IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 19 January 2024, accepted 20 February 2024, date of publication 27 February 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371013

TOPICAL REVIEW

Distributed Denial of Service Attack in HTTP/2: Review on Security Issues and Future Challenges

LIANG MING[1,2], YU-BENG LEAU[2], (Senior Member, IEEE), AND YING XIE[3]
[1]Office of Information Technology, Chongqing University of Arts and Sciences, Chongqing 402160, China
[2]Cybersecurity Research Laboratory, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu 88400, Malaysia
[3]The First Experimental Primary School, Yangtze Normal University, Chongqing 400030, China

Corresponding author: Yu-Beng Leau (lybeng@ums.edu.my)

The work was supported in part by the National Natural Science Foundation of China under Grant U22A20102 and in part by the key Science and Technology Research Program of Chongqing Municipal Education Commission, Ching under Grant KJZD-K202101305.

ABSTRACT This article offers a comprehensive overview of recent literature on the HTTP/2 protocol and conducts an analysis of the security threats and DDoS attack typologies associated with HTTP/2. The investigation revealed that the introduction of new features in HTTP/2 has significantly improved the network transmission speed and utilization. However, these advancements have also brought forth a series of emerging network security risks. This study examines the current state of the art in DDoS attacks tailored for HTTP/2 and their detection methods, proposing future research directions in the field of attack detection. By analyzing the distinctive features of HTTP/2 protocol, the study suggests extending DDoS attack detection techniques established for HTTP/1 to the realm of HTTP/2. Furthermore, the research underscores the ease with which adversaries can exploit the intrinsic multiplexing in HTTP/2 to launch a large number of malicious requests, leading to severe depletion of network bandwidth and exhaustion of valuable server resources. Additionally, it highlights the potential applicability of deep learning algorithms in the context of the HTTP/2 protocol. Additionally, the article proposes strategies to address challenges associated with DDoS attacks and the scarcity of adequate datasets for HTTP/2. This research contributes to a comprehensive understanding of the security implications surrounding the HTTP/2 protocol and provides valuable insights for advancing DDoS attack detection technologies.

INDEX TERMS HTTP, HTTP/2, DDoS, deep-learning, machine-learning.

I. INTRODUCTION

A. BACKGROUND

The widespread integration and advancement of informationization have resulted in a substantial dependence of businesses, governments, and society on it to fulfill their information retrieval requirements. Simultaneously, the importance of network communications is steadily escalating. However, as the volume of network traffic continues to surge, ensuring the availability of diverse communication standards becomes a challenging task. This underscores the necessity for continuous adaptation and enhancement of network security and communication standards to meet the escalating communication demands while upholding the reliability and security of networks. This becomes especially crucial for computer network security researchers who are engaged in addressing evolving network threats and attacks.

The inception of the World Wide Web (WWW) in the early 1990s aimed to facilitate users in accessing information consistently and simply from any source [1]. With the increasing richness and size of communication payload, the concern of network access latency has become notably prominent, particularly in mobile networks where data request and response times may extend up to 30 milliseconds,

The associate editor coordinating the review of this manuscript and approving it for publication was Valentina E. Balas.

despite the current capability of mobile devices to achieve link speeds exceeding 1 Gbit/s [2]. The widely accepted HTTP/1.1 protocol, which has served as the standard for the WWW, is no longer capable of meeting the demands of the present situation. In response to these challenges, significant efforts have led to the emergence of the innovative HTTP/2 protocol.

HTTP/2 has gained widespread adoption as a protocol on numerous servers across the Internet. However, subsequent to its release, a multitude of vulnerabilities associated with this protocol have been disclosed by various security researchers [2]. These vulnerabilities can be exploited to execute diverse types of Denial of Service (DoS) attacks. Consequently, to gain a deeper understanding of the security risks inherent in the implementation of the new version of the HTTP protocol, a systematic review of existing research has been conducted. The objective of this article is to underscore the necessity of addressing the implementation shortcomings of HTTP/2 and explore ways to overcome them within the context of current technological limitations.

This paper initially showcases the enhancements in the HTTP/2 protocol by conducting a comparative analysis with the HTTP/1 protocol. Subsequently, the compiled literature identifies various DDoS attacks that have been validated, utilizing the improvements introduced in the HTTP/2-based protocol. The available detection methods for these HTTP/2-based DDoS attacks are then delineated. Finally, the paper summarizes the limitations of current DDoS attack detection techniques for the HTTP/2 protocol and points towards future research directions in this domain.

B. CONTRIBUTIONS

In initiating our literature review, searches were conducted using the keywords "HTTP/2" and "DDoS" across well-established research databases. The selected databases encompassed Google Scholar, IEEE Xplore, Scopus, SpringerLink, ScienceDirect, and ACM Digital Library. A comprehensive search yielded 104 results, from which literature published before 2017 was excluded, resulting in a refined count of 64 pertinent articles. finally, Focused on the themes of DDoS attacks, the HTTP/2 protocol, and related attack detection techniques, a meticulous filtering process led to a curated collection of 28 articles for in-depth analysis. These articles were carefully studied and analysed from three main angles, i.e.

1). The DDoS attack methods targeting the HTTP/2 protocol were organized by their discovery time, and for each attack, the protocol vulnerabilities that were being exploited were categorized and the resulting impacts were summarized. The aim is to understand the currently identified DDoS attacks based on HTTP/2.
2). Based on the collected literature, a summary was provided for existing detection techniques for DDoS attacks with reference to the HTTP/2 protocol. Additionally, through the utilization of a matrix, mapping was conducted for DDoS attacks based on HTTP/2 and the corresponding applicable detection techniques.
3). By identifying the gaps and the real underlying reasons, this paper proposes future research directions for enhancing existing attack detection techniques in the context of HTTP/2.

It is worth emphasizing that although there exist numerous studies and progress made based just on HTTP/1.1 attacks and detection mechanisms, it is unable to fully cater to its successor, HTTP/2. Due to the fact that HTTP/2 is a more recent standard [2] with fundamentally different operations compared to HTTP/1.1 [6], [7].

II. THE IMPROVEMENTS IN HTTP/2

HTTP/2, represents the next generation of the HTTP web communication protocol. HTTP/2 was officially published in 2015. The motivation behind the new version of the HTTP protocol was primarily to address the various limitations and shortcomings identified in HTTP/1 (including HTTP/1.1).

A. THE HTTP/1 DILEMMA

In the early days, HTTP/1.1 was widely adopted as the communication protocol for almost all web content on the Internet. Its full name is Hypertext Transfer Protocol (HTTP), with the version number (1.1) appended. This protocol operates over the TCP protocol and follows a simple request-response-based mechanism. HTTP/1.1 defines the types of messages that a client (web browser) can send to a server and the format of the response data it can receive. In the case of unencrypted communication, client requests and server responses are transmitted in the form of ASCII codes. However, as applications built on this protocol evolved and the demand for faster network responses increased, this made the HTTP/1.1 outdated as it no longer was able to meet the expectations of the current needs and time. The most commonly known deficiency of HTTP/1 were:

- **Too many options in the protocol:** HTTP/1.1 not only contains a multitude of details but also includes numerous options reserved for future scalability. As time progressed, these seemingly superfluous functionalities have been brought into use but presents interoperability issues between clients and servers, e.g. HTTP Pipelining.
- **TCP performance is not fully utilized**: HTTP/1.1 fails to optimize the utilization of the TCP protocol, compelling HTTP clients (browsers) to seek alternative solutions for minimizing page load times. In essence, the underutilization of the capabilities of TCP results in interruptions during the transmission process. Improving this aspect by maximizing the potential of TCP would directly translate to enhanced performance times for both data sending and receiving. This optimization aims to streamline the communication process, mitigating interruptions and fostering more efficient data exchange between clients and servers.

- **Gradual increase in data size and resource count in transmission**: Through a meticulous examination of the initiation process of leading websites, a conspicuous trend emerges [3]. Over recent years, there has been a steady rise in the data volume required to load website pages, with certain pages surpassing the 2MB threshold. On average, each webpage mandates the download of over 110 individual resources to facilitate proper rendering and display [3]. This incremental surge in data consumption bears noteworthy implications for network access latency and the overall performance of the web. The growing demand for resources during page loading contributes to heightened latency and impacts the efficiency of web interactions.
- **The dreadful transmission latency**: Despite the substantial expansion of network bandwidth in recent years, there has not been a commensurate reduction in network latency. The proliferation of media content underscores the critical importance of low-latency communication to guarantee seamless and uninterrupted transmission. This is particularly crucial in the context of video services, encompassing applications, e.g. video conferencing, gaming, and services reliant on real-time data streaming. The necessity for low latency becomes increasingly apparent as the demand for swift and responsive communication in these domains continues to grow.
- **Unavoidable Head-of-line blocking:**HTTP/1.1 introduced Pipelining as a technique that is supposed to improve the efficiency of the request-response transmission process. It allows uninterrupted transmission of requests while waiting for the response of the current request. Pipelining was introduced to alleviate the issue and improve server resource utilization. However, it is prone to head-of-line blocking, increasing server memory overhead and potentially causing duplicate requests. It has also several other issues, often leading to it being disabled by default for most users. For instance, in the case of time-consuming requests, it is possible to opt for establishing a new TCP connection for subsequent requests.

B. THE HTTP/2 BENEFITS COMPARISON

In May 2015, the HTTP/2 specification was formally standardized as a response to Google's SPDY protocol, with the aim of ensuring compatibility with the HTTP protocol. In contrast, HTTP/1 primarily manages requests through two distinct approaches: serial processing, where a request is processed before sending another, and concurrent processing, wherein the return process of the request result must follow a sequential order, necessitating the processing of the first response before subsequent requests can be handled. This procedural arrangement often leads to a phenomenon known as blocking. In contrast, HTTP/2 exhibits the capability for concurrency processing, allowing the simultaneous handling of multiple requests. Once response data is generated, it can promptly be returned without strict adherence to a specific order. This advancement enhances overall efficiency and furnishes the critical capability to manage multi-threaded and parallel requests more effectively. The optimization process in HTTP/2 protocol is illustrated in Figure 1.

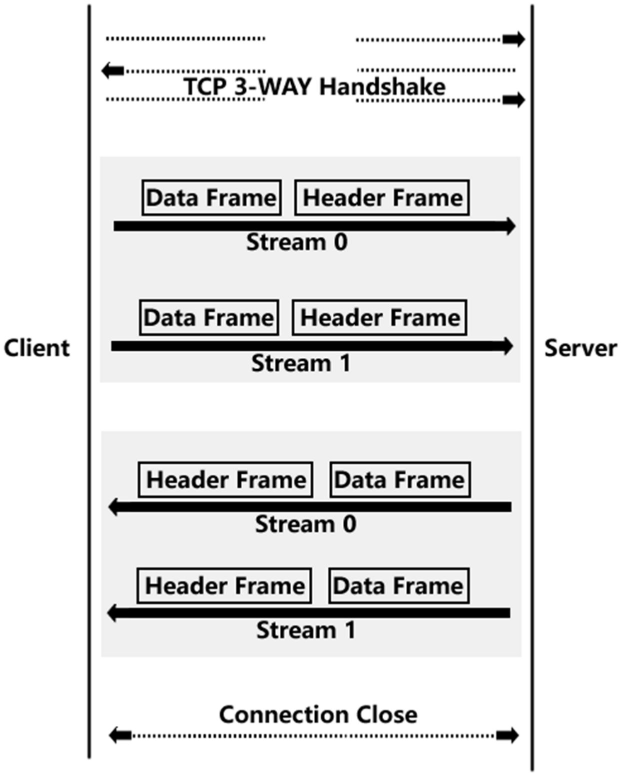

FIGURE 1. The HTTP/2 process of data exchange.

Similar to HTTP/1, HTTP/2 also employs the standard TCP three-way handshake during connection setup. Additionally, HTTP/2 introduces the concept of ''stream'', which in essence represents a bidirectional sequence of binary frames. Within a single message, back-and-forth frames are assigned a unique stream ID. Conceptually, a ''stream'' can be envisioned as a virtual ''data flow'' where ordered data frames flow sequentially. The advantage is that when these data frames are received, they are organized in sequence, forming the request and response messages similar to those in HTTP/1.

This ''stream'' is evidently a virtual concept and does not have a physical existence. As a result, HTTP/2 can utilize streams to simultaneously send multiple fragmented messages over a single TCP connection, a feature commonly referred to as ''multiplexing''. This multiplexing capability allows multiple round-trip communications to be handled on the same connection, thereby mitigating the issue of ''head-of-line blocking'' and significantly reducing latency. Consequently, the utilization of ''streams'' at the message level ensures an orderly sequence of frames, while at the connection level, messages are received and sent in a non-sequential manner, enhancing connection efficiency and overall performance.

In summary, in comparison to the HTTP/1 protocol, HTTP/2 has primarily achieved improvements and optimizations especially in the following ways:

- HTTP/2 employs a binary format for data transmission, diverging from the textual format employed by HTTP/1.1. The adoption of binary format offers additional advantages and potentials in terms of protocol parsing and optimization extension.
- HTTP/2 utilizes HPACK for header compression when transmitting message headers. This greatly reduced the network traffic load. In contrast, HTTP/1.1 includes redundant header information in each request, resulting in wastage of bandwidth resources.
- HTTP/2 uses multiplexing for data frame transmission, that is, multiple requests are completed concurrently through a TCP connection. Although HTTP/1.1 applied pipeline technology to achieve efficiency, it still suffered from blocking. In this respect, HTTP/2 is much more efficient as it supports true concurrent requests, and at the same time, the streaming process supports prioritization and flow control.
- HTTP/2 Server Push technology enables the server to proactively push resources to the client, such as through JavaScript (JS) and Cascading Style Sheets (CSS) files, without waiting for the client to request for them. This aspect optimizes the delivery of resources by eliminating the need for the client to parse HTML and then send separate requests for these resources. By pushing the resources to the client in advance, they are made readily available when the client needs them. This not only reduced the traffic load but also eliminate any bottlenecks at the server.
- Simplify Web application development. In comparison to HTTP/1, the adoption of HTTP/2 provides a significant reduction in the workload for web developers by eliminating a substantial portion of the optimization efforts required to enhance data transfer.
- Romuald Corbel et al. [8] compared HTTP/1 and HTTP/2 in terms of functionality and page download time. By taking all precautions in terms of set up to eliminate biasness, i.e. in terms of the hardware, software, and the transmission conditions, they applied a measurement by using multiple transmissions over a single TCP connection per domain technique and found that HTTP/2 consistently fared better than HTTP/1 in terms of performance. More precisely, as the network latency increases, the page download time (PDT) of HTTP/2 remains 15 % less than that of HTTP/1. Furthermore, they observed that the ratio of HTTP/2's page download time to HTTP/1's page download time decreases with increasing packet loss, suggesting that HTTP/2 is more superior and resilient to packet loss than HTTP/1. The results showed great promises that HTTP/2 is much more ideal for data streaming where current content tend to be heavier and sequentially important, e.g. in the case of web-video data transferring [9].

III. REVIEW OF EXISTING DDoS ATTACK ON HTTP/2

With the formal release of the HTTP/2 protocol, many software vendors have upgraded their applications to support HTTP/2 in recent years, resulting in significant performance improvements. However, recent research has also revealed certain vulnerabilities associated with the HTTP/2 protocol that can be exploited by malicious cyber-attacks. Table 1 provided a summary of those research which present significant insights into some of the most prominent and concerning weaknesses found in the HTTP/2.

Despite the gain of improved web performance, HTTP/2 has been criticized for using lower overhead requests to force servers to perform more computation and be more vulnerable to DDoS attacks.

Praseed et al. [22] have shown that the multiplexing technique of HTTP/2 may be easily exploited by attackers to launch DoS attack by initiating multiple requests for web resources using a single TCP connection. This enables malicious users to initiate DDoS attacks with ease. In their research, a comprehensive analysis was conducted to assess the performance of HTTP/2 servers and HTTP/1 servers under asymmetric DDoS attacks, under the same workloads. Under HTTP/2 protocol condition, they applied a new DDoS attack vector called multiplexed asymmetric DDoS attack and demonstrated that this type of attack can be initiated with just a small number of clients and causes server crashes. Additionally, it was demonstrated that asymmetric workloads on servers with server push enabled could easily lead to a flood attack at the network layer.

In fact, Beckett et al.'s [6] research validated the susceptibility of HTTP/2 to flood DDoS attacks. To better illustrate this issue, they compared the effectiveness of request flooding DDoS attacks using HTTP/2 with the previous HTTP/1.1 standard. The results indicated that, as HTTP/2 allowed multiple requests to be included in a single packet, which provided a significant advantage to attackers, resulting in an enhanced impact of the attack effectiveness.

In order to show the vulnerability and amplification risks which HTTP/2 may be exposed to, Beckett et al. [7] implemented an HTTP/2 testbed to measure bandwidth by collecting traffic in a private cloud environment. They showed how one may exploit HTTP/2's header compression, HPACK to generate an amplified packet payload to cause flooding at the back-end data link.

As 5G is being rolled out globally, HTTP/2 protocol is most likely to be used in all the Service Based Interfaces (SBI) of 5G [9]. Hu et al. [10] discussed several different kinds of attacks based on HTTP/2 that may be implemented in 5GS including stream reuse attack, flow control DoS, dependency cycle DoS, HPACK Bomb, Man-in-the-Middle

TABLE 1. Types of HTTP/2 DDoS attacks & its weaknesses.

Paper	Attack Method	Exploited	Attack Impact	Year
[11]	A total of 5 cases were tested, and for each case a specific HTTP/2 packet was crafted, then for each case this was repeatedly sent 30 times. For each case, the packet generator sends a test case of a forged HTTP/2 frame packet against the server.	HTTP/2 WINDOW_UPDATE Frame	The experimental results indicated that with the introduction of HTTP/2, it is possible to launch a DoS attack against a web server by sending slow-rate HTTP/2 data packets from a single client. The HTTP/2 protocol requires more computational resources compared to its predecessor, HTTP/1.1. Consequently, HTTP/2 is more susceptible to the impact of Denial of Service (DoS) attacks.	2015
[12]	This research modeled 2 types of attacks whose traffic continually consumed the victim's computing resources, and caused the machine learning techniques to incorrectly classify some traffic instances thereby yielding false alarms.	1)The Size of Packet Carrying the RSK-ACK Flag 2)WINDOW_UPDATE packets	This research demonstrated that it is possible to initiate DoS attacks by intelligently distributing them, allowing them to remain covert over an extended period. Such stealthy traffic can have a detrimental effect on the performance of machine learning classifiers, resulting in a higher percentage of false positives.	2017
[7]	This research formed a testing platform by connecting to an HTTP/1.1 backend web server through an HTTP/2 gateway proxy. Two commonly used HTTP/2 proxies, Nginx and nghttp2, were employed. The attack tool utilized was *h2load*. They captured traffic using the tcpdump tool, measuring packets containing payload data to mitigate the impact of TCP ACK frequency deployment choices. Finally, they measured the byte size of the packets, including Ethernet, IPv4, TCP headers, and HTTP payload. This experiment aimed to investigate the performance of HTTP/2 when facing DDoS attacks.	The Decompression Of HTTP/2 Request Headers (HPACK)	This study demonstrated the vulnerability in an environment transitioning from HTTP/2 to HTTP/1. This allowed a skilled attacker to leverage HTTP/2 header compression (HPACK) to generate large data packet payloads transmitted over the backend data link. Additionally, the server's maximum concurrency settings influence the attacker's amplification capabilities. On the other hand, applications that served as HTTP proxies supporting HTTP/2 can be utilized to amplify HTTP/2 traffic to any HTTP/1.1 website.	2017
[6]	Using the created attack tool, they initially launched repeated attacks on the web server using HTTP/1.1, with 800 concurrent connections. Subsequently, the attack was adapted to utilize the HTTP/2 protocol, exploiting its multiplexing capability. Different maximum concurrency values were tested in the attack simulation to assess their DDoS attacks impact risks.	HTTP/2 Multiplexing	For HTTP DoS attacks, it has been shown that the bottleneck for attackers lies mostly in packet generation. However, this research revealed that in HTTP/2 scenarios, the risk of DDoS attacks is greater than the threat posed by the previous version, HTTP/1, due to the fact that HTTP/2 allowed multiple requests per packet. This gives attackers a significant advantage and leads to increased effectiveness of the attacks.	2017
[16]	This research evaluated slow DoS attack named zAttack against the HTTP/2 protocol which was launched by injecting specially crafted HTTP requests through: 1) Payload from Client to Server 2) Payload from Server to Client 3) Payload from Client to Server	1) HTTP/2 SETTINGS_MAX_CONCURRENT_STREAMS of the SETTINGS Frame 2) HTTP/2 PUSH_PROMISE Frame 3) HTTP/2 RST_STREAM frame	In this experiment, attacks were initiated by sending specially designed HTTP/2 payloads to web servers. The results revealed that the majority of servers were susceptible to these attacks, whether initiated with plain text or encrypted HTTP/2 requests. Through a comparative study of slow DoS attacks on HTTP/1.1 and HTTP/2, it was concluded that HTTP/2 presents a greater number of threat vectors compared to HTTP/1.1.	2018

TABLE 1. ***(Continued.)*** **Types of HTTP/2 DDoS attacks & its weaknesses.**

[10]	This paper conducted a comprehensive study of the use of HTTP/2 in the 5G core network and revealed potential vulnerabilities and security issues.	1) Stream Reuse Attack 2) Flow control DoS 3) Dependency cycle DoS 4) HPACK Bomb 5) Man-in-the-Middle Attack (MITM) 6) Interconnect Attacks	Based on the analysis, there is a medium to high level of security risk existing in 5G network. This is because 5G uses common "Internet" protocols like HTTP/2, JSON, and REST API, which means grace period between vulnerability discovery and real exploitation will become much shorter compared to SS7 and Diameter.	2018
[22]	The authors conducted tests on three e-commerce web applications (Opencart, Magento, and Prestashop) using their proposed attack models. The tests included : 1) Performance Comparison of HTTP/1.1 and HTTP/2 Under a Simple Asymmetric DDoS Attack. 2) Analyzing the Performance of an HTTP/2 Server Under a Multiplexed Asymmetric Attack. 3) Analyzing the Impact of Server Push on an HTTP/2 Server Under a Multiplexed Asymmetric DDoS Attack.	1) Server Push 2) Multiplexing	In this study, the authors demonstrated that, for the same quantity and type of requests, HTTP/2 servers actually outperform servers running HTTP/1.1. Despite the performance improvement, HTTP/2 servers are more susceptible to Multiplexing Asymmetric Attacks. Moreover, if server push is enabled on HTTP/2 servers, utilizing Multiplexing Asymmetric Attacks can lead to network layer egress flooding attacks.	2019
[15]	Implemented an attack generator which contained a total of 11 slow DoS attacks: Slowloris, Slow POST, Slow Read, SlowDrop, Slow Next, Slowcomm, and a group of attacks focusing on the HTTP/2 protocol — Slow Read, Slow POST, Slow Preface, Slow Headers, and SlowSettings. The aim of this research was to test the vulnerability of popular HTTP2 web servers to these attacks.	GET or POST Request	The author's tests confirmed the effectiveness of the crafted slow DoS attack generator, successfully triggering a DoS attack on most web servers. The attack results on different web servers were as follows: 1) Microsoft IIS demonstrated the highest level of resistance to these attacks. 2) Nginx 1.14.0 could withstand all attacks without experiencing a denial of service, except for Slow Read and Slowcomm attacks. 3) Lighttpd 1.4.55 and Apache 2.4.17, on the other hand, lacked sufficient security against slow DoS attacks, making them susceptible to all the tested attack scenarios.	2021
[14]	This research generate 5 types of incomplete HTTP/2 request packets: 1) Incomplete GET/POST Request Header. 2) Sending Connection Preface Only. 3) Incomplete POST Request Message Body. 4) Advertising Zero Window Size. 5) Unacknowledged SETTINGS frame. These attacks were then sent to the web servers one after another.	1)HTTP/2 SETTINGS Frame 2)HTTP/2 WINDOW_UPDATE Frame 3)HTTP/2 HEADERS Frame 4)HTTP/2 CONTINUATION Frame	In this study, the authors initially conducted an empirical assessment of slow HTTP/2 DoS attacks on the internet. Subsequently, they proposed a real-time detection solution for these attacks. The experimental results revealed that multiple HTTP/2 servers on the internet experience delays when closing connections, particularly for connections that transmit incomplete requests. This behavior makes internet web servers susceptible to the impact of slow HTTP/2 DoS attacks.	2022
[17]	They conducted HTA (HTTP/2 Traffic Amplification) and HSR (HTTP/2 Slow Rate) attack experiments on 10 popular CDNs to assess their feasibility and impact in the real world.	1) HTTP/2 Traffic Amplification (HTA) attack 2) HTTP/2 Slow Rate (HSR) attack.	The experimental results of this study demonstrated that all of these CDN services were susceptible to HTA attacks, with four of them being particularly vulnerable to HSR attacks. In the worst-case scenario, attackers can amplify network traffic by a factor of 403,092, posing a significant DoS threat to CDN services.	2022

TABLE 2. Contemporary research in HTTP/2 DDoS detection methods.

Paper	Contribution For Detection	Limitation	Types of Dataset	Year
[18]	A lightweight machine learning-based method has been provided for classifying encrypted network traffic into their respective HTTP/1 and HTTP/2 versions. This approach is simple, scalable, and suitable for scenarios where deep packet inspection (DPI) is not feasible. This work is a step forward in the direction of understanding modern web traffic. It provides a methodology to identify HTTP/2 traffic in flow traces where no information about application layer protocols is available. It will improve visibility into network traffic and help in analyzing the adoption of HTTP/2 from passive traces.	C4.5 algorithm can be used in high speed online classification systems while Random Forest is a better choice for offline systems as it provides slightly higher accuracy.	1) Campus Dataset from border routers of a European university. 2) Residential Dataset: from a Point of Presence (PoP) of a European ISP.	2016
[12]	Applied four machine learning techniques, i.e. Naïve Bayes (NB), Decision Tree J48(DT), JRip and Support Vector Machines (SVM) to classify the attack traffic.	In the methods adopted, attackers may bypass the detection of the machine learning models by adopting adversarial tactics such as using stealthy attack techniques or deceiving the models. That is to say, the robustness of the models is challenged in the face of advanced and targeted attacks.	Self-Build Database Set	2017
[20]	This research applied anomaly detection scheme which uses chi-square test to differentiate between normal and attack traffic profiles. It showed that this detection scheme could detect the attacks with high accuracy.	Although the literature proposed a detection method for the Slow Rate DoS attack, it did not fully consider other possible attack types and threat vectors. Other types of attacks may require different detection methods and strategies.	1) Training Dataset: collected 14 hours of normal HTTP/2 traffic from the web server configured in their testbed. 2) Testing Dataset: Generated another 14 hours of normal traffic from the same setup.	2018
[25]	1) Define a test-of-time and test-of-space methodology for HTTPS classification including the specification of a crawling campaign to collect relevant datasets. 2) Evaluate this H2 classifier over around four months and the impact of a regular re-training. 3) Evaluate this H2 classifier on more than 3000 websites	This method needs regular updates, due to the constant changes in Internet content and server software updates. This is because the HTTP/2 Classifier method needs to be retrained periodically to adapt to new traffic patterns and features. In this study, performing weekly training updates has proven to be effective, but the frequency of updates may vary for different websites and application scenarios.	1)Temporal Dataset: crawled 500 key-words of four major services (Amazon, Instagram, Google and Google Images). 2) Service-wide Dataset: The dataset was constructed by requesting data from various popular websites. Starting with an initial list of popular websites in the United States, non-HTTP/2 websites were excluded from consideration.	2020
[23]	In this work, HTTP/2 request sets have been modelled as fuzzy multisets, and a trust scoring mechanism has been developed that helps in distinguishing between valid and invalid request sets. Experiments on public datasets demonstrate that the proposed mechanism is able to detect multiplexed asymmetric attacks with an accuracy of around 95%, while maintaining a false positive rate (FPR) of around 3%.	The dataset used for the study was limited in terms of size and the specificity of the data source. Hence, leading to a less comprehensive and accurate modeling and analysis of attack traffic. Although the method showed high accuracy in detecting multiplexed application layer DDoS attacks, there remains about 3% of false positive rate which labelled some legitimate traffic as attack traffic, thus, causing unnecessary disruptions to use As network traffic grows and scales, this method will have to confront the challenge of handling a multitude of requests and analyzing complex data. The scalability of the method may be limited and restricted especially in large-scale network environments.	HTTP/2 logs were generated by publicly available HTTP/1.1 traces from the SDSC and CLARKNET websites.	2021

TABLE 2. *(Continued.)* **Contemporary research in HTTP/2 DDoS detection methods.**

[24]	1) Generating and capturing slow rate attacks traffic. 2) Developing four one class classifier algorithms to classify slow rate DoS attacks or normal based on the input dataset. 3) Applying those one class models to the collected traffic data. 4) Compared the results of proposed algorithms with an existing related models. It was shown that those one classifier algorithms outperforms than other algorithms.	This research may have excluded other algorithms which may be actually more superior and perform better across various other attack types.	Self-Build Database Set	2023
[26]	1) Test the behaviour of HTTP/2 supporting web servers on the Internet against different Slow HTTP/2 DoS attacks and show that several are vulnerable to attacks. 2) Propose an event sequence analysis-based detection scheme to detect Slow HTTP/2 DoS attacks in real-time. 3) Test the detection performance of the proposed scheme in a real network and show that it can detect attacks with very high accuracy and marginal computational overhead. 4) Compare the detection performance of the proposed scheme with the previously known defence mechanisms to counter Slow HTTP/2 DoS attacks and show that it outperforms the previously known defence mechanisms.	This solution is highly focused only on Slow HTTP/2 DoS attacks and may not be able to adapt to new attack patterns or variants.	Self-Build Database Set	2023

Attack (MITM) and interconnect attacks. Based on the research analysis, there is a medium to high level of risk in the signaling security domain. It is paramount importance to be aware of these issues in order to prevent unwanted malicious attacks from being launched through them.

The HTTP/2 protocol has introduced flow control to regulate numerous streams within a single connection. This allowed servers to reduce traffic on one stream, enabling them to continue processing other streams within the same connection. Flow control for the HTTP/2 protocol can be accomplished through the WINDOW_UPDATE frame. Changing the value of the WINDOW_UPDATE frame causes the HTTP/2 device to spend a lot of time processing it, which ultimately leads to very high CPU utilization on the recipient machine. This problem is not present in HTTP/1 [11]. The attacker can exploit this vulnerability to launch an attack.

In attempt to further test the security of HTTP/2, Adi et al. [12] proposed an HTTP/2 stealth attack model by intelligently distributing network traffic load associated with a DoS attack that remain stealthy, and can be used for a long period of time. This covert attack is able to bypass intrusion detection systems and even affect the performance of the machine learning classifier, by resulting in an extremely high percentage of False Alarms. Additionally, to illustrate how covert attack traffic is classified from legitimate traffic, the researchers proposed a legitimate traffic model based on HTTP/2. This model is utilized to generate flash-crowd traffic.

It is also because the request header is not mandatory to be given, which allowed network intruders to generate the forged headers as legitimate HTTP requests. Jaafar et al. [13] also demonstrated this weaknesses by using used eight different types of attacks in which the results showed that, there are similarities and differences in the fake request headers in both internal and external networks, thus, this can make HTTP requests appear authentic when launching HTTP DDoS attacks. In fact, it was highlighted that the request header needs a major improvement in terms of security to prevent the request header from being easily manipulated by malicious attackers.

In 2022, Tripathi et al. [14] measured the behavior of slow-rate DoS attacks in the top 500K Alexa ranked websites on the Internet which provided HTTP/2 services, and observed that a large number of them were vulnerable to these attacks.

Sicora et al. [15] conducted research revealing the susceptibility of HTTP/2 to slow Denial of Service (DoS) attacks. Their study employed a comprehensive set of malicious packets, encompassing eleven types of slow DoS, targeting commonly used test targets, including Apache versions 2.4.17 and 2.4.29, Nginx version 1.14.0, Lighttpd version 1.4.55, and MS-IIS version 10.0. The findings indicated that HTTP/2 servers across these platforms encountered difficulties in identifying underlying threats and attacks. This investigation underscores the imperative need for continuous enhancement and modification of web server configurations to bolster security and mitigate the identified types of attacks. The authors strongly advocate the implementation of supplementary protection measures, such as intrusion prevention systems, to fortify the overall security posture.

In a similar fashion, Zhang et al. [16] demonstrated how a very slow DoS attack called zAttack, can causes the HTTP/2 server to wait while continuously consumes the server resources before ultimately leading to a DoS attack. Again this research tested slow DoS attack based on popular web servers. The test results demonstrated that mainstream web servers that support the HTTP/2 protocol are extremely vulnerable to this type of attack.

Manzoor et al. [18] compared the traffic-level characteristics of HTTP/1 and HTTP/2 traffic using traffic-level statistics, collected in a real operational network using Deep Packet Inspection (DPI) technology. Apart from confirming the rapid growth of HTTP/2 among top Internet companies, their investigations also showed the new features introduced in HTTP/2 lead to significant differences in the network fingerprinting (The network fingerprinting is a technique used in computer networking and cyber security to identify and characterize network devices, services, or systems based on their unique patterns of behavior or attributes.) of the HTTP/2 protocol.

Liu et al. [17] introduced two novel Content Delivery Network (CDN) DoS attacks: HTTP/2 Traffic Amplification (HTA) attack and HTTP/2 Slow Rate (HSR) attack. They conducted HTA and HSR attack experiments on ten popular CDNs to assess their feasibility and impact in the real world. The results showed that HTA attack enables malicious users to consume the source server's bandwidth, while the HSR attack can exhaust all available connections on the source server. Under the worst-case scenario, attackers have the capability to amplify network traffic by up to 403, 092 times, presenting a considerably very serious DoS threat to CDN services. As a result, it is recommended that mitigation measures are required to effectively counteract these attacks.

IV. RELATED WORK FOR HTTP/2 DDoS DETECTION

Firstly, HTTP/2, in comparison to HTTP/1.1, not only retains the core functionalities of HTTP/1.1 but also introduces new features such as HPACK header compression, request multiplexing, server push, and header compression. As outlined, these additions made it challenging for eavesdroppers to monitor or discern the operational state of activities on HTTP/2 websites. Network flooding attacks can bypass intrusion detection systems by utilizing a new network communication protocol (HTTP/2).

Machine learning techniques can be applied and used to categorize network traffic into attack traffic and normal traffic. This creates an urgent need to understand HTTP/2 characteristics and design customized network attack detection schemes.

Adi et al. [19] proposed a technique to generate the set of network traffic features for network intrusion detection. The technique showed that the intrusion detection system is able to categorize previously unseen network traffic samples with fewer false alarms than the techniques used in the literature.

Manzoor et al. [18] made two contributions in his research. First, the difference between HTTP/1 and HTTP/2 traffic was characterized using passive measurement datasets collected in operational networks and Deep Packet Inspection (DPI).

They noted that when comparing the same services on HTTP/1 and HTTP/2, HTTP/2 flows are longer and consist of smaller packets. This may be a result of the new protocol functional features of HTTP/2 both happening at the server and the client sides during communication.

Second, a lightweight method is proposed for categorizing encrypted web traffic into corresponding HTTP versions. To make the method practical, machine learning is used in conjunction with basic information commonly found in aggregated flow traces (e.g. NetFlow records). They evaluate five classic machine learning performance metrics ($\mathrm{F}-\mathrm{measure} = 2 \times (\mathrm{Precision} \times \mathrm{Recall})/(\mathit{Precision} + \mathit{Recall})$) and demonstrate that the decision tree is most suitable for solving this problem. The results showed that it can accurately categorize several months of traffic without retraining. The method is simple, scalable, and suitable for situations where DPI is not possible.

Tripathi et al. [20] proposed a solution based on event sequence analysis for detecting slow HTTP/2 DoS attacks with high accuracy and lower computational overhead. They found that some existing web servers are susceptible to such slow DoS attacks, emphasizing the need for protection against them. They transformed the continuous interactions between clients and HTTP/2 servers into event sequences, storing all possible normal event sequences in a database with a normal pattern. This database is then used to detect abnormal event sequences. This means that if the response event sequence generated by an HTTP/2 request is not found in the database, the event can be interpreted as an attack request.

Adi et al. [12] introduced a DoS attack traffic model specifically targeted at HTTP/2 web servers. Their research builds upon previous studies involving DoS attack models for HTTP/2 services, and by introducing and analyzing a novel stealthy variant of DoS attack capable of covertly disrupting regular web services in terms of higher rate of false alarms, the researchers used four machine learning techniques, namely Naive Bayes, Decision Tree, JRip, and Support Vector Machines to counter such attacks using the characteristics of false alarm rate. The simulation results exhibited promising outcomes, and provided valuable insights into potential future advancements to counteract new variants of such attacks.

In the case that recognized defenses against HTTP/1.1 attacks that do not work against HTTP/2 attacks, Anand et al. [24] proposed a machine learning based DDoS detection technique for slow rate attack. The method uses four one-class techniques to achieve HTTP/2 DDoS attack detection: SVM (Support Vector Machine), IF(Isolation Forest), MCD (Minimum Covariant Determinant), and LOF (Local Outlier Factor). The model of detection result was very good with accuracy being 0.99, sensitivity being 0.99. Furthermore, the experiment was carried out in a real operating environment, hence, their method holds great potential for being implemented in real-world applications.

From the research experiments based on HTTP/2 traffic data, Brissaud et al. [25] made two important observations. First, it was observed that the traffic generated by the same request may change over time. Secondly, the testing model can only be validated with a large number of different instances. Call their method the test-of-time and test-of-space respectively. Finally, it is noted that the HTTP/2 classifier is specifically designed to monitor user activity in HTTP/2-based services (HTTP/2 over TLS), but the designed methodology and results can be useful for other encrypted traffic classification or fingerprinting techniques that utilize machine learning.

Tripathi et al. [26] proposed a solution based on event sequence analysis for detecting slow HTTP/2 DoS attacks with high accuracy and lower computational overhead. They found that some existing web servers are susceptible to such slow DoS attacks, emphasizing the need for protection against them. They transformed the continuous interactions between clients and HTTP/2 servers into event sequences, storing all possible normal event sequences in a database with a normal pattern. This database was then used to detect abnormal event sequences. This means that if the response event sequence generated by an HTTP/2 request is not found in the database, the event can be interpreted as an attack request.

In this work, Praseed et al. [23] proposed a method for modeling HTTP/2 requests as fuzzy multiset. The proposed method uses a combination of relative base and request load to detect multiplexing based application layer DDoS attacks. The detection method was tested on an open source dataset and the results showed that the method is able to detect multiplexed AL-DDoS attacks with about 95% accuracy while maintaining a low False Positives Rate (FPR) of about 3%.

V. HTTP/2 DDoS ATTACK AND DETECTION TECHNIQUES MATRIX

To aid security professionals in more effectively detecting potential HTTP/2 DDoS threats, this study, based on the compilation and analysis of literature over the years, introduces a matrix outlining HTTP/2 DDoS attack types and detection techniques. This matrix aligns existing HTTP/2 DDoS attack types with their corresponding detection technologies, enabling the formulation of more comprehensive and effective attack detection strategies. Table 3 provides a detailed presentation of the matrix. It is noteworthy that, due to space constraints, author names are used instead of corresponding detection methods.

Utilizing this matrix, the current state of HTTP/2 DDoS attack types and their corresponding detection techniques is presented in the most concise and intuitive manner. For instance, concerning attack types, the achievements in detecting Slow HTTP/2 DDoS Attacks are more prominent, whereas research on Multiplexing-Based DDoS Attacks is relatively scarce. Additionally, research on detecting Stealthy HTTP/2 DDoS Attacks of this nature came to a standstill as early as 2017. From another perspective of the matrix, it is also evident that from 2016 to 2023, only six researchers have been engaged in the field of HTTP/2 DDoS attack detection. This number is significantly lower compared to research in the field of HTTP/1.

VI. FUTURE CHALLENGES ASSOCIATED WITH HTTP/2 DDoS DETECTION

The effort made in this paper is to extensively review all literature on HTTP/2 from 2015 to 2023. Through a thorough analysis of these research papers, it becomes evident that the HTTP/2 protocol has undergone significant optimizations, resulting in notable improvements in network transmission speed and efficiency, as well as a reduction in communication latency. However, these advancements have also triggered a new wave of attacks targeting HTTP/2. Indeed, attackers can leverage the multiplexing feature of HTTP/2 to generate a substantial volume of junk requests, leading to the consumption of network bandwidth and server resources. Additionally, slowloris attacks are also applicable in the context of HTTP/2 networks. Attackers operate under the fundamental principle of maximizing the impact and extent of their attacks while minimizing costs. As a consequence, the repercussions of these attacks far exceed those observed during the era of HTTP/1.

Although the study of how the security risks can be devised or initiated has been explained above, it is noted that during the period from 2015 to 2023, from a quantitative perspective, research on attack detection techniques specific to HTTP/2 remains notably less abundant compared to that concerning HTTP/1. A considerable portion of detection techniques still remains in the early stages of utilizing statistical or machine learning-based approaches. There remain the need of more research to devise a more superior methodology which can detect a wide variety of DDoS attacks. Nevertheless, as shown in Table 2, scholars in the relevant field have meticulously described their research work on HTTP/2 security risks and proposed corresponding countermeasures. For instance, as may be noted from Table 2, they have investigated and put forth various detection methods tailored to tackle the DDoS of the HTTP/2 protocol. They can

TABLE 3. The matrix of HTTP/2 DDoS attack detection techniques.

Paper	Method	Multiplexing-Based DDoS Attack	Slow HTTP/2 DDoS Attack	Stealthy HTTP/2 DDoS Attack	Encrypted HTTP/2 DDoS Attack Traffic	Year
[18]	Manzoor				√	2016
[12]	Adi			√		2017
[19]	Adi			√		2017
[20]	Tripathi		√			2018
[25]	Brissaud				√	2020
[23]	Praseed	√				2021
[24]	Anand		√			2023
[26]	Tripathi		√			2023

be used for providing a robust foundation platform for the investigation, research of attack detection methodology and countermeasures.

Considering the aforementioned issues, several recommendations based on our analysis of current HTTP/2 attacks and detection techniques have been proposed. These recommendations aim to provide guidance for enhancing HTTP2 server security as well as to define future research directions in the field of attack detection within the context of HTTP/2.

A. THE INHERITANCE AND RENEWAL OF KNOWLEDGE

When comparing literature related to HTTP/1 attack detection methods, it's easy to find no fewer than 10,000 articles. It is evident that existing HTTP/1 attack detection techniques are more mature and abundant. The HTTP/2 protocol, rather than being an entirely independent new standard, is an extension and improvement of its predecessor HTTP/1. As a result, it should be viewed as an enhancement and optimization of the previous version as it was designed to maintain compatibility with the semantics, patterns, and other aspects of HTTP/1. The updates can be regarded as more of making localized improvement.

As the overall technology has yet to mature, combining and building upon existing research outcomes in the field of HTTP/1 attack detection is crucial, with a focus on seamlessly transitioning to HTTP/2 scenarios, as a key point and breakthrough to consider.

Deep learning algorithm detection model has demonstrated remarkable successes in various diverse fields, as well as in HTTP/1 disruptive DDoS attacks [27], [28]. Hence, progressive development can leverage on some of the cutting-edge HTTP/1-based deep learning algorithms, to develop network attack detection techniques for the HTTP/2 protocol. This could actually provide stronger motivations and alleviate as well as to facilitate the transition of HTTP/1 servers to become HTTP/2 servers due to the enhanced DDoS security feature.

B. THE LIMITATIONS OF ENCRYPTED TRANSMISSION FOR ATTACK DETECTION

Even though the design of HTTP/2 itself allows for non-encrypted communication, and the protocol itself does not require the use of encryption, however, the developers' of all major web browsers (e.g. Chrome, Safari, Firefox, Opera, IE, Edge) currently stated that they will only implement HTTP/2 with TLS encryption as mandatory standard. As a result, due to the dichotomous standard between the HTTP/2 servers and the web browser clients, this present a challenge. This implies that from the web browsers' perspective, HTTP/2 based transmissions will be in an encrypted mode, which poses a significant limitation to DDoS attack detection. How to convert encrypted packets to plaintext in this case is a problem that need to solve in the future. Berman et al. [28] have assessed the security factor of HTTP/2. By using packet capturing tools, they intercepted data packages and showed how it is feasible to decrypt HTTP/2 encrypted traffic using SSL-KEY-LOGGING. This is a serious security issue as it may expose both sensitive and vital information.

C. FOCUS ON HTTP/2 DDoS ATTACKS BASED ON MULTIPLEXING

The proliferation of the Internet, in terms of service, data, domain and application have made it into a daily life routine of today. As DDoS attacks are a type of network attack that aims to overwhelm or exhaust a target system by simultaneously sending a large volume of requests and data traffic, this results in legitimate users being unable to access or use the system normally. DDoS attacks pose a significant and highly destructive network interruptions and threats. As the Internet becomes increasingly prevalent in various domains, the frequency and scale of DDoS attacks have been steadily rising too. Therefore, safeguarding systems and networks from DDoS attacks based on the HTTP/2 protocol is a crucial aspect of network security. Effective research and development of DDoS detection techniques based on the HTTP/2 protocol are essential to ensure network stability and enhance user experience in today's context.

In essence, the feature of multiplexing in HTTP/2 allows obtaining multiple resources with just once request, enabling DDoS attacks to more be able to effectively exploit this feature to execute sophisticated and devastating attacks.

Requires focused attention in terms of security analysis and protection against potential exploitation in HTTP/2 DDoS attacks.

D. THE LACK OF DATASETS

Many researchers currently utilize publicly accessible realistic datasets to validate their proposed methods. They mainly relied on KDD Cup 99 and CAIDA datasets to study DDoS traffic in HTTP/2. However, these datasets are outdated, and the proposed models may not efficiently identify the latest DDoS attacks in the HTTP/2 scenarios of today. To effectively model heterogeneous DDoS attack traffic, the use of realistic or carefully crafted and up-to-date HTTP/2 datasets is a significant requirement.

Simulating HTTP/2 traffic to generate dataset is an effective method. Use tools to simulate both normal and DDoS attack traffic, ensuring that the experiments cover various scenarios. Through such simulated data, it helps improve and validate your DDoS attack detection techniques. It is crucial to ensure that the dataset is representative enough, including various attack types and normal traffic, to enhance the reliability and applicability of the experiments.

VII. CONCLUSION

The introduction of new features in the HTTP/2 protocol offers significant benefits that should be acknowledged and harnessed. However, it is equally important not to overlook the potential network security risks associated with these advancements. Various research works have shown that DDoS attacks can exploit these new features by sending carefully crafted HTTP/2 Request to web servers, resulting in network service disruptions. There remain relatively low number of research on providing good references in counteracting HTTP/2 DDoS attacks. This article aims to take cognizant of the more contemporary research findings, analysis and ideas from the past 8 years, to spur more focused research in both the detection and countermeasures techniques of DDoS on HTTP/2 servers. Additionally, it provides relevant suggestions to guide future research in attack detection direction. It is crucial to highlight that the introduction of multiplexing and server push technologies in HTTP/2 enables attackers to launch Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks on HTTP/2 servers with low costs. Therefore, it is recommended that future research focuses more on the detection and prevention of such DDoS attacks which based on multiplexing and server push technologies.

ACKNOWLEDGMENT

The authors would like to thank the Faculty of Computing and Informatics, Universiti Malaysia Sabah, and also to thank the anonymous reviewers, whose valuable comments and suggestions have allowed a clear legible and developmental path to be taken in improving the quality of this article.

REFERENCES

[1] M. Jazayeri, "Some trends in web application development," in *Proc. Future Softw. Eng. (FOSE)*, May 2007, pp. 199–213.

[2] M. Laner, P. Svoboda, P. Romirer-Maierhofer, N. Nikaein, F. Ricciato, and M. Rupp, "A comparison between one-way delays in operating HSPA and LTE networks," in *Proc. 10th Int. Symp. Modeling Optim. Mobile, Ad Hoc Wireless Netw. (WiOpt)*, May 2012, pp. 286–292.

[3] M. Wijnants, R. Marx, P. Quax, and W. Lamotte, "HTTP/2 prioritization and its impact on web performance," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1755–1764.

[4] N. Tripathi and N. Hubballi, "Application layer denial-of-service attacks and defense mechanisms: A survey," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–33, 2021.

[5] N. Tripathi, N. Hubballi, and Y. Singh, "How secure are web servers? An empirical study of slow HTTP DoS attacks and detection," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 454–463.

[6] D. Beckett and S. Sezer, "HTTP/2 Cannon: Experimental analysis on HTTP/1 and HTTP/2 request flood DDoS attacks," in *Proc. 7th Int. Conf. Emerg. Secur. Technol. (EST)*, Sep. 2017, pp. 108–113.

[7] D. Beckett and S. Sezer, "HTTP/2 tsunami: Investigating HTTP/2 proxy amplification DDoS attacks," in *Proc. 7th Int. Conf. Emerg. Secur. Technol. (EST)*, Sep. 2017, pp. 128–133.

[8] R. Corbel, E. Stephan, and N. Omnes, "HTTP/1.1 pipelining vs HTTP2 in-the-clear: Performance comparison," in *Proc. 13th Int. Conf. New Technol. for Distrib. Syst. (NOTERE)*, Jul. 2016, pp. 1–6.

[9] *Study on CT WG3 Aspects of 5G System Phase 1*, document TR 29.890, 3GPP, 2018.

[10] X. Hu, C. Liu, S. Liu, W. You, and Y. Zhao, "Signalling security analysis: Is HTTP/2 secure in 5G core network?" in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.

[11] E. Adi, Z. Baig, C. P. Lam, and P. Hingston, "Low-rate denial-of-service attacks against HTTP/2 services," in *Proc. 5th Int. Conf. IT Converg. Secur. (ICITCS)*, Aug. 2015, pp. 1–5.

[12] E. Adi, Z. Baig, and P. Hingston, "Stealthy denial of service (DoS) attack modelling and detection for HTTP/2 services," *J. Netw. Comput. Appl.*, vol. 91, pp. 1–13, Aug. 2017, doi: 10.1016/j.jnca.2017.04.015.

[13] A. G. Jaafar, S. A. Ismail, M. S. Abdullah, N. Kama, A. Azmi, and O. M. Yusop, "Recent analysis of forged request headers constituted by HTTP DDoS," *Sensors*, vol. 20, no. 14, p. 3820, Jul. 2020, doi: 10.3390/s20143820.

[14] N. Tripathi and A. K. Shaji, "Defer no time, delays have dangerous ends: Slow HTTP/2 DoS attacks into the wild," in *Proc. 14th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2022, pp. 194–198.

[15] M. Sikora, R. Fujdiak, K. Kuchar, E. Holasova, and J. Misurec, "Generator of slow denial-of-service cyber attacks," *Sensors*, vol. 21, no. 16, p. 5473, Aug. 2021.

[16] Y. Zhang and Y. Shi, "A slow rate denial-of-service attack against HTTP/2," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018, pp. 1388–1391.

[17] H. Song, J. Liu, J. Yang, X. Lei, and G. Xue, "Two types of novel DoS attacks against CDNs based on HTTP/2 flow control mechanism," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, Sep. 2022, pp. 467–487.

[18] J. Manzoor, I. Drago, and R. Sadre, "How HTTP/2 is changing web traffic and how to detect it," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2017, pp. 1–9.

[19] E. Adi and Z. Baig, "Intelligent feature selection for detecting HTTP/2 denial of service attacks," in *Proc. 15th Austral. Inf. Secur. Manage. Conf.*, Dec. 2017, p. 57.

[20] N. Tripathi and N. Hubballi, "Slow rate denial of service attacks against HTTP/2 and detection," *Comput. Secur.*, vol. 72, pp. 255–272, Jan. 2018.

[21] N. Oda and S. Yamaguchi, "HTTP/2 performance evaluation with latency and packet losses," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–2.

[22] A. Praseed and P. S. Thilagam, "Multiplexed asymmetric attacks: Next-generation DDoS on HTTP/2 servers," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1790–1800, 2020.

[23] A. Praseed and P. S. Thilagam, "Fuzzy request set modelling for detecting multiplexed asymmetric DDoS attacks on HTTP/2 servers," *Expert Syst. Appl.*, vol. 186, Dec. 2021, Art. no. 115697.

[24] N. Anand and M. A. Saifulla, "An efficient IDS for slow rate HTTP/2.0 DoS attacks using one class classification," in *Proc. IEEE 8th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2023, pp. 1–9.

[25] P. O. Brissaud, J. François, I. Chrisment, and O. Bettan, "Encrypted HTTP/2 traffic monitoring: Standing the test of time and space," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Jun. 2020, pp. 1–6.

[26] N. Tripathi, ‘‘Delays have dangerous ends: Slow HTTP/2 DoS attacks into the wild and their real-time detection using event sequence analysis,’’ *IEEE Trans. Dependable Secure Comput.*, vol. 1, pp. 1–13, 2023.

[27] A. A. Abdul Lateef, S. T. F. Al-Janabi, and B. Al-Khateeb, ‘‘Survey on intrusion detection systems based on deep learning,’’ *Periodicals Eng. Natural Sci.*, vol. 7, no. 3, p. 1074, Aug. 2019.

[28] D. Berman, A. Buczak, J. Chavis, and C. Corbett, ‘‘A survey of deep learning methods for cyber security,’’ *Information*, vol. 10, no. 4, p. 122, Apr. 2019.

[29] M. Suresh, P. P. Amritha, A. K. Mohan, and V. A. Kumar, ‘‘An investigation on HTTP/2 security,’’ *J. Cyber Secur. Mobility*, vol. 10, pp. 161–180, Jan. 2018.

LIANG MING received the B.Sc. degree (Hons.) in physics from Chongqing College of Arts and Sciences, China, in 2005, and the M.Sc. degree in software development computer science from Southwest University, Chongqing, China, in 2013. He is currently pursuing the Doctor of Philosophy degree in computer science with Universiti Malaysia Sabah. He is also with the Office of Informatization (OOI), Chongqing College of Arts and Sciences, where he has engaged in various computer network security related works, which include big data analysis, software development, and IPv6 network design and construction.

YU-BENG LEAU (Senior Member, IEEE) received the B.Sc. degree in multimedia technology from Universiti Malaysia Sabah, the M.Sc. degree in information security from Universiti Teknologi Malaysia, and the Ph.D. degree in internet infrastructures security from Universiti Sains Malaysia. He is currently a Senior Lecturer with the Faculty of Computing and Informatics, Universiti Malaysia Sabah. His current research interests include intrusion detection and prediction, network security situation awareness, IPv6 security, the Internet of Things (IoT), and information centric networks (ICN).

YING XIE received the B.Sc. degree (Hons.) from Yangtze Normal University, China. She is a teacher and has engaged in relevant scientific research. She is proficient in data collection and analysis. She has published her research in Children’s International Understanding Education Research Workshop. Her current research interests include abnormal detection and deep learning.