

**Summary of "HTTP Request Smuggling in 2020 – New Variants, New Defenses and New Challenges" by Amit Klein**

**HTTP Request Smuggling** is an attack technique that exploits discrepancies in how different HTTP devices (such as proxies, load balancers, and web servers) interpret non-standard or ambiguous HTTP requests. By manipulating the way these devices split a stream of HTTP requests, an attacker can "smuggle" a malicious request through to the server, causing desynchronization between the intermediary and the server’s understanding of the HTTP request and response streams[[1]](#fn1)[[2]](#fn2)[[3]](#fn3).

**Key Contributions**

* **New Attack Variants:**  
  The paper introduces several new HTTP Request Smuggling variants that work against various proxy-server and proxy-proxy combinations. Notably:
  + **Header SP/CR Junk:** Exploits differences in how systems handle multiple or malformed Content-Length headers.
  + **Wait for It:** Uses timing and incomplete request bodies to manipulate server behavior.
  + **HTTP/1.2 to Bypass mod\_security:** Leverages the fact that many servers accept HTTP/1.2 as HTTP/1.1, bypassing some security rules.
  + **Plain Solution:** Abuses Content-Type text/plain to bypass certain security checks.
  + **CR Header ("0dwrap"):** Uses carriage return in header names to bypass parsing logic in some systems[[1]](#fn1)[[4]](#fn4)[[5]](#fn5).
* **New Defenses:**  
  The author proposes a robust, open-source C++ library ("Request Smuggling Firewall") that enforces strict, whitelist-based HTTP/1.x protocol validation at the socket layer. This defense inspects and validates incoming requests before they reach the application, terminating connections upon protocol violations. The solution is portable across Windows and Linux and can be applied to a wide range of servers and proxies[[1]](#fn1).
* **New Challenges:**  
  The research also documents borderline cases and anomalies in server behaviors that are not yet exploitable but may lead to new attack vectors in the future. Examples include headers with non-standard formatting, signed Content-Length values, and ambiguous whitespace handling[[1]](#fn1).

**Research Scope**

The study tested a wide range of web servers and proxies, including IIS, Apache, nginx, Node.js, Abyss, Tomcat, Varnish, lighttpd, Squid, Caddy, Traefik, and HAProxy, in both web server and proxy modes. The findings highlight that many combinations remain vulnerable to these new variants[[1]](#fn1)[[4]](#fn4).

**Impact and Risk**

* **Risks:**  
  HTTP Request Smuggling can lead to cache poisoning, session hijacking, bypassing of authentication and security controls, and even direct compromise of backend systems. The risk is compounded by the fact that vulnerable combinations can exist between unrelated organizations (e.g., a website’s server and a client’s ISP proxy), making comprehensive mitigation challenging[[4]](#fn4).
* **Vendor Response:**  
  The paper notes that several vendors have issued fixes or configuration workarounds in response to these findings. For example, Aprelium (Abyss) and Squid addressed multiple variants, and the OWASP Core Rule Set (CRS) implemented fixes for others[[1]](#fn1)[[4]](#fn4).

**Conclusions**

Despite being a 15-year-old class of vulnerabilities, HTTP Request Smuggling remains relevant and dangerous. The paper demonstrates new attack techniques, the limitations of current open-source defenses, and provides a proof-of-concept solution for robust protection. It also highlights that the field is still evolving, with uncharted territory for future research and exploitation[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)[[4]](#fn4).

**References:**  
[[1]](#fn1) us-20-Klein-HTTP-Request-Smuggling-In-2020-New-Variants-New-Defenses-And-New-Challenges-wp.pdf  
[[2]](#fn2) Black Hat 2020 PDF  
[[3]](#fn3) ExtraHop Black Hat summary  
[[4]](#fn4) PortSwigger/The Daily Swig coverage  
[[5]](#fn5) [Intruder.io](http://Intruder.io) practical demonstration

⁂

1. us-20-Klein-HTTP-Request-Smuggling-In-2020-New-Variants-New-Defenses-And-New-Challenges-wp.pdf

1. <https://i.blackhat.com/USA-20/Wednesday/us-20-Klein-HTTP-Request-Smuggling-In-2020-New-Variants-New-Defenses-And-New-Challenges-wp.pdf>

1. <https://i.blackhat.com/executive-interviews/2020/20201029_ExtraHop_HTTP-Request-Smuggling-in-2020_v2.pdf>

1. <https://portswigger.net/daily-swig/black-hat-2020-new-http-request-smuggling-variants-levied-against-modern-web-servers>

1. <https://www.intruder.io/research/practical-http-header-smuggling>