Commercial support for versions past the Maintenance LTS phase is available through our OpenJS Ecosystem Sustainability Program partners

Tuesday, March 24, 2026 Security Releases

TNJP The Node.js Project

Security releases available

Updates are now available for the 25.x, 24.x, 22.x, 20.x Node.js release lines for the following issues.

This security release includes the following dependency updates to address public vulnerabilities:

- undici (6.24.1, 7.24.4) on 22.x, 24.x, 25.x

Incomplete fix for CVE-2026-21637: `loadSNI()` in `_tls_wrap.js` lacks `try`/`catch` leading to Remote DoS (CVE-2026-21637) - (High)

A flaw in Node.js TLS error handling leaves `SNICallback` invocations unprotected against synchronous exceptions, while the equivalent ALPN and PSK callbacks were already addressed in CVE-2026-21637. This represents an incomplete fix of that prior vulnerability.

When an `SNICallback` throws synchronously on unexpected input the exception bypasses TLS error handlers and propagates as an uncaught exception, crashing the Node.js process.

- This vulnerability affects all Node.js versions that received the CVE-2026-21637 fix, including **20.x, 22.x, 24.x, and 25.x**, on any TLS server where `SNICallback` may throw on unexpected `servername` input.

Thank you, to mbarbs for reporting this vulnerability and thank you mcollina for fixing it.

Denial of Service via `__proto__` header name in `req.headersDistinct` (Uncaught `TypeError` crashes Node.js process) (CVE-2026-21710) - (High)

A flaw in Node.js HTTP request handling causes an uncaught `TypeError` when a request is received with a header named `__proto__` and the application accesses `req.headersDistinct`.

When this occurs, `dest["__proto__"]` resolves to `Object.prototype` rather than `undefined`, causing `.push()` to be called on a non-array. This exception is thrown synchronously inside a property getter and cannot be intercepted by `error` event listeners, meaning it cannot be handled without wrapping every `req.headersDistinct` access in a `try/catch`.

- This vulnerability affects all Node.js HTTP servers on **20.x, 22.x, 24.x, and v25.x**

Thank you, to yushengchen for reporting this vulnerability and thank you mcollina for fixing it.

Node.js Permission Model bypass: UDS server bind/listen works without `--allow-net` (CVE-2026-21711) - (Medium)

A flaw in Node.js Permission Model network enforcement leaves Unix Domain Socket (UDS) server operations without the required permission checks, while all comparable network paths correctly enforce them.

As a result, code running under `--permission` without `--allow-net` can create and expose local IPC endpoints, allowing communication with other processes on the same host outside of the intended network restriction boundary.

- This vulnerability affects Node.js **25.x** processes using the Permission Model where `--allow-net` is intentionally omitted to restrict network access. Note that `--allow-net` is currently an experimental feature.

Thank you, to xavlimsg for reporting this vulnerability and thank you RafaelGSS for fixing it.

Assertion error in `node_url.cc` via malformed URL format leads to Node.js crash (CVE-2026-21712) - (Medium)

A flaw in Node.js URL processing causes an assertion failure in native code when `url.format()` is called with a malformed internationalized domain name (IDN) containing invalid characters, crashing the Node.js process.

- This vulnerability affects **24.x and 25.x**.

Thank you, to wooffie for reporting this vulnerability and thank you RafaelGSS for fixing it.

Timing side-channel in HMAC verification via `memcmp()` in `crypto_hmac.cc` leads to potential MAC forgery (CVE-2026-21713) - (Medium)

A flaw in Node.js HMAC verification uses a non-constant-time comparison when validating user-provided signatures, potentially leaking timing information proportional to the number of matching bytes. Under certain threat models where high-resolution timing measurements are possible, this behavior could be exploited as a timing oracle to infer HMAC values.

Node.js already provides timing-safe comparison primitives used elsewhere in the codebase, indicating this is an oversight rather than an intentional design decision.

- This vulnerability affects **20.x, 22.x, 24.x, and 25.x**.

Thank you, to x_probe for reporting this vulnerability and thank you panva for fixing it.

Memory leak in Node.js HTTP/2 server via `WINDOW_UPDATE` on stream 0 leads to resource exhaustion (CVE-2026-21714) - (Medium)

A memory leak occurs in Node.js HTTP/2 servers when a client sends `WINDOW_UPDATE` frames on stream 0 (connection-level) that cause the flow control window to exceed the maximum value of 2^{31}-1. The server correctly sends a GOAWAY frame, but the Http2Session object is never cleaned up.

- This vulnerability affects HTTP2 users on Node.js 20, 22, 24 and 25.

Thank you, to galbarnahum for reporting this vulnerability and thank you RafaelGSS for fixing it.

HashDoS in V8 (CVE-2026-21717) - (Medium)

A flaw in V8's string hashing mechanism causes integer-like strings to be hashed to their numeric value, making hash collisions trivially predictable. By crafting a request that causes many such collisions in V8's internal string table, an attacker can significantly degrade performance of the Node.js process.

The most common trigger is any endpoint that calls `JSON.parse()` on attacker-controlled input, as JSON parsing automatically internalizes short strings into the affected hash table.

- This vulnerability affects **20.x, 22.x, 24.x, and 25.x**.

Thank you, to sharp_edged for reporting this vulnerability and thank you joyeecheung for fixing it.

Permission Model Bypass in realpathSync.native Allows File Existence Disclosure (CVE-2026-21715) - (Low)

A flaw in Node.js Permission Model filesystem enforcement leaves `fs.realpathSync.native()` without the required read permission checks, while all comparable filesystem functions correctly enforce them.

As a result, code running under `--permission` with restricted `--allow-fs-read` can still use `fs.realpathSync.native()` to check file existence, resolve symlink targets, and enumerate filesystem paths outside of permitted directories.

- This vulnerability affects **20.x, 22.x, 24.x, and 25.x** processes using the Permission Model where `--allow-fs-read` is intentionally restricted.

Thank you, to stif for reporting this vulnerability and thank you RafaelGSS for fixing it.

CVE-2024-36137 Patch Bypass - FileHandle.chmod/chown (CVE-2026-21716) - (Low)

An incomplete fix for CVE-2024-36137 leaves `FileHandle.chmod()` and `FileHandle.chown()` in the promises API without the required permission checks, while their callback-based equivalents (`fs.fchmod()`, `fs.fchown()`) were correctly patched.

As a result, code running under `--permission` with restricted `--allow-fs-write` can still use promise-based `FileHandle` methods to modify file permissions and ownership on already-open file descriptors, bypassing the intended write restrictions.

- This vulnerability affects **20.x, 22.x, 24.x, and 25.x** processes using the Permission Model where `--allow-fs-write` is intentionally restricted.

Thank you, to wooseokdotkim for reporting this vulnerability and thank you RafaelGSS for fixing it.

Downloads and release details

- Node.js v20.20.2
- Node.js v22.22.2
- Node.js v24.14.1
- Node.js v25.8.2

Summary

The Node.js project will release new versions of the 25.x, 24.x, 22.x, 20.x releases lines on or shortly after, Tuesday, March 24, 2026 in order to address:

- 2 high severity issues.
- 5 medium severity issues.
- 2 low severity issues.

Impact

The 25.x release line of Node.js is vulnerable to 2 high severity issues, 5 medium severity issues, 2 low severity issues. The 24.x release line of Node.js is vulnerable to 2 high severity issues, 4 medium severity issues, 2 low severity issues. The 22.x release line of Node.js is vulnerable to 2 high severity issues, 4 medium severity issues, 2 low severity issues. The 20.x release line of Node.js is vulnerable to 2 high severity issues, 4 medium severity issues, 2 low severity issues.

It's important to note that End-of-Life versions are always affected when a security release occurs. To ensure your system's security, please use an up-to-date version as outlined in our

Release Schedule.

Release timing

Releases will be available on, or shortly after, Tuesday, March 24, 2026.

Contact and future updates

The current Node.js security policy can be found at https://nodejs.org/en/security/. Please follow the process outlined in https://github.com/nodejs/node/blob/master/SECURITY.md if you wish to report a vulnerability in Node.js.

Subscribe to the low-volume announcement-only nodejs-sec mailing list at https://groups.google.com/forum/#!forum/nodejs-sec to stay up to date on security vulnerabilities and security-related releases of Node.js and the projects maintained in the nodejs GitHub organization.

Previous
Developing a minimally HashDoS resistant, yet quickly reversible integer hash for V8

Next
OpenSSL Security Advisory Assessment, January 2026

Last Updated

Mar 24, 2026

Reading Time

5 min

Contribute

Edit this page

Table of Contents