WEBINAR SUMMARY

October 29, 2020

HTTP Request Smuggling in 2020

Amit Klein, VP Security Research, SafeBreach
Jesse Munos, Technical Marketing Manager, ExtraHop

KEY TAKEAWAYS

- Web and proxy servers remain vulnerable to HTTP request smuggling.
- NDR shifts the advantage to the defender.
- Reveal(x) provides the information necessary to stop HTTP request smuggling.

in partnership with

ExtraHop

OVERVIEW

HTTPS request smuggling was first identified in 2005 as a way of attacking systems by exploiting how proxy and web servers receive HTTP requests. Fifteen years later, HTTP request smuggling, also known as an HTTP desync attack, remains a significant security problem for businesses.

Network detection and response (NDR) solutions, like ExtraHop Reveal(x), help organizations identify potential attacks, including modern variants of HTTP request smuggling attacks, so that they can be resolved.

CONTEXT

Amit Klein discussed HTTP request smuggling and its continued impact on today's IT environments. Jesse Munos shared the benefits of NDR solutions, focusing on ExtraHop Reveal(x) and how it helps identify HTTP request smuggling attacks.

KEY TAKEAWAYS

Web and proxy servers remain vulnerable to HTTP request smuggling.

HTTP request smuggling interferes with the way that a website processes how HTTP requests are received, allowing an attacker to gain unauthorized access to a system. These attacks have been a known issue since 2005 and continue to be problematic for modern web and proxy servers today.

> HTTP request smuggling is still seen in 2020, even in commercial software. Existing open source solutions are also lacking protection.
>
> *Amit Klein, SafeBreach*

This attack works because the proxy and web servers interpret the transmission control protocol (TCP) stream in different ways. In the example below, a request with two content-length (CL) headers is interpreted differently by the proxy and web servers; the web server cache is poisoned and the attacker is able to gain access. The green text shows the cache poisoning.

Table 1: Example : HTTP Request Smuggling Using Different Content Lengths

Request Sent	POST /hello.php HTTP/1.1 … Content-Length: 0 Content-Length: 44 GET/poison.html HTTP/1.1 Host: www.example.com Something: GET /target.html HTTP.1.1
Caching Proxy (uses the last CL)	1. /hello.php (44 bytes in body) 2. **/target.html**
Web Server (uses the first CL)	1. /hello.php (0 bytes in body) 2. **/poison.html** (+headers)

Although some web and proxy servers may be better protected today against basic, older HTTP request smuggling methods, new variants on these methods have been seen. While proxy server and web server vendors are closing some reported holes and the Open Web Application Security Project (OWASP) has modified its core rule set (CRS) to resolve issues, HTTP request smuggling still remains an issue.

NDR shifts the advantage to the defender.

Without proper detection and visibility solutions in place, organizations are likely to find they are at a disadvantage compared to attackers. NDR solutions, including ExtraHop Reveal(x), shift the advantage to the defender. Defending organizations are able to identify and stop incoming attacks, including HTTP request smuggling attempts.

Table 2: NDR Characteristics

- **Use out-of-band detection,** unlike most intrusion detection systems (IDS) and intrusion prevention systems (IPS).
- **Are agentless and tamperproof;** agents are not deployed to the endpoints, which simplifies implementation and makes the system harder for attackers to tamper with or disable.
- **Are difficult to detect or evade** because of the agentless and out-of-bend deployment.
- **Provide complete network visibility,** tracking North-South data coming into and out of the network as well as East-West data moving between systems on the network.
- **Provide full visibility into all traffic,** including transport layer security (TLS) 1.3 encryption.
- **Are complementary to other detection systems,** including endpoint detection and response (EDR) and security information and event management (SIEM) systems.

> If attackers want to get into an environment with this kind of technology, they have to focus on exactly how they can hide within that environment, and it's not going to apply to other environments.
>
> *Jesse Munos, ExtraHop*

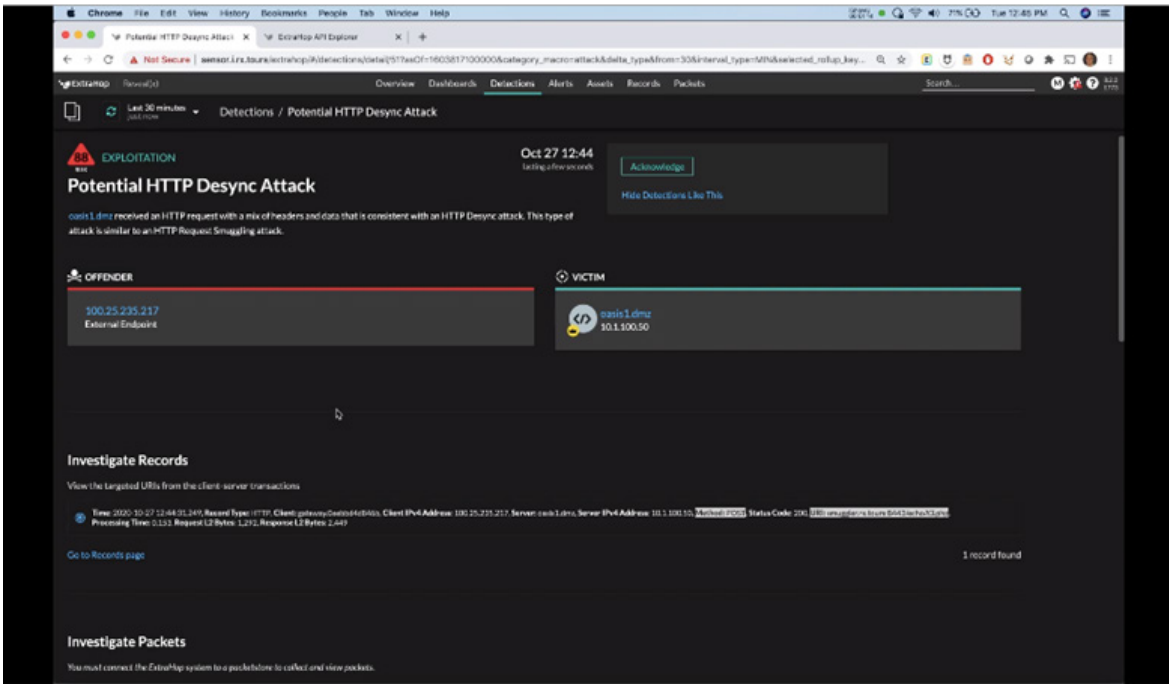
Reveal(x)'s advanced algorithmic detection running in both the cloud and on-premise systems further shifts the advantage in the defender's favor. Examples of what Reveal(x) can detect include:

- **Domain generation algorithms (DGA),** which are almost exclusively related to malware, with better than 98% accuracy.
- **Brute force attempts,** using deviations from normal baselines.
- **Command and control (C2) beaconing,** even those that are encrypted.

Reveal(x) provides the information necessary to stop HTTP request smuggling.

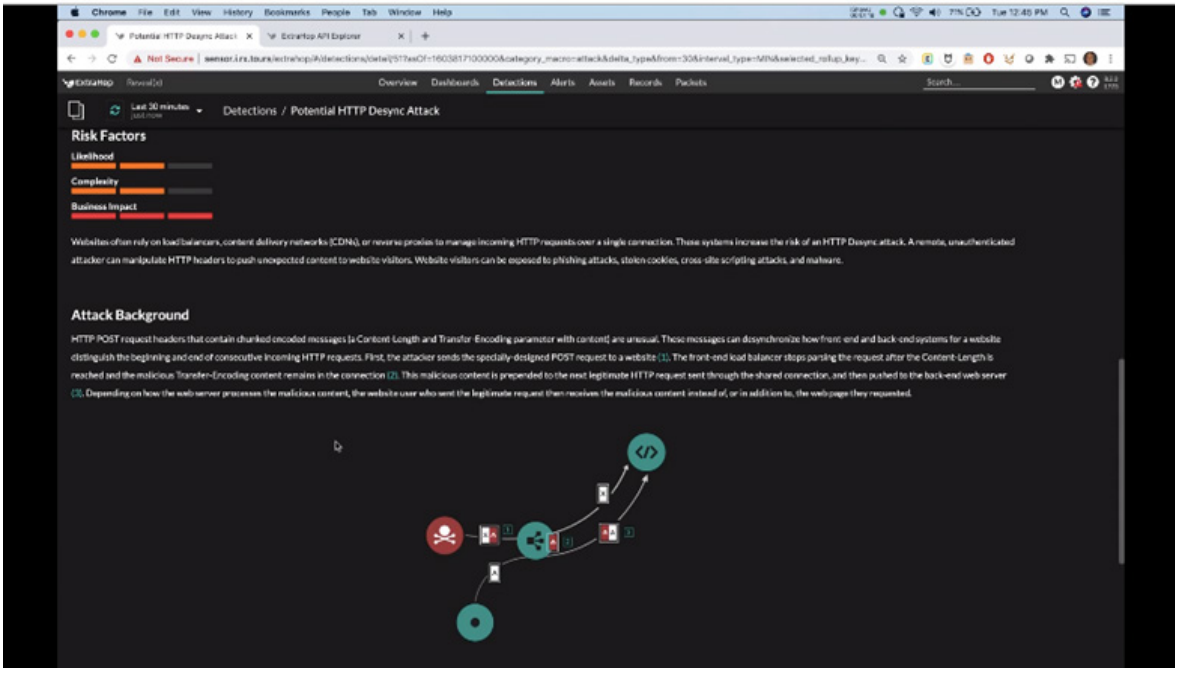
Security teams can use Reveal(x) to quickly identify HTTP request smuggling attempts, which the software refers to as HTTP desync attacks.

Figure 1: Investigating Potential HTTP Desync Attacks In Reveal(x)



Reveal(x) does not natively respond to malicious behavior, but it provides the information necessary for analysts to understand the attack and the potential impact on the business. ExtraHop also offers a broad spectrum of integrations with security products, including firewalls and EDR, to enable not just quick identification but quick response to limit impact.

Figure 2: Reveal(x) Details the Attack Background and Risk Factors to Guide Decision Making



BIOGRAPHIES

Amit Klein

VP Security Research, SafeBreach

Amit Klein is a world-renowned information security expert, with 29 years in information security and over 30 published technical and academic papers on this topic. Amit is the VP Security Research at SafeBreach, responsible for researching various infiltration, exfiltration, and lateral movement attacks. Before SafeBreach, Amit was the CTO for Trusteer (acquired by IBM) for 8.5 years. Prior to Trusteer, Amit was Chief Scientist for Cyota (acquired by RSA) for 2 years, and prior to that, Director of Security and Research for Sanctum (acquired by Watchfire, now part of IBM security division) for 7 years.

Amit has a B.Sc. from the Hebrew University in Mathematics and Physics (magna cum laude, Talpiot program), recognized by InfoWorld as a CTO of the year 2010, and has presented at Black Hat USA, DEF CON, Usenix, NDSS, InfoCom, DSN, HITB, RSA, OWASP, CertConf, BlueHat, CyberTech, APWG, and AusCERT.

Jesse Munos

Technical Marketing Manager, ExtraHop

Jesse Munos is Technical Marketing Manager for ExtraHop where he provides competitive analysis and technical content to his marketing-focused peers. Jesse started his career in 2014 as an escalations engineer with Cisco Systems where he focused on EDR and Malware Sandboxing technologies and API integrations. During that time he also presented at Cisco Live providing deep dive technical breakdowns and executive level briefings on Cisco's security portfolio. He focuses on pushing best-of-breed technology solutions that meet current customer needs while guiding product development to embrace the broader ecosystem integrations.