Security Bulletin: Vulnerabilities found

Security Bulletin

Summary

Vulnerabilities were found in the product libraries. Customers should update to the fixed versions at the earliest.

Vulnerability Details

CVEID: CVE-2022-42889 *(https://www.cve.org/CVERecord?id=CVE-2022-42889)*
DESCRIPTION: Apache Commons Text performs variable interpolation, allowing properties to be dynamically evaluated and expanded. The standard format for interpolation is "${prefix:name}", where "prefix" is used to locate an instance of org.apache.commons.text.lookup.StringLookup that performs the interpolation. Starting with version 1.5 and continuing through 1.9, the set of default Lookup instances included interpolators that could result in arbitrary code execution or contact with remote servers. These lookups are: - "script" - execute expressions using the JVM script execution engine (javax.script) - "dns" - resolve dns records - "url" - load values from urls, including from remote servers Applications using the interpolation defaults in the affected versions may be vulnerable to remote code execution or unintentional contact with remote servers if untrusted configuration values are used. Users are recommended to upgrade to Apache Commons Text 1.10.0, which disables the problematic interpolators by default.
CWE: CWE-94: Improper Control of Generation of Code ('Code Injection')
(https://cwe.mitre.org/data/definitions/94.html)
CVSS Source: NVD
CVSS Base score: 9.8
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-46295 *(https://www.cve.org/CVERecord?id=CVE-2025-46295)*
DESCRIPTION: Apache Commons Text versions prior to 1.10.0 included interpolation features that could be abused when applications passed untrusted input into the text-substitution API. Because some interpolators could trigger actions like executing commands or accessing external resources, an attacker could potentially achieve remote code execution. This vulnerability has been fully addressed in FileMaker Server 22.0.4.
CWE: CWE-94: Improper Control of Generation of Code ('Code Injection')
(https://cwe.mitre.org/data/definitions/94.html)
CVSS Source: CISA ADP
CVSS Base score: 9.8
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2026-23745 *(https://www.cve.org/CVERecord?id=CVE-2026-23745)*

DESCRIPTION: node-tar is a Tar for Node.js. The node-tar library (= 7.5.2) fails to sanitize the linkpath of Link (hardlink) and SymbolicLink entries when preservePaths is false (the default secure behavior). This allows malicious archives to bypass the extraction root restriction, leading to Arbitrary File Overwrite via hardlinks and Symlink Poisoning via absolute symlink targets. This vulnerability is fixed in 7.5.3.

CWE: CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') *(https://cwe.mitre.org/data/definitions/22.html)*

CVSS Source: NVD

CVSS Base score: 6.1

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N)

CVEID: CVE-2026-23950 *(https://www.cve.org/CVERecord?id=CVE-2026-23950)*

DESCRIPTION: node-tar,a Tar for Node.js, has a race condition vulnerability in versions up to and including 7.5.3. This is due to an incomplete handling of Unicode path collisions in the `path-reservations` system. On case-insensitive or normalization-insensitive filesystems (such as macOS APFS, In which it has been tested), the library fails to lock colliding paths (e.g., `ß` and `ss`), allowing them to be processed in parallel. This bypasses the library's internal concurrency safeguards and permits Symlink Poisoning attacks via race conditions. The library uses a `PathReservations` system to ensure that metadata checks and file operations for the same path are serialized. This prevents race conditions where one entry might clobber another concurrently. This is a Race Condition which enables Arbitrary File Overwrite. This vulnerability affects users and systems using node-tar on macOS (APFS/HFS+). Because of using `NFD` Unicode normalization (in which `ß` and `ss` are different), conflicting paths do not have their order properly preserved under filesystems that ignore Unicode normalization (e.g., APFS (in which `ß` causes an inode collision with `ss`)). This enables an attacker to circumvent internal parallelization locks (`PathReservations`) using conflicting filenames within a malicious tar archive. The patch in version 7.5.4 updates `path-reservations.js` to use a normalization form that matches the target filesystem's behavior (e.g., `NFKD`), followed by first `toLocaleLowerCase('en')` and then `toLocaleUpperCase('en')`. As a workaround, users who cannot upgrade promptly, and who are programmatically using `node-tar` to extract arbitrary tarball data should filter out all `SymbolicLink` entries (as npm does) to defend against arbitrary file writes via this file system entry name collision issue.

CWE: CWE-176: Improper Handling of Unicode Encoding *(https://cwe.mitre.org/data/definitions/176.html)*

CVSS Source: NVD

CVSS Base score: 5.9

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N)

CVEID: CVE-2026-24400 *(https://www.cve.org/CVERecord?id=CVE-2026-24400)*

DESCRIPTION: AssertJ provides Fluent testing assertions for Java and the Java Virtual Machine (JVM). Starting in version 1.4.0 and prior to version 3.27.7, an XML External Entity (XXE) vulnerability exists in `org.assertj.core.util.xml.XmlStringPrettyFormatter`: the `toXmlDocument(String)` method initializes `DocumentBuilderFactory` with default settings, without disabling DTDs or external entities. This formatter is used by the `isXmlEqualTo(CharSequence)` assertion for `CharSequence` values. An application is vulnerable only when it uses untrusted XML input with either `isXmlEqualTo(CharSequence)` from `org.assertj.core.api.AbstractCharSequenceAssert` or `xmlPrettyFormat(String)` from

`org.assertj.core.util.xml.XmlStringPrettyFormatter`. If untrusted XML input is processed by tone of these methods, an attacker couldnread arbitrary local files via `file://` URIs (e.g., `/etc/passwd`, application configuration files); perform Server-Side Request Forgery (SSRF) via HTTP/HTTPS URIs, and/or cause Denial of Service via "Billion Laughs" entity expansion attacks. `isXmlEqualTo(CharSequence)` has been deprecated in favor of XMLUnit in version 3.18.0 and will be removed in version 4.0. Users of affected versions should, in order of preference: replace `isXmlEqualTo(CharSequence)` with XMLUnit, upgrade to version 3.27.7, or avoid using `isXmlEqualTo(CharSequence)` or `XmlStringPrettyFormatter` with untrusted input. `XmlStringPrettyFormatter` has historically been considered a utility for `isXmlEqualTo(CharSequence)` rather than a feature for AssertJ users, so it is deprecated in version 3.27.7 and removed in version 4.0, with no replacement.

CWE: CWE-611: Improper Restriction of XML External Entity Reference *(https://cwe.mitre.org/data/definitions/611.html)*

CVSS Source: NVD

CVSS Base score: 9.1

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:H)

CVEID: CVE-2026-25639 *(https://www.cve.org/CVERecord?id=CVE-2026-25639)*

DESCRIPTION: Axios is a promise based HTTP client for the browser and Node.js. Prior to versions 0.30.3 and 1.13.5, the mergeConfig function in axios crashes with a TypeError when processing configuration objects containing __proto__ as an own property. An attacker can trigger this by providing a malicious configuration object created via JSON.parse(), causing complete denial of service. This vulnerability is fixed in versions 0.30.3 and 1.13.5.

CWE: CWE-754: Improper Check for Unusual or Exceptional Conditions *(https://cwe.mitre.org/data/definitions/754.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-64718 *(https://www.cve.org/CVERecord?id=CVE-2025-64718)*

DESCRIPTION: js-yaml is a JavaScript YAML parser and dumper. In js-yaml before 4.1.1 and 3.14.2, it's possible for an attacker to modify the prototype of the result of a parsed yaml document via prototype pollution (`__proto__`). All users who parse untrusted yaml documents may be impacted. The problem is patched in js-yaml 4.1.1 and 3.14.2. Users can protect against this kind of attack on the server by using `node --disable-proto=delete` or `deno` (in Deno, pollution protection is on by default).

CWE: CWE-1321: Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution') *(https://cwe.mitre.org/data/definitions/1321.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVEID: CVE-2026-27212 *(https://www.cve.org/CVERecord?id=CVE-2026-27212)*

DESCRIPTION: Swiper is a free and mobile touch slider with hardware accelerated transitions and native behavior. Versions 6.5.1 through 12.1.1 have a Prototype pollution vulnerability. The vulnerability resides in line 94 of shared/utils.mjs, where the indexOf() function is used to check whether user provided input contain

forbidden strings. Despite a previous fix that attempted to mitigate prototype pollution by checking whether user input contained a forbidden key, it is still possible to pollute Object.prototype via a crafted input using Array.prototype. The exploit works across Windows and Linux and on Node and Bun runtimes. Any application that processes attacker-controlled input using this package may be affected by the following: Authentication Bypass, Denial of Service and RCE. This issue is fixed in version 12.1.2.

CWE: CWE-1321: Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution') *(https://cwe.mitre.org/data/definitions/1321.html)*

CVSS Source: NVD

CVSS Base score: 7.8

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2026-0994 *(https://www.cve.org/CVERecord?id=CVE-2026-0994)*

DESCRIPTION: A denial-of-service (DoS) vulnerability exists in google.protobuf.json_format.ParseDict() in Python, where the max_recursion_depth limit can be bypassed when parsing nested google.protobuf.Any messages. Due to missing recursion depth accounting inside the internal Any-handling logic, an attacker can supply deeply nested Any structures that bypass the intended recursion limit, eventually exhausting Python's recursion stack and causing a RecursionError.

CWE: CWE-674: Uncontrolled Recursion *(https://cwe.mitre.org/data/definitions/674.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2022-1471 *(https://www.cve.org/CVERecord?id=CVE-2022-1471)*

DESCRIPTION: SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: CVE.org

CVSS Base score: 8.3

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L)

CVEID: CVE-2022-25857 *(https://www.cve.org/CVERecord?id=CVE-2022-25857)*

DESCRIPTION: The package org.yaml:snakeyaml from 0 and before 1.31 are vulnerable to Denial of Service (DoS) due missing to nested depth limitation for collections.

CWE: CWE-776: Improper Restriction of Recursive Entity References in DTDs ('XML Entity Expansion') *(https://cwe.mitre.org/data/definitions/776.html)*

CVSS Source: IBM X-Force

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2022-38749 *(https://www.cve.org/CVERecord?id=CVE-2022-38749)*

DESCRIPTION: Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.

CWE: CWE-121: Stack-based Buffer Overflow *(https://cwe.mitre.org/data/definitions/121.html)*

CVSS Source: IBM X-Force

CVSS Base score: 3.3

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:L)

CVEID: CVE-2022-38750 *(https://www.cve.org/CVERecord?id=CVE-2022-38750)*

DESCRIPTION: Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.

CWE: CWE-121: Stack-based Buffer Overflow *(https://cwe.mitre.org/data/definitions/121.html)*

CVSS Source: IBM X-Force

CVSS Base score: 3.3

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:L)

CVEID: CVE-2022-38751 *(https://www.cve.org/CVERecord?id=CVE-2022-38751)*

DESCRIPTION: Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.

CWE: CWE-121: Stack-based Buffer Overflow *(https://cwe.mitre.org/data/definitions/121.html)*

CVSS Source: IBM X-Force

CVSS Base score: 3.3

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:L)

CVEID: CVE-2022-38752 *(https://www.cve.org/CVERecord?id=CVE-2022-38752)*

DESCRIPTION: Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack-overflow.

CWE: CWE-121: Stack-based Buffer Overflow *(https://cwe.mitre.org/data/definitions/121.html)*

CVSS Source: IBM X-Force

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2022-41854 *(https://www.cve.org/CVERecord?id=CVE-2022-41854)*

DESCRIPTION: Those using Snakeyaml to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack overflow. This effect may support a denial of service attack.

CWE: CWE-121: Stack-based Buffer Overflow *(https://cwe.mitre.org/data/definitions/121.html)*

CVSS Source: CVE.org
CVSS Base score: 5.8
CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:L/UI:R/S:C/C:N/I:N/A:H)

CVEID: CVE-2025-67735 *(https://www.cve.org/CVERecord?id=CVE-2025-67735)*
DESCRIPTION: Netty is an asynchronous, event-driven network application framework. In versions prior to 4.1.129.Final and 4.2.8.Final, the `io.netty.handler.codec.http.HttpRequestEncoder` has a CRLF injection with the request URI when constructing a request. This leads to request smuggling when `HttpRequestEncoder` is used without proper sanitization of the URI. Any application / framework using `HttpRequestEncoder` can be subject to be abused to perform request smuggling using CRLF injection. Versions 4.1.129.Final and 4.2.8.Final fix the issue.
CWE: CWE-93: Improper Neutralization of CRLF Sequences ('CRLF Injection') *(https://cwe.mitre.org/data/definitions/93.html)*
CVSS Source: security-advisories@github.com
CVSS Base score: 6.5
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVEID: CVE-2025-13466 *(https://www.cve.org/CVERecord?id=CVE-2025-13466)*
DESCRIPTION: body-parser 2.2.0 is vulnerable to denial of service due to inefficient handling of URL-encoded bodies with very large numbers of parameters. An attacker can send payloads containing thousands of parameters within the default 100KB request size limit, causing elevated CPU and memory usage. This can lead to service slowdown or partial outages under sustained malicious traffic. This issue is addressed in version 2.2.1.
CWE: CWE-400: Uncontrolled Resource Consumption *(https://cwe.mitre.org/data/definitions/400.html)*
CVSS Source: openjs
CVSS Base score: 5.5
CVSS Vector: (CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:L/E:P/AU:Y)

CVEID: CVE-2021-31294 *(https://www.cve.org/CVERecord?id=CVE-2021-31294)*
DESCRIPTION: Redis before 6cbea7d allows a replica to cause an assertion failure in a primary server by sending a non-administrative command (specifically, a SET command). NOTE: this was fixed for Redis 6.2.x and 7.x in 2021. Versions before 6.2 were not intended to have safety guarantees related to this.
CWE: CWE-617: Reachable Assertion *(https://cwe.mitre.org/data/definitions/617.html)*
CVSS Source: IBM X-Force
CVSS Base score: 7.5
CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-15284 *(https://www.cve.org/CVERecord?id=CVE-2025-15284)*
DESCRIPTION: Improper Input Validation vulnerability in qs (parse modules) allows HTTP DoS.This issue affects qs: 6.14.1. Summary The arrayLimit option in qs did not enforce limits for bracket notation (a[]=1&a[]=2), only for indexed notation (a[0]=1). This is a consistency bug; arrayLimit should apply uniformly across all array

notations. Note: The default parameterLimit of 1000 effectively mitigates the DoS scenario originally described. With default options, bracket notation cannot produce arrays larger than parameterLimit regardless of arrayLimit, because each a[]=valueconsumes one parameter slot. The severity has been reduced accordingly. Details The arrayLimit option only checked limits for indexed notation (a[0]=1&a[1]=2) but did not enforce it for bracket notation (a[]=1&a[]=2). Vulnerable code (lib/parse.js:159-162): if (root === '[]' && options.parseArrays) { obj = utils.combine([], leaf); // No arrayLimit check } Working code (lib/parse.js:175): else if (index = options.arrayLimit) { // Limit checked here obj = []; obj[index] = leaf; } The bracket notation handler at line 159 uses utils.combine([], leaf) without validating against options.arrayLimit, while indexed notation at line 175 checks index = options.arrayLimit before creating arrays. PoC const qs = require('qs'); const result = qs.parse('a[]=1&a[]=2&a[]=3&a[]=4&a[]=5&a[]=6', { arrayLimit: 5 }); console.log(result.a.length); // Output: 6 (should be max 5) Note on parameterLimit interaction: The original advisory's "DoS demonstration" claimed a length of 10,000, but parameterLimit (default: 1000) caps parsing to 1,000 parameters. With default options, the actual output is 1,000, not 10,000. Impact Consistency bug in arrayLimit enforcement. With default parameterLimit, the practical DoS risk is negligible since parameterLimit already caps the total number of parsed parameters (and thus array elements from bracket notation). The risk increases only when parameterLimit is explicitly set to a very high value.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: harborist

CVSS Base score: 3.7

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2026-2391 *(https://www.cve.org/CVERecord?id=CVE-2026-2391)*

DESCRIPTION: ### Summary The `arrayLimit` option in qs does not enforce limits for comma-separated values when `comma: true` is enabled, allowing attackers to cause denial-of-service via memory exhaustion. This is a bypass of the array limit enforcement, similar to the bracket notation bypass addressed in GHSA-6rw7-vpxm-498p (CVE-2025-15284). ### Details When the `comma` option is set to `true` (not the default, but configurable in applications), qs allows parsing comma-separated strings as arrays (e.g., `?param=a,b,c` becomes `['a', 'b', 'c']`). However, the limit check for `arrayLimit` (default: 20) and the optional throwOnLimitExceeded occur after the comma-handling logic in `parseArrayValue`, enabling a bypass. This permits creation of arbitrarily large arrays from a single parameter, leading to excessive memory allocation. **Vulnerable code** (lib/parse.js: lines ~40-50): ```js if (val && typeof val === 'string' && options.comma && val.indexOf(',') -1) { return val.split(','); } if (options.throwOnLimitExceeded && currentArrayLength = options.arrayLimit) { throw new RangeError('Array limit exceeded. Only ' + options.arrayLimit + ' element' + (options.arrayLimit === 1 ? '' : 's') + ' allowed in an array.'); } return val; ``` The `split(',')` returns the array immediately, skipping the subsequent limit check. Downstream merging via `utils.combine` does not prevent allocation, even if it marks overflows for sparse arrays.This discrepancy allows attackers to send a single parameter with millions of commas (e.g., `?param=,,,,,,,,...`), allocating massive arrays in memory without triggering limits. It bypasses the intent of `arrayLimit`, which is enforced correctly for indexed (`a[0]=`) and bracket (`a[]=`) notations (the latter fixed in v6.14.1 per GHSA-6rw7-vpxm-498p). ### PoC **Test 1 - Basic bypass:** ``` npm install qs ``` ```js const qs = require('qs'); const payload = 'a=' + ','.repeat(25); // 26 elements after split (bypasses arrayLimit: 5) const options = { comma: true, arrayLimit: 5, throwOnLimitExceeded: true };

try { const result = qs.parse(payload, options); console.log(result.a.length); // Outputs: 26 (bypass successful) } catch (e) { console.log('Limit enforced:', e.message); // Not thrown } ``` **Configuration:** - `comma: true` - `arrayLimit: 5` - `throwOnLimitExceeded: true` Expected: Throws "Array limit exceeded" error. Actual: Parses successfully, creating an array of length 26. ### Impact Denial of Service (DoS) via memory exhaustion.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-14009 *(https://www.cve.org/CVERecord?id=CVE-2025-14009)*

DESCRIPTION: A critical vulnerability exists in the NLTK downloader component of nltk/nltk, affecting all versions. The _unzip_iter function in nltk/downloader.py uses zipfile.extractall() without performing path validation or security checks. This allows attackers to craft malicious zip packages that, when downloaded and extracted by NLTK, can execute arbitrary code. The vulnerability arises because NLTK assumes all downloaded packages are trusted and extracts them without validation. If a malicious package contains Python files, such as __init__.py, these files are executed automatically upon import, leading to remote code execution. This issue can result in full system compromise, including file system access, network access, and potential persistence mechanisms.

CWE: CWE-94: Improper Control of Generation of Code ('Code Injection') *(https://cwe.mitre.org/data/definitions/94.html)*

CVSS Source: security@huntr.dev

CVSS Base score: 10

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)

CVEID: CVE-2024-29025 *(https://www.cve.org/CVERecord?id=CVE-2024-29025)*

DESCRIPTION: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. The `HttpPostRequestDecoder` can be tricked to accumulate data. While the decoder can store items on the disk if configured so, there are no limits to the number of fields the form can have, an attacher can send a chunked post consisting of many small fields that will be accumulated in the `bodyListHttpData` list. The decoder cumulates bytes in the `undecodedChunk` buffer until it can decode a field, this field can cumulate data without limits. This vulnerability is fixed in 4.1.108.Final.

CWE: CWE-770: Allocation of Resources Without Limits or Throttling *(https://cwe.mitre.org/data/definitions/770.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2025-58056 *(https://www.cve.org/CVERecord?id=CVE-2025-58056)*

DESCRIPTION: Netty is an asynchronous event-driven network application framework for development of maintainable high performance protocol servers and clients. In versions 4.1.124.Final, and 4.2.0.Alpha3 through 4.2.4.Final, Netty incorrectly accepts standalone newline characters (LF) as a chunk-size line terminator, regardless of a preceding carriage return (CR), instead of requiring CRLF per HTTP/1.1 standards. When

combined with reverse proxies that parse LF differently (treating it as part of the chunk extension), attackers can craft requests that the proxy sees as one request but Netty processes as two, enabling request smuggling attacks. This is fixed in versions 4.1.125.Final and 4.2.5.Final.

CWE: CWE-444: Inconsistent Interpretation of HTTP Requests ('HTTP Request/Response Smuggling') *(https://cwe.mitre.org/data/definitions/444.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N)

CVEID: CVE-2025-58057 *(https://www.cve.org/CVERecord?id=CVE-2025-58057)*

DESCRIPTION: Netty is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. In netty-codec-compression versions 4.1.124.Final and below, and netty-codec versions 4.2.4.Final and below, when supplied with specially crafted input, BrotliDecoder and certain other decompression decoders will allocate a large number of reachable byte buffers, which can lead to denial of service. BrotliDecoder.decompress has no limit in how often it calls pull, decompressing data 64K bytes at a time. The buffers are saved in the output list, and remain reachable until OOM is hit. This is fixed in versions 4.1.125.Final of netty-codec and 4.2.5.Final of netty-codec-compression.

CWE: CWE-409: Improper Handling of Highly Compressed Data (Data Amplification) *(https://cwe.mitre.org/data/definitions/409.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2018-16487 *(https://www.cve.org/CVERecord?id=CVE-2018-16487)*

DESCRIPTION: A prototype pollution vulnerability was found in lodash 4.17.11 where the functions merge, mergeWith, and defaultsDeep can be tricked into adding or modifying properties of Object.prototype.

CWE: CWE-400: Uncontrolled Resource Consumption *(https://cwe.mitre.org/data/definitions/400.html)*

CVSS Source: IBM X-Force

CVSS Base score: 6.5

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:L)

CVEID: CVE-2025-68161 *(https://www.cve.org/CVERecord?id=CVE-2025-68161)*

DESCRIPTION: The Socket Appender in Apache Log4j Core versions 2.0-beta9 through 2.25.2 does not perform TLS hostname verification of the peer certificate, even when the verifyHostName https://logging.apache.org/log4j/2.x/manual/appenders/network.html#SslC… *(https://logging.apache.org/log4j/2.x/manual/appenders/network.html#SslConfiguration-attr-verifyHostName)* configuration attribute or the log4j2.sslVerifyHostName https://logging.apache.org/log4j/2.x/manual/systemproperties.html#log4j… *(https://logging.apache.org/log4j/2.x/manual/systemproperties.html#log4j2.sslVerifyHostName)* system property is set to true. This issue may allow a man-in-the-middle attacker to intercept or redirect log traffic under the following conditions: * The attacker is able to intercept or redirect network traffic between the client and the log receiver. * The attacker can present a server certificate issued by a certification authority trusted by the Socket Appender's configured

trust store (or by the default Java trust store if no custom trust store is configured). Users are advised to upgrade to Apache Log4j Core version 2.25.3, which addresses this issue. As an alternative mitigation, the Socket Appender may be configured to use a private or restricted trust root to limit the set of trusted certificates.

CWE: CWE-297: Improper Validation of Certificate with Host Mismatch *(https://cwe.mitre.org/data/definitions/297.html)*

CVSS Source: NVD

CVSS Base score: 4.8

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVEID: CVE-2026-21945 *(https://www.cve.org/CVERecord?id=CVE-2026-21945)*

DESCRIPTION: Java SE is vulnerable to a denial of service, caused by an easily exploitable vulnerability issue that allows an remote attacker to cause a hang or repeatable crash of the application.

CWE: CWE-400: Uncontrolled Resource Consumption *(https://cwe.mitre.org/data/definitions/400.html)*

CVSS Source: secalert_us@oracle.com

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-21932 *(https://www.cve.org/CVERecord?id=CVE-2026-21932)*

DESCRIPTION: Java SE could allow a remote attacker to bypass security controls and create, delete, or modify critical data or all accessible data, caused by an easily exploitable vulnerability.

CVSS Source: secalert_us@oracle.com

CVSS Base score: 7.4

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:N/I:H/A:N)

CVEID: CVE-2026-21933 *(https://www.cve.org/CVERecord?id=CVE-2026-21933)*

DESCRIPTION: Java SE could allow a remote attacker to bypass security controls and perform unauthorized update, insert, delete, or read operations on accessible data, caused by an easily exploitable vulnerability.

CVSS Source: secalert_us@oracle.com

CVSS Base score: 6.1

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N)

CVEID: CVE-2026-21925 *(https://www.cve.org/CVERecord?id=CVE-2026-21925)*

DESCRIPTION: Java SE could allow a remote unauthenticated attacker to bypass security controls and perform unauthorized update, insert, delete, or read operations on accessible data, caused by an difficult to exploit vulnerability.

CVSS Source: secalert_us@oracle.com

CVSS Base score: 4.8

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N)

CVEID: CVE-2026-1188 *(https://www.cve.org/CVERecord?id=CVE-2026-1188)*

DESCRIPTION: In the Eclipse OMR port library component since release 0.2.0, an API function to return the textual names of all supported processor features was not accounting for the separator inserted between

processor features. If the output buffer supplied to this function was incorrectly sized, failing to account for the separator when determining when a write to the buffer was safe could lead to a buffer overflow. This issue is fixed in Eclipse OMR version 0.8.0.

CWE: CWE-131: Incorrect Calculation of Buffer Size *(https://cwe.mitre.org/data/definitions/131.html)*

CVSS Source: NVD

CVSS Base score: 9.8

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-7339 *(https://www.cve.org/CVERecord?id=CVE-2025-7339)*

DESCRIPTION: on-headers is a node.js middleware for listening to when a response writes headers. A bug in on-headers versions `1.1.0` may result in response headers being inadvertently modified when an array is passed to `response.writeHead()`. Users should upgrade to version 1.1.0 to receive a patch. Uses are strongly encouraged to upgrade to `1.1.0`, but this issue can be worked around by passing an object to `response.writeHead()` rather than an array.

CWE: CWE-241: Improper Handling of Unexpected Data Type *(https://cwe.mitre.org/data/definitions/241.html)*

CVSS Source: openjs

CVSS Base score: 3.4

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N)

CVEID: CVE-2024-3884 *(https://www.cve.org/CVERecord?id=CVE-2024-3884)*

DESCRIPTION: A flaw was found in Undertow that can cause remote denial of service attacks. When the server uses the FormEncodedDataDefinition.doParse(StreamSourceChannel) method to parse large form data encoding with application/x-www-form-urlencoded, the method will cause an OutOfMemory issue. This flaw allows unauthorized users to cause a remote denial of service (DoS) attack.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: secalert@redhat.com

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-66418 *(https://www.cve.org/CVERecord?id=CVE-2025-66418)*

DESCRIPTION: urllib3 is a user-friendly HTTP client library for Python. Starting in version 1.24 and prior to 2.6.0, the number of links in the decompression chain was unbounded allowing a malicious server to insert a virtually unlimited number of compression steps leading to high CPU usage and massive memory allocation for the decompressed data. This vulnerability is fixed in 2.6.0.

CWE: CWE-770: Allocation of Resources Without Limits or Throttling *(https://cwe.mitre.org/data/definitions/770.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-66471 *(https://www.cve.org/CVERecord?id=CVE-2025-66471)*

DESCRIPTION: urllib3 is a user-friendly HTTP client library for Python. Starting in version 1.0 and prior to

2.6.0, the Streaming API improperly handles highly compressed data. urllib3's streaming API is designed for the efficient handling of large HTTP responses by reading the content in chunks, rather than loading the entire response body into memory at once. When streaming a compressed response, urllib3 can perform decoding or decompression based on the HTTP Content-Encoding header (e.g., gzip, deflate, br, or zstd). The library must read compressed data from the network and decompress it until the requested chunk size is met. Any resulting decompressed data that exceeds the requested amount is held in an internal buffer for the next read operation. The decompression logic could cause urllib3 to fully decode a small amount of highly compressed data in a single operation. This can result in excessive resource consumption (high CPU usage and massive memory allocation for the decompressed data.

CWE: CWE-409: Improper Handling of Highly Compressed Data (Data Amplification)

(https://cwe.mitre.org/data/definitions/409.html)

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-21441 *(https://www.cve.org/CVERecord?id=CVE-2026-21441)*

DESCRIPTION: urllib3 is an HTTP client library for Python. urllib3's streaming API is designed for the efficient handling of large HTTP responses by reading the content in chunks, rather than loading the entire response body into memory at once. urllib3 can perform decoding or decompression based on the HTTP `Content-Encoding` header (e.g., `gzip`, `deflate`, `br`, or `zstd`). When using the streaming API, the library decompresses only the necessary bytes, enabling partial content consumption. Starting in version 1.22 and prior to version 2.6.3, for HTTP redirect responses, the library would read the entire response body to drain the connection and decompress the content unnecessarily. This decompression occurred even before any read methods were called, and configured read limits did not restrict the amount of decompressed data. As a result, there was no safeguard against decompression bombs. A malicious server could exploit this to trigger excessive resource consumption on the client. Applications and libraries are affected when they stream content from untrusted sources by setting `preload_content=False` when they do not disable redirects. Users should upgrade to at least urllib3 v2.6.3, in which the library does not decode content of redirect responses when `preload_content=False`. If upgrading is not immediately possible, disable redirects by setting `redirect=False` for requests to untrusted source.

CWE: CWE-409: Improper Handling of Highly Compressed Data (Data Amplification)

(https://cwe.mitre.org/data/definitions/409.html)

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-66030 *(https://www.cve.org/CVERecord?id=CVE-2025-66030)*

DESCRIPTION: Forge (also called `node-forge`) is a native implementation of Transport Layer Security in JavaScript. An Integer Overflow vulnerability in node-forge versions 1.3.1 and below enables remote, unauthenticated attackers to craft ASN.1 structures containing OIDs with oversized arcs. These arcs may be decoded as smaller, trusted OIDs due to 32-bit bitwise truncation, enabling the bypass of downstream OID-

based security decisions. This issue has been patched in version 1.3.2.

CWE: CWE-190: Integer Overflow or Wraparound *(https://cwe.mitre.org/data/definitions/190.html)*

CVSS Source: NVD

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N)

CVEID: CVE-2025-66031 *(https://www.cve.org/CVERecord?id=CVE-2025-66031)*

DESCRIPTION: Forge (also called `node-forge`) is a native implementation of Transport Layer Security in JavaScript. An Uncontrolled Recursion vulnerability in node-forge versions 1.3.1 and below enables remote, unauthenticated attackers to craft deep ASN.1 structures that trigger unbounded recursive parsing. This leads to a Denial-of-Service (DoS) via stack exhaustion when parsing untrusted DER inputs. This issue has been patched in version 1.3.2.

CWE: CWE-674: Uncontrolled Recursion *(https://cwe.mitre.org/data/definitions/674.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-33228 *(https://www.cve.org/CVERecord?id=CVE-2026-33228)*

DESCRIPTION: flatted is a circular JSON parser. Prior to version 3.4.2, the parse() function in flatted can use attacker-controlled string values from the parsed JSON as direct array index keys, without validating that they are numeric. Since the internal input buffer is a JavaScript Array, accessing it with the key "__proto__" returns Array.prototype via the inherited getter. This object is then treated as a legitimate parsed value and assigned as a property of the output object, effectively leaking a live reference to Array.prototype to the consumer. Any code that subsequently writes to that property will pollute the global prototype. This issue has been patched in version 3.4.2.

CWE: CWE-1321: Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution') *(https://cwe.mitre.org/data/definitions/1321.html)*

CVSS Source: NVD

CVSS Base score: 9.8

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2023-3223 *(https://www.cve.org/CVERecord?id=CVE-2023-3223)*

DESCRIPTION: A flaw was found in undertow. Servlets annotated with @MultipartConfig may cause an OutOfMemoryError due to large multipart content. This may allow unauthorized users to cause remote Denial of Service (DoS) attack. If the server uses fileSizeThreshold to limit the file size, it's possible to bypass the limit by setting the file name in the request to null.

CWE: CWE-789: Memory Allocation with Excessive Size Value *(https://cwe.mitre.org/data/definitions/789.html)*

CVSS Source: IBM X-Force

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2024-3653 *(https://www.cve.org/CVERecord?id=CVE-2024-3653)*

DESCRIPTION: A vulnerability was found in Undertow. This issue requires enabling the learning-push handler in the server's config, which is disabled by default, leaving the maxAge config in the handler unconfigured. The default is -1, which makes the handler vulnerable. If someone overwrites that config, the server is not subject to the attack. The attacker needs to be able to reach the server with a normal HTTP request.

CWE: CWE-401: Missing Release of Memory after Effective Lifetime *(https://cwe.mitre.org/data/definitions/401.html)*

CVSS Source: CVE.org

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2025-12383 *(https://www.cve.org/CVERecord?id=CVE-2025-12383)*

DESCRIPTION: In Eclipse Jersey versions 2.45, 3.0.16, 3.1.9 a race condition can cause ignoring of critical SSL configurations - such as mutual authentication, custom key/trust stores, and other security settings. This issue may result in SSLHandshakeException under normal circumstances, but under certain conditions, it could lead to unauthorized trust in insecure servers (see PoC)

CWE: CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') *(https://cwe.mitre.org/data/definitions/362.html)*

CVSS Source: NVD

CVSS Base score: 7.4

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N)

CVEID: CVE-2022-45688 *(https://www.cve.org/CVERecord?id=CVE-2022-45688)*

DESCRIPTION: A stack overflow in the XML.toJSONObject component of hutool-json v5.8.10 allows attackers to cause a Denial of Service (DoS) via crafted JSON or XML data.

CWE: CWE-787: Out-of-bounds Write *(https://cwe.mitre.org/data/definitions/787.html)*

CVSS Source: IBM X-Force

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2023-5072 *(https://www.cve.org/CVERecord?id=CVE-2023-5072)*

DESCRIPTION: Denial of Service in JSON-Java versions up to and including 20230618. A bug in the parser means that an input string of modest size can lead to indefinite amounts of memory being used.

CWE: CWE-770: Allocation of Resources Without Limits or Throttling *(https://cwe.mitre.org/data/definitions/770.html)*

CVSS Source: IBM X-Force

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-26996 *(https://www.cve.org/CVERecord?id=CVE-2026-26996)*

DESCRIPTION: minimatch is a minimal matching utility for converting glob expressions into JavaScript RegExp objects. Versions 10.2.0 and below are vulnerable to Regular Expression Denial of Service (ReDoS) when a glob pattern contains many consecutive * wildcards followed by a literal character that doesn't appear in

the test string. Each * compiles to a separate [^/]*? regex group, and when the match fails, V8's regex engine backtracks exponentially across all possible splits. The time complexity is O(4^N) where N is the number of * characters. With N=15, a single minimatch() call takes ~2 seconds. With N=34, it hangs effectively forever. Any application that passes user-controlled strings to minimatch() as the pattern argument is vulnerable to DoS. This issue has been fixed in version 10.2.1.

CWE: CWE-1333: Inefficient Regular Expression Complexity *(https://cwe.mitre.org/data/definitions/1333.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-21884 *(https://www.cve.org/CVERecord?id=CVE-2026-21884)*

DESCRIPTION: React Router is a router for React. In @remix-run/react version prior to 2.17.3. and react-router 7.0.0 through 7.11.0, a XSS vulnerability exists in in React Router's ScrollRestoration API in Framework Mode when using the getKey/storageKey props during Server-Side Rendering which could allow arbitrary JavaScript execution during SSR if untrusted content is used to generate the keys. There is no impact if server-side rendering in Framework Mode is disabled, or if Declarative Mode (BrowserRouter) or Data Mode (createBrowserRouter/RouterProvider) is being used. This issue has been patched in @remix-run/react version 2.17.3 and react-router version 7.12.0.

CWE: CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') *(https://cwe.mitre.org/data/definitions/79.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 8.2

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:N)

CVEID: CVE-2026-22029 *(https://www.cve.org/CVERecord?id=CVE-2026-22029)*

DESCRIPTION: React Router is a router for React. In @remix-run/router version prior to 1.23.2. and react-router 7.0.0 through 7.11.0, React Router (and Remix v1/v2) SPA open navigation redirects originating from loaders or actions in Framework Mode, Data Mode, or the unstable RSC modes can result in unsafe URLs causing unintended javascript execution on the client. This is only an issue if you are creating redirect paths from untrusted content or via an open redirect. There is no impact if Declarative Mode (BrowserRouter) is being used. This issue has been patched in @remix-run/router version 1.23.2 and react-router version 7.12.0.

CWE: CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') *(https://cwe.mitre.org/data/definitions/79.html)*

CVSS Source: NVD

CVSS Base score: 6.1

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N)

CVEID: CVE-2026-22030 *(https://www.cve.org/CVERecord?id=CVE-2026-22030)*

DESCRIPTION: React Router is a router for React. In @remix-run/server-runtime version prior to 2.17.3. and react-router 7.0.0 through 7.11.0, React Router (or Remix v2) is vulnerable to CSRF attacks on document POST requests to UI routes when using server-side route action handlers in Framework Mode, or when using React

Server Actions in the new unstable RSC modes. There is no impact if Declarative Mode (BrowserRouter) or Data Mode (createBrowserRouter/RouterProvider) is being used. This issue has been patched in @remix-run/server-runtime version 2.17.3 and react-router version 7.12.0.

CWE: CWE-346: Origin Validation Error *(https://cwe.mitre.org/data/definitions/346.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 6.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N)

CVEID: CVE-2026-25896 *(https://www.cve.org/CVERecord?id=CVE-2026-25896)*

DESCRIPTION: fast-xml-parser allows users to validate XML, parse XML to JS object, or build XML from JS object without C/C++ based libraries and no callback. From 4.1.3to before 5.3.5, a dot (.) in a DOCTYPE entity name is treated as a regex wildcard during entity replacement, allowing an attacker to shadow built-in XML entities (<, >, &, ", ') with arbitrary values. This bypasses entity encoding and leads to XSS when parsed output is rendered. This vulnerability is fixed in 5.3.5.

CWE: CWE-185: Incorrect Regular Expression *(https://cwe.mitre.org/data/definitions/185.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 9.3

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:N)

CVEID: CVE-2025-2998 *(https://www.cve.org/CVERecord?id=CVE-2025-2998)*

DESCRIPTION: A vulnerability was found in PyTorch 2.6.0. It has been declared as critical. Affected by this vulnerability is the function torch.nn.utils.rnn.pad_packed_sequence. The manipulation leads to memory corruption. Local access is required to approach this attack. The exploit has been disclosed to the public and may be used.

CWE: CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer *(https://cwe.mitre.org/data/definitions/119.html)*

CVSS Source: cna@vuldb.com

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L)

CVEID: CVE-2025-2999 *(https://www.cve.org/CVERecord?id=CVE-2025-2999)*

DESCRIPTION: A vulnerability was found in PyTorch 2.6.0. It has been rated as critical. Affected by this issue is the function torch.nn.utils.rnn.unpack_sequence. The manipulation leads to memory corruption. Attacking locally is a requirement. The exploit has been disclosed to the public and may be used.

CWE: CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer *(https://cwe.mitre.org/data/definitions/119.html)*

CVSS Source: cna@vuldb.com

CVSS Base score: 5.3

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L)

CVEID: CVE-2025-63396 *(https://www.cve.org/CVERecord?id=CVE-2025-63396)*

DESCRIPTION: An issue was discovered in PyTorch v2.5 and v2.7.1. Omission of profiler.stop() can cause torch.profiler.profile (PythonTracer) to crash or hang during finalization, leading to a Denial of Service (DoS).
CWE: CWE-667: Improper Locking *(https://cwe.mitre.org/data/definitions/667.html)*
CVSS Source: CISA ADP
CVSS Base score: 3.3
CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2026-24842 *(https://www.cve.org/CVERecord?id=CVE-2026-24842)*
DESCRIPTION: node-tar,a Tar for Node.js, contains a vulnerability in versions prior to 7.5.7 where the security check for hardlink entries uses different path resolution semantics than the actual hardlink creation logic. This mismatch allows an attacker to craft a malicious TAR archive that bypasses path traversal protections and creates hardlinks to arbitrary files outside the extraction directory. Version 7.5.7 contains a fix for the issue.
CWE: CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') *(https://cwe.mitre.org/data/definitions/22.html)*
CVSS Source: security-advisories@github.com
CVSS Base score: 8.2
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:N)

CVEID: CVE-2025-66221 *(https://www.cve.org/CVERecord?id=CVE-2025-66221)*
DESCRIPTION: Werkzeug is a comprehensive WSGI web application library. Prior to version 3.1.4, Werkzeug's safe_join function allows path segments with Windows device names. On Windows, there are special device names such as CON, AUX, etc that are implicitly present and readable in every directory. send_from_directory uses safe_join to safely serve files at user-specified paths under a directory. If the application is running on Windows, and the requested path ends with a special device name, the file will be opened successfully, but reading will hang indefinitely. This issue has been patched in version 3.1.4.
CWE: CWE-67: Improper Handling of Windows Device Names *(https://cwe.mitre.org/data/definitions/67.html)*
CVSS Source: NVD
CVSS Base score: 5.3
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2026-22701 *(https://www.cve.org/CVERecord?id=CVE-2026-22701)*
DESCRIPTION: filelock is a platform-independent file lock for Python. Prior to version 3.20.3, a TOCTOU race condition vulnerability exists in the SoftFileLock implementation of the filelock package. An attacker with local filesystem access and permission to create symlinks can exploit a race condition between the permission validation and file creation to cause lock operations to fail or behave unexpectedly. The vulnerability occurs in the _acquire() method between raise_on_not_writable_file() (permission check) and os.open() (file creation). During this race window, an attacker can create a symlink at the lock file path, potentially causing the lock to operate on an unintended target file or leading to denial of service. This issue has been patched in version 3.20.3.
CWE: CWE-59: Improper Link Resolution Before File Access ('Link Following') *(https://cwe.mitre.org/data/definitions/59.html)*

CVSS Source: security-advisories@github.com
CVSS Base score: 5.3
CVSS Vector: (CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:L/A:H)

CVEID: CVE-2025-68146 *(https://www.cve.org/CVERecord?id=CVE-2025-68146)*
DESCRIPTION: filelock is a platform-independent file lock for Python. In versions prior to 3.20.1, a Time-of-Check-Time-of-Use (TOCTOU) race condition allows local attackers to corrupt or truncate arbitrary user files through symlink attacks. The vulnerability exists in both Unix and Windows lock file creation where filelock checks if a file exists before opening it with O_TRUNC. An attacker can create a symlink pointing to a victim file in the time gap between the check and open, causing os.open() to follow the symlink and truncate the target file. All users of filelock on Unix, Linux, macOS, and Windows systems are impacted. The vulnerability cascades to dependent libraries. The attack requires local filesystem access and ability to create symlinks (standard user permissions on Unix; Developer Mode on Windows 10+). Exploitation succeeds within 1-3 attempts when lock file paths are predictable. The issue is fixed in version 3.20.1. If immediate upgrade is not possible, use SoftFileLock instead of UnixFileLock/WindowsFileLock (note: different locking semantics, may not be suitable for all use cases); ensure lock file directories have restrictive permissions (chmod 0700) to prevent untrusted users from creating symlinks; and/or monitor lock file directories for suspicious symlinks before running trusted applications. These workarounds provide only partial mitigation. The race condition remains exploitable. Upgrading to version 3.20.1 is strongly recommended.
CWE: CWE-59: Improper Link Resolution Before File Access ('Link Following') *(https://cwe.mitre.org/data/definitions/59.html)*
CVSS Source: NVD
CVSS Base score: 6.5
CVSS Vector: (CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:L/I:H/A:H)

CVEID: CVE-2026-21860 *(https://www.cve.org/CVERecord?id=CVE-2026-21860)*
DESCRIPTION: Werkzeug is a comprehensive WSGI web application library. Prior to version 3.1.5, Werkzeug's safe_join function allows path segments with Windows device names that have file extensions or trailing spaces. On Windows, there are special device names such as CON, AUX, etc that are implicitly present and readable in every directory. Windows still accepts them with any file extension, such as CON.txt, or trailing spaces such as CON. This issue has been patched in version 3.1.5.
CWE: CWE-67: Improper Handling of Windows Device Names *(https://cwe.mitre.org/data/definitions/67.html)*
CVSS Source: NVD
CVSS Base score: 5.3
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L)

CVEID: CVE-2020-14340 *(https://www.cve.org/CVERecord?id=CVE-2020-14340)*
DESCRIPTION: A vulnerability was discovered in XNIO where file descriptor leak caused by growing amounts of NIO Selector file handles between garbage collection cycles. It may allow the attacker to cause a denial of service. It affects XNIO versions 3.6.0.Beta1 through 3.8.1.Final.
CWE: CWE-400: Uncontrolled Resource Consumption *(https://cwe.mitre.org/data/definitions/400.html)*

CVSS Source: IBM X-Force
CVSS Base score: 5.9
CVSS Vector: (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2022-0084 *(https://www.cve.org/CVERecord?id=CVE-2022-0084)*
DESCRIPTION: A flaw was found in XNIO, specifically in the notifyReadClosed method. The issue revealed this method was logging a message to another expected end. This flaw allows an attacker to send flawed requests to a server, possibly causing log contention-related performance concerns or an unwanted disk fill-up.
CWE: CWE-770: Allocation of Resources Without Limits or Throttling *(https://cwe.mitre.org/data/definitions/770.html)*
CVSS Source: IBM X-Force
CVSS Base score: 5.9
CVSS Vector: (CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2023-5685 *(https://www.cve.org/CVERecord?id=CVE-2023-5685)*
DESCRIPTION: A flaw was found in XNIO. The XNIO NotifierState that can cause a Stack Overflow Exception when the chain of notifier states becomes problematically large can lead to uncontrolled resource management and a possible denial of service (DoS).
CWE: CWE-400: Uncontrolled Resource Consumption *(https://cwe.mitre.org/data/definitions/400.html)*
CVSS Source: IBM X-Force
CVSS Base score: 7.5
CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-12635 *(https://www.cve.org/CVERecord?id=CVE-2025-12635)*
DESCRIPTION: IBM WebSphere Application Server 8.5, 9.0 and IBM WebSphere Application Server Liberty 17.0.0.3 through 25.0.0.12 are affected by cross-site scripting due to improper validation of user-supplied input. An attacker could exploit this vulnerability by using a specially crafted URL to redirect the user to a malicious site.
CWE: CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') *(https://cwe.mitre.org/data/definitions/79.html)*
CVSS Source: IBM
CVSS Base score: 5.4
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N)

CVEID: CVE-2025-26791 *(https://www.cve.org/CVERecord?id=CVE-2025-26791)*
DESCRIPTION: DOMPurify before 3.2.4 has an incorrect template literal regular expression, sometimes leading to mutation cross-site scripting (mXSS).
CWE: CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') *(https://cwe.mitre.org/data/definitions/79.html)*
CVSS Source: NVD
CVSS Base score: 6.1
CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N)

CVEID: CVE-2024-48910 *(https://www.cve.org/CVERecord?id=CVE-2024-48910)*

DESCRIPTION: DOMPurify is a DOM-only, super-fast, uber-tolerant XSS sanitizer for HTML, MathML and SVG. DOMPurify was vulnerable to prototype pollution. This vulnerability is fixed in 2.4.2.

CWE: CWE-1321: Improperly Controlled Modification of Object Prototype Attributes ('Prototype Pollution') *(https://cwe.mitre.org/data/definitions/1321.html)*

CVSS Source: NVD

CVSS Base score: 9.8

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2024-47875 *(https://www.cve.org/CVERecord?id=CVE-2024-47875)*

DESCRIPTION: DOMPurify is a DOM-only, super-fast, uber-tolerant XSS sanitizer for HTML, MathML and SVG. DOMpurify was vulnerable to nesting-based mXSS. This vulnerability is fixed in 2.5.0 and 3.1.3.

CWE: CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') *(https://cwe.mitre.org/data/definitions/79.html)*

CVSS Source: NVD

CVSS Base score: 6.1

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N)

CVEID: CVE-2026-26007 *(https://www.cve.org/CVERecord?id=CVE-2026-26007)*

DESCRIPTION: cryptography is a package designed to expose cryptographic primitives and recipes to Python developers. Prior to 46.0.5, the public_key_from_numbers (or EllipticCurvePublicNumbers.public_key()), EllipticCurvePublicNumbers.public_key(), load_der_public_key() and load_pem_public_key() functions do not verify that the point belongs to the expected prime-order subgroup of the curve. This missing validation allows an attacker to provide a public key point P from a small-order subgroup. This can lead to security issues in various situations, such as the most commonly used signature verification (ECDSA) and shared key negotiation (ECDH). When the victim computes the shared secret as S = [victim_private_key]P via ECDH, this leaks information about victim_private_key mod (small_subgroup_order). For curves with cofactor 1, this reveals the least significant bits of the private key. When these weak public keys are used in ECDSA , it's easy to forge signatures on the small subgroup. Only SECT curves are impacted by this. This vulnerability is fixed in 46.0.5.

CWE: CWE-345: Insufficient Verification of Data Authenticity *(https://cwe.mitre.org/data/definitions/345.html)*

CVSS Source: NVD

CVSS Base score: 6.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N)

CVEID: CVE-2026-1525 *(https://www.cve.org/CVERecord?id=CVE-2026-1525)*

DESCRIPTION: Undici allows duplicate HTTP Content-Length headers when they are provided in an array with case-variant names (e.g., Content-Length and content-length). This produces malformed HTTP/1.1 requests with multiple conflicting Content-Length values on the wire. Who is impacted: * Applications using undici.request(), undici.Client, or similar low-level APIs with headers passed as flat arrays * Applications that accept user-controlled header names without case-normalization Potential consequences: * Denial of

Service: Strict HTTP parsers (proxies, servers) will reject requests with duplicate Content-Length headers (400 Bad Request) * HTTP Request Smuggling: In deployments where an intermediary and backend interpret duplicate headers inconsistently (e.g., one uses the first value, the other uses the last), this can enable request smuggling attacks leading to ACL bypass, cache poisoning, or credential hijacking

CWE: CWE-444: Inconsistent Interpretation of HTTP Requests ('HTTP Request/Response Smuggling') *(https://cwe.mitre.org/data/definitions/444.html)*

CVSS Source: NVD

CVSS Base score: 9.8

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2026-1526 *(https://www.cve.org/CVERecord?id=CVE-2026-1526)*

DESCRIPTION: The undici WebSocket client is vulnerable to a denial-of-service attack via unbounded memory consumption during permessage-deflate decompression. When a WebSocket connection negotiates the permessage-deflate extension, the client decompresses incoming compressed frames without enforcing any limit on the decompressed data size. A malicious WebSocket server can send a small compressed frame (a "decompression bomb") that expands to an extremely large size in memory, causing the Node.js process to exhaust available memory and crash or become unresponsive. The vulnerability exists in the PerMessageDeflate.decompress() method, which accumulates all decompressed chunks in memory and concatenates them into a single Buffer without checking whether the total size exceeds a safe threshold.

CWE: CWE-409: Improper Handling of Highly Compressed Data (Data Amplification) *(https://cwe.mitre.org/data/definitions/409.html)*

CVSS Source: openjs

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-1527 *(https://www.cve.org/CVERecord?id=CVE-2026-1527)*

DESCRIPTION: ImpactWhen an application passes user-controlled input to the upgrade option of client.request(), an attacker can inject CRLF sequences (\r\n) to: * Inject arbitrary HTTP headers * Terminate the HTTP request prematurely and smuggle raw data to non-HTTP services (Redis, Memcached, Elasticsearch) The vulnerability exists because undici writes the upgrade value directly to the socket without validating for invalid header characters: // lib/dispatcher/client-h1.js:1121 if (upgrade) { header += `connection: upgrade\r\nupgrade: ${upgrade}\r\n` }

CWE: CWE-93: Improper Neutralization of CRLF Sequences ('CRLF Injection') *(https://cwe.mitre.org/data/definitions/93.html)*

CVSS Source: openjs

CVSS Base score: 4.6

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:N)

CVEID: CVE-2026-1528 *(https://www.cve.org/CVERecord?id=CVE-2026-1528)*

DESCRIPTION: ImpactA server can reply with a WebSocket frame using the 64-bit length form and an extremely large length. undici's ByteParser overflows internal math, ends up in an invalid state, and throws a

fatal TypeError that terminates the process. Patches Patched in the undici version v7.24.0 and v6.24.0. Users should upgrade to this version or later.

CWE: CWE-248: Uncaught Exception *(https://cwe.mitre.org/data/definitions/248.html)*

CVSS Source: openjs

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-2229 *(https://www.cve.org/CVERecord?id=CVE-2026-2229)*

DESCRIPTION: ImpactThe undici WebSocket client is vulnerable to a denial-of-service attack due to improper validation of the server_max_window_bits parameter in the permessage-deflate extension. When a WebSocket client connects to a server, it automatically advertises support for permessage-deflate compression. A malicious server can respond with an out-of-range server_max_window_bits value (outside zlib's valid range of 8-15). When the server subsequently sends a compressed frame, the client attempts to create a zlib InflateRaw instance with the invalid windowBits value, causing a synchronous RangeError exception that is not caught, resulting in immediate process termination. The vulnerability exists because: *
The isValidClientWindowBits() function only validates that the value contains ASCII digits, not that it falls within the valid range 8-15 * The createInflateRaw() call is not wrapped in a try-catch block * The resulting exception propagates up through the call stack and crashes the Node.js process

CWE: CWE-248: Uncaught Exception *(https://cwe.mitre.org/data/definitions/248.html)*

CVSS Source: openjs

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2026-26960 *(https://www.cve.org/CVERecord?id=CVE-2026-26960)*

DESCRIPTION: node-tar is a full-featured Tar for Node.js. When using default options in versions 7.5.7 and below, an attacker-controlled archive can create a hardlink inside the extraction directory that points to a file outside the extraction root, enabling arbitrary file read and write as the extracting user. Severity is high because the primitive bypasses path protections and turns archive extraction into a direct filesystem access primitive. This issue has been fixed in version 7.5.8.

CWE: CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') *(https://cwe.mitre.org/data/definitions/22.html)*

CVSS Source: NVD

CVSS Base score: 7.1

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N)

CVEID: CVE-2025-65945 *(https://www.cve.org/CVERecord?id=CVE-2025-65945)*

DESCRIPTION: auth0/node-jws is a JSON Web Signature implementation for Node.js. In versions 3.2.2 and earlier and version 4.0.0, auth0/node-jws has an improper signature verification vulnerability when using the HS256 algorithm under specific conditions. Applications are affected when they use the jws.createVerify() function for HMAC algorithms and use user-provided data from the JSON Web Signature protected header or payload in HMAC secret lookup routines, which can allow attackers to bypass signature verification. This issue

has been patched in versions 3.2.3 and 4.0.1.

CWE: CWE-347: Improper Verification of Cryptographic Signature *(https://cwe.mitre.org/data/definitions/347.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N)

CVEID: CVE-2026-24734 *(https://www.cve.org/CVERecord?id=CVE-2026-24734)*

DESCRIPTION: Improper Input Validation vulnerability in Apache Tomcat Native, Apache Tomcat. When using an OCSP responder, Tomcat Native (and Tomcat's FFM port of the Tomcat Native code) did not complete verification or freshness checks on the OCSP response which could allow certificate revocation to be bypassed. This issue affects Apache Tomcat Native: from 1.3.0 through 1.3.4, from 2.0.0 through 2.0.11; Apache Tomcat: from 11.0.0-M1 through 11.0.17, from 10.1.0-M7 through 10.1.51, from 9.0.83 through 9.0.114. The following versions were EOL at the time the CVE was created but are known to be affected: from 1.1.23 through 1.1.34, from 1.2.0 through 1.2.39. Older EOL versions are not affected. Apache Tomcat Native users are recommended to upgrade to versions 1.3.5 or later or 2.0.12 or later, which fix the issue. Apache Tomcat users are recommended to upgrade to versions 11.0.18 or later, 10.1.52 or later or 9.0.115 or later which fix the issue.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N)

CVEID: CVE-2025-12816 *(https://www.cve.org/CVERecord?id=CVE-2025-12816)*

DESCRIPTION: An interpretation-conflict (CWE-436) vulnerability in node-forge versions 1.3.1 and earlier enables unauthenticated attackers to craft ASN.1 structures to desynchronize schema validations, yielding a semantic divergence that may bypass downstream cryptographic verifications and security decisions.

CWE: CWE-436: Interpretation Conflict *(https://cwe.mitre.org/data/definitions/436.html)*

CVSS Source: CISA ADP

CVSS Base score: 8.6

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:N)

CVEID: CVE-2026-24486 *(https://www.cve.org/CVERecord?id=CVE-2026-24486)*

DESCRIPTION: Python-Multipart is a streaming multipart parser for Python. Prior to version 0.0.22, a Path Traversal vulnerability exists when using non-default configuration options `UPLOAD_DIR` and `UPLOAD_KEEP_FILENAME=True`. An attacker can write uploaded files to arbitrary locations on the filesystem by crafting a malicious filename. Users should upgrade to version 0.0.22 to receive a patch or, as a workaround, avoid using `UPLOAD_KEEP_FILENAME=True` in project configurations.

CWE: CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') *(https://cwe.mitre.org/data/definitions/22.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N)

CVEID: CVE-2026-25128 *(https://www.cve.org/CVERecord?id=CVE-2026-25128)*

DESCRIPTION: fast-xml-parser allows users to validate XML, parse XML to JS object, or build XML from JS object without C/C++ based libraries and no callback. In versions 5.0.9 through 5.3.3, a RangeError vulnerability exists in the numeric entity processing of fast-xml-parser when parsing XML with out-of-range entity code points (e.g., `陾` or ``). This causes the parser to throw an uncaught exception, crashing any application that processes untrusted XML input. Version 5.3.4 fixes the issue.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-36335 *(https://www.cve.org/CVERecord?id=CVE-2025-36335)*

DESCRIPTION: IBM watsonx.data intelligence stores user credentials in plain text which can be read by a local user.

CWE: CWE-256: Plaintext Storage of a Password *(https://cwe.mitre.org/data/definitions/256.html)*

CVSS Source: IBM

CVSS Base score: 6.2

CVSS Vector: (CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

CVEID: CVE-2025-64756 *(https://www.cve.org/CVERecord?id=CVE-2025-64756)*

DESCRIPTION: Glob matches files using patterns the shell uses. Starting in version 10.2.0 and prior to versions 10.5.0 and 11.1.0, the glob CLI contains a command injection vulnerability in its -c/--cmd option that allows arbitrary command execution when processing files with malicious names. When glob -c command patterns are used, matched filenames are passed to a shell with shell: true, enabling shell metacharacters in filenames to trigger command injection and achieve arbitrary code execution under the user or CI account privileges. This issue has been patched in versions 10.5.0 and 11.1.0.

CWE: CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') *(https://cwe.mitre.org/data/definitions/78.html)*

CVSS Source: security-advisories@github.com

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-14924 *(https://www.cve.org/CVERecord?id=CVE-2025-14924)*

DESCRIPTION: Hugging Face Transformers megatron_gpt2 Deserialization of Untrusted Data Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the parsing of checkpoints. The issue results from the lack of proper validation of user-supplied data, which can result in deserialization of untrusted data. An attacker can leverage this vulnerability to execute code in the context of the current process. Was ZDI-CAN-27984.

CWE: CWE-502: Deserialization of Untrusted Data *(https://cwe.mitre.org/data/definitions/502.html)*

CVSS Source: zdi-disclosures@trendmicro.com

CVSS Base score: 7.8

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-14928 *(https://www.cve.org/CVERecord?id=CVE-2025-14928)*

DESCRIPTION: Hugging Face Transformers HuBERT convert_config Code Injection Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must convert a malicious checkpoint. The specific flaw exists within the convert_config function. The issue results from the lack of proper validation of a user-supplied string before using it to execute Python code. An attacker can leverage this vulnerability to execute code in the context of the current user. Was ZDI-CAN-28253.

CWE: CWE-94: Improper Control of Generation of Code ('Code Injection') *(https://cwe.mitre.org/data/definitions/94.html)*

CVSS Source: zdi-disclosures@trendmicro.com

CVSS Base score: 7.8

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-14929 *(https://www.cve.org/CVERecord?id=CVE-2025-14929)*

DESCRIPTION: Hugging Face Transformers X-CLIP Checkpoint Conversion Deserialization of Untrusted Data Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the parsing of checkpoints. The issue results from the lack of proper validation of user-supplied data, which can result in deserialization of untrusted data. An attacker can leverage this vulnerability to execute code in the context of the current process. Was ZDI-CAN-28308.

CWE: CWE-502: Deserialization of Untrusted Data *(https://cwe.mitre.org/data/definitions/502.html)*

CVSS Source: zdi-disclosures@trendmicro.com

CVSS Base score: 7.8

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-14930 *(https://www.cve.org/CVERecord?id=CVE-2025-14930)*

DESCRIPTION: Hugging Face Transformers GLM4 Deserialization of Untrusted Data Remote Code Execution Vulnerability. This vulnerability allows remote attackers to execute arbitrary code on affected installations of Hugging Face Transformers. User interaction is required to exploit this vulnerability in that the target must visit a malicious page or open a malicious file. The specific flaw exists within the parsing of weights. The issue results from the lack of proper validation of user-supplied data, which can result in deserialization of untrusted data. An attacker can leverage this vulnerability to execute code in the context of the current process. Was ZDI-CAN-28309.

CWE: CWE-502: Deserialization of Untrusted Data *(https://cwe.mitre.org/data/definitions/502.html)*

CVSS Source: zdi-disclosures@trendmicro.com

CVSS Base score: 7.8

CVSS Vector: (CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H)

CVEID: CVE-2025-7962 *(https://www.cve.org/CVERecord?id=CVE-2025-7962)*

DESCRIPTION: In Jakarta Mail 2.0.2 it is possible to preform a SMTP Injection by utilizing the \r and \n UTF-8 characters to separate different messages.

CWE: CWE-147: Improper Neutralization of Input Terminators *(https://cwe.mitre.org/data/definitions/147.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N)

CVEID: CVE-2025-14914 *(https://www.cve.org/CVERecord?id=CVE-2025-14914)*

DESCRIPTION: IBM WebSphere Application Server Liberty 17.0.0.3 through 26.0.0.1 could allow a privileged user to upload a zip archive containing path traversal sequences resulting in an overwrite of files leading to arbitrary code execution.

CWE: CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') *(https://cwe.mitre.org/data/definitions/22.html)*

CVSS Source: IBM

CVSS Base score: 7.6

CVSS Vector: (CVSS:3.1/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:H)

CVEID: CVE-2026-25990 *(https://www.cve.org/CVERecord?id=CVE-2026-25990)*

DESCRIPTION: Pillow is a Python imaging library. From 10.3.0 to before 12.1.1, n out-of-bounds write may be triggered when loading a specially crafted PSD image. This vulnerability is fixed in 12.1.1.

CWE: CWE-787: Out-of-bounds Write *(https://cwe.mitre.org/data/definitions/787.html)*

CVSS Source: NVD

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H)

CVEID: CVE-2025-12543 *(https://www.cve.org/CVERecord?id=CVE-2025-12543)*

DESCRIPTION: A flaw was found in the Undertow HTTP server core, which is used in WildFly, JBoss EAP, and other Java applications. The Undertow library fails to properly validate the Host header in incoming HTTP requests.As a result, requests containing malformed or malicious Host headers are processed without rejection, enabling attackers to poison caches, perform internal network scans, or hijack user sessions.

CWE: CWE-20: Improper Input Validation *(https://cwe.mitre.org/data/definitions/20.html)*

CVSS Source: NVD

CVSS Base score: 9.6

CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H)

IBM X-Force ID: 177835

DESCRIPTION: Apache Commons Codec could allow a remote attacker to obtain sensitive information,

caused by the improper validation of input. An attacker could exploit this vulnerability using a method call to obtain sensitive information.

CWE: CWE-200: Exposure of Sensitive Information to an Unauthorized Actor *(https://cwe.mitre.org/data/definitions/200.html)*

CVSS Source: IBM X-Force

CVSS Base score: 7.5

CVSS Vector: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

Affected Products and Versions

Affected Product(s):	Version(s):
IBM watsonx.data intelligence	5.2.0, 5.2.1, 5.3.0, 5.3.1

Remediation/Fixes

Update version to 5.3.1-patch3

https://www.ibm.com/docs/en/software-hub/5.3.x?topic=overview-available-patches-software-hub-version-531 *(https://www.ibm.com/docs/en/software-hub/5.3.x?topic=overview-available-patches-software-hub-version-531)*

Workarounds and Mitigations

None

Get Notified about Future Security Bulletins

Subscribe to My Notifications *(https://www.ibm.com/support/pages/node/718119)* to be notified of important product support alerts like this.

References

Complete CVSS v3 Guide

On-line Calculator v3

Related Information

IBM Secure Engineering Web Portal *(http://www.ibm.com/security/secure-engineering/bulletins.html)*

IBM Product Security Incident Response Blog *(http://www.ibm.com/blogs/psirt)*

Acknowledgement

Change History

27 Apr 2026: Initial Publication

*The CVSS Environment Score is customer environment specific and will ultimately impact the Overall CVSS Score. Customers can evaluate the impact of this vulnerability in their environments by accessing the links in the

Reference section of this Security Bulletin.

Disclaimer

According to the Forum of Incident Response and Security Teams (FIRST), the Common Vulnerability Scoring System (CVSS) is an "industry open standard designed to convey vulnerability severity and help to determine urgency and priority of response." IBM PROVIDES THE CVSS SCORES ""AS IS"" WITHOUT WARRANTY OF ANY KIND, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. CUSTOMERS ARE RESPONSIBLE FOR ASSESSING THE IMPACT OF ANY ACTUAL OR POTENTIAL SECURITY VULNERABILITY. In addition to other efforts to address potential vulnerabilities, IBM periodically updates the record of components contained in our product offerings. As part of that effort, if IBM identifies previously unidentified packages in a product/service inventory, we address relevant vulnerabilities regardless of CVE date. Inclusion of an older CVEID does not demonstrate that the referenced product has been used by IBM since that date, nor that IBM was aware of a vulnerability as of that date. We are making clients aware of relevant vulnerabilities as we become aware of them. "Affected Products and Versions" referenced in IBM Security Bulletins are intended to be only products and versions that are supported by IBM and have not passed their end-of-support or warranty date. Thus, failure to reference unsupported or extended-support products and versions in this Security Bulletin does not constitute a determination by IBM that they are unaffected by the vulnerability. Reference to one or more unsupported versions in this Security Bulletin shall not create an obligation for IBM to provide fixes for any unsupported or extended-support products or versions.

Document Information

More support for:
IBM watsonx.data intelligence

Software version:
5.2.0, 5.2.1, 5.3.0, 5.3.1

Operating system(s):
Red Hat, RedHat OpenShift

Document number:
7270923

Modified date:
27 April 2026

Initial Publish date:
27 April 2026