

> BACKGROUND REPORT FOR PROFESSIONALS

Brotli flaw (Node2Shell/HTTP2Shell): out-of-band fixes in .NET, Node.js, PHP and Python packages

The compressor you didn't know you were using

Event: 21 October 2026 (out-of-band fixes: 21-26 October)

Closed: 27 October 2026 (research revision: v1)

Made with the AI crew, reviewed by Péter

English version of the Hungarian original

Summary

Google's Brotli compression library has an out-of-bounds write in its decoder (CVE-2026-111497, “Node2Shell/HTTP2Shell”): a specially crafted compressed stream can cause a crash, a memory leak and, in some cases, code execution. Because many products bundle their own copy of the library, within a week out-of-band fixes shipped for Node.js, .NET, PHP's Windows packages, Python packages, Bun, Envoy and Istio. No in-the-wild exploitation was known as of 23 October 2026, but a public PoC and a researcher demonstration exist. The most important action: an inventory of where Brotli lives in your environment, then patching, rebuilding containers and restarting, because updating the system library doesn't fix bundled copies.

KEY FINDINGS

- The flaw is in the Brotli C decoder (1.0.3–1.2.0), fixed in releases 1.0.10, 1.1.1 and 1.2.1 (21 October 2026).
- Updating the system Brotli doesn't fix applications that bundle their own copy (Node.js, some .NET builds, PHP Windows packages): these must be updated and restarted separately.
- Bad data can arrive from two directions: inbound (compressed request body) and outbound (a response downloaded from a foreign server) – the WAF doesn't see the latter, and changing Accept-Encoding doesn't protect either.
- Severity is disputed (CVSS 8.1–9.2): code execution has been demonstrated only with difficulty and unreliably; the more likely impact is denial of service.
- No Hungarian impact or domestic authority warning was known at the time of research.

What happened

There is an out-of-bounds write in the decoder (the decompression part) of Google's open-source **Brotli** compression library: **CVE-2026-111497**. A specially crafted Brotli stream can make the decoder write past one of its internal buffers during decompression. Possible consequences are a crash, a memory leak and, under certain conditions, remote code execution. The press and researchers use several names: **Node2Shell** (because it first surfaced in Node.js), **HTTP2Shell** (because it turned out to affect every ecosystem that speaks HTTP) and, jokingly, **Broccoli2Shell**. ([SecurityWeek](#); [Red Hat](#); [Aikido](#)) – **confirmed**

Brotli is the compression browsers and servers use on the web alongside or instead of gzip (Content-Encoding: br). Many products bundle their own copy, so this is not a flaw in one product but in a **built-in component**: wherever software decompresses Brotli data from an untrusted source, it may be reachable.

Affected and fixed Brotli versions: 1.0.3–1.2.0 affected; fixed in **1.0.10, 1.1.1 and 1.2.1** (21 October 2026). ([SecurityWeek](#); [Wiz](#)) – **confirmed**

Discovery: Node.js credits members of OpenAI's Codex Security team with the discovery; according to reports, OpenAI found the bug during red teaming and notified the Brotli developers and downstream projects. The change fixing the root cause had already landed in the Brotli source code in February 2026, but at that time it was not known to be a security bug and no release followed. ([SecurityWeek](#); [Broccoli2Shell tracker](#); [The Hacker News](#)) – **reported** (several sources; OpenAI's own announcement was not reviewed)

Exploitation: as of 23 October 2026, no in-the-wild attacks had been reported, and the flaw was not in CISA's Known Exploited Vulnerabilities (KEV) catalog. A public proof of concept (PoC) and a researcher demonstration exist (see below). ([Sysdig](#); [The Hacker News](#); [SecurityWeek](#)) – **confirmed** (status on 23 October)

Timeline

Date	Event	Source
February 2026	The change fixing the root cause lands in the Brotli source, without a security label or release (reported)	SecurityWeek , The Hacker News
14 October 2026	Node.js pre-announces the security release (affected lines, high severity)	Sysdig
21 October 2026	Node.js fixes: 22.23.4, 24.22.1, 26.12.1 (10 CVEs in the release in total); Brotli 1.0.10 / 1.1.1 / 1.2.1 released	The Hacker News , Sysdig
21 October 2026	A few hours after the fix, the first public repository for the CVE appears	Sysdig
22 October 2026	The CVE and GitHub advisory go public; Python Brotli 1.2.1 on PyPI; Imperva demonstrates code execution on a Node.js build	Broccoli2Shell tracker , The Hacker News
23 October 2026	brtlicffi 1.2.1.0; it becomes clear the impact goes beyond Node.js (Bun, Envoy, Istio fixes); no KEV entry	Aikido , Sysdig
25 October 2026 (Sunday)	Microsoft ships out-of-band fixes: .NET 8/9/10, .NET 11 RC3, Visual Studio, PowerShell	BleepingComputer , SecurityWeek
around 26 October 2026	PHP rebuilds its Windows packages for the second time in five days ("-1" builds with Brotli 1.2.1)	The Hacker News

Actors and background

- **Google / Brotli project:** owner of the library, released the fixed versions. Google scored it **9.2 (critical)** under CVSS 4.0 in the CVE record. ([The Hacker News](#); [Sysdig](#))
- **OpenAI Codex Security:** per Node.js, they reported the flaw. Interesting background: an AI-driven security team found a bug in a widely used C library. ([SecurityWeek](#); [Broccoli2Shell tracker](#))
- **Node.js project:** first to ship a fix (21 October), severity: high. ([The Hacker News](#))
- **Microsoft:** CVSS 3.1 score 8.1, classified as elevation of privilege, rated "Exploitation More Likely"; critical for .NET and Visual Studio, important for PowerShell. ([SecurityWeek](#))
- **PHP project:** rebuilt its Windows packages because of the bundled Brotli, just a few days after the PHP2Shell fixes. ([The Hacker News](#))
- **Python ecosystem:** the Brotli and brotlicffi packages were fixed; popular HTTP clients (urllib3, requests, httpx, aiohttp) use them automatically when installed. ([Aikido](#); [Broccoli2Shell tracker](#))
- **Researchers and security firms:** Imperva (exploitability research), Aikido and OpenSourceMalware (Python impact), Wiz, Sysdig (detection).

Context of the week: this is another big flaw of the "everything is a Shell" week, after PHP2Shell (two critical PHP flaws, fixed on 21 October). The shared lesson: flaws in rarely mentioned components deep in web infrastructure hit many ecosystems at once. ([The Hacker News](#))

Similar earlier cases (background): the problem of bundled libraries ("vendored dependencies") is an old one: with zlib, libwebp or xz-utils the difficulty was also that after a flaw in a single low-level component, every product carrying its own copy had to update separately. – **assumption** (editorial parallel, not from this research's sources)

How it works

RED // ATTACKER'S VIEW

- **The basic idea:** the point of compression is that the decompressor produces large data from small data, so the decompressor *trusts* what the compressed data describes. If that description is wrong and the decoder doesn't notice, it can write to the wrong place in memory.
- **The weakness, conceptually:** according to public analyses, one part of the decoder signalled "bad data" and "need more data" in the same way, and a fast path ignored the result. A deliberately corrupted stream could therefore make it keep accepting data into a small internal buffer and overrun it. The fix simply returns a hard format error. ([Broccoli2Shell tracker](#); [Red Hat](#))
- **Where bad data can come from:** two directions.
- **Inbound:** someone sends the server a compressed request body (Content-Encoding: br), and the web framework decompresses it automatically.
- **Outbound:** your application fetches something from a foreign server (webhook, link preview, crawler, API call), and the HTTP client automatically decompresses the Brotli response. This is the trickier direction, because the firewall or WAF doesn't see it. ([Aikido](#); [Broccoli2Shell tracker](#))
- **Why it can be hidden:** the word "Brotli" doesn't have to appear anywhere in the application code. It is enough that an HTTP client or middleware uses automatic decompression, or that the package is installed as a transitive dependency. ([Wiz](#); [Aikido](#))
- **Why it isn't trivial:** per Red Hat, attack complexity is high, and modern protections (ASLR, NX, FORTIFY_SOURCE, SELinux) make code execution harder but don't prevent a crash. The most likely

impact is therefore denial of service (DoS); code execution is possible but unreliable. ([Red Hat](#); [The Hacker News](#))

- **The "system update isn't enough" trap:** if an application bundles its own Brotli copy (Node.js, some .NET builds, PHP's Windows packages, tools like Git for Windows), updating the system libbrotli doesn't fix it. And conversely: the running process uses the old code until you restart it. ([BleepingComputer](#); [SecurityWeek](#))

Impact and Hungarian relevance

- **Scale:** Brotli is one of the web's core compression formats; the affected set is any software that decompresses untrusted Brotli data: Node.js, Bun, .NET, PHP Windows packages, Python HTTP clients, Envoy and Istio (service meshes, API gateways), a Go library, Java brotli4j, plus desktop tools. ([Wiz](#); [Aikido](#); [SecurityWeek](#))
- **Who is most affected:** organisations and developers running web services, APIs, containerised or serverless applications; especially where the service fetches user-supplied URLs. Individuals less directly: for them the usual browser and app updates are the task.
- **Damage:** no known damage or attack reported as of 23 October 2026. ([Sysdig](#))
- **Hungarian relevance: no known impact in Hungary.** At the time of research we found no warning about this flaw among the alerts of NBSZ NKI (Hungary's national cyber security authority), and no substantive coverage in the Hungarian press. Many Hungarian web services run on .NET or Node.js, which makes it likely that domestic organisations may also be affected, but there is no concrete data.
– **assumption**
- **Region:** we found no Central and Eastern Europe-specific data.

What to do

BLUE // WHAT TO DO NOW

Individuals

- **Update** your browser, phone and desktop apps when updates are offered; vendors fix their own bundled Brotli copies this way.
- **If you use developer tools** (Node.js, Python, Git, PHP on your own machine), update those to the latest version too.
- **Don't panic:** there are no reports of in-the-wild attacks, and the flaw mainly affects servers and developer tools. There is no verified data on browser exposure (see Open questions), so the usual automatic updates are the task.

Small business

- **Ask** your hosting or development partner: does your website or web shop run on Node.js, .NET, PHP (Windows) or Python, and has it been updated since 21 October?
- **If you run it yourself:** update Node.js (22.23.4 / 24.22.1 / 26.12.1), .NET (8.0.33 / 9.0.22 / 10.0.14), Python Brotli (1.2.1) and brotlicffi (1.2.1.0), then **restart** the services.
- **If you run PHP on Windows:** switch to the build ending in "-1" (e.g. 8.5.12-1) and verify the SHA-256 hash on the php.net download page; `php -v` shows the same version, so you can't tell from that.
- **If your Node.js is 20.x, 23.x or 25.x:** these are end-of-life and get no fix; plan the move to 22, 24 or 26.

Organisations

- **Build an inventory** of where Brotli lives in your environment: runtimes, packages, lockfiles, container images, CI, build and developer machines, bundled (vendored) copies. Don't rely only on the vulnerability scanner: according to reports, some tools initially didn't reliably flag the affected PyPI versions. ([Broccoli2Shell tracker](#); [Aikido](#))
- **Prioritise by exposure:** first update services that fetch foreign URLs or accept Brotli-compressed request bodies. ([Broccoli2Shell tracker](#); [Wiz](#))
- **Update, rebuild, restart:** after the package update, rebuild container images on a fixed base image, update lockfiles, restart processes, and verify the running binary is actually the fixed one. ([SecurityWeek](#); [SOC Prime](#))
- **Temporary mitigation if you can't update right away:** reject Content-Encoding: br request bodies at the proxy, load balancer or WAF; turn off automatic Brotli decompression; in Python remove the Brotli/brotlicffi package (clients fall back to gzip). Removing br from Accept-Encoding on its own **does not protect**, because the server can still send a Brotli response. ([Broccoli2Shell tracker](#); [The Hacker News](#); [BleepingComputer](#))
- **On Linux, check which Brotli is in use:** per Red Hat, some Node.js and .NET builds use the system Brotli, others the bundled one; this determines whether you update brotli/libbrotli or the runtime package. ([Red Hat](#))
- **Watch for signs:** repeated crashes of web-facing Node.js or .NET processes, a service spawning a shell, unexpected outbound connections. ([Sysdig](#))

Technical details

CVE: CVE-2026-111497 – out-of-bounds write in the block-type switching part of the Brotli C decoder.

Scores: - Google: CVSS 4.0 – 9.2 (critical) - Red Hat: CVSS 3.1 – 8.1 (important), AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H - Microsoft: CVSS 3.1 – 8.1, elevation of privilege, "Exploitation More Likely" - Node.js: "High" ([The Hacker News](#); [Red Hat](#); [SecurityWeek](#))

Affected → fixed versions:

Component	Affected	Fixed	Source
Google Brotli (C)	1.0.3–1.2.0	1.0.10, 1.1.1, 1.2.1	Wiz , SecurityWeek
Node.js	22.x, 24.x, 26.x before the fix; 20/23/25 EOL, no fix	22.23.4, 24.22.1, 26.12.1	The Hacker News , Sysdig
.NET	8, 9, 10, 11 preview	8.0.33, 9.0.22, 10.0.14, 11 RC3	BleepingComputer , SecurityWeek
Visual Studio, PowerShell	VS 2019/2022/2026; PowerShell 7.4/7.5/7.6	25 October 2026 updates	SecurityWeek
PHP Windows packages	8.2–8.5 (Brotli 1.2.0 in earlier builds)	8.5.12-1, 8.4.27-1, 8.3.36-1, 8.2.35-1	The Hacker News
Python Brotli	1.0.3–1.2.0	1.2.1	Aikido , Wiz
Python brotlicffi	1.0.3–1.2.0.x	1.2.1.0	Aikido , Wiz
Bun	1.0.21–1.4.3	1.4.4	Wiz , Sysdig
Envoy	1.36.11, 1.37.7, 1.38.5, 1.39.2	1.36.12, 1.37.8, 1.38.6, 1.39.3, 1.40.0	Wiz

Component	Affected	Fixed	Source
Istio	1.29.0–1.29.8, 1.30.0–1.30.5, 1.31.0–1.31.1	1.29.9, 1.30.6, 1.31.2	Wiz
Go andybalholm/brotli	< v1.2.7	v1.2.7	Wiz

Reachable paths (conceptually): - Node.js: direct use of `node:zlib`; automatic decompression in the built-in `fetch()` (`undici`); Express 5 default body parsing; Koa, Fastify with a compression plugin. ([Sysdig](#); [The Hacker News](#)) - .NET: ASP.NET Core request decompression, `HttpClient` automatic response decompression, `System.IO.Compression.Brotli` APIs; none is enabled by default. ([BleepingComputer](#)) - PHP (Windows): the `cURL` extension when automatic decompression is enabled (`CURLOPT_ENCODING`; empty value = all supported formats). Off by default, but HTTP client libraries may turn it on. ([The Hacker News](#)) - Python: `urllib3/requests`, `httpx` (with the `brotli` extra), `aiohttp`, `Scrapy` – if the package is installed; it doesn't need to be imported. ([Aikido](#)) - `curl`: only if built with Brotli and decompression is enabled; the server's `Content-Encoding: br` header alone triggers decompression. ([Lilting](#); [The Hacker News](#))

Still unpatched per the public tracker (26 October 2026): DuckDB, Apache Arrow, 7-Zip-zstd, p7zip, EDK II and others; the list is incomplete. ([SecurityWeek](#); [Broccoli2Shell tracker](#))

MITRE ATT&CK (editorial mapping): T1190 (Exploit Public-Facing Application), T1499 (Endpoint Denial of Service), T1059 (Command and Scripting Interpreter – after successful code execution), T1105 (Ingress Tool Transfer), T1552 (Unsecured Credentials from the runtime). – **assumption**

Detection ideas: - a runtime sensor doesn't see the overwrite itself; watch the behaviour *after* successful exploitation: process launch after a crash in the same workload, a shell under the web process, unexpected outbound connections; ([Sysdig](#)) - repeated crashes of the same service (may indicate attempts); - `Content-Encoding: br` requests to endpoints where this is unusual (an exposure signal, not a fix); - inventory: running Node.js version and bundled Brotli version, loaded `libbrotlidec`, `pip/lockfile` dependency trees. ([Broccoli2Shell tracker](#))

Indicators: no official IOC list is known; the Broccoli2Shell tracker explicitly publishes no IOCs or attack signatures.

Official guidance: Microsoft .NET security announcement (`dotnet/announcements #494`), Red Hat RHSB-2026-015, the Node.js October 2026 security release, the PHP Windows download page (SHA-256 hashes).

Contradictions and open questions

Open questions

- **How severe is it really?** Scores range from 8.1 to 9.2, and researchers are divided: Imperva demonstrated code execution on an official Node.js build (not on every attempt), while the public PoC only achieved code execution on a deliberately weakened build, causing a crash and limited memory read on a default build. Some researchers think the flaw is in practice only good for denial of service. ([The Hacker News](#); [SecurityWeek](#))
- **PoC descriptions differ:** BleepingComputer says "PoC exploits appeared on GitHub", SecurityWeek and The Hacker News clarify: the public PoC reached code execution only on a weakened build. ([BleepingComputer](#); [SecurityWeek](#))

- **Why such a late alert?** The root-cause fix was in the source in February, but without a security label, so those using bundled copies didn't know about it. We found no detailed explanation from the project of how this happened.
- **Browsers and desktop apps:** public material doesn't discuss in detail whether the major browsers and Electron-based apps are affected.
- **Linux distributions:** fixes are arriving gradually; backported version numbers may look older than upstream, so the distribution's own advisory is authoritative.
- **Hungary:** no domestic authority warning or domestic exposure data was known at the time of research.

Source assessment

Source	Type	Reliability	Supports
Red Hat – CVE-2026-111497	vendor (advisory)	high	root-cause location, CVSS 3.1, system vs. bundled Brotli, effect of protections
SecurityWeek	press	high	ecosystems, Microsoft fixes, discovery, PoC debate, tracker
BleepingComputer	press	high	.NET versions, reachable paths, mitigations
The Hacker News – PHP Windows	press	medium–high	PHP "-1" builds, cURL path, scores, CISA status
The Hacker News – Node.js release	press	medium–high	Node.js versions, other CVEs in the release, timeline
The Hacker News – Imperva demo	press	medium–high	exploitability, PoC status
Wiz	researcher (security firm)	high	version table (Bun, Envoy, Istio, Go), actions
Sysdig	researcher (security firm)	high	detailed timeline, detection, EOL lines
Aikido	researcher (security firm)	medium–high	Python impact, HTTP clients, naming
Broccoli2Shell tracker	background (independent aggregator)	medium	root cause in plain terms, affected/unpatched list, mitigations
SOC Prime	researcher (security firm)	medium	mitigations, incident response steps
Lilting	background (individual test blog)	medium	curl behaviour with automatic decompression