Cloudera Flow Management 4.12.0

Cloudera Flow Management Release Notes

Date published: 2019-06-26
Date modified: 2026-03-31

CLOUDERA

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. All rights reserved.

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

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

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

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

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

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

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

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

Contents

What's new in Cloudera Flow Management 4.12.0

Explore the latest features and improvements in Cloudera Flow Management and how they support modern data pipeline development and operations.

Cloudera Flow Management 4.12.0 running on Apache NiFi 2.6.0 with Cloudera Manager on Cloudera on premises delivers capabilities not available in earlier NiFi versions, combined with Cloudera-specific enhancements. This release also includes fixes for multiple Common Vulnerabilities and Exposures (CVEs), improving stability, reliability, and compliance for enterprise deployments.

In addition, Cloudera Flow Management 4.12.0.1 release includes the following important fixes identified in Cloudera Flow Management 4.12.0.0:

Angular fix

In Cloudera Flow Management 4.12.0.0, the bundled Angular version could, in certain cases, fail to send the authentication tokens with UI requests, causing them to be rejected. This issue is resolved with an updated Angular version.

Sparkplug component fix

In Cloudera Flow Management 4.12.0.0, the ConsumeMQTTIIoT processor could halt processing and continuously yield after MQTT client disconnects. This issue is now fixed.

Snowflake component fix

In Cloudera Flow Management 4.12.0.0, the SnowflakeComputingConnectionPool controller service could return an error. This issue is now fixed.

The sections below highlight the most important updates in the Cloudera Flow Management 4.12.0 release.

New NiFi components

- ConsumeKinesis - Enables users to receive messages from an AWS Kinesis source
- PutIcebergRecord - Enables users to write record-based FlowFiles into Iceberg tables
- AzureDevOpsFlowRegistryClient - Enables users to leverage Azure DevOps as a Flow Registry

For a full list of supported NiFi components, see the Support Matrix.

Improvements

Industrial Internet of Things (IIoT) support

ConsumeMQTTIIoT and MQTTIIoTReader allows flows to receive and parse Sparkplug messages, enabling IIoT/edge use cases as an industry standard format.

Kafka OAuth2 authentication support

Support for OAuth2 (OAUTHBEARER) authentication has been added for Kafka components. This enhancement enables secure access to Kafka brokers using an OAuth2 Token Provider controller service to obtain and manage access tokens.

The feature is available for the following Kafka component groups:

- Kafka_2_6 / Kafka2CDP
- Kafka3ConnectionService

New flow analysis rule

The RequireServerSSLContext flow analysis rule has been added to ensure that servers (using ListenHttp or ListenTCP) cannot be created without a secure communications layer.

Flow Migration Tool

Cloudera Flow Migration Tool 7.0.0 adds support for migrations to Cloudera Flow Management 4.12.0 and includes several bug fixes.

For more information, see the Migration Tool documentation.

Upgrade and migration options

Cloudera Flow Management 4.12.0 is supported on Cloudera platform 7.3.2.

You can upgrade to Cloudera Flow Management 4.12.0 from 4.10.0 and 4.11.0 using Cloudera Manager. However, because versions 4.10.0 and 4.11.0 are supported on Cloudera Base on premises 7.3.1 and its service packs, you must first upgrade the runtime to Cloudera platform 7.3.2, and then upgrade Cloudera Flow Management to 4.12.0. For instructions, see Upgrading from Cloudera Flow Management 4.10.0 or 4.11.0 to 4.12.0.

In-place upgrades from Cloudera Flow Management 2.1.7 and lower versions to Cloudera Flow Management 4.12.0 are not supported.

Flows created in NiFi 1 must be migrated to NiFi 2 before they can be used in Cloudera Flow Management 4.12.0 (which runs NiFi 2). After the NiFi 1 - NiFi 2 migration, you can import the flows into a 4.12.0 environment. For more information about the migration process and the Cloudera Flow Migration Tool, see the Migration documentation.

Important: To transition from Cloudera Flow Management versions powered by NiFi 1 (2.1.7 and lower) to versions powered by NiFi 2 (4.10.0 and higher) with the help of the Flow Migration Tool, you must first upgrade to Cloudera Flow Management 2.1.7 SP3. This version is required to use the Cloudera Flow Migration Tool. For detailed compatibility information between Apache NiFi and Cloudera Flow Management versions, see the Component versions page.

Installation

To perform a fresh installation of Cloudera Flow Management 4.12.0, follow the Cloudera Flow Management installation workflow.

Support matrix

Review the support matrix before installing or upgrading Cloudera Flow Management.

Component versions

Review the Cloudera Flow Management component versions for compatibility with other applications.

Note: NiFi is compatible with the version of NiFi Registry bundled with your Cloudera Flow Management release as well as any later version.

Cloudera Flow Management 4.12.0

- Apache NiFi 2.6.0.4.12.0.0
- Apache NiFi Registry 2.6.0.4.12.0.0

Cloudera Flow Management 4.11.0

- Apache NiFi 2.4.0.4.11.0.0
- Apache NiFi Registry 2.4.0.4.11.0.0

Cloudera Flow Management 4.10.0

- Apache NiFi 2.3.0.4.10.0.0
- Apache NiFi Registry 2.3.0.4.10.0.0

Cloudera Flow Management 4.0.0 [Technical Preview]

- Apache NiFi 2.0.0.4.0.0.0
- Apache NiFi Registry 2.0.0.4.0.0.0

Cloudera Flow Management 2.1.7.2000 (SP2)

- Apache NiFi 1.28.1.2.1.7.2000
- Apache NiFi Registry 1.28.1.2.1.7.2000

Cloudera Flow Management 2.1.7.1000 (SP1)

- Apache NiFi 1.26.0.2.1.7.1000
- Apache NiFi Registry 1.26.0.2.1.7.1000

Cloudera Flow Management 2.1.7

- Apache NiFi 1.26.0.2.1.7.0
- Apache NiFi Registry 1.26.0.2.1.7.0

Cloudera Flow Management 2.1.6.1000 (SP1)

- Apache NiFi 1.23.1.2.1.6.1000
- Apache NiFi Registry 1.23.1.2.1.6.1000

Cloudera Flow Management 2.1.6

- Apache NiFi 1.23.1.2.1.6.0
- Apache NiFi Registry 1.23.1.2.1.6.0

Cloudera Flow Management 2.1.5.1000 (SP1)

- Apache NiFi 1.18.0.2.1.5.1000
- Apache NiFi Registry 1.18.0.2.1.5.1000

Cloudera Flow Management 2.1.5

- Apache NiFi 1.18.0.2.1.5.0
- Apache NiFi Registry 1.18.0.2.1.5.0

Cloudera Flow Management 2.1.4.1000 (SP1)

- Apache NiFi 1.16.0.2.1.4.1000
- Apache NiFi Registry 1.16.0.2.1.4.1000

Cloudera Flow Management 2.1.4

- Apache NiFi 1.16.0.2.1.4.0
- Apache NiFi Registry 1.16.0.2.1.4.0

Cloudera Flow Management 2.1.3

- Apache NiFi 1.15.2.2.1.3.0

- Apache NiFi Registry 1.15.2.2.1.3.0

Note: Apache NiFi and Apache NiFi Registry versions are unified in the 1.15.x release.

Cloudera Flow Management 2.1.2

- Apache NiFi 1.13.2.2.1.2.0
- Apache NiFi Registry 0.8.0.2.1.2.0

Cloudera Flow Management 2.1.1

- Apache NiFi 1.13.2.2.1.1.0
- Apache NiFi Registry 0.8.0.2.1.1.0

Cloudera Flow Management 2.0.4

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

Cloudera Flow Management 2.0.1

- Apache NiFi 1.11.4
- Apache NiFi Registry 0.6.0

System requirements

Review the system requirements before installing or upgrading Cloudera Flow Management.

Supported Cloudera platform versions

Cloudera Flow Management 4.12.0 supports the following versions of Cloudera Base on premises:

- 7.3.2

Supported JAVA Development Kits (JDK)

Cloudera Flow Management requires a minimum of JDK 21 for proper functionality. Ensure your environment meets this requirement before installation.

Other system requirements

ZooKeeper

You need to install the ZooKeeper service included with your Cloudera Base on premises cluster.

Python

- Minimum requirement: Python 3.11
- Recommended version: Python 3.12

Number of cores

- Minimum: Four cores per NiFi node are required for Cloudera support.
- Recommended: Eight cores per NiFi node, which typically provides an optimal starting point for most common use cases.

Supported operating systems

Review the list of operating systems supported in Cloudera Flow Management 4.12.0.

Operating system	Versions
RHEL	• 8.8 • 8.10 • 9.1 • 9.2 • 9.4 • 9.6
SLES	• 15 SP5
Oracle	• 8.8
Ubuntu	• 22.04 • 24.04
Windows	• 10 • Server 2016 • Server 2019 • Server 2022 • Server 2025

Note:

NiFi on Windows is only supported in standalone mode, not managed by Cloudera Manager or as part of a Cloudera cluster, and as a single instance installation. Clustering NiFi on Windows is not supported.

NiFi Registry is not supported on Windows.

Supported NiFi Registry databases

Review the list of databases supported by NiFi Registry in Cloudera Flow Management 4.12.0.

- H2
- PostgreSQL 10.x
- PostgreSQL 11.x
- PostgreSQL 12.x
- PostgreSQL 13.x
- PostgreSQL 14.x
- PostgreSQL 17.x
- MySQL 8.x

Supported NiFi processors

Review the list of Apache NiFi processors supported in Cloudera Flow Management 4.12.0.

Apache NiFi provides a wide range of processors for building dataflows. Cloudera Flow Management 4.12.0 is based on Apache NiFi 2.6.0 and includes most of these processors, but not all are supported by Cloudera.

Some processors available in Apache NiFi are developed and maintained by the community and are not supported in Cloudera. These processors may be excluded due to factors such as limited testing, insufficient reliability, lack of production readiness, or misalignment with Cloudera best practices.

To ensure stability, performance, and full support coverage, use only processors that are officially supported by Cloudera in production environments.

Table 1: List of NiFi processors supported in Cloudera Flow Management 4.12.0

AttributesToCSV	GetAwsTranscribeJobStatus
AttributesToJSON	GetAwsTranslateJobStatus
CalculateParquetOffsets	GetAzureEventHub
CalculateParquetRowGroupOffsets	GetAzureQueueStorage_v12
CalculateRecordStats	GetBoxFileCollaborators
CaptureChangeDebeziumDB2	GetBoxGroupMembers
CaptureChangeDebeziumMongoDB	GetCouchbaseKey
CaptureChangeDebeziumMySQL	GetElasticsearch
CaptureChangeDebeziumOracle	GetFile
CaptureChangeDebeziumPostgreSQL	GetFTP
CaptureChangeDebeziumSQLServer	GetGcpVisionAnnotateFilesOperationStatus
CaptureChangeMySQL	GetGcpVisionAnnotateImagesOperationStatus
ChunkDocument	GetHBase
CompressContent	GetHDFS
ConnectWebSocket	GetHDFSFileInfo
ConsumeAMQP	GetHDFSSequenceFile
ConsumeAzureEventHub	GetHubSpot
ConsumeBoxEnterpriseEvents	GetJiraIssue
ConsumeBoxEvents	GetMongoRecord
ConsumeElasticsearch	GetS3ObjectTags
ConsumeGCPubSub	GetSFTP
ConsumeGCPubSubLite	GetShopify
ConsumeJMS	GetSNMP
ConsumeKafka_2_6	GetSnowflakeIngestStatus
ConsumeKafka2CDP	GetSolr
ConsumeKafka2RecordCDP	GetSplunk
ConsumeKafkaRecord_2_6	GetSQS
ConsumeKinesis	GetWorkdayReport
ConsumeKinesisStream	GetZendesk
ConsumeMQTT	HandleHttpRequest
ConsumeMQTTIIoT	HandleHttpResponse
ConsumePLC	IdentifyMimeType
ConsumeSlack	InvokeAWSGatewayApi
ConsumeTwitter	InvokeGRPC
ConsumeWindowsEventLog	InvokeHTTP
ControlRate	InvokeScriptedProcessor
ConvertAvroToJSON	JoinEnrichment

ConvertAvroToParquet	JoltTransformJSON
ConvertCharacterSet	JoltTransformRecord
ConvertJSONToSQL	JSLTTransformJSON
ConvertProtobuf	JsonQueryElasticsearch
ConvertRecord	ListAzureBlobStorage_v12
CopyAzureBlobStorage_v12	ListAzureDataLakeStorage
CountText	ListBoxFile
CreateBoxFileMetadataInstance	ListBoxFileInfo
CreateBoxMetadataTemplate	ListBoxFileMetadataInstances
CreateHadoopSequenceFile	ListBoxFileMetadataTemplates
CryptographicHashContent	ListCDPObjectStore
DecryptContentAge	ListDatabaseTables
DecryptContentPGP	ListDropbox
DeduplicateRecord	ListenBeats
DeleteAzureBlobStorage_v12	ListenFTP
DeleteAzureDataLakeStorage	ListenGRPC
DeleteBoxFileMetadataInstance	ListenHTTP
DeleteByQueryElasticsearch	ListenNetFlow
DeleteCDPObjectStore	ListenOTLP
DeleteDynamoDB	ListenRELP
DeleteGCSObject	ListenSlack
DeleteGridFS	ListenSyslog
DeleteHBaseCells	ListenTCP
DeleteHBaseRow	ListenTCPRecord
DeleteHDFS	ListenTrapSNMP
DeleteS3Object	ListenUDP
DeleteSQS	ListenUDPRecord
DetectDuplicate	ListenWebSocket
DistributeLoad	ListFile
DuplicateFlowFile	ListFTP
EncodeContent	ListGCSBucket
EncryptContentAge	ListGoogleDrive
EncryptContentPGP	ListHDFS
EnforceOrder	ListS3
EvaluateJsonPath	ListSFTP
EvaluateXPath	ListSmb
EvaluateXQuery	LogAttribute
ExecuteGraphQuery	LogMessage
ExecuteGraphQueryRecord	LookupAttribute
ExecuteGroovyScript	LookupRecord

ExecuteProcess	MergeContent
ExecuteScript	MergeRecord
ExecuteSQL	ModifyCompression
ExecuteSQLRecord	MonitorActivity
ExecuteStateless	MoveAzureDataLakeStorage
ExecuteStreamCommand	MoveHDFS
ExtractAvroMetadata	Notify
ExtractGrok	PackageFlowFile
ExtractHL7Attributes	PaginatedJsonQueryElasticsearch
ExtractImageMetadata	ParseCEF
ExtractRecordSchema	ParseDocument
ExtractStructuredBoxFileMetadata	ParseEvtx
ExtractText	ParseSyslog
FetchAzureBlobStorage_v12	PartitionRecord
FetchAzureDataLakeStorage	PromptAzureOpenAI
FetchBoxFile	PromptChatGPT
FetchBoxFileInfo	PromptClaude
FetchBoxFileMetadataInstance	PromptOpenAI
FetchBoxFileRepresentation	PublishAMQP
FetchCDPObjectStore	PublishGCPubSub
FetchDistributedMapCache	PublishGCPubSubLite
FetchDropbox	PublishJMS
FetchFile	PublishKafka_2_6
FetchFTP	PublishKafka2CDP
FetchGCSObject	PublishKafka2RecordCDP
FetchGoogleDrive	PublishKafkaRecord_2_6
FetchGridFS	PublishMQTT
FetchHBaseRow	PublishSlack
FetchHDFS	PutAccumuloRecord
FetchParquet	PutAzureBlobStorage_v12
FetchPLC	PutAzureCosmosDBRecord
FetchS3Object	PutAzureDataLakeStorage
FetchSFTP	PutAzureEventHub
FetchSmb	PutAzureQueueStorage_v12
FilterAttribute	PutBigQuery
FlattenJson	PutBoxFile
ForkEnrichment	PutCassandraQL
ForkRecord	PutCassandraRecord
GenerateFlowFile	PutCDPObjectStore
GenerateRecord	PutChroma

GenerateTableFetch	PutClouderaHiveQL
GeoEnrichIP	PutClouderaHiveStreaming
GeoEnrichIPRecord	PutClouderaORC
GeohashRecord	PutCloudWatchMetric
GetAsanaObject	PutCouchbaseKey
GetAwsPollyJobStatus	PutDatabaseRecord
GetAwsTextractJobStatus	PutDistributedMapCache

Related Information

Supported NiFi controller services
Supported NiFi reporting tasks
Supported NiFi parameter providers
Supported NiFi flow analysis rules
Supported NiFi flow registry clients

Supported NiFi controller services

Review the list of Apache NiFi controller services supported in Cloudera Flow Management 4.12.0.

Apache NiFi provides a wide range of controller services for building dataflows. Cloudera Flow Management 4.12.0 is based on Apache NiFi 2.6.0 and includes most of these controller services, but not all are supported by Cloudera.

Some controller services available in Apache NiFi are developed and maintained by the community and are not supported in Cloudera. These controller services may be excluded due to factors such as limited testing, insufficient reliability, lack of production readiness, or misalignment with Cloudera best practices.

To ensure stability, performance, and full support coverage, use only controller services that are officially supported by Cloudera in production environments.

Table 2: List of NiFi controller services supported in Cloudera Flow Management 4.12.0

AccumuloService	JsonRecordSetWriter
ActiveMQJMSConnectionFactoryProvider	JsonTreeReader
ADLSCredentialsControllerService	JWTBearerOAuth2AccessTokenProvider
ADLSCredentialsControllerServiceLookup	Kafka3ConnectionService
ADLSIDBrokerCloudCredentialsProviderControllerService	KafkaRecordSink_2_6
AmazonGlueEncodedSchemaReferenceReader	KafkaRecordSinkCDP
AmazonGlueSchemaRegistry	KerberosKeytabUserService
AmazonMSKConnectionService	KerberosPasswordUserService
ApicurioSchemaRegistry	KerberosTicketCacheUserService
AvroReader	KuduLookupService
AvroRecordSetWriter	LivySessionController
AvroSchemaRegistry	LoggingRecordSink
AWSCredentialsProviderControllerService	MapCacheClientService
AWSIDBrokerCloudCredentialsProviderControllerService	MapCacheServer
AzureBlobIDBrokerCloudCredentialsProviderControllerService	MongoDBControllerService

AzureBlobStorageFileResourceService	MongoDBLookupService
AzureCosmosDBClientService	MQTTIIoTReader
AzureDataLakeStorageFileResourceService	Neo4JCypherClientService
AzureEventHubRecordSink	ParquetReader
AzureServiceBusJMSConnectionFactoryProvider	ParquetRecordSetWriter
AzureStorageCredentialsControllerService_v12	PEMEncodedSSLContextProvider
AzureStorageCredentialsControllerServiceLookup_v12	PhoenixThickConnectionPool
CassandraDistributedMapCache	PhoenixThinConnectionPool
CassandraSessionProvider	PostgreSQLConnectionPool
CdpCredentialsProviderControllerService	PropertiesFileLookupService
CdpOauth2AccessTokenProviderControllerService	ProtobufReader
CEFReader	ProxyPLC4XConnectionPool
CiscoEmblemSyslogMessageReader	RabbitMQJMSConnectionFactoryProvider
ClouderaAttributeSchemaReferenceReader	ReaderLookup
ClouderaAttributeSchemaReferenceWriter	RecordSetWriterLookup
ClouderaEncodedSchemaReferenceReader	RecordSinkServiceLookup
ClouderaEncodedSchemaReferenceWriter	RedisConnectionPoolService
ClouderaHiveConnectionPool	RedisDistributedMapCacheClientService
ClouderaHiveConnectionPoolLookup	RedshiftConnectionPool
ClouderaSchemaRegistry	RESTCatalogService
CMLLookupService	RestLookupService
ConfluentEncodedSchemaReferenceReader	S3FileResourceService
ConfluentEncodedSchemaReferenceWriter	ScriptedLookupService
ConfluentProtobufMessageNameResolver	ScriptedReader
ConfluentSchemaRegistry	ScriptedRecordSetWriter
CouchbaseKeyValueLookupService	ScriptedRecordSink
CouchbaseMapCacheClient	SetCacheClientService
CouchbaseRecordLookupService	SetCacheServer
CSVReader	SimpleCsvFileLookupService
CSVRecordLookupService	SimpleDatabaseLookupService
CSVRecordSetWriter	SimpleKeyValueLookupService
DatabaseRecordLookupService	SimpleRedisDistributedMapCacheClientServic
DatabaseRecordSink	SimpleScriptedLookupService
DatabaseTableSchemaRegistry	SiteToSiteReportingRecordSink
DBCPConnectionPool	SlackRecordSink
DBCPConnectionPoolLookup	SmbjClientProviderService
DeveloperBoxClientService	SnowflakeComputingConnectionPool
DistributedMapCacheLookupService	StandardAsanaClientProviderService
EBCDICRecordReader	StandardAzureCredentialsControllerService
ElasticSearchClientServiceImpl	StandardCouchbaseConnectionService

ElasticSearchLookupService	StandardDatabaseDialectService
ElasticSearchStringLookupService	StandardDropboxCredentialService
EmailRecordSink	StandardFileResourceService
EmbeddedHazelcastCacheManager	StandardHashiCorpVaultClientService
ExcelReader	StandardHttpContextMap
ExternalHazelcastCacheManager	StandardJiraCredentialService
FreeFormTextRecordSetWriter	StandardJsonSchemaRegistry
GCPCredentialsControllerService	StandardKustoIngestService
GCSFileResourceService	StandardKustoQueryService
GenericPLC4XConnectionPool	StandardOauth2AccessTokenProvider
GrokReader	StandardPGPPrivateKeyService
HadoopCatalogService	StandardPGPPublicKeyService
HadoopDBCPConnectionPool	StandardPLC4XConnectionPool
HazelcastMapCacheClient	StandardPrivateKeyService
HBase_2_ClientMapCacheService	StandardProtobufReader
HBase_2_ClientService	StandardProxyConfigurationService
HBase_2_RecordLookupService	StandardRestrictedSSLContextService
HikariCPConnectionPool	StandardS3EncryptionService
HiveCatalogService	StandardSnowflakeIngestManagerProviderServ
HttpRecordSink	StandardSSLContextService
ImpalaConnectionPool	StandardWebClientServiceProvider
IPFIXReader	Syslog5424Reader
IPLookupService	SyslogReader
JASN1Reader	TinkerpopClientService
JdbcCatalogService	UDPEventRecordSink
JettyWebSocketClient	VolatileSchemaCache
JettyWebSocketServer	WindowsEventLogReader
JiraRecordSink	XMLFileLookupService
JMSConnectionFactoryProvider	XMLReader
JndiJmsConnectionFactoryProvider	XMLRecordSetWriter
JsonConfigBasedBoxClientService	YamlTreeReader
JsonPathReader	ZendeskRecordSink

Related Information

Supported NiFi processors
Supported NiFi reporting tasks
Supported NiFi parameter providers
Supported NiFi flow analysis rules
Supported NiFi flow registry clients

Supported NiFi reporting tasks

Review the list of Apache NiFi reporting tasks supported in Cloudera Flow Management 4.12.0.

Cloudera Flow Management is based on Apache NiFi and includes a set of reporting tasks, most of which are supported by Cloudera. To ensure optimal performance and reliable support, it is crucial to use only supported reporting tasks and avoid deploying unsupported ones in production environments.

- AzureLogAnalyticsProvenanceReportingTask
- AzureLogAnalyticsReportingTask
- ControllerStatusReportingTask
- MonitorDiskUsage
- MonitorMemory
- QueryNiFiReportingTask
- ReportLineageToAtlas
- ScriptedReportingTask
- SiteToSiteBulletinReportingTask
- SiteToSiteMetricsReportingTask
- SiteToSiteProvenanceReportingTask
- SiteToSiteStatusReportingTask

Additional reporting tasks are developed and tested by the community but are not officially supported by Cloudera. Reporting tasks may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

Related Information

Supported NiFi processors
Supported NiFi controller services
Supported NiFi parameter providers
Supported NiFi flow analysis rules
Supported NiFi flow registry clients

Supported NiFi parameter providers

Review the list of Apache NiFi parameter providers supported in Cloudera Flow Management 4.12.0.

Cloudera Flow Management is shipped with Apache NiFi and includes a set of parameter providers, most of which are supported by Cloudera. To ensure optimal performance and reliable support, it is crucial to use only supported parameter providers and avoid deploying unsupported ones in production environments.

- AwsSecretsManagerParameterProvider
- AzureKeyVaultSecretsParameterProvider
- CyberArkConjurParameterProvider
- DatabaseParameterProvider
- EnvironmentVariableParameterProvider
- GcpSecretManagerParameterProvider
- HashiCorpVaultParameterProvider
- KubernetesSecretParameterProvider
- OnePasswordParameterProvider
- PropertiesFileParameterProvider

Additional parameter providers are developed and tested by the community but are not officially supported by Cloudera. Parameter providers may be excluded for various reasons, including insufficient reliability, incomplete test coverage, community declaration of non-production readiness, or deviations from Cloudera best practices.

Related Information

Supported NiFi processors
Supported NiFi controller services
Supported NiFi reporting tasks
Supported NiFi flow analysis rules
Supported NiFi flow registry clients

Supported NiFi flow analysis rules

Review the list of Apache NiFi processors supported in Cloudera Flow Management 4.12.0.

Flow Analysis Rules allow analyzing components or parts of a flow to help maintain optimal design. Rules can be set as Recommendations (informational only) or Policies (enforceable). Recommendation violations are logged but do not affect functionality, while Policy violations invalidate components until resolved.

- DisallowComponentType
- DisallowConsecutiveConnectionsWithRoundRobinLB
- DisallowDeadEnd
- DisallowDeprecatedProcessor
- DisallowExtractTextForFullContent
- RecommendRecordProcessor
- RequireHandleHttpResponseAfterHandleHttpRequest
- RequireMergeBeforePutIceberg
- RequireServerSSLContextService
- RestrictBackpressureSettings
- RestrictComponentNaming
- RestrictConcurrentTasksVsThreadPoolSizeInProcessors
- RestrictFlowFileExpiration
- RestrictProcessorConcurrency
- RestrictSchedulingForListProcessors
- RestrictThreadPoolSize
- RestrictYieldDurationForConsumeKafkaProcessors

Related Information

Supported NiFi processors
Supported NiFi controller services
Supported NiFi reporting tasks
Supported NiFi parameter providers
Supported NiFi flow registry clients

Supported NiFi flow registry clients

Review the list of Apache NiFi flow registry clients supported in Cloudera Flow Management 4.12.0.

NiFi Flow Registry clients are components in Apache NiFi that allow it to connect to external Flow Registries, services used to store and manage versioned dataflows.

- AzureDevOpsFlowRegistryClient
- BitbucketFlowRegistryClient
- ClouderaDataFlowRegistryClient
- ClouderaFlowLibraryFlowRegistryClient
- GitHubFlowRegistryClient
- GitLabFlowRegistryClient

- NifiRegistryFlowRegistryClient

Related Information

Supported NiFi processors
Supported NiFi controller services
Supported NiFi reporting tasks
Supported NiFi parameter providers
Supported NiFi flow analysis rules

Supported Cloudera exclusive components

Review the list of Cloudera exclusive components supported in Cloudera Flow Management 4.12.0.

Cloudera Flow Management provides a set of NiFi components available only to Cloudera customers. These components provide additional functionality and are tailored to enhance the Cloudera NiFi experience. The list of these components is provided below.

Processors

- CaptureChangeDebeziumDB2
- CaptureChangeDebeziumMongoDB
- CaptureChangeDebeziumMySQL
- CaptureChangeDebeziumOracle
- CaptureChangeDebeziumPostgreSQL
- CaptureChangeDebeziumSQLServer
- ConsumeKafka2CDP
- ConsumeKafka2RecordCDP
- ConsumeMQTTIIoT
- ConsumePLC
- ConvertProtobuf
- DeleteCDPObjectStore
- FetchCDPObjectStore
- FetchPLC
- GetJiraIssue
- GetSlackReaction
- GetTCP
- InvokeGRPC
- ListCDPObjectStore
- ListenGRPC
- ListenNetFlow
- PublishKafka2CDP
- PublishKafka2RecordCDP
- PutCDPObjectStore
- PutClouderaHiveQL
- PutClouderaHiveStreaming
- PutClouderaORC
- PutIcebergCDC
- PutJiraIssue
- PutPLC
- SawmillTransformJSON
- SawmillTransformRecord
- SelectClouderaHiveQL

- TriggerClouderaHiveMetaStoreEvent
- UpdateClouderaHiveTable
- UpdateDeltaLakeTable
- UpdateJiraIssue

Controller services

- ActiveMQJMSConnectionFactoryProvider
- ADLSIDBrokerCloudCredentialsProviderControllerService
- AWSIDBrokerCloudCredentialsProviderControllerService
- AzureBlobIDBrokerCloudCredentialsProviderControllerService
- AzureServiceBusJMSConnectionFactoryProvider
- CdpCredentialsProviderControllerService
- CdpOauth2AccessTokenProviderControllerService
- CiscoEmblemSyslogMessageReader
- ClouderaAttributeSchemaReferenceReader
- ClouderaAttributeSchemaReferenceWriter
- ClouderaEncodedSchemaReferenceReader
- ClouderaEncodedSchemaReferenceWriter
- ClouderaHiveConnectionPool
- ClouderaHiveConnectionPoolLookup
- ClouderaSchemaRegistry
- CMLLookupService
- EBCDICRecordReader
- GenericPLC4XConnectionPool
- ImpalaConnectionPool
- IPFIXReader
- JiraRecordSink
- KafkaRecordSinkCDP
- MQTTIIoTReader
- PhoenixThickConnectionPool
- PhoenixThinConnectionPool
- PostgreSQLConnectionPool
- ProxyPLC4XConnectionPool
- RabbitMQJMSConnectionFactoryProvider
- RedshiftConnectionPool
- StandardJiraCredentialService
- StandardPLC4XConnectionPool

Parameter providers

- CyberArkConjurParameterProvider
- PropertiesFileParameterProvider

Flow analysis rules

- DisallowConsecutiveConnectionsWithRoundRobinLB
- DisallowDeadEnd
- DisallowDeprecatedProcessor
- DisallowExtractTextForFullContent
- RecommendRecordProcessor
- RequireHandleHttpResponseAfterHandleHttpRequest

- RequireMergeBeforePutIceberg
- RestrictComponentNaming
- RestrictConcurrentTasksVsThreadPoolSizeInProcessors
- RestrictProcessorConcurrency
- RestrictSchedulingForListProcessors
- RestrictThreadPoolSize
- RestrictYieldDurationForConsumeKafkaProcessors

Flow registry clients

- ClouderaDataFlowRegistryClient
- ClouderaFlowLibraryFlowRegistryClient

Download locations

You can download the Cloudera Flow Management software artifacts from the Cloudera Archive. There are different artifacts for different operating systems, standalone components, and Windows files.

Use the following tables to identify the latest available Cloudera Flow Management 4.12.0 release suitable for your operating system and deployment requirements.

Note:

You must have credentials to download Cloudera Flow Management files. Your download credential is not the same as the credential you use to access the Cloudera Support Portal.

You can get download credentials in the following ways:

- Contact your Cloudera sales representative.
- Check the Welcome email you have received for your Cloudera Flow Management account.
- File a non-technical case on the Cloudera Support Portal for the Cloudera Support team to assist you.

Cloudera Flow Management download links

Table 3: RHEL 8

File	Location
Manifest	https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat8/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat8/yum/tars/parcel/CFM-4.12.0.1-8-el8.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat8/yum/tars/parcel/CFM-4.12.0.1-8-el8.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat8/yum/tars/parcel/NIFI-2.6.0.4.12.0.1-8.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat8/yum/tars/parcel/NIFIREGISTRY-2.6.0.4.12.0.1-8.jar

Table 4: RHEL 9

File	Location
Manifest	https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat9/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat9/yum/tars/parcel/CFM-4.12.0.1-8-el9.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat9/yum/tars/parcel/CFM-4.12.0.1-8-el9.parcel.sha

File	Location
CSD	**NiFi:** https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat9/yum/tars/parcel/NIFI-2.6.0.4.12.0.1-8.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm4/4.12.0.1/redhat9/yum/tars/parcel/NIFIREGISTRY-2.6.0.4.12.0.1-8.jar

Table 5: SLES 15

File	Location
Manifest	https://archive.cloudera.com/p/cfm4/4.12.0.1/sles15/yum/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm4/4.12.0.1/sles15/yum/tars/parcel/CFM-4.12.0.1-8-sles15.parcel
Parcel sha file	https://archive.cloudera.com/p/cfm4/4.12.0.1/sles15/yum/tars/parcel/CFM-4.12.0.1-8-sles15.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm4/4.12.0.1/sles15/yum/tars/parcel/NIFI-2.6.0.4.12.0.1-8.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm4/4.12.0.1/sles15/yum/tars/parcel/NIFIREGISTRY-2.6.0.4.12.0.1-8.jar

Table 6: Ubuntu 22

File	Location
Manifest	https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu22/apt/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu22/apt/tars/parcel/CFM-4.12.0.1-8-jammy.parcel
Parcel SHA file	https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu22/apt/tars/parcel/CFM-4.12.0.1-8-jammy.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu22/apt/tars/parcel/NIFI-2.6.0.4.12.0.1-8.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu22/apt/tars/parcel/NIFIREGISTRY-2.6.0.4.12.0.1-8.jar

Table 7: Ubuntu 24

File	Location
Manifest	https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu24/apt/tars/parcel/manifest.json
Parcel	https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu24/apt/tars/parcel/CFM-4.12.0.1-8-noble.parcel
Parcel SHA file	https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu24/apt/tars/parcel/CFM-4.12.0.1-8-noble.parcel.sha
CSD	**NiFi:** https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu24/apt/tars/parcel/NIFI-2.6.0.4.12.0.1-8.jar **NiFi Registry:** https://archive.cloudera.com/p/cfm4/4.12.0.1/ubuntu24/apt/tars/parcel/NIFIREGISTRY-2.6.0.4.12.0.1-8.jar

Cloudera Flow Migration Tool download links

Cloudera Flow Migration Tool version	Source Cloudera Flow Management version	Target Cloudera Flow Management version	Cloudera Flow Migration Tool repository
7.0.1	2.1.7.300x	4.12.0.x	Cloudera Flow Migration Tool 7.0.1 repository
7.0.0			Cloudera Flow Migration Tool 7.0.0 repository

Unsupported features

Review the list of features that are not supported in Cloudera Flow Management 4.12.0.

Behavioral changes

Learn about behavioral changes in Cloudera Flow Management 4.12.0.

Known issues in Cloudera Flow Management

Review the list of known issues and limitations in Cloudera Flow Management 4.12.0.

Kafka OAuth authentication not supported across mixed Kafka component groups

Kafka OAuth2 (OAUTHBEARER) authentication cannot be used simultaneously by components from Kafka_2_6 / Kafka2CDP and Kafka3ConnectionService. If OAuth is enabled for one group, components in the other group will fail with authentication errors.

Use OAuth authentication with only one Kafka component group in a deployment, or restart before switching between them..

Fixed issues in Cloudera Flow Management

Review the list of issues resolved in Cloudera Flow Management 4.12.0.

CFM-6474: Improved schema conversion in EBCDICRecordReader for integral data types

Previously, all integral fields were converted to LONG, regardless of their size. The conversion now uses the PIC digit count to determine the appropriate type:

- PIC S9(1)–S9(9) # INTEGER
- PIC S9(10)–S9(18) # LONG

Fixed CVEs in Cloudera Flow Management

Review the list of Common Vulnerabilities and Exposures (CVE) resolved in Cloudera Flow Management 4.12.0.

CVE-2012-6708:

jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

CVE-2014-0114:

Apache Commons BeanUtils, as distributed in lib/commons-beanutils-1.8.0.jar in Apache Struts 1.x through 1.3.10 and in other products requiring commons-beanutils through 1.9.2, does not suppress the class property, which allows remote attackers to "manipulate" the ClassLoader and execute arbitrary code via the class parameter, as demonstrated by the passing of this parameter to the getClass method of the ActionForm object in Struts 1.

CVE-2015-9251:

jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

CVE-2016-3427:

Unspecified vulnerability in Oracle Java SE 6u113, 7u99, and 8u77; Java SE Embedded 8u77; and JRockit R28.3.9 allows remote attackers to affect confidentiality, integrity, and availability via vectors related to JMX.

CVE-2017-15288:

The compilation daemon in Scala before 2.10.7, 2.11.x before 2.11.12, and 2.12.x before 2.12.4 uses weak permissions for private files in /tmp/scala-devel/${USER:shared}/scalac-compile-server-port, which allows local users to write to arbitrary class files and consequently gain privileges.

CVE-2018-10237:

Unbounded memory allocation in Google Guava 11.0 through 24.x before 24.1.1 allows remote attackers to conduct denial of service attacks against servers that depend on this library and deserialize attacker-provided data, because the AtomicDoubleArray class (when serialized with Java serialization) and the CompoundOrdering class (when serialized with GWT serialization) perform eager allocation without appropriate checks on what a client has sent and whether the data size is reasonable.

CVE-2019-10086:

In Apache Commons Beanutils 1.9.2, a special BeanIntrospector class was added which allows suppressing the ability for an attacker to access the classloader via the class property available on all Java objects. We, however were not using this by default characteristic of the PropertyUtilsBean.

CVE-2019-11358:

jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

CVE-2019-2684:

Vulnerability in the Java SE, Java SE Embedded component of Oracle Java SE (subcomponent: RMI). Supported versions that are affected are Java SE: 7u211, 8u202, 11.0.2 and 12; Java SE Embedded: 8u201. Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Java SE, Java SE Embedded. Successful attacks of this vulnerability can result in unauthorized creation, deletion or modification access to critical data or all Java SE, Java SE Embedded accessible data. Note: This vulnerability applies to Java deployments, typically in clients running sandboxed Java Web Start applications or sandboxed Java applets (in Java SE 8), that load and run untrusted code (e.g., code that comes from the internet) and rely on the Java sandbox for security. This vulnerability can also be exploited by using APIs in the specified Component, e.g., through a web service which supplies data to the APIs. CVSS 3.0 Base Score 5.9 (Integrity impacts). CVSS Vector: (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N).

CVE-2020-11022: Potential XSS vulnerability in jQuery

In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

CVE-2020-11023: Potential XSS vulnerability in jQuery

In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

CVE-2020-13946:

In Apache Cassandra, all versions prior to 2.1.22, 2.2.18, 3.0.22, 3.11.8 and 4.0-beta2, it is possible for a local attacker without access to the Apache Cassandra process or configuration files to manipulate the RMI registry to perform a man-in-the-middle attack and capture user names and passwords used to access the JMX interface. The attacker can then use these credentials to access the JMX interface and perform unauthorised operations. Users should also be aware of CVE-2019-2684, a JRE vulnerability that enables this issue to be exploited remotely.

CVE-2020-17516:

Apache Cassandra versions 2.1.0 to 2.1.22, 2.2.0 to 2.2.19, 3.0.0 to 3.0.23, and 3.11.0 to 3.11.9, when using 'dc' or 'rack' internode_encryption setting, allows both encrypted and unencrypted internode connections. A misconfigured node or a malicious user can use the unencrypted connection despite not being in the same rack or dc, and bypass mutual TLS requirement.

CVE-2020-36518:

jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.

CVE-2020-7656:

jquery prior to 1.9.0 allows Cross-site Scripting attacks via the load method. The load method fails to recognize and remove "<script>" HTML tags that contain a whitespace character, i.e: "</script >", which results in the enclosed script logic to be executed.

CVE-2020-8908: Temp directory permission issue in Guava

A temp directory creation vulnerability exists in all versions of Guava, allowing an attacker with access to the machine to potentially access data in a temporary directory created by the Guava API com.google.common.io.Files.createTempDir(). By default, on unix-like systems, the created directory is world-readable (readable by an attacker with access to the system). The method in question has been marked @Deprecated in versions 30.0 and later and should not be used. For Android developers, we recommend choosing a temporary directory API provided by Android, such as context.getCacheDir(). For other Java developers, we recommend migrating to the Java 7 API java.nio.file.Files.createTempDirectory() which explicitly configures permissions of 700, or configuring the Java runtime's java.io.tmpdir system property to point to a location whose permissions are appropriately configured.

CVE-2020-9040:

Couchbase Server Java SDK before 2.7.1.1 allows a potential attacker to forge an SSL certificate and pose as the intended peer. An attacker can leverage this flaw by crafting a cryptographically valid certificate that will be accepted by Java SDK's Netty component due to missing hostname verification.

CVE-2021-44521: Remote code execution for scripted UDFs

When running Apache Cassandra with the following configuration: enable_user_defined_functions: true enable_scripted_user_defined_functions: true enable_user_defined_functions_threads: false it is possible for an attacker to execute arbitrary code on the host. The attacker would need to have enough permissions to create user defined functions in the cluster to be able to exploit this. Note that this configuration is documented as unsafe, and will continue to be considered unsafe after this CVE.

CVE-2021-46877:

jackson-databind 2.10.x through 2.12.x before 2.12.6 and 2.13.x before 2.13.1 allows attackers to cause a denial of service (2 GB transient heap usage per read) in uncommon situations involving JsonNode JDK serialization.

CVE-2022-25647: Deserialization of Untrusted Data

The package com.google.code.gson:gson before 2.8.9 are vulnerable to Deserialization of Untrusted Data via the writeReplace() method in internal classes, which may lead to DoS attacks.

CVE-2022-36944:

Scala 2.13.x before 2.13.9 has a Java deserialization chain in its JAR file. On its own, it cannot be exploited. There is only a risk in conjunction with Java object deserialization within an application. In such situations, it allows attackers to erase contents of arbitrary files, make network connections, or possibly run arbitrary code (specifically, Function0 functions) via a gadget chain.

CVE-2022-41881:

Netty project is an event-driven asynchronous network application framework. In versions prior to 4.1.86.Final, a StackOverflowError can be raised when parsing a malformed crafted message due to an infinite recursion. This issue is patched in version 4.1.86.Final. There is no workaround, except using a custom HaProxyMessageDecoder.

CVE-2022-41915:

Netty project is an event-driven asynchronous network application framework. Starting in version 4.1.83.Final and prior to 4.1.86.Final, when calling `DefaultHttpHeadesr.set` with an _iterator_ of values, header value validation was not performed, allowing malicious header values in the iterator to perform HTTP Response Splitting. This issue has been patched in version 4.1.86.Final. Integrators can work around the issue by changing the `DefaultHttpHeaders.set(CharSequence, Iterator<?>)` call, into a `remove()` call, and call `add()` in a loop over the iterator of values.

CVE-2022-42003:

In FasterXML jackson-databind before versions 2.13.4.1 and 2.12.17.1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled.

CVE-2022-42004:

In FasterXML jackson-databind before 2.13.4, resource exhaustion can occur because of a lack of a check in BeanDeserializer._deserializeFromArray to prevent use of deeply nested arrays. An application is vulnerable only with certain customized choices for deserialization.

CVE-2023-2976: Use of temporary directory for file creation in `FileBackedOutputStream` in Guava

Use of Java's default temporary directory for file creation in `FileBackedOutputStream` in Google Guava versions 1.0 to 31.1 on Unix systems and Android Ice Cream Sandwich allows other users and apps on the machine with access to the default Java temporary directory to be able to access the files created by the class.

Even though the security vulnerability is fixed in version 32.0.0, we recommend using version 32.0.1 as version 32.0.0 breaks some functionality under Windows.

CVE-2023-34462: netty-handler SniHandler 16MB allocation

Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. The `SniHandler` can allocate up to 16MB of heap for each channel during the TLS handshake. When the handler or the channel does not have an idle timeout, it can be used to make a TCP server using the `SniHandler` to allocate 16MB of heap. The `SniHandler` class is a handler that waits for the TLS handshake to configure a `SslHandler` according to the indicated server name by the `ClientHello` record. For this matter it allocates a `ByteBuf` using the value defined in the `ClientHello` record. Normally the value of the packet should be smaller than the handshake packet but there are not checks done here and the way the code is written, it is possible to craft a packet that makes the `SslClientHelloHandler`. This vulnerability has been fixed in version 4.1.94.Final.

CVE-2023-39410: Apache Avro Java SDK: Memory when deserializing untrusted data in Avro Java SDK

When deserializing untrusted or corrupted data, it is possible for a reader to consume memory beyond the allowed constraints and thus lead to out of memory on the system.

This issue affects Java applications using Apache Avro Java SDK up to and including 1.11.2. Users should update to apache-avro version 1.11.3 which addresses this issue.

CVE-2023-44487:

The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.

CVE-2024-1597: pgjdbc SQL Injection via line comment generation

pgjdbc, the PostgreSQL JDBC Driver, allows attacker to inject SQL if using PreferQueryMode=SIMPLE. Note this is not the default. In the default mode there is no vulnerability. A placeholder for a numeric value must be immediately preceded by a minus. There must be a second placeholder for a string value after the first placeholder; both must be on the same line. By constructing a matching string payload, the attacker can inject SQL to alter the query,bypassing the protections that parameterized queries bring against SQL Injection attacks. Versions before 42.7.2, 42.6.1, 42.5.5, 42.4.4, 42.3.9, and 42.2.28 are affected.

CVE-2024-29371:

In jose4j before 0.9.6, an attacker can cause a Denial-of-Service (DoS) condition by crafting a malicious JSON Web Encryption (JWE) token with an exceptionally high compression ratio. When this token is processed by the server, it results in significant memory allocation and processing time during decompression.

CVE-2024-36124: iq80 Snappy has an out-of-bounds read when uncompressing data, leading to JVM crash

iq80 Snappy is a compression/decompression library. When uncompressing certain data, Snappy tries to read outside the bounds of the given byte arrays. Because Snappy uses the JDK class `sun.misc.Unsafe` to speed up memory access, no additional bounds checks are performed and this has similar security consequences as out-of-bounds access in C or C++, namely it can lead to non-deterministic behavior or crash the JVM. iq80 Snappy is not actively maintained anymore. As quick fix users can upgrade to version 0.5.

CVE-2024-45216: Apache Solr: Authentication bypass possible using a fake URL Path ending

Improper Authentication vulnerability in Apache Solr.

Solr instances using the PKIAuthenticationPlugin, which is enabled by default when Solr Authentication is used, are vulnerable to Authentication bypass.

A fake ending at the end of any Solr API URL path, will allow requests to skip Authentication while maintaining the API contract with the original URL Path.

This fake ending looks like an unprotected API path, however it is stripped off internally after authentication but before API routing.

This issue affects Apache Solr: from 5.3.0 before 8.11.4, from 9.0.0 before 9.7.0.

Users are recommended to upgrade to version 9.7.0, or 8.11.4, which fix the issue.

CVE-2024-45217: Apache Solr: ConfigSets created during a backup restore command are trusted implicitly

Insecure Default Initialization of Resource vulnerability in Apache Solr.

New ConfigSets that are created via a Restore command, which copy a configSet from the backup and give it a new name, are created without setting the "trusted" metadata.

ConfigSets that do not contain the flag are trusted implicitly if the metadata is missing, therefore this leads to "trusted" ConfigSets that may not have been created with an Authenticated request.

"trusted" ConfigSets are able to load custom code into classloaders, therefore the flag is supposed to only be set when the request that uploads the ConfigSet is Authenticated & Authorized.

This issue affects Apache Solr: from 6.6.0 before 8.11.4, from 9.0.0 before 9.7.0. This issue does not affect Solr instances that are secured via Authentication/Authorization.

Users are primarily recommended to use Authentication and Authorization when running Solr. However, upgrading to version 9.7.0, or 8.11.4 will mitigate this issue otherwise.

CVE-2024-47561: Apache Avro Java SDK: Arbitrary Code Execution when reading Avro schema (Java SDK)

Schema parsing in the Java SDK of Apache Avro 1.11.3 and previous versions allows bad actors to execute arbitrary code.

Users are recommended to upgrade to version 1.11.4 or 1.12.0, which fix this issue.

CVE-2024-51504: Apache ZooKeeper: Authentication bypass with IP-based authentication in Admin Server

When using IPAuthenticationProvider in ZooKeeper Admin Server there is a possibility of Authentication Bypass by Spoofing -- this only impacts IP based authentication implemented in ZooKeeper Admin Server. Default configuration of client's IP address detection in IPAuthenticationProvider, which uses HTTP request headers, is weak and allows an attacker to bypass authentication via spoofing client's IP address in request headers. Default configuration honors X-Forwarded-For HTTP header to read client's IP address. X-Forwarded-For request header is mainly used by proxy servers to identify the client and can be easily spoofed by an attacker pretending that the request comes from a different IP address. Admin Server commands, such as snapshot and restore arbitrarily can be executed on successful exploitation which could potentially lead to information leakage or service availability issues. Users are recommended to upgrade to version 3.9.3, which fixes this issue.

CVE-2024-57699:

A security issue was found in Netplex Json-smart 2.5.0 through 2.5.1. When loading a specially crafted JSON input, containing a large number of '{', a stack exhaustion can be trigger, which could allow an attacker to cause a Denial of Service (DoS). This issue exists because of an incomplete fix for CVE-2023-1370.

CVE-2024-7246: HPACK table poisoning in gRPC C++, Python & Ruby

It's possible for a gRPC client communicating with a HTTP/2 proxy to poison the HPACK table between the proxy and the backend such that other clients see failed requests. It's also possible to use this vulnerability to leak other clients HTTP header keys, but not values.

This occurs because the error status for a misencoded header is not cleared between header reads, resulting in subsequent (incrementally indexed) added headers in the first request being poisoned until cleared from the HPACK table.

Please update to a fixed version of gRPC as soon as possible. This bug has been fixed in 1.58.3, 1.59.5, 1.60.2, 1.61.3, 1.62.3, 1.63.2, 1.64.3, 1.65.4.

CVE-2024-7254: Stack overflow in Protocol Buffers Java Lite

Any project that parses untrusted Protocol Buffers data containing an arbitrary number of nested groups / series of SGROUP tags can corrupted by exceeding the stack limit i.e. StackOverflow. Parsing nested groups as unknown fields with DiscardUnknownFieldsParser or Java Protobuf Lite parser, or against Protobuf map fields, creates unbounded recursions that can be abused by an attacker.

CVE-2025-11226: Conditional processing of logback.xml configuration file, in conjuction with Spring Framework and Janino

ACE vulnerability in conditional configuration file processing by QOS.CH logback-core up to and including version 1.5.18 in Java applications, allows an attacker to execute arbitrary code by compromising an existing logback configuration file or by injecting an environment variable before program execution.

A successful attack requires the presence of Janino library and Spring Framework to be present on the user's class path. In addition, the attacker must have write access to a

configuration file. Alternatively, the attacker could inject a malicious

environment variable pointing to a malicious configuration file. In both

cases, the attack requires existing privilege.

CVE-2025-22227: CVE-2025-22227: Authentication Leak On Redirect With Reactor Netty HTTP Client

In some specific scenarios with chained redirects, Reactor Netty HTTP client leaks credentials. In order for this to happen, the HTTP client must have been explicitly configured to follow redirects.

CVE-2025-25193: Denial of Service attack on windows app using Netty

Netty, an asynchronous, event-driven network application framework, has a vulnerability in versions up to and including 4.1.118.Final. An unsafe reading of environment file could potentially cause a denial of service in Netty. When loaded on an Windows application, Netty attempts to load a file that does not exist. If an attacker creates such a large file, the Netty application crash. A similar issue was previously reported as CVE-2024-47535. This

issue was fixed, but the fix was incomplete in that null-bytes were not counted against the input limit. Commit d1fbda62d3a47835d3fb35db8bd42ecc205a5386 contains an updated fix.

CVE-2025-27817: Apache Kafka Client: Arbitrary file read and SSRF vulnerability

A possible arbitrary file read and SSRF vulnerability has been identified in Apache Kafka Client. Apache Kafka Clients accept configuration data for setting the SASL/OAUTHBEARER connection with the brokers, including "sasl.oauthbearer.token.endpoint.url" and "sasl.oauthbearer.jwks.endpoint.url". Apache Kafka allows clients to read an arbitrary file and return the content in the error log, or sending requests to an unintended location. In applications where Apache Kafka Clients configurations can be specified by an untrusted party, attackers may use the "sasl.oauthbearer.token.endpoint.url" and "sasl.oauthbearer.jwks.endpoint.url" configuratin to read arbitrary contents of the disk and environment variables or make requests to an unintended location. In particular, this flaw may be used in Apache Kafka Connect to escalate from REST API access to filesystem/environment/URL access, which may be undesirable in certain environments, including SaaS products.

Since Apache Kafka 3.9.1/4.0.0, we have added a system property ("-Dorg.apache.kafka.sasl.oauthbearer.allowed.urls") to set the allowed urls in SASL JAAS configuration. In 3.9.1, it accepts all urls by default for backward compatibility. However in 4.0.0 and newer, the default value is empty list and users have to set the allowed urls explicitly.

CVE-2025-31672: Apache POI: parsing OOXML based files (xlsx, docx, etc.), poi-ooxml could read unexpected data if underlying zip has duplicate zip entry names

Improper Input Validation vulnerability in Apache POI. The issue affects the parsing of OOXML format files like xlsx, docx and pptx. These file formats are basically zip files and it is possible for malicious users to add zip entries with duplicate names (including the path) in the zip. In this case, products reading the affected file could read different data because 1 of the zip entries with the duplicate name is selected over another but different products may choose a different zip entry.

This issue affects Apache POI poi-ooxml before 5.4.0. poi-ooxml 5.4.0 has a check that throws an exception if zip entries with duplicate file names are found in the input file.

Users are recommended to upgrade to version poi-ooxml 5.4.0, which fixes the issue. Please read https://poi.apache.org/security.html for recommendations about how to use the POI libraries securely.

CVE-2025-33042: Apache Avro Java SDK: Code injection on Java generated code

Improper Control of Generation of Code ('Code Injection') vulnerability in Apache Avro Java SDK when generating specific records from untrusted Avro schemas.

This issue affects Apache Avro Java SDK: all versions through 1.11.4 and version 1.12.0.

Users are recommended to upgrade to version 1.12.1 or 1.11.5, which fix the issue.

CVE-2025-46762: Apache Parquet Java: Potential malicious code execution from trusted packages in the parquet-avro module when reading an Avro schema from a Parquet file metadata

Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code.

While 1.15.1 introduced a fix to restrict untrusted packages, the default setting of trusted packages still allows malicious classes from these packages to be executed.

The exploit is only applicable if the client code of parquet-avro uses the "specific" or the "reflect" models deliberately for reading Parquet files. ("generic" model is not impacted)

Users are recommended to upgrade to 1.15.2 or set the system property "org.apache.parquet.avro.SERIALIZABLE_PACKAGES" to an empty string on 1.15.1. Both are sufficient to fix the issue.

CVE-2025-48734: Apache Commons BeanUtils: PropertyUtilsBean does not suppresses an enum's declaredClass property by default

Improper Access Control vulnerability in Apache Commons.

A special BeanIntrospector class was added in version 1.9.2. This can be used to stop attackers from using the declared class property of Java enum objects to get access to the classloader. However this protection was not enabled by default. PropertyUtilsBean (and consequently BeanUtilsBean) now disallows declared class level property access by default.

Releases 1.11.0 and 2.0.0-M2 address a potential security issue when accessing enum properties in an uncontrolled way. If an application using Commons BeanUtils passes property paths from an external source directly to the getProperty() method of PropertyUtilsBean, an attacker can access the enum's class loader via the "declaredClass" property available on all Java "enum" objects. Accessing the enum's "declaredClass" allows remote attackers to access the ClassLoader and execute arbitrary code. The same issue exists with PropertyUtilsBean.getNestedProperty().

Starting in versions 1.11.0 and 2.0.0-M2 a special BeanIntrospector suppresses the "declaredClass" property. Note that this new BeanIntrospector is enabled by default, but you can disable it to regain the old behavior; see section 2.5 of the user's guide and the unit tests.

This issue affects Apache Commons BeanUtils 1.x before 1.11.0, and 2.x before 2.0.0-M2.Users of the artifact commons-beanutils:commons-beanutils

1.x are recommended to upgrade to version 1.11.0, which fixes the issue.

Users of the artifact org.apache.commons:commons-beanutils2

2.x are recommended to upgrade to version 2.0.0-M2, which fixes the issue.

CVE-2025-48924: Apache Commons Lang, Apache Commons Lang: ClassUtils.getClass(...) can throw a StackOverflowError on very long inputs

Uncontrolled Recursion vulnerability in Apache Commons Lang.

This issue affects Apache Commons Lang: Starting with commons-lang:commons-lang 2.0 to 2.6, and, from org.apache.commons:commons-lang3 3.0 before 3.18.0.

The methods ClassUtils.getClass(...) can throw StackOverflowError on very long inputs. Because an Error is usually not handled by applications and libraries, a

StackOverflowError could cause an application to stop.

Users are recommended to upgrade to version 3.18.0, which fixes the issue.

CVE-2025-49128: Jackson-core Vulnerable to Memory Disclosure via Source Snippet in JsonLocation

Jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. Starting in version 2.0.0 and prior to version 2.13.0, a flaw in jackson-core's `JsonLocation._appendSourceDesc` method allows up to 500 bytes of unintended memory content to be included in exception messages. When parsing JSON from a byte array with an offset and length, the exception message incorrectly reads from the beginning of the array instead of the logical payload start. This results in possible information disclosure in systems using pooled or reused buffers, like Netty or Vert.x. This issue was silently fixed in jackson-core version 2.13.0, released on September 30, 2021, via PR #652. All users should upgrade to version 2.13.0 or later. If upgrading is not immediately possible, applications can mitigate the issue by disabling exception message exposure to clients to avoid returning parsing exception messages in HTTP responses and/or disabling source inclusion in exceptions to prevent Jackson from embedding any source content in exception messages, avoiding leakage.

CVE-2025-5115: MadeYouReset HTTP/2 vulnerability

In Eclipse Jetty, versions <=9.4.57, <=10.0.25, <=11.0.25, <=12.0.21, <=12.1.0.alpha2, an HTTP/2 client may trigger the server to send RST_STREAM frames, for example by sending frames that are malformed or that should not be sent in a particular stream state, therefore forcing the server to consume resources such as CPU and memory.

For example, a client can open a stream and then send WINDOW_UPDATE frames with window size increment of 0, which is illegal.

Per specification https://www.rfc-editor.org/rfc/rfc9113.html#name-window_update , the server should send a RST_STREAM frame.

The client can now open another stream and send another bad WINDOW_UPDATE, therefore causing the server to consume more resources than necessary, as this case does not exceed the max number of concurrent streams, yet the client is able to create an enormous amount of streams in a short period of time.

The attack can be performed with other conditions (for example, a DATA frame for a closed stream) that cause the server to send a RST_STREAM frame.

Links:

* https://github.com/jetty/jetty.project/security/advisories/GHSA-mmxm-8w33-wc4h

CVE-2025-52999: jackson-core Has Potential for StackoverflowError if user parses an input file that contains very deeply nested data

jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. In versions prior to 2.15.0, if a user parses an input file and it has deeply nested data, Jackson could end up throwing a StackoverflowError if the depth is particularly large. jackson-core 2.15.0 contains a configurable limit for how deep Jackson will traverse in an input document, defaulting to an allowable depth of 1000. jackson-core will throw a StreamConstraintsException if the limit is reached. jackson-databind also benefits from this change because it uses jackson-core to parse JSON inputs. As a workaround, users should avoid parsing input files from untrusted sources.

CVE-2025-53864:

Connect2id Nimbus JOSE + JWT 10.0.x before 10.0.2 and 9.37.x before 9.37.4 allows a remote attacker to cause a denial of service via a deeply nested JSON object supplied in a JWT claim set, because of uncontrolled recursion. NOTE: this is independent of the Gson 2.11.0 issue because the Connect2id product could have checked the JSON object nesting depth, regardless of what limits (if any) were imposed by Gson.

CVE-2025-55752: Apache Tomcat: Directory traversal via rewrite with possible RCE if PUT is enabled

Relative Path Traversal vulnerability in Apache Tomcat.

The fix for bug 60013 introduced a regression where the rewritten URL was normalized before it was decoded. This introduced the possibility that, for rewrite rules that rewrite query parameters to the URL, an attacker could manipulate the request URI to bypass security constraints including the protection for /WEB-INF/ and /META-INF/. If PUT requests were also enabled then malicious files could be uploaded leading to remote code execution. PUT requests are normally limited to trusted users and it is considered unlikely that PUT requests would be enabled in conjunction with a rewrite that manipulated the URI.

This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.10, from 10.1.0-M1 through 10.1.44, from 9.0.0.M11 through 9.0.108.

The following versions were EOL at the time the CVE was created but are

known to be affected: 8.5.6 though 8.5.100. Other, older, EOL versions may also be affected.

Users are recommended to upgrade to version 11.0.11 or later, 10.1.45 or later or 9.0.109 or later, which fix the issue.

CVE-2025-55754: Apache Tomcat: console manipulation via escape sequences in log messages

Improper Neutralization of Escape, Meta, or Control Sequences vulnerability in Apache Tomcat.

Tomcat did not escape ANSI escape sequences in log messages. If Tomcat was running in a console on a Windows operating system, and the console supported ANSI escape sequences, it was possible for an attacker to use a specially crafted URL to inject ANSI escape sequences to manipulate the console and the clipboard and attempt to trick an administrator into running an attacker controlled command. While no attack vector was found, it may have been possible to mount this attack on other operating systems.

This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.10, from 10.1.0-M1 through 10.1.44, from 9.0.40 through 9.0.108.

The following versions were EOL at the time the CVE was created but are

known to be affected: 8.5.60 though 8.5.100. Other, older, EOL versions may also be affected.

Users are recommended to upgrade to version 11.0.11 or later, 10.1.45 or later or 9.0.109 or later, which fix the issue.

CVE-2025-58457: Apache ZooKeeper: Insufficient Permission Check in AdminServer Snapshot/Restore Commands

Improper permission check in ZooKeeper AdminServer lets authorized clients to run snapshot and restore command with insufficient permissions.

This issue affects Apache ZooKeeper: from 3.9.0 before 3.9.4.

Users are recommended to upgrade to version 3.9.4, which fixes the issue.

The issue can be mitigated by disabling both commands (via admin.snapshot.enabled and admin.restore.enabled), disabling the whole AdminServer interface (via admin.enableServer), or ensuring that the root ACL does not provide open permissions. (Note that ZooKeeper ACLs are not recursive, so this does not impact operations on child nodes besides notifications from recursive watches.)

CVE-2025-59250: JDBC Driver for SQL Server Spoofing Vulnerability

Improper input validation in JDBC Driver for SQL Server allows an unauthorized attacker to perform spoofing over a network.

CVE-2025-59340: jinjava Sandbox Bypass via JavaType-Based Deserialization

jinjava is a Java-based template engine based on django template syntax, adapted to render jinja templates. Priori to 2.8.1, by using mapper.getTypeFactory().constructFromCanonical(), it is possible to instruct the underlying ObjectMapper to deserialize attacker-controlled input into arbitrary classes. This enables the creation of semi-arbitrary class instances without directly invoking restricted methods or class literals. As a result, an attacker can escape the sandbox and instantiate classes such as java.net.URL, opening up the ability to access local files and URLs(e.g., file:///etc/passwd). With further chaining, this primitive can potentially lead to remote code execution (RCE). This vulnerability is fixed in 2.8.1.

CVE-2025-61795: Apache Tomcat: Delayed cleaning of multi-part upload temporary files may lead to DoS

Improper Resource Shutdown or Release vulnerability in Apache Tomcat.

If an error occurred (including exceeding limits) during the processing of a multipart upload, temporary copies of the uploaded parts written to disc were not cleaned up immediately but left for the garbage collection process to delete. Depending on JVM settings, application memory usage and application load, it was possible that space for the temporary copies of uploaded parts would be filled faster than GC cleared it, leading to a DoS.

This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.11, from 10.1.0-M1 through 10.1.46, from 9.0.0.M1 through 9.0.109.

The following versions were EOL at the time the CVE was created but are

known to be affected: 8.5.0 though 8.5.100. Other, older, EOL versions may also be affected.

Users are recommended to upgrade to version 11.0.12 or later, 10.1.47 or later or 9.0.110 or later which fixes the issue.

CVE-2025-66169: Apache Camel Neo4j: Cypher injection vulnerability in Camel-Neo4j component

Cypher Injection vulnerability in Apache Camel camel-neo4j component.

This issue affects Apache Camel: from 4.10.0 before 4.10.8, from 4.14.0 before 4.14.3, from 4.15.0 before 4.17.0

Users are recommended to upgrade to version 4.10.8 for 4.10.x LTS and 4.14.3 for 4.14.x LTS and 4.17.0.

CVE-2025-66524: Apache NiFi: Deserialization of Untrusted Data in GetAsanaObject Processor

Apache NiFi 1.20.0 through 2.6.0 include the GetAsanaObject Processor, which requires integration with a configurable Distribute Map Cache Client Service for storing and retrieving state information. The GetAsanaObject Processor used generic Java Object serialization and deserialization without filtering. Unfiltered Java object deserialization does not provide protection against crafted state information stored in the cache server configured for GetAsanaObject. Exploitation requires an Apache NiFi system running with the GetAsanaObject Processor, and direct access to the configured cache server. Upgrading to Apache NiFi 2.7.0 is the recommended mitigation, which replaces Java Object serialization with JSON serialization. Removing the GetAsanaObject Processor located in the nifi-asana-processors-nar bundle also prevents exploitation.

CVE-2026-1225: Malicious logback.xml configuration file allows instantiation of arbitrary classes

ACE vulnerability in configuration file processing by QOS.CH logback-core up to and including version 1.5.24 in Java applications, allows an attacker to instantiate classes already present on the class path by compromising an existing logback configuration file.

The instantiation of a potentially malicious Java class requires that said class is present on the user's class-path. In addition, the attacker must have write access to a

configuration file. However, after successful instantiation, the instance is very likely to be discarded with no further ado.

CVE-2026-22610: Angular has XSS Vulnerability via Unsanitized SVG Script Attributes

Angular is a development platform for building mobile and desktop web applications using TypeScript/JavaScript and other languages. Prior to versions 19.2.18, 20.3.16, 21.0.7, and 21.1.0-rc.0, a cross-site scripting (XSS) vulnerability has been identified in the Angular Template Compiler. The vulnerability exists because Angular’s internal sanitization schema fails to recognize the href and xlink:href attributes of SVG <script> elements as a Resource URL context. This issue has been patched in versions 19.2.18, 20.3.16, 21.0.7, and 21.1.0-rc.0.

CVE-2026-24015: Apache IoTDB: Insecure Default Configuration Vulnerability

A vulnerability in Apache IoTDB.

This issue affects Apache IoTDB: from 1.0.0 before 1.3.7, from 2.0.0 before 2.0.7.

Users are recommended to upgrade to version 1.3.7 or 2.0.7, which fixes the issue.

CVE-2026-24713: Apache IoTDB: JEXL Expression Injection Vulnerability

Improper Input Validation vulnerability in Apache IoTDB.

This issue affects Apache IoTDB: from 1.0.0 before 1.3.7, from 2.0.0 before 2.0.7.

Users are recommended to upgrade to version 1.3.7 or 2.0.7, which fixes the issue.

CVE-2026-25526: JinJava Bypass through ForTag leads to Arbitrary Java Execution

JinJava is a Java-based template engine based on django template syntax, adapted to render jinja templates. Prior to versions 2.7.6 and 2.8.3, JinJava is vulnerable to arbitrary Java execution via bypass through ForTag. This allows arbitrary Java class instantiation and file access bypassing built-in sandbox restrictions. This issue has been patched in versions 2.7.6 and 2.8.3.