June 4, 2026

AI-Discovered HTTP/2 Bomb Affects Major Web Servers

Key Insights:

- A remote denial-of-service (DoS) vulnerability, dubbed the “HTTP/2 Bomb,” affects major web servers including NGINX, Apache HTTP Server, Microsoft IIS, Envoy and Cloudflare Pingora.
- The vulnerability was discovered by the security firm Calif using OpenAI Codex, which uniquely chained two known techniques: a variation of the HPACK compression bomb and an HTTP/2 low-and-slow hold.
- An attacker needs only a single system and a 100Mbps internet connection to render vulnerable servers inaccessible within seconds.
- Public Python proof-of-concept (PoC) attack scripts were published to GitHub on June 2, 2026. NGINX (fixed in release 1.29.8) and Apache (assigned CVE-2026-49975, fixed in mod_http2 v2.0.41) have deployed patches. However, as of June 2, 2026, the fix status for Microsoft IIS, Envoy, and Cloudflare Pingora remained unknown.
- To maintain service availability, organizations should immediately patch exposed servers or isolate them behind reverse proxies, web application and API protection (WAAP) services, or Layer 7 load balancers. Radware products are validated as not affected and have provided protection against potential attacks since day one.
- This incident underscores a paradigm shift in cybersecurity, proving that frontier AI models can identify and chain decade-old public flaws at a speed and scale that human researchers cannot match.

Cybersecurity researchers have discovered a remote denial-of-service (DoS) exploit that affects major web servers, including NGINX, Apache HTTP Server, Microsoft Internet Information Services (IIS), Envoy and Cloudflare Pingora. The attack was discovered by the research team from Calif, a security firm from California. The team leveraged OpenAI Codex, a coding agent powered by a frontier AI model. Codex discovered the new vulnerability by chaining two known techniques: a compression bomb and a low-and-slow hold. The researchers dubbed the new vulnerability “HTTP/2 Bomb.”

Two Known Vulnerabilities

HPACK Bomb

PACK (RFC 7541) is a stateful compression mechanism designed to strip away the redundancy in HTTP/2 traffic. Instead of repeatedly transmitting bulky header strings, it utilizes a shared memory system between client and server. Instead of sending the same long header text over

and over, both sides of an HTTP/2 connection keep a matching list of the headers they are repeating. When a sender wants to use a header, it puts it on the list just once. For any future messages, the sender does not transmit the whole header again. Instead, it just sends the index number—usually only a single byte—referring to the header text. When the receiver gets this index number, it looks it up on its own list and then swaps the number back out for the original full header to rebuild the complete message.

Cory Benfield coined the term "HPACK Bomb" back in 2016 in CVE-2016-6581. In 2025, Gal Bar Nahum discovered a similar memory exhaustion vulnerability in the Apache HTTP/2 module, which was assigned CVE-2025-53020 and fixed by the Apache maintainers in July 2025.

HTTP/2 Low & Slow Attack

HTTP/2 (RFC 9113) has a built-in way to stop data from overloading the connection on any single stream. The side receiving the data controls the limit for the data the server can send by setting a limit on how much it can handle at one time. The sender cannot send any more data past that limit until the receiver sends an update message giving it permission to continue.

HTTP/2 low-and-slow exhaustion without the compression amplifier goes back just as far: CVE-2016-8740 for unbounded CONTINUATION frames and CVE-2016-1546 for worker-thread starvation, both in Apache httpd.

HTTP/2 Bomb Chained Vulnerability

By chaining the low-and-slow attack vector with a variation of the known HTTP Bomb vulnerability Codex uncovered a new, previously undisclosed vulnerability that can render web servers unresponsive within seconds and with very limited resources required by an attacker.

While the HPACK Bomb filled a server's index table with large values and referenced them repeatedly to achieve amplification, recent server implementations mitigated this vulnerability by limiting the length of the total decoded header size. In the new HTTP/2 Bomb vulnerability, headers are kept small or empty ("") and the amplification comes from the per-entry metadata overhead a server adds to it. The decoded size limit detection mitigating the HPACK Bomb will not detect this new memory exhaustion attack because it covers only the resulting string after reassembling all headers and not the individual header index table entries.

Some servers, such as Apache and Envoy, limit the header field count instead of the total header size. For these servers, the researchers used the cookie header as a bypass. RFC 9113 §8.2.3 says that in HTTP/2, one does not have to keep the cookie header fields into a single giant string. Instead, a server is allowed to chop a big cookie into smaller pieces, nicknamed crumbs. Instead of sending one massive cookie header, a browser can send multiple separate cookie headers with each one holding only a single crumb. This feature allows the HPACK compression to work

much more efficiently. If a browser sends one giant cookie string, and only one tiny part of it changes on subsequent requests, the server treats the whole cookie string as a brand new one and will not compress it efficiently. By splitting the cookie into crumbs, the server can save each tiny piece of the cookie into its memory index list. If only just one crumb changes later on, the browser only has to send the changed piece instead of the whole cookie. For all the other crumbs, it can just send the cheap, one-byte index numbers. Apache httpd and Envoy, however, are not counting cookie crumbs against the header field count limit, which allows the researchers to bypass the mitigation put in place to protect against the original HPACK Bomb attack.

Researchers noted that to make the most out of the attack vector, an attacker should not let the server process run out of memory. This would lead to the worker process to be killed and respawned clean. It is more effective to keep the memory pressure just under the kill threshold and push the system into a thrashing state where the system spends significantly more time moving memory pages between physical RAM and disk storage (swap space) than it does executing actual application instructions.

Impact

According to the researchers, a single system and a 100Mbps internet connection are enough to render a vulnerable server inaccessible within seconds. They demonstrated the attack against Apache httpd and Envoy with a single client being able to consume and hold 32GB of server memory in roughly 20 seconds. NGINX and Microsoft IIS resisted the attack for about 45 seconds, after which they were inaccessible and not responding to new legitimate requests.

Server	Amplification	Impact
Envoy 1.37.2	~5,700:1	~32 GB in ~10s
Apache httpd 2.4.67	~4,000:1	~32 GB in ~18s
nginx 1.29.7	~70:1	~32 GB in ~45s
Microsoft IIS (Windows Server 2025)	~68:1	~64 GB in ~45s

Table 1: Proof of concept demo test results by the Calif researchers (source: Calif)

Public Proof of Concept

The researchers published Python attack scripts that demonstrate the vulnerability for each affected server in a companion repository linked in their blog post. Before disclosing the vulnerabilities and publishing the PoC code, they responsibly disclosed the vulnerability to:

- The NGINX team, back in April, who immediately fixed the vulnerability in release 1.29.8

- The Apache team, on May 27, who assigned CVE-2026-49975 and fixed the vulnerability in mod_http2 v2.0.41
- Microsoft (IIS), Envoy and Cloudflare (Pingora), back in May 2026, but fix status was unknown by June 2, 2026

The companion repository was published on GitHub on June 2, 2026.

Mitigation

Because a working example of the attack is out in the open and an attacker needs only limited resources to conduct a successful attack, this vulnerability should not be ignored. It should be considered a threat to the availability of services that are directly exposed to the internet. The threat can be mitigated either by patching directly exposed servers or hosting services, applications and APIs behind reverse proxies or gateways that terminate public-facing sessions, such as web application and API protection services or L7 load balancers. Also, ensure adequate access controls are in place to avoid direct access to the origin services from the internet.

Radware Products Not Affected

At the time of disclosure, preliminary analysis of the latest versions of Radware products and services concluded that these were unaffected by the vulnerability. Radware products and services provide real-time protection against potential exploits and zero-days.

Updates and detailed product information are available through the knowledge base article "CVE-2026-49975 Codex Discovery HTTP/2 Bomb" on the Radware customer portal.

Conclusion

It is worth noting that this exploit was discovered by a frontier AI model from OpenAI. The individual flaws it chained together have been public for nearly ten years, but no human researcher ever thought to combine them until now.

This breakthrough comes just a few weeks after another frontier AI model, named Mythos, discovered a 27-year-old vulnerability in OpenBSD. Both cases prove how good frontier models have become at hunting down vulnerabilities. While human experts could have eventually discovered these vulnerabilities, we (humans) are simply no match for the speed and scale at which these frontier models operate.

EFFECTIVE DDOS PROTECTION ESSENTIALS

Hybrid DDoS Protection – Use on-premises and **cloud DDoS protection** for real-time **DDoS attack prevention** that also addresses high-volume attacks and protects from pipe saturation

Behavioral-Based Detection – Quickly and accurately identify and block anomalies while allowing legitimate traffic through

Real-Time Signature Creation – Promptly protect against unknown threats and zero-day attacks

Web DDOS Tsunami Protection – Automated immediate detection and mitigation of Web DDOS encrypted high RPS and morphing attacks

A Cybersecurity Emergency Response Plan – Turn to a dedicated emergency team of experts who have experience with Internet of Things security and handling IoT outbreaks

Intelligence on Active Threat Actors – High fidelity, correlated and analyzed data for preemptive protection against currently active known attackers

For further **network and application protection** measures, Radware urges companies to inspect and patch their network to defend against risks and threats.

EFFECTIVE WEB APPLICATION SECURITY ESSENTIALS

Full OWASP Top-10 coverage against defacements, injections, etc.

Low false positive rate using negative and positive security models for maximum accuracy

Auto-policy generation capabilities for the widest coverage with the lowest operational effort

Bot protection and device fingerprinting capabilities to overcome dynamic IP attacks and achieve improved bot detection and blocking

Securing APIs by filtering paths, understanding XML and JSON schemas for enforcement, and using activity tracking mechanisms to trace bots and guard internal resources

Flexible deployment options including on-premises, out-of-path, virtual or cloud-based

LEARN MORE AT RADWARE'S SECURITY RESEARCH CENTER

To know more about today's attack vector landscape, understand the business impact of cyberattacks, or learn more about emerging attack types and tools, visit Radware's **Security Research Center**. Additionally, visit Radware's **Quarterly DDoS & Application Threat Analysis Center** for quarter-over-quarter analysis of DDoS and application attack activity based on data from Radware's cloud security services and threat intelligence.