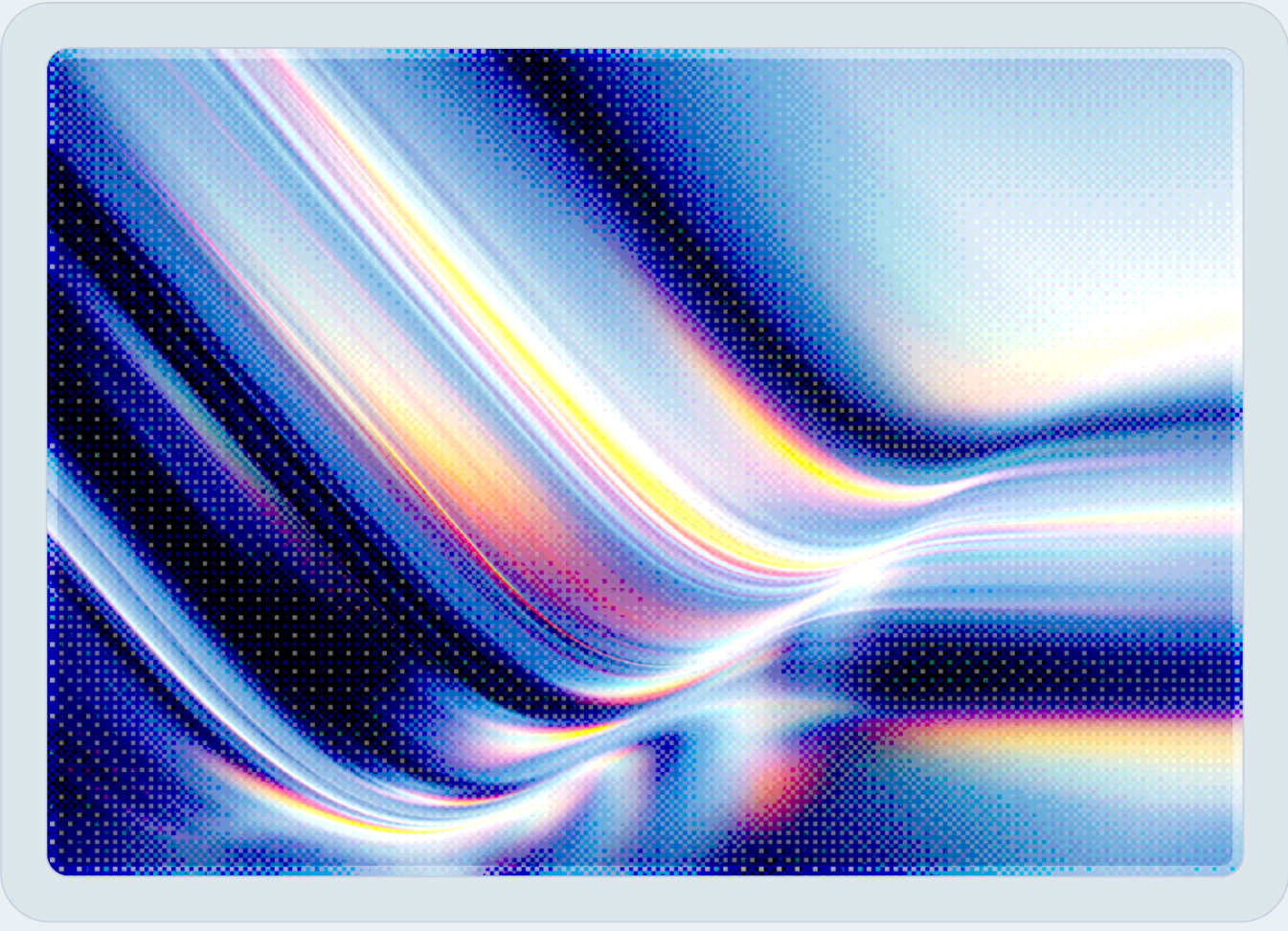
CSAI Foundation | Cloud Security Alliance

CVE-2026-119618: AI-Discovered Apache HTTP/2 RCE and the Distribution Patch Gap

Why the upstream fix is only the first step for mod_http2 operators

2026-11-12

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

CVE-2026-119618 is a critical race condition in the `assign_headers()` function of Apache HTTP Server's mod_http2 that can corrupt heap memory and lead to remote code execution. It affects httpd 2.4.55 through 2.4.70 and standalone mod_h2 2.0.8 through 2.0.42, and httpd 2.4.71 fixes it [1]. Two parties found the flaw independently: Anthropic's AI-based OSS Scanner reported it on October 26, 2026, and Google Cloud Vulnerability Research reported it eight days later. Apache shipped the fix 16 days after the first report [1]. Red Hat scores the issue CVSS 9.8, describes it as exploitable by an unauthenticated remote attacker, and states that public research has demonstrated a reliable exploit against a default-configured server with mod_http2 enabled [2]. CSA was unable to locate the primary exploit research, however, and no in-the-wild exploitation had been reported as of this writing.

Fix availability one day after the upstream release was uneven. Debian had fixed trixie and sid, while bookworm and forky remained listed as vulnerable, and RHEL and Ubuntu had not yet published fixes [2] [3][4]. How long these gaps persist will determine real-world exposure, and operators should treat the downstream packaging timeline as a risk to monitor closely. Only servers that negotiate HTTP/2 (`h2` or `h2c` in the `Protocols` directive) are exposed [2]. For hosts that cannot yet take a fixed package, removing HTTP/2 is a fast interim mitigation that is low-risk for typical web workloads, provided operators first verify that no services depend on HTTP/2, such as gRPC back ends.

Background

Apache HTTP Server 2.4.71 was released on November 11, 2026, and fixes three vulnerabilities. The most severe is CVE-2026-119618, which the Apache security team rates critical and describes as "Concurrent Execution using Shared Resource with Improper Synchronization" in mod_http2's `assign_headers()` function that "can lead to RCE" [1]. Red Hat explains the mechanism more directly: a thread-safety flaw lets one thread write into memory owned by another, which corrupts server memory. Red Hat scores it CVSS 3.1 9.8 with a network attack vector, low complexity, and no privileges or user interaction required [2]. At the time of writing, Red Hat's score breakdown showed no NVD or CVE.org score for the issue [2]. Organizations that prioritize by NVD scores may therefore underweight this flaw until those records are enriched.

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.

The affected range starts at httpd 2.4.55, released in January 2023, which brought in the mod_http2 2.0.x code line. It ends at 2.4.70, so the vulnerable code shipped in every httpd release from 2.4.55 through 2.4.70, a span of nearly four years [1]. Red Hat states that mod_http2 versions before 2.0.8 do not contain the affected code. As a result, Red Hat Enterprise Linux 8 and earlier are not affected, while RHEL 9 and RHEL 10 are [2]. Red Hat rates RHEL 9 and 10 as Important rather than Critical because their packages allocate each HTTP/2 stream from a separate allocator, which in Red Hat's assessment "substantially increases the complexity of exploiting the flaw" [2]. The difference shows that two packages built from the same upstream code can carry different risk, because the hardening and build choices a distribution makes can raise or lower that risk.

The discovery credit is also notable, because it bears on how long defenders can expect a flaw like this to stay secret. Apache credits two independent finders. Anthropic's OSS Scanner reported the issue on October 26, 2026, under identifier ANT-2026-MAPKT6MZ, and Anthony Weems and Simon Scannell of Google Cloud Vulnerability Research reported it separately on November 3 [1]. Anthropic launched OSS Scanner in October 2026 as a free, opt-in, AI-based service that runs its most capable models, including Claude Mythos, against enrolled open-source projects. The scanner was built on lessons from the company's earlier Project Glasswing [5]. According to reporting on the launch, the scanner runs "fully automated, with no human review before a report goes out," and each report includes reproduction tooling, an explanation, and a suggested fix where one is available [5].

This is not the first AI-assisted mod_http2 finding of 2026. In June, CVE-2026-49975 (the "HTTP/2 Bomb" denial-of-service issue) was credited to a Calif.IO researcher "in collaboration with OpenAI Codex," and its fix shipped in httpd 2.4.68 [1][7]. CVE-2026-119618 is more serious, however: it is memory corruption with demonstrated code-execution potential, not resource exhaustion.

Security Analysis

Exposure and Exploitability

Exposure depends on configuration more than on whether the module is installed. A server is reachable through this flaw only when mod_http2 is loaded and `h2` (HTTP/2 over TLS) or `h2c` (cleartext HTTP/2) appears in an active `Protocols` directive [2]. Some distribution packages ship mod_http2 without enabling HTTP/2 by default, while performance-tuning guides and hosting control panels may turn it on. Red Hat's phrase "default-configured server with mod_http2 enabled" appears to refer to an otherwise default configuration in which HTTP/2 has been turned on. Internet-facing TLS sites, reverse proxies, and load-balancer back ends that speak HTTP/2 are the main concern. Package-level inventories alone will therefore misstate both the true exposure and the true safety.

How easy the flaw is to exploit is still being assessed. Red Hat's statement says "public research has demonstrated a reliable exploit" against a default-configured server with mod_http2 enabled [2]. One secondary analysis reports that the demonstrated exploitation targets the event MPM and that reliability depends on threading and allocator behavior [6]. We did not locate the primary exploit write-up, and we found no reports of in-the-wild exploitation as of this writing. Even so, an unauthenticated network RCE in a widely deployed web server, combined with a public claim of reliable exploitation, warrants emergency-change treatment rather than routine patch cadence. Red Hat's lower rating for RHEL 9 and 10 reflects a harder exploit, not an impossible one, and operators should not read "Important" as permission to defer.

Dual Discovery Compresses the Safe Window

The two independent reports eight days apart are significant in their own right. Coordinated disclosure assumes that the reporter and the maintainer hold a secret for a while, and that the risk during that time is low because few others know of the flaw. When an automated AI scanner and a dedicated vendor vulnerability research team find the same bug in the same function within about a week, that assumption weakens. The finding suggests that, for high-value code paths in critical infrastructure software, other capable actors may find similar flaws on similar timelines, and some of those actors will not disclose what they find. CSA's earlier analysis of Project Glasswing reported that only 97 of 1,596 AI-discovered vulnerabilities disclosed to maintainers had been fixed [8]. Against that backdrop, Apache's 16-day turnaround from first report to release is short compared with typical coordinated disclosure windows of up to 90 days. The window that remains for defenders is the time between the upstream release and the actual deployment of a fix on each server.

The Distribution Patch Gap

Most production httpd instances likely run vendor packages rather than upstream tarballs, drawn from Debian, Ubuntu, RHEL and its rebuilds, SUSE, Alpine, container base images, or appliance firmware. Each of these channels has its own schedule for backporting and testing fixes. The following table records status on November 12, the day after the upstream release, and shows how uneven fix availability can be even within a single distribution family.

Channel	Status as of 2026-11-12	Source
Apache httpd upstream	Fixed in 2.4.71 (released 2026-11-11)	[1]
Debian 13 "trixie"	Fixed in 2.4.71-1~deb13u1 via DSA-6620-1	[3][4]

Channel	Status as of 2026-11-12	Source
Debian unstable "sid"	Fixed in 2.4.71-1	[3]
Debian 12 "bookworm"	Tracker lists vulnerable; not included in DSA-6620-1	[3][4]
Debian testing "forky"	Tracker lists 2.4.69-1 as vulnerable	[3]
RHEL 9 / RHEL 10	Affected, rated Important; no errata listed on CVE page at time of review	[2]
RHEL 8 and earlier	Not affected (pre-2.0.8 mod_http2)	[2]
Ubuntu	No CVE page or USN located at time of writing	—

Several points follow from this table. First, Debian's tracker lists a bookworm-security package at 2.4.71-1~deb12u1 that it still marks vulnerable. The tracker gives no reason for this, and the bookworm fix is absent from the DSA [3][4]. Operators on Debian 12 should confirm the status directly rather than assume that a version string containing "2.4.71" means the fix is in place. Second, enterprise distributions usually backport fixes into older version numbers, so a RHEL 9 package that fixes this CVE will probably not report itself as 2.4.71. Scanners that judge exposure by comparing banners or version strings against the upstream affected range will produce false positives on patched backports [6][11], and they may also miss vulnerable custom builds. Third, the standalone mod_h2 module distributed through GitHub has its own release line. CSA could not confirm that a fixed standalone mod_h2 release had shipped at the time of writing; one analysis lists it as pending [6]. Operators who build mod_h2 from source should check the upstream repository directly.

The pattern is not new, but AI-accelerated discovery raises its cost. In one earlier 2026 case, Ubuntu published USN-8384-1 for CVE-2026-49975 on June 4, four days before upstream httpd 2.4.68 shipped on June 8, because the fix was already available in upstream mod_h2 code since late May [9][1]. Distribution timelines depend heavily on when a fix becomes available to packagers and on how many supported releases need backports. For CVE-2026-119618, the visible one-day gap reflects the ordinary cost of testing and backporting across many supported releases at once. The concern is what happens if that gap stretches: when an AI scanner can produce a critical memory-corruption finding in a mature module, and a second team can find it independently within days, slow distribution channels could become the main exposure for many enterprises. A fix that sits in an upstream tarball protects few servers until it reaches the packages those servers actually install.

Recurring Memory Safety in mod_http2

CVE-2026-119618 is the latest in a series of 2026 mod_http2 flaws, including a double free with possible RCE (CVE-2026-23918, fixed in 2.4.67) and the HTTP/2 Bomb denial of service (CVE-2026-49975, fixed in 2.4.68) [1]. HTTP/2's multiplexed, stateful design requires concurrent stream handling in C. That makes the module a natural target for both AI-driven and human analysis, and it suggests more findings are likely. For operators, the practical point is that HTTP/2 support in httpd should be treated as attack surface that is enabled deliberately where it is needed, rather than as a default performance setting.

Recommendations

Immediate Actions

Organizations should first identify every httpd instance where HTTP/2 is actually negotiated, not merely where mod_http2 is installed. Running `apachectl -M | grep http2` together with a review of all `Protocols` directives, or an external ALPN probe for `h2`, gives a more accurate picture than package inventories. Where a fixed vendor package is available, as with Debian trixie, apply it as an emergency change and restart httpd. Where no fixed package exists, remove `h2` and `h2c` from `Protocols` (leaving `Protocols http/1.1`) or unload mod_http2 until a fix arrives [1][2]. Before doing so, verify that no services depend on HTTP/2, such as gRPC back ends proxied through httpd, h2c upgrade paths, or HTTP/2-only internal clients. For typical web workloads, clients will fall back to HTTP/1.1 with a modest performance impact. Front-end CDNs and load balancers that terminate HTTP/2 themselves can often keep HTTP/2 toward users while speaking HTTP/1.1 to the Apache origin.

Short-Term Mitigations

Over the next two to four weeks, security teams should track each distribution's advisory feed for this CVE directly. Waiting for scanner signatures is slower, because fix status is changing release by release. Container images need separate attention: base images built on `httpd`, Debian, or Ubuntu layers must be rebuilt and redeployed after vendor packages update, and images pinned to older digests will remain vulnerable no matter what the host OS does. Teams should validate remediation by package changelog or vendor errata ID rather than by upstream version number, given the backporting behavior described above [11]. Where web application firewalls or reverse proxies sit in front of Apache, operators should confirm whether they forward HTTP/2 to the origin, which preserves exposure, or downgrade it.

Strategic Considerations

The broader lesson concerns time. AI-assisted discovery appears to be increasing the rate at which long-standing flaws in mature code are found [12], and this incident shows that independent rediscovery within days is possible for critical code paths. Organizations should set remediation targets for critical, unauthenticated network RCE that are measured in days, and those targets should cover the full path from upstream release to vendor package to deployed workload. Organizations should also reduce their dependence on any single distribution's backport schedule for internet-facing services. That can mean keeping the ability to build and deploy upstream security releases quickly, or putting HTTP/2 termination on hardened, separately patched edge infrastructure. Finally, organizations that rely on open-source infrastructure should consider supporting both upstream maintainers and distribution security teams. The CSA Project Glasswing analysis identified remediation capacity, not discovery, as the binding constraint [8]. In this case upstream remediation was fast, and the constraint shifted downstream to distribution packaging and enterprise deployment.

CSA Resource Alignment

CSA's research note *Project Glasswing: AI Discovery Outpaces Open Source Patching Capacity* [8] is the most direct precedent for this incident. That analysis documented how Anthropic's AI-driven discovery effort produced findings far faster than open-source maintainers could fix them, and it recommended tiered actions to manage the gap between discovery and remediation. CVE-2026-119618 is a concrete case from the successor OSS Scanner program. Here the upstream maintainers responded quickly, and the remaining gap moved downstream to distribution packaging and enterprise deployment. Readers can use the Glasswing note's recommendations on compressed exploitation timelines as the strategic frame for the operational guidance above.

CSA's *AI-Accelerated Vulnerability Discovery and the Patch Debt Crisis* [12] examines the same dynamic at the enterprise level. It argues that AI-driven discovery is outpacing organizations' capacity to remediate, and that defenders need to shift from chasing full patch coverage toward risk-based prioritization, compensating controls, and detection. The interim HTTP/2 mitigation and the exposure-based inventory guidance in this note are examples of that approach applied to a single vulnerability.

CSA's June 2026 research note *HTTP/2 Bomb: AI-Discovered DoS Hits Every Major Web Server* [7] covered the earlier AI-assisted mod_http2 finding, CVE-2026-49975, and recommended disabling HTTP/2 where patches were unavailable. The two notes together show that AI tools from more than one vendor are now producing serious findings in the same protocol implementation. They also show that the same interim mitigation, restricting `Protocols` to HTTP/1.1, applies to both.

Within the AI Controls Matrix v1.1 [10], the Threat and Vulnerability Management (TVM) domain covers the remediation timelines, vulnerability prioritization, and patch-verification practices this note recommends. The domain's emphasis on prioritizing by actual exploitability and exposure fits the guidance to inventory negotiated HTTP/2 rather than installed packages. The Application and Interface Security (AIS) domain is relevant to organizations that operate their own AI-assisted code scanning, which incidents like this one may make an increasingly common practice among maintainers of critical software.

References

[1] Apache Software Foundation. "Apache HTTP Server 2.4 vulnerabilities." Apache HTTP Server Project, November 2026.

[2] Red Hat. "CVE-2026-119618." Red Hat Customer Portal, November 2026.

[3] Debian. "CVE-2026-119618." Debian Security Tracker, accessed November 12, 2026.

[4] Debian. "DSA-6620-1 apache2 security update." debian-security-announce, November 12, 2026.

[5] Paganini, Pierluigi. "Claude Helps Secure Open Source as Anthropic Offers Free Vulnerability Scanning." Security Affairs, October 9, 2026.

[6] Threat Landscape. "CVE-2026-119618 Apache mod_http2 RCE Analysis." Threat Landscape Blog, November 12, 2026.

[7] Cloud Security Alliance. "HTTP/2 Bomb: AI-Discovered DoS Hits Every Major Web Server." CSA Labs, June 4, 2026.

[8] Cloud Security Alliance. "Project Glasswing: AI Discovery Outpaces Open Source Patching Capacity." Cloud Security Alliance, June 2026.

[9] Canonical. "USN-8384-1: Apache HTTP Server vulnerability." ubuntu-security-announce, June 4, 2026.

[10] Cloud Security Alliance. "AI Controls Matrix v1.1." Cloud Security Alliance, 2026.

[11] Red Hat. "Security Backporting Practice." Red Hat Customer Portal, accessed November 12, 2026.

[12] Cloud Security Alliance. "AI-Accelerated Vulnerability Discovery and the Patch Debt Crisis." CSA Labs, June 17, 2026.

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.