

What is the Brotli RCE?

The Brotli RCE is CVE-2026-111497, an out-of-bounds write in the streaming decoder of Google's Brotli compression library, versions 1.0.3 to 1.2.0. A malformed block switch was treated as a need for more input instead of a format error, so the decoder kept writing into a small fixed buffer, which can permit remote code execution. Any software that decodes attacker-controlled Brotli data with that code is exposed, including Node.js (Node2Shell) and Python HTTP stacks (HTTP2Shell). Fixed in Brotli 1.0.10, 1.1.1 and 1.2.1, published as a security fix in October 2026.

HOW IT WORKS

01 Which software bundles the vulnerable decoder

The exposure follows the C library, not the language. Places it is known to be bundled:

- Node.js 22.x, 24.x and 26.x, through `node:zlib`, `fetch()`, `DecompressionStream` and middleware such as Express 5 and `body-parser` 2.x. Fixed in 22.23.4, 24.22.1 and 26.12.1. See Node2Shell.
- Bun, fixed in 1.4.4.
- Python: the Brotli package up to 1.2.0 and `brotlicffi` up to 1.2.0.2, which requests, `urllib3`, `httpx` and `aiohttp` use automatically when installed. Fixed in Brotli 1.2.1 and `brotlicffi` 1.2.1.0. See HTTP2Shell.
- Operating system packages (`libbrotli`, `brotli`) used by native software such as web servers, proxies and image or font tooling. Fixes arrive through your distribution's security updates.
- Anything statically linking the library, including Electron apps, single-executable binaries and vendor appliances, which only a rebuild or vendor update fixes.

Decoders that are independent reimplementations in another language, rather than wrappers around the C code, do not contain this exact code. Check each project's own advisories rather than assuming either way.

02 How an attacker reaches it

The bug only fires when attacker-controlled Brotli data is decoded. Shown for defensive context, the realistic delivery paths are:

HOW TO DEFEND

- Inventory every copy. `node -p process.versions.brotli` for Node.js, `pip show brotli brotlicffi` for Python, your package manager for `libbrotli`, and an image scan for statically linked copies. Dependency scanners may miss the Python fixes, which had no advisory database entry as of October 23, 2026.
- Upgrade Brotli to 1.0.10, 1.1.1 or 1.2.1 or later on your branch, or the runtime or package that bundles it, then rebuild container images.
- Reduce exposure while you patch. Reject inbound `Content-Encoding: br` where uploads do not need it, strip `br` from outbound `Accept-Encoding` to untrusted hosts, or remove the optional Brotli package so clients fall back to `gzip`.
- Watch for crashes in decompression code paths, correlated with the source of the compressed data, as an early signal of probing.

SOURCES

- [1] NVD: CVE-2026-111497
- [2] Google Brotli releases
- [3] Node.js October 2026 security releases
- [4] PyPI: `brotlicffi`

- Inbound request bodies with Content-Encoding: br sent to a server whose middleware inflates compressed bodies.
- Responses to outbound requests, when a server-side client advertises br in Accept-Encoding and fetches a URL the attacker controls, such as a webhook, link preview or feed reader.
- Files and messages decompressed by application code, such as uploaded archives or queue payloads.

As of October 23, 2026, no exploitation in the wild had been reported. Public proof-of-concept work showed code execution in a lab after multiple attempts and on deliberately weakened builds, and a crash on stock builds. A crash is still a reliable denial-of-service against an unpatched service.

Find the path to a shell before an attacker does.

securelayer7.net/learn/web-exploitation/what-is-brotli-rce

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