HTTP/2 implementations are vulnerable to "MadeYouReset" DoS attack through HTTP/2 control frames

Vulnerability Note VU#767506

Original Release Date: 2025-08-13 | Last Revised: 2026-03-17

Overview

A vulnerability has been discovered within many HTTP/2 implementations allowing for denial of service (DoS) attacks through HTTP/2 control frames. This vulnerability is colloquially known as "MadeYouReset" and is tracked as CVE-2025-8671. Some vendors have assigned a specific CVE to their products to describe the vulnerability, such as CVE-2025-48989, which is used to identify Apache Tomcat products affected by the vulnerability. MadeYouReset exploits a mismatch caused by stream resets between HTTP/2 specifications and the internal architectures of many real-world web servers. This results in resource exhaustion, and a threat actor can leverage this vulnerability to perform a distributed denial of service attack (DDoS). This vulnerability is similar to CVE-2023-44487,

ABOUT VULNERABILITY NOTES

CONTACT US ABOUT THIS VULNERABILITY

PROVIDE A VENDOR STATEMENT

colloquially known as "Rapid Reset." Multiple vendors have issued patches or responses to the vulnerability, and readers should review the statements provided by vendors at the end of this Vulnerability Note and patch as appropriate.

Description

A mismatch caused by client-triggered server-sent stream resets between HTTP/2 specifications and the internal architectures of some HTTP/2 implementations may result in excessive server resource consumption leading to denial-of-service (DoS). This vulnerability is tracked as CVE-2025-8671 and is known colloquially as "MadeYouReset." This vulnerability is similar to CVE-2023-44487, colloquially known as "Rapid Reset", which abused client-sent stream resets. HTTP/2 introduced stream cancellation - the ability of both client and server to immediately close a stream at any time. However, after a stream is canceled, many implementations keep processing the request, compute the response, but don't send it back to the client. This creates a mismatch between the amount of active streams from the HTTP/2 point of view, and the actual active HTTP requests the backend server is processing.

By opening streams and then rapidly triggering the server to reset them using malformed frames or flow control errors, an attacker can exploit a discrepancy created between HTTP/2 streams accounting and the servers active HTTP requests. Streams reset by the server are considered closed, even though backend processing continues. This allows a client to cause the server to handle an unbounded number of concurrent HTTP/2 requests on a single connection.

The flaw largely stems from many implementations of the HTTP/2 protocol equating resetting streams to closing them; however, in practice, the server will still process them. An attacker can exploit this to continually send reset requests, where the protocol is considering these reset streams as closed, but the server will still be processing them, causing a DoS.

HTTP/2 does support a parameter called SETTINGS_MAX_CONCURRENT_STREAMS, which defines a set of currently active streams per session. In theory, this setting would prevent an attacker from overloading the target server, as they would

max out the concurrent stream counter for their specific malicious session. In practice, when a stream is reset by the attacker, the protocol considers it no longer active and no longer accounts for it within this counter.

Impact

The main impact of this vulnerability is its potential usage in DDoS attacks. Threat actors exploiting the vulnerability will likely be able to force targets offline or heavily limit connection possibilities for clients by making the server process an extremely high number of concurrent requests. Victims will have to address either high CPU overload or memory exhaustion depending on their implementation of HTTP/2.

Solution

Various vendors have provided patches and statements to address the vulnerability. Please review their statements below. CERT/CC recommends that vendors who use HTTP/2 in their products review their implementation and limit the number/rate of RST_STREAMs sent from the server. Additionally, please review the supplemental materials provided by the reporters, which include additional mitigations and other potential solutions here: https://galbarnahum.com/made-you-reset

Acknowledgements

Thanks to the reporters, Gal Bar Nahum, Anat Bremler-Barr, and Yaniv Harel of Tel Aviv University. This document was written by Christopher Cullen.

Vendor Information

Filter by status: All

Filter by content:

Additional information available

Sort by: Status

Expand all

AMPHP	Affected
Apache Tomcat	Affected
Eclipse Foundation	Affected
Fastly	Affected
gRPC	Affected
Mozilla	Affected
Netty	Affected
Red Hat	Affected
SUSE Linux	Affected
Varnish Software	Affected

View all 119 vendors

References

- https://github.com/galbarnahum/MadeYouReset
- https://galbarnahum.com/made-you-reset
- https://deepness-lab.org/publications/madeyoureset/
- https://www.imperva.com/blog/madeyoureset-turning-http-2-server-against-itself/
- https://www.cve.org/CVERecord?id=CVE-2025-8671
- https://www.rfc-editor.org/rfc/rfc9113.html#name-rst_stream
- https://www.rfc-editor.org/rfc/rfc9113.html#section-6.5.2
- https://github.com/tempesta-tech/tempesta/issues/2439
- https://github.com/tempesta-tech/tempesta/issues/2451

- https://seanmonstar.com/blog/hyper-http2-didnt-madeyoureset/
- https://blog.litespeedtech.com/2025/08/13/litespeed-not-affected-by-madeyoureset/
- https://blog.cloudflare.com/madeyoureset-an-http-2-vulnerability-thwarted-by-rapid-reset-mitigations/
- https://www.akamai.com/blog/security/response-madeyoureset-http2-protocol-attacks
- https://www.windriver.com/security/vulnerability-responses/http2-madeyoureset-vulnerability
- https://thehackernews.com/2025/08/new-http2-madeyoureset-vulnerability.html
- https://tempesta-tech.com/blog/made-you-reset-http2-ddos-attack-analysis-and-mitigation/
- https://www.isc2.org/Insights/2023/10/The-HTTP2-Fast-Reset-Attack-Vulnerability-What-You-Need-To-Know
- https://gitlab.isc.org/isc-projects/bind9/-/issues/5325

Other Information

CVE IDs:	CVE-2025-36047 CVE-2025-48989 CVE-2025-5115 CVE-2025-54500 CVE-2025-55163 CVE-2025-8671 CVE-2025-9784
API URL:	VINCE JSON | CSAF
Date Public:	2025-08-13
Date First Published:	2025-08-13
Date Last Updated:	2026-03-17 16:01 UTC
Document Revision:	29

Download PGP Key

Read CERT/CC Blog

Learn about Vulnerability Analysis

Carnegie Mellon University
Software Engineering
Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
412-268-5800

Contact SEI

Contact CERT/CC

412-268-5800
cert@cert.org