

University ID : N17385760

Net ID : iv447

LAB3

1. NMAP

1. Using nmap, find all the open ports and OS on each host in the 10.10.111.0/24 network. List the command that is used. [10 points]

STEPS :

Power Up Internal Router, External Router and Kali virtual Machine.

Type the following command on kali's command terminal :

```
sudo nmap -O 10.10.111.0/24
```

This will find all the open ports and OS on each host in the 10.10.111.0/24 network.

```
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!
student@kali:~$ sudo nmap -O 10.10.111.0/24
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-27 22:19 EDT
Nmap scan report for 10.10.111.1
Host is up (0.00043s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
MAC Address: 00:00:00:00:00:02 (Xerox)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 1 hop

Nmap scan report for 10.10.111.2
Host is up (0.00037s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
MAC Address: 00:00:00:00:00:01 (Xerox)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 1 hop

Nmap scan report for 10.10.111.102
Host is up (0.0012s latency).
All 1000 scanned ports on 10.10.111.102 are filtered
```

```
Nmap scan report for 10.10.111.2
Host is up (0.00037s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
MAC Address: 00:00:00:00:00:01 (Xerox)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 1 hop

Nmap scan report for 10.10.111.102
Host is up (0.0012s latency).
All 1000 scanned ports on 10.10.111.102 are filtered
MAC Address: 00:00:00:00:00:06 (Xerox)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

Nmap scan report for 10.10.111.100
Host is up (0.000047s latency).
All 1000 scanned ports on 10.10.111.100 are closed
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (4 hosts up) scanned in 10.91 seconds
student@kali:~$
```

2. Using nmap, find all the open ports and OS on each host in the 10.20.111.0/24 network. List the command that is used. [10 points]

STEPS :

1. Power Up Internal Router, External Router and Kali virtual Machine.
2. Type the following command on kali's command terminal :
sudo nmap -O 10.20.111.0/24

This will find all the open ports and OS on each host in the 10.20.111.0/24 network.

Connected (unencrypted) to: QEMU (344_13_22)

Applications ▾ Places ▾ Terminal ▾ Tue 22:22 student@kali:~

```
File Edit View Search Terminal Help
student@kali:~$ sudo nmap -o 10.20.111.0/24
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-27 22:22 EDT
Nmap scan report for 10.20.111.1
Host is up (0.00071s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 2 hops

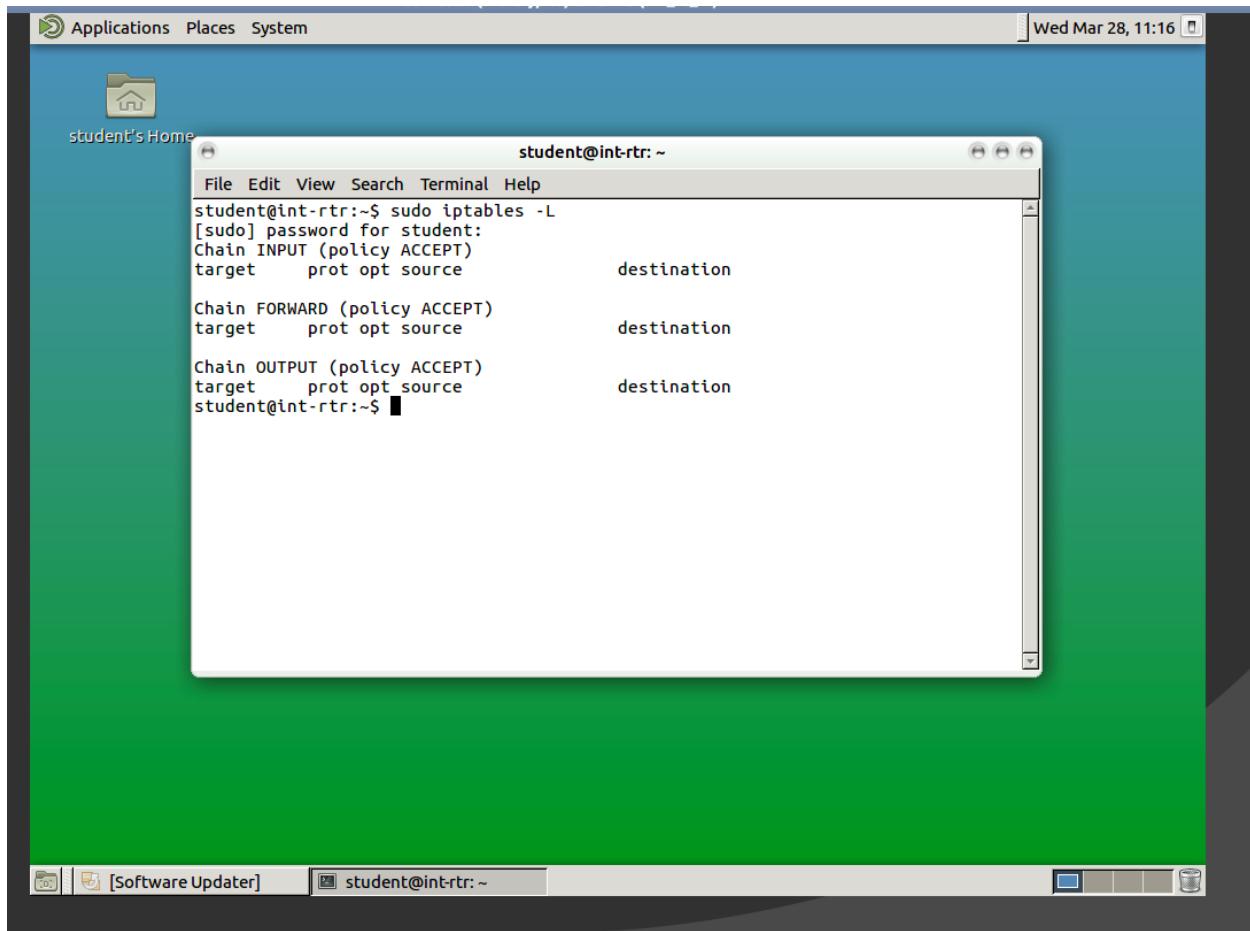
Nmap scan report for 10.20.111.2
Host is up (0.00094s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 3 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 38.06 seconds
student@kali:~$
```

2. IPTABLES

Iptables is a user space utility program that allows system administrator to configure the tables and apply firewall rules for incoming and outgoing traffic.

When the default rule is ACCEPT on INPUT, FORWARD and OUTPUT chains on Internal Router Iptables . External network can communicate with internal network and internal network can communicate with external network without any restriction.



For Eg. Kali Machine (on network 10.10.111.0/24) can perform following activities :

1. Ping from kali to Int-linux 10.20.111.2 :

Command : sudo ping 10.20.111.2

Connected (unencrypted) to: QEMU (344_13_22)

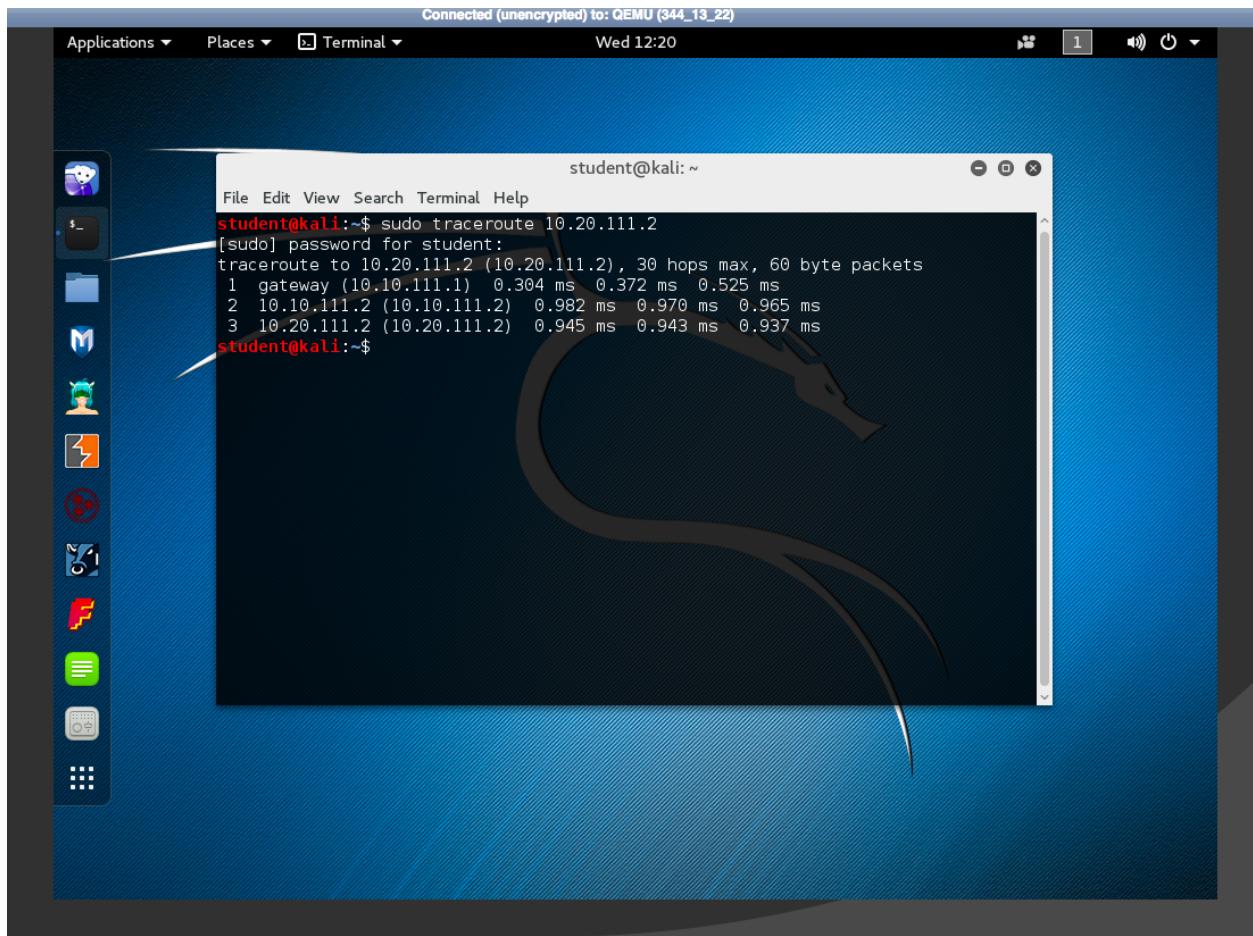
Applications ▾ Places ▾ Terminal ▾ Wed 11:27

student@kali:~

```
File Edit View Search Terminal Help
student@kali:~$ sudo ping 10.20.111.2
[sudo] password for student:
PING 10.20.111.2 (10.20.111.2) 56(84) bytes of data.
64 bytes from 10.20.111.2: icmp_seq=1 ttl=62 time=0.06 ms
64 bytes from 10.20.111.2: icmp_seq=2 ttl=62 time=0.01 ms
64 bytes from 10.20.111.2: icmp_seq=3 ttl=62 time=0.955 ms
64 bytes from 10.20.111.2: icmp_seq=4 ttl=62 time=0.898 ms
64 bytes from 10.20.111.2: icmp_seq=5 ttl=62 time=0.885 ms
64 bytes from 10.20.111.2: icmp_seq=6 ttl=62 time=0.940 ms
64 bytes from 10.20.111.2: icmp_seq=7 ttl=62 time=0.874 ms
64 bytes from 10.20.111.2: icmp_seq=8 ttl=62 time=0.971 ms
64 bytes from 10.20.111.2: icmp_seq=9 ttl=62 time=0.991 ms
64 bytes from 10.20.111.2: icmp_seq=10 ttl=62 time=0.881 ms
64 bytes from 10.20.111.2: icmp_seq=11 ttl=62 time=0.909 ms
64 bytes from 10.20.111.2: icmp_seq=12 ttl=62 time=0.914 ms
64 bytes from 10.20.111.2: icmp_seq=13 ttl=62 time=0.946 ms
64 bytes from 10.20.111.2: icmp_seq=14 ttl=62 time=0.870 ms
64 bytes from 10.20.111.2: icmp_seq=15 ttl=62 time=0.899 ms
64 bytes from 10.20.111.2: icmp_seq=16 ttl=62 time=0.818 ms
64 bytes from 10.20.111.2: icmp_seq=17 ttl=62 time=0.933 ms
64 bytes from 10.20.111.2: icmp_seq=18 ttl=62 time=0.828 ms
64 bytes from 10.20.111.2: icmp_seq=19 ttl=62 time=0.728 ms
64 bytes from 10.20.111.2: icmp_seq=20 ttl=62 time=0.962 ms
64 bytes from 10.20.111.2: icmp_seq=21 ttl=62 time=0.730 ms
```

2. Traceroute to 10.20.111.2 from kali:

Command : sudo traceroute 10.20.111.2



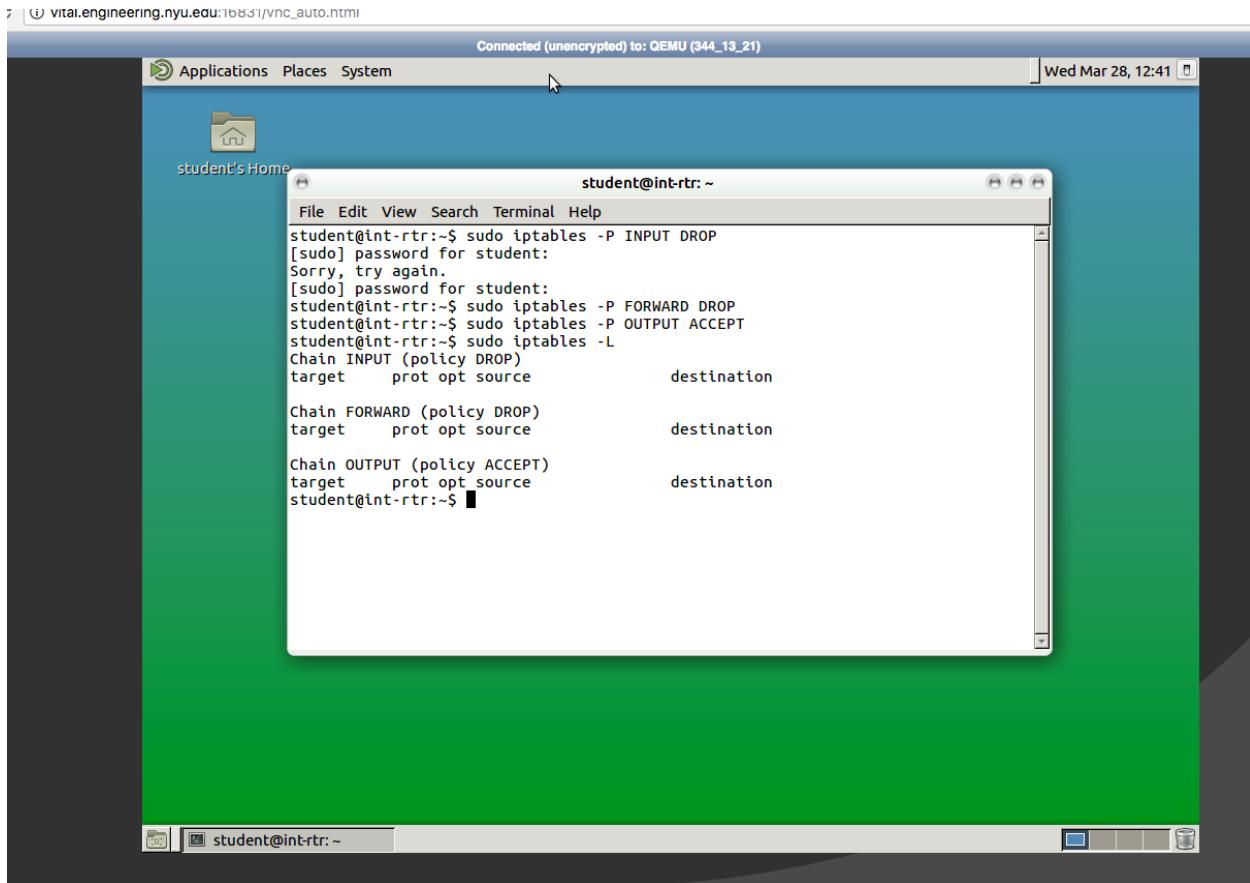
Now we can configure the firewall on Internal Router for following rules implementation :

- A) For outgoing traffic (from 10.20.111.0/24 to 10.10.111.0/24) - your internal machine should be able to communicate with the external network and the external machines without restrictions. [10 pts]
- B) For incoming traffic (from the 10.10.111.0/24 to the 10.20.111.0/24) - all incoming connection requests should be rejected with the following exceptions: 1) The internal machine (10.20.111.2) should respond to a ping from 10.10.111.0/24 [10 pts] 2) The internal machine (10.20.111.2) should block all incoming SSH and http requests from 10.10.111.0/24 [10 pts]

Type following rules for Iptables on Int-Router :

```
sudo iptables -P INPUT DROP
```

```
sudo iptables -P FORWARD DROP  
sudo iptables -P OUTPUT ACCEPT  
sudo iptables -L
```



These will be default rules for our firewall.

For outgoing traffic (from 10.20.111.0/24 to 10.10.111.0/24) We can add following rules to FORWARD chain so that internal machine is able to communicate with the external network and the external machines without restrictions :

```
sudo iptables -P INPUT DROP  
sudo iptables -P FORWARD DROP  
sudo iptables -P OUTPUT ACCEPT  
sudo iptables -L
```

```
sudo iptables -A FORWARD -s 10.20.111.0/24 -d 10.10.111.0/24 -i eth1 -o eth0 -j  
ACCEPT  
sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -i eth0 -j  
ACCEPT  
sudo iptables -L
```

The screenshot shows a terminal window titled "Connected (unencrypted) to: QEMU (344_13_21)". The window has a dark theme and displays the following command-line session:

```
student@int-rtr:~$ sudo iptables -P INPUT DROP  
[sudo] password for student:  
Sorry, try again.  
[sudo] password for student:  
student@int-rtr:~$ sudo iptables -P FORWARD DROP  
student@int-rtr:~$ sudo iptables -P OUTPUT ACCEPT  
student@int-rtr:~$ sudo iptables -L  
Chain INPUT (policy DROP)  
target prot opt source destination  
  
Chain FORWARD (policy DROP)  
target prot opt source destination  
  
Chain OUTPUT (policy ACCEPT)  
target prot opt source destination  
student@int-rtr:~$ sudo iptables -A FORWARD -s 10.20.111.0/24 -d 10.10.111.0/24  
-i eth1 -o eth0 -j ACCEPT  
student@int-rtr:~$ sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,R  
ELATED -i eth0 -j ACCEPT  
student@int-rtr:~$ sudo iptables -L  
Chain INPUT (policy DROP)  
target prot opt source destination  
  
Chain FORWARD (policy DROP)  
target prot opt source destination  
ACCEPT all -- 10.20.111.0/24 10.10.111.0/24  
ACCEPT all -- anywhere anywhere ctstate RELATED,ES  
TABLISHED  
  
Chain OUTPUT (policy ACCEPT)  
target prot opt source destination  
student@int-rtr:~$
```

Checking for traffic from int-linux machine :

Connected (unencrypted) to: QEMU (344_13_25)

Applications Places System student@int-linux: ~ Wed Mar 28, 12:53

File Edit View Search Terminal Help

```
student@int-linux:~$ sudo iptables -L
[sudo] password for student:
Chain INPUT (policy ACCEPT)
target     prot opt source          destination

Chain FORWARD (policy ACCEPT)
target     prot opt source          destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
student@int-linux:~$ sudo nmap 10.10.111.1

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 12:45 EDT
Nmap scan report for 10.10.111.1
Host is up (0.00083s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain

Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
student@int-linux:~$ sudo ssh 10.10.111.1
The authenticity of host '10.10.111.1 (10.10.111.1)' can't be established.
ECDSA key fingerprint is SHA256:cAq+RABxe0+4P7ibrg/xGbmpDYUkpo7NL9MVbRiXCUY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.111.1' (ECDSA) to the list of known hosts.
root@10.10.111.1's password:
```

Now for blocking all incoming traffic to 10.20.111.0/24 from external network except for ping, we can add following rules to table :

```
sudo iptables -A FORWARD -p icmp -s 10.10.111.0/24 -d 10.20.111.0/24 -i eth0 -j ACCEPT
sudo iptables -L
```

Connected (unencrypted) to: QEMU (344_13_21)

student@int-rtr: ~

```
File Edit View Search Terminal Help
Chain INPUT (policy DROP)
target    prot opt source          destination
Chain FORWARD (policy DROP)
target    prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
student@int-rtr:~$ sudo iptables -A FORWARD -s 10.20.111.0/24 -d 10.10.111.0/24
-i eth1 -o eth0 -j ACCEPT
student@int-rtr:~$ sudo iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,R
ELATED -i eth0 -j ACCEPT
student@int-rtr:~$ sudo iptables -L
Chain INPUT (policy DROP)
target    prot opt source          destination
Chain FORWARD (policy DROP)
target    prot opt source          destination
ACCEPT   all  --  10.20.111.0/24      10.10.111.0/24
ACCEPT   all  --  anywhere        anywhere        ctstate RELATED,ES
TABLISHED

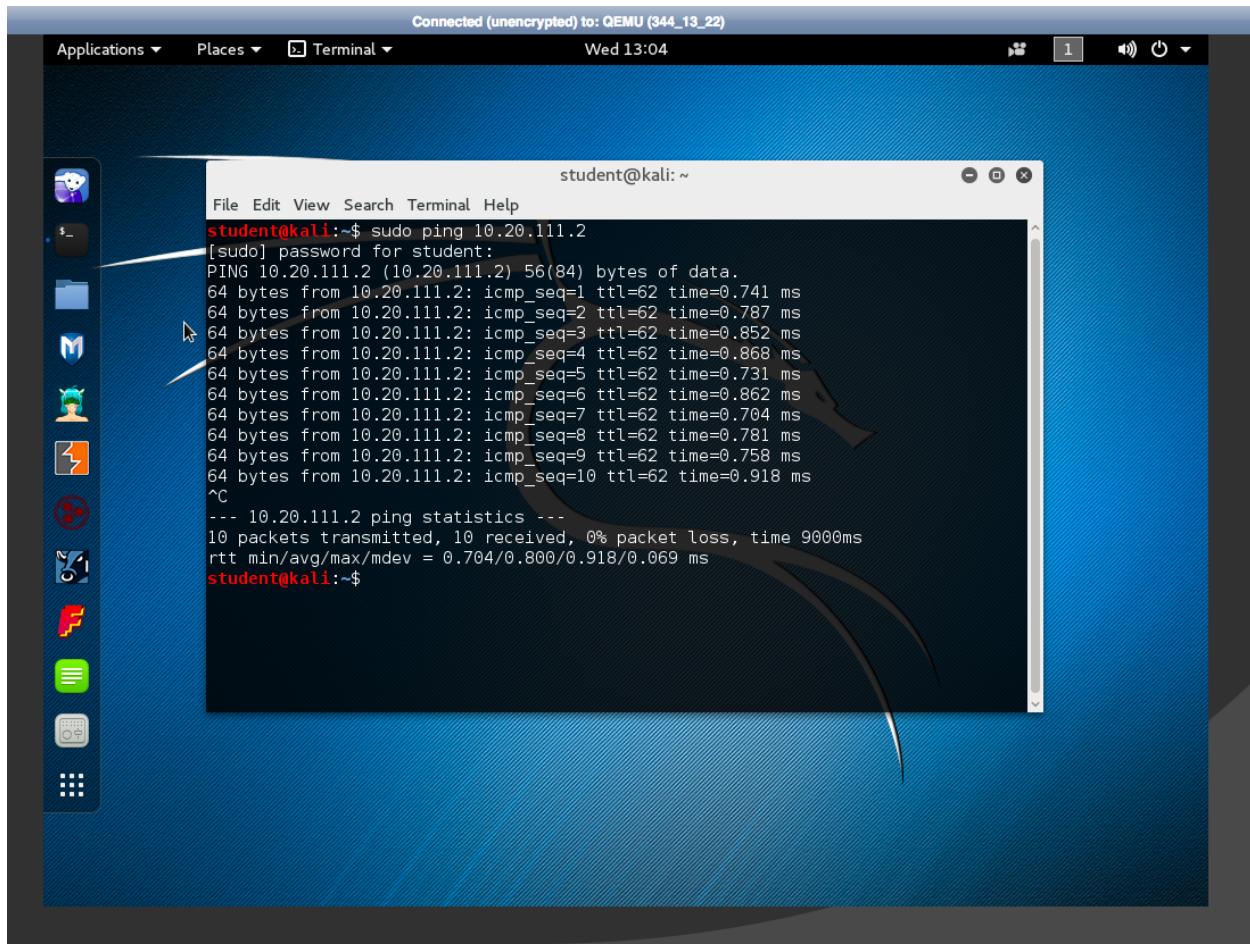
Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
student@int-rtr:~$ sudo iptables -A FORWARD -p icmp -s 10.10.111.0/24 -d 10.20.111.0/24 -i eth0 -j ACCEPT
student@int-rtr:~$ sudo iptables -L
Chain INPUT (policy DROP)
target    prot opt source          destination
Chain FORWARD (policy DROP)
target    prot opt source          destination
ACCEPT   all  --  10.20.111.0/24      10.10.111.0/24
ACCEPT   all  --  anywhere        anywhere        ctstate RELATED,ESTABLISHED
ACCEPT   icmp --  10.10.111.0/24     10.20.111.0/24

Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
student@int-rtr:~$
```

Ping from 10.10.111.0/24 network :

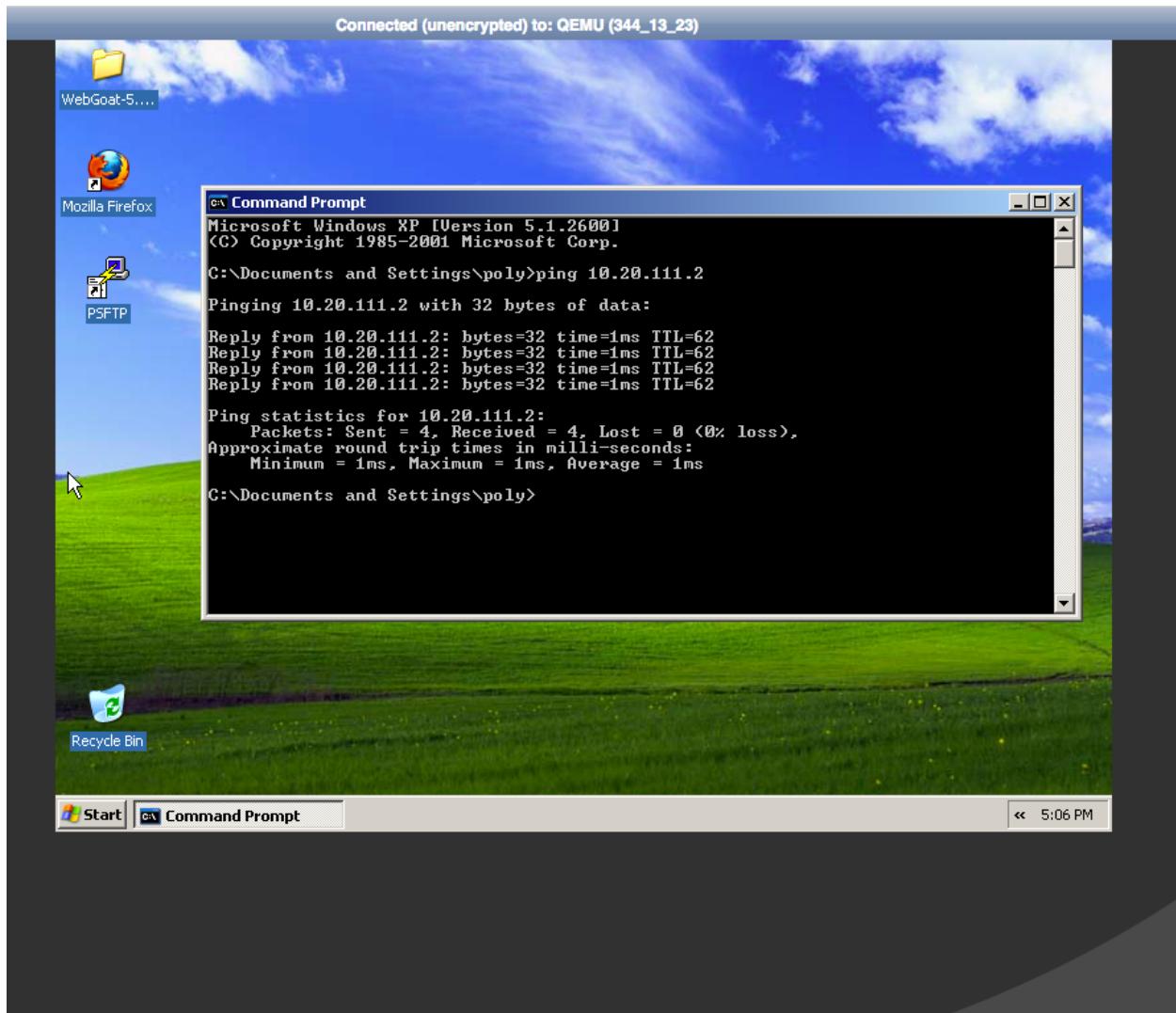
Ping from kali machine :

Command : sudo ping 10.20.111.2



ping from windows XP :

Command : ping 10.20.111.2



Ping from 10.20.111.2 to external network :

Command : sudo ping 10.10.111.1

Connected (unencrypted) to: QEMU (344_13_25)

student@int-linux: ~

```
File Edit View Search Terminal Help
student@int-linux:~$ sudo ping 10.10.111.1
PING 10.10.111.1 (10.10.111.1) 56(84) bytes of data.
64 bytes from 10.10.111.1: icmp_seq=1 ttl=63 time=0.434 ms
64 bytes from 10.10.111.1: icmp_seq=2 ttl=63 time=0.572 ms
64 bytes from 10.10.111.1: icmp_seq=3 ttl=63 time=0.600 ms
64 bytes from 10.10.111.1: icmp_seq=4 ttl=63 time=0.505 ms
64 bytes from 10.10.111.1: icmp_seq=5 ttl=63 time=0.484 ms
64 bytes from 10.10.111.1: icmp_seq=6 ttl=63 time=0.473 ms
^C
--- 10.10.111.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4997ms
rtt min/avg/max/mdev = 0.434/0.511/0.600/0.060 ms
student@int-linux:~$
```

The internal machine (10.20.111.2) will block all incoming SSH and http requests from 10.10.111.0/24, since we followed default drop rule for INPUT and FORWARD chains. Only pings will be allowed from external network.

Traceroute from external network for 10.20.111.2 :

Command : sudo traceroute 10.20.111.2

Connected (unencrypted) to: QEMU (344_13_22)

Applications ▾ Places ▾ Terminal ▾ Wed 13:13 student@kali: ~

File Edit View Search Terminal Help

```
^C
--- 10.20.111.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9000ms
rtt min/avg/max/mdev = 0.704/0.800/0.918/0.069 ms
[student@kali:~$ clear
student@kali:~$ sudo traceroute 10.20.111.2
traceroute to 10.20.111.2 (10.20.111.2), 30 hops max, 60 byte packets
 1 gateway (10.10.111.1)  0.369 ms  0.353 ms  0.336 ms
 2 10.10.111.2 (10.10.111.2)  0.598 ms  0.576 ms  0.570 ms
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *

student@kali:~$
```

Ssh and nmap from kali :

Command :

```
sudo ssh 10.20.111.2
sudo nmap 10.20.111.2
```

```
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
student@kali:~$ sudo ssh 10.20.111.2
^C
student@kali:~$ sudo nmap 10.20.111.2

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 13:16 EDT
Nmap scan report for 10.20.111.2
Host is up (0.00083s latency).
All 1000 scanned ports on 10.20.111.2 are filtered

Nmap done: 1 IP address (1 host up) scanned in 21.21 seconds
student@kali:~$
```

3. NMAP & IPTABLES:

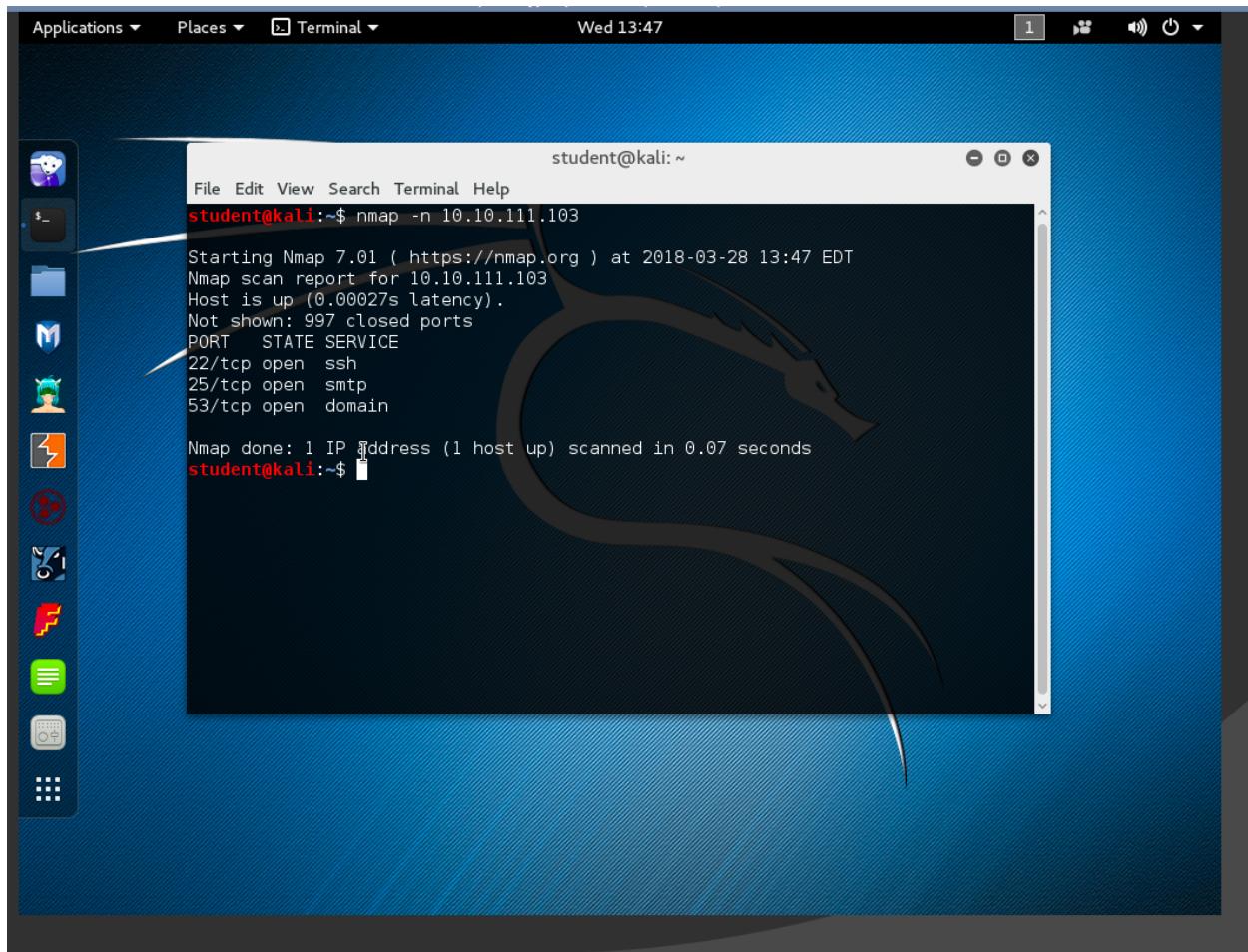
1) Following are the options you will find yourself often needing when using nmap. Use each of these options to perform a scan on the Ubuntu VM using Kali as an attacker machine (see Lab 0 for the login information for the Ubuntu machine). Submit a quick one-liner beside each to explain what each does and screenshots of each scan that you performed. (10 points)

We will first flush all the rules for part 3 that we applied on internal router.

a) -n

Command : nmap -n 10.10.111.103 / nmap 10.10.111.103 -n

Description : If we use -n with nmap it does not perform DNS resolution.



b) -P0

Command : nmap -p0 10.10.111.103

Description : Leaving off end port in range, makes the scan go through to port 65535.

The screenshot shows a Kali Linux desktop environment with a blue wavy background. A terminal window titled "student@kali: ~" is open, displaying the output of the Nmap command. The terminal window has a standard Xfce-style title bar with icons for minimize, maximize, and close. The terminal itself has a dark background with white text. The output of the command is as follows:

```
Connected (unencrypted) to: QEMU (344_13_22)
Applications ▾ Places ▾ Terminal ▾ Wed 13:54
[1] 1354 Terminal

student@kali:~$ nmap -p0 10.10.111.103
Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 13:54 EDT
Nmap scan report for 10.10.111.103
Host is up (0.00033s latency).
PORT      STATE SERVICE
0/tcp     closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
student@kali:~$
```

c) -O

Command : sudo nmap -O 10.10.111.103

Description : Above command performs remote OS detection using TCP/IP stack fingerprinting.

Connected (unencrypted) to: QEMU (344_13_22)

Applications ▾ Places ▾ Terminal ▾ Wed 14:01

student@kali:~

```
File Edit View Search Terminal Help
student@kali:~$ sudo nmap -o 10.10.111.103
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 14:01 EDT
Nmap scan report for 10.10.111.103
Host is up (0.00023s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
MAC Address: 00:00:00:00:00:05 (Xerox)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.0
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.83 seconds
student@kali:~$
```

d) -v

Command : nmap -v 10.10.111.103

Description : Increase verbosity level, description will be in more details.

Connected (unencrypted) to: QEMU (344_13_22)

Applications ▾ Places ▾ Terminal ▾

Wed 14:05

student@kali: ~

File Edit View Search Terminal Help

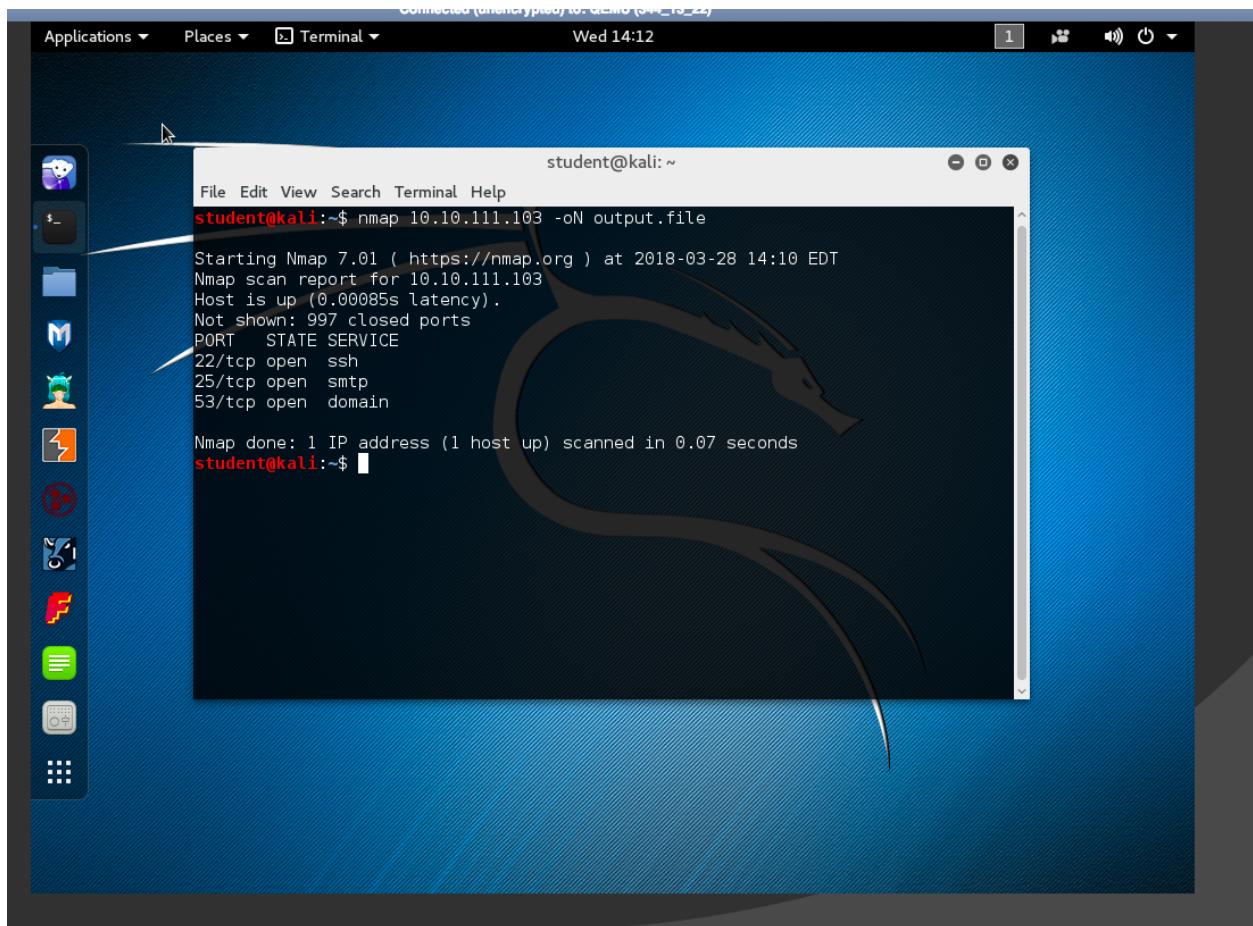
```
student@kali:~$ nmap -v 10.10.111.103
Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 14:05 EDT
Initiating Ping Scan at 14:05
Scanning 10.10.111.103 [2 ports]
Completed Ping Scan at 14:05, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:05
Completed Parallel DNS resolution of 1 host. at 14:05, 0.00s elapsed
Initiating Connect Scan at 14:05
Scanning 10.10.111.103 [1000 ports]
Discovered open port 22/tcp on 10.10.111.103
Discovered open port 25/tcp on 10.10.111.103
Discovered open port 53/tcp on 10.10.111.103
Completed Connect Scan at 14:05, 0.01s elapsed (1000 total ports)
Nmap scan report for 10.10.111.103
Host is up (0.00029s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
student@kali:~$
```

e) -oN

Command : nmap 10.10.111.103 -oN output.file

Description : Normal output to output.file

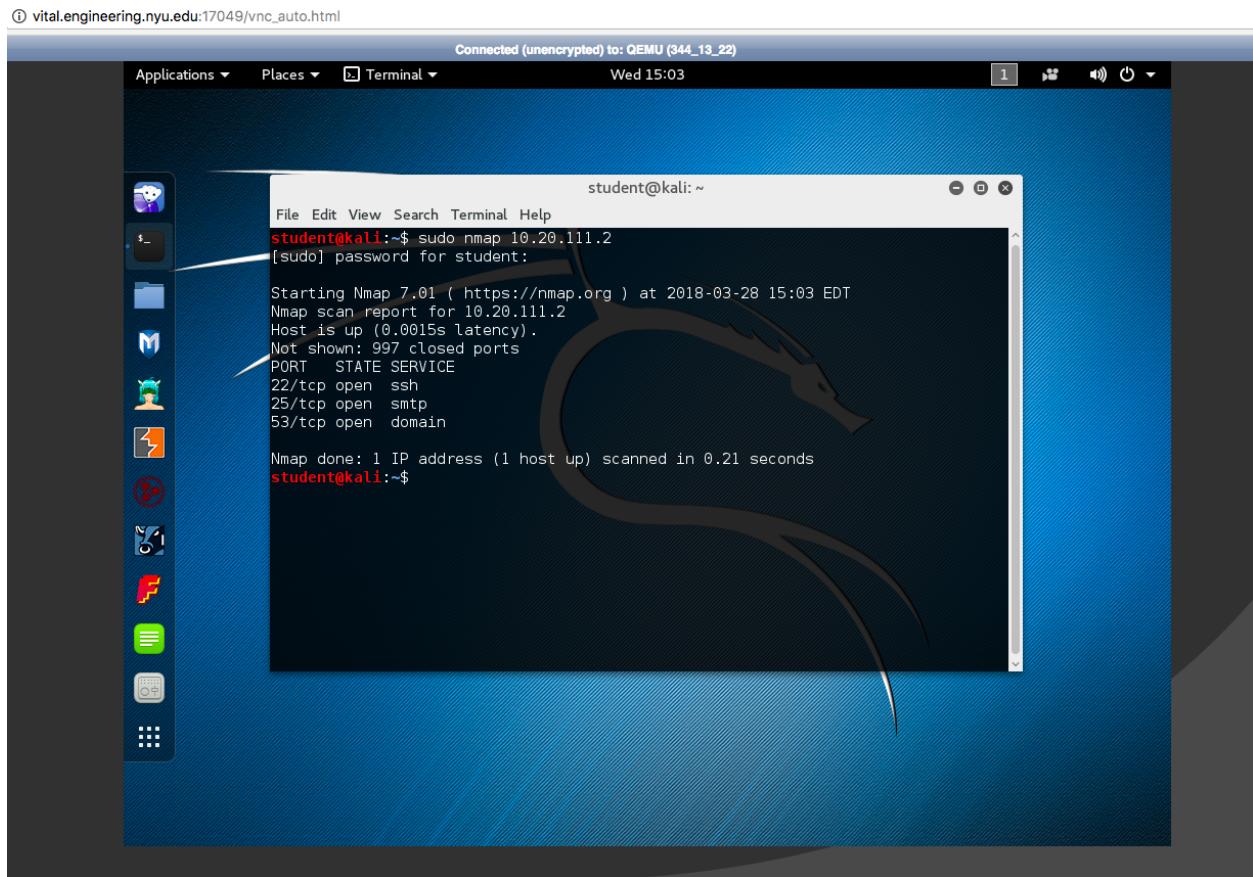


- 2) Using Kali as the scanning machine, perform an nmap scan on the Int-Linux VM (see Lab 0 for the login information for the Int-Linux machine). Include screenshots of the scan results in your report.
- Did the Int-Linux VM respond to nmap's probes? If yes, write firewall rules to stop it. This involves blocking incoming ICMP packets, and ports 443 and 80. If you write firewall rules, make sure to verify that they are installed correctly. (10 points)
 - Now that you have implemented the appropriate rules on the Int-Linux VM, execute nmap from Kali. Submit screenshots of your nmap command and results of your scan. (10 points)
 - There is a method of forcing nmap to scan hosts even if the initial nmap probes are blocked. Leaving the iptables in place that block nmap's initial probe requests, run nmap with a set of options that scans IntLinux even when it doesn't reply to nmap's initial probe requests. Include the nmap options you used and a screenshot of the scan. (10 points)

Nmap of int-linux from kali :

Command : sudo nmap 10.20.111.2

Output :



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "student@kali: ~". The terminal content shows the following Nmap command and its output:

```
student@kali:~$ sudo nmap 10.20.111.2
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 15:03 EDT
Nmap scan report for 10.20.111.2
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
student@kali:~$
```

We add following rules to iptable of int-linux machine to perform above mentioned tasks:

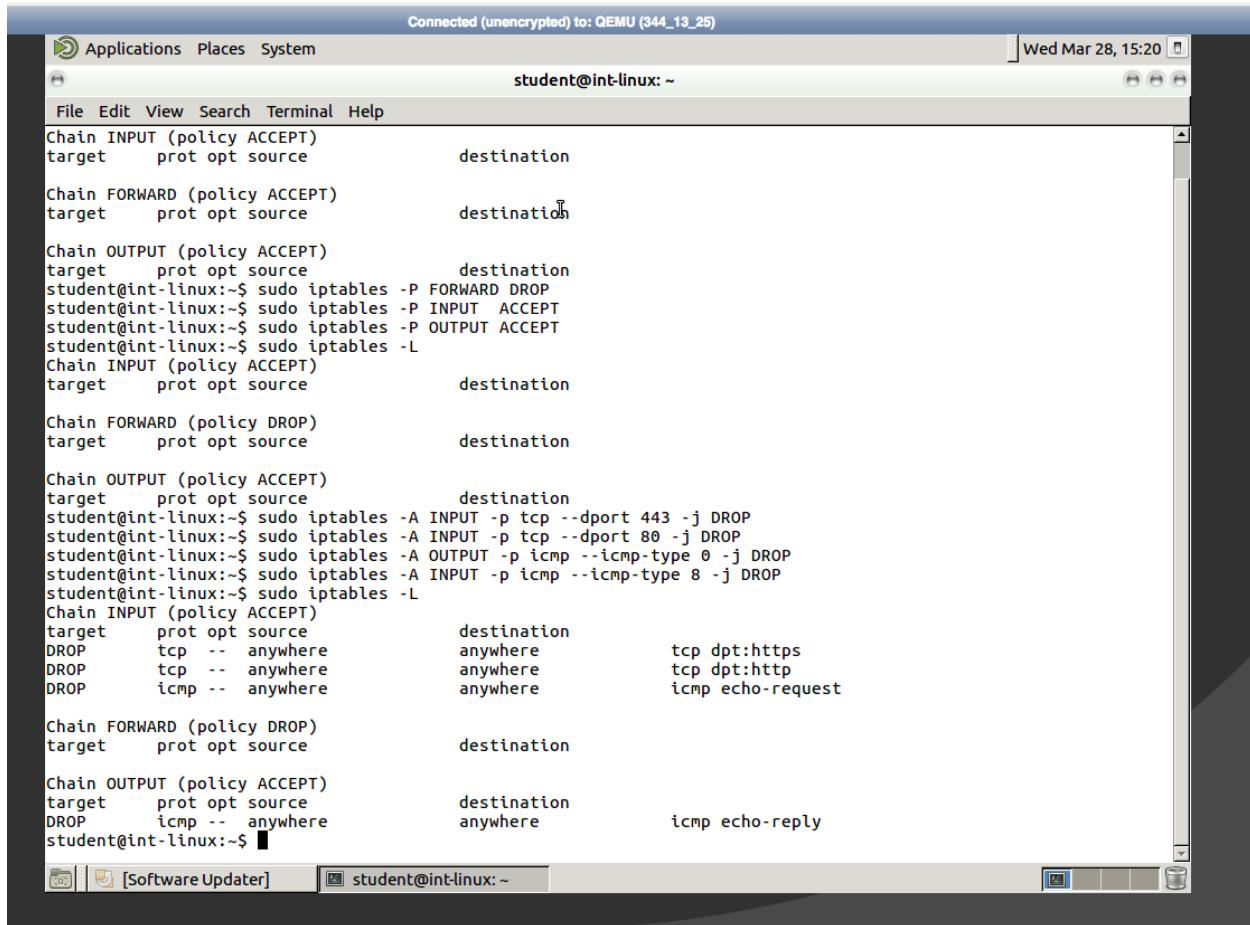
Set the following default rules to iptables :

```
sudo iptables -P FORWARD DROP
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -L
```

Rules to block icmp from external network and block port 443 and 80 on int-linux machine :

```
sudo iptables -A INPUT -p tcp --dport 443 -j DROP  
sudo iptables -A INPUT -p tcp --dport 80 -j DROP  
sudo iptables -A OUTPUT -p icmp --icmp-type 0 -j DROP  
sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP  
sudo iptables -L
```

Iptable on int-linux after above rules are added to the table :

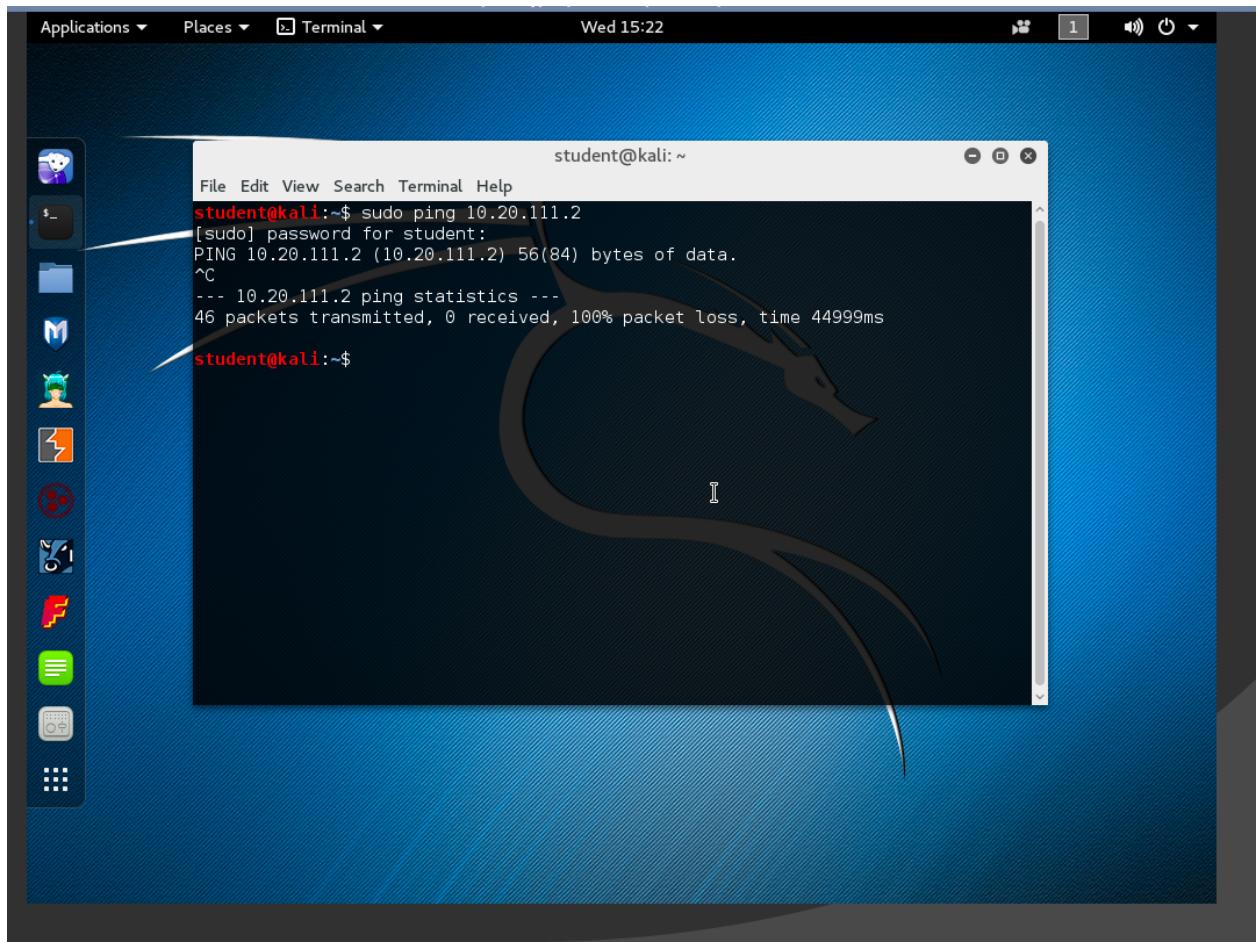


The screenshot shows a terminal window titled "student@int-linux: ~" running on a Kali Linux desktop environment. The window displays the results of the command "sudo iptables -L". The output lists several chains and their rules:

- Chain INPUT (policy ACCEPT)**
 - target prot opt source destination
- Chain FORWARD (policy ACCEPT)**
 - target prot opt source destination
- Chain OUTPUT (policy ACCEPT)**
 - target prot opt source destination
- student@int-linux:~\$ sudo iptables -P FORWARD DROP
- student@int-linux:~\$ sudo iptables -P INPUT ACCEPT
- student@int-linux:~\$ sudo iptables -P OUTPUT ACCEPT
- student@int-linux:~\$ sudo iptables -L
- Chain INPUT (policy ACCEPT)**
 - target prot opt source destination
- Chain FORWARD (policy DROP)**
 - target prot opt source destination
- Chain OUTPUT (policy ACCEPT)**
 - target prot opt source destination
- student@int-linux:~\$ sudo iptables -A INPUT -p tcp --dport 443 -j DROP
- student@int-linux:~\$ sudo iptables -A INPUT -p tcp --dport 80 -j DROP
- student@int-linux:~\$ sudo iptables -A OUTPUT -p icmp --icmp-type 0 -j DROP
- student@int-linux:~\$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP
- student@int-linux:~\$ sudo iptables -L
- Chain INPUT (policy ACCEPT)**
 - target prot opt source destination
 - DROP tcp -- anywhere anywhere tcp dpt:https
 - DROP tcp -- anywhere anywhere tcp dpt:http
 - DROP icmp -- anywhere anywhere icmp echo-request
- Chain FORWARD (policy DROP)**
 - target prot opt source destination
- Chain OUTPUT (policy ACCEPT)**
 - target prot opt source destination
 - DROP icmp -- anywhere anywhere icmp echo-reply

ICMP ping from kali to 10.20.111.2 :

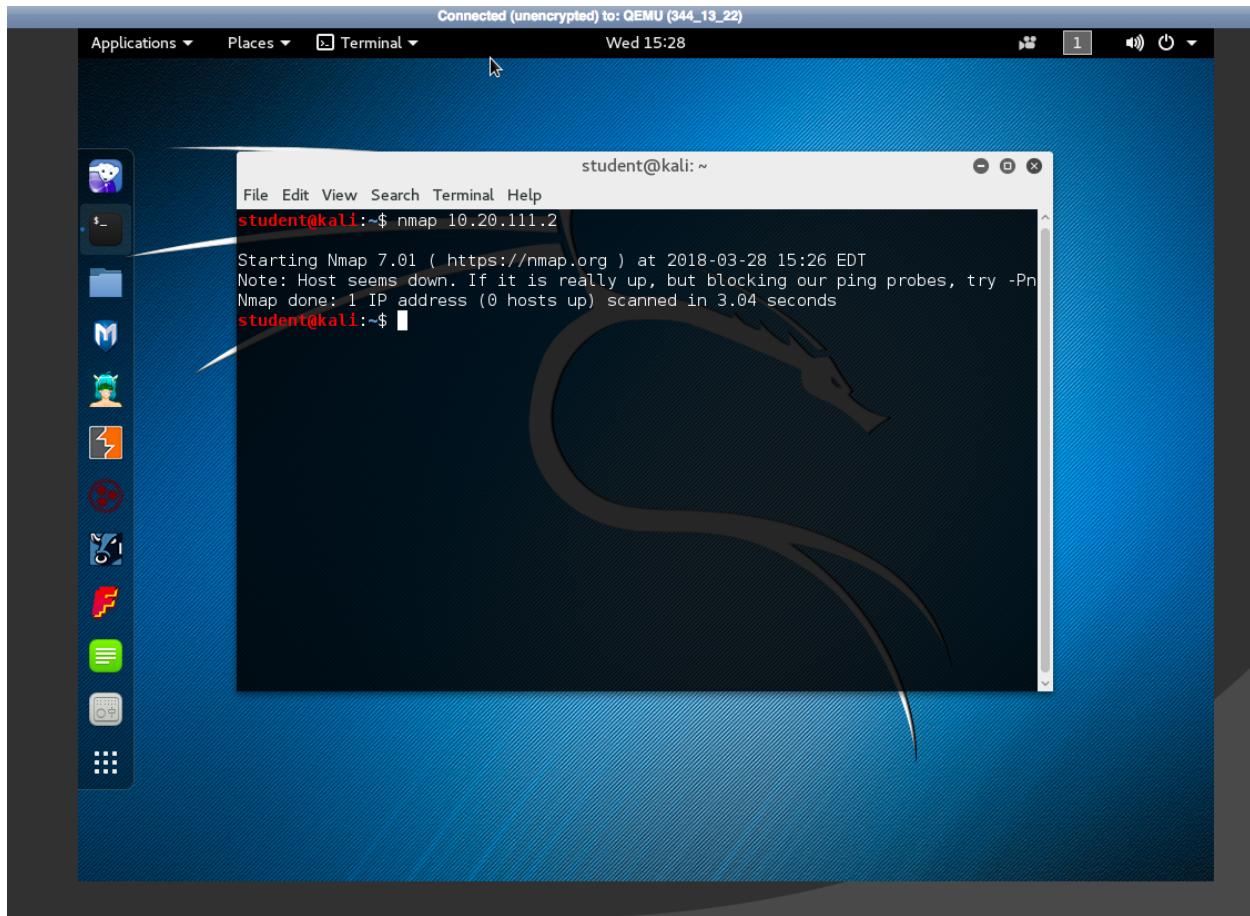
Command : sudo ping 10.20.111.2



nmap after applying above rule :

Command : nmap 10.20.111.2

After applying above rules int-linux will block nmap probes.

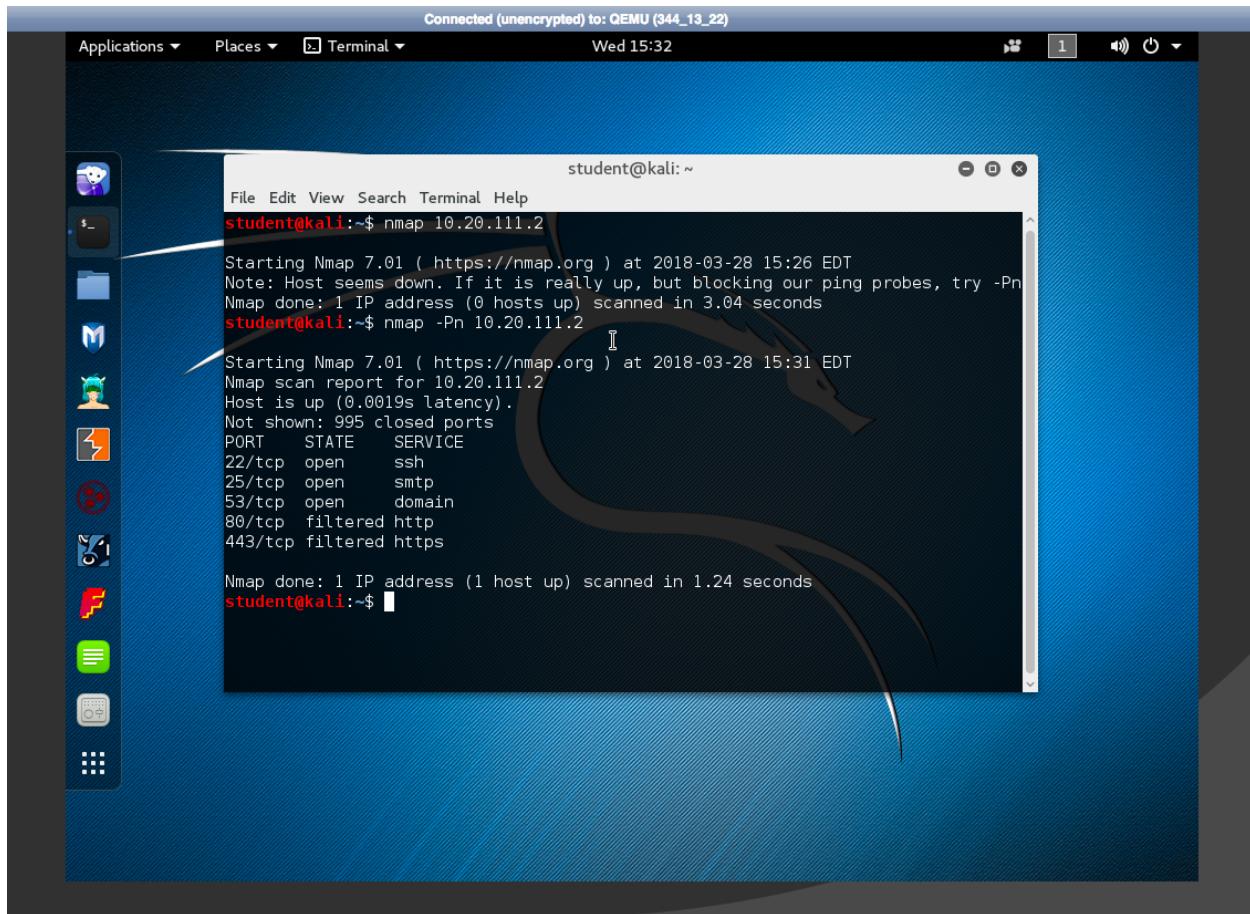


There is a method of forcing nmap to scan hosts even if the initial nmap probes are blocked. Leaving the iptables in place that block nmap's initial probe requests, run nmap with a set of options that scans IntLinux even when it doesn't reply to nmap's initial probe requests.

Following command can be used for this :

Command :

```
nmap -Pn 10.20.111.2
```



3) Using the Kali machine as an attacker machine, perform a nmap TCP SYN scan on the Metasploitable VM (see Lab 0 for the login information for the Metasploitable machine). Then construct an iptable rule to block all incoming TCP SYN packets only from the Kali scanning server's IP address. Explain the trade offs of blocking all TCP SYN packets from an IP address. Submit screenshots of your TCP SYN scans before and after applying the iptable rule. (10 points)

For this part we will apply the rules on metasploitable VM :

Perform nmap probe from kali on metasploitable VM :

Command : sudo nmap -sS 10.10.111.102

vital.engineering.nyu.edu:17049/vnc_auto.html

Connected (unencrypted) to: QEMU (344_13_22)

student@kali: ~

```
File Edit View Search Terminal Help
student@kali:~$ sudo nmap -sS 10.10.111.102
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 15:39 EDT
Nmap scan report for 10.10.111.102
Host is up (0.020s latency).
Not shown: 977 closed ports
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
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:00:00:00:00:06 (Xerox)

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
student@kali:~$
```

Command : sudo nmap -sT 10.10.111.102

Connected (unencrypted) to: QEMU (344_13_22)

Applications ▾ Places ▾ Terminal ▾ Wed 15:42 student@kali: ~

```
student@kali:~$ sudo nmap -sT 10.10.111.102
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 15:42 EDT
Nmap scan report for 10.10.111.102
Host is up (0.0032s latency).
Not shown: 977 closed ports
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
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:00:00:00:00:06 (Xerox)

Nmap done: 1 IP address (1 host up) scanned in 8.23 seconds
student@kali:~$
```

Apply rule to metasploitable iptable to all block tcp syn packets coming from kali server :

Command :

```
sudo iptables -A INPUT -p tcp --syn -s 10.10.111.100 -j DROP
```

Connected (unencrypted) to: QEMU (344_13_26)

```
No mail.  
msfadmin@metasploitable:~$ sudo iptables -L  
[sudo] password for msfadmin:  
Chain INPUT (policy ACCEPT)  
target     prot opt source          destination  
  
Chain FORWARD (policy ACCEPT)  
target     prot opt source          destination  
  
Chain OUTPUT (policy ACCEPT)  
target     prot opt source          destination  
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --syn -s 10.10.111.100  
-j DROP  
msfadmin@metasploitable:~$ sudo iptables -L  
Chain INPUT (policy ACCEPT)  
target     prot opt source          destination  
DROP      tcp   --  10.10.111.100      anywhere           tcp flags:FIN,SYN,R  
ST,ACK/SYN  
  
Chain FORWARD (policy ACCEPT)  
target     prot opt source          destination  
  
Chain OUTPUT (policy ACCEPT)  
target     prot opt source          destination  
msfadmin@metasploitable:~$
```

Connected (unencrypted) to: QEMU (344_13_22)

Applications ▾ Places ▾ Terminal ▾ Wed 16:00 student@kali: ~

```
student@kali:~$ sudo nmap -sS 10.10.111.102
[sudo] password for student:

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 15:59 EDT
Nmap scan report for 10.10.111.102
Host is up (0.00096s latency).
All 1000 scanned ports on 10.10.111.102 are filtered
MAC Address: 00:00:00:00:00:06 (Xerox)

Nmap done: 1 IP address (1 host up) scanned in 21.21 seconds
student@kali:~$ sudo nmap -sT 10.10.111.102

Starting Nmap 7.01 ( https://nmap.org ) at 2018-03-28 15:59 EDT
Nmap scan report for 10.10.111.102
Host is up (0.0042s latency).
All 1000 scanned ports on 10.10.111.102 are filtered
MAC Address: 00:00:00:00:00:06 (Xerox)

Nmap done: 1 IP address (1 host up) scanned in 21.17 seconds
student@kali:~$
```

Trade off of blocking all TCP SYN packets from an IP address :

There might be possibility of TCP SYN Flooding attack from server that we are blocking in our iptables rule, but instead of blocking all tcp syn packets, we can apply hashlimit to limit all incoming tcp connections. Filter that denies all tcp syn packets from specific ip address can be effective but temporarily, it's easy for attacker to adapt and he can use random ports or spoof it's ip address. Also blocking all tcp syn packet traffic, we are denying the legitimate request made by server.