An Expert Analysis of the HTTP/2 Rapid Reset DDoS Attack: A New Paradigm of Protocol-Based Exploitation

Executive Summary: The New Paradigm of Protocol-Based DDoS

The late 2023 HTTP/2 Rapid Reset attack represents a watershed moment in the evolution of distributed denial-of-service (DDoS) cyberattacks. It is distinguished not by its brute-force volumetric scale alone, but by a novel and unprecedented level of technical sophistication and efficiency. The attack exploited a zero-day vulnerability (CVE-2023-44487) in the fundamental HTTP/2 web protocol, demonstrating a critical shift in the threat landscape. The report's investigation found that the primary targets were major internet infrastructure providers—Google, Cloudflare, and AWS—which were hit by a sustained campaign of attacks. The attack mechanism leveraged a simple yet devastating "request, cancel" loop, allowing a relatively small botnet to generate an astonishing peak of 398 million requests per second (RPS), a figure that shattered all previous records. This incident revealed a systemic vulnerability and underscored that traditional, volume-based DDoS defenses are now obsolete against intelligent, protocol-level exploits. The strategic takeaway from this event is clear: cybersecurity strategies must pivot from a focus on bandwidth saturation to a nuanced, intelligent defense capable of detecting and mitigating resource exhaustion at the application layer.

1. Introduction: A New Frontier in Cyberwarfare

1.1 Context and Significance

The global landscape of cyber threats continues to evolve at an accelerating pace. While the frequency of distributed denial-of-service (DDoS) attacks has seen a dramatic increase, with projections indicating a doubling from 2018 to 2023 [1] and a 117% year-over-year rise in network-layer attacks in late 2023 [2], the HTTP/2 Rapid Reset attack marks a qualitative leap in complexity. This incident, which unfolded in late 2023, was not merely a larger variant of prior attacks; it was a seminal event that fundamentally redefined the capabilities of threat actors. It was a zero-day vulnerability [3], a previously unknown flaw in a core internet protocol, providing adversaries with a "critical new tool" [5] that could be exploited to launch attacks at a scale and efficiency never before witnessed. The fact that the vulnerability was found within HTTP/2, a protocol that is integral to how a significant portion of the internet operates, elevates this event from a routine cyberattack to a matter of systemic digital security.

1.2 Selection Rationale

This analysis focuses on the HTTP/2 Rapid Reset attack due to its profound significance and the unique opportunity it presents for in-depth study. The incident's recent occurrence in the latter half of 2023 [6] and the subsequent public disclosure by multiple major technology firms—including Google, Cloudflare, and AWS [5]—provided an unprecedented wealth of detailed, first-hand information. Unlike many other cyber events that remain cloaked in secrecy, the collaborative and responsible disclosure process initiated by these companies allows for a thorough, multi-faceted investigation into the attack's technical methodology, its unprecedented impact, and the critical lessons learned. This detailed information makes the HTTP/2 Rapid Reset attack an ideal case study for an expert-level report, enabling a deep, technical analysis that goes far beyond a simple recitation of facts.

2. Attack Deconstruction: The Technical Anatomy of a Protocol Exploit

2.1 The HTTP/2 Protocol Explained

To understand the attack, a foundational knowledge of the HTTP/2 protocol is essential. HTTP/2, a major revision of the HTTP network protocol, was designed to improve web performance by addressing the limitations of its predecessor, HTTP/1.1. A key feature is **stream multiplexing**, which allows for multiple requests and responses to be sent over a single TCP connection concurrently. This contrasts sharply with HTTP/1.1, where each request typically required its own separate connection. Within the HTTP/2 framework, a client and server communicate via a series of binary-encoded messages known as "frames".[8] One such frame is the

RST_STREAM frame, a legitimate and a necessary component of the protocol. Its purpose is to prematurely terminate a stream or to signal that no further request or response data will be processed on that stream.[8] In normal operation, a client might use this frame to cancel a request it no longer needs. The HTTP/2 Rapid Reset attack, however, demonstrates how this seemingly benign feature can be weaponized with devastating effect.

2.2 The Attack Mechanism (CVE-2023-44487)

The core of the HTTP/2 Rapid Reset attack, officially designated as CVE-2023-44487, lay in the malicious exploitation of HTTP/2's stream cancellation mechanism. The attack unfolded as a deceptively simple "request, cancel" loop.[4] An attacker would initiate a large number of requests over a single HTTP/2 connection but, instead of waiting for a response, would immediately cancel each request using the

RST_STREAM frame.[7] This process was then repeated continuously and at immense scale. The key to the attack's success was that by immediately canceling the requests, the attacker never violated the server's pre-defined limit on the number of concurrent open streams.[9] To the server's stateful logic, it appeared as though the attacker was operating within the normal bounds of a single connection. However, the rapid "churn" of continuously opening and immediately closing streams forced the server to expend significant computational resources on starting and then discarding operations.[8] This led to resource exhaustion and a denial-of-service condition without ever needing to saturate the network link.

The efficacy of this method was its most striking feature. As reported by Cloudflare, the record-breaking attack involved a "modestly-sized botnet, consisting of roughly 20,000 machines".[4] A botnet of this size is not considered large in the context of historical DDoS attacks, some of which have leveraged millions of compromised devices. Yet, this relatively small number of machines was able to generate a peak of 398 million requests per second, a figure that dwarfs all previous records.[5] This profound disparity between the size of the botnet and the scale of the resulting attack traffic demonstrates a fundamental shift in the threat model. The goal was no longer to overwhelm a network with sheer volumetric force, but to exploit a protocol-level vulnerability to achieve a massively amplified attack with minimal resources. This approach also allowed the attack to bypass traditional DDoS defenses, which are typically designed to detect and block traffic from high-volume sources or to rate-limit a large number of inbound connections.[6] Since the attack leveraged a single, seemingly normal connection, these defenses were rendered ineffective.

2.3 Variants and the Evolution of the Threat

Following the initial wave of attacks, threat actors demonstrated a rapid and adaptive evolution of their tactics, leading to the discovery of a new variant dubbed "MadeYouReset." This new attack method further complicated detection and mitigation efforts.[10] While the original Rapid Reset attack relied on the client sending a flood of

RST_STREAM frames, the new variant was designed to make the server perform the attack on itself.[10] Attackers would send subtly malformed frames that, while not immediately breaking the protocol at a packet level, would trigger a protocol violation according to the HTTP/2 specification (RFC 9113).[10] Examples of these violations include a

WINDOW_UPDATE frame with a zero increment, which is explicitly disallowed, or a PRIORITY frame with an incorrect length.[10] In response to these violations, the specification dictates that the server must respond with a

RST_STREAM frame to close the stream cleanly.[10] By carefully crafting these violations, the attacker could force the server into a loop of error-handling and resource cleanup, causing the same resource exhaustion as the original attack but without a high volume of suspicious

RST_STREAM frames originating from the client.[10] This innovation highlighted a continuous and rapid cat-and-mouse game between defenders and attackers, as the latter quickly developed a means to bypass the most obvious mitigation strategies.

3. Targets, Motives, and Impact Assessment

3.1 Primary Targets and Affected Parties

The HTTP/2 Rapid Reset attacks were not directed at isolated, niche targets. Instead, they were a "persistent and deliberately engineered campaign" [2] aimed at the core pillars of modern internet infrastructure. The primary targets included Google, Cloudflare, and AWS [6], which collectively host a vast portion of the global web. The attacks were not merely a nuisance but a serious threat to the stability of the internet itself. Google reported that its services, its Google Cloud infrastructure, and its customers were targeted.[5] While the attacks were largely mitigated, Cloudflare's network experienced "intermittent edge instability," which led to performance impacts and temporary errors for a small number of customers.[4] This demonstrated that even with state-of-the-art defenses, the sheer scale and novel nature of the attack posed a significant operational challenge.

3.2 Attacker Motives and Attribution

The specific threat actor behind the HTTP/2 Rapid Reset attacks remains "unknown".[4] Given the high level of technical sophistication and the resources required to develop and execute a zero-day exploit, the actor was likely well-resourced. While the provided data references general trends of hacktivism and politically motivated attacks [1], the targeting of foundational internet infrastructure suggests motives that could range from financial gain through extortion to state-sponsored sabotage or simply the demonstration of a new, powerful cyberweapon. The attacks were a "critical new tool" [5] that could be leveraged for any of these purposes.

3.3 Scale and Operational Impact

The most compelling aspect of the HTTP/2 Rapid Reset attacks was their unprecedented scale, measured in requests per second (RPS). The attack on Google, which peaked at 398 million RPS, shattered all previous records.[5] For context, Google noted that this peak, which was sustained for two minutes, generated more requests than the total number of article views on Wikipedia during the entire month of September 2023.[5] Cloudflare also reported mitigating a record-breaking attack that peaked at 201 million RPS, a figure three times larger than its previous record of 71 million RPS from earlier in the year.[7] AWS reported a similar attack peaking at 155 million RPS in August 2023.[5] The collective response from these major competitors, who put aside competitive differences to engage in a collaborative "responsible disclosure" process [4], serves as a powerful testament to the gravity of the threat. The attack was not just a threat to a single business; it was a systemic risk to the digital security of the global internet. The following table provides a clear comparison of the key metrics from the attacks on these providers.

Table 1: Key Metrics of the HTTP/2 Rapid Reset Attacks on Major Providers

Company	Peak Requests Per Second (RPS)	Reported Botnet Size	Attack Vector	Date of Attack
Google	398 million	Not specified	HTTP/2 Rapid Reset	October 2023
Cloudflare	201 million	~20,000 machines	HTTP/2 Rapid Reset	August 2023
AWS	155 million	Not specified	HTTP/2 Rapid Reset	August 2023

The operational impact of these attacks, while successfully mitigated, was significant. The attacks were designed to cause "downtime" [6] and were so effective they "overloaded some components" in Cloudflare's network, forcing the company to develop "purpose-built technology" to mitigate them.[2] The sheer resource consumption required for the server to process and respond to the unending stream of requests, even when canceled, highlighted the inherent vulnerability in HTTP/2's implementation. This demonstrated that the threat was not merely about clogging a network pipe but about exploiting a fundamental design flaw to cause catastrophic resource exhaustion at the application layer.

4. Mitigation and Defensive Strategies

4.1 The Challenge of Mitigation

The HTTP/2 Rapid Reset attack exposed a critical weakness in existing cybersecurity defenses. Traditional DDoS protection, which relies on measures like rate limiting or simple blocklists to defend against volumetric flood attacks, proved ineffective.[6] The attack’s traffic did not appear as a flood of disconnected packets from a multitude of sources but rather as legitimate-looking, stateful connections. Since the attacker's goal was not to saturate network bandwidth but to exhaust CPU and memory resources by forcing the server to process an infinite number of stream-cancellation operations, these conventional defenses were rendered obsolete.[9] This incident served as a stark reminder that a defense model focused solely on volume and network capacity is no longer sufficient; a new, intelligent approach is required.

4.2 Technical Countermeasures

The response to the HTTP/2 Rapid Reset vulnerability (CVE-2023-44487) necessitated a multi-pronged defensive strategy. The most immediate and critical step was the development and deployment of **patches** for all vulnerable implementations of the HTTP/2 protocol.[3] This included a wide array of server software and libraries, such as Netty, Jetty, NGINX, and Apache Tomcat.[3]

Beyond patching, specialized mitigations were developed to address the unique nature of the exploit:

- **Rate Limiting on RST_STREAM Frames:** Instead of limiting overall requests per second, a more granular approach was required. Defenders implemented rate-limiting specifically on the number of RST_STREAM frames per connection or IP address to detect and stop the malicious "request, cancel" loop.[10]
- **Limiting Concurrent Streams:** Network providers and security vendors reinforced limits on the maximum number of concurrent streams allowed per connection to prevent the attacker from initiating an overwhelming number of requests.[10]
- **Behavioral Detection and Anomaly Monitoring:** The most advanced defenses moved beyond simple rule-based blocking to behavioral analysis. This involved detecting unusual "stream churn"—the rapid opening and closing of streams—and flagging connections that exhibit a high volume of protocol violations, which are often precursors to resource-exhaustion exploits.[10]
- **Client Downgrade:** As a last resort, systems were configured to downgrade clients exhibiting suspicious behavior from the efficient HTTP/2 protocol to the less efficient HTTP/1.1 protocol, effectively mitigating the amplification factor of the attack.[10]

The table below provides a practical mapping of the attack's technical vectors to the corresponding defensive strategies.

Table 2: Defensive Strategies vs. Attack Vectors

Attack Vector	Weakness Exploited	Corresponding Mitigation Strategy
Client-side RST_STREAM flooding	Protocol's stream cancellation mechanism	Rate-limit RST_STREAM frames per connection/IP
Resource exhaustion via "churn"	Server's computational effort in starting and discarding streams	Limit maximum concurrent streams per connection; implement behavioral detection of "churn"
Server-side forced resets ("MadeYouReset")	Protocol's mandatory error-handling (PROTOCOL_ERROR triggers RST_STREAM)	Implement stricter protocol validation checks; monitor for high volumes of protocol violations

4.3 Strategic and Collaborative Recommendations

The success of the HTTP/2 Rapid Reset attacks, and the subsequent countermeasures, highlighted the necessity of a multi-layered, holistic approach to cybersecurity.[6] Reliance on a single defensive measure is no longer viable. Instead, organizations must integrate network-layer DDoS protection with sophisticated application-layer security. Perhaps the most critical lesson was the importance of

industry-wide collaboration. The responsible disclosure and joint mitigation efforts by major players like Google, Cloudflare, and AWS set a new standard for a coordinated response to zero-day vulnerabilities that threaten the entire digital ecosystem.[4] This model of shared intelligence and cooperative defense is no longer a luxury but a fundamental requirement for securing an increasingly interconnected world.

5. Conclusion: Lessons Learned and Future Outlook

5.1 A New Era of Protocol-Level Attacks

The HTTP/2 Rapid Reset attack of late 2023 was a landmark event that signals a new era in cyber threats. It unequivocally proved that a high-impact DDoS attack does not require a massive botnet or overwhelming traffic volume. Instead, by exploiting a zero-day vulnerability in a core internet protocol, a modestly-sized botnet can achieve record-breaking levels of requests per second and threaten the operational stability of a significant portion of the internet. The primary threat vector has pivoted from volumetric saturation to intelligent resource exhaustion at the application layer, forcing a fundamental rethink of DDoS defense strategies. The new benchmark for a successful attack is efficiency, not sheer size.

5.2 Future Threat Landscape

The discovery and subsequent evolution of the HTTP/2 Rapid Reset vulnerability serve as a powerful harbinger of future threats. It is highly probable that threat actors are already actively investigating other internet protocols for similar vulnerabilities. The rapid development of the "MadeYouReset" variant demonstrates the attackers' ability to quickly adapt their tactics in response to new defenses, creating a continuous and challenging arms race. The cybersecurity community should anticipate a growing number of sophisticated, low-volume but high-impact attacks that exploit subtle flaws in common protocols.

5.3 Final Recommendations

In light of the HTTP/2 Rapid Reset attacks, the following recommendations are critical for hardening digital infrastructure:

- **Proactive Vulnerability Management:** Organizations must prioritize patching and security updates for all server implementations, frameworks, and libraries that handle internet protocols.
- **Invest in Behavioral Security:** Defense must move beyond simple traffic analysis. Investment in advanced behavioral monitoring and anomaly detection is essential to identify and mitigate attacks that exploit protocol logic or trigger subtle, server-side resource exhaustion.
- **Embrace Multi-Layered Defense:** A comprehensive security posture must combine network-level defenses with robust application-layer security, including specialized DDoS protection that can analyze and respond to threats at the protocol and application levels.

- **Foster Collaborative Intelligence:** The collaborative response from Google, Cloudflare, and AWS provides a blueprint for the future. Industry-wide information sharing and responsible disclosure must become the standard operating procedure to collectively address systemic cyber risks and preemptively fortify the internet against future, similar threats.

Works cited

1. Top +35 DDoS Statistics (2025) - StationX, accessed September 21, 2025, https://www.stationx.net/ddos-statistics/
2. DDoS threat report for 2023 Q4 - The Cloudflare Blog, accessed September 21, 2025, https://blog.cloudflare.com/ddos-threat-report-2023-q4/
3. Find and fix HTTP/2 rapid reset zero-day vulnerability CVE-2023-44487 | Snyk, accessed September 21, 2025, https://snyk.io/blog/find-fix-http-2-rapid-reset-zero-day-vulnerability-cve-2023-44487/
4. HTTP/2 Zero-Day vulnerability results in record-breaking DDoS attacks - The Cloudflare Blog, accessed September 21, 2025, https://blog.cloudflare.com/zero-day-rapid-reset-http2-record-breaking-ddos-attack/
5. Big Tech firms reveal record-breaking DDoS attacks - Silicon Republic, accessed September 21, 2025, https://www.siliconrepublic.com/enterprise/big-tech-ddos-attacks-record-google-amazon-cloudflare
6. Five Most Famous DDoS Attacks and Then Some | A10 Networks, accessed September 21, 2025, https://www.a10networks.com/blog/5-most-famous-ddos-attacks/
7. Famous DDoS attacks | Biggest DDoS attacks | Cloudflare, accessed September 21, 2025, https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/
8. MadeYouReset: An HTTP/2 vulnerability thwarted by Rapid Reset mitigations, accessed September 21, 2025, https://blog.cloudflare.com/madeyoureset-an-http-2-vulnerability-thwarted-by-rapid-reset-mitigations/
9. How it works: The novel HTTP/2 'Rapid Reset' DDoS attack | Google Cloud Blog, accessed September 21, 2025, https://cloud.google.com/blog/products/identity-security/how-it-works-the-novel-http2-rapid-reset-ddos-attack
10. MadeYouReset: Turning HTTP/2 Server Against Itself - Imperva, accessed September 21, 2025, https://www.imperva.com/blog/madeyoureset-turning-http-2-server-against-itself/