vulnerability has been resolved: <code>platform/x86/amd/hsmpp</code> : Ensure <code>sock->metric_tbl_addr</code> is non-NULL If metric table address is not allocated, accessing <code>metrics_bin</code> will result in a NULL pointer dereference, so add a check.
CVE-2025-39679	In the Linux kernel, the following vulnerability has been resolved: <code>drm/nouveau/nvif</code> : Fix potential memory leak in <code>nvif_vmm_ctor()</code> . When the <code>nvif_vmm_type</code> is invalid, we will return error directly without freeing the args in <code>nvif_vmm_ctor()</code> , which leading a memory leak. Fix it by setting the <code>ret</code> - <code>EINVAL</code> and <code>goto done</code> .
CVE-2025-39681	In the Linux kernel, the following vulnerability has been resolved: <code>x86/cpu/hygon</code> : Add missing <code>resctrl_cpu_detect()</code> in <code>bsp_init</code> helper Since 923f3a2b48bd ("x86/resctrl: Query LLC monitoring properties once during boot") <code>resctrl_cpu_detect()</code> has been moved from common CPU initialization code to the vendor-specific BSP init helper, while Hygon didn't put that call in their code. This triggers a division by zero fault during early booting stage on our machines with <code>X86_FEATURE_CQM*</code> supported, where <code>get_rdt_mon_resources()</code> tries to calculate <code>mon_l3_config</code> with uninitialized <code>boot_cpu_data.x86_cache_occ_scale</code> . Add the missing <code>resctrl_cpu_detect()</code> in the Hygon BSP init helper. [bp: Message commit message.]
CVE-2025-39682	In the Linux kernel, the following vulnerability has been resolved: <code>tls</code> : fix handling of zero-length records on the <code>rx_list</code> Each <code>recvmsg()</code> call must process either - only contiguous DATA records (any number of them) - one non-DATA record If the next record has different type than what has already been processed we break out of the main processing loop. If the record has already been decrypted (which may be the case for TLS 1.3 where we don't know type until decryption) we queue the pending record to the <code>rx_list</code> . Next <code>recvmsg()</code> will pick it up from there. Queuing the <code>skb</code> to <code>rx_list</code> after zero-copy decrypt is not possible, since in that case we decrypted directly to the user space buffer, and we don't have an <code>skb</code> to queue (<code>darg.skb</code> points to the ciphertext <code>skb</code> for access to metadata like length). Only data records are allowed zero-copy, and we break the processing loop after each non-data record. So we should never zero-copy and then find out that the record type has changed. The corner case we missed is when the initial record comes from <code>rx_list</code> , and it's zero length.

CVE ID	Description
CVE-2025-39683	In the Linux kernel, the following vulnerability has been resolved: tracing: Limit access to parser->buffer when trace_get_user failed When the length of the string written to set_ftrace_filter exceeds FTRACE_BUFF_MAX, the following KASAN alarm will be triggered: BUG: KASAN: slab-out-of-bounds in strsep+0x18c/0x1b0 Read of size 1 at addr ffff0000d00bd5ba by task ash/165 CPU: 1 UID: 0 PID: 165 Comm: ash Not tainted 6.16.0-g6bcd62bd56-dirty Hardware name: linux,dummy-virt (DT) Call trace: show_stack+0x34/0x50 (C) dump_stack_lvl+0xa0/0x158 print_address_description.constprop.0+0x88/0x398 print_report+0xb0/0x280 kasan_report+0xa4/0xf0 __asan_report_load1_noabort+0x20/0x30 strsep+0x18c/0x1b0 ftrace_process_regex.isra.0+0x100/0x2d8 ftrace_regex_release+0x484/0x618 __fput+0x364/0xa58 ____fput+0x28/0x40 task_work_run+0x154/0x278 do_notify_resume+0x1f0/0x220 el0_svc+0xec/0xf0 el0t_64_sync_handler+0xa0/0xe8 el0t_64_sync+0x1ac/0x1b0 The reason is that trace_get_user will fail when processing a string longer than FTRACE_BUFF_MAX, but not set the end of parser->buffer to 0. Then an OOB access will be triggered in ftrace_regex_release-> ftrace_process_regex->strsep->strpbrk. We can solve this problem by limiting access to parser->buffer when trace_get_user failed.
CVE-2025-39684	In the Linux kernel, the following vulnerability has been resolved: comedi: Fix use of uninitialized memory in do_insn_ioctl() and do_insnlist_ioctl() syzbot reports a KMSAN kernel-infoleak in `do_insn_ioctl()`. A kernel buffer is allocated to hold `insn->n` samples (each of which is an `unsigned int`). For some instruction types, `insn->n` samples are copied back to user-space, unless an error code is being returned. The problem is that not all the instruction handlers that need to return data to userspace fill in the whole `insn->n` samples, so that there is an information leak. There is a similar syzbot report for `do_insnlist_ioctl()`, although it does not have a reproducer for it at the time of writing. One culprit is `insn_rw_emulate_bits()` which is used as the handler for `INSN_READ` or `INSN_WRITE` instructions for subdevices that do not have a specific handler for that instruction, but do have an `INSN_BITS` handler. For `INSN_READ` it only fills in at most 1 sample, so if `insn->n` is greater than 1, the remaining `insn->n - 1` samples copied to userspace will be uninitialized kernel data. Another culprit is `vm80xx_ai_insn_read()` in the "vm80xx" driver. It never returns an error, even if it fails to fill the buffer. Fix it in `do_insn_ioctl()` and `do_insnlist_ioctl()` by making sure that uninitialized parts of the allocated buffer are zeroed before handling each instruction. Thanks to Arnaud Lecomte for their fix to `do_insn_ioctl()`. That fix replaced the call to `kmalloc_array()` with `kcalloc()`, but it is not always necessary to clear the whole buffer.
CVE-2025-39685	In the Linux kernel, the following vulnerability has been resolved: comedi: pcl726: Prevent invalid irq number The reproducer passed in an irq number(0x80008000) that was too large, which triggered the oob. Added an interrupt number check to prevent users from passing in an irq number that was too large. If `it->options[1]` is 31, then `1 << it->options[1]` is still invalid because it shifts a 1-bit into the sign bit (which is UB in C). Possible solutions include reducing the upper bound on the `it->options[1]` value to 30 or lower, or using `1U << it->options[1]`. The old code would just not attempt to request the IRQ if the `options[1]` value were invalid. And it would still configure the device without interrupts even if the call to `request_irq` returned an error. So it would be better to combine this test with the test below.
CVE-2025-39686	In the Linux kernel, the following vulnerability has been resolved: comedi: Make insn_rw_emulate_bits() do insn->n samples The `insn_rw_emulate_bits()` function is used as a default handler for `INSN_READ` instructions for subdevices that have a handler for `INSN_BITS` but not for `INSN_READ`. Similarly, it is used as a default handler for `INSN_WRITE` instructions for subdevices that have a handler for `INSN_BITS` but not for `INSN_WRITE`. It works by emulating the `INSN_READ` or `INSN_WRITE` instruction handling with a constructed `INSN_BITS` instruction. However, `INSN_READ` and `INSN_WRITE` instructions are supposed to be able read or write multiple samples, indicated by the `insn->n` value, but `insn_rw_emulate_bits()` currently only handles a single sample. For `INSN_READ`, the comedi core will copy `insn->n` samples back to user-space. (That triggered KASAN kernel-infoleak errors when `insn->n` was greater than 1, but that is being fixed more generally elsewhere in the comedi core.) Make `insn_rw_emulate_bits()` either handle `insn->n` samples, or return an error, to conform to the general expectation for `INSN_READ` and `INSN_WRITE` handlers.
CVE-2025-39687	In the Linux kernel, the following vulnerability has been resolved: iio: light: as73211: Ensure buffer holes are zeroed Given that the buffer is copied to a kfifo that ultimately user space can read, ensure we zero it.
CVE-2025-39689	In the Linux kernel, the following vulnerability has been resolved: ftrace: Also allocate and copy hash for reading of filter files Currently the reader of set_ftrace_filter and set_ftrace_notrace just adds the pointer to the global tracer hash to its iterator. Unlike the writer that allocates a copy of the hash, the reader keeps the pointer to the filter hashes. This is problematic because this pointer is static across function calls that release the locks that can update the global tracer hashes. This can cause UAF and similar bugs. Allocate and copy the hash for reading the filter files like it is done for the writers. This not only fixes UAF bugs, but also makes the code a bit simpler as it doesn't have to differentiate when to free the iterator's hash between writers and readers.

CVE ID	Description
CVE-2025-39691	In the Linux kernel, the following vulnerability has been resolved: fs/buffer: fix use-after-free when call bh_read() helper There's issue as follows: BUG: KASAN: stack-out-of-bounds in end_buffer_read_sync+0xe3/0x110 Read of size 8 at addr ffffc9000168f7f8 by task swapper/3/0 CPU: 3 UID: 0 PID: 0 Comm: swapper/3 Not tainted 6.16.0-862.14.0.6.x86_64 Hardware name: QEMU Standard PC (i440FX + PIIX, 1996) Call Trace: <IRQ> dump_stack_lvl+0x55/0x70 print_address_description.constprop.0+0x2c/0x390 print_report+0xb4/0x270 kasan_report+0xb8/0xf0 end_buffer_read_sync+0xe3/0x110 end_bio_bh_io_sync+0x56/0x80 blk_update_request+0x30a/0x720 scsi_end_request+0x51/0x2b0 scsi_io_completion+0xe3/0x480 ? scsi_device_unbusy+0x11e/0x160 blk_complete_reqs+0x7b/0x90 handle_softirqs+0xef/0x370 irq_exit_rcu+0xa5/0xd0 sysvec_apic_timer_interrupt+0x6e/0x90 </IRQ> Above issue happens when do ntfs3 filesystem mount, issue may happens as follows: mount IRQ ntfs_fill_super read_cache_page do_read_cache_folio filemap_read_folio mpage_read_folio do_mpage_readpage ntfs_get_block_vbo bh_read submit_bh wait_on_buffer(bh); blk_complete_reqs scsi_io_completion scsi_end_request blk_update_request end_bio_bh_io_sync end_buffer_read_sync __end_buffer_read_notouch unlock_buffer wait_on_buffer(bh);--> return will return to caller put_bh --> trigger stack-out-of-bounds In the mpage_read_folio() function, the stack variable 'map_bh' is passed to ntfs_get_block_vbo(). Once unlock_buffer() unlocks and wait_on_buffer() returns to continue processing, the stack variable is likely to be reclaimed. Consequently, during the end_buffer_read_sync() process, calling put_bh() may result in stack overrun. If the bh is not allocated on the stack, it belongs to a folio. Freeing a buffer head which belongs to a folio is done by drop_buffers() which will fail to free buffers which are still locked. So it is safe to call put_bh() before __end_buffer_read_notouch().
CVE-2025-39692	In the Linux kernel, the following vulnerability has been resolved: smb: server: split ksmbd_rdma_stop_listening() out of ksmbd_rdma_destroy() We can't call destroy_workqueue(smb_direct_wq); before stop_sessions()! Otherwise already existing connections try to use smb_direct_wq as a NULL pointer.
CVE-2025-39693	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: Avoid a NULL pointer dereference [WHY] Although unlikely drm_atomic_get_new_connector_state() or drm_atomic_get_old_connector_state() can return NULL. [HOW] Check returns before dereference. (cherry picked from commit 1e5e8d672fec9f2ab352be121be971877bffa2af9)
CVE-2025-39694	In the Linux kernel, the following vulnerability has been resolved: s390/sclp: Fix SCCB present check Tracing code called by the SCLP interrupt handler contains early exits if the SCCB address associated with an interrupt is NULL. This check is performed after physical to virtual address translation. If the kernel identity mapping does not start at address zero, the resulting virtual address is never zero, so that the NULL checks won't work. Subsequently this may result in incorrect accesses to the first page of the identity mapping. Fix this by introducing a function that handles the NULL case before address translation.
CVE-2025-39697	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix a race when updating an existing write After nfs_lock_and_join_requests() tests for whether the request is still attached to the mapping, nothing prevents a call to nfs_inode_remove_request() from succeeding until we actually lock the page group. The reason is that whoever called nfs_inode_remove_request() doesn't necessarily have a lock on the page group head. So in order to avoid races, let's take the page group lock earlier in nfs_lock_and_join_requests(), and hold it across the removal of the request in nfs_inode_remove_request().
CVE-2025-39698	In the Linux kernel, the following vulnerability has been resolved: io_uring/futex: ensure io_futex_wait() cleans up properly on failure The io_futex_data is allocated upfront and assigned to the io_kiocb async_data field, but the request isn't marked with REQ_F_ASYNC_DATA at that point. Those two should always go together, as the flag tells io_uring whether the field is valid or not. Additionally, on failure cleanup, the futex handler frees the data but does not clear ->async_data. Clear the data and the flag in the error path as well. Thanks to Trend Micro Zero Day Initiative and particularly ReDress for reporting this.
CVE-2025-39701	In the Linux kernel, the following vulnerability has been resolved: ACPI: pfr_update: Fix the driver update version check The security-version-number check should be used rather than the runtime version check for driver updates. Otherwise, the firmware update would fail when the update binary had a lower runtime version number than the current one. [rjw: Changelog edits]
CVE-2025-39702	In the Linux kernel, the following vulnerability has been resolved: ipv6: sr: Fix MAC comparison to be constant-time To prevent timing attacks, MACs need to be compared in constant time. Use the appropriate helper function for this.

CVE ID	Description
CVE-2025-39703	In the Linux kernel, the following vulnerability has been resolved: net, hsr: reject HSR frame if skb can't hold tag Receiving HSR frame with insufficient space to hold HSR tag in the skb can result in a crash (kernel BUG): [45.390915] skbuff: skb_under_panic: text:ffff880f32cac len:26 put:14 head:ffff888042418000 data:ffff888042417ff4 tail:0xe end:0x180 dev:bridge_slave_1 [45.392559] -----[cut here]----- [45.392912] kernel BUG at net/core/skbuff.c:211! [45.393276] Oops: invalid opcode: 0000 [#1] SMP DEBUG_PAGEALLOC KASAN NOPTI [45.393809] CPU: 1 UID: 0 PID: 2496 Comm: reproducer Not tainted 6.15.0 #12 PREEMPT(undef) [45.394433] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 [45.395273] RIP: 0010:skb_panic+0x15b/0x1d0 <snip registers, remove unreliable trace> [45.402911] Call Trace: [45.403105] <IRQ> [45.404470] skb_push+0xcd/0xf0 [45.404726] br_dev_queue_push_xmit+0x7c/0x6c0 [45.406513] br_forward_finish+0x128/0x260 [45.408483] __br_forward+0x42d/0x590 [45.409464] maybe_deliver+0x2eb/0x420 [45.409763] br_flood+0x174/0x4a0 [45.410030] br_handle_frame_finish+0xc7c/0x1bc0 [45.411618] br_handle_frame+0xac3/0x1230 [45.413674] __netif_receive_skb_core.constprop.0+0x808/0x3df0 [45.422966] __netif_receive_skb_one_core+0xb4/0x1f0 [45.424478] __netif_receive_skb+0x22/0x170 [45.424806] process_backlog+0x242/0x6d0 [45.425116] __napi_poll+0xbb/0x630 [45.425394] net_rx_action+0x4d1/0xcc0 [45.427613] handle_softirqs+0x1a4/0x580 [45.427926] do_softirq+0x74/0x90 [45.428196] </IRQ> This issue was found by syzkaller. The panic happens in br_dev_queue_push_xmit() once it receives a corrupted skb with ETH header already pushed in linear data. When it attempts the skb_push() call, there's not enough headroom and skb_push() panics. The corrupted skb is put on the queue by HSR layer, which makes a sequence of unintended transformations when it receives a specific corrupted HSR frame (with incomplete TAG). Fix it by dropping and consuming frames that are not long enough to contain both ethernet and hsr headers. Alternative fix would be to check for enough headroom before skb_push() in br_dev_queue_push_xmit(). In the reproducer, this is injected via AF_PACKET, but I don't easily see why it couldn't be sent over the wire from adjacent network. Further Details: In the reproducer, the following network interface chain is set up: ##### # veth0_to_hsr ##### hsr_slave0 ##### # hsr0 ##### # ##### # veth1_to_hsr ##### hsr_slave1 ##### ## # bridge # ## # ##### # # ... ##### To trigger the events leading up to crash, reproducer sends a corrupted HSR fr --- truncated---
CVE-2025-39705	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: fix a Null pointer dereference vulnerability [Why] A null pointer dereference vulnerability exists in the AMD display driver's (DC module) cleanup function dc_destruct(). When display control context (dc->ctx) construction fails (due to memory allocation failure), this pointer remains NULL. During subsequent error handling when dc_destruct() is called, there's no NULL check before dereferencing the perf_trace member (dc->ctx->perf_trace), causing a kernel null pointer dereference crash. [How] Check if dc->ctx is non-NULL before dereferencing. (Updated commit text and removed unnecessary error message) (cherry picked from commit 9dd8e2ba268c636c240a918e0a31e6feae19404)
CVE-2025-39706	In the Linux kernel, the following vulnerability has been resolved: drm/amdkfd: Destroy KFD debugfs after destroy KFD wq Since KFD proc content was moved to kernel debugfs, we can't destroy KFD debugfs before kfd_process_destroy_wq. Move kfd_process_destroy_wq prior to kfd_debugfs_fini to fix a kernel NULL pointer problem. It happens when /sys/kernel/debug/kfd was already destroyed in kfd_debugfs_fini but kfd_process_destroy_wq calls kfd_debugfs_remove_process. This line debugfs_remove_recursive(entry->proc_dentry); tries to remove /sys/kernel/debug/kfd/proc/<pid> while /sys/kernel/debug/kfd is already gone. It hangs the kernel by kernel NULL pointer. (cherry picked from commit 0333052d90683d88531558dcfdbf2525cc37c233)
CVE-2025-39707	In the Linux kernel, the following vulnerability has been resolved: drm/amdgpu: check if hubbub is NULL in debugfs/amdgpu_dm_capabilities HUBBUB structure is not initialized on DCE hardware, so check if it is NULL to avoid null dereference while accessing amdgpu_dm_capabilities file in debugfs.
CVE-2025-39709	In the Linux kernel, the following vulnerability has been resolved: media: venus: protect against spurious interrupts during probe Make sure the interrupt handler is initialized before the interrupt is registered. If the IRQ is registered before hfi_create(), it's possible that an interrupt fires before the handler setup is complete, leading to a NULL dereference. This error condition has been observed during system boot on Rb3Gen2.
CVE-2025-39710	In the Linux kernel, the following vulnerability has been resolved: media: venus: Add a check for packet size after reading from shared memory Add a check to ensure that the packet size does not exceed the number of available words after reading the packet header from shared memory. This ensures that the size provided by the firmware is safe to process and prevent potential out-of-bounds memory access.

CVE ID	Description
CVE-2025-39711	In the Linux kernel, the following vulnerability has been resolved: media: ivsc: Fix crash at shutdown due to missing mei_cldev_disable() calls Both the ACE and CSI driver are missing a mei_cldev_disable() call in their remove() function. This causes the mei_cl client to stay part of the mei_device->file_list list even though its memory is freed by mei_cl_bus_dev_release() calling kfree(cldev->cl). This leads to a use-after-free when mei_vsc_remove() runs mei_stop() which first removes all mei bus devices calling mei_ace_remove() and mei_csi_remove() followed by mei_cl_bus_dev_release() and then calls mei_cl_all_disconnect() which walks over mei_device->file_list dereferencing the just freed cldev->cl. And mei_vsc_remove() it self is run at shutdown because of the platform_device_unregister(tp->pdev) in vsc_tp_shutdown() When building a kernel with KASAN this leads to the following KASAN report: [106.634504] ===== <pre>[106.634623] BUG: KASAN: slab-use-after-free in mei_cl_set_disconnected (drivers/misc/mei/ client.c:783) mei [106.634683] Read of size 4 at addr ffff88819cb62018 by task systemd-shutdow/1 [106.634729] [106.634767] Tainted: [E]=UNSIGNED_MODULE [106.634770] Hardware name: Dell Inc. XPS 16 9640/09CK4V, BIOS 1.12.0 02/10/2025 [106.634773] Call Trace: [106.634777] <TASK> ... [106.634871] kasan_report (mm/kasan/report.c:221 mm/kasan/report.c:636) [106.634901] mei_cl_set_disconnected (drivers/misc/mei/client.c:783) mei [106.634921] mei_cl_all_disconnect (drivers/misc/mei/client.c:2165 (discriminator 4)) mei [106.634941] mei_reset (drivers/misc/ mei/init.c:163) mei ... [106.635042] mei_stop (drivers/misc/mei/init.c:348) mei [106.635062] mei_vsc_remove (drivers/misc/mei/mei_dev.h:784 drivers/misc/mei/platform-vsc.c:393) mei_vsc [106.635066] platform_remove (drivers/base/platform.c:1424) Add the missing mei_cldev_disable() calls so that the mei_cl gets removed from mei_device->file_list before it is freed to fix this.</pre>
CVE-2025-39712	In the Linux kernel, the following vulnerability has been resolved: media: mt9m114: Fix deadlock in get_frame_interval/set_frame_interval Getting / Setting the frame interval using the V4L2 subdev pad ops get_frame_interval/set_frame_interval causes a deadlock, as the subdev state is locked in the [1] but also in the driver itself. In [2] it's described that the caller is responsible to acquire and release the lock in this case. Therefore, acquiring the lock in the driver is wrong. Remove the lock acquisitions/releases from mt9m114_ifp_get_frame_interval() and mt9m114_ifp_set_frame_interval(). [1] drivers/media/v4l2-core/v4l2-subdev.c - line 1129 [2] Documentation/driver-api/media/v4l2-subdev.rst
CVE-2025-39713	In the Linux kernel, the following vulnerability has been resolved: media: rainshadow-cec: fix TOCTOU race condition in rain_interrupt() In the interrupt handler rain_interrupt(), the buffer full check on rain->buf_len is performed before acquiring rain->buf_lock. This creates a Time-of-Check to Time-of-Use (TOCTOU) race condition, as rain->buf_len is concurrently accessed and modified in the work handler rain_irq_work_handler() under the same lock. Multiple interrupt invocations can race, with each reading buf_len before it becomes full and then proceeding. This can lead to both interrupts attempting to write to the buffer, incrementing buf_len beyond its capacity (DATA_SIZE) and causing a buffer overflow. Fix this bug by moving the spin_lock() to before the buffer full check. This ensures that the check and the subsequent buffer modification are performed atomically, preventing the race condition. An corresponding spin_unlock() is added to the overflow path to correctly release the lock. This possible bug was found by an experimental static analysis tool developed by our team.
CVE-2025-39714	In the Linux kernel, the following vulnerability has been resolved: media: usbtv: Lock resolution while streaming When an program is streaming (ffplay) and another program (qv4l2) changes the TV standard from NTSC to PAL, the kernel crashes due to trying to copy to unmapped memory. Changing from NTSC to PAL increases the resolution in the usbtv struct, but the video plane buffer isn't adjusted, so it overflows. [hverkuil: call vb2_is_busy instead of vb2_is_streaming]
CVE-2025-39715	In the Linux kernel, the following vulnerability has been resolved: parisc: Revise gateway LWS calls to probe user read access We use load and stbys,e instructions to trigger memory reference interruptions without writing to memory. Because of the way read access support is implemented, read access interruptions are only triggered at privilege levels 2 and 3. The kernel and gateway page execute at privilege level 0, so this code never triggers a read access interruption. Thus, it is currently possible for user code to execute a LWS compare and swap operation at an address that is read protected at privilege level 3 (PRIV_USER). Fix this by probing read access rights at privilege level 3 and branching to lws_fault if access isn't allowed.
CVE-2025-39716	In the Linux kernel, the following vulnerability has been resolved: parisc: Revise __get_user() to probe user read access Because of the way read access support is implemented, read access interruptions are only triggered at privilege levels 2 and 3. The kernel executes at privilege level 0, so __get_user() never triggers a read access interruption (code 26). Thus, it is currently possible for user code to access a read protected address via a system call. Fix this by probing read access rights at privilege level 3 (PRIV_USER) and setting __gu_err to -EFAULT (-14) if access isn't allowed. Note the cmpiclr instruction does a 32-bit compare because COND macro doesn't work inside asm.

CVE ID	Description
CVE-2025-39718	In the Linux kernel, the following vulnerability has been resolved: vsock/virtio: Validate length in packet header before skb_put() When receiving a vsock packet in the guest, only the virtqueue buffer size is validated prior to virtio_vsock_skb_rx_put(). Unfortunately, virtio_vsock_skb_rx_put() uses the length from the packet header as the length argument to skb_put(), potentially resulting in SKB overflow if the host has gone wonky. Validate the length as advertised by the packet header before calling virtio_vsock_skb_rx_put().
CVE-2025-39719	In the Linux kernel, the following vulnerability has been resolved: iio: imu: bno055: fix OOB access of hw_xlate array Fix a potential out-of-bounds array access of the hw_xlate array in bno055.c. In bno055_get_regmask(), hw_xlate was iterated over the length of the vals array instead of the length of the hw_xlate array. In the case of bno055_gyr_scale, the vals array is larger than the hw_xlate array, so this could result in an out-of-bounds access. In practice, this shouldn't happen though because a match should always be found which breaks out of the for loop before it iterates beyond the end of the hw_xlate array. By adding a new hw_xlate_len field to the bno055_sysfs_attr, we can be sure we are iterating over the correct length.
CVE-2025-39720	In the Linux kernel, the following vulnerability has been resolved: ksmbd: fix refcount leak causing resource not released When ksmbd_conn_releasing(opinfo->conn) returns true, the refcount was not decremented properly, causing a refcount leak that prevents the count from reaching zero and the memory from being released.
CVE-2025-39721	In the Linux kernel, the following vulnerability has been resolved: crypto: qat - flush misc workqueue during device shutdown Repeated loading and unloading of a device specific QAT driver, for example qat_4xxx, in a tight loop can lead to a crash due to a use-after-free scenario. This occurs when a power management (PM) interrupt triggers just before the device-specific driver (e.g., qat_4xxx.ko) is unloaded, while the core driver (intel_qat.ko) remains loaded. Since the driver uses a shared workqueue ('qat_misc_wq') across all devices and owned by intel_qat.ko, a deferred routine from the device-specific driver may still be pending in the queue. If this routine executes after the driver is unloaded, it can dereference freed memory, resulting in a page fault and kernel crash like the following: BUG: unable to handle page fault for address: ffa000002e50a01c #PF: supervisor read access in kernel mode RIP: 0010:pm_bh_handler+0x1d2/0x250 [intel_qat] Call Trace: pm_bh_handler+0x1d2/0x250 [intel_qat] process_one_work+0x171/0x340 worker_thread+0x277/0x3a0 kthread+0xf0/0x120 ret_from_fork+0x2d/0x50 To prevent this, flush the misc workqueue during device shutdown to ensure that all pending work items are completed before the driver is unloaded. Note: This approach may slightly increase shutdown latency if the workqueue contains jobs from other devices, but it ensures correctness and stability.
CVE-2025-39724	In the Linux kernel, the following vulnerability has been resolved: serial: 8250: fix panic due to PSLVERR When the PSLVERR_RESP_EN parameter is set to 1, the device generates an error response if an attempt is made to read an empty RBR (Receive Buffer Register) while the FIFO is enabled. In serial8250_do_startup(), calling serial_port_out(port, UART_LCR, UART_LCR_WLEN8) triggers dw8250_check_lcr(), which invokes dw8250_force_idle() and serial8250_clear_and_reinit_fifos(). The latter function enables the FIFO via serial_out(p, UART_FCR, p->fcr). Execution proceeds to the serial_port_in(port, UART_RX). This satisfies the PSLVERR trigger condition. When another CPU (e.g., using printk()) is accessing the UART (UART is busy), the current CPU fails the check (value & ~UART_LCR_SPAR) == (lcr & ~UART_LCR_SPAR) in dw8250_check_lcr(), causing it to enter dw8250_force_idle(). Put serial_port_out(port, UART_LCR, UART_LCR_WLEN8) under the port->lock to fix this issue. Panic backtrace: [0.442336] Oops - unknown exception [#1] [0.442343] epc : dw8250_serial_in32+0x1e/0x4a [0.442351] ra : serial8250_do_startup+0x2c8/0x88e ... [0.442416] console_on_rootfs+0x26/0x70

CVE ID	Description
CVE-2025-39726	In the Linux kernel, the following vulnerability has been resolved: s390/ism: fix concurrency management in ism_cmd() The s390x ISM device data sheet clearly states that only one request-response sequence is allowable per ISM function at any point in time. Unfortunately as of today the s390/ism driver in Linux does not honor that requirement. This patch aims to rectify that. This problem was discovered based on Aliaksei's bug report which states that for certain workloads the ISM functions end up entering error state (with PEC 2 as seen from the logs) after a while and as a consequence connections handled by the respective function break, and for future connection requests the ISM device is not considered -- given it is in a dysfunctional state. During further debugging PEC 3A was observed as well. A kernel message like [1211.244319] zpci: 061a:00:00.0: Event 0x2 reports an error for PCI function 0x61a is a reliable indicator of the stated function entering error state with PEC 2. Let me also point out that a kernel message like [1211.244325] zpci: 061a:00:00.0: The ism driver bound to the device does not support error recovery is a reliable indicator that the ISM function won't be auto-recovered because the ISM driver currently lacks support for it. On a technical level, without this synchronization, commands (inputs to the FW) may be partially or fully overwritten (corrupted) by another CPU trying to issue commands on the same function. There is hard evidence that this can lead to DMB token values being used as DMB IOVAs, leading to PEC 2 PCI events indicating invalid DMA. But this is only one of the failure modes imaginable. In theory even completely losing one command and executing another one twice and then trying to interpret the outputs as if the command we intended to execute was actually executed and not the other one is also possible. Frankly, I don't feel confident about providing an exhaustive list of possible consequences.
CVE-2025-39728	In the Linux kernel, the following vulnerability has been resolved: clk: samsung: Fix UBSAN panic in samsung_clk_init() With UBSAN_ARRAY_BOUNDS=y, I'm hitting the below panic due to dereferencing `ctx->clk_data.hws` before setting `ctx->clk_data.num = nr_clks`. Move that up to fix the crash. UBSAN: array index out of bounds: 00000000f2005512 [#1] PREEMPT SMP <snip> Call trace: samsung_clk_init+0x110/0x124 (P) samsung_clk_init+0x48/0x124 (L) samsung_cmu_register_one+0x3c/0xa0 exynos_arm64_register_cmu+0x54/0x64 __gs101_cmu_top_of_clk_init_declare+0x28/0x60 ...
CVE-2025-39730	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix filehandle bounds checking in nfs_fh_to_dentry() The function needs to check the minimal filehandle length before it can access the embedded filehandle.
CVE-2025-39731	In the Linux kernel, the following vulnerability has been resolved: f2fs: vm_unmap_ram() may be called from an invalid context When testing F2FS with xfstests using UFS backed virtual disks the kernel complains sometimes that f2fs_release_decomp_mem() calls vm_unmap_ram() from an invalid context. Example trace from f2fs/007 test: f2fs/007 5s ... [12:59:38][8.902525] run fstests f2fs/007 [11.468026] BUG: sleeping function called from invalid context at mm/vmalloc.c:2978 [11.471849] in_atomic(): 1, irqs_disabled(): 1, non_block: 0, pid: 68, name: irq/22-ufshcd [11.475357] preempt_count: 1, expected: 0 [11.476970] RCU nest depth: 0, expected: 0 [11.478531] CPU: 0 UID: 0 PID: 68 Comm: irq/22-ufshcd Tainted: G W 6.16.0-rc5-xfstests-ufs-g40f92e79b0aa #9 PREEMPT(none) [11.478535] Tainted: [W]=WARN [11.478536] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [11.478537] Call Trace: [11.478543] <TASK> [11.478545] dump_stack_lvl+0x4e/0x70 [11.478554] __might_resched.cold+0xaf/0xbe [11.478557] vm_unmap_ram+0x21/0xb0 [11.478560] f2fs_release_decomp_mem+0x59/0x80 [11.478563] f2fs_free_dic+0x18/0x1a0 [11.478565] f2fs_finish_read_bio+0xd7/0x290 [11.478570] blk_update_request+0xec/0x3b0 [11.478574] ? sbitmap_queue_clear+0x3b/0x60 [11.478576] scsi_end_request+0x27/0x1a0 [11.478582] scsi_io_completion+0x40/0x300 [11.478583] ufshcd_mcq_poll_cqe_lock+0xa3/0xe0 [11.478588] ufshcd_sl_intr+0x194/0x1f0 [11.478592] ufshcd_threaded_intr+0x68/0xb0 [11.478594] ? __pfx_irq_thread_fn+0x10/0x10 [11.478599] irq_thread_fn+0x20/0x60 [11.478602] ? __pfx_irq_thread_fn+0x10/0x10 [11.478603] irq_thread+0xb9/0x180 [11.478605] ? __pfx_irq_thread_dtor+0x10/0x10 [11.478607] ? __pfx_irq_thread+0x10/0x10 [11.478609] kthread+0x10a/0x230 [11.478614] ? __pfx_kthread+0x10/0x10 [11.478615] ret_from_fork+0x7e/0xd0 [11.478619] ? __pfx_kthread+0x10/0x10 [11.478621] ret_from_fork_asm+0x1a/0x30 [11.478623] </TASK> This patch modifies in_task() check inside f2fs_read_end_io() to also check if interrupts are disabled. This ensures that pages are unmapped asynchronously in an interrupt handler.
CVE-2025-39732	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix sleeping-in-atomic in ath11k_mac_op_set_bitrate_mask() ath11k_mac_disable_peer_fixed_rate() is passed as the iterator to ieee80211_iterate_stations_atomic(). Note in this case the iterator is required to be atomic, however ath11k_mac_disable_peer_fixed_rate() does not follow it as it might sleep. Consequently below warning is seen: BUG: sleeping function called from invalid context at wmi.c:304 Call Trace: <TASK> dump_stack_lvl __might_resched.cold ath11k_wmi_cmd_send ath11k_wmi_set_peer_param ath11k_mac_disable_peer_fixed_rate ieee80211_iterate_stations_atomic ath11k_mac_op_set_bitrate_mask.cold Change to ieee80211_iterate_stations_mtx() to fix this issue. Tested-on: WCN6855 hw2.0 PCI WLAN.HSP.1.1-03125-QCAHSPSWPL_V1_V2_SILICONZ_LITE-3.6510.30

CVE ID	Description
CVE-2025-39734	In the Linux kernel, the following vulnerability has been resolved: Revert "fs/ntfs3: Replace inode_trylock with inode_lock" This reverts commit 69505fe98f198ee813898cbcaf6770949636430b. Initially, conditional lock acquisition was removed to fix an xfstest bug that was observed during internal testing. The deadlock reported by syzbot is resolved by reintroducing conditional acquisition. The xfstest bug no longer occurs on kernel version 6.16-rc1 during internal testing. I assume that changes in other modules may have contributed to this.
CVE-2025-39735	In the Linux kernel, the following vulnerability has been resolved: jfs: fix slab-out-of-bounds read in ea_get() During the "size_check" label in ea_get(), the code checks if the extended attribute list (xattr) size matches ea_size. If not, it logs "ea_get: invalid extended attribute" and calls print_hex_dump(). Here, EALIST_SIZE(ea_buf->xattr) returns 4110417968, which exceeds INT_MAX (2,147,483,647). Then ea_size is clamped: int size = clamp_t(int, ea_size, 0, EALIST_SIZE(ea_buf->xattr)); Although clamp_t aims to bound ea_size between 0 and 4110417968, the upper limit is treated as an int, causing an overflow above 2^31 - 1. This leads "size" to wrap around and become negative (-184549328). The "size" is then passed to print_hex_dump() (called "len" in print_hex_dump()), it is passed as type size_t (an unsigned type), this is then stored inside a variable called "int remaining", which is then assigned to "int linelen" which is then passed to hex_dump_to_buffer(). In print_hex_dump() the for loop, iterates through 0 to len-1, where len is 18446744073525002176, calling hex_dump_to_buffer() on each iteration: for (i = 0; i < len; i += rowsize) { linelen = min(remaining, rowsize); remaining -= rowsize; hex_dump_to_buffer(ptr + i, linelen, rowsize, groupsize, linebuf, sizeof(linebuf), ascii); ... } The expected stopping condition (i < len) is effectively broken since len is corrupted and very large. This eventually leads to the "ptr+i" being passed to hex_dump_to_buffer() to get closer to the end of the actual bounds of "ptr", eventually an out of bounds access is done in hex_dump_to_buffer() in the following for loop: for (j = 0; j < len; j++) { if (linebuflen < lx + 2) goto overflow2; ch = ptr[j]; ... } To fix this we should validate "EALIST_SIZE(ea_buf->xattr)" before it is utilised.
CVE-2025-39736	In the Linux kernel, the following vulnerability has been resolved: mm/kmemleak: avoid deadlock by moving pr_warn() outside kmemleak_lock When netpoll is enabled, calling pr_warn_once() while holding kmemleak_lock in mem_pool_alloc() can cause a deadlock due to lock inversion with the netconsole subsystem. This occurs because pr_warn_once() may trigger netpoll, which eventually leads to __alloc_skb() and back into kmemleak code, attempting to reacquire kmemleak_lock. This is the path for the deadlock. mem_pool_alloc() -> raw_spin_lock_irqsave(&kmemleak_lock, flags); -> pr_warn_once() -> netconsole subsystem -> netpoll -> __alloc_skb -> __create_object -> raw_spin_lock_irqsave(&kmemleak_lock, flags); Fix this by setting a flag and issuing the pr_warn_once() after kmemleak_lock is released.
CVE-2025-39737	In the Linux kernel, the following vulnerability has been resolved: mm/kmemleak: avoid soft lockup in __kmemleak_do_cleanup() A soft lockup warning was observed on a relative small system x86-64 system with 16 GB of memory when running a debug kernel with kmemleak enabled. watchdog: BUG: soft lockup - CPU#8 stuck for 33s! [kworker/8:1:134] The test system was running a workload with hot unplug happening in parallel. Then kmemleak decided to disable itself due to its inability to allocate more kmemleak objects. The debug kernel has its CONFIG_DEBUG_KMEMLEAK_MEM_POOL_SIZE set to 40,000. The soft lockup happened in kmemleak_do_cleanup() when the existing kmemleak objects were being removed and deleted one-by-one in a loop via a workqueue. In this particular case, there are at least 40,000 objects that need to be processed and given the slowness of a debug kernel and the fact that a raw_spinlock has to be acquired and released in __delete_object(), it could take a while to properly handle all these objects. As kmemleak has been disabled in this case, the object removal and deletion process can be further optimized as locking isn't really needed. However, it is probably not worth the effort to optimize for such an edge case that should rarely happen. So the simple solution is to call cond_resched() at periodic interval in the iteration loop to avoid soft lockup.

CVE ID	Description
CVE-2025-39738	In the Linux kernel, the following vulnerability has been resolved: btrfs: do not allow relocation of partially dropped subvolumes [BUG] There is an internal report that balance triggered transaction abort, with the following call trace: item 85 key (594509824 169 0) itemoff 12599 itemsize 33 extent refs 1 gen 197740 flags 2 ref#0: tree block backref root 7 item 86 key (594558976 169 0) itemoff 12566 itemsize 33 extent refs 1 gen 197522 flags 2 ref#0: tree block backref root 7 ... BTRFS error (device loop0): extent item not found for insert, bytenr 594526208 num_bytes 16384 parent 449921024 root_objectid 934 owner 1 offset 0 BTRFS error (device loop0): failed to run delayed ref for logical 594526208 num_bytes 16384 type 182 action 1 ref_mod 1: -117 -----[cut here]----- BTRFS: Transaction aborted (error -117) WARNING: CPU: 1 PID: 6963 at ../fs/btrfs/extent-tree.c:2168 btrfs_run_delayed_refs+0xfa/0x110 [btrfs] And btrfs check doesn't report anything wrong related to the extent tree. [CAUSE] The cause is a little complex, firstly the extent tree indeed doesn't have the backref for 594526208. The extent tree only have the following two backrefs around that bytenr on-disk: item 65 key (594509824 METADATA_ITEM 0) itemoff 13880 itemsize 33 refs 1 gen 197740 flags TREE_BLOCK tree block skinny level 0 (176 0x7) tree block backref root CSUM_TREE item 66 key (594558976 METADATA_ITEM 0) itemoff 13847 itemsize 33 refs 1 gen 197522 flags TREE_BLOCK tree block skinny level 0 (176 0x7) tree block backref root CSUM_TREE But the such missing backref item is not an corruption on disk, as the offending delayed ref belongs to subvolume 934, and that subvolume is being dropped: item 0 key (934 ROOT_ITEM 198229) itemoff 15844 itemsize 439 generation 198229 root_dirid 256 bytenr 10741039104 byte_limit 0 bytes_used 345571328 last_snapshot 198229 flags 0x1000000000001(RDONLY) refs 0 drop_progress key (206324 EXTENT_DATA 2711650304) drop_level 2 level 2 generation_v2 198229 And that offending tree block 594526208 is inside the dropped range of that subvolume. That explains why there is no backref item for that bytenr and why btrfs check is not reporting anything wrong. But this also shows another problem, as btrfs will do all the orphan subvolume cleanup at a read-write mount. So half-dropped subvolume should not exist after an RW mount, and balance itself is also exclusive to subvolume cleanup, meaning we shouldn't hit a subvolume half-dropped during relocation. The root cause is, there is no orphan item for this subvolume. In fact there are 5 subvolumes from around 2021 that have the same problem. It looks like the original report has some older kernels running, and caused those zombie subvolumes. Thankfully upstream commit 8d488a8c7ba2 ("btrfs: fix subvolume/snapshot deletion not triggered on mount") has long fixed the bug. [ENHANCEMENT] For repairing such old fs, btrfs-progs will be enhanced. Considering how delayed the problem will show up (at run delayed ref time) and at that time we have to abort transaction already, it is too late. Instead here we reject any half-dropped subvolume for reloc tree at the earliest time, preventing confusion and extra time wasted on debugging similar bugs.

CVE ID	Description
CVE-2025-39739	In the Linux kernel, the following vulnerability has been resolved: iommu/arm-smmu-qcom: Add SM6115 MDSS compatible Add the SM6115 MDSS compatible to clients compatible list, as it also needs that workaround. Without this workaround, for example, QRB4210 RB2 which is based on SM4250/SM6115 generates a lot of smmu unhandled context faults during boot: arm_smmu_context_fault: 116854 callbacks suppressed arm-smmu c600000.iommu: Unhandled context fault: fsr=0x402, iova=0x5c0ec600, fsynr=0x320021, cbfrsynra=0x420, cb=5 arm-smmu c600000.iommu: FSR = 00000402 [Format=2 TF], SID=0x420 arm-smmu c600000.iommu: FSYNR0 = 00320021 [S1CBNDX=50 PNU PLVL=1] arm-smmu c600000.iommu: Unhandled context fault: fsr=0x402, iova=0x5c0d7800, fsynr=0x320021, cbfrsynra=0x420, cb=5 arm-smmu c600000.iommu: FSR = 00000402 [Format=2 TF], SID=0x420 and also failed initialisation of lontium lt9611uxc, gpu and dpu is observed: (binding MDSS components triggered by lt9611uxc have failed) -----[cut here]----- !aspace WARNING: CPU: 6 PID: 324 at drivers/gpu/drm/msm/msm_gem_vma.c:130 msm_gem_vma_init+0x150/0x18c [msm] Modules linked in: ... (long list of modules) CPU: 6 UID: 0 PID: 324 Comm: (udev-worker) Not tainted 6.15.0-03037-gaacc73ceeb8b #4 PREEMPT Hardware name: Qualcomm Technologies, Inc. QRB4210 RB2 (DT) pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : msm_gem_vma_init+0x150/0x18c [msm] lr : msm_gem_vma_init+0x150/0x18c [msm] sp : ffff80008144b280 ... Call trace: msm_gem_vma_init+0x150/0x18c [msm] (P) get_vma_locked+0xc0/0x194 [msm] msm_gem_get_and_pin_iova_range+0x4c/0x9c [msm] msm_gem_kernel_new+0x48/0x160 [msm] msm_gpu_init+0x34c/0x53c [msm] adreno_gpu_init+0x1b0/0x2d8 [msm] a6xx_gpu_init+0x1e8/0x9e0 [msm] adreno_bind+0x2b8/0x348 [msm] component_bind_all+0x100/0x230 msm_drm_bind+0x13c/0x3d0 [msm] try_to_bring_up_aggregate_device+0x164/0x1d0 __component_add+0xa4/0x174 component_add+0x14/0x20 dsi_dev_attach+0x20/0x34 [msm] dsi_host_attach+0x58/0x98 [msm] devm_mipi_dsi_attach+0x34/0x90 lt9611uxc_attach_dsi.isra.0+0x94/0x124 [lontium_lt9611uxc] lt9611uxc_probe+0x540/0x5fc [lontium_lt9611uxc] i2c_device_probe+0x148/0x2a8 really_probe+0x9c/0x2c0 __driver_probe_device+0x78/0x120 driver_probe_device+0x3c/0x154 __driver_attach+0x90/0x1a0 bus_for_each_dev+0x68/0xb8 driver_attach+0x24/0x30 bus_add_driver+0xe4/0x208 driver_register+0x68/0x124 i2c_register_driver+0x48/0xcc lt9611uxc_driver_init+0x20/0x1000 [lontium_lt9611uxc] do_one_initcall+0x60/0x1d4 do_init_module+0x54/0x1fc load_module+0x1748/0x1c8c init_module_from_file+0x74/0xa0 __arm64_sys_finit_module+0x130/0x2f8 invoke_syscall+0x48/0x104 el0_svc_common.constprop.0+0xc0/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x2c/0x80 el0t_64_sync_handler+0x10c/0x138 el0t_64_sync+0x198/0x19c ---[end trace 0000000000000000]--- msm_dpu 5e01000.display-controller: [drm:msm_gpu_init [msm]] *ERROR* could not allocate memptrs: -22 msm_dpu 5e01000.display-controller: failed to load adreno gpu platform a400000.remoteproc:glink-edge:apr:service@7:dais: Adding to iommu group 19 msm_dpu 5e01000.display-controller: failed to bind 5900000.gpu (ops a3xx_ops [msm]): -22 msm_dpu 5e01000.display-controller: adev bind failed: -22 lt9611uxc 0-002b: failed to attach dsi to host lt9611uxc 0-002b: probe with driver lt9611uxc failed with error -22
CVE-2025-39742	In the Linux kernel, the following vulnerability has been resolved: RDMA: hfi1: fix possible divide-by-zero in find_hw_thread_mask() The function divides number of online CPUs by num_core_siblings, and later checks the divider by zero. This implies a possibility to get and divide-by-zero runtime error. Fix it by moving the check prior to division. This also helps to save one indentation level.
CVE-2025-39743	In the Linux kernel, the following vulnerability has been resolved: jfs: truncate good inode pages when hard link is 0 The fileset value of the inode copy from the disk by the reproducer is AGGR_RESERVED_I. When executing evict, its hard link number is 0, so its inode pages are not truncated. This causes the bugon to be triggered when executing clear_inode() because numpages is greater than 0.
CVE-2025-39744	In the Linux kernel, the following vulnerability has been resolved: rcu: Fix rcu_read_unlock() deadlock due to IRQ work During rcu_read_unlock_special(), if this happens during irq_exit(), we can lockup if an IPI is issued. This is because the IPI itself triggers the irq_exit() path causing a recursive lock up. This is precisely what Xiongfeng found when invoking a BPF program on the trace_tick_stop() tracepoint As shown in the trace below. Fix by managing the irq_work state correctly. irq_exit() __irq_exit_rcu() / * in_hardirq() returns false after this */ preempt_count_sub(HARDIRQ_OFFSET) tick_irq_exit() tick_nohz_irq_exit() tick_nohz_stop_sched_tick() trace_tick_stop() /* a bpf prog is hooked on this trace point */ __bpf_trace_tick_stop() bpf_trace_run2() rcu_read_unlock_special() /* will send a IPI to itself */ irq_work_queue_on(&rdp->defer_qs_iw, rdp->cpu); A simple reproducer can also be obtained by doing the following in tick_irq_exit(). It will hang on boot without the patch: static inline void tick_irq_exit(void) { + rcu_read_lock(); + WRITE_ONCE(current->rcu_read_unlock_special.b.need_qs, true); + rcu_read_unlock(); + [neeraj: Apply Frederic's suggested fix for PREEMPT_RT]

CVE ID	Description
CVE-2025-39745	In the Linux kernel, the following vulnerability has been resolved: rcutorture: Fix rcutorture_one_extend_check() splat in RT kernels For built with CONFIG_PREEMPT_RT=y kernels, running rcutorture tests resulted in the following splat: [68.797425] rcutorture_one_extend_check during change: Current 0x1 To add 0x1 To remove 0x0 preempt_count() 0x0 [68.797533] WARNING: CPU: 2 PID: 512 at kernel/rcu/rcutorture.c:1993 rcutorture_one_extend_check+0x419/0x560 [rcutorture] [68.797601] Call Trace: [68.797602] <TASK> [68.797619] ? lockdep_softirqs_off+0xa5/0x160 [68.797631] rcutorture_one_extend+0x18e/0xccc [rcutorture 2466dbd2ff34dbaa36049cb323a80c3306ac997c] [68.797646] ? local_clock+0x19/0x40 [68.797659] rcu_torture_one_read+0xf0/0x280 [rcutorture 2466dbd2ff34dbaa36049cb323a80c3306ac997c] [68.797678] ? __pfx_rcu_torture_one_read+0x10/0x10 [rcutorture 2466dbd2ff34dbaa36049cb323a80c3306ac997c] [68.797804] ? __pfx_rcu_torture_timer+0x10/0x10 [rcutorture 2466dbd2ff34dbaa36049cb323a80c3306ac997c] [68.797815] rcu-torture: rcu_torture_reader task started [68.797824] rcu-torture: Creating rcu_torture_reader task [68.797824] rcu_torture_reader+0x238/0x580 [rcutorture 2466dbd2ff34dbaa36049cb323a80c3306ac997c] [68.797836] ? kvm_sched_clock_read+0x15/0x30 Disable BH does not change the SOFTIRQ corresponding bits in preempt_count() for RT kernels, this commit therefore use softirq_count() to check the if BH is disabled.
CVE-2025-39746	In the Linux kernel, the following vulnerability has been resolved: wifi: ath10k: shutdown driver when hardware is unreliable In rare cases, ath10k may lose connection with the PCIe bus due to some unknown reasons, which could further lead to system crashes during resuming due to watchdog timeout: ath10k_pci 0000:01:00.0: wmi command 20486 timeout, restarting hardware ath10k_pci 0000:01:00.0: already restarting ath10k_pci 0000:01:00.0: failed to stop WMI vdev 0: -11 ath10k_pci 0000:01:00.0: failed to stop vdev 0: -11 ieee80211 phy0: PM: **** DPM device timeout **** Call Trace: panic+0x125/0x315 dpm_watchdog_set+0x54/0x54 dpm_watchdog_handler+0x57/0x57 call_timer_fn+0x31/0x13c At this point, all WMI commands will timeout and attempt to restart device. So set a threshold for consecutive restart failures. If the threshold is exceeded, consider the hardware is unreliable and all ath10k operations should be skipped to avoid system crash. fail_cont_count and pending_recovery are atomic variables, and do not involve complex conditional logic. Therefore, even if recovery check and reconfig complete are executed concurrently, the recovery mechanism will not be broken. Tested-on: QCA6174 hw3.2 PCI WLAN.RM.4.4.1-00288-QCARMSWPZ-1
CVE-2025-39747	In the Linux kernel, the following vulnerability has been resolved: drm/msm: Add error handling for krealloc in metadata setup Function msm_ioctl_gem_info_set_metadata() now checks for krealloc failure and returns -ENOMEM, avoiding potential NULL pointer dereference. Explicitly avoids __GFP_NOFAIL due to deadlock risks and allocation constraints. Patchwork: https://patchwork.freedesktop.org/patch/661235/
CVE-2025-39748	In the Linux kernel, the following vulnerability has been resolved: bpf: Forget ranges when refining tnum after JSET Syzbot reported a kernel warning due to a range invariant violation on the following BPF program. 0: call bpf_get_netns_cookie 1: if r0 == 0 goto <exit> 2: if r0 & 0xffffffff goto <exit> The issue is on the path where we fall through both jumps. That path is unreachable at runtime: after insn 1, we know r0 != 0, but with the sign extension on the jset, we would only fallthrough insn 2 if r0 == 0. Unfortunately, is_branch_taken() isn't currently able to figure this out, so the verifier walks all branches. The verifier then refines the register bounds using the second condition and we end up with inconsistent bounds on this unreachable path: 1: if r0 == 0 goto <exit> r0: u64=[0x1, 0xffffffffffffffff] var_off=(0, 0xffffffffffffffff) 2: if r0 & 0xffffffff goto <exit> r0 before reg_bounds_sync: u64=[0x1, 0xffffffffffffffff] var_off=(0, 0) r0 after reg_bounds_sync: u64=[0x1, 0] var_off=(0, 0) Improving the range refinement for JSET to cover all cases is tricky. We also don't expect many users to rely on JSET given LLVM doesn't generate those instructions. So instead of improving the range refinement for JSETs, Eduard suggested we forget the ranges whenever we're narrowing tnums after a JSET. This patch implements that approach.

CVE ID	Description
CVE-2025-39749	In the Linux kernel, the following vulnerability has been resolved: rcu: Protect ->defer_qs_iw_pending from data race On kernels built with CONFIG_IRQ_WORK=y, when rcu_read_unlock() is invoked within an interrupts-disabled region of code [1], it will invoke rcu_read_unlock_special(), which uses an irq-work handler to force the system to notice when the RCU read-side critical section actually ends. That end won't happen until interrupts are enabled at the soonest. In some kernels, such as those booted with rcutree.use_softirq=y, the irq-work handler is used unconditionally. The per-CPU rcu_data structure's ->defer_qs_iw_pending field is updated by the irq-work handler and is both read and updated by rcu_read_unlock_special(). This resulted in the following KCSAN splat: ----- BUG: KCSAN: data-race in rcu_preempt_deferred_qs_handler / rcu_read_unlock_special read to 0xffff96b95f42d8d8 of 1 bytes by task 90 on cpu 8: rcu_read_unlock_special+0x175/0x260 __rcu_read_unlock+0x92/0xa0 rt_spin_unlock+0x9b/0xc0 __local_bh_enable+0x10d/0x170 __local_bh_enable_ip+0xfb/0x150 rcu_do_batch+0x595/0xc40 rcu_cpu_kthread+0x4e9/0x830 smpboot_thread_fn+0x24d/0x3b0 kthread+0x3bd/0x410 ret_from_fork+0x35/0x40 ret_from_fork_asm+0x1a/0x30 write to 0xffff96b95f42d8d8 of 1 bytes by task 88 on cpu 8: rcu_preempt_deferred_qs_handler+0x1e/0x30 irq_work_single+0xaf/0x160 run_irq_workd+0x91/0xc0 smpboot_thread_fn+0x24d/0x3b0 kthread+0x3bd/0x410 ret_from_fork+0x35/0x40 ret_from_fork_asm+0x1a/0x30 no locks held by irq_work/8/88. irq event stamp: 200272 hardirqs last enabled at (200272): [<ffffffffffb0f56121>] finish_task_switch+0x131/0x320 hardirqs last disabled at (200271): [<ffffffffffb25c7859>] __schedule+0x129/0xd70 softirqs last enabled at (0): [<ffffffffffb0ee093f>] copy_process+0x4df/0x1cc0 softirqs last disabled at (0): [<0000000000000000>] 0x0 ----- The problem is that irq-work handlers run with interrupts enabled, which means that rcu_preempt_deferred_qs_handler() could be interrupted, and that interrupt handler might contain an RCU read-side critical section, which might invoke rcu_read_unlock_special(). In the strict KCSAN mode of operation used by RCU, this constitutes a data race on the ->defer_qs_iw_pending field. This commit therefore disables interrupts across the portion of the rcu_preempt_deferred_qs_handler() that updates the ->defer_qs_iw_pending field. This suffices because this handler is not a fast path.
CVE-2025-39750	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Correct tid cleanup when tid setup fails Currently, if any error occurs during ath12k_dp_rx_peer_tid_setup(), the tid value is already incremented, even though the corresponding TID is not actually allocated. Proceed to ath12k_dp_rx_peer_tid_delete() starting from unallocated tid, which might leads to freeing unallocated TID and cause potential crash or out-of-bounds access. Hence, fix by correctly decrementing tid before cleanup to match only the successfully allocated TIDs. Also, remove tid-- from failure case of ath12k_dp_rx_peer_frag_setup(), as decrementing the tid before cleanup in loop will take care of this. Compile tested only.
CVE-2025-39752	In the Linux kernel, the following vulnerability has been resolved: ARM: rockchip: fix kernel hang during smp initialization In order to bring up secondary CPUs main CPU write trampoline code to SRAM. The trampoline code is written while secondary CPUs are powered on (at least that true for RK3188 CPU). Sometimes that leads to kernel hang. Probably because secondary CPU execute trampoline code while kernel doesn't expect. The patch moves SRAM initialization step to the point where all secondary CPUs are powered down. That fixes rarely hangs on RK3188: [0.091568] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000 [0.091996] rockchip_smp_prepare_cpus: ncores 4
CVE-2025-39753	In the Linux kernel, the following vulnerability has been resolved: gfs2: Set .migrate_folio in gfs2_{rggrp,meta}_aops Clears up the warning added in 7ee3647243e5 ("migrate: Remove call to ->writepage") that occurs in various xfstests, causing "something found in dmesg" failures. [341.136573] gfs2_meta_aops does not implement migrate_folio [341.136953] WARNING: CPU: 1 PID: 36 at mm/migrate.c:944 move_to_new_folio+0x2f8/0x300
CVE-2025-39754	In the Linux kernel, the following vulnerability has been resolved: mm/smmaps: fix race between smaps_hugetlb_range and migration smaps_hugetlb_range() handles the pte without holding ptl, and may be concurrent with migration, leading to BUG_ON in pfn_swap_entry_to_page(). The race is as follows. smaps_hugetlb_range migrate_pages huge_ptep_get remove_migration_ptes folio_unlock pfn_swap_entry_folio BUG_ON To fix it, hold ptl lock in smaps_hugetlb_range().

CVE ID	Description
CVE-2025-39756	In the Linux kernel, the following vulnerability has been resolved: fs: Prevent file descriptor table allocations exceeding INT_MAX When sysctl_nr_open is set to a very high value (for example, 1073741816 as set by systemd), processes attempting to use file descriptors near the limit can trigger massive memory allocation attempts that exceed INT_MAX, resulting in a WARNING in mm/slub.c: WARNING: CPU: 0 PID: 44 at mm/slub.c:5027 __kvmalloc_node_noprof+0x21a/0x288 This happens because kvmalloc_array() and kvmalloc() check if the requested size exceeds INT_MAX and emit a warning when the allocation is not flagged with __GFP_NOWARN. Specifically, when nr_open is set to 1073741816 (0x3fffffff) and a process calls dup2(oldfd, 1073741880), the kernel attempts to allocate: - File descriptor array: 1073741880 * 8 bytes = 8,589,935,040 bytes - Multiple bitmaps: ~400MB - Total allocation size: > 8GB (exceeding INT_MAX = 2,147,483,647) Reproducer: 1. Set /proc/sys/fs/nr_open to 1073741816: # echo 1073741816 > /proc/sys/fs/nr_open 2. Run a program that uses a high file descriptor: #include <unistd.h> #include <sys/resource.h> int main() { struct rlimit rlim = {1073741824, 1073741824}; setrlimit(RLIMIT_NOFILE, &rlim); dup2(2, 1073741880); // Triggers the warning return 0; } 3. Observe WARNING in dmesg at mm/slub.c:5027 systemd commit a8b627a introduced automatic bumping of fs.nr_open to the maximum possible value. The rationale was that systems with memory control groups (memcg) no longer need separate file descriptor limits since memory is properly accounted. However, this change overlooked that: 1. The kernel's allocation functions still enforce INT_MAX as a maximum size regardless of memcg accounting 2. Programs and tests that legitimately test file descriptor limits can inadvertently trigger massive allocations 3. The resulting allocations (>8GB) are impractical and will always fail systemd's algorithm starts with INT_MAX and keeps halving the value until the kernel accepts it. On most systems, this results in nr_open being set to 1073741816 (0x3fffffff), which is just under 1GB of file descriptors. While processes rarely use file descriptors near this limit in normal operation, certain selftests (like tools/testing/selftests/core/unshare_test.c) and programs that test file descriptor limits can trigger this issue. Fix this by adding a check in alloc_fdtable() to ensure the requested allocation size does not exceed INT_MAX. This causes the operation to fail with -EMFILE instead of triggering a kernel warning and avoids the impractical >8GB memory allocation request.
CVE-2025-39757	In the Linux kernel, the following vulnerability has been resolved: ALSA: usb-audio: Validate UAC3 cluster segment descriptors UAC3 class segment descriptors need to be verified whether their sizes match with the declared lengths and whether they fit with the allocated buffer sizes, too. Otherwise malicious firmware may lead to the unexpected OOB accesses.
CVE-2025-39758	In the Linux kernel, the following vulnerability has been resolved: RDMA/siw: Fix the sendmsg byte count in siw_tcp_sendpages Ever since commit c2ff29e99a76 ("siw: Inline do_tcp_sendpages()"), we have been doing this: static int siw_tcp_sendpages(struct socket *, struct page **page, int offset, size_t size) [...] /* Calculate the number of bytes we need to push, for this page * specifically */ size_t bytes = min_t(size_t, PAGE_SIZE - offset, size); /* If we can't splice it, then copy it in, as normal */ if (!sendpage_ok(page[i])) msg.msg_flags &= ~MSG_SPLICE_PAGES; /* Set the bvec pointing to the page, with len \$bytes */ bvec_set_page(&bvec, page[i], bytes, offset); /* Set the iter to \$size, aka the size of the whole sendpages (!!!) */ iov_iter_bvec(&msg.msg_iter, ITER_SOURCE, &bvec, 1, size); try_page_again: lock_sock(sk); /* Sendmsg with \$size size (!!!) */ rv = tcp_sendmsg_locked(sk, &msg, size); This means we've been sending oversized iov_iters and tcp_sendmsg calls for a while. This has been a benign bug because sendpage_ok() always returned true. With the recent slab allocator changes being slowly introduced into next (that disallow sendpage on large kmalloc allocations), we have recently hit out-of-bounds crashes, due to slight differences in iov_iter behavior between the MSG_SPLICE_PAGES and "regular" copy paths: (MSG_SPLICE_PAGES) skb_splice_from_iter iov_iter_extract_pages iov_iter_extract_bvec_pages uses i->nr_segs to correctly stop in its tracks before OoB'ing everywhere skb_splice_from_iter gets a "short" read (!MSG_SPLICE_PAGES) skb_copy_to_page_nocache copy=iov_iter_count [...] copy_from_iter /* this doesn't help */ if (unlikely(iter->count < len)) len = iter->count; iterate_bvec ... and we run off the bvecs Fix this by properly setting the iov_iter's byte count, plus sending the correct byte count to tcp_sendmsg_locked.
CVE-2025-39759	In the Linux kernel, the following vulnerability has been resolved: btrfs: qgroup: fix race between quota disable and quota rescan ioctl There's a race between a task disabling quotas and another running the rescan ioctl that can result in a use-after-free of qgroup records from the fs_info->qgroup_tree rbtree. This happens as follows: 1) Task A enters btrfs_ioctl_quota_rescan() -> btrfs_qgroup_rescan(); 2) Task B enters btrfs_quota_disable() and calls btrfs_qgroup_wait_for_completion(), which does nothing because at that point fs_info->qgroup_rescan_running is false (it wasn't set yet by task A); 3) Task B calls btrfs_free_qgroup_config() which starts freeing qgroups from fs_info->qgroup_tree without taking the lock fs_info->qgroup_lock; 4) Task A enters qgroup_rescan_zero_tracking() which starts iterating the fs_info->qgroup_tree tree while holding fs_info->qgroup_lock, but task B is freeing qgroup records from that tree without holding the lock, resulting in a use-after-free. Fix this by taking fs_info->qgroup_lock at btrfs_free_qgroup_config(). Also at btrfs_qgroup_rescan() don't start the rescan worker if quotas were already disabled.

CVE ID	Description
CVE-2025-39760	In the Linux kernel, the following vulnerability has been resolved: usb: core: config: Prevent OOB read in SS endpoint companion parsing usb_parse_ss_endpoint_companion() checks descriptor type before length, enabling a potentially odd read outside of the buffer size. Fix this up by checking the size first before looking at any of the fields in the descriptor.
CVE-2025-39761	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: Decrement TID on RX peer frag setup error handling Currently, TID is not decremented before peer cleanup, during error handling path of ath12k_dp_rx_peer_frag_setup(). This could lead to out-of-bounds access in peer->rx_tid[]. Hence, add a decrement operation for TID, before peer cleanup to ensures proper cleanup and prevents out-of-bounds access issues when the RX peer frag setup fails. Found during code review. Compile tested only.
CVE-2025-39762	In the Linux kernel, the following vulnerability has been resolved: drm/amd/display: add null check [WHY] Prevents null pointer dereferences to enhance function robustness [HOW] Adds early null check and return false if invalid.
CVE-2025-39763	In the Linux kernel, the following vulnerability has been resolved: ACPI: APEI: send SIBUS to current task if synchronous memory error not recovered If a synchronous error is detected as a result of user-space process triggering a 2-bit uncorrected error, the CPU will take a synchronous error exception such as Synchronous External Abort (SEA) on Arm64. The kernel will queue a memory_failure() work which poisons the related page, unmaps the page, and then sends a SIBUS to the process, so that a system wide panic can be avoided. However, no memory_failure() work will be queued when abnormal synchronous errors occur. These errors can include situations like invalid PA, unexpected severity, no memory failure config support, invalid GUID section, etc. In such a case, the user-space process will trigger SEA again. This loop can potentially exceed the platform firmware threshold or even trigger a kernel hard lockup, leading to a system reboot. Fix it by performing a force kill if no memory_failure() work is queued for synchronous errors. [rjw: Changelog edits]
CVE-2025-39764	In the Linux kernel, the following vulnerability has been resolved: netfilter: ctnetlink: remove refcounting in expectation dumpers Same pattern as previous patch: do not keep the expectation object alive via refcount, only store a cookie value and then use that as the skip hint for dump resumption. AFAICS this has the same issue as the one resolved in the conntrack dumper, when we do if (!refcount_inc_not_zero(&exp->use)) to increment the refcount, there is a chance that exp == last, which causes a double-increment of the refcount and subsequent memory leak.
CVE-2025-39766	In the Linux kernel, the following vulnerability has been resolved: net/sched: Make cake_enqueue return NET_XMIT_CN when past buffer_limit The following setup can trigger a WARNING in htb_activate due to the condition: !cl->leaf.q->q.qlen tc qdisc del dev lo root tc qdisc add dev lo root handle 1: htb default 1 tc class add dev lo parent 1: classid 1:1 \ htb rate 64bit tc qdisc add dev lo parent 1:1 handle f: \ cake memlimit 1b ping -I lo -f -c1 -s64 -W0.001 127.0.0.1 This is because the low memlimit leads to a low buffer_limit, which causes packet dropping. However, cake_enqueue still returns NET_XMIT_SUCCESS, causing htb_enqueue to call htb_activate with an empty child qdisc. We should return NET_XMIT_CN when packets are dropped from the same tin and flow. I do not believe return value of NET_XMIT_CN is necessary for packet drops in the case of ack filtering, as that is meant to optimize performance, not to signal congestion.

CVE ID	Description
CVE-2025-39770	In the Linux kernel, the following vulnerability has been resolved: net: gso: Forbid IPv6 TSO with extensions on devices with only IPV6_CSUM When performing Generic Segmentation Offload (GSO) on an IPv6 packet that contains extension headers, the kernel incorrectly requests checksum offload if the egress device only advertises NETIF_F_IPV6_CSUM feature, which has a strict contract: it supports checksum offload only for plain TCP or UDP over IPv6 and explicitly does not support packets with extension headers. The current GSO logic violates this contract by failing to disable the feature for packets with extension headers, such as those used in GREoIPv6 tunnels. This violation results in the device being asked to perform an operation it cannot support, leading to a 'skb_warn_bad_offload' warning and a collapse of network throughput. While device TSO/USO is correctly bypassed in favor of software GSO for these packets, the GSO stack must be explicitly told not to request checksum offload. Mask NETIF_F_IPV6_CSUM, NETIF_F_TSO6 and NETIF_F_GSO_UDP_L4 in gso_features_check if the IPv6 header contains extension headers to compute checksum in software. The exception is a BIG TCP extension, which, as stated in commit 68e068cabd2c6c53 ("net: reenables NETIF_F_IPV6_CSUM offload for BIG TCP packets"): "The feature is only enabled on devices that support BIG TCP TSO. The header is only present for PF_PACKET taps like tcpdump, and not transmitted by physical devices." kernel log output (truncated): WARNING: CPU: 1 PID: 5273 at net/core/dev.c:3535 skb_warn_bad_offload+0x81/0x140 ... Call Trace: <TASK> skb_checksum_help+0x12a/0x1f0 validate_xmit_skb+0x1a3/0x2d0 validate_xmit_skb_list+0x4f/0x80 sch_direct_xmit+0x1a2/0x380 __dev_xmit_skb+0x242/0x670 __dev_queue_xmit+0x3fc/0x7f0 ip6_finish_output2+0x25e/0x5d0 ip6_finish_output+0x1fc/0x3f0 ip6_tnl_xmit+0x608/0xc00 [ip6_tunnel] ip6gre_tunnel_xmit+0x1c0/0x390 [ip6_gre] dev_hard_start_xmit+0x63/0x1c0 __dev_queue_xmit+0x6d0/0x7f0 ip6_finish_output2+0x214/0x5d0 ip6_finish_output+0x1fc/0x3f0 ip6_xmit+0x2ca/0x6f0 ip6_finish_output+0x1fc/0x3f0 ip6_xmit+0x2ca/0x6f0 inet6_csk_xmit+0xeb/0x150 __tcp_transmit_skb+0x555/0xa80 tcp_write_xmit+0x32a/0xe90 tcp_sendmsg_locked+0x437/0x1110 tcp_sendmsg+0x2f/0x50 ... skb linear: 00000000: e4 3d 1a 7d ec 30 e4 3d 1a 7e 5d 90 86 dd 60 0e skb linear: 00000010: 00 0a 1b 34 3c 40 20 11 00 00 00 00 00 00 00 00 skb linear: 00000020: 00 00 00 00 00 12 20 11 00 00 00 00 00 00 00 00 skb linear: 00000030: 00 00 00 00 00 11 2f 00 04 01 04 01 01 00 00 00 skb linear: 00000040: 86 dd 60 0e 00 0a 1b 00 06 40 20 23 00 00 00 00 skb linear: 00000050: 00 00 00 00 00 00 00 00 12 20 23 00 00 00 00 00 skb linear: 00000060: 00 00 00 00 00 00 00 00 00 11 bf 96 14 51 13 f9 skb linear: 00000070: ae 27 a0 a8 2b e3 80 18 00 40 5b 6f 00 00 01 01 skb linear: 00000080: 08 0a 42 d4 50 d5 4b 70 f8 1a
CVE-2025-39771	In the Linux kernel, the following vulnerability has been resolved: regulator: pca9450: Use devm_register_sys_off_handler With module test, there is error dump: -----[cut here]----- notifier callback pca9450_i2c_restart_handler already registered WARNING: kernel/notifier.c:23 at notifier_chain_register+0x5c/0x88, CPU#0: kworker/u16:3/50 Call trace: notifier_chain_register+0x5c/0x88 (P) atomic_notifier_chain_register+0x30/0x58 register_restart_handler+0x1c/0x28 pca9450_i2c_probe+0x418/0x538 i2c_device_probe+0x220/0x3d0 really_probe+0x114/0x410 __driver_probe_device+0xa0/0x150 driver_probe_device+0x40/0x114 __device_attach_driver+0xd4/0x12c So use devm_register_sys_off_handler to let kernel handle the resource free to avoid kernel dump.
CVE-2025-39772	In the Linux kernel, the following vulnerability has been resolved: drm/hisilicon/hibmc: fix the hibmc loaded failed bug When hibmc loaded failed, the driver use hibmc_unload to free the resource, but the mutexes in mode.config are not init, which will access an NULL pointer. Just change goto statement to return, because hibmc_hw_init() doesn't need to free anything.
CVE-2025-39773	In the Linux kernel, the following vulnerability has been resolved: net: bridge: fix soft lockup in br_multicast_query_expired() When set multicast_query_interval to a large value, the local variable 'time' in br_multicast_send_query() may overflow. If the time is smaller than jiffies, the timer will expire immediately, and then call mod_timer() again, which creates a loop and may trigger the following soft lockup issue. watchdog: BUG: soft lockup - CPU#1 stuck for 221s! [rb_consumer:66] CPU: 1 UID: 0 PID: 66 Comm: rb_consumer Not tainted 6.16.0+ #259 PREEMPT(none) Call Trace: <IRQ> __netdev_alloc_skb+0x2e/0x3a0 br_ip6_multicast_alloc_query+0x212/0x1b70 __br_multicast_send_query+0x376/0xac0 br_multicast_send_query+0x299/0x510 br_multicast_query_expired.constprop.0+0x16d/0x1b0 call_timer_fn+0x3b/0x2a0 __run_timers+0x619/0x950 run_timer_softirq+0x11c/0x220 handle_softirqs+0x18e/0x560 __irq_exit_rcu+0x158/0x1a0 sysvec_apic_timer_interrupt+0x76/0x90 </IRQ> This issue can be reproduced with: ip link add br0 type bridge echo 1 > /sys/class/net/br0/bridge/multicast_querier echo 0xffffffffffff > /sys/class/net/br0/bridge/multicast_query_interval ip link set dev br0 up The multicast_startup_query_interval can also cause this issue. Similar to the commit 99b40610956a ("net: bridge: mcast: add and enforce query interval minimum"), add check for the query interval maximum to fix this issue.

CVE ID	Description
CVE-2025-39776	In the Linux kernel, the following vulnerability has been resolved: mm/debug_vm_pgtable: clear page table entries at destroy_args() The mm/debug_vm_pagetable test allocates manually page table entries for the tests it runs, using also its manually allocated mm_struct. That in itself is ok, but when it exits, at destroy_args() it fails to clear those entries with the *_clear functions. The problem is that leaves stale entries. If another process allocates an mm_struct with a pgd at the same address, it may end up running into the stale entry. This is happening in practice on a debug kernel with CONFIG_DEBUG_VM_PGTABLE=y, for example this is the output with some extra debugging I added (it prints a warning trace if pgtables_bytes goes negative, in addition to the warning at check_mm() function): [2.539353] debug_vm_pgtable: [get_random_vaddr]: random_vaddr is 0x7ea247140000 [2.539366] kmem_cache info [2.539374] kmem_cache 0x000000002ce82385 - freelist 0x0000000000000000 - offset 0x508 [2.539447] debug_vm_pgtable: [init_args]: args->mm is 0x000000002267cc9e (...) [2.552800] WARNING: CPU: 5 PID: 116 at include/linux/mm.h:2841 free_pud_range+0x8bc/0x8d0 [2.552816] Modules linked in: [2.552843] CPU: 5 UID: 0 PID: 116 Comm: modprobe Not tainted 6.12.0-105.debug_vm2.el10.ppc64le+debug #1 VOLUNTARY [2.552859] Hardware name: IBM,9009-41A POWER9 (architected) 0x4e0202 0xf000005 of:IBM,FW910.00 (VL910_062) hv:phyp pSeries [2.552872] NIP: c0000000007eef3c LR: c0000000007eef30 CTR: c0000000003d8c90 [2.552885] REGS: c0000000622e73b0 TRAP: 0700 Not tainted (6.12.0-105.debug_vm2.el10.ppc64le+debug) [2.552899] MSR: 800000000282b033 <SF,VEC,VXS,EE,FP,ME,IR,DR,RI,LE> CR: 24002822 XER: 0000000a [2.552954] CFAR: c0000000008f03f0 IRQMASK: 0 [2.552954] GPR00: c0000000007eef30 c0000000622e7650 c000000002b1ac00 0000000000000001 [2.552954] GPR04: 0000000000000008 0000000000000000 c0000000007eef30 ffffffff [2.552954] GPR08: 00000000ffff00f5 0000000000000001 0000000000000048 0000000000004000 [2.552954] GPR12: 00000003fa440000 c000000017ffa300 c0000000051d9f80 ffffffff [2.552954] GPR16: 0000000000000000 0000000000000008 000000000000000a 60000000000000e0 [2.552954] GPR20: 4080000000000000 c0000000113af038 00007ffcf130000 0000700000000000 [2.552954] GPR24: c000000062a6a000 0000000000000001 8000000062a68000 0000000000000001 [2.552954] GPR28: 000000000000000a c000000062ebc600 0000000000002000 c000000062ebc760 [2.553170] NIP [c0000000007eef3c] free_pud_range+0x8bc/0x8d0 [2.553185] LR [c0000000007eef30] free_pud_range+0x8b0/0x8d0 [2.553199] Call Trace: [2.553207] [c0000000622e7650] [c0000000007eef30] free_pud_range+0x8b0/0x8d0 (unreliable) [2.553229] [c0000000622e7750] [c0000000007f40b4] free_pgd_range+0x284/0x3b0 [2.553248] [c0000000622e7800] [c0000000007f4630] free_pgtables+0x450/0x570 [2.553274] [c0000000622e78e0] [c0000000008161c0] exit_mmap+0x250/0x650 [2.553292] [c0000000622e7a30] [c0000000001b95b8] __mmap+0x98/0x290 [2.558344] [c0000000622e7a80] [c0000000001d1018] exit_mm+0x118/0x1b0 [2.558361] [c0000000622e7ac0] [c0000000001d141c] do_exit+0x2ec/0x870 [2.558376] [c0000000622e7b60] [c0000000001d1ca8] do_group_exit+0x88/0x150 [2.558391] [c0000000622e7bb0] [c0000000001d1db8] sys_exit_group+0x48/0x50 [2.558407] [c0000000622e7be0] [c00000000003d810] system_call_exception+0x1e0/0x4c0 [2.558423] [c0000000622e7e50] [c0000000000d05c] system_call_vectored_common+0x15c/0x2ec (...) [2.558892] ---[end trace 0000000000000000]--- [2.559022] BUG: Bad rss-counter state mm:000000002267cc9e type:MM_ANONPAGES val:1 [2.559037] BUG: non-zero pgtables_bytes on freeing mm: -6144 Here the modprobe process ended up with an allocated mm_struct from the mm_struct slab that was used before by the debug_vm_pgtable test. That is not a problem, since the mm_stru ---truncated---

CVE ID	Description
CVE-2025-39779	In the Linux kernel, the following vulnerability has been resolved: btrfs: subpage: keep TOWRITE tag until folio is cleaned btrfs_subpage_set_writeback() calls folio_start_writeback() the first time a folio is written back, and it also clears the PAGECACHE_TAG_TOWRITE tag even if there are still dirty blocks in the folio. This can break ordering guarantees, such as those required by btrfs_wait_ordered_extents(). That ordering breakage leads to a real failure. For example, running generic/464 on a zoned setup will hit the following ASSERT. This happens because the broken ordering fails to flush existing dirty pages before the file size is truncated. assertion failed: ! list_empty(&ordered->list) :: 0, in fs/btrfs/zoned.c:1899 -----[cut here]----- kernel BUG at fs/btrfs/zoned.c:1899! Oops: invalid opcode: 0000 [#1] SMP NOPTI CPU: 2 UID: 0 PID: 1906169 Comm: kworker/u130:2 Kdump: loaded Not tainted 6.16.0-rc6-BTRFS-ZNS+ #554 PREEMPT(voluntary) Hardware name: Supermicro Super Server/H12SSL-NT, BIOS 2.0 02/22/2021 Workqueue: btrfs-endio-write btrfs_work_helper [btrfs] RIP: 0010:btrfs_finish_ordered_zoned.cold+0x50/0x52 [btrfs] RSP: 0018:ffffc9002efdbd60 EFLAGS: 00010246 RAX: 000000000000004c RBX: ffff88811923c4e0 RCX: 0000000000000000 RDX: 0000000000000000 RSI: ffffffff827e38b1 RDI: 00000000fffffffb RBP: ffff88810005d000 R08: 00000000ffffdfff R09: ffffffff831051c8 R10: ffffffff83055220 R11: 0000000000000000 R12: ffff8881c2458c00 R13: ffff88811923c540 R14: ffff88811923c5e8 R15: ffff8881c1bd9680 FS: 0000000000000000(0000) GS:ffff88a04acd0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007f907c7a918c CR3: 0000000004024000 CR4: 0000000000350ef0 Call Trace: <TASK> ? srso_return_thunk+0x5/0x5f btrfs_finish_ordered_io+0x4a/0x60 [btrfs] btrfs_work_helper+0xf9/0x490 [btrfs] process_one_work+0x204/0x590 ? srso_return_thunk+0x5/0x5f worker_thread+0x1d6/0x3d0 ? __pfx_worker_thread+0x10/0x10 kthread+0x118/0x230 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x205/0x260 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK> Consider process A calling writepages() with WB_SYNC_NONE. In zoned mode or for compressed writes, it locks several folios for delalloc and starts writing them out. Let's call the last locked folio folio X. Suppose the write range only partially covers folio X, leaving some pages dirty. Process A calls btrfs_subpage_set_writeback() when building a bio. This function call clears the TOWRITE tag of folio X, whose size = 8K and the block size = 4K. It is following state. 0 4K 8K (flag: DIRTY, tag: DIRTY) <-----> Process A will write this range. Now suppose process B concurrently calls writepages() with WB_SYNC_ALL. It calls tag_pages_for_writeback() to tag dirty folios with PAGECACHE_TAG_TOWRITE. Since folio X is still dirty, it gets tagged. Then, B collects tagged folios using filemap_get_folios_tag() and must wait for folio X to be written before returning from writepages(). 0 4K 8K (flag: DIRTY, tag: DIRTY TOWRITE) However, between tagging and collecting, process A may call btrfs_subpage_set_writeback() and clear folio X's TOWRITE tag. 0 4K 8K (flag: DIRTY WRITEBACK, tag: DIRTY) As a result, process B won't see folio X in its batch, and returns without waiting for it. This breaks the WB_SYNC_ALL ordering requirement. Fix this by using btrfs_subpage_set_writeback_keepwrite(), which retains the TOWRITE tag. We now manually clear the tag only after the folio becomes clean, via the xas operation.
CVE-2025-39781	In the Linux kernel, the following vulnerability has been resolved: parisc: Drop WARN_ON_ONCE() from flush_cache_vmap I have observed warning to occasionally trigger.
CVE-2025-39782	In the Linux kernel, the following vulnerability has been resolved: jbd2: prevent softlockup in jbd2_log_do_checkpoint() Both jbd2_log_do_checkpoint() and jbd2_journal_shrink_checkpoint_list() periodically release j_list_lock after processing a batch of buffers to avoid long hold times on the j_list_lock. However, since both functions contend for j_list_lock, the combined time spent waiting and processing can be significant. jbd2_journal_shrink_checkpoint_list() explicitly calls cond_resched() when need_resched() is true to avoid softlockups during prolonged operations. But jbd2_log_do_checkpoint() only exits its loop when need_resched() is true, relying on potentially sleeping functions like __flush_batch() or wait_on_buffer() to trigger rescheduling. If those functions do not sleep, the kernel may hit a softlockup. watchdog: BUG: soft lockup - CPU#3 stuck for 156s! [kworker/u129:2:373] CPU: 3 PID: 373 Comm: kworker/u129:2 Kdump: loaded Not tainted 6.6.0+ #10 Hardware name: Huawei TaiShan 2280 /BC11SPCD, BIOS 1.27 06/13/2017 Workqueue: writeback wb_workfn (flush-7:2) pstate: 20000005 (nzCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : native_queued_spin_lock_slowpath+0x358/0x418 lr : jbd2_log_do_checkpoint+0x31c/0x438 [jbd2] Call trace: native_queued_spin_lock_slowpath+0x358/0x418 jbd2_log_do_checkpoint+0x31c/0x438 [jbd2] __jbd2_log_wait_for_space+0xf8/0x2f8 [jbd2] add_transaction_credits+0x3bc/0x418 [jbd2] start_this_handle+0xf8/0x560 [jbd2] jbd2__journal_start+0x118/0x228 [jbd2] __ext4_journal_start_sb+0x110/0x188 [ext4] ext4_do_writepages+0x3dc/0x740 [ext4] ext4_writepages+0xa4/0x190 [ext4] do_writepages+0x94/0x228 __writeback_single_inode+0x48/0x318 writeback_sb_inodes+0x204/0x590 __writeback_inodes_wb+0x54/0xf8 wb_writeback+0x2cc/0x3d8 wb_do_writeback+0x2e0/0x2f8 wb_workfn+0x80/0x2a8 process_one_work+0x178/0x3e8 worker_thread+0x234/0x3b8 kthread+0xf0/0x108 ret_from_fork+0x10/0x20 So explicitly call cond_resched() in jbd2_log_do_checkpoint() to avoid softlockup.

CVE ID	Description
CVE-2025-39783	In the Linux kernel, the following vulnerability has been resolved: PCI: endpoint: Fix configfs group list head handling Doing a list_del() on the epf_group field of struct pci_epf_driver in pci_epf_remove_cfs() is not correct as this field is a list head, not a list entry. This list_del() call triggers a KASAN warning when an endpoint function driver which has a configfs attribute group is torn down: ===== BUG: KASAN: slab-use-after-free in pci_epf_remove_cfs+0x17c/0x198 Write of size 8 at addr ffff00010f4a0d80 by task rmmmod/319 CPU: 3 UID: 0 PID: 319 Comm: rmmmod Not tainted 6.16.0-rc2 #1 NONE Hardware name: Radxa ROCK 5B (DT) Call trace: show_stack+0x2c/0x84 (C) dump_stack_lvl+0x70/0x98 print_report+0x17c/0x538 kasan_report+0xb8/0x190 __asan_report_store8_noabort+0x20/0x2c pci_epf_remove_cfs+0x17c/0x198 pci_epf_unregister_driver+0x18/0x30 nvmet_pci_epf_cleanup_module+0x24/0x30 [nvmet_pci_epf] __arm64_sys_delete_module+0x264/0x424 invoke_syscall+0x70/0x260 el0_svc_common.constprop.0+0xac/0x230 do_el0_svc+0x40/0x58 el0_svc+0x48/0xdc el0t_64_sync_handler+0x10c/0x138 el0t_64_sync+0x198/0x19c ... Remove this incorrect list_del() call from pci_epf_remove_cfs().
CVE-2025-39787	In the Linux kernel, the following vulnerability has been resolved: soc: qcom: mdt_loader: Ensure we don't read past the ELF header When the MDT loader is used in remoteproc, the ELF header is sanitized beforehand, but that's not necessary the case for other clients. Validate the size of the firmware buffer to ensure that we don't read past the end as we iterate over the header. e_phentsize and e_shentsize are validated as well, to ensure that the assumptions about step size in the traversal are valid.
CVE-2025-39788	In the Linux kernel, the following vulnerability has been resolved: scsi: ufs: exynos: Fix programming of HCI_UTRL_NEXUS_TYPE On Google gs101, the number of UTP transfer request slots (nutrs) is 32, and in this case the driver ends up programming the UTRL_NEXUS_TYPE incorrectly as 0. This is because the left hand side of the shift is 1, which is of type int, i.e. 31 bits wide. Shifting by more than that width results in undefined behaviour. Fix this by switching to the BIT() macro, which applies correct type casting as required. This ensures the correct value is written to UTRL_NEXUS_TYPE (0xffffffff on gs101), and it also fixes a UBSAN shift warning: UBSAN: shift-out-of-bounds in drivers/ufs/host/ufs-exynos.c:1113:21 shift exponent 32 is too large for 32-bit type 'int' For consistency, apply the same change to the nutmrs / UTMRL_NEXUS_TYPE write.
CVE-2025-39789	In the Linux kernel, the following vulnerability has been resolved: crypto: x86/aegis - Add missing error checks The skcipher_walk functions can allocate memory and can fail, so checking for errors is necessary.
CVE-2025-39790	In the Linux kernel, the following vulnerability has been resolved: bus: mhi: host: Detect events pointing to unexpected TREs When a remote device sends a completion event to the host, it contains a pointer to the consumed TRE. The host uses this pointer to process all of the TREs between it and the host's local copy of the ring's read pointer. This works when processing completion for chained transactions, but can lead to nasty results if the device sends an event for a single-element transaction with a read pointer that is multiple elements ahead of the host's read pointer. For instance, if the host accesses an event ring while the device is updating it, the pointer inside of the event might still point to an old TRE. If the host uses the channel's xfer_cb() to directly free the buffer pointed to by the TRE, the buffer will be double-freed. This behavior was observed on an ep that used upstream EP stack without 'commit 6f18d174b73d ("bus: mhi: ep: Update read pointer only after buffer is written")'. Where the device updated the events ring pointer before updating the event contents, so it left a window where the host was able to access the stale data the event pointed to, before the device had the chance to update them. The usual pattern was that the host received an event pointing to a TRE that is not immediately after the last processed one, so it got treated as if it was a chained transaction, processing all of the TREs in between the two read pointers. This commit aims to harden the host by ensuring transactions where the event points to a TRE that isn't local_rp + 1 are chained. [mani: added stable tag and reworded commit message]
CVE-2025-39794	In the Linux kernel, the following vulnerability has been resolved: ARM: tegra: Use I/O memcpy to write to IRAM Kasan crashes the kernel trying to check boundaries when using the normal memcpy.
CVE-2025-39795	In the Linux kernel, the following vulnerability has been resolved: block: avoid possible overflow for chunk_sectors check in blk_stack_limits() In blk_stack_limits(), we check that the t->chunk_sectors value is a multiple of the t->physical_block_size value. However, by finding the chunk_sectors value in bytes, we may overflow the unsigned int which holds chunk_sectors, so change the check to be based on sectors.

CVE ID	Description
CVE-2025-39797	In the Linux kernel, the following vulnerability has been resolved: xfrm: Duplicate SPI Handling The issue originates when Strongswan initiates an XFRM_MSG_ALLOCSPI Netlink message, which triggers the kernel function xfrm_alloc_spi(). This function is expected to ensure uniqueness of the Security Parameter Index (SPI) for inbound Security Associations (SAs). However, it can return success even when the requested SPI is already in use, leading to duplicate SPIs assigned to multiple inbound SAs, differentiated only by their destination addresses. This behavior causes inconsistencies during SPI lookups for inbound packets. Since the lookup may return an arbitrary SA among those with the same SPI, packet processing can fail, resulting in packet drops. According to RFC 4301 section 4.4.2, for inbound processing a unicast SA is uniquely identified by the SPI and optionally protocol. Reproducing the Issue Reliably: To consistently reproduce the problem, restrict the available SPI range in charon.conf : spi_min = 0x10000000 spi_max = 0x10000002 This limits the system to only 2 usable SPI values. Next, create more than 2 Child SA. each using unique pair of src/dst address. As soon as the 3rd Child SA is initiated, it will be assigned a duplicate SPI, since the SPI pool is already exhausted. With a narrow SPI range, the issue is consistently reproducible. With a broader/default range, it becomes rare and unpredictable. Current implementation: xfrm_spi_hash() lookup function computes hash using daddr, proto, and family. So if two SAs have the same SPI but different destination addresses, then they will: a. Hash into different buckets b. Be stored in different linked lists (byspi + h) c. Not be seen in the same hlist_for_each_entry_rcu() iteration. As a result, the lookup will result in NULL and kernel allows that Duplicate SPI Proposed Change: xfrm_state_lookup_spi_proto() does a truly global search - across all states, regardless of hash bucket and matches SPI and proto.
CVE-2025-39798	In the Linux kernel, the following vulnerability has been resolved: NFS: Fix the setting of capabilities when automounting a new filesystem Capabilities cannot be inherited when we cross into a new filesystem. They need to be reset to the minimal defaults, and then probed for again.
CVE-2025-39800	In the Linux kernel, the following vulnerability has been resolved: btrfs: abort transaction on unexpected eb generation at btrfs_copy_root() If we find an unexpected generation for the extent buffer we are cloning at btrfs_copy_root(), we just WARN_ON() and don't error out and abort the transaction, meaning we allow to persist metadata with an unexpected generation. Instead of warning only, abort the transaction and return -EUCLEAN.
CVE-2025-39801	In the Linux kernel, the following vulnerability has been resolved: usb: dwc3: Remove WARN_ON for device endpoint command timeouts This commit addresses a rarely observed endpoint command timeout which causes kernel panic due to warn when 'panic_on_warn' is enabled and unnecessary call trace prints when 'panic_on_warn' is disabled. It is seen during fast software-controlled connect/disconnect testcases. The following is one such endpoint command timeout that we observed: 1. Connect =====>dwc3_thread_interrupt->dwc3_ep0_interrupt->configs_composite_setup->composite_setup->usb_ep_queue->dwc3_gadget_ep0_queue->__dwc3_gadget_ep0_queue->__dwc3_ep0_do_control_data->dwc3_send_gadget_ep_cmd 2. Disconnect =====>dwc3_thread_interrupt->dwc3_gadget_disconnect_interrupt->dwc3_ep0_reset_state->dwc3_ep0_end_control_data->dwc3_send_gadget_ep_cmd In the issue scenario, in Exynos platforms, we observed that control transfers for the previous connect have not yet been completed and end transfer command sent as a part of the disconnect sequence and processing of USB_ENDPOINT_HALT feature request from the host timeout. This maybe an expected scenario since the controller is processing EP commands sent as a part of the previous connect. It maybe better to remove WARN_ON in all places where device endpoint commands are sent to avoid unnecessary kernel panic due to warn.
CVE-2025-39805	In the Linux kernel, the following vulnerability has been resolved: net: macb: fix unregister_netdev call order in macb_remove() When removing a macb device, the driver calls phy_exit() before unregister_netdev(). This leads to a WARN from kernfs: -----[cut here]----- kernfs: can not remove 'attached_dev', no directory WARNING: CPU: 1 PID: 27146 at fs/kernfs/dir.c:1683 Call trace: kernfs_remove_by_name_ns+0xd8/0xf0 sysfs_remove_link+0x24/0x58 phy_detach+0x5c/0x168 phy_disconnect+0x4c/0x70 phylink_disconnect_phy+0x6c/0xc0 [phylink] macb_close+0x6c/0x170 [macb] ... macb_remove+0x60/0x168 [macb] platform_remove+0x5c/0x80 ... The warning happens because the PHY is being exited while the netdev is still registered. The correct order is to unregister the netdev before shutting down the PHY and cleaning up the MDIO bus. Fix this by moving unregister_netdev() ahead of phy_exit() in macb_remove().

CVE ID	Description
CVE-2025-39806	In the Linux kernel, the following vulnerability has been resolved: HID: multitouch: fix slab out-of-bounds access in mt_report_fixup() A malicious HID device can trigger a slab out-of-bounds during mt_report_fixup() by passing in report descriptor smaller than 607 bytes. mt_report_fixup() attempts to patch byte offset 607 of the descriptor with 0x25 by first checking if byte offset 607 is 0x15 however it lacks bounds checks to verify if the descriptor is big enough before conducting this check. Fix this bug by ensuring the descriptor size is at least 608 bytes before accessing it. Below is the KASAN splat after the out of bounds access happens: [13.671954] ===== [13.672667] BUG: KASAN: slab-out-of-bounds in mt_report_fixup+0x103/0x110 [13.673297] Read of size 1 at addr ffff888103df39df by task kworker/0:1/10 [13.673297] [13.673297] CPU: 0 UID: 0 PID: 10 Comm: kworker/0:1 Not tainted 6.15.0-00005-gec5d573d83f4-dirty #3 [13.673297] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.2-debian-1.16.2-1 04/04 [13.673297] Call Trace: [13.673297] <TASK> [13.673297] dump_stack_lvl+0x5f/0x80 [13.673297] print_report+0xd1/0x660 [13.673297] kasan_report+0xe5/0x120 [13.673297] __asan_report_load1_noabort+0x18/0x20 [13.673297] mt_report_fixup+0x103/0x110 [13.673297] hid_open_report+0x1ef/0x810 [13.673297] mt_probe+0x422/0x960 [13.673297] hid_device_probe+0x2e2/0x6f0 [13.673297] really_probe+0x1c6/0x6b0 [13.673297] __driver_probe_device+0x24f/0x310 [13.673297] driver_probe_device+0x4e/0x220 [13.673297] __device_attach_driver+0x169/0x320 [13.673297] bus_for_each_drv+0x11d/0x1b0 [13.673297] __device_attach+0x1b8/0x3e0 [13.673297] device_initial_probe+0x12/0x20 [13.673297] bus_probe_device+0x13d/0x180 [13.673297] device_add+0xe3a/0x1670 [13.673297] hid_add_device+0x31d/0xa40 [...]
CVE-2025-39808	In the Linux kernel, the following vulnerability has been resolved: HID: hid-ntrig: fix unable to handle page fault in ntrig_report_version() in ntrig_report_version(), hdev parameter passed from hid_probe(). sending descriptor to /dev/uhid can make hdev->dev.parent->parent to null if hdev->dev.parent->parent is null, usb_dev has invalid address(0xffffffffffff58) that hid_to_usb_dev(hdev) returned when usb_rcvctrlpipe() use usb_dev, it trigger page fault error for address(0xffffffffffff58) add null check logic to ntrig_report_version() before calling hid_to_usb_dev()
CVE-2025-39810	In the Linux kernel, the following vulnerability has been resolved: bnxt_en: Fix memory corruption when FW resources change during ifdown bnxt_set_dflt_rings() assumes that it is always called before any TC has been created. So it doesn't take bp->num_tc into account and assumes that it is always 0 or 1. In the FW resource or capability change scenario, the FW will return flags in bnxt_hwrw_if_change() that will cause the driver to reinitialize and call bnxt_cancel_reservations(). This will lead to bnxt_init_dflt_ring_mode() calling bnxt_set_dflt_rings() and bp->num_tc may be greater than 1. This will cause bp->tx_ring[] to be sized too small and cause memory corruption in bnxt_alloc_cp_rings(). Fix it by properly scaling the TX rings by bp->num_tc in the code paths mentioned above. Add 2 helper functions to determine bp->tx_nr_rings and bp->tx_nr_rings_per_tc.
CVE-2025-39811	In the Linux kernel, the following vulnerability has been resolved: drm/xe/vm: Clear the scratch_pt pointer on error Avoid triggering a dereference of an error pointer on cleanup in xe_vm_free_scratch() by clearing any scratch_pt error pointer. (cherry picked from commit 358ee50ab565f3c8ea32480e9d03127a81ba32f8)
CVE-2025-39812	In the Linux kernel, the following vulnerability has been resolved: sctp: initialize more fields in sctp_v6_from_sk() syzbot found that sin6_scope_id was not properly initialized, leading to undefined behavior. Clear sin6_scope_id and sin6_flowinfo. BUG: KMSAN: uninit-value in __sctp_v6_cmp_addr+0x887/0x8c0 net/sctp/ipv6.c:649 __sctp_v6_cmp_addr+0x887/0x8c0 net/sctp/ipv6.c:649 sctp_inet6_cmp_addr+0x4f2/0x510 net/sctp/ipv6.c:983 sctp_bind_addr_conflict+0x22a/0x3b0 net/sctp/bind_addr.c:390 sctp_get_port_local+0x21eb/0x2440 net/sctp/socket.c:8452 sctp_get_port net/sctp/socket.c:8523 [inline] sctp_listen_start net/sctp/socket.c:8567 [inline] sctp_inet_listen+0x710/0xfdf0 net/sctp/socket.c:8636 __sys_listen_socket net/socket.c:1912 [inline] __sys_listen net/socket.c:1927 [inline] __do_sys_listen net/socket.c:1932 [inline] __se_sys_listen net/socket.c:1930 [inline] __x64_sys_listen+0x343/0x4c0 net/socket.c:1930 x64_sys_call+0x271d/0x3e20 arch/x86/include/generated/asm/syscalls_64.h:51 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xd9/0x210 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f Local variable addr.i.i created at: sctp_get_port net/sctp/socket.c:8515 [inline] sctp_listen_start net/sctp/socket.c:8567 [inline] sctp_inet_listen+0x650/0xfdf0 net/sctp/socket.c:8636 __sys_listen_socket net/socket.c:1912 [inline] __sys_listen net/socket.c:1927 [inline] __do_sys_listen net/socket.c:1932 [inline] __se_sys_listen net/socket.c:1930 [inline] __x64_sys_listen+0x343/0x4c0 net/socket.c:1930

CVE ID	Description
CVE-2025-39813	In the Linux kernel, the following vulnerability has been resolved: ftrace: Fix potential warning in trace_printk_seq during ftrace_dump When calling ftrace_dump_one() concurrently with reading trace_pipe, a WARN_ON_ONCE() in trace_printk_seq() can be triggered due to a race condition. The issue occurs because: CPU0 (ftrace_dump) CPU1 (reader) echo z > /proc/sysrq-trigger ! trace_empty(&iter) trace_iterator_reset(&iter) <- len = size = 0 cat /sys/kernel/tracing/trace_pipe trace_find_next_entry_inc(&iter) __find_next_entry ring_buffer_empty_cpu <- all empty return NULL trace_printk_seq(&iter.seq) WARN_ON_ONCE(s->seq.len >= s->seq.size) In the context between trace_empty() and trace_find_next_entry_inc() during ftrace_dump, the ring buffer data was consumed by other readers. This caused trace_find_next_entry_inc to return NULL, failing to populate `iter.seq`. At this point, due to the prior trace_iterator_reset, both `iter.seq.len` and `iter.seq.size` were set to 0. Since they are equal, the WARN_ON_ONCE condition is triggered. Move the trace_printk_seq() into the if block that checks to make sure the return value of trace_find_next_entry_inc() is non-NULL in ftrace_dump_one(), ensuring the `iter.seq` is properly populated before subsequent operations.
CVE-2025-39815	In the Linux kernel, the following vulnerability has been resolved: RISC-V: KVM: fix stack overrun when loading vlenb The userspace load can put up to 2048 bits into an xlen bit stack buffer. We want only xlen bits, so check the size beforehand.
CVE-2025-39817	In the Linux kernel, the following vulnerability has been resolved: efivarfs: Fix slab-out-of-bounds in efivarfs_d_compare Observed on kernel 6.6 (present on master as well): BUG: KASAN: slab-out-of-bounds in memcmp+0x98/0xd0 Call trace: kasan_check_range+0xe8/0x190 __asan_loadN+0x1c/0x28 memcmp+0x98/0xd0 efivarfs_d_compare+0x68/0xd8 __d_lookup_rcu_op_compare+0x178/0x218 __d_lookup_rcu+0x1f8/0x228 d_alloc_parallel+0x150/0x648 lookup_open.isra.0+0x5f0/0x8d0 open_last_lookups+0x264/0x828 path_openat+0x130/0x3f8 do_filp_open+0x114/0x248 do_sys_openat2+0x340/0x3c0 __arm64_sys_openat+0x120/0x1a0 If dentry->d_name.len < EFI_VARIABLE_GUID_LEN, 'guid' can become negative, leadings to oob. The issue can be triggered by parallel lookups using invalid filename: T1 T2 lookup_open ->lookup simple_lookup d_add // invalid dentry is added to hash list lookup_open d_alloc_parallel __d_lookup_rcu __d_lookup_rcu_op_compare hlist_bl_for_each_entry_rcu // invalid dentry can be retrieved ->d_compare efivarfs_d_compare // oob Fix it by checking 'guid' before cmp.
CVE-2025-39819	In the Linux kernel, the following vulnerability has been resolved: fs/smb: Fix inconsistent refcnt update A possible inconsistent update of refcount was identified in `smb2_compound_op`. Such inconsistent update could lead to possible resource leaks. Why it is a possible bug: 1. In the comment section of the function, it clearly states that the reference to `cfile` should be dropped after calling this function. 2. Every control flow path would check and drop the reference to `cfile`, except the patched one. 3. Existing callers would not handle refcount update of `cfile` if -ENOMEM is returned. To fix the bug, an extra goto label "out" is added, to make sure that the cleanup logic would always be respected. As the problem is caused by the allocation failure of `vars`, the cleanup logic between label "finished" and "out" can be safely ignored. According to the definition of function `is_replayable_error`, the error code of "-ENOMEM" is not recoverable. Therefore, the replay logic also gets ignored.
CVE-2025-39823	In the Linux kernel, the following vulnerability has been resolved: KVM: x86: use array_index_nospec with indices that come from guest min and dest_id are guest-controlled indices. Using array_index_nospec() after the bounds checks clamps these values to mitigate speculative execution side-channels.

CVE ID	Description
CVE-2025-39824	In the Linux kernel, the following vulnerability has been resolved: HID: asus: fix UAF via HID_CLAIMED_INPUT validation After hid_hw_start() is called hidinput_connect() will eventually be called to set up the device with the input layer since the HID_CONNECT_DEFAULT connect mask is used. During hidinput_connect() all input and output reports are processed and corresponding hid_inputs are allocated and configured via hidinput_configure_usages(). This process involves slot tagging report fields and configuring usages by setting relevant bits in the capability bitmaps. However it is possible that the capability bitmaps are not set at all leading to the subsequent hidinput_has_been_populated() check to fail leading to the freeing of the hid_input and the underlying input device. This becomes problematic because a malicious HID device like a ASUS ROG N-Key keyboard can trigger the above scenario via a specially crafted descriptor which then leads to a user-after-free when the name of the freed input device is written to later on after hid_hw_start(). Below, report 93 intentionally utilises the HID_UP_UNDEFINED Usage Page which is skipped during usage configuration, leading to the frees. 0x05, 0x0D, // Usage Page (Digitizer) 0x09, 0x05, // Usage (Touch Pad) 0xA1, 0x01, // Collection (Application) 0x85, 0x0D, // Report ID (13) 0x06, 0x00, 0xFF, // Usage Page (Vendor Defined 0xFF00) 0x09, 0xC5, // Usage (0xC5) 0x15, 0x00, // Logical Minimum (0) 0x26, 0xFF, 0x00, // Logical Maximum (255) 0x75, 0x08, // Report Size (8) 0x95, 0x04, // Report Count (4) 0xB1, 0x02, // Feature (Data,Var,Abs) 0x85, 0x5D, // Report ID (93) 0x06, 0x00, 0x00, // Usage Page (Undefined) 0x09, 0x01, // Usage (0x01) 0x15, 0x00, // Logical Minimum (0) 0x26, 0xFF, 0x00, // Logical Maximum (255) 0x75, 0x08, // Report Size (8) 0x95, 0x1B, // Report Count (27) 0x81, 0x02, // Input (Data,Var,Abs) 0xC0, // End Collection Below is the KASAN splat after triggering the UAF: [21.672709] ===== <pre>[21.673700] BUG: KASAN: slab-use-after-free in asus_probe+0xeeb/0xf80 [21.673700] Write of size 8 at addr ffff88810a0ac000 by task kworker/1:2/54 [21.673700] [21.673700] CPU: 1 UID: 0 PID: 54 Comm: kworker/1:2 Not tainted 6.16.0-rc4-g9773391cf4dd-dirty #36 PREEMPT(voluntary) [21.673700] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.2-debian-1.16.2-1 04/01/2014 [21.673700] Call Trace: [21.673700] <TASK> [21.673700] dump_stack_lvl+0x5f/0x80 [21.673700] print_report+0xd1/0x660 [21.673700] kasan_report +0xe5/0x120 [21.673700] __asan_report_store8_noabort+0x1b/0x30 [21.673700] asus_probe +0xeeb/0xf80 [21.673700] hid_device_probe+0x2ee/0x700 [21.673700] really_probe+0x1c6/0x6b0 [21.673700] __driver_probe_device+0x24f/0x310 [21.673700] driver_probe_device+0x4e/0x220 [...] [21.673700] [21.673700] Allocated by task 54: [21.673700] kasan_save_stack+0x3d/0x60 [21.673700] kasan_save_track+0x18/0x40 [21.673700] kasan_save_alloc_info+0x3b/0x50 [21.673700] __kasan_kmalloc+0x9c/0xa0 [21.673700] __kmalloc_cache_noprof+0x139/0x340 [21.673700] input_allocate_device+0x44/0x370 [21.673700] hidinput_connect+0xcb6/0x2630 [21.673700] hid_connect+0xf74/0x1d60 [21.673700] hid_hw_start+0x8c/0x110 [21.673700] asus_probe+0x5a3/0xf80 [21.673700] hid_device_probe+0x2ee/0x700 [21.673700] really_probe +0x1c6/0x6b0 [21.673700] __driver_probe_device+0x24f/0x310 [21.673700] driver_probe_device +0x4e/0x220 [...] [21.673700] [21.673700] Freed by task 54: [21.673700] kasan_save_stack +0x3d/0x60 [21.673700] kasan_save_track+0x18/0x40 [21.673700] kasan_save_free_info+0x3f/0x60 [21.673700] __kasan_slab_free+0x3c/0x50 [21.673700] kfre ---truncated---</pre>
CVE-2025-39825	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix race with concurrent opens in rename(2) Besides sending the rename request to the server, the rename process also involves closing any deferred close, waiting for outstanding I/O to complete as well as marking all existing open handles as deleted to prevent them from deferring closes, which increases the race window for potential concurrent opens on the target file. Fix this by unhashing the dentry in advance to prevent any concurrent opens on the target.
CVE-2025-39826	In the Linux kernel, the following vulnerability has been resolved: net: rose: convert 'use' field to refcount_t The 'use' field in struct rose_neigh is used as a reference counter but lacks atomicity. This can lead to race conditions where a rose_neigh structure is freed while still being referenced by other code paths. For example, when rose_neigh->use becomes zero during an ioctl operation via rose_rt_ioctl(), the structure may be removed while its timer is still active, potentially causing use-after-free issues. This patch changes the type of 'use' from unsigned short to refcount_t and updates all code paths to use rose_neigh_hold() and rose_neigh_put() which operate reference counts atomically.
CVE-2025-39827	In the Linux kernel, the following vulnerability has been resolved: net: rose: include node references in rose_neigh refcount Current implementation maintains two separate reference counting mechanisms: the 'count' field in struct rose_neigh tracks references from rose_node structures, while the 'use' field (now refcount_t) tracks references from rose_sock. This patch merges these two reference counting systems using 'use' field for proper reference management. Specifically, this patch adds incrementing and decrementing of rose_neigh->use when rose_neigh->count is incremented or decremented. This patch also modifies rose_rt_free(), rose_rt_device_down() and rose_clear_route() to properly release references to rose_neigh objects before freeing a rose_node through rose_remove_node(). These changes ensure rose_neigh structures are properly freed only when all references, including those from rose_node structures, are released. As a result, this resolves a slab-use-after-free issue reported by Syzbot.

CVE ID	Description
CVE-2025-39828	In the Linux kernel, the following vulnerability has been resolved: atm: atmtcp: Prevent arbitrary write in atmtcp_recv_control(). syzbot reported the splat below. [0] When atmtcp_v_open() or atmtcp_v_close() is called via connect() or close(), atmtcp_send_control() is called to send an in-kernel special message. The message has ATMTCP_HDR_MAGIC in atmtcp_control.hdr.length. Also, a pointer of struct atm_vcc is set to atmtcp_control.vcc. The notable thing is struct atmtcp_control is uAPI but has a space for an in-kernel pointer. struct atmtcp_control { struct atmtcp_hdr hdr; /* must be first */ ... atm_kptr_t vcc; /* both directions */ ... } __ATM_API_ALIGN; typedef struct { unsigned char _[8]; } __ATM_API_ALIGN atm_kptr_t; The special message is processed in atmtcp_recv_control() called from atmtcp_c_send(). atmtcp_c_send() is vcc->dev->ops->send() and called from 2 paths: 1. .ndo_start_xmit() (vcc->send() == atm_send_aal0()) 2. vcc_sendmsg() The problem is sendmsg() does not validate the message length and userspace can abuse atmtcp_recv_control() to overwrite any kptr by atmtcp_control. Let's add a new ->pre_send() hook to validate messages from sendmsg(). [0]: Oops: general protection fault, probably for non-canonical address 0xdffffc00200000ab: 0000 [#1] SMP KASAN PTI KASAN: probably user-memory-access in range [0x0000000100000558-0x000000010000055f] CPU: 0 UID: 0 PID: 5865 Comm: syz-executor331 Not tainted 6.17.0-rc1-syzkaller-00215-gbab3ce404553 #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 07/12/2025 RIP: 0010:atmtcp_recv_control drivers/atm/atmtcp.c:93 [inline] RIP: 0010:atmtcp_c_send+0x1da/0x950 drivers/atm/atmtcp.c:297 Code: 4d 8d 75 1a 4c 89 f0 48 c1 e8 03 42 0f b6 04 20 84 c0 0f 85 15 06 00 00 41 0f b7 1e 4d 8d b7 60 05 00 00 4c 89 f0 48 c1 e8 03 <42> 0f b6 04 20 84 c0 0f 85 13 06 00 00 66 41 89 1e 4d 8d 75 1c 4c RSP: 0018:ffff90003f5f810 EFLAGS: 00010203 RAX: 00000000200000ab RBX: 0000000000000000 RCX: 0000000000000000 RDX: ffff88802a510000 RSI: 00000000ffffff RDI: ffff888030a6068c RBP: ffff88802699fb40 R08: ffff888030a606eb R09: 1ffff1100614c0dd R10: dffffc0000000000 R11: ffffffff8718fc40 R12: dffffc0000000000 R13: ffff888030a60680 R14: 000000010000055f R15: 00000000ffffff FS: 00007f8d7e9236c0(0000) GS:ffff888125c1c000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 000000000045ad50 CR3: 0000000075bde000 CR4: 00000000003526f0 Call Trace: <TASK> vcc_sendmsg+0xa10/0xc60 net/atm/common.c:645 sock_sendmsg_nosec net/socket.c:714 [inline] __sock_sendmsg+0x219/0x270 net/socket.c:729 __sys_sendmsg+0x505/0x830 net/socket.c:2614 __sys_sendmsg+0x21f/0x2a0 net/socket.c:2668 __sys_sendmsg net/socket.c:2700 [inline] __do_sys_sendmsg net/socket.c:2705 [inline] __se_sys_sendmsg net/socket.c:2703 [inline] __x64_sys_sendmsg+0x19b/0x260 net/socket.c:2703 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f8d7e96a4a9 Code: 28 00 00 00 75 05 48 83 c4 28 c3 e8 51 18 00 00 90 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 b0 ff ff ff f7 d8 64 89 01 48 RSP: 002b:00007f8d7e923198 EFLAGS: 00000246 ORIG_RAX: 000000000000002e RAX: ffffffff8718fc40 RBX: 00007f8d7e9f4308 RCX: 00007f8d7e96a4a9 RDX: 0000000000000000 RSI: 0000200000000240 RDI: 0000000000000005 RBP: 00007f8d7e9f4300 R08: 65732f636f72702f R09: 65732f636f72702f R10: 65732f636f72702f R11: 0000000000000246 R12: 00007f8d7e9c10ac R13: 00007f8d7e9231a0 R14: 0000200000000200 R15: 0000200000000250 </TASK> Modules linked in:
CVE-2025-39829	In the Linux kernel, the following vulnerability has been resolved: trace/fgraph: Fix the warning caused by missing unregister notifier This warning was triggered during testing on v6.16: notifier callback ftrace_suspend_notifier_call already registered WARNING: CPU: 2 PID: 86 at kernel/notifier.c:23 notifier_chain_register+0x44/0xb0 ... Call Trace: <TASK> blocking_notifier_chain_register+0x34/0x60 register_ftrace_graph+0x330/0x410 ftrace_profile_write+0x1e9/0x340 vfs_write+0xf8/0x420 ? filp_flush+0x8a/0xa0 ? filp_close+0x1f/0x30 ? do_dup2+0xaf/0x160 ksys_write+0x65/0xe0 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x77/0x7f When writing to the function_profile_enabled interface, the notifier was not unregistered after start_graph_tracing failed, causing a warning the next time function_profile_enabled was written. Fixed by adding unregister_pm_notifier in the exception path.
CVE-2025-39832	In the Linux kernel, the following vulnerability has been resolved: net/mlx5: Fix lockdep assertion on sync reset unload event Fix lockdep assertion triggered during sync reset unload event. When the sync reset flow is initiated using the devlink reload fw_activate option, the PF already holds the devlink lock while handling unload event. In this case, delegate sync reset unload event handling back to the devlink callback process to avoid double-locking and resolve the lockdep warning. Kernel log: WARNING: CPU: 9 PID: 1578 at devl_assert_locked+0x31/0x40 [...] Call Trace: <TASK> mlx5_unload_one_devl_locked+0x2c/0xc0 [mlx5_core] mlx5_sync_reset_unload_event+0xaf/0x2f0 [mlx5_core] process_one_work+0x222/0x640 worker_thread+0x199/0x350 kthread+0x10b/0x230 ? __pfx_worker_thread+0x10/0x10 ? __pfx_kthread+0x10/0x10 ret_from_fork+0x8e/0x100 ? __pfx_kthread+0x10/0x10 ret_from_fork_asm+0x1a/0x30 </TASK>

CVE ID	Description
CVE-2025-39833	In the Linux kernel, the following vulnerability has been resolved: mISDN: hfcpcli: Fix warning when deleting uninitialized timer With CONFIG_DEBUG_OBJECTS_TIMERS unloading hfcpcli module leads to the following splat: [250.215892] ODEBUG: assert_init not available (active state 0) object: ffffffff01a3dc0 object type: timer_list hint: 0x0 [250.217520] WARNING: CPU: 0 PID: 233 at lib/debugobjects.c:612 debug_print_object+0x1b6/0x2c0 [250.218775] Modules linked in: hfcpcli(-) mISDN_core [250.219537] CPU: 0 UID: 0 PID: 233 Comm: rmmmod Not tainted 6.17.0-rc2-g6f713187ac98 #2 PREEMPT(voluntary) [250.220940] Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 [250.222377] RIP: 0010:debug_print_object+0x1b6/0x2c0 [250.223131] Code: fc ff df 48 89 fa 48 c1 ea 03 80 3c 02 00 75 4f 41 56 48 8b 14 dd a0 4e 01 9f 48 89 ee 48 c7 c7 20 46 01 9f e8 cb 84d [250.225805] RSP: 0018:ffff888015ea7c08 EFLAGS: 00010286 [250.226608] RAX: 0000000000000000 RBX: 0000000000000005 RCX: ffffffff9be93a95 [250.227708] RDX: 1ffff1100d945138 RSI: 0000000000000008 RDI: ffff88806ca289c0 [250.228993] RBP: ffffffff9f014a00 R08: 0000000000000001 R09: ffffed1002bd4f39 [250.230043] R10: ffff888015ea79cf R11: 0000000000000001 R12: 0000000000000001 [250.231185] R13: ffffffff9eea0520 R14: 0000000000000000 R15: ffff888015ea7cc8 [250.232454] FS: 00007f3208f01540(0000) GS:ffff8880caf5a000(0000) knlGS:0000000000000000 [250.233851] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [250.234856] CR2: 00007f32090a7421 CR3: 0000000004d63000 CR4: 0000000000000060 [250.236117] Call Trace: [250.236599] <TASK> [250.236967] ? trace_irq_enable.constprop.0+0xd4/0x130 [250.237920] debug_object_assert_init+0x1f6/0x310 [250.238762] ? __pfx_debug_object_assert_init+0x10/0x10 [250.239658] ? __lock_acquire+0xdea/0x1c70 [250.240369] __try_to_del_timer_sync+0x69/0x140 [250.241172] ? __pfx__try_to_del_timer_sync+0x10/0x10 [250.242058] ? __timer_delete_sync+0xc6/0x120 [250.242842] ? lock_acquire+0x30/0x80 [250.243474] ? __timer_delete_sync+0xc6/0x120 [250.244262] __timer_delete_sync+0x98/0x120 [250.245015] HFC_cleanup+0x10/0x20 [hfcpcli] [250.245704] __do_sys_delete_module+0x348/0x510 [250.246461] ? __pfx__do_sys_delete_module+0x10/0x10 [250.247338] do_syscall_64+0xc1/0x360 [250.247924] entry_SYSCALL_64_after_hwframe+0x77/0x7f Fix this by initializing hfc_tl timer with DEFINE_TIMER macro. Also, use mod_timer instead of manual timeout update.
CVE-2025-39835	In the Linux kernel, the following vulnerability has been resolved: xfs: do not propagate ENODATA disk errors into xattr code ENODATA (aka ENOATTR) has a very specific meaning in the xfs xattr code; namely, that the requested attribute name could not be found. However, a medium error from disk may also return ENODATA. At best, this medium error may escape to userspace as "attribute not found" when in fact it's an IO (disk) error. At worst, we may oops in xfs_attr_leaf_get() when we do: error = xfs_attr_leaf_hasname(args, &bp); if (error == -ENOATTR) { xfs_trans_brelse(args->trans, bp); return error; } because an ENODATA/ENOATTR error from disk leaves us with a null bp, and the xfs_trans_brelse will then null-deref it. As discussed on the list, we really need to modify the lower level IO functions to trap all disk errors and ensure that we don't let unique errors like this leak up into higher xfs functions - many like this should be remapped to EIO. However, this patch directly addresses a reported bug in the xattr code, and should be safe to backport to stable kernels. A larger-scope patch to handle more unique errors at lower levels can follow later. (Note, prior to 07120f1abdff we did not oops, but we did return the wrong error code to userspace.)
CVE-2025-39836	In the Linux kernel, the following vulnerability has been resolved: efi: stmm: Fix incorrect buffer allocation method The communication buffer allocated by setup_mm_hdr() is later on passed to tee_shm_register_kernel_buf(). The latter expects those buffers to be contiguous pages, but setup_mm_hdr() just uses kmalloc(). That can cause various corruptions or BUGs, specifically since commit 9aec2fb0fd5e ("slab: allocate frozen pages"), though it was broken before as well. Fix this by using alloc_pages_exact() instead of kmalloc().
CVE-2025-39838	In the Linux kernel, the following vulnerability has been resolved: cifs: prevent NULL pointer dereference in UTF16 conversion There can be a NULL pointer dereference bug here. NULL is passed to __cifs_sfu_make_node without checks, which passes it unchecked to cifs_strndup_to_utf16, which in turn passes it to cifs_local_to_utf16_bytes where 'from' is dereferenced, causing a crash. This patch adds a check for NULL 'src' in cifs_strndup_to_utf16 and returns NULL early to prevent dereferencing NULL pointer. Found by Linux Verification Center (linuxtesting.org) with SVACE
CVE-2025-39839	In the Linux kernel, the following vulnerability has been resolved: batman-adv: fix OOB read/write in network-coding decode batadv_nc_skb_decode_packet() trusts coded_len and checks only against skb->len. XOR starts at sizeof(struct batadv_unicast_packet), reducing payload headroom, and the source skb length is not verified, allowing an out-of-bounds read and a small out-of-bounds write. Validate that coded_len fits within the payload area of both destination and source sk_buffs before XORing.

CVE ID	Description
CVE-2025-39841	In the Linux kernel, the following vulnerability has been resolved: scsi: lpfc: Fix buffer free/clear order in deferred receive path Fix a use-after-free window by correcting the buffer release sequence in the deferred receive path. The code freed the RQ buffer first and only then cleared the context pointer under the lock. Concurrent paths (e.g., ABTS and the repost path) also inspect and release the same pointer under the lock, so the old order could lead to double-free/UAF. Note that the repost path already uses the correct pattern: detach the pointer under the lock, then free it after dropping the lock. The deferred path should do the same.
CVE-2025-39842	In the Linux kernel, the following vulnerability has been resolved: ocfs2: prevent release journal inode after journal shutdown Before calling ocfs2_delete_osb(), ocfs2_journal_shutdown() has already been executed in ocfs2_dismount_volume(), so osb->journal must be NULL. Therefore, the following calltrace will inevitably fail when it reaches jbd2_journal_release_jbd_inode(). ocfs2_dismount_volume()-> ocfs2_delete_osb()-> ocfs2_free_slot_info()-> __ocfs2_free_slot_info()-> evict()-> ocfs2_evict_inode()-> ocfs2_clear_inode()-> jbd2_journal_release_jbd_inode(osb->journal->j_journal, Adding osb->journal checks will prevent null-ptr-deref during the above execution path.
CVE-2025-39843	In the Linux kernel, the following vulnerability has been resolved: mm: slub: avoid wake up kswapd in set_track_prepare set_track_prepare() can incur lock recursion. The issue is that it is called from hrtimer_start_range_ns holding the per_cpu(hrtimer_bases)[n].lock, but when enabled CONFIG_DEBUG_OBJECTS_TIMERS, may wake up kswapd in set_track_prepare, and try to hold the per_cpu(hrtimer_bases)[n].lock. Avoid deadlock caused by implicitly waking up kswapd by passing in allocation flags, which do not contain __GFP_KSWAPD_RECLAIM in the debug_objects_fill_pool() case. Inside stack depot they are processed by gfp_nested_mask(). Since __slab_alloc() has preemption disabled, we mask out __GFP_DIRECT_RECLAIM from the flags there. The oops looks something like: BUG: spinlock recursion on CPU#3, swapper/3/0 lock: 0xfffff8a4bf29c80, .magic: dead4ead, .owner: swapper/3/0, .owner_cpu: 3 Hardware name: Qualcomm Technologies, Inc. Popsicle based on SM8850 (DT) Call trace: spin_bug+0x0 _raw_spin_lock_irqsave+0x80 hrtimer_try_to_cancel+0x94 task_contending+0x10c enqueue_dl_entity+0x2a4 dl_server_start+0x74 enqueue_task_fair+0x568 enqueue_task+0xac do_activate_task+0x14c ttwu_do_activate+0xcc try_to_wake_up+0x6c8 default_wake_function+0x20 autoremove_wake_function+0x1c __wake_up+0xac wakeup_kswapd+0x19c wake_all_kswapds+0x78 __alloc_pages_slowpath+0x1ac __alloc_pages_noprof+0x298 stack_depot_save_flags+0x6b0 stack_depot_save+0x14 set_track_prepare+0x5c __slab_alloc+0xccc __kmalloccache_noprof+0x470 __set_page_owner+0x2bc post_alloc_hook[jt]+0x1b8 prep_new_page+0x28 get_page_from_freelist+0x1edc __alloc_pages_noprof+0x13c alloc_slab_page+0x244 allocate_slab+0x7c __slab_alloc+0x8e8 kmem_cache_alloc_noprof+0x450 debug_objects_fill_pool+0x22c debug_object_activate+0x40 enqueue_hrtimer[jt]+0xdc hrtimer_start_range_ns+0x5f8 ...

CVE ID	Description
CVE-2025-39844	In the Linux kernel, the following vulnerability has been resolved: mm: move page table sync declarations to linux/pgtable.h During our internal testing, we started observing intermittent boot failures when the machine uses 4-level paging and has a large amount of persistent memory: BUG: unable to handle page fault for address: fffff70000000034 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 0 P4D 0 Oops: 0002 [#1] SMP NOPTI RIP: 0010: __init_single_page+0x9/0x6d Call Trace: <TASK> __init_zone_device_page+0x17/0x5d memmap_init_zone_device+0x154/0x1bb pagemap_range+0x2e0/0x40f memremap_pages+0x10b/0x2f0 devm_memremap_pages+0x1e/0x60 dev_dax_probe+0xce/0x2ec [device_dax] dax_bus_probe+0x6d/0xc9 [... snip ...] </TASK> It turns out that the kernel panics while initializing vmemmap (struct page array) when the vmemmap region spans two PGD entries, because the new PGD entry is only installed in init_mm.pgd, but not in the page tables of other tasks. And looking at __populate_section_memmap(): if (vmemmap_can_optimize(altmap, pgmap)) // does not sync top level page tables r = vmemmap_populate_compound_pages(pfn, start, end, nid, pgmap); else // sync top level page tables in x86 r = vmemmap_populate(start, end, nid, altmap); In the normal path, vmemmap_populate() in arch/x86/mm/init_64.c synchronizes the top level page table (See commit 9b861528a801 ("x86-64, mem: Update all PGDs for direct mapping and vmemmap mapping changes")) so that all tasks in the system can see the new vmemmap area. However, when vmemmap_can_optimize() returns true, the optimized path skips synchronization of top-level page tables. This is because vmemmap_populate_compound_pages() is implemented in core MM code, which does not handle synchronization of the top-level page tables. Instead, the core MM has historically relied on each architecture to perform this synchronization manually. We're not the first party to encounter a crash caused by not-sync'd top level page tables: earlier this year, Gwan-gyeong Mun attempted to address the issue [1] [2] after hitting a kernel panic when x86 code accessed the vmemmap area before the corresponding top-level entries were synced. At that time, the issue was believed to be triggered only when struct page was enlarged for debugging purposes, and the patch did not get further updates. It turns out that current approach of relying on each arch to handle the page table sync manually is fragile because 1) it's easy to forget to sync the top level page table, and 2) it's also easy to overlook that the kernel should not access the vmemmap and direct mapping areas before the sync. # The solution: Make page table sync more code robust and harder to miss To address this, Dave Hansen suggested [3] [4] introducing {pgd,p4d} _populate_kernel() for updating kernel portion of the page tables and allow each architecture to explicitly perform synchronization when installing top-level entries. With this approach, we no longer need to worry about missing the sync step, reducing the risk of future regressions. The new interface reuses existing ARCH_PAGE_TABLE_SYNC_MASK, PGTBL_P*D_MODIFIED and arch_sync_kernel_mappings() facility used by vmalloc and ioremap to synchronize page tables. pgd_populate_kernel() looks like this: static inline void pgd_populate_kernel(unsigned long addr, pgd_t *pgd, p4d_t *p4d) { pgd_populate(&init_mm, pgd, p4d); if (ARCH_PAGE_TABLE_SYNC_MASK & PGTBL_PGD_MODIFIED) arch_sync_kernel_mappings(addr, addr); } It is worth noting that vmalloc() and apply_to_range() carefully synchronizes page tables by calling p*d_alloc_track() and arch_sync_kernel_mappings(), and thus they are not affected by ---truncated---
CVE-2025-39845	In the Linux kernel, the following vulnerability has been resolved: x86/mm/64: define ARCH_PAGE_TABLE_SYNC_MASK and arch_sync_kernel_mappings() Define ARCH_PAGE_TABLE_SYNC_MASK and arch_sync_kernel_mappings() to ensure page tables are properly synchronized when calling p*d_populate_kernel(). For 5-level paging, synchronization is performed via pgd_populate_kernel(). In 4-level paging, pgd_populate() is a no-op, so synchronization is instead performed at the P4D level via p4d_populate_kernel(). This fixes intermittent boot failures on systems using 4-level paging and a large amount of persistent memory: BUG: unable to handle page fault for address: fffff70000000034 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 0 P4D 0 Oops: 0002 [#1] SMP NOPTI RIP: 0010: __init_single_page+0x9/0x6d Call Trace: <TASK> __init_zone_device_page+0x17/0x5d memmap_init_zone_device+0x154/0x1bb pagemap_range+0x2e0/0x40f memremap_pages+0x10b/0x2f0 devm_memremap_pages+0x1e/0x60 dev_dax_probe+0xce/0x2ec [device_dax] dax_bus_probe+0x6d/0xc9 [... snip ...] </TASK> It also fixes a crash in vmemmap_set_pmd() caused by accessing vmemmap before sync_global_pgds() [1]: BUG: unable to handle page fault for address: fffffb3ff1200000 #PF: supervisor write access in kernel mode #PF: error_code(0x0002) - not-present page PGD 0 P4D 0 Oops: Oops: 0002 [#1] PREEMPT SMP NOPTI Tainted: [W]=WARN RIP: 0010:vmemmap_set_pmd+0xff/0x230 <TASK> vmemmap_populate_hugepages+0x176/0x180 vmemmap_populate+0x34/0x80 __populate_section_memmap+0x41/0x90 sparse_add_section+0x121/0x3e0 __add_pages+0xba/0x150 add_pages+0x1d/0x70 memremap_pages+0x3dc/0x810 devm_memremap_pages+0x1c/0x60 xe_devmem_add+0x8b/0x100 [xe] xe_tile_init_noalloc+0x6a/0x70 [xe] xe_device_probe+0x48c/0x740 [xe] [... snip ...]
CVE-2025-39846	In the Linux kernel, the following vulnerability has been resolved: pcmcia: Fix a NULL pointer dereference in __iodyn_find_io_region() In __iodyn_find_io_region(), pcmcia_make_resource() is assigned to res and used in pci_bus_alloc_resource(). There is a dereference of res in pci_bus_alloc_resource(), which could lead to a NULL pointer dereference on failure of pcmcia_make_resource(). Fix this bug by adding a check of res.

CVE ID	Description
CVE-2025-39847	In the Linux kernel, the following vulnerability has been resolved: ppp: fix memory leak in pad_compress_skb If alloc_skb() fails in pad_compress_skb(), it returns NULL without releasing the old skb. The caller does: skb = pad_compress_skb(ppp, skb); if (!skb) goto drop; drop: kfree_skb(skb); When pad_compress_skb() returns NULL, the reference to the old skb is lost and kfree_skb(skb) ends up doing nothing, leading to a memory leak. Align pad_compress_skb() semantics with realloc(): only free the old skb if allocation and compression succeed. At the call site, use the new_skb variable so the original skb is not lost when pad_compress_skb() fails.
CVE-2025-39848	In the Linux kernel, the following vulnerability has been resolved: ax25: properly unshare skbs in ax25_kiss_rcv() Bernard Pidoux reported a regression apparently caused by commit c353e8983e0d ("net: introduce per netns packet chains"). skb->dev becomes NULL and we crash in __netif_receive_skb_core(). Before above commit, different kind of bugs or corruptions could happen without a major crash. But the root cause is that ax25_kiss_rcv() can queue/mangle input skb without checking if this skb is shared or not. Many thanks to Bernard Pidoux for his help, diagnosis and tests. We had a similar issue years ago fixed with commit 7aaed57c5c28 ("phonet: properly unshare skbs in phonet_rcv()").
CVE-2025-39849	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: sme: cap SSID length in __cfg80211_connect_result() If the ssid->datalen is more than IEEE80211_MAX_SSID_LEN (32) it would lead to memory corruption so add some bounds checking.
CVE-2025-39850	In the Linux kernel, the following vulnerability has been resolved: vxlan: Fix NPD in {arp,neigh}_reduce() when using nexthop objects When the "proxy" option is enabled on a VXLAN device, the device will suppress ARP requests and IPv6 Neighbor Solicitation messages if it is able to reply on behalf of the remote host. That is, if a matching and valid neighbor entry is configured on the VXLAN device whose MAC address is not behind the "any" remote (0.0.0.0 / ::). The code currently assumes that the FDB entry for the neighbor's MAC address points to a valid remote destination, but this is incorrect if the entry is associated with an FDB nexthop group. This can result in a NPD [1] [3] which can be reproduced using [2][4]. Fix by checking that the remote destination exists before dereferencing it. [1] BUG: kernel NULL pointer dereference, address: 0000000000000000 [...] CPU: 4 UID: 0 PID: 365 Comm: arping Not tainted 6.17.0-rc2-virtme-g2a89cb21162c #2 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-4.fc41 04/01/2014 RIP: 0010:vxlan_xmit+0xb58/0x15f0 [...] Call Trace: <TASK> dev_hard_start_xmit+0x5d/0x1c0 __dev_queue_xmit+0x246/0xf0 packet_sendmsg+0x113a/0x1850 __sock_sendmsg+0x38/0x70 __sys_sendto+0x126/0x180 __x64_sys_sendto+0x24/0x30 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x4b/0x53 [2] #!/bin/bash ip address add 192.0.2.1/32 dev lo ip nexthop add id 1 via 192.0.2.2 fdb ip nexthop add id 10 group 1 fdb ip link add name vx0 up type vxlan id 10010 local 192.0.2.1 dstport 4789 proxy ip neigh add 192.0.2.3 lladdr 00:11:22:33:44:55 nud perm dev vx0 bridge fdb add 00:11:22:33:44:55 dev vx0 self static nhid 10 arping -b -c 1 -s 192.0.2.1 -I vx0 192.0.2.3 [3] BUG: kernel NULL pointer dereference, address: 0000000000000000 [...] CPU: 13 UID: 0 PID: 372 Comm: ndisc6 Not tainted 6.17.0-rc2-virtme-g6ee90cb26014 #3 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1v996), BIOS 1.17.0-4.fc41 04/01/2x014 RIP: 0010:vxlan_xmit+0x803/0x1600 [...] Call Trace: <TASK> dev_hard_start_xmit+0x5d/0x1c0 __dev_queue_xmit+0x246/0xf0 ip6_finish_output2+0x210/0x6c0 ip6_finish_output+0x1af/0x2b0 ip6_mr_output+0x92/0x3e0 ip6_send_skb+0x30/0x90 rawv6_sendmsg+0xe6e/0x12e0 __sock_sendmsg+0x38/0x70 __sys_sendto+0x126/0x180 __x64_sys_sendto+0x24/0x30 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x4b/0x53 RIP: 0033:0x7f383422ec77 [4] #!/bin/bash ip address add 2001:db8:1::1/128 dev lo ip nexthop add id 1 via 2001:db8:1::1 fdb ip nexthop add id 10 group 1 fdb ip link add name vx0 up type vxlan id 10010 local 2001:db8:1::1 dstport 4789 proxy ip neigh add 2001:db8:1::3 lladdr 00:11:22:33:44:55 nud perm dev vx0 bridge fdb add 00:11:22:33:44:55 dev vx0 self static nhid 10 ndisc6 -r 1 -s 2001:db8:1::1 -w 1 2001:db8:1::3 vx0

CVE ID	Description
CVE-2025-39851	In the Linux kernel, the following vulnerability has been resolved: vxlan: Fix NPD when refreshing an FDB entry with a nexthop object VXLAN FDB entries can point to either a remote destination or an FDB nexthop group. The latter is usually used in EVPN deployments where learning is disabled. However, when learning is enabled, an incoming packet might try to refresh an FDB entry that points to an FDB nexthop group and therefore does not have a remote. Such packets should be dropped, but they are only dropped after dereferencing the non-existent remote, resulting in a NPD [1] which can be reproduced using [2]. Fix by dropping such packets earlier. Remove the misleading comment from first_remote_rcu(). [1] BUG: kernel NULL pointer dereference, address: 0000000000000000 [...] CPU: 13 UID: 0 PID: 361 Comm: mausezahn Not tainted 6.17.0-rc1-virtme-g9f6b606b6b37 #1 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-4.fc41 04/01/2014 RIP: 0010:vxlan_snoop+0x98/0x1e0 [...] Call Trace: <TASK> vxlan_encap_bypass+0x209/0x240 encap_bypass_if_local+0xb1/0x100 vxlan_xmit_one+0x1375/0x17e0 vxlan_xmit+0x6b4/0x15f0 dev_hard_start_xmit+0x5d/0x1c0 __dev_queue_xmit+0x246/0xf0 packet_sendmsg+0x113a/0x1850 __sock_sendmsg+0x38/0x70 __sys_sendto+0x126/0x180 __x64_sys_sendto+0x24/0x30 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x4b/0x53 [2] #!/bin/bash ip address add 192.0.2.1/32 dev lo ip address add 192.0.2.2/32 dev lo ip nexthop add id 1 via 192.0.2.3 fdb ip nexthop add id 10 group 1 fdb ip link add name vx0 up type vxlan id 10010 local 192.0.2.1 dstport 12345 localbypass ip link add name vx1 up type vxlan id 10020 local 192.0.2.2 dstport 54321 learning bridge fdb add 00:11:22:33:44:55 dev vx0 self static dst 192.0.2.2 port 54321 vni 10020 bridge fdb add 00:aa:bb:cc:dd:ee dev vx1 self static nhid 10 mausezahn vx0 -a 00:aa:bb:cc:dd:ee -b 00:11:22:33:44:55 -c 1 -q
CVE-2025-39852	In the Linux kernel, the following vulnerability has been resolved: net/tcp: Fix socket memory leak in TCP-AO failure handling for IPv6 When tcp_ao_copy_all_matching() fails in tcp_v6_syn_recv_sock() it just exits the function. This ends up causing a memory-leak: unreferenced object 0xffff0000281a8200 (size 2496): comm "softirq", pid 0, jiffies 4295174684 hex dump (first 32 bytes): 7f 00 00 06 7f 00 00 06 00 00 00 00 cb a8 88 13 0a 00 03 61 00 00 00 00 00 00 00 00 00 00 00 00 00 ...a..... backtrace (crc 5ebdb15): kmemleak_alloc+0x44/0xe0 kmem_cache_alloc_noprof+0x248/0x470 sk_prot_alloc+0x48/0x120 sk_clone_lock+0x38/0x3b0 inet_csk_clone_lock+0x34/0x150 tcp_create_openreq_child+0x3c/0x4a8 tcp_v6_syn_recv_sock+0x1c0/0x620 tcp_check_req+0x588/0x790 tcp_v6_rcv+0x5d0/0xc18 ip6_protocol_deliver_rcu+0x2d8/0x4c0 ip6_input_finish+0x74/0x148 ip6_input+0x50/0x118 ip6_sublist_rcv+0x2fc/0x3b0 ipv6_list_rcv+0x114/0x170 __netif_receive_skb_list_core+0x16c/0x200 netif_receive_skb_list_internal+0x1f0/0x2d0 This is because in tcp_v6_syn_recv_sock (and the IPv4 counterpart), when exiting upon error, inet_csk_prepare_forced_close() and tcp_done() need to be called. They make sure the newsk will end up being correctly free'd. tcp_v4_syn_recv_sock() makes this very clear by having the put_and_exit label that takes care of things. So, this patch here makes sure tcp_v4_syn_recv_sock and tcp_v6_syn_recv_sock have similar error-handling and thus fixes the leak for TCP-AO.
CVE-2025-39853	In the Linux kernel, the following vulnerability has been resolved: i40e: Fix potential invalid access when MAC list is empty list_first_entry() never returns NULL - if the list is empty, it still returns a pointer to an invalid object, leading to potential invalid memory access when dereferenced. Fix this by using list_first_entry_or_null instead of list_first_entry.
CVE-2025-39854	In the Linux kernel, the following vulnerability has been resolved: ice: fix NULL access of tx->in_use in ice_ll_ts_intr Recent versions of the E810 firmware have support for an extra interrupt to handle report of the "low latency" Tx timestamps coming from the specialized low latency firmware interface. Instead of polling the registers, software can wait until the low latency interrupt is fired. This logic makes use of the Tx timestamp tracking structure, ice_ptp_tx, as it uses the same "ready" bitmap to track which Tx timestamps complete. Unfortunately, the ice_ll_ts_intr() function does not check if the tracker is initialized before its first access. This results in NULL dereference or use-after-free bugs similar to the issues fixed in the ice_ptp_ts_irq() function. Fix this by only checking the in_use bitmap (and other fields) if the tracker is marked as initialized. The reset flow will clear the init field under lock before it tears the tracker down, thus preventing any use-after-free or NULL access.
CVE-2025-39857	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix one NULL pointer dereference in smc_ib_is_sg_need_sync() BUG: kernel NULL pointer dereference, address: 00000000000002ec PGD 0 P4D 0 Oops: Oops: 0000 [#1] SMP PTI CPU: 28 UID: 0 PID: 343 Comm: kworker/28:1 Kdump: loaded Tainted: G OE 6.17.0-rc2+ #9 NONE Tainted: [O]=OOT_MODULE, [E]=UNSIGNED_MODULE Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1 04/01/2014 Workqueue: smc_hs_wq smc_listen_work [smc] RIP: 0010:smc_ib_is_sg_need_sync+0x9e/0xd0 [smc] ... Call Trace: <TASK> smcr_buf_map_link+0x211/0x2a0 [smc] __smc_buf_create+0x522/0x970 [smc] smc_buf_create+0x3a/0x110 [smc] smc_find_rdma_v2_device_serv+0x18f/0x240 [smc] ? smc_vlan_by_tcpsk+0x7e/0xe0 [smc] smc_listen_find_device+0x1dd/0x2b0 [smc] smc_listen_work+0x30f/0x580 [smc] process_one_work+0x18c/0x340 worker_thread+0x242/0x360 kthread+0xe7/0x220 ret_from_fork+0x13a/0x160 ret_from_fork_asm+0x1a/0x30 </TASK> If the software RoCE device is used, ibdev->dma_device is a null pointer. As a result, the problem occurs. Null pointer detection is added to prevent problems.

CVE ID	Description
CVE-2025-39859	In the Linux kernel, the following vulnerability has been resolved: ptp: ocp: fix use-after-free bugs causing by ptp_ocp_watchdog The ptp_ocp_detach() only shuts down the watchdog timer if it is pending. However, if the timer handler is already running, the timer_delete_sync() is not called. This leads to race conditions where the devlink that contains the ptp_ocp is deallocated while the timer handler is still accessing it, resulting in use-after-free bugs. The following details one of the race scenarios. (thread 1) (thread 2) ptp_ocp_remove() ptp_ocp_detach() ptp_ocp_watchdog() if (timer_pending(&bp->watchdog)) bp = timer_container_of() timer_delete_sync() devlink_free(devlink) //free bp-> //use Resolve this by unconditionally calling timer_delete_sync() to ensure the timer is reliably deactivated, preventing any access after free.
CVE-2025-39860	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: Fix use-after-free in l2cap_sock_cleanup_listen() syzbot reported the splat below without a repro. In the splat, a single thread calling bt_accept_dequeue() freed sk and touched it after that. The root cause would be the racy l2cap_sock_cleanup_listen() call added by the cited commit. bt_accept_dequeue() is called under lock_sock() except for l2cap_sock_release(). Two threads could see the same socket during the list iteration in bt_accept_dequeue(): CPU1 CPU2 (close()) ---- sock_hold(sk) sock_hold(sk); lock_sock(sk) <-- block close() sock_put(sk) bt_accept_unlink(sk) sock_put(sk) <-- refcnt by bt_accept_enqueue() release_sock(sk) lock_sock(sk) sock_put(sk) bt_accept_unlink(sk) sock_put(sk) <-- last refcnt bt_accept_unlink(sk) <-- UAF Depending on the timing, the other thread could show up in the "Freed by task" part. Let's call l2cap_sock_cleanup_listen() under lock_sock() in l2cap_sock_release(). [0]: BUG: KASAN: slab-use-after-free in debug_spin_lock_before kernel/locking/spinlock_debug.c:86 [inline] BUG: KASAN: slab-use-after-free in do_raw_spin_lock+0x26f/0x2b0 kernel/locking/spinlock_debug.c:115 Read of size 4 at addr ffff88803b7eb1c4 by task syz.5.3276/16995 CPU: 3 UID: 0 PID: 16995 Comm: syz.5.3276 Not tainted syzkaller #0 PREEMPT(full) Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.16.3-debian-1.16.3-2~bpo12+1 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:94 [inline] dump_stack_lvl+0x116/0x1f0 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xcd/0x630 mm/kasan/report.c:482 kasan_report+0xe0/0x110 mm/kasan/report.c:595 debug_spin_lock_before kernel/locking/spinlock_debug.c:86 [inline] do_raw_spin_lock+0x26f/0x2b0 kernel/locking/spinlock_debug.c:115 spin_lock_bh include/linux/spinlock.h:356 [inline] release_sock+0x21/0x220 net/core/socket.c:3746 bt_accept_dequeue+0x505/0x600 net/bluetooth/af_bluetooth.c:312 l2cap_sock_cleanup_listen+0x5c/0x2a0 net/bluetooth/l2cap_sock.c:1451 l2cap_sock_release+0x5c/0x210 net/bluetooth/l2cap_sock.c:1425 __sock_release+0xb3/0x270 net/socket.c:649 sock_close+0x1c/0x30 net/socket.c:1439 __fput+0x3ff/0xb70 fs/file_table.c:468 task_work_run+0x14d/0x240 kernel/task_work.c:227 resume_user_mode_work include/linux/resume_user_mode.h:50 [inline] exit_to_user_mode_loop+0xeb/0x110 kernel/entry/common.c:43 exit_to_user_mode_prepare include/linux/irq-entry-common.h:225 [inline] syscall_exit_to_user_mode_work include/linux/entry-common.h:175 [inline] syscall_exit_to_user_mode include/linux/entry-common.h:210 [inline] do_syscall_64+0x3f6/0x4c0 arch/x86/entry/syscall_64.c:100 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f2accf8ebe9 Code: ff ff c3 66 2e 0f 1f 84 00 00 00 00 0f 1f 40 00 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 c7 c1 a8 ff ff f7 d8 64 89 01 48 RSP: 002b:00007ffdb6cb1378 EFLAGS: 00000246 ORIG_RAX: 00000000000001b4 RAX: 0000000000000000 RBX: 00000000000426fb RCX: 00007f2accf8ebe9 RDX: 0000000000000000 RSI: 000000000000001e RDI: 0000000000000003 RBP: 00007f2acd1b7da0 R08: 0000000000000001 R09: 00000012b6cb166f R10: 0000001b30e20000 R11: 0000000000000246 R12: 00007f2acd1b609c R13: 00007f2acd1b6090 R14: ffffffff ffffffff R15: 00007ffdb6cb1490 </TASK> Allocated by task 5326: kasan_save_stack+0x33/0x60 mm/kasan/common.c:47 kasan_save_track+0x14/0x30 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:388 [inline] __kasan_kmalloc+0xaa/0xb0 mm/kasan/common.c:405 kasan_kmalloc include/linux/kasan.h:260 [inline] __do_kmalloc_node mm/slub.c:4365 [inline] __kmalloc_nopro ---truncated---
CVE-2025-39861	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: vhci: Prevent use-after-free by removing debugfs files early Move the creation of debugfs files into a dedicated function, and ensure they are explicitly removed during vhci_release(), before associated data structures are freed. Previously, debugfs files such as "force_suspend", "force_wakeup", and others were created under hdev->debugfs but not removed in vhci_release(). Since vhci_release() frees the backing vhci_data structure, any access to these files after release would result in use-after-free errors. Although hdev->debugfs is later freed in hci_release_dev(), user can access files after vhci_data is freed but before hdev->debugfs is released.
CVE-2025-39862	In the Linux kernel, the following vulnerability has been resolved: wifi: mt76: mt7915: fix list corruption after hardware restart Since stations are recreated from scratch, all lists that wcid are added to must be cleared before calling ieee80211_restart_hw. Set wcid->sta = 0 for each wcid entry in order to ensure that they are not added again before they are ready.

CVE ID	Description
CVE-2025-39863	In the Linux kernel, the following vulnerability has been resolved: wifi: brcmfmac: fix use-after-free when rescheduling brcmf_btcoex_info work The brcmf_btcoex_detach() only shuts down the btcoex timer, if the flag timer_on is false. However, the brcmf_btcoex_timerfunc(), which runs as timer handler, sets timer_on to false. This creates critical race conditions: 1.If brcmf_btcoex_detach() is called while brcmf_btcoex_timerfunc() is executing, it may observe timer_on as false and skip the call to timer_shutdown_sync(). 2.The brcmf_btcoex_timerfunc() may then reschedule the brcmf_btcoex_info worker after the cancel_work_sync() has been executed, resulting in use-after-free bugs. The use-after-free bugs occur in two distinct scenarios, depending on the timing of when the brcmf_btcoex_info struct is freed relative to the execution of its worker thread. Scenario 1: Freed before the worker is scheduled The brcmf_btcoex_info is deallocated before the worker is scheduled. A race condition can occur when schedule_work(&bt_local->work) is called after the target memory has been freed. The sequence of events is detailed below: CPU0 CPU1 brcmf_btcoex_detach brcmf_btcoex_timerfunc bt_local->timer_on = false; if (cfg->btcoex->timer_on) ... cancel_work_sync(); ... kfree(cfg->btcoex); // FREE schedule_work(&bt_local->work); // USE Scenario 2: Freed after the worker is scheduled The brcmf_btcoex_info is freed after the worker has been scheduled but before or during its execution. In this case, statements within the brcmf_btcoex_handler() — such as the container_of macro and subsequent dereferences of the brcmf_btcoex_info object will cause a use-after-free access. The following timeline illustrates this scenario: CPU0 CPU1 brcmf_btcoex_detach brcmf_btcoex_timerfunc bt_local->timer_on = false; if (cfg->btcoex->timer_on) ... cancel_work_sync(); ... schedule_work(); // Reschedule kfree(cfg->btcoex); // FREE brcmf_btcoex_handler() // Worker /* btci = container_of(...); // USE The kfree() above could ... also occur at any point btci-> // USE during the worker's execution! */ To resolve the race conditions, drop the conditional check and call timer_shutdown_sync() directly. It can deactivate the timer reliably, regardless of its current state. Once stopped, the timer_on state is then set to false.
CVE-2025-39864	In the Linux kernel, the following vulnerability has been resolved: wifi: cfg80211: fix use-after-free in cmp_bss() Following bss_free() quirk introduced in commit 776b3580178f ("cfg80211: track hidden SSID networks properly"), adjust cfg80211_update_known_bss() to free the last beacon frame elements only if they're not shared via the corresponding 'hidden_beacon_bss' pointer.
CVE-2025-39865	In the Linux kernel, the following vulnerability has been resolved: tee: fix NULL pointer dereference in tee_shm_put tee_shm_put have NULL pointer dereference: __optee_disable_shm_cache -> shm = reg_pair_to_ptr(...); //shm maybe return NULL tee_shm_free(shm); -> tee_shm_put(shm); //crash Add check in tee_shm_put to fix it. panic log: Unable to handle kernel paging request at virtual address 0000000000100cca Mem abort info: ESR = 0x0000000096000004 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x04: level 0 translation fault Data abort info: ISV = 0, ISS = 0x00000004, ISS2 = 0x00000000 CM = 0, WnR = 0, TnD = 0, TagAccess = 0 GCS = 0, Overlay = 0, DirtyBit = 0, Xs = 0 user pgtbl: 4k pages, 48-bit VAs, pgdp=0000002049d07000 [0000000000100cca] pgd=0000000000000000, p4d=0000000000000000 Internal error: Oops: 0000000096000004 [#1] SMP CPU: 2 PID: 14442 Comm: systemd-sleep Tainted: P OE ----- 6.6.0-39-generic #38 Source Version: 938b255f6cb8817c95b0dd5c8c2944acfce94b07 Hardware name: greatwall GW-001Y1A-FTH, BIOS Great Wall BIOS V3.0 10/26/2022 pstate: 80000005 (Nzcv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--) pc : tee_shm_put+0x24/0x188 lr : tee_shm_free+0x14/0x28 sp : ffff001f98f9faf0 x29: ffff001f98f9faf0 x28: ffff0020df543cc0 x27: 0000000000000000 x26: ffff001f811344a0 x25: ffff8000818dac00 x24: ffff800082d8d048 x23: ffff001f850fcd18 x22: 0000000000000001 x21: ffff001f98f9fb88 x20: ffff001f83e76218 x19: ffff001f83e761e0 x18: 000000000000ffff x17: 303a30303a303030 x16: 0000000000000000 x15: 0000000000000003 x14: 0000000000000001 x13: 0000000000000000 x12: 0101010101010101 x11: 0000000000000001 x10: 0000000000000001 x9: ffff800080e08d0c x8: ffff001f98f9fb88 x7: 0000000000000000 x6: 0000000000000000 x5: 0000000000000000 x4: 0000000000000000 x3: 0000000000000000 x2: ffff001f83e761e0 x1: 00000000ffff001f x0: 0000000000100cca Call trace: tee_shm_put+0x24/0x188 tee_shm_free+0x14/0x28 __optee_disable_shm_cache+0xa8/0x108 optee_shutdown+0x28/0x38 platform_shutdown+0x28/0x40 device_shutdown+0x144/0x2b0 kernel_power_off+0x3c/0x80 hibernate+0x35c/0x388 state_store+0x64/0x80 kobj_attr_store+0x14/0x28 sysfs_kf_write+0x48/0x60 kernfs_fop_write_iter+0x128/0x1c0 vfs_write+0x270/0x370 ksys_write+0x6c/0x100 __arm64_sys_write+0x20/0x30 invoke_syscall+0x4c/0x120 el0_svc_common.constprop.0+0x44/0xf0 do_el0_svc+0x24/0x38 el0_svc+0x24/0x88 el0t_64_sync_handler+0x134/0x150 el0t_64_sync+0x14c/0x15

CVE ID	Description
CVE-2025-39866	In the Linux kernel, the following vulnerability has been resolved: fs: writeback: fix use-after-free in __mark_inode_dirty() An use-after-free issue occurred when __mark_inode_dirty() get the bdi_writeback that was in the progress of switching. CPU: 1 PID: 562 Comm: systemd-random-Not tainted 6.6.56-gb4403bd46a8e #1 pstate: 60400005 (nZCv daif +PAN -UAO -TCO -DIT -SSBS BTYP=--) pc : __mark_inode_dirty+0x124/0x418 lr : __mark_inode_dirty+0x118/0x418 sp : ffffffff08c9dbbc0 Call trace: __mark_inode_dirty+0x124/0x418 generic_update_time+0x4c/0x60 file_modified+0xcc/0xd0 ext4_buffered_write_iter+0x58/0x124 ext4_file_write_iter+0x54/0x704 vfs_write+0x1c0/0x308 ksys_write+0x74/0x10c __arm64_sys_write+0x1c/0x28 invoke_syscall+0x48/0x114 el0_svc_common.constprop.0+0xc0/0xe0 do_el0_svc+0x1c/0x28 el0_svc+0x40/0xe4 el0t_64_sync_handler+0x120/0x12c el0t_64_sync+0x194/0x198 Root cause is: systemd-random-seed kworker ----- __mark_inode_dirty inode_switch_wbs_work_fn spin_lock(&inode->i_lock); inode_attach_wb locked_inode_to_wb_and_lock_list get_inode->i_wb spin_unlock(&inode->i_lock); spin_lock(&wb->list_lock) spin_lock(&inode->i_lock) inode_io_list_move_locked spin_unlock(&wb->list_lock) spin_unlock(&inode->i_lock) spin_lock(&old_wb->list_lock) inode_do_switch_wbs spin_lock(&inode->i_lock) inode->i_wb = new_wb spin_unlock(&inode->i_lock) spin_unlock(&old_wb->list_lock) wb_put_many(old_wb, nr_switched) cgwb_release old wb released wb_wakeup_delayed() accesses wb, then trigger the use-after-free issue Fix this race condition by holding inode spinlock until wb_wakeup_delayed() finished.
CVE-2025-39869	In the Linux kernel, the following vulnerability has been resolved: dmaengine: ti: edma: Fix memory allocation size for queue_priority_map Fix a critical memory allocation bug in edma_setup_from_hw() where queue_priority_map was allocated with insufficient memory. The code declared queue_priority_map as s8 (*)[2] (pointer to array of 2 s8), but allocated memory using sizeof(s8) instead of the correct size. This caused out-of-bounds memory writes when accessing: queue_priority_map[i][0] = i; queue_priority_map[i][1] = i; The bug manifested as kernel crashes with "Oops - undefined instruction" on ARM platforms (BeagleBoard-X15) during EDMA driver probe, as the memory corruption triggered kernel hardening features on Clang. Change the allocation to use sizeof(*queue_priority_map) which automatically gets the correct size for the 2D array structure.
CVE-2025-39873	In the Linux kernel, the following vulnerability has been resolved: can: xilinx_can: xcan_write_frame(): fix use-after-free of transmitted SKB can_put_echo_skb() takes ownership of the SKB and it may be freed during or after the call. However, xilinx_can xcan_write_frame() keeps using SKB after the call. Fix that by only calling can_put_echo_skb() after the code is done touching the SKB. The tx_lock is held for the entire xcan_write_frame() execution and also on the can_get_echo_skb() side so the order of operations does not matter. An earlier fix commit 3d3c817c3a40 ("can: xilinx_can: Fix usage of skb memory") did not move the can_put_echo_skb() call far enough. [mkl: add "commit" in front of sha1 in patch description] [mkl: fix indentation]
CVE-2025-39876	In the Linux kernel, the following vulnerability has been resolved: net: fec: Fix possible NPD in fec_enet_phy_reset_after_clk_enable() The function of_phy_find_device may return NULL, so we need to take care before dereferencing phy_dev.
CVE-2025-39877	In the Linux kernel, the following vulnerability has been resolved: mm/damon/sysfs: fix use-after-free in state_show() state_show() reads kdamond->damon_ctx without holding damon_sysfs_lock. This allows a use-after-free race: CPU 0 CPU 1 ----- state_show() damon_sysfs_turn_damon_on() ctx = kdamond->damon_ctx; mutex_lock(&damon_sysfs_lock); damon_destroy_ctx(kdamond->damon_ctx); kdamond->damon_ctx = NULL; mutex_unlock(&damon_sysfs_lock); damon_is_running(ctx); / * ctx is freed */ mutex_lock(&ctx->kdamond_lock); /* UAF */ (The race can also occur with damon_sysfs_kdamonds_rm_dirs() and damon_sysfs_kdamond_release(), which free or replace the context under damon_sysfs_lock.) Fix by taking damon_sysfs_lock before dereferencing the context, mirroring the locking used in pid_show(). The bug has existed since state_show() first accessed kdamond->damon_ctx.
CVE-2025-39880	In the Linux kernel, the following vulnerability has been resolved: libceph: fix invalid accesses to ceph_connection_v1_info There is a place where generic code in messenger.c is reading and another place where it is writing to con->v1 union member without checking that the union member is active (i.e. msgr1 is in use). On 64-bit systems, con->v1.auth_retry overlaps with con->v2.out_iter, so such a read is almost guaranteed to return a bogus value instead of 0 when msgr2 is in use. This ends up being fairly benign because the side effect is just the invalidation of the authorizer and successive fetching of new tickets. con->v1.connect_seq overlaps with con->v2.conn_bufs and the fact that it's being written to can cause more serious consequences, but luckily it's not something that happens often.

CVE ID	Description
CVE-2025-39881	In the Linux kernel, the following vulnerability has been resolved: kernfs: Fix UAF in polling when open file is released A use-after-free (UAF) vulnerability was identified in the PSI (Pressure Stall Information) monitoring mechanism: BUG: KASAN: slab-use-after-free in psi_trigger_poll +0x3c/0x140 Read of size 8 at addr ffff3de3d50bd308 by task systemd/1 psi_trigger_poll+0x3c/0x140 cgroup_pressure_poll+0x70/0xa0 cgroup_file_poll+0x8c/0x100 kernfs_fop_poll+0x11c/0x1c0 ep_item_poll.isra.0+0x188/0x2c0 Allocated by task 1: cgroup_file_open+0x88/0x388 kernfs_fop_open+0x73c/0xaf0 do_dentry_open+0x5fc/0x1200 vfs_open+0xa0/0x3f0 do_open+0x7e8/0xd08 path_openat+0x2fc/0x6b0 do_filp_open+0x174/0x368 Freed by task 8462: cgroup_file_release+0x130/0x1f8 kernfs_drain_open_files+0x17c/0x440 kernfs_drain+0x2dc/0x360 kernfs_show+0x1b8/0x288 cgroup_file_show+0x150/0x268 cgroup_pressure_write+0x1dc/0x340 cgroup_file_write+0x274/0x548 Reproduction Steps: 1. Open test/cpu.pressure and establish epoll monitoring 2. Disable monitoring: echo 0 > test/cgroup.pressure 3. Re-enable monitoring: echo 1 > test/cgroup.pressure The race condition occurs because: 1. When cgroup.pressure is disabled (echo 0 > cgroup.pressure), it: - Releases PSI triggers via cgroup_file_release() - Frees of->priv through kernfs_drain_open_files() 2. While epoll still holds reference to the file and continues polling 3. Re-enabling (echo 1 > cgroup.pressure) accesses freed of->priv epolling disable/enable cgroup.pressure fd=open(cpu.pressure) while(1) ... epoll_wait kernfs_fop_poll kernfs_get_active = true echo 0 > cgroup.pressure ... cgroup_file_show kernfs_show // inactive kn kernfs_drain_open_files cft->release(of); kfree(ctx); ... kernfs_get_active = false echo 1 > cgroup.pressure kernfs_show kernfs_activate_one(kn); kernfs_fop_poll kernfs_get_active = true cgroup_file_poll psi_trigger_poll // UAF ... end: close(fd) To address this issue, introduce kernfs_get_active_of() for kernfs open files to obtain active references. This function will fail if the open file has been released. Replace kernfs_get_active() with kernfs_get_active_of() to prevent further operations on released file descriptors.
CVE-2025-39883	In the Linux kernel, the following vulnerability has been resolved: mm/memory-failure: fix VM_BUG_ON_PAGE(PagePoisoned(page)) when unpoison memory When I did memory failure tests, below panic occurs: page dumped because: VM_BUG_ON_PAGE(PagePoisoned(page)) kernel BUG at include/linux/page-flags.h:616! Oops: invalid opcode: 0000 [#1] PREEMPT SMP NOPTI CPU: 3 PID: 720 Comm: bash Not tainted 6.10.0-rc1-00195-g148743902568 #40 RIP: 0010:unpoison_memory+0x2f3/0x590 RSP: 0018:ffffa57fc8787d60 EFLAGS: 00000246 RAX: 0000000000000037 RBX: 0000000000000009 RCX: ffff9be25fcdc9c8 RDX: 0000000000000000 RSI: 0000000000000027 RDI: ffff9be25fcdc9c0 RBP: 0000000000300000 R08: ffffffffb4956f88 R09: 00000000000009ffb R10: 0000000000000284 R11: ffffffffb4926fa0 R12: ffffe6b00c000000 R13: ffff9bdb453dfd00 R14: 0000000000000000 R15: ffffffffbffffe FS: 00007f08f04e4740(0000) GS:ffff9be25fcc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000564787a30410 CR3: 000000010d4e2000 CR4: 00000000000006f0 Call Trace: <TASK> unpoison_memory+0x2f3/0x590 simple_attr_write_xsigned.constprop.0.isra.0+0xb3/0x110 debugfs_attr_write+0x42/0x60 full_proxy_write+0x5b/0x80 vfs_write+0xd5/0x540 ksys_write+0x64/0xe0 do_syscall_64+0xb9/0x1d0 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f08f0314887 RSP: 002b:00007ffce710078 EFLAGS: 00000246 ORIG_RAX: 0000000000000001 RAX: ffffffffbffffda RBX: 0000000000000009 RCX: 00007f08f0314887 RDX: 0000000000000009 RSI: 0000564787a30410 RDI: 0000000000000001 RBP: 0000564787a30410 R08: 000000000000fefe R09: 000000007ffffff R10: 0000000000000000 R11: 0000000000000246 R12: 0000000000000009 R13: 00007f08f041b780 R14: 00007f08f0417600 R15: 00007f08f0416a00 </TASK> Modules linked in: hwpoison_inject ---[end trace 0000000000000000]--- RIP: 0010:unpoison_memory+0x2f3/0x590 RSP: 0018:ffffa57fc8787d60 EFLAGS: 00000246 RAX: 0000000000000037 RBX: 0000000000000009 RCX: ffff9be25fcdc9c8 RDX: 0000000000000000 RSI: 0000000000000027 RDI: ffff9be25fcdc9c0 RBP: 0000000000300000 R08: ffffffffb4956f88 R09: 00000000000009ffb R10: 0000000000000284 R11: ffffffffb4926fa0 R12: ffffe6b00c000000 R13: ffff9bdb453dfd00 R14: 0000000000000000 R15: ffffffffbffffe FS: 00007f08f04e4740(0000) GS:ffff9be25fcc0000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000564787a30410 CR3: 000000010d4e2000 CR4: 00000000000006f0 Kernel panic - not syncing: Fatal exception Kernel Offset: 0x31c00000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-0xffffffffbfffffff) ---[end Kernel panic - not syncing: Fatal exception]--- The root cause is that unpoison_memory() tries to check the PG_HWPoison flag of an uninitialized page. So VM_BUG_ON_PAGE(PagePoisoned(page)) is triggered. This can be reproduced by below steps: 1.Offline memory block: echo offline > /sys/devices/system/memory/memory12/state 2.Get offlined memory pfn: page-types -b n -rln 3.Write pfn to unpoison-pfn echo <pfn> > /sys/kernel/debug/hwpoison/unpoison-pfn This scenario can be identified by pfn_to_online_page() returning NULL. And ZONE_DEVICE pages are never expected, so we can simply fail if pfn_to_online_page() == NULL to fix the bug.

CVE ID	Description
CVE-2025-39884	In the Linux kernel, the following vulnerability has been resolved: btrfs: fix subvolume deletion lockup caused by inodes xarray race There is a race condition between inode eviction and inode caching that can cause a live struct btrfs_inode to be missing from the root->inodes xarray. Specifically, there is a window during evict() between the inode being unhashed and deleted from the xarray. If btrfs_iget() is called for the same inode in that window, it will be recreated and inserted into the xarray, but then eviction will delete the new entry, leaving nothing in the xarray: Thread 1 Thread 2 ----- evict() remove_inode_hash() btrfs_iget_path() btrfs_iget_locked() btrfs_read_locked_inode() btrfs_add_inode_to_root() destroy_inode() btrfs_destroy_inode() btrfs_del_inode_from_root() __xa_erase In turn, this can cause issues for subvolume deletion. Specifically, if an inode is in this lost state, and all other inodes are evicted, then btrfs_del_inode_from_root() will call btrfs_add_dead_root() prematurely. If the lost inode has a delayed_node attached to it, then when btrfs_clean_one_deleted_snapshot() calls btrfs_kill_all_delayed_nodes(), it will loop forever because the delayed_nodes xarray will never become empty (unless memory pressure forces the inode out). We saw this manifest as soft lockups in production. Fix it by only deleting the xarray entry if it matches the given inode (using __xa_cmpxchg()).
CVE-2025-39885	In the Linux kernel, the following vulnerability has been resolved: ocfs2: fix recursive semaphore deadlock in fiemap call syzbot detected a OCFS2 hang due to a recursive semaphore on a FS_IOC_FIEMAP of the extent list on a specially crafted mmap file. context_switch kernel/sched/core.c:5357 [inline] __schedule+0x1798/0x4cc0 kernel/sched/core.c:6961 __schedule_loop kernel/sched/core.c:7043 [inline] schedule+0x165/0x360 kernel/sched/core.c:7058 schedule_preempt_disabled+0x13/0x30 kernel/sched/core.c:7115 rwsem_down_write_slowpath+0x872/0xfe0 kernel/locking/rwsem.c:1185 __down_write_common kernel/locking/rwsem.c:1317 [inline] __down_write kernel/locking/rwsem.c:1326 [inline] down_write+0x1ab/0x1f0 kernel/locking/rwsem.c:1591 ocfs2_page_mkwrite+0x2ff/0xc40 fs/ocfs2/mmap.c:142 do_page_mkwrite+0x14d/0x310 mm/memory.c:3361 wp_page_shared mm/memory.c:3762 [inline] do_wp_page+0x268d/0x5800 mm/memory.c:3981 handle_pte_fault mm/memory.c:6068 [inline] __handle_mm_fault+0x1033/0x5440 mm/memory.c:6195 handle_mm_fault+0x40a/0x8e0 mm/memory.c:6364 do_user_addr_fault+0x764/0x1390 arch/x86/mm/fault.c:1387 handle_page_fault arch/x86/mm/fault.c:1476 [inline] exc_page_fault+0x76/0xf0 arch/x86/mm/fault.c:1532 asm_exc_page_fault+0x26/0x30 arch/x86/include/asm/identry.h:623 RIP: 0010:copy_user_generic arch/x86/include/asm/uaccess_64.h:126 [inline] RIP: 0010:raw_copy_to_user arch/x86/include/asm/uaccess_64.h:147 [inline] RIP: 0010:_inline_copy_to_user include/linux/uaccess.h:197 [inline] RIP: 0010:_copy_to_user+0x85/0xb0 lib/usercopy.c:26 Code: e8 00 bc f7 fc 4d 39 fc 72 3d 4d 39 ec 77 38 e8 91 b9 f7 fc 4c 89 f7 89 de e8 47 25 5b fd 0f 01 cb 4c 89 ff 48 89 d9 4c 89 f6 <f3> a4 0f 1f 00 48 89 cb 0f 01 ca 48 89 d8 5b 41 5c 41 5d 41 5e 41 RSP: 0018:ffffc9000403f950 EFLAGS: 00050256 RAX: ffffffff84c7f101 RBX: 0000000000000038 RCX: 0000000000000038 RDX: 0000000000000000 RSI: fffffc9000403f9e0 RDI: 0000200000000060 RBP: fffffc9000403fa90 R08: fffffc9000403fa17 R09: 1ffff92000807f42 R10: dffffc0000000000 R11: fffff52000807f43 R12: 0000200000000098 R13: 00007fffffff000 R14: fffffc9000403f9e0 R15: 0000200000000060 copy_to_user include/linux/uaccess.h:225 [inline] fiemap_fill_next_extent+0x1c0/0x390 fs/ioctl.c:145 ocfs2_fiemap+0x888/0xc90 fs/ocfs2/extent_map.c:806 ioctl_fiemap fs/ioctl.c:220 [inline] do_vfs_ioctl+0x1173/0x1430 fs/ioctl.c:532 __do_sys_ioctl fs/ioctl.c:596 [inline] __se_sys_ioctl+0x82/0x170 fs/ioctl.c:584 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f RIP: 0033:0x7f5f13850fd9 RSP: 002b:00007ffe3b3518b8 EFLAGS: 00000246 ORIG_RAX: 0000000000000010 RAX: ffffffff84c7f101 RBX: 0000200000000000 RCX: 00007f5f13850fd9 RDX: 0000200000000040 RSI: 00000000c020660b RDI: 0000000000000004 RBP: 6165627472616568 R08: 0000000000000000 R09: 0000000000000000 R10: 0000000000000000 R11: 0000000000000246 R12: 00007ffe3b3518f0 R13: 00007ffe3b351b18 R14: 431bde82d7b634db R15: 00007f5f1389a03b ocfs2_fiemap() takes a read lock of the ip_alloc_sem semaphore (since v2.6.22-527-g7307de80510a) and calls fiemap_fill_next_extent() to read the extent list of this running mmap executable. The user supplied buffer to hold the fiemap information page faults calling ocfs2_page_mkwrite() which will take a write lock (since v2.6.27-38-g00dc417fa3e7) of the same semaphore. This recursive semaphore will hold filesystem locks and causes a hang of the filesystem. The ip_alloc_sem protects the inode extent list and size. Release the read semaphore before calling fiemap_fill_next_extent() in ocfs2_fiemap() and ocfs2_fiemap_inline(). This does an unnecessary semaphore lock/unlock on the last extent but simplifies the error path.

CVE ID	Description
CVE-2025-39886	In the Linux kernel, the following vulnerability has been resolved: bpf: Tell memcg to use allow_spinning=false path in bpf_timer_init() Currently, calling bpf_map_kmalloc_node() from __bpf_async_init() can cause various locking issues; see the following stack trace (edited for style) as one example: ... [10.011566] do_raw_spin_lock.cold [10.011570] try_to_wake_up (5) double-acquiring the same [10.011575] kick_pool rq_lock, causing a hardlockup [10.011579] __queue_work [10.011582] queue_work_on [10.011585] kernfs_notify [10.011589] cgroup_file_notify [10.011593] try_charge_memcg (4) memcg accounting raises an [10.011597] obj_cgroup_charge_pages MEMCG_MAX event [10.011599] obj_cgroup_charge_account [10.011600] __memcg_slab_post_alloc_hook [10.011603] __kmalloc_node_noprof ... [10.011611] bpf_map_kmalloc_node [10.011612] __bpf_async_init [10.011615] bpf_timer_init (3) BPF calls bpf_timer_init() [10.011617] bpf_prog_XXXXXXXXXXXXXXX_fcg_runnable [10.011619] bpf__sched_ext_ops_runnable [10.011620] enqueue_task_scx (2) BPF runs with rq_lock held [10.011622] enqueue_task [10.011626] ttwu_do_activate [10.011629] sched_ttwu_pending (1) grabs rq_lock ... The above was reproduced on bpf-next (b338cf849ec8) by modifying ./tools/sched_ext/scx_flatcg.bpf.c to call bpf_timer_init() during ops.runnable(), and hacking the memcg accounting code a bit to make a bpf_timer_init() call more likely to raise an MEMCG_MAX event. We have also run into other similar variants (both internally and on bpf-next), including double-acquiring cgroup_file_kn_lock, the same worker_pool::lock, etc. As suggested by Shakeel, fix this by using __GFP_HIGH instead of GFP_ATOMIC in __bpf_async_init(), so that e.g. if try_charge_memcg() raises an MEMCG_MAX event, we call __memcg_memory_event() with @allow_spinning=false and avoid calling cgroup_file_notify() there. Depends on mm patch "memcg: skip cgroup_file_notify if spinning is not allowed": https://lore.kernel.org/bpf/20250905201606.66198-1-shakeel.butt@linux.dev/ v0 approach s/bpf_map_kmalloc_node/bpf_mem_alloc/ https://lore.kernel.org/bpf/20250905061919.439648-1-yepeilin@google.com/ v1 approach: https://lore.kernel.org/bpf/20250905234547.862249-1-yepeilin@google.com/
CVE-2025-39889	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: l2cap: Check encryption key size on incoming connection This is required for passing GAP/SEC/SEM/BI-04-C PTS test case: Security Mode 4 Level 4, Responder - Invalid Encryption Key Size - 128 bit This tests the security key with size from 1 to 15 bytes while the Security Mode 4 Level 4 requests 16 bytes key size. Currently PTS fails with the following logs: - expected:Connection Response: Code: [3 (0x03)] Code Identifier: (lt)Wildcard: Exists(gt) Length: [8 (0x0008)] Destination CID: (lt)Wildcard: Exists(gt) Source CID: [64 (0x0040)] Result: [3 (0x0003)] Connection refused - Security block Status: (lt)Wildcard: Exists(gt), but received:Connection Response: Code: [3 (0x03)] Code Identifier: [1 (0x01)] Length: [8 (0x0008)] Destination CID: [64 (0x0040)] Source CID: [64 (0x0040)] Result: [0 (0x0000)] Connection Successful Status: [0 (0x0000)] No further information available And HCI logs: < HCI Command: Read Encrypti.. (0x05 0x0008) plen 2 Handle: 14 Address: 00:1B:DC:F2:24:10 (Vencer Co., Ltd.) > HCI Event: Command Complete (0x0e) plen 7 Read Encryption Key Size (0x05 0x0008) ncmd 1 Status: Success (0x00) Handle: 14 Address: 00:1B:DC:F2:24:10 (Vencer Co., Ltd.) Key size: 7 > ACL Data RX: Handle 14 flags 0x02 dlen 12 L2CAP: Connection Request (0x02) ident 1 len 4 PSM: 4097 (0x1001) Source CID: 64 < ACL Data TX: Handle 14 flags 0x00 dlen 16 L2CAP: Connection Response (0x03) ident 1 len 8 Destination CID: 64 Source CID: 64 Result: Connection successful (0x0000) Status: No further information available (0x0000)
CVE-2025-39890	In the Linux kernel, the following vulnerability has been resolved: wifi: ath12k: fix memory leak in ath12k_service_ready_ext_event Currently, in ath12k_service_ready_ext_event(), svc_rdy_ext.mac_phy_caps is not freed in the failure case, causing a memory leak. The following trace is observed in kmemleak: unreferenced object 0xffff8b3eb5789c00 (size 1024): comm "softirq", pid 0, jiffies 4294942577 hex dump (first 32 bytes): 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 7b 00 00 10{ ... 01 00 00 00 00 00 00 00 01 00 00 00 1f 38 00 00 8.. backtrace (crc 44e1c357): __kmalloc_noprof+0x30b/0x410 ath12k_wmi_mac_phy_caps_parse+0x84/0x100 [ath12k] ath12k_wmi_tlv_iter+0x5e/0x140 [ath12k] ath12k_wmi_svc_rdy_ext_parse+0x308/0x4c0 [ath12k] ath12k_wmi_tlv_iter+0x5e/0x140 [ath12k] ath12k_service_ready_ext_event.isra.0+0x44/0xd0 [ath12k] ath12k_wmi_op_rx+0x2eb/0xd70 [ath12k] ath12k_htc_rx_completion_handler+0x1f4/0x330 [ath12k] ath12k_ce_recv_process_cb+0x218/0x300 [ath12k] ath12k_pci_ce_workqueue+0x1b/0x30 [ath12k] process_one_work+0x219/0x680 bh_worker+0x198/0x1f0 tasklet_action+0x13/0x30 handle_softirqs+0xca/0x460 __irq_exit_rcu+0xbe/0x110 irq_exit_rcu+0x9/0x30 Free svc_rdy_ext.mac_phy_caps in the error case to fix this memory leak. Tested-on: QCN9274 hw2.0 PCI WLAN.WBE.1.4.1-00199-QCAHKSUPL_SILICONZ-1
CVE-2025-39891	In the Linux kernel, the following vulnerability has been resolved: wifi: mwifiex: Initialize the chan_stats array to zero The adapter->chan_stats[] array is initialized in mwifiex_init_channel_scan_gap() with vmalloc(), which doesn't zero out memory. The array is filled in mwifiex_update_chan_statistics() and then the user can query the data in mwifiex_cfg80211_dump_survey(). There are two potential issues here. What if the user calls mwifiex_cfg80211_dump_survey() before the data has been filled in. Also the mwifiex_update_chan_statistics() function doesn't necessarily initialize the whole array. Since the array was not initialized at the start that could result in an information leak. Also this array is pretty small. It's a maximum of 900 bytes so it's more appropriate to use kcalloc() instead vmalloc().

CVE ID	Description
CVE-2025-39894	In the Linux kernel, the following vulnerability has been resolved: netfilter: br_netfilter: do not check confirmed bit in br_nf_local_in() after confirm When send a broadcast packet to a tap device, which was added to a bridge, br_nf_local_in() is called to confirm the conntrack. If another conntrack with the same hash value is added to the hash table, which can be triggered by a normal packet to a non-bridge device, the below warning may happen. -----[cut here]----- WARNING: CPU: 1 PID: 96 at net/bridge/br_netfilter_hooks.c:632 br_nf_local_in+0x168/0x200 CPU: 1 UID: 0 PID: 96 Comm: tap_send Not tainted 6.17.0-rc2-dirty #44 PREEMPT(voluntary) RIP: 0010:br_nf_local_in+0x168/0x200 Call Trace: <TASK> nf_hook_slow+0x3e/0xf0 br_pass_frame_up+0x103/0x180 br_handle_frame_finish+0x2de/0x5b0 br_nf_hook_thresh+0xc0/0x120 br_nf_pre_routing_finish+0x168/0x3a0 br_nf_pre_routing+0x237/0x5e0 br_handle_frame+0x1ec/0x3c0 __netif_receive_skb_core+0x225/0x1210 __netif_receive_skb_one_core+0x37/0xa0 netif_receive_skb +0x36/0x160 tun_get_user+0xa54/0x10c0 tun_chr_write_iter+0x65/0xb0 vfs_write+0x305/0x410 ksys_write+0x60/0xd0 do_syscall_64+0xa4/0x260 entry_SYSCALL_64_after_hwframe+0x77/0x7f </TASK> ---[end trace 0000000000000000]--- To solve the hash conflict, nf_ct_resolve_clash() try to merge the conntracks, and update skb->nfct. However, br_nf_local_in() still use the old ct from local variable 'nfct' after confirm(), which leads to this warning. If confirm() does not insert the conntrack entry and return NF_DROP, the warning may also occur. There is no need to reserve the WARN_ON_ONCE, just remove it.
CVE-2025-39895	In the Linux kernel, the following vulnerability has been resolved: sched: Fix sched_numa_find_nth_cpu() if mask offline sched_numa_find_nth_cpu() uses a bsearch to look for the 'closest' CPU in sched_domains_numa_masks and given cpus mask. However they might not intersect if all CPUs in the cpus mask are offline. bsearch will return NULL in that case, bail out instead of dereferencing a bogus pointer. The previous behaviour lead to this bug when using maxcpus=4 on an rk3399 (LLLLbb) (i.e. booting with all big CPUs offline): [1.422922] Unable to handle kernel paging request at virtual address fffffff800000000 [1.423635] Mem abort info: [1.423889] ESR = 0x0000000096000006 [1.424227] EC = 0x25: DABT (current EL), IL = 32 bits [1.424715] SET = 0, FnV = 0 [1.424995] EA = 0, S1PTW = 0 [1.425279] FSC = 0x06: level 2 translation fault [1.425735] Data abort info: [1.425998] ISV = 0, ISS = 0x000000006, ISS2 = 0x00000000 [1.426499] CM = 0, WnR = 0, TnD = 0, TagAccess = 0 [1.426952] GCS = 0, DirtyBit = 0, Xs = 0 [1.427428] swapper pgtable: 4k pages, 39-bit VAs, pgdp=0000000004a9f000 [1.428038] [fffffff8000000000] pgd=18000000f7fff403, p4d=18000000f7fff403, pud=18000000f7fff403, pmd=0000000000000000 [1.429014] Internal error: Oops: 0000000096000006 [#1] SMP [1.429525] Modules linked in: [1.429813] CPU: 3 UID: 0 PID: 1 Comm: swapper/0 Not tainted 6.17.0-rc4-dirty #343 PREEMPT [1.430559] Hardware name: Pine64 RockPro64 v2.1 (DT) [1.431012] pstate: 60000005 (nZCv daif - PAN -UAO -TCO -DIT -SSBS BTYPE=--) [1.431634] pc : sched_numa_find_nth_cpu+0x2a0/0x488 [1.432094] lr : sched_numa_find_nth_cpu+0x284/0x488 [1.432543] sp : fffffffc084e1b960 [1.432843] x29: fffffffc084e1b960 x28: fffffff80078a8800 x27: fffffffc0846eb1d0 [1.433495] x26: 0000000000000000 x25: 0000000000000000 x24: 0000000000000000 [1.434144] x23: 0000000000000000 x22: fffffff7f093 x21: fffffffc081de6378 [1.434792] x20: 0000000000000000 x19: 0000000fff7f093 x18: 0000000ffffff [1.435441] x17: 3030303866666666 x16: 66663d736b73616d x15: fffffffc104e1b5b7 [1.436091] x14: 0000000000000000 x13: fffffffc084712860 x12: 00000000000000372 [1.436739] x11: 0000000000000126 x10: fffffffc08476a860 x9 : ffffffc084712860 [1.437389] x8 : 00000000ffffff x7 : fffffffc08476a860 x6 : 0000000000000000 [1.438036] x5 : 000000000000bffa x4 : 0000000000000000 x3 : 0000000000000000 [1.438683] x2 : 0000000000000000 x1 : fffffffc0846eb000 x0 : fffffff8000407b68 [1.439332] Call trace: [1.439559] sched_numa_find_nth_cpu+0x2a0/0x488 (P) [1.440016] smp_call_function_any +0xc8/0xd0 [1.440416] armv8_pmu_init+0x58/0x27c [1.440770] armv8_cortex_a72_pmu_init +0x20/0x2c [1.441199] arm_pmu_device_probe+0x1e4/0x5e8 [1.441603] armv8_pmu_device_probe +0x1c/0x28 [1.442007] platform_probe+0x5c/0xac [1.442347] really_probe+0xbc/0x298 [1.442683] __driver_probe_device+0x78/0x12c [1.443087] driver_probe_device+0xdc/0x160 [1.443475] __driver_attach+0x94/0x19c [1.443833] bus_for_each_dev+0x74/0xd4 [1.444190] driver_attach +0x24/0x30 [1.444525] bus_add_driver+0xe4/0x208 [1.444874] driver_register+0x60/0x128 [1.445233] __platform_driver_register+0x24/0x30 [1.445662] armv8_pmu_driver_init+0x28/0x4c [1.446059] do_one_initcall+0x44/0x25c [1.446416] kernel_init_freeable+0x1dc/0x3bc [1.446820] kernel_init+0x20/0x1d8 [1.447151] ret_from_fork+0x10/0x20 [1.447493] Code: 90022e21 f000e5f5 910de2b5 2a1703e2 (f8767803) [1.448040] ---[end trace 0000000000000000]--- [1.448483] note: swapper/0[1] exited with preempt_count 1 [1.449047] Kernel panic - not syncing: Attempted to kill init! exitcode=0x0000000b [1.449741] SMP: stopping secondary CPUs [1.450105] Kernel Offset: disabled [1.450419] CPU features: 0x000000,00080000,20002001,0400421b [---truncated---
CVE-2025-39896	In the Linux kernel, the following vulnerability has been resolved: accel/ivpu: Prevent recovery work from being queued during device removal Use disable_work_sync() instead of cancel_work_sync() in ivpu_dev_fini() to ensure that no new recovery work items can be queued after device removal has started. Previously, recovery work could be scheduled even after canceling existing work, potentially leading to use-after-free bugs if recovery accessed freed resources. Rename ivpu_pm_cancel_recovery() to ivpu_pm_disable_recovery() to better reflect its new behavior.

CVE ID	Description
CVE-2025-39897	In the Linux kernel, the following vulnerability has been resolved: net: xilinx: axienet: Add error handling for RX metadata pointer retrieval Add proper error checking for dmaengine_desc_get_metadata_ptr() which can return an error pointer and lead to potential crashes or undefined behaviour if the pointer retrieval fails. Properly handle the error by unmapping DMA buffer, freeing the skb and returning early to prevent further processing with invalid data.
CVE-2025-39898	Rejected reason: This CVE ID has been rejected or withdrawn by its CVE Numbering Authority.
CVE-2025-39899	In the Linux kernel, the following vulnerability has been resolved: mm/userfaultfd: fix kmap_local LIFO ordering for CONFIG_HIGHPTPE With CONFIG_HIGHPTPE on 32-bit ARM, move_pages_pte() maps PTE pages using kmap_local_page(), which requires unmapping in Last-In-First-Out order. The current code maps dst_pte first, then src_pte, but unmaps them in the same order (dst_pte, src_pte), violating the LIFO requirement. This causes the warning in kunmap_local_indexed(): WARNING: CPU: 0 PID: 604 at mm/highmem.c:622 kunmap_local_indexed+0x178/0x17c addr !\n = __fix_to_virt(FIX_KMAP_BEGIN + idx) Fix this by reversing the unmap order to respect LIFO ordering. This issue follows the same pattern as similar fixes: - commit eca6828403b8 ("crypto: skcipher - fix mismatch between mapping and unmapping order") - commit 8cf57c6df818 ("nilfs2: eliminate staggered calls to kunmap in nilfs_rename") Both of which addressed the same fundamental requirement that kmap_local operations must follow LIFO ordering.
CVE-2025-39901	In the Linux kernel, the following vulnerability has been resolved: i40e: remove read access to debugfs files The 'command' and 'netdev_ops' debugfs files are a legacy debugging interface supported by the i40e driver since its early days by commit 02e9c290814c ("i40e: debugfs interface"). Both of these debugfs files provide a read handler which is mostly useless, and which is implemented with questionable logic. They both use a static 256 byte buffer which is initialized to the empty string. In the case of the 'command' file this buffer is literally never used and simply wastes space. In the case of the 'netdev_ops' file, the last command written is saved here. On read, the file contents are presented as the name of the device followed by a colon and then the contents of their respective static buffer. For 'command' this will always be "<device>: ". For 'netdev_ops', this will be "<device>: <last command written>". But note the buffer is shared between all devices operated by this module. At best, it is mostly meaningless information, and at worse it could be accessed simultaneously as there doesn't appear to be any locking mechanism. We have also recently received multiple reports for both read functions about their use of snprintf and potential overflow that could result in reading arbitrary kernel memory. For the 'command' file, this is definitely impossible, since the static buffer is always zero and never written to. For the 'netdev_ops' file, it does appear to be possible, if the user carefully crafts the command input, it will be copied into the buffer, which could be large enough to cause snprintf to truncate, which then causes the copy_to_user to read beyond the length of the buffer allocated by kzalloc. A minimal fix would be to replace snprintf() with scnprintf() which would cap the return to the number of bytes written, preventing an overflow. A more involved fix would be to drop the mostly useless static buffers, saving 512 bytes and modifying the read functions to stop needing those as input. Instead, lets just completely drop the read access to these files. These are debug interfaces exposed as part of debugfs, and I don't believe that dropping read access will break any script, as the provided output is pretty useless. You can find the netdev name through other more standard interfaces, and the 'netdev_ops' interface can easily result in garbage if you issue simultaneous writes to multiple devices at once. In order to properly remove the i40e_dbg_netdev_ops_buf, we need to refactor its write function to avoid using the static buffer. Instead, use the same logic as the i40e_dbg_command_write, with an allocated buffer. Update the code to use this instead of the static buffer, and ensure we free the buffer on exit. This fixes simultaneous writes to 'netdev_ops' on multiple devices, and allows us to remove the now unused static buffer along with removing the read access.
CVE-2025-39902	In the Linux kernel, the following vulnerability has been resolved: mm/slab: avoid accessing metadata when pointer is invalid in object_err() object_err() reports details of an object for further debugging, such as the freelist pointer, redzone, etc. However, if the pointer is invalid, attempting to access object metadata can lead to a crash since it does not point to a valid object. One known path to the crash is when alloc_consistency_checks() determines the pointer to the allocated object is invalid because of a freelist corruption, and calls object_err() to report it. The debug code should report and handle the corruption gracefully and not crash in the process. In case the pointer is NULL or check_valid_pointer() returns false for the pointer, only print the pointer value and skip accessing metadata.

CVE ID	Description
CVE-2025-39905	In the Linux kernel, the following vulnerability has been resolved: net: phylink: add lock for serializing concurrent pl->phydev writes with resolver Currently phylink_resolve() protects itself against concurrent phylink_bringup_phy() or phylink_disconnect_phy() calls which modify pl->phydev by relying on pl->state_mutex. The problem is that in phylink_resolve(), pl->state_mutex is in a lock inversion state with pl->phydev->lock. So pl->phydev->lock needs to be acquired prior to pl->state_mutex. But that requires dereferencing pl->phydev in the first place, and without pl->state_mutex, that is racy. Hence the reason for the extra lock. Currently it is redundant, but it will serve a functional purpose once mutex_lock(&phy->lock) will be moved outside of the mutex_lock(&pl->state_mutex) section. Another alternative considered would have been to let phylink_resolve() acquire the rtnl_mutex, which is also held when phylink_bringup_phy() and phylink_disconnect_phy() are called. But since phylink_disconnect_phy() runs under rtnl_lock(), it would deadlock with phylink_resolve() when calling flush_work(&pl->resolve). Additionally, it would have been undesirable because it would have unnecessarily blocked many other call paths as well in the entire kernel, so the smaller-scoped lock was preferred.
CVE-2025-39907	In the Linux kernel, the following vulnerability has been resolved: mtd: rawnand: stm32_fmc2: avoid overlapping mappings on ECC buffer Avoid below overlapping mappings by using a contiguous non-cacheable buffer. [4.077708] DMA-API: stm32_fmc2_nfc 48810000.nand-controller: cacheline tracking EEXIST, overlapping mappings aren't supported [4.089103] WARNING: CPU: 1 PID: 44 at kernel/dma/debug.c:568 add_dma_entry+0x23c/0x300 [4.097071] Modules linked in: [4.100101] CPU: 1 PID: 44 Comm: kworker/u4:2 Not tainted 6.1.82 #1 [4.106346] Hardware name: STMicroelectronics STM32MP257F VALID1 SNOR / MB1704 (LPDDR4 Power discrete) + MB1703 + MB1708 (SNOR MB1730) (DT) [4.118824] Workqueue: events_unbound deferred_probe_work_func [4.124674] pstate: 60000005 (nZCv daif -PAN -UAO -TCO -DIT -SSBS BTYPE=--)[4.131624] pc : add_dma_entry+0x23c/0x300 [4.135658] lr : add_dma_entry+0x23c/0x300 [4.139792] sp : ffff800009dbb490 [4.143016] x29: ffff800009dbb4a0 x28: 0000000004008022 x27: ffff8000098a6000 [4.150174] x26: 0000000000000000 x25: ffff8000099e7000 x24: ffff8000099e7de8 [4.157231] x23: 00000000fffff x22: 0000000000000000 x21: ffff8000098a6a20 [4.164388] x20: ffff000080964180 x19: ffff800009819ba0 x18: 0000000000000006 [4.171545] x17: 6361727420656e69 x16: 6c6568636163203a x15: 72656c6c6f72746e [4.178602] x14: 6f632d646e616e2e x13: ffff800009832f58 x12: 00000000000004ec [4.185759] x11: 00000000000001a4 x10: ffff80000988af58 x9 : ffff800009832f58 [4.192916] x8 : 00000000ffffeff x7 : ffff80000988af58 x6 : 80000000fffff000 [4.199972] x5 : 000000000000bff4 x4 : 0000000000000000 x3 : 0000000000000000 [4.207128] x2 : 0000000000000000 x1 : 0000000000000000 x0 : ffff0000812d2c40 [4.214185] Call trace: [4.216605] add_dma_entry+0x23c/0x300 [4.220338] debug_dma_map_sg+0x198/0x350 [4.224373] __dma_map_sg_attrs+0xa0/0x110 [4.228411] dma_map_sg_attrs+0x10/0x2c [4.232247] stm32_fmc2_nfc_xfer.isra.0+0x1c8/0x3fc [4.237088] stm32_fmc2_nfc_seq_read_page+0xc8/0x174 [4.242127] nand_read_oob+0x1d4/0x8e0 [4.245861] mtd_read_oob_std+0x58/0x84 [4.249596] mtd_read_oob+0x90/0x150 [4.253231] mtd_read+0x68/0xac
CVE-2025-39908	In the Linux kernel, the following vulnerability has been resolved: net: dev_ioctl: take ops lock in hwtstamp lower paths ndo hwtstamp callbacks are expected to run under the per-device ops lock. Make the lower get/set paths consistent with the rest of ndo invocations. Kernel log: WARNING: CPU: 13 PID: 51364 at ./include/net/netdev_lock.h:70 __netdev_update_features+0x4bd/0xe60 ... RIP: 0010: __netdev_update_features+0x4bd/0xe60 ... Call Trace: <TASK> netdev_update_features+0x1f/0x60 mlx5_hwtstamp_set+0x181/0x290 [mlx5_core] mlx5e_hwtstamp_set+0x19/0x30 [mlx5_core] dev_set_hwtstamp_phylib+0x9f/0x220 dev_set_hwtstamp_phylib+0x9f/0x220 dev_set_hwtstamp+0x13d/0x240 dev_ioctl+0x12f/0x4b0 sock_ioctl+0x171/0x370 __x64_sys_ioctl+0x3f7/0x900 ? __sys_setsockopt+0x69/0xb0 do_syscall_64+0x6f/0x2e0 entry_SYSCALL_64_after_hwframe+0x4b/0x53 ... </TASK> ... ---[end trace 0000000000000000]--- Note that the mlx5_hwtstamp_set and mlx5e_hwtstamp_set functions shown in the trace come from an in progress patch converting the legacy ioctl to ndo_hwtstamp_get/set and are not present in mainline.
CVE-2025-39909	In the Linux kernel, the following vulnerability has been resolved: mm/damon/lru_sort: avoid divide-by-zero in damon_lru_sort_apply_parameters() Patch series "mm/damon: avoid divide-by-zero in DAMON module's parameters application". DAMON's RECLAIM and LRU_SORT modules perform no validation on user-configured parameters during application, which may lead to division-by-zero errors. Avoid the divide-by-zero by adding validation checks when DAMON modules attempt to apply the parameters. This patch (of 2): During the calculation of 'hot_thres' and 'cold_thres', either 'sample_interval' or 'aggr_interval' is used as the divisor, which may lead to division-by-zero errors. Fix it by directly returning -EINVAL when such a case occurs. Additionally, since 'aggr_interval' is already required to be set no smaller than 'sample_interval' in damon_set_attrs(), only the case where 'sample_interval' is zero needs to be checked.

CVE ID	Description
CVE-2025-39910	In the Linux kernel, the following vulnerability has been resolved: mm/vmalloc, mm/kasan: respect gfp mask in kasan_populate_vmalloc() kasan_populate_vmalloc() and its helpers ignore the caller's gfp_mask and always allocate memory using the hardcoded GFP_KERNEL flag. This makes them inconsistent with vmalloc(), which was recently extended to support GFP_NOFS and GFP_NOIO allocations. Page table allocations performed during shadow population also ignore the external gfp_mask. To preserve the intended semantics of GFP_NOFS and GFP_NOIO, wrap the apply_to_page_range() calls into the appropriate memalloc scope. xfs calls vmalloc with GFP_NOFS, so this bug could lead to deadlock. There was a report here https://lkml.kernel.org/r/686ea951.050a0220.385921.0016.GAE@google.com This patch: - Extends kasan_populate_vmalloc() and helpers to take gfp_mask; - Passes gfp_mask down to alloc_pages_bulk() and __get_free_page(); - Enforces GFP_NOFS/NOIO semantics with memalloc_*_save()/restore() around apply_to_page_range(); - Updates vmalloc.c and percpu allocator call sites accordingly.
CVE-2025-39911	In the Linux kernel, the following vulnerability has been resolved: i40e: fix IRQ freeing in i40e_vsi_request_irq_msix error path If request_irq() in i40e_vsi_request_irq_msix() fails in an iteration later than the first, the error path wants to free the IRQs requested so far. However, it uses the wrong dev_id argument for free_irq(), so it does not free the IRQs correctly and instead triggers the warning: Trying to free already-free IRQ 173 WARNING: CPU: 25 PID: 1091 at kernel/irq/manage.c:1829 __free_irq+0x192/0x2c0 Modules linked in: i40e(+) [...] CPU: 25 UID: 0 PID: 1091 Comm: NetworkManager Not tainted 6.17.0-rc1+ #1 PREEMPT(lazy) Hardware name: [...] RIP: 0010: __free_irq+0x192/0x2c0 [...] Call Trace: <TASK> free_irq+0x32/0x70 i40e_vsi_request_irq_msix.cold+0x63/0x8b [i40e] i40e_vsi_request_irq+0x79/0x80 [i40e] i40e_vsi_open+0x21f/0x2f0 [i40e] i40e_open+0x63/0x130 [i40e] __dev_open+0xfc/0x210 __dev_change_flags+0x1fc/0x240 netif_change_flags+0x27/0x70 do_setlink.isra.0+0x341/0xc70 rtnl_newlink+0x468/0x860 rtnetlink_rcv_msg+0x375/0x450 netlink_rcv_skb+0x5c/0x110 netlink_unicast+0x288/0x3c0 netlink_sendmsg+0x20d/0x430 ____sys_sendmsg+0x3a2/0x3d0 ____sys_sendmsg+0x99/0xe0 __sys_sendmsg+0x8a/0xf0 do_syscall_64+0x82/0x2c0 entry_SYSCALL_64_after_hwframe+0x76/0x7e [...] </TASK> ---[end trace 0000000000000000]--- Use the same dev_id for free_irq() as for request_irq(). I tested this with inserting code to fail intentionally.
CVE-2025-39913	In the Linux kernel, the following vulnerability has been resolved: tcp_bpf: Call sk_msg_free() when tcp_bpf_send_verdict() fails to allocate psock->cork. syzbot reported the splat below. [0] The repro does the following: 1. Load a sk_msg prog that calls bpf_msg_cork_bytes(msg, cork_bytes) 2. Attach the prog to a SOCKMAP 3. Add a socket to the SOCKMAP 4. Activate fault injection 5. Send data less than cork_bytes At 5., the data is carried over to the next sendmsg() as it is smaller than the cork_bytes specified by bpf_msg_cork_bytes(). Then, tcp_bpf_send_verdict() tries to allocate psock->cork to hold the data, but this fails silently due to fault injection + __GFP_NOWARN. If the allocation fails, we need to revert the sk->sk_forward_alloc change done by sk_msg_alloc(). Let's call sk_msg_free() when tcp_bpf_send_verdict fails to allocate psock->cork. The "*copied" also needs to be updated such that a proper error can be returned to the caller, sendmsg. It fails to allocate psock->cork. Nothing has been corked so far, so this patch simply sets "*copied" to 0. [0]: WARNING: net/ipv4/af_inet.c:156 at inet_sock_destruct+0x623/0x730 net/ipv4/af_inet.c:156, CPU#1: syz-executor/5983 Modules linked in: CPU: 1 UID: 0 PID: 5983 Comm: syz-executor Not tainted syzkaller #0 PREEMPT(full) Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 07/12/2025 RIP: 0010:inet_sock_destruct+0x623/0x730 net/ipv4/af_inet.c:156 Code: 0f 0b 90 e9 62 fe ff ff e8 7a db b5 f7 90 0f 0b 90 e9 95 fe ff ff e8 6c db b5 f7 90 0f 0b 90 e9 bb fe ff ff e8 5e db b5 f7 90 <0f> 0b 90 e9 e1 fe ff ff 89 f9 80 e1 07 80 c1 03 38 c1 0f 8c 9f fc RSP: 0018:ffff90000a08b48 EFLAGS: 00010246 RAX: ffffffff8a09d0b2 RBX: dffffc0000000000 RCX: ffff888024a23c80 RDX: 0000000000000100 RSI: 00000000000000ff RDI: 0000000000000000 RBP: 00000000000000ff R08: ffff88807e07c627 R09: 1ffff1100fc0f8c4 R10: dffffc0000000000 R11: fffff1100fc0f8c5 R12: ffff88807e07c380 R13: dffffc0000000000 R14: ffff88807e07c60c R15: 1ffff1100fc0f872 FS: 00005555604c4500(0000) GS:ffff888125af1000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00005555604df5c8 CR3: 0000000032b06000 CR4: 00000000003526f0 Call Trace: <IRQ> __sk_destruct+0x86/0x660 net/core/sock.c:2339 rcu_do_batch kernel/rcu/tree.c:2605 [inline] rcu_core+0xca8/0x1770 kernel/rcu/tree.c:2861 handle_softirqs+0x286/0x870 kernel/softirq.c:579 __do_softirq kernel/softirq.c:613 [inline] invoke_softirq kernel/softirq.c:453 [inline] __irq_exit_rcu+0xca/0x1f0 kernel/softirq.c:680 irq_exit_rcu+0x9/0x30 kernel/softirq.c:696 instr_sysvec_apic_timer_interrupt arch/x86/kernel/apic/apic.c:1052 [inline] sysvec_apic_timer_interrupt+0xa6/0xc0 arch/x86/kernel/apic/apic.c:1052 </IRQ>

CVE ID	Description
CVE-2025-39914	In the Linux kernel, the following vulnerability has been resolved: tracing: Silence warning when chunk allocation fails in trace_pid_write Syzkaller trigger a fault injection warning: WARNING: CPU: 1 PID: 12326 at tracepoint_add_func+0xbfc/0xeb0 Modules linked in: CPU: 1 UID: 0 PID: 12326 Comm: syz.6.10325 Tainted: G U 6.14.0-rc5-syzkaller #0 Tainted: [U]=USER Hardware name: Google Compute Engine/Google Compute Engine RIP: 0010:tracepoint_add_func+0xbfc/0xeb0 kernel/tracepoint.c:294 Code: 09 fe ff 90 0f 0b 90 0f b6 74 24 43 31 ff 41 bc ea ff ff ff RSP: 0018:ffffc9000414fb48 EFLAGS: 00010283 RAX: 00000000000012a1 RBX: ffffffff8e240ae0 RCX: fffff90014b78000 RDX: 00000000000080000 RSI: ffffffff81bbd78b RDI: 0000000000000001 RBP: 0000000000000000 R08: 0000000000000001 R09: 0000000000000000 R10: 0000000000000001 R11: 0000000000000001 R12: ffffffff81bbd78b R13: 0000000000000000 R14: dffffc0000000000 R15: ffffffff81c264f0 FS: 00007f27217f66c0(0000) GS:ffff8880b8700000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 0000001b2e80dff8 CR3: 00000000268f8000 CR4: 0000000003526f0 DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000 DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400 Call Trace: <TASK> tracepoint_probe_register_prio+0xc0/0x110 kernel/tracepoint.c:464 register_trace_prio_sched_switch include/trace/events/sched.h:222 [inline] register_pid_events kernel/trace/trace_events.c:2354 [inline] event_pid_write.isra.0+0x439/0x7a0 kernel/trace/trace_events.c:2425 vfs_write+0x24c/0x1150 fs/read_write.c:677 ksys_write+0x12b/0x250 fs/read_write.c:731 do_syscall_x64 arch/x86/entry/common.c:52 [inline] do_syscall_64+0xcd/0x250 arch/x86/entry/common.c:83 entry_SYSCALL_64_after_hwframe+0x77/0x7f We can reproduce the warning by following the steps below: 1. echo 8 >> set_event_notrace_pid. Let tr->filtered_pids owns one pid and register sched_switch tracepoint. 2. echo '' >> set_event_pid, and perform fault injection during chunk allocation of trace_pid_list_alloc. Let pid_list with no pid and assign to tr->filtered_pids. 3. echo '' >> set_event_pid. Let pid_list is NULL and assign to tr->filtered_pids. 4. echo 9 >> set_event_pid, will trigger the double register sched_switch tracepoint warning. The reason is that syzkaller injects a fault into the chunk allocation in trace_pid_list_alloc, causing a failure in trace_pid_list_set, which may trigger double register of the same tracepoint. This only occurs when the system is about to crash, but to suppress this warning, let's add failure handling logic to trace_pid_list_set.
CVE-2025-39916	In the Linux kernel, the following vulnerability has been resolved: mm/damon/reclaim: avoid divide-by-zero in damon_reclaim_apply_parameters() When creating a new scheme of DAMON_RECLAIM, the calculation of 'min_age_region' uses 'aggr_interval' as the divisor, which may lead to division-by-zero errors. Fix it by directly returning -EINVAL when such a case occurs.
CVE-2025-39920	In the Linux kernel, the following vulnerability has been resolved: pcmcia: Add error handling for add_interval() in do_validate_mem() In the do_validate_mem(), the call to add_interval() does not handle errors. If kmalloc() fails in add_interval(), it could result in a null pointer being inserted into the linked list, leading to illegal memory access when sub_interval() is called next. This patch adds an error handling for the add_interval(). If add_interval() returns an error, the function will return early with the error code.
CVE-2025-39923	In the Linux kernel, the following vulnerability has been resolved: dmaengine: qcom: bam_dma: Fix DT error handling for num-channels/ees When we don't have a clock specified in the device tree, we have no way to ensure the BAM is on. This is often the case for remotely-controlled or remotely-powered BAM instances. In this case, we need to read num-channels from the DT to have all the necessary information to complete probing. However, at the moment invalid device trees without clock and without num-channels still continue probing, because the error handling is missing return statements. The driver will then later try to read the number of channels from the registers. This is unsafe, because it relies on boot firmware and lucky timing to succeed. Unfortunately, the lack of proper error handling here has been abused for several Qualcomm SoCs upstream, causing early boot crashes in several situations [1, 2]. Avoid these early crashes by erroring out when any of the required DT properties are missing. Note that this will break some of the existing DTs upstream (mainly BAM instances related to the crypto engine). However, clearly these DTs have never been tested properly, since the error in the kernel log was just ignored. It's safer to disable the crypto engine for these broken DTBs. [1]: https://lore.kernel.org/r/CY01EKQVWE36.B9X5TDXAREPF@fairphone.com/ [2]: https://lore.kernel.org/r/20230626145959.646747-1-krzysztof.kozlowski@linaro.org/
CVE-2025-39925	In the Linux kernel, the following vulnerability has been resolved: can: j1939: implement NETDEV_UNREGISTER notification handler syzbot is reporting unregister_netdevice: waiting for vcan0 to become free. Usage count = 2 problem, for j1939 protocol did not have NETDEV_UNREGISTER notification handler for undoing changes made by j1939_sk_bind(). Commit 25fe97cb7620 ("can: j1939: move j1939_priv_put() into sk_destruct callback") expects that a call to j1939_priv_put() can be unconditionally delayed until j1939_sk_sock_destruct() is called. But we need to call j1939_priv_put() against an extra ref held by j1939_sk_bind() call (as a part of undoing changes made by j1939_sk_bind()) as soon as NETDEV_UNREGISTER notification fires (i.e. before j1939_sk_sock_destruct() is called via j1939_sk_release()). Otherwise, the extra ref on "struct j1939_priv" held by j1939_sk_bind() call prevents "struct net_device" from dropping the usage count to 1; making it impossible for unregister_netdevice() to continue. [mkl: remove space in front of label]

CVE ID	Description
CVE-2025-39927	In the Linux kernel, the following vulnerability has been resolved: ceph: fix race condition validating r_parent before applying state Add validation to ensure the cached parent directory inode matches the directory info in MDS replies. This prevents client-side race conditions where concurrent operations (e.g. rename) cause r_parent to become stale between request initiation and reply processing, which could lead to applying state changes to incorrect directory inodes. [idryomov: folded a kerneldoc fixup and a follow-up fix from Alex to move CEPH_CAP_PIN reference when r_parent is updated: When the parent directory lock is not held, req->r_parent can become stale and is updated to point to the correct inode. However, the associated CEPH_CAP_PIN reference was not being adjusted. The CEPH_CAP_PIN is a reference on an inode that is tracked for accounting purposes. Moving this pin is important to keep the accounting balanced. When the pin was not moved from the old parent to the new one, it created two problems: The reference on the old, stale parent was never released, causing a reference leak. A reference for the new parent was never acquired, creating the risk of a reference underflow later in ceph_mdsc_release_request(). This patch corrects the logic by releasing the pin from the old parent and acquiring it for the new parent when r_parent is switched. This ensures reference accounting stays balanced.]
CVE-2025-39929	In the Linux kernel, the following vulnerability has been resolved: smb: client: fix smbdirect_recv_io leak in smb_negotiate() error path During tests of another unrelated patch I was able to trigger this error: Objects remaining on __kmem_cache_shutdown()
CVE-2025-39931	In the Linux kernel, the following vulnerability has been resolved: crypto: af_alg - Set merge to zero early in af_alg_sendmsg If an error causes af_alg_sendmsg to abort, ctx->merge may contain a garbage value from the previous loop. This may then trigger a crash on the next entry into af_alg_sendmsg when it attempts to do a merge that can't be done. Fix this by setting ctx->merge to zero near the start of the loop.
CVE-2025-39932	In the Linux kernel, the following vulnerability has been resolved: smb: client: let smbd_destroy() call disable_work_sync(&info->post_send_credits_work) In smbd_destroy() we may destroy the memory so we better wait until post_send_credits_work is no longer pending and will never be started again. I actually just hit the case using rxe: WARNING: CPU: 0 PID: 138 at drivers/infiniband/sw/rxe/rxe_verbs.c:1032 rxe_post_recv+0x1ee/0x480 [rdma_rxe] ... [5305.686979] [T138] smbd_post_recv+0x445/0xc10 [cifs] [5305.687135] [T138] ? srso_alias_return_thunk+0x5/0xfbef5 [5305.687149] [T138] ? __kasan_check_write+0x14/0x30 [5305.687185] [T138] ? __pfx_smbd_post_recv+0x10/0x10 [cifs] [5305.687329] [T138] ? __pfx_raw_spin_lock_irqsave+0x10/0x10 [5305.687356] [T138] ? srso_alias_return_thunk+0x5/0xfbef5 [5305.687368] [T138] ? srso_alias_return_thunk+0x5/0xfbef5 [5305.687378] [T138] ? _raw_spin_unlock_irqrestore+0x11/0x60 [5305.687389] [T138] ? srso_alias_return_thunk+0x5/0xfbef5 [5305.687399] [T138] ? get_receive_buffer+0x168/0x210 [cifs] [5305.687555] [T138] smbd_post_send_credits+0x382/0x4b0 [cifs] [5305.687701] [T138] ? __pfx_smbd_post_send_credits+0x10/0x10 [cifs] [5305.687855] [T138] ? __pfx__schedule+0x10/0x10 [5305.687865] [T138] ? __pfx_raw_spin_lock_irq+0x10/0x10 [5305.687875] [T138] ? queue_delayed_work_on+0x8e/0xa0 [5305.687889] [T138] process_one_work+0x629/0xf80 [5305.687908] [T138] ? srso_alias_return_thunk+0x5/0xfbef5 [5305.687917] [T138] ? __kasan_check_write+0x14/0x30 [5305.687933] [T138] worker_thread+0x87f/0x1570 ... It means rxe_post_recv was called after rdma_destroy_qp(). This happened because put_receive_buffer() was triggered by ib_drain_qp() and called: queue_work(info->workqueue, &info->post_send_credits_work);
CVE-2025-39933	In the Linux kernel, the following vulnerability has been resolved: smb: client: let recv_done verify data_offset, data_length and remaining_data_length This is inspired by the related server fixes.
CVE-2025-39934	In the Linux kernel, the following vulnerability has been resolved: drm: bridge: anx7625: Fix NULL pointer dereference with early IRQ If the interrupt occurs before resource initialization is complete, the interrupt handler/worker may access uninitialized data such as the I2C tpc_client device, potentially leading to NULL pointer dereference.
CVE-2025-39937	In the Linux kernel, the following vulnerability has been resolved: net: rkill: gpio: Fix crash due to dereferencing uninitialized pointer Since commit 7d5e9737efda ("net: rkill: gpio: get the name and type from device property") rkill_find_type() gets called with the possibly uninitialized "const char *type_name;" local variable. On x86 systems when rkill-gpio binds to a "BCM4752" or "LNV4752" acpi_device, the rkill->type is set based on the ACPI acpi_device_id: rkill->type = (unsigned)id->driver_data; and there is no "type" property so device_property_read_string() will fail and leave type_name uninitialized, leading to a potential crash. rkill_find_type() does accept a NULL pointer, fix the potential crash by initializing type_name to NULL. Note likely sofar this has not been caught because: 1. Not many x86 machines actually have a "BCM4752"/"LNV4752" acpi_device 2. The stack happened to contain NULL where type_name is stored

CVE ID	Description
CVE-2025-39938	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: q6apm-lpass-dais: Fix NULL pointer dereference if source graph failed If earlier opening of source graph fails (e.g. ADSP rejects due to incorrect audioreach topology), the graph is closed and "dai_data->graph[dai->id]" is assigned NULL. Preparing the DAI for sink graph continues though and next call to q6apm_lpass_dai_prepare() receives dai_data->graph[dai->id]=NULL leading to NULL pointer exception: qcom-apm gprsvc:service:2:1: Error (1) Processing 0x01001002 cmd qcom-apm gprsvc:service:2:1: DSP returned error[1001002] 1 q6apm-lpass-dais 30000000.remoteproc:glink-edge:gpr:service@1:bedais: fail to start APM port 78 q6apm-lpass-dais 30000000.remoteproc:glink-edge:gpr:service@1:bedais: ASoC: error at snd_soc_pcm_dai_prepare on TX_CODECDMA_TX_3: -22 Unable to handle kernel NULL pointer dereference at virtual address 00000000000000a8 ... Call trace: q6apm_graph_media_format_pcm+0x48/0x120 (P) q6apm_lpass_dai_prepare+0x110/0x1b4 snd_soc_pcm_dai_prepare+0x74/0x108 __soc_pcm_prepare+0x44/0x160 dpcm_be_dai_prepare+0x124/0x1c0
CVE-2025-39940	In the Linux kernel, the following vulnerability has been resolved: dm-stripe: fix a possible integer overflow There's a possible integer overflow in stripe_io_hints if we have too large chunk size. Test if the overflow happened, and if it did, don't set limits->io_min and limits->io_opt;
CVE-2025-39942	In the Linux kernel, the following vulnerability has been resolved: ksmbd: smbdirect: verify remaining_data_length respects max_fragmented_rcv_size This is inspired by the check for data_offset + data_length.
CVE-2025-39943	In the Linux kernel, the following vulnerability has been resolved: ksmbd: smbdirect: validate data_offset and data_length field of smb_direct_data_transfer If data_offset and data_length of smb_direct_data_transfer struct are invalid, out of bounds issue could happen. This patch validate data_offset and data_length field in rcv_done.
CVE-2025-39944	In the Linux kernel, the following vulnerability has been resolved: octeonx2-pf: Fix use-after-free bugs in otx2_sync_tstamp() The original code relies on cancel_delayed_work() in otx2_ptp_destroy(), which does not ensure that the delayed work item syncstamp_work has fully completed if it was already running. This leads to use-after-free scenarios where otx2_ptp is deallocated by otx2_ptp_destroy(), while syncstamp_work remains active and attempts to dereference otx2_ptp in otx2_sync_tstamp(). Furthermore, the syncstamp_work is cyclic, the likelihood of triggering the bug is nonnegligible. A typical race condition is illustrated below: CPU 0 (cleanup) CPU 1 (delayed work callback) otx2_remove() otx2_ptp_destroy() otx2_sync_tstamp() cancel_delayed_work() kfree(ptp) ptp = container_of(...); //UAF ptp-> //UAF This is confirmed by a KASAN report: BUG: KASAN: slab-use-after-free in __run_timer_base.part.0+0x7d7/0x8c0 Write of size 8 at addr ffff88800aa09a18 by task bash/136 ... Call Trace: <IRQ> dump_stack_lvl+0x55/0x70 print_report+0xcf/0x610 ? __run_timer_base.part.0+0x7d7/0x8c0 kasan_report+0xb8/0xf0 ? __run_timer_base.part.0+0x7d7/0x8c0 __run_timer_base.part.0+0x7d7/0x8c0 ? __pfx__run_timer_base.part.0+0x10/0x10 ? __pfx_read_tsc+0x10/0x10 ? ktime_get+0x60/0x140 ? lapic_next_event+0x11/0x20 ? clockevents_program_event+0x1d4/0x2a0 run_timer_softirq+0xd1/0x190 handle_softirqs+0x16a/0x550 irq_exit_rcu+0xaf/0xe0 sysvec_apic_timer_interrupt+0x70/0x80 </IRQ> ... Allocated by task 1: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 __kasan_kmalloc+0x7f/0x90 otx2_ptp_init+0xb1/0x860 otx2_probe+0x4eb/0xc30 local_pci_probe+0xdc/0x190 pci_device_probe+0x2fe/0x470 really_probe+0x1ca/0x5c0 __driver_probe_device+0x248/0x310 driver_probe_device+0x44/0x120 __driver_attach+0xd2/0x310 bus_for_each_dev+0xed/0x170 bus_add_driver+0x208/0x500 driver_register+0x132/0x460 do_one_initcall+0x89/0x300 kernel_init_freeable+0x40d/0x720 kernel_init+0x1a/0x150 ret_from_fork+0x10c/0x1a0 ret_from_fork_asm+0x1a/0x30 Freed by task 136: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3a/0x60 __kasan_slab_free+0x3f/0x50 kfree+0x137/0x370 otx2_ptp_destroy+0x38/0x80 otx2_remove+0x10d/0x4c0 pci_device_remove+0xa6/0x1d0 device_release_driver_internal+0xf8/0x210 pci_stop_bus_device+0x105/0x150 pci_stop_and_remove_bus_device_locked+0x15/0x30 remove_store+0xcc/0xe0 kernfs_fop_write_iter+0x2c3/0x440 vfs_write+0x871/0xd70 ksys_write+0xee/0x1c0 do_syscall_64+0xac/0x280 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... Replace cancel_delayed_work() with cancel_delayed_work_sync() to ensure that the delayed work item is properly canceled before the otx2_ptp is deallocated. This bug was initially identified through static analysis. To reproduce and test it, I simulated the OcteonTX2 PCI device in QEMU and introduced artificial delays within the otx2_sync_tstamp() function to increase the likelihood of triggering the bug.

CVE ID	Description
CVE-2025-39945	In the Linux kernel, the following vulnerability has been resolved: cnic: Fix use-after-free bugs in cnic_delete_task The original code uses cancel_delayed_work() in cnic_cm_stop_bnx2x_hw(), which does not guarantee that the delayed work item 'delete_task' has fully completed if it was already running. Additionally, the delayed work item is cyclic, the flush_workqueue() in cnic_cm_stop_bnx2x_hw() only blocks and waits for work items that were already queued to the workqueue prior to its invocation. Any work items submitted after flush_workqueue() is called are not included in the set of tasks that the flush operation awaits. This means that after the cyclic work items have finished executing, a delayed work item may still exist in the workqueue. This leads to use-after-free scenarios where the cnic_dev is deallocated by cnic_free_dev(), while delete_task remains active and attempt to dereference cnic_dev in cnic_delete_task(). A typical race condition is illustrated below: CPU 0 (cleanup) CPU 1 (delayed work callback) cnic_netdev_event() cnic_stop_hw() cnic_delete_task() cnic_cm_stop_bnx2x_hw() ... cancel_delayed_work() /* the queue_delayed_work() flush_workqueue() executes after flush_workqueue() */ queue_delayed_work() cnic_free_dev(dev)//free cnic_delete_task() //new instance dev = cp->dev; //use Replace cancel_delayed_work() with cancel_delayed_work_sync() to ensure that the cyclic delayed work item is properly canceled and that any ongoing execution of the work item completes before the cnic_dev is deallocated. Furthermore, since cancel_delayed_work_sync() uses __flush_work(work, true) to synchronously wait for any currently executing instance of the work item to finish, the flush_workqueue() becomes redundant and should be removed. This bug was identified through static analysis. To reproduce the issue and validate the fix, I simulated the cnic PCI device in QEMU and introduced intentional delays — such as inserting calls to ssleep() within the cnic_delete_task() function — to increase the likelihood of triggering the bug.
CVE-2025-39946	In the Linux kernel, the following vulnerability has been resolved: tls: make sure to abort the stream if headers are bogus Normally we wait for the socket to buffer up the whole record before we service it. If the socket has a tiny buffer, however, we read out the data sooner, to prevent connection stalls. Make sure that we abort the connection when we find out late that the record is actually invalid. Retrying the parsing is fine in itself but since we copy some more data each time before we parse we can overflow the allocated skb space. Constructing a scenario in which we're under pressure without enough data in the socket to parse the length upfront is quite hard. syzbot figured out a way to do this by serving us the header in small OOB sends, and then filling in the recvbuf with a large normal send. Make sure that tls_rx_msg_size() aborts strp, if we reach an invalid record there's really no way to recover.
CVE-2025-39947	In the Linux kernel, the following vulnerability has been resolved: net/mlx5e: Harden uplink netdev access against device unbind The function mlx5_uplink_netdev_get() gets the uplink netdevice pointer from mdev->mlx5e_res.uplink_netdev. However, the netdevice can be removed and its pointer cleared when unbound from the mlx5_core.eth driver. This results in a NULL pointer, causing a kernel panic. BUG: unable to handle page fault for address: 0000000000001300 at RIP: 0010:mlx5e_vport_rep_load+0x22a/0x270 [mlx5_core] Call Trace: <TASK> mlx5_esw_offloads_rep_load+0x68/0xe0 [mlx5_core] esw_offloads_enable+0x593/0x910 [mlx5_core] mlx5_eswitch_enable_locked+0x341/0x420 [mlx5_core] mlx5_devlink_eswitch_mode_set+0x17e/0x3a0 [mlx5_core] devlink_nl_eswitch_set_doit+0x60/0xd0 genl_family_rcv_msg_doit+0xe0/0x130 genl_rcv_msg+0x183/0x290 netlink_rcv_skb+0x4b/0xf0 genl_rcv+0x24/0x40 netlink_unicast+0x255/0x380 netlink_sendmsg+0x1f3/0x420 __sock_sendmsg+0x38/0x60 __sys_sendto+0x119/0x180 do_syscall_64+0x53/0x1d0 entry_SYSCALL_64_after_hwframe+0x4b/0x53 Ensure the pointer is valid before use by checking it for NULL. If it is valid, immediately call netdev_hold() to take a reference, and preventing the netdevice from being freed while it is in use.

CVE ID	Description
CVE-2025-39948	In the Linux kernel, the following vulnerability has been resolved: ice: fix Rx page leak on multi-buffer frames The ice_put_rx_mbuf() function handles calling ice_put_rx_buf() for each buffer in the current frame. This function was introduced as part of handling multi-buffer XDP support in the ice driver. It works by iterating over the buffers from first_desc up to 1 plus the total number of fragments in the frame, cached from before the XDP program was executed. If the hardware posts a descriptor with a size of 0, the logic used in ice_put_rx_mbuf() breaks. Such descriptors get skipped and don't get added as fragments in ice_add_xdp_frag. Since the buffer isn't counted as a fragment, we do not iterate over it in ice_put_rx_mbuf(), and thus we don't call ice_put_rx_buf(). Because we don't call ice_put_rx_buf(), we don't attempt to re-use the page or free it. This leaves a stale page in the ring, as we don't increment next_to_alloc. The ice_reuse_rx_page() assumes that the next_to_alloc has been incremented properly, and that it always points to a buffer with a NULL page. Since this function doesn't check, it will happily recycle a page over the top of the next_to_alloc buffer, losing track of the old page. Note that this leak only occurs for multi-buffer frames. The ice_put_rx_mbuf() function always handles at least one buffer, so a single-buffer frame will always get handled correctly. It is not clear precisely why the hardware hands us descriptors with a size of 0 sometimes, but it happens somewhat regularly with "jumbo frames" used by 9K MTU. To fix ice_put_rx_mbuf(), we need to make sure to call ice_put_rx_buf() on all buffers between first_desc and next_to_clean. Borrow the logic of a similar function in i40e used for this same purpose. Use the same logic also in ice_get_pgcnts(). Instead of iterating over just the number of fragments, use a loop which iterates until the current index reaches to the next_to_clean element just past the current frame. Unlike i40e, the ice_put_rx_mbuf() function does call ice_put_rx_buf() on the last buffer of the frame indicating the end of packet. For non-linear (multi-buffer) frames, we need to take care when adjusting the pagecnt_bias. An XDP program might release fragments from the tail of the frame, in which case that fragment page is already released. Only update the pagecnt_bias for the first descriptor and fragments still remaining post-XDP program. Take care to only access the shared info for fragmented buffers, as this avoids a significant cache miss. The xdp_xmit value only needs to be updated if an XDP program is run, and only once per packet. Drop the xdp_xmit pointer argument from ice_put_rx_mbuf(). Instead, set xdp_xmit in the ice_clean_rx_irq() function directly. This avoids needing to pass the argument and avoids an extra bit-wise OR for each buffer in the frame. Move the increment of the ntc local variable to ensure its updated *before* all calls to ice_get_pgcnts() or ice_put_rx_mbuf(), as the loop logic requires the index of the element just after the current frame. Now that we use an index pointer in the ring to identify the packet, we no longer need to track or cache the number of fragments in the rx_ring.
CVE-2025-39949	In the Linux kernel, the following vulnerability has been resolved: qed: Don't collect too many protection override GRC elements In the protection override dump path, the firmware can return far too many GRC elements, resulting in attempting to write past the end of the previously-kmalloc'ed dump buffer. This will result in a kernel panic with reason: BUG: unable to handle kernel paging request at ADDRESS where "ADDRESS" is just past the end of the protection override dump buffer. The start address of the buffer is: p_hwfn->cdev->dbg_features[DBG_FEATURE_PROTECTION_OVERRIDE].dump_buf and the size of the buffer is buf_size in the same data structure. The panic can be arrived at from either the qede Ethernet driver path: [exception RIP: qed_grc_dump_addr_range+0x108] qed_protection_override_dump at ffffffff02662ed [qed] qed_dbg_protection_override_dump at ffffffff0267792 [qed] qed_dbg_feature at ffffffff026aa8f [qed] qed_dbg_all_data at ffffffff026b211 [qed] qed_fw_fatal_reporter_dump at ffffffff027298a [qed] devlink_health_do_dump at ffffffff82497f61 devlink_health_report at ffffffff8249cf29 qed_report_fatal_error at ffffffff0272baf [qed] qede_sp_task at ffffffff045ed32 [qede] process_one_work at ffffffff81d19783 or the qedf storage driver path: [exception RIP: qed_grc_dump_addr_range+0x108] qed_protection_override_dump at ffffffff068b2ed [qed] qed_dbg_protection_override_dump at ffffffff068c792 [qed] qed_dbg_feature at ffffffff068fa8f [qed] qed_dbg_all_data at ffffffff0690211 [qed] qed_fw_fatal_reporter_dump at ffffffff069798a [qed] devlink_health_do_dump at ffffffff8aa95e51 devlink_health_report at ffffffff8aa9ae19 qed_report_fatal_error at ffffffff0697baf [qed] qed_hw_err_notify at ffffffff06d32d7 [qed] qed_spq_post at ffffffff06b1011 [qed] qed_fcoe_destroy_conn at ffffffff06b2e91 [qed] qedf_cleanup_fcport at ffffffff05e7597 [qedf] qedf_rport_event_handler at ffffffff05e7bf7 [qedf] fc_rport_work at ffffffff02da715 [libfc] process_one_work at ffffffff8a319663 Resolve this by clamping the firmware's return value to the maximum number of legal elements the firmware should return.

CVE ID	Description
CVE-2025-39950	In the Linux kernel, the following vulnerability has been resolved: net/tcp: Fix a NULL pointer dereference when using TCP-AO with TCP_REPAIR A NULL pointer dereference can occur in tcp_ao_finish_connect() during a connect() system call on a socket with a TCP-AO key added and TCP_REPAIR enabled. The function is called with skb being NULL and attempts to dereference it on tcp_hdr(skb)->seq without a prior skb validation. Fix this by checking if skb is NULL before dereferencing it. The commentary is taken from bpf_skops_established(), which is also called in the same flow. Unlike the function being patched, bpf_skops_established() validates the skb before dereferencing it. int main(void){ struct sockaddr_in sockaddr; struct tcp_ao_add tcp_ao; int sk; int one = 1; memset(&sockaddr,"0",sizeof(sockaddr)); memset(&tcp_ao,"0",sizeof(tcp_ao)); sk = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); sockaddr.sin_family = AF_INET; memcpy(tcp_ao.alg_name,"cmac(aes128)",12); memcpy(tcp_ao.key,"ABCDEFGHABCDEFGH",16); tcp_ao.keylen = 16; memcpy(&tcp_ao.addr,&sockaddr,sizeof(sockaddr)); setsockopt(sk, IPPROTO_TCP, TCP_AO_ADD_KEY, &tcp_ao, sizeof(tcp_ao)); setsockopt(sk, IPPROTO_TCP, TCP_REPAIR, &one, sizeof(one)); sockaddr.sin_family = AF_INET; sockaddr.sin_port = htons(123); inet_aton("127.0.0.1", &sockaddr.sin_addr); connect(sk,(struct sockaddr *)&sockaddr,sizeof(sockaddr)); return 0; } \$ gcc tcp-ao-nullptr.c -o tcp-ao-nullptr -Wall \$ unshare -Urn BUG: kernel NULL pointer dereference, address: 00000000000000b6 PGD 1f648d067 P4D 1f648d067 PUD 1982e8067 PMD 0 Oops: Oops: 0000 [#1] SMP NOPTI Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020 RIP: 0010:tcp_ao_finish_connect (net/ipv4/tcp_ao.c:1182)
CVE-2025-39951	In the Linux kernel, the following vulnerability has been resolved: um: virtio_uml: Fix use-after-free after put_device in probe When register_virtio_device() fails in virtio_uml_probe(), the code sets vu_dev->registered = 1 even though the device was not successfully registered. This can lead to use-after-free or other issues.
CVE-2025-39952	In the Linux kernel, the following vulnerability has been resolved: wifi: wilc1000: avoid buffer overflow in WID string configuration Fix the following copy overflow warning identified by Smatch checker. drivers/net/wireless/microchip/wilc1000/wlan_cfg.c:184 wilc_wlan_parse_response_frame() error: '__memcpy()' 'cfg->s[i]->str' copy overflow (512 vs 65537) This patch introduces size check before accessing the memory buffer. The checks are base on the WID type of received data from the firmware. For WID string configuration, the size limit is determined by individual element size in 'struct wilc_cfg_str_vals' that is maintained in 'len' field of 'struct wilc_cfg_str'.
CVE-2025-39953	In the Linux kernel, the following vulnerability has been resolved: cgroup: split cgroup_destroy_wq into 3 workqueues A hung task can occur during [1] LTP cgroup testing when repeatedly mounting/unmounting perf_event and net_prio controllers with systemd.unified_cgroup_hierarchy=1. The hang manifests in cgroup_lock_and_drain_offline() during root destruction. Related case: cgroup_fj_function_perf_event cgroup_fj_function.sh perf_event cgroup_fj_function_net_prio cgroup_fj_function.sh net_prio Call Trace: cgroup_lock_and_drain_offline+0x14c/0x1e8 cgroup_destroy_root+0x3c/0x2c0 css_free_rwork_fn+0x248/0x338 process_one_work+0x16c/0x3b8 worker_thread+0x22c/0x3b0 kthread+0xec/0x100 ret_from_fork+0x10/0x20 Root Cause: CPU0 CPU1 mount perf_event umount net_prio cgroup1_get_tree cgroup_kill_sb rebind_subsystems // root destruction enqueues // cgroup_destroy_wq // kill all perf_event css // one perf_event css A is dying // css A offline enqueues cgroup_destroy_wq // root destruction will be executed first css_free_rwork_fn cgroup_destroy_root cgroup_lock_and_drain_offline // some perf descendants are dying // cgroup_destroy_wq max_active = 1 // waiting for css A to die Problem scenario: 1. CPU0 mounts perf_event (rebind_subsystems) 2. CPU1 unmounts net_prio (cgroup_kill_sb), queuing root destruction work 3. A dying perf_event CSS gets queued for offline after root destruction 4. Root destruction waits for offline completion, but offline work is blocked behind root destruction in cgroup_destroy_wq (max_active=1) Solution: Split cgroup_destroy_wq into three dedicated workqueues: cgroup_offline_wq – Handles CSS offline operations cgroup_release_wq – Manages resource release cgroup_free_wq – Performs final memory deallocation This separation eliminates blocking in the CSS free path while waiting for offline operations to complete. [1] https://github.com/linux-test-project/ltp/blob/master/runtest/controllers

CVE ID	Description
CVE-2025-39955	In the Linux kernel, the following vulnerability has been resolved: tcp: Clear tcp_sk(sk)->fastopen_rsk in tcp_disconnect(). syzbot reported the splat below where a socket had tcp_sk(sk)->fastopen_rsk in the TCP_ESTABLISHED state. [0] syzbot reused the server-side TCP Fast Open socket as a new client before the TFO socket completes 3WHS: 1. accept() 2. connect(AF_UNSPEC) 3. connect() to another destination As of accept(), sk->sk_state is TCP_SYN_RECV, and tcp_disconnect() changes it to TCP_CLOSE and makes connect() possible, which restarts timers. Since tcp_disconnect() forgot to clear tcp_sk(sk)->fastopen_rsk, the retransmit timer triggered the warning and the intended packet was not retransmitted. Let's call reqsk_fastopen_remove() in tcp_disconnect(). [0]: WARNING: CPU: 2 PID: 0 at net/ipv4/tcp_timer.c:542 tcp_retransmit_timer (net/ipv4/tcp_timer.c:542 (discriminator 7)) Modules linked in: CPU: 2 UID: 0 PID: 0 Comm: swapper/2 Not tainted 6.17.0-rc5-g201825fb4278 #62 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.16.3-debian-1.16.3-2 04/01/2014 RIP: 0010:tcp_retransmit_timer (net/ipv4/tcp_timer.c:542 (discriminator 7)) Code: 41 55 41 54 55 53 48 8b af b8 08 00 00 48 89 fb 48 85 ed 0f 84 55 01 00 00 0f b6 47 12 3c 03 74 0c 0f b6 47 12 3c 04 74 04 90 <0f> 0b 90 48 8b 85 c0 00 00 00 48 89 ef 48 8b 40 30 e8 6a 4f 06 3e RSP: 0018:ffffc900002f8d40 EFLAGS: 00010293 RAX: 0000000000000002 RBX: ffff888106911400 RCX: 0000000000000017 RDX: 00000000002517619 RSI: ffffffff83764080 RDI: ffff888106911400 RBP: ffff888106d5c000 R08: 0000000000000001 R09: ffff888106911480 R10: 00000000000000c2 R11: ffff888106911540 R12: ffff888106911540 R13: ffff888106911480 R14: ffff888106911840 R15: ffff888106911840 FS: 0000000000000000(0000) GS:ffff88907b768000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000000000033 CR2: 00007f8044d69d90 CR3: 00000000002c30003 CR4: 00000000000370ef0 Call Trace: <IRQ> tcp_write_timer (net/ipv4/tcp_timer.c:738) call_timer_fn (kernel/time/timer.c:1747) __run_timers (kernel/time/timer.c:1799 kernel/time/timer.c:2372) timer_expire_remote (kernel/time/timer.c:2385 kernel/time/timer.c:2376 kernel/time/timer.c:2135) tmigr_handle_remote_up (kernel/time/timer_migration.c:944 kernel/time/timer_migration.c:1035) __walk_groups.isra.0 (kernel/time/timer_migration.c:533 (discriminator 1)) tmigr_handle_remote (kernel/time/timer_migration.c:1096) handle_softirqs (./arch/x86/include/asm/jump_label.h:36 ./include/trace/events/irq.h:142 kernel/softirq.c:580) irq_exit_rcu (kernel/softirq.c:614 kernel/softirq.c:453 kernel/softirq.c:680 kernel/softirq.c:696) sysvec_apic_timer_interrupt (arch/x86/kernel/apic/apic.c:1050 (discriminator 35)) arch/x86/kernel/apic/apic.c:1050 (discriminator 35)) </IRQ>
CVE-2025-39957	In the Linux kernel, the following vulnerability has been resolved: wifi: mac80211: increase scan_ies_len for S1G Currently the S1G capability element is not taken into account for the scan_ies_len, which leads to a buffer length validation failure in ieee80211_prep_hw_scan() and subsequent WARN in __ieee80211_start_scan(). This prevents hw scanning from functioning. To fix ensure we accommodate for the S1G capability length.
CVE-2025-39958	In the Linux kernel, the following vulnerability has been resolved: iommu/s390: Make attach succeed when the device was surprise removed When a PCI device is removed with surprise hotplug, there may still be attempts to attach the device to the default domain as part of tear down via (__iommu_release_dma_ownership()), or because the removal happens during probe (__iommu_probe_device()). In both cases zpci_register_ioat() fails with a cc value indicating that the device handle is invalid. This is because the device is no longer part of the instance as far as the hypervisor is concerned. Currently this leads to an error return and s390_iommu_attach_device() fails. This triggers the WARN_ON() in __iommu_group_set_domain_nofail() because attaching to the default domain must never fail. With the device fenced by the hypervisor no DMAs to or from memory are possible and the IOMMU translations have no effect. Proceed as if the registration was successful and let the hotplug event handling clean up the device. This is similar to how devices in the error state are handled since commit 59bbf596791b ("iommu/s390: Make attach succeed even if the device is in error state") except that for removal the domain will not be registered later. This approach was also previously discussed at the link. Handle both cases, error state and removal, in a helper which checks if the error needs to be propagated or ignored. Avoid magic number condition codes by using the pre-existing, but never used, defines for PCI load/store condition codes and rename them to reflect that they apply to all PCI instructions.

CVE ID	Description
CVE-2025-39961	In the Linux kernel, the following vulnerability has been resolved: iommu/amd/pgtbl: Fix possible race while increase page table level The AMD IOMMU host page table implementation supports dynamic page table levels (up to 6 levels), starting with a 3-level configuration that expands based on IOVA address. The kernel maintains a root pointer and current page table level to enable proper page table walks in alloc_pte()/fetch_pte() operations. The IOMMU IOVA allocator initially starts with 32-bit address and once its exhausted it switches to 64-bit address (max address is determined based on IOMMU and device DMA capability). To support larger IOVA, AMD IOMMU driver increases page table level. But in unmap path (iommu_v1_unmap_pages()), fetch_pte() reads pgtable->[root/mode] without lock. So its possible that in exteme corner case, when increase_address_space() is updating pgtable->[root/mode], fetch_pte() reads wrong page table level (pgtable->mode). It does compare the value with level encoded in page table and returns NULL. This will result is iommu_unmap ops to fail and upper layer may retry/log WARN_ON. CPU 0 CPU 1 ----- ----- map pages unmap pages alloc_pte() -> increase_address_space() iommu_v1_unmap_pages() -> fetch_pte() pgtable->root = pte (new root value) READ pgtable->[mode/root] Reads new root, old mode Updates mode (pgtable->mode += 1) Since Page table level updates are infrequent and already synchronized with a spinlock, implement seqcount to enable lock-free read operations on the read path.
CVE-2025-39964	In the Linux kernel, the following vulnerability has been resolved: crypto: af_alg - Disallow concurrent writes in af_alg_sendmsg Issuing two writes to the same af_alg socket is bogus as the data will be interleaved in an unpredictable fashion. Furthermore, concurrent writes may create inconsistencies in the internal socket state. Disallow this by adding a new ctx->write field that indicates exclusive ownership for writing.
CVE-2025-39965	In the Linux kernel, the following vulnerability has been resolved: xfrm: xfrm_alloc_spi shouldn't use 0 as SPI x->id.spi == 0 means "no SPI assigned", but since commit 94f39804d891 ("xfrm: Duplicate SPI Handling"), we now create states and add them to the byspi list with this value. __xfrm_state_delete doesn't remove those states from the byspi list, since they shouldn't be there, and this shows up as a UAF the next time we go through the byspi list.
CVE-2025-39967	In the Linux kernel, the following vulnerability has been resolved: fbcon: fix integer overflow in fbcon_do_set_font Fix integer overflow vulnerabilities in fbcon_do_set_font() where font size calculations could overflow when handling user-controlled font parameters. The vulnerabilities occur when: 1. CALC_FONTSZ(h, pitch, charcount) performs h * pith * charcount multiplication with user-controlled values that can overflow. 2. FONT_EXTRA_WORDS * sizeof(int) + size addition can also overflow 3. This results in smaller allocations than expected, leading to buffer overflows during font data copying. Add explicit overflow checking using check_mul_overflow() and check_add_overflow() kernel helpers to safety validate all size calculations before allocation.
CVE-2025-39968	In the Linux kernel, the following vulnerability has been resolved: i40e: add max boundary check for VF filters There is no check for max filters that VF can request. Add it.
CVE-2025-39969	In the Linux kernel, the following vulnerability has been resolved: i40e: fix validation of VF state in get resources VF state I40E_VF_STATE_ACTIVE is not the only state in which VF is actually active so it should not be used to determine if a VF is allowed to obtain resources. Use I40E_VF_STATE_RESOURCES_LOADED that is set only in i40e_vc_get_vf_resources_msg() and cleared during reset.
CVE-2025-39970	In the Linux kernel, the following vulnerability has been resolved: i40e: fix input validation logic for action_meta Fix condition to check 'greater or equal' to prevent OOB dereference.
CVE-2025-39971	In the Linux kernel, the following vulnerability has been resolved: i40e: fix idx validation in config queues msg Ensure idx is within range of active/initialized TCs when iterating over vf->ch[idx] in i40e_vc_config_queues_msg().
CVE-2025-39972	In the Linux kernel, the following vulnerability has been resolved: i40e: fix idx validation in i40e_validate_queue_map Ensure idx is within range of active/initialized TCs when iterating over vf->ch[idx] in i40e_validate_queue_map().
CVE-2025-39973	In the Linux kernel, the following vulnerability has been resolved: i40e: add validation for ring_len param The `ring_len` parameter provided by the virtual function (VF) is assigned directly to the hardware memory context (HMC) without any validation. To address this, introduce an upper boundary check for both Tx and Rx queue lengths. The maximum number of descriptors supported by the hardware is 8k-32. Additionally, enforce alignment constraints: Tx rings must be a multiple of 8, and Rx rings must be a multiple of 32.

CVE ID	Description
CVE-2025-39977	In the Linux kernel, the following vulnerability has been resolved: futex: Prevent use-after-free during requeue-PI syzbot managed to trigger the following race: T1 T2 futex_wait_requeue_pi() futex_do_wait() schedule() futex_requeue() futex_proxy_trylock_atomic() futex_requeue_pi_prepare() requeue_pi_wake_futex() futex_requeue_pi_complete() /* preempt */ /* timeout/ signal wakes T1 * futex_requeue_pi_wakeup_sync() // Q_REQUEUE_PI_LOCKED futex_hash_put() // back to userland, on stack futex_q is garbage /* back */ wake_up_state(q->task, TASK_NORMAL); In this scenario futex_wait_requeue_pi() is able to leave without using futex_q::lock_ptr for synchronization. This can be prevented by reading futex_q::task before updating the futex_q::requeue_state. A reference on the task_struct is not needed because requeue_pi_wake_futex() is invoked with a spinlock_t held which implies a RCU read section. Even if T1 terminates immediately after, the task_struct will remain valid during T2's wake_up_state(). A READ_ONCE on futex_q::task before futex_requeue_pi_complete() is enough because it ensures that the variable is read before the state is updated. Read futex_q::task before updating the requeue state, use it for the following wakeup.
CVE-2025-39978	In the Linux kernel, the following vulnerability has been resolved: ooctontx2-pf: Fix potential use after free in otx2_tc_add_flow() This code calls kfree_rcu(new_node, rcu) and then dereferences "new_node" and then dereferences it on the next line. Two lines later, we take a mutex so I don't think this is an RCU safe region. Re-order it to do the dereferences before queuing up the free.
CVE-2025-39980	In the Linux kernel, the following vulnerability has been resolved: nexthop: Forbid FDB status change while nexthop is in a group The kernel forbids the creation of non-FDB nexthop groups with FDB nexthops: # ip nexthop add id 1 via 192.0.2.1 fdb # ip nexthop add id 2 group 1 Error: Non FDB nexthop group cannot have fdb nexthops. And vice versa: # ip nexthop add id 3 via 192.0.2.2 dev dummy1 # ip nexthop add id 4 group 3 fdb Error: FDB nexthop group can only have fdb nexthops. However, as long as no routes are pointing to a non-FDB nexthop group, the kernel allows changing the type of a nexthop from FDB to non-FDB and vice versa: # ip nexthop add id 5 via 192.0.2.2 dev dummy1 # ip nexthop add id 6 group 5 # ip nexthop replace id 5 via 192.0.2.2 fdb # echo \$? 0 This configuration is invalid and can result in a NPD [1] since FDB nexthops are not associated with a nexthop device: # ip route add 198.51.100.1/32 nhid 6 # ping 198.51.100.1 Fix by preventing nexthop FDB status change while the nexthop is in a group: # ip nexthop add id 7 via 192.0.2.2 dev dummy1 # ip nexthop add id 8 group 7 # ip nexthop replace id 7 via 192.0.2.2 fdb Error: Cannot change nexthop FDB status while in a group. [1] BUG: kernel NULL pointer dereference, address: 00000000000003c0 [...] Oops: Oops: 0000 [#1] SMP CPU: 6 UID: 0 PID: 367 Comm: ping Not tainted 6.17.0-rc6-virtme-gb65678cacc03 #1 PREEMPT(voluntary) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.17.0-4.fc41 04/01/2014 RIP: 0010:fib_lookup_good_nhc+0x1e/0x80 [...] Call Trace: <TASK> fib_table_lookup+0x541/0x650 ip_route_output_key_hash_rcu+0x2ea/0x970 ip_route_output_key_hash+0x55/0x80 __ip4_datagram_connect+0x250/0x330 udp_connect+0x2b/0x60 __sys_connect+0x9c/0xd0 __x64_sys_connect+0x18/0x20 do_syscall_64+0xa4/0x2a0 entry_SYSCALL_64_after_hwframe+0x4b/0x53

CVE ID	Description
CVE-2025-39981	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: MGMT: Fix possible UAFs This attempts to fix possible UAFs caused by struct mgmt_pending being freed while still being processed like in the following trace, in order to fix mgmt_pending_valid is introduce and use to check if the mgmt_pending hasn't been removed from the pending list, on the complete callbacks it is used to check and in addition remove the cmd from the list while holding mgmt_pending_lock to avoid TOCTOU problems since if the cmd is left on the list it can still be accessed and freed. BUG: KASAN: slab-use-after-free in mgmt_add_adv_patterns_monitor_sync+0x35/0x50 net/bluetooth/mgmt.c:5223 Read of size 8 at addr ffff8880709d4dc0 by task kworker/u11:0/55 CPU: 0 UID: 0 PID: 55 Comm: kworker/u11:0 Not tainted 6.16.4 #2 PREEMPT(full) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014 Workqueue: hci0 hci_cmd_sync_work Call Trace: <TASK> dump_stack_lvl+0x189/0x250 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xca/0x240 mm/kasan/report.c:482 kasan_report+0x118/0x150 mm/kasan/report.c:595 mgmt_add_adv_patterns_monitor_sync+0x35/0x50 net/bluetooth/mgmt.c:5223 hci_cmd_sync_work+0x210/0x3a0 net/bluetooth/hci_sync.c:332 process_one_work kernel/workqueue.c:3238 [inline] process_scheduled_works+0xade/0x17b0 kernel/workqueue.c:3321 worker_thread+0x8a0/0xda0 kernel/workqueue.c:3402 kthread+0x711/0x8a0 kernel/kthread.c:464 ret_from_fork+0x3fc/0x770 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 home/kwqcheii/source/fuzzing/kernel/kasan/linux-6.16.4/arch/x86/entry/entry_64.S:245 <TASK> Allocated by task 12210: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x93/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x230/0x3d0 mm/slub.c:4364 kmalloc_noprof include/linux/slab.h:905 [inline] kzalloc_noprof include/linux/slab.h:1039 [inline] mgmt_pending_new+0x65/0x1e0 net/bluetooth/mgmt_util.c:269 mgmt_pending_add+0x35/0x140 net/bluetooth/mgmt_util.c:296 __add_adv_patterns_monitor+0x130/0x200 net/bluetooth/mgmt.c:5247 add_adv_patterns_monitor+0x214/0x360 net/bluetooth/mgmt.c:5364 hci_mgmt_cmd+0x9c9/0xef0 net/bluetooth/hci_sock.c:1719 hci_sock_sendmsg+0x6ca/0xef0 net/bluetooth/hci_sock.c:1839 sock_sendmsg_nosec net/socket.c:714 [inline] __sock_sendmsg+0x219/0x270 net/socket.c:729 sock_write_iter+0x258/0x330 net/socket.c:1133 new_sync_write fs/read_write.c:593 [inline] vfs_write+0x5c9/0xb30 fs/read_write.c:686 ksys_write+0x145/0x250 fs/read_write.c:738 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 12221: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x46/0x50 mm/kasan/generic.c:576 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x62/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slub.c:2381 [inline] slab_free mm/slub.c:4648 [inline] kfree+0x18e/0x440 mm/slub.c:4847 mgmt_pending_free net/bluetooth/mgmt_util.c:311 [inline] mgmt_pending_foreach+0x30d/0x380 net/bluetooth/mgmt_util.c:257 __mgmt_power_off+0x169/0x350 net/bluetooth/mgmt.c:9444 hci_dev_close_sync+0x754/0x1330 net/bluetooth/hci_sync.c:5290 hci_dev_do_close net/bluetooth/hci_core.c:501 [inline] hci_dev_close+0x108/0x200 net/bluetooth/hci_core.c:526 sock_do_ioctl+0xd9/0x300 net/socket.c:1192 sock_ioctl+0x576/0x790 net/socket.c:1313 vfs_ioctl fs/ioctl.c:51 [inline] __do_sys_ioctl fs/ioctl.c:907 [inline] __se_sys_ioctl+0xf9/0x170 fs/ioctl.c:893 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xf ---truncated---

CVE ID	Description
CVE-2025-39982	In the Linux kernel, the following vulnerability has been resolved: Bluetooth: hci_event: Fix UAF in hci_acl_create_conn_sync This fixes the following UFA in hci_acl_create_conn_sync where a connection still pending is command submission (conn->state == BT_OPEN) maybe freed, also since this also can happen with the likes of hci_le_create_conn_sync fix it as well: BUG: KASAN: slab-use-after-free in hci_acl_create_conn_sync+0x5ef/0x790 net/bluetooth/hci_sync.c:6861 Write of size 2 at addr ffff88805ffcc038 by task kworker/u11:2/9541 CPU: 1 UID: 0 PID: 9541 Comm: kworker/u11:2 Not tainted 6.16.0-rc7 #3 PREEMPT(full) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014 Workqueue: hci3 hci_cmd_sync_work Call Trace: <TASK> dump_stack_lvl+0x189/0x250 lib/dump_stack.c:120 print_address_description mm/kasan/report.c:378 [inline] print_report+0xca/0x230 mm/kasan/report.c:480 kasan_report+0x118/0x150 mm/kasan/report.c:593 hci_acl_create_conn_sync+0x5ef/0x790 net/bluetooth/hci_sync.c:6861 hci_cmd_sync_work+0x210/0x3a0 net/bluetooth/hci_sync.c:332 process_one_work kernel/workqueue.c:3238 [inline] process_scheduled_works+0xae1/0x17b0 kernel/workqueue.c:3321 worker_thread+0x8a0/0xda0 kernel/workqueue.c:3402 kthread+0x70e/0x8a0 kernel/kthread.c:464 ret_from_fork+0x3fc/0x770 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 home/kwqcheii/source/fuzzing/kernel/kasan/linux-6.16-rc7/arch/x86/entry/entry_64.S:245 </TASK> Allocated by task 123736: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 poison_kmalloc_redzone mm/kasan/common.c:377 [inline] __kasan_kmalloc+0x93/0xb0 mm/kasan/common.c:394 kasan_kmalloc include/linux/kasan.h:260 [inline] __kmalloc_cache_noprof+0x230/0x3d0 mm/slab.c:4359 kmalloc_noprof include/linux/slab.h:905 [inline] kzalloc_noprof include/linux/slab.h:1039 [inline] __hci_conn_add+0x233/0x1b30 net/bluetooth/hci_conn.c:939 hci_conn_add_unset net/bluetooth/hci_conn.c:1051 [inline] hci_connect_acl+0x16c/0x4e0 net/bluetooth/hci_conn.c:1634 pair_device+0x418/0xa70 net/bluetooth/mgmt.c:3556 hci_mgmt_cmd+0x9c9/0xef0 net/bluetooth/hci_sock.c:1719 hci_sock_sendmsg+0x6ca/0xef0 net/bluetooth/hci_sock.c:1839 sock_sendmsg_nosec net/socket.c:712 [inline] __sock_sendmsg+0x219/0x270 net/socket.c:727 sock_write_iter+0x258/0x330 net/socket.c:1131 new_sync_write fs/read_write.c:593 [inline] vfs_write+0x54b/0xa90 fs/read_write.c:686 ksys_write+0x145/0x250 fs/read_write.c:738 do_syscall_x64 arch/x86/entry/syscall_64.c:63 [inline] do_syscall_64+0xfa/0x3b0 arch/x86/entry/syscall_64.c:94 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 103680: kasan_save_stack mm/kasan/common.c:47 [inline] kasan_save_track+0x3e/0x80 mm/kasan/common.c:68 kasan_save_free_info+0x46/0x50 mm/kasan/generic.c:576 poison_slab_object mm/kasan/common.c:247 [inline] __kasan_slab_free+0x62/0x70 mm/kasan/common.c:264 kasan_slab_free include/linux/kasan.h:233 [inline] slab_free_hook mm/slab.c:2381 [inline] slab_free mm/slab.c:4643 [inline] kfree+0x18e/0x440 mm/slab.c:4842 device_release+0x9c/0x1c0 kobject_cleanup lib/kobject.c:689 [inline] kobject_release lib/kobject.c:720 [inline] kref_put include/linux/kref.h:65 [inline] kobject_put+0x22b/0x480 lib/kobject.c:737 hci_conn_cleanup net/bluetooth/hci_conn.c:175 [inline] hci_conn_del+0x8ff/0xcb0 net/bluetooth/hci_conn.c:1173 hci_conn_complete_evt+0x3c7/0x1040 net/bluetooth/hci_event.c:3199 hci_event_func net/bluetooth/hci_event.c:7477 [inline] hci_event_packet+0x7e0/0x1200 net/bluetooth/hci_event.c:7531 hci_rx_work+0x46a/0xe80 net/bluetooth/hci_core.c:4070 process_one_work kernel/workqueue.c:3238 [inline] process_scheduled_works+0xae1/0x17b0 kernel/workqueue.c:3321 worker_thread+0x8a0/0xda0 kernel/workqueue.c:3402 kthread+0x70e/0x8a0 kernel/kthread.c:464 ret_from_fork+0x3fc/0x770 arch/x86/kernel/process.c:148 ret_from_fork_asm+0x1a/0x30 home/kwqcheii/sour ---truncated---
CVE-2025-39985	In the Linux kernel, the following vulnerability has been resolved: can: mcba_usb: populate ndo_change_mtu() to prevent buffer overflow Sending an PF_PACKET allows to bypass the CAN framework logic and to directly reach the xmit() function of a CAN driver. The only check which is performed by the PF_PACKET framework is to make sure that skb->len fits the interface's MTU. Unfortunately, because the mcba_usb driver does not populate its net_device_ops->ndo_change_mtu(), it is possible for an attacker to configure an invalid MTU by doing: \$ ip link set can0 mtu 9999 After doing so, the attacker could open a PF_PACKET socket using the ETH_P_CANXL protocol: socket(PF_PACKET, SOCK_RAW, htons(ETH_P_CANXL)) to inject a malicious CAN XL frames. For example: struct canxl_frame frame = { .flags = 0xff, .len = 2048, }; The CAN drivers' xmit() function are calling can_dev_dropped_skb() to check that the skb is valid, unfortunately under above conditions, the malicious packet is able to go through can_dev_dropped_skb() checks: 1. the skb->protocol is set to ETH_P_CANXL which is valid (the function does not check the actual device capabilities). 2. the length is a valid CAN XL length. And so, mcba_usb_start_xmit() receives a CAN XL frame which it is not able to correctly handle and will thus misinterpret it as a CAN frame. This can result in a buffer overflow. The driver will consume cf->len as-is with no further checks on these lines: usb_msg.dlc = cf->len; memcpy(usb_msg.data, cf->data, usb_msg.dlc); Here, cf->len corresponds to the flags field of the CAN XL frame. In our previous example, we set canxl_frame->flags to 0xff. Because the maximum expected length is 8, a buffer overflow of 247 bytes occurs! Populate net_device_ops->ndo_change_mtu() to ensure that the interface's MTU can not be set to anything bigger than CAN_MTU. By fixing the root cause, this prevents the buffer overflow.

CVE ID	Description
CVE-2025-39986	In the Linux kernel, the following vulnerability has been resolved: can: sun4i_can: populate <code>ndo_change_mtu()</code> to prevent buffer overflow Sending an <code>PF_PACKET</code> allows to bypass the CAN framework logic and to directly reach the <code>xmit()</code> function of a CAN driver. The only check which is performed by the <code>PF_PACKET</code> framework is to make sure that <code>skb->len</code> fits the interface's MTU. Unfortunately, because the <code>sun4i_can</code> driver does not populate its <code>net_device_ops->ndo_change_mtu()</code>, it is possible for an attacker to configure an invalid MTU by doing, for example: <code>\$ ip link set can0 mtu 9999</code> After doing so, the attacker could open a <code>PF_PACKET</code> socket using the <code>ETH_P_CANXL</code> protocol: <code>socket(PF_PACKET, SOCK_RAW, htons(ETH_P_CANXL))</code> to inject a malicious CAN XL frames. For example: <code>struct canxl_frame frame = { .flags = 0xff, .len = 2048, };</code> The CAN drivers' <code>xmit()</code> function are calling <code>can_dev_dropped_skb()</code> to check that the <code>skb</code> is valid, unfortunately under above conditions, the malicious packet is able to go through <code>can_dev_dropped_skb()</code> checks: 1. the <code>skb->protocol</code> is set to <code>ETH_P_CANXL</code> which is valid (the function does not check the actual device capabilities). 2. the length is a valid CAN XL length. And so, <code>sun4ican_start_xmit()</code> receives a CAN XL frame which it is not able to correctly handle and will thus misinterpret it as a CAN frame. This can result in a buffer overflow. The driver will consume <code>cf->len</code> as-is with no further checks on this line: <code>dlc = cf->len;</code> Here, <code>cf->len</code> corresponds to the flags field of the CAN XL frame. In our previous example, we set <code>canxl_frame->flags</code> to <code>0xff</code>. Because the maximum expected length is 8, a buffer overflow of 247 bytes occurs a couple line below when doing: <code>for (i = 0; i < dlc; i++) writel(cf->data[i], priv->base + (dreg + i * 4));</code> Populate <code>net_device_ops->ndo_change_mtu()</code> to ensure that the interface's MTU can not be set to anything bigger than <code>CAN_MTU</code>. By fixing the root cause, this prevents the buffer overflow.
CVE-2025-39987	In the Linux kernel, the following vulnerability has been resolved: can: hi311x: populate <code>ndo_change_mtu()</code> to prevent buffer overflow Sending an <code>PF_PACKET</code> allows to bypass the CAN framework logic and to directly reach the <code>xmit()</code> function of a CAN driver. The only check which is performed by the <code>PF_PACKET</code> framework is to make sure that <code>skb->len</code> fits the interface's MTU. Unfortunately, because the <code>sun4i_can</code> driver does not populate its <code>net_device_ops->ndo_change_mtu()</code>, it is possible for an attacker to configure an invalid MTU by doing, for example: <code>\$ ip link set can0 mtu 9999</code> After doing so, the attacker could open a <code>PF_PACKET</code> socket using the <code>ETH_P_CANXL</code> protocol: <code>socket(PF_PACKET, SOCK_RAW, htons(ETH_P_CANXL))</code> to inject a malicious CAN XL frames. For example: <code>struct canxl_frame frame = { .flags = 0xff, .len = 2048, };</code> The CAN drivers' <code>xmit()</code> function are calling <code>can_dev_dropped_skb()</code> to check that the <code>skb</code> is valid, unfortunately under above conditions, the malicious packet is able to go through <code>can_dev_dropped_skb()</code> checks: 1. the <code>skb->protocol</code> is set to <code>ETH_P_CANXL</code> which is valid (the function does not check the actual device capabilities). 2. the length is a valid CAN XL length. And so, <code>hi3110_hard_start_xmit()</code> receives a CAN XL frame which it is not able to correctly handle and will thus misinterpret it as a CAN frame. The driver will consume <code>frame->len</code> as-is with no further checks. This can result in a buffer overflow later on in <code>hi3110_hw_tx()</code> on this line: <code>memcpy(buf + HI3110_FIFO_EXT_DATA_OFF, frame->data, frame->len);</code> Here, <code>frame->len</code> corresponds to the flags field of the CAN XL frame. In our previous example, we set <code>canxl_frame->flags</code> to <code>0xff</code>. Because the maximum expected length is 8, a buffer overflow of 247 bytes occurs! Populate <code>net_device_ops->ndo_change_mtu()</code> to ensure that the interface's MTU can not be set to anything bigger than <code>CAN_MTU</code>. By fixing the root cause, this prevents the buffer overflow.
CVE-2025-39988	In the Linux kernel, the following vulnerability has been resolved: can: etas_es58x: populate <code>ndo_change_mtu()</code> to prevent buffer overflow Sending an <code>PF_PACKET</code> allows to bypass the CAN framework logic and to directly reach the <code>xmit()</code> function of a CAN driver. The only check which is performed by the <code>PF_PACKET</code> framework is to make sure that <code>skb->len</code> fits the interface's MTU. Unfortunately, because the <code>etas_es58x</code> driver does not populate its <code>net_device_ops->ndo_change_mtu()</code>, it is possible for an attacker to configure an invalid MTU by doing, for example: <code>\$ ip link set can0 mtu 9999</code> After doing so, the attacker could open a <code>PF_PACKET</code> socket using the <code>ETH_P_CANXL</code> protocol: <code>socket(PF_PACKET, SOCK_RAW, htons(ETH_P_CANXL))</code> to inject a malicious CAN XL frames. For example: <code>struct canxl_frame frame = { .flags = 0xff, .len = 2048, };</code> The CAN drivers' <code>xmit()</code> function are calling <code>can_dev_dropped_skb()</code> to check that the <code>skb</code> is valid, unfortunately under above conditions, the malicious packet is able to go through <code>can_dev_dropped_skb()</code> checks: 1. the <code>skb->protocol</code> is set to <code>ETH_P_CANXL</code> which is valid (the function does not check the actual device capabilities). 2. the length is a valid CAN XL length. And so, <code>es58x_start_xmit()</code> receives a CAN XL frame which it is not able to correctly handle and will thus misinterpret it as a <code>CAN(FD)</code> frame. This can result in a buffer overflow. For example, using the <code>es581.4</code> variant, the frame will be dispatched to <code>es581_4_tx_can_msg()</code>, go through the last check at the beginning of this function: <code>if (can_is_canfd_skb(skb)) return -EMSGSIZE;</code> and reach this line: <code>memcpy(tx_can_msg->data, cf->data, cf->len);</code> Here, <code>cf->len</code> corresponds to the flags field of the CAN XL frame. In our previous example, we set <code>canxl_frame->flags</code> to <code>0xff</code>. Because the maximum expected length is 8, a buffer overflow of 247 bytes occurs! Populate <code>net_device_ops->ndo_change_mtu()</code> to ensure that the interface's MTU can not be set to anything bigger than <code>CAN_MTU</code> or <code>CANFD_MTU</code> (depending on the device capabilities). By fixing the root cause, this prevents the buffer overflow.

CVE ID	Description
CVE-2025-39989	In the Linux kernel, the following vulnerability has been resolved: x86/mce: use is_copy_from_user() to determine copy-from-user context Patch series "mm/hwpoison: Fix regressions in memory failure handling", v4. ## 1. What am I trying to do: This patchset resolves two critical regressions related to memory failure handling that have appeared in the upstream kernel since version 5.17, as compared to 5.10 LTS. - copyin case: poison found in user page while kernel copying from user space - instr case: poison found while instruction fetching in user space ## 2. What is the expected outcome and why - For copyin case: Kernel can recover from poison found where kernel is doing get_user() or copy_from_user() if those places get an error return and the kernel return -EFAULT to the process instead of crashing. More specifily, MCE handler checks the fixup handler type to decide whether an in kernel #MC can be recovered. When EX_TYPE_UACCESS is found, the PC jumps to recovery code specified in _ASM_EXTABLE_FAULT() and return a -EFAULT to user space. - For instr case: If a poison found while instruction fetching in user space, full recovery is possible. User process takes #PF, Linux allocates a new page and fills by reading from storage. ## 3. What actually happens and why - For copyin case: kernel panic since v5.17 Commit 4c132d1d844a ("x86/futex: Remove .fixup usage") introduced a new extable fixup type, EX_TYPE_EFAULT_REG, and later patches updated the extable fixup type for copy-from-user operations, changing it from EX_TYPE_UACCESS to EX_TYPE_EFAULT_REG. It breaks previous EX_TYPE_UACCESS handling when posion found in get_user() or copy_from_user(). - For instr case: user process is killed by a SIGBUS signal due to #CMCI and #MCE race When an uncorrected memory error is consumed there is a race between the CMCI from the memory controller reporting an uncorrected error with a UCNA signature, and the core reporting and SRAR signature machine check when the data is about to be consumed. ### Background: why *UN*corrected errors tied to *C*MCI in Intel platform [1] Prior to Icelake memory controllers reported patrol scrub events that detected a previously unseen uncorrected error in memory by signaling a broadcast machine check with an SRAO (Software Recoverable Action Optional) signature in the machine check bank. This was overkill because it's not an urgent problem that no core is on the verge of consuming that bad data. It's also found that multi SRAO UCE may cause nested MCE interrupts and finally become an IERR. Hence, Intel downgrades the machine check bank signature of patrol scrub from SRAO to UCNA (Uncorrected, No Action required), and signal changed to #CMCI. Just to add to the confusion, Linux does take an action (in uc_decode_notifier()) to try to offline the page despite the UC*NA* signature name. ### Background: why #CMCI and #MCE race when poison is consuming in Intel platform [1] Having decided that CMCI/UCNA is the best action for patrol scrub errors, the memory controller uses it for reads too. But the memory controller is executing asynchronously from the core, and can't tell the difference between a "real" read and a speculative read. So it will do CMCI/UCNA if an error is found in any read. Thus: 1) Core is clever and thinks address A is needed soon, issues a speculative read. 2) Core finds it is going to use address A soon after sending the read request 3) The CMCI from the memory controller is in a race with MCE from the core that will soon try to retire the load from address A. Quite often (because speculation has got better) the CMCI from the memory controller is delivered before the core is committed to the instruction reading address A, so the interrupt is taken, and Linux offlines the page (marking it as poison). ## Why user process is killed for instr case Commit 046545a661af ("mm/hwpoison: fix error page recovered but reported "not ---truncated---
CVE-2025-39990	In the Linux kernel, the following vulnerability has been resolved: bpf: Check the helper function is valid in get_helper_proto kernel test robot reported verifier bug [1] where the helper func pointer could be NULL due to disabled config option. As Alexei suggested we could check on that in get_helper_proto directly. Marking tail_call helper func with BPF_PTR_POISON, because it is unused by design. [1] https://lore.kernel.org/oe-lkp/202507160818.68358831-lkp@intel.com
CVE-2025-39991	In the Linux kernel, the following vulnerability has been resolved: wifi: ath11k: fix NULL dereference in ath11k_qmi_m3_load() If ab->fw.m3_data points to data, then fw pointer remains null. Further, if m3_mem is not allocated, then fw is dereferenced to be passed to ath11k_err function. Replace fw->size by m3_len. Found by Linux Verification Center (linuxtesting.org) with SVACE.

CVE ID	Description
CVE-2025-39992	In the Linux kernel, the following vulnerability has been resolved: mm: swap: check for stable address space before operating on the VMA It is possible to hit a zero entry while traversing the vmas in unuse_mm() called from swapoff path and accessing it causes the OOPS: Unable to handle kernel NULL pointer dereference at virtual address 0000000000000446--> Loading the memory from offset 0x40 on the XA_ZERO_ENTRY as address. Mem abort info: ESR = 0x0000000096000005 EC = 0x25: DABT (current EL), IL = 32 bits SET = 0, FnV = 0 EA = 0, S1PTW = 0 FSC = 0x05: level 1 translation fault The issue is manifested from the below race between the fork() on a process and swapoff: fork(dup_mmap()) swapoff(unuse_mm) ----- 1) Identical mtree is built using __mt_dup(). 2) copy_pte_range()--> copy_nonpresent_pte(): The dst mm is added into the mmlist to be visible to the swapoff operation. 3) Fatal signal is sent to the parent process(which is the current during the fork) thus skip the duplication of the vmas and mark the vma range with XA_ZERO_ENTRY as a marker for this process that helps during exit_mmap(). 4) swapoff is tried on the 'mm' added to the 'mmlist' as part of the 2. 5) unuse_mm(), that iterates through the vma's of this 'mm' will hit the non-NULL zero entry and operating on this zero entry as a vma is resulting into the oops. The proper fix would be around not exposing this partially-valid tree to others when dropping the mmap lock, which is being solved with [1]. A simpler solution would be checking for MMF_UNSTABLE, as it is set if mm_struct is not fully initialized in dup_mmap(). Thanks to Liam/Lorenzo/David for all the suggestions in fixing this issue.
CVE-2025-39993	In the Linux kernel, the following vulnerability has been resolved: media: rc: fix races with imon_disconnect() Syzbot reports a KASAN issue as below: BUG: KASAN: use-after-free in __create_pipe include/linux/usb.h:1945 [inline] BUG: KASAN: use-after-free in send_packet +0xa2d/0xbc0 drivers/media/rc/imon.c:627 Read of size 4 at addr ffff8880256fb000 by task syz-executor314/4465 CPU: 2 PID: 4465 Comm: syz-executor314 Not tainted 6.0.0-rc1-syzkaller #0 Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.14.0-2 04/01/2014 Call Trace: <TASK> __dump_stack lib/dump_stack.c:88 [inline] dump_stack_lvl+0xcd/0x134 lib/dump_stack.c:106 print_address_description mm/kasan/report.c:317 [inline] print_report.cold +0x2ba/0x6e9 mm/kasan/report.c:433 kasan_report+0xb1/0x1e0 mm/kasan/report.c:495 __create_pipe include/linux/usb.h:1945 [inline] send_packet+0xa2d/0xbc0 drivers/media/rc/imon.c:627 vfd_write+0x2d9/0x550 drivers/media/rc/imon.c:991 vfs_write+0x2d7/0xdd0 fs/read_write.c:576 ksys_write+0x127/0x250 fs/read_write.c:631 do_syscall_x64 arch/x86/entry/common.c:50 [inline] do_syscall_64+0x35/0xb0 arch/x86/entry/common.c:80 entry_SYSCALL_64_after_hwframe +0x63/0xcd The iMON driver improperly releases the usb_device reference in imon_disconnect without coordinating with active users of the device. Specifically, the fields usbdev_intf0 and usbdev_intf1 are not protected by the users counter (ictx->users). During probe, imon_init_intf0 or imon_init_intf1 increments the usb_device reference count depending on the interface. However, during disconnect, usb_put_dev is called unconditionally, regardless of actual usage. As a result, if vfd_write or other operations are still in progress after disconnect, this can lead to a use-after-free of the usb_device pointer. Thread 1 vfd_write Thread 2 imon_disconnect ... if usb_put_dev(ictx->usbdev_intf0) else usb_put_dev(ictx->usbdev_intf1) ... while send_packet if pipe = usb_sndintpipe(ictx->usbdev_intf0) UAF else pipe = usb_sndctrlpipe(ictx->usbdev_intf0, 0) UAF Guard access to usbdev_intf0 and usbdev_intf1 after disconnect by checking ictx->disconnected in all writer paths. Add early return with -ENODEV in send_packet(), vfd_write(), lcd_write() and display_open() if the device is no longer present. Set and read ictx->disconnected under ictx->lock to ensure memory synchronization. Acquire the lock in imon_disconnect() before setting the flag to synchronize with any ongoing operations. Ensure writers exit early and safely after disconnect before the USB core proceeds with cleanup. Found by Linux Verification Center (linuxtesting.org) with Syzkaller.
CVE-2025-39994	In the Linux kernel, the following vulnerability has been resolved: media: tuner: xc5000: Fix use-after-free in xc5000_release The original code uses cancel_delayed_work() in xc5000_release(), which does not guarantee that the delayed work item timer_sleep has fully completed if it was already running. This leads to use-after-free scenarios where xc5000_release() may free the xc5000_priv while timer_sleep is still active and attempts to dereference the xc5000_priv. A typical race condition is illustrated below: CPU 0 (release thread) CPU 1 (delayed work callback) xc5000_release() xc5000_do_timer_sleep() cancel_delayed_work() hybrid_tuner_release_state(priv) kfree(priv) priv = container_of() // UAF Replace cancel_delayed_work() with cancel_delayed_work_sync() to ensure that the timer_sleep is properly canceled before the xc5000_priv memory is deallocated. A deadlock concern was considered: xc5000_release() is called in a process context and is not holding any locks that the timer_sleep work item might also need. Therefore, the use of the _sync() variant is safe here. This bug was initially identified through static analysis. [hverkuil: fix typo in Subject: tuner -> tuner]

CVE ID	Description
CVE-2025-39995	In the Linux kernel, the following vulnerability has been resolved: media: i2c: tc358743: Fix use-after-free bugs caused by orphan timer in probe The state->timer is a cyclic timer that schedules work_i2c_poll and delayed_work_enable_hotplug, while rearming itself. Using timer_delete() fails to guarantee the timer isn't still running when destroyed, similarly cancel_delayed_work() cannot ensure delayed_work_enable_hotplug has terminated if already executing. During probe failure after timer initialization, these may continue running as orphans and reference the already-freed tc358743_state object through tc358743_irq_poll_timer. The following is the trace captured by KASAN. BUG: KASAN: slab-use-after-free in __run_timer_base.part.0+0x7d7/0x8c0 Write of size 8 at addr ffff88800ded83c8 by task swapper/1/0 ... Call Trace: <IRQ> dump_stack_lvl+0x55/0x70 print_report+0xcfc/0x610 ? __pfx_sched_balance_find_src_group+0x10/0x10 ? __run_timer_base.part.0+0x7d7/0x8c0 kasan_report+0xb8/0xf0 ? __run_timer_base.part.0+0x7d7/0x8c0 __run_timer_base.part.0+0x7d7/0x8c0 ? rcu_sched_clock_irq+0xb06/0x27d0 ? __pfx__run_timer_base.part.0+0x10/0x10 ? try_to_wake_up+0xb15/0x1960 ? tmigr_update_events+0x280/0x740 ? _raw_spin_lock_irq+0x80/0xe0 ? __pfx__raw_spin_lock_irq+0x10/0x10 tmigr_handle_remote_up+0x603/0x7e0 ? __pfx_tmigr_handle_remote_up+0x10/0x10 ? sched_balance_trigger+0x98/0x9f0 ? sched_tick+0x221/0x5a0 ? _raw_spin_lock_irq+0x80/0xe0 ? __pfx__raw_spin_lock_irq+0x10/0x10 ? tick_nohz_handler+0x339/0x440 ? __pfx_tmigr_handle_remote_up+0x10/0x10 __walk_groups.isra.0+0x42/0x150 tmigr_handle_remote+0x1f4/0x2e0 ? __pfx_tmigr_handle_remote+0x10/0x10 ? ktime_get+0x60/0x140 ? lapic_next_event+0x11/0x20 ? clockevents_program_event+0x1d4/0x2a0 ? hrtimer_interrupt+0x322/0x780 handle_softirqs+0x16a/0x550 irq_exit_rcu+0xaf/0xe0 sysvec_apic_timer_interrupt+0x70/0x80 </IRQ> ... Allocated by task 141: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 __kasan_kmalloc+0x7f/0x90 __kmalloc_node_track_caller_noprof+0x198/0x430 devm_kmalloc+0x7b/0x1e0 tc358743_probe+0xb7/0x610 i2c_device_probe+0x51d/0x880 really_probe+0x1ca/0x5c0 __driver_probe_device+0x248/0x310 driver_probe_device+0x44/0x120 __device_attach_driver+0x174/0x220 bus_for_each_drv+0x100/0x190 __device_attach+0x206/0x370 bus_probe_device+0x123/0x170 device_add+0xd25/0x1470 i2c_new_client_device+0x7a0/0xcd0 do_one_initcall+0x89/0x300 do_init_module+0x29d/0x7f0 load_module+0x4f48/0x69e0 init_module_from_file+0xe4/0x150 idempotent_init_module+0x320/0x670 __x64_sys_finit_module+0xbd/0x120 do_syscall_64+0xac/0x280 entry_SYSCALL_64_after_hwframe+0x77/0x7f Freed by task 141: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3a/0x60 __kasan_slab_free+0x3f/0x50 kfree+0x137/0x370 release_nodes+0xa4/0x100 devres_release_group+0x1b2/0x380 i2c_device_probe+0x694/0x880 really_probe+0x1ca/0x5c0 __driver_probe_device+0x248/0x310 driver_probe_device+0x44/0x120 __device_attach_driver+0x174/0x220 bus_for_each_drv+0x100/0x190 __device_attach+0x206/0x370 bus_probe_device+0x123/0x170 device_add+0xd25/0x1470 i2c_new_client_device+0x7a0/0xcd0 do_one_initcall+0x89/0x300 do_init_module+0x29d/0x7f0 load_module+0x4f48/0x69e0 init_module_from_file+0xe4/0x150 idempotent_init_module+0x320/0x670 __x64_sys_finit_module+0xbd/0x120 do_syscall_64+0xac/0x280 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... Replace timer_delete() with timer_delete_sync() and cancel_delayed_work() with cancel_delayed_work_sync() to ensure proper termination of timer and work items before resource cleanup. This bug was initially identified through static analysis. For reproduction and testing, I created a functional emulation of the tc358743 device via a kernel module and introduced faults through the debugfs interface.

CVE ID	Description
CVE-2025-39996	In the Linux kernel, the following vulnerability has been resolved: media: b2c2: Fix use-after-free causing by irq_check_work in flexcop_pci_remove The original code uses cancel_delayed_work() in flexcop_pci_remove(), which does not guarantee that the delayed work item irq_check_work has fully completed if it was already running. This leads to use-after-free scenarios where flexcop_pci_remove() may free the flexcop_device while irq_check_work is still active and attempts to dereference the device. A typical race condition is illustrated below: CPU 0 (remove) CPU 1 (delayed work callback) flexcop_pci_remove() flexcop_pci_irq_check_work() cancel_delayed_work() flexcop_device_kfree(fc_pci->fc_dev) fc = fc_pci->fc_dev; // UAF This is confirmed by a KASAN report: ===== <pre>BUG: KASAN: slab-use-after-free in __run_timer_base.part.0+0x7d7/0x8c0 Write of size 8 at addr ffff8880093aa8c8 by task bash/135 ... Call Trace: <IRQ> dump_stack_lvl +0x55/0x70 print_report+0xcf/0x610 ? __run_timer_base.part.0+0x7d7/0x8c0 kasan_report +0xb8/0xf0 ? __run_timer_base.part.0+0x7d7/0x8c0 __run_timer_base.part.0+0x7d7/0x8c0 ? __pfx__run_timer_base.part.0+0x10/0x10 ? __pfx_read_tsc+0x10/0x10 ? ktime_get+0x60/0x140 ? lapic_next_event+0x11/0x20 ? clockevents_program_event+0x1d4/0x2a0 run_timer_softirq +0xd1/0x190 handle_softirqs+0x16a/0x550 irq_exit_rcu+0xaf/0xe0 sysvec_apic_timer_interrupt +0x70/0x80 </IRQ> ... Allocated by task 1: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 __kasan_kmalloc+0x7f/0x90 __kmalloc_noprof+0x1be/0x460 flexcop_device_kmalloc+0x54/0xe0 flexcop_pci_probe+0x1f/0x9d0 local_pci_probe+0xdc/0x190 pci_device_probe+0x2fe/0x470 really_probe+0x1ca/0x5c0 __driver_probe_device+0x248/0x310 driver_probe_device+0x44/0x120 __driver_attach+0xd2/0x310 bus_for_each_dev+0xed/0x170 bus_add_driver+0x208/0x500 driver_register+0x132/0x460 do_one_initcall+0x89/0x300 kernel_init_freeable+0x40d/0x720 kernel_init+0x1a/0x150 ret_from_fork+0x10c/0x1a0 ret_from_fork_asm+0x1a/0x30 Freed by task 135: kasan_save_stack+0x24/0x50 kasan_save_track+0x14/0x30 kasan_save_free_info+0x3a/0x60 __kasan_slab_free+0x3f/0x50 kfree+0x137/0x370 flexcop_device_kfree+0x32/0x50 pci_device_remove +0xa6/0x1d0 device_release_driver_internal+0xf8/0x210 pci_stop_bus_device+0x105/0x150 pci_stop_and_remove_bus_device_locked+0x15/0x30 remove_store+0xcc/0xe0 kernfs_fop_write_iter +0x2c3/0x440 vfs_write+0x871/0xd70 ksys_write+0xee/0x1c0 do_syscall_64+0xac/0x280 entry_SYSCALL_64_after_hwframe+0x77/0x7f ... Replace cancel_delayed_work() with cancel_delayed_work_sync() to ensure that the delayed work item is properly canceled and any executing delayed work has finished before the device memory is deallocated. This bug was initially identified through static analysis. To reproduce and test it, I simulated the B2C2 FlexCop PCI device in QEMU and introduced artificial delays within the flexcop_pci_irq_check_work() function to increase the likelihood of triggering the bug.</pre>
CVE-2025-39998	In the Linux kernel, the following vulnerability has been resolved: scsi: target: target_core_configs: Add length check to avoid buffer overflow A buffer overflow arises from the usage of snprintf to write into the buffer "buf" in target_lu_gp_members_show function located in /drivers/target/target_core_configs.c. This buffer is allocated with size LU_GROUP_NAME_BUF (256 bytes). snprintf(...) formats multiple strings into buf with the HBA name (hba->hba_group.cg_item), a slash character, a devicename (dev-> dev_group.cg_item) and a newline character, the total formatted string length may exceed the buffer size of 256 bytes. Since snprintf() returns the total number of bytes that would have been written (the length of %s/%sn), this value may exceed the buffer length (256 bytes) passed to memcpy(), this will ultimately cause function memcpy reporting a buffer overflow error. An additional check of the return value of snprintf() can avoid this buffer overflow.

CVE ID	Description
CVE-2025-40000	In the Linux kernel, the following vulnerability has been resolved: wifi: rtw89: fix use-after-free in <code>rtw89_core_tx_kick_off_and_wait()</code> There is a bug observed when <code>rtw89_core_tx_kick_off_and_wait()</code> tries to access already freed <code>skb_data</code>: BUG: KFENCE: use-after-free write in <code>rtw89_core_tx_kick_off_and_wait</code> drivers/net/wireless/realtek/rtw89/core.c:1110 CPU: 6 UID: 0 PID: 41377 Comm: kworker/u64:24 Not tainted 6.17.0-rc1+ #1 PREEMPT(lazy) Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS edk2-20250523-14.fc42 05/23/2025 Workqueue: events_unbound cfg80211_wiphy_work [cfg80211] Use-after-free write at 0x0000000020309d9d (in kfence-#251): <code>rtw89_core_tx_kick_off_and_wait</code> drivers/net/wireless/realtek/rtw89/core.c:1110 <code>rtw89_core_scan_complete</code> drivers/net/wireless/realtek/rtw89/core.c:5338 <code>rtw89_hw_scan_complete_cb</code> drivers/net/wireless/realtek/rtw89/fw.c:7979 <code>rtw89_chancx_proceed_cb</code> drivers/net/wireless/realtek/rtw89/chan.c:3165 <code>rtw89_chancx_proceed</code> drivers/net/wireless/realtek/rtw89/chan.h:141 <code>rtw89_hw_scan_complete</code> drivers/net/wireless/realtek/rtw89/fw.c:8012 <code>rtw89_mac_c2h_scanofld_rsp</code> drivers/net/wireless/realtek/rtw89/mac.c:5059 <code>rtw89_fw_c2h_work</code> drivers/net/wireless/realtek/rtw89/fw.c:6758 <code>process_one_work</code> kernel/workqueue.c:3241 <code>worker_thread</code> kernel/workqueue.c:3400 <code>kthread</code> kernel/kthread.c:463 <code>ret_from_fork</code> arch/x86/kernel/process.c:154 <code>ret_from_fork_asm</code> arch/x86/entry/entry_64.S:258 kfence-#251: 0x0000000056e2393d-0x000000009943cb62, size=232, cache=skbuff_head_cache allocated by task 41377 on cpu 6 at 77869.159548s (0.009551s ago): <code>__alloc_skb</code> net/core/skbuff.c:659 <code>__netdev_alloc_skb</code> net/core/skbuff.c:734 <code>ieee80211_nullfunc_get</code> net/mac80211/tx.c:5844 <code>rtw89_core_send_nullfunc</code> drivers/net/wireless/realtek/rtw89/core.c:3431 <code>rtw89_core_scan_complete</code> drivers/net/wireless/realtek/rtw89/core.c:5338 <code>rtw89_hw_scan_complete_cb</code> drivers/net/wireless/realtek/rtw89/fw.c:7979 <code>rtw89_chancx_proceed_cb</code> drivers/net/wireless/realtek/rtw89/chan.c:3165 <code>rtw89_chancx_proceed</code> drivers/net/wireless/realtek/rtw89/chan.c:3194 <code>rtw89_hw_scan_complete</code> drivers/net/wireless/realtek/rtw89/fw.c:8012 <code>rtw89_mac_c2h_scanofld_rsp</code> drivers/net/wireless/realtek/rtw89/mac.c:5059 <code>rtw89_fw_c2h_work</code> drivers/net/wireless/realtek/rtw89/fw.c:6758 <code>process_one_work</code> kernel/workqueue.c:3241 <code>worker_thread</code> kernel/workqueue.c:3400 <code>kthread</code> kernel/kthread.c:463 <code>ret_from_fork</code> arch/x86/kernel/process.c:154 <code>ret_from_fork_asm</code> arch/x86/entry/entry_64.S:258 freed by task 1045 on cpu 9 at 77869.168393s (0.001557s ago): <code>ieee80211_tx_status_skb</code> net/mac80211/status.c:1117 <code>rtw89_pci_release_txwd_skb</code> drivers/net/wireless/realtek/rtw89/pci.c:564 <code>rtw89_pci_release_tx_skbs</code> .isra.0 drivers/net/wireless/realtek/rtw89/pci.c:651 <code>rtw89_pci_release_tx</code> drivers/net/wireless/realtek/rtw89/pci.c:676 <code>rtw89_pci_napi_poll</code> drivers/net/wireless/realtek/rtw89/pci.c:4238 <code>__napi_poll</code> net/core/dev.c:7495 <code>net_rx_action</code> net/core/dev.c:7557 <code>net/core/dev.c:7684</code> <code>handle_softirqs</code> kernel/softirq.c:580 <code>do_softirq.part.0</code> kernel/softirq.c:480 <code>__local_bh_enable_ip</code> kernel/softirq.c:407 <code>rtw89_pci_interrupt_threadfn</code> drivers/net/wireless/realtek/rtw89/pci.c:927 <code>irq_thread_fn</code> kernel/irq/manage.c:1133 <code>irq_thread</code> kernel/irq/manage.c:1257 <code>kthread</code> kernel/kthread.c:463 <code>ret_from_fork</code> arch/x86/kernel/process.c:154 <code>ret_from_fork_asm</code> arch/x86/entry/entry_64.S:258 It is a consequence of a race between the waiting and the signaling side of the completion: Waiting thread Completing thread <code>rtw89_core_tx_kick_off_and_wait()</code> <code>rcu_assign_pointer(skb_data->wait, wait) /* start waiting */ wait_for_completion_timeout()</code> <code>rtw89_pci_tx_status()</code> <code>rtw89_core_tx_wait_complete()</code> <code>rcu_read_lock()</code> /* signals completion and ---truncated---
CVE-2025-40001	In the Linux kernel, the following vulnerability has been resolved: scsi: mvsas: Fix use-after-free bugs in <code>mvs_work_queue</code> During the detaching of Marvell's SAS/SATA controller, the original code calls <code>cancel_delayed_work()</code> in <code>mvs_free()</code> to cancel the delayed work item <code>mwq->work_q</code>. However, if <code>mwq->work_q</code> is already running, the <code>cancel_delayed_work()</code> may fail to cancel it. This can lead to use-after-free scenarios where <code>mvs_free()</code> frees the <code>mvs_info</code> while <code>mvs_work_queue()</code> is still executing and attempts to access the already-freed <code>mvs_info</code>. A typical race condition is illustrated below: CPU 0 (remove) CPU 1 (delayed work callback) <code>mvs_pci_remove()</code> <code>mvs_free()</code> <code>mvs_work_queue()</code> <code>cancel_delayed_work()</code> <code>kfree(mvi)</code> <code>mvi-> // UAF Replace <code>cancel_delayed_work()</code> with <code>cancel_delayed_work_sync()</code> to ensure that the delayed work item is properly canceled and any executing delayed work item completes before the <code>mvs_info</code> is deallocated. This bug was found by static analysis.</code>

CVE ID	Description
CVE-2025-40003	In the Linux kernel, the following vulnerability has been resolved: net: msc: ocelot: Fix use-after-free caused by cyclic delayed work The origin code calls cancel_delayed_work() in ocelot_stats_deinit() to cancel the cyclic delayed work item ocelot->stats_work. However, cancel_delayed_work() may fail to cancel the work item if it is already executing. While destroy_workqueue() does wait for all pending work items in the work queue to complete before destroying the work queue, it cannot prevent the delayed work item from being rescheduled within the ocelot_check_stats_work() function. This limitation exists because the delayed work item is only enqueued into the work queue after its timer expires. Before the timer expiration, destroy_workqueue() has no visibility of this pending work item. Once the work queue appears empty, destroy_workqueue() proceeds with destruction. When the timer eventually expires, the delayed work item gets queued again, leading to the following warning: workqueue: cannot queue ocelot_check_stats_work on wq ocelot-switch-stats WARNING: CPU: 2 PID: 0 at kernel/workqueue.c:2255 __queue_work +0x875/0xaf0 ... RIP: 0010: __queue_work+0x875/0xaf0 ... RSP: 0018:ffff88806d108b10 EFLAGS: 00010086 RAX: 0000000000000000 RBX: 0000000000000101 RCX: 0000000000000027 RDX: 0000000000000027 RSI: 0000000000000004 RDI: ffff88806d123e88 RBP: ffffffff813c3170 R08: 0000000000000000 R09: ffffd100da247d2 R10: ffffd100da247d1 R11: ffff88806d123e8b R12: ffff88800c00f000 R13: ffff88800d7285c0 R14: ffff88806d0a5580 R15: ffff88800d7285a0 FS: 0000000000000000(0000) GS:ffff8880e5725000(0000) knlGS:0000000000000000 CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 CR2: 00007fe18e45ea10 CR3: 0000000005e6c000 CR4: 00000000000000f0 Call Trace: <IRQ> ? kasan_report+0xc6/0xf0 ? __pfx_delayed_work_timer_fn+0x10/0x10 ? __pfx_delayed_work_timer_fn+0x10/0x10 call_timer_fn+0x25/0x1c0 __run_timer_base.part.0+0x3be/0x8c0 ? __pfx_delayed_work_timer_fn+0x10/0x10 ? rcu_sched_clock_irq+0xb06/0x27d0 ? __pfx__run_timer_base.part.0+0x10/0x10 ? try_to_wake_up+0xb15/0x1960 ? _raw_spin_lock_irq+0x80/0xe0 ? __pfx_raw_spin_lock_irq+0x10/0x10 tmigr_handle_remote_up+0x603/0x7e0 ? __pfx_tmigr_handle_remote_up+0x10/0x10 ? sched_balance_trigger+0x1c0/0x9f0 ? sched_tick+0x221/0x5a0 ? _raw_spin_lock_irq+0x80/0xe0 ? __pfx_raw_spin_lock_irq+0x10/0x10 ? tick_nohz_handler+0x339/0x440 ? __pfx_tmigr_handle_remote_up+0x10/0x10 __walk_groups.isra.0+0x42/0x150 tmigr_handle_remote+0x1f4/0x2e0 ? __pfx_tmigr_handle_remote+0x10/0x10 ? ktime_get+0x60/0x140 ? lapic_next_event+0x11/0x20 ? clockevents_program_event+0x1d4/0x2a0 ? hrtimer_interrupt+0x322/0x780 handle_softirqs+0x16a/0x550 irq_exit_rcu+0xaf/0xe0 sysvec_apic_timer_interrupt+0x70/0x80 </IRQ> ... The following diagram reveals the cause of the above warning: CPU 0 (remove) CPU 1 (delayed work callback) msc_ocelot_remove() ocelot_deinit() ocelot_check_stats_work() ocelot_stats_deinit() cancel_delayed_work() ... queue_delayed_work() destroy_workqueue() (wait a time) __queue_work() //UAF The above scenario actually constitutes a UAF vulnerability. The ocelot_stats_deinit() is only invoked when initialization failure or resource destruction, so we must ensure that any delayed work items cannot be rescheduled. Replace cancel_delayed_work() with disable_delayed_work_sync() to guarantee proper cancellation of the delayed work item and ensure completion of any currently executing work before the workqueue is deallocated. A deadlock concern was considered: ocelot_stats_deinit() is called in a process context and is not holding any locks that the delayed work item might also need. Therefore, the use of the _sync() variant is safe here. This bug was identified through static analysis. To reproduce the issue and validate the fix, I simulated ocelot-swit --- truncated---
CVE-2025-40005	In the Linux kernel, the following vulnerability has been resolved: spi: cadence-quadsapi: Implement refcount to handle unbind during busy driver support indirect read and indirect write operation with assumption no force device removal(unbind) operation. However force device removal(removal) is still available to root superuser. Unbinding driver during operation causes kernel crash. This changes ensure driver able to handle such operation for indirect read and indirect write by implementing refcount to track attached devices to the controller and gracefully wait and until attached devices remove operation completed before proceed with removal operation.

CVE ID	Description
CVE-2025-40006	In the Linux kernel, the following vulnerability has been resolved: mm/hugetlb: fix folio is still mapped when deleted Migration may be raced with fallocation hole. remove_inode_single_folio will unmap the folio if the folio is still mapped. However, it's called without folio lock. If the folio is migrated and the mapped pte has been converted to migration entry, folio_mapped() returns false, and won't unmap it. Due to extra refcount held by remove_inode_single_folio, migration fails, restores migration entry to normal pte, and the folio is mapped again. As a result, we triggered BUG in filemap_unaccount_folio. The log is as follows: BUG: Bad page cache in process hugetlb pfn:156c00 page: refcount:515 mapcount:0 mapping:0000000099fef6e1 index:0x0 pfn:0x156c00 head: order:9 mapcount:1 entire_mapcount:1 nr_pages_mapped:0 pincount:0 aops:hugetlbfs_aops ino:ccc dentry name(?):"my_hugepage_file" flags: 0x17ffffc00000c1(locked waiters head node=0 zone=2 lastcpupid=0x1fffff) page_type: f4(hugetlb) page dumped because: still mapped when deleted CPU: 1 UID: 0 PID: 395 Comm: hugetlb Not tainted 6.17.0-rc5-00044-g7aac71907bde-dirty #484 NONE Hardware name: QEMU Ubuntu 24.04 PC (i440FX + PIIX, 1996), BIOS 0.0.0 02/06/2015 Call Trace: <TASK> dump_stack_lvl+0x4f/0x70 filemap_unaccount_folio+0xc4/0x1c0 __filemap_remove_folio+0x38/0x1c0 filemap_remove_folio+0x41/0xd0 remove_inode_hugepages+0x142/0x250 hugetlbfs_fallocate+0x471/0x5a0 vfs_fallocate+0x149/0x380 Hold folio lock before checking if the folio is mapped to avoid race with migration.
CVE-2025-40008	In the Linux kernel, the following vulnerability has been resolved: kmsan: fix out-of-bounds access to shadow memory Running sha224_kunit on a KMSAN-enabled kernel results in a crash in kmsan_internal_set_shadow_origin(): BUG: unable to handle page fault for address: fffffb3840291000 #PF: supervisor read access in kernel mode #PF: error_code(0x0000) - not-present page PGD 1810067 P4D 1810067 PUD 192d067 PMD 3c17067 PTE 0 Oops: 0000 [#1] SMP NOPTI CPU: 0 UID: 0 PID: 81 Comm: kunit_try_catch Tainted: G N 6.17.0-rc3 #10 PREEMPT(voluntary) Tainted: [N]=TEST Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.17.0-0-gb52ca86e094d-prebuilt.qemu.org 04/01/2014 RIP: 0010:kmsan_internal_set_shadow_origin+0x91/0x100 [...] Call Trace: <TASK> __msan_memset+0xee/0x1a0 sha224_final+0x9e/0x350 test_hash_buffer_overruns+0x46f/0x5f0 ? kmsan_get_shadow_origin_ptr+0x46/0xa0 ? __pfx_test_hash_buffer_overruns+0x10/0x10 kunit_try_run_case+0x198/0xa00 This occurs when memset() is called on a buffer that is not 4-byte aligned and extends to the end of a guard page, i.e. the next page is unmapped. The bug is that the loop at the end of kmsan_internal_set_shadow_origin() accesses the wrong shadow memory bytes when the address is not 4-byte aligned. Since each 4 bytes are associated with an origin, it rounds the address and size so that it can access all the origins that contain the buffer. However, when it checks the corresponding shadow bytes for a particular origin, it incorrectly uses the original unrounded shadow address. This results in reads from shadow memory beyond the end of the buffer's shadow memory, which crashes when that memory is not mapped. To fix this, correctly align the shadow address before accessing the 4 shadow bytes corresponding to each origin.
CVE-2025-40009	In the Linux kernel, the following vulnerability has been resolved: fs/proc/task_mmu: check p->vec_buf for NULL When the PAGEMAP_SCAN ioctl is invoked with vec_len = 0 reaches pagemap_scan_backout_range(), kernel panics with null-ptr-deref: [44.936808] Oops: general protection fault, probably for non-canonical address 0xdffffc0000000000: 0000 [#1] SMP DEBUG_PAGEALLOC KASAN NOPTI [44.937797] KASAN: null-ptr-deref in range [0x0000000000000000-0x0000000000000000] [44.938391] CPU: 1 UID: 0 PID: 2480 Comm: reproducer Not tainted 6.17.0-rc6 #22 PREEMPT(none) [44.939062] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org 04/01/2014 [44.939935] RIP: 0010:pagemap_scan_thp_entry.isra.0+0x741/0xa80 <snip registers, unreliable trace> [44.946828] Call Trace: [44.947030] <TASK> [44.949219] pagemap_scan_pmd_entry+0xec/0xfa0 [44.952593] walk_pmd_range.isra.0+0x302/0x910 [44.954069] walk_pud_range.isra.0+0x419/0x790 [44.954427] walk_p4d_range+0x41e/0x620 [44.954743] walk_pgd_range+0x31e/0x630 [44.955057] __walk_page_range+0x160/0x670 [44.956883] walk_page_range_mm+0x408/0x980 [44.958677] walk_page_range+0x66/0x90 [44.958984] do_pagemap_scan+0x28d/0x9c0 [44.961833] do_pagemap_cmd+0x59/0x80 [44.962484] __x64_sys_ioctl+0x18d/0x210 [44.962804] do_syscall_64+0x5b/0x290 [44.963111] entry_SYSCALL_64_after_hwframe+0x76/0x7e vec_len = 0 in pagemap_scan_init_bounce_buffer() means no buffers are allocated and p->vec_buf remains set to NULL. This breaks an assumption made later in pagemap_scan_backout_range(), that page_region is always allocated for p->vec_buf_index. Fix it by explicitly checking p->vec_buf for NULL before dereferencing. Other sites that might run into same deref-issue are already (directly or transitively) protected by checking p->vec_buf. Note: From PAGEMAP_SCAN man page, it seems vec_len = 0 is valid when no output is requested and it's only the side effects caller is interested in, hence it passes check in pagemap_scan_get_args(). This issue was found by syzkaller.
CVE-2025-40010	In the Linux kernel, the following vulnerability has been resolved: afs: Fix potential null pointer dereference in afs_put_server afs_put_server() accessed server->debug_id before the NULL check, which could lead to a null pointer dereference. Move the debug_id assignment, ensuring we never dereference a NULL server pointer.

CVE ID	Description
CVE-2025-40011	In the Linux kernel, the following vulnerability has been resolved: drm/gma500: Fix null dereference in hdmi teardown pci_set_drvdata sets the value of pdev->driver_data to NULL, after which the driver_data obtained from the same dev is dereferenced in oaktrail_hdmi_i2c_exit, and the i2c_dev is extracted from it. To prevent this, swap these calls. Found by Linux Verification Center (linuxtesting.org) with Svacer.
CVE-2025-40012	In the Linux kernel, the following vulnerability has been resolved: net/smc: fix warning in smc_rx_splice() when calling get_page() smc_lo_register_dmb() allocates DMB buffers with kzalloc(), which are later passed to get_page() in smc_rx_splice(). Since kmalloc memory is not page-backed, this triggers WARN_ON_ONCE() in get_page() and prevents holding a refcount on the buffer. This can lead to use-after-free if the memory is released before splice_to_pipe() completes. Use folio_alloc() instead, ensuring DMBs are page-backed and safe for get_page(). WARNING: CPU: 18 PID: 12152 at ./include/linux/mm.h:1330 smc_rx_splice+0xaf8/0xe20 [smc] CPU: 18 UID: 0 PID: 12152 Comm: smcapp Kdump: loaded Not tainted 6.17.0-rc3-11705-g9cf4672ecfee #10 NONE Hardware name: IBM 3931 A01 704 (z/VMM 7.4.0) Krrl PSW : 0704e00180000000 000793161032696c (smc_rx_splice+0xafc/0xe20 [smc]) R:0 T:1 IO:1 EX:1 Key:0 M:1 W:0 P:0 AS:3 CC:2 PM:0 RI:0 EA:3 Krrl GPRS: 0000000000000000 001cee80007d3001 00077400000000f8 0000000000000005 0000000000000001 001cee80007d3006 0007740000001000 001c000000000000 000000009b0c99e0 0000000000001000 001c0000000000f8 001c000000000000 000003ffcc6f7c88 0007740003e98000 0007931600000005 000792969b2ff7b8 Krrl Code: 0007931610326960: af000000 mc 0,0 0007931610326964: a7f4ff43 brc 15,00079316103267ea #0007931610326968: af000000 mc 0,0 >000793161032696c: a7f4ff3f brc 15,00079316103267ea 0007931610326970: e320f1000004 lg %r2,256(%r15) 0007931610326976: c0e53fd1b5f5 brasl %r14,000793168fd5d560 000793161032697c: a7f4fbb5 brc 15,00079316103260e6 0007931610326980: b904002b lgr %r2,%r11 Call Trace: smc_rx_splice+0xafc/0xe20 [smc] smc_rx_splice+0x756/0xe20 [smc] smc_rx_recvmg+0xa74/0xe00 [smc] smc_splice_read+0x1ce/0x3b0 [smc] sock_splice_read+0xa2/0xf0 do_splice_read+0x198/0x240 splice_file_to_pipe+0x7e/0x110 do_splice+0x59e/0xde0 __do_splice+0x11a/0x2d0 __s390x_sys_splice+0x140/0x1f0 __do_syscall+0x122/0x280 system_call+0x6e/0x90 Last Breaking-Event-Address: smc_rx_splice+0x960/0xe20 [smc] ---[end trace 0000000000000000]---
CVE-2025-40013	In the Linux kernel, the following vulnerability has been resolved: ASoC: qcom: audioreach: fix potential null pointer dereference It is possible that the topology parsing function audioreach_widget_load_module_common() could return NULL or an error pointer. Add missing NULL check so that we do not dereference it.
CVE-2025-40014	In the Linux kernel, the following vulnerability has been resolved: objtool, spi: amd: Fix out-of-bounds stack access in amd_set_spi_freq() If speed_hz < AMD_SPI_MIN_HZ, amd_set_spi_freq() iterates over the entire amd_spi_freq array without breaking out early, causing 'i' to go beyond the array bounds. Fix that by stopping the loop when it gets to the last entry, so the low speed_hz value gets clamped up to AMD_SPI_MIN_HZ. Fixes the following warning with an UBSAN kernel: drivers/spi/spi-amd.o: error: objtool: amd_set_spi_freq() falls through to next function amd_spi_set_opcode()

CVE ID	Description
CVE-2025-40016	In the Linux kernel, the following vulnerability has been resolved: media: uvcvideo: Mark invalid entities with id UVC_INVALID_ENTITY_ID Per UVC 1.1+ specification 3.7.2, units and terminals must have a non-zero unique ID. `` Each Unit and Terminal within the video function is assigned a unique identification number, the Unit ID (UID) or Terminal ID (TID), contained in the bUnitID or bTerminalID field of the descriptor. The value 0x00 is reserved for undefined ID, `` If we add a new entity with id 0 or a duplicated ID, it will be marked as UVC_INVALID_ENTITY_ID. In a previous attempt commit 3dd075fe8ebb ("media: uvcvideo: Require entities to have a non-zero unique ID"), we ignored all the invalid units, this broke a lot of non-compatible cameras. Hopefully we are more lucky this time. This also prevents some syzkaller reproducers from triggering warnings due to a chain of entities referring to themselves. In one particular case, an Output Unit is connected to an Input Unit, both with the same ID of 1. But when looking up for the source ID of the Output Unit, that same entity is found instead of the input entity, which leads to such warnings. In another case, a backward chain was considered finished as the source ID was 0. Later on, that entity was found, but its pads were not valid. Here is a sample stack trace for one of those cases. [20.650953] usb 1-1: new high-speed USB device number 2 using dummy_hcd [20.830206] usb 1-1: Using ep0 maxpacket: 8 [20.833501] usb 1-1: config 0 descriptor?? [21.038518] usb 1-1: string descriptor 0 read error: -71 [21.038893] usb 1-1: Found UVC 0.00 device <unnamed> (2833:0201) [21.039299] uvcvideo 1-1:0.0: Entity type for entity Output 1 was not initialized! [21.041583] uvcvideo 1-1:0.0: Entity type for entity Input 1 was not initialized! [21.042218] -----[cut here]----- [21.042536] WARNING: CPU: 0 PID: 9 at drivers/media/mc/mc-entity.c:1147 media_create_pad_link+0x2c4/0x2e0 [21.043195] Modules linked in: [21.043535] CPU: 0 UID: 0 PID: 9 Comm: kworker/0:1 Not tainted 6.11.0-rc7-00030-g3480e43a6ccf #444 [21.044101] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS 1.15.0-1 04/01/2014 [21.044639] Workqueue: usb_hub_wq hub_event [21.045100] RIP: 0010:media_create_pad_link+0x2c4/0x2e0 [21.045508] Code: fe e8 20 01 00 00 b8 f4 ff ff ff 48 83 c4 30 5b 41 5c 41 5d 41 5e 41 5f 5d c3 cc cc cc cc 0f 0b eb e9 0f 0b eb 0a 0f 0b eb 06 <0f> 0b eb 02 0f 0b b8 ea ff ff ff eb d4 66 2e 0f 1f 84 00 00 00 00 [21.046801] RSP: 0018:ffff90000004b318 EFLAGS: 00010246 [21.047227] RAX: ffff888004e5d458 RBX: 0000000000000000 RCX: ffffffff818fccf1 [21.047719] RDX: 000000000000007b RSI: 0000000000000000 RDI: ffff888004313290 [21.048241] RBP: ffff888004313290 R08: 0001ffffffff R09: 0000000000000000 [21.048701] R10: 0000000000000013 R11: 0001888004313290 R12: 0000000000000003 [21.049138] R13: ffff888004313080 R14: ffff888004313080 R15: 0000000000000000 [21.049648] FS: 0000000000000000(0000) GS:ffff88803ec00000(0000) knlGS:0000000000000000 [21.050271] CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033 [21.050688] CR2: 0000592cc27635b0 CR3: 000000000431c000 CR4: 0000000000750ef0 [21.051136] PKRU: 55555554 [21.051331] Call Trace: [21.051480] <TASK> [21.051611] ? __warn+0xc4/0x210 [21.051861] ? media_create_pad_link+0x2c4/0x2e0 [21.052252] ? report_bug+0x11b/0x1a0 [21.052540] ? trace_hardirqs_on+0x31/0x40 [21.052901] ? handle_bug+0x3d/0x70 [21.053197] ? exc_invalid_op+0x1a/0x50 [21.053511] ? asm_exc_invalid_op+0x1a/0x20 [21.053924] ? media_create_pad_link+0x91/0x2e0 [21.054364] ? media_create_pad_link+0x2c4/0x2e0 [21.054834] ? media_create_pad_link+0x91/0x2e0 [21.055131] ? _raw_spin_unlock+0x1e/0x40 [21.055441] ? __v4l2_device_register_subdev+0x202/0x210 [21.055837] uvc_mc_register_entities+0x358/0x400 [21.056144] uvc_register_chains+0x1 --- truncated---
CVE-2025-40114	In the Linux kernel, the following vulnerability has been resolved: iio: light: Add check for array bounds in veml6075_read_int_time_ms The array contains only 5 elements, but the index calculated by veml6075_read_int_time_index can range from 0 to 7, which could lead to out-of-bounds access. The check prevents this issue. Coverity Issue CID 1574309: (#1 of 1): Out-of-bounds read (OVERRUN) overrun-local: Overrunning array veml6075_it_ms of 5 4-byte elements at element index 7 (byte offset 31) using index int_index (which evaluates to 7) This is hardening against potentially broken hardware. Good to have but not necessary to backport.
CVE-2025-40300	In the Linux kernel, the following vulnerability has been resolved: x86/vmscape: Add conditional IBPB mitigation VMSCAPE is a vulnerability that exploits insufficient branch predictor isolation between a guest and a userspace hypervisor (like QEMU). Existing mitigations already protect kernel/KVM from a malicious guest. Userspace can additionally be protected by flushing the branch predictors after a VMexit. Since it is the userspace that consumes the poisoned branch predictors, conditionally issue an IBPB after a VMexit and before returning to userspace. Workloads that frequently switch between hypervisor and userspace will incur the most overhead from the new IBPB. This new IBPB is not integrated with the existing IBPB sites. For instance, a task can use the existing speculation control prctl() to get an IBPB at context switch time. With this implementation, the IBPB is doubled up: one at context switch and another before running userspace. The intent is to integrate and optimize these cases post-embargo. [dhansen: elaborate on suboptimal IBPB solution]
CVE-2025-40325	In the Linux kernel, the following vulnerability has been resolved: md/raid10: wait barrier before returning discard request with REQ_NOWAIT raid10_handle_discard should wait barrier before returning a discard bio which has REQ_NOWAIT. And there is no need to print warning calltrace if a discard bio has REQ_NOWAIT flag. Quality engineer usually checks dmesg and reports error if dmesg has warning/error calltrace.

CVE ID	Description
CVE-2025-40364	In the Linux kernel, the following vulnerability has been resolved: io_uring: fix io_req_prep_async with provided buffers io_req_prep_async() can import provided buffers, commit the ring state by giving up on that before, it'll be reimported later if needed.
CVE-2025-40778	Under certain circumstances, BIND is too lenient when accepting records from answers, allowing an attacker to inject forged data into the cache. This issue affects BIND 9 versions 9.11.0 through 9.16.50, 9.18.0 through 9.18.39, 9.20.0 through 9.20.13, 9.21.0 through 9.21.12, 9.11.3-S1 through 9.16.50-S1, 9.18.11-S1 through 9.18.39-S1, and 9.20.9-S1 through 9.20.13-S1.
CVE-2025-40780	In specific circumstances, due to a weakness in the Pseudo Random Number Generator (PRNG) that is used, it is possible for an attacker to predict the source port and query ID that BIND will use. This issue affects BIND 9 versions 9.16.0 through 9.16.50, 9.18.0 through 9.18.39, 9.20.0 through 9.20.13, 9.21.0 through 9.21.12, 9.16.8-S1 through 9.16.50-S1, 9.18.11-S1 through 9.18.39-S1, and 9.20.9-S1 through 9.20.13-S1.
CVE-2025-40909	Perl threads have a working directory race condition where file operations may target unintended paths. If a directory handle is open at thread creation, the process-wide current working directory is temporarily changed in order to clone that handle for the new thread, which is visible from any third (or more) thread already running. This may lead to unintended operations such as loading code or accessing files from unexpected locations, which a local attacker may be able to exploit. The bug was introduced in commit 11a11ecf4bea72b17d250cfb43c897be1341861e and released in Perl version 5.13.6
CVE-2025-41234	Description In Spring Framework, versions 6.0.x as of 6.0.5, versions 6.1.x and 6.2.x, an application is vulnerable to a reflected file download (RFD) attack when it sets a "Content-Disposition" header with a non-ASCII charset, where the filename attribute is derived from user-supplied input. Specifically, an application is vulnerable when all the following are true: * The header is prepared with org.springframework.http.ContentDisposition. * The filename is set via ContentDisposition.Builder#filename(String, Charset). * The value for the filename is derived from user-supplied input. * The application does not sanitize the user-supplied input. * The downloaded content of the response is injected with malicious commands by the attacker (see RFD paper reference for details). An application is not vulnerable if any of the following is true: * The application does not set a "Content-Disposition" response header. * The header is not prepared with org.springframework.http.ContentDisposition. * The filename is set via one of: * ContentDisposition.Builder#filename(String), or * ContentDisposition.Builder#filename(String, ASCII) * The filename is not derived from user-supplied input. * The filename is derived from user-supplied input but sanitized by the application. * The attacker cannot inject malicious content in the downloaded content of the response. Affected Spring Products and Versions Spring Framework: * 6.2.0 - 6.2.7 * 6.1.0 - 6.1.20 * 6.0.5 - 6.0.28 * Older, unsupported versions are not affected Mitigation Users of affected versions should upgrade to the corresponding fixed version. Affected version(s) Fix version Availability 6.2.x 6.2.8 OSS 6.1.x 6.1.21 OSS 6.0.x 6.0.29 Commercial https://enterprise.spring.io/ No further mitigation steps are necessary. CWE-113 in 'Content-Disposition' handling in VMware Spring Framework versions 6.0.5 to 6.2.7 allows remote attackers to launch Reflected File Download (RFD) attacks via unsanitized user input in 'ContentDisposition.Builder#filename(String, Charset)' with non-ASCII charsets.
CVE-2025-41242	Spring Framework MVC applications can be vulnerable to a "Path Traversal Vulnerability" when deployed on a non-compliant Servlet container. An application can be vulnerable when all the following are true: * the application is deployed as a WAR or with an embedded Servlet container * the Servlet container does not reject suspicious sequences https://jakarta.ee/specifications/servlet/6.1/jakarta-servlet-spec-6.1.html#uri-path-canonicalization * the application serves static resources https://docs.spring.io/spring-framework/reference/web/webmvc/mvc-config/static-resources.html#page-title with Spring resource handling We have verified that applications deployed on Apache Tomcat or Eclipse Jetty are not vulnerable, as long as default security features are not disabled in the configuration. Because we cannot check exploits against all Servlet containers and configuration variants, we strongly recommend upgrading your application.
CVE-2025-41249	The Spring Framework annotation detection mechanism may not correctly resolve annotations on methods within type hierarchies with a parameterized super type with unbounded generics. This can be an issue if such annotations are used for authorization decisions. Your application may be affected by this if you are using Spring Security's @EnableMethodSecurity feature. You are not affected by this if you are not using @EnableMethodSecurity or if you do not use security annotations on methods in generic superclasses or generic interfaces. This CVE is published in conjunction with CVE-2025-41248 https://spring.io/security/cve-2025-41248 .

CVE ID	Description
CVE-2025-4138	Allows the extraction filter to be ignored, allowing symlink targets to point outside the destination directory, and the modification of some file metadata. You are affected by this vulnerability if using the tarfile module to extract untrusted tar archives using <code>TarFile.extractall()</code> or <code>TarFile.extract()</code> using the <code>filter=</code> parameter with a value of "data" or "tar". See the tarfile extraction filters documentation https://docs.python.org/3/library/tarfile.html#tarfile-extraction-filter for more information. Note that for Python 3.14 or later the default value of <code>filter=</code> changed from "no filtering" to "data", so if you are relying on this new default behavior then your usage is also affected. Note that none of these vulnerabilities significantly affect the installation of source distributions which are tar archives as source distributions already allow arbitrary code execution during the build process. However when evaluating source distributions it's important to avoid installing source distributions with suspicious links.
CVE-2025-4207	Buffer over-read in PostgreSQL GB18030 encoding validation allows a database input provider to achieve temporary denial of service on platforms where a 1-byte over-read can elicit process termination. This affects the database server and also libpq. Versions before PostgreSQL 17.5, 16.9, 15.13, 14.18, and 13.21 are affected.
CVE-2025-4330	Allows the extraction filter to be ignored, allowing symlink targets to point outside the destination directory, and the modification of some file metadata. You are affected by this vulnerability if using the tarfile module to extract untrusted tar archives using <code>TarFile.extractall()</code> or <code>TarFile.extract()</code> using the <code>filter=</code> parameter with a value of "data" or "tar". See the tarfile extraction filters documentation https://docs.python.org/3/library/tarfile.html#tarfile-extraction-filter for more information. Note that for Python 3.14 or later the default value of <code>filter=</code> changed from "no filtering" to "data", so if you are relying on this new default behavior then your usage is also affected. Note that none of these vulnerabilities significantly affect the installation of source distributions which are tar archives as source distributions already allow arbitrary code execution during the build process. However when evaluating source distributions it's important to avoid installing source distributions with suspicious links.
CVE-2025-4373	A flaw was found in GLib, which is vulnerable to an integer overflow in the <code>g_string_insert_unichar()</code> function. When the position at which to insert the character is large, the position will overflow, leading to a buffer underwrite.
CVE-2025-43857	Net::IMAP implements Internet Message Access Protocol (IMAP) client functionality in Ruby. Prior to versions 0.5.7, 0.4.20, 0.3.9, and 0.2.5, there is a possibility for denial of service by memory exhaustion when net-imap reads server responses. At any time while the client is connected, a malicious server can send a "literal" byte count, which is automatically read by the client's receiver thread. The response reader immediately allocates memory for the number of bytes indicated by the server response. This should not be an issue when securely connecting to trusted IMAP servers that are well-behaved. It can affect insecure connections and buggy, untrusted, or compromised servers (for example, connecting to a user supplied hostname). This issue has been patched in versions 0.5.7, 0.4.20, 0.3.9, and 0.2.5.
CVE-2025-43859	h11 is a Python implementation of HTTP/1.1. Prior to version 0.16.0, a leniency in h11's parsing of line terminators in chunked-coding message bodies can lead to request smuggling vulnerabilities under certain conditions. This issue has been patched in version 0.16.0. Since exploitation requires the combination of buggy h11 with a buggy (reverse) proxy, fixing either component is sufficient to mitigate this issue.
CVE-2025-4435	When using a <code>TarFile.errorlevel = 0</code> and extracting with a filter the documented behavior is that any filtered members would be skipped and not extracted. However the actual behavior of <code>TarFile.errorlevel = 0</code> in affected versions is that the member would still be extracted and not skipped.
CVE-2025-4516	There is an issue in CPython when using <code>bytes.decode("unicode_escape", error="ignore replace")</code> . If you are not using the "unicode_escape" encoding or an error handler your usage is not affected. To work-around this issue you may stop using the <code>error=</code> handler and instead wrap the <code>bytes.decode()</code> call in a try-except catching the <code>DecodeError</code> .
CVE-2025-4517	Allows arbitrary filesystem writes outside the extraction directory during extraction with <code>filter="data"</code> . You are affected by this vulnerability if using the tarfile module to extract untrusted tar archives using <code>TarFile.extractall()</code> or <code>TarFile.extract()</code> using the <code>filter=</code> parameter with a value of "data" or "tar". See the tarfile extraction filters documentation https://docs.python.org/3/library/tarfile.html#tarfile-extraction-filter for more information. Note that for Python 3.14 or later the default value of <code>filter=</code> changed from "no filtering" to "data", so if you are relying on this new default behavior then your usage is also affected. Note that none of these vulnerabilities significantly affect the installation of source distributions which are tar archives as source distributions already allow arbitrary code execution during the build process. However when evaluating source distributions it's important to avoid installing source distributions with suspicious links.

CVE ID	Description
CVE-2025-45582	GNU Tar through 1.35 allows file overwrite via directory traversal in crafted TAR archives, with a certain two-step process. First, the victim must extract an archive that contains a ../ symlink to a critical directory. Second, the victim must extract an archive that contains a critical file, specified via a relative pathname that begins with the symlink name and ends with that critical file's name. Here, the extraction follows the symlink and overwrites the critical file. This bypasses the protection mechanism of "Member name contains '.'" that would occur for a single TAR archive that attempted to specify the critical file via a ../ approach. For example, the first archive can contain "x -> .././../././home/victim/.ssh" and the second archive can contain x/authorized_keys. This can affect server applications that automatically extract any number of user-supplied TAR archives, and were relying on the blocking of traversal. This can also affect software installation processes in which "tar xf" is run more than once (e.g., when installing a package can automatically install two dependencies that are set up as untrusted tarballs instead of official packages).
CVE-2025-4563	A vulnerability exists in the NodeRestriction admission controller where nodes can bypass dynamic resource allocation authorization checks. When the DynamicResourceAllocation feature gate is enabled, the controller properly validates resource claim statuses during pod status updates but fails to perform equivalent validation during pod creation. This allows a compromised node to create mirror pods that access unauthorized dynamic resources, potentially leading to privilege escalation.
CVE-2025-4565	Any project that uses Protobuf Pure-Python backend to parse untrusted Protocol Buffers data containing an arbitrary number of recursive groups, recursive messages or a series of SGROUP tags can be corrupted by exceeding the Python recursion limit. This can result in a Denial of service by crashing the application with a RecursionError. We recommend upgrading to version =>6.31.1 or beyond commit 17838beda2943d08b8a9d4df5b68f5f04f26d901
CVE-2025-4575	Issue summary: Use of -adreject option with the openssl x509 application adds a trusted use instead of a rejected use for a certificate. Impact summary: If a user intends to make a trusted certificate rejected for a particular use it will be instead marked as trusted for that use. A copy & paste error during minor refactoring of the code introduced this issue in the OpenSSL 3.5 version. If, for example, a trusted CA certificate should be trusted only for the purpose of authenticating TLS servers but not for CMS signature verification and the CMS signature verification is intended to be marked as rejected with the -adreject option, the resulting CA certificate will be trusted for CMS signature verification purpose instead. Only users which use the trusted certificate format who use the openssl x509 command line application to add rejected uses are affected by this issue. The issues affecting only the command line application are considered to be Low severity. The FIPS modules in 3.5, 3.4, 3.3, 3.2, 3.1 and 3.0 are not affected by this issue. OpenSSL 3.4, 3.3, 3.2, 3.1, 3.0, 1.1.1 and 1.0.2 are also not affected by this issue.
CVE-2025-45768	pyjwt v2.10.1 was discovered to contain weak encryption. NOTE: this is disputed by the Supplier because the key length is chosen by the application that uses the library (admittedly, library users may benefit from a minimum value and a mechanism for opting in to strict enforcement).
CVE-2025-4598	A vulnerability was found in systemd-coredump. This flaw allows an attacker to force a SUID process to crash and replace it with a non-SUID binary to access the original's privileged process coredump, allowing the attacker to read sensitive data, such as /etc/shadow content, loaded by the original process. A SUID binary or process has a special type of permission, which allows the process to run with the file owner's permissions, regardless of the user executing the binary. This allows the process to access more restricted data than unprivileged users or processes would be able to. An attacker can leverage this flaw by forcing a SUID process to crash and force the Linux kernel to recycle the process PID before systemd-coredump can analyze the /proc/pid/auxv file. If the attacker wins the race condition, they gain access to the original's SUID process coredump file. They can read sensitive content loaded into memory by the original binary, affecting data confidentiality.
CVE-2025-46392	Uncontrolled Resource Consumption vulnerability in Apache Commons Configuration 1.x. There are a number of issues in Apache Commons Configuration 1.x that allow excessive resource consumption when loading untrusted configurations or using unexpected usage patterns. The Apache Commons Configuration team does not intend to fix these issues in 1.x. Apache Commons Configuration 1.x is still safe to use in scenario's where you only load trusted configurations. Users that load untrusted configurations or give attackers control over usage patterns are recommended to upgrade to the 2.x version line, which fixes these issues. Apache Commons Configuration 2.x is not a drop-in replacement, but as it uses a separate Maven groupId and Java package namespace they can be loaded side-by-side, making it possible to do a gradual migration.
CVE-2025-46394	In tar in BusyBox through 1.37.0, a TAR archive can have filenames hidden from a listing through the use of terminal escape sequences.

CVE ID	Description
CVE-2025-46560	vLLM is a high-throughput and memory-efficient inference and serving engine for LLMs. Versions starting from 0.8.0 and prior to 0.8.5 are affected by a critical performance vulnerability in the input preprocessing logic of the multimodal tokenizer. The code dynamically replaces placeholder tokens (e.g., < audio_ >, < image_ >) with repeated tokens based on precomputed lengths. Due to inefficient list concatenation operations, the algorithm exhibits quadratic time complexity ($O(n^2)$), allowing malicious actors to trigger resource exhaustion via specially crafted inputs. This issue has been patched in version 0.8.5.
CVE-2025-46701	Improper Handling of Case Sensitivity vulnerability in Apache Tomcat's GCI servlet allows security constraint bypass of security constraints that apply to the pathInfo component of a URI mapped to the CGI servlet. This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.6, from 10.1.0-M1 through 10.1.40, from 9.0.0.M1 through 9.0.104. The following versions were EOL at the time the CVE was created but are known to be affected: 8.5.0 through 8.5.100. Other, older, EOL versions may also be affected. Users are recommended to upgrade to version 11.0.7, 10.1.41 or 9.0.105, which fixes the issue.
CVE-2025-46727	Rack is a modular Ruby web server interface. Prior to versions 2.2.14, 3.0.16, and 3.1.14, `Rack::QueryParser` parses query strings and `application/x-www-form-urlencoded` bodies into Ruby data structures without imposing any limit on the number of parameters, allowing attackers to send requests with extremely large numbers of parameters. The vulnerability arises because `Rack::QueryParser` iterates over each `&`-separated key-value pair and adds it to a Hash without enforcing an upper bound on the total number of parameters. This allows an attacker to send a single request containing hundreds of thousands (or more) of parameters, which consumes excessive memory and CPU during parsing. An attacker can trigger denial of service by sending specifically crafted HTTP requests, which can cause memory exhaustion or pin CPU resources, stalling or crashing the Rack server. This results in full service disruption until the affected worker is restarted. Versions 2.2.14, 3.0.16, and 3.1.14 fix the issue. Some other mitigations are available. One may use middleware to enforce a maximum query string size or parameter count, or employ a reverse proxy (such as Nginx) to limit request sizes and reject oversized query strings or bodies. Limiting request body sizes and query string lengths at the web server or CDN level is an effective mitigation.
CVE-2025-4673	Proxy-Authorization and Proxy-Authenticate headers persisted on cross-origin redirects potentially leaking sensitive information.
CVE-2025-46762	Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code. While 1.15.1 introduced a fix to restrict untrusted packages, the default setting of trusted packages still allows malicious classes from these packages to be executed. The exploit is only applicable if the client code of parquet-avro uses the "specific" or the "reflect" models deliberately for reading Parquet files. ("generic" model is not impacted) Users are recommended to upgrade to 1.15.2 or set the system property "org.apache.parquet.avro.SERIALIZABLE_PACKAGES" to an empty string on 1.15.1. Both are sufficient to fix the issue.
CVE-2025-46835	Git GUI allows you to use the Git source control management tools via a GUI. When a user clones an untrusted repository and is tricked into editing a file located in a maliciously named directory in the repository, then Git GUI can create and overwrite files for which the user has write permission. This vulnerability is fixed in 2.43.7, 2.44.4, 2.45.4, 2.46.4, 2.47.3, 2.48.2, 2.49.1, and 2.50.1.
CVE-2025-47273	setuptools is a package that allows users to download, build, install, upgrade, and uninstall Python packages. A path traversal vulnerability in `PackageIndex` is present in setuptools prior to version 78.1.1. An attacker would be allowed to write files to arbitrary locations on the filesystem with the permissions of the process running the Python code, which could escalate to remote code execution depending on the context. Version 78.1.1 fixes the issue.
CVE-2025-47277	vLLM, an inference and serving engine for large language models (LLMs), has an issue in versions 0.6.5 through 0.8.4 that ONLY impacts environments using the `PyNcclPipe` KV cache transfer integration with the V0 engine. No other configurations are affected. vLLM supports the use of the `PyNcclPipe` class to establish a peer-to-peer communication domain for data transmission between distributed nodes. The GPU-side KV-Cache transmission is implemented through the `PyNcclCommunicator` class, while CPU-side control message passing is handled via the `send_obj` and `recv_obj` methods on the CPU side. The intention was that this interface should only be exposed to a private network using the IP address specified by the `--kv-ip` CLI parameter. The vLLM documentation covers how this must be limited to a secured network. The default and intentional behavior from PyTorch is that the `TCPStore` interface listens on ALL interfaces, regardless of what IP address is provided. The IP address given was only used as a client-side address to use. vLLM was fixed to use a workaround to force the `TCPStore` instance to bind its socket to a specified private interface. As of version 0.8.5, vLLM limits the `TCPStore` socket to the private interface as configured.

CVE ID	Description
CVE-2025-47287	Tornado is a Python web framework and asynchronous networking library. When Tornado's ``multipart/form-data`` parser encounters certain errors, it logs a warning but continues trying to parse the remainder of the data. This allows remote attackers to generate an extremely high volume of logs, constituting a DoS attack. This DoS is compounded by the fact that the logging subsystem is synchronous. All versions of Tornado prior to 6.5.0 are affected. The vulnerable parser is enabled by default. Upgrade to Tornado version 6.5.0 to receive a patch. As a workaround, risk can be mitigated by blocking `Content-Type: multipart/form-data` in a proxy.
CVE-2025-47291	containerd is an open-source container runtime. A bug was found in the containerd's CRI implementation where containerd, starting in version 2.0.1 and prior to version 2.0.5, doesn't put usernamespaced containers under the Kubernetes' cgroup hierarchy, therefore some Kubernetes limits are not honored. This may cause a denial of service of the Kubernetes node. This bug has been fixed in containerd 2.0.5+ and 2.1.0+. Users should update to these versions to resolve the issue. As a workaround, disable usernamespaced pods in Kubernetes temporarily.
CVE-2025-47906	If the PATH environment variable contains paths which are executables (rather than just directories), passing certain strings to LookPath("", ".", and ".."), can result in the binaries listed in the PATH being unexpectedly returned.
CVE-2025-47907	Cancelling a query (e.g. by cancelling the context passed to one of the query methods) during a call to the Scan method of the returned Rows can result in unexpected results if other queries are being made in parallel. This can result in a race condition that may overwrite the expected results with those of another query, causing the call to Scan to return either unexpected results from the other query or an error.
CVE-2025-47910	When using http.CrossOriginProtection, the AddInsecureBypassPattern method can unexpectedly bypass more requests than intended. CrossOriginProtection then skips validation, but forwards the original request path, which may be served by a different handler without the intended security protections.
CVE-2025-47917	Mbed TLS before 3.6.4 allows a use-after-free in certain situations of applications that are developed in accordance with the documentation. The function mbedtls_x509_string_to_names() takes a head argument that is documented as an output argument. The documentation does not suggest that the function will free that pointer; however, the function does call mbedtls_asn1_free_named_data_list() on that argument, which performs a deep free(). As a result, application code that uses this function (relying only on documented behavior) is likely to still hold pointers to the memory blocks that were freed, resulting in a high risk of use-after-free or double-free. In particular, the two sample programs x509/cert_write and x509/cert_req are affected (use-after-free if the san string contains more than one DN).
CVE-2025-48371	OpenFGA is an authorization/permission engine. OpenFGA versions 1.8.0 through 1.8.12 (corresponding to Helm chart openfga-0.2.16 through openfga-0.2.30 and docker 1.8.0 through 1.8.12) are vulnerable to authorization bypass when certain Check and ListObject calls are executed. Users are affected under four specific conditions: First, calling Check API or ListObjects with an authorization model that has a relationship directly assignable by both type bound public access and user; second, there are check or list object queries with contextual tuples for the relationship that can be directly assignable by both type bound public access and user; third, those contextual tuples's user field is an user; and finally, type bound public access tuples are not assigned to the relationship. Users should upgrade to version 1.8.13 to receive a patch. The upgrade is backwards compatible.
CVE-2025-48379	Pillow is a Python imaging library. In versions 11.2.0 to before 11.3.0, there is a heap buffer overflow when writing a sufficiently large (>64k encoded with default settings) image in the DDS format due to writing into a buffer without checking for available space. This only affects users who save untrusted data as a compressed DDS image. This issue has been patched in version 11.3.0.
CVE-2025-48384	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals. When reading a config value, Git strips any trailing carriage return and line feed (CRLF). When writing a config entry, values with a trailing CR are not quoted, causing the CR to be lost when the config is later read. When initializing a submodule, if the submodule path contains a trailing CR, the altered path is read resulting in the submodule being checked out to an incorrect location. If a symlink exists that points the altered path to the submodule hooks directory, and the submodule contains an executable post-checkout hook, the script may be unintentionally executed after checkout. This vulnerability is fixed in v2.43.7, v2.44.4, v2.45.4, v2.46.4, v2.47.3, v2.48.2, v2.49.1, and v2.50.1.

CVE ID	Description
CVE-2025-48385	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals. When cloning a repository Git knows to optionally fetch a bundle advertised by the remote server, which allows the server-side to offload parts of the clone to a CDN. The Git client does not perform sufficient validation of the advertised bundles, which allows the remote side to perform protocol injection. This protocol injection can cause the client to write the fetched bundle to a location controlled by the adversary. The fetched content is fully controlled by the server, which can in the worst case lead to arbitrary code execution. The use of bundle URIs is not enabled by default and can be controlled by the bundle.heuristic config option. Some cases of the vulnerability require that the adversary is in control of where a repository will be cloned to. This either requires social engineering or a recursive clone with submodules. These cases can thus be avoided by disabling recursive clones. This vulnerability is fixed in v2.43.7, v2.44.4, v2.45.4, v2.46.4, v2.47.3, v2.48.2, v2.49.1, and v2.50.1.
CVE-2025-48386	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals. The wincred credential helper uses a static buffer (target) as a unique key for storing and comparing against internal storage. This credential helper does not properly bounds check the available space remaining in the buffer before appending to it with wcsncat(), leading to potential buffer overflows. This vulnerability is fixed in v2.43.7, v2.44.4, v2.45.4, v2.46.4, v2.47.3, v2.48.2, v2.49.1, and v2.50.1.
CVE-2025-48387	tar-fs provides filesystem bindings for tar-stream. Versions prior to 3.0.9, 2.1.3, and 1.16.5 have an issue where an extract can write outside the specified dir with a specific tarball. This has been patched in versions 3.0.9, 2.1.3, and 1.16.5. As a workaround, use the ignore option to ignore non files/directories.
CVE-2025-48734	Improper Access Control vulnerability in Apache Commons. A special BeanIntrospector class was added in version 1.9.2. This can be used to stop attackers from using the declared class property of Java enum objects to get access to the classloader. However this protection was not enabled by default. PropertyUtilsBean (and consequently BeanUtilsBean) now disallows declared class level property access by default. Releases 1.11.0 and 2.0.0-M2 address a potential security issue when accessing enum properties in an uncontrolled way. If an application using Commons BeanUtils passes property paths from an external source directly to the getProperty() method of PropertyUtilsBean, an attacker can access the enum's class loader via the "declaredClass" property available on all Java "enum" objects. Accessing the enum's "declaredClass" allows remote attackers to access the ClassLoader and execute arbitrary code. The same issue exists with PropertyUtilsBean.getNestedProperty(). Starting in versions 1.11.0 and 2.0.0-M2 a special BeanIntrospector suppresses the "declaredClass" property. Note that this new BeanIntrospector is enabled by default, but you can disable it to regain the old behavior; see section 2.5 of the user's guide and the unit tests. This issue affects Apache Commons BeanUtils 1.x before 1.11.0, and 2.x before 2.0.0-M2. Users of the artifact commons-beanutils:commons-beanutils 1.x are recommended to upgrade to version 1.11.0, which fixes the issue. Users of the artifact org.apache.commons:commons-beanutils2 2.x are recommended to upgrade to version 2.0.0-M2, which fixes the issue.
CVE-2025-4877	There's a vulnerability in the libssh package where when a libssh consumer passes in an unexpectedly large input buffer to ssh_get_fingerprint_hash() function. In such cases the bin_to_base64() function can experience an integer overflow leading to a memory under allocation, when that happens it's possible that the program perform out of bounds write leading to a heap corruption. This issue affects only 32-bits builds of libssh.
CVE-2025-4878	A vulnerability was found in libssh, where an uninitialized variable exists under certain conditions in the privatekey_from_file() function. This flaw can be triggered if the file specified by the filename doesn't exist and may lead to possible signing failures or heap corruption.
CVE-2025-48795	Apache CXF stores large stream based messages as temporary files on the local filesystem. A bug was introduced which means that the entire temporary file is read into memory and then logged. An attacker might be able to exploit this to cause a denial of service attack by causing an out of memory exception. In addition, it is possible to configure CXF to encrypt temporary files to prevent sensitive credentials from being cached unencrypted on the local filesystem, however this bug means that the cached files are written out to logs unencrypted. Users are recommended to upgrade to versions 3.5.11, 3.6.6, 4.0.7 or 4.1.1, which fixes this issue.
CVE-2025-48924	Uncontrolled Recursion vulnerability in Apache Commons Lang. This issue affects Apache Commons Lang: Starting with commons-lang:commons-lang 2.0 to 2.6, and, from org.apache.commons:commons-lang3 3.0 before 3.18.0. The methods ClassUtils.getClass(...) can throw StackOverflowError on very long inputs. Because an Error is usually not handled by applications and libraries, a StackOverflowError could cause an application to stop. Users are recommended to upgrade to version 3.18.0, which fixes the issue.
CVE-2025-48965	Mbed TLS before 3.6.4 has a NULL pointer dereference because mbedtls_asn1_store_named_data can trigger conflicting data with val.p of NULL but val.len greater than zero.

CVE ID	Description
CVE-2025-48976	Allocation of resources for multipart headers with insufficient limits enabled a DoS vulnerability in Apache Commons FileUpload. This issue affects Apache Commons FileUpload: from 1.0 before 1.6; from 2.0.0-M1 before 2.0.0-M4. Users are recommended to upgrade to versions 1.6 or 2.0.0-M4, which fix the issue.
CVE-2025-48988	Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat. This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.7, from 10.1.0-M1 through 10.1.41, from 9.0.0.M1 through 9.0.105. The following versions were EOL at the time the CVE was created but are known to be affected: 8.5.0 through 8.5.100. Other, older, EOL versions may also be affected. Users are recommended to upgrade to version 11.0.8, 10.1.42 or 9.0.106, which fix the issue.
CVE-2025-48989	Improper Resource Shutdown or Release vulnerability in Apache Tomcat made Tomcat vulnerable to the made you reset attack. This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.9, from 10.1.0-M1 through 10.1.43 and from 9.0.0.M1 through 9.0.107. Older, EOL versions may also be affected. Users are recommended to upgrade to one of versions 11.0.10, 10.1.44 or 9.0.108 which fix the issue.
CVE-2025-49007	Rack is a modular Ruby web server interface. Starting in version 3.1.0 and prior to version 3.1.16, there is a denial of service vulnerability in the Content-Disposition parsing component of Rack. This is very similar to the previous security issue CVE-2022-44571. Carefully crafted input can cause Content-Disposition header parsing in Rack to take an unexpected amount of time, possibly resulting in a denial of service attack vector. This header is used typically used in multipart parsing. Any applications that parse multipart posts using Rack (virtually all Rails applications) are impacted. Version 3.1.16 contains a patch for the vulnerability.
CVE-2025-49125	Authentication Bypass Using an Alternate Path or Channel vulnerability in Apache Tomcat. When using PreResources or PostResources mounted other than at the root of the web application, it was possible to access those resources via an unexpected path. That path was likely not to be protected by the same security constraints as the expected path, allowing those security constraints to be bypassed. This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.7, from 10.1.0-M1 through 10.1.41, from 9.0.0.M1 through 9.0.105. The following versions were EOL at the time the CVE was created but are known to be affected: 8.5.0 through 8.5.100. Other, older, EOL versions may also be affected. Users are recommended to upgrade to version 11.0.8, 10.1.42 or 9.0.106, which fix the issue.
CVE-2025-49128	Jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. Starting in version 2.0.0 and prior to version 2.13.0, a flaw in jackson-core's <code>JsonLocation.appendSourceDesc</code> method allows up to 500 bytes of unintended memory content to be included in exception messages. When parsing JSON from a byte array with an offset and length, the exception message incorrectly reads from the beginning of the array instead of the logical payload start. This results in possible information disclosure in systems using pooled or reused buffers, like Netty or Vert.x. This issue was silently fixed in jackson-core version 2.13.0, released on September 30, 2021, via PR #652. All users should upgrade to version 2.13.0 or later. If upgrading is not immediately possible, applications can mitigate the issue by disabling exception message exposure to clients to avoid returning parsing exception messages in HTTP responses and/or disabling source inclusion in exceptions to prevent Jackson from embedding any source content in exception messages, avoiding leakage.
CVE-2025-4947	libcurl accidentally skips the certificate verification for QUIC connections when connecting to a host specified as an IP address in the URL. Therefore, it does not detect impostors or man-in-the-middle attacks.
CVE-2025-49794	A use-after-free vulnerability was found in libxml2. This issue occurs when parsing XPath elements under certain circumstances when the XML schematron has the <code><sch:name path="..."></code> schema elements. This flaw allows a malicious actor to craft a malicious XML document used as input for libxml, resulting in the program's crash using libxml or other possible undefined behaviors.
CVE-2025-49796	A vulnerability was found in libxml2. Processing certain <code>sch:name</code> elements from the input XML file can trigger a memory corruption issue. This flaw allows an attacker to craft a malicious XML input file that can lead libxml to crash, resulting in a denial of service or other possible undefined behavior due to sensitive data being corrupted in memory.
CVE-2025-50181	urllib3 is a user-friendly HTTP client library for Python. Prior to 2.5.0, it is possible to disable redirects for all requests by instantiating a PoolManager and specifying retries in a way that disable redirects. By default, requests and botocore users are not affected. An application attempting to mitigate SSRF or open redirect vulnerabilities by disabling redirects at the PoolManager level will remain vulnerable. This issue has been patched in version 2.5.0.

CVE ID	Description
CVE-2025-50182	urllib3 is a user-friendly HTTP client library for Python. Starting in version 2.2.0 and prior to 2.5.0, urllib3 does not control redirects in browsers and Node.js. urllib3 supports being used in a Pyodide runtime utilizing the JavaScript Fetch API or falling back on XMLHttpRequest. This means Python libraries can be used to make HTTP requests from a browser or Node.js. Additionally, urllib3 provides a mechanism to control redirects, but the retries and redirect parameters are ignored with Pyodide; the runtime itself determines redirect behavior. This issue has been patched in version 2.5.0.
CVE-2025-50213	Failure to Sanitize Special Elements into a Different Plane (Special Element Injection) vulnerability in Apache Airflow Providers Snowflake. This issue affects Apache Airflow Providers Snowflake: before 6.4.0. Sanitation of table and stage parameters were added in CopyFromExternalStageToSnowflakeOperator to prevent SQL injection. Users are recommended to upgrade to version 6.4.0, which fixes the issue.
CVE-2025-5025	libcurl supports *pinning* of the server certificate public key for HTTPS transfers. Due to an omission, this check is not performed when connecting with QUIC for HTTP/3, when the TLS backend is wolfSSL. Documentation says the option works with wolfSSL, failing to specify that it does not for QUIC and HTTP/3. Since pinning makes the transfer succeed if the pin is fine, users could unwittingly connect to an impostor server without noticing.
CVE-2025-50817	A vulnerability in the Python-Future 1.0.0 module allows for arbitrary code execution via the unintended import of a file named test.py. When the module is loaded, it automatically imports test.py, if present in the same directory or in the sys.path. This behavior can be exploited by an attacker who has the ability to write files to the server, allowing the execution of arbitrary code.
CVE-2025-50952	openjpeg v 2.5.0 was discovered to contain a NULL pointer dereference via the component /openjp2/dwt.c.
CVE-2025-5187	A vulnerability exists in the NodeRestriction admission controller in Kubernetes clusters where node users can delete their corresponding node object by patching themselves with an OwnerReference to a cluster-scoped resource. If the OwnerReference resource does not exist or is subsequently deleted, the given node object will be deleted via garbage collection.
CVE-2025-52194	A buffer overflow vulnerability exists in libsndfile version 1.2.2 and potentially earlier versions when processing malformed IRCAM audio files. The vulnerability occurs in the ircam_read_header function at src/ircam.c:164 during sample rate processing, leading to memory corruption and potential code execution.
CVE-2025-5222	A stack buffer overflow was found in International components for unicode (ICU). While running the genrb binary, the 'subtag' struct overflowed at the SRBRoot::addTag function. This issue may lead to memory corruption and local arbitrary code execution.
CVE-2025-5244	A vulnerability was found in GNU Binutils up to 2.44. It has been rated as critical. Affected by this issue is the function elf_gc_sweep of the file bfd/elflink.c of the component ld. The manipulation leads to memory corruption. An attack has to be approached locally. The exploit has been disclosed to the public and may be used. Upgrading to version 2.45 is able to address this issue. It is recommended to upgrade the affected component.
CVE-2025-5245	A vulnerability classified as critical has been found in GNU Binutils up to 2.44. This affects the function debug_type_samep of the file /binutils/debug.c of the component objdump. The manipulation leads to memory corruption. Local access is required to approach this attack. The exploit has been disclosed to the public and may be used. It is recommended to apply a patch to fix this issue.
CVE-2025-52496	Mbed TLS before 3.6.4 has a race condition in AESNI detection if certain compiler optimizations occur. An attacker may be able to extract an AES key from a multithreaded program, or perform a GCM forgery.
CVE-2025-52497	Mbed TLS before 3.6.4 has a PEM parsing one-byte heap-based buffer underflow, in mbedtls_pem_read_buffer and two mbedtls_pk_parse functions, via untrusted PEM input.
CVE-2025-5278	A flaw was found in GNU Coreutils. The sort utility's begfield() function is vulnerable to a heap buffer under-read. The program may access memory outside the allocated buffer if a user runs a crafted command using the traditional key format. A malicious input could lead to a crash or leak sensitive data.
CVE-2025-5279	When the Amazon Redshift Python Connector is configured with the BrowserAzureOAuth2CredentialsProvider plugin, the driver skips the SSL certificate validation step for the Identity Provider. An insecure connection could allow an actor to intercept the token exchange process and retrieve an access token. This issue has been addressed in driver version 2.1.7. Users should upgrade to address this issue and ensure any forked or derivative code is patched to incorporate the new fixes.
CVE-2025-5283	Use after free in libvpx in Google Chrome prior to 137.0.7151.55 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: Medium)

CVE ID	Description
CVE-2025-52999	jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. In versions prior to 2.15.0, if a user parses an input file and it has deeply nested data, Jackson could end up throwing a StackoverflowError if the depth is particularly large. jackson-core 2.15.0 contains a configurable limit for how deep Jackson will traverse in an input document, defaulting to an allowable depth of 1000. jackson-core will throw a StreamConstraintsException if the limit is reached. jackson-databind also benefits from this change because it uses jackson-core to parse JSON inputs. As a workaround, users should avoid parsing input files from untrusted sources.
CVE-2025-53020	Late Release of Memory after Effective Lifetime vulnerability in Apache HTTP Server. This issue affects Apache HTTP Server: from 2.4.17 up to 2.4.63. Users are recommended to upgrade to version 2.4.64, which fixes the issue.
CVE-2025-5318	A flaw was found in the libssh library. An out-of-bounds read can be triggered in the sftp_handle function due to an incorrect comparison check that permits the function to access memory beyond the valid handle list and to return an invalid pointer, which is used in further processing. This vulnerability allows an authenticated remote attacker to potentially read unintended memory regions, exposing sensitive information or affect service behavior.
CVE-2025-5351	A flaw was found in the key export functionality of libssh. The issue occurs in the internal function responsible for converting cryptographic keys into serialized formats. During error handling, a memory structure is freed but not cleared, leading to a potential double free issue if an additional failure occurs later in the function. This condition may result in heap corruption or application instability in low-memory scenarios, posing a risk to system reliability where key export operations are performed.
CVE-2025-53547	Helm is a package manager for Charts for Kubernetes. Prior to 3.18.4, a specially crafted Chart.yaml file along with a specially linked Chart.lock file can lead to local code execution when dependencies are updated. Fields in a Chart.yaml file, that are carried over to a Chart.lock file when dependencies are updated and this file is written, can be crafted in a way that can cause execution if that same content were in a file that is executed (e.g., a bash.rc file or shell script). If the Chart.lock file is symlinked to one of these files updating dependencies will write the lock file content to the symlinked file. This can lead to unwanted execution. Helm warns of the symlinked file but did not stop execution due to symlinking. This issue has been resolved in Helm v3.18.4.
CVE-2025-53643	AIOHTTP is an asynchronous HTTP client/server framework for asyncio and Python. Prior to version 3.12.14, the Python parser is vulnerable to a request smuggling vulnerability due to not parsing trailer sections of an HTTP request. If a pure Python version of aiohttp is installed (i.e. without the usual C extensions) or AIOHTTP_NO_EXTENSIONS is enabled, then an attacker may be able to execute a request smuggling attack to bypass certain firewalls or proxy protections. Version 3.12.14 contains a patch for this issue.
CVE-2025-5372	A flaw was found in libssh versions built with OpenSSL versions older than 3.0, specifically in the ssh_kdf() function responsible for key derivation. Due to inconsistent interpretation of return values where OpenSSL uses 0 to indicate failure and libssh uses 0 for success—the function may mistakenly return a success status even when key derivation fails. This results in uninitialized cryptographic key buffers being used in subsequent communication, potentially compromising SSH sessions' confidentiality, integrity, and availability.
CVE-2025-53864	Connect2id Nimbus JOSE + JWT 10.0.x before 10.0.2 and 9.37.x before 9.37.4 allows a remote attacker to cause a denial of service via a deeply nested JSON object supplied in a JWT claim set, because of uncontrolled recursion. NOTE: this is independent of the Gson 2.11.0 issue because the Connect2id product could have checked the JSON object nesting depth, regardless of what limits (if any) were imposed by Gson.
CVE-2025-54121	Starlette is a lightweight ASGI (Asynchronous Server Gateway Interface) framework/toolkit, designed for building async web services in Python. In versions 0.47.1 and below, when parsing a multi-part form with large files (greater than the default max spool size) starlette will block the main thread to roll the file over to disk. This blocks the event thread which means the application can't accept new connections. The UploadFile code has a minor bug where instead of just checking for self._in_memory, the logic should also check if the additional bytes will cause a rollover. The vulnerability is fixed in version 0.47.2.
CVE-2025-54368	uv is a Python package and project manager written in Rust. In versions 0.8.5 and earlier, remote ZIP archives were handled in a streamwise fashion, and file entries were not reconciled against the archive's central directory. An attacker could contrive a ZIP archive that would extract with legitimate contents on some package installers, and malicious contents on others due to multiple local file entries. An attacker could also contrive a "stacked" ZIP input with multiple internal ZIPs, which would be handled differently by different package installers. The attacker could choose which installer to target in both scenarios. This issue is fixed in version 0.8.6. To work around this issue, users may choose to set UV_INSECURE_NO_ZIP_VALIDATION=1 to revert to the previous behavior.

CVE ID	Description
CVE-2025-54410	Moby is an open source container framework developed by Docker Inc. that is distributed as Docker Engine, Mirantis Container Runtime, and various other downstream projects/products. A firewall vulnerability affects Moby releases before 28.0.0. When firewall reloads, Docker fails to re-create iptables rules that isolate bridge networks, allowing any container to access all ports on any other container across different bridge networks on the same host. This breaks network segmentation between containers that should be isolated, creating significant risk in multi-tenant environments. Only containers in --internal networks remain protected. Workarounds include reloading firewall and either restarting the docker daemon, re-creating bridge networks, or using rootless mode. Maintainers anticipate a fix for this issue in version 25.0.13.
CVE-2025-55163	Netty is an asynchronous, event-driven network application framework. Prior to versions 4.1.124.Final and 4.2.4.Final, Netty is vulnerable to MadeYouReset DDoS. This is a logical vulnerability in the HTTP/2 protocol, that uses malformed HTTP/2 control frames in order to break the max concurrent streams limit - which results in resource exhaustion and distributed denial of service. This issue has been patched in versions 4.1.124.Final and 4.2.4.Final.
CVE-2025-5683	When loading a specifically crafted ICNS format image file in QImage then it will trigger a crash. This issue affects Qt from versions 6.3.0 through 6.5.9, from 6.6.0 through 6.8.4, 6.9.0. This is fixed in 6.5.10, 6.8.5 and 6.9.1.
CVE-2025-5702	The strcmp implementation optimized for the Power10 processor in the GNU C Library version 2.39 and later writes to vector registers v20 to v31 without saving contents from the caller (those registers are defined as non-volatile registers by the powerpc64le ABI), resulting in overwriting of its contents and potentially altering control flow of the caller, or leaking the input strings to the function to other parts of the program.
CVE-2025-57804	h2 is a pure-Python implementation of a HTTP/2 protocol stack. Prior to version 4.3.0, an HTTP/2 request splitting vulnerability allows attackers to perform request smuggling attacks by injecting CRLF characters into headers. This occurs when servers downgrade HTTP/2 requests to HTTP/1.1 without properly validating header names/values, enabling attackers to manipulate request boundaries and bypass security controls. This issue has been patched in version 4.3.0.
CVE-2025-58056	Netty is an asynchronous event-driven network application framework for development of maintainable high performance protocol servers and clients. In versions 4.1.124.Final, and 4.2.0.Alpha3 through 4.2.4.Final, Netty incorrectly accepts standalone newline characters (LF) as a chunk-size line terminator, regardless of a preceding carriage return (CR), instead of requiring CRLF per HTTP/1.1 standards. When combined with reverse proxies that parse LF differently (treating it as part of the chunk extension), attackers can craft requests that the proxy sees as one request but Netty processes as two, enabling request smuggling attacks. This is fixed in versions 4.1.125.Final and 4.2.5.Final.
CVE-2025-58057	Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. In netty-codec-compression versions 4.1.124.Final and below, and netty-codec versions 4.2.4.Final and below, when supplied with specially crafted input, BrotliDecoder and certain other decompression decoders will allocate a large number of reachable byte buffers, which can lead to denial of service. BrotliDecoder.decompress has no limit in how often it calls pull, decompressing data 64K bytes at a time. The buffers are saved in the output list, and remain reachable until OOM is hit. This is fixed in versions 4.1.125.Final of netty-codec and 4.2.5.Final of netty-codec-compression.
CVE-2025-58065	Flask-AppBuilder is an application development framework. Prior to version 4.8.1, when Flask-AppBuilder is configured to use OAuth, LDAP, or other non-database authentication methods, the password reset endpoint remains registered and accessible, despite not being displayed in the user interface. This allows an enabled user to reset their password and be able to create JWT tokens even after the user is disabled on the authentication provider. Users should upgrade to Flask-AppBuilder version 4.8.1 or later to receive a fix. If immediate upgrade is not possible, manually disable password reset routes in the application configuration; implement additional access controls at the web server or proxy level to block access to the reset my password URL; and/or monitor for suspicious password reset attempts from disabled accounts.
CVE-2025-58068	Eventlet is a concurrent networking library for Python. Prior to version 0.40.3, the Eventlet WSGI parser is vulnerable to HTTP Request Smuggling due to improper handling of HTTP trailer sections. This vulnerability could enable attackers to, bypass front-end security controls, launch targeted attacks against active site users, and poison web caches. This problem has been patched in Eventlet 0.40.3 by dropping trailers which is a breaking change if a backend behind eventlet.wsgi proxy requires trailers. A workaround involves not using eventlet.wsgi facing untrusted clients.

CVE ID	Description
CVE-2025-58364	OpenPrinting CUPS is an open source printing system for Linux and other Unix-like operating systems. In versions 2.4.12 and earlier, an unsafe deserialization and validation of printer attributes causes null dereference in the libcups library. This is a remote DoS vulnerability available in local subnet in default configurations. It can cause the cups & cups-browsed to crash, on all the machines in local network who are listening for printers (so by default for all regular linux machines). On systems where the vulnerability CVE-2024-47176 (cups-filters 1.x/cups-browsed 2.x vulnerability) was not fixed, and the firewall on the machine does not reject incoming communication to IPP port, and the machine is set to be available to public internet, attack vector "Network" is possible. The current versions of CUPS and cups-browsed projects have the attack vector "Adjacent" in their default configurations. Version 2.4.13 contains a patch for CVE-2025-58364.
CVE-2025-58767	REXML is an XML toolkit for Ruby. The REXML gems from 3.3.3 to 3.4.1 has a DoS vulnerability when parsing XML containing multiple XML declarations. If you need to parse untrusted XMLs, you may be impacted to these vulnerabilities. The REXML gem 3.4.2 or later include the patches to fix these vulnerabilities.
CVE-2025-5889	A vulnerability was found in juliangruber brace-expansion up to 1.1.11/2.0.1/3.0.0/4.0.0. It has been rated as problematic. Affected by this issue is the function expand of the file index.js. The manipulation leads to inefficient regular expression complexity. The attack may be launched remotely. The complexity of an attack is rather high. The exploitation is known to be difficult. The exploit has been disclosed to the public and may be used. Upgrading to version 1.1.12, 2.0.2, 3.0.1 and 4.0.1 is able to address this issue. The name of the patch is a5b98a4f30d7813266b221435e1eaf25a1b0ac5. It is recommended to upgrade the affected component.
CVE-2025-5914	A vulnerability has been identified in the libarchive library, specifically within the archive_read_format_rar_seek_data() function. This flaw involves an integer overflow that can ultimately lead to a double-free condition. Exploiting a double-free vulnerability can result in memory corruption, enabling an attacker to execute arbitrary code or cause a denial-of-service condition.
CVE-2025-5915	A vulnerability has been identified in the libarchive library. This flaw can lead to a heap buffer over-read due to the size of a filter block potentially exceeding the Lempel-Ziv-Storer-Schieber (LZSS) window. This means the library may attempt to read beyond the allocated memory buffer, which can result in unpredictable program behavior, crashes (denial of service), or the disclosure of sensitive information from adjacent memory regions.
CVE-2025-5916	A vulnerability has been identified in the libarchive library. This flaw involves an integer overflow that can be triggered when processing a Web Archive (WARC) file that claims to have more than INT64_MAX - 4 content bytes. An attacker could craft a malicious WARC archive to induce this overflow, potentially leading to unpredictable program behavior, memory corruption, or a denial-of-service condition within applications that process such archives using libarchive.
CVE-2025-5917	A vulnerability has been identified in the libarchive library. This flaw involves an 'off-by-one' miscalculation when handling prefixes and suffixes for file names. This can lead to a 1-byte write overflow. While seemingly small, such an overflow can corrupt adjacent memory, leading to unpredictable program behavior, crashes, or in specific circumstances, could be leveraged as a building block for more sophisticated exploitation.
CVE-2025-5918	A vulnerability has been identified in the libarchive library. This flaw can be triggered when file streams are piped into bsdtar, potentially allowing for reading past the end of the file. This out-of-bounds read can lead to unintended consequences, including unpredictable program behavior, memory corruption, or a denial-of-service condition.
CVE-2025-59343	tar-fs provides filesystem bindings for tar-stream. Versions prior to 3.1.1, 2.1.3, and 1.16.5 are vulnerable to symlink validation bypass if the destination directory is predictable with a specific tarball. This issue has been patched in version 3.1.1, 2.1.4, and 1.16.6. A workaround involves using the ignore option on non files/directories.
CVE-2025-59375	libexpat in Expat before 2.7.2 allows attackers to trigger large dynamic memory allocations via a small document that is submitted for parsing.

CVE ID	Description
CVE-2025-59419	Netty is an asynchronous, event-driven network application framework. In versions prior to 4.1.128.Final and 4.2.7.Final, the SMTP codec in Netty contains an SMTP command injection vulnerability due to insufficient input validation for Carriage Return (r) and Line Feed (n) characters in user-supplied parameters. The vulnerability exists in <code>io.netty.handler.codec.smtp.DefaultSmtpRequest</code> , where parameters are directly concatenated into the SMTP command string without sanitization. When methods such as <code>SmtpRequests.rcpt(recipient)</code> are called with a malicious string containing CRLF sequences, attackers can inject arbitrary SMTP commands. Because the injected commands are sent from the server's trusted IP address, resulting emails will likely pass SPF and DKIM authentication checks, making them appear legitimate. This allows remote attackers who can control SMTP command parameters (such as email recipients) to forge arbitrary emails from the trusted server, potentially impersonating executives and forging high-stakes corporate communications. This issue has been patched in versions 4.1.129.Final and 4.2.8.Final. No known workarounds exist.
CVE-2025-59420	Authlib is a Python library which builds OAuth and OpenID Connect servers. Prior to version 1.6.4, Authlib's JWS verification accepts tokens that declare unknown critical header parameters (crit), violating RFC 7515 "must#understand" semantics. An attacker can craft a signed token with a critical header (for example, bork or cnf) that strict verifiers reject but Authlib accepts. In mixed#language fleets, this enables split#brain verification and can lead to policy bypass, replay, or privilege escalation. This issue has been patched in version 1.6.4.
CVE-2025-5987	A flaw was found in libssh when using the ChaCha20 cipher with the OpenSSL library. If an attacker manages to exhaust the heap space, this error is not detected and may lead to libssh using a partially initialized cipher context. This occurs because the OpenSSL error code returned aliases with the SSH_OK code, resulting in libssh not properly detecting the error returned by the OpenSSL library. This issue can lead to undefined behavior, including compromised data confidentiality and integrity or crashes.
CVE-2025-6020	A flaw was found in linux-pam. The module <code>pam_namespace</code> may use access user-controlled paths without proper protection, allowing local users to elevate their privileges to root via multiple symlink attacks and race conditions.
CVE-2025-6021	A flaw was found in libxml2's <code>xmlBuildQName</code> function, where integer overflows in buffer size calculations can lead to a stack-based buffer overflow. This issue can result in memory corruption or a denial of service when processing crafted input.
CVE-2025-6052	A flaw was found in how GLib's GString manages memory when adding data to strings. If a string is already very large, combining it with more input can cause a hidden overflow in the size calculation. This makes the system think it has enough memory when it doesn't. As a result, data may be written past the end of the allocated memory, leading to crashes or memory corruption.
CVE-2025-6069	The <code>html.parser.HTMLParser</code> class had worse-case quadratic complexity when processing certain crafted malformed inputs potentially leading to amplified denial-of-service.
CVE-2025-61594	In affected URI version, a bypass exists for the fix to CVE-2025-27221 that can expose user credentials. This vulnerability has been assigned the CVE identifier CVE-2025-61594. We recommend upgrading the <code>uri</code> gem. ## Details When using the <code>+</code> operator to combine URIs, sensitive information like passwords from the original URI can be leaked, violating RFC3986 and making applications vulnerable to credential exposure. Please update URI gem to version 0.12.5, 0.13.3, 1.0.4 or later. ## Affected versions <code>uri</code> gem versions <code>< 0.12.5</code> , <code>0.13.0</code> to <code>0.13.2</code> and <code>1.0.0</code> to <code>1.0.3</code> . ## Credits Thanks to <code>junfuchong</code> (<code>chongfujun</code>) for discovering this issue. Also thanks to <code>nobu</code> for additional fixes of this vulnerability.
CVE-2025-6170	A flaw was found in the interactive shell of the <code>xmllint</code> command-line tool, used for parsing XML files. When a user inputs an overly long command, the program does not check the input size properly, which can cause it to crash. This issue might allow attackers to run harmful code in rare configurations without modern protections.
CVE-2025-61770	Rack is a modular Ruby web server interface. In versions prior to 2.2.19, 3.1.17, and 3.2.2, <code>Rack::Multipart::Parser</code> buffers the entire multipart preamble (bytes before the first boundary) in memory without any size limit. A client can send a large preamble followed by a valid boundary, causing significant memory use and potential process termination due to out-of-memory (OOM) conditions. Remote attackers can trigger large transient memory spikes by including a long preamble in multipart/form-data requests. The impact scales with allowed request sizes and concurrency, potentially causing worker crashes or severe slowdown due to garbage collection. Versions 2.2.19, 3.1.17, and 3.2.2 enforce a preamble size limit (e.g., 16 KiB) or discard preamble data entirely. Workarounds include limiting total request body size at the proxy or web server level and monitoring memory and set per-process limits to prevent OOM conditions.

CVE ID	Description
CVE-2025-61771	Rack is a modular Ruby web server interface. In versions prior to 2.2.19, 3.1.17, and 3.2.2, <code>Rack::Multipart::Parser</code> stores non-file form fields (parts without a <code>filename</code>) entirely in memory as Ruby <code>String</code> objects. A single large text field in a multipart/form-data request (hundreds of megabytes or more) can consume equivalent process memory, potentially leading to out-of-memory (OOM) conditions and denial of service (DoS). Attackers can send large non-file fields to trigger excessive memory usage. Impact scales with request size and concurrency, potentially leading to worker crashes or severe garbage-collection overhead. All Rack applications processing multipart form submissions are affected. Versions 2.2.19, 3.1.17, and 3.2.2 enforce a reasonable size cap for non-file fields (e.g., 2 MiB). Workarounds include restricting maximum request body size at the web-server or proxy layer (e.g., Nginx <code>client_max_body_size</code>) and validating and rejecting unusually large form fields at the application level.
CVE-2025-61772	Rack is a modular Ruby web server interface. In versions prior to 2.2.19, 3.1.17, and 3.2.2, <code>Rack::Multipart::Parser</code> can accumulate unbounded data when a multipart part's header block never terminates with the required blank line (<code>CRLF</code>). The parser keeps appending incoming bytes to memory without a size cap, allowing a remote attacker to exhaust memory and cause a denial of service (DoS). Attackers can send incomplete multipart headers to trigger high memory use, leading to process termination (OOM) or severe slowdown. The effect scales with request size limits and concurrency. All applications handling multipart uploads may be affected. Versions 2.2.19, 3.1.17, and 3.2.2 cap per-part header size (e.g., 64 KiB). As a workaround, restrict maximum request sizes at the proxy or web server layer (e.g., Nginx <code>client_max_body_size</code>).
CVE-2025-61780	Rack is a modular Ruby web server interface. Prior to versions 2.2.20, 3.1.18, and 3.2.3, a possible information disclosure vulnerability existed in <code>Rack::Sendfile</code> when running behind a proxy that supports <code>x-sendfile</code> headers (such as Nginx). Specially crafted headers could cause <code>Rack::Sendfile</code> to miscommunicate with the proxy and trigger unintended internal requests, potentially bypassing proxy-level access restrictions. When <code>Rack::Sendfile</code> received untrusted <code>x-sendfile-type</code> or <code>x-accel-mapping</code> headers from a client, it would interpret them as proxy configuration directives. This could cause the middleware to send a "redirect" response to the proxy, prompting it to reissue a new internal request that was not subject to the proxy's access controls. An attacker could exploit this by setting a crafted <code>x-sendfile-type: x-accel-redirect</code> header, setting a crafted <code>x-accel-mapping</code> header, and requesting a path that qualifies for proxy-based acceleration. Attackers could bypass proxy-enforced restrictions and access internal endpoints intended to be protected (such as administrative pages). The vulnerability did not allow arbitrary file reads but could expose sensitive application routes. This issue only affected systems meeting all of the following conditions: The application used <code>Rack::Sendfile</code> with a proxy that supports <code>x-accel-redirect</code> (e.g., Nginx); the proxy did not always set or remove the <code>x-sendfile-type</code> and <code>x-accel-mapping</code> headers; and the application exposed an endpoint that returned a body responding to <code>.to_path</code> . Users should upgrade to Rack versions 2.2.20, 3.1.18, or 3.2.3, which require explicit configuration to enable <code>x-accel-redirect</code> . Alternatively, configure the proxy to always set or strip the header, or in Rails applications, disable sendfile completely.
CVE-2025-61911	python-ldap is a lightweight directory access protocol (LDAP) client API for Python. In versions prior to 3.4.5, the sanitization method <code>ldap.filter.escape_filter_chars</code> can be tricked to skip escaping of special characters when a crafted <code>'list'</code> or <code>'dict'</code> is supplied as the <code>'assertion_value'</code> parameter, and the non-default <code>'escape_mode=1'</code> is configured. The method <code>ldap.filter.escape_filter_chars</code> supports 3 different escaping modes. <code>'escape_mode=0'</code> (default) and <code>'escape_mode=2'</code> happen to raise exceptions when a <code>'list'</code> or <code>'dict'</code> object is supplied as the <code>'assertion_value'</code> parameter. However, <code>'escape_mode=1'</code> computes without performing adequate logic to ensure a fully escaped return value. If an application relies on the vulnerable method in the <code>'python-ldap'</code> library to escape untrusted user input, an attacker might be able to abuse the vulnerability to launch ldap injection attacks which could potentially disclose or manipulate ldap data meant to be inaccessible to them. Version 3.4.5 fixes the issue by adding a type check at the start of the <code>ldap.filter.escape_filter_chars</code> method to raise an exception when the supplied <code>'assertion_value'</code> parameter is not of type <code>'str'</code> .
CVE-2025-61912	python-ldap is a lightweight directory access protocol (LDAP) client API for Python. In versions prior to 3.4.5, <code>ldap.dn.escape_dn_chars()</code> escapes <code>\x00</code> incorrectly by emitting a backslash followed by a literal NUL byte instead of the RFC-4514 hex form <code>\00</code> . Any application that uses this helper to construct DN's from untrusted input can be made to consistently fail before a request is sent to the LDAP server (e.g., AD), resulting in a client-side denial of service. Version 3.4.5 contains a patch for the issue.
CVE-2025-61919	Rack is a modular Ruby web server interface. Prior to versions 2.2.20, 3.1.18, and 3.2.3, <code>Rack::Request#POST</code> reads the entire request body into memory for <code>'Content-Type: application/x-www-form-urlencoded'</code> , calling <code>rack.input.read(nil)</code> without enforcing a length or cap. Large request bodies can therefore be buffered completely into process memory before parsing, leading to denial of service (DoS) through memory exhaustion. Users should upgrade to Rack version 2.2.20, 3.1.18, or 3.2.3, any of which enforces form parameter limits using <code>'query_parser.bytesize_limit'</code> , preventing unbounded reads of <code>'application/x-www-form-urlencoded'</code> bodies. Additionally, enforce strict maximum body size at the proxy or web server layer (e.g., Nginx <code>client_max_body_size</code> , Apache <code>LimitRequestBody</code>).

CVE ID	Description
CVE-2025-61920	Authlib is a Python library which builds OAuth and OpenID Connect servers. Prior to version 1.6.5, Authlib's JOSE implementation accepts unbounded JWS/JWT header and signature segments. A remote attacker can craft a token whose base64url#encoded header or signature spans hundreds of megabytes. During verification, Authlib decodes and parses the full input before it is rejected, driving CPU and memory consumption to hostile levels and enabling denial of service. Version 1.6.5 patches the issue. Some temporary workarounds are available. Enforce input size limits before handing tokens to Authlib and/or use application-level throttling to reduce amplification risk.
CVE-2025-61984	ssh in OpenSSH before 10.1 allows control characters in usernames that originate from certain possibly untrusted sources, potentially leading to code execution when a ProxyCommand is used. The untrusted sources are the command line and %-sequence expansion of a configuration file. (A configuration file that provides a complete literal username is not categorized as an untrusted source.)
CVE-2025-61985	ssh in OpenSSH before 10.1 allows the %0' character in an ssh:// URI, potentially leading to code execution when a ProxyCommand is used.
CVE-2025-6199	A flaw was found in the GIF parser of GdKPixbuf's LZW decoder. When an invalid symbol is encountered during decompression, the decoder sets the reported output size to the full buffer length rather than the actual number of written bytes. This logic error results in uninitialized sections of the buffer being included in the output, potentially leaking arbitrary memory contents in the processed image.
CVE-2025-62706	Authlib is a Python library which builds OAuth and OpenID Connect servers. Prior to version 1.6.5, Authlib's JWE zip=DEF path performs unbounded DEFLATE decompression. A very small ciphertext can expand into tens or hundreds of megabytes on decrypt, allowing an attacker who can supply decryptable tokens to exhaust memory and CPU and cause denial of service. This issue has been patched in version 1.6.5. Workarounds for this issue involve rejecting or stripping zip=DEF for inbound JWEs at the application boundary, forking and add a bounded decompression guard via decompressobj().decompress(data, MAX_SIZE)) and returning an error when output exceeds a safe limit, or enforcing strict maximum token sizes and fail fast on oversized inputs; combine with rate limiting.
CVE-2025-6297	It was discovered that dpkg-deb does not properly sanitize directory permissions when extracting a control member into a temporary directory, which is documented as being a safe operation even on untrusted data. This may result in leaving temporary files behind on cleanup. Given automated and repeated execution of dpkg-deb commands on adversarial .deb packages or with well compressible files, placed inside a directory with permissions not allowing removal by a non-root user, this can end up in a DoS scenario due to causing disk quota exhaustion or disk full conditions.
CVE-2025-6395	A NULL pointer dereference flaw was found in the GnuTLS software in _gnutls_figure_common_ciphersuite(). When it reads certain settings from a template file, it can allow an attacker to cause an out-of-bounds (OOB) NULL pointer write, resulting in memory corruption and a denial of service (DoS) that could crash the system.
CVE-2025-6442	Ruby WEBrick read_header HTTP Request Smuggling Vulnerability. This vulnerability allows remote attackers to smuggle arbitrary HTTP requests on affected installations of Ruby WEBrick. This issue is exploitable when the product is deployed behind an HTTP proxy that fulfills specific conditions. The specific flaw exists within the read_headers method. The issue results from the inconsistent parsing of terminators of HTTP headers. An attacker can leverage this vulnerability to smuggle arbitrary HTTP requests. Was ZDI-CAN-21876.
CVE-2025-6965	There exists a vulnerability in SQLite versions before 3.50.2 where the number of aggregate terms could exceed the number of columns available. This could lead to a memory corruption issue. We recommend upgrading to version 3.50.2 or above.
CVE-2025-7039	A flaw was found in glib. An integer overflow during temporary file creation leads to an out-of-bounds memory access, allowing an attacker to potentially perform path traversal or access private temporary file content by creating symbolic links. This vulnerability allows a local attacker to manipulate file paths and access unauthorized data. The core issue stems from insufficient validation of file path lengths during temporary file operations.
CVE-2025-7345	A flaw exists in gdk#pixbuf within the gdk_pixbuf__jpeg_image_load_increment function (io-jpeg.c) and in glib's g_base64_encode_step (glib/gbase64.c). When processing maliciously crafted JPEG images, a heap buffer overflow can occur during Base64 encoding, allowing out-of-bounds reads from heap memory, potentially causing application crashes or arbitrary code execution.
CVE-2025-7545	A vulnerability classified as problematic was found in GNU Binutils 2.45. Affected by this vulnerability is the function copy_section of the file binutils/objcopy.c. The manipulation leads to heap-based buffer overflow. Attacking locally is a requirement. The exploit has been disclosed to the public and may be used. The patch is named 08c3cbe5926e4d355b5cb70bbec2b1eeb40c2944. It is recommended to apply a patch to fix this issue.

CVE ID	Description
CVE-2025-7546	A vulnerability, which was classified as problematic, has been found in GNU Binutils 2.45. Affected by this issue is the function <code>bfd_elf_set_group_contents</code> of the file <code>bfd/elf.c</code> . The manipulation leads to out-of-bounds write. It is possible to launch the attack on the local host. The exploit has been disclosed to the public and may be used. The name of the patch is <code>41461010eb7c79fee7a9d5f6209accdaac66cc6b</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-7700	A flaw was found in FFmpeg's ALS audio decoder, where it does not properly check for memory allocation failures. This can cause the application to crash when processing certain malformed audio files. While it does not lead to data theft or system control, it can be used to disrupt services and cause a denial of service.
CVE-2025-7709	An integer overflow exists in the FTS5 https://sqlite.org/fts5.html extension. It occurs when the size of an array of tombstone pointers is calculated and truncated into a 32-bit integer. A pointer to partially controlled data can then be written out of bounds.
CVE-2025-8058	The <code>regcomp</code> function in the GNU C library version from 2.4 to 2.41 is subject to a double free if some previous allocation fails. It can be accomplished either by a <code>malloc</code> failure or by using an interposed <code>malloc</code> that injects random <code>malloc</code> failures. The double free can allow buffer manipulation depending of how the <code>regex</code> is constructed. This issue affects all architectures and ABIs supported by the GNU C library.
CVE-2025-8114	A flaw was found in <code>libssh</code> , a library that implements the SSH protocol. When calculating the session ID during the key exchange (KEX) process, an allocation failure in cryptographic functions may lead to a NULL pointer dereference. This issue can cause the client or server to crash.
CVE-2025-8176	A vulnerability was found in <code>LibTIFF</code> up to 4.7.0. It has been declared as critical. This vulnerability affects the function <code>get_histogram</code> of the file <code>tools/tiffmedian.c</code> . The manipulation leads to use after free. The attack needs to be approached locally. The exploit has been disclosed to the public and may be used. The patch is identified as <code>fe10872e53efba9cc36c66ac4ab3b41a839d5172</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-8177	A vulnerability was found in <code>LibTIFF</code> up to 4.7.0. It has been rated as critical. This issue affects the function <code>setrow</code> of the file <code>tools/thumbnail.c</code> . The manipulation leads to buffer overflow. An attack has to be approached locally. The patch is named <code>e8c9d6c616b19438695fd829e58ae4fde5bfb22</code> . It is recommended to apply a patch to fix this issue. This vulnerability only affects products that are no longer supported by the maintainer.
CVE-2025-8194	There is a defect in the CPython "tarfile" module affecting the "TarFile" extraction and entry enumeration APIs. The tar implementation would process tar archives with negative offsets without error, resulting in an infinite loop and deadlock during the parsing of maliciously crafted tar archives. This vulnerability can be mitigated by including the following patch after importing the "tarfile" module: https://gist.github.com/sethmlarson/1716ac5b82b73dbcbf23ad2eff8b33e1
CVE-2025-8225	A vulnerability was found in GNU Binutils 2.44 and classified as problematic. This issue affects the function <code>process_debug_info</code> of the file <code>binutils/dwarf.c</code> of the component DWARF Section Handler. The manipulation leads to memory leak. Attacking locally is a requirement. The identifier of the patch is <code>e51fdff7d2e538c0e5accdd65649ac68e6e0ddd4</code> . It is recommended to apply a patch to fix this issue.
CVE-2025-8277	A flaw was found in <code>libssh</code> 's handling of key exchange (KEX) processes when a client repeatedly sends incorrect KEX guesses. The library fails to free memory during these rekey operations, which can gradually exhaust system memory. This issue can lead to crashes on the client side, particularly when using <code>libgcrypt</code> , which impacts application stability and availability.
CVE-2025-8291	The 'zipfile' module would not check the validity of the ZIP64 End of Central Directory (EOCD) Locator record offset value would not be used to locate the ZIP64 EOCD record, instead the ZIP64 EOCD record would be assumed to be the previous record in the ZIP archive. This could be abused to create ZIP archives that are handled differently by the 'zipfile' module compared to other ZIP implementations. Remediation maintains this behavior, but checks that the offset specified in the ZIP64 EOCD Locator record matches the expected value.
CVE-2025-8534	A vulnerability classified as problematic was found in <code>libtiff</code> 4.6.0. This vulnerability affects the function <code>PS_Lvl2page</code> of the file <code>tools/tiff2ps.c</code> of the component <code>tiff2ps</code> . The manipulation leads to null pointer dereference. It is possible to launch the attack on the local host. The complexity of an attack is rather high. The exploitation appears to be difficult. The exploit has been disclosed to the public and may be used. The name of the patch is <code>6ba36f159fd396ad11bf6b7874554197736ecc8b</code> . It is recommended to apply a patch to fix this issue. One of the maintainers explains, that "[t]his error only occurs if <code>DEFER_STRILE_LOAD</code> (<code>defer-strile-load:BOOL=ON</code>) or <code>TIFFOpen</code> (<code>.. "rD"</code>) option is used."
CVE-2025-8556	A flaw was found in <code>CIRCL</code> 's implementation of the FourQ elliptic curve. This vulnerability allows an attacker to compromise session security via low-order point injection and incorrect point validation during Diffie-Hellman key exchange.

CVE ID	Description
CVE-2025-8677	Querying for records within a specially crafted zone containing certain malformed DNSKEY records can lead to CPU exhaustion. This issue affects BIND 9 versions 9.18.0 through 9.18.39, 9.20.0 through 9.20.13, 9.21.0 through 9.21.12, 9.18.11-S1 through 9.18.39-S1, and 9.20.9-S1 through 9.20.13-S1.
CVE-2025-8713	PostgreSQL optimizer statistics allow a user to read sampled data within a view that the user cannot access. Separately, statistics allow a user to read sampled data that a row security policy intended to hide. PostgreSQL maintains statistics for tables by sampling data available in columns; this data is consulted during the query planning process. Prior to this release, a user could craft a leaky operator that bypassed view access control lists (ACLs) and bypassed row security policies in partitioning or table inheritance hierarchies. Reachable statistics data notably included histograms and most-common-values lists. CVE-2017-7484 and CVE-2019-10130 intended to close this class of vulnerability, but this gap remained. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.
CVE-2025-8714	Untrusted data inclusion in pg_dump in PostgreSQL allows a malicious superuser of the origin server to inject arbitrary code for restore-time execution as the client operating system account running psql to restore the dump, via psql meta-commands. pg_dumpall is also affected. pg_restore is affected when used to generate a plain-format dump. This is similar to MySQL CVE-2024-21096. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected.
CVE-2025-8715	Improper neutralization of newlines in pg_dump in PostgreSQL allows a user of the origin server to inject arbitrary code for restore-time execution as the client operating system account running psql to restore the dump, via psql meta-commands inside a purpose-crafted object name. The same attacks can achieve SQL injection as a superuser of the restore target server. pg_dumpall, pg_restore, and pg_upgrade are also affected. Versions before PostgreSQL 17.6, 16.10, 15.14, 14.19, and 13.22 are affected. Versions before 11.20 are unaffected. CVE-2012-0868 had fixed this class of problem, but version 11.20 reintroduced it.
CVE-2025-8851	A vulnerability was determined in LibTIFF up to 4.5.1. Affected by this issue is the function readSeparateStripsetoBuffer of the file tools/tiffcrop.c of the component tiffcrop. The manipulation leads to stack-based buffer overflow. Local access is required to approach this attack. The patch is identified as 8a7a48d7a645992ca83062b3a1873c951661e2b3. It is recommended to apply a patch to fix this issue.
CVE-2025-8869	When extracting a tar archive pip may not check symbolic links point into the extraction directory if the tarfile module doesn't implement PEP 706. Note that upgrading pip to a "fixed" version for this vulnerability doesn't fix all known vulnerabilities that are remediated by using a Python version that implements PEP 706. Note that this is a vulnerability in pip's fallback implementation of tar extraction for Python versions that don't implement PEP 706 and therefore are not secure to all vulnerabilities in the Python 'tarfile' module. If you're using a Python version that implements PEP 706 then pip doesn't use the "vulnerable" fallback code. Mitigations include upgrading to a version of pip that includes the fix, upgrading to a Python version that implements PEP 706 (Python >=3.9.17, >=3.10.12, >=3.11.4, or >=3.12), applying the linked patch, or inspecting source distributions (sdist) before installation as is already a best-practice.
CVE-2025-8916	Allocation of Resources Without Limits or Throttling vulnerability in Legion of the Bouncy Castle Inc. BC Java bcpxix on All (API modules), Legion of the Bouncy Castle Inc. BC Java bcprov on All (API modules), Legion of the Bouncy Castle Inc. BCPKIX FIPS bcpxix-fips on All (API modules) allows Excessive Allocation. This vulnerability is associated with program files https://github.com/bcgit/bc-java/blob/main/pkix/src/main/java/org/bouncycastle/pkix/jcajce/PKIXCertPathReviewer.Java , https://github.com/bcgit/bc-java/blob/main/prov/src/main/java/org/bouncycastle/x509/PKIXCertPathReviewer.Java . This issue affects BC Java: from 1.44 through 1.78; BC Java: from 1.44 through 1.78; BCPKIX FIPS: from 1.0.0 through 1.0.7, from 2.0.0 through 2.0.7.
CVE-2025-8941	A flaw was found in linux-pam. The pam_namespace module may improperly handle user-controlled paths, allowing local users to exploit symlink attacks and race conditions to elevate their privileges to root. This CVE provides a "complete" fix for CVE-2025-6020.
CVE-2025-8961	A vulnerability was identified in LibTIFF 4.7.0. This issue affects the function May of the file tiffcrop.c of the component tiffcrop. The manipulation leads to memory corruption. The attack needs to be approached locally. The exploit has been disclosed to the public and may be used.
CVE-2025-9086	1. A cookie is set using the `secure` keyword for `https://target` 2. curl is redirected to or otherwise made to speak with `http://target` (same hostname, but using clear text HTTP) using the same cookie set 3. The same cookie name is set - but with just a slash as path (`path=/`). Since this site is not secure, the cookie *should* just be ignored. 4. A bug in the path comparison logic makes curl read outside a heap buffer boundary The bug either causes a crash or it potentially makes the comparison come to the wrong conclusion and lets the clear-text site override the contents of the secure cookie, contrary to expectations and depending on the memory contents immediately following the single-byte allocation that holds the path. The presumed and correct behavior would be to plainly ignore the second set of the cookie since it was already set as secure on a secure host so overriding it on an insecure host should not be okay.

CVE ID	Description
CVE-2025-9165	A flaw has been found in LibTIFF 4.7.0. This affects the function <code>_TIFFmallocExt/_TIFFCheckRealloc/TIFFHashSetNew/InitCCITTFax3</code> of the file <code>tools/tiffcmp.c</code> of the component <code>tiffcmp</code> . Executing manipulation can lead to memory leak. The attack is restricted to local execution. This attack is characterized by high complexity. It is indicated that the exploitability is difficult. The exploit has been published and may be used. There is ongoing doubt regarding the real existence of this vulnerability. This patch is called <code>ed141286a37f6e5ddafb5069347ff5d587e7a4e0</code> . It is best practice to apply a patch to resolve this issue. A researcher disputes the security impact of this issue, because "this is a memory leak on a command line tool that is about to exit anyway". In the reply the project maintainer declares this issue as "a simple 'bug' when leaving the command line tool and (...) not a security issue at all".
CVE-2025-9230	Issue summary: An application trying to decrypt CMS messages encrypted using password based encryption can trigger an out-of-bounds read and write. Impact summary: This out-of-bounds read may trigger a crash which leads to Denial of Service for an application. The out-of-bounds write can cause a memory corruption which can have various consequences including a Denial of Service or Execution of attacker-supplied code. Although the consequences of a successful exploit of this vulnerability could be severe, the probability that the attacker would be able to perform it is low. Besides, password based (PWR) encryption support in CMS messages is very rarely used. For that reason the issue was assessed as Moderate severity according to our Security Policy. The FIPS modules in 3.5, 3.4, 3.3, 3.2, 3.1 and 3.0 are not affected by this issue, as the CMS implementation is outside the OpenSSL FIPS module boundary.
CVE-2025-9231	Issue summary: A timing side-channel which could potentially allow remote recovery of the private key exists in the SM2 algorithm implementation on 64 bit ARM platforms. Impact summary: A timing side-channel in SM2 signature computations on 64 bit ARM platforms could allow recovering the private key by an attacker.. While remote key recovery over a network was not attempted by the reporter, timing measurements revealed a timing signal which may allow such an attack. OpenSSL does not directly support certificates with SM2 keys in TLS, and so this CVE is not relevant in most TLS contexts. However, given that it is possible to add support for such certificates via a custom provider, coupled with the fact that in such a custom provider context the private key may be recoverable via remote timing measurements, we consider this to be a Moderate severity issue. The FIPS modules in 3.5, 3.4, 3.3, 3.2, 3.1 and 3.0 are not affected by this issue, as SM2 is not an approved algorithm.
CVE-2025-9232	Issue summary: An application using the OpenSSL HTTP client API functions may trigger an out-of-bounds read if the 'no_proxy' environment variable is set and the host portion of the authority component of the HTTP URL is an IPv6 address. Impact summary: An out-of-bounds read can trigger a crash which leads to Denial of Service for an application. The OpenSSL HTTP client API functions can be used directly by applications but they are also used by the OCSP client functions and CMP (Certificate Management Protocol) client implementation in OpenSSL. However the URLs used by these implementations are unlikely to be controlled by an attacker. In this vulnerable code the out of bounds read can only trigger a crash. Furthermore the vulnerability requires an attacker-controlled URL to be passed from an application to the OpenSSL function and the user has to have a 'no_proxy' environment variable set. For the aforementioned reasons the issue was assessed as Low severity. The vulnerable code was introduced in the following patch releases: 3.0.16, 3.1.8, 3.2.4, 3.3.3, 3.4.0 and 3.5.0. The FIPS modules in 3.5, 3.4, 3.3, 3.2, 3.1 and 3.0 are not affected by this issue, as the HTTP client implementation is outside the OpenSSL FIPS module boundary.
CVE-2025-9301	A vulnerability was determined in <code>cmake 4.1.20250725-gb5cce23</code> . This affects the function <code>cmForEachFunctionBlocker::ReplayItems</code> of the file <code>cmForEachCommand.cxx</code> . This manipulation causes reachable assertion. The attack needs to be launched locally. The exploit has been publicly disclosed and may be utilized. Patch name: <code>37e27f71bc356d880c908040cd0cb68fa2c371b8</code> . It is suggested to install a patch to address this issue.
CVE-2025-9714	Uncontrolled recursion in XPath evaluation in <code>libxml2</code> up to and including version 2.9.14 allows a local attacker to cause a stack overflow via crafted expressions. XPath processing functions <code>`xmlXPathRunEval`</code> , <code>`xmlXPathCtxtCompile`</code> , and <code>`xmlXPathEvalExpr`</code> were resetting recursion depth to zero before making potentially recursive calls. When such functions were called recursively this could allow for uncontrolled recursion and lead to a stack overflow. These functions now preserve recursion depth across recursive calls, allowing recursion depth to be controlled.
CVE-2025-9900	A flaw was found in <code>Libtiff</code> . This vulnerability is a "write-what-where" condition, triggered when the library processes a specially crafted TIFF image file. By providing an abnormally large image height value in the file's metadata, an attacker can trick the library into writing attacker-controlled color data to an arbitrary memory location. This memory corruption can be exploited to cause a denial of service (application crash) or to achieve arbitrary code execution with the permissions of the user.
CVE-2025-9951	A heap-buffer-overflow write exists in <code>jpeg2000dec FFmpeg</code> which allows an attacker to potentially gain remote code execution or cause denial of service via the channel definition <code>cdef atom</code> of JPEG2000.
DLA-3859-1	systemd - security update

CVE ID	Description
DLA-3875-1	gnutls28 - security update
DLA-3910-1	e2fsprogs - security update
DLA-3926-1	perl - security update
DLA-3930-1	libsepol - security update
DLA-3942-1	openssl - security update
DLA-3972-1	tzdata - new timezone database
DLA-4061-1	libtasn1-6 - security update
DLA-4063-1	gnutls28 - security update
DLA-4065-1	krb5 - security update
DLA-4085-1	tzdata - new timezone database
DLA-4105-1	tzdata - new timezone database
DLA-4130-1	shadow - security update
DLA-4143-1	glibc - security update
DLA-4176-1	openssl - security update
DLA-4181-1	glibc - security update
DLA-4195-1	krb5 - security update
DLA-4259-1	systemd - security update
DLA-4267-1	gnutls28 - security update
DLA-4306-1	pam - security update
DLA-4321-1	openssl - security update
DSA-5514-1	glibc - security update
DSA-5650-1	util-linux - security update
DSA-5673-1	glibc - security update
DSA-5678-1	glibc - security update
DSA-5726-1	krb5 - security update
GHSA-2464-8j7c-4cjm	### Summary Use of this library in a security-critical context may result in leaking sensitive information, if used to process sensitive fields. ### Details
GHSA-353f-x4gh-cqq8	## Summary Nokogiri v1.18.9 patches the vendored libxml2 to address CVE-2025-6021, CVE-2025-6170, CVE-2025-49794, CVE-2025-49795, and CVE-2025-49796. ## Impact and severity
GHSA-58qw-p7qm-5rvh	### From the reporter > `XmlParser` is vulnerable to XML external entity (XXE) vulnerability. > XmlParser is being used when parsing Jetty's xml configuration files. An attacker might exploit > this vulnerability in order to achieve SSRF or cause a denial of service. > One possible scenario is importing a (remote) malicious WAR into a Jetty's server, while the > WAR includes a malicious web.xml. ### Impact There are no circumstances in a normally deployed Jetty server where potentially hostile XML is given to the XmlParser class without the attacker already having arbitrary access to the server. I.e. in order to exploit `XmlParser` the attacker would already have the ability to deploy and execute hostile code. Specifically, Jetty has no protection against malicious web application and potentially hostile web applications should only be run on an isolated virtualisation. Thus this is not considered a vulnerability of the Jetty server itself, as any such usage of the jetty XmlParser is equally vulnerable as a direct usage of the JVM supplied SAX parser. No CVE will be allocated to this advisory. However, any direct usage of the `XmlParser` class by an application may be vulnerable. The impact would greatly depend on how the application uses `XmlParser`, but it could be a denial of service due to large entity expansion, or possibly the revealing local files if the XML results are accessible remotely. ### Patches Ability to configure the SAXParserFactory to fit the needs of your particular XML parser implementation have been merged as part of PR #10067 ### Workarounds Don't use `XmlParser` to parse data from users.
GHSA-5w6v-399v-w3cc	## Summary Nokogiri v1.18.8 upgrades its dependency libxml2 to [v2.13.8](https://gitlab.gnome.org/GNOME/libxml2/-/releases/v2.13.8). libxml2 v2.13.8 addresses:

CVE ID	Description
GHSA-fv92-fjc5-jj9h	### Summary Use of this library in a security-critical context may result in leaking sensitive information, if used to process sensitive fields. ### Details
GHSA-ggpf-24jw-3fcw	## Description https://github.com/vllm-project/vllm/security/advisories/GHSA-rh4j-5rhw-hr54 reported a vulnerability where loading a malicious model could result in code execution on the vllm host. The fix applied to specify `weights_only=True` to calls to `torch.load()` did not solve the problem prior to PyTorch 2.6.0. PyTorch has issued a new CVE about this problem: https://github.com/advisories/GHSA-53q9-r3pm-6pq6
GHSA-h4gh-qq45-vh27	pyca/cryptography's wheels include a statically linked copy of OpenSSL. The versions of OpenSSL included in cryptography 37.0.0-43.0.0 are vulnerable to a security issue. More details about the vulnerability itself can be found in https://openssl-library.org/news/secadv/20240903.txt . If you are building cryptography source ("sdist") then you are responsible for upgrading your copy of OpenSSL. Only users installing from wheels built by the cryptography project (i.e., those distributed on PyPI) need to update their cryptography versions.
GHSA-hf3c-wxg2-49q9	### Impact This report is to highlight a vulnerability in XGrammar, a library used by the structured output feature in vLLM. The XGrammar advisory is here: https://github.com/mlc-ai/xgrammar/security/advisories/GHSA-389x-67px-mjg3 The [xgrammar](https://xgrammar.mlc.ai/docs/) library is the default backend used by vLLM to support structured output (a.k.a. guided decoding). Xgrammar provides a required, built-in cache for its compiled grammars stored in RAM. xgrammar is available by default through the OpenAI compatible API server with both the V0 and V1 engines.
GHSA-m425-mq94-257g	### Impact In affected releases of gRPC-Go, it is possible for an attacker to send HTTP/2 requests, cancel them, and send subsequent requests, which is valid by the HTTP/2 protocol, but would cause the gRPC-Go server to launch more concurrent method handlers than the configured maximum stream limit. ### Patches This vulnerability was addressed by #6703 and has been included in patch releases: 1.56.3, 1.57.1, 1.58.3. It is also included in the latest release, 1.59.0. Along with applying the patch, users should also ensure they are using the `grpc.MaxConcurrentStreams` server option to apply a limit to the server's resources used for any single connection. ### Workarounds None. ### References #6703
GHSA-mrxw-mxhj-p664	## Summary Nokogiri v1.18.4 upgrades its dependency libxslt to [v1.1.43](https://gitlab.gnome.org/GNOME/libxslt/-/releases/v1.1.43). libxslt v1.1.43 resolves:
GHSA-r4pg-vg54-wxx4	### Impact cert-manager packages which call the standard library `pem.Decode()` function can take a long time to process specially crafted invalid PEM data. If an attacker is able to modify PEM data which cert-manager reads (e.g. in a Secret resource), they may be able to use large amounts of CPU in the cert-manager controller pod to effectively create a denial-of-service (DoS) vector for cert-manager in the cluster. Secrets are limited in size to [1MiB](https://kubernetes.io/docs/concepts/configuration/secret/#restriction-data-size), which reduces the impact of this issue; it was discovered through an ~856kB fuzz test input which causes `pem.Decode` to take roughly 750ms to reject the input on an M2 Max Macbook Pro. By way of comparison, a valid PEM-encoded 4096-bit RSA key takes roughly 70µs to parse on the same machine. Given the required size of PEM data needed to present a realistic DoS vector, an attacker would need to create or insert many different large sized resources in the cluster, and so the best secondary defense is to ensure that sensible limits are placed via RBAC. This issue affects all versions of cert-manager to have been released since at least v0.1.0 (since `pem.Decode` is core functionality for cert-manager). All [supported releases](https://cert-manager.io/docs/releases/) are patched. ### Patches The fixed versions are v1.16.2, v1.15.4 and v1.12.14. - master branch: https://github.com/cert-manager/cert-manager/pull/7400 - release-1.16 branch: https://github.com/cert-manager/cert-manager/pull/7401 - release-1.15 branch: https://github.com/cert-manager/cert-manager/pull/7402 - release-1.12 branch: https://github.com/cert-manager/cert-manager/pull/7403 ### Workarounds Ensure that RBAC is scoped correctly in your cluster. If a user is able to modify resources containing PEM data to be able to exploit this, it's like that those permissions are a bigger security threat than this issue - especially for Secret resources. ### References - Upstream issue: https://github.com/golang/go/issues/50116 - Similar issue: https://github.com/sigstore/sigstore/issues/198 - Google OSSFuzz: https://issues.oss-fuzz.com/issues/376728466
GHSA-vvfq-8hwr-qm4m	## Summary Nokogiri v1.18.3 upgrades its dependency libxml2 to [v2.13.6](https://gitlab.gnome.org/GNOME/libxml2/-/releases/v2.13.6). libxml2 v2.13.6 addresses:
GHSA-w476-p2h3-79g9	### Impact In versions 0.9.4 and earlier of uv, tar archives containing PAX headers with file size overrides were not handled properly. As a result, an attacker could contrive a source distribution (as a tar archive) that would extract differently when installed via uv versus other Python package installers. The underlying parsing differential here originates with astral-tokio-tar, which disclosed this vulnerability as CVE-2025-62518.

CVE ID	Description
GHSA-xpw8-rcwv-8f8p	A client might overload the server by issue frequent RST frames. This can cause a massive amount of load on the remote system and so cause a DDOS attack. ### Impact This is a DDOS attack, any http2 server is affected and so you should update as soon as possible. ### Patches This is patched in version 4.1.100.Final. ### Workarounds A user can limit the amount of RST frames that are accepted per connection over a timeframe manually using either an own `Http2FrameListener` implementation or an `ChannelInboundHandler` implementation (depending which http2 API is used). ### References - https://www.cve.org/CVERecord?id=CVE-2023-44487 - https://blog.cloudflare.com/technical-breakdown-http2-rapid-reset-ddos-attack/ - https://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps/
NSWG-ECO-328	Jquery is a javascript library for DOM traversal and manipulation, event handling, animation, and Ajax. When text/javascript responses are received from cross-origin ajax requests not containing the option `dataType`, the result is executed in `jQuery.globalEval` potentially allowing an attacker to execute arbitrary code on the origin.
NSWG-ECO-329	jQuery is a javascript library for DOM manipulation. jQuery's main method in affected versions contains an unreliable way of detecting whether the input to the `jQuery(strInput)` function is intended to be a selector or HTML. For example, this code would be parsed as a selector, executing the code in the `onerror` attribute: `` `\$("#log").html( \$("element{attribute=<img src=\"x\" onerror=\"alert(1)\" />"]").html() ); `` The fix in v1.9.0 updates a regular expression for detecting whether the input is HTML or a selector. HTML input must now explicitly start with `<`, rather than previously assuming that the input was HTML if the string contained `<` anywhere.
RHBA-2022:2065	For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.6 Release Notes linked from the References section.
RHBA-2024:5736	The ca-certificates package contains a set of Certificate Authority (CA) certificates chosen by the Mozilla Foundation for use with the Internet Public Key Infrastructure (PKI).
RHBA-2024:6680	Network Security Services (NSS) is a set of libraries designed to support the cross-platform development of security-enabled client and server applications.
RHEA-2023:3850	Kerberos is a network authentication system, which can improve the security of your network by eliminating the insecure practice of sending passwords over the network in unencrypted form. It allows clients and servers to authenticate to each other with the help of a trusted third party, the Kerberos key distribution center (KDC).
RHSA-2022:1065	OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, as well as a full-strength general-purpose cryptography library.
RHSA-2022:1537	The gzip packages contain the gzip (GNU zip) data compression utility. gzip is used to compress regular files. It replaces them with files containing the .gz extension, while retaining ownership modes, access, and modification times.
RHSA-2022:1552	An update for vim is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Vim (Vi IMproved) is an updated and improved version of the vi editor. Security Fix(es): * vim: use after free in utf_ptr2char (CVE-2022-1154) This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. vim: use after free in utf_ptr2char
RHSA-2022:1642	The zlib packages provide a general-purpose lossless data compression library that is used by many different programs.
RHSA-2022:1891	The libpq package provides the PostgreSQL client library, which allows client programs to connect to PostgreSQL servers.

CVE ID	Description
RHSA-2022:1986	An update for python3 is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems. Security Fix(es): * python: ftplib should not use the host from the PASV response (CVE-2021-4189) * python: urllib: HTTP client possible infinite loop on a 100 Continue response (CVE-2021-3737) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.6 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. python: urllib: HTTP client possible infinite loop on a 100 Continue response python: ftplib should not use the host from the PASV response
RHSA-2022:2031	libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2022:2092	An update for bind is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly. The following packages have been upgraded to a later upstream version: bind (9.11.36). (BZ#2013993) Security Fix(es): * bind: Lame cache can be abused to severely degrade resolver performance (CVE-2021-25219) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.6 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. bind: Lame cache can be abused to severely degrade resolver performance
RHSA-2022:2201	The rsync utility enables the users to copy and synchronize files locally or across a network. Synchronization with rsync is fast because rsync only sends the differences in files over the network instead of sending whole files. The rsync utility is also used as a mirroring tool.
RHSA-2022:4807	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2022:4991	XZ Utils is an integrated collection of user-space file compression utilities based on the Lempel-Ziv-Markov chain algorithm (LZMA), which performs lossless data compression. The algorithm provides a high compression ratio while keeping the decompression time short.
RHSA-2022:5311	The libcrypto library provides general-purpose implementations of various cryptographic algorithms.
RHSA-2022:5313	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2022:5314	Expat is a C library for parsing XML documents.
RHSA-2022:5317	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2022:5319	An update for vim is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Vim (Vi IMproved) is an updated and improved version of the vi editor. Security Fix(es): * vim: heap buffer overflow (CVE-2022-1621) * vim: buffer over-read (CVE-2022-1629) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. vim: heap buffer overflow vim: buffer over-read
RHSA-2022:5809	The pcre2 package contains a new generation of the Perl Compatible Regular Expression libraries for implementing regular expression pattern matching using the same syntax and semantics as Perl.

CVE ID	Description
RHSAs-2022:5813	An update for vim is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Vim (Vi IMproved) is an updated and improved version of the vi editor. Security Fix(es): * vim: Out-of-bounds Write (CVE-2022-1785) * vim: out-of-bounds write in vim_regsub_both() in regexp.c (CVE-2022-1897) * vim: buffer over-read in utf_ptr2char() in mbyte.c (CVE-2022-1927) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. vim: Out-of-bounds Write vim: out-of-bounds write in vim_regsub_both() in regexp.c vim: buffer over-read in utf_ptr2char() in mbyte.c
RHSAs-2022:5818	OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, as well as a full-strength general-purpose cryptography library.
RHSAs-2022:6159	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSAs-2022:6180	The rsync utility enables the users to copy and synchronize files locally or across a network. Synchronization with rsync is fast because rsync only sends the differences in files over the network instead of sending whole files. The rsync utility is also used as a mirroring tool.
RHSAs-2022:6206	The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelism capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, and keeps track of processes using Linux cgroups. In addition, it supports snapshotting and restoring of the system state, maintains mount and automount points, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for sysvinit.
RHSAs-2022:6457	An update for python3 is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems. Security Fix(es): * python(mailcap): findmatch() function does not sanitise the second argument (CVE-2015-20107) * python: urllib.parse does not sanitize URLs containing ASCII newline and tabs (CVE-2022-0391) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. python(mailcap): findmatch() function does not sanitise the second argument python: urllib.parse does not sanitize URLs containing ASCII newline and tabs
RHSAs-2022:6463	The GNU Privacy Guard (GnuPG or GPG) is a tool for encrypting data and creating digital signatures, compliant with OpenPGP and S/MIME standards.
RHSAs-2022:6778	An update for bind is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Important. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly. Security Fix(es): * bind: memory leak in ECDSA DNSSEC verification code (CVE-2022-38177) * bind: memory leaks in EdDSA DNSSEC verification code (CVE-2022-38178) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. bind: memory leak in ECDSA DNSSEC verification code bind: memory leaks in EdDSA DNSSEC verification code
RHSAs-2022:6878	Expat is a C library for parsing XML documents.
RHSAs-2022:7089	KSBA (pronounced Kasbah) is a library to make X.509 certificates as well as the CMS easily accessible by other applications. Both specifications are building blocks of S/MIME and TLS.

CVE ID	Description
RHSA-2022:7105	The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic algorithms and protocols such as SSL, TLS, and DTLS.
RHSA-2022:7106	The zlib packages provide a general-purpose lossless data compression library that is used by many different programs.
RHSA-2022:7108	SQLite is a C library that implements an SQL database engine. A large subset of SQL92 is supported. A complete database is stored in a single disk file. The API is designed for convenience and ease of use. Applications that link against SQLite can enjoy the power and flexibility of an SQL database without the administrative hassles of supporting a separate database server.
RHSA-2022:7128	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2022:7704	WebKitGTK is the port of the portable web rendering engine WebKit to the GTK platform.
RHSA-2022:7715	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2022:7720	An update for e2fsprogs is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The e2fsprogs packages provide a number of utilities for creating, checking, modifying, and correcting the ext2, ext3, and ext4 file systems. Security Fix(es): * e2fsprogs: out-of-bounds read/write via crafted filesystem (CVE-2022-1304) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.7 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. e2fsprogs: out-of-bounds read/write via crafted filesystem
RHSA-2022:7790	The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly.
RHSA-2022:7793	The rsync utility enables the users to copy and synchronize files locally or across a network. Synchronization with rsync is fast because rsync only sends the differences in files over the network instead of sending whole files. The rsync utility is also used as a mirroring tool.
RHSA-2023:0096	D-Bus is a system for sending messages between applications. It is used both for the system-wide message bus service, and as a per-user-login-session messaging facility.
RHSA-2023:0100	The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelism capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, and keeps track of processes using Linux cgroups. In addition, it supports snapshotting and restoring of the system state, maintains mount and automount points, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for sysvinit.
RHSA-2023:0103	Expat is a C library for parsing XML documents.
RHSA-2023:0110	SQLite is a C library that implements an SQL database engine. A large subset of SQL92 is supported. A complete database is stored in a single disk file. The API is designed for convenience and ease of use. Applications that link against SQLite can enjoy the power and flexibility of an SQL database without the administrative hassles of supporting a separate database server.
RHSA-2023:0116	A library that provides Abstract Syntax Notation One (ASN.1, as specified by the X.680 ITU-T recommendation) parsing and structures management, and Distinguished Encoding Rules (DER, as per X.690) encoding and decoding functions.
RHSA-2023:0173	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2023:0625	KSBA (pronounced Kasbah) is a library to make X.509 certificates as well as the CMS easily accessible by other applications. Both specifications are building blocks of S/MIME and TLS.
RHSA-2023:0833	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2023:0835	The python-setuptools package provides a collection of enhancements to Python distribution utilities allowing convenient building and distribution of Python packages.

CVE ID	Description
RHSA-2023:0837	The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelism capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, and keeps track of processes using Linux cgroups. In addition, it supports snapshotting and restoring of the system state, maintains mount and automount points, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for sysvinit.
RHSA-2023:0842	The GNU tar program can save multiple files in an archive and restore files from an archive.
RHSA-2023:1140	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2023:1405	OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, as well as a full-strength general-purpose cryptography library.
RHSA-2023:1569	The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic algorithms and protocols such as SSL, TLS, and DTLS.
RHSA-2023:1930	GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news.
RHSA-2023:2963	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2023:3002	The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly.
RHSA-2023:3018	The libarchive programming library can create and read several different streaming archive formats, including GNU tar, cpio, and ISO 9660 CD-ROM images. Libarchive is used notably in the bsdtar utility, scripting language bindings such as python-libarchive, and several popular desktop file managers.
RHSA-2023:3042	GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news.
RHSA-2023:3104	GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news.
RHSA-2023:3106	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2023:3591	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2023:3837	The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelism capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, and keeps track of processes using Linux cgroups. In addition, it supports snapshotting and restoring of the system state, maintains mount and automount points, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for sysvinit.
RHSA-2023:3839	libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2023:3840	SQLite is a C library that implements an SQL database engine. A large subset of SQL92 is supported. A complete database is stored in a single disk file. The API is designed for convenience and ease of use. Applications that link against SQLite can enjoy the power and flexibility of an SQL database without the administrative hassles of supporting a separate database server.

CVE ID	Description
RHSA-2023:4102	An update for bind is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Important. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly. Security Fix(es): * bind: named's configured cache size limit can be significantly exceeded (CVE-2023-2828) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A vulnerability was found in BIND. The effectiveness of the cache-cleaning algorithm used in named can be severely diminished by querying the resolver for specific RRsets in a certain order, effectively allowing the configured max-cache-size limit to exceed significantly.
RHSA-2023:4498	D-Bus is a system for sending messages between applications. It is used both for the system-wide message bus service, and as a per-user-login-session messaging facility.
RHSA-2023:4520	The python-requests package contains a library designed to make HTTP requests easy for developers.
RHSA-2023:4523	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2023:4524	Libcap is a library for getting and setting POSIX.1e (formerly POSIX 6) draft 15 capabilities.
RHSA-2023:4529	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2023:4535	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2023:4539	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2023:4706	The subscription-manager packages provide programs and libraries to allow users to manage subscriptions and yum repositories from the Red Hat entitlement platform.
RHSA-2023:5249	The ncurses (new curses) library routines are a terminal-independent method of updating character screens with reasonable optimization. The ncurses packages contain support utilities including a terminfo compiler tic, a decompiler infocmp, clear, tput, tset, and a termcap conversion tool captoinfo.
RHSA-2023:5252	The dmidecode packages provide utilities for extracting Intel 64 and Intel Itanium hardware information from the system BIOS or Extensible Firmware Interface (EFI), depending on the SMBIOS/DMI standard. This information typically includes system manufacturer, model name, serial number, BIOS version, and asset tag, as well as other details, depending on the manufacturer.
RHSA-2023:5455	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2023:5474	The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly.
RHSA-2023:5837	nghttp2 contains the Hypertext Transfer Protocol version 2 (HTTP/2) client, server, and proxy programs as well as a library implementing the HTTP/2 protocol in C.
RHSA-2023:5997	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2023:6944	The protobuf-c packages provide C bindings for Google's Protocol Buffers.
RHSA-2023:7016	The libpq package provides the PostgreSQL client library, which allows client programs to connect to PostgreSQL servers.

CVE ID	Description
RHSA-2023:7083	An update for emacs is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news. Security Fix(es): * emacs: command execution via shell metacharacters (CVE-2022-48337) * emacs: command injection vulnerability in htmlfontify.el (CVE-2022-48339) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.9 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A flaw was found in the Emacs package. This flaw allows attackers to execute commands via shell metacharacters in the name of a source-code file. emacs: command execution via shell metacharacters This vulnerability is only triggered when a local user introduces untrusted input, via a file with a crafted name. For this reason, this flaw has been rated with a Moderate security impact. The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability. A flaw was found in the Emacs package. If a file name or directory name contains shell metacharacters, arbitrary code may be executed. emacs: command injection vulnerability in htmlfontify.el This vulnerability is only triggered when a local user introduces untrusted input, via a file or directory with a crafted name. For this reason, this flaw has been rated with a Moderate security impact.
RHSA-2023:7112	The shadow-utils packages include programs for converting UNIX password files to the shadow password format, as well as utilities for managing user and group accounts.
RHSA-2023:7151	An update for python3 is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems. Security Fix(es): * python: tarfile module directory traversal (CVE-2007-4559) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.9 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A flaw was found in the Python tarfile module. Extracting a crafted TAR archive with the tarfile.extract or tarfile.extractall functions could lead to a directory traversal vulnerability, resulting in overwrite of arbitrary files. python: tarfile module directory traversal The Red Hat Product Security has rated this issue as having a Moderate security impact, a future update may address this flaw. More information regarding issue severity can be found here: https://access.redhat.com/security/updates/classification. Versions of `python36:3.6/python36` as shipped with Red Hat Enterprise Linux 8 are marked as 'Not affected' as they just provide "symlinks" to the main `python3` component, which provides the actual interpreter of the Python programming language. The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability.
RHSA-2023:7166	The tpm2-tss packages provide the Intel implementation of the Trusted Platform Module (TPM) 2.0 System API library. This library enables programs to interact with TPM 2.0 devices
RHSA-2023:7174	HTTP::Tiny is a small and simple HTTP/1.1 client written in Perl.

CVE ID	Description
RHSA-2023:7176	An update for python-pip is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. pip is a package management system used to install and manage software packages written in Python. Many packages can be found in the Python Package Index (PyPI). pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python". Security Fix(es): * python: tarfile module directory traversal (CVE-2007-4559) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.9 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A flaw was found in the Python tarfile module. Extracting a crafted TAR archive with the tarfile.extract or tarfile.extractall functions could lead to a directory traversal vulnerability, resulting in overwrite of arbitrary files. python: tarfile module directory traversal The Red Hat Product Security has rated this issue as having a Moderate security impact, a future update may address this flaw. More information regarding issue severity can be found here: https://access.redhat.com/security/updates/classification . Versions of `python36:3.6/python36` as shipped with Red Hat Enterprise Linux 8 are marked as 'Not affected' as they just provide "symlinks" to the main `python3` component, which provides the actual interpreter of the Python programming language. The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability.
RHSA-2023:7177	An update for bind is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly. Security Fix(es): * bind: flooding with UPDATE requests may lead to DoS (CVE-2022-3094) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 8.9 Release Notes linked from the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A flaw was found in Bind, where sending a flood of dynamic DNS updates may cause named to allocate large amounts of memory. This issue may cause named to slow down due to a lack of free memory, resulting in a denial of service (DoS). bind: flooding with UPDATE requests may lead to DoS Exploitation of this vulnerability is limited to trusted clients who are permitted to make dynamic zone changes. The impact on the 'named' service is directly related to the volume of requests being sent in. The service will recover to normal once an attacker stops sending dynamic updates. While a performance impact can be triggered, it is unlikely to result in a crash. The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability.
RHSA-2023:7187	The procps-ng packages contain a set of system utilities that provide system information, including ps, free, skill, pkill, pgrep, snice, tload, top, uptime, vmstat, w, watch, and pwdx.
RHSA-2023:7714	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2023:7790	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2023:7877	OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, as well as a full-strength general-purpose cryptography library.

CVE ID	Description
RHSA-2024:0114	An update for python3 is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems. Security Fix(es): * python: use after free in heappushpop() of heapq module (CVE-2022-48560) * python: DoS when processing malformed Apple Property List files in binary format (CVE-2022-48564) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A use-after-free vulnerability was found in Python via the heappushpop function in the heapq module. This flaw allows an attacker to submit a specially crafted request, causing a service disruption that leads to a denial of service attack. python: use after free in heappushpop() of heapq module Versions of python36:3.6/python36 as shipped with Red Hat Enterprise Linux 8 are marked as 'Not affected' as they just provide "symlinks" to the main python3 component, which provides the actual interpreter of the Python programming language. The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability. A vulnerability was found in the Python core plistlib library within the read_ints() function in the plistlib.py file. In malformed input, the implementation can be manipulated to create an argument for struct.unpack(). This issue can lead to excessive CPU and memory consumption, resulting in a MemError, as it constructs the 'format' argument for unpack(). This flaw allows an attacker to employ a binary plist input, potentially executing a denial of service (DoS) attack by exhausting CPU and RAM resources. python: DoS when processing malformed Apple Property List files in binary format
RHSA-2024:0116	An update for python-urllib3 is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The python-urllib3 package provides the Python HTTP module with connection pooling and file POST abilities. Security Fix(es): * python-urllib3: Cookie request header isn't stripped during cross-origin redirects (CVE-2023-43804) * urllib3: Request body not stripped after redirect from 303 status changes request method to GET (CVE-2023-45803) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A flaw was found in urllib3, a user-friendly HTTP client library for Python. urllib3 doesn't treat the 'Cookie' HTTP header special or provide any helpers for managing cookies over HTTP, which is the responsibility of the user. However, it is possible for a user to specify a 'Cookie' header and unknowingly leak information via HTTP redirects to a different origin if that user doesn't disable redirects explicitly. python-urllib3: Cookie request header isn't stripped during cross-origin redirects The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability. A flaw was found in urllib3, an HTTP client library for Python. urllib3 doesn't remove the HTTP request body when an HTTP redirect response using status 301, 302, or 303, after changing the method in a request from one that could accept a request body such as 'POST' to 'GET', as is required by HTTP RFCs. This issue requires a previously trusted service to become compromised in order to have an impact on confidentiality, therefore, the exploitability of this vulnerability is low. Additionally, many users aren't putting sensitive data in HTTP request bodies; if this is the case, this vulnerability isn't exploitable. urllib3: Request body not stripped after redirect from 303 status changes request method to GET Both of the following conditions must be true to be affected by this vulnerability: 1. Using urllib3 and submitting sensitive information in the HTTP request body such as form data or JSON 2. The origin service is compromised and starts redirecting using 301, 302, or 303 to a malicious peer or the redirected-to service becomes compromised
RHSA-2024:0119	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2024:0155	The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic algorithms and protocols such as SSL, TLS, and DTLS.
RHSA-2024:0253	SQLite is a C library that implements an SQL database engine. A large subset of SQL92 is supported. A complete database is stored in a single disk file. The API is designed for convenience and ease of use. Applications that link against SQLite can enjoy the power and flexibility of an SQL database without the administrative hassles of supporting a separate database server.

CVE ID	Description
RHSA-2024:0256	An update for python3 is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Moderate. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems. Security Fix(es): * python: Parsing errors in email/_parseaddr.py lead to incorrect value in email address part of tuple (CVE-2023-27043) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. The email module of Python through 3.11.3 incorrectly parses e-mail addresses that contain a special character. The wrong portion of an RFC2822 header is identified as the value of the addr-spec. In some applications, an attacker can bypass a protection mechanism in which application access is granted only after verifying receipt of e-mail to a specific domain (e.g., only @company.example.com addresses may be used for signup). This occurs in email/_parseaddr.py in recent versions of Python. python: Parsing errors in email/_parseaddr.py lead to incorrect value in email address part of tuple Versions of python36:3.6/python36 as shipped with Red Hat Enterprise Linux 8 are marked as 'Not affected' as they just provide "symlinks" to the main python3 component, which provides the actual interpreter of the Python programming language. The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability.
RHSA-2024:0627	The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic algorithms and protocols such as SSL, TLS, and DTLS.
RHSA-2024:0628	libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2024:0647	The RPM Package Manager (RPM) is a command-line driven package management system capable of installing, uninstalling, verifying, querying, and updating software packages.
RHSA-2024:0768	The libmaxminddb package contains the MaxMind DB library.
RHSA-2024:0786	Network Security Services (NSS) is a set of libraries designed to support the cross-platform development of security-enabled client and server applications.
RHSA-2024:0956	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2024:0974	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2024:10281	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2024:10379	Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the need to recompile programs to handle authentication.
RHSA-2024:10779	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2024:10785	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2024:10943	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2024:1601	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2024:1610	The "less" utility is a text file browser that resembles "more", but allows users to move backwards in the file as well as forwards. Since "less" does not read the entire input file at startup, it also starts more quickly than ordinary text editors.
RHSA-2024:1615	Expat is a C library for parsing XML documents.

CVE ID	Description
RHSA-2024:1782	An update for bind and dhcp is now available for Red Hat Enterprise Linux 8. Red Hat Product Security has rated this update as having a security impact of Important. A Common Vulnerability Scoring System (CVSS) base score, which gives a detailed severity rating, is available for each vulnerability from the CVE link(s) in the References section. The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly. The Dynamic Host Configuration Protocol (DHCP) is a protocol that allows individual devices on an IP network to get their own network configuration information, including an IP address, a subnet mask, and a broadcast address. The dhcp packages provide a relay agent and ISC DHCP service required to enable and administer DHCP on a network. Security Fix(es): * bind9: Parsing large DNS messages may cause excessive CPU load (CVE-2023-4408) * bind9: KeyTrap - Extreme CPU consumption in DNSSEC validator (CVE-2023-50387) * bind9: Preparing an NSEC3 closest encoder proof can exhaust CPU resources (CVE-2023-50868) For more details about the security issue(s), including the impact, a CVSS score, acknowledgments, and other related information, refer to the CVE page(s) listed in the References section. This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original. A flaw was found in the bind package. This issue may allow a remote attacker with no specific privileges to craft a specially long DNS message leading to an excessive and uncontrolled CPU usage, the server being unavailable, and a Denial of Service. bind9: Parsing large DNS messages may cause excessive CPU load The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability. Processing specially crafted responses coming from DNSSEC-signed zones can lead to uncontrolled CPU usage, leading to a Denial of Service in the DNSSEC-validating resolver side. This vulnerability applies only for systems where DNSSEC validation is enabled. bind9: KeyTrap - Extreme CPU consumption in DNSSEC validator A flaw was found in bind9. By flooding a DNSSEC resolver with responses coming from a DNSSEC-signed zone using NSEC3, an attacker can lead the targeted resolver to a CPU exhaustion, further leading to a Denial of Service on the targeted host. This vulnerability applies only for systems where DNSSEC validation is enabled. bind9: Preparing an NSEC3 closest encoder proof can exhaust CPU resources
RHSA-2024:1784	The gnutls package provide the GNU Transport Layer Security (GnuTLS) library,
RHSA-2024:1825	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2024:2722	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2024:3163	Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the need to recompile programs to handle authentication.
RHSA-2024:3203	The systemd packages contain systemd, a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelism capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, and keeps track of processes using Linux cgroups. In addition, it supports snapshotting and restoring of the system state, maintains mount and automount points, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for sysvinit.
RHSA-2024:3214	The gmp packages contain GNU MP, a library for arbitrary precision arithmetics, signed integers operations, rational numbers, and floating point numbers.
RHSA-2024:3233	libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2024:3268	Kerberos is a network authentication system, which can improve the security of your network by eliminating the insecure practice of sending passwords over the network in unencrypted form. It allows clients and servers to authenticate to each other with the help of a trusted third party, the Kerberos key distribution center (KDC).
RHSA-2024:3344	The glibc packages provide the standard C libraries (libc), POSIX thread
RHSA-2024:3347	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2024:3626	The libxml2 library is a development toolbox providing the implementation of
RHSA-2024:4197	The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.

CVE ID	Description
RHSA-2024:4211	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2024:4247	libuv is a multi-platform support library with a focus on asynchronous I/O.
RHSA-2024:4252	libnghttp2 is a library implementing the Hypertext Transfer Protocol version 2 (HTTP/2) protocol in C.
RHSA-2024:4256	The "less" utility is a text file browser that resembles "more", but allows users to move backwards in the file as well as forwards. Since "less" does not read the entire input file at startup, it also starts more quickly than ordinary text editors.
RHSA-2024:4260	The hsakmt packages include a thunk library for AMD's Heterogeneous System Architecture (HSA) Linux kernel driver (amdkfd).
RHSA-2024:4264	OpenLDAP is an open-source suite of Lightweight Directory Access Protocol (LDAP) applications and development tools. LDAP is a set of protocols used to access and maintain distributed directory information services over an IP network.
RHSA-2024:4265	The Common UNIX Printing System (CUPS) provides a portable printing layer for Linux, UNIX, and similar operating systems.
RHSA-2024:4568	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2024:4720	The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.
RHSA-2024:5079	The libtiff packages contain a library of functions for manipulating Tagged Image File Format (TIFF) files.
RHSA-2024:5193	The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.
RHSA-2024:5299	The wget packages provide the GNU Wget file retrieval utility for HTTP, HTTPS, and FTP protocols.
RHSA-2024:5309	The python-urllib3 package provides the Python HTTP module with connection pooling and file POST abilities.
RHSA-2024:5312	Kerberos is a network authentication system, which can improve the security of your network by eliminating the insecure practice of sending passwords over the network in unencrypted form. It allows clients and servers to authenticate to each other with the help of a trusted third party, the Kerberos key distribution center (KDC).
RHSA-2024:5524	The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly.
RHSA-2024:5530	The python-setuptools package provides a collection of enhancements to Python distribution utilities allowing convenient building and distribution of Python packages.
RHSA-2024:5654	The curl packages provide the libcurl library and the curl utility for downloading files from servers using various protocols, including HTTP, FTP, and LDAP.
RHSA-2024:5814	Node.js is a software development platform for building fast and scalable network applications in the JavaScript programming language.
RHSA-2024:5962	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2024:6000	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2024:6975	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2024:6987	GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news.
RHSA-2024:6989	Expat is a C library for parsing XML documents.
RHSA-2024:7000	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2024:7848	OpenSSL is a toolkit that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, as well as a full-strength general-purpose cryptography library.
RHSA-2024:8117	The java-1.8.0-openjdk packages provide the OpenJDK 8 Java Runtime Environment and the OpenJDK 8 Java Software Development Kit.

CVE ID	Description
RHSA-2024:8124	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2024:8359	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2024:8833	The libtiff packages contain a library of functions for manipulating Tagged Image File Format (TIFF) files.
RHSA-2024:8860	Kerberos is a network authentication system, which can improve the security of your network by eliminating the insecure practice of sending passwords over the network in unencrypted form. It allows clients and servers to authenticate to each other with the help of a trusted third party, the Kerberos key distribution center (KDC).
RHSA-2024:8922	The bzip2 packages contain a freely available, high-quality data compressor. It provides both standalone compression and decompression utilities, as well as a shared library for use with other programs.
RHSA-2024:9502	Expat is a C library for parsing XML documents.
RHSA-2024:9689	The binutils packages provide a collection of binary utilities for the manipulation of object code in various object file formats. It includes the ar, as, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings, strip, and addr2line utilities.
RHSA-2025:0012	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2025:0065	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:0083	The Common UNIX Printing System (CUPS) provides a portable printing layer for Linux, UNIX, and similar operating systems.
RHSA-2025:0325	The rsync utility enables the users to copy and synchronize files locally or across a network. Synchronization with rsync is fast because rsync only sends the differences in files over the network instead of sending whole files. The rsync utility is also used as a mirroring tool.
RHSA-2025:0422	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2025:10027	Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the need to recompile programs to handle authentication.
RHSA-2025:10128	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2025:10669	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:1068	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:10698	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2025:10862	The java-1.8.0-openjdk packages provide the OpenJDK 8 Java Runtime Environment and the OpenJDK 8 Java Software Development Kit.
RHSA-2025:10867	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2025:11030	GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news.
RHSA-2025:11035	The lz4 packages provide support for LZ4, a very fast, lossless compression algorithm that provides compression speeds of 400 MB/s per core and scales with multicore CPUs. It also features an extremely fast decoder that reaches speeds of multiple GB/s per core and typically reaches RAM speed limits on multicore systems.
RHSA-2025:11036	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2025:11298	The kernel packages contain the Linux kernel, the core of any Linux operating system.

CVE ID	Description
RHSA-2025:11327	GLib provides the core application building blocks for libraries and applications written in C. It provides the core object system used in GNOME, the main loop implementation, and a large set of utility functions for strings and common data structures.
RHSA-2025:11455	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:11534	Git is a distributed revision control system with a decentralized architecture. As opposed to centralized version control systems with a client-server model, Git ensures that each working copy of a Git repository is an exact copy with complete revision history. This not only allows the user to work on and contribute to projects without the need to have permission to push the changes to their official repositories, but also makes it possible for the user to work with no network connection.
RHSA-2025:11805	Perl is a high-level programming language that is commonly used for system administration utilities and web programming.
RHSA-2025:11850	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:12010	SQLite is a C library that implements an SQL database engine. A large subset of SQL92 is supported. A complete database is stored in a single disk file. The API is designed for convenience and ease of use. Applications that link against SQLite can enjoy the power and flexibility of an SQL database without the administrative hassles of supporting a separate database server.
RHSA-2025:12450	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2025:1266	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:12748	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2025:12752	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:12876	The ncurses (new curses) library routines are a terminal-independent method of updating character screens with reasonable optimization. The ncurses packages contain support utilities including a terminfo compiler tic, a decompiler infocmp, clear, tput, tset, and a termcap conversion tool captoinfo.
RHSA-2025:12980	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2025:1301	The gcc packages provide compilers for C, C++, Java, Fortran, Objective C, and Ada 95 GNU, as well as related support libraries.
RHSA-2025:13203	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2025:13234	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2025:13315	The gdk-pixbuf2 packages provide an image loading library that can be extended by loadable modules for new image formats. It is used by toolkits such as GTK+ or clutter.
RHSA-2025:1346	The gcc packages provide compilers for C, C++, Java, Fortran, Objective C, and Ada 95 GNU, as well as related support libraries.
RHSA-2025:1351	Node.js is a software development platform for building fast and scalable network applications in the JavaScript programming language.
RHSA-2025:14135	The libarchive programming library can create and read several different streaming archive formats, including GNU tar, cpio, and ISO 9660 CD-ROM images. Libarchive is used notably in the bsdtar utility, scripting language bindings such as python-libarchive, and several popular desktop file managers.
RHSA-2025:14438	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:14557	Pluggable Authentication Modules (PAM) provide a system to set up authentication policies without the need to recompile programs to handle authentication.
RHSA-2025:14560	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2025:14900	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.

CVE ID	Description
RHSA-2025:15115	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2025:15123	The httpd packages provide the Apache HTTP Server, a powerful, efficient, and extensible web server.
RHSA-2025:1517	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2025:15702	The Common UNIX Printing System (CUPS) provides a portable printing layer for
RHSA-2025:1675	The Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols. BIND includes a DNS server (named); a resolver library (routines for applications to use when interfacing with DNS); and tools for verifying that the DNS server is operating correctly.
RHSA-2025:16823	OpenSSH is an SSH protocol implementation supported by a number of Linux, UNIX, and similar operating systems. It includes the core files necessary for both the OpenSSH client and server.
RHSA-2025:1737	The libpq package provides the PostgreSQL client library, which allows client programs to connect to PostgreSQL servers.
RHSA-2025:17415	The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic algorithms and protocols such as SSL, TLS, and DTLS.
RHSA-2025:17715	Vim (Vi IMproved) is an updated and improved version of the vi editor.
RHSA-2025:18286	libssh is a library which implements the SSH protocol. It can be used to implement client and server applications.
RHSA-2025:18821	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2025:1917	GNU Emacs is a powerful, customizable, self-documenting text editor. It provides special code editing features, a scripting language (elisp), and the capability to read e-mail and news.
RHSA-2025:2473	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:2600	The rsync utility enables the users to copy and synchronize files locally or across a network. Synchronization with rsync is fast because rsync only sends the differences in files over the network instead of sending whole files. The rsync utility is also used as a mirroring tool.
RHSA-2025:2686	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2025:2722	Kerberos is a network authentication system, which can improve the security of your network by eliminating the insecure practice of sending passwords over the network in unencrypted form. It allows clients and servers to authenticate to each other with the help of a trusted third party, the Kerberos key distribution center (KDC).
RHSA-2025:3026	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:3082	PostgreSQL is an advanced object-relational database management system (DBMS).
RHSA-2025:3260	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:3421	FreeType is a free, high-quality, portable font engine that can open and manage font files. FreeType loads, hints, and renders individual glyphs efficiently.
RHSA-2025:3615	libxslt is a library for transforming XML files into other textual formats (including HTML, plain text, and other XML representations of the underlying data) using the standard XSLT stylesheet transformation mechanism.
RHSA-2025:3828	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2025:3845	The java-1.8.0-openjdk packages provide the OpenJDK 8 Java Runtime Environment and the OpenJDK 8 Java Software Development Kit.
RHSA-2025:3852	The java-17-openjdk packages provide the OpenJDK 17 Java Runtime Environment and the OpenJDK 17 Java Software Development Kit.
RHSA-2025:3893	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:3913	Expat is a C library for parsing XML documents.
RHSA-2025:4049	A library that provides Abstract Syntax Notation One (ASN.1, as specified by the X.680 ITU-T recommendation) parsing and structures management, and Distinguished Encoding Rules (DER, as per X.690) encoding and decoding functions.

CVE ID	Description
RHSA-2025:4051	The gnutls packages provide the GNU Transport Layer Security (GnuTLS) library, which implements cryptographic algorithms and protocols such as SSL, TLS, and DTLS.
RHSA-2025:4244	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2025:4461	Node.js is a software development platform for building fast and scalable network applications in the JavaScript programming language.
RHSA-2025:4658	The libtiff packages contain a library of functions for manipulating Tagged Image File Format (TIFF) files.
RHSA-2025:4791	Python is an interpreted, interactive, object-oriented programming language, which includes modules, classes, exceptions, very high level dynamic data types and dynamic typing. Python supports interfaces to many system calls and libraries, as well as to various windowing systems.
RHSA-2025:7531	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:7540	The libjpeg-turbo packages contain a library of functions for manipulating JPEG images. They also contain simple client programs for accessing the libjpeg functions. These packages provide the same functionality and API as libjpeg but with better performance.
RHSA-2025:8056	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:8246	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:8395	The rsync utility enables the users to copy and synchronize files locally or across a network. Synchronization with rsync is fast because rsync only sends the differences in files over the network instead of sending whole files. The rsync utility is also used as a mirroring tool.
RHSA-2025:8411	Kerberos is a network authentication system, which can improve the security of your network by eliminating the insecure practice of sending passwords over the network in unencrypted form. It allows clients and servers to authenticate to each other with the help of a trusted third party, the Kerberos key distribution center (KDC).
RHSA-2025:8414	Git is a distributed revision control system with a decentralized architecture. As opposed to centralized version control systems with a client-server model, Git ensures that each working copy of a Git repository is an exact copy with complete revision history. This not only allows the user to work on and contribute to projects without the need to have permission to push the changes to their official repositories, but also makes it possible for the user to work with no network connection.
RHSA-2025:8514	Node.js is a software development platform for building fast and scalable network applications in the JavaScript programming language.
RHSA-2025:8655	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2025:8676	libxslt is a library for transforming XML files into other textual formats (including HTML, plain text, and other XML representations of the underlying data) using the standard XSLT stylesheet transformation mechanism.
RHSA-2025:8686	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
RHSA-2025:8743	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:8958	The libxml2 library is a development toolbox providing the implementation of various XML standards.
RHSA-2025:9580	The kernel packages contain the Linux kernel, the core of any Linux operating system.
RHSA-2025:9877	The glibc packages provide the standard C libraries (libc), POSIX thread libraries (libpthread), standard math libraries (libm), and the name service cache daemon (nscd) used by multiple programs on the system. Without these libraries, the Linux system cannot function correctly.
SUSE-SU-2025:0283-1	Security update for nginx
SUSE-SU-2025:0291-1	Security update for iperf
SUSE-SU-2025:0348-1	Security update for libxml2
SUSE-SU-2025:0355-1	Security update for bind

CVE ID	Description
SUSE-SU-2025:0369-1	Security update for curl
SUSE-SU-2025:0401-1	Security update for crypto-policies, krb5
SUSE-SU-2025:0430-1	Security update for openssl-3
SUSE-SU-2025:0548-1	Security update for libtasn1
SUSE-SU-2025:0554-1	Security update for python3
TEMP-0290435-0B57B5	tar's rmt command may have undesired side effects
TEMP-0517018-A83CE6	sysvinit: no-root option in expert installer exposes locally exploitable security flaw
TEMP-0628843-DBAD28	more related to CVE-2005-4890