1-1-2020

Thwarting ICMP low-rate attacks against firewalls while minimizing legitimate traffic loss

Kadhim Hayawi
Zayed University

Zouheir Trabelsi
United Arab Emirates University

Safaa Zeidan
United Arab Emirates University

Mohammad Mehedy Masud
United Arab Emirates University

Recommended Citation

Hayawi, Kadhim; Trabelsi, Zouheir; Zeidan, Safaa; and Masud, Mohammad Mehedy, "Thwarting ICMP low-rate attacks against firewalls while minimizing legitimate traffic loss" (2020). *All Works*. 3655.
https://zuscholars.zu.ac.ae/works/3655

Received March 26, 2020, accepted April 4, 2020, date of publication April 13, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2987479

Thwarting ICMP Low-Rate Attacks Against Firewalls While Minimizing Legitimate Traffic Loss

KADHIM HAYAWI[1], ZOUHEIR TRABELSI[2], SAFAA ZEIDAN[2], AND MOHAMMAD MEHEDY MASUD[2], (Member, IEEE)

[1]College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates
[2]College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates

Corresponding author: Zouheir Trabelsi (trabelsi@uaeu.ac.ae)

This work was supported by the UAE University UPAR Research Grant Program under Grant 31T122.

ABSTRACT Low-rate distributed denial of service (LDDoS) attacks pose more challenging threats that disrupt network security devices and services. Such type of attacks is difficult to detect and mitigate. In LDDoS attacks, attacker uses low-volume of malicious traffic that looks alike legitimate traffic. Thus, it can enter the network in silence without any notice. However, it may have severe effect on disrupting network services, depleting system resources, and degrading network speed to a point considering them as one of the most damaging attack types. There are many types of LDDoS such as application server and ICMP error messages based LDDoS. This paper is solely concerned with the ICMP error messages based LDDoS. The paper proposes a mechanism to mitigate low-rate ICMP error message attacks targeting security devices, such as firewalls. The mechanism is based on triggering a rejection rule to defend against corresponding detected attack as early as possible, in order to preserve firewall resources. The rejection rule has certain adaptive activity time, during which the rule continues to reject related low-rate attack packets. This activity time is dynamically predicted for the next rule activation period according to current and previous attack severity and statistical parameters. However, the rule activity time needs to be stabilized in a manner in order to prevent any additional overhead to the system as well as to prevent incremental loss of corresponding legitimate packets. Experimental results demonstrate that the proposed mechanism can efficiently defend against incremental evasion cycle of low-rate attacks, and monitor rejection rule activity duration to minimize legitimate traffic loss.

INDEX TERMS Low-rate attacks, BlackNurse attack, Stateful firewall, session table, attack probabilistic modeling.

I. INTRODUCTION

Interconnected computer systems and networks still suffer from security threats especially Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. DoS attack is considered as one-to-one-attack that uses only one compromised host to influence the network bandwidth and degrade the provided services. However, DDoS attack uses multiple compromised hosts (botnet) that were previously infected with malwares to make them under malicious control. The common goal is to overload the targeted victim with a tremendous amount of bogus traffic until it becomes slow or unresponsive to legitimate requests. While servers such as Web servers are the common targeted victims of DoS attacks, firewalls and other security devices become a desirable target of sophisticated types of these attacks recently [1], [2].

Firewalls are considered as the first line of defense in network security. Based on the way they make their decisions, firewalls are of two main types: stateless and stateful. Stateless packet filtering firewalls base their protection decision on a single packet. They filter incoming and outgoing packet traffic on hosts and networks according to a predefined set of filtering rules. Each rule has an action associated with it either to accept or drop a packet, depending on the packet's header

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

information (Protocol (Prot), IP source (Src-IP), source Port (Src-P), IP destination (Dst-IP), destination Port (Dst-P), TCP flags (SYN, ACK, FIN, RST, PSH), ICMP messages, etc.).

On the other hand, stateful inspection firewall make their decision on a connection or session basis. Session table, which keeps record of states of connections, is considered as the heart of stateful firewall and it is usually used to improve its performance. It safely considers that packets belonging to a particular flow do not need to be re-checked against the filtering rules. However, session table may become a firewall bottleneck part and can be used against the firewall. Attackers may craft special type of packets to flood this limited resource. As a result, the session table is filled with illegitimate flows that prevent legitimate flows from being established.

While firewalls are designed to protect networks behind them, they can suffer themselves from DDoS attacks intending particularly to overwhelm them and bringing them to knees [1], [3]. Such types of DDoS attacks are known as Denial of Firewalling (DoF) attacks. DoF attacks may target the firewall filtering rules or the session table. The effect can appear as increase in packet filtering time, CPU utilization, memory usage and the number of allocated sessions in the session table. This may make the firewall unresponsive and lead to denial of service for all devices located behind the firewall. Since the firewall has finite amount of memory and CPU power the problem may get worse, as these finite firewall resources define an upper bound on the number of accepted traffic flows and the number of established sessions.

In DoF attacks two aspects are important to classify the type of attacks launched against a firewall: the targeted firewall's part and the used attack traffic volume, as shown in Fig. 1. Depending on the intended firewall part, attackers may target the firewall rule base or the session table. In DoF attacks targeting the rule base, malicious packets are sent to match the longest filtering path until either denied by the default rule or accepted by firewall bottom rules with high rules indices. This long path of packet matching process increases firewall filtering time and degrade its performance considerably. Likewise, DoF attacks targeting firewall's session table intend to make the session table full by sending massive fake requests that add redundant session entries, preventing the process of legitimate connection requests.

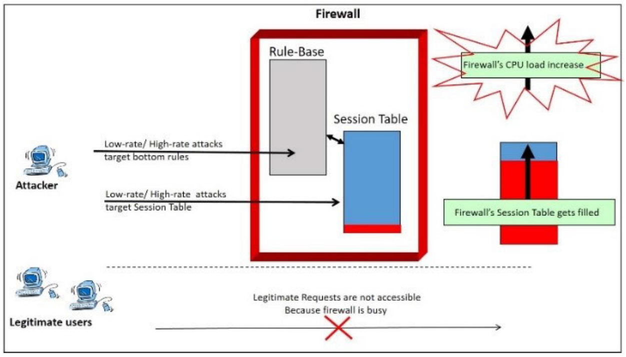

Firewall
Rule-Base
Session Table
Firewall's CPU load increase
Low-rate/ High-rate attacks
target bottom rules
Attacker
Low-rate/ High-rate attacks
target Session Table
Firewall's Session Table gets filled
Legitimate users
Legitimate Requests are not accessible
Because firewall is busy

FIGURE 1. DoF attacks targeting firewall's rule base or the session table.

The other criteria used for the classification of DoF attacks is the attack traffic volume. Attackers may use high or low-rate attack traffic to overwhelm the target firewall. In high-rate attacks, attackers use massive volume of attack traffic to overwhelm the available bandwidth and to flood a firewall with redundant packets, forcing it to perform purposeless extra work. This extra work usually degrades the firewall performance and holds up legitimate users' traffic. Contrary, a low-rate of special crafted malicious packets may have more prolong effect on firewalls causing additional harm and forcing them to work harder [4]. According to Corero research [5], [6], vast majority of DDoS attacks are relatively low-rate with 98% of these attacks were less than 10Gbps, and the average attack duration is short, with 81% lasting less than 10 minutes. Usually, a Low-rate attack is launched periodically with high narrow spike and low frequency. The main problem with such kind of attack traffic is that it is close to the behavior of normal traffic making it difficult to detect and mitigate. An example of such low-rate DoF attack is the emerging BlackNurse attack. The attack is considered as one of the LDDoS ICMP error message attacks. It launches a low-rate of special crafted ICMP packets (Destination unreachable Type: 3, Port unreachable Code: 3) that can overwhelm targeted firewalls' processors. When the firewall is under this attack, its CPU utilization increases sharply until the firewall becomes unresponsive. As a result, users from the LAN side can no longer access the Internet [7], [8].

This paper proposes a mechanism to defend against low-rate DoF attacks that use ICMP error messages. The mechanism generalizes the idea proposed in [9] and offers efficient and effective solution to its limitations. In [9], an approach is proposed to defend particularly against the BlackNurse attack. The approach uses an early rejection rule that is triggered when a certain threshold is reached. The rule has a time activity duration that is calculated according to previous attack severity and statistical parameters of attack's traffic. However, under certain circumstance, attacker may evade this approach by sending an incremental cycle of the BlackNurse attack rate. The goal of the attacker is to continuously increase the rule activity duration time, and this consequently will increase corresponding legitimate ICMP packet loss. As a result, in the worst case the rule will be activated all the time rejecting all incoming ICMP packets of Type: 3, Code: 3 including legitimate packets. Thus there is a natural trade-off between the firewall rule's activation period and the legitimate packets loss.

In order to address the aforementioned trade-off problem, this paper proposes a mechanism that can be used to mitigate low-rate ICMP error messages attacks. The mechanism ensures that the rule corresponding to the detected low-rate attack is activated as early as possible to preserve firewall resources. During rule activation time, the firewall rejects all packets related to the specified low-rate attack, provided that rule activity duration is dynamically adjusted depending on current and previous attack intensity and statistical properties of attack's traffic. Moreover, the rule activation time is stabilized in a manner preventing continuous attackers' trails to evade the proposed mechanism. The paper also includes detailed mathematical derivations of continuous time Markov chain modeling for low- rate attacks.

The rest of the paper is organized as follows. Section II discusses related work. Section III presents background on types of LDDoS attacks, possible available mitigations and their limitations which motivate the work in this paper. Section IV discuses continuous time Markov chain modeling based on Possion counting process of low-rate attacks. Section V proposes an ICMP timed early rejection rule to mitigate low-rate ICMP error message attacks. Section VI illustrates possible new attacks against timed early rejection rule that can cause legitimate packet loss. Section VII proposes algorithms to defend against incremental evasion cycle of low-rate ICMP error message attacks. Section VIII includes performance evaluation of the proposed mechanism. Finally, section IX concludes the paper.

II. RELATED WORK

LDDoS attacks are usually difficult to detect at the network level as they occupy less bandwidth. These attacks are often shorter in duration and can be launched through a single computer as in the case of the BlackNurse attack [7], [8]. They look alike legitimate traffic and often do not get the attention of IT security staff as they use low malicious traffic volume. However, they can deplete system resources and degrade network performance severely which make them the most damaging attacks type [5], [6]. Some research works are done on the detection of low-rate attacks; however, they lack proposal for real time mitigation mechanisms for such attacks. Wenke Lee in [10] states that "Existing defenses against low-volume DDoS attacks lack precision and they cannot create a response in a timely manner".

In [11], a technique is proposed to identify two LDDoS attacks, the constant and the pulsing attack from legitimate traffic based on their distribution difference of packet size. In [12] a method is proposed based on optimal objective entropy to detect LDDoS attacks. The idea generalizes the traditional entropy metric. However, the distance value between normal traffic and attack traffic is quite small and therefore, increasing the false-positive rate. Likewise, a generalized entropy and information distance between legitimate and attack traffic to detect LDDoS attacks is proposed in [13]. The proposed measurement outperforms the tradition entropy in terms of false-positive rate and distance gap, however distance gap is still small. These research work falls on the area of anomaly detection of low-rate attacks. In the area of signature detection of low-rate attacks, a distributed detection mechanism is presented in [14]. The mechanism is based on dynamic time-warping method to calculate the cumulative distance of the time warping between sampled and template flows which gives the degree of similarities between the two flows. However, this calculation is based on attack flow periodicity. Authors in [15] evaluated the combined impact of attack pattern and network environment. They studied system sophisticated attacks and the model of the minimum transmission rate of attack packets to tune the attack effect. However, all these research works fall in the low-rate detection area where no specified mitigations technique were proposed.

The majority of research work that exists in the literature attempt to improve the overall firewall performance by proposing techniques to eliminate or minimize DoF attacks targeting firewall's filtering rules. These can be classified into the following categories: Firewall's Rule/rule-field reordering techniques [16]–[19], Early packet rejection techniques [20]–[23], and Tree-based decision techniques with dynamic behavior [24]–[27].

Contrary, common mitigation techniques related to DoF attacks on firewall session table are based on attack rate limiting, as in Juniper Networks Screening features [28]. However, if the activation limit is over the desired boundary, then attack traffic may pass through and consumes firewall resources. Another point to mention is that these rate limiting features requires administration attention to enable them. Once enabled, they will be activated all the time unless disabled back. This may dramatically increase CPU and session table utilization, such as in TCP SYN flood mechanism [29]. In [30], a mechanism is proposed to defend against DoF attacks targeting session table. The mechanism used the natural properties of the splay tree firewall, and a session table architecture that is based on session attributes separation to deal with costly timeout attribute. In [31], fast session table manipulation algorithm is proposed to improve session timeout process. The algorithm covers multi-queue architectures however; it defends hosts only against SYN flood attack.

The work in this paper builds on and significantly extends the preliminary work presented in [9]. The most noticeable extensions include a model to handle incremental evasion cycle of low-rate attacks that can break the system proposed in [8], causing severe increment in the rejection rule activity time. The continuous increment in rule activity duration results in a direct increase in legitimate packet loss. The paper also includes detailed mathematical derivations of continuous time Markov chain modeling of low-rate attacks upon which we base our proposed solution. In addition, this paper presents an easy to implement algorithms to mitigate low-rate ICMP error message attacks.

III. LOW-RATE ATTACKS BACKGROUND

High-rate DDoS attacks are usually forceful, and flood the target with an overwhelming quantity of malicious packets to deplete its resources. These flooding attacks use a lot of bandwidth which requires that the attacker controls multiple machines in order to direct the massive traffic towards the victim network and the application layers. For instance, the SYN flooding attack exploits the connection limit of the target by sending massive SYN packets with spoofed Src_IP that add entries in the target session table.

However, SYN-ACK packets are never received, forcing these entries in the session table to remain until their timeout is reached. Thus, preventing benign requests.

In contrast, low-rate or slow-rate attacks are difficult to detect and handle, as their goal is to overwhelm the victim slowly and quietly. Instead of sending requests as fast as possible as in the case of high-rate attacks, they are sent as slow as possible consuming less bandwidth. The attack traffic is difficult to distinguish from normal traffic as well as hard to mitigate. Launching the attack does not require a lot of resources, thus a single computer would be enough with no need for additional bots. The victim can be a server such as HTTP server or a network security device such as a firewall.

A. LOW-RATE ATTACKS TARGETING SERVERS

Usually, low-rate attacks target the application layer of the server rather than the communication layers, as the former does not require huge resources, only few packets request with appropriate content are enough to launch the attack. These bogus requests force the server to allocate fictitious connections, reducing its ability to accept new requests. As a result, preventing legitimate user requests. This is accomplished by transmitting requests to the server very slowly, but faster than the server timeout. Most of the low-rate attacks target the server's HTTP protocol. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service. Attackers can use HTTP header and HTTP post requests to carry out the low-rate attacks [32], [33].

1) SLOWLORIS OR SLOW HEADER ATTACK

The attacker launches this attack with the help of a tool called Slowloris. By which, the attacker can send multiple partial HTTP header requests to the targeted server [34]. For each coming request, the server opens a thread which should be closed once the connection is completed. Normally, the server closes exceedingly long-time connections in order to free threads for upcoming requests. However, the attacker keeps sending partial header requests periodically to hold up the connection and prevent it from timing out. As a result, the server can not release any of the partially opened connections and keeps waiting for the requests' termination. As all available threads get in use, the server would not be able to accomplish legitimate user requests, as shown in Fig 2.

To be specific, all HTTP requests end with special characters that are parsed by the target server indicating that the request packet is received. For instance, in a complete HTTP GET request, each line in the header ends with CRLF (Carriage Return+Line Feed) character, whereas two CRLF characters are used in the last header row line to denote a blank line. The HTTP protocol uses the blank line to indicate the completion of the request header. Attackers make use of this information and send HTTP GET requests without the blank line, by suppressing one of the CRLF character from the header's last row line. Thus, the presence of only one single CRLF characters would indicate that the request headers are incomplete forcing the server to wait for long time and consuming its resources.

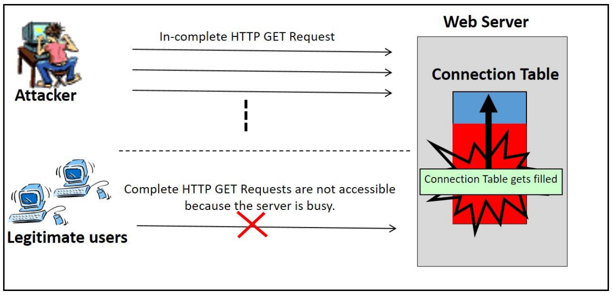

In-complete HTTP GET Request
Web Server
Connection Table
Attacker
Connection Table gets filled
Complete HTTP GET Requests are not accessible because the server is busy.
Legitimate users

FIGURE 2. Slow header attack.

2) SLOW BODY OR SLOW POST ATTACK

This attack is similar to Slow header attack concept. The attacker with the help of a tool called RUDY, "Are You Dead Yet?" sends partial POST body messages to the target server [35]. Normally, the HTTP post request header contains the size of the body message to be sent next. However, the attacker sends the body message very slowly, forcing the server to keep the connection open as more data is expected to be received. More similar connections are created until server resources are consumed in order to block legitimate user requests, as shown in Fig. 3.

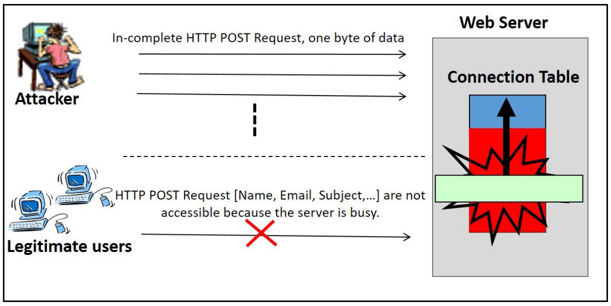

In-complete HTTP POST Request, one byte of data
Web Server
Connection Table
Attacker
HTTP POST Request [Name, Email, Subject,...] are not accessible because the server is busy.
Legitimate users

FIGURE 3. Slow body attack.

Usually, when a user fills in a web form, the data are sent to the HTTP server using HTTP POST request using one or two packets, and the server closes the connection. However, in Slow body attack the form data is split into many packets each contains one byte of the data to be sent. These packets are sent at random time intervals preventing the server from closing the connection and keeping it waiting for the request completion.

3) SLOW READ

The Slow header and Slow body attacks depend on sending slow HTTP requests, however, Slow read attack depends on reading HTTP responses slowly. In this attack, the attacker initiates legitimate connection with the target server and sends an appropriate HTTP request. However, the attacker reads the server replies very slowly. This would force the server to slow down its responses, and therefore the connection is kept alive for a long time. Moreover, the attacker may not read the server response for a long time, and then before the connection times out, starts reading data slowly one byte at a time. This is accomplished by sending a Zero window to the server, consequently, the server assumes that the requester is busy reading the sent data. As a result, the connection remains opened. Similar multiple connections would exhaust the server resources and lead to DoS situation at a very slow speed, preventing new requests [36]. The attack is illustrated in Fig. 4.

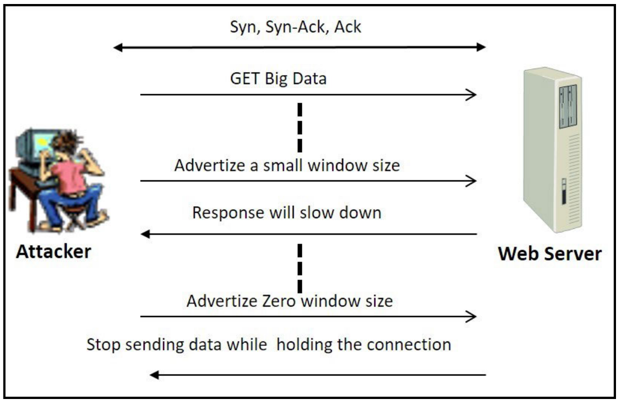

Syn, Syn-Ack, Ack
GET Big Data
Advertize a small window size
Response will slow down
Advertize Zero window size
Stop sending data while holding the connection
Attacker
Web Server

FIGURE 4. Slow read attack.

B. LOW-RATE ATTACKS TARGETING FIREWALLS

Attackers may target security devices such as firewalls, using a special crafted packet sent at low-rate that can consume firewall resources until it becomes unresponsive to legitimate request. The most known LDDoS attacks on firewall are based on the generation of ICMP error messages.

ICMP is an auxiliary protocol, which provides diagnostic information when requested [39]. ICMP can be divided into two broad categories: Error reporting and Query/Control, Fig. 5.

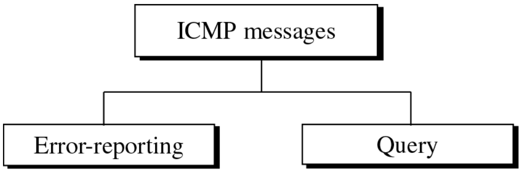


FIGURE 5. ICMP packet's types.

The header of an ICMP packet is made mainly from four fields, namely Type, Code, Checksum and Option fields, Fig.6.

ICMP Query messages are generated when requested. Usually, these messages are used to collect information about specific hosts or networks, or for troubleshooting. There are 4 types of ICMP Query messages: Echo request or reply, Timestamp request or reply, Address mask request or reply, and Router solicitation or advertisement. ICMP Query messages can help in identifying problems and diagnosing them. For example, ICMP Echo request or reply is the first step towards checking if the destination device is alive or not. To check it, the source device sends an ICMP Echo message to the destination (Type:8, Code:0). Upon receiving the Echo request, the destination device replies with an ICMP Echo reply message (Type: 0, Code: 0). Once the source node receives the Echo Reply from the destination, it understands that the remote device is alive.

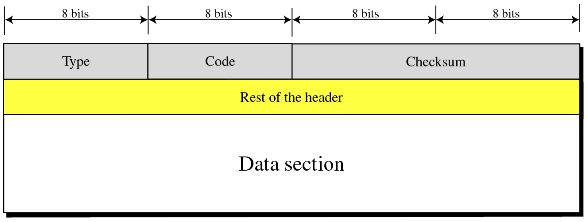


FIGURE 6. The fields of an ICMP packet.

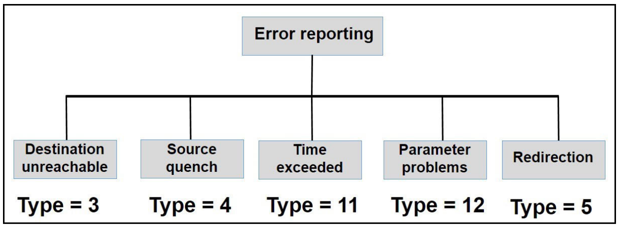


FIGURE 7. Types of ICMP error-reporting messages.

ICMP Error-reporting messages, on the hand, are generated during the processing of any Internet packet when a host or network device encounters a problem. That is, these messages are used to report issues or errors that took place in the Internet. There are 5 types of ICMP Error-reporting messages shown in Fig. 7: Destination unreachable, Source Quench, Time Exceeded, Parameter problem, and Redirection. For example, an ICMP error-reporting message of Type 3 (Destination unreachable) is generated when the destination of the packet cannot be reached. The Code field in the ICMP header provides further details about the message's type (e.g., Code = 0 for network unreachable, Code = 1 for host unreachable, Code = 2 for protocol unreachable, Code = 3 for port unreachable). Another example is the ICMP error-reporting message of Type 4 (Source Quench), which is a message from one host to another asking the other host to slow down the speed at which the packets are being sent. Source Quench is one of the ways to control the packet flow on the Internet.

The Option field of an ICMP error message usually includes information about the packet that caused the error. That is, the Option field includes part of the received IP packet, which is represented by the IP header, plus the first 8 bytes of the message data, as shown in Fig. 8.

Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

FIGURE 8. **The option field of an ICMP error-reporting message.**

Firewalls may require huge processing time to process ICMP Error-reporting messages. That is, once an ICMP Error-reporting packet reaches a firewall, the data in the Option field is extracted and the firewall's session table is searched for any session related to the received ICMP packet. Usually, this process may take enormous processing time and affect the overall firewall's performance. Hence, malicious users may succeed to create a DoF attack situation within the firewall by sending ICMP Error-reporting packets to the firewall targeting its session table. In contrast to common DoS attacks that are based on high-rate traffic generation, a low-rate of ICMP Error-reporting traffic have been found to be able to degrade the performance of firewalls significantly [4]–[9]. Such attacks are known under the name of low-rate DoF attacks. The BlackNurse attack is a well-known example of such attacks, and is specifically based on generating ICMP error message of type 3.

1) BLACKNURSE ATTACK

This attack depends on sending low-rate ICMP error message [Type:3 (Host unreachable), Code:3 (Port unreachable)] to the targeted firewall [7]–[9]. These error messages are considered among the most expensive computationally, because they consume much of the processing power of the stateful firewall. The attack can drive high firewall CPU loads and make the firewall unresponsive, as shown in Fig. 9. As a result, users from the LAN side will no longer be able to exchange network traffic.

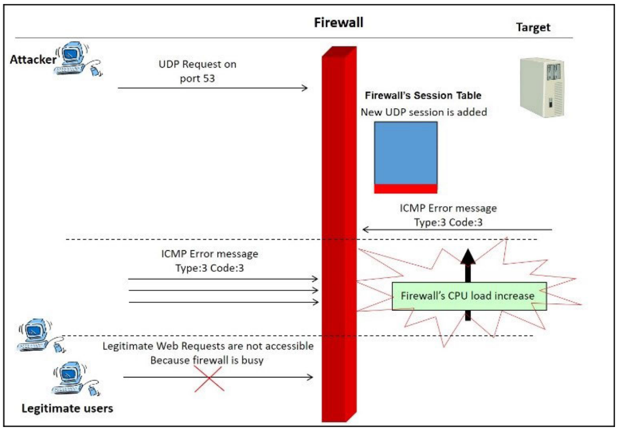

Firewall
Target
Attacker
UDP Request on port 53
Firewall's Session Table
New UDP session is added
ICMP Error message Type:3 Code:3
ICMP Error message Type:3 Code:3
Firewall's CPU load increase
Legitimate Web Requests are not accessible Because firewall is busy
Legitimate users

FIGURE 9. **The BlackNurse attack.**

Usually, ICMP port unreachable message [Type:3, Code:3] is sent when a destination host cannot deliver a reply packet because the intended port is not active. For instance, if a source computer sends a UDP port 53 request to a target computer that is not a DNS server, the target will generate ICMP port unreachable message to the source. Attacker makes use of such packets to launch the BlackNurse attack. The reason behind the increase in firewall CPU loads is due to the tracking process of ICMP error messages used in stateful firewalls. After receiving a UDP port 53 request shown in Fig. 9, the firewall inspects the source and destination addresses and UDP port numbers. If there is a filtering rule that allows the packet to across, the firewall inserts a new UDP entry in the session table. If there is no DNS service running on the target computer, the target will generate ICMP port unreachable message. Once the firewall receives the ICMP port unreachable message, it extracts from the packet's payload the attribute of the original packet (UDP request) that caused this error message to be sent. Then, the firewall searches its session table for a related session entry with similar attributes. If a match is found, the error message is embedded to its related session entry and is allowed to pass through the firewall in order to notify the sender that the request sent is not accomplished. This costly process of stateful analysis of ICMP error messages would consume the firewall resources and prevent it from processing normal traffic when more superfluous requests are sent.

C. LIMITATIONS OF COMMON AVAILABLE LOW-RATE ATTACK MITIGATIONS

Mitigations of low-rate attacks targeting servers are made with special configurations that can track server resources allocation such as memory and CPU usage, connection tables and application threads to identify any abuse of these resources. Also, monitoring long and idle open connections, and stuck application processes would help. Indeed, configuring server connection timeout would prevent low-rate attacks. For example, a connection timeout in an Apache server can be configured as follows:

<IfModule mod_reqtimeout.c> RequestReadTimeout header=20, MinRate=500 body=20, MinRate=500 </IfModule>

This configuration will timeout any connection if the sender fails to send the header or the body data within 20 seconds each. However, as the rate of attack's traffic is reduced, the quality of service would also be degraded. This is because short timeout would wrongly disconnect legitimate connections.

Likewise, common DoF attacks are mitigated using threshold-based mechanisms, such as Screen features used in Juniper Networks [28]. Usually, these mechanisms require balanced threshold configurations, as high threshold activation values may allow attack traffic to pass through the firewall as well as low threshold values may introduce legitimate packet loss. In addition, expert knowledge is required as most of these threshold-based mechanisms are often not enabled, and even if enabled, they would dramatically increase CPU and session table utilization, such as in TCP SYN flood mechanism [29]. Furthermore, the values entered for threshold mechanisms remain indefinitely the same until updated by the administrator.

For instant, the BlackNurse attack can be mitigated by denying ICMP packets arriving at the WAN interface of the firewall. However, this would prevent inside hosts from using ping command as reply will never get back. Even if ICMP [Type: 3] messages sent to the firewall WAN interface are only denied, as recommended in [7], the ICMP Path MTU discovery will be disabled because ICMP [Type: 3, Code: 4] packets will also be denied. Contrary, thinking of using a filtering rule to block ICMP [Type: 3, Code: 3] packets is also not a valid choice, as this can affect a DNS resolver when attempting to connect to a non-existing DNS server because it never receives the ICMP port unreachable message. The following Snort rule is proposed in [7] as a solution to defend against the BlackNurse attack:

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"TDC-SOC - Possible BlackNurse attack from external source"; itype:3; icode:3; count 250, seconds 1; reference:url, soc.tdc.dk/blacknurse/blacknurse.pdf; metadata:TDC-SOC-CERT,18032016; priority:3; sid:88000012; rev:1;

However, this Snort rule is also a threshold-based solution and is applied all the time, where fake and legitimate port unreachable messages are dropped if a limit of 250 packets per second is reached.

Iptables rate limit ICMP port unreachable messages by default. This explains the reason that iptable-based products are unaffected by the BlackNurse attack [7]. The Linux 2.4.20 kernel limits destination unreachable messages to one per second in net/ipv4/icmp.c. However, this method would even prevent other benign ICMP destination unreachable messages from reaching their proper destinations, if their rates are more than the desired limit. This means that the iptable treats fake and legitimate destination unreachable messages in the same manner.

All aforementioned limitations pose a crucial need for scalable mechanism that update threshold values dynamically according to traffic behavior, attack intensity and system's needs in order to address the trade-off problem between the rule's activation period and legitimate packets loss. First, we adopt stochastic process based on continuous-time Markov Chains to model low-rate attack traffic behavior. Accordingly, Poisson counting process can be used in the proposed mechanism for identifying the transition probabilities of low-rate attack incidents.

IV. CONTINUOUS-TIME MARKOV CHAINS MODELING OF LOW-RATE DoS ATTACKS

Let the total number of low-rate DoS attacks incidents $N(t)$ by the observation time t be the state of the continuous-time Markov chain process $N(t)$; $t \geq 0$ at time t. As such, this process is characterized by the Markovian property that the conditional probability distribution of future state $N(t+r)$ of total number of attacks at time $t + r$ given the present state $N(r)$ at time r and all past states $N(h)$ at times $0 \leq h < s$ is independent of the past. In other words, the future state of the total number of attacks depends only on its present state. That is,

$$P\{N(t+r) = k|N(r) = I, N(h) = n(h)\} = P\{N(t+r) = k|N(r) = I\} \quad (1)$$

This way, we can define a continuous-time Markov chain based on Poisson counting process that have the following states: $(0, 1, 2, 3, \ldots, k, k+1, ..)$ assuming that the attack incidents are independent and happen at random time with a rate of λ. This can be achieved by dividing the observation time t into infinitesimally small values of times at which the probability of the attack (P) tends to be zero, i.e. $\lim_{P\to 0}$, while the number of attacks incidents during observation time t tends to be ∞, i.e. $\lim_{N\to\infty}$. Thus, the attacks occur at a constant rate of $\lambda = N \times P$.

This process is an example of a pure birth process since the transition occurs only in the forward direction from k to $k+1$, where $k \geq 0$ [37]. Moreover, the times between each pair of attack incidents are independent and identically distributed exponential random variable with parameter λ [38]. The transition graph of this process is shown in Fig. 10 in which each state k is connected to itself and the next state $k+1$ as only one attack incident can occur at any instant of time.

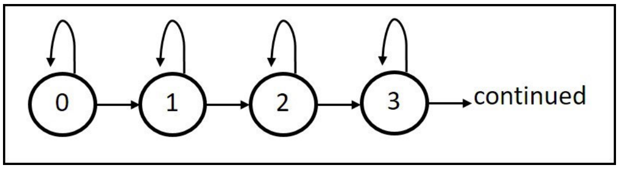


FIGURE 10. The transition graph of continuous-time Markov chain of studied attacks.

Using Champan-Kolmogorov relationship, we have the following differential equation that governs the behavior of its transition probabilities matrix $P(t)$ in terms of the jump rate matrix Q,

$$\frac{dP(t)}{dt} = P(t)Q \quad (2)$$

where the jump rate matrix Q for such a stochastic process is defined as follows by picking an infinitesimally small value of t to represent an instant of time.

$$Q = \lim_{\delta t \to 0} \frac{P(\delta t) - I}{\delta t} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ 0 & -\lambda & \lambda & 0 & \cdots \\ 0 & 0 & -\lambda & \lambda & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \end{pmatrix} \quad (3)$$

Since it is a pure birth process, all elements below the diagonal are zeros. The rates for the transition from state k to $k+i$ where $i > 1$ are also zeros because the allowed increment in the total number of attacks for a Poisson counting process is 1. Furthermore, The Q matrix has a bi-diagonal structure in which all the elements along the diagonal are equal to $-\lambda$, and all the elements on the upper diagonal are equal to λ. That is the transition rate from state k, where k is the total number of attacks, to itself is $-\lambda$, whilst it is λ for transition from state k to state $k+1$.

To find the transition probability $P_{0k}(t)$ of k low-rate DoS attacks incidents by the observation time t, we write the matrices representation of the above Champan-Kolmogorov family of equations (Eq. 2) as follows.

$$\begin{pmatrix} \frac{dP_{00}(t)}{dt} & \frac{dP_{01}(t)}{dt} & \frac{dP_{02}(t)}{dt} & \frac{dP_{03}(t)}{dt} & \cdots \\ \frac{dP_{10}(t)}{dt} & \frac{dP_{11}(t)}{dt} & \frac{dP_{12}(t)}{dt} & \frac{dP_{13}(t)}{dt} & \cdots \\ \frac{dP_{20}(t)}{dt} & \frac{dP_{21}(t)}{dt} & \frac{dP_{22}(t)}{dt} & \frac{dP_{23}(t)}{dt} & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \end{pmatrix}$$

$$\begin{pmatrix} \frac{dP_{00}(t)}{dt} & \frac{dP_{01}(t)}{dt} & \frac{dP_{02}(t)}{dt} & \frac{dP_{03}(t)}{dt} & \cdots \\ \frac{dP_{10}(t)}{dt} & \frac{dP_{11}(t)}{dt} & \frac{dP_{12}(t)}{dt} & \frac{dP_{13}(t)}{dt} & \cdots \\ \frac{dP_{20}(t)}{dt} & \frac{dP_{21}(t)}{dt} & \frac{dP_{22}(t)}{dt} & \frac{dP_{23}(t)}{dt} & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \end{pmatrix}$$

$$\begin{pmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ 0 & -\lambda & \lambda & 0 & \cdots \\ 0 & 0 & -\lambda & \lambda & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \end{pmatrix} \tag{4}$$

For example, to find $P_{00}(t)$, we first calculate its corresponding derivative $\frac{dP_{00}(t)}{dt}$ using matrix multiplication rules as in the following,

$$\frac{dP_{00}(t)}{dt} = -\lambda P_{00}(t) \tag{5}$$

Then we take the integral of both sides,

$$\int_{1}^{P_{00}(t)} \frac{dP_{00}(t)}{P_{00}(t)} = \int_{0}^{t} -\lambda dt \tag{6}$$

Finally, we simplify to get an expression for $P_{00}(t)$ as follows,

$$P_{00}(t) = e^{-\lambda t} \tag{7}$$

This shows that the probability of one or more low-rate DoS attacks by the observation time t increases as the time gets larger. Similarly, we calculate the rest of transition probabilities as follows,

$$\begin{aligned} P_{01}(t) &= \lambda t e^{-\lambda t} \\ P_{02}(t) &= \frac{\lambda^2 t^2}{2} e^{-\lambda t} \\ P_{03}(t) &= \frac{\lambda^3 t^3}{6} e^{-\lambda t} \end{aligned} \tag{8}$$

Consequently, the general term for the transition probability of k low-rate DoS attacks incidents by the observation time t given that it is a Poisson counting process is as follows,

$$P_{0k}(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \tag{9}$$

V. ICMP EARLY REJECTION RULE WITH TIME-TO-DEFEND TO MITIGATE LOW-RATE ICMP ATTACKS

We propose for each low-rate ICMP error message attack an early rejection rule that is identified by related attack type and code values. The rule is given a high priority precedency in the firewall rule base in order to be evaluated as early as possible to preserve the firewall resources during the attack. The general early rejection rule format is shown in Table 1.

TABLE 1. General format of low-rate ICMP error message early rejection rule.

Rule_Field	Value
Priority	1
Prot	1
IP Addresses: Scr_IP Dst_IP	 ∗ ∗
Type Code	x y
Action	Deny
Time	TTD seconds

The rule consists of two parts: Identifier part *<ICMP (x,y)>* and Time-to-Defend part *<TTD>*. The identifier part includes Protocol, IP addresses, Type, Code, and Action fields, where (x,y) represent valid ICMP error message type and code values, respectively. For instance, the BlackNurse attack identifier part would look like *<ICMP* (3,3)>, which corresponds to: <Prot:1, Scr_IP:∗, Dst_IP:∗, Type:3, Code:3, Action: Deny>. The time to defend part *<TTD>* represents the rule activity time duration. This determines the rule life-time for which the rule is enabled to mitigate the corresponding ICMP error message attack. The *<TTD>* is calculated dynamically depending on the attack statistics and severity parameters using the derived Poisson distribution for continuous time Markov chains, Eq. (9).

The *<ICMP* (x,y), *TTD>* rule is triggered once corresponding ICMP error messages reach a *minimum* certain rate limit, $R_{trigger}$. Beyond that *minimum* $R_{trigger}$, the firewall resources usage increases, and may become unresponsive to legitimate requests. In the case of the BlackNurse attack, $R_{trigger}$ is around 15-18 Mbit/s (40-50) k packets per second and is able to overwhelm the affected firewall's CPU regardless of the Internet connection capacity [7]. However, in [8] a 7K packets per second of the BlackNurse attack was enough to increase the CPU utilization of Juniper NetScreen SSG 20 to 89%, as shown in Fig. 11 and 12.

The process of triggering *<ICMP (x,y), TTD>* rule is illustrated in algorithm 1. The firewall extracts the ICMP error message type and code values *(x,y)*, then searches the session table for a related session using the payload data. If a related

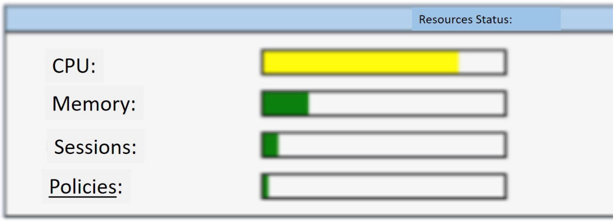
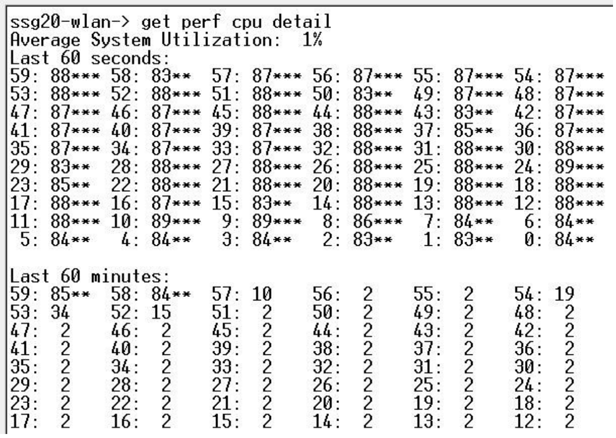


FIGURE 11. The effect of the BlackNurse attack on Juniper SSG 20 CPU performance.



```
ssg20-wlan-> get perf cpu detail
Average System Utilization:  1%
Last 60 seconds:
59: 88*** 58: 83**  57: 87*** 56: 87*** 55: 87*** 54: 87***
53: 88*** 52: 88*** 51: 88*** 50: 83**  49: 87*** 48: 87***
47: 87*** 46: 87*** 45: 88*** 44: 88*** 43: 83**  42: 87***
41: 87*** 40: 87*** 39: 87*** 38: 88*** 37: 85**  36: 87***
35: 87*** 34: 87*** 33: 87*** 32: 88*** 31: 88*** 30: 88***
29: 83**  28: 88*** 27: 88*** 26: 88*** 25: 88*** 24: 89***
23: 85**  22: 88*** 21: 88*** 20: 88*** 19: 88*** 18: 88***
17: 88*** 16: 87*** 15: 83**  14: 88*** 13: 88*** 12: 88***
11: 88*** 10: 89***  9: 89***  8: 86***  7: 84**   6: 84**
 5: 84**   4: 84**   3: 84**   2: 83**   1: 83**   0: 84**

Last 60 minutes:
59: 85**  58: 84**  57: 10    56:  2    55:  2    54: 19
53: 34    52: 15    51:  2    50:  2    49:  2    48:  2
47:  2    46:  2    45:  2    44:  2    43:  2    42:  2
41:  2    40:  2    39:  2    38:  2    37:  2    36:  2
35:  2    34:  2    33:  2    32:  2    31:  2    30:  2
29:  2    28:  2    27:  2    26:  2    25:  2    24:  2
23:  2    22:  2    21:  2    20:  2    19:  2    18:  2
17:  2    16:  2    15:  2    14:  2    13:  2    12:  2
```

FIGURE 12. Juniper NetScreen SSG 20 CPU status during the BlackNurse attack.

Algorithm 1 rule_Trigger(P, $R_{trigger}$, $TTD(t)$)

```
Input: ICMP error message $P$, $R_{trigger}$.
Output: True if <ICMP (x,y), TTD> rule is triggered,
        False otherwise.
$X \leftarrow P$.type
$y \leftarrow P$.code
$s \leftarrow$ findSession($P$)
if $s \neq NULL$ then                    // P is not fake
    deleteSessions($s$)
    forward($P$, $P$.getinitiatorSrc())
    return false
else
    $r \leftarrow r + 1$
    if $r == R_{trigger}$ then
        tiggerICMP($x, y, TTD(t)$)
        r$\leftarrow$ 0
        return true
    end if
    return false // P is fake but r has not
    reached $R_{trigger}$.
end if
```

session is found, the firewall deletes the session to free its resources, and forwards the error message to the initiator. However, if no related session is found the counter r for this type of error message is incremented, and if it reaches $R_{trigger}$ indicating an attack is undergoing, then the corresponding *<ICMP (x,y)>* rule is triggered for *<TTD>* seconds.

For the rule activity time part *<TTD>*, recall that textitTTD is defined as the time needed for a countermeasure c (e.g. access control policy) to gain some level of control on some attack vector A (e.g. BlackNurse) [9].

Algorithm 2 **calc_TTD**($TTD(t)$, $TTD(t-1)$, $R_{max}(t)$, $R_{max}(t-1)$))

```
Input: $TTD(t)$ and $TTD(t-1)$, current and previous
       maximum attack rate $R_{max}(t)$ and $R_{max}(t-1)$.
Output: $TTD(t+1)$ to be used next time once <ICMP
        (x,y)> is triggered.
Calculate $\alpha(t)$ using Eq.(10)
Calculate $P(t)$ using Eq.(11)
Calculate $\beta(t)$ using Eq.(12)
Calculate $TTD(t+1)$ using Eq.(13)
return $TTD(t+1)$
```

We use $t-1, t, t+1$ to represent parameters in previous, current and next observation windows of time, respectively. Given the current $TTD(t)$ value, the current and previous attack statistics and severity parameters, algorithm 2 objective is to predict $TTD(t+1)$ that will be used next time, once *<ICMP (x,y)>* rule is triggered. The current and previous attack statistics and severity parameters are defined as follows

$$\alpha(t) = (R_{max}(t) - R_{max}(t-1))/R_{max}(t-1) \tag{10}$$

$$P(t) = 1 - e^{-\lambda TTD(t)} \tag{11}$$

$$\beta(t) = (P(t) - P(t-1))/P(t-1) \tag{12}$$

where:

$R_{max}(t)$: is the maximum rate of the $ICMP(x, y)$ attacks during current observation window t.

$R_{max}(t-1)$: is the maximum rate of the $ICMP(x, y)$ attacks during previous observation window $t-1$.

$P(t)$: is the Poisson distribution of the $ICMP(x, y)$ attacks during current observation window t, with mean $\lambda TTD(t)$.

$P(t-1)$: is the Poisson distribution of the $ICMP(x, y)$ attacks during previous observation window $t-1$, with mean $\lambda TTD(t-1)$.

$R_{max}(t-1)$ and $P(t-1)$ were sufficient to induce $TTD(t)$. Hence, we use the change in R_{max} and P relative to their original values to update $TTD(t+1)$ as follows:

$$TTD(t+1) = (1 + \alpha(t) + \beta(t))TTD(t) \tag{13}$$

where:

$TTD(t)$: is the current rule activity time duration in seconds during current observation window t, initialized to $TTD(0)$ in the beginning.

$TTD(t+1)$: is the predicted rule activity time duration in seconds to be used in the next observation window $t+1$, once *<ICMP (x,y)* > rule is triggered.

VI. EVASION ATTACK AGAINST TIMED EARLY REJECTION RULE

The previous work on the BlackNurse attack detection and prevention [9] is vulnerable to a special type of attack, where the attacker can try to progressivley increase the rule activity duration time. Consequently, this will increase corresponding legitimate ICMP packet loss. For instance, in the worst case, the BlackNurse early rejection rule *<ICMP (3,3), TTD>* will be activated all the time, rejecting all incoming *ICMP* packets of Type: 3, Code: 3.

An attacker may manipulate the attack rate, so that $TTD(t)$ keeps increasing. This can be done by continuously increasing $R_{max}(t)$. Fig. 13 shows the resulting effect.

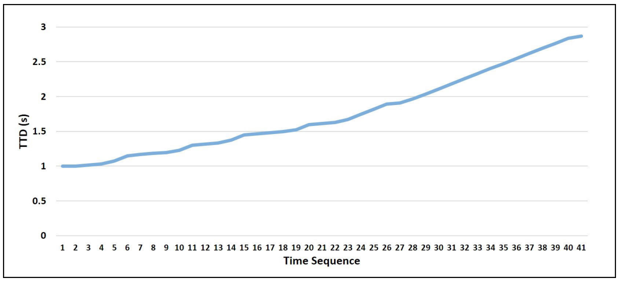


FIGURE 13. ***TTD*(*t*) can increase indefinitely if attacker keeps increasing the attack rate, $R_{max}(t)$.**

Here, the goal of the attacker is to increase $TTD(t)$, and thus make the firewall deny legitimate packets for longer periods of time. Fig. 14 shows the effect by plotting amount of loss in legitimate packets due to the increase of $TTD(t)$ manipulated by the attacker.

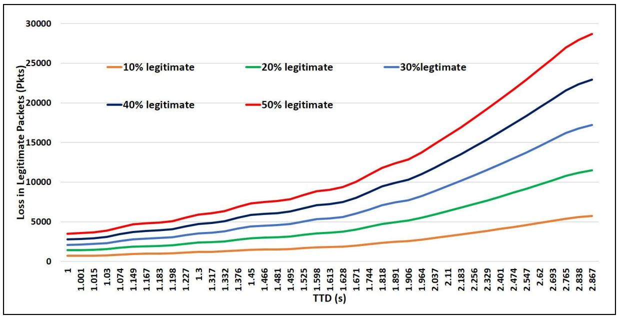


FIGURE 14. **Loss in legitimate packets increases when TTD increases.**

As seen in Fig. 14, as the percentage of legitimate packets increases, the total number of legitimate packets loss also increases. For example, if 10% packets are legitimate, then the number of legitimate packets lost for $TTD = 2.5$ is about 4,200, but if 50% are legitimate, then the total loss is 21600 for the same *TTD*. From this analysis, we infer that if the attacker keeps increasing the *TTD*, it will approach infinity and ultimately cause the firewall to deny all legitimate packets. Therefore, in this work, we propose a mechanism to detect this kind of attack and avoid *TTD* to increase indefinitely. Next section details this approach.

VII. PROPOSED MODIFIED MECHANISM TO MITIGATE ICMP LOW-RATE ATTACKS

The main idea behind mitigation of the aforementioned attack is to set a threshold on *<TTD>* so that it cannot increase indefinitely. We propose three techniques to adaptively choose the threshold. Following subsections describe them in detail.

```
Algorithm 3 TTD_Adjust($TTD(t), TTD_U, \zeta(t)$)
  Input: $TTD(t), TTD_U, \zeta(t)$
  Output: Updated $TTD(t+1)$
  $TTD(t+1) \leftarrow$
  calc_TTD($TTD(t), TTD(t-1), R_{max}(t), R_{max}(t-1)$)
  if $TTD(t+1) > TTD_U$ then
      Calculate $\zeta(t)$ using Eq.14
      Update $TTD(t+1)$ using Eq.15 // Update the
      start index needed for next
      adjustment.
      $s \leftarrow t+1$
  end if
  return $TTD(t+1)$
```

A. TTD ADJUSTMENT PROCESS

Algorithm 3 illustrates the process of *TTD* adjustment to eliminate evasion attack impact using an upper threshold. Let *<TTD>* has an upper threshold, TTD_U. If the predicted $TTD(t+1)$ in Eq. (13) is larger than TTD_U, then it is adjusted by an amount ζ, computed using the following equation:

$$\zeta(t) = \frac{\delta \sum_{i=s}^{t}(TTD(i+1) - TTD(0))^2}{t-s+1} \tag{14}$$

where:

$\zeta(t)$: is the correction offset at time t.

δ: is an adjustable weight parameter.

$t - s + 1$: is the number of time-stamps passed since the last $TTD(t+1)$ correction. In other words, the last $TTD(t+1)$ correction was at time $t - s$.

$TTD(0)$: is the reference $TTD(t)$ value.

Eq. (14) computes the *mean squared* error, weighted by δ, for the last $TTD(t+1)$ values from s until t, with respect to the reference $TTD(0)$. If the $TTD(t+1)$ value is higher than TTD_U, it is reduced by $\zeta(t)$ as follows:

$$TTD(t+1) = TTD(t+1) - \zeta(t) \tag{15}$$

Algorithm 3, calls algorithm 2 which is responsible for calculating $TTD(t+1)$. If the calculated $TTD(t+1)$ is greater than TTD_U, then $\zeta(t)$ is computed and $TTD(t+1)$ is adjusted accordingly. By this, *<TTD>* can be protected against attacker evasion attempt to increase it indefinitely, and accordingly legitimate packet loss is preserved.

We combine the proposed early rule triggering process and *TTD* adjustment algorithms to form an effective collaborative defense mechanism against low-rate ICMP error

Algorithm 4 thwart_low_rate_ICMP( )

```
$t = r = s \leftarrow 0$
while true do
    $P \leftarrow$ nextICMP_Error_Packet
    trigger $\leftarrow$ rule_Trigger($P$, $R_{trigger}$, $TTD(t)$)
    if trigger==true then
        $TTD(t+1) \leftarrow$ TTD_Adjust($TTD(t)$, $TTD_U$, $\zeta(t)$)
        $t \leftarrow t+1$
    end if
end while
```

message threats. Algorithm 4 illustrates the overall concatenation for the previous proposed algorithms. The firewall continues filtering incoming packets, once ICMP error message P is received, *rule_Trigger* algorithm is invoked to keep trace of P until it is either forwarded to the initiator to inform for an error, or P is tracked to check if more of such packets are received previously indicating a low-rate attack is undergoing. If the later was the case, <*ICMP (x,y), TTD*> rule is triggered to preserve firewall resources. Following that *TTD_Adjust* algorithm is invoked to keep track of the corresponding rule activity time, and dynamically complete required calculations to adjust <*TTD*> in case of increment attack evasion. Finally, the system behavior and time-to-defense are predicted dynamically to be used in case of similar attacks happen in the future.

B. METHODOLOGY TO DETERMINE THE UPPER THRESHOLD TTD_U

Algorithm 3 calibrates the predicted $TTD(t+1)$, to be used in the future in case of similar attacks, against an upper threshold $TTDU$. If $TTD(t+1) > TTDU$, then adjustment process follows accordingly. In this section we propose methods for determining TTD_U.

1) BUSINESS MODEL THRESHOLD

The business model threshold is based on a predefined upper bound on the number of legitimate packets lost. For example, if the business model restricts the number of legitimate packet loss to maximum 1000, then according to Fig. 14, $TTD(t+1)$ must be 1.613 or less. Therefore, we will choose $TTD_U = 1.613$ in algorithm 3, and apply the TTD adjustment process accordingly.

2) POSITIVE INCREMENT THRESHOLD

The positive increment threshold is based on an upper limit on the number of times $TTD(t+1)$ increases consecutively. Let this limit be L. Then TTD_U can be determined using algorithm 5. If the number of consecutive increments of $TTD(t+1)$ values is L, then we set the TTD_U to the last $TTD(t+1)$ value before the consecutive increments happen.

Algorithm 5 PosIncr_Threshold(L)

```
Input: $L$// Limit on the number of times
       $TTD(t+1)$ increases consecutively.
Output: $TTD_U$// The positive increment
        threshold.
if $t$=0 then
    temp ←0
    ctr ←0// Initialize counter ctr.
end if
if $TTD(t+1) > TTD(t)$ then
    $ctr = ctr + 1$
    if $ctr == L$ then
        $TTD_U = TTD(temp)$
        ctr=0
    end if

else
    $ctr \leftarrow 0$
    $temp \leftarrow t$
end if
return $TTD_U$
```

The variable *ctr* is used to keep track of the number of consecutive increments.

3) ATTACK PROBABILITY THRESHOLD

This threshold is based on an upper limit on $P(t)$, which is the probability of attack during $TTD(t)$ period. From Eq. (11), we can infer that as $TTD(t)$ tends to infinity, $P(t)$ approaches 1, and β approaches zero (Eq. 12). Therefore, an upper limit on $P(t)$ will also set an upper limit to TTD. Let the upper limit on $P(t)$ be P_{MAX}. This follows that:

$$P(t) \leq P_{MAX}$$

or, $1 - e^{-\lambda TTD(t)} \leq P_{MAX}$, using Eq. (11).

Thus, $TTD(t) \leq -ln(1 - P_{MAX})/\lambda$.

Therefore, TTD_U in algorithm 3 can be set to:

$$TTD_U = -ln(1 - P_{MAX})/\lambda \tag{16}$$

VIII. PERFOMANCE EVALUATION

The proposed algorithms to defend against low-rate ICMP error message attacks are implemented using Java programming language. Despite that during some time intervals where low-rate attacks are not launched and therefore the triggering process is not accomplished, <*TTD*> calculations and adjustments are kept from the last previous time interval during which the corresponding low-rate attack had occurred. The parameters λ and $R_{max}(t)$ are used in the proposed model to define the attacker skill level. For each occurrence of a low-rate ICMP attack that trigger the corresponding <*ICMP (x,y), TTD*> rule, *TTD(t + 1)* is calculated to be used in the next time for similar attack type occurrence. For instance, we simulated the proposed mechanism for expert attacker skill level since this type of attacker has high experience and tries to manipulate $R_{max}(t)$ in order to evade the system and increase $<TTD>$ indefinitely. Thus, in this section we evaluate the effectiveness of the proposed method in thwarting the low-rate ICMP error message attacks and analyze its performance under different parameter settings. The triggering rate $R_{trigger}$ is set as 7K packets per second. All experiments parameters are plotted for 40 consecutive time sequences.

A. TTD ADJUSTMENT

In this section we show how $TTD(t+1)$ are adjusted based on three proposed threshold methods for $TTD(t+1)$ adjustment, as proposed in Section VII(B).

1) ADJUSTMENT BY BUSINESS MODEL THRESHOLD

As explained before, the business model threshold is based on a predefined upper bound on the number of legitimate packets lost.

In this experiment, we show how effectively $TTD(t + 1)$ adjustment process works using this threshold. The results are shown in Fig. 15 for different values of δ between 0.5 and 0.9, and $TTD_U = 1.6(s)$, which was selected based on business requirement as explained in Section VII(B)

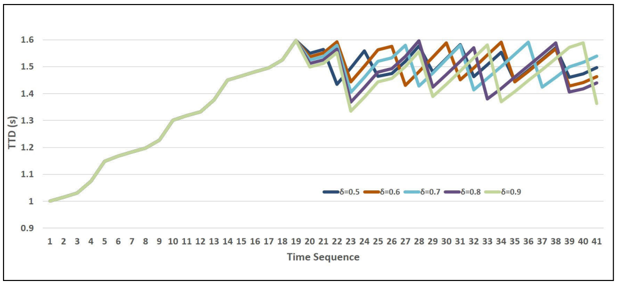


FIGURE 15. Effect of ***TTD*(*t* + 1)** correction using algorithm 3. The graphs show TTD_U = 1.6(*s*), and updated value of ***TTD*(*t*)** over time.

As shown in Fig. 15, from the time TTD(t + 1) reaches the threshold TTDU, TTD(t) values fluctuate between the threshold and a lower value, but never exceed the threshold. Besides, the parameter controls how far the TTD(t + 1) value can go down after it reaches the threshold using Eq.15. For example, higher values of tend to dip TTD(t + 1) higher, and vice-versa.

2) ADJUSTMENT BY POSITIVE INCREMENT THRESHOLD

Recall that positive increment threshold (L) puts an upper bound on the number of consecutive increment of TTD(t + 1). We experimented with different values of L, the results of which are shown in Fig. 16.

We observe that for lower values of L, the threshold is adjusted more frequently and vice-versa. Furthermore, for higher values of L, the TTD(t + 1) value may also reach much higher (e.g. 2.6 as shown in the figure).

3) ADJUSTMENT BY ATTACK PROBABILITY THRESHOLD

As mentioned in Section VII(B), this threshold (P_{MAX}) is based on an upper limit on P(t), which is the probability of

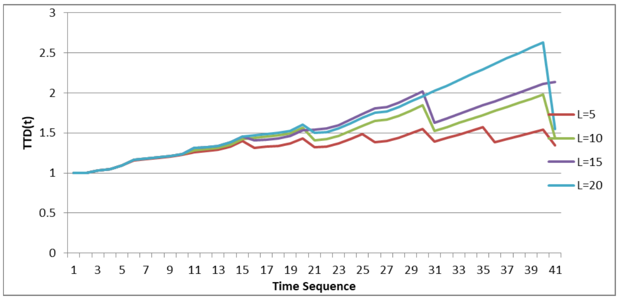


FIGURE 16. Effect of ***TTD*(*t* + 1)** correction using positive increment threshold (L) for different values of L between 5 to 20.

attack during TTD(t) period. We have conducted the experiment with different values of P_{MAX} ranging from 0.995 to 0.9999995, and the results are shown in Fig. 17.

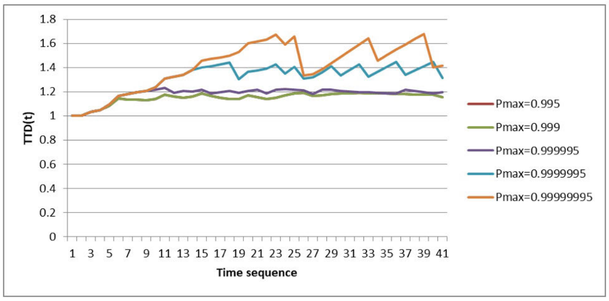


FIGURE 17. Effect of ***TTD*(*t* + 1)** correction using positive increment threshold (L) for different values of L between 5 to 20.

It is evident from Fig. 17 that higher values of PMAX result in not only higher values of TTD_u but also less frequent adjustment of TTD(t + 1).

Comparing the three different thresholds we can observe that they exhibit a similar trend: higher values of threshold result in less frequent adjustment of TTD(t + 1), which in turn lead to more legitimate packet loss. However, less frequent adjustment can also save processing time. Therefore, this tradeoff should be considered in choosing the threshold values. On the other hand, each threshold has its own merit based on specific circumstances and priorities. For example, if business requirement is very well defined then we can choose specific value of the business threshold that meets the business policy. On the contrary, if the attack probability is very high, then we opt to choose the maximum probability threshold P_{MAX}. This is because higher attack probability means the value of β(t) approaches zero and thus the frequency of attack does not affect the adjustment of TTD(t + 1) value (refer to Eq. 13). Therefore, choosing P_{MAX} as the threshold, we try to avoid this scenario. Finally, if none of the above scenarios apply, we can choose the positive increment threshold (L).

B. EFFECT OF PARAMETER λ

Fig. 18 shows the effect of varying parameter λ between 1 and 10 on $TTD(t)$. It can be seen that higher values of λ tend

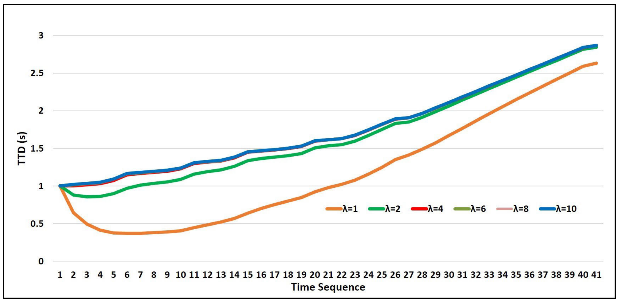


FIGURE 18. The Effect of λ on the *TTD*(*t*) growth rate.

to increase $TTD(t)$ more rapidly towards the threshold. This is important desirable property of proposed adaptive Time-To-Defend ($TTD(t)$) because it models that fact that the defense system reacts dynamically to the level of skills of attacker modeled by values of λ. In this fashion, $TTD(t)$ is adapted such that the likelihood of successful low-rate ICMP error message attacks and their impact on the firewall performance tend to be decreased.

C. INFLUENCE OF λ ON PARAMETER β

In this experiment, we analyze the effect on β by changing parameter λ between 1 and 10. As seen in the previous experiment (Fig. 16), $TTD(t)$ increases more rapidly for a higher value of λ. As a result, $P(t)$ also increases (Eq.11), and approaches 1, as $TTD(t)$ approaches infinity (Section VII-B-3). Therefore, as Fig. 19 shows, β reaches 0 more quickly for higher values of λ (Eq.12). This is because β measures the change in probabilities between two consecutive time points.

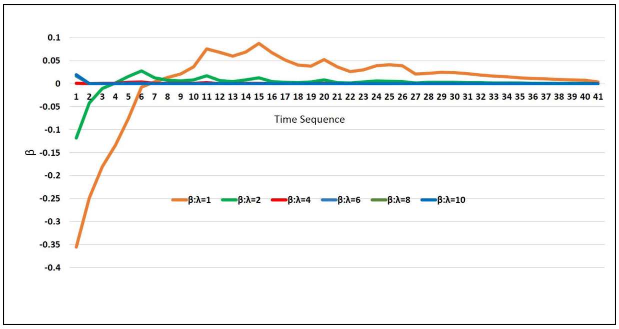


FIGURE 19. How β is affected by λ and *TTD*(*t*).

D. MINIMIZING LEGITIMATE PACKET LOSS

This experiment studies the effect of using the $TTD(t+1)$ adjustment algorithm with $TTD_U = 1.6(s)$ in minimizing legitimate packet loss for different values of between 0.5 and 0.9. As indicated in experiment A on $TTD(t)$ adjustment, greater values of tends to have greater affect in $TTD(t)$ adjustment. Also, this would affect accordingly legitimate packet loss as shown in Fig. 20. To be specific Table 2 compares accumulative legitimate packet loss without using $TTD(t)$ adjustment in prior work [9] with $TTD(t)$ adjustment mechanism proposed in this paper for different ratios of legitimate packets relative to low-rate DoS attack traffic (10%-50%). It is clearly shown that $TTD(t)$ adjustment tends to minimize legitimate packet loss by 25.3-26.4% when δ values vary between 0.5 and 0.9, due to the correction offset, $\zeta(t)$.

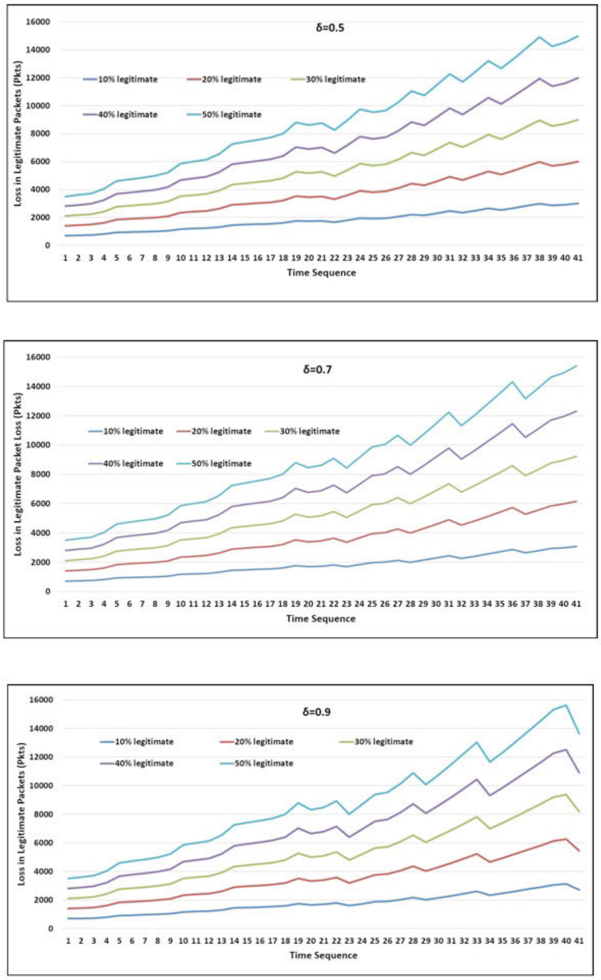


FIGURE 20. Loss in legitimate packets for TTD(t + 1) adjustment using algorithm 3 when δ values vary between 0.5-0.9.

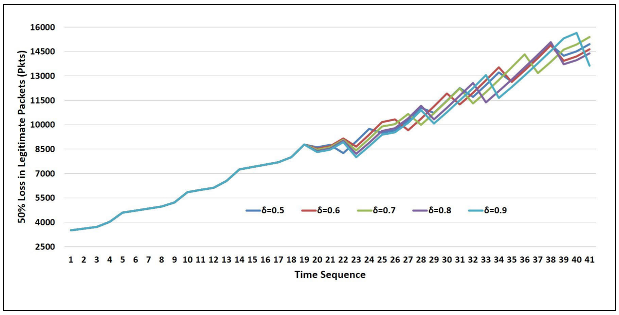


FIGURE 21. Effect of *TTD*(*t* + 1) adjustment using algorithm 3 for different δ values, and coresponding values of legitimate packet loss over time.

TABLE 2. Accumulative legitimate packet loss when using TTD with and without adjustment algorithm.

Legitimate Packets Ratio	Legitimate packets Loss (Pkts)					
	TTD without adjustment [8]	TTD with adjustment				
		δ=0.5	δ=0.6	δ=0.7	δ=0.8	δ=0.9
10%	97966	73153	73110	72996	72261	72055
20%	195931	146305	146219	145992	144522	144109
30%	293896	219457	219328	218987	216783	216163
40%	391861	292609	292437	291983	289044	288218
50%	489826	365762	365547	364979	361305	360272
Percentage gain in minimizing packet loss using TTD Adjustment		25.33%	25.37%	25.49%	26.24%	26.45%

Fig. 21 shows a specific example for minimizing legitimate packet loss when 50% of traffic is legitimate for δ values vary between 0.5-0.9.

IX. CONCLUSION

LDDoS attacks are usually difficult to detect as they occupy low bandwidth. These attacks are often shorter in duration and can be launched using a single computer. As they use low-volume of malicious traffic that looks alike legitimate traffic, they can enter the network in silence depleting its resources and degrading the performance. DoF attacks are special type of DDoS attacks that target firewalls. Their effect can appear as increase in packet filtering time, CPU utilization, memory usage and the number of allocated sessions in the session table. This may make the firewall unresponsive and degrade its performance. Low-rate ICMP error message attacks such as the BlackNurse attack, target firewall devices using low volume of ICMP error message [Type:3, Code:3].

This paper proposes a mechanism to defend against low-rate ICMP error message attacks that are intended to degrade firewalls performance. The mechanism is based on generating an early rejection rule with the format: *<ICMP (x,y), TTD>* that is identified by related ICMP error message attack type and code values. The rule has an activity duration, *TTD* that is determined based on current and previous attack severity and statistical parameters. The mechanism includes an easy to implement algorithms to address the natural trade-off between firewall rule's activation period and the legitimate packets loss. It achieves that by preventing incremental evasion cycle of low-rate attacks that can cause severe increment in the rejection rule activity time and therefore increase legitimate packet loss.

Experiments are conducted to simulate the proposed mechanism against expert attacker behavior. Despite expert attacker trials to manipulate $R_{max}(t)$ in order to evade the system and increase <*TTD*> indefinitely, the proposed model ensures that *TTD(t)* values fluctuate between a predefined threshold TTD_U and a lower value controlled by parameter δ using Eq.15, but never exceeding TTD_U. The experiments confirm that the *TTD* adjustment model tends to be stable without additional overhead and increment in the rule defense duration, regardless of the nature and the number of attack botnets. The proposed mechanism tends to minimize legitimate packet loss by 25.3-26.4% when values vary between 0.5 and 0.9. For future work, we intend to generalize the idea of early rejection rule with time to defend duration to cover other classes of low-rate DoF attacks. Iptables can be used as open source firewall to implement the proposed mechanism.

REFERENCES

[1] A. X. Liu, A. R. Khakpour, J. W. Hulst, Z. Ge, D. Pei, and J. Wang, "Firewall fingerprinting and denial of firewalling attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1699–1712, Jul. 2017.

[2] Z. Trabelsi, K. Hayawi, A. Al Braiki, and S. S. Mathew, *Network Attacks and Defenses: A Hands-On Approach*. Boca Raton, FL, USA: Auerbach, 2012.

[3] S. Prabhakar, "Network security in digitalization: Attacks and defence," *Int. J. Res. Comput. Appl. Robot.*, vol. 5, no. 5, pp. 46–52, May 2017.

[4] K. Salah, K. Sattar, M. Sqalli, and E. Al-Shaer, "A potential low-rate DoS attack against network firewalls," *Secur. Commun. Netw.*, vol. 4, no. 2, pp. 136–146, Feb. 2011.

[5] S. Weagle. (Jul. 2017). *Short, Low-Volume DDoS Attacks Pose Greatest Security and Availability Threat to Businesses*. [Online]. Available: https://www.itproportal.com/features/short-low-volume-ddos-attacks-pose-greatest-security-and-availability-threat-to-businesses/

[6] S. Newman. (2016). *DDoS Attacks on the Rise Again and Size Doesn't Matter*. TDC Security Operation Center. [Online]. Available: http://soc.tdc.dklblacknurse/blacknurse.pdf

[7] L. Hansson, P. Hogh, B. Bachmann, K. Jor-gensen, and D. Rand. (2016). *The BlackNurse Attac*. TDC Security Operation Center. [Online]. Available: http://soc.tdc.dklblacknurse/blacknurse.pdf

[8] S. Khandelwal. (Nov. 2016). *Even A Single Computer Can Take Down Big Servers Using BlackNurse Attack*. [Online]. Available: http://thehackernews.com/2016/11/dos-attack-server-firewall.html

[9] Z. Trabelsi, S. Zeidan, and K. Hayawi, "Denial of firewalling attacks (DoF): The case study of the emerging BlackNurse attack," *IEEE Access*, vol. 7, pp. 61596–61609, 2019.

[10] E. Chabrow. *Researchers' Goal: Mitigate DDoS Attacks Within 10 Seconds*. [Online]. Available: https://www.bankinfosecurity.com/researchers-hope-to-mitigate-ddos-attacks-within-10-seconds-a-9091

[11] L. Zhou, M. Liao, C. Yuan, and H. Zhang, "Low-rate DDoS attack detection using expectation of packet size," *Secur. Commun. Netw.*, vol. 2017, pp. 1–14, Oct. 2017.

[12] P. N. Jadhav and B. M. Patil, "Low-rate DDOS attack detection using optimal objective entropy method," *Int. J. Comput. Appl.*, vol. 78, no. 3, pp. 33–38, 2013.

[13] 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.

[14] 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 (ICNP)*, Oct. 2004, pp. 196–205.

[15] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, "On a mathematical model for low-rate shrew DDoS," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1069–1083, Jul. 2014.

[16] W. Wang, R. Ji, W. Chen, B. Chen, and Z. Li, "Firewall rules sorting based on Markov model," in *Proc. 1st Int. Symp. Data, Privacy, E-Commerce (ISDPE)*, Nov. 2007, pp. 203–208.

[17] W. Wang, H. Chen, J. Chen, and B. Liu, "Firewall rule ordering based on statistical model," in *Proc. Int. Conf. Comput. Eng. Technol.*, Jan. 2009, pp. 185–188.

[18] Z. Trabelsi, L. Zhang, and S. Zeidan, "Dynamic rule and rule-field optimisation for improving firewall performance and security," *IET Inf. Secur.*, vol. 8, no. 4, pp. 250–257, Jul. 2014.

[19] Z. Trabelsi, L. Zhang, S. Zeidan, and K. Ghoudi, "Dynamic traffic awareness statistical model for firewall performance enhancement," *Comput. Secur.*, vol. 39, pp. 160–172, Nov. 2013.

[20] H. Hamed, A. El-Atawy, and E. Al-Shaer, "On dynamic optimization of packet matching in high-speed firewalls," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 10, pp. 1817–1830, Oct. 2006.

[21] H. Hamed, A. El-Atawy, and E. Al-Shaer, "Adaptive statistical optimization techniques for firewall packet filtering," in *Proc. 25TH IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2006, pp. 1–12.

[22] A. El-Atawy, T. Samak, E. Al-Shaer, and H. Li, "Using online traffic statistical matching for optimizing packet filtering performance," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2007, pp. 866–874.

[23] A. El-Atawy, E. Al-Shaer, T. Tran, and R. Boutaba, "Adaptive early packet filtering for defending firewalls against DoS attacks," in *Proc. 28th Conf. Comput. Commun. (INFOCOM)*, Apr. 2009, pp. 1–9.

[24] T. Chomsiri, X. He, P. Nanda, and Z. Tan, "A stateful mechanism for the tree-rule firewall," in *Proc. IEEE 13th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sep. 2014, pp. 122–129.

[25] T. Chomsiri, X. He, P. Nanda, and Z. Tan, "Hybrid tree-rule firewall for high speed data transmission," *IEEE Trans. Cloud Comput.*, early access, Apr. 14, 2016, doi: 10.1109/TCC.2016.2554548.

[26] Z. Trabelsi and S. Zeidan, "Multilevel early packet filtering technique based on traffic statistics and splay trees for firewall performance improvement," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 1089–1093.

[27] Z. Trabelsi, S. Zeidan, M. M. Masud, and K. Ghoudi, "Statistical dynamic splay tree filters towards multilevel firewall packet filtering enhancement," *Comput. Secur.*, vol. 53, pp. 109–131, Sep. 2015.

[28] (Sep. 2015). *Denial-of-Service Attacks Feature Guide for Security Devices*. [Online]. Available: http:www.juniper.net/techpubs/en_US/junos12./security-attack-denial-of-service.pdf

[29] Redwolfsecurity.com. *The Next BIG Thing: 'Small' DDoS Attacks are Often Hardest to Block*. Accessed: Dec. 9, 2016. [Online]. Available: https://www.redwolfsecurity.com/small-ddos-attacks/

[30] Z. Trabelsi, S. Zeidan, K. Shuaib, and K. Salah, "Improved session table architecture for denial of stateful firewall attacks," *IEEE Access*, vol. 6, pp. 35528–35543, 2018.

[31] X. Li, Z. Z. Ji, and M. Z. Hu, "Session table architecture for defending SYN flood attack," in *Proc. 7th Int. Conf. Inf. Commun. Secur. (ICICS)*, 2005, pp. 220–230.

[32] S. Suroto, "A review of defense against slow HTTP attack," *Int. J. Informat. Vis.*, vol. 1, no. 4, pp. 127–134, 2017, doi: 10.30630/joiv.1.4.51.

[33] S. Kumar. (2012). *How Slow Http Can Knock Down a Server?* Geeks website. [Online]. Available: http://www.geeksforgeeks.org/slow-http-can-knock-server/

[34] L. Perlman. (2018). *Slow Loris—Rethinking DoS Attacks*. [Online]. Available: https://medium.com/front-end-weekly/slow-loris-rethinking-dos-attacks-bd1ca5091bfe

[35] M. Najafabadi, T. Khoshgoftaar, A. Napolitano, and C. Wheelus, "RUDY-Attack: Detection at the network level and its important features," in *Proc. 29th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2016, pp. 282–287.

[36] S. Tayama and H. Tanaka, "Analysis of slow read DoS attack and communication environment," in *Proc. Int. Conf. Mobile Wireless Technol.*, 2017, pp. 350–359.

[37] S. M. Ross, *Introduction to Probability Models*. New York, NY, USA: Academic, 2014.

[38] S. Karlin, *Arst Course in Stochastic Processes*. New York, NY, USA: Academic, 2014.

[39] (2019). *What is the ICMP (Internet Control Message Protocol)?* [Online]. Available: https://www.ionos.com/digitalguide/server/know-how/what-is-icmp-protocol-and-how-does-it-work/

KADHIM HAYAWI received the M.Sc. degree in computer science from Dalhousie University, Canada, in 2004, and the Ph.D. degree from the University of Waterloo, Canada, in 2018. He is currently an Assistant Professor with the College of Technological Innovation, Zayed University, United Arab Emirates, and a member of Cyber Security and Digital Forensics research group, where he teaches a wide variety of courses, and pursues his research endeavors. He earned several prestigious industry certifications, and has over 17 years of experience in academia and industry. His research interests are in information security and privacy challenges of emerging technologies, such as the IoT, Cloud, social networks, blockchain, zero trust architecture, digital health, and healthcare data analytics.

ZOUHEIR TRABELSI received the Ph.D. degree in computer science from the Tokyo University of Technology and Agriculture, Japan. He is currently a Professor of network security with the College of Information Technology (CIT), United Arab Emirates University (UAEU). Prior joining UAEU, he was a Computer Science Researcher at the Central Research Laboratory of Hitachi, Tokyo, Japan, for four years. His research interests include zero trust architecture, network security, intrusion detection and prevention, firewalls, network covert channels, information security education, and curriculum development.

SAFAA ZEIDAN received the B.Sc. degree (Hons.) in computer engineering from the University of Sharjah, United Arab Emirates. She is currently a Research Assistant with the College of Information Technology, United Arab Emirates University (UAEU). She has around 13 publications in well-known conferences and journals. Her research interests focus mainly on firewall optimization techniques during normal and attack situations.

MOHAMMAD MEHEDY MASUD (Member, IEEE) received the Ph.D. degree in computer science from The University of Texas at Dallas, USA, in December 2009. He is currently an Associate Professor with the College of Information Technology (CIT), United Arab Emirates University (UAEU). Prior to joining UAEU in January 2012, he worked at the University of Texas at Dallas as a Research Associate for two years. His research interests include Big Data mining, healthcare data analytics, e-health, data streammining, and machine learning.