

What is Node2Shell?

Node2Shell is the name for CVE-2026-111497, an out-of-bounds write in the Google Brotli decoder (versions 1.0.3 to 1.2.0) bundled with Node.js. A malformed Brotli stream makes the decoder keep writing past a small internal buffer, which can permit remote code execution. Node.js 22.x, 24.x and 26.x are affected and fixed in 22.23.4, 24.22.1 and 26.12.1 (October 21, 2026). Apps are exposed when they decompress attacker-controlled Brotli data, for example Express 5 request bodies or `fetch()` responses.

HOW IT WORKS

01 Who is exposed and how the attack works

A bare Node.js HTTP server does not decode Brotli bodies on its own. Exposure depends on whether attacker-controlled Brotli data reaches the decoder. Shown for defensive context, the paths are:

- Inbound request bodies. Express 5 (5.0.0 to 5.3.0) and body-parser 2.x (2.0.0 to 2.3.0) inflate bodies sent with `Content-Encoding: br` when using `express.json()` or `raw()`. A single crafted POST with that header and a malformed Brotli body reaches the bug.
- Outbound responses. The built-in `fetch()` decodes Brotli responses. A server-side app that fetches URLs an attacker can influence, such as webhooks, link previews or SSRF-prone features, can be fed a malicious response.
- Direct decompression. Any code calling `zlib.brotliDecompress()` or `createBrotliDecompress()` on uploaded files or message payloads.

The researchers report that code execution took multiple attempts in their lab, and as of October 23, 2026 no exploitation in the wild had been confirmed and the CVE was not on the CISA KEV list. Crashing the process is far easier, so even a failed exploit is a denial-of-service.

02 How to check if you are exposed

Check the runtime that is actually deployed, not the version in your laptop's shell:

HOW TO DEFEND

- Reject `Content-Encoding: br` on inbound requests at the reverse proxy, load balancer or middleware for paths that do not need compressed uploads. Few public APIs do.
- Limit what server-side `fetch` can reach, and strip `br` from the `Accept-Encoding` header on outbound requests to untrusted hosts.
- Move off end-of-life lines (20.x and older, 23.x, 25.x), which will not receive this or future fixes.

SOURCES

- [1] Node.js October 2026 security releases
- [2] NVD: CVE-2026-111497
- [3] Google Brotli releases
- [4] MITRE CWE-787: Out-of-bounds Write

- Node version: `node -v` inside every container image, serverless runtime and host. Vulnerable on 22.x below 22.23.4, 24.x below 24.22.1 and 26.x below 26.12.1. End-of-life lines 20.x, 23.x and 25.x get no fix.
- Bundled Brotli: `node -p process.versions.brotli` shows the decoder version compiled into the runtime.
- Other runtimes: Bun is fixed in 1.4.4. Electron apps and single-executable Node binaries carry their own runtime and need their own rebuild.
- Middleware: search your code for `express.json-`, `body-parser` and `brotliDecompress`, and check whether any service accepts `Content-Encoding: br` from the internet.
- Signals: unexpected process exits or decoder errors during decompression. They are not proof of exploitation, but correlate them with the runtime version and the source of the compressed data.

Find the path to a shell before an attacker does.

securelayer7.net/learn/web-exploitation/what-is-node2shell

[Open online](https://securelayer7.net/learn/web-exploitation/what-is-node2shell)