University ID : N17385760
Net ID : iv447

LAB3

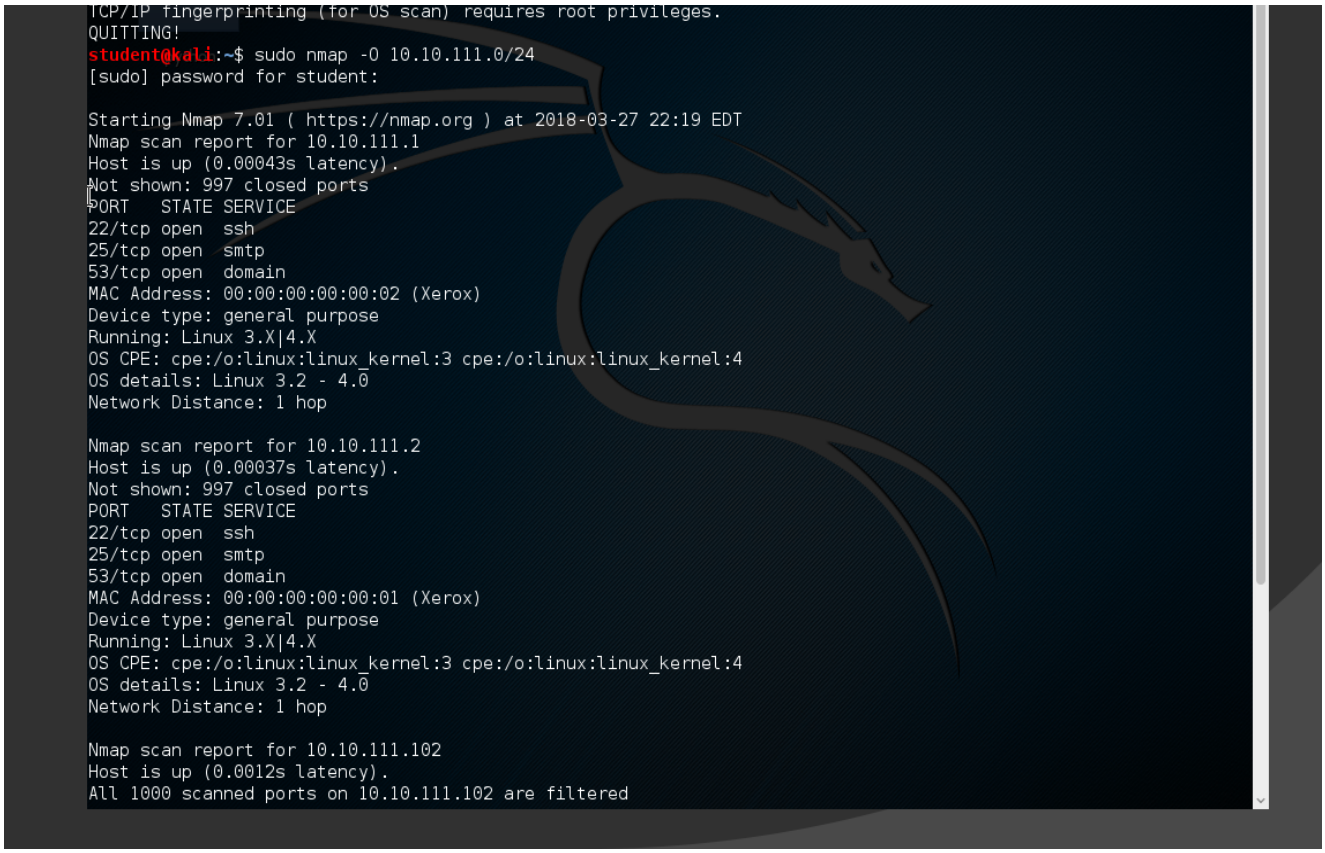
1. NMAP

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.

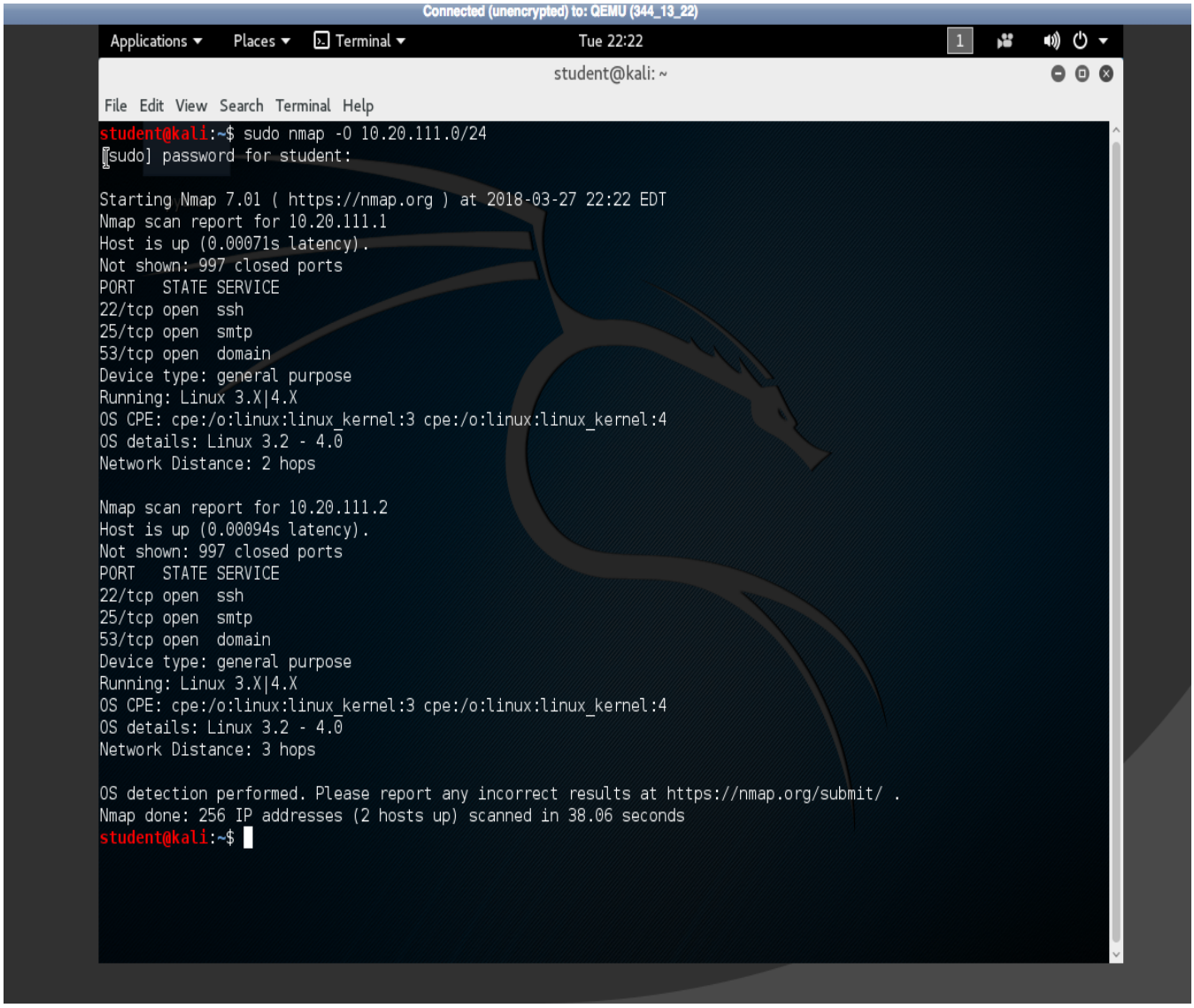


2. Using nmap, find all the open ports and OS on each host in the 10.20.111.0/24 network. List the command that is used. [10 points]

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

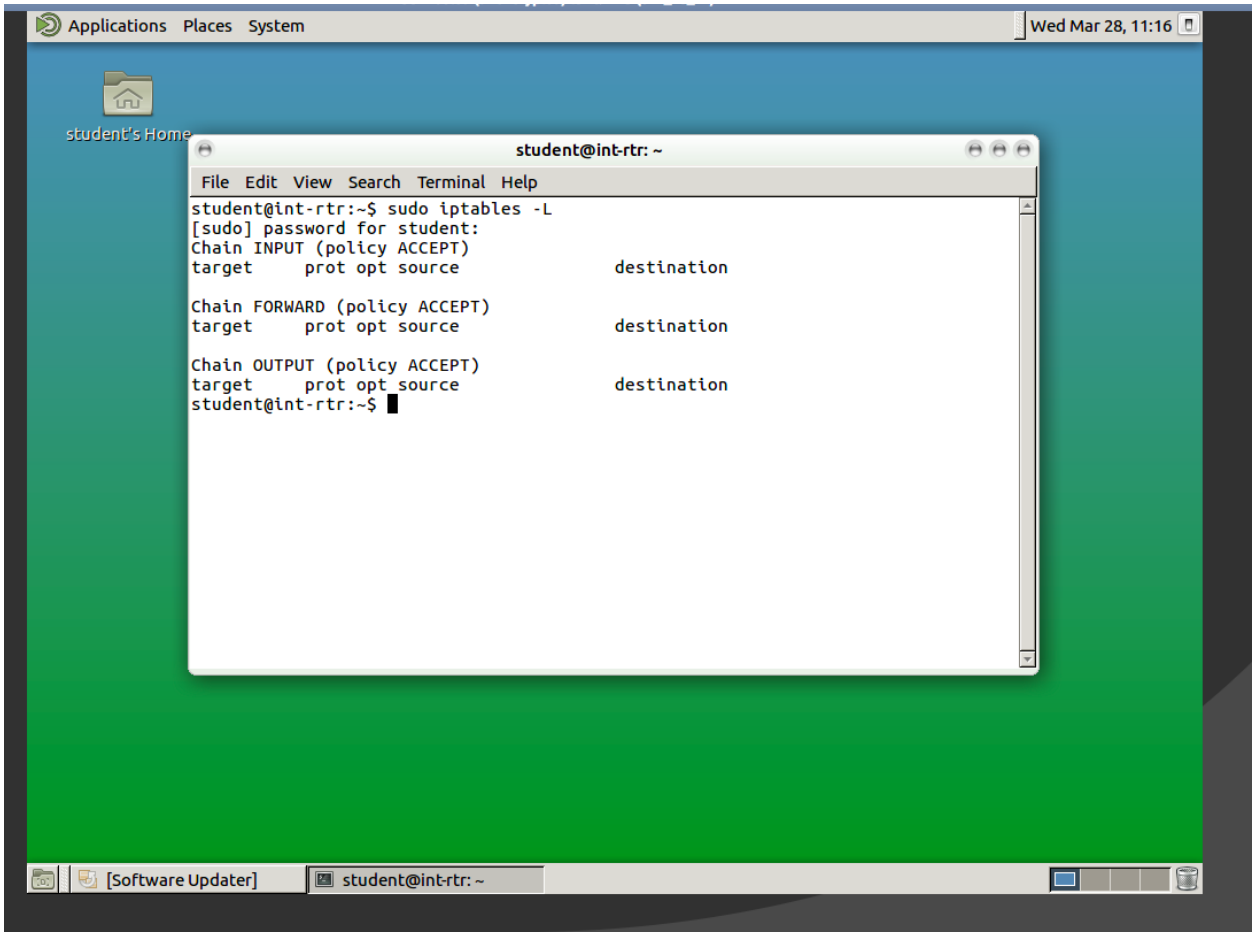
This will find all the open ports and OS on each host in the 10.20.111.0/24 network.



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.

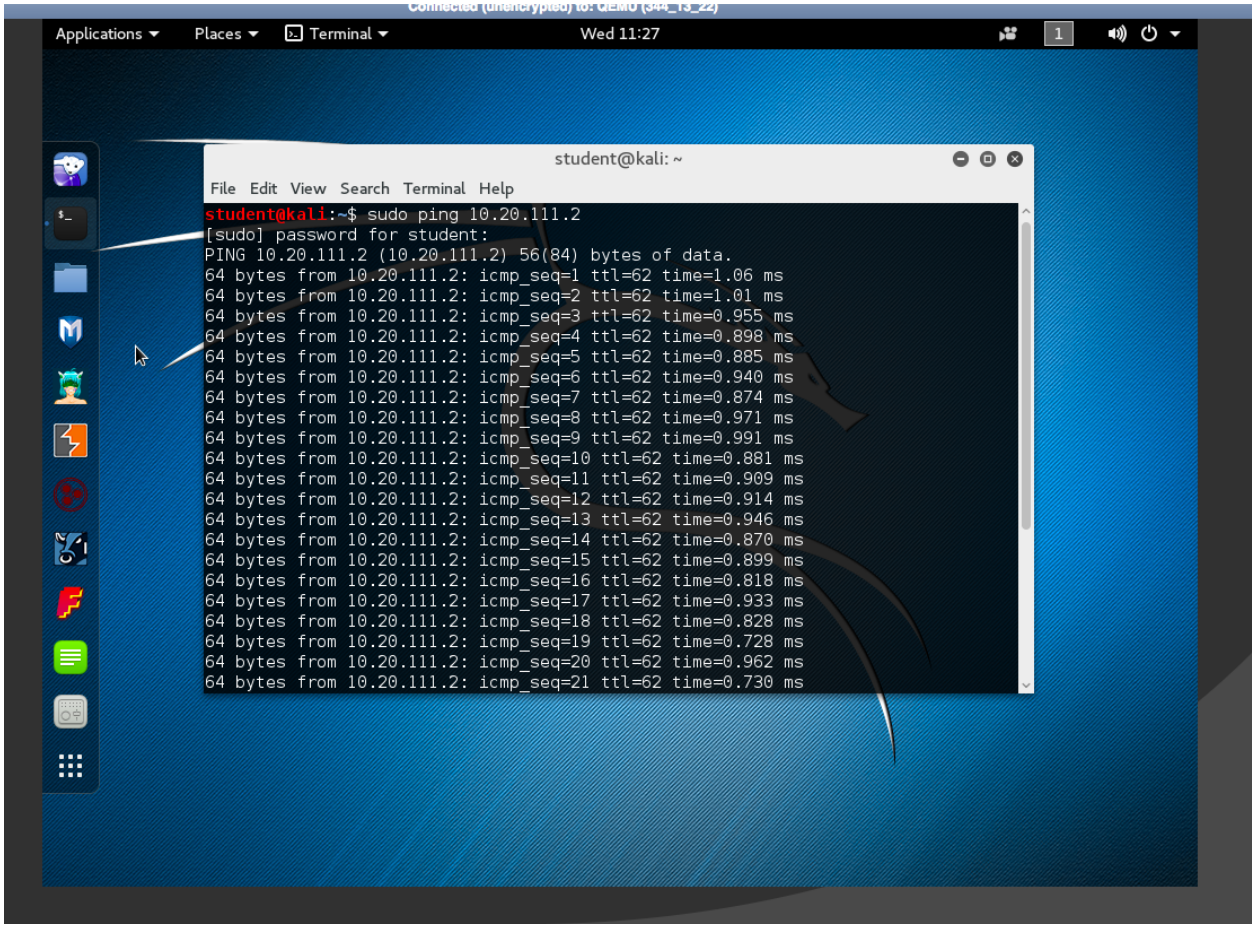
When the default rule is ACCEPT on INPUT, FORWARD and OUTPUT chains on Internal Router Iptables . External network can communicate with internal network and internal network can communicate with external network without any restriction.



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

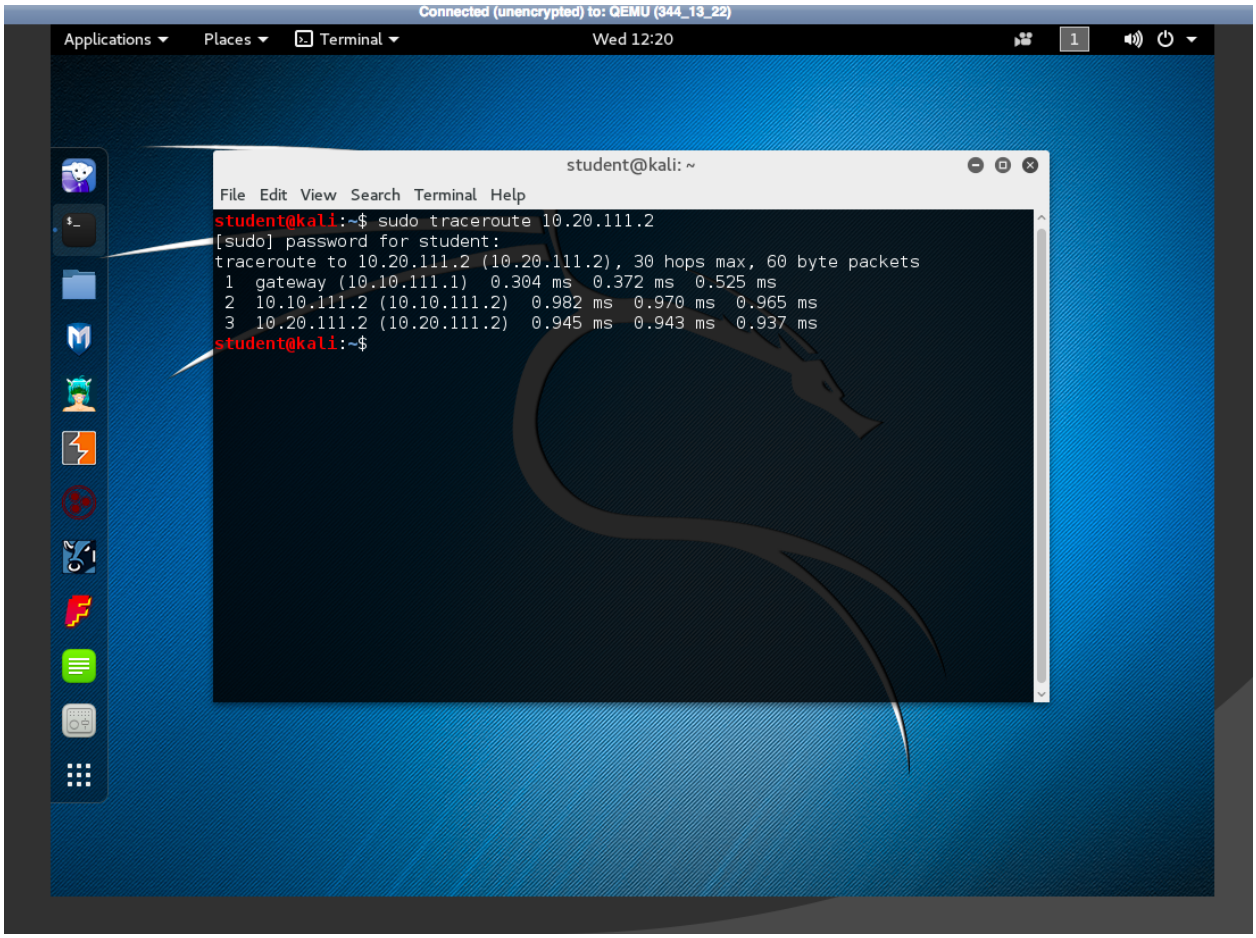
1. Ping from kali to Int-linux 10.20.111.2 :

   Command : sudo ping 10.20.111.2



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 :

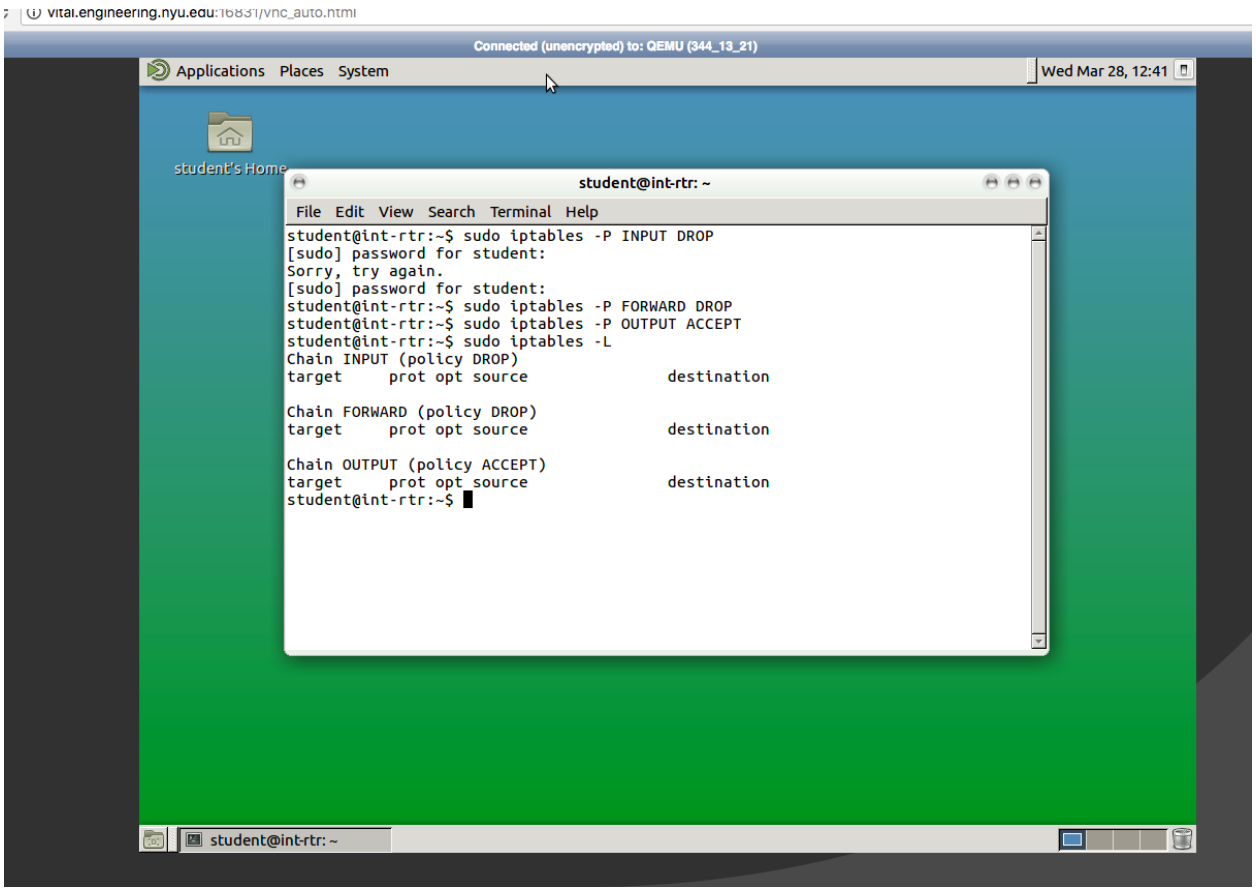
A) For outgoing traffic (from 10.20.111.0/24 to 10.10.111.0/24) - your internal machine should be able to communicate with the external network and the external machines without restrictions. [10 pts]

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

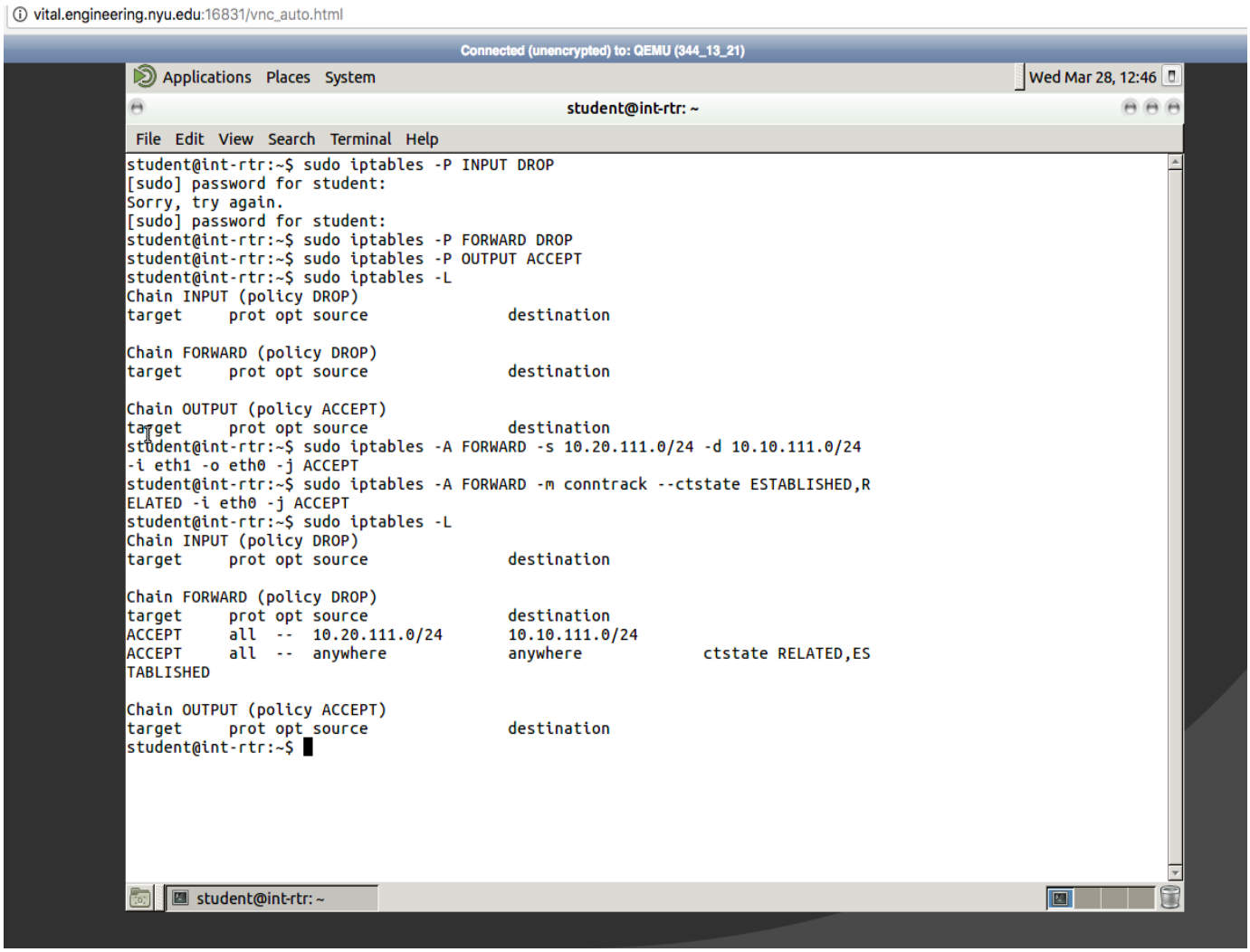


These will be default rules for our firewall.

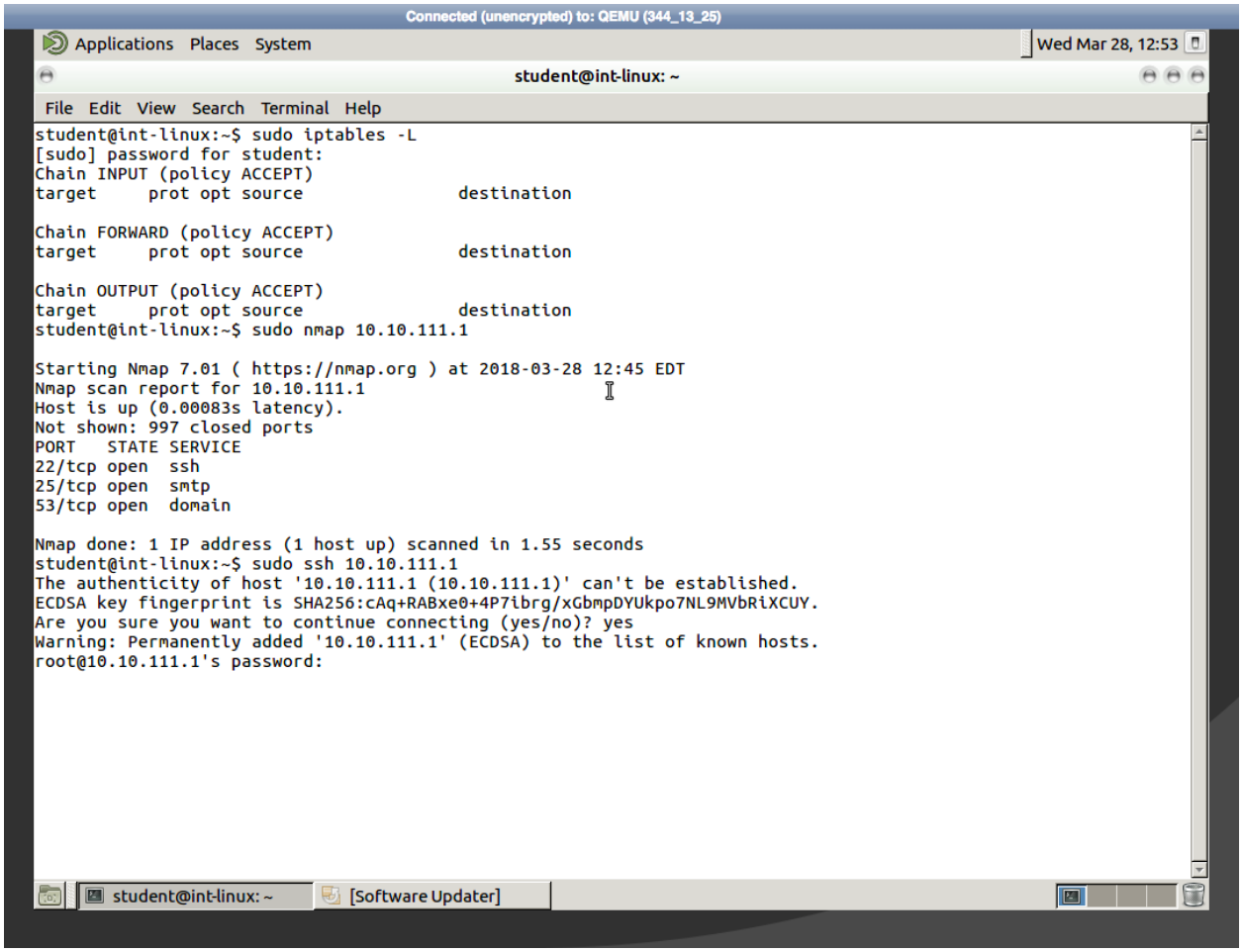
For outgoing traffic (from 10.20.111.0/24 to 10.10.111.0/24) We can add following rules to FORWARD chain so that internal machine is able to communicate with the external network and the external machines without restrictions :

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

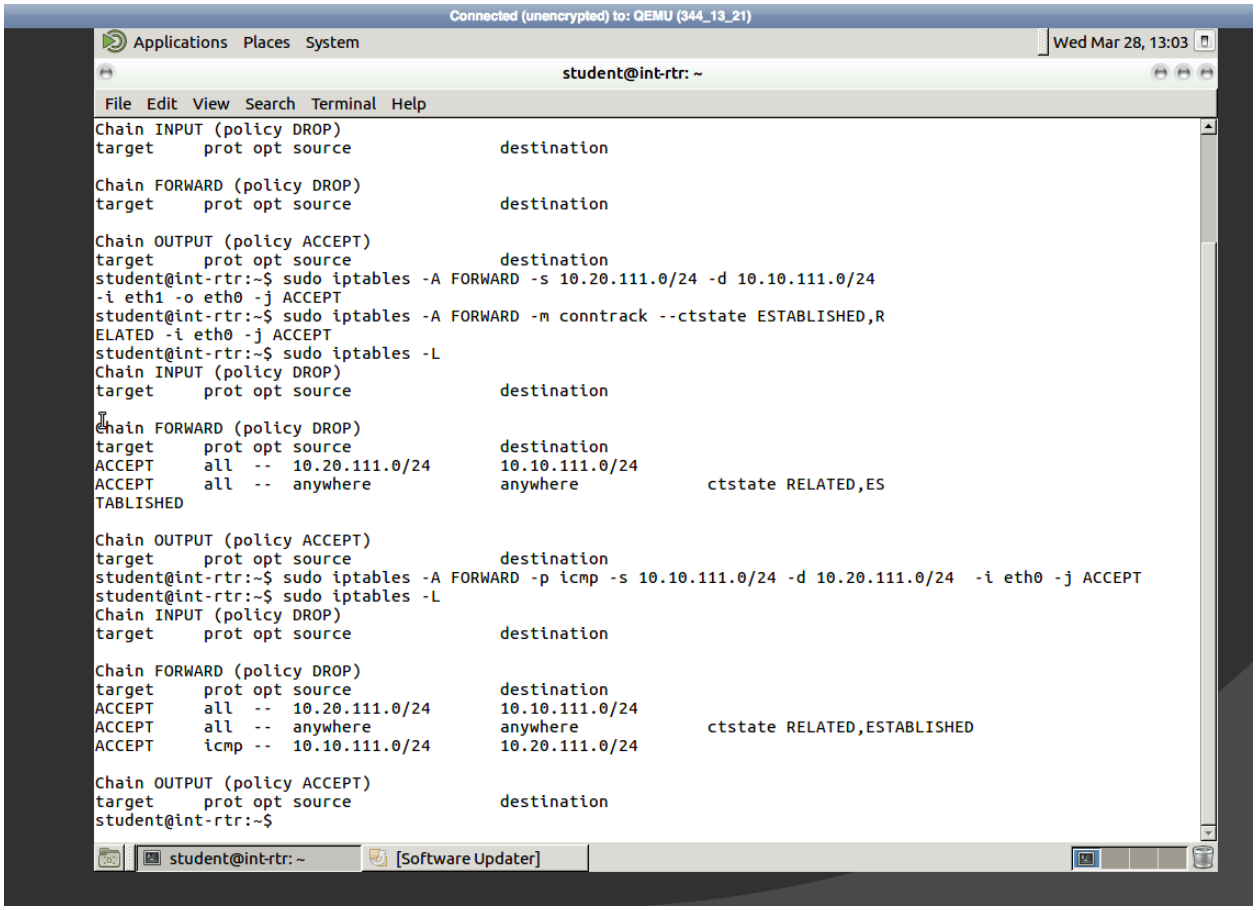


Checking for traffic from int-linux machine :



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 :

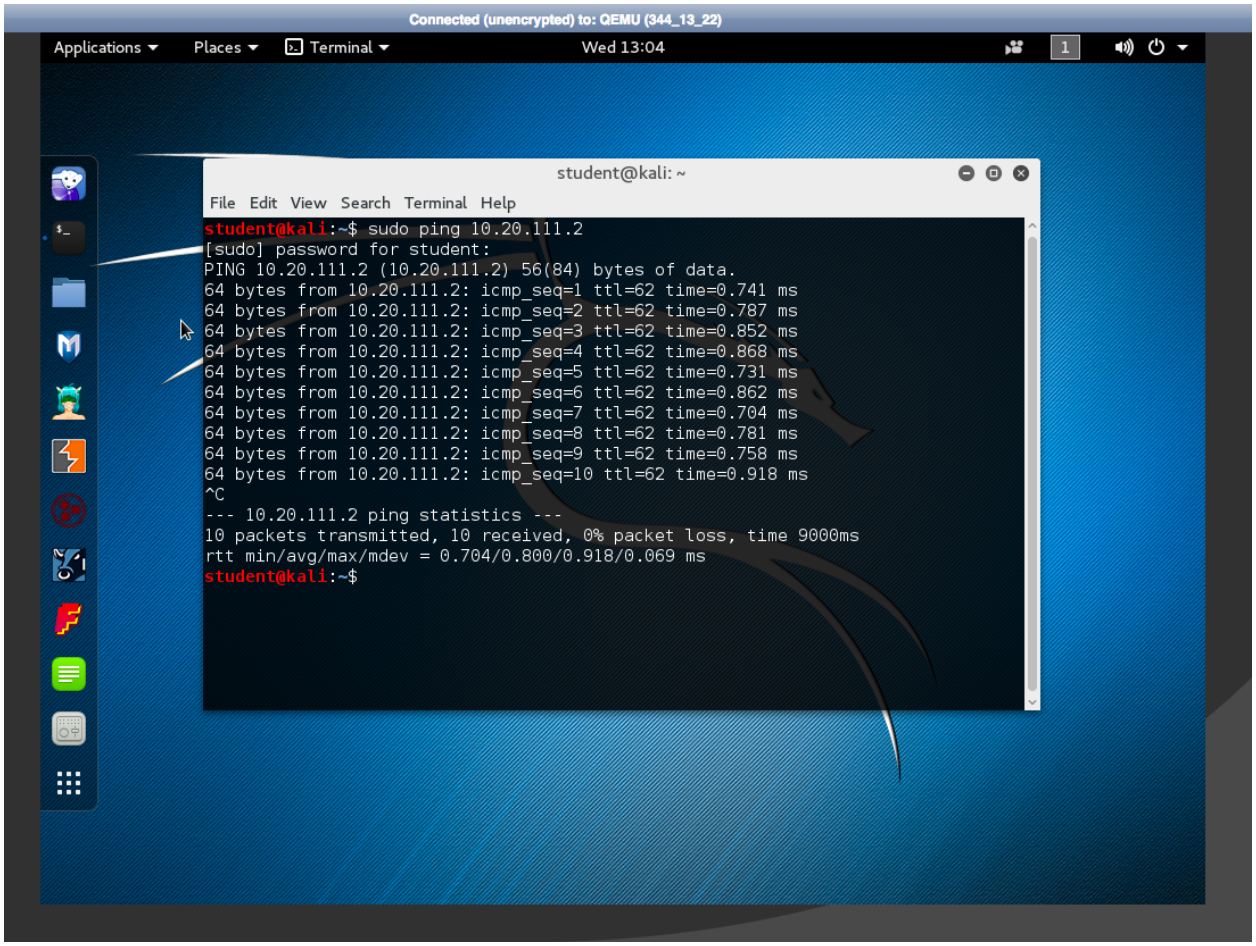
sudo iptables -A FORWARD -p icmp -s 10.10.111.0/24 -d 10.20.111.0/24 -i eth0 -j ACCEPT
sudo iptables -L



Ping from 10.10.111.0/24 network :

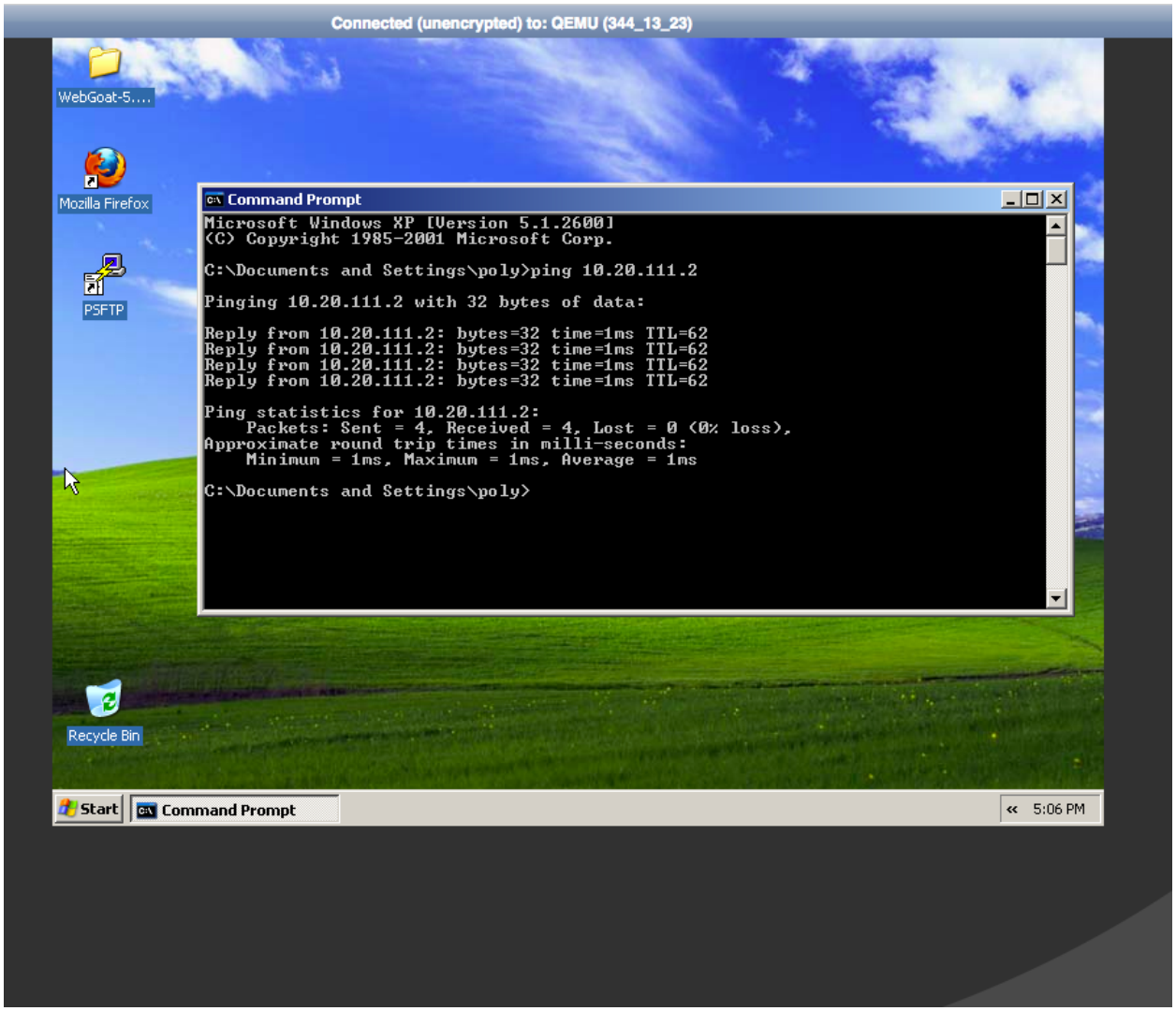
Ping from kali machine :

Command : sudo ping 10.20.111.2



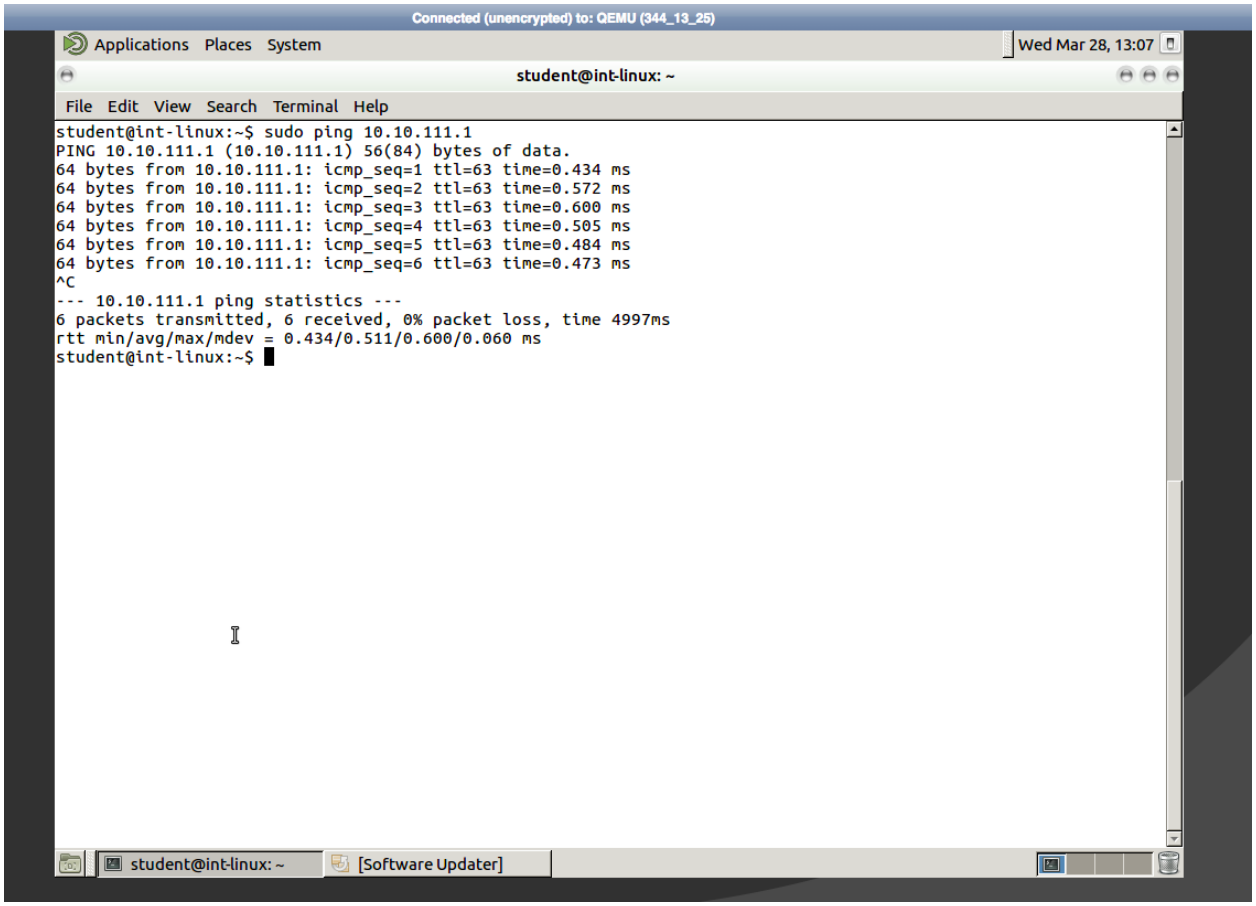
ping from windows XP :

Command : ping 10.20.111.2



Ping from 10.20.111.2 to external network :

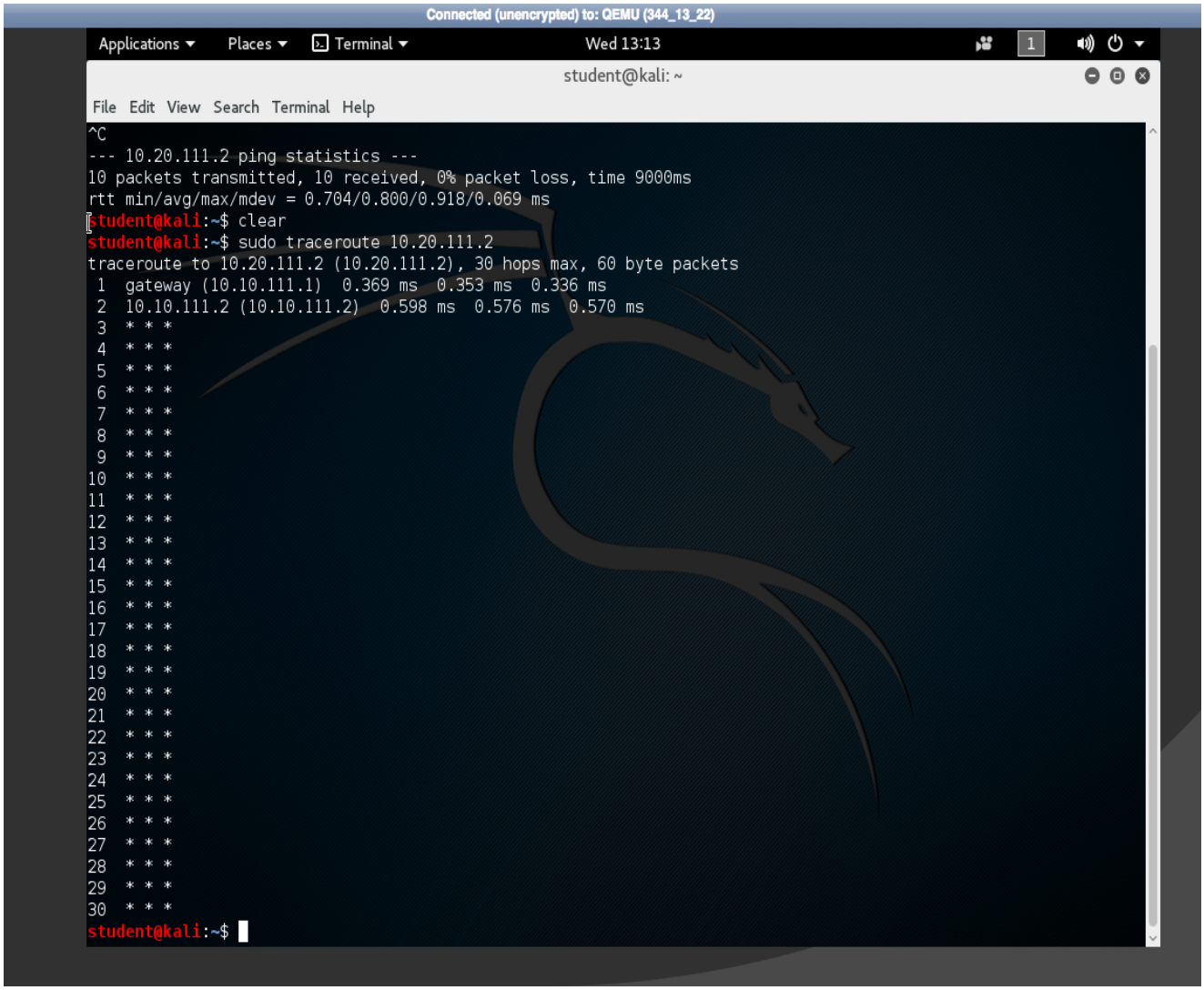
Command : sudo ping 10.10.111.1



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.

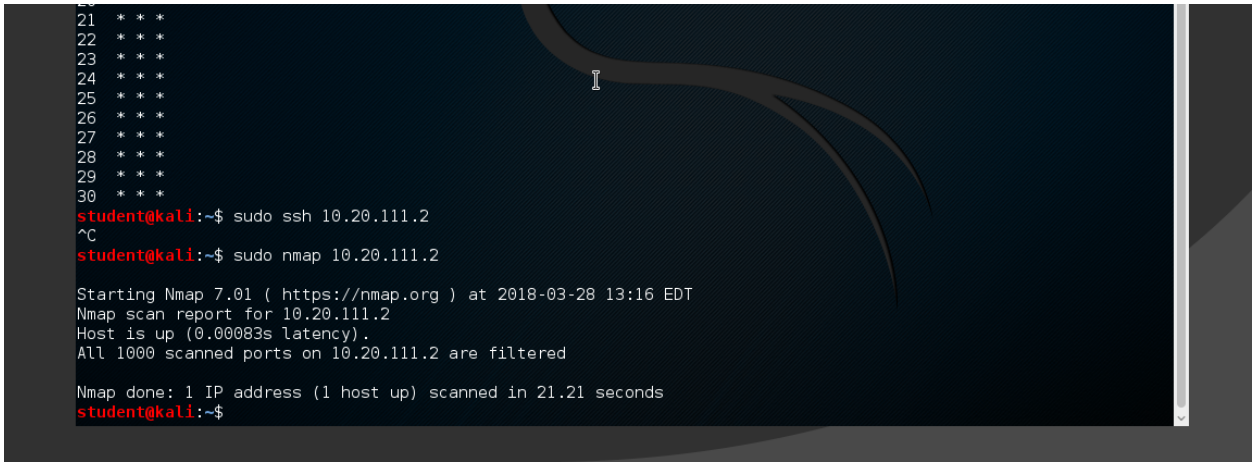
Traceroute from external network for 10.20.111.2 :

Command : sudo traceroute 10.20.111.2



Ssh and nmap from kali :

Command :
sudo ssh 10.20.111.2
sudo nmap 10.20.111.2



3. NMAP & IPTABLES:

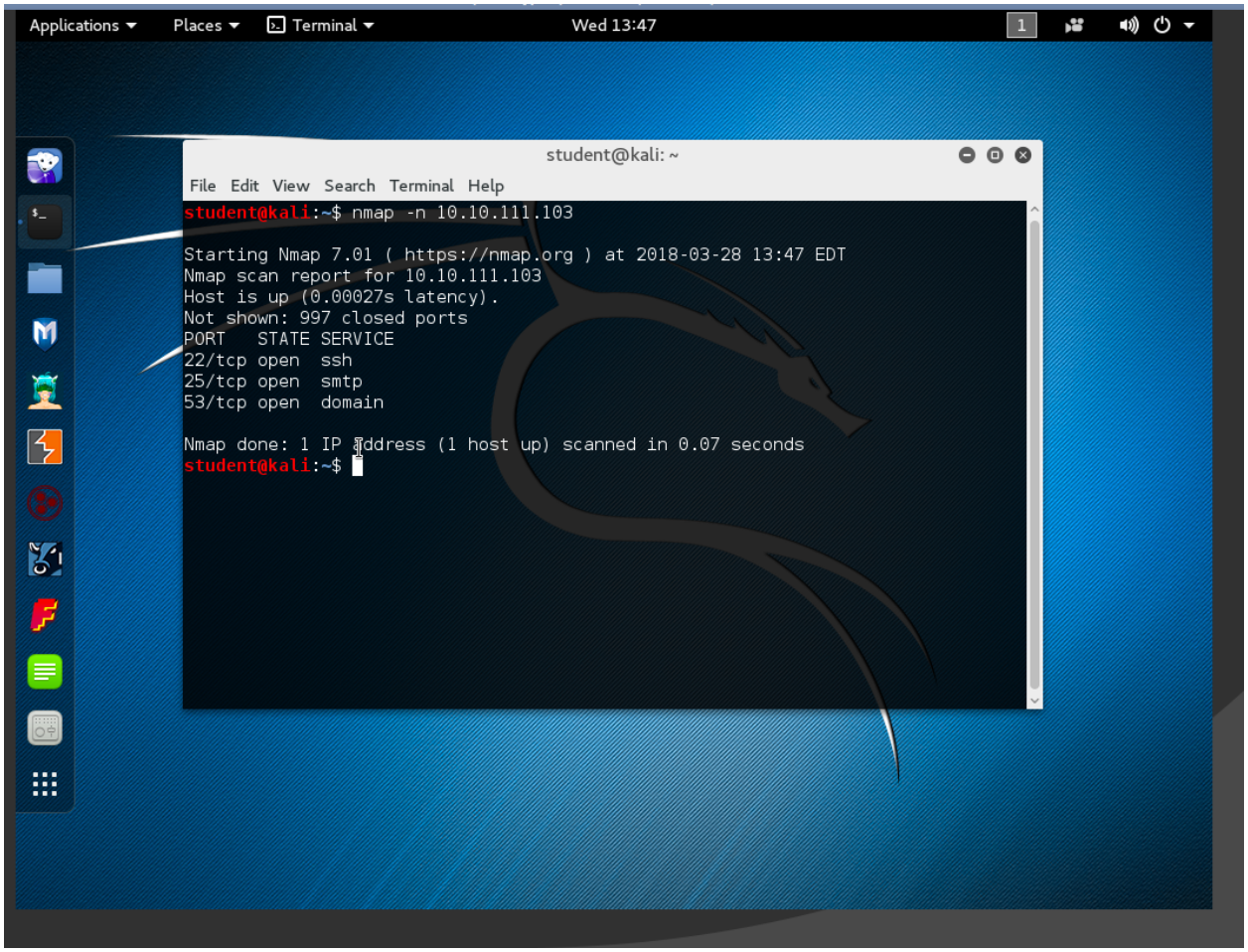
1) Following are the options you will find yourself often needing when using nmap. Use each of these options to perform a scan on the Ubuntu VM using Kali as an attacker machine (see Lab 0 for the login information for the Ubuntu machine). Submit a quick one-liner beside each to explain what each does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

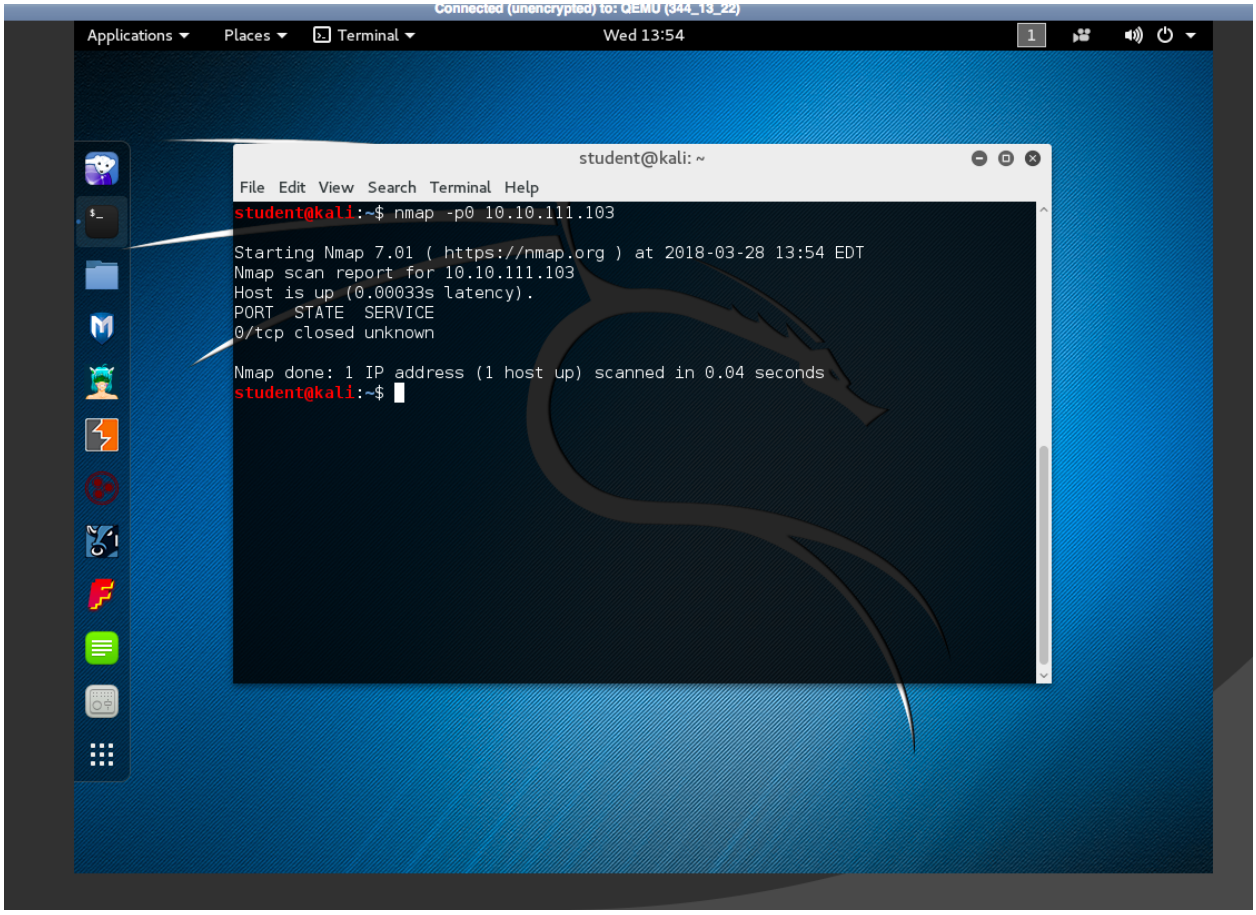
Description : If we use -n with nmap it does not perform DNS resolution.



b) -P0

Command : nmap -p0 10.10.111.103

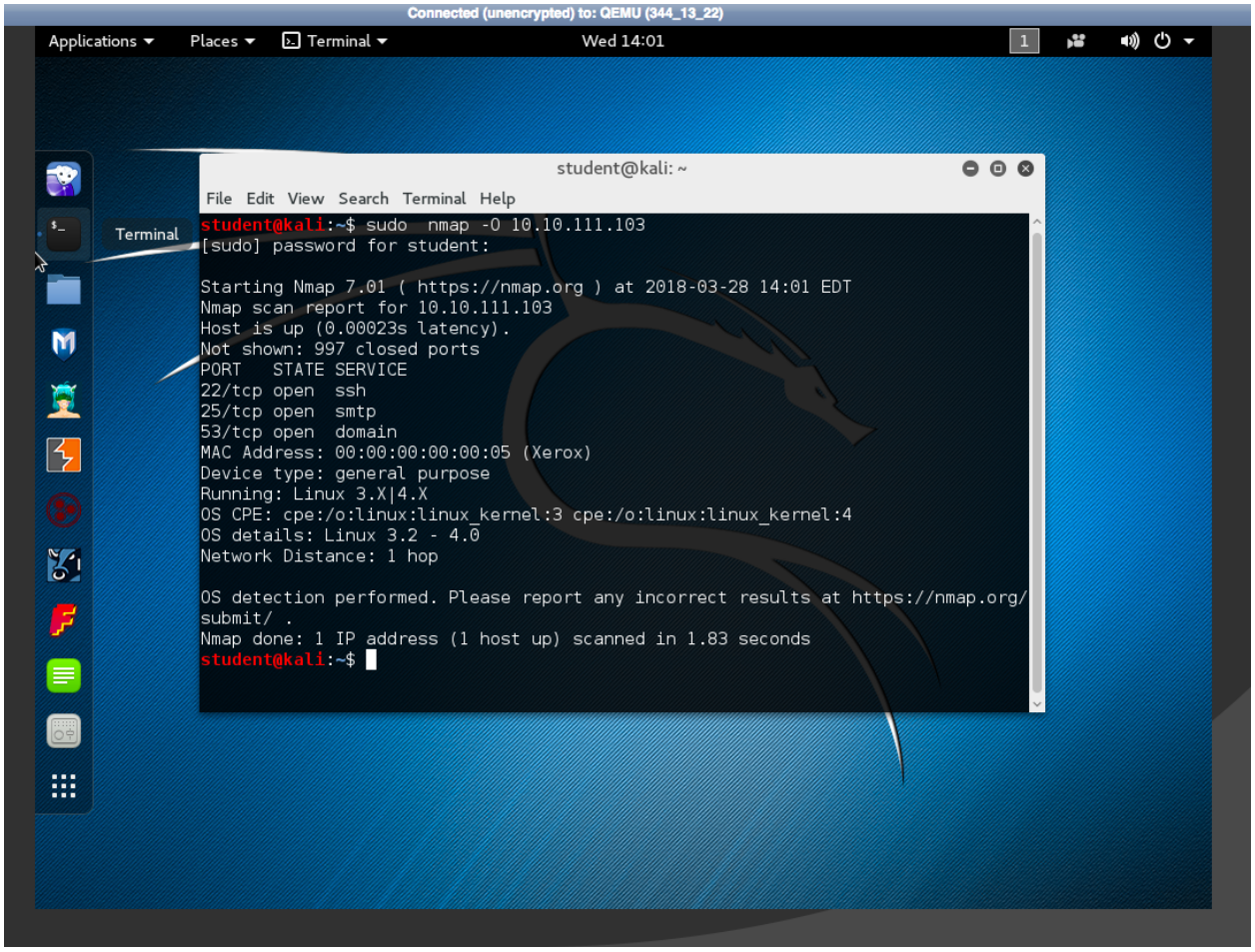
Description : Leaving off end port in range, makes the scan go through to port 65535.



c) -O

Command : sudo nmap -O 10.10.111.103

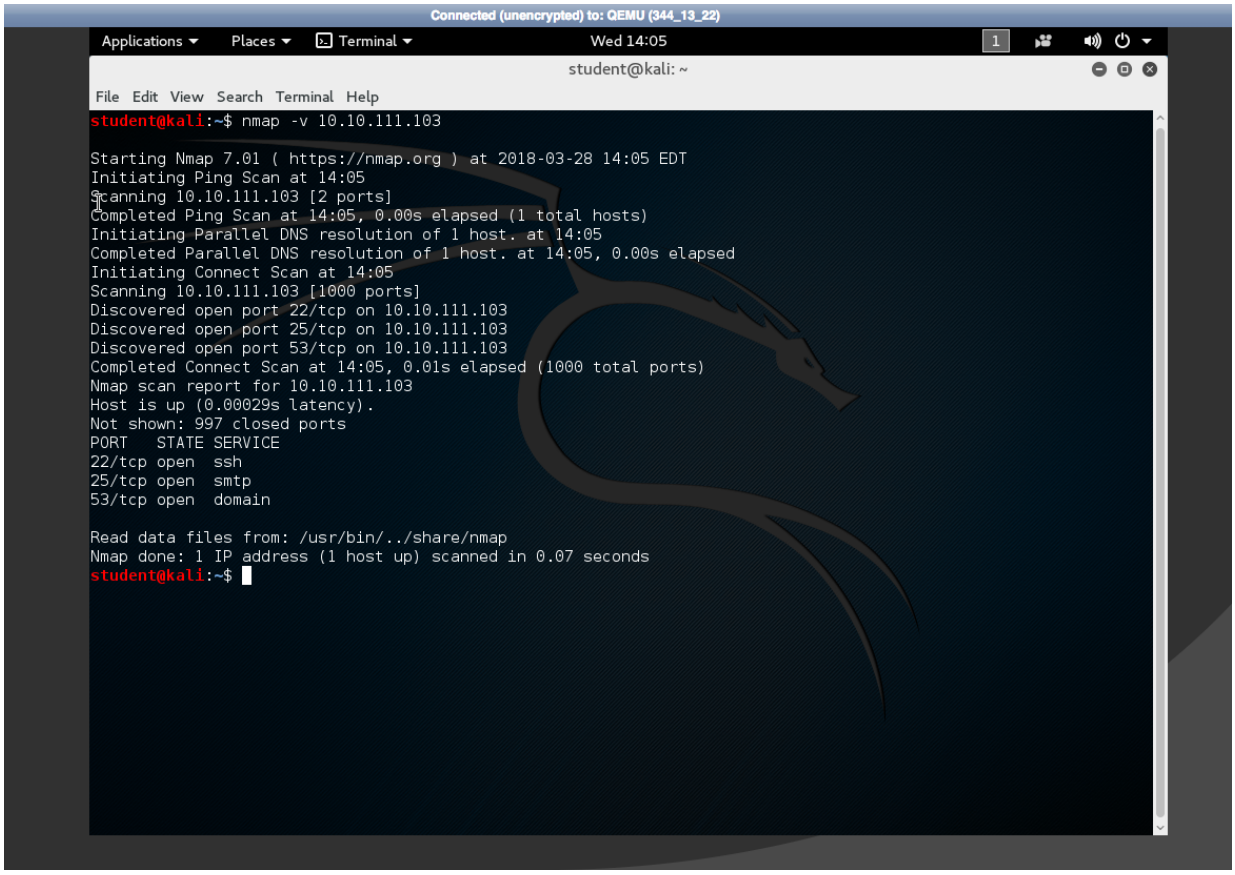
Description : Above command performs remote OS detection using TCP/IP stack fingerprinting.



d) -v

Command : nmap -v 10.10.111.103

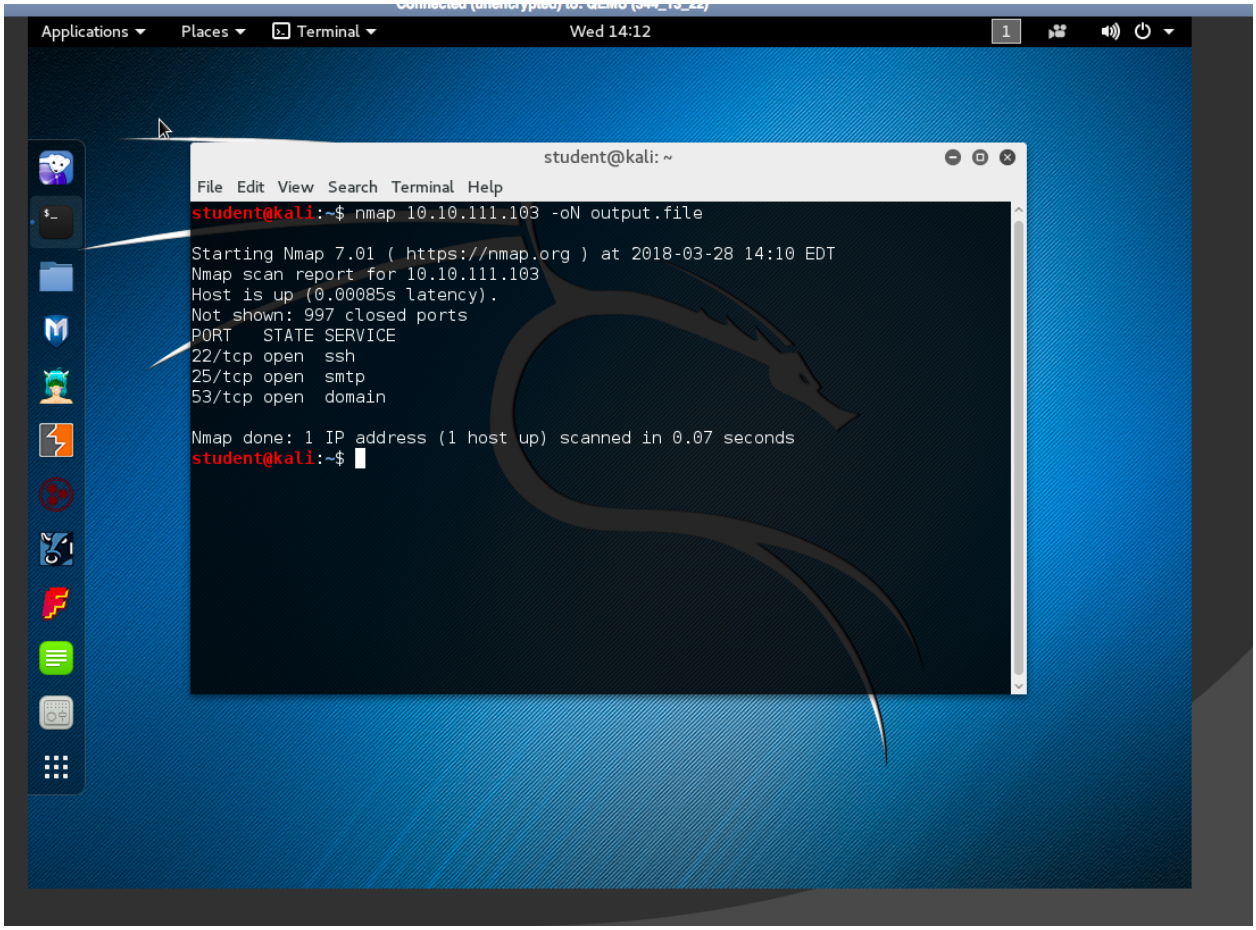
Description : Increase verbosity level, description will be in more details.



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.

a) 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)

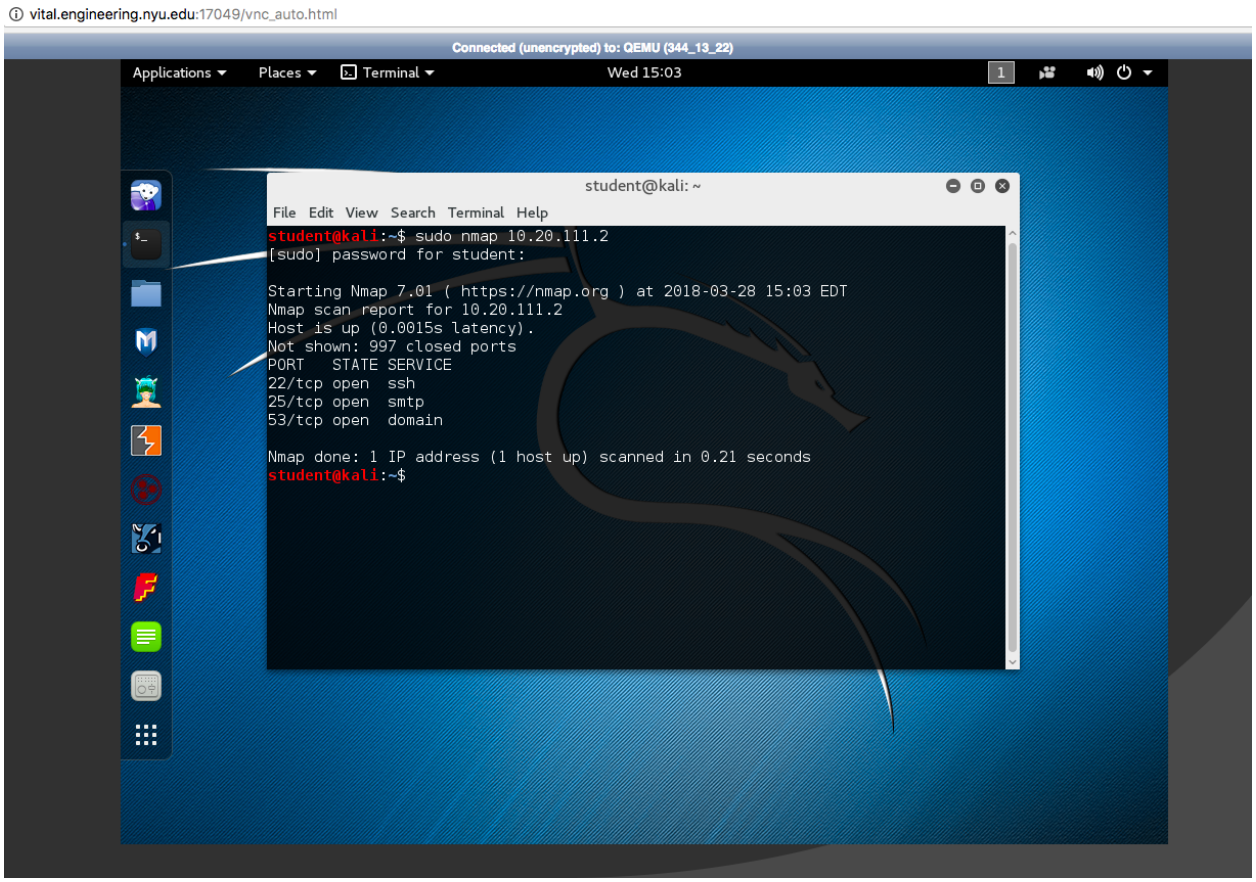
b) Now that you have implemented the appropriate rules on the Int-Linux VM, execute nmap from Kali. Submit screenshots of your nmap command and results of your scan. (10 points)

c) 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 :



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

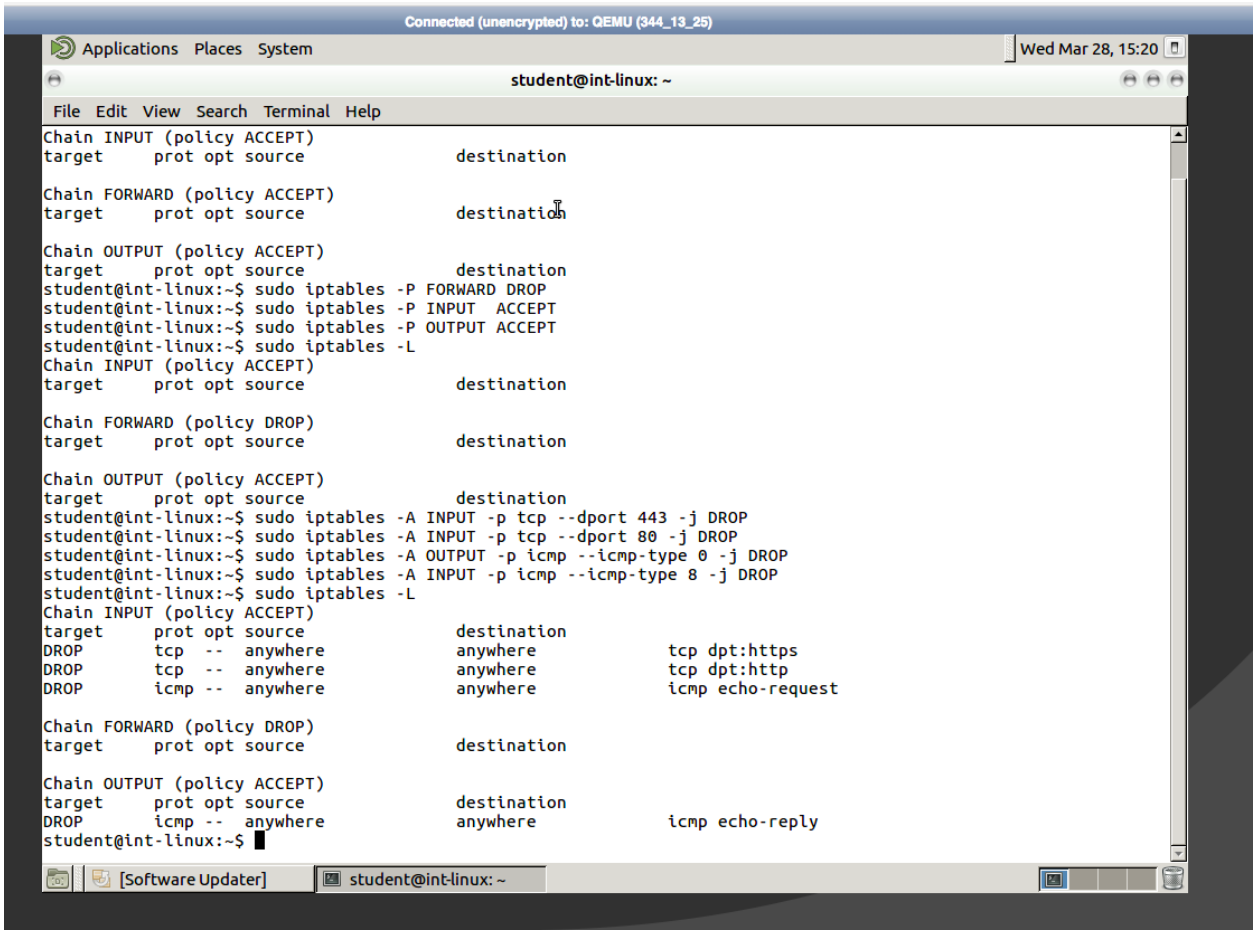
Set the following default rules to iptables :

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 :

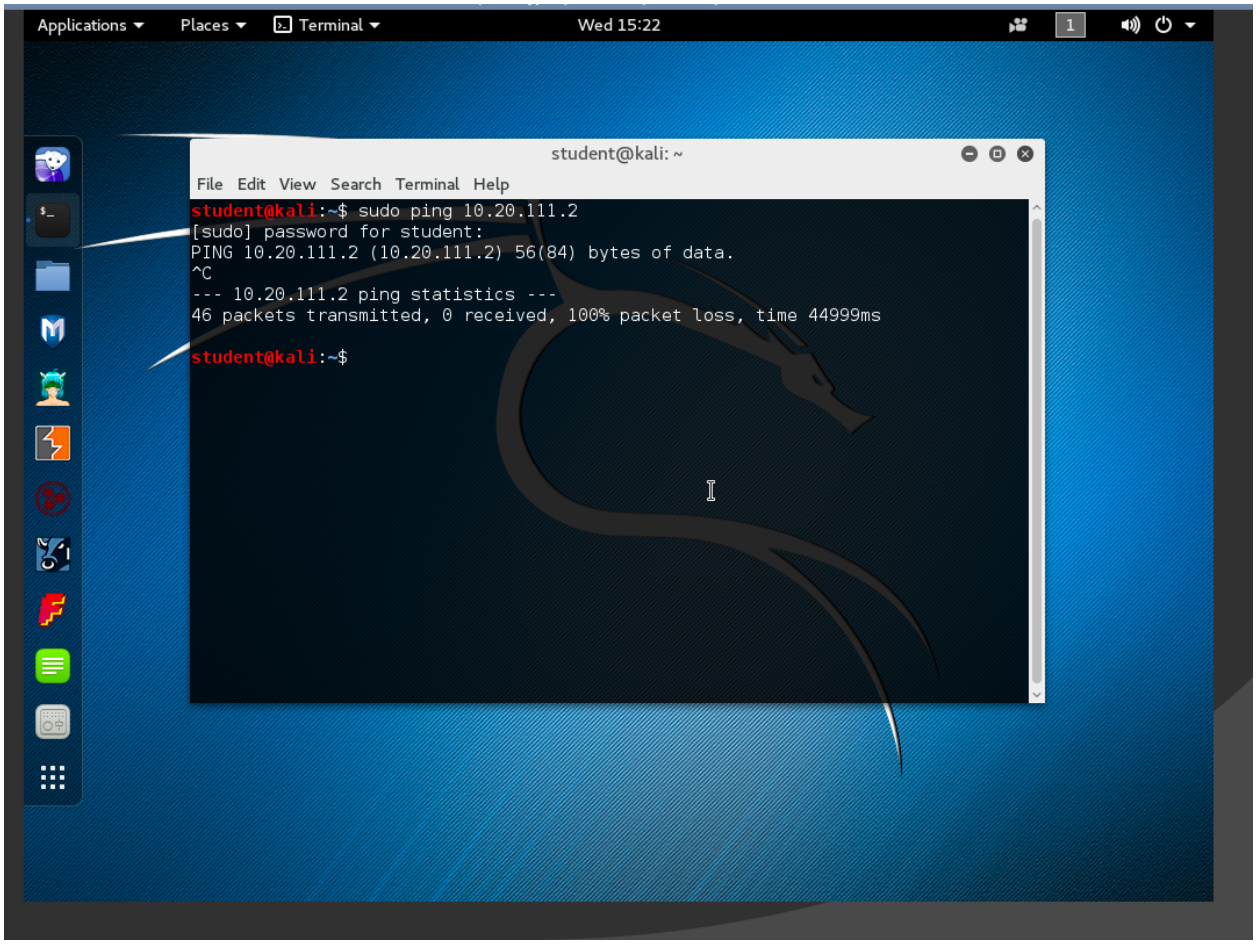
sudo iptables -A INPUT -p tcp --dport 443 -j DROP
sudo iptables -A INPUT -p tcp --dport 80 -j DROP
sudo iptables -A OUTPUT -p icmp --icmp-type 0 -j DROP
sudo iptables -A INPUT -p icmp --icmp-type 8 -j DROP
sudo iptables -L

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



ICMP ping from kali to 10.20.111.2 :

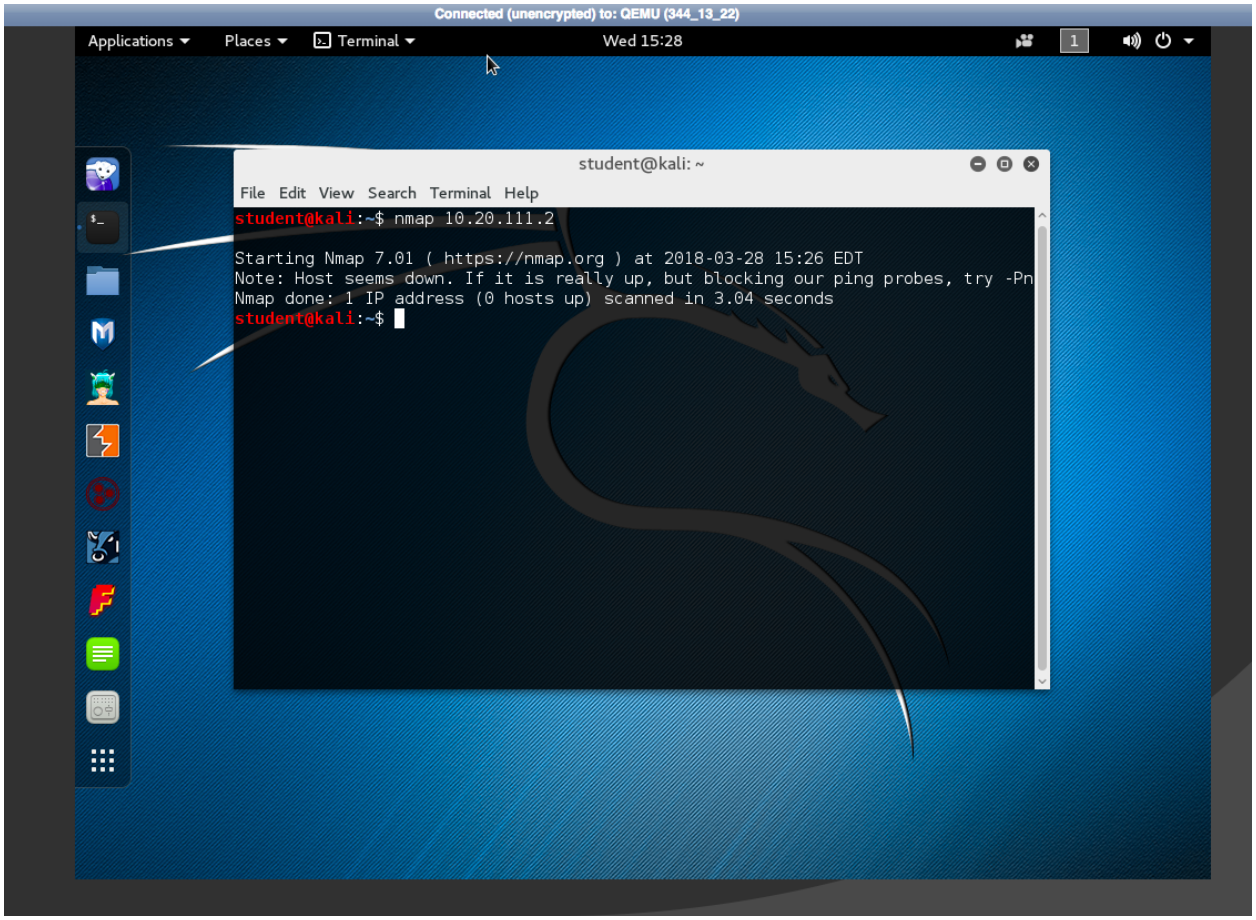
Command : sudo ping 10.20.111.2



nmap after applying above rule :

Command : nmap 10.20.111.2

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

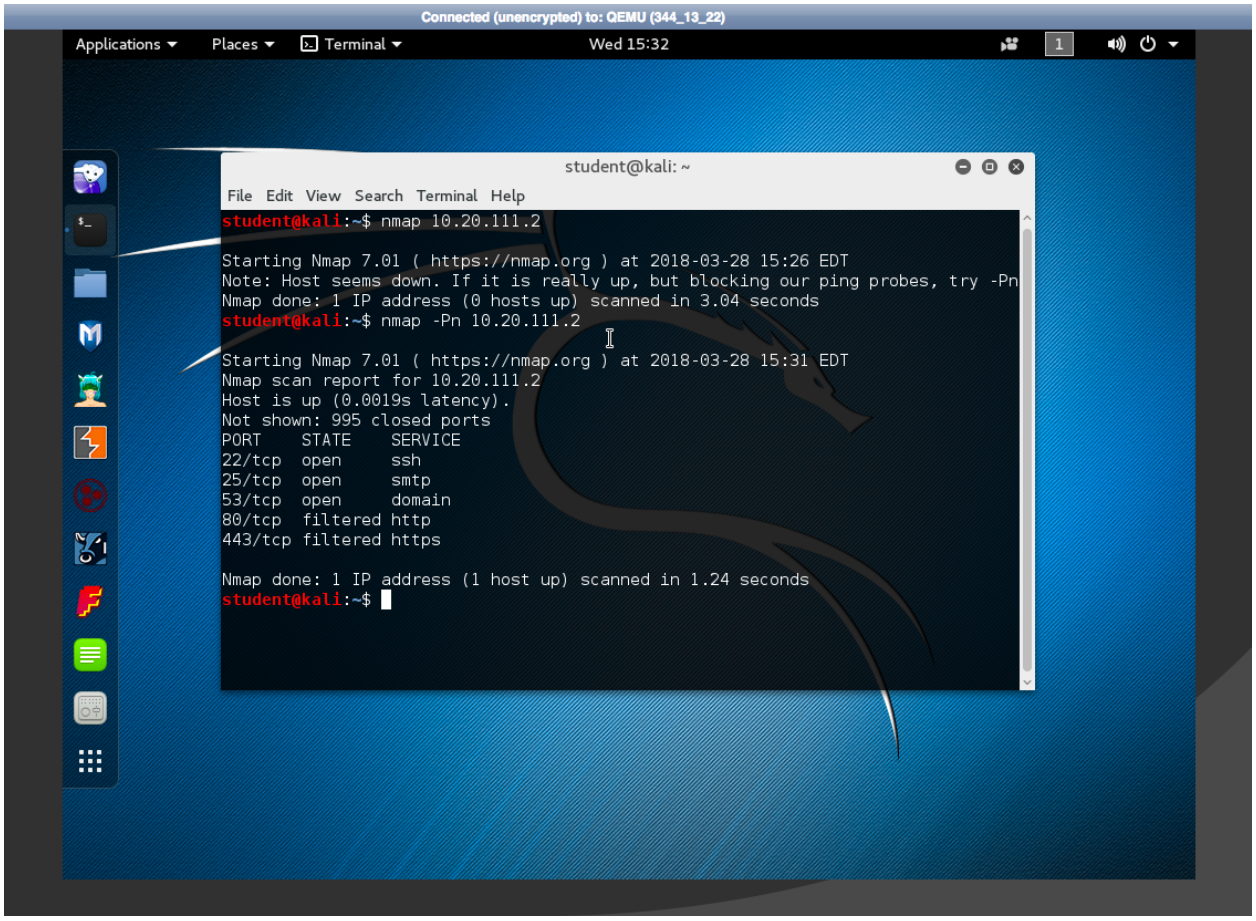


There is a method of forcing nmap to scan hosts even if the initial nmap probes are blocked. Leaving the iptables in place that block nmap's initial probe requests, run nmap with a set of options that scans IntLinux even when it doesn't reply to nmap's initial probe requests.

Following command can be used for this :

Command :

nmap -Pn 10.20.111.2

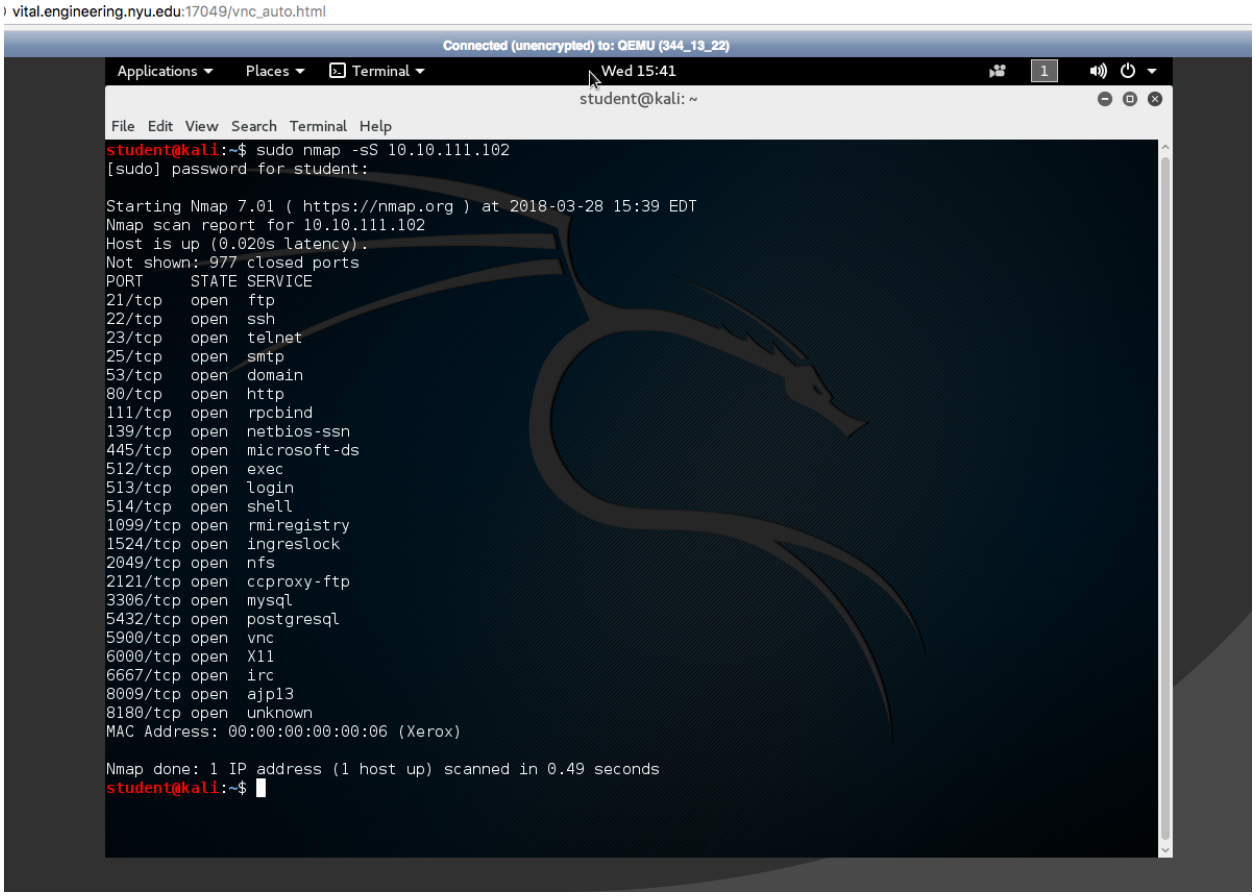


3) Using the Kali machine as an attacker machine, perform a nmap TCP SYN scan on the Metasploitable VM (see Lab 0 for the login information for the Metasploitable machine). Then construct an iptable rule to block all incoming TCP SYN packets only from the Kali scanning server's IP address. Explain the 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)

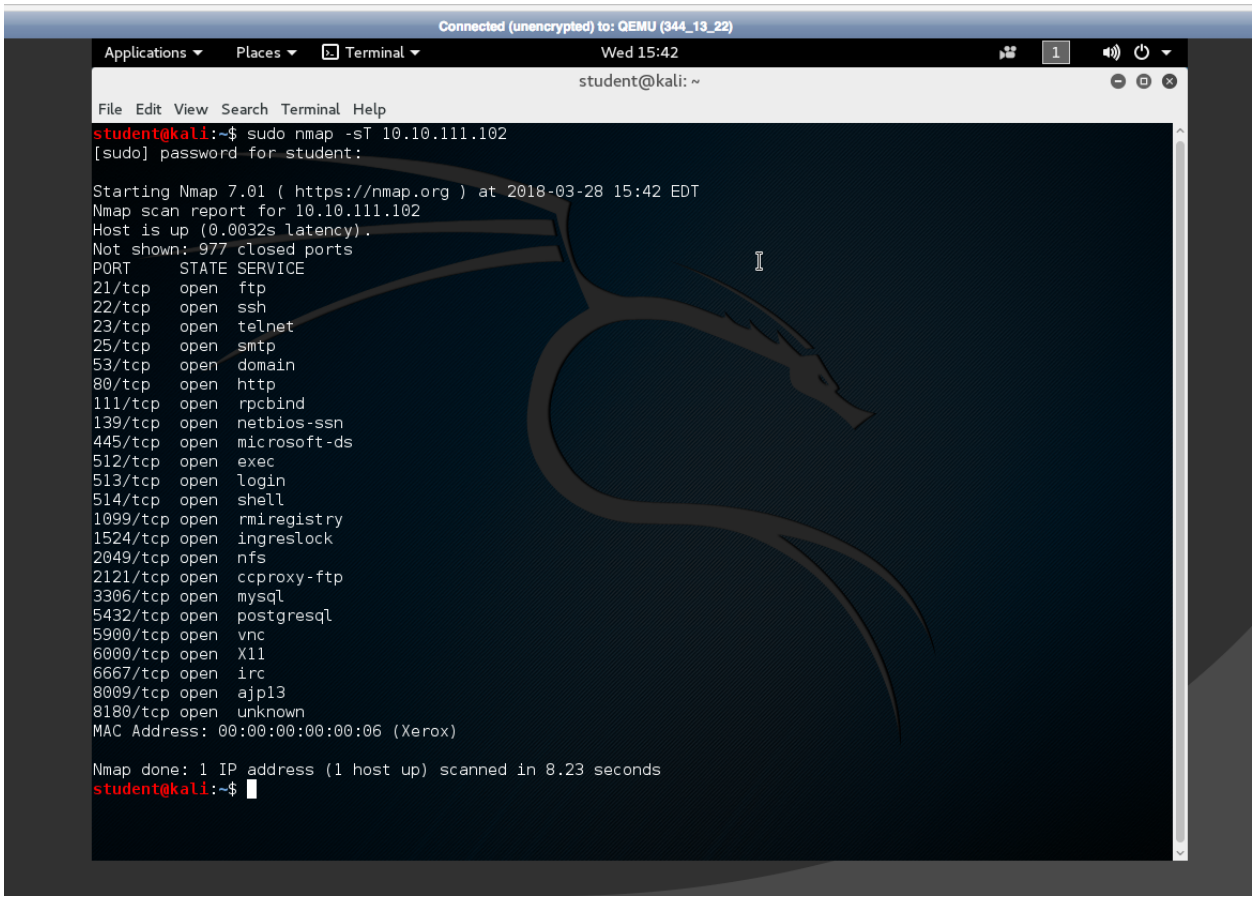
For this part we will apply the rules on metasploitable VM :

Perform nmap probe from kali on metasploitable VM :

Command : sudo nmap -sS 10.10.111.102

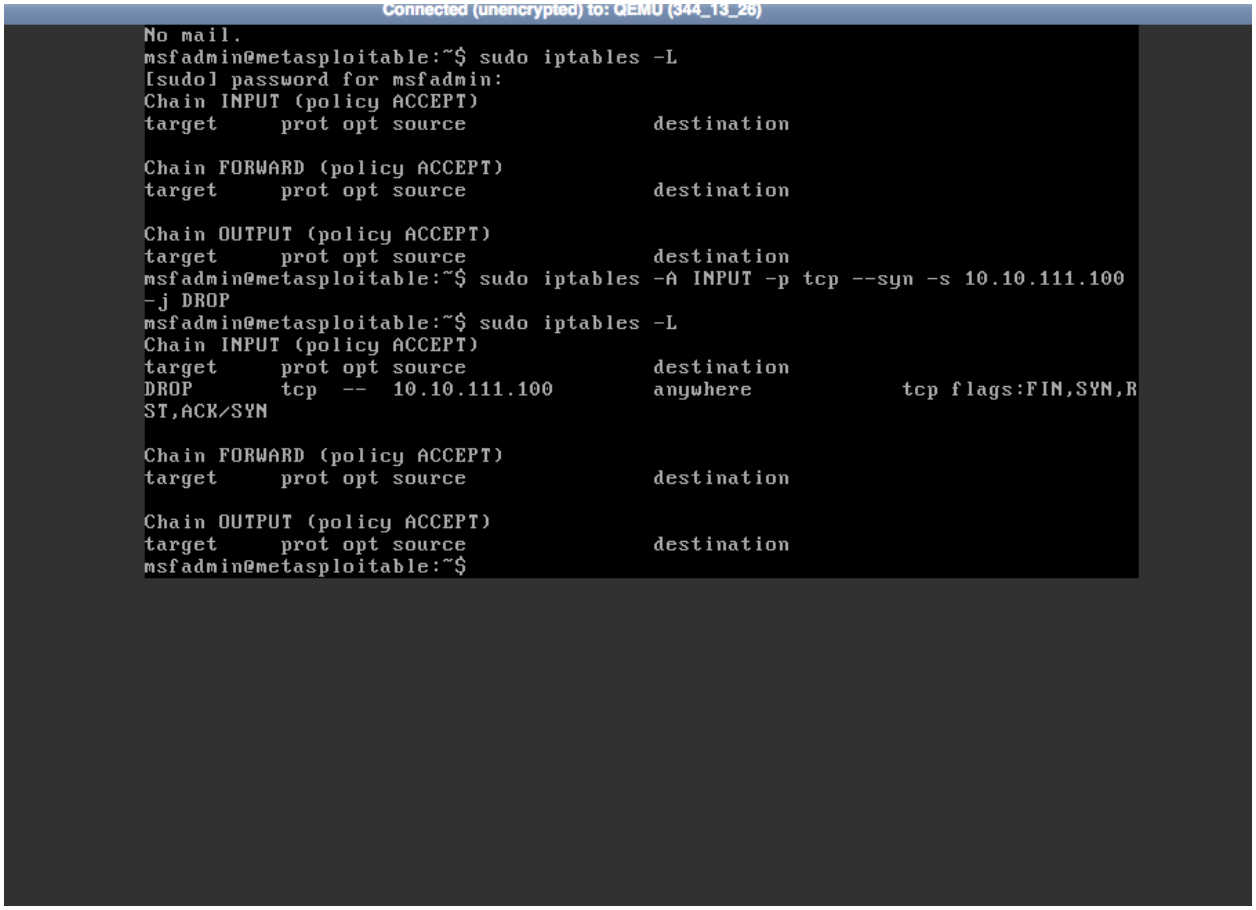


Command : sudo nmap -sT 10.10.111.102



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

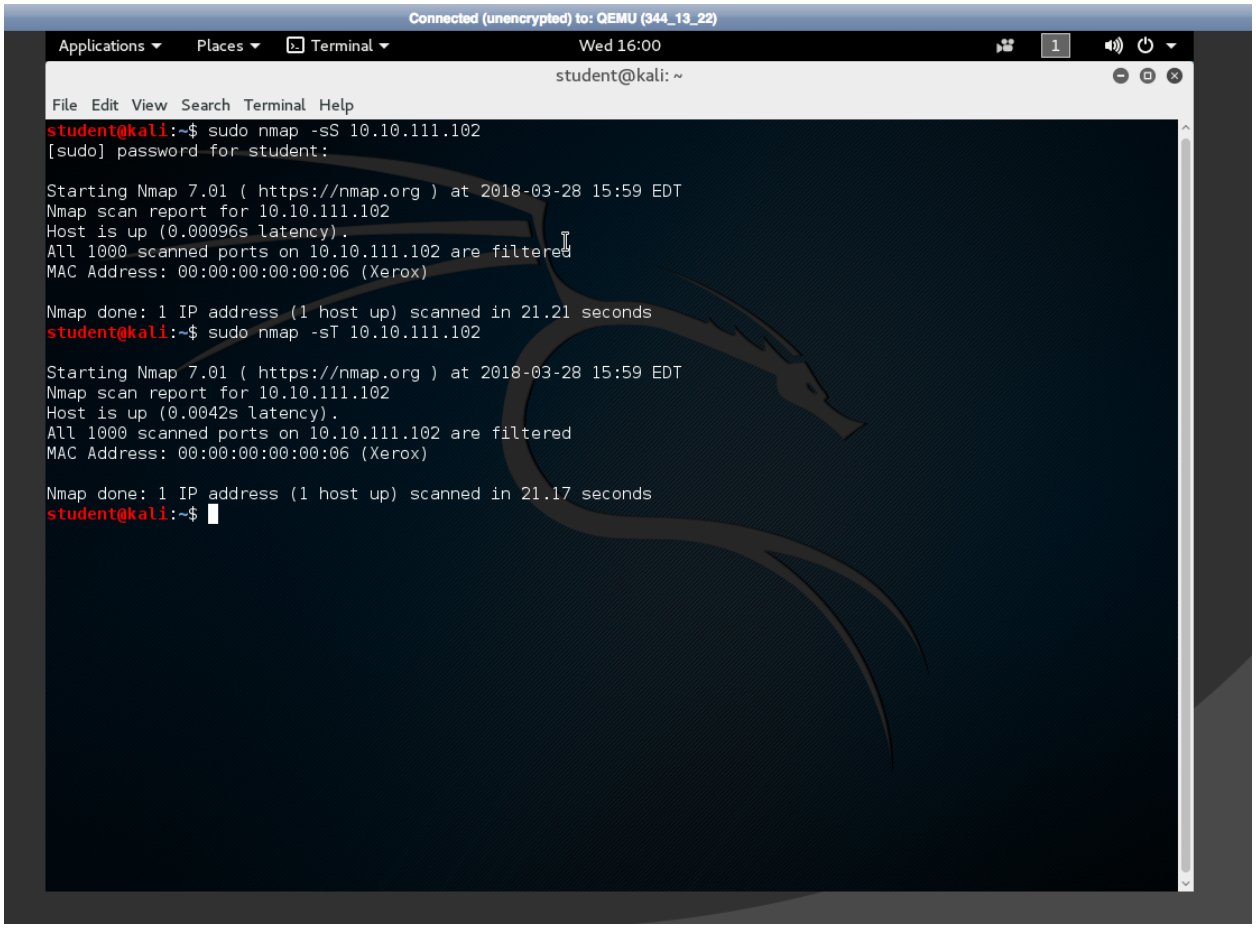

Connected (unencrypted) to: QEMU (344_13_26)
No mail.
msfadmin@metasploitable:~$ sudo iptables -L
[sudo] password for msfadmin:
Chain INPUT (policy ACCEPT)
target   prot opt source   destination

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:~$



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.