sAIfer Lab
Joint lab on Safety and Security of AI

HTTP/3 will not Save you from Request Smuggling: A Methodology to Detect HTTP/3 Header (mis)Validations

Lorenzo Pisu, Federico Loi, Davide Maiorca, Giorgio Giacinto
lorenzo.pisu@unica.it, federico.loi@mindedsecurity.com, {davide.maiorca, giacinto}@unica.it
Affiliations: University Of Cagliari, IMQ Minded Security

The 22nd International Symposium on Network Computing and Applications (NCA 2024)

UNICA
UNIVERSITÀ DEGLI STUDI DI CAGLIARI

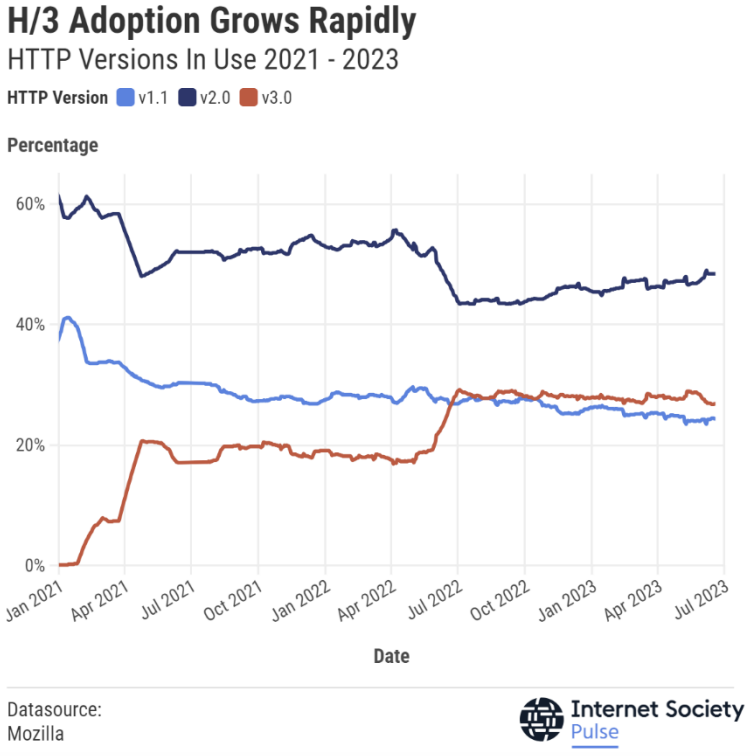
HTTP/3

How it works

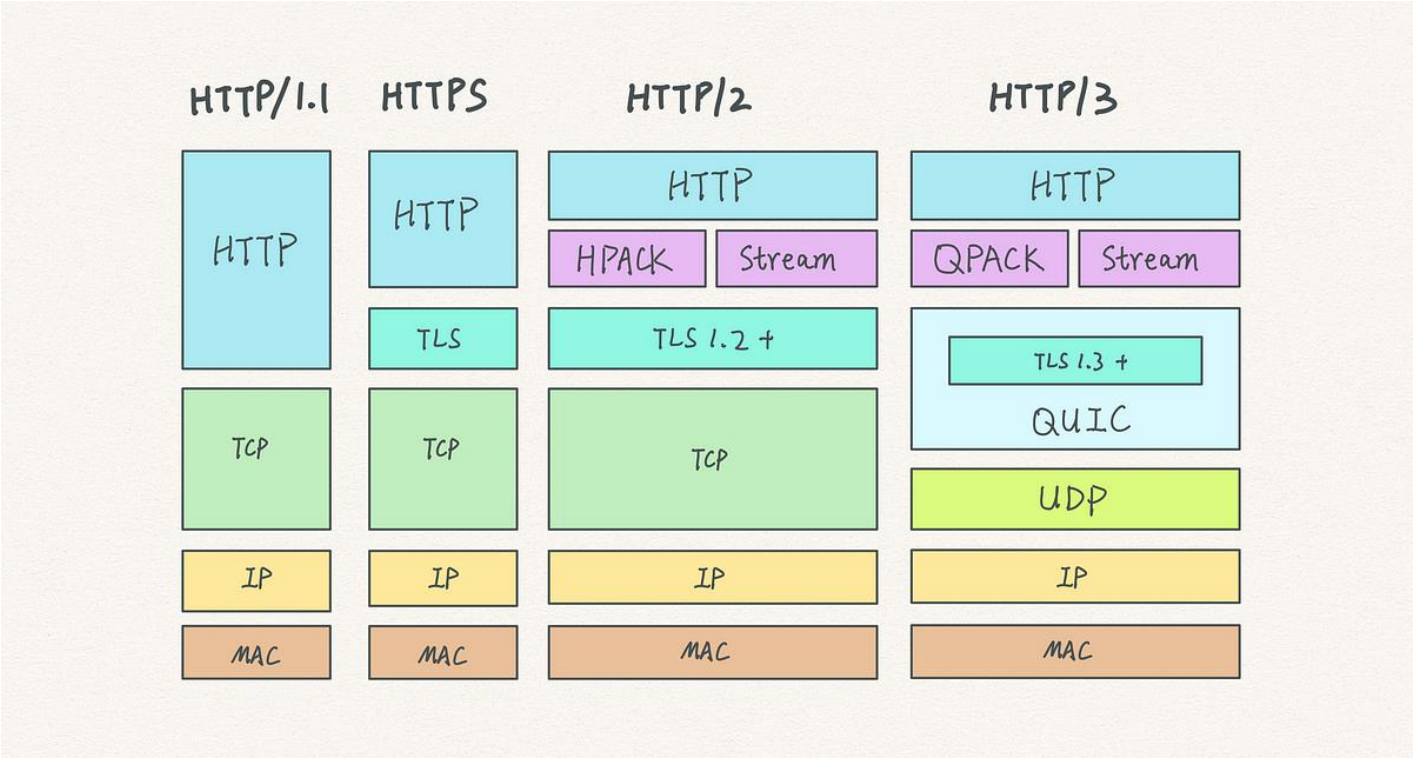
HTTP/3 Usage

- HTTP/3 is quickly **expanding** and it is widely **supported** by browsers

- However, **few research papers explore its security**

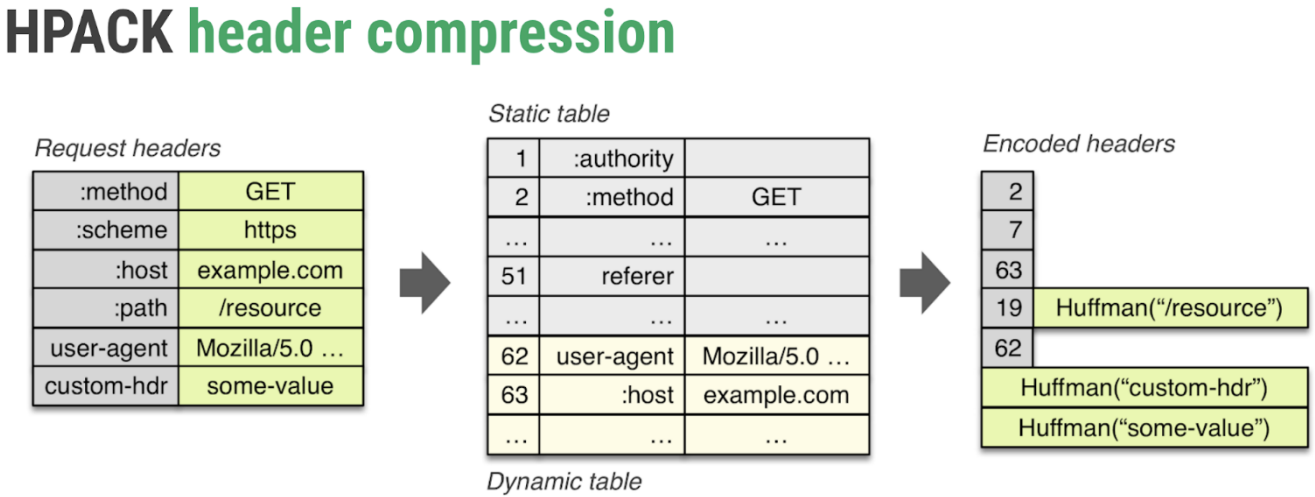


HTTP/3 vs HTTP/1 and 2



HTTP/3 vs HTTP/1 and 2

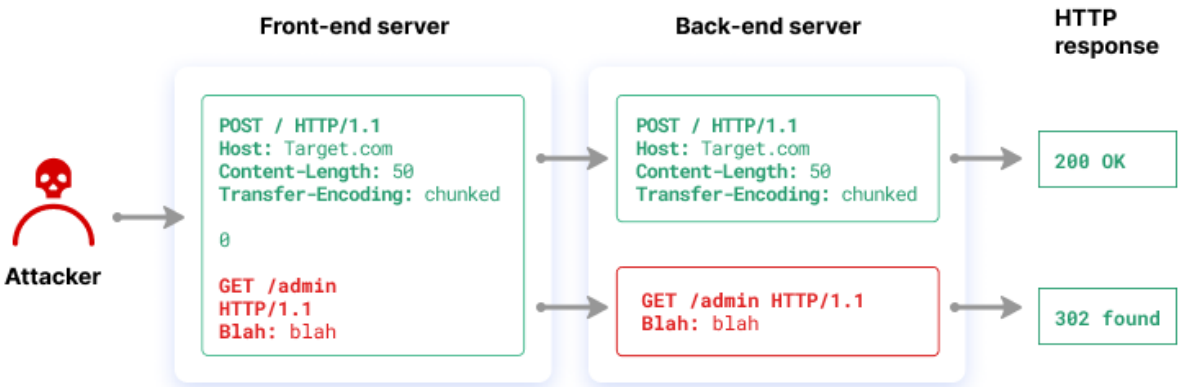
- **HTTP/1** sends requests **without compressions**
- **HTTP/2 and 3 apply compressions** (HPACK and QPACK)



Background

Request Smuggling

- Request smuggling is an attack that arises when two or more servers parse the same request in different ways
- **Example: conflicting headers**



HTTP versions conversion

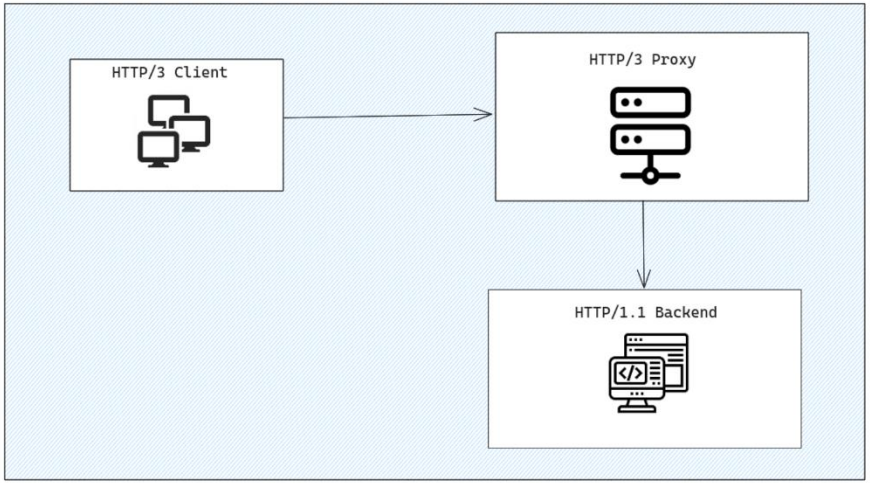
- Proxies support HTTP version conversion
- Example: **HTTP/3 to HTTP/1**
- **This can be dangerous**

HTTP/3 does not use the Connection header field to indicate connection-specific fields; in this protocol, connection-specific metadata is conveyed by other means. An endpoint **MUST NOT** generate an HTTP/3 field section containing connection-specific fields; any message containing connection-specific fields **MUST** be treated as malformed.

The only exception to this is the TE header field, which **MAY** be present in an HTTP/3 request header; when it is, it **MUST NOT** contain any value other than "trailers".

An intermediary transforming an HTTP/1.x message to HTTP/3 **MUST** remove connection-specific header fields as discussed in Section 7.6.1 of [HTTP], or their messages will be treated by other HTTP/3 endpoints as malformed.

Source: https://datatracker.ietf.org/doc/html/rfc9114



Detection Methodology

Methodology

- Request smuggling arises from not following RFC specifications
- We extracted from the RFCs a set of restrictions

RFC Restriction	Description
Field Name Restrictions	The presence of characters within the forbidden ranges (0x00-0x20, 0x41-0x5a, 0x7f-0xff) in field names must be avoided to ensure compliance with RFC standards. Emphasis was placed on detecting non-visible ASCII characters, uppercase characters, and ASCII SP (0x20) occurrences within field names.
Colon Restrictions	The prohibition of colons (ASCII COLON, 0x3a) in field names, except for pseudo-header fields, is essential to prevent ambiguity and parsing errors in HTTP/3 requests.
Field Value Constraints	The absence of zero values (ASCII NUL, 0x00), line feeds (ASCII LF, 0x0a), carriage returns (ASCII CR, 0x0d), and leading/trailing ASCII whitespace characters (ASCII SP or HTAB, 0x20 or 0x09) within field values must be validated to ensure data integrity and prevent injection attacks.
Transfer-Encoding header	Transfer codings are not defined in HTTP/3. The `transfer-encoding` header must not be used in HTTP/3. The only exception is when the header contains the value `trailer`.
Content-Length header	`Content-Length` headers are allowed in HTTP/3, although they are not necessary as the length of the request is calculated automatically. However, if a `Content-Length` header is present, its length must equal the length of the data in the request body.

TABLE II: HTTP/3 Header Restrictions (as per RFC 9114)

Methodology

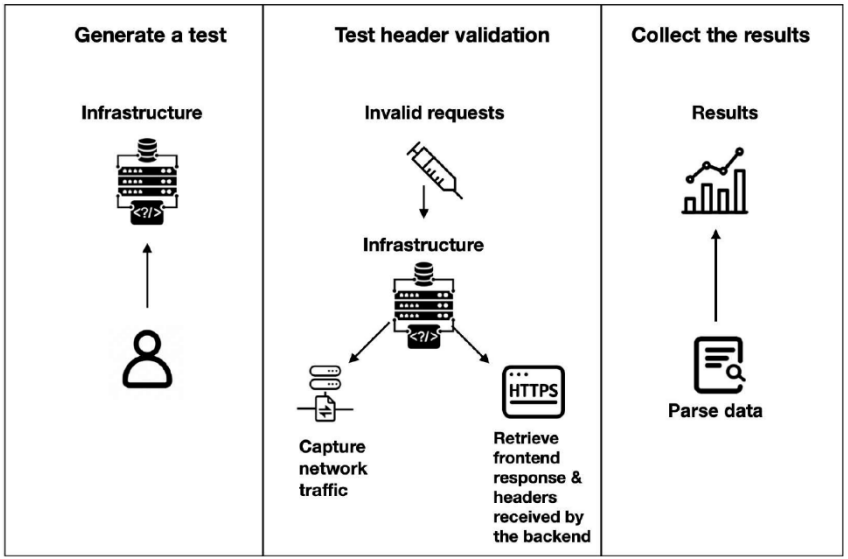
From each RFC violation we extract possible vulnerabilities, defining a taxonomy of HTTP/3 request smuggling attacks

- HTTP/3 Content-Length
- HTTP/3 Transfer-Encoding
- HTTP/3 Request splitting and response queue poisoning
- HTTP/3 Tunneling
- HTTP/3 Conflicting headers

Methodology

- We created a tool that crafts malicious requests

https://github.com/lpisu98/HTTP3-Smuggling-Tool



Experimental Evaluation

- We tested our tool against 5 popular proxies:
  - **Aioquic**
  - **Caddy**
  - **Haproxy (2.7 and 3.0)**
  - **Nginx**
  - **Traefik**
- Haproxy 2.7 has a vulnerability (**CVE-2023-25950**) related to request smuggling
- We use this version of Haproxy to confirm that **our tool can spot the vulnerability**

Results

- ✗ indicates failed validations
- ⚠ indicates connection timeouts
- ✂ indicates modifications of the request
- ✓ indicates successful validations

Proxy	Header value	Header name	Other	Total
Aioquic	5 ✗	26 ✓ 162 ✗	7 ✓ 3 ✗	33 ✓ 170 ✗
Caddy	3 ✓ 2 ✗	162 ✓26 ✂	1 ✓3 ⚠6 ✂	166 ✓ 3 ⚠ 32 ✂ 2 ✗
Haproxy (2.7)	5 ✗	188 ⚠	10 ⚠	198 ⚠ 5 ✗
Haproxy (3.0)	3 ⚠ 2 ✗	188 ⚠	10 ⚠	201 ⚠ 2 ✗
Nginx	3 ✓ 2 ✗	58 ✓ 130 ✂	9 ✓ 1 ✂	70 ✓ 131 ✂ 2 ✗
Traefik	3 ✓ 2 ✗	188 ⚠	7 ⚠ 3 ✂	3 ✓ 195 ⚠ 3 ✂ 2 ✗

Conclusion and future works

- HTTP/3 proxies can have security problems
- To prevent vulnerabilities, **proxies should strictly adhere to RFC specifications**

Future works:

- **More proxies can be analyzed with our tool**
- Of each proxy **multiple versions can be analyzed**, based on their real-world spreading

sAIfer Lab

Joint lab on Safety and Security of AI

Thank you for the attention

Lorenzo Pisu

lorenzo.pisu@unica.it

www.saiferlab.ai

UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI