

Packet filter firewall project 1

Student Name: Mohammed Ahmed Ragab

Email: maragab@asu.edu

Submission Date: 24 October 2023

Class Name and Term: CSE548 Fall 2023

I. PROJECT OVERVIEW

This project is about creating a stateless firewall that block/allow the packets on a server_gateway virtual machine which is connected to a client virtual machine on the same NAT network. The firewall can be achieved by using the linux command “iptables” that allow the user to write rules which specifies whether an IP packets is accepted or dropped based on criteria such as source IP and protocol (ICMP, TCP ,... etc).

II. NETWORK SETUP

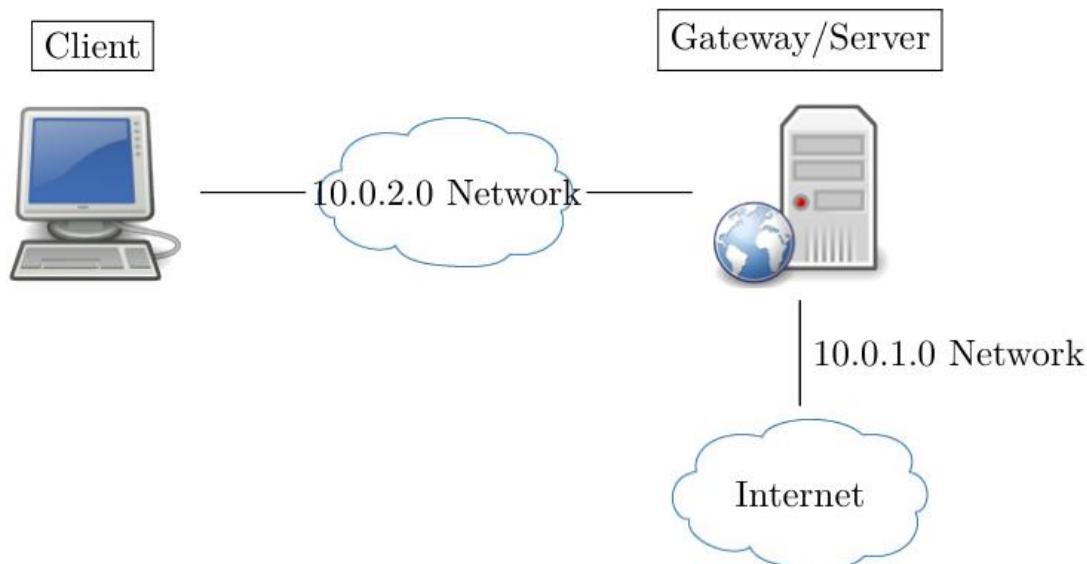
The network topology

client IP = 10.0.2.6

client interface = enp0s3

gateway/server IP for NATnetwork = 10.0.2.7

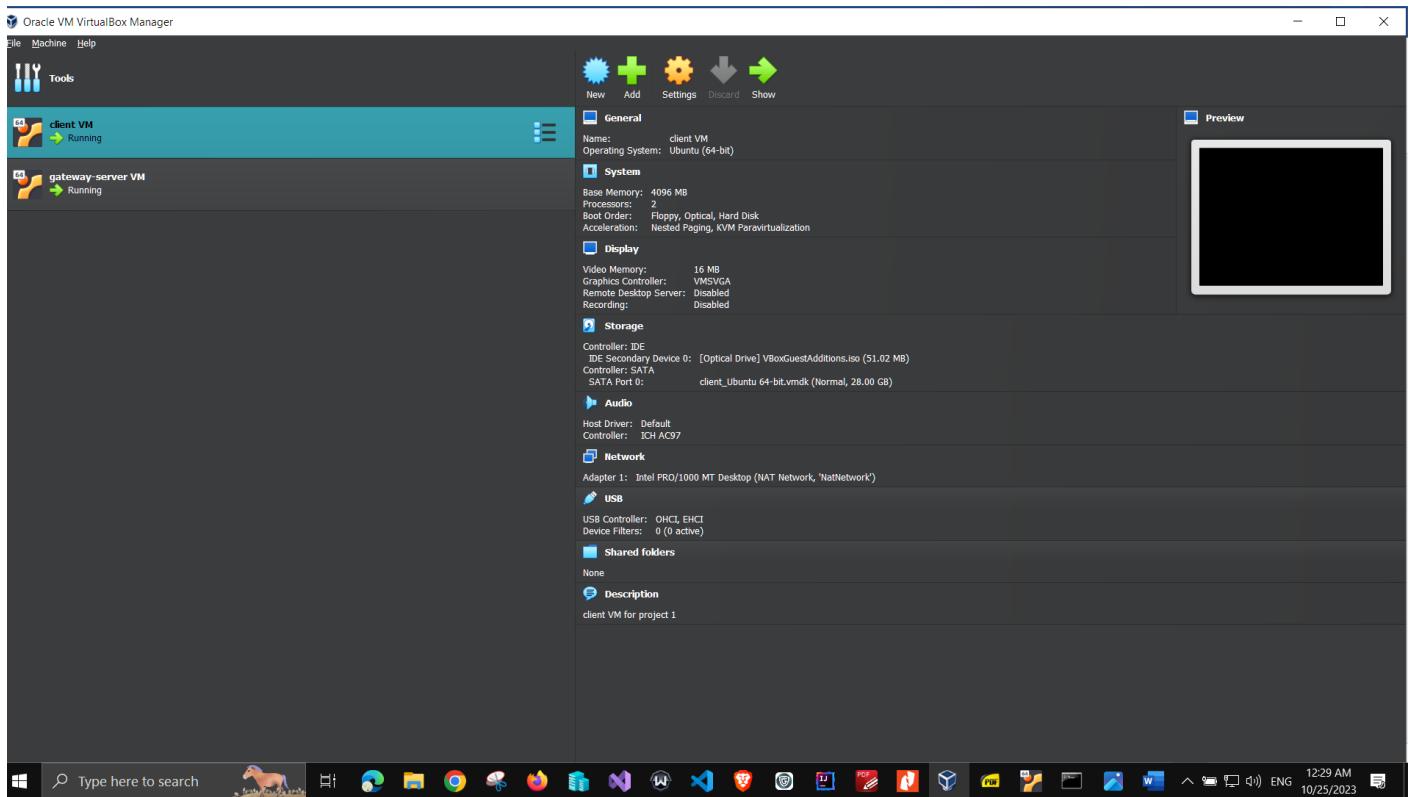
gateway/server IP for NATnetwork1 = 10.0.1.4 , interface (to the internet) = enp0s8



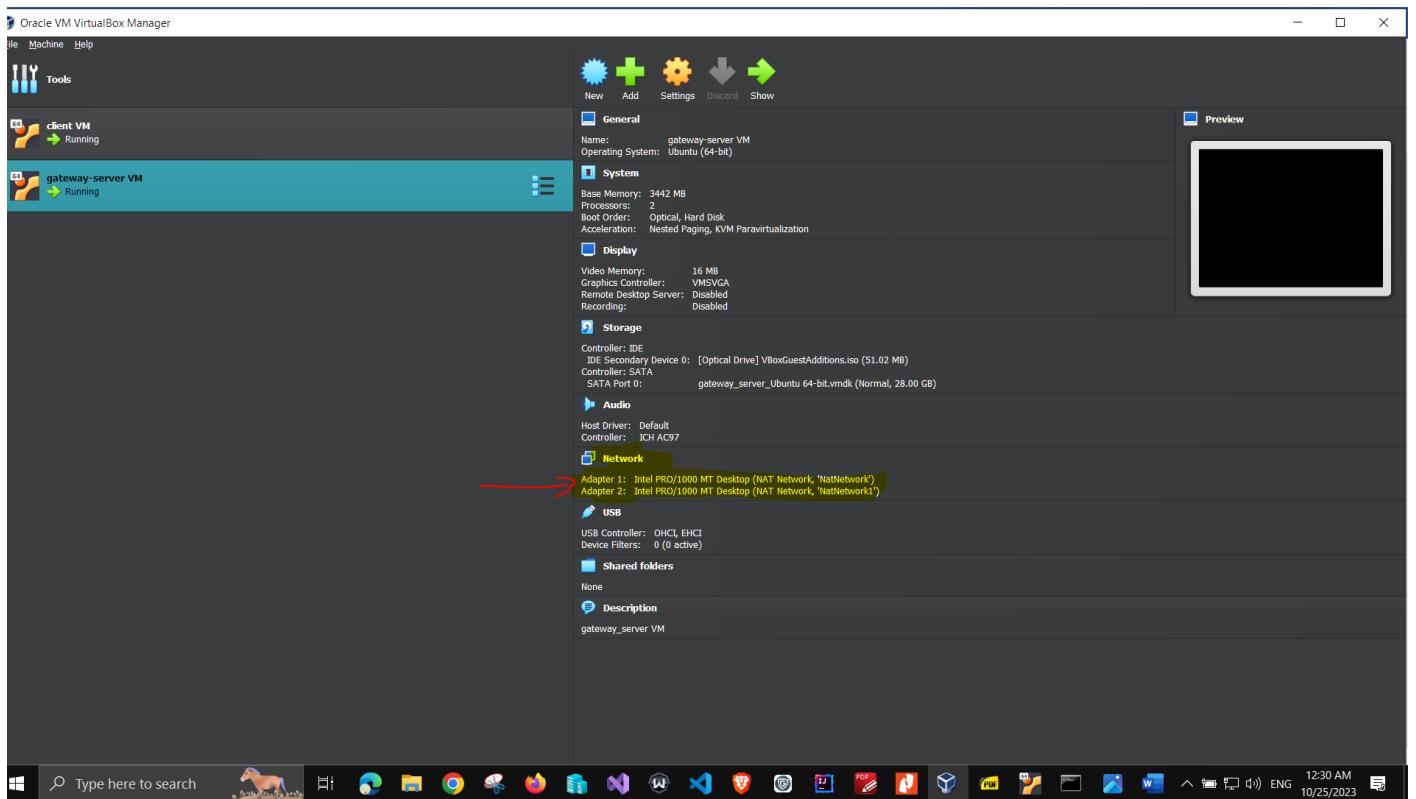
The client VM in virtual box

Please edit the highlighted portion.

2



The server/gateway VM in virtual box



III. SOFTWARE

I have used the following tools to test the connection from client VM to the gateway/server VM so that I make sure that the rules which have been written on the gateway/server VM are working fine.

- Firefox browser (to test connecting with the apache server and view the index.html welcome webpage)
- virtualbox
- Ping

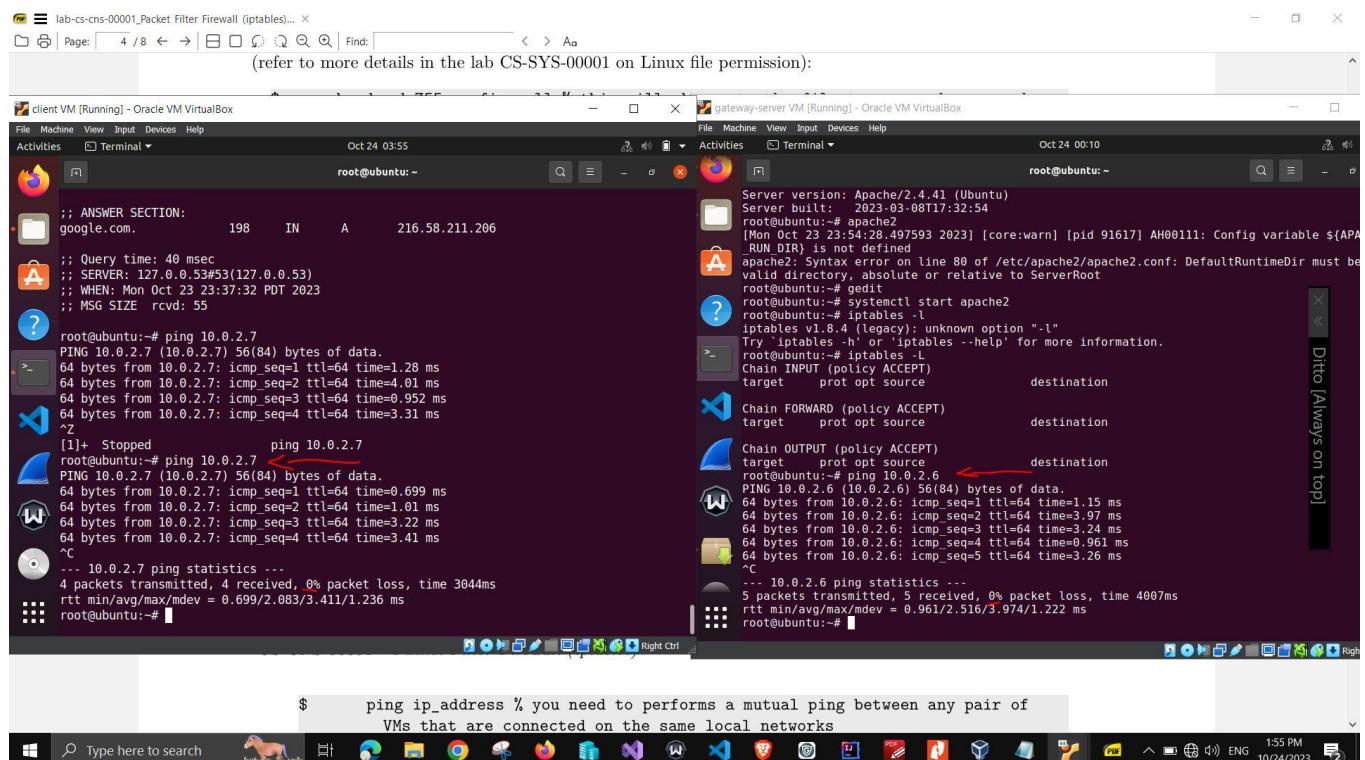
- Traceroute
- Ifconfig
- Gedit (for editing the files)
- Apache2 web server
- Iptables commands
- nmap

IV. PROJECT DESCRIPTION

1. First, I have to prepare the test environment by following and apply all the steps in the provided PDF files and run the 2 virtual machines (VMs) at the same time. Then I assigned “natNetwork” to client VM , and “natNetwork”, “natNetwork1” to the gateway_server VM from the settings for VirtualBox software.
2. I followed the instructions in the PDF file “lab-cs-cns-00001” which describes the rules need to be written using iptables command.
3. The following is a step-by-step screenshots of the work I have done to complete the project with some linux commands I used to define the rules for filter and NAT tables:

task 1.1

task 1.1 each VM can ping each other



task 1.1 each VM can ping each other

```
$ ping ip_address % you need to performs a mutual ping between any pair of
VMs that are connected on the same local network
$ ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=4.01 ms
64 bytes from 10.0.2.7: icmp_seq=3 ttl=64 time=0.952 ms
64 bytes from 10.0.2.7: icmp_seq=4 ttl=64 time=3.31 ms
^Z
[1]+ Stopped ping 10.0.2.7
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=0.699 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 10.0.2.7: icmp_seq=3 ttl=64 time=3.22 ms
64 bytes from 10.0.2.7: icmp_seq=4 ttl=64 time=3.41 ms
^C
--- 10.0.2.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3044ms
rtt min/avg/max/mdev = 0.699/2.083/3.411/1.236 ms
root@ubuntu:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 10.0.2.7 0.0.0.0 UG 0 0 0 enp0s3
0.0.0.0 10.0.2.1 0.0.0.0 UG 29100 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3
169.254.0.0 0.0.0.0 255.255.0.0 U 1000 0 0 enp0s3
root@ubuntu:~#
$ ping ip_address % you need to performs a mutual ping between any pair of
VMs that are connected on the same local network
$ ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
64 bytes from 10.0.2.6: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 10.0.2.6: icmp_seq=2 ttl=64 time=3.97 ms
64 bytes from 10.0.2.6: icmp_seq=3 ttl=64 time=3.24 ms
64 bytes from 10.0.2.6: icmp_seq=4 ttl=64 time=0.961 ms
64 bytes from 10.0.2.6: icmp_seq=5 ttl=64 time=3.26 ms
^C
--- 10.0.2.6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.961/2.516/3.974/1.222 ms
root@ubuntu:~#

```

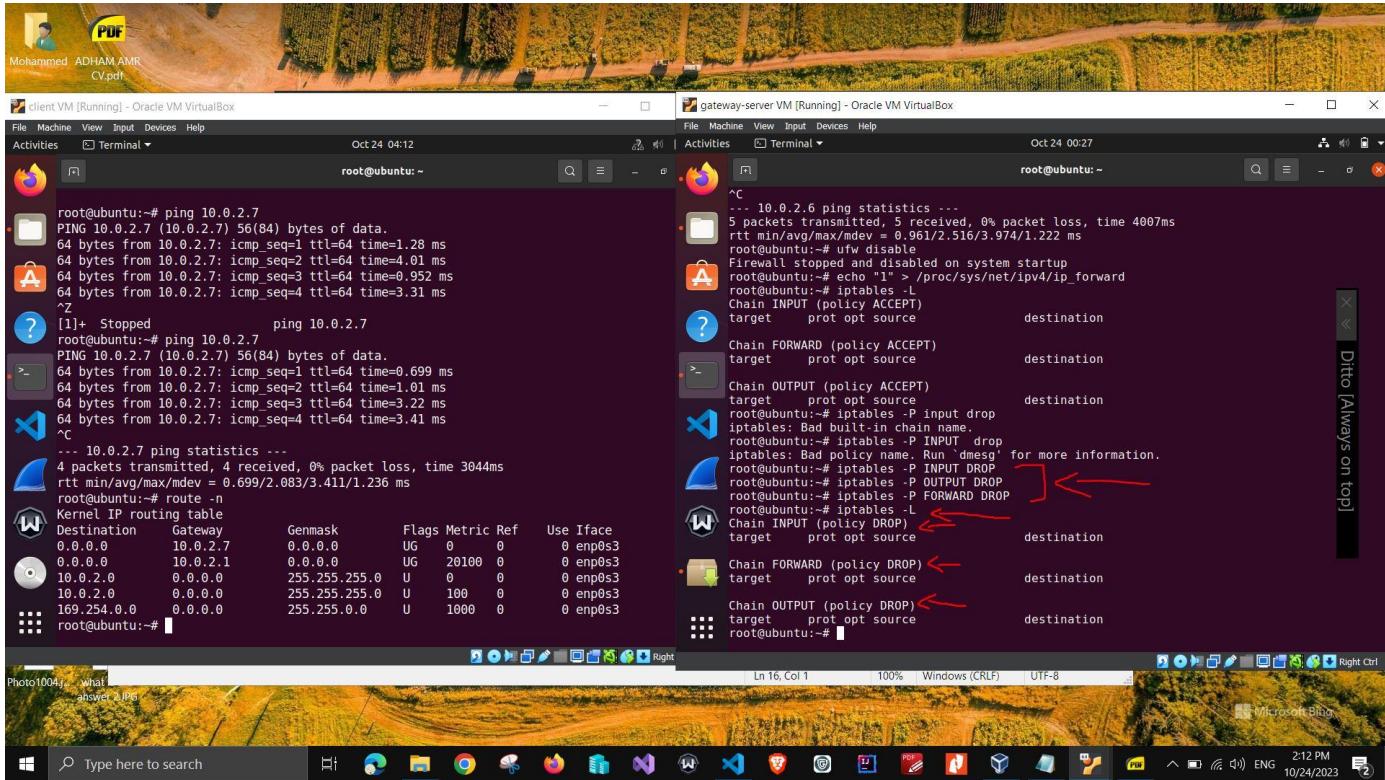
(a) Enable packet forwarding on the gateway

task 1.1 enable packet forwarding on gateway VM

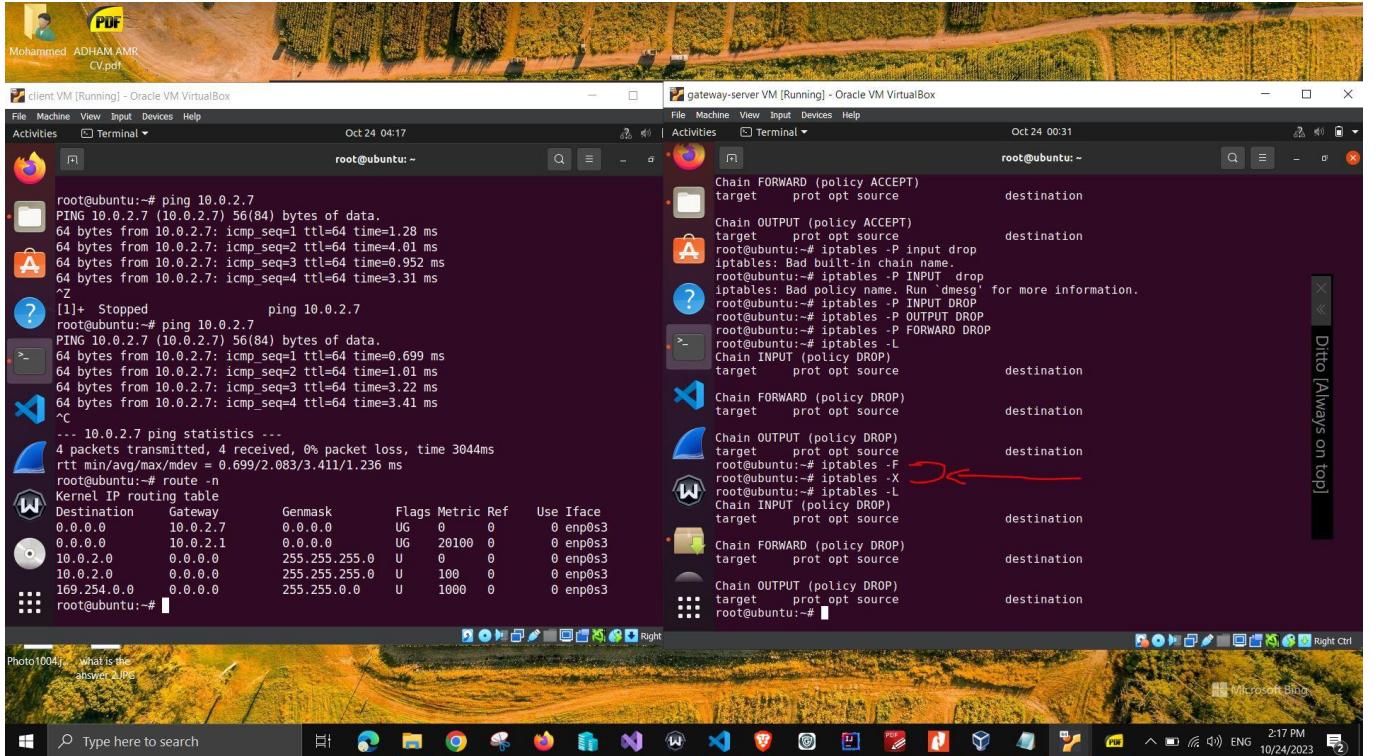
```
$ default gw to the default gw IP though the directly connected interface
$ ping ip_address % you need to performs a mutual ping between any pair of
VMs that are connected on the same local network
$ ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=4.01 ms
64 bytes from 10.0.2.7: icmp_seq=3 ttl=64 time=0.952 ms
64 bytes from 10.0.2.7: icmp_seq=4 ttl=64 time=3.31 ms
^Z
[1]+ Stopped ping 10.0.2.7
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
64 bytes from 10.0.2.7: icmp_seq=1 ttl=64 time=0.699 ms
64 bytes from 10.0.2.7: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 10.0.2.7: icmp_seq=3 ttl=64 time=3.22 ms
64 bytes from 10.0.2.7: icmp_seq=4 ttl=64 time=3.41 ms
^C
--- 10.0.2.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3044ms
rtt min/avg/max/mdev = 0.699/2.083/3.411/1.236 ms
root@ubuntu:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 10.0.2.7 0.0.0.0 UG 0 0 0 enp0s3
0.0.0.0 10.0.2.1 0.0.0.0 UG 29100 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3
169.254.0.0 0.0.0.0 255.255.0.0 U 1000 0 0 enp0s3
root@ubuntu:~#
$ default gw to the default gw IP though the directly connected interface
$ sudo iptables -P INPUT ACCEPT % option -P means default policy
$ ping ip_address % you need to performs a mutual ping between any pair of
VMs that are connected on the same local network
$ ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
64 bytes from 10.0.2.6: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 10.0.2.6: icmp_seq=2 ttl=64 time=3.97 ms
64 bytes from 10.0.2.6: icmp_seq=3 ttl=64 time=3.24 ms
64 bytes from 10.0.2.6: icmp_seq=4 ttl=64 time=0.961 ms
64 bytes from 10.0.2.6: icmp_seq=5 ttl=64 time=3.26 ms
^C
--- 10.0.2.6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.961/2.516/3.974/1.222 ms
root@ubuntu:~# ufw disable
Firewall stopped and disabled on system startup
root@ubuntu:~# echo "1" > /proc/sys/net/ipv4/ip_forward
root@ubuntu:~#

```

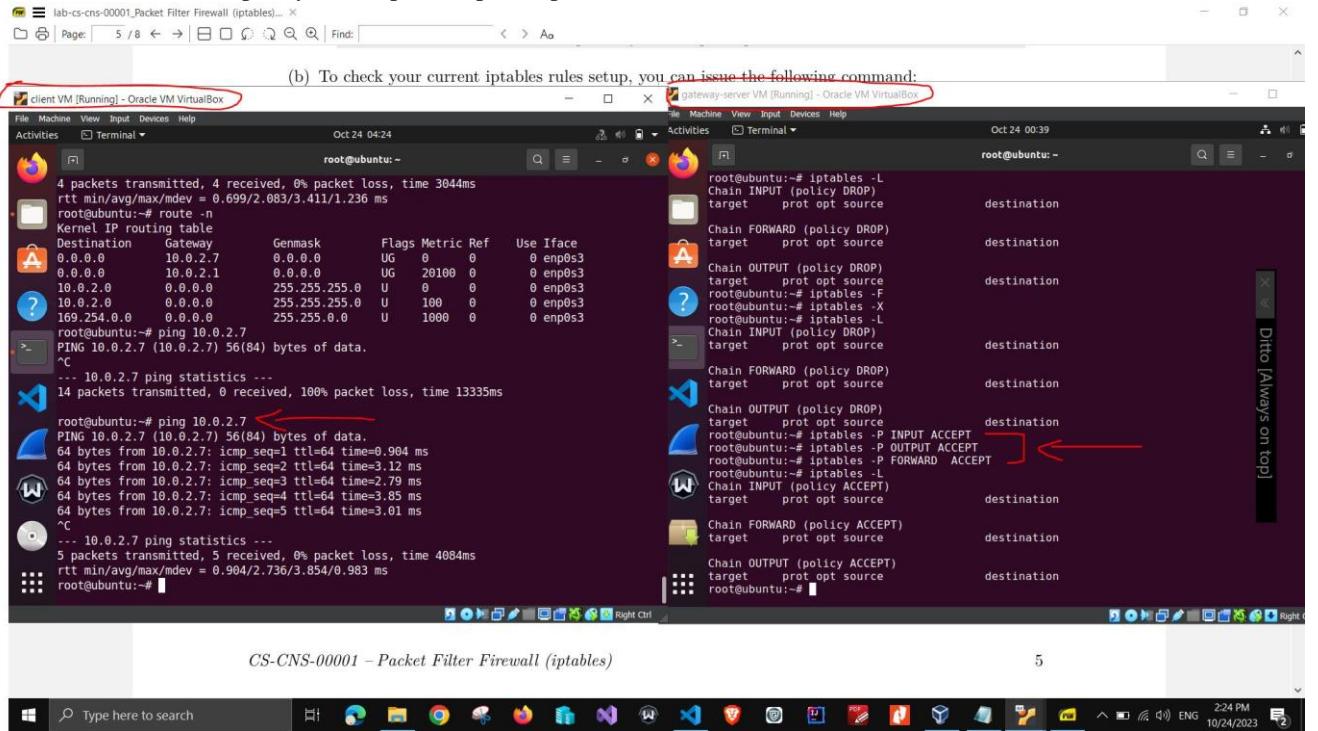
task 1.1.4.b drop policy for input output forward chaines



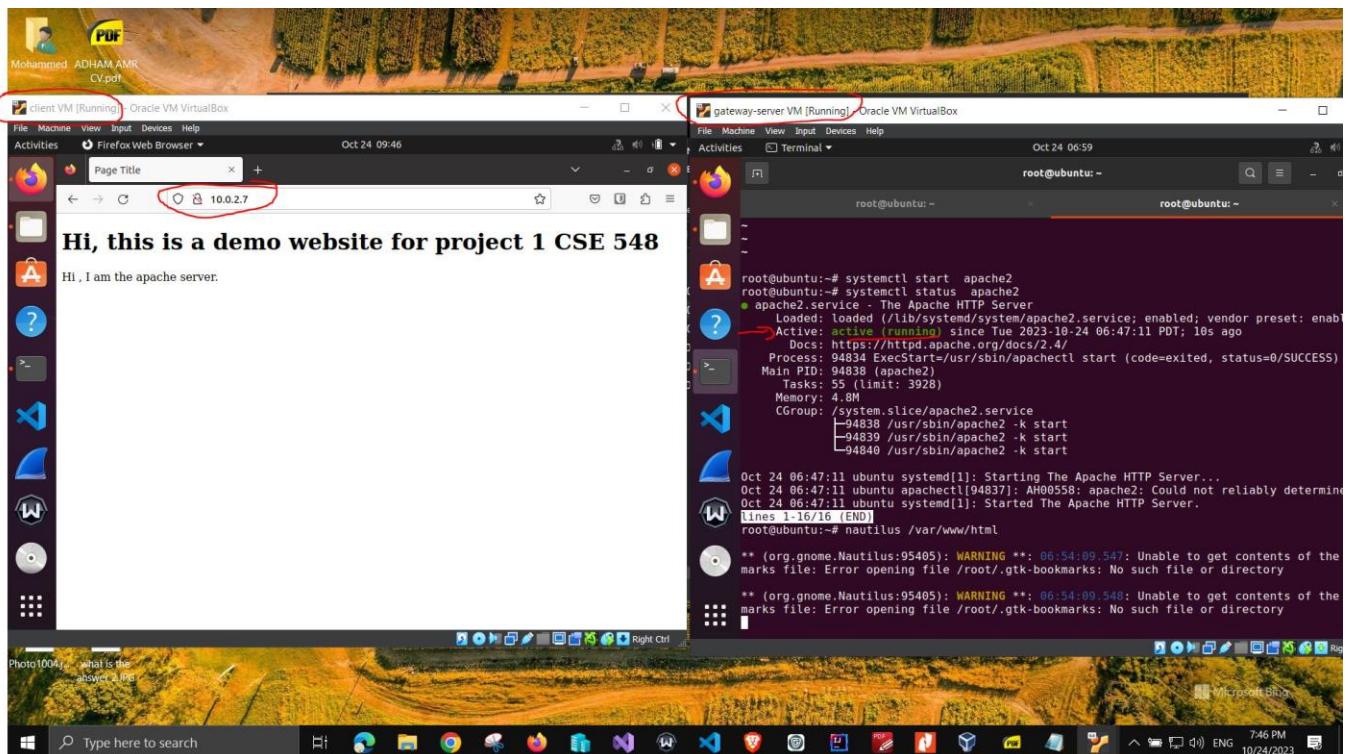
task 1.1.4.c flush all existing chains and delete all user defined chains for iptables



task 1.1.4.d set default policy to accept for input output and forward chains



task 1.2 Apache server is running on gateway VM and client can access the server from the web browser



task 1.3 Reset firewall to whitelist and client VM cannot ping server VM

(a) Test Apache server running the following command:

```

client VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 24 09:53
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
... 10.0.2.7 ping statistics ...
14 packets transmitted, 0 received, 100% packet loss, time 13335ms
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
... 10.0.2.7 ping statistics ...
5 packets transmitted, 5 received, 0% packet loss, time 4084ms
rtt min/avg/max/mdev = 0.904/2.736/3.854/0.983 ms
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
... 10.0.2.7 ping statistics ...
24 packets transmitted, 0 received, 100% packet loss, time 23543ms
root@ubuntu:~#

```

gateway-server VM [Running] - Oracle VM VirtualBox

```

File Machine View Input Devices Help
Activities Terminal Oct 24 07:06
root@ubuntu:~# root@ubuntu:~# root@ubuntu:~#
root@ubuntu:~# CGroup: /system.slice/apache2.service
root@ubuntu:~# -[09762 /usr/sbin/apache2 -k start
root@ubuntu:~# -[09763 /usr/sbin/apache2 -k start
root@ubuntu:~# -[09764 /usr/sbin/apache2 -k start
Oct 23 23:45:55 ubuntu systemd[1]: Starting The Apache HTTP Server...
Oct 23 23:45:55 ubuntu apachectl[90761]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for Port 80
Oct 23 23:45:55 ubuntu systemd[1]: Started The Apache HTTP Server.
root@ubuntu:~# gedit
(gedit:94257): Tepl:WARNING **: 06:35:33.869: GVfs metadata is not supported. Fallback to eplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-data.
^[[B^[[A^C
root@ubuntu:~# iptables -F
root@ubuntu:~# iptables -X
root@ubuntu:~# iptables -P INPUT DROP
root@ubuntu:~# iptables -P FORWARD DROP
root@ubuntu:~# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
Chain FORWARD (policy DROP)
target prot opt source destination
Chain OUTPUT (policy DROP)
target prot opt source destination
root@ubuntu:~#

```

task 1.3 Reset firewall to whitelist and server VM cannot ping client VM

(a) Test Apache server running the following command:

```

client VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 24 09:55
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
... 10.0.2.7 ping statistics ...
14 packets transmitted, 0 received, 100% packet loss, time 13335ms
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
... 10.0.2.7 ping statistics ...
5 packets transmitted, 5 received, 0% packet loss, time 4084ms
rtt min/avg/max/mdev = 0.904/2.736/3.854/0.983 ms
root@ubuntu:~# ping 10.0.2.7
PING 10.0.2.7 (10.0.2.7) 56(84) bytes of data.
^C
... 10.0.2.7 ping statistics ...
24 packets transmitted, 0 received, 100% packet loss, time 23543ms
root@ubuntu:~#

```

gateway-server VM [Running] - Oracle VM VirtualBox

```

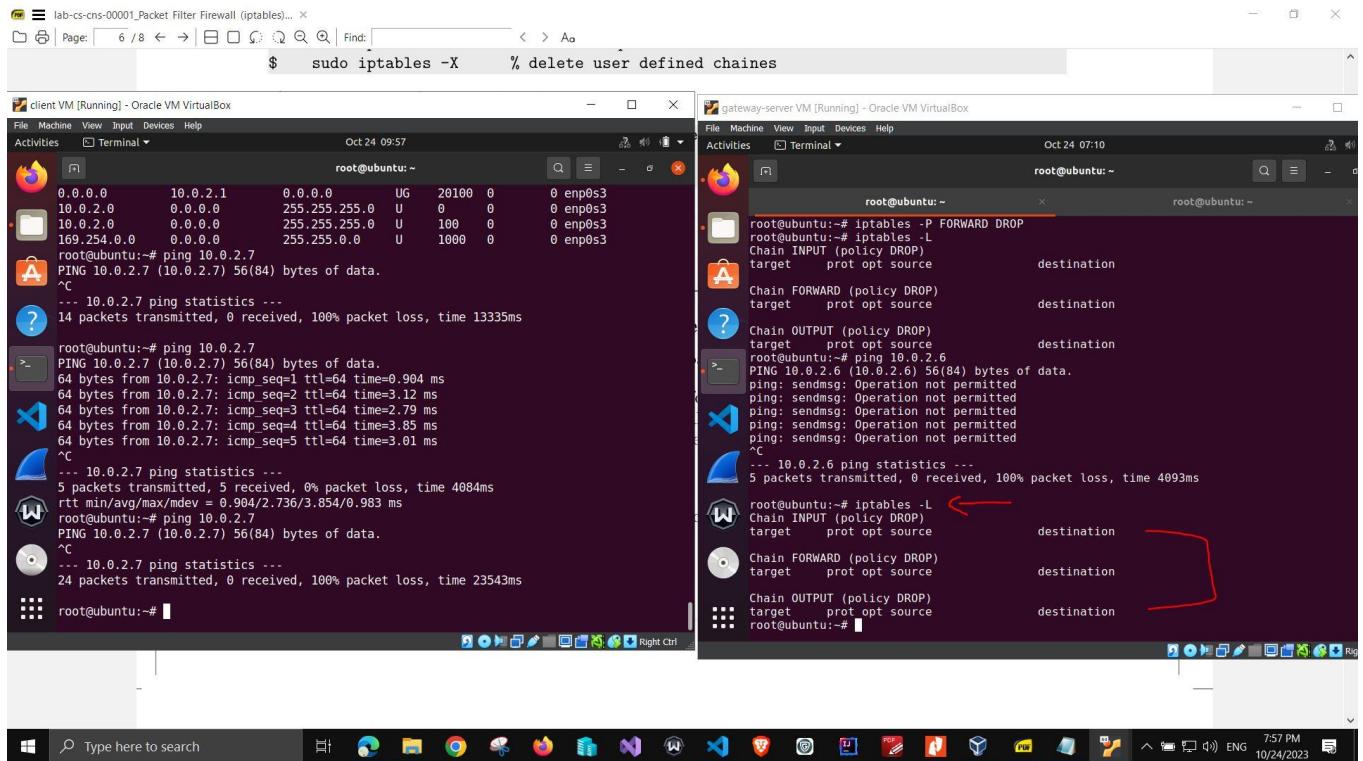
File Machine View Input Devices Help
Activities Terminal Oct 24 07:08
root@ubuntu:~# root@ubuntu:~# root@ubuntu:~#
root@ubuntu:~# (gedit:94257): Tepl:WARNING **: 06:35:33.869: GVfs metadata is not supported. Fallback to eplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-data.
^[[B^[[A^C
root@ubuntu:~# iptables -F
root@ubuntu:~# iptables -X
root@ubuntu:~# iptables -P INPUT DROP
root@ubuntu:~# iptables -P FORWARD DROP
root@ubuntu:~# iptables -L
Chain INPUT (policy DROP)
target prot opt source destination
Chain FORWARD (policy DROP)
target prot opt source destination
Chain OUTPUT (policy DROP)
target prot opt source destination
root@ubuntu:~# ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
^C
... 10.0.2.6 ping statistics ...
5 packets transmitted, 0 received, 100% packet loss, time 4093ms
root@ubuntu:~#

```

task 2.1 allow whitelist (drop for input output and forward chain) on server_gateway VM

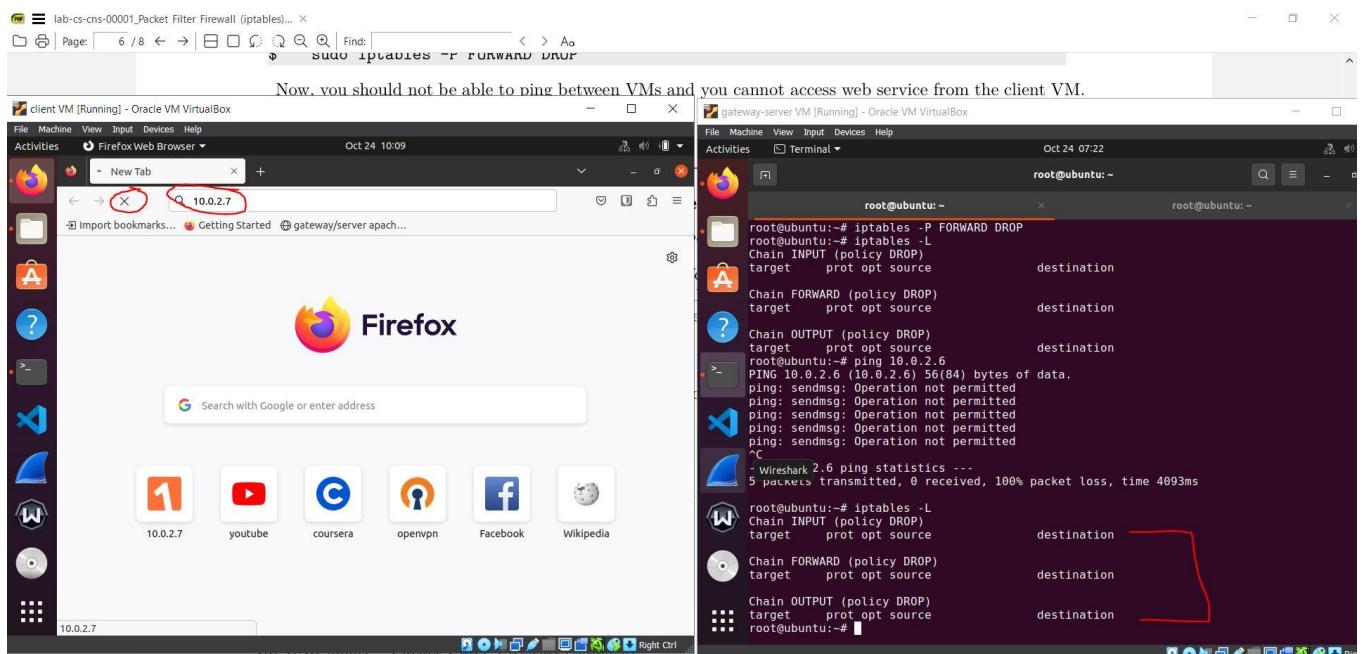
Please edit the highlighted portion.

8



task 2.1 after allow whitelist on server_gateway VM , client VM cannot access apache web server web page fom the browser

in the following screenshot, the page was loading and the Apache welcome webpage has not loaded as the client VM is blocked from accessing the server_gateway VM



3. Stop the client from pinging the Gateway/Server VM's IP address.



task 2.2 define the rule to allow client access the server apache on server VM

```

root@ubuntu:~ # iptables -A OUTPUT -p TCP --sport 80 -o enp0s3 -s 10.0.2.7 -d 10.0.2.6 -j ACCEPT
root@ubuntu:~ # iptables -A INPUT -p TCP --dport 80 -i enp0s3 -s 10.0.2.6 -j ACCEPT
root@ubuntu:~ # iptables -L
Chain INPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
Chain FORWARD (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
Chain OUTPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.7        10.0.2.6     tcp spt:http
root@ubuntu:~ # iptables -A INPUT -p TCP --dport 80 -i enp0s3 -s 10.0.2.6 -j ACCEPT
root@ubuntu:~ # iptables -L
Chain INPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
Chain FORWARD (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.7        10.0.2.6     tcp spt:http
root@ubuntu:~ # iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target  prot opt  source          destination
1   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
2   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
Chain FORWARD (policy DROP)
num  target  prot opt  source          destination
1   ACCEPT  tcp  --  10.0.2.7        10.0.2.6     tcp spt:http
root@ubuntu:~ # 

```

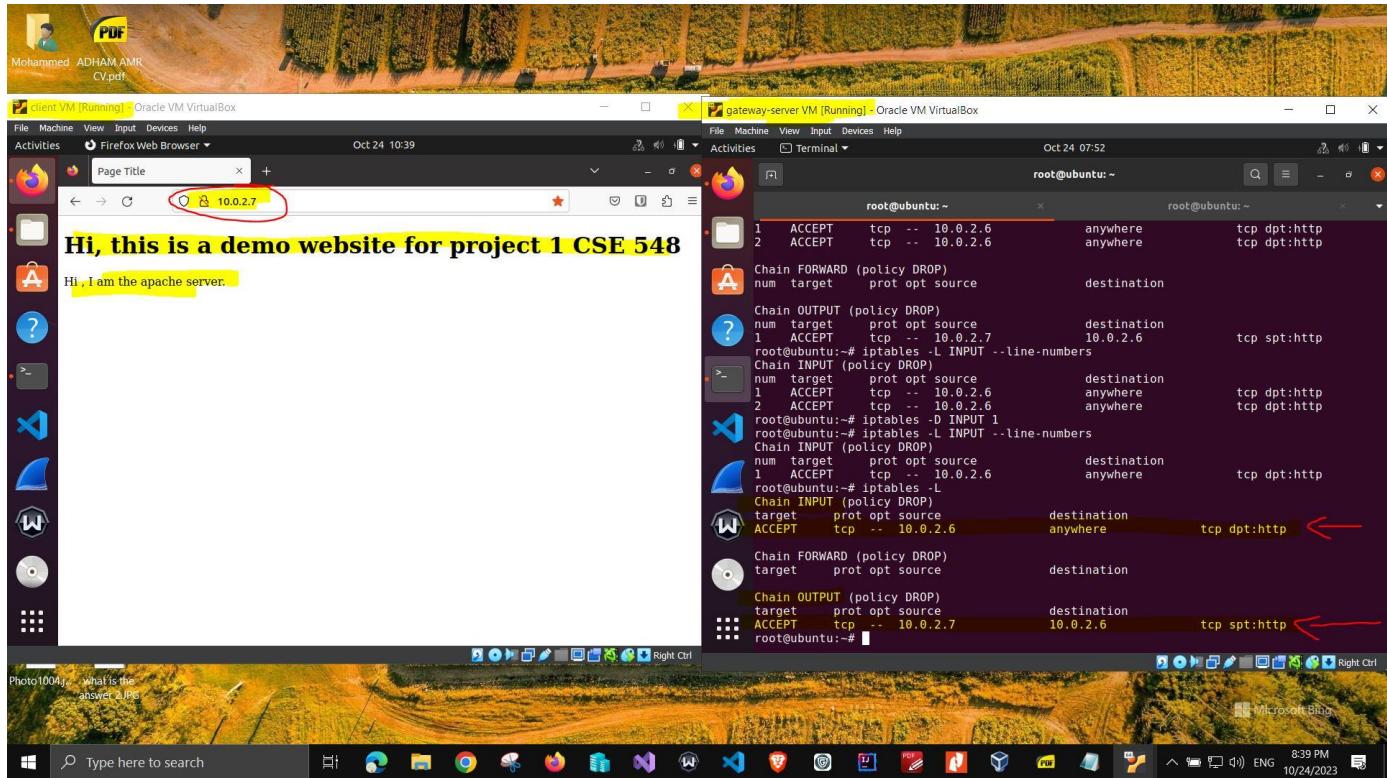
task 2.2 view the rule to allow client access the server apache on server VM

```

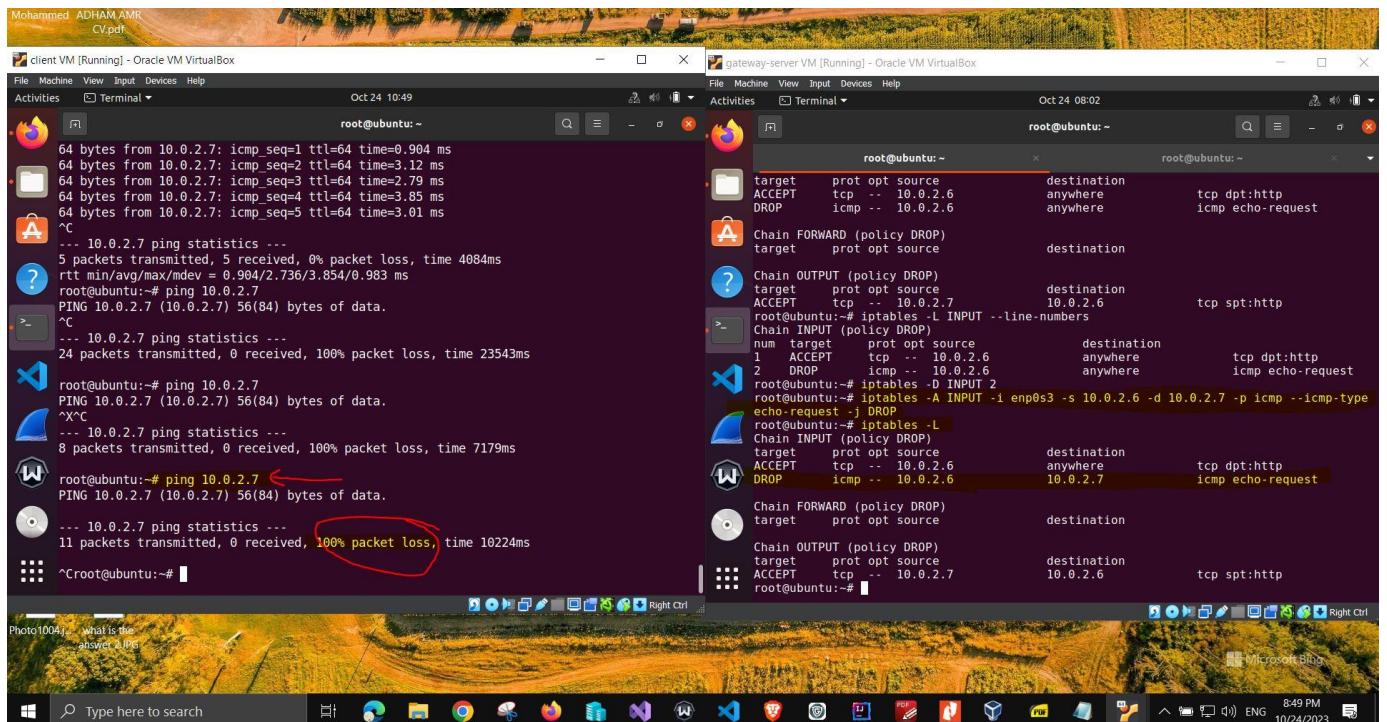
root@ubuntu:~ # iptables -A INPUT -p TCP --dport 80 -i enp0s3 -s 10.0.2.6 -j ACCEPT
root@ubuntu:~ # iptables -A INPUT -p TCP --dport 80 -i enp0s3 -s 10.0.2.7 -j ACCEPT
root@ubuntu:~ # iptables -L
Chain INPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
ACCEPT  tcp  --  10.0.2.7        anywhere    tcp dpt:http
Chain FORWARD (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.7        10.0.2.6     tcp spt:http
root@ubuntu:~ # iptables -L --line-numbers
Chain INPUT (policy DROP)
num  target  prot opt  source          destination
1   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
2   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
Chain FORWARD (policy DROP)
num  target  prot opt  source          destination
1   ACCEPT  tcp  --  10.0.2.7        10.0.2.6     tcp spt:http
root@ubuntu:~ # iptables -L INPUT --line-numbers
Chain INPUT (policy DROP)
num  target  prot opt  source          destination
1   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
2   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
root@ubuntu:~ # iptables -D INPUT 1
root@ubuntu:~ # iptables -L INPUT --line-numbers
Chain INPUT (policy DROP)
num  target  prot opt  source          destination
1   ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
root@ubuntu:~ # iptables -L
Chain INPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.6        anywhere    tcp dpt:http
Chain FORWARD (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.7        10.0.2.6     tcp spt:http
root@ubuntu:~ # 

```

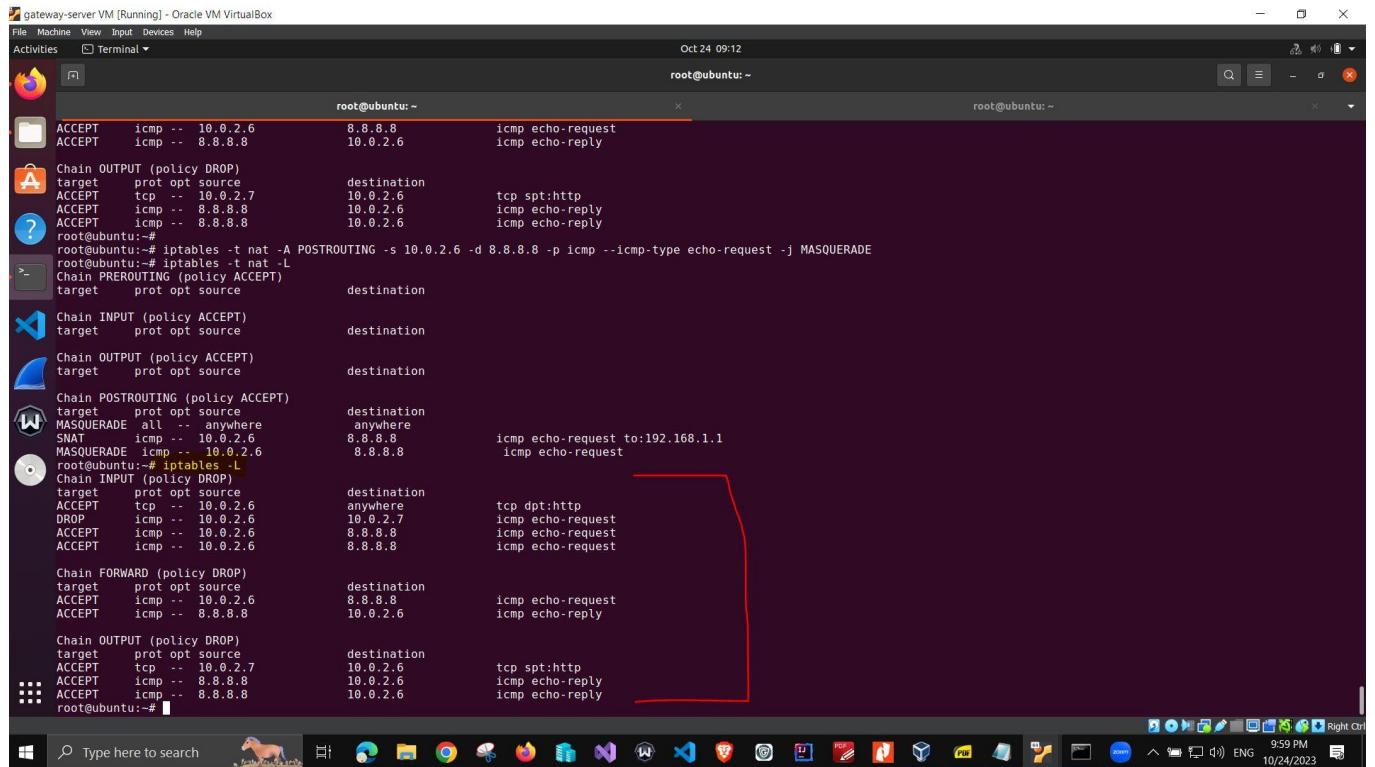
task 2.2 client can access the server apache on server VM and welcome webpage is viewed



task 2.3 Stop the client from pinging the Gateway_Server VM IP address



task 2.3 Stop the client from pinging the Gateway_Server VM IP address _iptables rules for filter table



```

gateway-server VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 24 09:12
root@ubuntu: ~
root@ubuntu: ~
ACCEPT icmp -- 10.0.2.6 8.8.8     icmp echo-request
ACCEPT icmp -- 8.8.8.8   10.0.2.6     icmp echo-reply

Chain OUTPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.7    10.0.2.6      tcp spt:http
ACCEPT  icmp --  8.8.8.8    10.0.2.6      icmp echo-reply
ACCEPT  icmp --  8.8.8.8    10.0.2.6      icmp echo-reply

root@ubuntu:~# iptables -t nat -A POSTROUTING -s 10.0.2.6 -d 8.8.8.8 -p icmp --icmp-type echo-request -j MASQUERADE
root@ubuntu:~# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target  prot opt source          destination

Chain INPUT (policy ACCEPT)
target  prot opt source          destination

Chain OUTPUT (policy ACCEPT)
target  prot opt source          destination

Chain POSTROUTING (policy ACCEPT)
target  prot opt source          destination
MASQUERADE all  --  anywhere    anywhere
SNAT    icmp --  10.0.2.6    8.8.8.8      icmp echo-request to:192.168.1.1
MASQUERADE icmp --  10.0.2.6    8.8.8.8      icmp echo-request
root@ubuntu:~# iptables -L
Chain INPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.6    anywhere    tcp dpt:http
DROP    icmp --  10.0.2.6    8.8.8.8      icmp echo-request
ACCEPT  icmp --  10.0.2.6    8.8.8.8      icmp echo-request
ACCEPT  icmp --  10.0.2.6    8.8.8.8      icmp echo-request

Chain FORWARD (policy DROP)
target  prot opt source          destination
ACCEPT  icmp --  10.0.2.6    8.8.8.8      icmp echo-request
ACCEPT  icmp --  8.8.8.8    10.0.2.6      icmp echo-reply

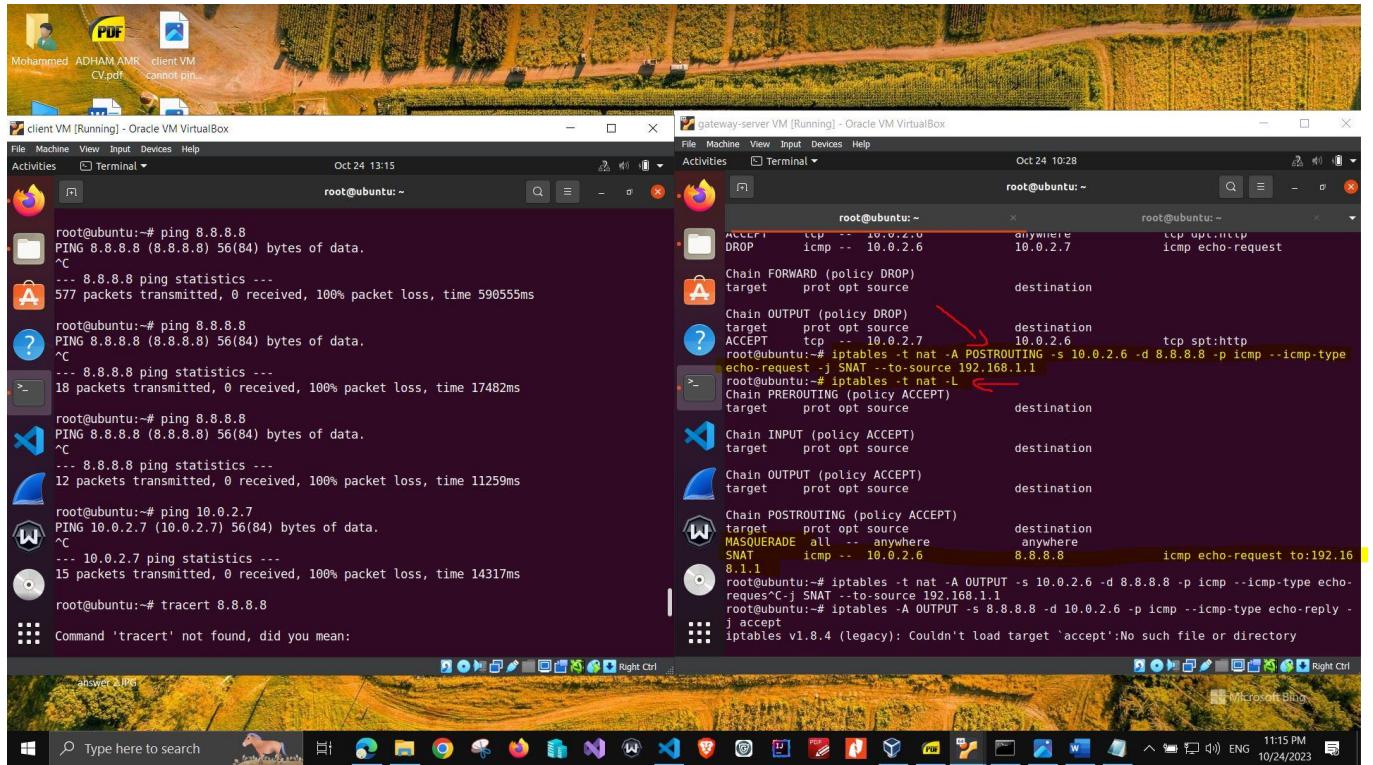
Chain OUTPUTPUT (policy DROP)
target  prot opt source          destination
ACCEPT  tcp  --  10.0.2.7    10.0.2.6      tcp spt:http
ACCEPT  icmp --  8.8.8.8    10.0.2.6      icmp echo-reply
ACCEPT  icmp --  8.8.8.8    10.0.2.6      icmp echo-reply
root@ubuntu:~#

```

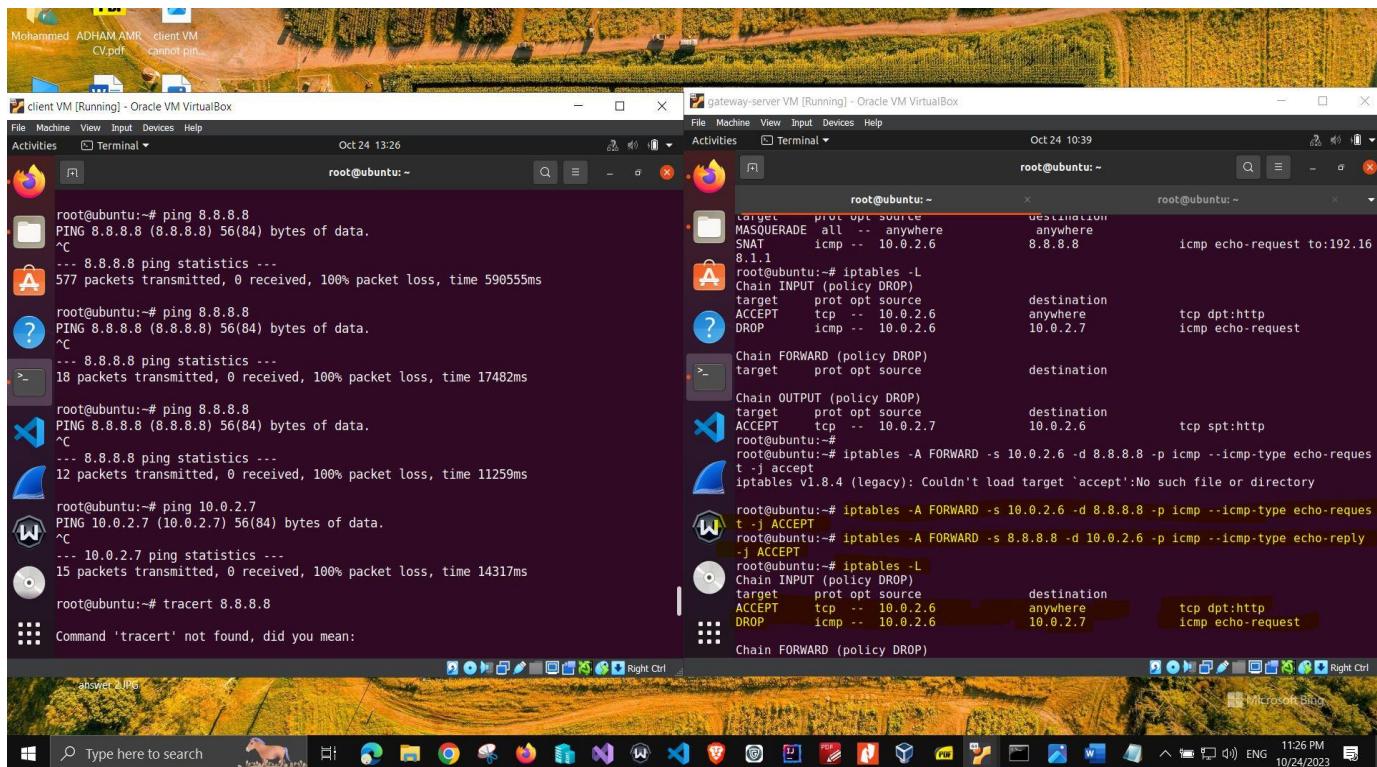
task 2.4 Allow the client to ping 8.8.8.8 (an public IP address on internet)_iptables rule for NAT table for POSTROUTING chain

note: the client VM cannot ping 8.8.8.8 , the TA reviewed my rules for the NAT table and it seems everything is correct, but the issue still appear. I posted this on the discussion website (Ed) website and also I added another rules for the filter table for INPUT and OUTPUT chain to allow ICMP request and ICMP response from client IP = 10.0.2.6 to 8.8.8.8 , unfortunately client VM still cannot ping 8.8.8.8

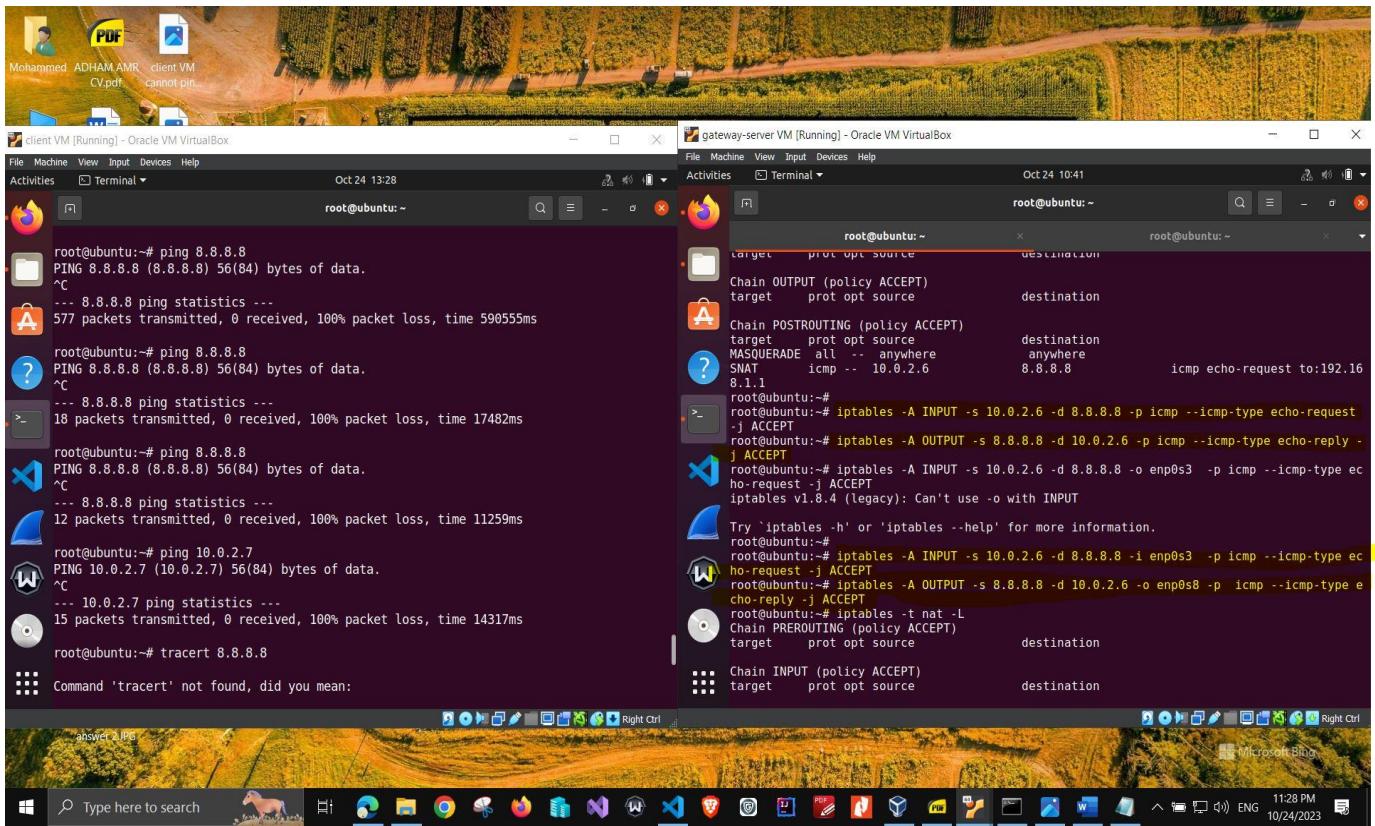
also I added a rule for FORWARD chain for filter table , and added a rule for INPUT and OUTPUT chains for filter table, but still client VM still cannot ping 8.8.8.8



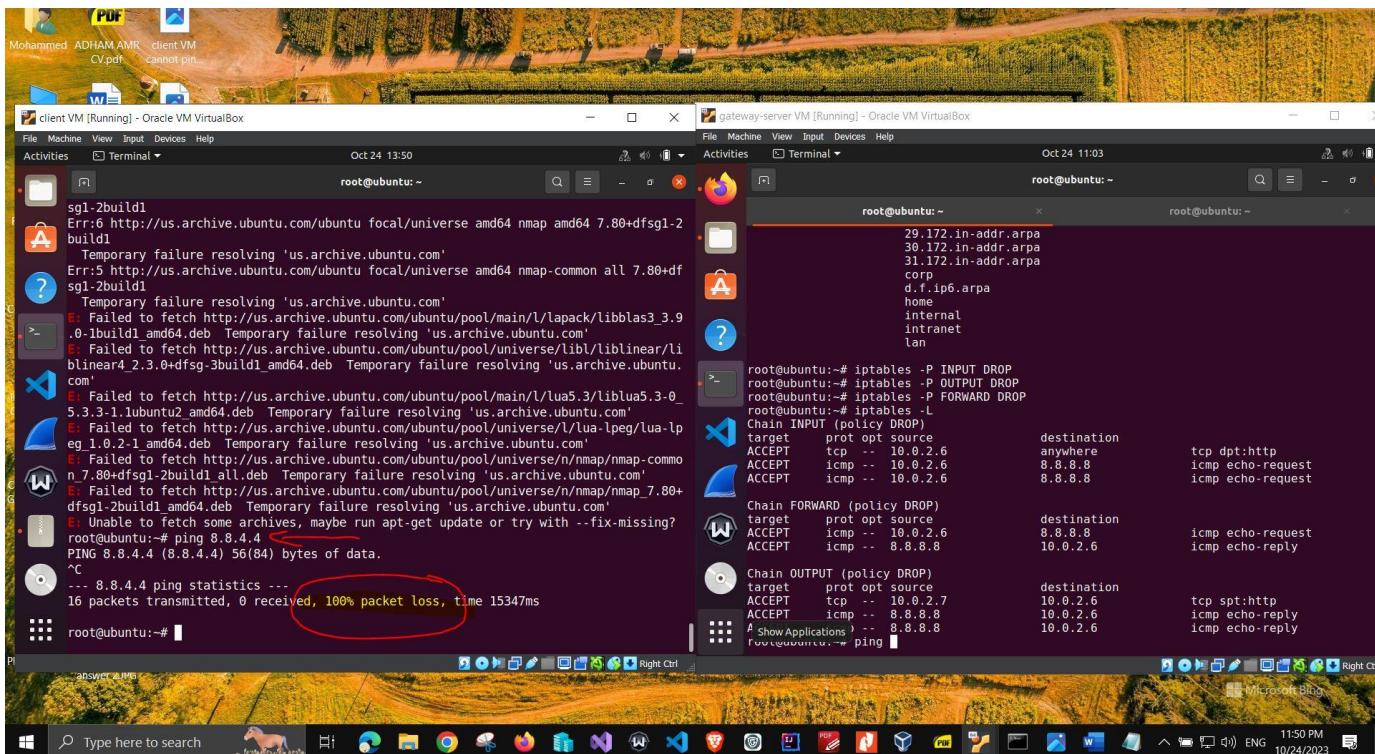
task 2.4 Allow the client to ping 8.8.8.8 (an public IP address on internet)_iptables rule for FORWARD chain for filter table



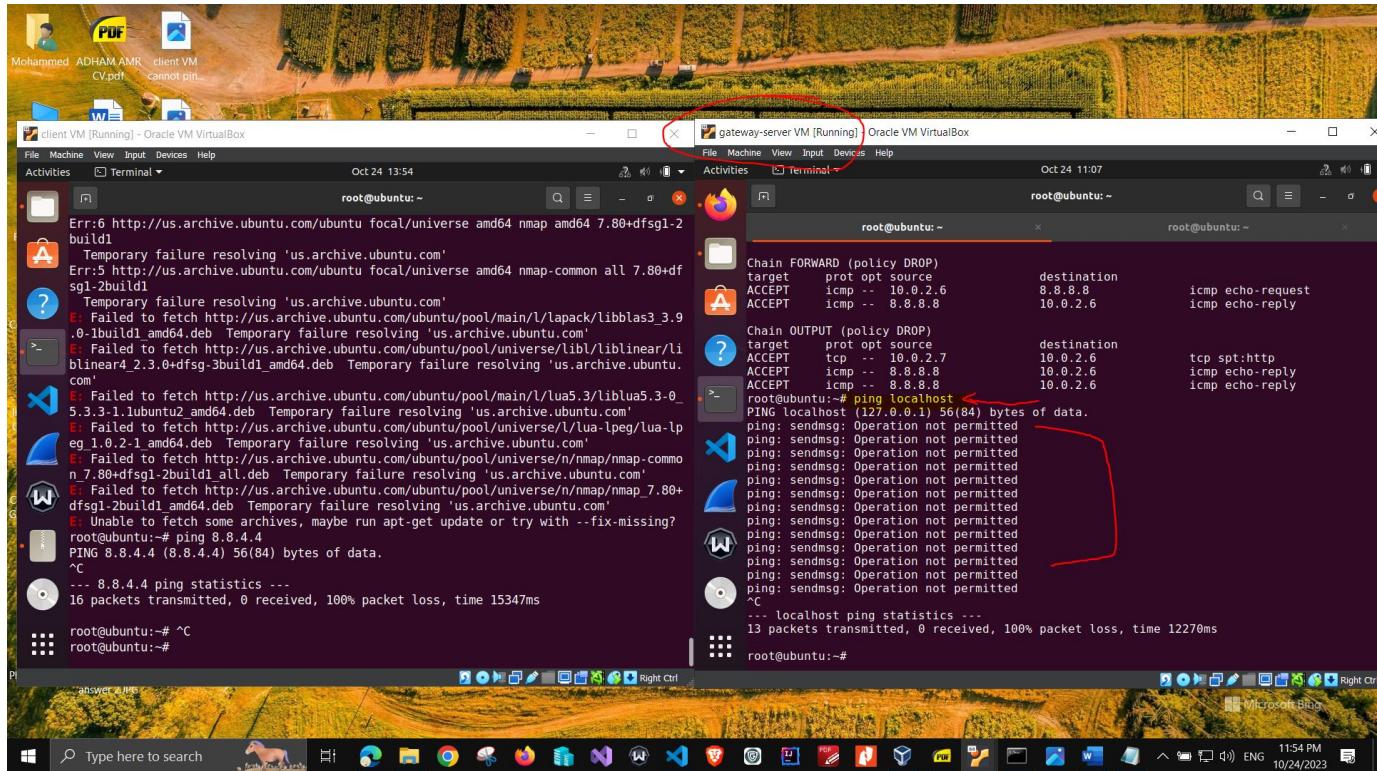
task 2.4 Allow the client to ping 8.8.8.8 (an public IP address on internet)_iptables rule for INPUT and OUTPUT chains for filter table.JPG



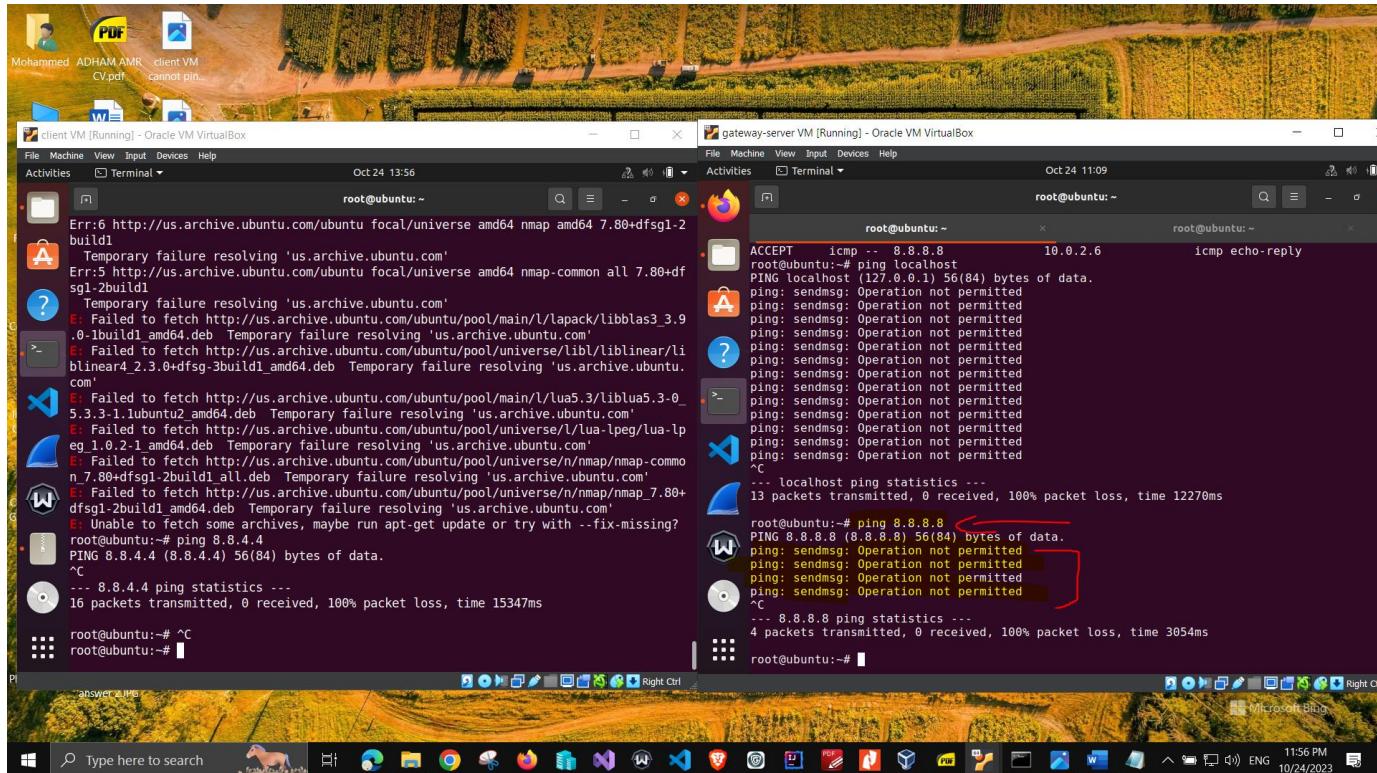
task 2.4 the client cannot ping 8.8.4.4 (an public IP address on internet)



task 2.4 cannot ping localhost on gateway_server VM

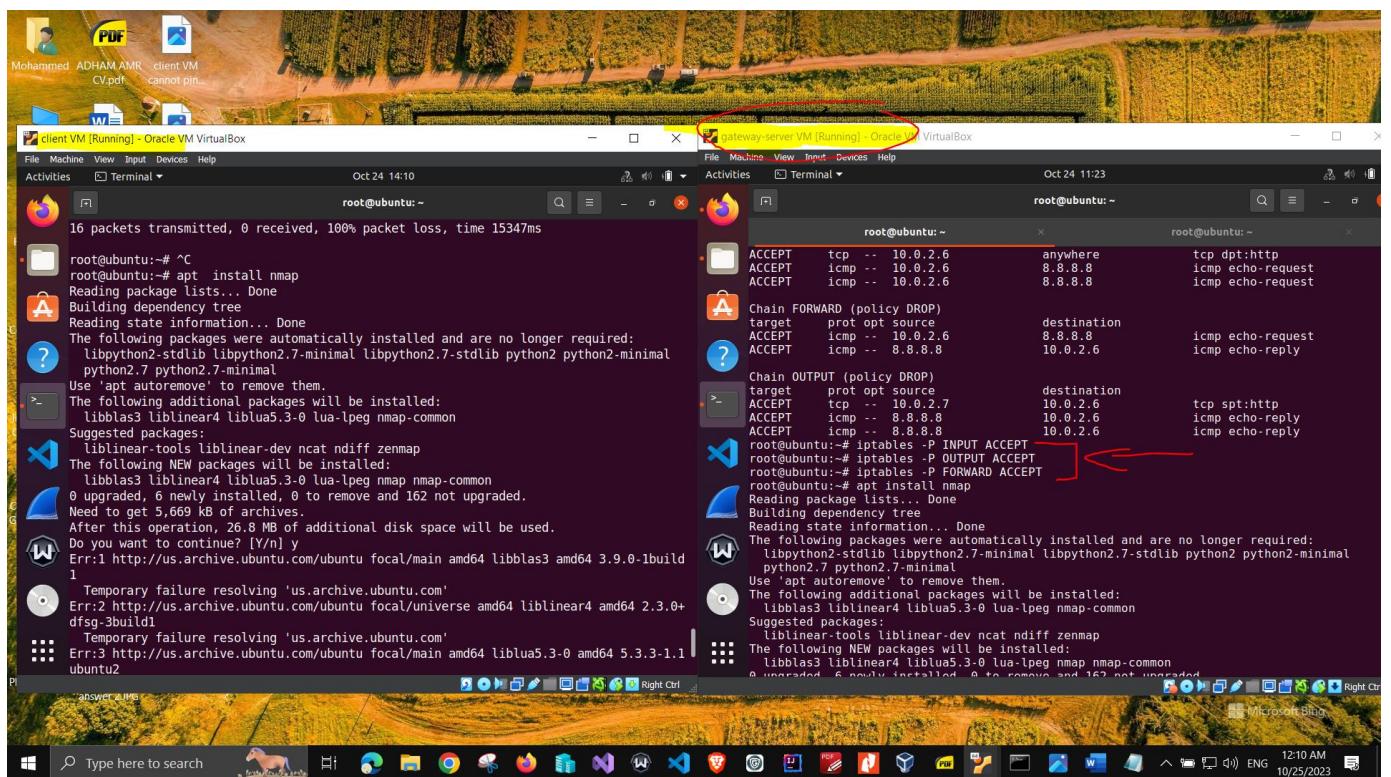
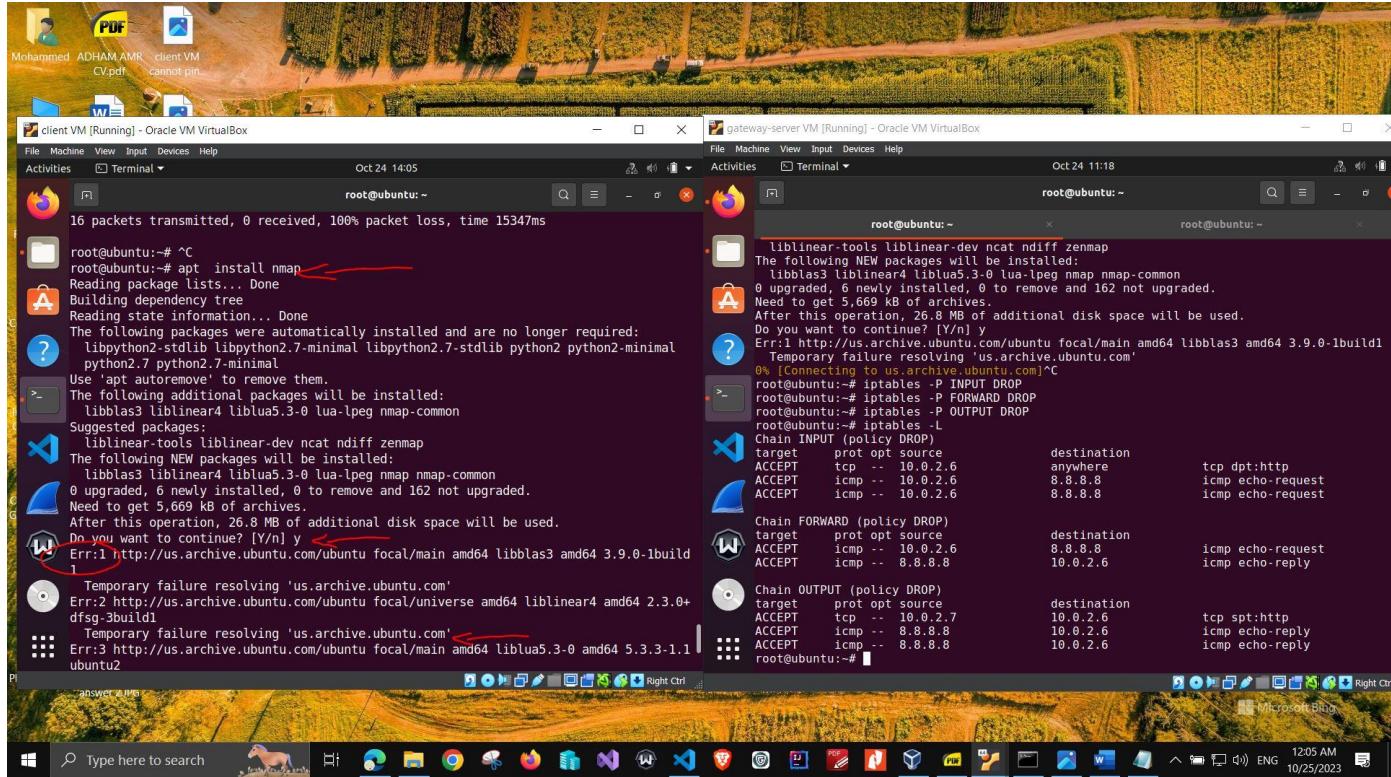


task 2.4 cannot ping localhost on gateway_server VM



Note: I failed to install NMAP tool on client VM after I SET the rules for filter table although I set to ACCEPT for INPUT , OUTPUT, and FORWARD chain for filter table , I tried to install NMAP tool offline and it failed, so I could not run the following commands on client VM:

sudo nmap -sT -p- 10.0.2.x % x is the value of your Gateway/Server VM's IP address
 sudo nmap -sU -p- 10.0.2.x % x is the value of your Gateway/Server VM's IP address



V. CONCLUSION

I have learned many useful and practical linux commands to filter the network packets , this helped me to discover the iptables command and run 2 virtual machines simultaneously so that I have a home lab to simulate 2 machines on the same NAT network. I really liked the instructions provided because it is easy to follow with and apply hands-on experience.

VI. APPENDIX B: ATTACHED FILES

I have not used configuration file to run the iptables commands, I used the terminal on the virtual machine directly and I asked the professor on the zoom live session on 22-10-2023 and she said it is fine to do this .

I uploaded a text file on GitHub that contains a draft for the commands I used and my process of thinking of meeting the required tasks.

<https://github.com/Mohammed-Ragab/CSE-548-advanced-network-security/blob/main/project%201%20draft%20and%20notes.txt>

References

Reference is optional, but nice to have to allow others to read your report with additional linked source for validation and learning.

- (n.d.). Retrieved from [https://linux-training.be/networking/ch14.html#:~:text=The%20filter%20table%20in%20iptables,\(routed\)%20through%20the%20system](https://linux-training.be/networking/ch14.html#:~:text=The%20filter%20table%20in%20iptables,(routed)%20through%20the%20system)
- (n.d.). Retrieved from YOUTUBE: <https://www.youtube.com/watch?v=H1WPwAjMXRo>