Detection and Mitigation of Low-Rate Denial-of-Service Attacks: A Survey

Vinicius de Miranda Rios, Pedro R M Inacio, Damien Magoni, Mario M Freire

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 June 2022, accepted 7 July 2022, date of publication 15 July 2022, date of current version 26 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3191430

SURVEY

Detection and Mitigation of Low-Rate Denial-of-Service Attacks: A Survey

VINÍCIUS DE MIRANDA RIOS[1,2], PEDRO R. M. INÁCIO[2], (Senior Member, IEEE), DAMIEN MAGONI[3], (Senior Member, IEEE), AND MÁRIO M. FREIRE[2], (Member, IEEE)

[1]Instituto de Educação, Ciência e Tecnologia do Tocantins-IFTO, Palmas 77021-090, Brazil
[2]Instituto de Telecomunicações e Departamento de Informática, Universidade da Beira Interior, 6201-001 Covilhã, Portugal
[3]LaBRI, CNRS, University of Bordeaux, 33400 Talence, France

Corresponding author: Vinícius de Miranda Rios (vinicius.rios@ifto.edu.br)

This work was supported in part by the Portuguese Fundação para a Ciência e a Tecnologia (FCT)/Ministério da Ciência, Tecnologia e Ensino Superior (MCTES) through national funds and, when applicable, co-funded by EU funds under Project UIDB/50008/2020; in part by FCT/Programa Operacional Competitividade e Internacionalização (COMPETE)/Fundo Europeu de Desenvolvimento Regional (FEDER) under project SECURIoTESIGN under Grant POCI-01-0145-FEDER-030657; in part by C4-Centro de Competências em Cloud Computing under Grant Centro-01-0145-FEDER-000019; and in part by the European Regional Development Fund (ERDF) through the Programa Operacional Regional do Centro, Centro 2020. The work of Vinícius de Miranda Rios was supported by the Brazilian Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) Foundation under Contract BEX 9095/13-6.

ABSTRACT The potential for being the target of Denial of Service (DoS) attacks is one of the most severe security threats on the Internet. Attackers have been modifying their attack format over the years, damaging specific conditions of operating systems and protocols in an attempt to deny or diminish the quality of the service provided to legitimate users. Nowadays, attacks are stealthier and mimic legitimate user traffic in such a way that detection mechanisms against High-rate DoS attacks are no longer sufficient. This evolving type of attack, known as LDoS (Low-rate Denial of Service) attacks, has the potential to produce more damage than its predecessor due to its stealth nature and the lack of suitable detection and defense methods. This survey summarizes and complements previous studies and surveys related to this specific type of attack. First, we propose a taxonomy of the LDoS attacks, which were divided into three broad categories based on their modus operandi: QoS attacks, Slow rate attacks, and Service queue attacks. Next, we detail numerous detection mechanisms and counter-measures available against eight types of LDoS attacks. More specifically, we describe the methods used to throttle the attack traffic. Finally, we provide a feature comparison table for some existing attack tools. This survey aims at providing an extensive review of the literature for helping researchers and network administrators find up-to-date knowledge on LDoS attacks.

INDEX TERMS Denial-of-service, DoS, distributed DoS, DDoS, low-rate DoS attack, LDoS attack, LDDoS attack, DoS threat, LDoS detection mechanisms, slow DoS attack.

I. INTRODUCTION

The services offered today by the Internet are of various types and features, ranging from a simple exchange of messages via e-mail to video streaming and bill payments. These services are of great importance to persons, companies, and governments, providing convenience and speed for contacting a customer, watching a movie, or requesting the delivery of products or food. However, such services can become a major inconvenience to vendors if they do not work properly for even a few minutes, since an unavailable service may cause damage to revenues and the trust of customers. A major problem with Internet protocols is that they were not originally designed to include security. Therefore, several threats related to them have emerged over the last few decades. Among the existing threats against security systems are Denial-of-Service (DoS) attacks, which since their inception have caused major monetary losses for people, companies, and governments that rely on the Web as their main source of revenue [1], [2]. High-rate DoS attacks have

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

become a significant threat to Internet safety by adding the many-to-one dimension to the DoS (one computer and one Internet connection is used to swamp a target) problem [3] and amplifying it into a lethal traffic force. DDoS (Distributed Denial-of-Service) attacks have the aim of making resources or services unavailable to legitimate users by using a distributed, cooperative, and large-scale attack that causes damages to the system server, which may shut systems down, corrupt files, and partially or even completely compromise services [3]–[5]. The attack begins with the attacker, also called the botmaster or botherder (whose motivation behind DDoS attacks is mainly political/ideological, financial gain, or competitive advantage) infecting vulnerable computers (second victims) also called ''zombie machines'' (generally recruited through worms, Trojan horses, or backdoors), which form a Botnet (roBot Networks - networks formed by malware-compromised machines through the Internet), launching thousands of network traffic to the victim [6]–[8] in order to ruin their service, profits, and reputation. For example, from 2000 until now some of the largest and best-known companies on the Internet and entertainment (e.g. Yahoo.com, Buy.com, eBay, CNN, Amazon.com, etc) and many smaller ones have suffered DoS attacks with a huge amount of distributed network traffic that has paralysed their services [2], [9].

However, the key characteristics of DDoS flooding attacks are now well-known, and countermeasures that protect or at least attempt to protect victims from them can already be found in the literature. Thus, attackers have begun to change their strategy, denying service to the victims in a different way. They mimic legitimate user network traffic patterns by using low many-to-one dimension rate attacks, which are being called the LDoS attacks. Therefore, almost all defense mechanisms against DDoS strikes are not effective against this new attack format [10], [11].

LDoS attacks exploit specific vulnerabilities by focusing on semantic methods, but with the main feature of sending an amount of data traffic not exceeding the victim's bandwidth. The attack signature sends a high speed burst flow, repeated at a fixed low-time-scale frequency in a square wave pattern in order to cheat the system whose network links are being occupied [12]–[14]. For example, in 2001, Internet2 Abilene backbone was hit by short bursts of attack traffic rather than a large amount of attack traffic [15]. Asa Networks discovered that pulsing zombies are causing this type of attack. Later in 2004, the website qq.com in China was hit by some kind of Low-rate DoS attack [11]. Therefore, distinguishing LDoS attacks is a difficult task and requires solutions that can be at the same time scalable, accurate, and effective, allowing the legitimate traffic user to suffer less impact (false positive) or, in a different way, by not allowing malicious traffic to hit the victim (false negative).

The remainder of this article is divided as follows. Section 2 presents the previous surveys on LDoS. Section 3 describes a group division and concepts regarding LDoS attacks. Section 4 describes the detection methods and compares them to each other. Section 5 shows the tools used to launch the LDoS attacks traffic against the target and the tools used to defending against those attacks, while Section 6 concludes the survey.

II. REVIEW OF PRIOR SURVEYS ON LDoS ATTACKS

This section presents an overview of the prior surveys targeting exclusively low-rate DoS attacks, with the aim of comparing them by their content regarding attack taxonomy, attack classification, and defense mechanisms.

In 2011, Zhu *et al.* provided a cursory and quick introduction about LDoS attacks [11]. They showed how shrew, RoQ and Pulsing DoS attacks work and demonstrate through simulations how such attacks perform in a general way.

At the same time, a comparison of attack detection mechanisms was made by Mathew and Katkar between two types of LDoS, namely LoRDAS (Low-Rate DoS attacks against Application Servers) and Shrew attacks, describing how such mechanisms act by avoiding a succession of attacks as well as pointing out the characteristics related to the features that an ideal solution must have [16].

In 2012, a brief comparison between flood-based DoS attacks and LDoS attacks was made by Liu *et al.* in [10]. They compared both DoS concepts and showed that new defense mechanisms needed to be created in order to stop LDoS attacks due to their stealthy mode of operation.

In the same year, Mohan *et al.* in [17] produced a survey of the evolution proposed by Mathew and Katkar in [16], adding other detection mechanisms, of which some are based on the AQM (Active Queue Management) RED (Random Early Detection) algorithm and its variations. The authors also presented a defense mechanism based on a modified version of RED-PD (Preferential Dropping) that detected LDoS attacks, but did not specify which LDoS type of attack had been detected, even though the architecture of the proposal demonstrated that the attack was probably a Shrew attack.

A taxonomy of Slow DoS attacks was proposed by Cambiaso *et al.* in 2012, classifying them based on the attack characteristics, covering only malicious activities that target the application layer [18]. One interesting fact was that they considered the LoRDAS attack as a web threat that had the same slow DoS attack characteristics, which in other related papers were classified as a low-rate attack.

The following year, Cambiaso *et al.* in [19] described the evolution of the taxonomy since their previous paper in [18]. In the latter, they added new attacks and divided them into 3 new divisions based on the attack type and feasibility. They also adjusted the LoRDAS attack with an appropriate classification that was related to target an application running in a server not being exclusively a web application.

In early 2020, a survey on low-rate DoS attacks written by Zhijun *et al.* presented the most important active LDoS attacks, with an extensive overview of these malicious activities [20]. They classified and categorized them based on their characteristics. They also expanded the discussion about the detection and defense mechanisms used by researchers,

TABLE 1. Comparison of prior surveys and this article.

Works	Network security attack contextualization	Attack type classification	Attack tools	Attack target			Attack performance	Low-rate DoS attacks			Detection mechanisms
				Application layer	Transport layer	Network layer		QoS attacks	Slow DoS attacks	Service queue attacks	
Zhu et al. (2011) [11]	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗
Mathew and Katkar (2011) [16]	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
Liu et al. (2012) [10]	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓
Mohan et al. (2012) [17]	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
Cambiaso et al. (2012) [18]	✓	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓
Cambiaso et al. (2013) [19]	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✗
Zhijun et al. (2020) [20]	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✓
Tripathi and Hubballi (2021) [21]	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓
This survey (2022)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

comparing their methods and techniques in order to mitigate the attacks. However, the papers studied in this survey date back to 2018 and earlier. Since then, tens of papers on LDoS attacks have been published.

In 2021, a research study presented by Tripathi and Hubballi in [21] classified the attacks based on application layer protocols (ALP). This survey does not specifically target LDoS attacks but covers ALP-only attacks. They divided them into two categories: protocol specific and generic. The protocol specific category summarizes the application protocols that have a specific vulnerability to be hit. The generic category summarizes the application protocols that have generic vulnerabilities to be hit. In all the categories the authors presented a comparison among the attacks as well as the defense mechanisms.

In the same year, a survey [22] and several papers focusing on regular DoS attacks [23]–[27] were published. Our survey aims to close the gap concerning recently published papers focusing on LDoS attacks. Indeed, the aforementioned studies about LDoS attacks need to be updated with new information on issues such as new attacks, new defense mechanisms, and new classifications due to this ever-growing attack paradigm. Therefore, in order to stop or at least mitigate them, it is vital that researchers and network managers recognize and envision the LDoS attacks as a whole. Thus, one of the main contributions of this paper is to provide a state-of-the-art perspective, as well as an extensive and comprehensive high-level guide to LDoS attacks, specifically isolating and emphasizing the characteristics related to these malicious activities.

Table 1 provides a macro perspective on the major surveys and related papers in this area, comparing them in several important aspects related to Low-rate DoS attacks. The *Network security attack contextualization* field provides papers that have a detailed description of the LDoS. The *Attack type classification* field provides papers which have classified the threats according to certain characteristics. The *Attack tools* field provides papers that explain the tools built for each threat to be successful in its task. The *Attack target* layer field is divided into 3 categories, which are: *Application layer*, *Transport layer* and *Network layer*. It includes papers which have attacks that damage the target in each one of these categories. The *Attack performance* field shows the accuracy result for each detection/defense mechanism against

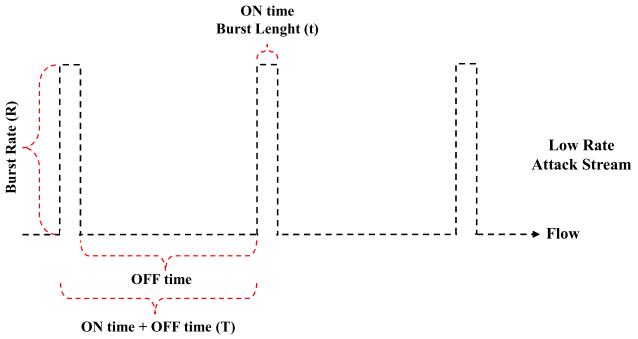

FIGURE 1. QoS attacks format (based on [28]–[30]). Where *t* is burst length of the traffic, *R* is the burst rate of the traffic and *T* is the total time for the attack period.

the LDoS attacks. The *Low-rate DoS attacks* field is divided into 3 categories: *QoS (Quality of Services) attacks*, these include attacks that aim at reducing the throughput of data traffic transmission; *Slow DoS attacks*, which include attacks that have the aim of stopping the target service, consuming its resources and damaging the protocols of the application layer, and the *Service queue attacks*, which include attacks that have the aim of degrading the performance of the applications processing by damaging the processor queuing of the incoming packets. And finally, the *Detection mechanism field*, which includes papers that explain the methods and techniques for mitigating LDoS attacks.

III. LOW-RATE DoS ATTACKS

This section presents the Low-rate DoS attack taxonomy, schematized in Figure 2, which is categorized into 3 division types, based on the damage caused to the protocol used by the services, that are: QoS attacks, Slow DoS attacks and Service queue attacks as described below.

A. QoS ATTACKS

The attacks in this category have the aim of damaging the quality of services provided by applications hitting the protocols at the transport and network layers. In this case, the protocols that needed the confirmation statement are the ones most affected by these types of attacks. Figure 1 shows the traffic model used as basis for generating the attack traffic pattern.

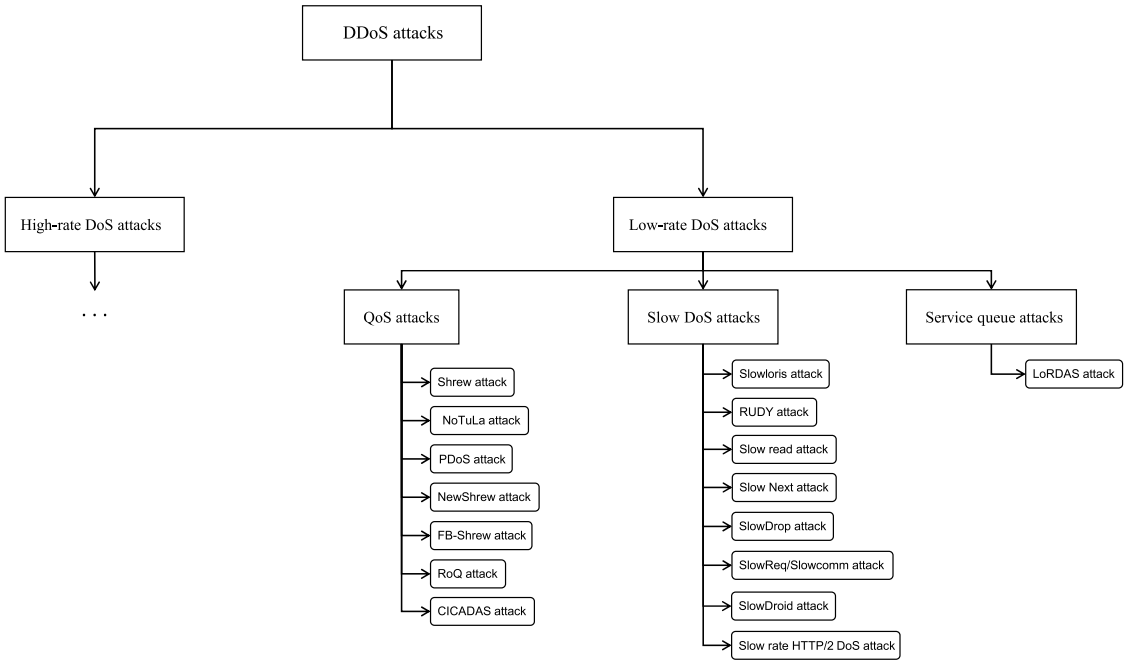

FIGURE 2. Low-rate DoS attacks taxonomy.

1) SHREW ATTACK

This attack has the aim of reducing the traffic throughput of the TCP-based application services to nearly zero. The attacker throttles the legitimate TCP (Transmission Control Protocol) flows with low-rate periodic on-off "square-wave" packets, using multiple distributed or single sources, synchronizing the attack period with the TCP minimum RTO (Retransmission TimeOut), making the target system service consecutively repeat frequency states of overburden with a fixed frequency, shutting off most of the legitimate TCP sources [12], [31]–[33].

2) NoTuLA (NOVEL TUNEABLE LOW-INTENSITY ADVERSARIAL ATTACK)

This attack has the aim of reducing the QoS of the victim's services by sending tunable traffic bursts at tunable periodic times in order to strangle the throughput for the victim [34]. The difference between that and the Shrew attack is that NoTuLA tunes the attack traffic based on 2 stages that are: the monitoring phase and link capacity estimation phase. Instead of sending traffic in a fixed burst periodic time transmission, it is sent in a tuned burst (large enough to create transient congestion) with tuned periodic time (adjusting the inter-arrival time) transmission, with moments of silence when needed.

3) PDoS (PULSING DENIAL-OF-SERVICE) ATTACK

Such an attack also aims at degrading the QoS of the victim's services, but targeting two TCP characteristics [35]. The first focuses on the default target of most of low-rate attacks, which is the retransmission timeout that the authors called timeout-based attack. On the other hand, the second target focuses on the congestion window (cwnd), which was called aimd-based attack. The former attack forces the victim to enter the fast recovery state endlessly. It is important to mention that both attacks are made in the synchronous and asynchronous mode. The main differences between the Shrew attack and the PDoS attack are the aimd-based attack and the asynchronous attack mode.

4) NewShrew ATTACK

This attack has the aim of degrading the QoS of the victim's services by targeting the retransmission timeout and slow start mechanism [36]. The first target is the default for TCP-based attacks. The second one has the aim of disrupting the TCP throughput in order to send burst traffic at the moment that the TCP is recovering from the timeout phase and entering the slow start phase, which quickly begins regaining its transmission rate. Next, the union of the TCP traffic with little attack traffic, forces the TCP to enter in the timeout stage again. The main difference between the NewShrew attack and the Shrew attack is the slow start target.

5) FULL-BUFFER SHREW ATTACK

This attack was first idealized by Guirguis *et al.* in [37] and subsequently improved by Yue *et al.* in [38] and [39]. The aim is to send high-rate traffic bursts only after the queue router buffer is full. This tactic produces maximum damage with minimum resources. The main difference between the Full-Buffer Shrew attack and the Shrew attack is that the first one does not need to match the minimum TCP RTO.

6) RoQ ATTACK

This attack has the aim of reducing service quality to legitimate users. The attacker throttles the network traffic in order to end systems, transmitting high-rate bursts on longer

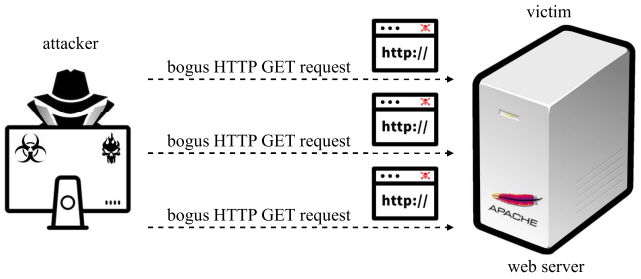

FIGURE 3. Slowloris attack.

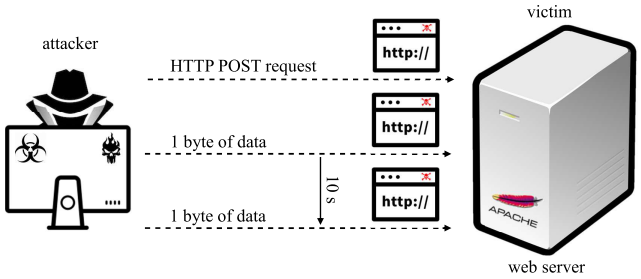

FIGURE 4. Rudy attack.

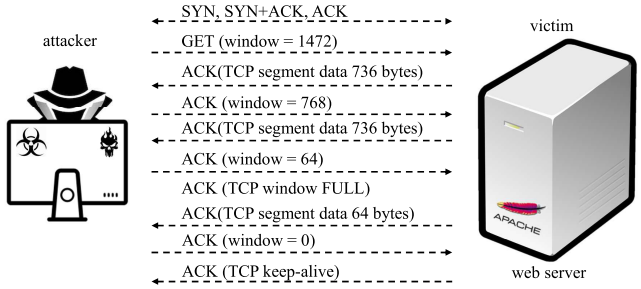

FIGURE 5. Slow read attack (based on [52]).

timescales, and flooding the border router queue on which most legitimate user packets are dropped [40]–[43]. The attack target, which in this case can be any transport layer protocol, is what differentiates the Shrew from the RoQ attacks.

7) CICADAS ATTACK

This attack has the aim of degrading the QoS of the victim's services to legitimate users in synchronized coordination. The bots scattered around the internet send low-rate periodic on-off ''square-wave'' packets towards the victim. Due to the fact that there is no central controller of the bots, the attack traffic synchronization is made by the feedback-based algorithm, adjusting the phase and magnitude of each attack stream based on previous RTT measurements in a dynamic manner [44].

B. SLOW DoS ATTACKS

The attacks in this category ''slow down'' the HTTP (Hyper-Text Transfer Protocol) traffic connections, manipulating the methods and characteristics of the protocol architecture, hence, keeping the network channel active for as much time as it can, consuming resources from the server. In 2009 Iran government servers were hit by this type of attack [18].

1) SLOWLORIS ATTACK

This attack has the aim of stopping web server services by opening hundreds of connections and keeping them open as long as it can. The attacker sends partial HTTP GET requests (adding additional $\backslash r \backslash n$ tags at the end of the request) periodically and simultaneously, with different source ports to the web server opening several other connections, causing it to wait until each of the attack packet requests is completed [18], [45]. Figure 3 illustrates this attack representation.

Once it has filled the targeted concurrent connection at its maximum, all additional connection attempts are denied.

2) SlowReq/SLOWCOMM ATTACK

This attack is a type mix between the Slowloris attack and RUDY attack and has the aim of disrupting web server services. The attacker initially sends partial HTTP GET requests (with no $\backslash r \backslash n$ tags at the end of the request) towards the victim and after that begins to send small packets (e.g. a single character) as a form of keeping the connections alive, denying the services to the legitimate users [46], [47].

3) RUDY (R-U-DEAD-YET) ATTACK

This attack is also called slow HTTP post attack and it has the aim to send fake HTTP POST requests disguised as the legitimate form submission, with the content length header field set as abnormally long. Additionally, the data is broken into packets as small as 1 byte each, which are sent at randomized 10-second intervals that keep the web server tied up endlessly [48], [49]. Figure 4 illustrates this attack.

The length of the long content field forces server connections to stay open, causing them to crash.

4) SLOW READ ATTACK

This attack explores the flow control of TCP by slowly reading the response with the window size smaller than usual [50], [51]. As a result, the attacker forces the web server to operate several connections simultaneously, which consume the resources until it can no longer receive any further requests.

Figure 5 illustrates the attack. This malicious activity is also known as the *low and slow attack* [53].

5) SlowDroid ATTACK

This attack has the same attack characteristics as the SlowReq/Slowcomm attack except that it runs on an android phone [54].

6) SLOW NEXT ATTACK

This attack explores the persistent connections of the HTTP protocol, targeting the timeout connection in order to keep

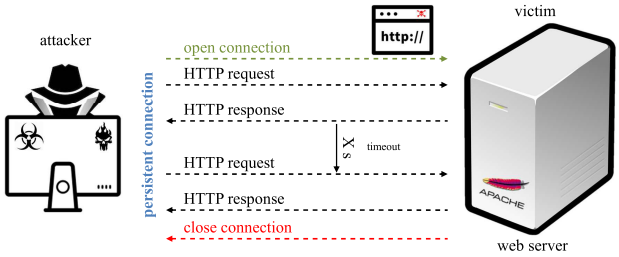

FIGURE 6. Slow Next attack.

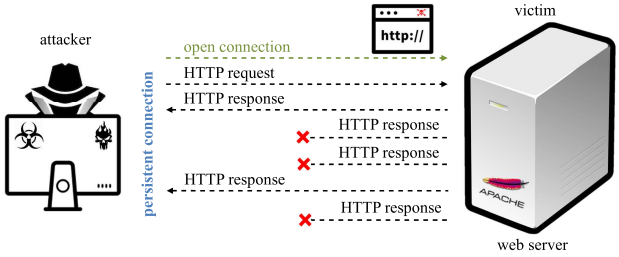

FIGURE 7. SlowDrop attack (based on [56]).

messages alive [55]. It sends a considerable amount of traffic with bogus timeout connection values higher than usual, keeping the web server channel open and maintaining the resource busy endlessly. As a result, legitimate connections are dropped.

Where X is the amount of bogus waiting time in seconds used to keep the channel open. Figure 6 illustrates this attack.

7) SlowDrop ATTACK

This attack opens several connections towards the web server, dropping the HTTP responses.

The aim is to simulate an environment where responses packets can be easily lost, such as a weak wireless connection, hence obliging the web server to endlessly respond to the attacker [56], causing the legitimate connections to drop. Figure 7 illustrates this attack.

8) HTTP/2 DoS ATTACK

This attack can render a web server useless by using specially crafted HTTP/2 requests [57]. In the first crafted HTTP/2 request, the malicious client sends a modified HTTP header with the settings_initial_window_size field configured to zero, which indicates that the client cannot receive any data at that moment. This starts a waiting time by the server for the window_update frame for a specific set of time. In the second crafted HTTP/2 request the malicious client sets and resets the end_headers and end_stream fields inside a complete HTTP POST method to the web server. This causes a waiting time by the server for one or more data frames that have not yet been received, resulting in a particular time duration. In the third crafted HTTP/2 request, the malicious client sends a connection preface to the web server and

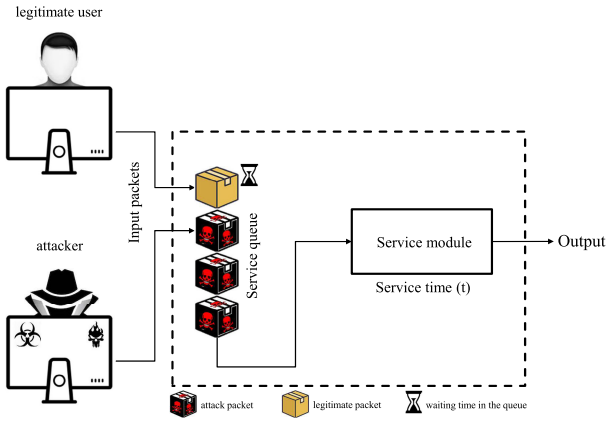

FIGURE 8. LoRDAS attack (based on [58] and [59]).

thus begins waiting for a GET/POST HTTP request which never arrives, causing a particular time duration. In the fourth crafted HTTP/2 request, the malicious client sends two types of incomplete bogus HTTP header methods. The GET and POST methods reset the end_headers and end_stream fields which indicate to the web server that there are other frames coming, causing a particular time duration. In the fifth crafted HTTP/2 requests the malicious client sends complete HTTP GET/POST requests in that the web server answers with a data frame along with two settings frame in that the second settings frame needs to be acknowledged by the client. If this does not happen the web server waits for a specific period of time. This particular time duration for each type of crafted HTTP/2 requests can be unlimited for some web servers, denying the services for the legitimate users or forcing them to wait for an extended period of time before beginning to process the requests again. That causes a long wait to access the service, leading the legitimate clients to give up.

C. SERVICE QUEUE ATTACKS

The attacks in this category directly affect the data processing queue of the services provided by applications in the Internet.

1) LoRDAS ATTACK

This attack has the aim of not allowing legitimate incoming packets to be processed by the application services for the legitimate users at the moment of response.

The attacker causes the service queues of the server to be overloaded and watches to see whenever responses to requests received for a particular service occur, in order to repeatedly insert a malicious request into the service queue. This way, the server is induced by the attacker to be occupied most of the time serving its requests instead of legitimate users [58]–[60]. Figure 8 illustrates the attack.

Table 2 shows a concise overview of the attacks presented in this section where the *Attack* field indicates the name of the menace. The *Target* field informs what the attack targets are in order to achieve success with its goal. The *Goal* field

TABLE 2. Comparison among LDoS attacks.

Attack	Target	Goal	Can IP be spoofed ?	Attack software	Previous knowledge
Shrew attack (2003) in [12]	TCP minimum RTO	Decrease QoS of the applications	Yes	No	TCP retransmission timeout
RoQ attack (2004) in [41]	Router queue	Decrease QoS of the applications	Yes	No	Border router
NoTuLa attack (2005) in [34]	TCP minimum RTO	Decrease QoS of the applications	Yes	No	TCP retransmission timeout value
PDoS attack (2005) in [35]	TCP minimum RTO and AIMD mechanism	Decrease QoS of the applications	Yes	No	TCP retransmission and congestion window
FB-Shrew attack (2006) in [37]	Router queue	Decrease QoS of the applications	Yes	No	Border router
NewShrew attack (2014) in [36]	TCP minimum RTO and slow start mechanism	Decrease QoS of the applications	Yes	No	TCP retransmission timeout and slow start initial phase
CICADAS attack (2016) in [44]	TCP minimum RTO	Decrease QoS of the applications	Yes	No	Previous RTT measurements
Slowloris attack (2009) in [18]	GET method	Stop web server services	No	Yes	HTTP header field
Slow read attack (2012) in [50]	TCP window field	Stop web server services	No	Yes	TCP window size
SlowReq/Slowcomm attack (2014) in [46]	GET method + content length field	Stop web server services	No	No	HTTP header field
SlowDroid attack (2014) in [54]	GET method + content length field	Stop web server services	No	No	HTTP header field
Slow Next attack (2015) in [55]	Wait timeout	Stop web server services	No	No	HTTP persistent connections
RUDY attack (2016) in [48]	POST method + content length field	Stop web server services	No	Yes	HTTP header field
HTTP/2 DoS attack (2018) in [57]	GET/POST methods + settings_initial_window_size field, window_update field, connection preface, end_headers field, end_stream field and settings field	Stop web server services	No	No	HTTP header field
SlowDrop attack (2019) in [56]	HTTP response	Stop web server services	No	No	webserver resources
LoRDAS attack (2007) in [58]	Data processing queue	Decrease QoS or deny the applications services	Yes	No	Queue inter-output time

informs what the purpose of the attack is. The *Can the IP be spoofed?* field tells whether it is possible to spoof the attacker's IP address. The *Attack software* field informs if there is a software to launch the attack and the *Previous knowledge* field includes information on previous knowledge in order for the attack to be successful.

IV. DETECTION AND DEFENSE MECHANISMS

This section presents the detection mechanisms and countermeasures against the attacks presented in the previous section. Tables 3, 4, 5 and 6 provide a brief comparison among the strategies for stopping/defending against the threats. These Tables are compounded by the Work field, which represents the papers that produce the technology, the Detection/Defense mechanism field which represents the methodology for fighting against the attacks, the Detection/Defense location which represents the place where the methodology acts, the Testbed field which represents how the experiments were put in practice, and the Performance field which represents the performance of the methodology in detecting and/or defending against LDoS attacks.

A. SHREW ATTACK

This section was divided into 5 parts, which are: simulated, emulated, real environment, traffic traces and multiple testbeds. Part 1 is related to research that simulates software such as ns-2, ns-3 etc as the testbed. Part 2 is related to research that uses emulated software like netkit, CORE (Common Open Research Emulator) etc as the testbed. Part 3 is related to research that uses real environments such as a lab., university infrastructure etc. Part 4 is related to research that uses traffic traces collected in the internet as the basis for the traffic classification. Part 5 is related to research that uses multiple testbeds.

1) SIMULATED

Guang *et al.* in [61] decided to randomize the TCP RTO (Retransmission TimeOut) with the aim of eliminating the future prediction of the synchronized next round of the timeout value by the attacker. They specified 3 value ranges to randomize the retransmission timeout, which worked very well in preventing the attack from learning the next RTO value as well as preserving TCP fairness against the shrew attack.

In order to detect shrew attacks Haibin *et al.* in [62] chose to install detection mechanisms in the routers that were 1 hop away from the victim. The routers sniff the input port searching for anomalies in the network traffic using the DTW (Dynamic Time Wrapping) method for that purpose. This method works by comparing the similarity among the signatures of the attack with the input signal extracted from the network traffic. Once the attack is confirmed, the attacked router sends back the detection to upstream routers connected to the input port, trying to detect the attack as close to its malicious source as possible. The results showed that the detection mechanism is accurate and robust in detecting the attack.

Yu-Kwong *et al.* in [63] chose to detect shrew attacks in the bottleneck link router, using a method that they called HAWK (Halting Anomaly with Weighted Choking). This

method used a flow table that maintained the statistics about the flows which were marked as ''potentially malicious'' or ''confirmed malicious.'' Each incoming packet was checked and if it matched some flow in the table entry its statistics were updated. After that, the router's queue was analyzed and if it was larger in size than the maximum threshold the incoming packet was dropped, but if the size was within the threshold averages (both minimum and maximum), then the packet was admitted with a P probability, otherwise the packet was truly admitted. The results showed that HAWK outperformed other router AQMs and provided an adequate experience for the legitimate user traffic.

Detecting shrew attacks is not an easy task; therefore Yu *et al.* in [64] developed an algorithm by analyzing the frequency-domain characteristics, aiming at filtering, identifying, and detecting the attacks, thus preventing the legitimate traffic from being dropped. For this task, the authors considered the number of packets caught by the signal, sampling them in every 1 ms and then converting the time-domain series into their frequency domain representation using DFT (Discrete Fourier Transform). After that the NCAS (Normalized Cumulative Amplitude Spectrums) of TCP and shrew flows were compared. A flow was considered legitimate if the value of NCAS was greater than the threshold, otherwise the flow was considered as an attack. The results showed a high accuracy in detecting the attack.

Amey *et al.* in [65] based their detection mechanism against shrew attacks in the RTT (Round Trip Time) traffic. The detection module was hosted in the edge routers and computed the RTT average high and low of the flows in both directions. Additionally, the RTO period was computed and estimated. If the average RTT high repeated periodically and the RTO estimated was second, then the traffic collected was malicious. The results were satisfactory and the module was easily implantable.

Shrew attacks can be defeated by increasing the buffer size of the queue routers. Sandeep and Andreas in [66] concluded this using a mathematical model that calculates the amount of the packets in the router's queue based on the burst traffic arriving to the victim. The results showed that the attacks can be mitigated by increasing the router buffer.

Amey *et al.* in [67] showed through experiments that the quality of voice services was compromised by Shrew and RoQ attacks, eliminating the premise that these attacks were only effective against applications that used TCP protocol, since VOIP (Voice Over Internet Protocol) applications used UDP (User Datagram Protocol) traffic. Based on that they decided to use the FEC (Forward Error Correction) model with Reed-Solomon code to recover the VOIP traffic from packet loss. The results showed that the FEC with RS(3,2) was effective in detecting the anomalies inside the legitimate traffic.

Shrew attacks remain as a potential strike against costumer's services. So, Zenghui and Liguo in [68] suggested to detect this attack based on a threshold, which was composed by the average of the percentage of the attack divided by the legitimate packets in the router's queue. The results were satisfactory in detecting the attack.

Changwang *et al.* in [69] proposed a structure that filtered and detected Shrew attack packets passing through the route, which they called RRED (Robust RED). The idea was to detect the attack packets according to the time between arrivals, based on the dropped packets using RED algorithms. A packet arriving at the server queue would be considered suspicious if it arrived within a brief time interval after a packet coming from the same flow was dropped by the Detection and Filter block or after a packet belonging to any flow that was dropped by the RED block. The results showed that the tool was able to improve the TCP traffic under shrew attack.

In order to mitigate the impacts of shrew attacks against the target, Huaping *et al.* in [70] made a comparison between two queue managements, RED and Droptail. They used both in the same scenario conditions. The results showed that the greater the distribution of the attack nodes, the greater was the effectiveness of droptail defense and the worse the effectiveness of the RED.

Kumawat and Meena in [71] created a framework for detecting LDoS attacks. First of all the traffic was characterized as malicious or normal through the Entropy attribute. If the Entropy was greater than the threshold, then the traffic would be considered malicious, otherwise, the traffic would be regarded as normal. Next, in order to detect the type of the attack the traffic entropy was compared to a threshold. If the Entropy was high it would be marked as a High-rate attack; if the Entropy was low it would be marked as a Low-rate attack, otherwise it would be marked as normal. Finally, the attack was mitigated based on the flow ID of the traffic. The results were successful in detecting the attacks.

Detecting shrew attacks in a MANET (Mobile Ad-hoc Network) is a difficult task. With that in mind, Singh *et al.* in [72] proposed a mechanism that set the congestion bit based on three status values, these being frequency of receiving RTS/CTS packets, frequency of sensing a busy channel, and the number of RTS/DATA retransmissions, which were observed by a passive server. It verified whether the three status values exceeded the limit values set by the threshold. If that were true, the traffic would be marked as an attack and all the traffic would be blocked or rejected. The results showed a reduction in both the attack and the packet loss of legitimate users.

A detection mechanism called SEDP (Spectral Energy Distribution Probability) was created by Wu *et al.* in [73] to detect shrew attacks. The idea behind the approach was to work with the signal generated by the amount of attack and legitimate traffic. The authors treated a shrew attack as a short signal and the TCP traffic as a long signal. Based on that it was necessary to change the time domain to a frequency domain using Fourier transform. Next, the spectral energy of legitimate and attack traffic was calculated in such a way that if the energy exceeded the threshold the alarm would be

triggered. The results showed better accuracy in detection and less consumption in computation.

An extended approach based on Xiang *et al.* in [74] was made by Sahoo *et al.* in [75], using the union of the Generalized Entropy (GE) and Generalized Information Divergence (GID) metrics. In order to compute the flow entropy it was verified whether the packet window sizes of the traffic flow were equal to 80; if that was the case, then the destination IP and occurrence were computed in a hash table. This action also computed the information distance among the flows. After computing the GE and GID from the flows, the values generated were compared to the set thresholds. If the values were found to be greater than or equal to the threshold, then the counter parameter increased by 1. If the counter parameter was equal to 5, then the flow was marked as an attack. The results made it possible to identify the attack traffic and the legitimate traffic with an improved false negative rate.

Zhang *et al.* in [76] proposed detecting the shrew attack based on the extension of the PCA algorithm called adaptive KPCA (Kernel Principal Component Analysis). First, all the sample data were extracted from the wavelet multi-scale analysis and then the RBF parameters (Radial Basis Function), number of main components, and SPE (Squared Prediction Error) trust limit were trained and adjusted regarding the network environment. If the values of the parameter exceeded a certain threshold, then an attack alarm would be triggered. The results showed a 99.2% accuracy with a 0.8% false negative rate and 2% false positive rate.

Liu *et al.* in [77] created a detection mechanism that classifies the network traffic based on frequency-domain, grouping them in clusters based on NCAS features. The proposal was to separate the sampled traffic in clusters, using the BIRCH algorithm for this and then blocking the clusters with NCAS values above the threshold. The results showed an accuracy beyond 70% and a rapid response time.

A defense/detection mechanism called FRRED (Fair Robust Random Early Detection) algorithm was developed by Lin *et al.* in [78]. The proposal for detecting the attack was to record the packet drop time, checking if the incoming packet belonged to the same flow of the packet dropped and if it arrived in a short-range time after the dropped one. Based on that, the indicator labeled as f.I was used to classify the flow as ''under attack'' or ''legitimate,'' which meant that if the verified packet was under attack, then the f.I would decrease by one, otherwise it would be increased by one. The results showed an effective throughput and fairness for TCP flows and mitigation for attack flow traffic.

Huang *et al.* in [79] used the CCID (Cross-Correlation Identity Distinction) method with the aim of identifying the stealth Shrew attack traffic mixed with the legitimate traffic. For this task the authors converted different data flows into the appropriate time domain signal and frequency domain. After that the cross-correlation of both domain type was calculated. Next, the traffic with high cross-correlation coefficient was marked as an attack traffic. The results showed that in both domain type the CCID was effective in detecting the attack.

In order to detect and filter Shrew attacks, Şimşek and Şentürk in [80] based their study on the congestion queue and standard deviation of the traffic in the router. For detection, the main difference from other AQM-based researches was that they based it on the pre-congestion period of the router queue. They assumed that the attack would begin after a packet was dropped. In order to filter the attack traffic they used the standard deviation method for the variation in time of the packets arrival. The results showed an efficient detection with no false positive and negative rates.

In 2018, Siracusano *et al.* [81] have proposed a methodology for the detection of LDDoS attacks based on the characteristics of malicious TCP flows. They have used two datasets: one generated from a simulated network, the other from the publically available CIC DoS dataset. Both contain the attacks slowread, slowheaders and slowbody, alongside legitimate web browsing. They have extracted TCP flow features from all connections and have shown that decision trees and kNN supervised algorithms accurately classified up to 99.99% of flows. In the same year, Cotae et Rabie [82] proposed a game theoretic approach to detect LDoS attacks. They simulated network congestion attacks and created a threshold bandwidth filter at the router that allows a specific bandwidth. They considered the game players in a static simultaneous game and found the Nash Equilibrium where players do not have any profit. They concluded that a mixed strategy will the best response for an organization using this approach.

In 2021, Liu *et al.* [83] have developed a novel semi-supervised locality sensitive incremental transductive support vector machine (LS-ITSVM) method. Their method incorporates local frequency-domain features from the autocorrelation sequence of network flows into the regularization time-domain framework of TSVM. Simulation results show an higher detection accuracy of abnormal network flows, a faster training and better response times.

2) EMULATED

The detection approach against Shrew attack by Kaur and Agrawal in [84] involved two phases. The first was to generate a log file with the traffic network (attack + legitimate) as the input data. The second was to apply the detection attack into the log file aiming at analyzing how accurate the suggested method was. The chosen method was based on a modified version of the CDA (Changepoint Detection Algorithm), which they called QCD (Quickest Changepoint Detection) algorithm. The algorithm has the aim of observing any small changes in the traffic probability deviating from the previous normal behavior and locating in real time the exact moment where the anomaly change occurs.

3) REAL ENVIRONMENT

Ying *et al.* in [85] described how the BGP (Border Gateway Protocol) could be affected by shrew attacks as well as techniques that mitigated the damage generated by this attack. The first thing they did was to hide information such as the

min RTO value from the BGP packet, and the second thing was to guarantee the bandwidth and to prioritize scheduling for BGP traffic using for this task a set of solutions as filters and queuing methods, e.g., WRED (Weighted RED). The results were effective in prevent the attacks.

In order to detect shrew attacks, Gautam *et al.* in [86] made a comparison between two detection techniques called MAD (Modeled Attack Detector) and the modified PAD (Periodic Attack Detector) based on Xinming *et al.* in [87]. The MAD technique operated on the sampled time-series of network traffic, whereas PAD worked in the spectrum of network traffic. The results showed that MAD was able to detect attacks faster than PAD.

Yang *et al.* in [74] proposed a combination of two information metrics for shrew attack detection, generalized entropy and information distance. These metrics were compared to two known approaches, Shannon entropy and Kullback-Leibler distance. The detection process started by collecting all data traffic passing through the routers 2 steps away from the victim and then it calculated the distribution probability from all the sampled data. After that the values were sent to routers 1 step away from the victim and their distances were calculated. The DDoS attack was detected in case the distance summed was greater than the threshold. The experimental results showed an effective detection and low false positive rate.

The detection system created by Rejo and Vijay in [88] used IDS (Intrusion Detection System) software in addition to a software-based approach to detect shrew attacks. The software worked in the first moment, collecting the data traffic and calculating the average traffic data, number of timeouts and the amount of dropped packets. In the second moment, it used the values generated in the previous step in order to calculate the current and average inter-arrival time in the time slot. After that, it verified whether the values were higher than the threshold; if that was true, the traffic would be an attack. The results were shown to be effective in detecting the attacks.

In order to keep the target of the shrew attack from being overwhelmed, Wu *et al.* in [89] used the PCA (Principal Component Analysis) algorithm to define the Open Supervised Device Protocol matrix of a normal flow, and one under attack. The flow ID is composed of source address, source port, destination address, and destination port. Based on that, a waveform of the normal and attack traffic can be drawn to select the appropriate threshold. That way it is possible to detect the attack with good performance and achieve a high detection rate, low false alarm, and low missed alarm probability.

In order to detect shrew attacks in SDN (Software Defined Networks) networks, Agrawal and Tapaswi in [90] created a defense scheme containing the detection of the attack through entropy and tracing back the traffic attack to its origin. The entropy mechanism was used for detecting the attack, which was able to differentiate the attacked traffic from the traffic without attack. After that, with the attack source IP addresses in hand it was possible to verify the origin of the attack using a DPM traceback scheme, blocking the attack using ACL (Access Control List) and SDN flow-table rules. The results showed a 97.6% accuracy in detecting Shrew attacks.

Boro *et al.* in [91] designed a mechanism that rapidly detects the shrew attack in the traffic inflow. For this purpose they used the SSM (self-similarity matrix) across multiple time scales, computing the similarity between pairs of features, which are Average packets per network flow, Number of packets per interval or sample, Number of network flows and Server outflow performance in a set of time-ordered data samples. The results showed that the mechanism was effective in terms of accuracy and computation speed.‘

Tang *et al.* in [92] developed a framework called P&F (Performance and Features) for real-time LDoS attack detection in SDN networks. The framework is divided into three parts, which are: Feature Extraction and Classification, Attack Detection and Attack Mitigation. The Feature Extraction and Classification module has the aim of extracting features from the perfomance of the TCP traffic under LDoS attack and from the characteristics of the LDoS attack traffic. Next, the Attack Detection module has of aim of detecting anomalies based on the features collected from the Feature Extraction and Classification module. Finally, the Attack Mitigation module will locate the source IP of attackers and the victim ports, based on the rules that it sends according to the locating result to filter LDoS attack traffic. The results showed a detection accuracy performance of 96%.

4) TRAFFIC TRACES

Bhuyan *et al.* in [93] experimented with information metric theory algorithms, namely Hartley entropy, Shannon entropy, Renyi's entropy and Generalized entropy with the aim of quantifying their success in detecting shrew attacks. Basically, the data was sampled into 10s period received by the upstream routers and then the distribution probability was calculated based on the flow ID (Source IP, Destination IP and Protocol) to generate the Entropy. And finally the value obtained was checked in order to verify whether it was greater than the threshold. If true, the flow would be marked as an attack, otherwise the flow would go to the next router. The results were effective in detecting the attack.

Bhushan and Gupta in [98] based their detection tool on the hypothesis test defining the H_0 as $\sigma_N = \sigma_R$ which represented the legitimate traffic and H_1 as $\sigma_N > \sigma_R$ which represented the attack traffic. The edge routers would sample the traffic arriving to them and calculate the standard deviation σ_R based on the packet size of each packet and a threshold. After that the hypothesis was tested in order to verify whether the traffic was legitimate or not. The results showed effective detection by the tool.

5) MULTIPLE TESTBEDS

Zhijun *et al.* in [94] detected attacks using a comparison between the similarity of the signal from a synthetic shrew attack and the heterogeneous signal composed of the TCP

TABLE 3. Comparison among detection/defense mechanisms against Shrew attacks.

Work	Used approach for traffic classification	Detection/Defense mechanism	Detection/Defense location	Testbed	Performance
Guang et al. in [61]	-	RTO randomized	Victim	Simulated	Effective
Haibin et al. in [62]	Auto-correlation plot signature of the traffic attack	Dynamic Time Wrapping	Upper routers one hop away from the victim	Simulated	-
Yu-Kwong et al. in [63]	Average queue size of the router	Halting Anomaly with Weighted choKing	Border router	Simulated	HAWK outperforms other router's AQM and provide a fair experience to the legitimate users traffic
Yu et al. in [64]	Flow ID (IP source address, IP destination address and IP destination port)	Normalized Cumulative Amplitude Spectrum	Server	Simulated	High detection accuracy was achieved using a collaborative distributed detection mechanism
Amey et al. in [65]	Flow ID (IP source address, IP source port, IP destination address and IP destination port)	Average RTT and estimated RTO of the traffic	Edge router	Simulated	-
Sandeep and Andreas in [66]	-	Increase buffer size of the routers	Border router	Simulated	-
Ying et al. in [85]	-	Hide BGP topology information and prioritize routing traffic	Border router	Real environment	Effective
Gautam et al. in [86]	Sampled time-series of network traffic	Modeled Attack Detector	-	Real environment	-
Amey et al. in [67]	-	FEC model with Reed-Solomon code	-	Simulated	The FEC model with RS(3,2) is effective in detect the anomalies inside the legitimate traffic
Zenghui and Liguo in [68]	-	The rate of the dropping packets	Border router	Simulated	-
Changwang et al. in [69]	Flow ID (IP source address, IP source port, IP destination address, IP destination port and Protocol)	RRED	Border router	Simulated	Is highly robust, can significantly improve the performance of TCP under attacks, outperforming existing RED-like algorithms
Huaping et al. in [70]	-	Droptail and RED	Border router	Simulated	The deeper degree of distribution, the better defense performance of Droptail (passive queue management) and the worse defense performance of RED
Yang et al. in [74]	IP characteristics	Generalized entropy and information distance metric	Border router and routers from 1 and 2 hops away from the victim	Real environment	Effective detection and low false positive rate
Rejo and Vijay in [88]	Traffic inter-arrival time and dropped packets	Software-based approach integrated with existing Intrusion detection system	Detection server	Real environment	-
Zhijun et al. in [94]	The correlation between the signal of simulated shrew attack and the hybrid signal (TCP flow plus shrew attack)	Spectral intervals	Border router	Real environment and simulated	False negative alarm rate is less than 1.1% False positive alarm rate is less than 0.8% The detect rate is more than 98%
Kai et al. in [95]	Variation of TCP traffic	Exponential Weighted Move Average	Border router	Simulated and traffic traces	Effective detection with low false positive rate
Changwang et al. in [96]	Congestion router queue and drop packets	Congestion Participation Rate	Border router	Simulated, real environment and traffic traces	Effective
Kumawat and Meena in [71]	Entropy	Low false positive rate	Intermediated routers	Simulated	Effective
Singh et al. in [72]	Frequency of receiving RTS/CTS packets, frequency of sensing a busy channel and the number of RTS/DATA retransmissions	Status values threshold	Passive server	Simulated	Reduce attack throughput there by increasing the received bandwidth and reducing the packet loss of legitimate users
Wu et al. in [89]	Flow ID (IP source address, IP source port, IP destination address and IP destination port)	PCA algorithm	Firewall	Real environment	High detection rate, low false alarm and low missed alarm probability
Bhuyan et al. in [93]	Flow ID (IP source address, IP destination address and IP destination port)	Entropy methods	-	Traffic traces	Effective
Wu et al. in [73]	Network traffic	Spectral Energy Distribution Probability	Victim	Simulated	The detection accuracy is 100% with low false positive and negative rate
Wu et al. in [97]	Holder parameter	Multifractal analyzes	Victim	Real environment and simulated	Simulation achieved 92% of accuracy and False positive rate of 9% while the testbed environment achieve 91% of accuracy and False positive rate of 10% and low complexity
Sahoo et al. in [75]	Destination IP and occurrance	Generalized Entropy and Generalized Information divergence metrics	Controller	Simulated	Improved false negative rate
Zhang et al. in [76]	Network traffic	Kernel Principal Component Analysis	-	Simulated	99.2% of accuracy with 0.8% of false negative rate and 2% of false positive rate
Kaur and Agrawal in [84]	Network traffic	Changepoint Detection Algorithm	-	Emulated	-
Agrawal and Tapaswi in [90]	IP source and IP packet size	Entropy and IP traceback mechanism	SDN Controller	Real environment	97.6% detection accuracy
Boro et al. in [91]	Network traffic	Self-Similarity Matrix	Firewall	Real environment	Effective
Liu et al. in [77]	Frequency-domain	Network flow grouping method + NCAS	Border router	Simulated	Detection accuracy greater than 70%
Lin et al. in [78]	Flow ID (IP source address, IP source port, IP destination address, IP destination port and Protocol)	FRRED	Border router	Simulated	Effectively preserve the throughput of TCP flows Significantly improve fairness among TCP flows Effectively mitigate address-spoofing LDoS attacks
Huang et al. in [79]	Frequency domain Time Domain Signal and	Cross-Correlation Identity Distinction	Victim	Simulated	Effective
Şimşek and Şentürk in [80]	Router queue and delay time	Precongestion period and arrival times of packets	Border router	Simulated	Zero false positive and false negative rates
Bhushan and Gupta in [98]	Packet size	Hypothesis test	Edge routers	Traffic traces	Effective
Tang et al. in [99]	Variance and mean deviation parameters	SADBSCAN algorithm and cosine similarity	Victim	Simulated, real environment and traffic traces	Improve the detection accuracy and reduce the false negative rates
Tang et al. in [92]	Time-frequency analysis, packet transfer speed, average packet transfer speed, average traffic speed, packet variance, packet standard deviation, coefficient of variation of packet and average packet size	P&F framework	Target switch	Real environment	96% of detection accuracy

flows plus the shrew attack traffic. The results of the calculation were compared to the correlation value based on a threshold, which was calculated based on experiments. If the value of the similarity overstepped the threshold, the traffic would be classified as an attack. The results had high accuracy and performance.

A Shrew attack is an anomaly traffic, which can be defeated. The authors Kai *et al.* in [95] showed that the attack can be detected based on the abnormal phenomena of network traffic. First, all the traffic data is sampled and next, the EWMA (Exponential Weighted Move Average) algorithm generated statistics about the hybrid traffic. Second, the detection method observes the behavior of the traffic distribution generated by EWMA algorithm and makes various judgment criteria. If all criteria match in the time window then the attack will be detected. The results were effective in detecting attacks on the traffic.

Changwang *et al.* in [96] proposed a measurement defined as CPR (Congestion Participation Rate) for detecting and filtering shrew attacks. The measurement was based on the proportional relationship between congested incoming packets versus the total incoming packets in the stream. If in this proportional relationship there was a CPR greater than the predefined threshold, then the flow would be considered an attack and all packets would be dropped. The results were shown to be effective against the attacks.

Wu *et al.* in [97] proposed an approach based on a multi-fractal analysis of network traffic. Protocols such TCP, IP and HTTP have this characteristic, while UDP is a monofractal. The approach needed to set the Hölder exponent parameter in order to detect the attack. Therefore, the smaller the exponent α was, the more confident would be the result that the traffic was malicious. The results based on the simulation achieved 92% of accuracy and a false positive rate of 9% while the testbed environment achieved 91% of accuracy and a false positive rate of 10%.

Tang *et al.* in [99] proposed a Low-rate DoS attack detection mechanism that adaptively identifies traffic attacks in clusters of data in multidensity datasets. To achieve this goal the authors created an algorithm called SADBSCAN (Self-Adaptive Density-Based Spatial Clustering of Applications with Noise) which has the aim of improving noise and data distribution as well as mitigating the effects of data class imbalance produced by classical clustering algorithms. To detect the LDoS attack the data traffic is first split into equal amounts of multiple data units where in each unit the variance and mean deviation parameters are extracted from the TCP and UDP traffic to be used as eingevalues into the SADBSCAN algorithm. Next, the cosine similarity is introduced to label the clusters and noise points resulting from SADBSCAN algorithm. This label could be attack traffic or benign traffic based on the value threshold provided. The results showed that the detection mechanism improves accuracy in distinguishing Low-rate DoS attacks from legitimate traffic, which reduces the false negative rates.

B. RoQ ATTACK

The detection mechanism provided by Arunmozhi and Venkataramani in [40] was an integrated scheme in that the intermediate nodes kept on sending an FMT (Flow Monitoring Table) file to the destination mode, which contained the flow ID, source ID, packet sending rate, and destination ID. If a node was experiencing a congestion, it would send a packet to the destination node with the CE (congestion experienced) bit marked in the IP header. That way, the destination node that received the flow with the CE bit marked would send a control warning to the source asking it to diminish the sending rate. If the source node did not stop the flow, then it would be considered an attacker and all the flow belonging to it would be discarded.

Guirguis *et al.* in [42] studied the impact of the RoQ attacks on admission controllers and load balancing servers. They observed that the attack could be very powerful in disrupting the efficiency of the servers. Based on that, an admission ratio β parameter value has been inserted in order to be dynamically adaptive, which was based on the equation:

$$\alpha^n(i) = \frac{1}{N} + \beta \sum_{j=1}^{N} (q^j(i-a) - q^n(i-1)) \tag{1}$$

Setting the β value to 0.03 the load balancers might display a quicker reaction to the changes in traffic, keeping services available. The admission controllers' K parameter has the same effect.

Ren *et al.* in [100] created a defense scheme with the aim of avoiding a decrease in quality of data related to voice and video traffic in a MANET (Mobile Ad-hoc Network). This scheme was divided into 2 parts. The first part acted in the MAC layer by keeping track of the frequency and retransmission of the packet flow. After that, the monitored values were analyzed and if they exceeded a certain threshold an alert would be triggered. The second part of the defense scheme was indicating the packets with the congestion bit field and warning the sender nodes about the congestion problem. If they did not reduce their transmission data, they would then be considered as attacking nodes. The results showed that the delay and decrease of traffic quality was proportional to the sum of attack traffic flows.

Shevtekar and Ansari in [101] proposed a detection environment system that was sectioned into two phases. The first or detection phase started with the sudden increase of the packets above a certain virtual queue threshold. Next, all the new arriving packets were analyzed to identify whether they belonged to the benign flow table. If that was true the packets would be forwarded to the attack filtering, otherwise they would go to the normal path. In the attack filtering stage, the algorithm checked how long it took for each packet to arrive in each flow, thus examining the disparity in time between each arrival. If the time difference between those arrivals was very small, the detection mechanism module would perform a packet load count, and in case the valued exceed a certain threshold, the result would be that the

attack was detected. Once the attack was detected, malicious packets were dropped. The detection system for RoQ attack and Low-rate DoS attack successfully achieved its purpose. Chen and Hwang in [102] designed a detection scheme that differentiated a normal TCP flow from a RoQ attack flow by making use of spectral analysis. The scheme used the hypothesis testing method in order to differentiate the traffic under attack from the legitimate one based on the spectrum of the flow in the frequency band. The results showed that the scheme was able to cut off RoQ attack flow and effectively save 99% of legitimate TCP flows.

Gulati and Dhaliwal in [103] began the detection system by choosing the computer selected to be the attack monitor. After the selection, if the elected computer node verified that, within a small period of time above the threshold there was a rapid increase in the amount of traffic, the flow would be placed on a suspect list. If any suspect node appeared a large number of times, it would be added to a checking table list. This table list was later sent to every node within the network environment whose amount of traffic passing through was above a certain threshold throughout a specific time period. If this was true, then the node would be placed in the attacker table; otherwise, it would be dropped from the suspect table list. Each node listed in the ''attacker table list'' was to be blocked. The results showed that it was viable to diminish packet loss and to have better throughput rates.

Wen *et al.* in [104] created a detection system divided into two stages. the first stage was to analyze suddenly anomalies in the collected traffic data. Next, the sampled traffic was sliced by time and checked by the CUSUM method. If any slice showed any anomaly, then the second stage was removed. At this point an auto-correlation analysis was used to verify the signal periodicity. If it was consistent along with time, an alarm would be triggered confirming that the RoQ attack was present. The results showed an accurate detection scheme with low false positive and false negative rates.

Gang *et al.* in [105] tested the network traffic performance of the Mobile Internet Protocols such as MIPv4, MIPv6 and FMIPv6 against the RoQ attack. In this scenario the proposed detection solution used the Hamilton-path-based scheme as the main factor to decrease the packet loss of the legitimate traffic nodes. The aim was to create a reserved bandwidth in the overlay networks to allow the legitimate traffic to be forwarded to the destiny. As the results showed, without the detection solution the mobile protocols suffered more damage due to the amount of traffic passing throw them. On the other hand, with the detection scheme, the packet loss and delay decreased in the overlay networks.

Hongsong *et al.* in [106] explored three methods (Ensemble Empirical Mode Decomposition (EEMD), Correlated Coefficient Method and Hilbert-Huang Transform (HHT)) in order to create a detection mechanism that could eliminate issues that were likely to be related to the signal generated by the RoQ attack, such as mode mixing and fake components as well as analyzing and comparing the time–frequency distribution of routing message traffic. Hence, based on the Intrinsic Mode Function (IMF) spectrum of the traffic generated by the detection scheme, it was easy to detect the attack. The results showed a very accurate detection of the RoQ attack traffic.

In order to detect RoQ attacks, Rios *et al.* in [107] proposed a detection mechanism which was composed of fuzzy logic, neural network and Euclidean distance. All these methods used a training data set as the basis for classifying the data as attack or legitimate. The aim was to classify the data using fuzzy logic and neural networks and then to compare the classification generated by both. If the same data values had different classifications, the Euclidean distance was used to check them with the training data set classification. After that, the result of each classification method was compared and if there were more attack classifications than legitimate ones, the data would be classified as an attack, otherwise it would be classified as legitimate. However, if there was a tie, the data would be classified as a warning that would require a closer inspection by the network administrators. The results showed an excellent performance with 100% of accuracy in the emulated environment and 99.3% of accuracy in the real environment in detecting the attack traffic.

Liu *et al.* in [108] created a Low-rate detection mechanism which was compounded by the union of the self-adjusting SVM method with the APSO (Adaptive Particle Swarm Optimization) algorithm. The self-adjusting SVM method has the aim of enhancing the generalization ability of the pure SVM algorithm and it was made using 2 approaches, which are: the adjustment of the gamma parameter and the adoption of the optimal degree value from the training data. The APSO algorithm was used to enhance the adjustment of the attack and background traffic band average values and the energy intensity of the attack flow sub-band parameters. The results showed excellent detection performance with the accuracy ranging from 92.36% to 96.65%.

C. SLOWLORIS ATTACK

Aiello *et al.* in [109] proposed a Statistical Based Intrusion Detection (SBID) approach, which had the aim of detecting slowloris attacks based on the probability distribution of the parameters Δrequest, Δresponse, Δdelay and Δnext for two distinct traffic $f(x)$ and $g(x)$ belonging to the same network situation. If the two traffics had distinct distributions, the alarm was triggered. The results showed 100% of anomaly detection, with 1% probability of false positive rate.

In order to detect and soften the application DoS attack slowloris, Dantas *et al.* in [110] proposed a defense mechanism that they called SeVen (Selective Defense for Application Layer DDoS Attacks). It assumed state-dependent protocols and used the notion of state in the messages of the HTTP protocol. The software did not immediately respond to requests received by the application, but instead waited for a period of time called the round. During a round, the messages were accumulated in an internal buffer and when the buffer reached its limit, it was decided whether a new request would be accepted or not, based on certain criteria. It is important

TABLE 4. Comparison among detection/defense mechanisms against RoQ attacks.

Work	Used approach for traffic classification	Detection/Defense mechanism	Detection/Defense location	Testbed	Performance
Arunmozhi and Venkataramani in [40]	-	Flow Monitoring	Destination node	Simulated	Achieves higher throughput and packet delivery ratio
Guirguis et al. in [42]	-	In the admission controller's equation Dynamically adapt beta and k values	-	Simulated	-
Ren et al. in [100]	-	Based on the threshold of the frequency of receiving RTS/CTS packets, frequency of sensing a busy channel, and the number of RTS/DATA retransmissions	-	Simulated	-
Shevtekar and Ansari in [101]	Flow ID (packet count, packet size, createtime and lastaccessed time)	Router-based approach	Routers close to the victim	Simulated	Can successfully detect and mitigate RoQ attacks
Chen and Hwang in [102]	Flow ID (IP source address, IP source port, IP destination address and IP destination port)	Flow-level spectral analysis with sequential hypothesis testing	-	Simulated	Effectively rescue 99% of legitimate TCP flows
Gulati and Dhaliwal in [103]	-	Flow Monitoring Table	Chosen attack monitor	Simulated	Reduce packet loss and improves throughput
Wen et al. in [104]	Abruptly traffic changes	CUSUM method with auto-correlation analysis	-	Simulated and Real environment	High accuracy and high Effective
Gang et al. in [105]	-	Hamilton-path-based scheme	-	Simulated	Reduced packet loss
Hongsong et al. in [106]	Intrinsic Mode Function spectrum	Hilbert-Huang Transform with ensemble empirical mode decomposition and Correlated coefficient method	-	Simulated	Highly effective
Rios et al. in [107]	Flow ID (IP source address, IP source port, IP destination address, and IP destination port)	Fuzzy logic, Neural Network and Euclidean distance	Victim	Emulated and Real environment	100% of accuracy in the emulated environment and 99.3% of accuracy in the real environment
Liu et al. in [108]	Average values of the attack and background traffic bands and energy intensity of the attack flow sub-band	Self-adjusting SVM method with APSO algorithm	Victim	Emulated	Higher detection accuracy, which ranges from 92.36% to 96.65%

to notice whether the new request was accepted or not, based on the probability, since the attack packets were more likely to be dropped. The results showed high levels of availability, greater robustness, and they also led to less traffic than other techniques.

In order to detect slow DoS attacks, including the slowloris attack, Mongelli *et al.* in [111] proposed a detection method with the aim of analyzing specific spectral features, such as the number of received packets of traffic over small time horizons. The time horizon analyzed the current and previous temporal packets behavior in order to see significant changes in both traffic. Hence, in order to identify the attack they used the Fast Fourier Transform in the current time horizon in frequency domain. The attack was detected due to smoother behavior of the legitimate traffic.

Katkar *et al.* in [112] configured an IDS software along with Naive Bayesian algorithm with the aim of quickly detecting the slowloris attack. To validate the results they created a test-bed environment that was composed of web servers clusters, due to the fact that this approach has benefits for the web server application. All the traffic arriving to the web servers was collected and formatted for a traffic data file type and transmitted to be analyzed by the IDS software. It categorized the received aggregated records into normal or intrusive ones. Whenever a record was categorized as intrusive, it was an indication that the server was under DoS/DDoS attack.

Based on the HTTP traffic request and response, Aqil *et al.* in [113] determined the optimum detection high and low threshold based on features for detecting slowloris attacks. Basically, the detection algorithm checked every 3 seconds if there were feature values above or below the determined thresholds. If that was the case, the algorithm warned of an attack. The results were shown to be an extremely effective approach.

Hirakawa *et al.* in [114] based the detection scheme on 3 steps. The first was to check whether the numbers of connections that were monitored were greater than a threshold during a normal time. If the threshold was trespassed, then in step 2 all the IPs belonging to the same flow and that were appearing most frequently would be disconnected. In step 3, if the connections were above the threshold, that stopped the disconnection process and all the cycles started over. The results showed enough resistance against the strikes from a single attacker as well as from distributed strikes from 30 attackers by calibrating the threshold appropriately.

In order to detect the slowloris attack Singh and De in [115] created a scheme by uniting the genetic algorithm method, which was used to improve the MLP Neural Network weights and the MLP Neural Network algorithm. They made a comparison between the proposal with other machine learning algorithms such as Naive Bayes, Radial Basis Function (RBF) Network, MLP, J48, and C45. In the tests the authors' algorithm was better at detecting the slowloris attack, reaching 98% of accuracy.

Faria *et al.* in [116] developed a tool which they called SDToW (Slowloris Detecting Tool for WMNs) that detected and blocked slowloris attack traffic in wireless mesh networks (WMNs). First, they analyzed the slowloris traffic behavior. After that, they based the detection on three information items, the HTTP GET method, The Reassembled PDU, and the Packets with 296 bytes and TCP set on protocol field. In every 5 seconds, traffic was collected by the Collection Module (CM) in the web server and it was sent to the concentrator where the Analysis and Filtering Module (AFM) selected the connections with the information listed above. If there was traffic within these three information inside the traffic connections, so there was an ongoing attack and the IP and MAC addresses of the connections would be extracted

TABLE 5. Comparison among detection/defense mechanisms against Slowloris attacks.

Work	Used approach for traffic classification	Detection/Defense mechanism	Detection/Defense location	Testbed	Performance
Aiello et al. in [109]	Probability distributions parameters Δrequest, Δresponse, Δdelay and Δnext	Statistical Based Intrusion Detection	Victim	-	100% of anomalies detection, with 1% probability of false positive rate.
Dantas et al. in [110]	-	SeVen	Victim	Simulated	High levels of availability,greater robustness and also leads to less traffic
Mongelli et al. in [111]	-	Fast Fourier Transform	Victim	Traffic traces	Frequency domain
Katkar et al. in [112]	-	IDS with Naive Bayesian classifier	Victim	Traffic traces	IDS signature
Aqil et al. in [113]	The volume of data sent, the volume of data received, Interrupts, context switches and TCP Sockets	High and low traffic threshold	Victim	Real environment	The true positive rate is close to 100% and the false positive rate is decreased by about 66% as compared to traditional detectors
Hirakawa et al. in [114]	The number of connections and duration time for each IP address	Amount of connections per IP address	Victim	Simulated	Enough resistance and effective
Singh and De in [115]	The number of HTTP count, the number of the IP addresses, the constant mapping function and the fixed frame length	MLP-GA	Victim	-	Detection accuracy of 98.04%
Faria et al. in [116]	HTTP GET method, The Reassembled PDU and the Packets with 296 bytes and TCP set on protocol field	SDToW	Victim	Real environment	Lower incidence of false positive errors

and included into the blacklist to be blocked. The results showed a lower incidence of false positive errors.

D. NewShrew ATTACK

Cotae *et al.* in [117] based their detection proposal on the Fisher g-statistics test method. In order to detect the attack they set a frequency interval in the range of 0.01Hz to 1.1Hz which was labeled as Shrew frequency attack interval. They performed a simulation of the Fisher g-statistics tests for one and multiple time series. For both time series they considered an attack if the content was in the Shrew frequency attack interval and the p-value was less than 10^{-3}. The results showed effective detection, with all attacks being identified.

Luo and Chang in [35] proposed a detection system based on two stages, having as reference the presence of two traffic anomalies. The first one was the incoming fluctuate data traffic and the second one was the low amount of outgoing ACK traffic. In the first stage it applied the discrete wavelet transform, which was used to monitor these two anomalies. In the second stage the CUSUM algorithm was applied in order to detect both anomalies. The experiments were highly accurate in detecting PDoS attack traffic.

E. PDoS ATTACK

The Vanguard detection system developed by Luo *et al.* in [118] was an extension of the Luo and Chang approach in [35]. This system employed more metrics to detect three traffic anomalies induced by the attackers using a CUSUM algorithm. The first anomaly was the incoming fluctuate data traffic, the second one was the decline of the outgoing ACK traffic, and third, the changes in the distribution of the incoming TCP data rate. The results showed an accurate detection system, capable of detecting in a short-range time a wide range of attack traffic.

F. RUDY ATTACK

Najafabadi *et al.* in [119] made a classification comparison among 3 machine learning algorithms namely K-NN and two forms of C4.5 decision trees (C4.5D and C4.5N). The algorithm classification comparison firstly used a set of 'N' features and then a set of 7 features (selected as the best ones among the set of 'N' features). The first result showed that the features related to traffic size, self-similarity, and speed were very effective in detecting the RUDY attack. The second result showed that in both classification results C4.5N was the best algorithm with low false positive rate among the machine learning algorithms.

G. HTTP/2 DoS ATTACK

Tripathi and Hubballi in [57] mitigated the HTTP/2 DoS attack using the Chi-square test. First, in the training phase the authors collected data from legitimate traffic and then in the testing phase, the current traffic was compared to collected traffic. For this comparison, the Chi-square test was used along with five features that are: settings frame having settings_initial_window_size set to 0, headers frame having end_stream_flag reset, flows having Connection Preface only, headers frame having end_headers_flag reset and Server's settings frames which were not recognized. Therefore, in a ΔT time the current traffic was collected and compared to the training phase traffic and then it was calculated through the Chi-square test. If the obtained calculated value was less than the predefined threshold, the detector did not contain anomalous ?ows. However, if the calculated value was higher than the previously established threshold, the detector contained anomalous ?ows. The results from the 7 scenarios had a recall ranging from 8.33% to 100% and a FPR (False Positive Rate) of 0%.

H. LoRDAS ATTACK

Maciá-Fernández *et al.* in [13] proposed 4 defense mechanisms against LoRDAS attack that were:Random answer instant (RAI), Random service time (RST), Improved Random Time Queue Blocking (IRTQB) and Random time queue blocking (RTQB). The RST method had the aim of randomizing the answer instant response, thus avoiding allowing the attacker to predict the correct moment to send the attack. In the RAI method the authors decoupled the instant method from the enable method, thus avoiding the attacker to event predict the instant method; the enable instant would not suffer

TABLE 6. Comparison among detection/defense mechanisms against NewShrew attack, PDoS attack, RUDY attack, HTTP/2 DoS attack and LoRDAS attack.

Work	Used approach for traffic classification	Detection/Defense mechanism	Detection/Defense location	Testbed	Performance	Attack type
Cotae et al. in [117]	Frequency interval in the traffic range	Fisher G-statistic test	-	Simulated	Correctly identified all attacks	NewShrew
Luo and Chang in [35]	Incoming fluctuate data traffic and the amount of outgoing ACK traffic	Wavelet transform and CUSUM	Victim	Emulated	Very effective	PDoS
Luo et al. in [118]	Incoming fluctuate data traffic and amount of outgoing ACK traffic and distribution of the incoming TCP data rate	CUSUM	SNORT server	Real environment	Effective	PDoS
Najafabadi et al. in [119]	Features: Outbound session convergence, Inbound session convergence, Packets, Bytes, RIOT Bytes, Outbound velocity bpp and Outbound velocity bps	K-Nearest Neighbor and two forms of C4.5 decision trees (C4.5D and C4.5N)	Border router	Real environment	Very good classification performance	RUDY
Tripathi and Hubballi in [57]	Source IP address and source port number	Chi-square test	Victim	Real environment	Recall ranging from 8.33% to 100% for and FPR of 0% for all the scenarios	HTTP/2 DoS
Maciá-Fernández et al. in [13]	-	RST, RAI, RTQB and IRTQB	Victim	Simulated	-	LoRDAS

any damage. Due to RST and RAI limitations related to time, which directly impacted the server behavior the RTQB method was developed with the aim of inserting a blocking interval between the answer instant and the enable instant, thus preventing the attacker from successfully damaging the legitimate data processing and discarding all the requests arriving on it. Due to RTQB discard process, the authors improved it (IRTQB) in a way that the attack process was selectively chosen to be discarded. The results showed an effective defense by the IRTQB method if compared to the other methods.

I. LEVERAGING SDN AGAINST LDoS ATTACKS

In recent years, several papers have leveraged the use of SDN to mitigate LDoS attacks. An SDN controller has a global view of the whole network and can aggregate very fine-grained information on specific flows. As the controller is cloud-based, it can leverage much processing power and memory volume which enables it to use centralized algorithms such as those based on machine learning. In 2019, SoftGuard [120] was proposed to defend against low-rate TCP attacks in SDN. Another framework called Q-MIND [121] was proposed that same year by Phan *et al.* to reduce LDoS attacks bu using reinforcement learning. The following year, Perez-Diaz *et al.* have used six machine learning algorithms [122] in their IDS inside an SDN simulated environment. Still in 2020, Balarezo *et al.* have investigated the feasibility and mitigation of LDoS shrew attacks on the control plane connections of an SDN network [123]. Several other papers related to SDN appeared in 2021. Tang *et al.* proposed a framework based on the histogram-based gradient boosting and finding peaks (HGB-FP) algorithm to detect LDoS attacks and mitigate their influence in SDN in real-time [124]. Ilango *et al.* have designed a deep learning scheme called FFCNN to detect LDoS attacks in an IoT-SDN environment [125]. Lui *et al.* proposed an LDoS attack detection method based on a two-step self-adjusting support vector machine (TS-SVM) [126]. Their proposal was evaluated by simulation and showed an improvement over a traditional SVM approach. Finally, Li *et al.* used a Bat Algorithm associated with a BP Neural Network (BA-BNN) to detect LDoS attacks [127]. As the previous study, the evaluation was performed on a simulated environment using *mininet* and the Ryu controller.

V. TOOLS

This section presents the attack and defense tools used to build most of the LDoS attack scenarios.

A. ATTACK TOOLS

In order to launch the attack traffic towards the victim, a tool specifically developed for this purpose is necessary as can be seen in Table 7. Some of the malicious activities presented here have a software which is open source and can be changed for the most varied purposes. The QoS attack and Service queue attacks presented in Section 3 did not have a software that was specific to the aim of the attack. Usually a CBR (Constant Bit Rate) traffic was used to meet the LDoS attack requirements as depicted in Figure 1. In the simulated environments, most of the papers (if not all) used CBR traffic burst as the attack traffic. In the real environment, most authors did not mention what kind of software they were using and a few others used some stress tool in a CBR traffic burst manner for the attack traffic. One exception was the paper by Rios *et al.* in [107] in which the authors created a script that manipulated the Hping3 tool output flood traffic to meet the RoQ attack requirements. On the other hand the Slow DoS attack had some threats with software developed to fill its DoS requirements. The slowloris attack was the threat with most software developed for damaging the web server services followed by Slow Read, RUDY, SlowReq and Slowcomm, that were: slowloris.pl [128], PyLoris [129], QSlowloris [130], an unnamed PHP version [131], [143], SlowHTTPTest [45], Http bog [132], Torshammer [133], Goloris [134], SlowDroid [54], slowloris.py [135], R-U-Dead-Yet [136], Cyphon [137], sloww [138], dotloris [139] and pwnloris [140]. Table 7 describes the attack tools in detail where the *Year* field indicates the birth of the defense tool. The *Tool name* field indicates the name of the tool. The *Attack mode* field indicates the format of the attack, which are: single

TABLE 7. Comparison among Slow DoS attacks tools.

Year	Tool name	Attack mode		Operating System				Programming language	Interface		Attack type
		DoS	DDoS	Windows	Mac OS	Linux	Android		CLI	GUI	
2009	slowloris.pl in [128]	✓				✓		Perl	✓		Slowloris
2009	Pyloris in [129]	✓		✓	✓	✓		Python	✓	✓	Slowloris
2009	QSlowloris in [130]	✓		✓					✓		Slowloris
2009	unnamed PHP version in [131]							PHP	✓		Slowloris
2011	Slowhttptest in [45]	✓		✓	✓			Python	✓	✓	Slowloris Slow Read
2011	Http bog in [132]	✓		✓				C#	✓		Slow Read
2012	Torshammer in [133]	✓				✓		Python	✓		RUDY
2014	Goloris in [134]	✓			✓	✓		Go	✓		Slowloris
2014	SlowDroid in [54]	✓					✓			✓	SlowReq Slowcomm
2015	slowloris.py in [135]	✓			✓	✓		Python	✓		Slowloris
2016	R-U-Dead-Yet in [136]	✓				✓		Python	✓		RUDY
2018	Cyphon in [137]	✓			✓			Perl		✓	Slowloris
2018	sloww in [138]	✓				✓		JavaScript	✓		Slowloris
2018	dotloris in [139]	✓		✓				C#	✓		Slowloris
2018	pwnloris in [140]	✓		✓		✓		Python	✓		Slowloris

TABLE 8. Comparison among Slow DoS attacks defense tools.

Year	Tool name	Defense attack mode		Operating System			Programming language	Interface		Defense against
		LDoS	LDDoS	Windows	Mac OS	Linux		CLI	GUI	
1998	Snort in [116] and [141]		✓			✓	C	✓		Slowloris
1998	Iptables in [142]	✓				✓	C	✓		Slowloris
2010	Suricata in [141]		✓			✓	C and Rust	✓		Slowloris
2020	SDToW in [116]		✓			✓	Python	✓		Slowloris

or distributed. The *Operating System* field indicates which system is used by the tool. The *Programming language* field indicates which language the defense tool was developed. The *Interface* field indicates which interface type is used to configure the parameters of the defense tool, if it is a terminal type or a graphical type. The *Attack type* field indicates the type o attack which the attack tool is launching.

B. DEFENSE TOOLS

In order to create a barrier to prevent that malicious traffic arrive to the target's attack, some existing defense tools were used for this purpose as can be seen in Table 8. The IDS (Intrusion Detection System) tools Snort and Suricata [144] were adapted to be used for DDoS attacks detection and more recently used for slowloris attack detection as in [116] and [141]. The recent SDToW (Slowloris Detecting Tool for WMNs) tool, developed for the slowloris attack detection for mobile networks. As the IDS tools, the iptables firewall was adapted for slowloris attack detection setting rules accordingly [142]. Table 8 describes the defense tools in detail where the *Year* field indicates the birth of the defense tool. The *Tool name* field indicates the name of the tool. The *Defense attack mode* field indicates the format of the attack, which are: single or distributed. The *Operating System* field indicates which system is used by the tools. The *Programming language* field indicates which language the defense tool was developed. The *Interface* field indicates which interface type is used to configure the parameters of the defense tool, if it is a terminal type or a graphical type. The *Defense against field* indicates the type o attack which the defense tool is stopping.

VI. CONCLUSION

LDoS attacks, although still at an early stage, are receiving increased attention due to the fact that most of them are not detected by High-rate DoS attack detection mechanisms. Such an attack method has been widely adopted by attackers, since it makes them stealthier and more capable of shutting down their targets.

In this paper, we presented a comprehensive study guide of the LDoS attacks, defense/detection mechanisms and tools. First, we divided the attacks into three categories for the purpose of making them more comprehensible so as to be distinguished in terms of format, target, and the purpose of the attack. Second, we presented the mechanisms that can identify the attacks based on the fingerprints left over from the attack traffic and then stop or mitigate them. Third, we showed a list of attack tools that can reproduce some of the LDoS attack concepts.

REFERENCES

[1] (2001). *The Changing Face of Distributed Denial of Service Mitigation*. Accessed: Feb. 13, 2021. [Online]. Available: https://www.sans.org/white-papers/462/

[2] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surveys*, vol. 39, no. 1, p. 3, Apr. 2007.

[3] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: Classification and state-of-the-art," *Comput. Netw.*, vol. 44, no. 5, pp. 643–666, Apr. 2004.

[4] R. K. Chang, "Defending against flooding-based distributed denial-of-service attacks: A tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42–51, Mar. 2002.

[5] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[6] H. Beitollahi and G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," *Comput. Commun.*, vol. 35, no. 11, pp. 1312–1332, Jun. 2012.

[7] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, 2013.

[8] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2046–2069, Mar. 2013.

[9] L. Garber, "Denial-of-service attacks rip the internet," *Computer*, vol. 33, no. 4, pp. 12–17, Apr. 2000.

[10] X.-M. Liu, G. Cheng, Q. Li, and M. Zhang, "A comparative study on flood DoS and low-rate DoS attacks," *J. China Universities Posts Telecommun.*, vol. 19, pp. 116–121, Jun. 2012.

[11] Q. Zhu, Z. Yizhi, and X. Chuiyi, "Research and survey of low-rate denial of service attacks," in *Proc. 13th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2011, pp. 1195–1198.

[12] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks: The shrew vs. the mice and elephants," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2003, pp. 75–86.

[13] G. Maciá-Fernández, R. A. Rodríguez-Gómez, and J. E. Díaz-Verdejo, "Defense techniques for low-rate DoS attacks against application servers," *Comput. Netw.*, vol. 54, no. 15, pp. 2711–2727, Oct. 2010.

[14] Z. Wu, C. Wang, and H. Zeng, "Research on the comparison of flood DDoS and low-rate DDoS," in *Proc. Int. Conf. Multimedia Technol.*, Jul. 2011, pp. 5503–5506.

[15] (2001). *New Breed of Attack Zombies Lurk*. Accessed: Apr. 19, 2022. [Online]. Available: https://www.wired.com/2001/05/new-breed-of-attack-zombies-lurk/

[16] R. Mathew and V. Katkar, "Survey of low rate DoS attack detection mechanisms," in *Proc. Int. Conf. Workshop Emerg. Trends Technol.*, 2011, pp. 955–958.

[17] L. Mohan, M. G. Bijesh, and J. K. John, "Survey of low rate denial of service (LDoS) attack on RED and its counter strategies," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2012, pp. 1–7.

[18] (2012). *Taxonomy of Slow DoS Attacks to Web Applications*. Accessed: Feb. 13, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-34135-9_20

[19] E. Cambiaso, G. Papaleo, G. Chiola, and M. Aiello, "Slow DoS attacks: Definition and categorisation," *Int. J. Trust Manage. Comp. Commun.*, vol. 1, nos. 3–4, pp. 300–319, Jan. 2013.

[20] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, "Low-rate DoS attacks, detection, defense, and challenges: A survey," *IEEE Access*, vol. 8, pp. 43920–43943, 2020.

[21] N. Tripathi and N. Hubballi, "Application layer denial-of-service attacks and defense mechanisms: A survey," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–33, May 2022, doi: 10.1145/3448291.

[22] D. Zhang, Q.-G. Wang, G. Feng, Y. Shi, and A. V. Vasilakos, "A survey on attack detection, estimation and control of industrial cyber–physical systems," *ISA Trans.*, vol. 116, pp. 1–16, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001905782100046X

[23] H. A. P. and K. K., "Secure-MQTT: An efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for Internet of Things," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–15, Dec. 2019.

[24] S. Iranmanesh, F. S. Abkenar, A. Jamalipour, and R. Raad, "A heuristic distributed scheme to detect falsification of mobility patterns in internet of vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 719–727, Jan. 2022.

[25] S. Ramesh, C. Yaashuwanth, K. Prathibanandhi, A. R. Basha, and T. Jayasankar, "An optimized deep neural network based DoS attack detection in wireless video sensor network," *J. Ambient Intell. Humanized Comput.*, vol. 4, pp. 1–14, Jan. 2021.

[26] R. SaiSindhuTheja and G. K. Shyam, "An efficient Metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620309364

[27] F. Hussain, S. G. Abbas, G. A. Shah, I. M. Pires, U. U. Fayyaz, F. Shahzad, N. M. Garcia, and E. Zdravevski, "A framework for malicious traffic detection in IoT healthcare environment," *Sensors*, vol. 21, no. 9, p. 3025, Apr. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3025

[28] A. Kuzmanovic and E. W. Knightly, "Low-rate TCP-targeted denial of service attacks: The shrew vs. the mice and elephants," in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun.*, 2003, pp. 75–86.

[29] A. Shevtekar and N. Ansari, "A router-based technique to mitigate reduction of quality (RoQ) attacks," *Comput. Netw.*, vol. 52, no. 5, pp. 957–970, 2008.

[30] C. Xu, J. Shen, and X. Du, "Low-rate DoS attack detection method based on hybrid deep neural networks," *J. Inf. Secur. Appl.*, vol. 60, Aug. 2021, Art. no. 102879.

[31] Y. Chen and K. Hwang, "Collaborative detection and filtering of shrew DDoS attacks using spectral analysis," *J. Parallel Distrib. Comput.*, vol. 66, no. 9, pp. 1137–1151, Sep. 2006.

[32] H. Sun, J. C. S. Lui, and D. K. Y. Yau, "Distributed mechanism in detecting and defending against the low-rate TCP attack," *Comput. Netw.*, vol. 50, no. 13, pp. 2312–2330, Sep. 2006.

[33] Z.-J. Wu, J. Lei, D. Yao, M.-H. Wang, and S. M. Musa, "Chaos-based detection of LDoS attacks," *J. Syst. Softw.*, vol. 86, no. 1, pp. 211–221, Jan. 2013.

[34] S. S. Kanhere and A. Naveed, "A novel tuneable low-intensity adversarial attack," in *Proc. IEEE Conf. Local Comput. Netw. 30th Anniversary (LCN)*, 2005, pp. 794–801.

[35] X. Luo and R. K. C. Chang, "On a new class of pulsing denial-of-service attacks and the defense," in *Proc. Netw. Distrib. Syst. Symp.*, 2005, pp. 1–19.

[36] J. Luo and X. Yang, "The NewShrew attack: A new type of low-rate TCP-targeted DoS attack," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 713–718.

[37] M. Guirguis, A. Bestavros, and I. Matta, "On the impact of low-rate attacks," in *Proc. IEEE Int. Conf. Commun.*, Jan. 2006, pp. 2316–2321.

[38] M. Yue, Z. Wu, and M. Wang, "A new exploration of FB-shrew attack," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1987–1990, Oct. 2016.

[39] M. Yue, M. Wang, and Z. Wu, "Low-high burst: A double potency varying-RTT based full-buffer shrew attack model," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2285–2300, Oct. 2021.

[40] S. A. Arunmozhi and Y. Venkataramani, "A flow monitoring scheme to defend reduction-of-quality (RoQ) attacks in mobile ad-hoc networks," *Inf. Secur. J. A, Global Perspective*, vol. 19, no. 5, pp. 263–272, Oct. 2010.

[41] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the transients of adaptation for RoQ attacks on internet resources," in *Proc. 12th IEEE Int. Conf. Netw. Protocols*, Oct. 2004, pp. 184–195.

[42] M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang, "Adversarial exploits of end-systems adaptation dynamics," *J. Parallel Distrib. Comput.*, vol. 67, no. 3, pp. 318–335, Mar. 2007.

[43] A. Shevtekar and N. Ansari, "A router-based technique to mitigate reduction of quality (RoQ) attacks," *Comput. Netw.*, vol. 52, no. 5, pp. 957–970, 2008.

[44] Y.-M. Ke, C.-W. Chen, H.-C. Hsiao, A. Perrig, and V. Sekar, "CICADAS: Congesting the internet with coordinated and decentralized pulsating attacks," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, May 2016, pp. 699–710.

[45] (2015). *SlowHTTPTest*. Accessed: Feb. 13, 2019. [Online]. Available: https://github.com/shekyan/slowhttptest/wiki

[46] M. Aiello, G. Papaleo, and E. Cambiaso, "SlowReq: A weapon for cyberwarfare operations. Characteristics, limits, performance, remediations," in *Proc. Int. Joint Conf. SOCO-CISIS-ICEUTE*. Cham, Switzerland: Springer, 2014, pp. 537–546.

[47] E. Cambiaso, G. Papaleo, and M. Aiello, "Slowcomm: Design, development and performance evaluation of a new slow DoS attack," *J. Inf. Secur. Appl.*, vol. 35, pp. 23–31, Aug. 2017.

[48] M. M. Najafabadi, T. M. Khoshgoftaar, A. Napolitano, and C. Wheelus, "Rudy attack: Detection at the network level and its important features," in *Proc. FLAIRS Conf.*, 2016, pp. 288–293.

[49] (2018). *Cloudfare*. Accessed: Feb. 13, 2019. [Online]. Available: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/r-u-dead-yet-rudy/

[50] (2012). *Are You Ready For Slow Reading?* Accessed: Feb. 13, 2019. [Online]. Available: https://blog.shekyan.com/2012/01/are-you-ready-for-slow-reading.html

[51] J. Park, "Analysis of slow read DoS attack and countermeasures on web servers," *Int. J. Cyber-Secur. Digit. Forensics*, vol. 4, no. 2, pp. 339–353, 2015.
[52] S. Tayama and H. Tanaka, "Analysis of slow read dos attack and communication environment," in *Proc. Int. Conf. Mobile Wireless Technol.* Cham, Switzerland: Springer, 2017, pp. 350–359.
[53] C. Kemp, C. Calvert, and T. Khoshgoftaar, "Utilizing netflow data to detect slow read attacks," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, Jul. 2018, pp. 108–116.
[54] E. Cambiaso, G. Papaleo, and M. Aiello, "SlowDroid: Turning a smartphone into a mobile attack vector," in *Proc. Int. Conf. Future Internet Things Cloud*, Aug. 2014, pp. 405–410.
[55] E. Cambiaso, G. Papaleo, G. Chiola, and M. Aiello, "Designing and modeling the slow next DoS attack," in *Computational Intelligence in Security for Information Systems Conference*. Cham, Switzerland: Springer, 2015, pp. 249–259.
[56] E. Cambiaso, G. Chiola, and M. Aiello, "Introducing the SlowDrop attack," *Comput. Netw.*, vol. 150, pp. 234–249, Feb. 2019.
[57] 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.
[58] G. Maciá-Fernández, J. E. Díaz-Verdejo, and P. García-Teodoro, "Evaluation of a low-rate DoS attack against iterative servers," *Comput. Netw.*, vol. 51, no. 4, pp. 1013–1030, Mar. 2007.
[59] G. Maciá-Fernández, J. E. Díaz-Verdejo, and P. García-Teodoro, "Evaluation of a low-rate DoS attack against application servers," *Comput. Secur.*, vol. 27, nos. 7–8, pp. 335–354, Dec. 2008.
[60] G. Macia-Fernandez, J. E. Diaz-Verdejo, and P. Garcia-Teodoro, "Mathematical model for low-rate DoS attacks against application servers," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 519–529, Sep. 2009.
[61] G. Yang, M. Gerla, and M. Y. Sanadidi, "Defense against low-rate TCP-targeted denial-of-service attacks," in *Proc. 9th Int. Symp. Comput. Commun.*, 2004, pp. 345–350.
[62] H. Sun, J. C. S. Lui, and D. K. Y. Yau, "Defending against low-rate TCP attacks: Dynamic detection and protection," in *Proc. 12th IEEE Int. Conf. Netw. Protocols*, Oct. 2004, pp. 196–205.
[63] Y. Kwok, R. Tripathi, Y. Chen, and K. Hwang, "HAWK: Halting anomalies with weighted choking to rescue well-behaved TCP sessions from shrew DDoS attacks," in *Proc. Netw. Mobile Comput.*, 2005, pp. 423–432.
[64] Y. Chen, K. Hwang, and Y.-K. Kwok, "Filtering of shrew DDoS attacks in frequency domain," in *Proc. IEEE Conf. Local Comput. Netw. 30th Anniversary (LCN)*, May 2005, pp. 1–8.
[65] A. Shevtekar, K. Anantharam, and N. Ansari, "Low rate TCP denial-of-service attack detection at edge routers," *IEEE Commun. Lett.*, vol. 9, no. 4, pp. 363–365, Apr. 2005.
[66] S. Sarat and A. Terzis, "On the effect of router buffer sizes on low-rate denial of service attacks," in *Proc. 14th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2005, pp. 281–286.
[67] A. Shevtekar, J. Stille, and N. Ansari, "On the impacts of low rate DoS attacks on VoIP traffic," *Secur. Commun. Netw.*, vol. 1, no. 1, pp. 45–56, 2008.
[68] Z. Liu and L. Guan, "Attack simulation and signature extraction of low-rate DoS," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Secur. Informat.*, Apr. 2010, pp. 544–548.
[69] C. Zhang, J. Yin, Z. Cai, and W. Chen, "RRED: Robust RED algorithm to counter low-rate denial-of-service attacks," *IEEE Commun. Lett.*, vol. 14, no. 5, pp. 489–491, May 2010.
[70] H. Hu, J. Zhang, B. Liu, L. Chen, and X. Chen, "Simulation and analysis of distributed low-rate denial-of-service attacks," in *Proc. 5th Int. Conf. Comput. Sci. Converg. Inf. Technol.*, Nov. 2010, pp. 620–626.
[71] H. Kumawat and G. Meena, "Characterization, detection and mitigation of low-rate DoS attack," in *Proc. Int. Conf. Inf. Commun. Technol. Competitive Strategies*, 2014, p. 69.
[72] J. Singh, S. Gupta, and L. Kaur, "A MAC layer based defense architecture for reduction of quality (RoQ) attacks in wireless LAN," *Int. J. Comput. Sci. Inf. Secur.*, vol. 7, no. 1, pp. 284–291, 2010.
[73] Z. Wu, M. Yue, D. Li, and K. Xie, "SEDP-based detection of low-rate DoS attacks," *Int. J. Commun. Syst.*, vol. 28, no. 11, pp. 1772–1788, Jul. 2015.
[74] Y. Xiang, K. Li, and W. Zhou, "Low-rate DDoS attacks detection and traceback by using new information metrics," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 426–437, Jun. 2011.
[75] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," *Future Gener. Comput. Syst.*, vol. 89, pp. 685–697, Dec. 2018.
[76] X. Zhang, Z. Wu, J. Chen, and M. Yue, "An adaptive KPCA approach for detecting LDoS attack," *Int. J. Commun. Syst.*, vol. 30, no. 4, p. e2993, Mar. 2017.
[77] Z. Liu, X. Yin, and H. J. Lee, "A new network flow grouping method for preventing periodic shrew DDoS attacks in cloud computing," in *Proc. 18th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2016, pp. 66–69.
[78] J. Lin, C. Zhang, Z. Cai, Q. Liu, and J. Yin, "A TCP-friendly AQM algorithm to mitigate low-rate DDoS attacks," *Int. J. Auto. Adapt. Commun. Syst.*, vol. 9, nos. 1–2, pp. 149–163, 2016.
[79] C. Huang, P. Yi, F. Zou, Y. Yao, W. Wang, and T. Zhu, "CCID: Cross-correlation identity distinction method for detecting shrew DDoS," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–9, Feb. 2019.
[80] M. Şimşek and A. Şentürk, "Fast and lightweight detection and filtering method for low-rate TCP targeted distributed denial of service (LDDoS) attacks," *Int. J. Commun. Syst.*, vol. 31, no. 18, p. e3823, Dec. 2018.
[81] M. Siracusano, S. Shiaeles, and B. Ghita, "Detection of LDDoS attacks based on TCP connection parameters," in *Proc. Global Inf. Infrastructure Netw. Symp. (GIIS)*, Oct. 2018, pp. 1–6.
[82] P. Cotae and R. Rabie, "On a game theoretic approach to detect the low-rate denial of service attacks," in *Proc. Int. Conf. Commun. (COMM)*, Jun. 2018, pp. 19–26.
[83] Z. Liu, X. Yin, R. Yang, and A. Dong, "Early detection of LDDoS attacks in IoT utilizing locality sensitive incremental TSVM method," in *Proc. 23rd Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2021, pp. 194–199.
[84] G. Kaur and P. Agrawal, "Detection of LDoS attacks using variant of CUSUM and Shiryaev—Roberts's algorithm," in *Proc. 4th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, 2016, pp. 363–369.
[85] (2007). *Low-Rate TCP-Targeted DoS Attack Disrupts Internet Routing*. Accessed: Feb. 13, 2018. [Online]. Available: https://www.ndss-symposium.org/ndss2007/low-rate-tcp-targeted-dos-attack-disrupts-internet-routing/
[86] G. Thatte, U. Mitra, and J. Heidemann, "Detection of low-rate attacks in computer networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1–6.
[87] X. He, C. Papadopoulos, J. Heidemann, U. Mitra, U. Riaz, and A. Hussain, "Spectral analysis of bottleneck traffic," Dept. Comput. Sci., Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. USC-CSD-TR-05-854, 2005.
[88] R. Mathew and V. Katkar, "Software based low rate DoS attack detection mechanism," *Int. J. Comput. Appl.*, vol. 20, no. 6, pp. 14–18, Apr. 2011.
[89] Z. Wu, R. Hu, and M. Yue, "Flow-oriented detection of low-rate denial of service attacks," *Int. J. Commun. Syst.*, vol. 29, no. 1, pp. 130–141, Jan. 2016.
[90] N. Agrawal and S. Tapaswi, "An SDN-assisted defense mechanism for the shrew DDoS attack in a cloud computing environment," *J. Netw. Syst. Manage.*, vol. 29, no. 2, pp. 1–28, Apr. 2021.
[91] D. Boro, M. Haloi, and D. K. Bhattacharyya, "A fast self-similarity matrix-based method for shrew DDoS attack detection," *Inf. Secur. J. A, Global Perspective*, vol. 29, no. 2, pp. 73–90, Mar. 2020.
[92] D. Tang, Y. Yan, S. Zhang, J. Chen, and Z. Qin, "Performance and features: Mitigating the low-rate TCP-targeted DoS attack via SDN," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 428–444, Jan. 2022.
[93] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Information metrics for low-rate DDoS attack detection: A comparative evaluation," in *Proc. 7th Int. Conf. Contemp. Comput. (IC)*, Aug. 2014, pp. 80–84.
[94] Z. Wu, L. Liu, and X. Liu, "The approach of detecting LDoS attack based on correlative parameters," in *Proc. Int. Conf. Multimedia Technol.*, Jul. 2011, pp. 5587–5590.
[95] K. Chen, H. Liu, and X. Chen, "EBDT: A method for detecting LDoS attack," in *Proc. Int. Conf. Inf. Automat. (ICIA)*, Jun. 2012, pp. 911–916.
[96] C. Zhang, Z. Cai, W. Chen, X. Luo, and J. Yin, "Flow level detection and filtering of low-rate DDoS," *Comput. Netw.*, vol. 56, no. 15, pp. 3417–3431, 2012.
[97] Z. Wu, L. Zhang, and M. Yue, "Low-rate DoS attacks detection based on network multifractal," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 5, pp. 559–567, Sep. 2015.
[98] K. Bhushan and B. B. Gupta, "Hypothesis test for low-rate DDoS attack detection in cloud computing environment," *Proc. Comput. Sci.*, vol. 132, pp. 947–955, Jan. 2018.

[99] D. Tang, S. Zhang, J. Chen, and X. Wang, "The detection of low-rate DoS attacks using the SADBSCAN algorithm," *Inf. Sci.*, vol. 565, pp. 229–247, Jul. 2021.

[100] W. Ren, D.-Y. Yeung, H. Jin, and M. Yang, "Pulsing RoQ DDoS attack and defense scheme in mobile ad hoc networks," *Int. J. Netw. Secur.*, vol. 4, no. 2, pp. 227–234, 2007.

[101] A. Shevtekar and N. Ansari, "A router-based technique to mitigate reduction of quality (RoQ) attacks," *Comput. Netw.*, vol. 52, no. 5, pp. 957–970, 2008.

[102] Y. Chen and K. Hwang, "Spectral analysis of TCP flows for defense against reduction-of-quality attacks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 1203–1210.

[103] S. Gulati and A. Dhaliwal, "Mitigating RoQ attacks using flow monitoring method," *Int. J. Eng. Trends Technol.*, vol. 4, pp. 4074–4079, Jan. 2013.

[104] K. Wen, J. Yang, F. Cheng, C. Li, Z. Wang, and H. Yin, "Two-stage detection algorithm for RoQ attack based on localized periodicity analysis of traffic anomaly," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–6.

[105] G. Yu, T. Li, J. Wei, and C. Liu, "Assessment of reduction of quality attacks on mobile IP networks," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl., IEEE Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, Dec. 2017, pp. 449–453.

[106] H. Chen, M. Liu, and F. Zhongchuan, "Using improved Hilbert–Huang transformation method to detect routing-layer reduce of quality attack in wireless sensor network," *Wireless Pers. Commun.*, vol. 104, no. 2, pp. 595–615, Jan. 2019.

[107] V. de Miranda Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms," *Comput. Netw.*, vol. 186, pp. 1–18, Feb. 2021.

[108] B. Liu, D. Tang, Y. Yan, Z. Zheng, S. Zhang, and J. Zhou, "TS-SVM: Detect LDoS attack in SDN based on two-step self-adjusting SVM," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 678–685.

[109] M. Aiello, E. Cambiaso, S. Scaglione, and G. Papaleo, "A similarity based approach for application DoS attacks detection," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2013, pp. 430–435.

[110] Y. G. Dantas, V. Nigam, and I. E. Fonseca, "A selective defense for application layer DDoS attacks," in *Proc. IEEE Joint Intell. Secur. Informat. Conf.*, Sep. 2014, pp. 75–82.

[111] M. Mongelli, M. Aiello, E. Cambiaso, and G. Papaleo, "Detection of DoS attacks through Fourier transform and mutual information," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 7204–7209.

[112] V. Katkar, A. Zinjade, S. Dalvi, T. Bafna, and R. Mahajan, "Detection of DoS/DDoS attack against HTTP servers using naive Bayesian," in *Proc. Int. Conf. Comput. Commun. Control Autom.*, Feb. 2015, pp. 280–285.

[113] A. Aqil, A. O. F. Atya, T. Jaeger, S. V. Krishnamurthy, K. Levitt, P. D. McDaniel, J. Rowe, and A. Swami, "Detection of stealthy TCP-based DoS attacks," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2015, pp. 348–353.

[114] T. Hirakawa, K. Ogura, B. B. Bista, and T. Takata, "A defense method against distributed slow HTTP DoS attack," in *Proc. 19th Int. Conf. Netw.-Based Inf. Syst. (NBiS)*, Sep. 2016, pp. 152–158.

[115] K. J. Singh and T. De, "MLP-GA based algorithm to detect application layer DDoS attack," *J. Inf. Secur. Appl.*, vol. 36, pp. 145–153, Oct. 2017.

[116] V. D. S. Faria, J. A. Gonçalves, C. A. M. D. Silva, G. D. B. Vieira, and D. M. Mascarenhas, "SDToW: A slowloris detecting tool for WMNs," *Information*, vol. 11, no. 12, p. 544, Nov. 2020.

[117] P. Cotae, M. Kang, and A. Velazquez, "Multiple time series Fisher periodicity test for the detection of the distributed new shrew attacks," in *Proc. Int. Conf. Commun. (COMM)*, Jun. 2016, pp. 9–14.

[118] X. Luo, E. W. W. Chan, and R. K. C. Chang, "Detecting pulsing denial-of-service attacks with nondeterministic attack intervals," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, pp. 1–13, Dec. 2009.

[119] M. M. Najafabadi, T. M. Khoshgoftaar, A. Napolitano, and C. Wheelus, "Rudy attack: Detection at the network level and its important features," in *Proc. 29th Int. Flairs Conf.*, 2016, pp. 282–287.

[120] R. Xie, M. Xu, J. Cao, and Q. Li, "SoftGuard: Defend against the low-rate TCP attack in SDN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[121] T. V. Phan, T. M. R. Gias, S. T. Islam, T. T. Huong, N. H. Thanh, and T. Bauschert, "Q-MIND: Defeating stealthy DoS attacks in SDN with a machine-learning based defense framework," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[122] J. A. Pérez-Díaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.

[123] J. F. Balarezo, S. Wang, K. G. Chavez, A. Al-Hourani, J. Fu, and K. Sithamparanathan, "Low-rate TCP DDoS attack model in the southbound channel of software defined networks," in *Proc. 14th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2020, pp. 1–10.

[124] D. Tang, S. Zhang, Y. Yan, J. Chen, and Z. Qin, "Real-time detection and mitigation of LDoS attacks in the SDN using the HGB-FP algorithm," *IEEE Trans. Services Comput.*, early access, Aug. 4, 2022, doi: 10.1109/TSC.2021.3102046.

[125] H. S. Ilango, M. Ma, and R. Su, "Low rate DoS attack detection in IoT–SDN using deep learning," in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData), IEEE Congr. Cybermatics (Cybermatics)*, Dec. 2021, pp. 115–120.

[126] B. Liu, D. Tang, Y. Yan, Z. Zheng, S. Zhang, and J. Zhou, "TS-SVM: Detect LDoS attack in SDN based on two-step self-adjusting SVM," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 678–685.

[127] X. Li, N. Luo, D. Tang, Z. Zheng, Z. Qin, and X. Gao, "BA-BNN: Detect LDoS attacks in SDN based on bat algorithm and BP neural network," in *Proc. IEEE Int. Conf Parallel, Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput., Commun., Social Comput., Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Sep. 2021, pp. 300–307.

[128] (2018). *Orignal-Slowloris-HTTP-DoS*. Accessed: Feb. 13, 2021. [Online]. Available: https://github.com/Ogglas/Orignal-Slowloris-HTTP-DoS

[129] (2008). *PyLoris: A Testing Tool for Web Server DoS Vulnerabilites*. Accessed: Feb. 13, 2021. [Online]. Available: https://web.archive.org/web/20090715100428/http://motomastyle.com/ pyloris/

[130] (2009). *How to Help Take Down Gerdab.IR in 5 Easy Steps*. Accessed: Feb. 13, 2021. [Online]. Available: http://cyberwar4iran.blogspot.com/

[131] (2009). *Apache HTTP DoS Tool Released*. Accessed: Feb. 13, 2021. [Online]. Available: https://isc.sans.edu/diary/Apache+HTTP+DoS+tool+released/6601

[132] (2018). *HTTP Bog: A Slow HTTP Denial-of-Service Tool*. Accessed: Apr. 13, 2022. [Online]. Available: https://www.softpedia.com/get/Internet/Servers/Server-Tools/HTTP-Bog.shtml

[133] (2012). *Torshammer*. Accessed: Apr. 19, 2022. [Online]. Available: https://sourceforge.net/projects/torshammer/

[134] (2014). *Goloris—Slowloris for NGINX DoS*. Accessed: Feb. 13, 2021. [Online]. Available: https://github.com/valyala/goloris

[135] G. Yaltirakli. (2015). *Slowloris*. [Online]. Available: https://github.com/gkbrk/slowloris

[136] (2016). *R-U-Dead-Yet*. Accessed: Apr. 19, 2022. [Online]. Available: https://sourceforge.net/projects/r-u-dead-yet/

[137] (2018). *Cyphon*. Accessed: Feb. 13, 2021. [Online]. Available: https://github.com/abila5h/Cyphon-DoS

[138] (2018). *SLOWW*. Accessed: Feb. 13, 2021. [Online]. Available: https://github.com/ethanent/sloww

[139] (2018). *Dotloris*. Accessed: Feb. 13, 2021. [Online]. Available: https://github.com/bass3l/dotloris

[140] (2018). *Pwnloris: An Improved Slowloris DoS Tool*. Accessed: Apr. 13, 2022. [Online]. Available: https://github.com/h0ussni/pwnloris

[141] T. E. de Sousa Araujo, F. M. Matos, and J. A. Moreira, "Intrusion detection systems' performance for distributed denial-of-service attack," in *Proc. Conf. Electr., Electron. Eng., Inf. Commun. Technol. (CHILECON)*, Oct. 2017, pp. 1–6.

[142] R. K. Sharma, B. Issac, and H. K. Kalita, "Intrusion detection and response system inspired by the defense mechanism of plants," *IEEE Access*, vol. 7, pp. 52427–52439, 2019.

[143] (2009). *Apache and Squid DoS*. Accessed: Feb. 13, 2021. [Online]. Available: https://seclists.org/fulldisclosure/2009/Jun/207

[144] W. Park and S. Ahn, "Performance comparison and detection analysis in Snort and Suricata environment," *Wireless Pers. Commun.*, vol. 94, no. 2, pp. 241–252, May 2017.

VINÍCIUS DE MIRANDA RIOS was born in Janaúba, Minas Gerais, Brazil, in 1980. He received the B.Sc. degree in information systems from the Centro Universitário Luterano do Brasil (CEULP-ULBRA), Brazil, in 2005, and the master's degree in electrical engineering from the Universidade de Brasília (UnB), Brazil, in 2012. He is currently pursuing the Ph.D. degree with the Universidade da Beira Interior (UBI), Covilhã, Portugal. He received the grant from the Brazilian CAPES foundation for his Ph.D. degree. He has been a Professor of computer science with the Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO), since 2015, where he teaches subjects related to computer networks and programming to graduate courses. He is an Instructor with IFTO Huawei ICT Academy.

PEDRO R. M. INÁCIO (Senior Member, IEEE) was born in Covilhã, Portugal, in 1988. He received the B.Sc. degree in mathematics/computer science and the Ph.D. degree in computer science and engineering from the Universidade da Beira Interior (UBI), Portugal, in 2005 and 2009, respectively. The Ph.D. work was performed in the enterprise environment at Nokia Siemens Networks Portugal S.A., through a Ph.D. granted by the Portuguese Foundation for Science and Technology. He has been a Professor of computer science with UBI, since 2010, where he teaches subjects related to information assurance and security, programming of mobile devices and computer based simulation, to graduate and undergraduate courses, namely to the B.Sc., M.Sc., and Ph.D. programs in computer science and engineering. He is currently the Head of the Department of Computer Science, UBI. He is an Instructor with UBI Cisco Academy. He is a Researcher with the Instituto de Telecomunicações (IT). He has about 30 publications in the form of book chapters and papers in international peer-reviewed books, conferences, and journals. His main research interests include information assurance and security, computer based simulation, and network traffic monitoring, and analysis and classification. He frequently reviews papers for IEEE, Springer, Wiley, and Elsevier journals. He is a member of the Technical Program Committee of international conferences, such as the ACM Symposium on Applied Computing-Track on Networking.

DAMIEN MAGONI (Senior Member, IEEE) received the M.Eng. degree from Télécom Paris, in 1995, and the M.Sc. and Ph.D. degrees from the University of Strasbourg in 1999 and 2002, respectively. He has been a Visiting Researcher at various institutions around the world, including the AIST at Tsukuba, the University of Sydney, the University of Michigan at Ann Arbor, and University College Dublin. He has been a Full Professor of computer science with the University of Bordeaux, since 2008. From 2002 to 2008, he was an Associate Professor at the University of Strasbourg. Some of his research has been supported by grants from the European Union, the CNRS, and Science Foundation Ireland. He has co-published over 90 refereed research papers. He also has authored several open-source software programs for networking research and teaching. His latest contributions are the virtual network device and the network mobilizer, which jointly enable the emulation of mobile networks. His main research interests include computer communications and networking, with a focus on internet architecture, protocols, and applications. He has been a reviewer for over 20 academic journals and has been in the TPC of numerous high-level conferences. He is a senior member of ACM.

MÁRIO M. FREIRE (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in systems and automation from the University of Coimbra, Portugal, in 1992 and 1994, respectively, the Ph.D. degree in electrical engineering, in 2000, and the Habilitation degree in computer science from the University of Beira Interior (UBI), Portugal, in 2007. He is a Full Professor of computer science with UBI, which he joined in the Fall of 1994. In April 1993, he did a one-month internship at the Research Centre of Alcatel-SEL (now Nokia Networks), Stuttgart, Germany. He is the coauthor of seven international patents, coeditor of eight books published in the Springer LNCS book series, and coauthor of about 130 papers in international journals and conferences. His main research interests include computer systems and networks, including network and systems virtualization, cloud and edge computing, and security and privacy in computer systems and networks. He is a member of the Editorial Board of the ACM SIGAPP Applied Computing Review. He has served as a technical program committee member for several IEEE international conferences and is Co-Chair of the track on Networking of ACM SAC 2021. He is a Chartered Engineer by the Portuguese Order of Engineers and he is a member of the IEEE Computer Society and of the Association for Computing Machinery. He was the Editor of IEEE Communications Surveys and Tutorials, from 2007 to 2011, and is an Associate Editor of the *Wiley Security and Privacy Journal* and the *Wiley International Journal of Communication Systems*.