IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 25, 2019, accepted April 28, 2019, date of publication May 9, 2019, date of current version May 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915792

Denial of Firewalling Attacks (DoF): The Case Study of the Emerging BlackNurse Attack

ZOUHEIR TRABELSI, SAFAA ZEIDAN, AND KADHIM HAYAWI
College of Information Technology, United Arab Emirates University, Al-Ain, United Arab Emirates

Corresponding author: Safaa Zeidan (safaa.z@uaeu.ac.ae)

This work was supported by UAE University UPAR, under Grant 31T080.

ABSTRACT Traditional Distributed Denial of Service (DDoS) attacks usually flood network targets with malicious traffic. Recently, new types of DDoS attacks have emerged and target specifically network security devices, mainly firewalls and intrusion prevention systems (IPS). In contrast to traditional DDoS attacks, these emerging attacks use a low volume of malicious traffic. This paper is concerned solely with an emerging denial of firewalling attack (DoF), called the BlackNurse attack. The attack uses specially formatted ICMP error messages to overwhelm targeted firewalls' CPUs. This paper offers detailed insights into the understanding of DoF attacks and classifying them according to the targeted firewall resources, traffic volume, and attack effect. This paper also concentrates on the BlackNurse attack principles, practical attack generation, and its general effect on impacted firewalls and the networks. The performance evaluations are conducted on commercial grades, namely, Juniper NetScreen SSG 20 and Cisco ASA 5540 firewalls. The pros and cons of the available attack mitigations are discussed. OS screening features on Juniper NetScreen SSG 20 are used, for an example, to test their effectiveness in thwarting the attack. Furthermore, this paper proposes a novel mechanism to defend against the BlackNurse attack using an early rejection rule with dynamic activity time duration that depends on current and previous attack statistics and severity parameters. The evaluation is conducted to simulate the proposed mechanism defense against novice and expert BlackNurse attackers.

INDEX TERMS DDoS attack, DoF attack, BlackNurse attack, stateful firewall, session table ICMP error messages.

I. INTRODUCTION

In DDoS attacks, attackers intend to overwhelm network system resources or services until legitimate requests cannot be processed any more. This is usually accomplished using a net of compromised clients (botnet) that were previously infected with malwares to make them under malicious control. The common goal is to overload the target with massive traffic until it becomes slow or unresponsive to legitimate requests.

Firewalls are the first line of defense against DDoS attacks and threats targeting networks and services. The primary functionality of a firewall is to filter traffic routed in and out of a network. This is done according to predefined filtering policy rules, which are typically constructed to allow or deny a packet to pass through, depending on the packet's header information (Protocol (Prot), IP source (Src-IP), source Port (Src-P), IP destination (Dst-IP), destination Port (Dst-P).

However, firewalls themselves are also susceptible to malicious attacks from the Internet as they are deployed at the perimeter of the network [1], [2]. Such types of DDoS attacks are called Denial of Firewalling (DoF) attacks. DoF attacks use techniques to abuse firewalls through expanding their resources with minimal amount of effort. These attacks may depend on malicious traffic volume to flood a firewall with redundant packets and force it to perform purposeless extra work. This extra work usually degrades the firewall performance and holds up legitimate users traffic. Contrary, a low volume of special crafted malicious packets may have more prolong effect on firewalls causing additional harm and forcing them to work hard [3], [4]. DoF attacks may target firewall filtering rules or the heart of the stateful firewalls which is the session table.

In DoF attacks targeting firewalls filtering rules, attackers send malicious packets that can follow the longest matching path until processed by the default rule or the last matching rules with high index. This long path of matching

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

process increases the firewall packet filtering time and degrade its performance considerably. Many research works have been proposed to eliminate or minimize such attack effect, which include mainly: Rule/rule-field reordering techniques [5]–[8], Early packet rejection techniques [9]–[12], and Tree-based decision techniques with dynamic behavior [13]–[16].

Firewall session table keeps track of individual packets and associate them with their respective flows. Whenever a legitimate request reaches the firewall, an entry is added to the session table representing that flow. Consequent packets are checked against the session table instead of the filtering rules. Despite that session table is designed to increase firewalls security, it can be itself vulnerable to attacks [17]–[19]. Attackers may send more requests to add entries in the session table more rapidly than the firewall can remove. In this case, new legitimate connections cannot be established, resulting in an another type of DoF attack situation.

Recently, researchers have discovered a new DoF attack, called the ‘‘BlackNurse’’, which specifically targets network firewalls. The attack launches low volume of special crafted ICMP packets, that can overwhelm targeted firewalls’ processors [20], [21]. 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.

This paper offers more insights into the understanding of the BlackNurse attack principals as a case study of DoF attacks. A brief description of ICMP error messages and stateful firewall session table are introduced in order to illustrate reasons for increasing firewalls’ CPU utilization during the BlackNurse attack. The paper discusses practical attack generation and its effect on commercial grades, namely: Juniper NetScreen SSG 20 and Cisco ASA 5540 firewalls. Experiments showed that these devices are affected significantly with the BlackNurse attack using a very low attack volume traffic of 7k packets per second from a single PC. In addition, available mitigation techniques are investigated. OS screening features is used on Juniper NetScreen SSG 20, as an example to defend against the attack. The paper proposes a mechanism to defend against the BlackNurse attack using an early rejection rule with dynamic time activity duration. The proposed rule has an adaptive triggering process, and an activity duration model to infer the activity duration time. Experiments are conducted to simulate the proposed mechanism against novice and expert BlackNurse attackers.

The rest of the paper is organized as follows. Section II and III give brief background abound ICMP error messages and stateful firewall and session table, respectively, and their relation to the BlackNurse attack. Section IV explains the types of DoF attacks on stateful firewall, and gives examples of possible available mitigations. Section V investigates the principle of the emerging BlackNurse attack, provides lab activities for the attack generation and its effect on different commercial firewalls as well as available mitigation techniques. Section VI proposes a mitigation mechanism to defend against the BlackNurse attack. Finally, section VII concludes the paper.

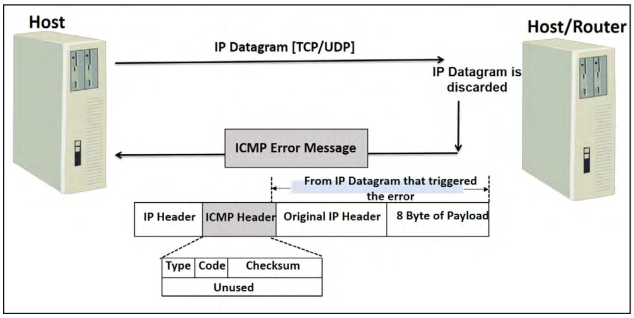


FIGURE 1. ICMP error message format.

II. ICMP ERROR MESSAGES

The BlackNurse attack is based on sending special crafted ICMP messages to degrade firewall performance. This section gives brief background on ICMP protocol especially unreachable error messages.

ICMP allows to deliver different types of error reporting messages from the network and transport layers, such as when a particular end system is not responding, an IP network is not reachable, or a node is overloaded. These ICMP error reporting messages involve one-way communication. However, ICMP is also used to perform a variety of administrative and control functions, such as to check that routers are correctly routing packets to the specified destination address and to send echo and reply messages (the Ping command). Such control messages involve two-way communications.

ICMP messages are encapsulated in an IP datagram. The IP protocol ensures that the ICMP message is sent to the correct destination. This is achieved by assigning the destination address to the IP destination address field. The source address is set to the address of the computer that generated the ICMP message and assigned to the IP source address field. The IP protocol type is set to ‘‘ICMP’’ to indicate that the packet is to be handled by the destination ICMP client interface [22].

ICMP error messages report error conditions and are sent when a datagram is discarded. Figure 1 shows a general ICMP error message format. In ICMP error messages, the first four bytes always have the same format. The first byte identifies the message ‘‘Type’’, the second byte is the ‘‘Code’’ representing the message condition, and the other two bytes are used for the checksum. What comes after can vary and depend upon the error condition being reported. The ICMP error message contains as a payload the original IP packet header and the first eight bytes of the transport protocol that triggers this error message, typically TCP or UDP. This payload is used by stateful firewalls to relate ICMP error messages to their appropriate connections in the session table.

ICMP error messages of Type: 3 represent a "destination unreachable" situation, where code values clarify the type of unreachability, as follows:

0 = net unreachable;
1 = host unreachable;
2 = protocol unreachable;
3 = port unreachable;
4 = fragmentation needed and DF is set;
5 = source route failed

The ICMP error message "destination unreachable, port unreachable" [Type:3, Code:3] is used to launch the BlackNurse attack [20], [21]. What makes this packet so special is that it consumes the targeted firewall's CPU to a point where the firewall becomes completely unresponsive.

III. STATEFUL FIREWALL AND SESSION TABLE

A stateful firewalls uses an internal session table to keep track of packets passing through it [23]–[25]. If a packet has the expected properties that the session table predicts, this means that the packet is part of an active connection, and therefore it is forwarded without any processing by the firewall filtering rules. Stateful firewalls are more secure than stateless firewalls, as the administrator no longer needs to write filtering rules to allow return traffic. This closes the security hole opened by rules that allow return traffic which can be exploited by an attacker to launch a DoS attack. Session table entries depend on the firewall vendor. But, they typically include < protocol (Prot), IP source (Src-IP), source Port (Src-P), IP destination (Dst-IP), destination Port (Dst-P), connection state and timeouts >, where connection state is tracked until a connection is torn down as in a TCP connection or until a preconfigured timeout is reached as in a UDP connection. To better understand session table entries for TCP, UDP and ICMP protocols, let's assume the following security policies:

TCP SP: Allow internal hosts at 192.168.1.1/24 to connect to external Web servers at 192.168.2.1/24 and not vice versa.

UDP SP: Allow internal hosts at 192.168.1.1/24 to request DNS service from 192.168.2.2 and not vice versa.

ICMP SP: Allow internal hosts at 192.168.1.1/24 to ping external hosts at 192.168.2.1/24 and not vice versa.

These security policies are translated to the following firewall rules, shown in Table 1.

TABLE 1. Firewall filtering rules example.

TCP SP									
Rule	**Direction**	**Prot**	**Src-IP**	**Dst-IP**	**SP**	**DP**	**SYN**	**ACK**	**Action**
R1	Out	TCP	192.168.1.1/24	192.168.2.1/24	Any	80	1	0	Allow
R2	In	TCP	192.168.2.1/24	192.168.1.1/24	80	Any	1	1	Allow
R3	Out	TCP	192.168.1.1/24	192.168.2.1/24	Any	80	0	1	Allow
R4	In	TCP	192.168.2.1/24	192.168.1.1/24	80	Any	0	1	Allow
UDP SP									
Rule	**Direction**	**Prot**	**Src-IP**	**Dst-IP**	**SP**	**DP**	**-**	**-**	**Action**
R1	Out	UDP	192.168.1.1/24	192.168.2.2	Any	53			Allow
R2	In	UDP	192.168.2.2	192.168.1.1	53	Any			Allow
ICMP SP									
Rule	**Direction**	**Prot**	**Src-IP**	**Dst-IP**	**Type**	**Code**	**-**	**-**	**Action**
R1	out	ICMP	192.168.1.1/24	192.168.2.1/24	8	0			Allow
R2	In	ICMP	192.168.2.1/24	192.168.1.1/24	0	8			Allow
Default	Any	Any	Any	Any	Any	Any	Any	Any	Deny

A. TCP STATEFUL INSPECTION

Tracking TCP connections is based on the TCP three-way hand shack process. When a SYN request reaches the firewall, the firewall matches it against the set of filtering rules. If there is a filtering rule that allows the packet to across, the firewall inserts a new TCP entry into the session table, and the TCP connection state is set to the SYN_RCVD state. Once the two other remaining packets of the three-way handshake process are received, the TCP connection state transits to the ESTABLISHED state. Table 2 shows an example of a TCP SYN packet accepted by TCP R1 in Table 1, and the corresponding session entries upon receiving the TCP first packet and after completing the three-way hand shake process.

TABLE 2. Process of adding TCP connection entry in the session table.

TCP SYN Packet

Packet	Prot	Src-IP	Dst-IP	SP	DP	SYN	ACK
Packet#1	TCP	192.168.1.3	192.168.2.2	2235	80	1	0

Session Table entry after receiving the SYN packet

TCP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	TCP	192.168.1.3	192.168.2.2	2235	80	SYN_RCVD	Half Open connection, default 10s

Session Table entry after completing the three-way hand shaking

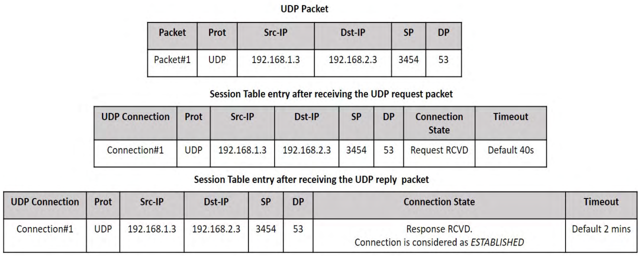
TCP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	TCP	192.168.1.3	192.168.2.2	2235	80	ESTABLISHED	Full connection, default 3600s

Therefore, the first SYN packet of a TCP connection effectively opens a hole in the firewall, and the returned traffic is allowed through this hole within a predefined timeout limit; the default is 10s [26], [27]. Other subsequent TCP packets with the flag SYN unset and the flag ACK set, are checked if they belong to an active connection. If matching entry is found in the session table, packets are allowed through immediately. If no such matching exists, then the packets are rejected. A TCP session entry is removed from the session table when FIN or RST packets are received. However, A TCP session entry is removed also, if a preconfigured idle timeout is reached; the default is usually 1 hour [26], [27].

B. UDP STATEFUL INSPECTION

UDP is a connectionless protocol since it does not have flags or sequence numbers, which make tracking the state of a UDP connection more complicated process compared to TCP. Stateful firewalls use pseudo-stateful mechanism to treat UDP traffic as a stateful traffic and document it in the session table. Upon receiving a UDP packet, the firewall inspects the source and destination addresses and UDP port numbers. If there is a filtering rule that allow the packet

TABLE 3. Process of adding UDP connection entry in the session table.



UDP Packet

Packet	Prot	Src-IP	Dst-IP	SP	DP
Packet#1	UDP	192.168.1.3	192.168.2.3	3454	53

Session Table entry after receiving the UDP request packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	UDP	192.168.1.3	192.168.2.3	3454	53	Request RCVD	Default 40s

Session Table entry after receiving the UDP reply packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#1	UDP	192.168.1.3	192.168.2.3	3454	53	Response RCVD. Connection is considered as *ESTABLISHED*	Default 2 mins

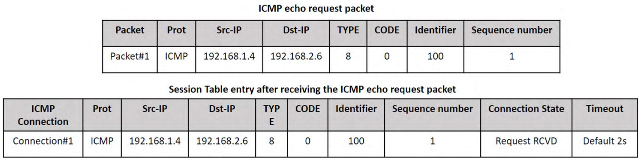
across, the firewall inserts a new UDP entry into the session table. Then, any UDP packet between the source and destination over the specified port numbers can pass back and forth freely. The firewall allows the connection to remain up as long as there are packets flowing through it. Since the firewall cannot track the state of packet exchange in a UDP connection, it cannot determine when the connection is over. The solution for this is to use a preconfigured *idle timeout*to tear down the connection. Idle timeout is a configurable option, the default for UDP traffic is usually 2 mins [26], [27]. The receiving of any UDP packet will update the idle timeout of its connection to the default value. However, once the idle time expires the connection is torn down and its corresponding entry is removed from the session table. Table 3 shows an example of a UDP packet accepted by UDP R1 in Table 1, and the corresponding added session entry. In this example, the DNS query information is added to the session table and a timeout period of 40s is set for the return packet to be received. Once the reply is received, the session is considered as if it is *established* and the idle timeout is set usually to 2 mins [26], [27].

The iptables developers consider that a connection state becomes ''ESTABLISHED'' when packets are seen in both directions whatever the protocol between the two hosts [28].

C. ICMP STATEFUL INSPECTION

ICMP is also a connectionless protocol. However, like UDP, it also has attributes that allow its connections to be tracked. The ICMP attributes are usually the Type, Code, Identifier and Sequence number fields in the ICMP header. Since ICMP has no mechanism to announce the end of its connection, a predetermined timeout is associated with each connection entry.

TABLE 4. Process of adding ICMP connection entry in the session table.



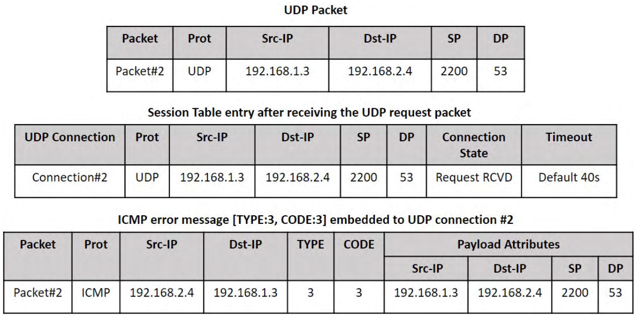
ICMP echo request packet

Packet	Prot	Src-IP	Dst-IP	TYPE	CODE	Identifier	Sequence number
Packet#1	ICMP	192.168.1.4	192.168.2.6	8	0	100	1

Session Table entry after receiving the ICMP echo request packet

ICMP Connection	Prot	Src-IP	Dst-IP	TYP E	CODE	Identifier	Sequence number	Connection State	Timeout
Connection#1	ICMP	192.168.1.4	192.168.2.6	8	0	100	1	Request RCVD	Default 2s

TABLE 5. Process of adding ICMP connection entry in the session table.



UDP Packet

Packet	Prot	Src-IP	Dst-IP	SP	DP
Packet#2	UDP	192.168.1.3	192.168.2.4	2200	53

Session Table entry after receiving the UDP request packet

UDP Connection	Prot	Src-IP	Dst-IP	SP	DP	Connection State	Timeout
Connection#2	UDP	192.168.1.3	192.168.2.4	2200	53	Request RCVD	Default 40s

ICMP error message [TYPE:3, CODE:3] embedded to UDP connection #2

Packet	Prot	Src-IP	Dst-IP	TYPE	CODE	Payload Attributes			
						Src-IP	Dst-IP	SP	DP
Packet#2	ICMP	192.168.2.4	192.168.1.3	3	3	192.168.1.3	192.168.2.4	2200	53

ICMP control traffic that involves two-way communication such as, Ping command is based on an ICMP echo request message sent first followed by a response; these can be treated as a connection in its own. When receiving an echo request message, the firewall extracts source and destination addresses, Type and Code, and then matches it against the set of filtering rules. If a match is found, the firewall inserts a new ICMP session entry in the session table. In ICMP control traffic, the Identifier, Sequence number and Data fields should be returned to the sender unaltered. Table 4 shows an ICMP echo request packet accepted by ICMP R1 in Table 1, and the corresponding added session entry. In this example, the echo request packet parameters are added to the session table and a timeout is set for the echo reply to reach the firewall. Once the reply is received, the firewall realizes that there is a corresponding entry in its session table with the same attributes. Therefore, it allows the reply to pass through and consequently this entry is removed from the session table.

However, tracking the state of ICMP error messages is a much more complicated process than UDP. Since these error messages involve one-way communication and they are precipitated by requests from other protocols like TCP or UDP. Because of this multiprotocol issue, extra work is required from the firewall to match each ICMP error message to its corresponding session, which contains attributes about the request packet that triggers this error message to be sent. Once the firewall receives an ICMP error message, it extracts from its payload the attributes of the original packet that caused this error message to be sent. Then, the firewall searches in its session table for a session entry with similar attributes. If a match is found, the error message is embedded to its corresponding session entry and is allowed to pass through the firewall in order to notify the sender that the sent request is not accomplished. Table 5 shows ICMP port unreachable error message that is generated after receiving a DNS request packet and the corresponding session entry. In this example, a UDP packet requesting DNS service at port 53 is accepted by UDP R1 in Table 1. The firewall adds a UDP session entry and waits for the destination reply. However, no DNS service is available in the destination server. Thus, the destination generates ICMP port unreachable message [Type: 3, Code: 3]. The error message reaches the firewall in which the payload is extracted and the firewall recognize that the payload has similar attributes to UDP connection #2. Therefore, the firewall allows the port unreachable error message to pass through to the sender.

IV. STATEFUL FIREWALL ATTACKS

DoF attacks define methods of abusing firewalls through expending firewall resources with minimal amount of effort. In DoF, attackers use special crafted packets to effectively overload the firewall device itself instead of the network behind it. Attackers may target the firewall filtering rules or the session table. The effect can appear as increase in firewall CPU utilization, memory usage or number of allocated sessions. This may make the firewall unresponsive and lead to denial of service for all devices located behind the firewall. Finite amount of firewall memory and CPU power impose finite upper bound on the accepted traffic flows and the number of established sessions. Usually, session table improves firewall performance as it safely considers that packets belonging to a particular flow do not need to be re-checked against the filtering rules. However, session table maybe used against the firewall as attackers may craft special type of packets that flood this limited resource. As a result, the session table is filled with illegitimate flows that prevent legitimate flows from being established.

A. ATTACKS TARGETING FIREWALL FILTERING RULES

Attacks targeting firewall filtering rules depend mainly on the used packet filtering mechanism and the size of the firewall policy. In firewalls that use sequential search algorithm for packet filtering, an attacker may launch a DoS attack that primarily target a filtering rule with high index in the firewall policy. This makes the attack more effective as the illegitimate packets will traverse a long filtering path until reaching the intended bottom rule, degrading rapidly the firewall performance [29]. It was shown in [7], [8] that dynamic reordering of filtering rules and rule-fields can effectively defend against such attacks. In [15], [16], a splay tree firewall was proposed that can early reject or accept packets and therefore enhancing packets filtering process. Also, the splay filters are reordered based on a statistical model that utilizes traffic characteristic. Hence, the mechanism is considered as a device protection against DoS attacks targeting different rules positions.

B. ATTACKS TARGETING FIREWALL SESSION TABLE

As the session table is a limited resource, it can be a target for attackers to degrade the firewall performance. Attackers may flood the session table with illegitimate requests to increase session entries until it is full, so that the firewall can no longer handle the legitimate connection requests. Contrary, attackers may use special crafted packets at low volume that can derive high CPU load until the firewall becomes unresponsive to legitimate requests.

1) SESSION TABLE VOLUMETRIC ATTACKS (FLOODING)

When packets that belong to new flows pass the firewall, new session entries are added to the session table. However, if the number of flows exceeds the size of the session table, the firewall cannot create any new sessions and starts dropping new connection requests. This is the case when attacker floods the session table with massive barged traffic such as: SYN flood, SYN-ACK-ACK flood, UDP flood, ICMP flood and SYN-ACK-ACK proxy flood.

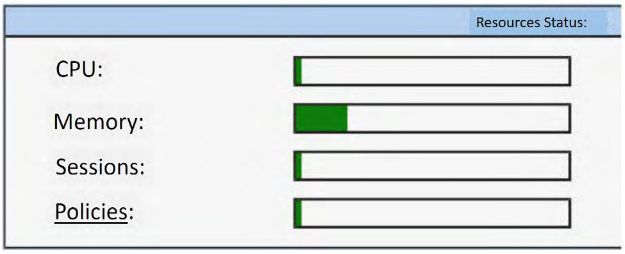


FIGURE 2. Juniper NetScreen SSG 20 under normal traffic situation as shown in the device's GUI interface.

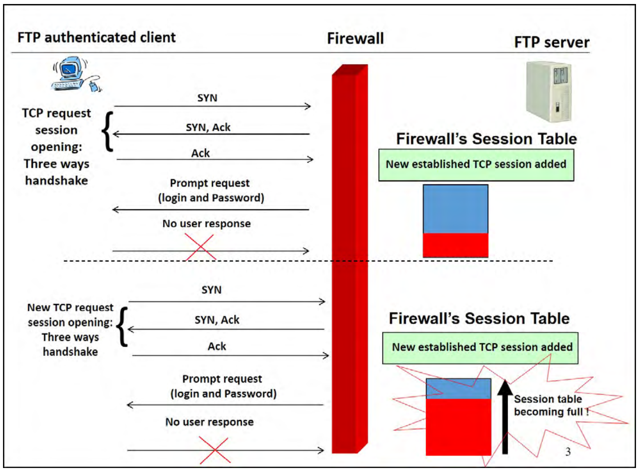

FTP authenticated client
Firewall
FTP server
TCP request session opening: Three ways handshake
SYN
SYN, Ack
Ack
Prompt request (login and Password)
No user response
Firewall's Session Table
New established TCP session added
New TCP request session opening: Three ways handshake
SYN
SYN, Ack
Ack
Prompt request (login and Password)
No user response
Firewall's Session Table
New established TCP session added
Session table becoming full !
3

FIGURE 3. SYN-ACK-ACK proxy flood process.

The impact of session table flooding will be demonstrated through explaining SYN-ACK-ACK proxy flood and ICMP flood on Juniper NetScreen SSG 20. Figure 2 shows a screen shot of Juniper NetScreen SSG 20 under normal traffic situation, where CPU, memory and sessions resources status are normal.

a: SYN-ACK-ACK proxy flood

To initiate a normal FTP connection, an authenticated user sends a SYN request to FTP server on port 21. This request is received by the firewall, and checked against the filtering rules, as it is a SYN request an entry is created accordingly in the session table. After that, the firewall proxies a SYN-ACK packet to the user, then the user responds with ACK packet. At this stage, the initial three-way hand shack process is completed between the user and the firewall. Accordingly, the firewall sends to the user the login prompt. However, malicious user will not log in, but instead keeps on initiating SYN-ACK-ACK requests. This will fill up the firewall's session table and prevent it from accepting newly legitimate connection requests. Figure 3 illustrates the aforementioned SYN-ACK-ACK flood process.

To initiate this attack on Juniper NetScreen SSG 20, we used frameip packet generator tool [30] with the following command:

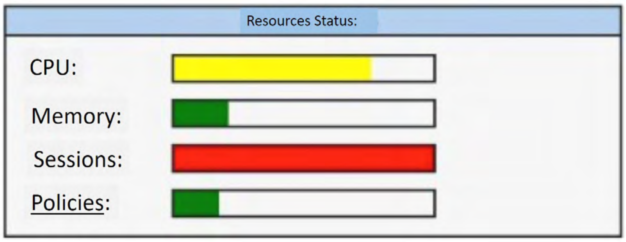
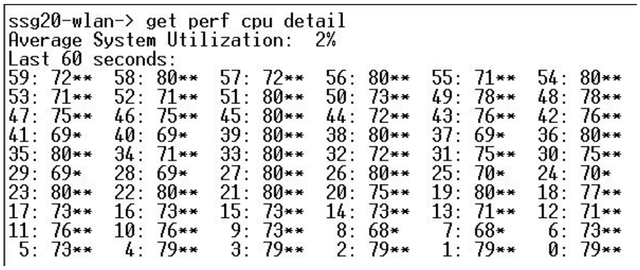
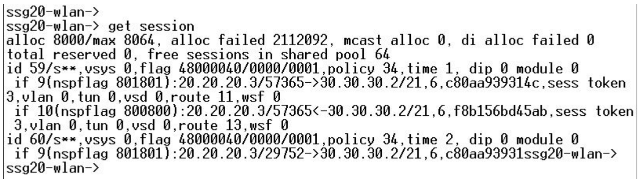


FIGURE 4. **Juniper NetScreen SSG 20 resources status during the SYN-ACK-ACK proxy flood attack.**



```
ssg20-wlan-> get perf cpu detail
Average System Utilization:  2%
Last 60 seconds:
59: 72**  58: 80**  57: 72**  56: 80**  55: 71**  54: 80**
53: 71**  52: 71**  51: 80**  50: 73**  49: 78**  48: 78**
47: 75**  46: 75**  45: 80**  44: 72**  43: 76**  42: 76**
41: 69*   40: 69*   39: 80**  38: 80**  37: 69*   36: 80**
35: 80**  34: 71**  33: 80**  32: 72**  31: 75**  30: 75**
29: 69*   28: 69*   27: 80**  26: 80**  25: 70*   24: 70*
23: 80**  22: 80**  21: 80**  20: 75**  19: 80**  18: 77**
17: 73**  16: 73**  15: 73**  14: 73**  13: 71**  12: 71**
11: 76**  10: 76**   9: 73**   8: 68*    7: 68*    6: 73**
 5: 73**   4: 79**   3: 79**   2: 79**   1: 79**   0: 79**
```

FIGURE 5. **Juniper NetScreen SSG 20 CPU utilization during the SYN-ACK-ACK proxy flood attack.**



```
ssg20-wlan->
ssg20-wlan-> get session
alloc 8000/max 8064, alloc failed 2112092, mcast alloc 0, di alloc failed 0
total reserved 0, free sessions in shared pool 64
id 59/s**,vsys 0,flag 48000040/0000/0001,policy 34,time 1, dip 0 module 0
 if 9(nspflag 801801):20.20.20.3/57365->30.30.30.2/21,6,c80aa939314c,sess token
3,vlan 0,tun 0,vsd 0,route 11,wsf 0
 if 10(nspflag 800800):20.20.20.3/57365<-30.30.30.2/21,6,f8b156bd45ab,sess token
 3,vlan 0,tun 0,vsd 0,route 13,wsf 0
id 60/s**,vsys 0,flag 48000040/0000/0001,policy 34,time 2, dip 0 module 0
 if 9(nspflag 801801):20.20.20.3/29752->30.30.30.2/21,6,c80aa93931ssg20-wlan->
ssg20-wlan->
```

FIGURE 6. **Juniper NetScreen SSG 20 session allocation during the SYN-ACK-ACK proxy flood attack.**

C:\*Frameip>frameip -interface 0 -send_mode 1 -ip_ destination [] -ip_ type 6 -tcp_ port_ destination 21 -ip_source [] -wait 0 -loops 0*

This command floods (*-wait 0 -loops 0*) the firewall with TCP SYN packets (*-ip_type 6, SYN:1 default*) from the authenticated user (*ip_source []*) targeting FTP server (*ip_destination []*) at port 21(*tcp_port_destination 21*).

The impact of this attack on Juniper firewall resources status is shown in Figure 4. The online command "*get perf CPU detail"* shows a noticeable instant increase in the CPU utilization that reached 80%, as illustrated in Figure 5. Also, the online command *"get session"* shows a dramatic increase in the firewall session table entries, where 8000 sessions are allocated out of 8064, as illustrated in Figure 6.

b: ICMP Flood

In this attack the attacker floods a target with so many ICMP echo request packets [Type: 8, Code: 0] aiming to degrade the firewall performance and preventing it from processing valid traffic.

To initiate this attack on Juniper NetScreen SSG 20, we used frameip packet generator tool [30] with the following command:

C:\*Frameip>frameip -interface 0 –send mode 1 -ip_destination [] -ip_source r -wait 0 -loops 0*

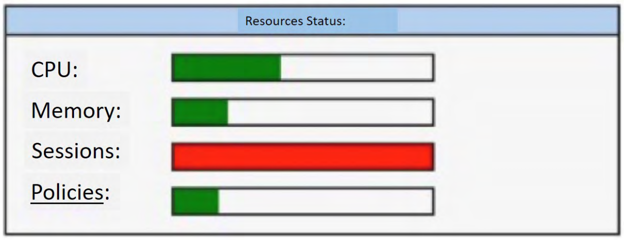
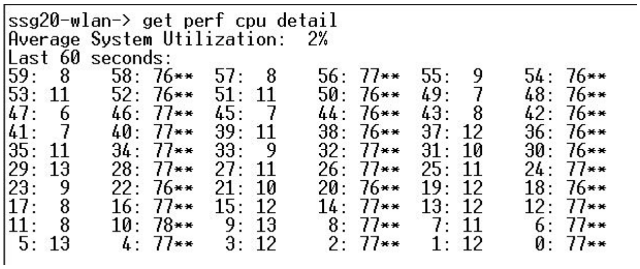
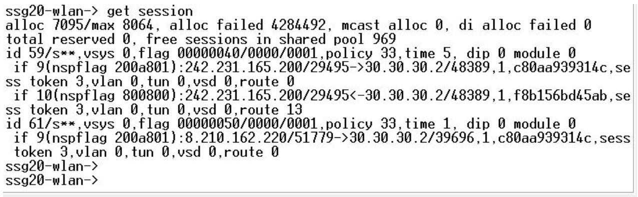


FIGURE 7. **Juniper NetScreen SSG 20 resources status during the ICMP flood attack.**



```
ssg20-wlan-> get perf cpu detail
Average System Utilization:  2%
Last 60 seconds:
59:  8    58: 76**  57:  8    56: 77**  55:  9    54: 76**
53: 11    52: 76**  51: 11    50: 76**  49:  7    48: 76**
47:  6    46: 77**  45:  7    44: 76**  43:  8    42: 76**
41:  7    40: 77**  39: 11    38: 76**  37: 12    36: 76**
35: 11    34: 77**  33:  9    32: 77**  31: 10    30: 76**
29: 13    28: 77**  27: 11    26: 77**  25: 11    24: 77**
23:  9    22: 76**  21: 10    20: 76**  19: 12    18: 76**
17:  8    16: 77**  15: 12    14: 77**  13: 12    12: 77**
11:  8    10: 78**   9: 13     8: 77**   7: 11     6: 77**
 5: 13     4: 77**   3: 12     2: 77**   1: 12     0: 77**
```

FIGURE 8. **Juniper NetScreen SSG 20 CPU utilization during the ICMP flood attack.**



```
ssg20-wlan-> get session
alloc 7095/max 8064, alloc failed 4284492, mcast alloc 0, di alloc failed 0
total reserved 0, free sessions in shared pool 969
id 59/s**,vsys 0,flag 00000040/0000/0001,policy 33,time 5, dip 0 module 0
 if 9(nspflag 200a801):242.231.165.200/29495->30.30.30.2/48389,1,c80aa939314c,se
ss token 3,vlan 0,tun 0,vsd 0,route 0
 if 10(nspflag 800800):242.231.165.200/29495<-30.30.30.2/48389,1,f8b156bd45ab,se
ss token 3,vlan 0,tun 0,vsd 0,route 13
id 61/s**,vsys 0,flag 00000050/0000/0001,policy 33,time 1, dip 0 module 0
 if 9(nspflag 200a801):8.210.162.220/51779->30.30.30.2/39696,1,c80aa939314c,sess
 token 3,vlan 0,tun 0,vsd 0,route 0
ssg20-wlan->
ssg20-wlan->
```

FIGURE 9. **Juniper NetScreen SSG 20 session allocation during the ICMP flood attack.**

This command floods the target with ICMP echo request packets using random IP source address.

Figure 7 shows the impact of this attack on the firewall resources status. The firewall CPU utilization under the attack reached 78% using the online command *"get perf CPU detail"*, as shown in Figure 8. Likewise, Figure 9 shows the significant increase in the session table entries using the online command *"get session"*.

c: Mitigations techniques for session table volumetric attacks

Different mitigations techniques are proposed by vendors to mitigate common flood attacks such as limit the amount of sessions per source and destination IP addresses. Another mitigation technique named "Aggressive Aging" is used when the session table becomes full and the firewall is unable to accept new connection requests. According to this feature inactive sessions can be removed from the session table to provide slots for newly connections.

2) SESSION TABLE LOW VOLUMETRIC ATTACKS

Traditional volumetric attacks usually require large network traffic volume to be able to crash or degrade the performance of target servers. However, recently attackers use special crafted packets that target particularly firewall session table

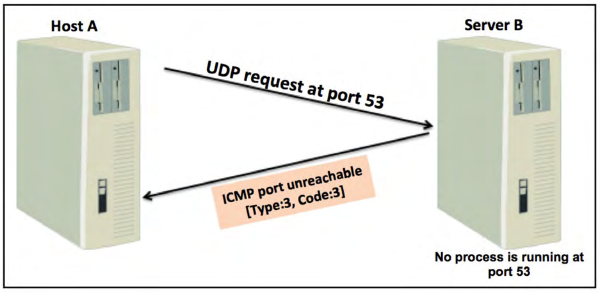

Host A
Server B
UDP request at port 53
ICMP port unreachable [Type:3, Code:3]
No process is running at port 53

FIGURE 10. ICMP port unreachable error message generation.

using very low volume of malicious traffic. An example of such emerging attack is called the BlackNurse attack.

V. BLACKNURSE ATTACK

Traditional ICMP flooding attack floods target hosts with massive echo request packets [Type 8, Code 0]. In contrast, TDC Security Operations Center has discovered in 2016 another type of ICMP based attack known as BlackNurse attack that explicitly targets firewalls and routers [20]. The BlackNurse attack is considered as a special type of ICMP attack that depends on sending low volume traffic of specific ICMP error messages [Type: 3 (Destination unreachable), Code: 3 (Port unreachable)]. This low volume DDoS attack is effective because the objective is not to flood the firewall with illegitimate traffic, but rather to craft packets that consume firewall resources and drive high CPU workload. Usually, firewalls with single CPU are more likely to be vulnerable to this attack than firewalls with multicore.

A. BLACKNURSE ATTACK PRINCIPLE

ICMP port unreachable error message [Type 3, Code 3] is generated when a destination host cannot deliver a reply packet because the intended port is not active, for example, when a source computer A sends UDP request to port 53 of a target computer B that is not a DNS server. DNS reply will never be sent back; instead the target will generate ICMP port unreachable message to the source, as shown in Figure 10.

To be more precise, upon receiving the UDP request packet shown in Figure 10, 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. Since there is no DNS service running on target B, it 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 in its session table for a session entry with similar attributes. If a match is found, the error message is embedded to its corresponding session entry and is allowed to pass through the firewall in order to notify the sender that the request sent is not accomplished. This process of stateful analysis of ICMP error messages can consume most firewall resources and prevent it from processing normal traffic.

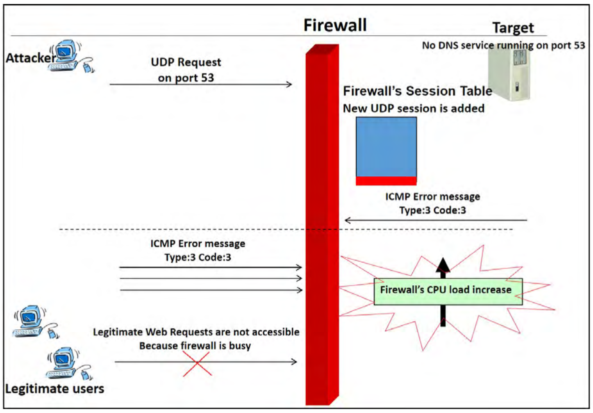

Firewall
Target
No DNS service running on port 53
Attacker
UDP Request on port 53
Firewall's Session Table
New UDP session is added
ICMP Error message Type:3 Code:3
ICMP Error message Type:3 Code:3
Firewall's CPU load increase
Legitimate Web Requests are not accessible Because firewall is busy
Legitimate users

FIGURE 11. BlackNurse attack principle and generation.

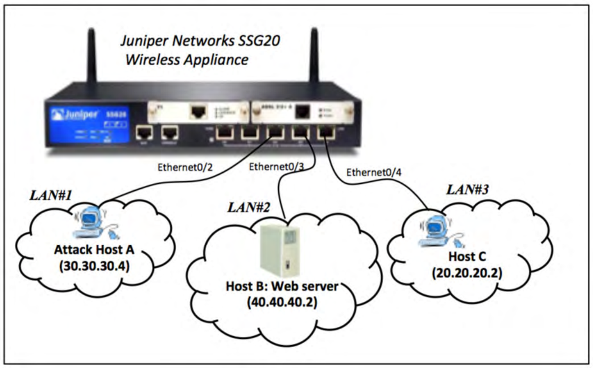

Juniper Networks SSG20 Wireless Appliance
Ethernet0/2
Ethernet0/3
Ethernet0/4
LAN#1
LAN#2
LAN#3
Attack Host A (30.30.30.4)
Host B: Web server (40.40.40.2)
Host C (20.20.20.2)

FIGURE 12. Network architecture for the BlackNurse experiments.

The attacker now can get benefit from such ICMP port unreachable message to launch the BlackNurse attack, as shown in Figure 11. Sending a low volume traffic of ICMP port unreachable message, even from a single host, at a rate of 15-18 Mbps (40-50 K packets per second) [20], can drive high firewall CPU loads and make the firewall unresponsive, as shown in Figure 11. These packets are considered among the most expensive computationally, because they consume much of the processing power of the firewall. During the attack, the firewall's CPU utilization may reach up to 90%. In addition, when the attack is undergoing, users from the LAN side will no longer be able to exchange network traffic.

B. BLACKNURSE ATTACK GENERATION

Practically, to generate the BlackNurse attack as illustrated in [20], any network packet generator tool can be used. For example, the following Hping3 tool's online commands allow to generate the attack [31]:

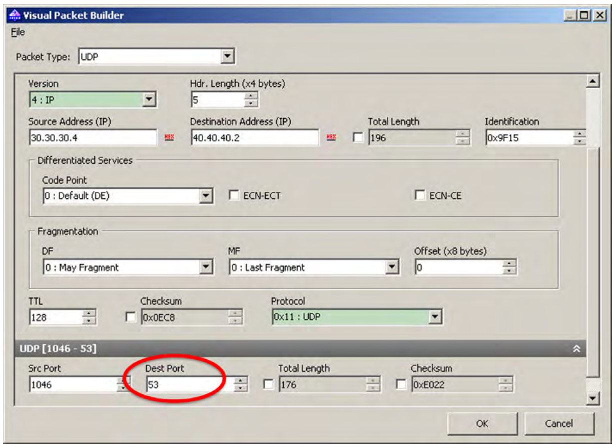
# hping3 -1 –C 3 –K 3 –i u20 dest-ip.

This command sends ICMP port unreachable message [Type: 3, Code: 3] to the target dest-ip, where –i u20 sends one packet every 20 ms.

# hping3 -1 –C 3 –K 3 –flood dest-ip

While this command floods the target dest-ip with ICMP port unreachable message [Type: 3, Code: 3].

Experiments are conducted to generate the BlackNurse attack using only 7K packets per second and show its impact



Packet Type: UDP

Version: 4 : IP
Hdr. Length (x4 bytes): 5
Source Address (IP): 30.30.30.4
Destination Address (IP): 40.40.40.2
Total Length: □ 196
Identification: 0x9F15

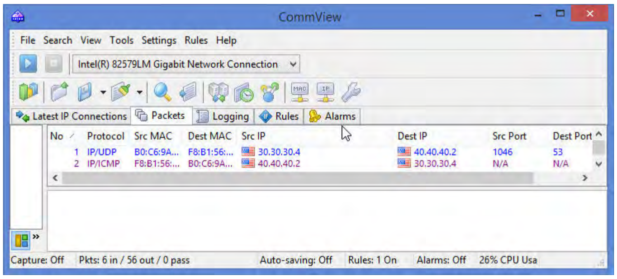
Differentiated Services
Code Point: 0 : Default (DE)
□ ECN-ECT
□ ECN-CE

Fragmentation
DF: 0 : May Fragment
MF: 0 : Last Fragment
Offset (x8 bytes): 0

TTL: 128
Checksum: □ 0x0EC8
Protocol: 0x11 : UDP

UDP [1046 - 53]
Src Port: 1046
Dest Port: 53
Total Length: □ 176
Checksum: □ 0xE022

FIGURE 13. UDP packet from host A to host B requesting DNS service.


CommView
File Search View Tools Settings Rules Help
Intel(R) 82579LM Gigabit Network Connection
Latest IP Connections
Packets
Logging
Rules
Alarms

No	Protocol	Src MAC	Dest MAC	Src IP	Dest IP	Src Port	Dest Port
1	IP/UDP	B0:C6:9A...	F8:B1:56:...	30.30.30.4	40.40.40.2	1046	53
2	IP/ICMP	F8:B1:56:...	B0:C6:9A...	40.40.40.2	30.30.30.4	N/A	N/A

Capture: Off | Pkts: 6 in / 56 out / 0 pass | Auto-saving: Off | Rules: 1 On | Alarms: Off | 26% CPU Usa

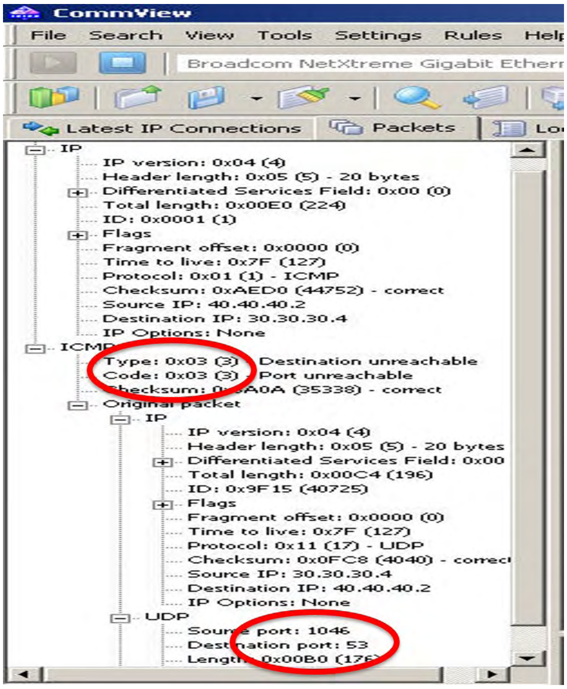
FIGURE 14. ICMP port unreachable message Type 3, Code 3 generated on host B upon UDP request received.

on commercial grades, Juniper NetScreen SSG 20, and Cisco ASA 5540 firewalls. Figure 12 shows the network architecture used in the hands-on lab activities. Three hosts belonging to three different networks are connected to Juniper NetScreen SSG 20 firewall device. Host A is the host that will generate the BlackNurse attack. Hosts B and C are a Web server and a Web client, respectively. LiteServer [32] is the Web server software. Host B does not have a DNS service running on it. All hosts use CommView sniffer tool [33] to capture and build the desired network traffic. Juniper NetScreen SSG 20 will be replaced later by Cisco ASA 5540 using the same network architecture shown in Figure 12.

The firewall should have filtering rules to allow UDP and TCP to pass through. Using the firewall GUI interface, a TCP rule is implemented to allow standard web traffic (TCP/80) between hosts C and B. In addition, a UDP rule is implemented to allow DNS request (UDP/53) between hosts A and B.

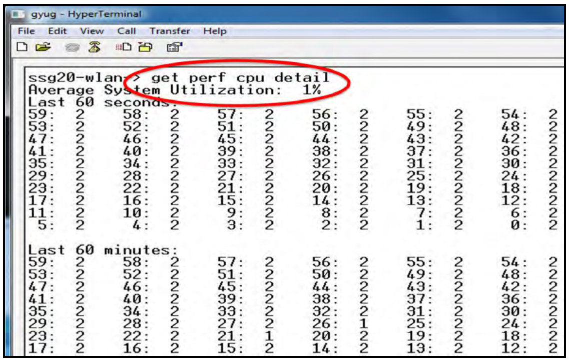
The experiment consists of the following steps:

1) Commview's packet generator is used on Host A to build a UDP request to host B on port 53 (Figure 13).
2) The UDP request is sent to host B. Since, host B has no DNS service running on it, it will automatically generate ICMP port unreachable message to host A (Figure 14).
3) Commview's packet generator is used to sniff the ICMP port unreachable message on host A. Figure 15 shows a screenshot of the capture ICMP error message.


CommView
File Search View Tools Settings Rules Help
Broadcom NetXtreme Gigabit Etherr
Latest IP Connections
Packets
Lo

```
IP
    IP version: 0x04 (4)
    Header length: 0x05 (5) - 20 bytes
    Differentiated Services Field: 0x00 (0)
    Total length: 0x00E0 (224)
    ID: 0x0001 (1)
    Flags
    Fragment offset: 0x0000 (0)
    Time to live: 0x7F (127)
    Protocol: 0x01 (1) - ICMP
    Checksum: 0xAED0 (44752) - correct
    Source IP: 40.40.40.2
    Destination IP: 30.30.30.4
    IP Options: None
ICMP
    Type: 0x03 (3) - Destination unreachable
    Code: 0x03 (3) - Port unreachable
    Checksum: 0x8A0A (35338) - correct
    Original packet
        IP
            IP version: 0x04 (4)
            Header length: 0x05 (5) - 20 bytes
            Differentiated Services Field: 0x00
            Total length: 0x00C4 (196)
            ID: 0x9F15 (40725)
            Flags
            Fragment offset: 0x0000 (0)
            Time to live: 0x7F (127)
            Protocol: 0x11 (17) - UDP
            Checksum: 0x0FC8 (4040) - correct
            Source IP: 30.30.30.4
            Destination IP: 40.40.40.2
            IP Options: None
        UDP
            Source port: 1046
            Destination port: 53
            Length: 0x00B0 (176)
```

FIGURE 15. ICMP port unrachable message Type 3, Code 3 at host A.

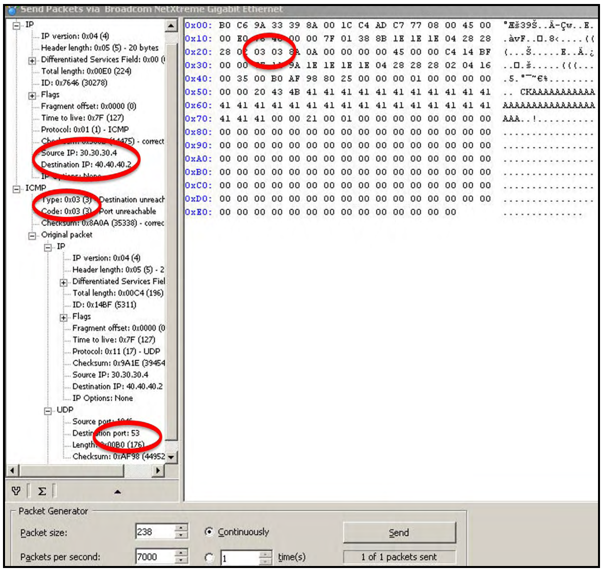


```
ssg20-wlan-> get perf cpu detail
Average System Utilization:  1%
Last 60 seconds:
59:  2     58:  2     57:  2     56:  2     55:  2     54:  2
53:  2     52:  2     51:  2     50:  2     49:  2     48:  2
47:  2     46:  2     45:  2     44:  2     43:  2     42:  2
41:  2     40:  2     39:  2     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
11:  2     10:  2      9:  2      8:  2      7:  2      6:  2
 5:  2      4:  2      3:  2      2:  2      1:  2      0:  2

Last 60 minutes:
59:  2     58:  2     57:  2     56:  2     55:  2     54:  2
53:  2     52:  2     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:  1     25:  2     24:  2
23:  2     22:  2     21:  1     20:  2     19:  2     18:  2
17:  2     16:  2     15:  2     14:  2     13:  2     12:  2
```

FIGURE 16. Firewall CPU status before the attack.

4) At this point, the attack is ready to be launched. Figure 2 shows Juniper device's status and CPU utilization under normal condition before the attack.
   The Juniper device's status and CPU utilization can be also verified by using the online command "get perf CPU detail", as shown in Figure 16. In fact, Figure 16 shows that the firewall CPU utilization is around 2% sixty seconds before launching the attack.
5) The sniffed ICMP port unreachable packet on host A is used to generate the BlackNurse attack, as shown in Figure 17.
6) As long as the firewall is under the BlackNurse attack, its processor will continue to be overwhelmed. The firewall GUI interface becomes unresponsive, as shown in Figure 18.
   Figure 19 shows the effect of the attack on Juniper NetScreen SSG 20 CPU performance, as indicated in yellow color, compared to the CPU status shown in Figure 2.



Send Packets via Broadcom NetXtreme Gigabit Ethernet

```
IP
  IP version: 0x04 (4)
  Header length: 0x05 (5) - 20 bytes
  Differentiated Services Field: 0x00 (
  Total length: 0x00E0 (224)
  ID: 0x7646 (30278)
  Flags
  Fragment offset: 0x0000 (0)
  Time to live: 0x7F (127)
  Protocol: 0x01 (1) - ICMP
  Checksum: 0x388B (14475) - correct
  Source IP: 30.30.30.4
  Destination IP: 40.40.40.2
  IP Options: None
ICMP
  Type: 0x03 (3) - Destination unreach
  Code: 0x03 (3) - Port unreachable
  Checksum: 0x8A0A (35338) - correc
  Original packet
    IP
      IP version: 0x04 (4)
      Header length: 0x05 (5) - 2
      Differentiated Services Fiel
      Total length: 0x00C4 (196)
      ID: 0x14BF (5311)
      Flags
      Fragment offset: 0x0000 (0
      Time to live: 0x7F (127)
      Protocol: 0x11 (17) - UDP
      Checksum: 0x9A1E (39454
      Source IP: 30.30.30.4
      Destination IP: 40.40.40.2
      IP Options: None
    UDP
      Source port: 1046
      Destination port: 53
      Length: 0x00B0 (176)
      Checksum: 0xAF98 (44952
```

```
0x00: B0 C6 9A 33 39 8A 00 1C C4 AD C7 77 08 00 45 00   °Æš39Š..Ä-Çw..E.
0x10: 00 E0 76 46 00 00 7F 01 38 8B 1E 1E 1E 04 28 28   .àvF..□.8‹....((
0x20: 28 02 03 03 8A 0A 00 00 00 00 45 00 00 C4 14 BF   (...Š.....E..Ä.¿
0x30: 00 00 7F 11 9A 1E 1E 1E 1E 04 28 28 28 02 04 16   ..□.š.....(((...
0x40: 00 35 00 B0 AF 98 80 25 00 00 00 01 00 00 00 00   .5.°¯˜€%........
0x50: 00 00 20 43 4B 41 41 41 41 41 41 41 41 41 41 41   .. CKAAAAAAAAAAA
0x60: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0x70: 41 41 41 00 00 21 00 01 00 00 00 00 00 00 00 00   AAA..!..........
0x80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0xA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0xB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0xC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0xD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0xE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ..............
```

Packet Generator

Packet size: 238 ● Continuously

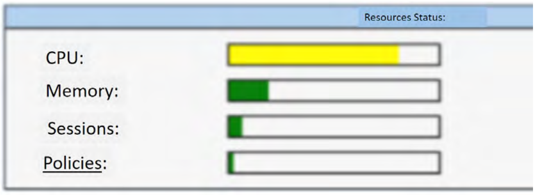
Packets per second: 7000 ○ 1 time(s)

FIGURE 17. BlackNurse attack generation at host A.

http://20.20.20.1/nswebui.html

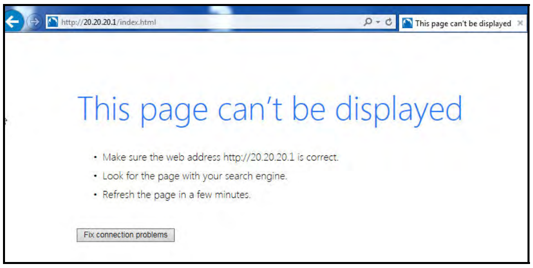
ssg20-wlan:Juniper-Screen...

FIGURE 18. Firewall GUI interface during the attack.



Resources Status:

FIGURE 19. The effect of the attack on Juniper SSG 20 CPU performance.



http://20.20.20.1/index.html

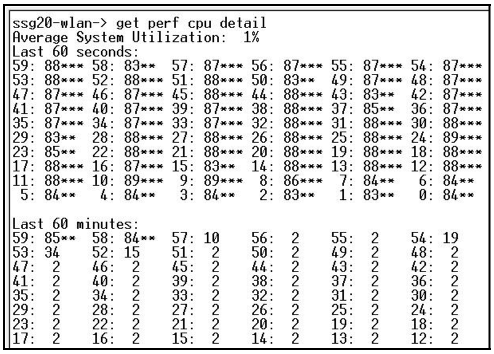
This page can't be displayed

This page can't be displayed

- Make sure the web address http://20.20.20.1 is correct.
- Look for the page with your search engine.
- Refresh the page in a few minutes.

FIGURE 20. The effect of the attack on normal users accessing Web server.

7) Another effect of this attack is that normal users will experiment difficulties to access the Web server running on host B, since the firewall CPU is overloaded with the attack traffic, as shown in Figure 20.
8) Once the BlackNurse attack is stopped, online command "*get perf CPU detail*" is used to display the history of Juniper SSG 20 CPU utilization during the attack. Figure 21 shows that the firewall CPU utilization reached about 89% during the sixty seconds after launching the attack.



```
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 21. Juniper NetScreen SSG 20 CPU status after the attack.

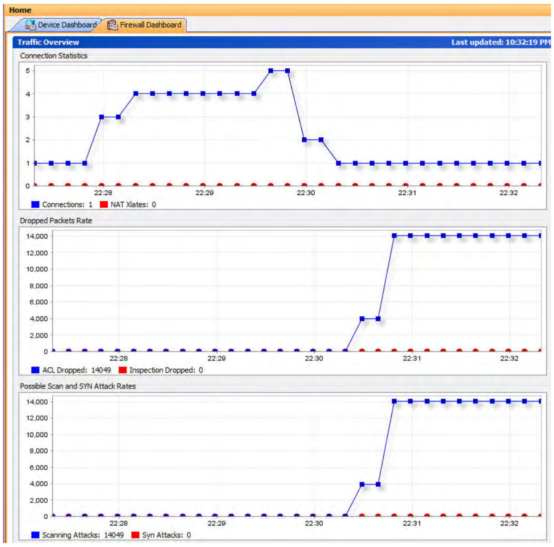


FIGURE 22. Cisco ASA 5540 real time Dashboard before and after the attack.

9) Juniper NetScreen SSG 20 is replaced with Cisco ASA 5540 firewall using the same network architecture in Figure 12. All steps 1-5 are repeated for Cisco ASA 5540 firewall. Figure 22 shows different Cisco ASA dashboard parameters before and after the attack, while Figure 23 shows the attack impact on its CPU usage. As shown clearly in Figure 23, Cisco ASA 5540 CPU usage started to increase in less than a second after launching the attack until it reaches 20%.

C. GENERAL BLACKNURSE ATTACK MITIGATION

In [20], it has been found that the BlackNurse attack has severe negative impact on the performance of many popular firewalls and routers, such as:

- Cisco ASA 5505, 5506, 5515, 5525 and 5540
- Cisco ASA 5550 (Legacy) and 5515-X
- Cisco 897 router
- Cisco 6500 router
- Fortigate 60c and 100D.
- Fortinet v5.4.1
- Palo Alto SonicWall
- Zyxel NWA3560-N
- Zyxel Zywall USG50

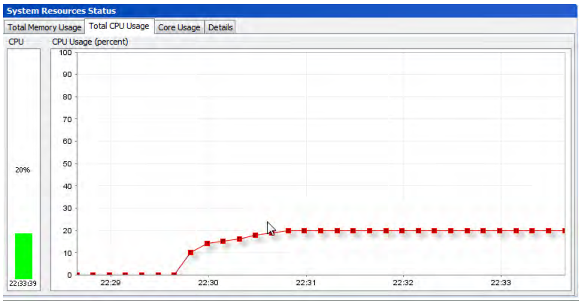


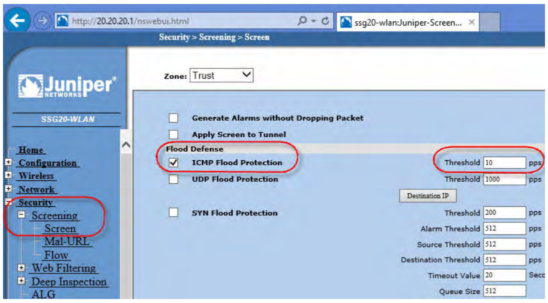
FIGURE 23. Cisco ASA 5540 real time CPU usage before and after the attack.

However, any iptable-based product is unaffected by the BlackNurse attack [20]. In general, the decision regarding whether or not a given firewall is vulnerable to the BlackNurse attack, depends entirely on the device architecture and its protocol stack implementation. Different kinds of mitigation techniques can be implemented to minimize the impact of the BlackNurse attack, namely:

- Drop ICMP packets arriving at the WAN interface of the firewall. This may cause more problems, as any inside host, using ping command to see if a server is alive, will never get a reply.
- The recommendation in [20] is to deny ICMP type 3 messages sent to the firewall WAN interface. However, denying ICMP unreachable messages disables ICMP Path MTU discovery, which can halt IPSec and PPTP traffic. In order to allow Path MTU discovery to function properly, ICMP type 3 code 4 packets (fragmentation needed) should be allowed.
- Use a filtering rule to block ICMP type 3, code 3 packets. This can affect a DNS resolver when attempting to connect to a non-existing DNS server. The DNS resolver delays trying a secondary server as it never receives the ICMP port unreachable message.
- Rate-limit incoming ICMP traffic. This may prevent other legitimate ICMP traffic from reaching its destination.
- Create a list of trusted source hosts for which ICMP packets are allowed.
- Since the BlackNurse attack affects mostly firewalls with one CPU, another alternative mitigation solution is to upgrade a firewall with multiple CPU cores. However, enforcing vendors to produce only multi-core CPU Firewalls is not realistic.

The following steps describe an experiment example for BlackNurse attack mitigation on Juniper NetScreen SSG 20 using the same network architecture shown in Figure 12.

Step #1: Using Juniper firewall's GUI interface, ICMP flood protection screening is enabled, as shown in Figure 24. The threshold should be specified based on the firewall environment and the usage of the ICMP protocol. As an example, the threshold in this experiment has been set to 10 packets per second (pps). This means that the firewall will accept only 10 ICMP packets per second.



Zone: Trust

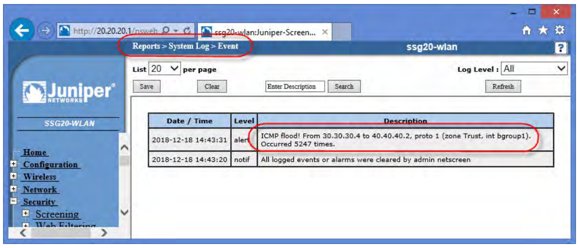
□ Generate Alarms without Dropping Packet

□ Apply Screen to Tunnel

Flood Defense

☑ ICMP Flood Protection	Threshold 10	pps
□ UDP Flood Protection	Threshold 1000	pps
□ SYN Flood Protection	Threshold 200	pps
	Alarm Threshold 512	pps
	Source Threshold 512	pps
	Destination Threshold 512	pps
	Timeout Value 20	Secc
	Queue Size 512	

FIGURE 24. Enabling ICMP flood protection in Juniper NetScreen SSG 20 firewall.



List 20 per page   Log Level : All

Date / Time	Level	Description
2018-12-18 14:43:31	alert	ICMP flood! From 30.30.30.4 to 40.40.40.2, proto 1 (zone Trust, int bgroup1). Occurred 5247 times.
2018-12-18 14:43:20	notif	All logged events or alarms were cleared by admin netscreen

FIGURE 25. Juniper NetScreen SSG 20 firewall screeing log event.

Step #2: Using CommView packet generator in host A, BlackNurse attack traffic is generated, as shown in Figure 17.

Step #3: After the attack generation, Figure 25 shows that Juniper firewall screening limited the rate of accepted ICMP packets and consequently protected the firewall from the attack.

VI. PROPOSED MITIGATION MECHANISM TO DEFEND AGAINST THE BLACKNURSE ATTACK

Stateful firewalls still suffer from DoF attacks that can wreak havoc and exhaust all of the firewall's resources, especially the session table [1], [17] and [34]. Common mitigation mechanisms dealing with DoF session table attacks are threshold based mechanisms, as in Screen features used in Juniper Networks [34]. However, if threshold activation is set high, then attack traffic will pass through the firewall and can consume its resources. Moreover, most of these threshold based mechanisms are often not enabled and even if enabled, they dramatically increase CPU and session table utilization, such as in TCP SYN flood mechanism [35]. To solve threshold mechanisms complications, a mechanism is proposed in [18] to defend 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.

On the contrary, DoF BlackNurse attack is difficult to detect and manage. This is due to the fact that the ICMP port unreachable error message used in the BlackNurse attack is considered by the firewall as a related packet of a legitimate session. All firewalls have to inspect these related packets and combine them with their corresponding sessions if any, as discussed in Section V. The other issue is that the BlackNurse attack uses low volume traffic that is difficult to be detected and classified by the firewall as a DoF flooding traffic. Furthermore, all available BlackNurse attack mitigations have its own limitation as discussed in Section C. It is important to mention here that iptables rate limit ICMP port unreachable messages by default. This explains the reason that iptable-based products are unaffected by the BlackNurse attack [20]. 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. In addition, in [20] the following Snort rule is proposed as 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;

Likewise, 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.

To address the aforementioned limitations, we propose a reactive approach as opposed to the proactive approach used in the iptables. The approach is based on an adaptive early rejection rule to defend specifically against the BlackNurse attack. This rule will have a dynamic time of activity, during which the rule will be active. The rule is adaptive since it will not be active all the time, contrary to the discussed available BlackNurse mitigations. In addition, the rule activity time will not be fixed; however, it will be dynamic accommodating the statistics and severity of the BlackNurse attack history parameters.

Let's denote the BlackNurse attack early rejection rule with dynamic time-to-defend duration by *AER_BN*(*TTD*). The *AER_BN*(*TTD*) is triggered under certain conditions and given high priority in order to preserve firewall resources during the BlackNurse attack, especially the CPU. The *AER_BN*(*TTD*) is activated only once *fake* port unreachable messages reach a *minimum* certain rate limit R_x. Beyond that *minimum* R_x, the firewall CPU usage increases to $u\%$, and the firewall may become unresponsive. R_x and $u\%$ are firewall vendor dependent; however, their average values can be found experimentally. In [20], it was stated that a low bandwidth of around 15-18 Mbit/s (40-50) k packets of the BlackNurse per second is enough to overwhelm the affected firewall's CPU regardless of Internet connection capacity. Contrary, in the conducted lab experiments, we found that 7K packets of the BlackNurse per second was enough to increase Juniper NetScreen SSG 20 CPU utilization to 89%, while increase Cisco ASA 5540 to 20%.

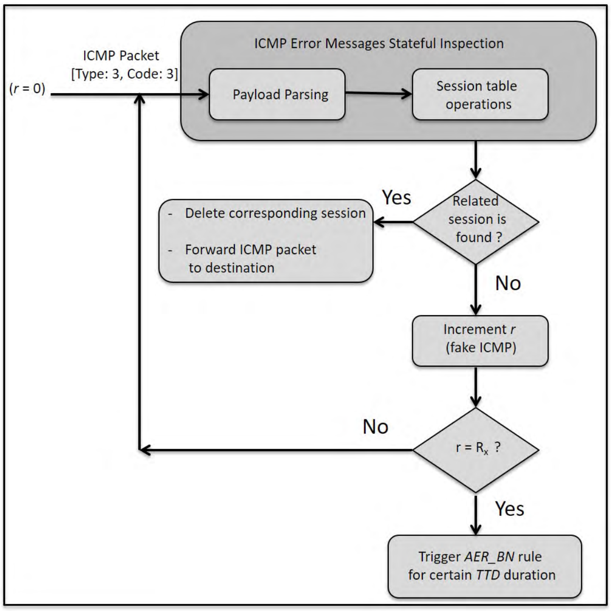

ICMP Error Messages Stateful Inspection
ICMP Packet
[Type: 3, Code: 3]
(r = 0)
Payload Parsing
Session table operations
Yes
Related session is found ?
- Delete corresponding session
- Forward ICMP packet to destination
No
Increment r (fake ICMP)
No
r = R_x ?
Yes
Trigger AER_BN rule for certain TTD duration

FIGURE 26. **AER_BN(TTD) rule triggering process.**

The *AER_BN*(*TTD*) rule has the following format:

Priority : **1**, **Prot** : **1**, **Scr_IP** :*, **Dst_IP** :*, **Type** : **3**, **Code** : **3**, **Action** : **Deny** ***for TTD section***

The *AER_BN*(*TTD*) rule is based on a triggering process, and a *TTD* duration modeling.

A. AER_BN(TTD) RULE TRIGGERING PROCESS

Upon receiving port unreachable message [Type:3, code:3], the firewall parses the payload and verifies it against the session table to find related session, as shown in Figure 26. If, no related session is found, this is considered as a fake message and the corresponding counter r is incremented. If the rate of such fake packets r reaches R_x, this gives an indication that a BlackNurse attack is undergoing. To preserve firewall resources especially the CPU utilization, the*AER_BN*(*TTD*) is activated for a certain time-to-defend duration (*TTD*), which is determined based on previous and current statistics and severity parameters of the BlackNurse attack.

B. TIME-TO-DEFEND (TTD) MODELING

Time-To-Defend (*TTD*) is defined as the time needed for a countermeasure *c* to gain some level of control on some attack vector *A*. Particular to our settings, *TTD* can be defined as the time required by the access control policy *c* to defend against low rate DoS attack targeting a firewall, specifically the BlackNurse attack *A*. Given the current *TTD* value and the current and previous BlackNurse attack statistics and severity parameters, the objective is to predict *TTD* that will be used next time when *AER_BN* rule is triggered.

We model the BlackNurse probability using *Poisson* counting process, making use of the simplified assumption that the BlackNurse attacks are discrete independent events, and their number BN_i in any observation interval of length *TTD* is Poisson distributed with mean λ *TTD* [36]. Thus, the probability of one or more BlackNurse attacks BN_i launched against the target firewall during an observation period *TTD* is given by the distribution:

$$P = 1 - e^{-\lambda TTD} \tag{1}$$

We model the severity of the BlackNurse attack during an observation period *TTD* by the maximum attack rate launched among $BN_i(R_i)$ during the observation period *TTD*, denoted by*Rmax*, where R_i is the i^{th} attack rate which may differ from one *TTD* to another.

Generally, at the current time t where $AER\_BN(TTD_{(t)})$ is activated for a $TTD_{(t)}$ duration, different BlackNurse attacks $BN_i(R_i)$ may happen. Thus, *TTD* for the next time $t + 1$, denoted by $TTD_{(t+1)}$ is computed based on the previous used$TTD_{(t)}$ and the previous distribution $P_{(t)}$ and $Rmax_{(t)}$ as follows:

$$TTD(\mathrm{t}+1) = (1 + \alpha + \beta) \times TTD(\mathrm{t}) \tag{2}$$

where:

$TTD_{(t)}$: is the current estimated state of *TTD* which is initialized to TTD_0 in the beginning.

$TTD_{(t+1)}$: is the next estimated state of *TTD*to be used when *AER_BN* is triggered next time.

$$\alpha = Rmax_{(t)} - -Rmax_{(t-1)}/Rmax_{(t-1)} \tag{3}$$

$$\beta = P_{(t)} - P_{(t-1)}/P_{(t-1)} \tag{4}$$

where:

$Rmax_{(t)}$: is the maximum rate of $BN_i(R_i)$ during current$TTD_{(t)}$ duration.

$Rmax_{(t-1)}$: is the maximum rate of $BN_i(R_i)$ during previous$TTD_{(t-1)}$ duration.

$\mathrm{P}_{(t)}$: is the Poisson distribution of $BN_i(R_i)$ during current$TTD_{(t)}$ duration.

$\mathrm{P}_{(t-1)}$: is the Poisson distribution of $BN_i(R_i)$ during previous$TTD_{(t-1)}$ duration.

Note that: $BN_i(R_i)$ may differ from one*TTD*to another.

The basic intuition behind α and β (Eq. 3 and 4) is that both of $R_{(t-1)}$ and $\mathrm{P}_{(t-1)}$ were sufficient to induce $TTD_{(t)}$. Hence, we use the change in R and P relative to their values to update *TTD*. Figure 28 illustrates the aforementioned *TTD* duration modeling process.

C. PERFORMANCE EVALUATION

The proposed *AER_BN(TTD)* rule triggering process and the time-to-defend modeling are implemented using Java programming language. According to [37], a system with no known vulnerabilities continues to be at risk because of vulnerabilities that exist, but are currently unknown or not active. The same concept we applied when implementing the proposed mechanism, a firewall continues to be at risk because of the BlackNurse vulnerability that exists but is not launched currently. We believe that *TTD* is adapted such that the likelihood of the BlackNurse attack and its risk on a firewall tend to be decreased. Thus, despite that during some time intervals where the BlackNurse attack is not launched and therefore the triggering process is not accomplished, *TTD* calculations are kept from the last previous time interval during which the BlackNurse attack had occurred. The parameters λ and *Rmax* are used in the proposed model to define the attacker skill level. For each occurrence of a BlackNurse attack $BN_i(R_i)$ that trigger the *AER_BN(TTD)* rule, *TTD* is calculated to be used in the next time of the attack occurrence. For instance, we simulated the proposed mechanism for two attacker skill levels, namely novice and expert. The triggering rate R_x is set for both as 7K packets per second. Two experiments are conducted to estimate the *TTD* using Eq.2 according to attacker skill levels.

Experiment 1: For a novice attacker, we choose λ = 2, and *Rmax* to vary between 2K-11K packets per second within a 30 time sequences. During this interval, *AER_BN(TTD)* rule tends to be on '1' and off '0' reflecting a novice attacker behavior, and *TTD* is calculated accordingly. Figure 28 shows the corresponding calculated *TTD* and rule activation time. For a novice attacker, some *Rmax*tends to be below the triggering R_x. This will result in a frequent inactivity of the *AER_BN(TTD)* rule, as the novice attacker is still learning and trying to configure the system behavior. From the experiment,

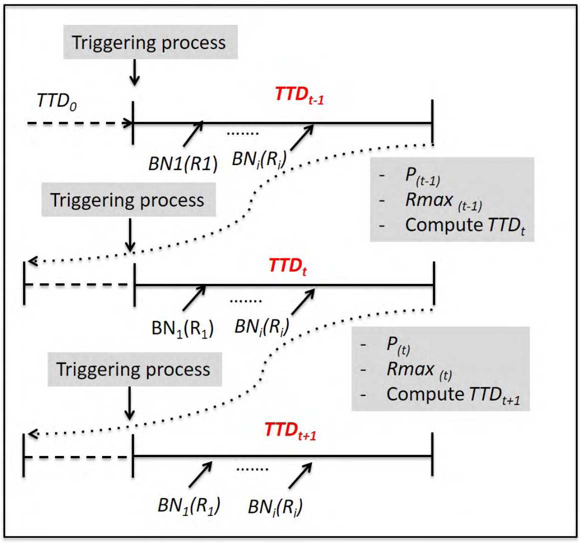


FIGURE 27. TTD duration modeling process.

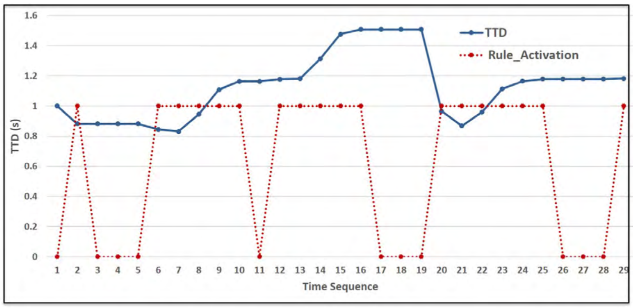


FIGURE 28. TTD and AER_BN(TTD) rule activation model for a novice attacker.

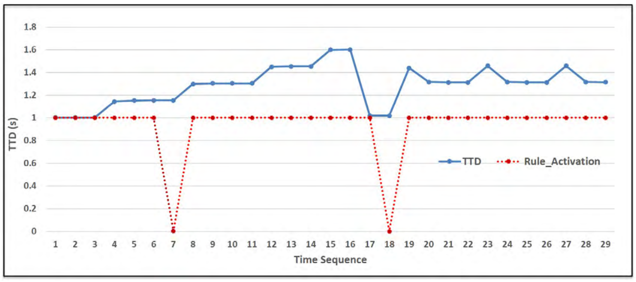


FIGURE 29. TTD and AER_BN(TTD) rule activation model for an expert attacker.

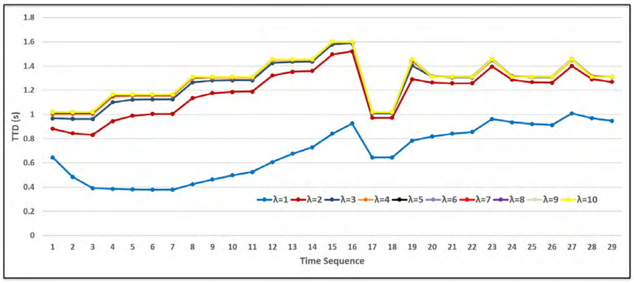


FIGURE 30. The impact of varying λ on TTD model for an expert attacker.

the calculated average *TTD* to represent a novice attacker is about 1.128 s.

Experiment 2: Higher skill levels are defined by increasing the value of the parameters λ and *Rmax*. For an expert attacker, we choose λ = 4 and *Rmax* to vary between 4K-11K within a 30 time sequences. Also, during this interval *AER_BN(TTD)* rule tends to be on '1' and off '0', and *TTD* is calculated accordingly. However, in the expert attacker model, the rule activation tends to be on '1' most of the time. Figure 29 shows the calculated *TTD* and rule activation time for an expert attacker. In this model, most of *Rmax* tends to be greater than the triggering R_x. Indeed, *Rmax* tends to reach the maximum rate in the specified range, as the attacker objective in this case is to overwhelm the firewall CPU. From the experiment, the calculated average *TTD* to represent an expert attacker is about 1.284 s which is 13% increase compared to *TTD* for a novice attacker.

As the attacker skill level increases, the average number of attack trials increases as expected. Usually, expert attacker uses DDoS attack traffic issued from many compromised hosts (botnets). To simulate BlackNurse DDoS attack situation, we used the same previous attack rate in Experiment 2, but the values of the parameter λ are varied from 1-10. Figure 30 shows the impact of λ on the *TTD* model. It is clearly shown that regardless of the nature and the number of botnets generating the BlackNurse DDoS traffic, the proposed *TTD* model tends to be stable in defending the attack and preserving the firewall resources.

VII. CONCLUSION

Usually, users with large bandwidth connections deemed themselves protected by professional firewalls against DDoS attacks that intend to overload their networks with malicious traffic. However, these firewalls themselves are exactly what the emerging BlackNurse attack is targeting. Instead of flooding the network with ICMP ping packets, the attack uses low volume of special crafted ICMP error messages of [Type:3, Code:3]. Susceptible firewalls show an immense increase in CPU utilization once the attack is launched. In addition, users or systems in the LAN side cannot exchange traffic with the WAN side as the intended firewall becomes unresponsive.

This paper gives a classification of DoF attacks according to the targeted firewall part, traffic volume and possible attack effect. The paper illustrates in details the BlackNurse attack principles as an example of DoF attacks. Practical experiments are provided to illustrate attack generation and available mitigation techniques. Experiments are conducted on commercial grade Juniper NetScreen SSG 20 and Cisco ASA 5540 to investigate the attack effect. The experiments showed that these firewalls can be vulnerable using a very low rate of 7K packets per second of the BlackNurse traffic from a single PC. The paper addresses limitations in the available BlackNurse attack mitigation mechanisms and offers a proposal based on an early rejection rule to defend against the BlackNurse attack. The proposed rule is adaptive with a triggering process to ensure rule activity only when the BlackNurse attack is undergoing. This feature will address the limitations in the available mitigation mechanisms, as iptable treats fake and legitimate destination unreachable messages in the same manner. Furthermore, the proposed rule has a dynamic time-to-defend duration that is estimated based on current and history attack statistics and severity parameters. Experiments are conducted to simulate the proposed mechanism against novice and expert attackers' behaviors. *TTD* tends to increase by 13% for expert attackers compared to novice attacker. In addition, the experiments guaranty that the *TDD* model tends to be stable without additional overhead and increment in the rule defense duration, regardless of the nature and the number of botnets in the case of DDoS BlackNurse attack.

For future work, we intend to generalize the idea of early rejection rule with time to defend duration to cover all ICMP hard error messages that can be used against the firewall itself. Each attack will have its corresponding early rejection rule which is triggered under certain circumstances with estimated time-to-defend duration. Iptables can be used as open source firewall to implement the proposed mechanism.

REFERENCES

[1] L. Alex, K. Amir, H. Joshua, G. Zihui, P. Dan, and W. Jia, "Firewall fingerprinting and denial of firewalling attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1699–1712 Jul. 2017.

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

[3] K. Salah, K. Sattar, M. Sqalli1, and E. Al-Shaer, "A potential low-rate DoS attack against network firewalls," *Secur. Commun. Netw.*, vol. 4, pp. 136–146, Jan. 2011.

[4] K. Sattar, K. Salah, M. Sqalli, R. Rafiq, and M. Rizwan, "A delay-based countermeasure against the discovery of default rules in firewalls," *Arabian J. Sci. Eng.*, vol. 42, no. 2, pp. 833–844, Feb. 2017.

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

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

[7] Z. Trabelsi, L. Zhang, and S. Zeidan, "Dynamic rule and rule-field optimisation for improving firewall performance and security," *IET Inf. Secur. J.*, vol. 8, no. 4, pp. 250–257, Jul. 2013.

[8] 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.

[9] 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.

[10] H. Hamed, A. El-Atawy, and E. Al-Shaer, "Adaptive Statistical Optimization Techniques for Firewall Packet Filtering," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[11] A. El-Atawy, T. Samak, E. Al-Shaer, and H. Li, "Using online traffic statistical matching for optimizing packet filtering performance," in *Proc. IEEE INFOCOM*, May 2007, pp. 866–874.

[12] E. Al-Shaer, A. El-Atawy, and T. Tran, "Adaptive early packet filtering for defending firewalls against DoS attacks," *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1–9.

[13] 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.

[14] T. Chomsiri, X. He, P. Nanda, and Z. Tan, "Hybrid tree-rule firewall for high speed data transmission," *IEEE Trans. Cloud Comput.*, to be published.

[15] 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.*, Jun. 2012, pp. 1089–1093.

[16] Z. Trabelsi, S. Zeidan, 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.

[17] S. Gill. (Jul. 2009). *Maximizing Firewall Availability*. [Online]. Available: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.4204&rep=rep1&type=pdf

[18] 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.

[19] Z. Trabelsi and S. Zeidan, "Enhanced Session Table Architecture for Stateful Firewalls," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

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

[21] S. Khandelwal. (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

[22] K. R. Fall and W. R. Stevens, *TCP/IP Illustrated*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2011.

[23] L. Zeltser, K. Kent, S. Northcutt, R. W. Ritchey, and S. Winters, *Inside Network Perimeter Security: Stateful Firewalls*. 2nd ed. 2005, p. 768.

[24] Avishai Wool. (2018). *Packet Filtering and Stateful Firewalls*. [Online]. Available: http://www.eng.tau.ac.il/~yash/hinsec-171.pdf

[25] Hank and Foo. (2016). *Stateful Firewalls*. [Online]. Available: http:// 1164 docplayer.net/3420645-Stateful-?rewalls-hank-and-foo.html

[26] (2018). *Connection Settings*. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/security/asa/asa93/configuration/firewall/asa-firewall-cli/conns-connlimits.pdf

[27] ScreenOS. (2016). *How to Determine the Timeout of A Session and How do They Work*. [Online]. Available: https://kb.juniper.net/InfoCenter/index?page=content&id=kb7046

[28] (2015). *Iptables: Difference Between NEW, ESTABLISHED and RELATED Packets*. [Online]. Available: https://serverfault.com/questions/371316/iptables-difference-between-new-established-and-related-packets

[29] K. Salah, K. Elbadawi, and R. Boutaba, "Performance modeling and analysis of network firewalls," *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 1, pp. 12–21, Mar. 2012.

[30] (2015). *Frameip Packet Generator Tool*. [Online]. Available: https://www.frameip.com

[31] (2000). *Hping3 Tool*. [Online]. Available: http://www.hping.org

[32] (2016). *LiteServe Software*. [Online]. Available: http://www.cmfperception.com

[33] (2019). *CommView Tool*. [Online]. Available: http://www.tamos.com

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

[35] (2016). *The Next BIG Thing: 'Small,' DDoS Attacks are Often Hardest to Block*. [Online]. Available: https://www.redwolfsecurity.com/small-ddos-attacks/

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

[37] M. McQueen, W. Boyer Mark, A. George, and A. Beitel, "Time-to-compromise model for cyber risk reduction estimation," in *Proc. Eur. Symp. Res. Comput. Secur.*, Aug. 2005, pp. 12–25.

ZOUHEIR TRABELSI received the Ph.D. degree in computer science from the Tokyo University of Technology and Agriculture, Japan, in 1994. From 1994 to 1998, he was a Computer Science Researcher with the Central Research Laboratory of Hitachi, Tokyo, Japan. From 2001 to 2002, he was a Visiting Assistant Professor with Pace University, New York, NY, USA. In 2005, he joined the College of Information Technology, United Arab Emirates University, where he is currently a Full Professor of information security and the Master Programs Coordinator. His research interests include network security, intrusion detection and prevention, firewalls, TCP/IP covert channels, and 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. She has around 13 publications in well-known conferences and journals. Her research interest includes firewall optimization techniques during normal and attack situations.

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 a Member of the Cyber Security Research Group, College of Information Technology, United Arab Emirates University, 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 include information security and privacy challenges of emerging technologies, such as the IoT, cloud computing, social networks, blockchain, autonomous vehicles, and intrusion detection systems.