جامعة جدة
University of Jeddah

Network Security Project

Made By :

Osama Abdulaziz Zarad 2040912

Yousef Alfuhaid 2040300

Abdulrahman Aladdin 2040606

Network Security (T29)

Introduction

At this project we are going to simulate and perform a passive attack "Nmap Scan" to web server, then we are going to analysis the traffic in the server using "Wireshark" to know how to detect Nmap scan. Next, we are going to apply firewall and IDS to avoid such attacks.

In our scenario we have two virtual machines , the first one is Web Server "Ubuntu" which has IP address is 10.0.2.15 , the attacker machine "Ubuntu" his IP address is 10.0.2.4. All the machines are in the same network.

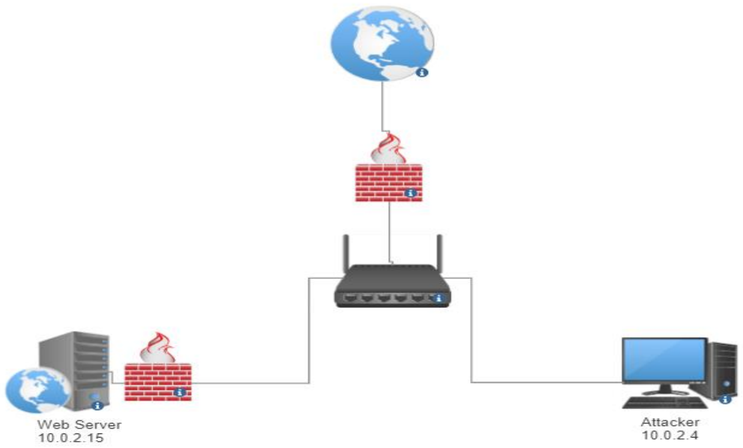


Figure 1: Network Diagram AS shown above the network diagram with IP addresses of each end point.

Tools will be needed.

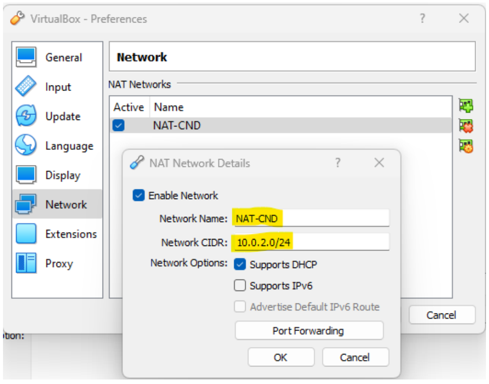
Attacker Machine	Server Machine
Nmap	Apache server
	Wireshark
	Snort
	Iptables

Part 1: Network Setup

1.1: Server Setting

In our project we are going to use VirtualBox as hypervisor and we are going to configure the server and attacker Ubuntu machines as following.

First configure NAT "VLAN" in VirtualBox , file→ properties→ Network



Network

NAT Networks

Active	Name
☑	NAT-CND

☑ Enable Network

Network Name: NAT-CND

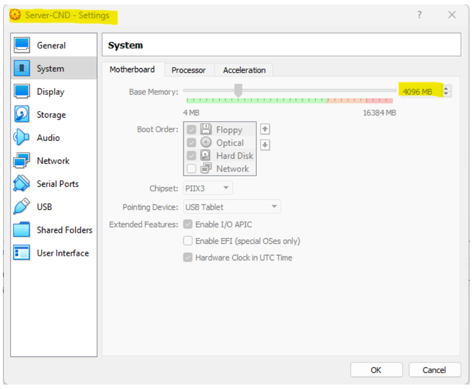
Network CIDR: 10.0.2.0/24

Network Options: ☑ Supports DHCP

□ Supports IPv6

□ Advertise Default IPv6 Route

Then configuring server machine in VirtualBox:



System

Base Memory: 4096 MB

Boot Order: ☑ Floppy ☑ Optical ☑ Hard Disk □ Network

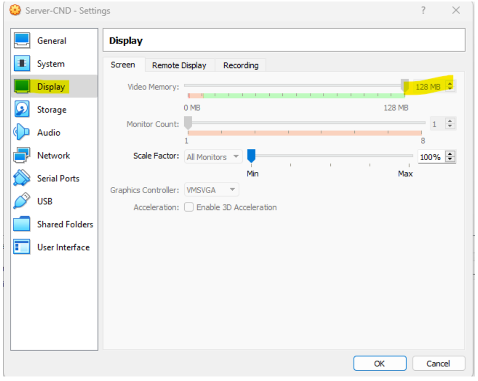
Chipset: PIIX3

Pointing Device: USB Tablet

Extended Features: ☑ Enable I/O APIC

□ Enable EFI (special OSes only)

☑ Hardware Clock in UTC Time



Display

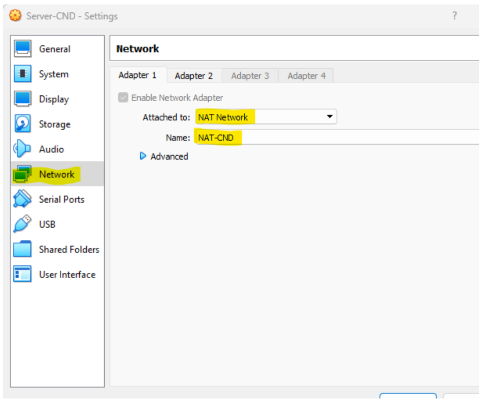
Video Memory: 128 MB

Monitor Count: 1

Scale Factor: All Monitors 100%

Graphics Controller: VMSVGA

Acceleration: □ Enable 3D Acceleration



Network

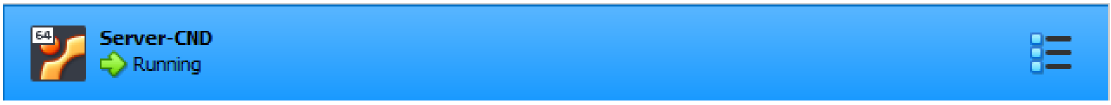
☑ Enable Network Adapter

Attached to: NAT Network

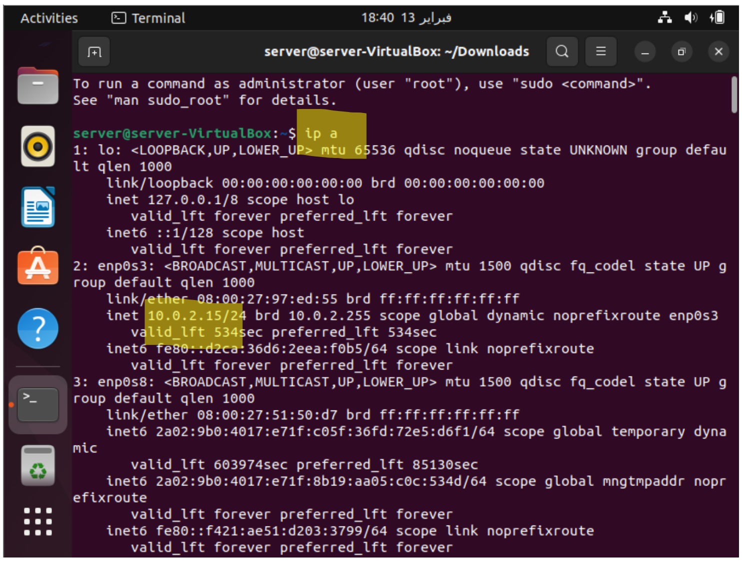
Name: NAT-CND

1.1: Server Setting

After we configure the server in VirtualBox lets lunch it.



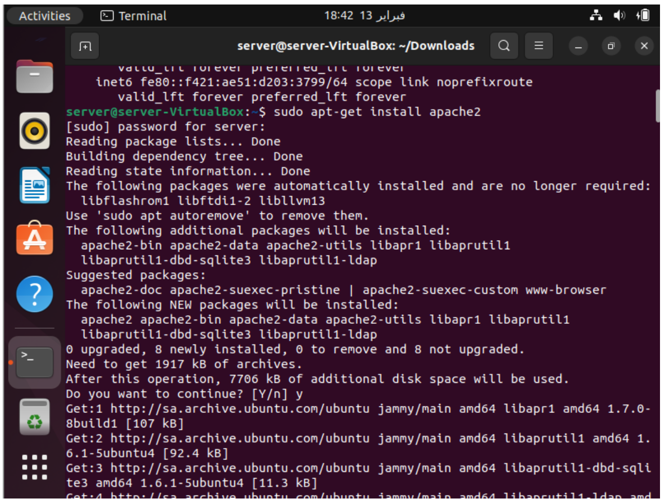
Now let’s check our IP address , → IP a



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

server@server-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:97:ed:55 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 534sec preferred_lft 534sec
    inet6 fe80::d2ca:36d6:2eea:f0b5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:51:50:d7 brd ff:ff:ff:ff:ff:ff
    inet6 2a02:9b0:4017:e71f:c05f:36fd:72e5:d6f1/64 scope global temporary dyna
mic
       valid_lft 603974sec preferred_lft 85130sec
    inet6 2a02:9b0:4017:e71f:8b19:aa05:c0c:534d/64 scope global mngtmpaddr nopr
efixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::f421:ae51:d203:3799/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

As we can see it is 10.0.2.15

The next step is installing Apache2 server. → sudo apt-get install apache2

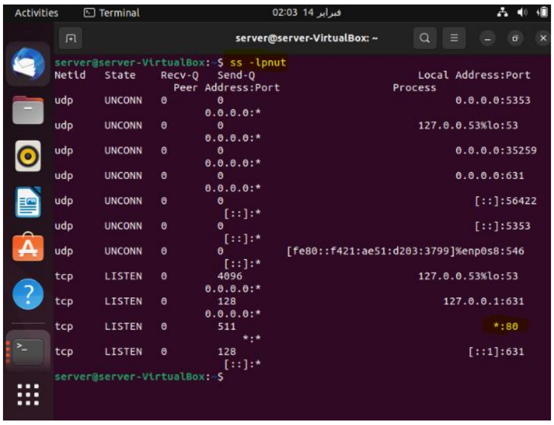


```
       valid_lft forever preferred_lft forever
    inet6 fe80::f421:ae51:d203:3799/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
server@server-VirtualBox:~$ sudo apt-get install apache2
[sudo] password for server:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 8 not upgraded.
Need to get 1917 kB of archives.
After this operation, 7706 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://sa.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1 amd64 1.7.0-
8build1 [107 kB]
Get:2 http://sa.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1 amd64 1.
6.1-5ubuntu4 [92.4 kB]
Get:3 http://sa.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-dbd-sqli
te3 amd64 1.6.1-5ubuntu4 [11.3 kB]
Get:4 http://sa.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-ldap amd
```

1.1: Server Setting

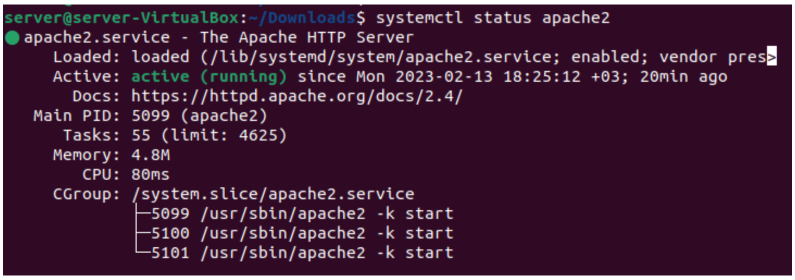
As you can see in the figures below Apache server is running and port 80 is open and listing.

- ➔ **ss -lpnut**



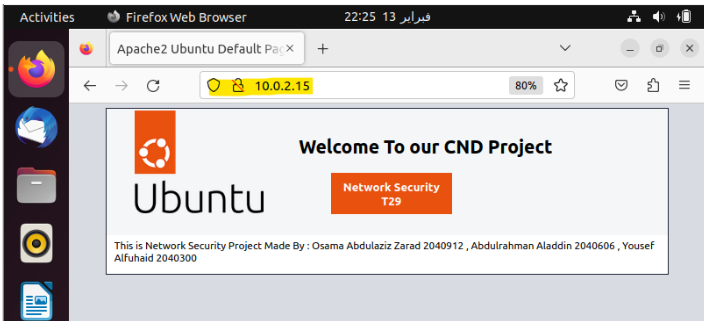
```
server@server-VirtualBox:~$ ss -lpnut
Netid   State    Recv-Q   Send-Q                        Local Address:Port
                    Peer Address:Port                 Process
udp     UNCONN   0        0                              0.0.0.0:5353
                       0.0.0.0:*
udp     UNCONN   0        0                         127.0.0.53%lo:53
                       0.0.0.0:*
udp     UNCONN   0        0                              0.0.0.0:35259
                       0.0.0.0:*
udp     UNCONN   0        0                              0.0.0.0:631
                       0.0.0.0:*
udp     UNCONN   0        0                                 [::]:56422
                          [::]:*
udp     UNCONN   0        0                                 [::]:5353
                          [::]:*
udp     UNCONN   0        0          [fe80::f421:ae51:d203:3799]%enp0s8:546
                          [::]:*
tcp     LISTEN   0        4096                      127.0.0.53%lo:53
                       0.0.0.0:*
tcp     LISTEN   0        128                          127.0.0.1:631
                       0.0.0.0:*
tcp     LISTEN   0        511                                *:80
                             *:*
tcp     LISTEN   0        128                              [::1]:631
                          [::]:*
server@server-VirtualBox:~$
```

- ➔ **systemctl status apache2**



```
server@server-VirtualBox:~/Downloads$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres>
     Active: active (running) since Mon 2023-02-13 18:25:12 +03; 20min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 5099 (apache2)
      Tasks: 55 (limit: 4625)
     Memory: 4.8M
        CPU: 80ms
     CGroup: /system.slice/apache2.service
             ├─5099 /usr/sbin/apache2 -k start
             ├─5100 /usr/sbin/apache2 -k start
             └─5101 /usr/sbin/apache2 -k start
```

web page is working successfully through http://localhost , you can edit the web page using this command → sudo nano /var/www/html/index.html



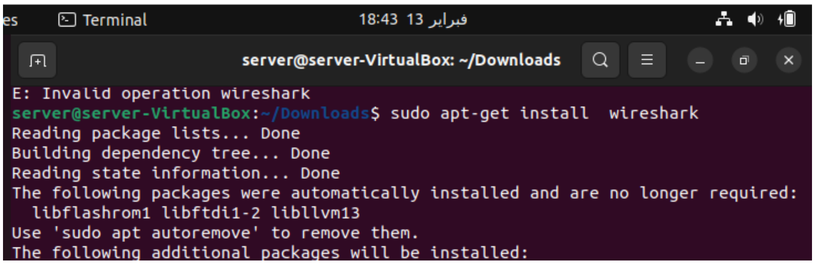
Welcome To our CND Project

Network Security
T29

This is Network Security Project Made By : Osama Abdulaziz Zarad 2040912 , Abdulrahman Aladdin 2040606 , Yousef Alfuhaid 2040300

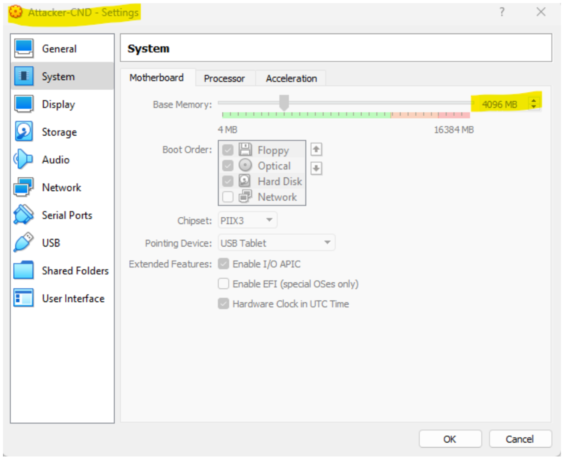
1.1: Server Setting

Now let's download Wireshark in our server. → sudo apt install Wireshark



```
E: Invalid operation wireshark
server@server-VirtualBox:~/Downloads$ sudo apt-get install  wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
```

1.2: Attacker Setting



Attacker-CND - Settings

System

Base Memory: 4096 MB

Boot Order: ☑ Floppy, ☑ Optical, ☑ Hard Disk, □ Network

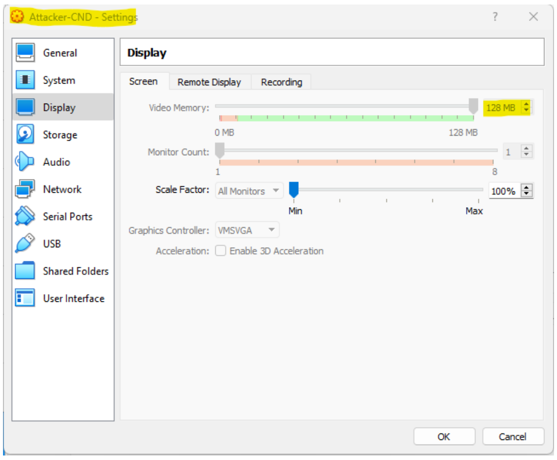
Chipset: PIIX3

Pointing Device: USB Tablet

Extended Features: ☑ Enable I/O APIC

□ Enable EFI (special OSes only)

☑ Hardware Clock in UTC Time



Attacker-CND - Settings

Display

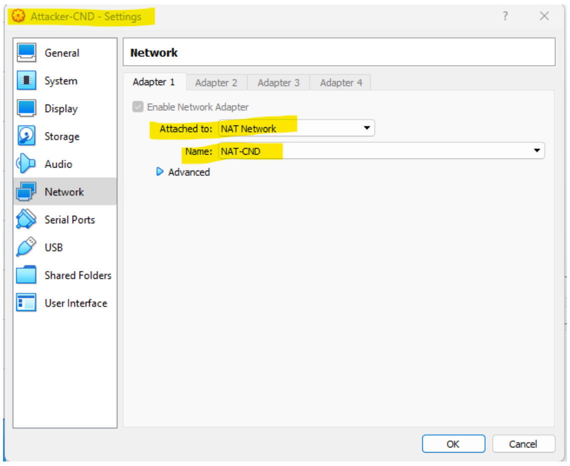
Video Memory: 128 MB

Monitor Count: 1

Scale Factor: All Monitors 100%

Graphics Controller: VMSVGA

Acceleration: □ Enable 3D Acceleration



Attacker-CND - Settings

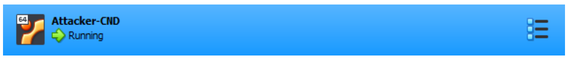
Network

☑ Enable Network Adapter

Attached to: NAT Network

Name: NAT-CND

Now let's lunch the machine.

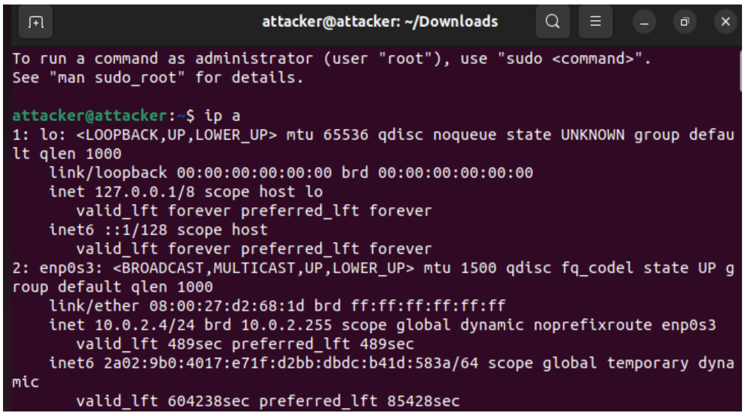


Attacker-CND

Running

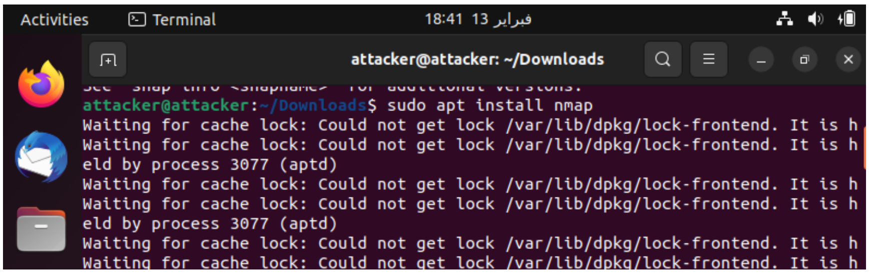
1.2: Attacker Setting

Check what is the IP address which is 10.0.2.4 → ip a



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

attacker@attacker:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:d2:68:1d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 489sec preferred_lft 489sec
    inet6 2a02:9b0:4017:e71f:d2bb:dbdc:b41d:583a/64 scope global temporary dyna
mic
       valid_lft 604238sec preferred_lft 85428sec
```

Now let's download Nmap in the attacker machine. → sudo apt install nmap



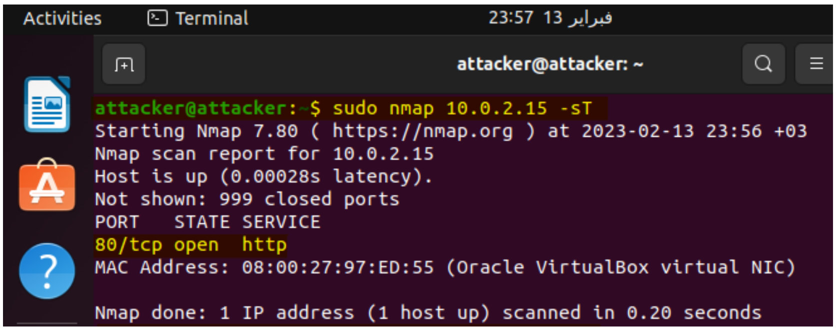
```
See 'snap info <snapname>' for additional versions.
attacker@attacker:~/Downloads$ sudo apt install nmap
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is h
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is h
eld by process 3077 (aptd)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is h
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is h
eld by process 3077 (aptd)
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is h
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-frontend. It is h
```

Part 2: Performing passive attack

2.1 : Perform TCP scan using Nmap :

TCP scan is passive attack used to discover whether a port is open, closed or filtered , Nmap will send TCP connect and implement the concept of a full three-way handshake , SYN , SYN ACK , ACK.

To perform this type of attack → nmap IPaddress -sT

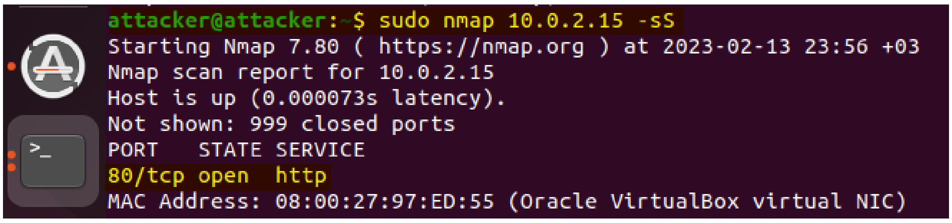


```
attacker@attacker:~$ sudo nmap 10.0.2.15 -sT
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-13 23:56 +03
Nmap scan report for 10.0.2.15
Host is up (0.00028s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:97:ED:55 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

2.2 : Perform stealth "Syn" scan using Nmap :

Stealth or Syn scan is passive attack used to discover whether port is open, closed or filtered, but it is faster than TCP scan because Nmap will send packets but without having a fully TCP three-way handshake ,moreover syn scan not hampered by IDS and firewall and can pass them.

To perform this type of attack → nmap Ipaddress -sS

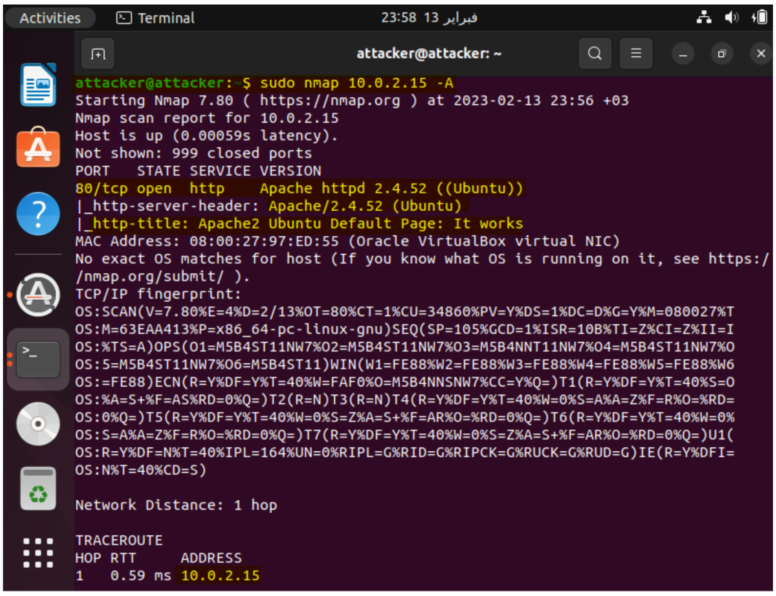


```
attacker@attacker:~$ sudo nmap 10.0.2.15 -sS
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-13 23:56 +03
Nmap scan report for 10.0.2.15
Host is up (0.000073s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:97:ED:55 (Oracle VirtualBox virtual NIC)
```

2.3 : Perform a scan that enables OS detection, version detection, script scanning, and traceroute

To perform this type of scan you have two ways implement each scan individually , option -O for OS detection , -sV for version detection, -sC for script scanning , -packet-trace fore trace route or you can use option -A which will run all the previous options.

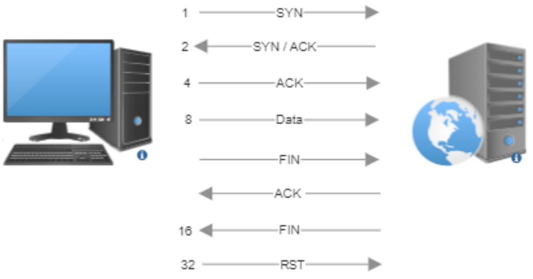
➔ **nmap Ipaddress -A**



```
attacker@attacker:~$ sudo nmap 10.0.2.15 -A
Starting Nmap 7.80 ( https://nmap.org ) at 2023-02-13 23:56 +03
Nmap scan report for 10.0.2.15
Host is up (0.00059s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 08:00:27:97:ED:55 (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https:/
/nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=2/13%OT=80%CT=1%CU=34860%PV=Y%DS=1%DC=D%G=Y%M=080027%T
OS:M=63EAA413%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=10B%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%
OS:S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.59 ms 10.0.2.15
```

Part 3: Analysis the attack using Wireshark

Before we start the analysis, we need to understand the concept of three-way handshake, the healthy connection must has data transfer, to know what if the connection is healthy or not , check the value of conversation completeness if is it more than 39 or not , next figure will explain it more.



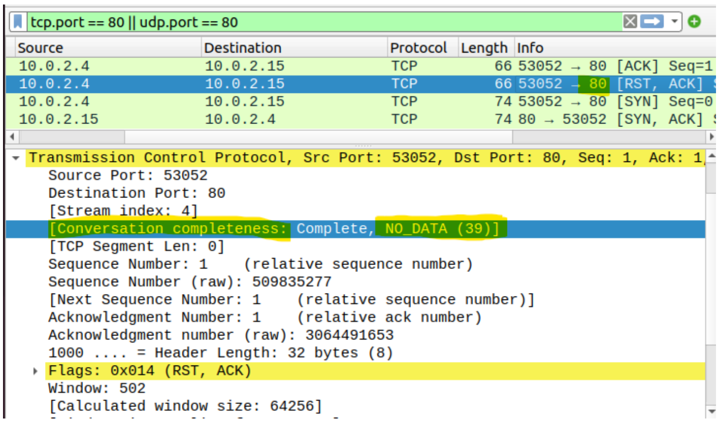
Values of Conversation completeness :

SYN = 1
SYN / ACK = 2
ACK = 4
DATA = 8
FIN =16
RST = 32

As you can see the value of each step , if conversation completeness was equal to 39 that means this connection is not healthy and subspecies because , 1 for SYN , 2 for SYN / ACK , 4 for ACK , 32 for RST combining them equals 39 , that means there was a full hand shake but the connection terminate and there is no data transferred , which tells you that might there is a scan running on your network.

3.1: analysis traffic of TCP scan

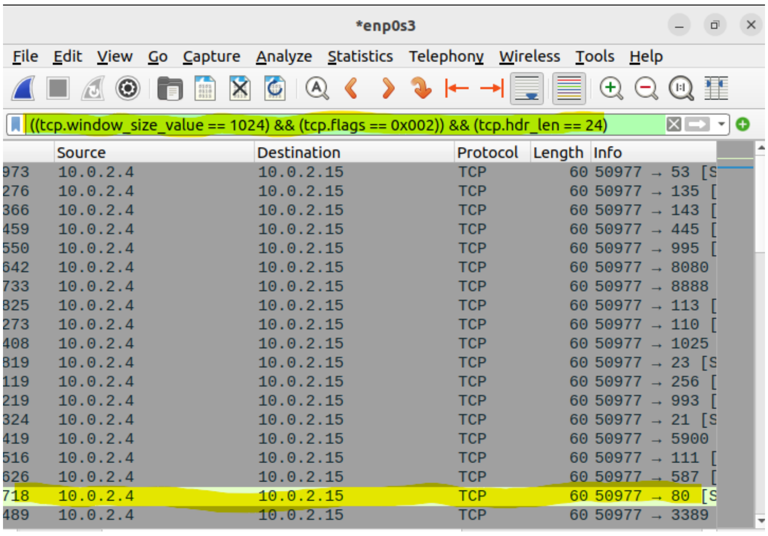
By analysis the PCAP file we can see clearly that there was hundred of packets that the value of conversation completeness equal to 39 and the connection terminate without transferring any data, and as you can see the three-way handshakes are appears which also tell us the type of the scan was TCP scan .



```
tcp.port == 80 || udp.port == 80

Source         Destination    Protocol  Length  Info
10.0.2.4       10.0.2.15      TCP           66 53052 → 80 [ACK] Seq=1
10.0.2.4       10.0.2.15      TCP           66 53052 → 80 [RST, ACK] S
10.0.2.4       10.0.2.15      TCP           74 53052 → 80 [SYN] Seq=0
10.0.2.15      10.0.2.4       TCP           74 80 → 53052 [SYN, ACK] S

Transmission Control Protocol, Src Port: 53052, Dst Port: 80, Seq: 1, Ack: 1
   Source Port: 53052
   Destination Port: 80
   [Stream index: 4]
   [Conversation completeness: Complete, NO_DATA (39)]
   [TCP Segment Len: 0]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 509835277
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 3064491653
   1000 .... = Header Length: 32 bytes (8)
 › Flags: 0x014 (RST, ACK)
   Window: 502
   [Calculated window size: 64256]
```

3.2: analysis traffic of Stealth scan

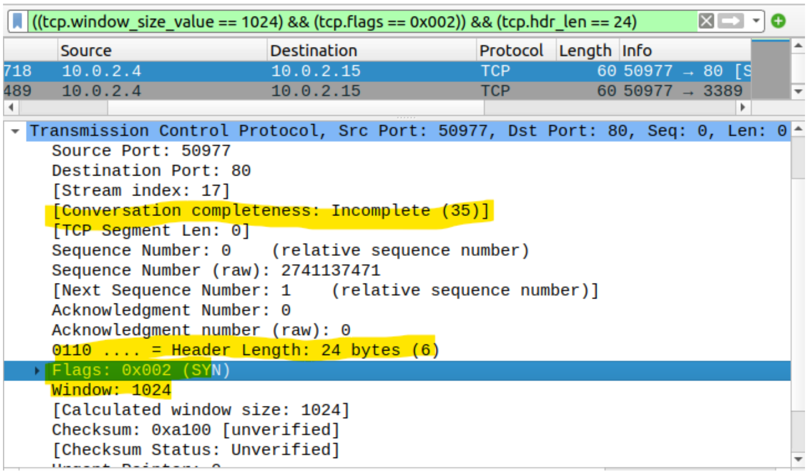
First of all we need to know that TCP header without any option is equal to 20 bytes so if there is any more option TCP header will be bigger, another thing window size usually in healthy connection or full three-way handshake will be much bigger as we see previously , but usually in stealth scan Nmap will generate packets with 1024 windows size , last thing that we can look for it is the size of conversation completeness if it is equal to 35 → 1 for SYN , 2 for SYN/ ACK , 32 for RST .So, let's prepare our filter , we need the size of the window equal to 1024 and flag of syn packet which it's value in hex equal to 0x002 and the TCP header equal to 24.


*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
((tcp.window_size_value == 1024) && (tcp.flags == 0x002)) && (tcp.hdr_len == 24)
```

```
    Source          Destination     Protocol Length Info
973 10.0.2.4        10.0.2.15       TCP          60 50977 → 53 [S
276 10.0.2.4        10.0.2.15       TCP          60 50977 → 135 [
366 10.0.2.4        10.0.2.15       TCP          60 50977 → 143 [
459 10.0.2.4        10.0.2.15       TCP          60 50977 → 445 [
550 10.0.2.4        10.0.2.15       TCP          60 50977 → 995 [
642 10.0.2.4        10.0.2.15       TCP          60 50977 → 8080
733 10.0.2.4        10.0.2.15       TCP          60 50977 → 8888
825 10.0.2.4        10.0.2.15       TCP          60 50977 → 113 [
273 10.0.2.4        10.0.2.15       TCP          60 50977 → 110 [
408 10.0.2.4        10.0.2.15       TCP          60 50977 → 1025
819 10.0.2.4        10.0.2.15       TCP          60 50977 → 23 [S
119 10.0.2.4        10.0.2.15       TCP          60 50977 → 256 [
219 10.0.2.4        10.0.2.15       TCP          60 50977 → 993 [
324 10.0.2.4        10.0.2.15       TCP          60 50977 → 21 [S
419 10.0.2.4        10.0.2.15       TCP          60 50977 → 5900
516 10.0.2.4        10.0.2.15       TCP          60 50977 → 111 [
626 10.0.2.4        10.0.2.15       TCP          60 50977 → 587 [
718 10.0.2.4        10.0.2.15       TCP          60 50977 → 80 [S
489 10.0.2.4        10.0.2.15       TCP          60 50977 → 3389
```

As you can see, we capture the packets successfully

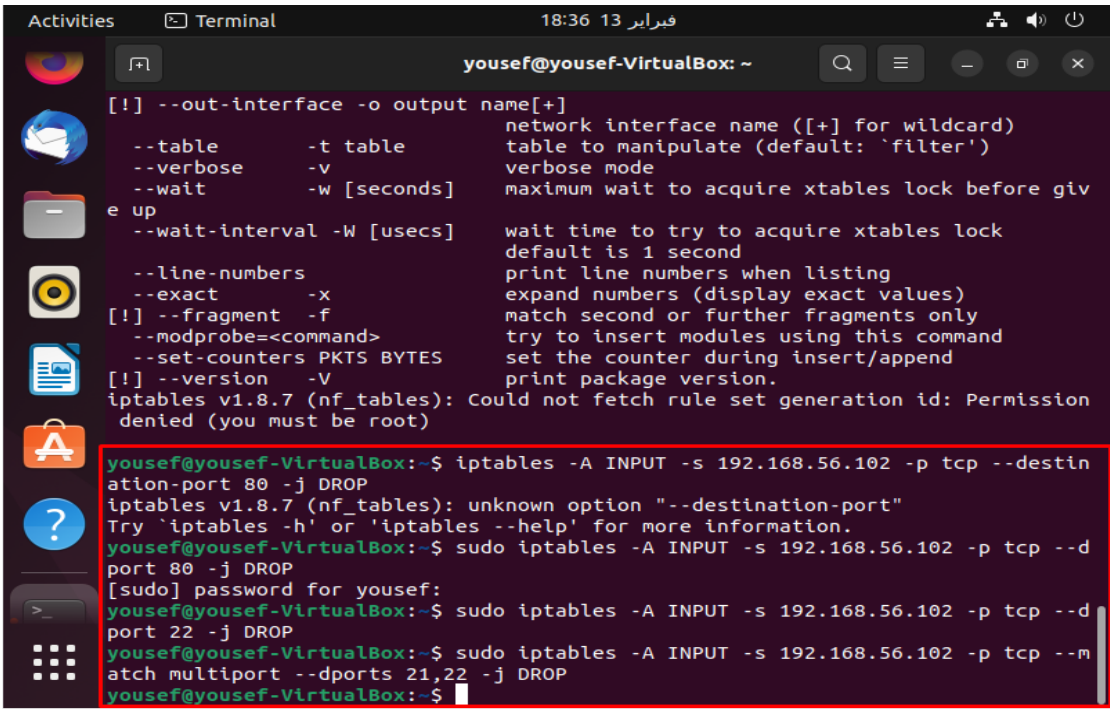


```
((tcp.window_size_value == 1024) && (tcp.flags == 0x002)) && (tcp.hdr_len == 24)
```

```
    Source          Destination     Protocol Length Info
718 10.0.2.4        10.0.2.15       TCP          60 50977 → 80 [S
489 10.0.2.4        10.0.2.15       TCP          60 50977 → 3389
```

```
Transmission Control Protocol, Src Port: 50977, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 50977
  Destination Port: 80
  [Stream index: 17]
  [Conversation completeness: Incomplete (35)]
  [TCP Segment Len: 0]
  Sequence Number: 0    (relative sequence number)
  Sequence Number (raw): 2741137471
  [Next Sequence Number: 1    (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  0110 .... = Header Length: 24 bytes (6)
▸ Flags: 0x002 (SYN)
  Window: 1024
  [Calculated window size: 1024]
  Checksum: 0xa100 [unverified]
  [Checksum Status: Unverified]
```

Part 4: Firewall



```
[!] --out-interface -o output name[+]
                                 network interface name ([+] for wildcard)
  --table       -t table         table to manipulate (default: `filter')
  --verbose     -v               verbose mode
  --wait        -w [seconds]     maximum wait to acquire xtables lock before giv
e up
  --wait-interval -W [usecs]     wait time to try to acquire xtables lock
                                 default is 1 second
  --line-numbers                 print line numbers when listing
  --exact       -x               expand numbers (display exact values)
[!] --fragment  -f               match second or further fragments only
  --modprobe=<command>           try to insert modules using this command
  --set-counters PKTS BYTES      set the counter during insert/append
[!] --version   -V               print package version.
iptables v1.8.7 (nf_tables): Could not fetch rule set generation id: Permission
 denied (you must be root)
yousef@yousef-VirtualBox:~$ iptables -A INPUT -s 192.168.56.102 -p tcp --destin
ation-port 80 -j DROP
iptables v1.8.7 (nf_tables): unknown option "--destination-port"
Try `iptables -h' or 'iptables --help' for more information.
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --d
port 80 -j DROP
[sudo] password for yousef:
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --d
port 22 -j DROP
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --m
atch multiport --dports 21,22 -j DROP
yousef@yousef-VirtualBox:~$
```

The first command: iptables -A INPUT -s 192.168.56.102 -p tcp --dport 80 -j DROP

adds a new rule to the INPUT chain in Iptables to block incoming traffic from the IP address

The options used in the command are as follows:

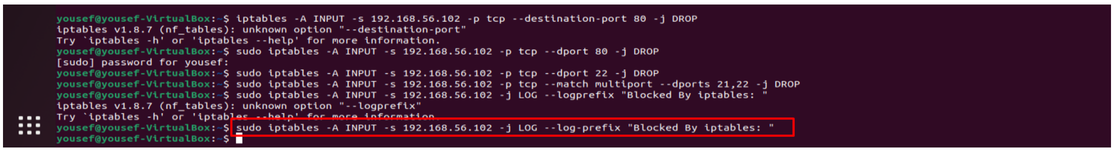
-A: This option adds a new rule to the end of the INPUT chain.

-s: This option specifies the source IP address

-p: This option specifies the protocol

--dport: This option specifies the destination port

-j: This option specifies the target of the rule



```
yousef@yousef-VirtualBox:~$ iptables -A INPUT -s 192.168.56.102 -p tcp --destination-port 80 -j DROP
iptables v1.8.7 (nf_tables): unknown option "--destination-port"
Try `iptables -h' or 'iptables --help' for more information.
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --dport 80 -j DROP
[sudo] password for yousef:
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --dport 22 -j DROP
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --match multiport --dports 21,22 -j DROP
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -j LOG --logprefix "Blocked By iptables: "
iptables v1.8.7 (nf_tables): unknown option "--logprefix"
Try `iptables -h' or 'iptables --help' for more information.
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -j LOG --log-prefix "Blocked By iptables: "
yousef@yousef-VirtualBox:~$
```

The command iptables -A INPUT -s 192.168.56.102 -j LOG --log-prefix "Blocked by iptables: "

adds a new rule to the INPUT chain in Iptables to log incoming traffic from the IP address specified as 192.168.56.102

The options used in the command are as follows:

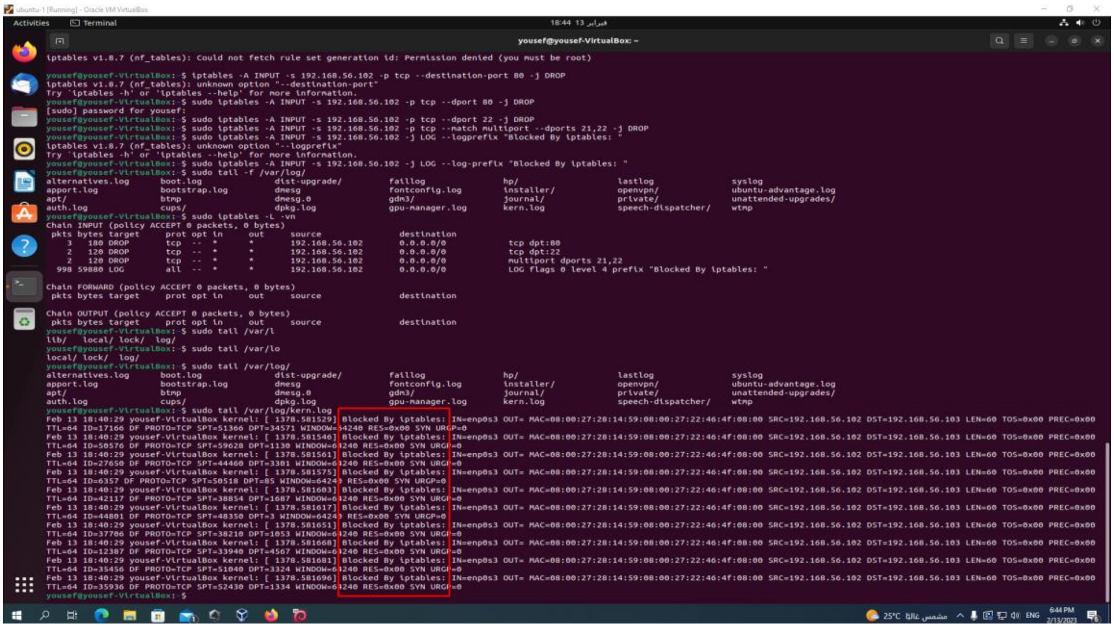
-A: This option adds a new rule to the end of the INPUT chain.

-s: This option specifies the source IP address, in this case, the IP address of the attacker.

-j: This option specifies the target of the rule, in this case, the LOG target, which will log all incoming traffic that matches the rule.

--log-prefix: This option specifies a prefix for the log messages generated by this rule.

The prefix "Blocked by iptables: " will be added to the beginning of each log message generated by this rule.

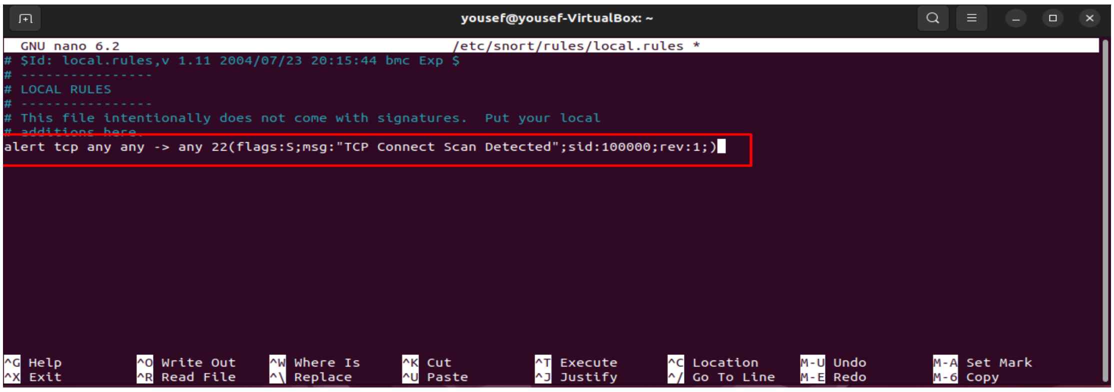


```
iptables v1.8.7 (nf_tables): Could not fetch rule set generation id: Permission denied (you must be root)

yousef@yousef-VirtualBox:~$ iptables -A INPUT -s 192.168.56.102 -p tcp --destination-port 80 -j DROP
iptables v1.8.7 (nf_tables): unknown option "--destination-port"
Try `iptables -h' or 'iptables --help' for more information.
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --dport 80 -j DROP
[sudo] password for yousef:
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --dport 22 -j DROP
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -p tcp --match multiport --dports 21,22 -j DROP
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -j LOG --logprefix "Blocked By iptables: "
iptables v1.8.7 (nf_tables): unknown option "--logprefix"
Try `iptables -h' or 'iptables --help' for more information.
yousef@yousef-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.56.102 -j LOG --log-prefix "Blocked By iptables: "
yousef@yousef-VirtualBox:~$ sudo tail -f /var/log/
alternatives.log      boot.log            dist-upgrade/       faillog             hp/                 lastlog             syslog
apport.log            bootstrap.log       dmesg               fontconfig.log      installer/          openvpn/            ubuntu-advantage.log
apt/                  btmp                dmesg.0             gdm3/               journal/            private/            unattended-upgrades/
auth.log              cups/               dpkg.log            gpu-manager.log     kern.log            speech-dispatcher/  wtmp
yousef@yousef-VirtualBox:~$ sudo iptables -L -vn
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    3   180 DROP       tcp  --  *      *       192.168.56.102       0.0.0.0/0            tcp dpt:80
    2   120 DROP       tcp  --  *      *       192.168.56.102       0.0.0.0/0            tcp dpt:22
    2   120 DROP       tcp  --  *      *       192.168.56.102       0.0.0.0/0            multiport dports 21,22
  998 59880 LOG        all  --  *      *       192.168.56.102       0.0.0.0/0            LOG flags 0 level 4 prefix "Blocked By iptables: "

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
yousef@yousef-VirtualBox:~$ sudo tail /var/l
lib/   local/ lock/  log/
yousef@yousef-VirtualBox:~$ sudo tail /var/lo
local/ lock/  log/
yousef@yousef-VirtualBox:~$ sudo tail /var/log/
alternatives.log      boot.log            dist-upgrade/       faillog             hp/                 lastlog             syslog
apport.log            bootstrap.log       dmesg               fontconfig.log      installer/          openvpn/            ubuntu-advantage.log
apt/                  btmp                dmesg.0             gdm3/               journal/            private/            unattended-upgrades/
auth.log              cups/               dpkg.log            gpu-manager.log     kern.log            speech-dispatcher/  wtmp
yousef@yousef-VirtualBox:~$ sudo tail /var/log/kern.log
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581529] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=17166 DF PROTO=TCP SPT=51366 DPT=34571 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581546] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=50576 DF PROTO=TCP SPT=59628 DPT=1130 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581561] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=27650 DF PROTO=TCP SPT=44460 DPT=3301 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581575] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=6357 DF PROTO=TCP SPT=50518 DPT=85 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581603] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=42117 DF PROTO=TCP SPT=38854 DPT=1687 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581617] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=44801 DF PROTO=TCP SPT=48350 DPT=3 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581651] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=37706 DF PROTO=TCP SPT=38210 DPT=1053 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581668] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=12387 DF PROTO=TCP SPT=33940 DPT=4567 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581681] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=35456 DF PROTO=TCP SPT=51040 DPT=3324 WINDOW=64240 RES=0x00 SYN URGP=0
Feb 13 18:40:29 yousef-VirtualBox kernel: [ 1378.581696] Blocked By iptables: IN=enp0s3 OUT= MAC=08:00:27:28:14:59:08:00:27:22:46:4f:08:00 SRC=192.168.56.102 DST=192.168.56.103 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=35936 DF PROTO=TCP SPT=52430 DPT=1334 WINDOW=64240 RES=0x00 SYN URGP=0
yousef@yousef-VirtualBox:~$
```

ubuntu-1 [Running] - Oracle VM VirtualBox
Activities
Terminal
18:44 13 فبراير
yousef@yousef-VirtualBox: ~
25°C مشمس غالبًا
ENG
6:44 PM
2/13/2023

iptables detection from the log

Part 5: Firewall



```
  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 22(flags:S;msg:"TCP Connect Scan Detected";sid:100000;rev:1;)








^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy
```

The options used in the rule are as follows:

tcp: This specifies the protocol type, in this case, TCP.

any: This specifies the source and destination IP addresses, in this case, any IP address.

any: This specifies the source and destination ports, in this case, any port.

->: This specifies the direction of the traffic, in this case, incoming traffic.

22: This specifies the destination port, in this case, port 22 (SSH).

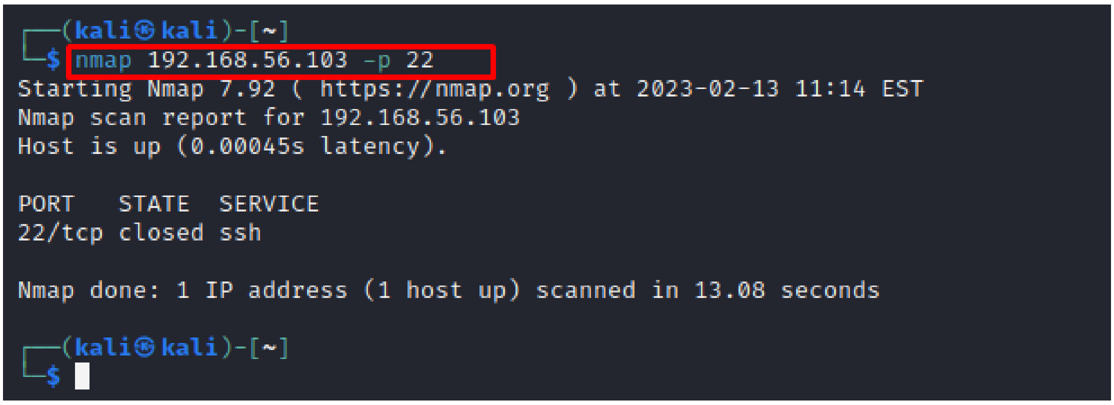
flags:S: This option specifies the TCP flag that is used to trigger the alert, in this case, the SYN (Synchronize) flag.

msg: This option specifies the message that will be displayed when the alert is triggered, in this case, "NMAP TCP Scan".

sid: This option specifies a unique identifier for the rule, in this case, 10000.

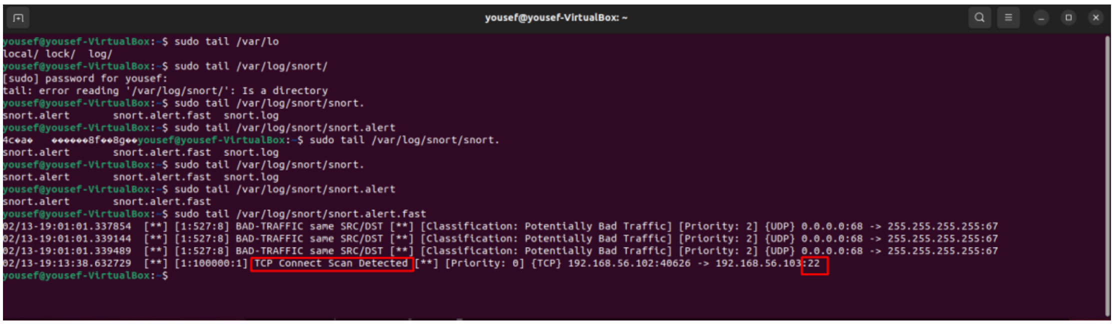
rev: This option specifies the revision number for the rule, in this case, 1.

This rule is designed to detect and alert on a TCP connect scan that is initiated from any IP address and directed to port 22 (SSH)



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.56.103 -p 22
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-13 11:14 EST
Nmap scan report for 192.168.56.103
Host is up (0.00045s latency).

PORT   STATE  SERVICE
22/tcp closed ssh

Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Here is the attacker trying to scan the port 22 of the server

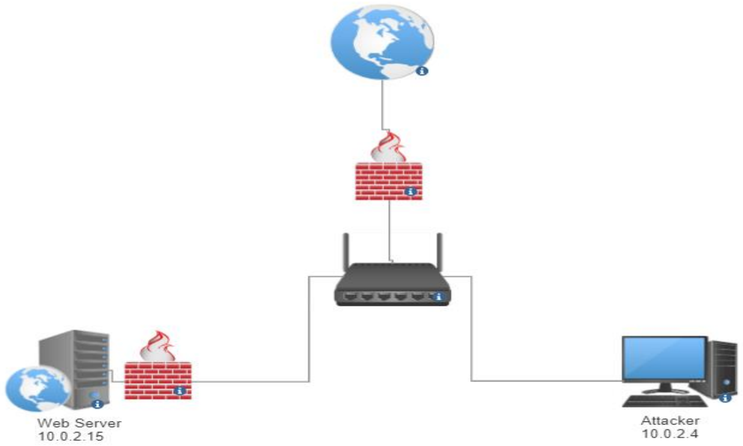


```
yousef@yousef-VirtualBox:~$ sudo tail /var/lo
local/ lock/  log/
yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/
[sudo] password for yousef:
tail: error reading '/var/log/snort/': Is a directory
yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/snort.
snort.alert       snort.alert.fast  snort.log
yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/snort.alert
4c◆a◆   ◆◆◆◆◆◆8f◆◆8g◆◆yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/snort.
snort.alert       snort.alert.fast  snort.log
yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/snort.
snort.alert       snort.alert.fast  snort.log
yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/snort.alert
snort.alert       snort.alert.fast
yousef@yousef-VirtualBox:~$ sudo tail /var/log/snort/snort.alert.fast
02/13-19:01:01.337854  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
02/13-19:01:01.339144  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
02/13-19:01:01.339489  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
02/13-19:13:38.632729  [**] [1:100000:1] TCP Connect Scan Detected [**] [Priority: 0] {TCP} 192.168.56.102:40626 -> 192.168.56.103:22
yousef@yousef-VirtualBox:~$
```

And lastly here is the IDS detection “snort” after the attacker tries to scan port 22

Part 6 : report

Network setup of the project (a diagram with labels and IP addresses) :



List of software or tools used in the project and their configurations. :

Attacker Machine	Server Machine
Nmap	Apache server
	Wireshark
	Snort
	Iptables

Step-by-step project description (screenshots are required) :

shown above

Conclusion :

After we finish the tasks, we have the ability to detect, protect, and attack the network.

At part 2 task 1, we are the attacker; we scan the TCP connection and stealthily scan

At part 3, we install the Wireshark tool to use for capturing the TCP scan and stealth scan.

Its job is to provide monitoring to discover and then take action.

At part 4, we configure the firewall to block the HTTP, SSH, and FTP connections from the attacker.

At part 5, we install and configure SNORT on the server VM to alert on TCP connect scan on port 22 from the attacker VM.