

LAB REPORT

Subject: Intrusion Detection and Prevention System

Session 03

Topic name: Write Snort rule

1. GENERAL INFORMATION:

Num	Full name	Student ID	Email
1	Nguyen Dinh Kha	20520562	20520562@gm.uit.edu.vn
2	Le Sy Cuong	20521149	20521149@gm.uit.edu.vn

2. IMPLEMENTATION CONTENT

Num	Work	Personal Responsible	Self-assessment result
1	Requirement 1.1	Nguyen Dinh Kha	100%
2	Requirement 1.2	Nguyen Dinh Kha	100%
3	Requirement 1.3	Nguyen Dinh Kha	100%
4	Requirement 1.4	Le Sy Cuong	100%
5	Requirement 1.5	Le Sy Cuong	100%

The section below of this report is the detailed documentation from the practical group.

DETAILED REPORT

Students undertake the practical exercise with the requirements below.

A. OVERVIEW

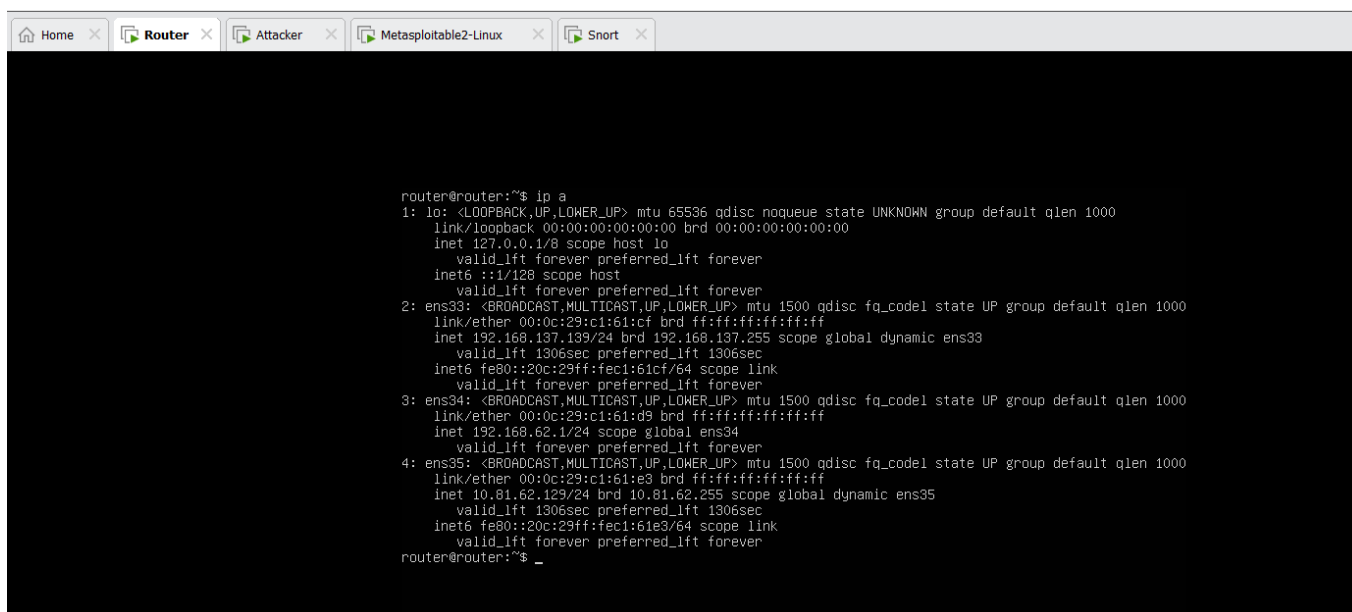
A.1 Objectives

- Learn and write rules for Snort
(https://www.snort.org/documents#latest_rule_documents)
- Analyze traffic flows before and after rule deployment.

A.2 Setting up the environment

- Use the environment set up in practical exercise 02.

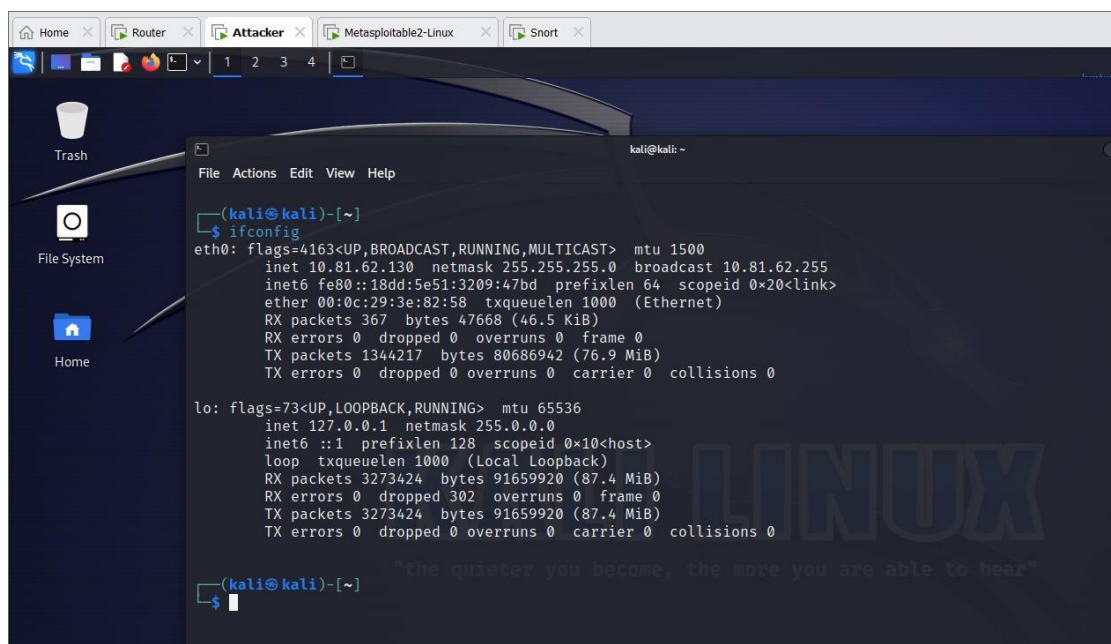
- Configure the IP of the Router:



The screenshot shows a terminal window with the title bar containing tabs for Home, Router, Attacker, Metasploitable2-Linux, and Snort. The terminal is running the 'ip a' command on the Router, displaying the following output:

```
router@router:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c1:61:cf brd ff:ff:ff:ff:ff:ff
    inet 192.168.137.139/24 brd 192.168.137.255 scope global dynamic ens33
        valid_lft 1306sec preferred_lft 1306sec
    inet6 fe80::20c:29ff:fe01:61cf/64 scope link
        valid_lft forever preferred_lft forever
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c1:61:d9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.62.1/24 scope global ens34
        valid_lft forever preferred_lft forever
4: ens35: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c1:61:e3 brd ff:ff:ff:ff:ff:ff
    inet 10.81.62.129/24 brd 10.81.62.255 scope global dynamic ens35
        valid_lft 1306sec preferred_lft 1306sec
    inet6 fe80::20c:29ff:fe01:61e3/64 scope link
        valid_lft forever preferred_lft forever
router@router:~$ _
```

- Configure the IP of the Attacker's machine:



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window has a title bar with tabs for Home, Router, Attacker, Metasploitable2-Linux, and Snort. The terminal is running the 'ifconfig' command on the Attacker, displaying the following output:

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.81.62.130 netmask 255.255.255.0 broadcast 10.81.62.255
    inet6 fe80::18dd:5e51:3209:47bd prefixlen 64 scopeid 0<link>
    ether 00:0c:29:3e:82:58 txqueuelen 1000 (Ethernet)
    RX packets 367 bytes 47668 (46.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 134217 bytes 80686942 (76.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 3273424 bytes 91659920 (87.4 MiB)
    RX errors 0 dropped 302 overruns 0 frame 0
    TX packets 3273424 bytes 91659920 (87.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
$
```

- Configure the IP of the Victim's machine:

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:c2:c9:0c brd ff:ff:ff:ff:ff:ff
    inet 192.168.62.200/24 brd 192.168.62.255 scope global eth0
        inet6 fe80::20c:29ff:fec2:c90c/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

- Configure the IP of the Snort machine:

```
snort@snort:/var/log/snort$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:c2:cf:48:69 brd ff:ff:ff:ff:ff:ff
    altnme enp2s1
    inet 192.168.137.141/24 metric 100 brd 192.168.137.255 scope global dynamic ens33
        valid_lft 1498sec preferred_lft 1498sec
    inet6 fe80::20c:29ff:fece:4869/64 scope link
        valid_lft forever preferred_lft forever
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:c2:cf:48:73 brd ff:ff:ff:ff:ff:ff
    altnme enp2s2
    inet6 fe80::20c:29ff:fece:4873/64 scope link
        valid_lft forever preferred_lft forever
4: ens35: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:c2:cf:48:7d brd ff:ff:ff:ff:ff:ff
    altnme enp2s3
    inet6 fe80::20c:29ff:fece:487d/64 scope link
        valid_lft forever preferred_lft forever
snort@snort:/var/log/snort$ _
```

- Run Snort Inline:

```
snort@snort:/var/log/snort$ sudo snort -c /etc/snort/nhom4-snort.conf -Q -i ens35:ens34_
```

```
Home Router Attacker Metasploitable2-Linux Snort

+-----[Rule Port Counts]-----+
|   src   tcp   udp   icmp   ip   |
|   dst   0     0     0     0     |
|   any   0     0     0     0     |
|   nc    0     0     0     0     |
|   s+d   0     0     0     0     |
+-----+

+-----[detection-filter-config]-----+
| memory-cap : 1048576 bytes          |
+-----+
+-----[detection-filter-rules]-----+
| none                                |
+-----+

+-----[rate-filter-config]-----+
| memory-cap : 1048576 bytes          |
+-----+
+-----[rate-filter-rules]-----+
| none                                |
+-----+

+-----[event-filter-config]-----+
| memory-cap : 1048576 bytes          |
+-----+
+-----[event-filter-global]-----+
| none                                |
+-----+
+-----[event-filter-local]-----+
| none                                |
+-----+
+-----[suppression]-----+
| none                                |
+-----+

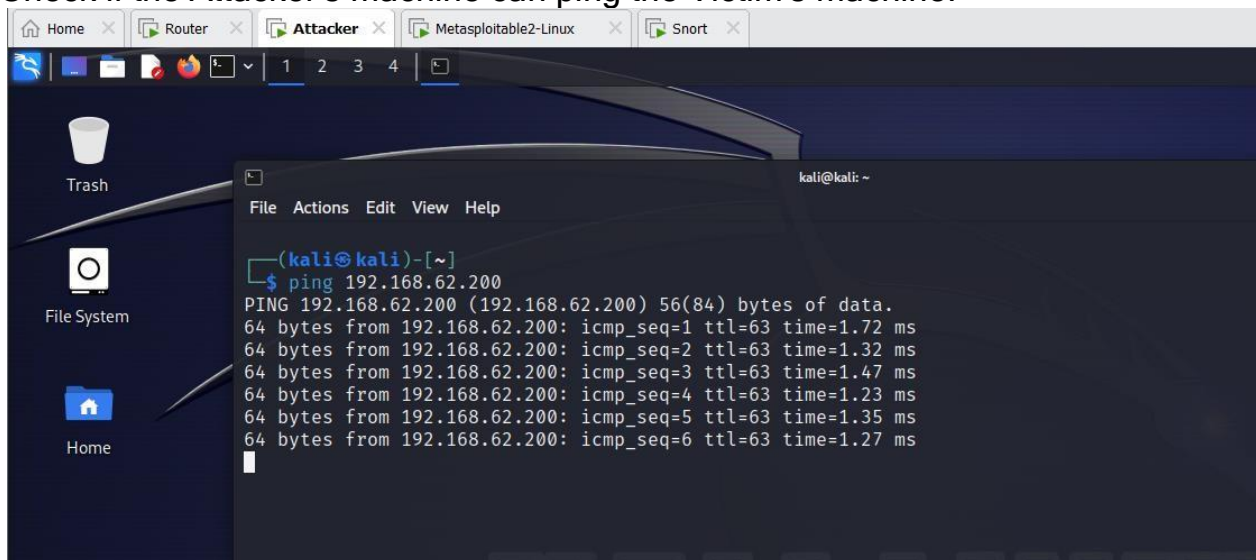
Rule application order: pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!
afpacket DAQ configured to inline.
Acquiring network traffic from "ens3:ens34".
Reload thread starting...
Reload thread started, thread 0x7fa8dde206c0 (2037)

--= Initialization Complete ==--

~*~ Snort! ~*~
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.3 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.13

Commencing packet processing (pid=2028)
Decoding Ethernet
```

- Check if the Attacker's machine can ping the Victim's machine:



```
kali@kali: ~
File Actions Edit View Help

(kali@kali)-[~]
$ ping 192.168.62.200
PING 192.168.62.200 (192.168.62.200) 56(84) bytes of data.
64 bytes from 192.168.62.200: icmp_seq=1 ttl=63 time=1.72 ms
64 bytes from 192.168.62.200: icmp_seq=2 ttl=63 time=1.32 ms
64 bytes from 192.168.62.200: icmp_seq=3 ttl=63 time=1.47 ms
64 bytes from 192.168.62.200: icmp_seq=4 ttl=63 time=1.23 ms
64 bytes from 192.168.62.200: icmp_seq=5 ttl=63 time=1.35 ms
64 bytes from 192.168.62.200: icmp_seq=6 ttl=63 time=1.27 ms
```

- Check if the Victim's machine can ping the Attacker's machine:

```
msfadmin@metasploitable:~$ ping 10.81.62.130
PING 10.81.62.130 (10.81.62.130) 56(84) bytes of data:
64 bytes from 10.81.62.130: icmp_seq=1 ttl=63 time=1.28 ms
64 bytes from 10.81.62.130: icmp_seq=2 ttl=63 time=1.24 ms
64 bytes from 10.81.62.130: icmp_seq=3 ttl=63 time=1.71 ms
64 bytes from 10.81.62.130: icmp_seq=4 ttl=63 time=1.26 ms
64 bytes from 10.81.62.130: icmp_seq=5 ttl=63 time=1.40 ms
64 bytes from 10.81.62.130: icmp_seq=6 ttl=63 time=1.31 ms
64 bytes from 10.81.62.130: icmp_seq=7 ttl=63 time=1.44 ms
64 bytes from 10.81.62.130: icmp_seq=8 ttl=63 time=1.65 ms
64 bytes from 10.81.62.130: icmp_seq=9 ttl=63 time=1.21 ms
64 bytes from 10.81.62.130: icmp_seq=10 ttl=63 time=1.30 ms
64 bytes from 10.81.62.130: icmp_seq=11 ttl=63 time=1.43 ms
64 bytes from 10.81.62.130: icmp_seq=12 ttl=63 time=1.23 ms
64 bytes from 10.81.62.130: icmp_seq=13 ttl=63 time=1.37 ms
64 bytes from 10.81.62.130: icmp_seq=14 ttl=63 time=1.76 ms
64 bytes from 10.81.62.130: icmp_seq=15 ttl=63 time=1.41 ms
```

- Kiểm tra máy Attacker ping ra internet:

```
(kali@kali)-[~]
$ ping google.com
PING google.com (216.58.200.238) 56(84) bytes of data:
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=1 ttl=127 time=58.6 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=2 ttl=127 time=33.9 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=3 ttl=127 time=33.8 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=4 ttl=127 time=35.1 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=5 ttl=127 time=33.5 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=6 ttl=127 time=34.1 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=7 ttl=127 time=33.9 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=8 ttl=127 time=34.2 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=9 ttl=127 time=34.9 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=10 ttl=127 time=36.8 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=11 ttl=127 time=33.6 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=12 ttl=127 time=33.7 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=13 ttl=127 time=34.4 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=14 ttl=127 time=34.4 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=15 ttl=127 time=34.9 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=16 ttl=127 time=34.0 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=17 ttl=127 time=34.8 ms
```

- Check if the Attacker's machine can ping the internet:

```
msfadmin@metasploitable:~$ ping google.com
PING google.com (216.58.200.238) 56(84) bytes of data:
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=1 ttl=127 time=34.2 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=2 ttl=127 time=34.9 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=3 ttl=127 time=34.3 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=4 ttl=127 time=38.0 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=5 ttl=127 time=34.1 ms
64 bytes from 238.200.58.216.in-addr.arpa (216.58.200.238): icmp_seq=6 ttl=127 time=35.4 ms
```

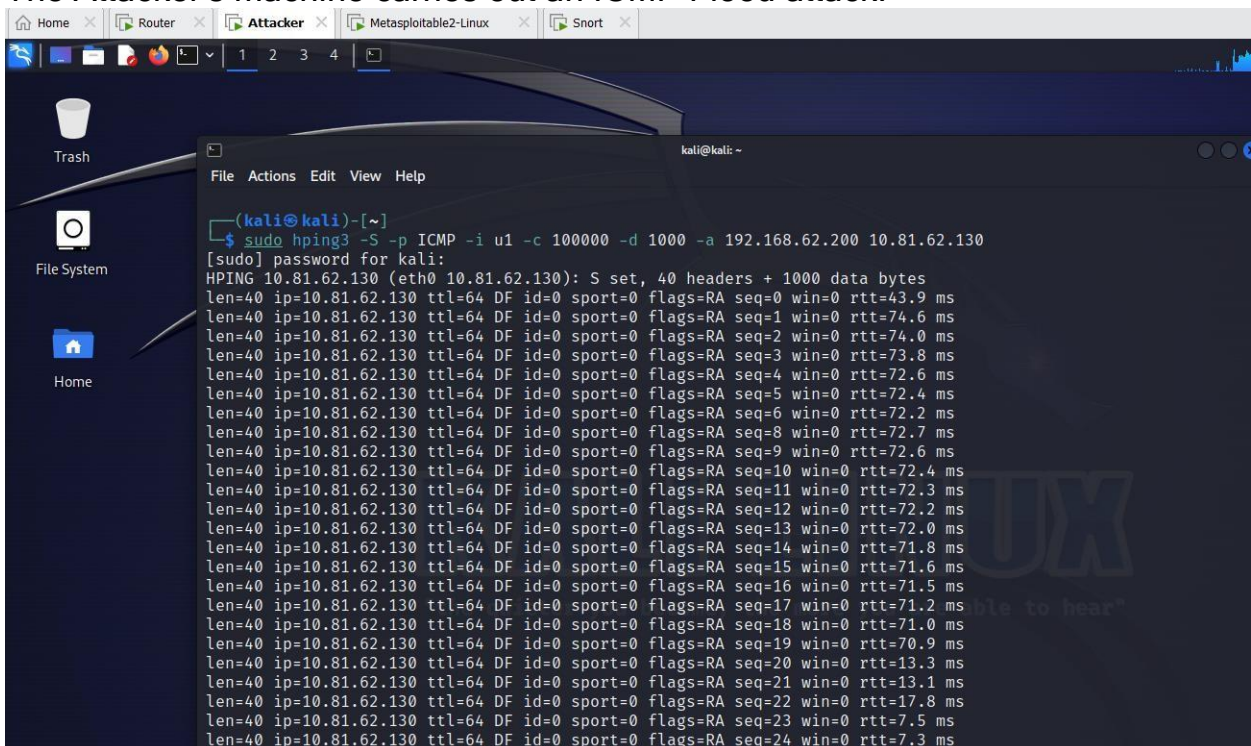

B. PRACTICE

Students will perform practical exercises with the following requirements.

B.1 Write a rule for Snort

Requirement 1.1 Prevent ICMP Flood attacks

- Write a Snort rule to limit ICMP packets to the Victim's machine. The threshold is not more than 23 packets/5s.
 - Use the hping3 tool on the Attacker's machine to carry out the attack.
 - Check the results before and after installing the rule.
- Before installing the rule:
 - The Attacker's machine carries out an ICMP Flood attack:



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal displays the execution of the hping3 tool to perform an ICMP flood attack. The command used is `sudo hping3 -S -p ICMP -i u1 -c 100000 -d 1000 -a 192.168.62.200 10.81.62.130`. The output shows a series of 24 ICMP echo requests (seq=0 to seq=24) sent to the victim IP 10.81.62.130. Each request is 40 bytes long (4 bytes header + 36 bytes data) and has a TTL of 64. The round-trip times (rtt) are consistently around 70-75 ms. The attack is being performed from the attacker's IP 192.168.62.200.

```
(kali@kali)-[~]
$ sudo hping3 -S -p ICMP -i u1 -c 100000 -d 1000 -a 192.168.62.200 10.81.62.130
[sudo] password for kali:
HPING 10.81.62.130 (eth0 10.81.62.130): S set, 40 headers + 1000 data bytes
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=43.9 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=74.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=74.0 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=73.8 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=72.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=72.4 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=72.2 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=72.7 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=72.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=10 win=0 rtt=72.4 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=11 win=0 rtt=72.3 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=12 win=0 rtt=72.2 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=13 win=0 rtt=72.0 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=14 win=0 rtt=71.8 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=15 win=0 rtt=71.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=16 win=0 rtt=71.5 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=17 win=0 rtt=71.2 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=18 win=0 rtt=71.0 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=19 win=0 rtt=70.9 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=20 win=0 rtt=13.3 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=21 win=0 rtt=13.1 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=22 win=0 rtt=17.8 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=23 win=0 rtt=7.5 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=24 win=0 rtt=7.3 ms
```

- Victim's machine: Use tcpdump to check if the Victim's machine has been attacked by ICMP Flood:

```
Home Router Attacker Metasploitable2-Linux Snort

02:58:59.580366 IP 10.81.62.130.0 > 192.168.62.200.9937: R 0:0(0) ack 1271452230
win 0
02:58:59.580454 IP 10.81.62.130.0 > 192.168.62.200.9938: R 0:0(0) ack 1352815421
win 0
02:58:59.580556 IP 10.81.62.130.0 > 192.168.62.200.9939: R 0:0(0) ack 1263648525
win 0
02:58:59.580663 IP 10.81.62.130.0 > 192.168.62.200.9940: R 0:0(0) ack 976794618
win 0
02:58:59.580781 IP 10.81.62.130.0 > 192.168.62.200.9941: R 0:0(0) ack 2021910026
win 0
02:58:59.580852 IP 10.81.62.130.0 > 192.168.62.200.9942: R 0:0(0) ack 13970830 u
in 0
02:58:59.582903 IP 10.81.62.130.0 > 192.168.62.200.9960: R 0:0(0) ack 1121557075
win 0
02:58:59.588014 IP 10.81.62.130.0 > 192.168.62.200.10013: R 0:0(0) ack 193465547
6 win 0
02:58:59.592707 IP 10.81.62.130.0 > 192.168.62.200.10062: R 0:0(0) ack 235795573
win 0
02:58:59.599124 IP 10.81.62.130.0 > 192.168.62.200.10123: R 0:0(0) ack 44824735
win 0
02:58:59.601404 IP 10.81.62.130.0 > 192.168.62.200.10124: R 0:0(0) ack 139468436
win 0
02:58:59.604462 IP 10.81.62.130.0 > 192.168.62.200.10125: R 0:0(0) ack 60913987
win 0
02:58:59.605970 IP 10.81.62.130.0 > 192.168.62.200.10126: R 0:0(0) ack 157488336
```

```
Home Router Attacker Metasploitable2-Linux Snort

02:59:08.943865 IP 10.81.62.130.0 > 192.168.62.200.27962: R 0:0(0) ack 143405319
2 win 0
02:59:08.945379 IP 10.81.62.130.0 > 192.168.62.200.27963: R 0:0(0) ack 176849522
5 win 0
02:59:08.946879 IP 10.81.62.130.0 > 192.168.62.200.27964: R 0:0(0) ack 142953669
2 win 0
02:59:08.948460 IP 10.81.62.130.0 > 192.168.62.200.27965: R 0:0(0) ack 656500861
win 0
02:59:08.949956 IP 10.81.62.130.0 > 192.168.62.200.27966: R 0:0(0) ack 311628228
win 0
02:59:08.951540 IP 10.81.62.130.0 > 192.168.62.200.27967: R 0:0(0) ack 536389711
win 0
02:59:08.952959 IP 10.81.62.130.0 > 192.168.62.200.27968: R 0:0(0) ack 148007702
1 win 0
02:59:08.954297 IP 10.81.62.130.0 > 192.168.62.200.27969: R 0:0(0) ack 822012934
win 0
02:59:08.955880 IP 10.81.62.130.0 > 192.168.62.200.27970: R 0:0(0) ack 224322355
win 0
02:59:08.957309 IP 10.81.62.130.0 > 192.168.62.200.27971: R 0:0(0) ack 195403632
6 win 0

7906 packets captured
60739 packets received by filter
52470 packets dropped by kernel
msfadmin@metasploitable:~$ _
```

⇒ Using tcpdump, we observe that the Victim's machine has been attacked by ICMP Flood.

⇒ However, Snort did not generate an alert because no rule has been set for Snort yet:

```
Home Router Attacker Metasploit2-Linux Snort

Would icmp4::echo: 0
Would ip6::ttl: 0
Would ip6::ttl: 0
Would ip6::opts: 10
Would ip6::opts: 0
Would icmp6::echo: 0
Would icmp6::echo: 0
Would tcp::syn_opt: 0
Would tcp::syn_opt: 0
Would tcp::opt: 0
Would tcp::opt: 0
Would tcp::pad: 0
Would tcp::pad: 0
Would tcp::rsv: 0
Would tcp::rsv: 0
Would tcp::ns: 0
Would tcp::ns: 0
Would tcp::urp: 0
Would tcp::urp: 0
Would tcp::ecn_pkt: 0
Would tcp::ecn_pkt: 0
Would tcp::ts_ecn: 0
Would tcp::ts_ecn: 0
Would tcp::req_urg: 0
Would tcp::req_urg: 0
Would tcp::req_pay: 0
Would tcp::req_pay: 0
Would tcp::req_urp: 0
Would tcp::req_urp: 0
Would tcp::ecn_ssn: 0
Would tcp::ecn_ssn: 0
Would tcp::ts_nop: 0
Would tcp::ts_nop: 0
Would tcp::ips_data: 0
Would tcp::ips_data: 0
Would tcp::block: 0
Would tcp::block: 0
Would tcp::trim_syn: 0
Would tcp::trim_syn: 0
Would tcp::trim_rst: 0
Would tcp::trim_rst: 0
Would tcp::trim_win: 0
Would tcp::trim_win: 0
Would tcp::trim_mss: 0
Would tcp::trim_mss: 0
=====
Snort exiting
snort@snort:/var/log/snort$
snort@snort:/var/log/snort$ cat alert
snort@snort:/var/log/snort$
```

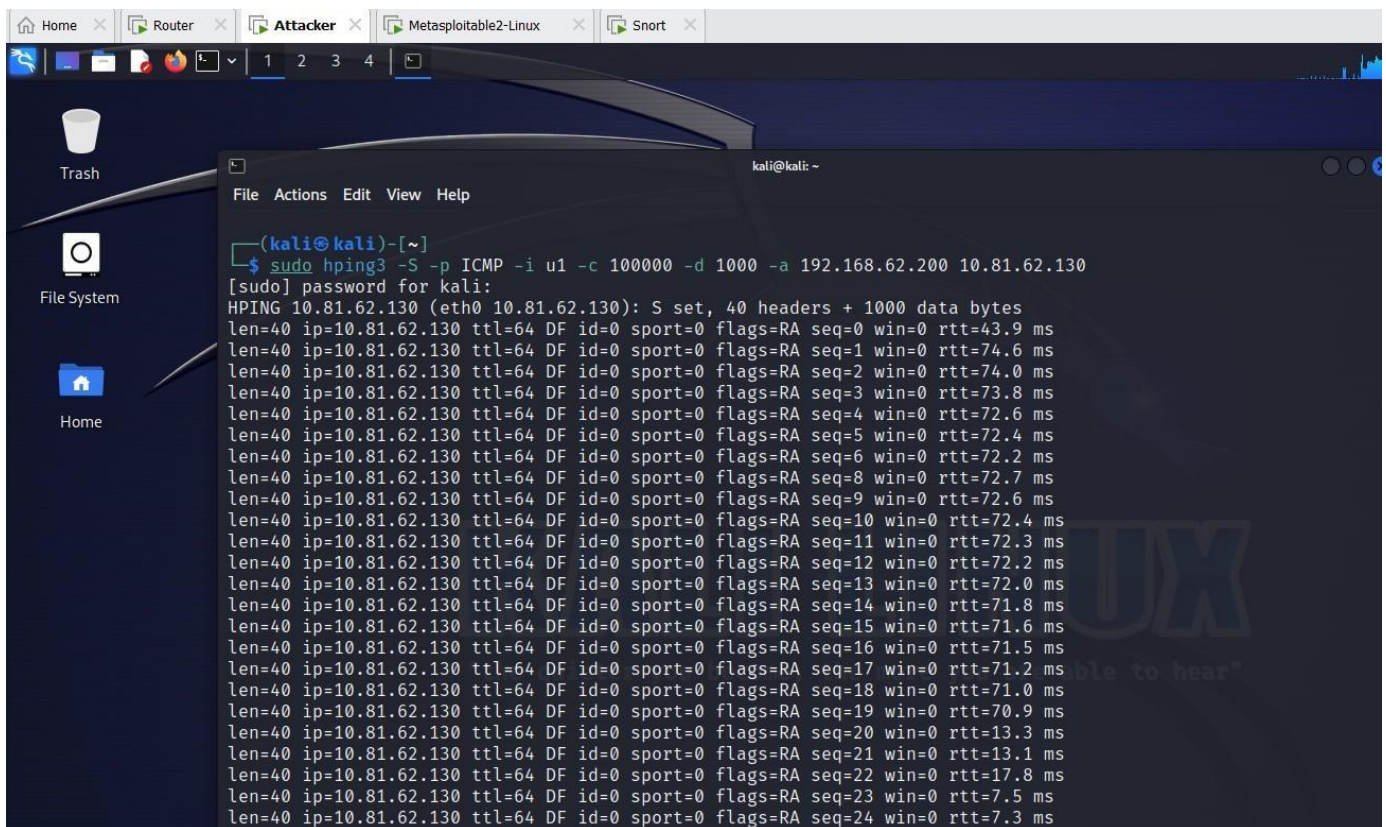
⇒ Checking the alert file of Snort, we see that it's empty, meaning Snort has not detected anything because no rule has been set.

- After setting the rule:
- We proceed to set the rule for Snort:

```
snort@snort:/var/log/snort$
snort@snort:/var/log/snort$ cat /etc/snort/rules/rhom4.rules
alert icmp any any -> 192.168.62.200 any (msg: "ICMP Flood to 192.168.62.200"; itype:8; threshold: type both ,track by_src, count 23, seconds 5; sid:1000001; rev:1;)
snort@snort:/var/log/snort$ _
```

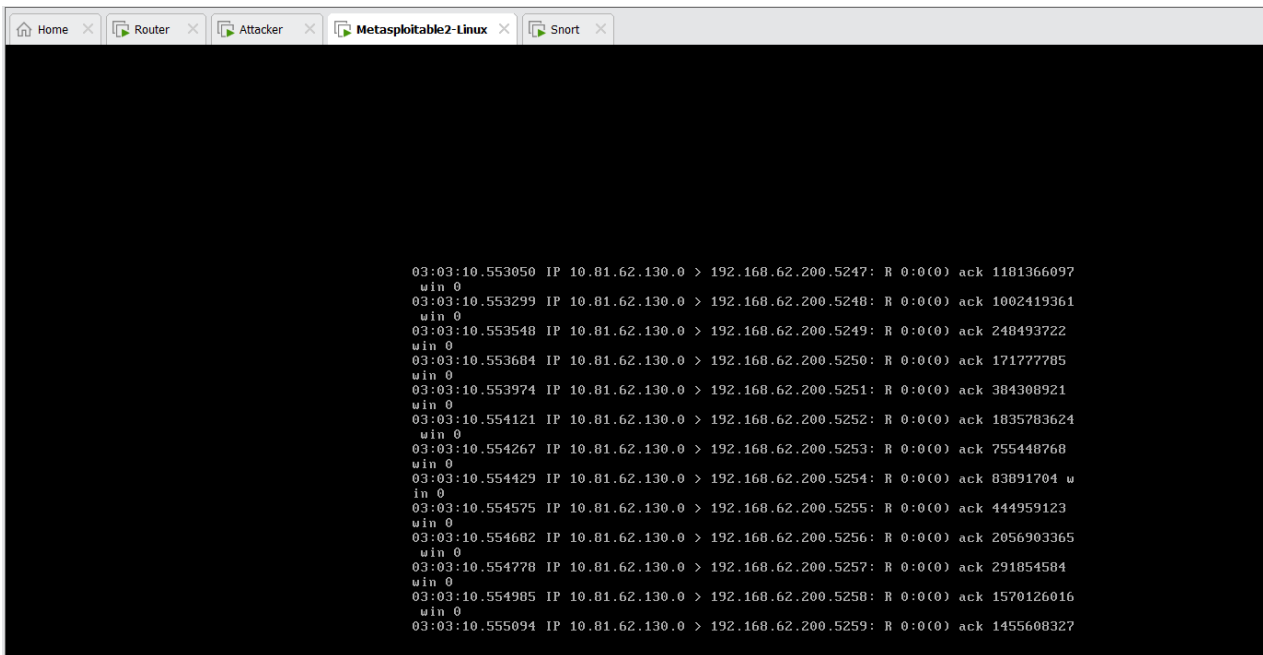
(alert icmp any any -> 192.168.62.200 any (msg: "ICMP Flood to 192.168.62.200"; itype:8; threshold: type both ,track by_src, count 23, seconds 5; sid: 1000001; rev:1;)

- After setting the rule, we carry out the ICMP Flood attack again on the Attacker's machine:



```
(kali@kali)-[~]
$ sudo hping3 -S -p ICMP -i u1 -c 100000 -d 1000 -a 192.168.62.200 10.81.62.130
[sudo] password for kali:
HPING 10.81.62.130 (eth0 10.81.62.130): S set, 40 headers + 1000 data bytes
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=43.9 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=74.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=74.0 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=73.8 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=72.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=72.4 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=72.2 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=8 win=0 rtt=72.7 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=9 win=0 rtt=72.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=10 win=0 rtt=72.4 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=11 win=0 rtt=72.3 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=12 win=0 rtt=72.2 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=13 win=0 rtt=72.0 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=14 win=0 rtt=71.8 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=15 win=0 rtt=71.6 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=16 win=0 rtt=71.5 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=17 win=0 rtt=71.2 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=18 win=0 rtt=71.0 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=19 win=0 rtt=70.9 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=20 win=0 rtt=13.3 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=21 win=0 rtt=13.1 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=22 win=0 rtt=17.8 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=23 win=0 rtt=7.5 ms
len=40 ip=10.81.62.130 ttl=64 DF id=0 sport=0 flags=RA seq=24 win=0 rtt=7.3 ms
```

- The Victim's machine uses tcpdump to check:



```
03:03:10.553050 IP 10.81.62.130.0 > 192.168.62.200.5247: R 0:0(0) ack 1181366097
win 0
03:03:10.553299 IP 10.81.62.130.0 > 192.168.62.200.5248: R 0:0(0) ack 1002419361
win 0
03:03:10.553548 IP 10.81.62.130.0 > 192.168.62.200.5249: R 0:0(0) ack 248493722
win 0
03:03:10.553684 IP 10.81.62.130.0 > 192.168.62.200.5250: R 0:0(0) ack 171777785
win 0
03:03:10.553974 IP 10.81.62.130.0 > 192.168.62.200.5251: R 0:0(0) ack 384308921
win 0
03:03:10.554121 IP 10.81.62.130.0 > 192.168.62.200.5252: R 0:0(0) ack 1835783624
win 0
03:03:10.554267 IP 10.81.62.130.0 > 192.168.62.200.5253: R 0:0(0) ack 755448768
win 0
03:03:10.554429 IP 10.81.62.130.0 > 192.168.62.200.5254: R 0:0(0) ack 83891704 u
in 0
03:03:10.554575 IP 10.81.62.130.0 > 192.168.62.200.5255: R 0:0(0) ack 444959123
win 0
03:03:10.554682 IP 10.81.62.130.0 > 192.168.62.200.5256: R 0:0(0) ack 2056903365
win 0
03:03:10.554778 IP 10.81.62.130.0 > 192.168.62.200.5257: R 0:0(0) ack 291854584
win 0
03:03:10.554985 IP 10.81.62.130.0 > 192.168.62.200.5258: R 0:0(0) ack 1570126016
win 0
03:03:10.555094 IP 10.81.62.130.0 > 192.168.62.200.5259: R 0:0(0) ack 1455608327
```

- On the Snort side, ICMP Flood has been detected:

```
Home Router Attacker Metasploitable2-Linux Snort

snort@snort:/var/log/snort$ cat alert
[**] [1:1000001:1] ICMP Flood to 192.168.62.200 [**]
[Priority: 0]
11/02-07:19:35.300132 10.81.62.130 -> 192.168.62.200
ICMP TTL:63 TOS:0x0 ID:52448 IpLen:20 DgmLen:28
Type:8 Code:0 ID:52749 Seq:5632 ECHO

[**] [1:1000001:1] ICMP Flood to 192.168.62.200 [**]
[Priority: 0]
11/02-07:19:40.211232 10.81.62.130 -> 192.168.62.200
ICMP TTL:63 TOS:0x0 ID:50122 IpLen:20 DgmLen:28
Type:8 Code:0 ID:52749 Seq:35681 ECHO

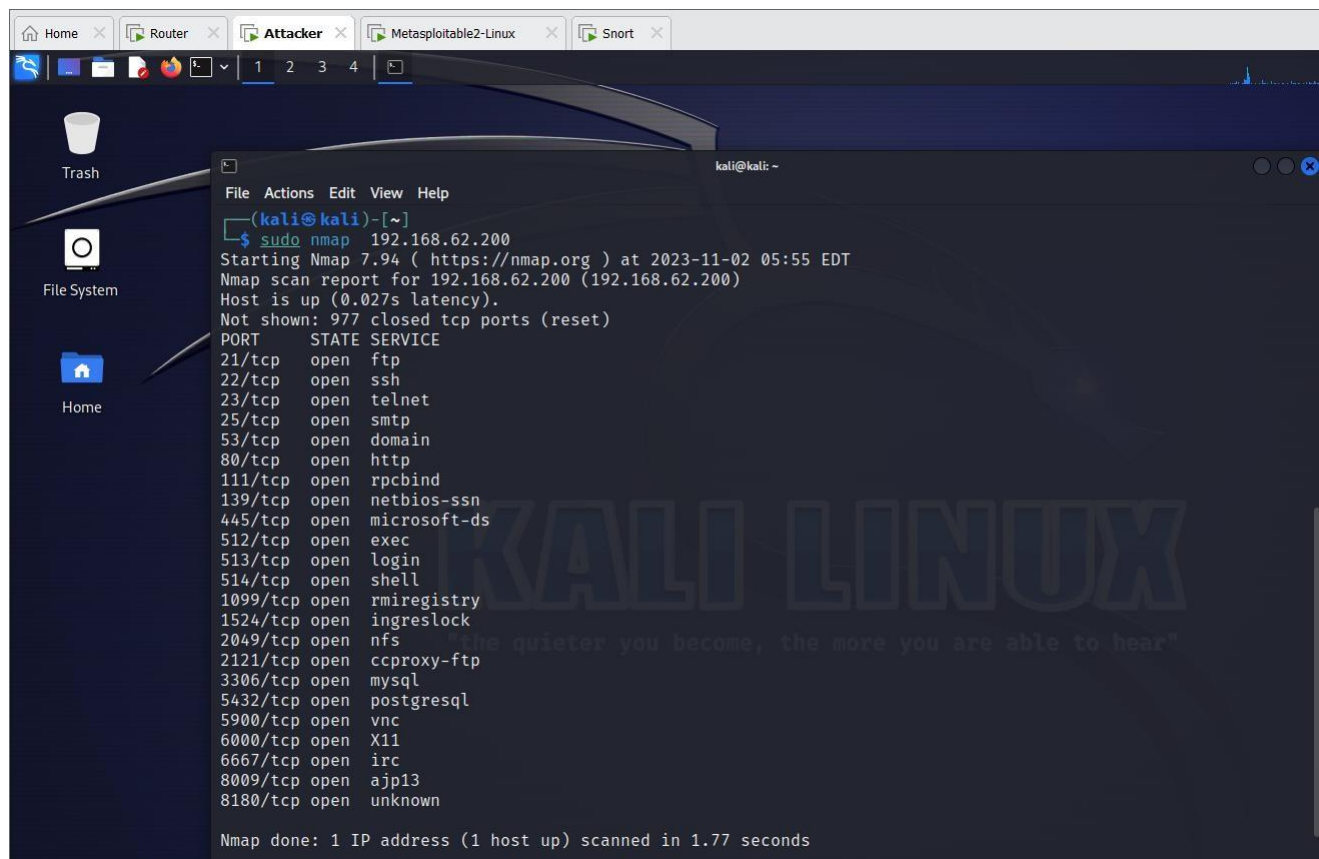
[**] [1:1000001:1] ICMP Flood to 192.168.62.200 [**]
[Priority: 0]
11/02-07:19:54.590885 10.81.62.130 -> 192.168.62.200
ICMP TTL:63 TOS:0x0 ID:62682 IpLen:20 DgmLen:28
Type:8 Code:0 ID:52749 Seq:58947 ECHO

snort@snort:/var/log/snort$ _
```

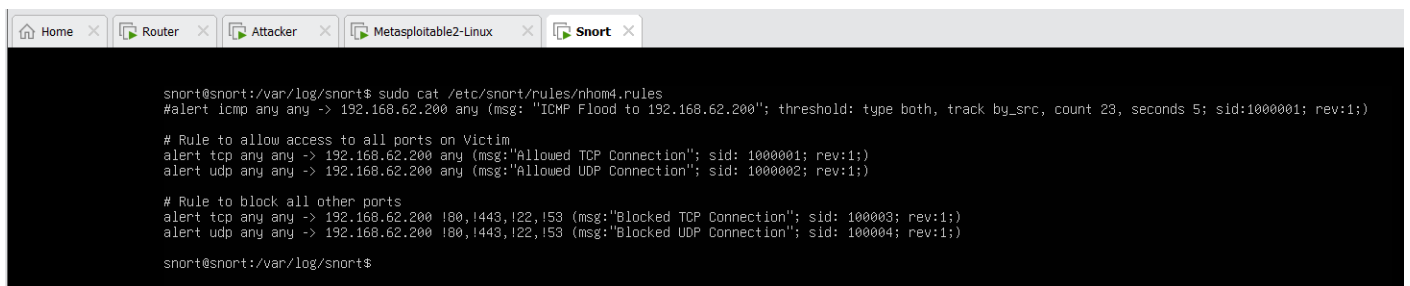
⇒ Snort has detected and alerted ICMP Flood to the Victim's machine (192.168.62.200)

Requirement 1.2 Allow access only to services running on the Victim

- Use nmap to scan open ports on the Victim's machine.
 - Write a Snort rule to only allow access to the open ports of the Victim's machine. Block all other ports.
 - Use telnet or nmap tools on the Attacker's machine to carry out the attack.
 - Check the results before and after installing the rule.
-
- Before setting the rule:
 - The Attacker's machine conducts an nmap attack:



- After setting the rule:
- We proceed to set the rule for Snort:



- ⇒ The # Rule to allow access to ports on the victim should be changed from alert to pass because alert is used to see if the rule works or not.
- ⇒ The part to Block ports should set a rule using alert except for ports (80, 443, 22, 53)

The Attacker attacks nmap on some specified ports:

```
(kali@kali)-[~]
$ sudo nmap -sT -p 80 -Pn 192.168.62.200

Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-02 06:32 EDT
Nmap scan report for 192.168.62.200 (192.168.62.200)
Host is up (0.0018s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds

(kali@kali)-[~]
$ sudo nmap -sT -p 443 -Pn 192.168.62.200

Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-02 06:32 EDT
Nmap scan report for 192.168.62.200 (192.168.62.200)
Host is up (0.016s latency).

PORT      STATE SERVICE
443/tcp   closed https

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds

(kali@kali)-[~]
$ sudo nmap -sT -p 22 -Pn 192.168.62.200

Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-02 06:33 EDT
Nmap scan report for 192.168.62.200 (192.168.62.200)
Host is up (0.0032s latency).

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds

(kali@kali)-[~]
$ sudo nmap -sT -p 53 -Pn 192.168.62.200

Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-02 06:33 EDT
Nmap scan report for 192.168.62.200 (192.168.62.200)
Host is up (0.0018s latency).

PORT      STATE SERVICE
53/tcp    open  domain

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

- On the Snort machine, alerts have appeared:

```
Home x Router x Attacker x Metasploitable2-Linux x Snort x

[**] [1:100003:1] Blocked TCP Connection [**]
[Priority: 0]
11/02-10:34:12.469425 10.81.62.130:32982 -> 192.168.62.200:25
TCP TTL:63 TOS:0x0 ID:43925 IpLen:20 DgmLen:52 DF
*****R** Seq: 0x44903587 Ack: 0x26F50536 Win: 0x1F6 TcpLen: 32
TCP Options (3) => NOP NOP TS: 8044956 959815

[**] [1:100001:1] Allowed TCP Connection [**]
[Priority: 0]
11/02-10:34:31.907143 10.81.62.130:34684 -> 192.168.62.200:23
TCP TTL:63 TOS:0x0 ID:32943 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x584FF22D Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 8064394 0 NOP WS: 7

[**] [1:100003:1] Blocked TCP Connection [**]
[Priority: 0]
11/02-10:34:31.907143 10.81.62.130:34684 -> 192.168.62.200:23
TCP TTL:63 TOS:0x0 ID:32943 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x584FF22D Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 8064394 0 NOP WS: 7

[**] [1:100001:1] Allowed TCP Connection [**]
[Priority: 0]
11/02-10:34:31.908460 10.81.62.130:34684 -> 192.168.62.200:23
TCP TTL:63 TOS:0x0 ID:32944 IpLen:20 DgmLen:52 DF
***** Seq: 0x584FF22E Ack: 0x39FA6C6F Win: 0x1F6 TcpLen: 32
TCP Options (3) => NOP NOP TS: 8064395 961759
```

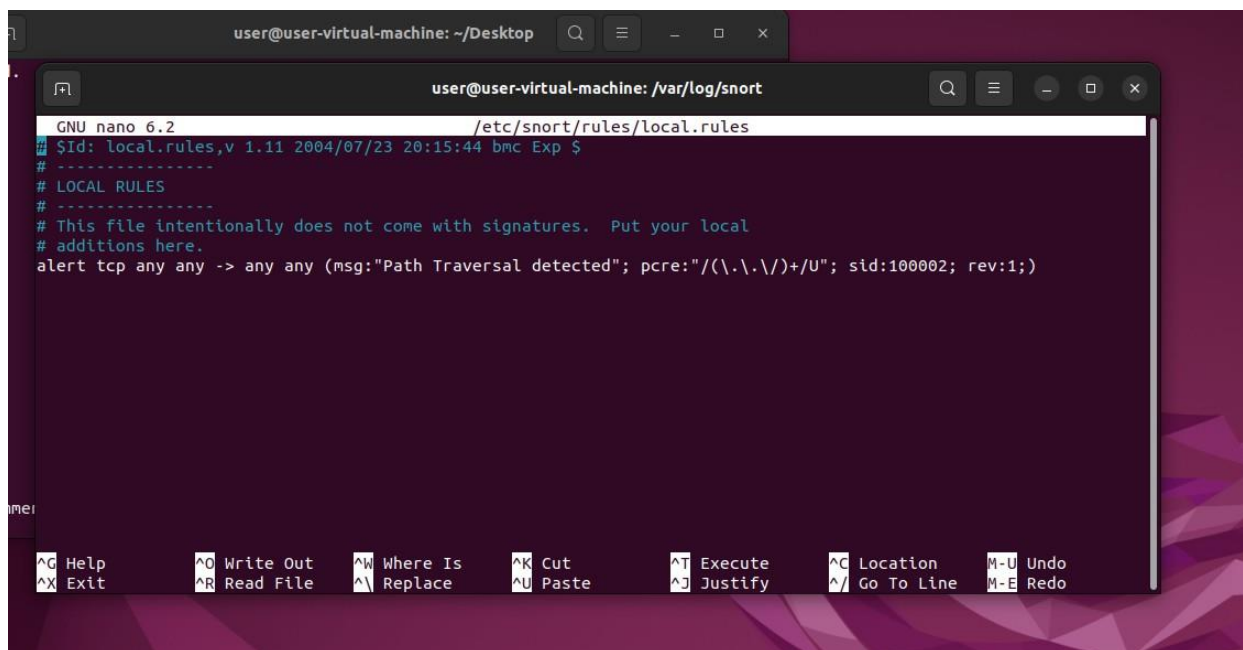
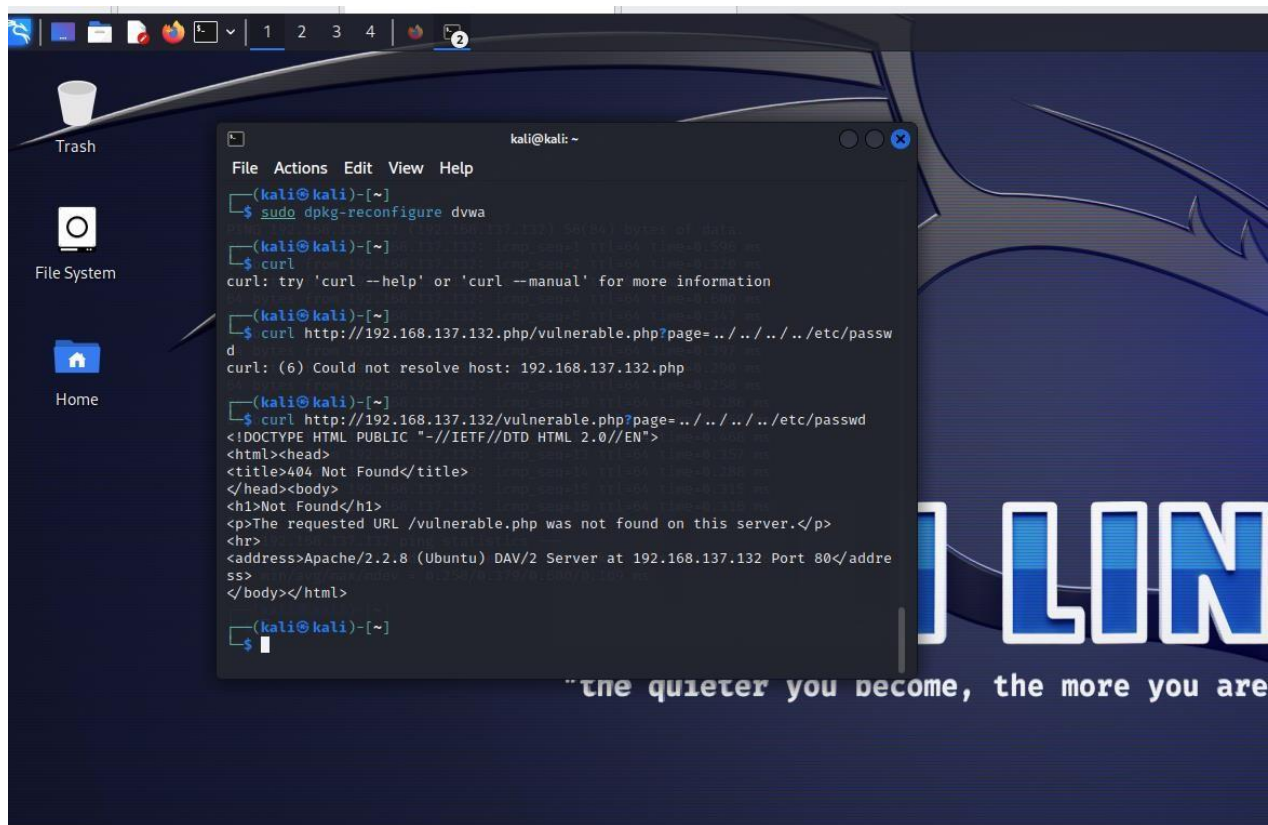
Requirement 1.3 Prevent password scanning attacks on the Web application

- Access the Mutillidae web application (/mutillidae/index.php?page=login.php) on the Victim's machine. Write a Snort rule to prevent password login scanning attacks on this web application. Note: only block password scanning, the web application should still be accessible normally.
- Use the hydra tool on the Attacker's machine to carry out the attack.
- Check the results before and after installing the rule.

Requirement 1.4 Prevent Path Traversal attacks

- Write a Snort rule to block Path Traversal attacks.
- On the Attacker's machine, access the path <http://192.168.x.200/dvwa/> to carry out the attack.
- Check the results before and after carrying out the attack.

● We proceed to carry out the Path Traversal attack with the Curl command, the Victim's machine IP is 192.168.137.132



- ⇒ Set Snort rule to detect Path Traversal with content that will detect strings containing one or more repetitions of "../", which is also a sign of Path Traversal


```
user@user-virtual-machine: ~/Desktop
ved.
user@user-virtual-machine: /var/log/snort
10/31-13:18:25.713689  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [C
lassification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.137.1:5
4354 -> 239.255.255.250:1900
10/31-13:18:26.727373  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [C
lassification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.137.1:5
4354 -> 239.255.255.250:1900
10/31-13:18:26.727376  [**] [1:1917:6] SCAN UPnP service discover attempt [**] [C
lassification: Detection of a Network Scan] [Priority: 3] {UDP} 192.168.137.1:5
4354 -> 239.255.255.250:1900
10/31-13:18:35.513784  [**] [1:1122:5] WEB-MISC /etc/passwd [**] [Classification
: Attempted Information Leak] [Priority: 2] {TCP} 192.168.137.134:60154 -> 192.1
68.137.132:80
10/31-13:18:35.513784  [**] [1:1113:5] WEB-MISC http directory traversal [**] [C
lassification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.137.134:6
0154 -> 192.168.137.132:80
10/31-13:18:35.513783  [**] [1:1122:5] WEB-MISC /etc/passwd [**] [Classification
: Attempted Information Leak] [Priority: 2] {TCP} 192.168.137.134:60154 -> 192.1
68.137.132:80
10/31-13:18:35.513783  [**] [1:1113:5] WEB-MISC http directory traversal [**] [C
lassification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.137.134:6
0154 -> 192.168.137.132:80
10/31-13:18:35.513783  [**] [1:100002:1] Path Traversal detected [**] [Priority:
0] {TCP} 192.168.137.134:60154 -> 192.168.137.132:80
user@user-virtual-machine: /var/log/snort$ S
```

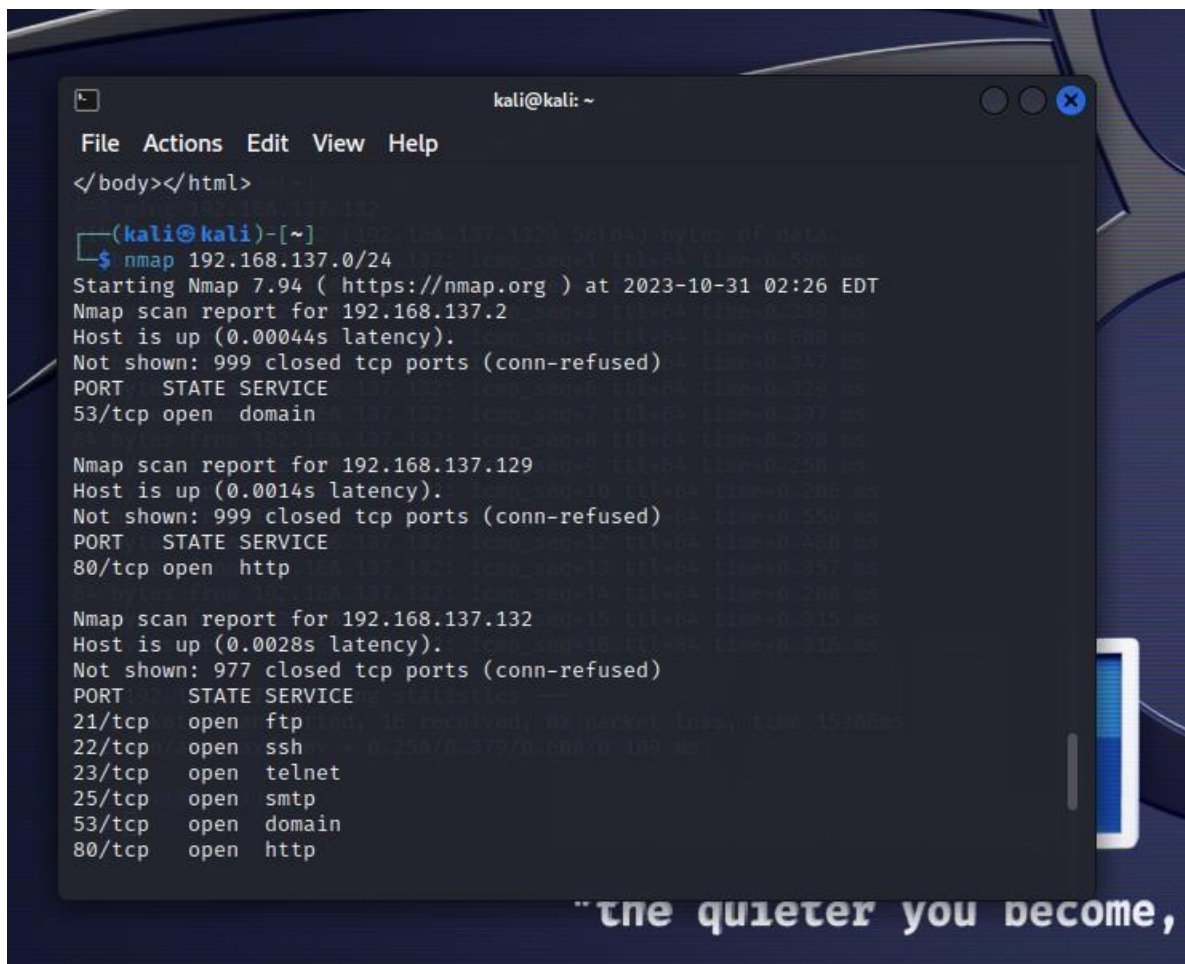
⇒ On the Snort machine, Path Traversal attack has been detected

Requirement 1.5 Students are required to develop 2 additional attack scenarios and write Snort rules to prevent attacks.

- Students independently develop 2 unrelated attack scenarios not related to DoS and web attacks, then write Snort rules to prevent the attacks.
- Perform writing Snort rules, check the results before and after the attacks as above requirements.
- Evaluation depends on the complexity of the scenarios.

⇒ .

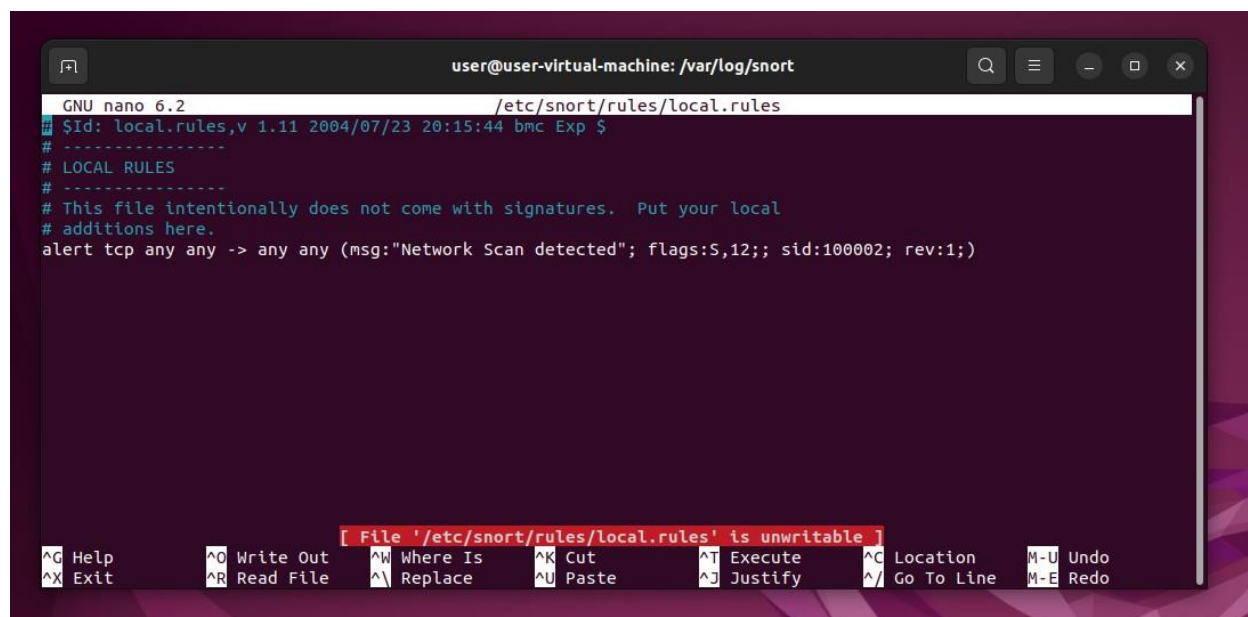
Network scan attack:



```
kali@kali: ~  
File Actions Edit View Help  
</body></html>  
(kali@kali)-[~]  
$ nmap 192.168.137.0/24  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-31 02:26 EDT  
Nmap scan report for 192.168.137.2  
Host is up (0.00044s latency).  
Not shown: 999 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
53/tcp    open  domain  
Nmap scan report for 192.168.137.129  
Host is up (0.0014s latency).  
Not shown: 999 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
80/tcp    open  http  
Nmap scan report for 192.168.137.132  
Host is up (0.0028s latency).  
Not shown: 977 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http
```

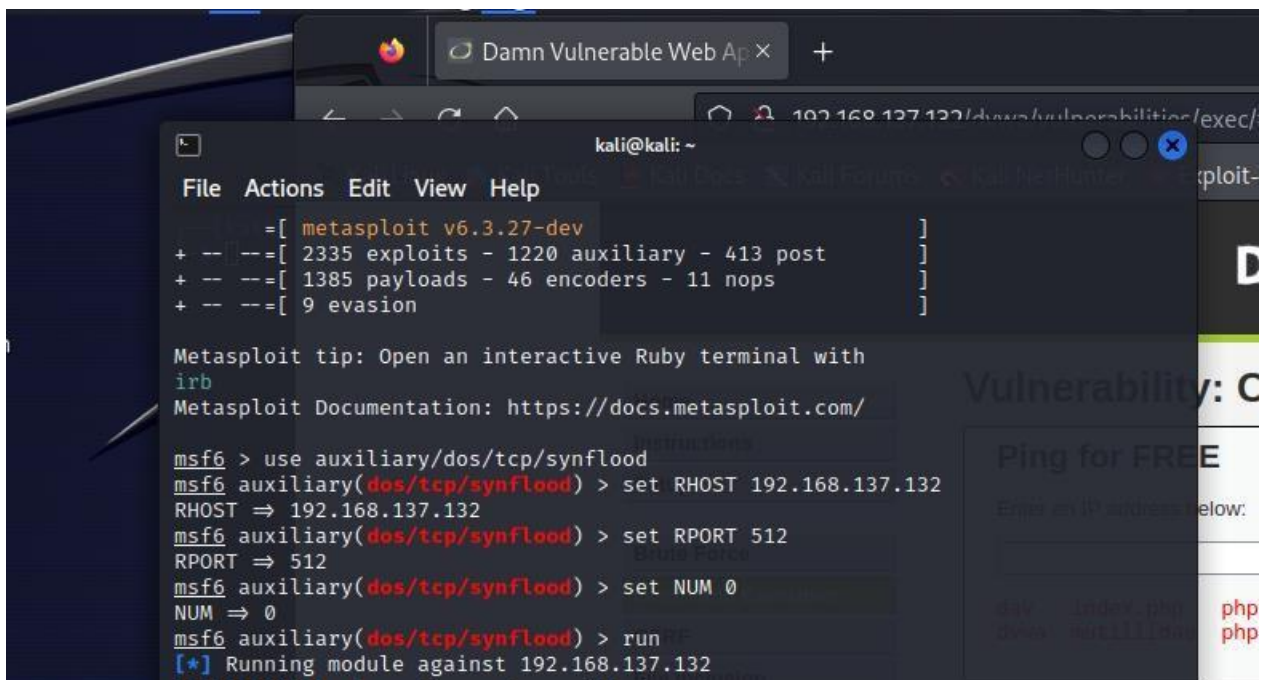
On the Attacker's machine, we perform a scan using nmap.

On the Snort machine, we set rules to detect Network Scan:



```
user@user-virtual-machine: /var/log/snort  
GNU nano 6.2 /etc/snort/rules/local.rules  
$Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $  
#  
# LOCAL RULES  
#  
# This file intentionally does not come with signatures. Put your local  
# additions here.  
alert tcp any any -> any any (msg:"Network Scan detected"; flags:S,12;; sid:100002; rev:1;)  
[ File '/etc/snort/rules/local.rules' is unwritable ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/_ Go To Line  M-E Redo
```

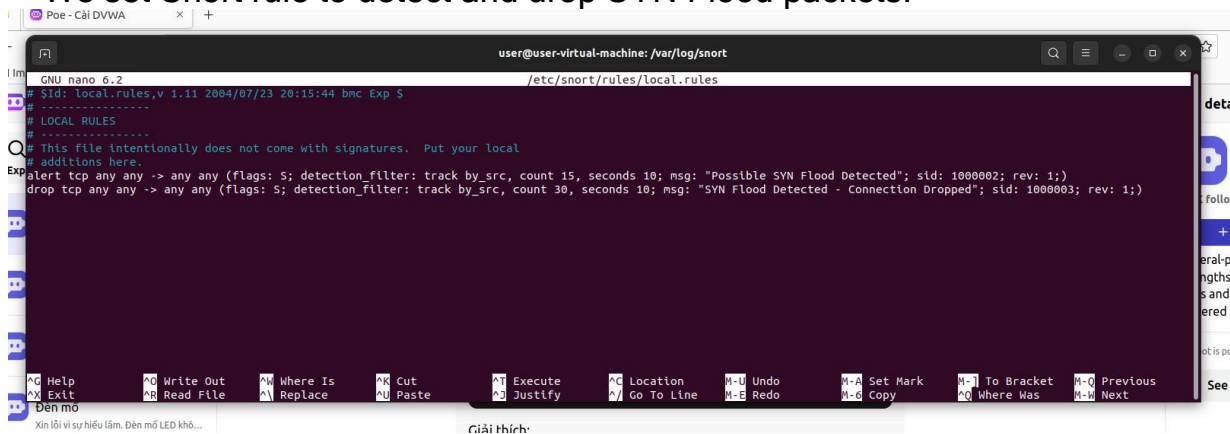
⇒ Alert tcp any any -> any any (msg:"Network Scan detected"; flags:S,12;; sid:100002; rev:1;)



```
kali@kali: ~  
File Actions Edit View Help  
=[ metasploit v6.3.27-dev ]  
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post ]  
+ -- --=[ 1385 payloads - 46 encoders - 11 nops ]  
+ -- --=[ 9 evasion ]  
  
Metasploit tip: Open an interactive Ruby terminal with  
irb  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use auxiliary/dos/tcp/synflood  
msf6 auxiliary(dos/tcp/synflood) > set RHOST 192.168.137.132  
RHOST => 192.168.137.132  
msf6 auxiliary(dos/tcp/synflood) > set RPORT 512  
RPORT => 512  
msf6 auxiliary(dos/tcp/synflood) > set NUM 0  
NUM => 0  
msf6 auxiliary(dos/tcp/synflood) > run  
[*] Running module against 192.168.137.132
```

⇒ Proceed to set RHOST, RPORT, NUM values and carry out the SYN Flood attack to the Victim's machine

- We set Snort rule to detect and drop SYN Flood packets:



```
user@user-virtual-machine: /var/log/snort  
GNU nano 6.2 /etc/snort/rules/local.rules  
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $  
# LOCAL RULES  
# -----  
# This file intentionally does not come with signatures. Put your local  
# additions here.  
alert tcp any any -> any any (flags: S; detection_filter: track by_src, count 15, seconds 10; msg: "Possible SYN Flood Detected"; sid: 1000002; rev: 1;)  
drop tcp any any -> any any (flags: S; detection_filter: track by_src, count 30, seconds 10; msg: "SYN Flood Detected - Connection Dropped"; sid: 1000003; rev: 1;)  
  
Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous  
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was Next  
Den mo  
Xin lỗi vì sự hiểu lầm. Đèn mô LED khó...  
Giải thích
```

Activities Terminal 99 1 18:45

Poe - Cài DVWA

user@user-virtual-machine: /var/log/snort

```
4598 -> 192.168.137.132:2525
11/01-18:37:56.062968 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:5
2016 -> 192.168.137.132:2041
11/01-18:37:56.063029 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:5
2436 -> 192.168.137.132:32775
11/01-18:37:56.063036 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:5
0838 -> 192.168.137.132:6001
11/01-18:37:56.063114 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:3
8848 -> 192.168.137.132:1272
11/01-18:37:56.063114 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:3
9088 -> 192.168.137.132:2717
11/01-18:37:56.063181 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:5
2902 -> 192.168.137.132:1098
11/01-18:37:56.063182 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:4
7646 -> 192.168.137.132:32773
11/01-18:37:56.063272 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:5
2848 -> 192.168.137.132:4125
11/01-18:37:56.063285 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:5
2750 -> 192.168.137.132:6666
11/01-18:37:56.063578 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:4
5370 -> 192.168.137.132:3889
11/01-18:37:56.063635 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:4
5906 -> 192.168.137.132:2022
11/01-18:37:56.063649 [**] [1:1000002:1] Possible SYN Flood Detected [**] [Priority: 0] {TCP} 192.168.137.134:6
```

Đèn mờ

Xin lỗi vì sự hiểu lầm. Đèn mờ LED khó...

Assistant Oct 31 >

Cách tấn công

Tôi xin lỗi vì sự nhầm lẫn. Nếu bạn gặp...

Giải thích:

- Luật đầu tiên (sid: 1000002) được sử dụng để phát hiện SYN Flood. Nó sẽ tạo một cảnh báo khi Snort phát hiện một lượng SYN packet lớn từ một nguồn IP trong một khoảng thời gian

và chặn synflood

bạn có thể sử

filter: track

ood Detecte

filter: track b

od - Connecti

Copy



After setting the rule, the result shows that Snort has detected suspicious attacks