

HiveForce Labs

THREAT ADVISORY



VULNERABILITY REPORT

Node2Shell: Node.js Brotli Flaw Turns Compressed Data into Code Execution

Date of Publication

October 23, 2026

Admiralty Code

A1

TA Number

TA2026314

Summary

First Seen: October 21, 2026

Affected Products: Node.js, Bun

Impact: CVE-2026-111497, dubbed Node2Shell, is a publicly disclosed out-of-bounds write in the Brotli decoder bundled with Node.js. An unauthenticated remote attacker can trigger it with a crafted Brotli-compressed stream to corrupt memory, leak process data, crash the application, or execute arbitrary code. Exposure arises wherever untrusted Brotli data is decompressed, including Express 5 and body-parser 2.x request parsing and responses consumed through fetch(). Proof-of-concept exploits are publicly available, and code execution has been separately demonstrated against an official Node.js build, which makes immediate patching critical.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-111497	Node2Shell (Node.js Brotli Decoder Out-of-Bounds Write Vulnerability)	Node.js, Bun			

Vulnerability Details

#1

CVE-2026-111497 is an out-of-bounds write vulnerability (CWE-787) in the copy of Google's Brotli decompression library that Node.js bundles into its runtime. The flaw is triggered by malformed compressed input that drives the decoder to a block-type transition the stream's own structure makes impossible. Instead of rejecting this condition, the vulnerable decoder interprets it as a request for more input and continues processing with an invalid internal state, which can cause it to write beyond an intended memory buffer. The corrected decoder returns a format error for the invalid transition and passes it to the caller, halting decompression before memory is corrupted. This fix had been committed to the upstream Brotli source in February 2026, months before it reached the Node.js runtime.

#2

The attack surface covers any path in which Node.js decompresses attacker-controlled Brotli data. Express 5 and body-parser 2.x inflate incoming request bodies carrying the Content-Encoding: br header when their raw() or json() parsers are configured, letting any client deliver a payload, while the built-in fetch() API automatically decompresses Brotli responses, exposing services that call attacker-controlled or compromised servers through webhooks or user-supplied URLs. Untrusted input passed directly to node:zlib Brotli functions is equally exposed, whereas a bare Node.js HTTP server does not decode incoming Brotli bodies by itself.

#3

The vulnerability affects Node.js 22.x before 22.23.4, 24.x before 24.22.1, and 26.x before 26.12.1, along with End-of-Life lines such as 25.x, 23.x, and 20.x that will receive no fix. The Bun runtime was also affected before version 1.4.4, and Node.js builds from Linux distribution packages that rely on the system Brotli library depend on the distribution's own Brotli update.

#4

The Node.js project rates the issue High severity and characterizes its impact as remote code execution. Exploit research progressed rapidly after disclosure, with code execution reproduced against the official, unmodified Node.js 24.22.0 Linux arm64 build through both consumed Brotli responses and configured request parsers, although success required multiple attempts. The flaw is publicly disclosed, and proof-of-concept exploits appeared on GitHub from October 22 but reach code execution only on deliberately weakened builds, while no in-the-wild exploitation has been reported.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-111497	Node.js (22.x before 22.23.4, 24.x before 24.22.1, 26.x before 26.12.1, and End-of-Life versions), Bun (before 1.4.4), Google Brotli (1.0.3 through 1.2.0)	cpe:2.3:a:nodejs:nod e.js:*:*:*:*:*:* cpe:2.3:a:google:brotl i:*:*:*:*:*:*	CWE-787 CWE-252

Recommendations



Upgrade Node.js Runtimes Immediately: Update all Node.js deployments to 22.23.4, 24.22.1, or 26.12.1, or to a newer patch release within the same line. Prioritize services that accept Brotli-compressed request bodies through Express 5 or body-parser 2.x and those that fetch content from external or user-supplied URLs, as these offer the most direct routes to the vulnerable decoder. The same releases also fix other high-severity issues, including TLS hostname verification bypasses, and remove trust in compromised eMudhra certificates, so upgrading reduces risk beyond this CVE.



Rebuild and Redeploy Container Images: Verify the runtime version inside every container image, CI runner, and host using `node --version`, since a package inventory can miss a Node.js binary bundled within an image. Images pinned to a vulnerable tag such as `node:24.22.0` remain exposed no matter how often they are rebuilt, so move the pin to a fixed release and redeploy.



Block Untrusted Brotli Request Bodies: The vendor has published no workaround, so where an immediate upgrade is not feasible, reject inbound requests carrying Content-Encoding: br at the reverse proxy, load balancer, WAF, or application middleware. This reduces inbound exposure only and does not protect against Brotli responses that applications fetch from external servers, so treat it strictly as a bridge to the runtime upgrade.



Map Brotli Decompression Paths: Identify every point where untrusted Brotli data enters applications by auditing dependencies for Express 5 and body-parser 2.x, searching code and dependencies for brotliDecompress and createBrotliDecompress calls from node:zlib, and cataloguing outbound fetch() calls to servers outside organizational control, including webhook deliveries and user-supplied URLs.



Update Bundled and System Brotli Components: Upgrade Bun to version 1.4.4 or later, and for Node.js installed from Linux distribution packages that use the system Brotli library, apply the distribution's Brotli security update in addition to checking the runtime version. Review other software that bundles Brotli, as the same fix shipped in Brotli 1.2.1, 1.1.1, and 1.0.10.



Monitor for Decompression Anomalies: Alert on unexpected Node.js process exits and decoder errors during decompression, and correlate them with the deployed runtime version, application logs, and the origin of the compressed data. These signals help prioritize investigation but are not reliable standalone indicators of exploitation.



Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	<u>T1190</u> : Exploit Public-Facing Application	
Execution*	<u>T1059</u> : Command and Scripting Interpreter	
Collection*	<u>T1005</u> : Data from Local System	
Impact	<u>T1499</u> : Endpoint Denial of Service	<u>T1499.004</u> : Application or System Exploitation

*Inferred



Patch Links

<https://nodejs.org/en/blog/release/v22.23.4/>

<https://nodejs.org/en/blog/release/v24.22.1/>

<https://nodejs.org/en/blog/release/v26.12.1/>

<https://bun.com/blog/bun-v1.4.4>



References

<https://www.imperva.com/blog/node2shell-from-brotli-data-to-remote-code-execution/>

<https://www.aikido.dev/blog/node2shell-nodejs-brotli-rce-cve-2026-111497>

<https://nodejs.org/en/blog/vulnerability/october-2026-security-releases#nodejs-brotli-decoder-out-of-bounds-write-permits-remote-code-execution-cve-2026-111497---high>

<https://github.com/oven-sh/bun/pull/46392>



What Next?

At Hive Pro, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a Demo of HivePro.

REPORT GENERATED ON

October 23, 2026 • 06:35 PM

© 2026 All Rights are Reserved by Hive Pro



More at www.hivepro.com