ECLIPSE®
FOUNDATION

Security Training for ●●●●●● Developers

Part 2: Section 3: Vulnerability Response Coordination and Embargoes

About This Training

- For Developers
- Designed in three parts
  - **Foundations and Systems Security** (for all Contributors)
  - **Vulnerability Management** (for all Contributors)
  - **Advanced Subjects** (for Committers and future Committers)

- Each part has a number of sections
  - Typically 15 minutes of study
  - Choose what matches your needs
- Content
  - Videos (live or pre-recorded)
  - Readings
  - Exercises

Complex Vulnerability Cases

- Multi-project (copied code, similar issues in similar projects)
  - Critical or exploited issues
- Communication issues

Multi-project issues and embargoes

Multi-project issues

- The same code has been copied between projects, and an issue has been discovered in that shared code
- Similar issue in projects with the same function (architectural issue or common implementation flaw). Example: in HTTP/2
- Projects with heavy inter-dependencies. Example: a fix requires changes in multiple projects

What is an embargo? (1)

- The period of time when a vulnerability is privately discussed, fixed, and then disclosed
- “Must to know” people only
  - Each additional person adds risk
  - Managers AREN’T “must to know” in vulnerability management
- Goal: deliver the fix to users before bad actors use the vulnerability

What is an embargo? (2)

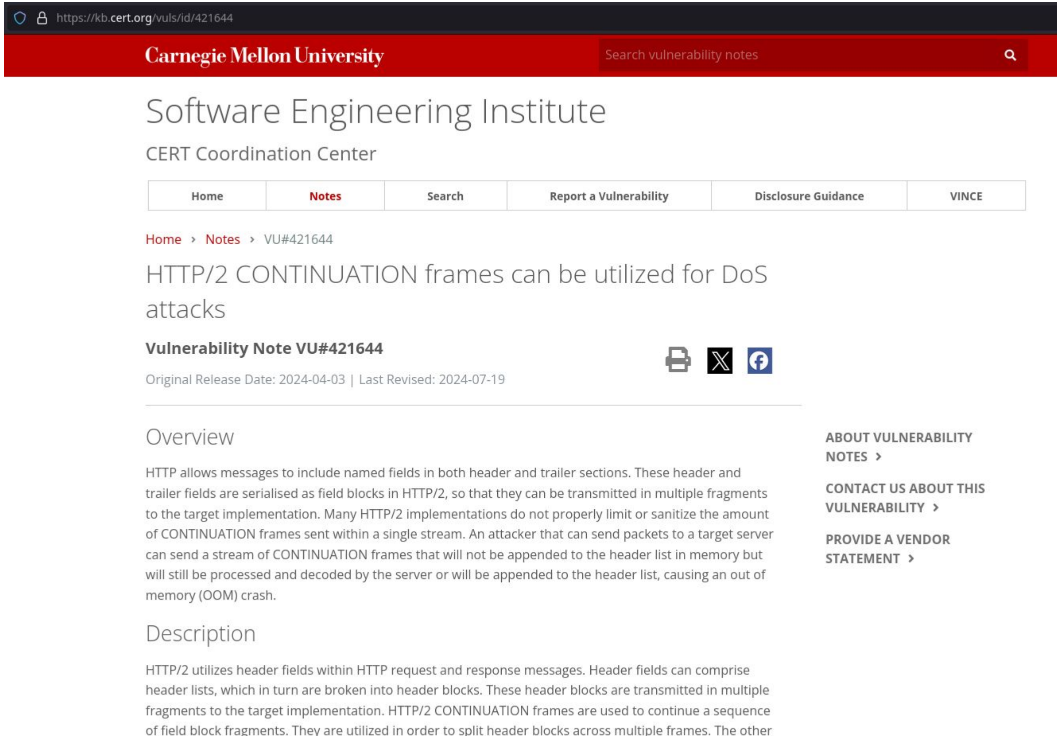
- What is the duration? Typically lasts days to weeks
  - But can last from hours to years (hardware vulnerabilities)
  - Involved parties decide together on the release/publication date
- How strict it is?
  - Depends on the vulnerability!
  - In a strict embargo, you shouldn’t tell outside of the group that there’s an embargo
- Embargo can be “broken”
  - … when someone else learns about the vulnerability before the fix is ready
  - Good practice: have a plan on how to deal with it (for example: a workaround)

Multi-project issue workflow

- Find all stakeholders
  - Check SECURITY files for security contacts
- Establish a communication channel
  - Depends on the severity of the issue, for example a dedicated mailing list
- Agree on the common release date
  - And also the date to merge fixes
- Work on all related fixes and advisories
- Stay in contact with other parties to confirm the release date
- Perform the release, publish advisories, and the global advisory if agreed on

Example: the VINCE platform

- When published, like: https://kb.cert.org/vuls/id/421644



https://kb.cert.org/vuls/id/421644

Carnegie Mellon University

Search vulnerability notes

Software Engineering Institute

CERT Coordination Center

Home › Notes › VU#421644

HTTP/2 CONTINUATION frames can be utilized for DoS attacks

Vulnerability Note VU#421644

Original Release Date: 2024-04-03 | Last Revised: 2024-07-19

Overview

HTTP allows messages to include named fields in both header and trailer sections. These header and trailer fields are serialised as field blocks in HTTP/2, so that they can be transmitted in multiple fragments to the target implementation. Many HTTP/2 implementations do not properly limit or sanitize the amount of CONTINUATION frames sent within a single stream. An attacker that can send packets to a target server can send a stream of CONTINUATION frames that will not be appended to the header list in memory but will still be processed and decoded by the server or will be appended to the header list, causing an out of memory (OOM) crash.

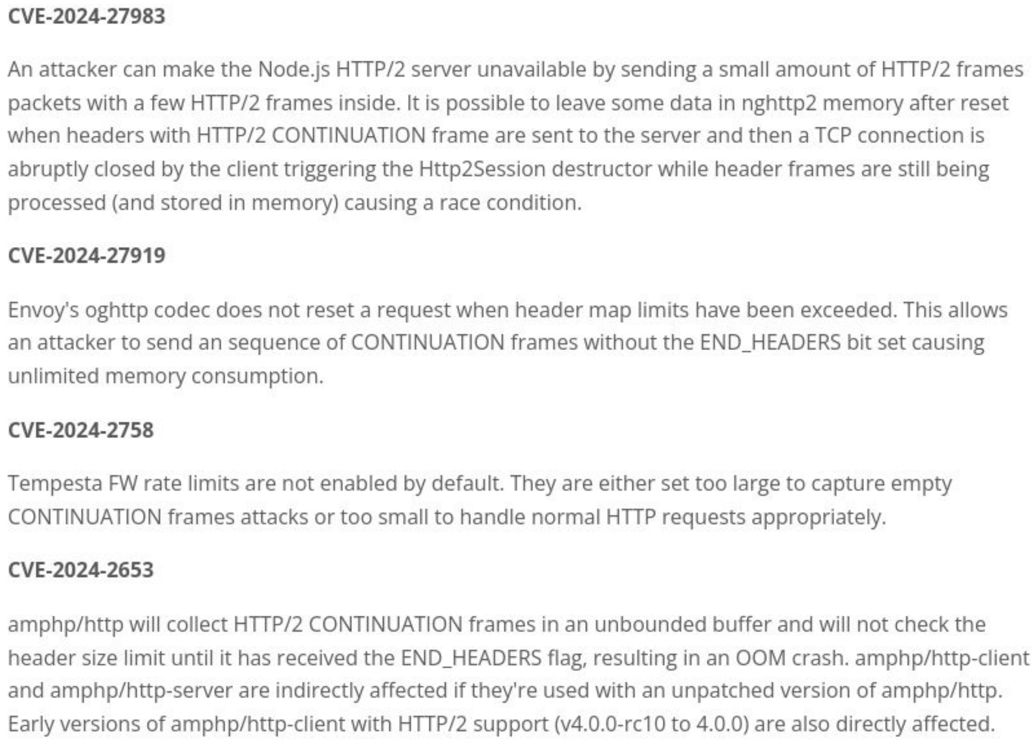
Description

HTTP/2 utilizes header fields within HTTP request and response messages. Header fields can comprise header lists, which in turn are broken into header blocks. These header blocks are transmitted in multiple fragments to the target implementation. HTTP/2 CONTINUATION frames are used to continue a sequence of field block fragments. They are utilized in order to split header blocks across multiple frames. The other

ABOUT VULNERABILITY NOTES ›
CONTACT US ABOUT THIS VULNERABILITY ›
PROVIDE A VENDOR STATEMENT ›

Example: the VINCE platform

- When published, like: https://kb.cert.org/vuls/id/421644



CVE-2024-27983

An attacker can make the Node.js HTTP/2 server unavailable by sending a small amount of HTTP/2 frames packets with a few HTTP/2 frames inside. It is possible to leave some data in nghttp2 memory after reset when headers with HTTP/2 CONTINUATION frame are sent to the server and then a TCP connection is abruptly closed by the client triggering the Http2Session destructor while header frames are still being processed (and stored in memory) causing a race condition.

CVE-2024-27919

Envoy's oghttp codec does not reset a request when header map limits have been exceeded. This allows an attacker to send an sequence of CONTINUATION frames without the END_HEADERS bit set causing unlimited memory consumption.

CVE-2024-2758

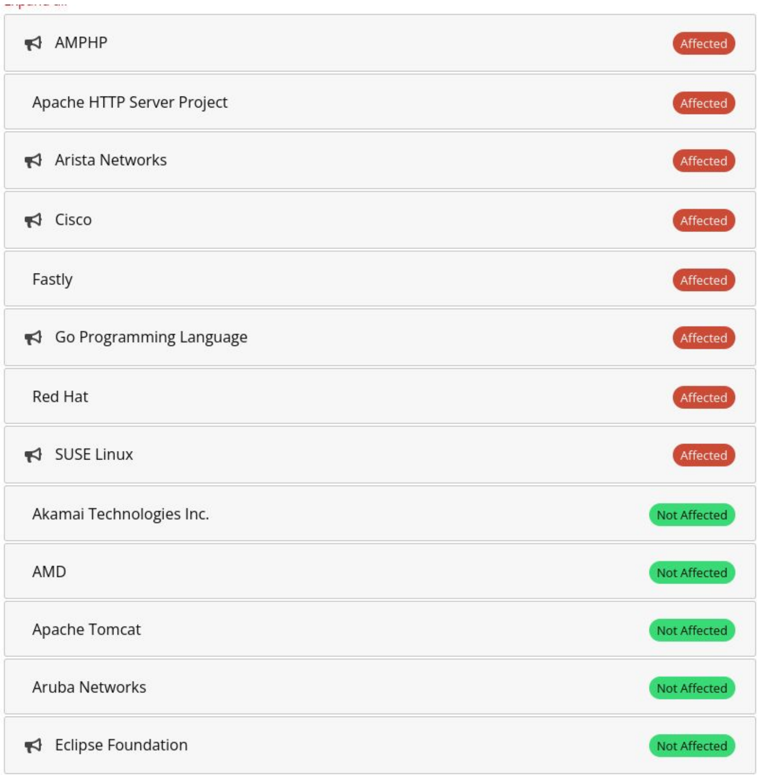
Tempesta FW rate limits are not enabled by default. They are either set too large to capture empty CONTINUATION frames attacks or too small to handle normal HTTP requests appropriately.

CVE-2024-2653

amphp/http will collect HTTP/2 CONTINUATION frames in an unbounded buffer and will not check the header size limit until it has received the END_HEADERS flag, resulting in an OOM crash. amphp/http-client and amphp/http-server are indirectly affected if they're used with an unpatched version of amphp/http. Early versions of amphp/http-client with HTTP/2 support (v4.0.0-rc10 to 4.0.0) are also directly affected.

Example: the VINCE platform

- When published, like: https://kb.cert.org/vuls/id/421644



📢 AMPHP	Affected
Apache HTTP Server Project	Affected
📢 Arista Networks	Affected
📢 Cisco	Affected
Fastly	Affected
📢 Go Programming Language	Affected
Red Hat	Affected
📢 SUSE Linux	Affected
Akamai Technologies Inc.	Not Affected
AMD	Not Affected
Apache Tomcat	Not Affected
Aruba Networks	Not Affected
📢 Eclipse Foundation	Not Affected

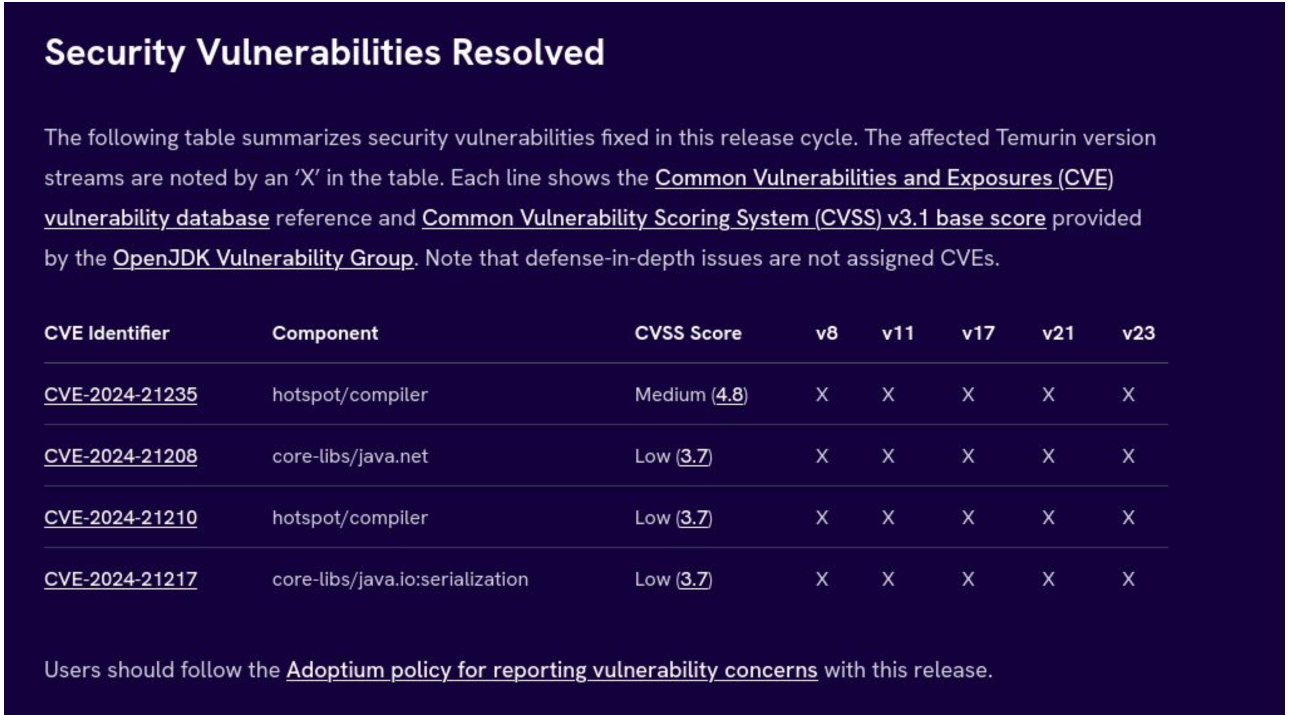
Security advisories

Security advisories

- Information for users
  - Explains the impact of the vulnerability
  - Gives workarounds, additional information
- Recommended content
  - An informative title.
  - A description of the issue.
  - A description of possible workarounds, if applicable.
  - The severity and impact of the issue.
  - A list of affected products (projects) along with the version numbers affected. If a project maintains multiple branches, each should be listed separately, specifying the exact version containing a solution or clearly indicating if a particular version is deprecated.
  - The list of related CVEs, if any.
  - Contact information for inquiries or questions.
  - The name of the issuing authority (e.g., Eclipse Foundation, the project itself, etc.).
  - A list of changes to the advisory, if applicable, in reverse chronological order.

Human-readable advisories

- Example: https://adoptium.net/news/2024/11/eclipse-temurin-8u432-11025-1713-2105-2301-available/



Security Vulnerabilities Resolved

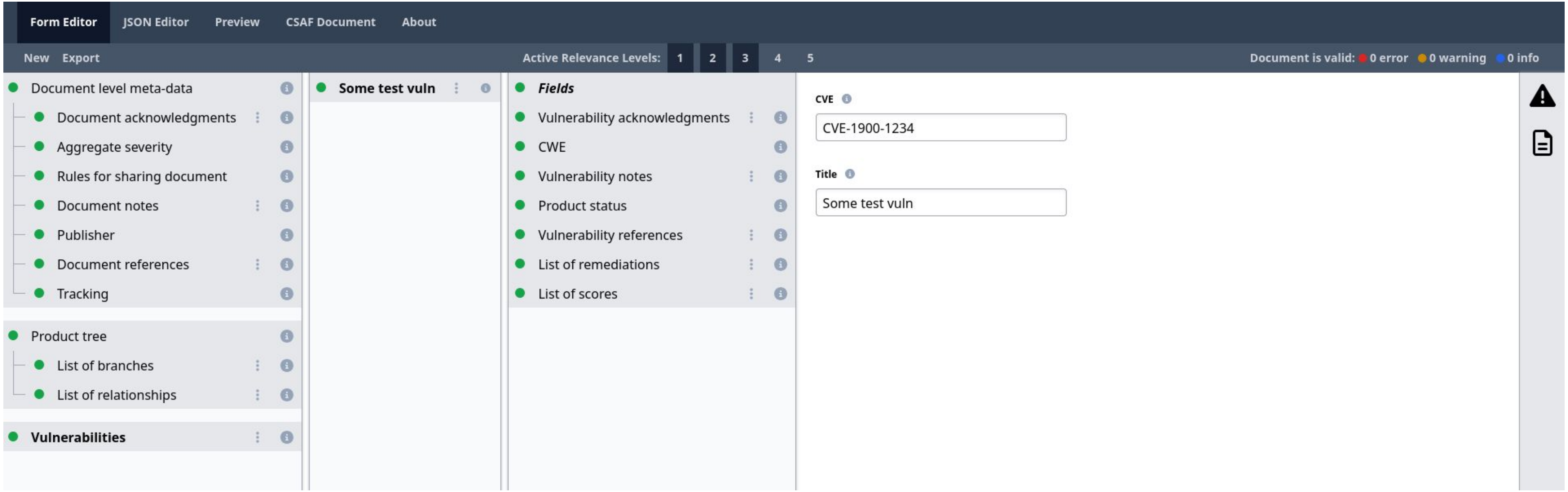
The following table summarizes security vulnerabilities fixed in this release cycle. The affected Temurin version streams are noted by an 'X' in the table. Each line shows the Common Vulnerabilities and Exposures (CVE) vulnerability database reference and Common Vulnerability Scoring System (CVSS) v3.1 base score provided by the OpenJDK Vulnerability Group. Note that defense-in-depth issues are not assigned CVEs.

CVE Identifier	Component	CVSS Score	v8	v11	v17	v21	v23
CVE-2024-21235	hotspot/compiler	Medium (4.8)	X	X	X	X	X
CVE-2024-21208	core-libs/java.net	Low (3.7)	X	X	X	X	X
CVE-2024-21210	hotspot/compiler	Low (3.7)	X	X	X	X	X
CVE-2024-21217	core-libs/java.io:serialization	Low (3.7)	X	X	X	X	X

Users should follow the Adoptium policy for reporting vulnerability concerns with this release.

Machine-readable advisories

- Popular format: CSAF
  - Generator tool: https://secvisogram.github.io/



Document is valid: ● 0 error ● 0 warning ● 0 info

CVE: CVE-1900-1234

Title: Some test vuln

Communication with stakeholders

Developers communicate with

- Security researchers
  - They may NOT have software development background
  - They usually want a CVE number
- Users
  - Might not have knowledge of security processes
- Other developers (upstream or downstream)
  - Usually submit useful reports
  - Want notification of a fix release
- Media
  - Might request statements (for severe vulnerabilities)
  - Their messages are a sign on broken embargo

Typical communication issues

- Urgent fix requests
- High loads of reports (vulnerability batches)
- Reports from automatic tools, not verified
- Regular bug reports as security issues
- Disagreements between reporters and developers
  - “It is a vulnerability” - “No, it isn’t”
- Difficulties in English

Communication tips

- Stay polite
- Ask for a coordinator if needed
  - Eclipse Foundation Security Team is one
- Know your security policy
  - But reporters MAY release vulnerability information when they want
- Use simple English, reword if necessary

Quiz

Readings

- Guide to implementing a coordinated vulnerability disclosure process for open source projects https://github.com/ossf/oss-vulnerability-guide/blob/main/maintainer-guide.md#readme
- Eclipse Foundation Security Handbook https://eclipse-csi.github.io/security-handbook/index.html
- CSAF Author Guide: https://secvisogram.github.io/secvisogram-documentation/

Thank you!