PACKET FILTER FIREWALL (iptables)

Student Name: Nana Ama Ahenkan Boakye-Ansah

Email: nboakyea@asu.edu

Submission Date: May 9, 2024

Class Name and Term: CSE548 Summer 2024

I. Project Overview

In this project lab, we want to set up an environment using two virtual machines running a Linux OS to learn how packet filter works. One machine is Client, and the other is a Gateway virtual machine. The Gateway has two Nat networks, and it can reach external networks on an interface and the Client has only one Nat Network which can use the Gateway to reach out to other networks. The iptables firewall will be used when configuring the gateway and NAT selected protocols will be enabled. A web server on the Gateway will be set up and a test-and-demo web page will be used. The Client should be able to access the webserver on the Gateway

At the end, we will not be able to ping the Gateway/Server VM IP address. Also, the Client can ping 8.8.8.8. We will want to control network traffic, allowing only particular protocols to a particular destination.

II. Network Setup

The diagram below shows show the topology of my network set up and how I did configuration on a Virtual Box Machine, I first installed some useful tools on the Linux machine so that my configuration will run smoothly.

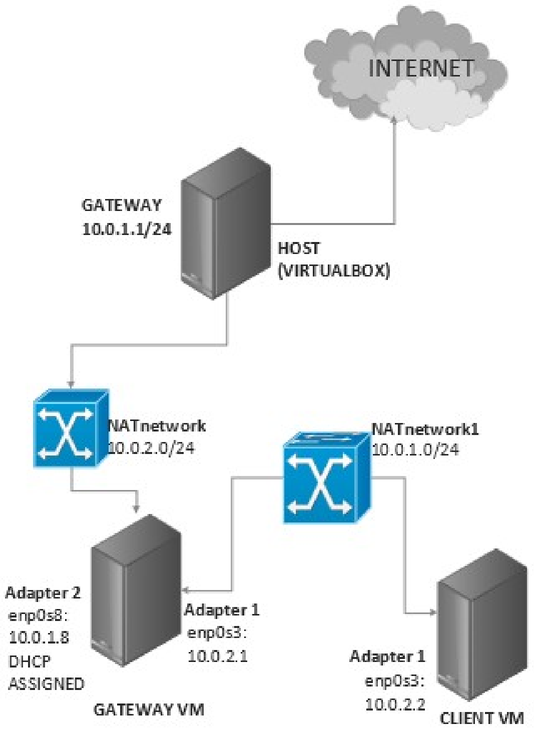
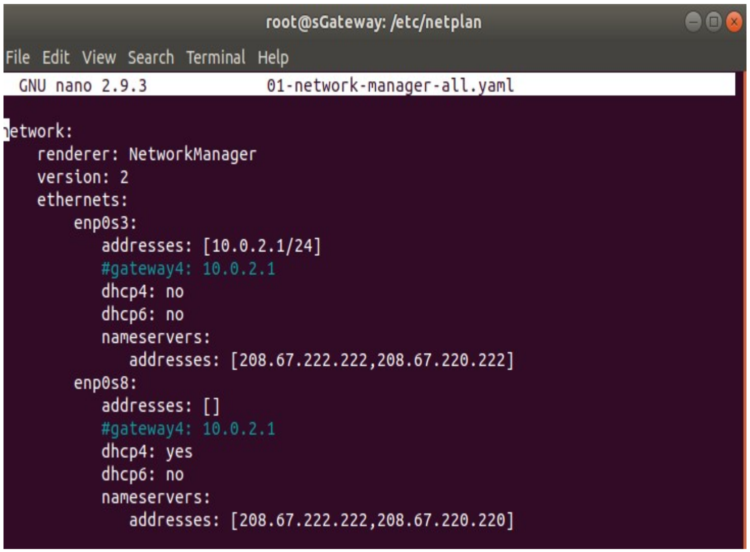


Figure 1- Network Set-up

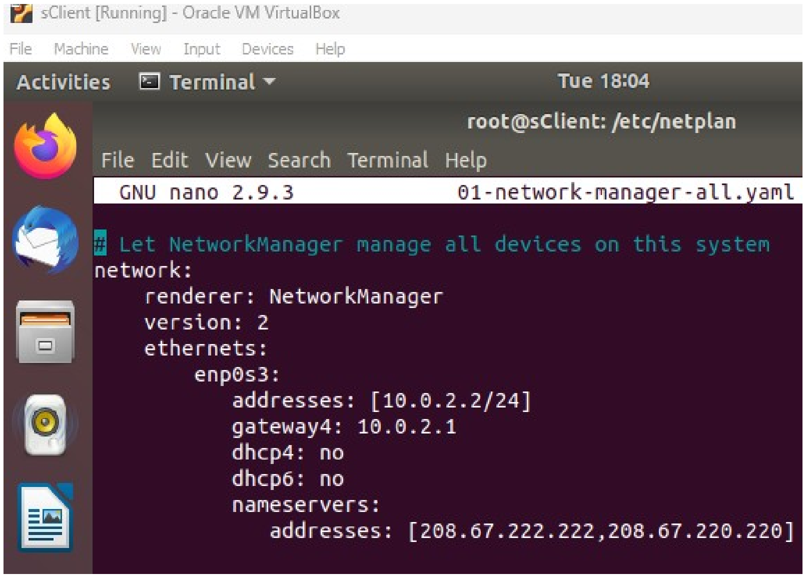
Initially, when I used "ifconfig” to check the interfaces, it had already be assigned using the DHCP Server provided by the VM, so I had to set up addresses manually to both Gateway and the Client and also let the Gateway VM use the default gateway and where DHCP addresses are required. The way I configured the network interfaces was through the netplan through /etc/netplan for both Gateway and Client VM's


root@sGateway: /etc/netplan
File Edit View Search Terminal Help

```
 GNU nano 2.9.3              01-network-manager-all.yaml

network:
   renderer: NetworkManager
   version: 2
   ethernets:
       enp0s3:
          addresses: [10.0.2.1/24]
          #gateway4: 10.0.2.1
          dhcp4: no
          dhcp6: no
          nameservers:
             addresses: [208.67.222.222,208.67.220.222]
       enp0s8:
          addresses: []
          #gateway4: 10.0.2.1
          dhcp4: yes
          dhcp6: no
          nameservers:
             addresses: [208.67.222.222,208.67.220.220]
```

Figure 1- Gateway Configurations

Figure 3-Gateway Configurations


sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 18:04
root@sClient: /etc/netplan
File Edit View Search Terminal Help

```
 GNU nano 2.9.3              01-network-manager-all.yaml

# Let NetworkManager manage all devices on this system
network:
   renderer: NetworkManager
   version: 2
   ethernets:
       enp0s3:
          addresses: [10.0.2.2/24]
          gateway4: 10.0.2.1
          dhcp4: no
          dhcp6: no
          nameservers:
             addresses: [208.67.222.222,208.67.220.220]
```

Figure 2-Client Configurations

On VirtualBox, the Gateway and Client Virtual Machines were set-up as internal networks: On the File menu select "Tools" then "Network Manger" then you configure your network interface as 10.0.2.0/24.

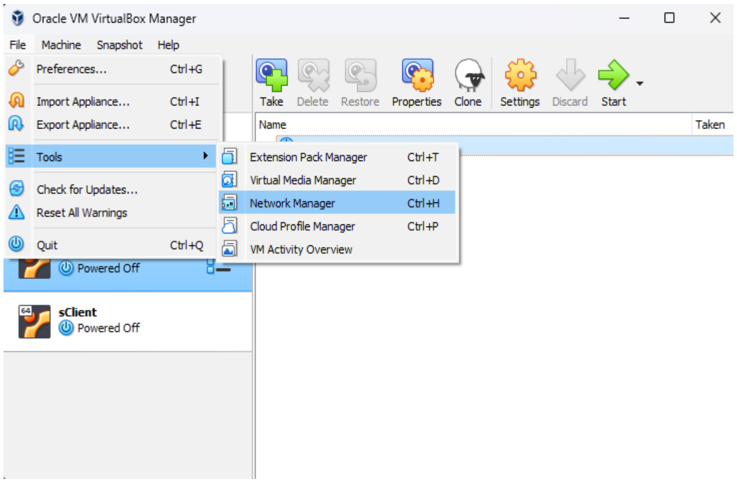


Figure 5-Network Manager for client

Figure 6 N

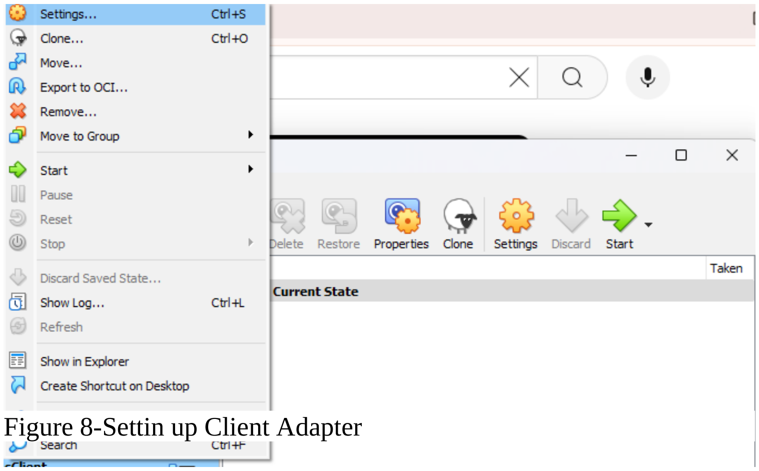


Figure 8-Settin up Client Adapter

I did same for the Gateway but at this point I had to set up two

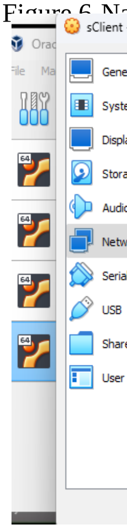
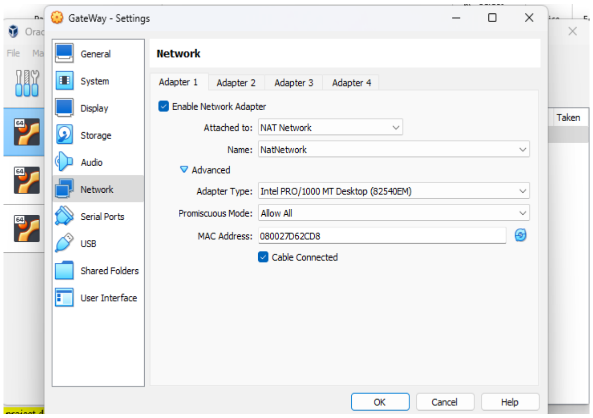


Figure 7-C

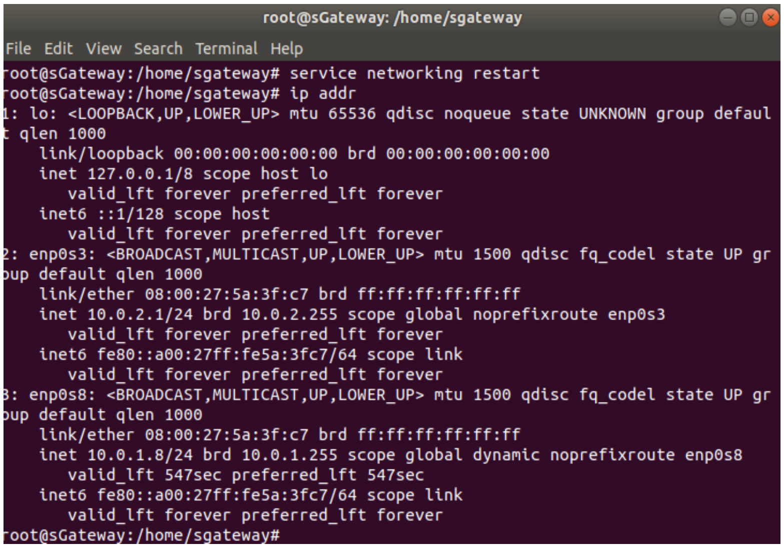
interfaces NatNetwork1 and NatNetwork2 on Adapter1 and Adapter2 respectively. The figure below shows how it was done.



☑ Enable Network Adapter
Attached to: NAT Network
Name: NatNetwork
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Allow All
MAC Address: 080027D62CD8
☑ Cable Connected

Figure 9-Set-Up for Gateway

On both Machines, I went to the root file and restarted the network using the “ **service networking restart**” command so that new network configurations are enabled.

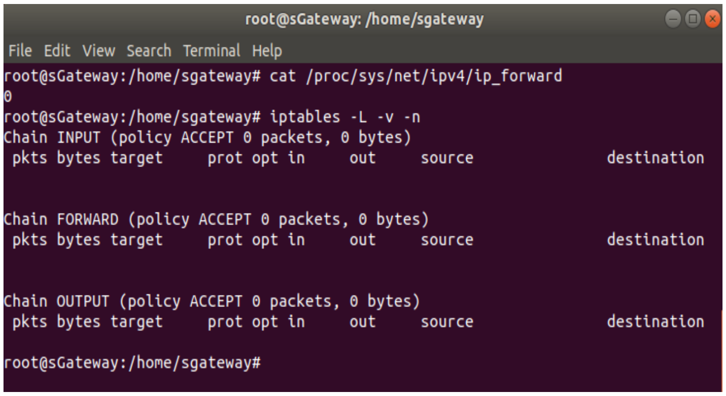

root@sGateway: /home/sgateway
File Edit View Search Terminal Help

```
root@sGateway:/home/sgateway# service networking restart
root@sGateway:/home/sgateway# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 gr
oup default qlen 1000
    link/ether 08:00:27:5a:3f:c7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.1/24 brd 10.0.2.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe5a:3fc7/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:5a:3f:c7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.8/24 brd 10.0.1.255 scope global dynamic noprefixroute enp0s8
       valid_lft 547sec preferred_lft 547sec
    inet6 fe80::a00:27ff:fe5a:3fc7/64 scope link
       valid_lft forever preferred_lft forever
root@sGateway:/home/sgateway#
```

Figure 10-Gateway Interface

Figure 1

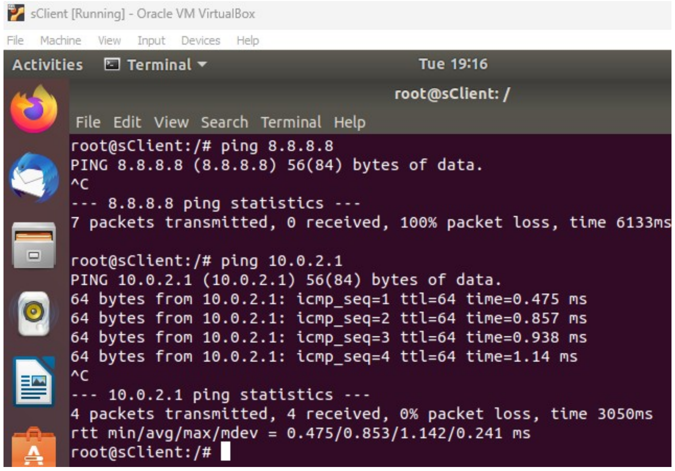
Initially, we haven’t configured anything to forward traffic on the Gateway and the Client is the only machine in the InternalNetwork and can only ping the Gateway’s VM interface which is 10.0.2.1.


root@sGateway: /home/sgateway
File Edit View Search Terminal Help

```
root@sGateway:/home/sgateway# cat /proc/sys/net/ipv4/ip_forward
0
root@sGateway:/home/sgateway# iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


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

root@sGateway:/home/sgateway#
```

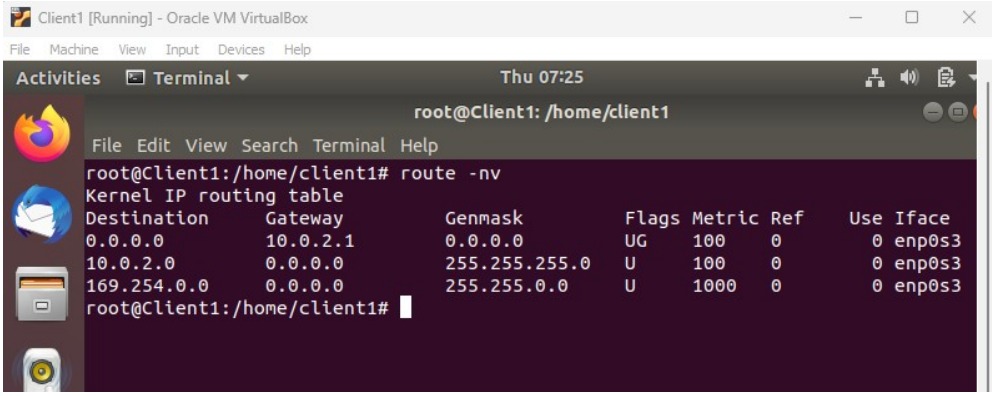
Figure 12-Gateway initial forward routing and iptable rules

Initial routing table for the client:


sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 19:16
root@sClient: /
File Edit View Search Terminal Help

```
root@sClient:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6133ms

root@sClient:/# ping 10.0.2.1
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data.
64 bytes from 10.0.2.1: icmp_seq=1 ttl=64 time=0.475 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=64 time=0.857 ms
64 bytes from 10.0.2.1: icmp_seq=3 ttl=64 time=0.938 ms
64 bytes from 10.0.2.1: icmp_seq=4 ttl=64 time=1.14 ms
^C
--- 10.0.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.475/0.853/1.142/0.241 ms
root@sClient:/#
```

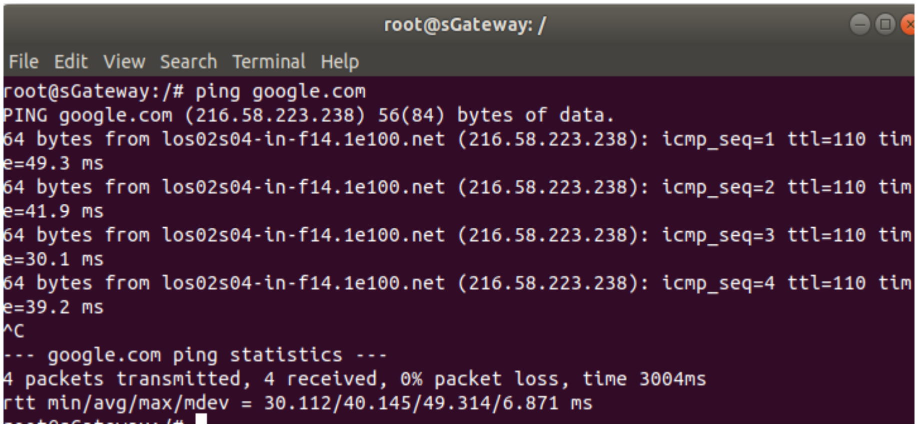
Figure 13-Initial network situation-Client


Client1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Thu 07:25
root@Client1: /home/client1
File Edit View Search Terminal Help

```
root@Client1:/home/client1# route -nv
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG    100    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@Client1:/home/client1#
```

Figure 2-Client Initial Routing Table

The Gateway has no issues resolving names with DNS and reaching also the internet by its default route:



```
root@sGateway:/# ping google.com
PING google.com (216.58.223.238) 56(84) bytes of data.
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=1 ttl=110 tim
e=49.3 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=2 ttl=110 tim
e=41.9 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=3 ttl=110 tim
e=30.1 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=4 ttl=110 tim
e=39.2 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 30.112/40.145/49.314/6.871 ms
```

Figure 15-DNS Resolution

III. Software

The tools used are:

1. Apache web server.
2. Packet inspection and NAT firewall, iptables
3. Various network tools such as WireShark or tcpdump, ping, traceroute
4. Packet forwarding router

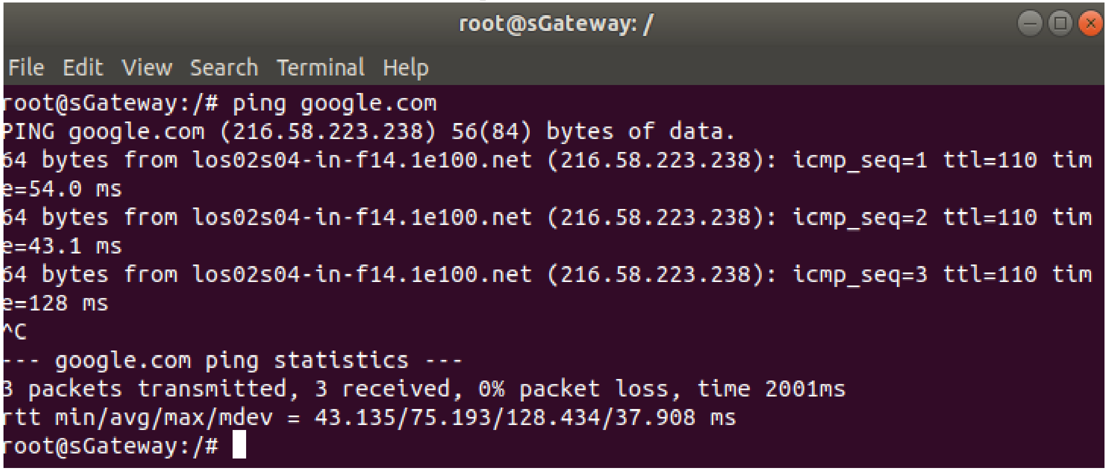
IV. Project Description

This is a step-by-step description of how the project was done.

A. The first step is to check the connectivity among VMs. So, I ping to check connectivity to make sure it’s successful.

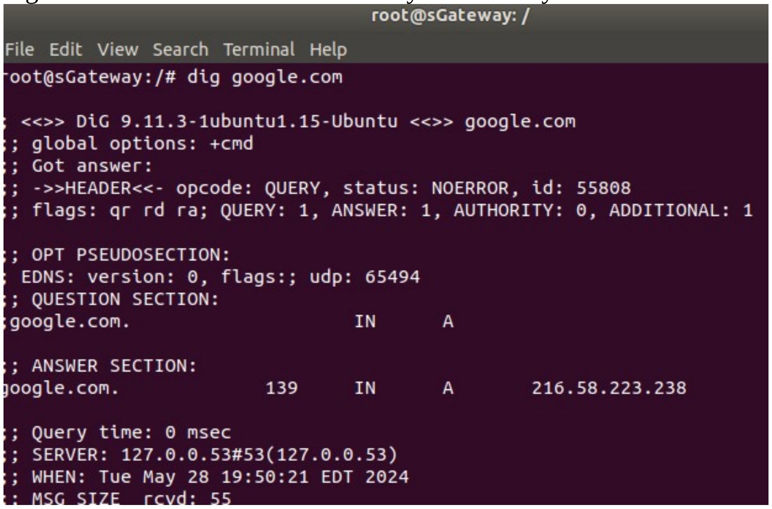
So, I did an initial set up on the machines in the Network Setup page that was given. The result must be so that the gateway can reach the internet, but the client cannot.

B. Check the network setup on the Gateway/Server VM. From a terminal window on the gateway, use the ping command to ensure internet reachability. So, I used **ping google.com** to ensure whether my gateway can reach the internet and it was successful. You have to interrupt it with CTRL+C.



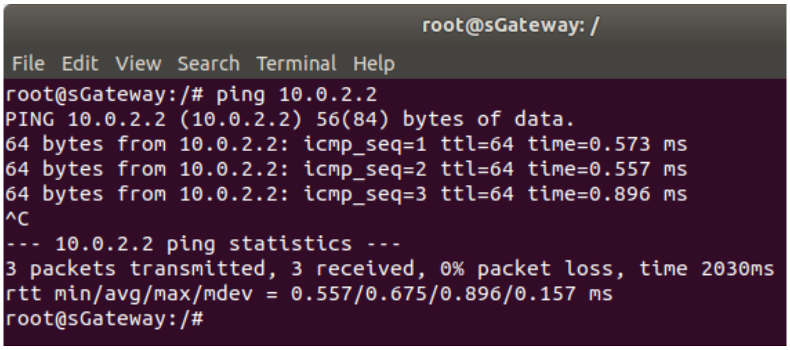
```
root@sGateway:/# ping google.com
PING google.com (216.58.223.238) 56(84) bytes of data.
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=1 ttl=110 tim
e=54.0 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=2 ttl=110 tim
e=43.1 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=3 ttl=110 tim
e=128 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 43.135/75.193/128.434/37.908 ms
root@sGateway:/# 
```

Figure 16-Ensure Internet Reachability on Gateway



```
root@sGateway:/# dig google.com

; <<>> DiG 9.11.3-1ubuntu1.15-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55808
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             139     IN      A       216.58.223.238

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue May 28 19:50:21 EDT 2024
;; MSG SIZE  rcvd: 55
```

Make sure that the Gateway can connect Client after testing internet reachability, issuing the command **ping 10.0.2.2**

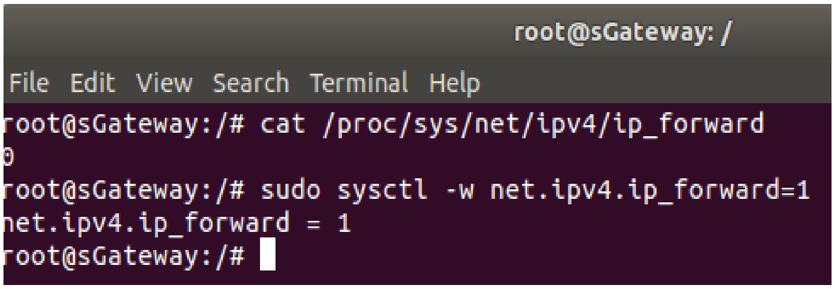


```
root@sGateway:/# ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.
64 bytes from 10.0.2.2: icmp_seq=1 ttl=64 time=0.573 ms
64 bytes from 10.0.2.2: icmp_seq=2 ttl=64 time=0.557 ms
64 bytes from 10.0.2.2: icmp_seq=3 ttl=64 time=0.896 ms
^C
--- 10.0.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.557/0.675/0.896/0.157 ms
root@sGateway:/#
```

Figure 3-Internet Reachability from Gateway to Client

C. Enable Packet forwarding

The Gateway must route traffic between the internal and the External networks in other to connect to the internet and this can be done by enabling packet forwarding and also configuring NAT. Initially, “**cat**” prints the value of the parameter to kno whether is has been set or not. **cat /proc/sys/net/ipv4/ip_forward** and we use the command **“sudo sysctl -w net.ipv4.ip_forward=1”**to enable it

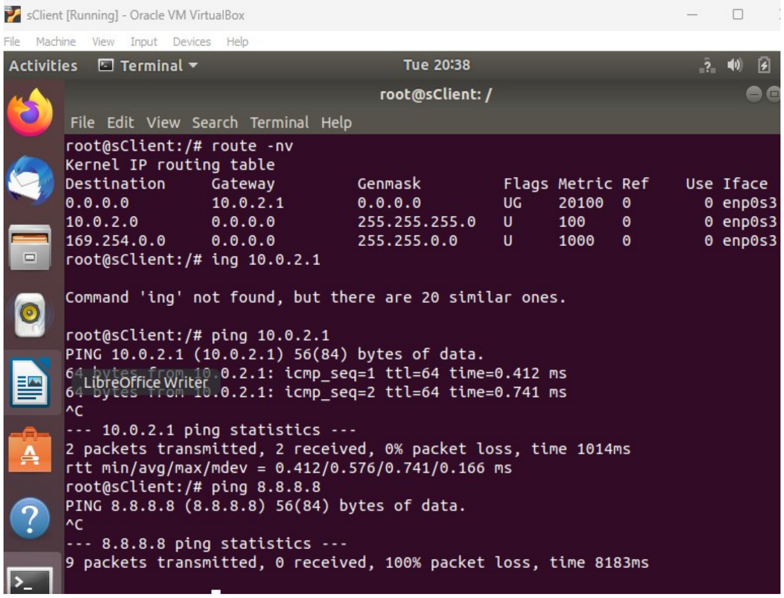


```
root@sGateway:/# cat /proc/sys/net/ipv4/ip_forward
0
root@sGateway:/# sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@sGateway:/# 
```

Figure 18- Enable Packet Forwarding

D. Testing Route. Client Reachability to Gateway.
The route is configured by default so we want to test whether the Client can reach the Gateway.

.

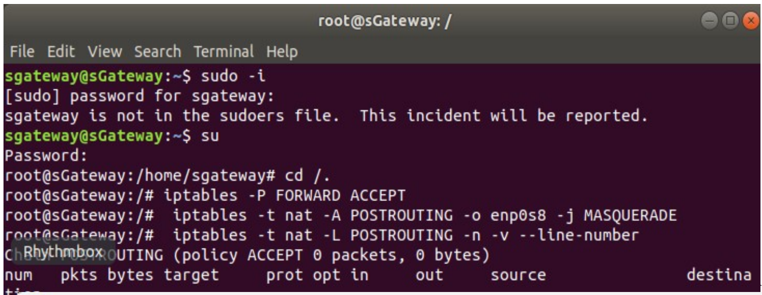


sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 20:38
root@sClient: /
File Edit View Search Terminal Help
LibreOffice Writer

```
root@sClient:/# route -nv
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG    20100  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@sClient:/# ing 10.0.2.1

Command 'ing' not found, but there are 20 similar ones.

root@sClient:/# ping 10.0.2.1
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data.
64 bytes from 10.0.2.1: icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=64 time=0.741 ms
^C
--- 10.0.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1014ms
rtt min/avg/max/mdev = 0.412/0.576/0.741/0.166 ms
root@sClient:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8183ms
```

Figure 19-Routing on Client

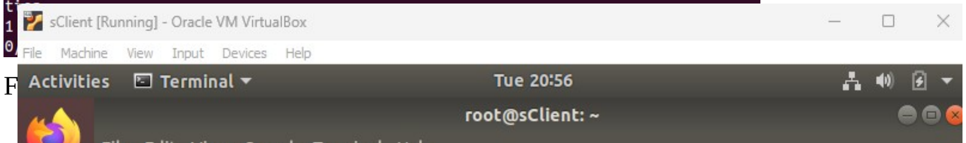
E. Network Traffic to the Internet is Enabled
The Client is supposed to reach the internet, so we must enable NAT rules on the Gateway. To achieve that, we need to use the following commands.
iptables -P FORWARD ACCEPT
iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE
iptables -t nat -L POSTROUTING -n -v --line-number



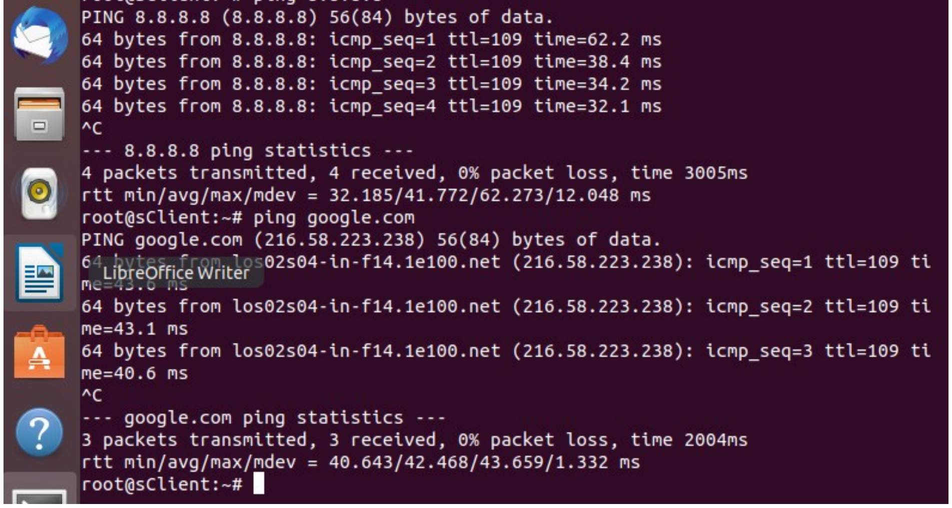
```
sgateway@sGateway:~$ sudo -i
[sudo] password for sgateway:
sgateway is not in the sudoers file.  This incident will be reported.
sgateway@sGateway:~$ su
Password:
root@sGateway:/home/sgateway# cd /.
root@sGateway:/# iptables -P FORWARD ACCEPT
root@sGateway:/#  iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE
root@sGateway:/#  iptables -t nat -L POSTROUTING -n -v --line-number
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destina
ti
1
0
```

n connect to some IP addresses and websites

F



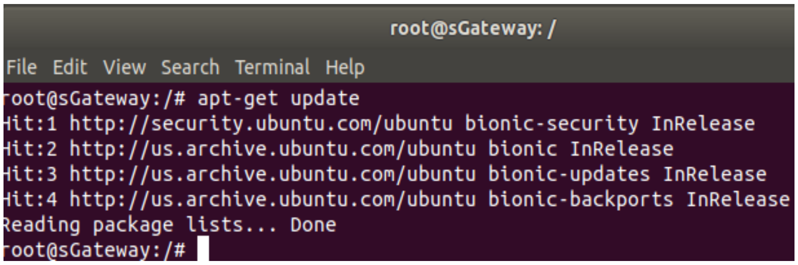
sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 20:56
root@sClient: ~



```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=62.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=38.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=34.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=109 time=32.1 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 32.185/41.772/62.273/12.048 ms
root@sClient:~# ping google.com
PING google.com (216.58.223.238) 56(84) bytes of data.
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=1 ttl=109 ti
me=43.6 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=2 ttl=109 ti
me=43.1 ms
64 bytes from los02s04-in-f14.1e100.net (216.58.223.238): icmp_seq=3 ttl=109 ti
me=40.6 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 40.643/42.468/43.659/1.332 ms
root@sClient:~#
```

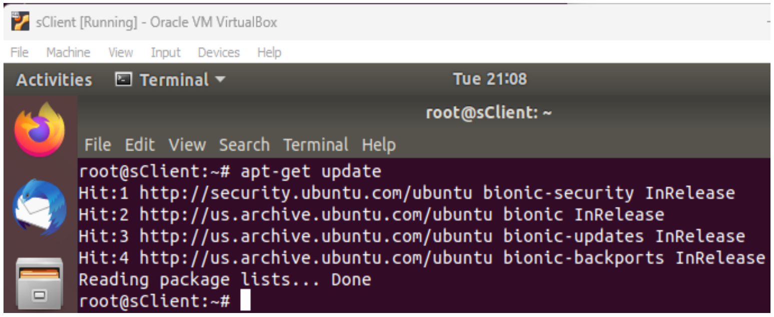
Figure 21-Testing Other IP address by using Ping

F. Test installed software and services Update Softwares and Install Softwares:
Run **sudo apt-get update** and **“sudo apt -y dist-upgrade”** to update softwares to the latest version on both VM’s



```
root@sGateway:/# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
root@sGateway:/#
```

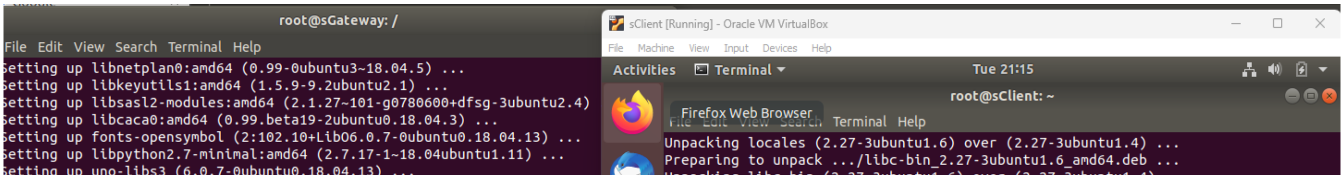
Figure 22-Update Softwares on Gateway



sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 21:08
root@sClient: ~
File Edit View Search Terminal Help

```
root@sClient:~# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
root@sClient:~#
```

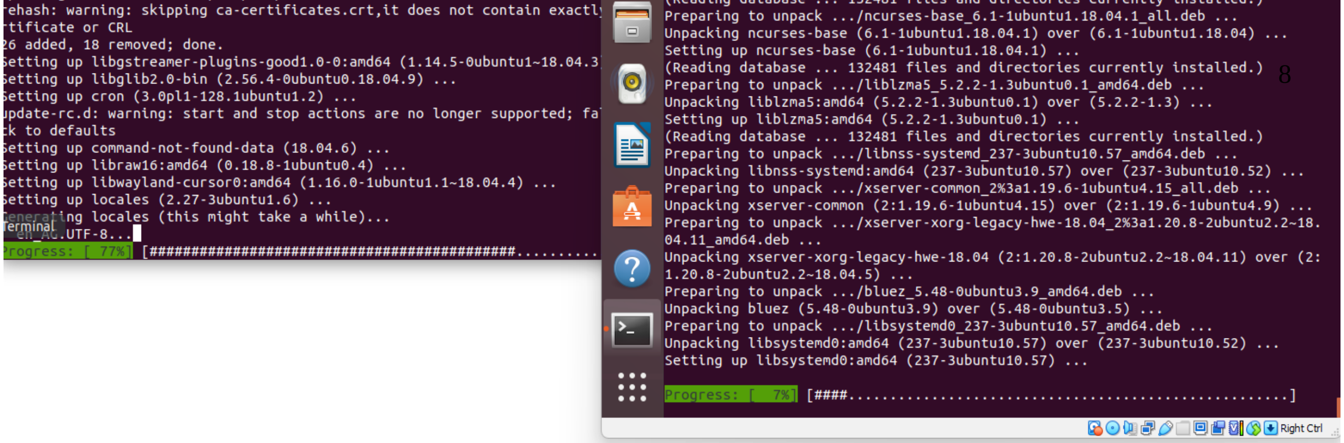
Figure 23-Update Softwares on Client



```
Setting up libnetplan0:amd64 (0.99-0ubuntu3~18.04.5) ...
Setting up libkeyutils1:amd64 (1.5.9-9.2ubuntu2.1) ...
Setting up libsasl2-modules:amd64 (2.1.27~101-g0780600+dfsg-3ubuntu2.4)
Setting up libcaca0:amd64 (0.99.beta19-2ubuntu0.18.04.3) ...
Setting up fonts-opensymbol (2:102.10+LibO6.0.7-0ubuntu0.18.04.13) ...
Setting up libpython2.7-minimal:amd64 (2.7.17-1~18.04ubuntu1.11) ...
Setting up uno-libs3 (6.0.7-0ubuntu0.18.04.13) ...
```

sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 21:15
root@sClient: ~
Firefox Web Browser
Terminal Help

```
Unpacking locales (2.27-3ubuntu1.6) over (2.27-3ubuntu1.4) ...
Preparing to unpack .../libc-bin_2.27-3ubuntu1.6_amd64.deb ...
```

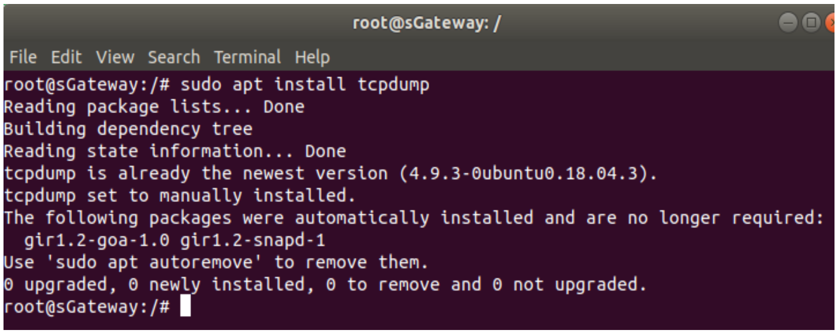


```
rehash: warning: skipping ca-certificates.crt,it does not contain exactly
rtificate or CRL
26 added, 18 removed; done.
Setting up libgstreamer-plugins-good1.0-0:amd64 (1.14.5-0ubuntu1~18.04.3
Setting up libglib2.0-bin (2.56.4-0ubuntu0.18.04.9) ...
Setting up cron (3.0pl1-128.1ubuntu1.2) ...
update-rc.d: warning: start and stop actions are no longer supported; fa
ck to defaults
Setting up command-not-found-data (18.04.6) ...
Setting up libraw16:amd64 (0.18.8-1ubuntu0.4) ...
Setting up libwayland-cursor0:amd64 (1.16.0-1ubuntu1.1~18.04.4) ...
Setting up locales (2.27-3ubuntu1.6) ...
Generating locales (this might take a while)...
  en_AG.UTF-8...
Progress: [ 77%] [###########################################.........
```

```
Preparing to unpack .../ncurses-base_6.1-1ubuntu1.18.04.1_all.deb ...
Unpacking ncurses-base (6.1-1ubuntu1.18.04.1) over (6.1-1ubuntu1.18.04) ...
Setting up ncurses-base (6.1-1ubuntu1.18.04.1) ...
(Reading database ... 132481 files and directories currently installed.)
Preparing to unpack .../liblzma5_5.2.2-1.3ubuntu0.1_amd64.deb ...
Unpacking liblzma5:amd64 (5.2.2-1.3ubuntu0.1) over (5.2.2-1.3) ...
Setting up liblzma5:amd64 (5.2.2-1.3ubuntu0.1) ...
(Reading database ... 132481 files and directories currently installed.)
Preparing to unpack .../libnss-systemd_237-3ubuntu10.57_amd64.deb ...
Unpacking libnss-systemd:amd64 (237-3ubuntu10.57) over (237-3ubuntu10.52) ...
Preparing to unpack .../xserver-common_2%3a1.19.6-1ubuntu4.15_all.deb ...
Unpacking xserver-common (2:1.19.6-1ubuntu4.15) over (2:1.19.6-1ubuntu4.9) ...
Preparing to unpack .../xserver-xorg-legacy-hwe-18.04_2%3a1.20.8-2ubuntu2.2~18.
04.11_amd64.deb ...
Unpacking xserver-xorg-legacy-hwe-18.04 (2:1.20.8-2ubuntu2.2~18.04.11) over (2:
1.20.8-2ubuntu2.2~18.04.5) ...
Preparing to unpack .../bluez_5.48-0ubuntu3.9_amd64.deb ...
Unpacking bluez (5.48-0ubuntu3.9) over (5.48-0ubuntu3.5) ...
Preparing to unpack .../libsystemd0_237-3ubuntu10.57_amd64.deb ...
Unpacking libsystemd0:amd64 (237-3ubuntu10.57) over (237-3ubuntu10.52) ...
Setting up libsystemd0:amd64 (237-3ubuntu10.57) ...

Progress: [  7%] [####................................................]
```

Figure 24- Update Softwares to current versions on Both Machines

Install the “tcpdump” package using “**sudo apt install tcpdump**” command

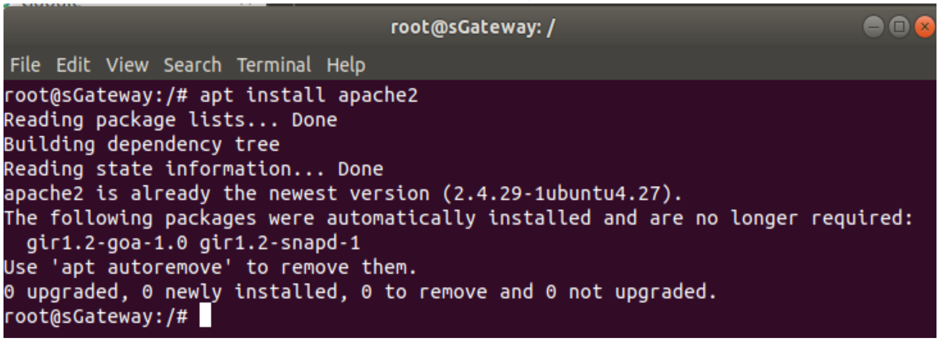


```
root@sGateway:/# sudo apt install tcpdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-0ubuntu0.18.04.3).
tcpdump set to manually installed.
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@sGateway:/#
```

Figure 25-Install tcpdump

G. Install Apache2 on the Gateway

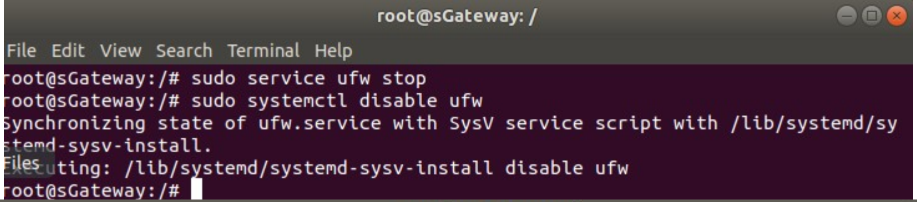
Use “**sudo apt install apache2**” on the Gateway to install apache2



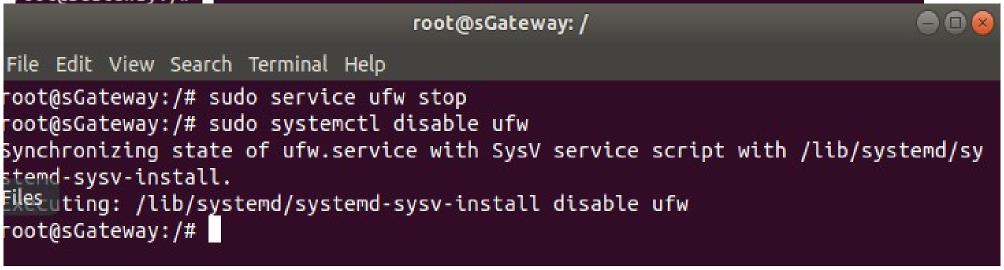
```
root@sGateway:/# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.29-1ubuntu4.27).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@sGateway:/#
```

Figure 26-Install Apache2 on Gateway

We must disable firewalls application and stop and will not activate on restart by issuing the commands: “**sudo service**



```
root@sGateway:/# sudo service ufw stop
root@sGateway:/# sudo systemctl disable ufw
Synchronizing state of ufw.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable ufw
root@sGateway:/# 
```

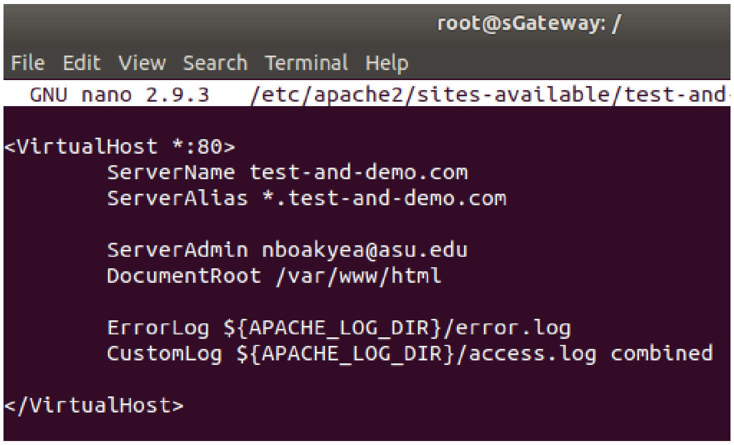


```
root@sGateway:/# sudo service ufw stop
root@sGateway:/# sudo systemctl disable ufw
Synchronizing state of ufw.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable ufw
root@sGateway:/# 
```

Figure 27-Disabling Firewalls

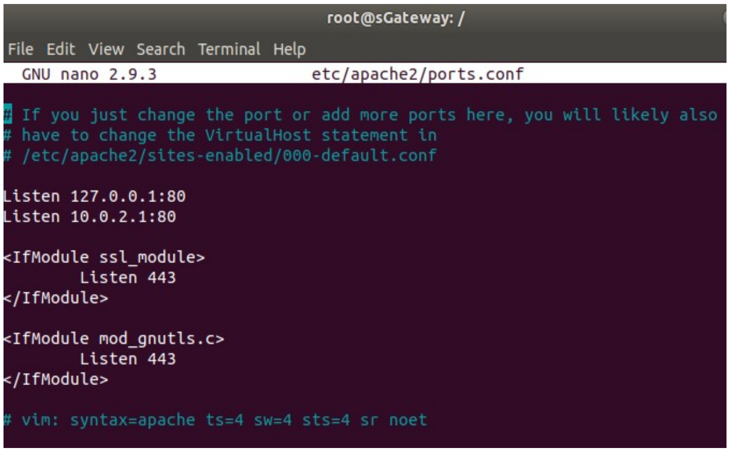
H. Apache2 Set-Up

The configuration file **“/etc/apache2/sites-available/test-and-demo.conf**” needs to be created and be equal to the listing below. It can be accomplished by using the “**sudo**” command with nano editor and the next configuration file to change is “**/etc/apache2/ports.conf**”.



```
 GNU nano 2.9.3   /etc/apache2/sites-available/test-and

<VirtualHost *:80>
        ServerName test-and-demo.com
        ServerAlias *.test-and-demo.com

        ServerAdmin nboakyea@asu.edu
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

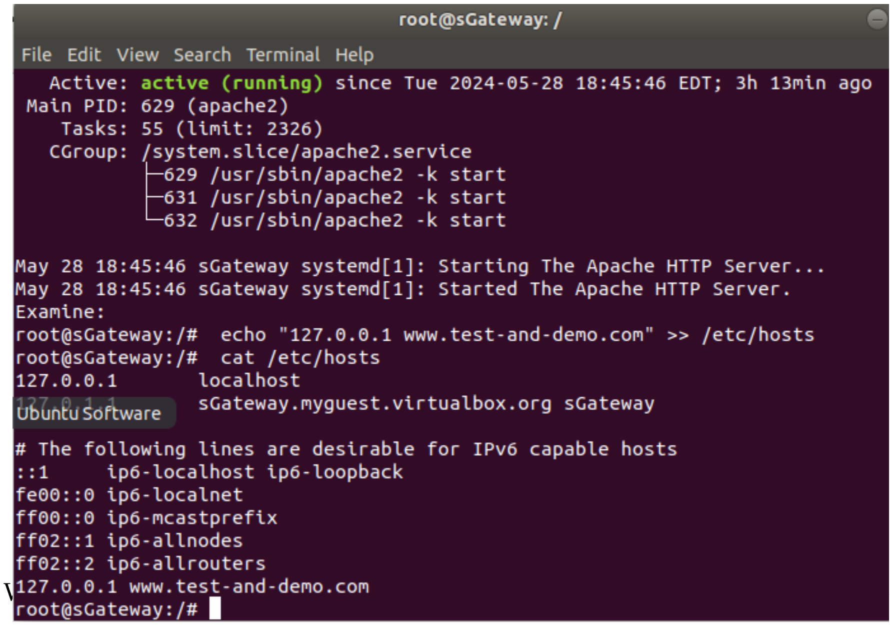
Figure 4-Configure the test-and-demo.conf file



```
 GNU nano 2.9.3              etc/apache2/ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 127.0.0.1:80
Listen 10.0.2.1:80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figure 28-Port.conf

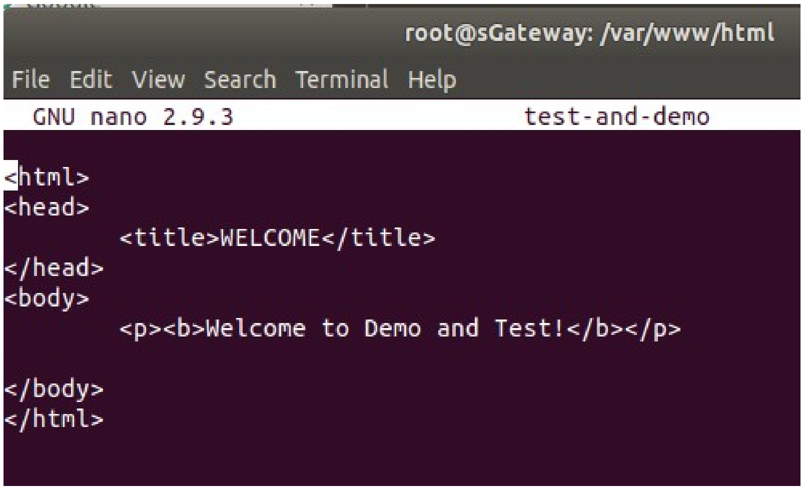
While still on the Gateway, add the hostname “www.test-and-demo.com
” on the /etc/hosts file by typing “**echo "127.0.0.1 www.test-and-demo.com" >> /etc/hosts**”



```
   Active: active (running) since Tue 2024-05-28 18:45:46 EDT; 3h 13min ago
 Main PID: 629 (apache2)
    Tasks: 55 (limit: 2326)
   CGroup: /system.slice/apache2.service
           ├─629 /usr/sbin/apache2 -k start
           ├─631 /usr/sbin/apache2 -k start
           └─632 /usr/sbin/apache2 -k start

May 28 18:45:46 sGateway systemd[1]: Starting The Apache HTTP Server...
May 28 18:45:46 sGateway systemd[1]: Started The Apache HTTP Server.
Examine:
root@sGateway:/#  echo "127.0.0.1 www.test-and-demo.com" >> /etc/hosts
root@sGateway:/#  cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       sGateway.myguest.virtualbox.org sGateway

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 www.test-and-demo.com
root@sGateway:/# 
```

V **ectory**

Figure 5-Add the hostname “www.test-and-demo.com

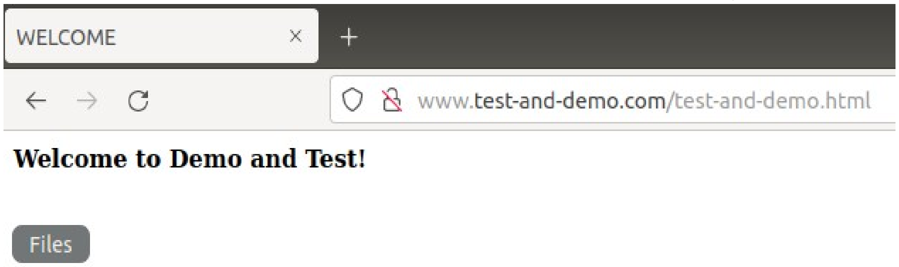

root@sGateway: /var/www/html
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                    test-and-demo

<html>
<head>
        <title>WELCOME</title>
</head>
<body>
        <p><b>Welcome to Demo and Test!</b></p>

</body>
</html>
```

Figure 31-Html File for Webserver

Enable the Apache2 VirtualHost and restart Apache2 with “**sudo a2ensite test-and-demo” and “sudo systemctl restart apache2.service”**

I. Connectivity to the Gateway from the web Server
You should be able to open the www.test-and -demo.com site on your browser

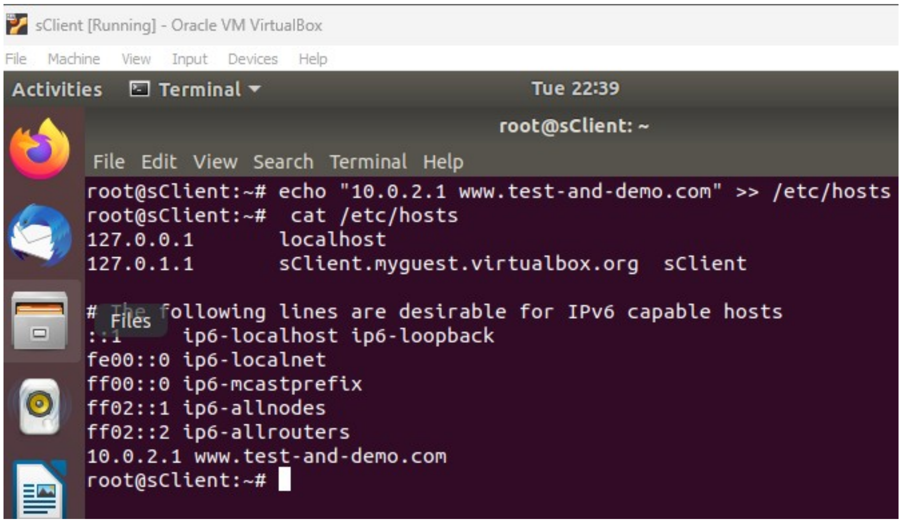


Welcome to Demo and Test!

Figure 32-Connecting to the web server

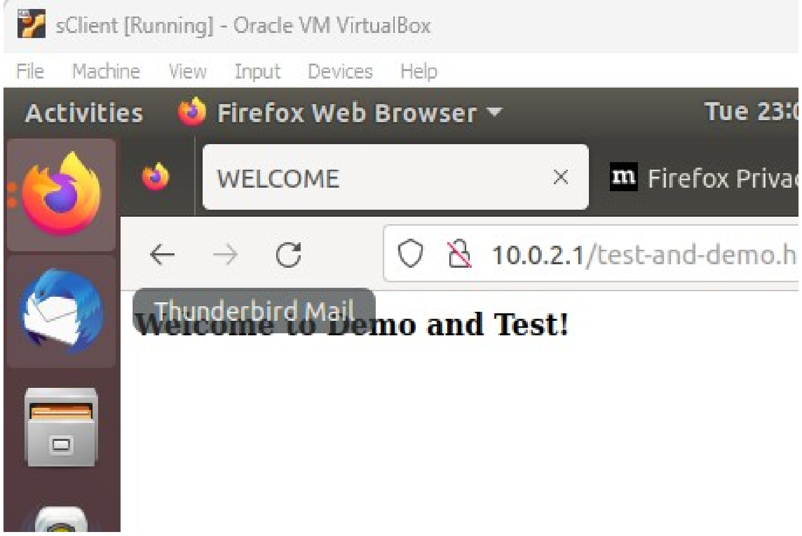
J. Apache2 should be reachable to the Client
Use this command in other for the Client to be reachable to the Gateway on the Client , /etc/hosts

“echo "10.0.2.1 www.test-and-demo.com" >>”
“cat /etc/hosts”


sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 22:39
root@sClient: ~
File Edit View Search Terminal Help
Files

```
root@sClient:~# echo "10.0.2.1 www.test-and-demo.com" >> /etc/hosts
root@sClient:~#  cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       sClient.myguest.virtualbox.org  sClient

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.0.2.1 www.test-and-demo.com
root@sClient:~# 
```

nd-demo” in the url

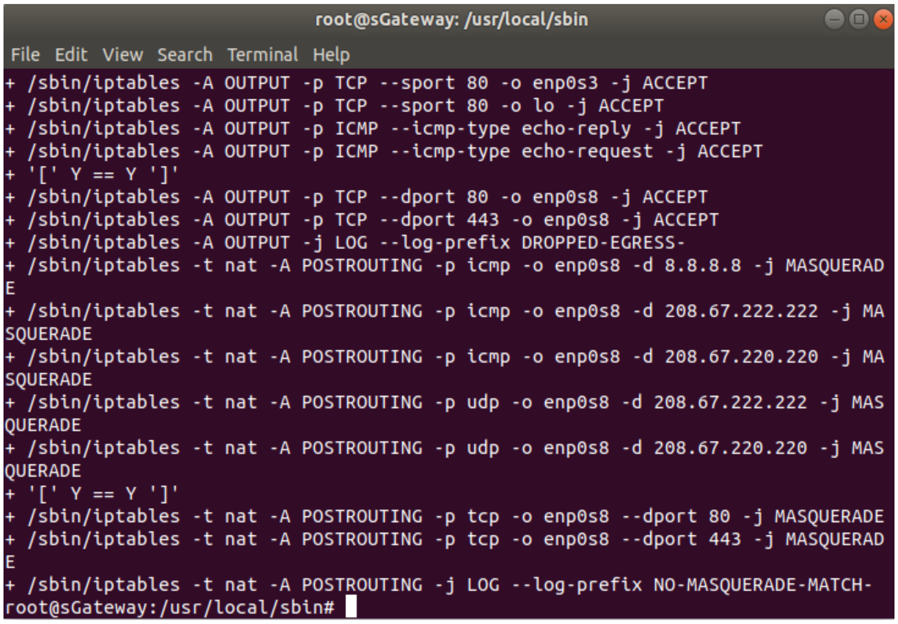


Welcome to Demo and Test!

Figure 33-Client Reachability to the web Server

K. Packet filter firewall Set-up

By Setting up a packet filter, it contributes to the overall integrity and reliability of the network. Packet filter helps ensure the flow of traffic, protect private data, prevent attackers from access to networks, and protect the network from all kinds of attacks. By defining and enforcing security rules, packet filters contribute to the overall. We have to execute the file by using this command "**chmod +x rc.firewall**", and after that you run the file **". ./rc.firewall**".


root@sGateway: /usr/local/sbin
File Edit View Search Terminal Help

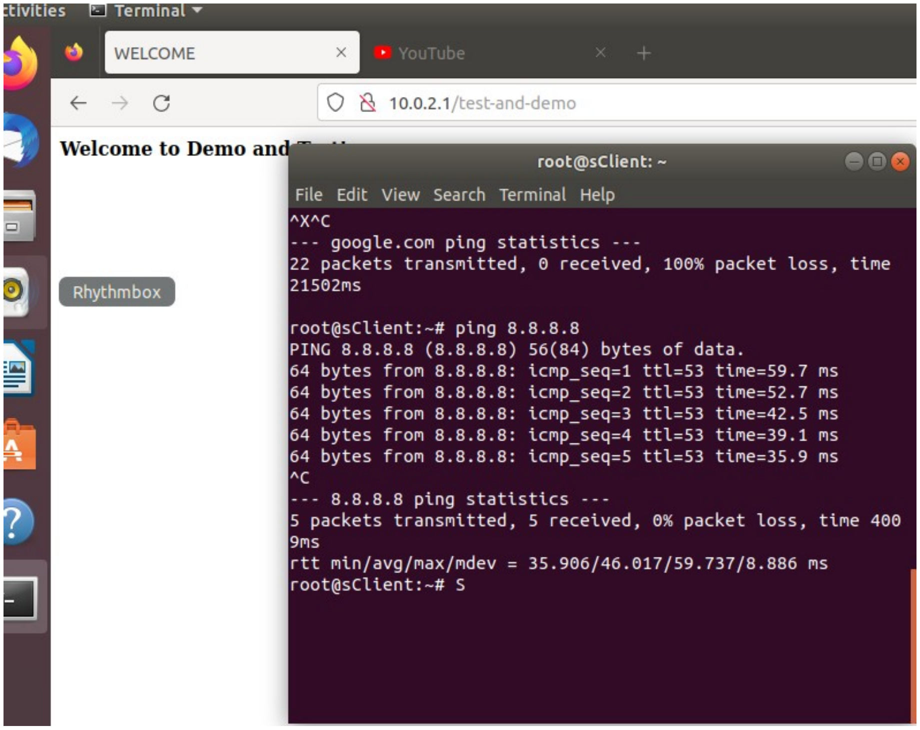
```
+ /sbin/iptables -A OUTPUT -p TCP --sport 80 -o enp0s3 -j ACCEPT
+ /sbin/iptables -A OUTPUT -p TCP --sport 80 -o lo -j ACCEPT
+ /sbin/iptables -A OUTPUT -p ICMP --icmp-type echo-reply -j ACCEPT
+ /sbin/iptables -A OUTPUT -p ICMP --icmp-type echo-request -j ACCEPT
+ '[' Y == Y ']'
+ /sbin/iptables -A OUTPUT -p TCP --dport 80 -o enp0s8 -j ACCEPT
+ /sbin/iptables -A OUTPUT -p TCP --dport 443 -o enp0s8 -j ACCEPT
+ /sbin/iptables -A OUTPUT -j LOG --log-prefix DROPPED-EGRESS-
+ /sbin/iptables -t nat -A POSTROUTING -p icmp -o enp0s8 -d 8.8.8.8 -j MASQUERAD
E
+ /sbin/iptables -t nat -A POSTROUTING -p icmp -o enp0s8 -d 208.67.222.222 -j MA
SQUERADE
+ /sbin/iptables -t nat -A POSTROUTING -p icmp -o enp0s8 -d 208.67.220.220 -j MA
SQUERADE
+ /sbin/iptables -t nat -A POSTROUTING -p udp -o enp0s8 -d 208.67.222.222 -j MAS
QUERADE
+ /sbin/iptables -t nat -A POSTROUTING -p udp -o enp0s8 -d 208.67.220.220 -j MAS
QUERADE
+ '[' Y == Y ']'
+ /sbin/iptables -t nat -A POSTROUTING -p tcp -o enp0s8 --dport 80 -j MASQUERADE
+ /sbin/iptables -t nat -A POSTROUTING -p tcp -o enp0s8 --dport 443 -j MASQUERAD
E
+ /sbin/iptables -t nat -A POSTROUTING -j LOG --log-prefix NO-MASQUERADE-MATCH-
root@sGateway:/usr/local/sbin#
```

Figure 34-Packet Filter Set up

The table below shows the configurations on the script for the Gateway to filter certain ports, propagating and blocking those ports.

Source	Destination	Protocol	Rule	Comment
Gateway	Loopback	*	Allowed	Allow loopback
Gateway	Client	ICMP	Allowed	Ping the Client
Gateway	Internet	TCP/80	Allowed	Connects to Web for updates
Gateway	Internet	TCP/443	Allowed	Connects to Web for updates
Gateway	DNS servers	ICMP	Allowed	Ping DNS Servers
Gateway	DNS servers	DNS	Allowed	Allow DNS Resolutions
Gateway	8.8.8.8	ICMP	Allowed	Ping Google DNS
Gateway	*	*	Deny	Deny the Rest
Client	Loopback	TCP/80	Allowed	Allow the Client
Client	Gateway	TCP/80	Allowed	Connects to the Web Server on the Gateway
Client	Internet	TCP/443	Allowed	Connects to the Web for updates
Client	Internet	ICMP	Allowed	Connects to the Web for updates
Client	DNS servers	DNS	Allowed	Allow DNS Resolution
Client	DNS servers	ICMP	Allowed	Ping the Gateway
Client	8.8.8.8	ICMP	Allowed	Ping Google DNS
Client	*	*	Deny	Deny the Rest

The Client can ping the DNS server of google but cannot resolve the name of the url, "google.com" and the DNS Server is used to do that. The NAT rules are set up to allow protocols to perform this. Interestingly, the Client can browse the web and reach the web server on the Gateway.

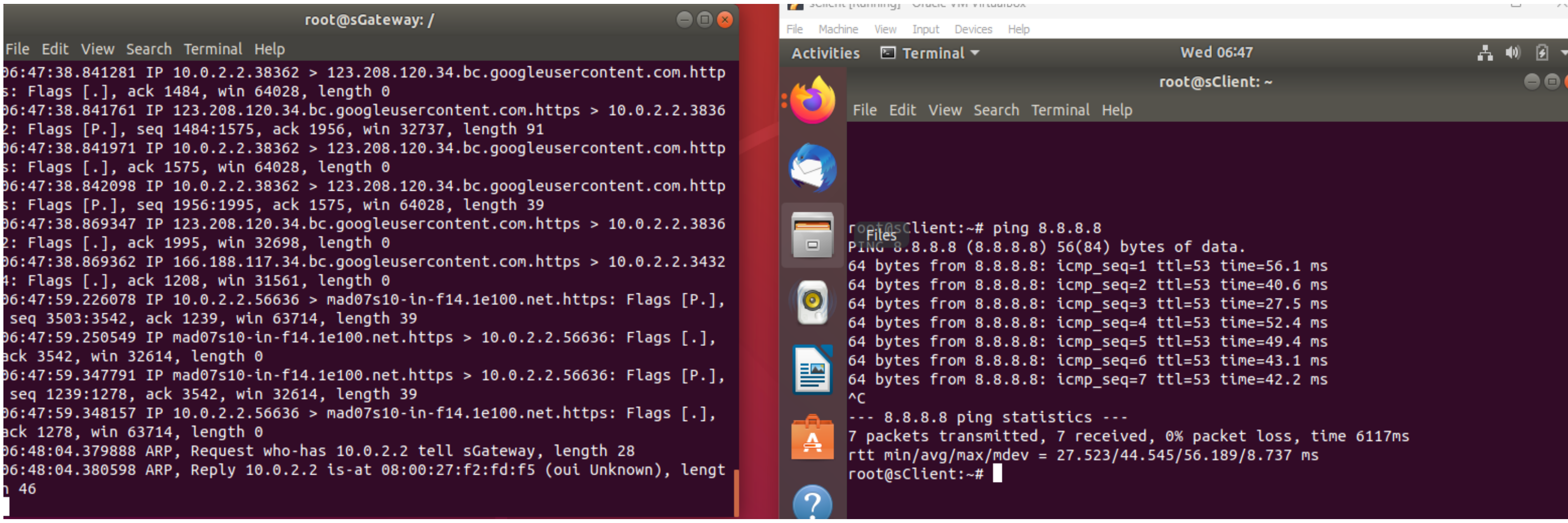


```
^X^C
--- google.com ping statistics ---
22 packets transmitted, 0 received, 100% packet loss, time
21502ms

root@sClient:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=59.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=52.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=42.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=39.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=53 time=35.9 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 400
9ms
rtt min/avg/max/mdev = 35.906/46.017/59.737/8.886 ms
root@sClient:~# S
```

Figure 35-Reachability from the Client to the Gateway

L. Ability of the Gateway to Sniff all traffic to the Client

Before a traffic can get to the Client, the gateway can sniff the traffic and this can be run with this command "**sudo tcpdump -i enp0s3**" on the Gateway, right after, you get on the Client to ping 8.8.8.8. WireShark can also be installed to give a better and graphical view of the sniffing traffic

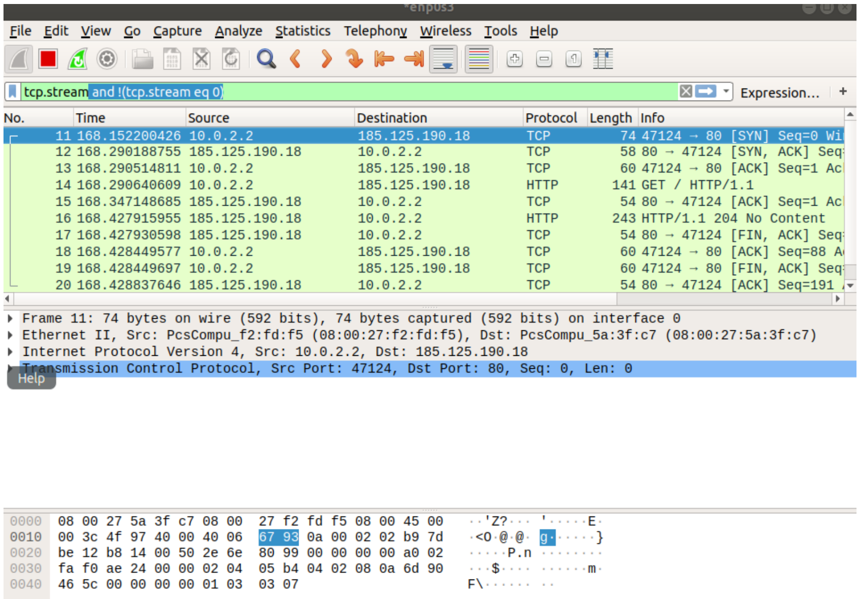


root@sGateway: /
File Edit View Search Terminal Help
Activities
Terminal
Wed 06:47
root@sClient: ~
File Edit View Search Terminal Help
Files

```
06:47:38.841281 IP 10.0.2.2.38362 > 123.208.120.34.bc.googleusercontent.com.http
s: Flags [.], ack 1484, win 64028, length 0
06:47:38.841761 IP 123.208.120.34.bc.googleusercontent.com.https > 10.0.2.2.3836
2: Flags [P.], seq 1484:1575, ack 1956, win 32737, length 91
06:47:38.841971 IP 10.0.2.2.38362 > 123.208.120.34.bc.googleusercontent.com.http
s: Flags [.], ack 1575, win 64028, length 0
06:47:38.842098 IP 10.0.2.2.38362 > 123.208.120.34.bc.googleusercontent.com.http
s: Flags [P.], seq 1956:1995, ack 1575, win 64028, length 39
06:47:38.869347 IP 123.208.120.34.bc.googleusercontent.com.https > 10.0.2.2.3836
2: Flags [.], ack 1995, win 32698, length 0
06:47:38.869362 IP 166.188.117.34.bc.googleusercontent.com.https > 10.0.2.2.3432
4: Flags [.], ack 1208, win 31561, length 0
06:47:59.226078 IP 10.0.2.2.56636 > mad07s10-in-f14.1e100.net.https: Flags [P.],
 seq 3503:3542, ack 1239, win 63714, length 39
06:47:59.250549 IP mad07s10-in-f14.1e100.net.https > 10.0.2.2.56636: Flags [.],
ack 3542, win 32614, length 0
06:47:59.347791 IP mad07s10-in-f14.1e100.net.https > 10.0.2.2.56636: Flags [P.],
 seq 1239:1278, ack 3542, win 32614, length 39
06:47:59.348157 IP 10.0.2.2.56636 > mad07s10-in-f14.1e100.net.https: Flags [.],
ack 1278, win 63714, length 0
06:48:04.379888 ARP, Request who-has 10.0.2.2 tell sGateway, length 28
06:48:04.380598 ARP, Reply 10.0.2.2 is-at 08:00:27:f2:fd:f5 (oui Unknown), lengt
h 46
```

```
root@sClient:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=56.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=40.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=27.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=52.4 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=53 time=49.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=53 time=43.1 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=53 time=42.2 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6117ms
rtt min/avg/max/mdev = 27.523/44.545/56.189/8.737 ms
root@sClient:~#
```

Figure 36-Gateway Sniffing Traffic to Client



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

```
tcp.stream and !(tcp.stream eq 0)
```

```
No.   Time           Source          Destination     Protocol Length Info
   11 168.152200426 10.0.2.2        185.125.190.18  TCP        74 47124 → 80 [SYN] Seq=0 Wi
   12 168.290188755 185.125.190.18  10.0.2.2        TCP        58 80 → 47124 [SYN, ACK] Seq
   13 168.290514811 10.0.2.2        185.125.190.18  TCP        60 47124 → 80 [ACK] Seq=1 Ac
   14 168.290640609 10.0.2.2        185.125.190.18  HTTP      141 GET / HTTP/1.1
   15 168.347148685 185.125.190.18  10.0.2.2        TCP        54 80 → 47124 [ACK] Seq=1 Ac
   16 168.427915955 185.125.190.18  10.0.2.2        HTTP      243 HTTP/1.1 204 No Content
   17 168.427930598 185.125.190.18  10.0.2.2        TCP        54 80 → 47124 [FIN, ACK] Seq
   18 168.428449577 10.0.2.2        185.125.190.18  TCP        60 47124 → 80 [ACK] Seq=88 A
   19 168.428449697 10.0.2.2        185.125.190.18  TCP        60 47124 → 80 [FIN, ACK] Seq
   20 168.428837646 185.125.190.18  10.0.2.2        TCP        54 80 → 47124 [ACK] Seq=191
```

```
▸ Frame 11: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_f2:fd:f5 (08:00:27:f2:fd:f5), Dst: PcsCompu_5a:3f:c7 (08:00:27:5a:3f:c7)
▸ Internet Protocol Version 4, Src: 10.0.2.2, Dst: 185.125.190.18
▸ Transmission Control Protocol, Src Port: 47124, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  08 00 27 5a 3f c7 08 00  27 f2 fd f5 08 00 45 00   ··'Z?··· '·····E·
0010  00 3c 4f 97 40 00 40 06  67 93 0a 00 02 02 b9 7d   ·<O·@·@· g······}
0020  be 12 b8 14 00 50 2e 6e  80 99 00 00 00 00 a0 02   ·····P.n ········
0030  fa f0 ae 24 00 00 02 04  05 b4 04 02 08 0a 6d 90   ···$···· ······m·
0040  46 5c 00 00 00 00 01 03  03 07                     F\······ ··
```

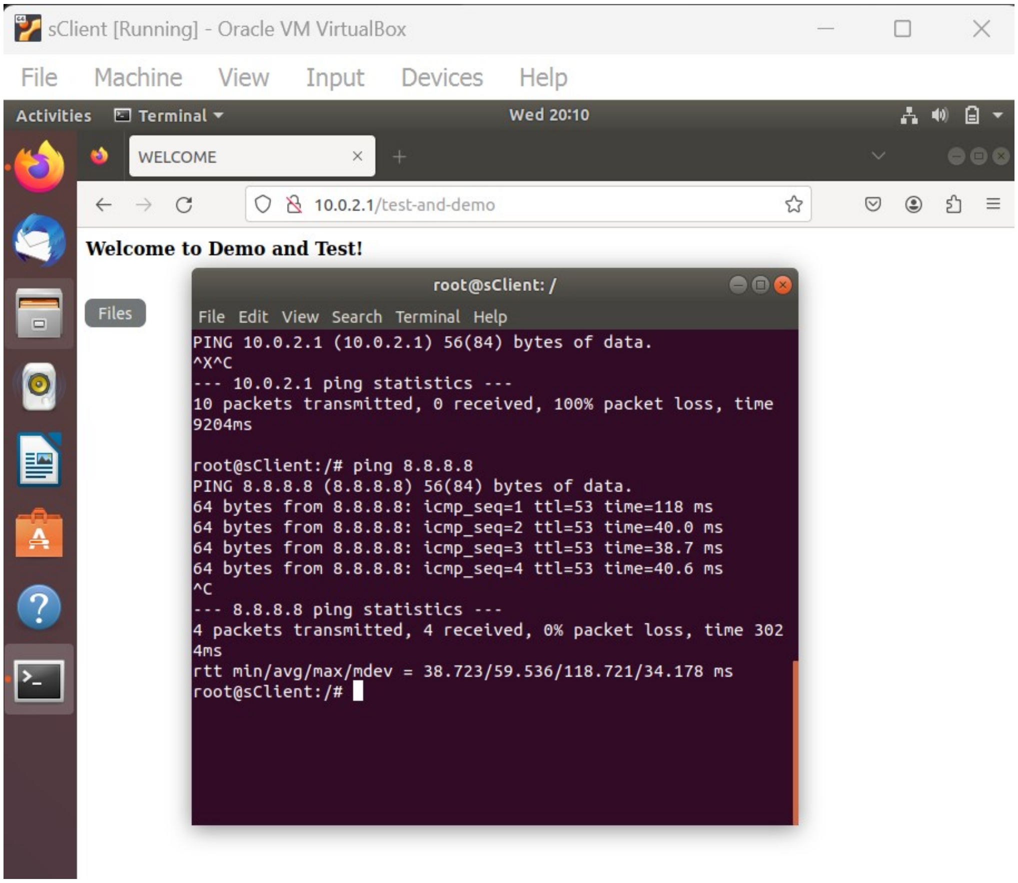
Figure 37-Using WireShark for a better View

M. The Client VM cannot ping the Gateway/Server VM IP address and can access the demo webpage on Gateway/Server VM by access the IP address of Gateway/Server VM in browser (the returning page must contain" Welcome ....", you can also use a web browser) and can ping 8.8.8.8. We add these rules to the our script.

SERVER_IP=10.0.2.1

# Drop ping requests to the server IP

sudo iptables -A INPUT -p icmp --icmp-type echo-request -d $SERVER_IP -j DROP
# Allow HTTP traffic to the server IP
sudo iptables -A INPUT -p tcp --dport 80 -d $SERVER_IP -j ACCEPT
# Allow ping to 8.8.8.8
sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -d 8.8.8.8 -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 8.8.8.8 -j ACCEPT

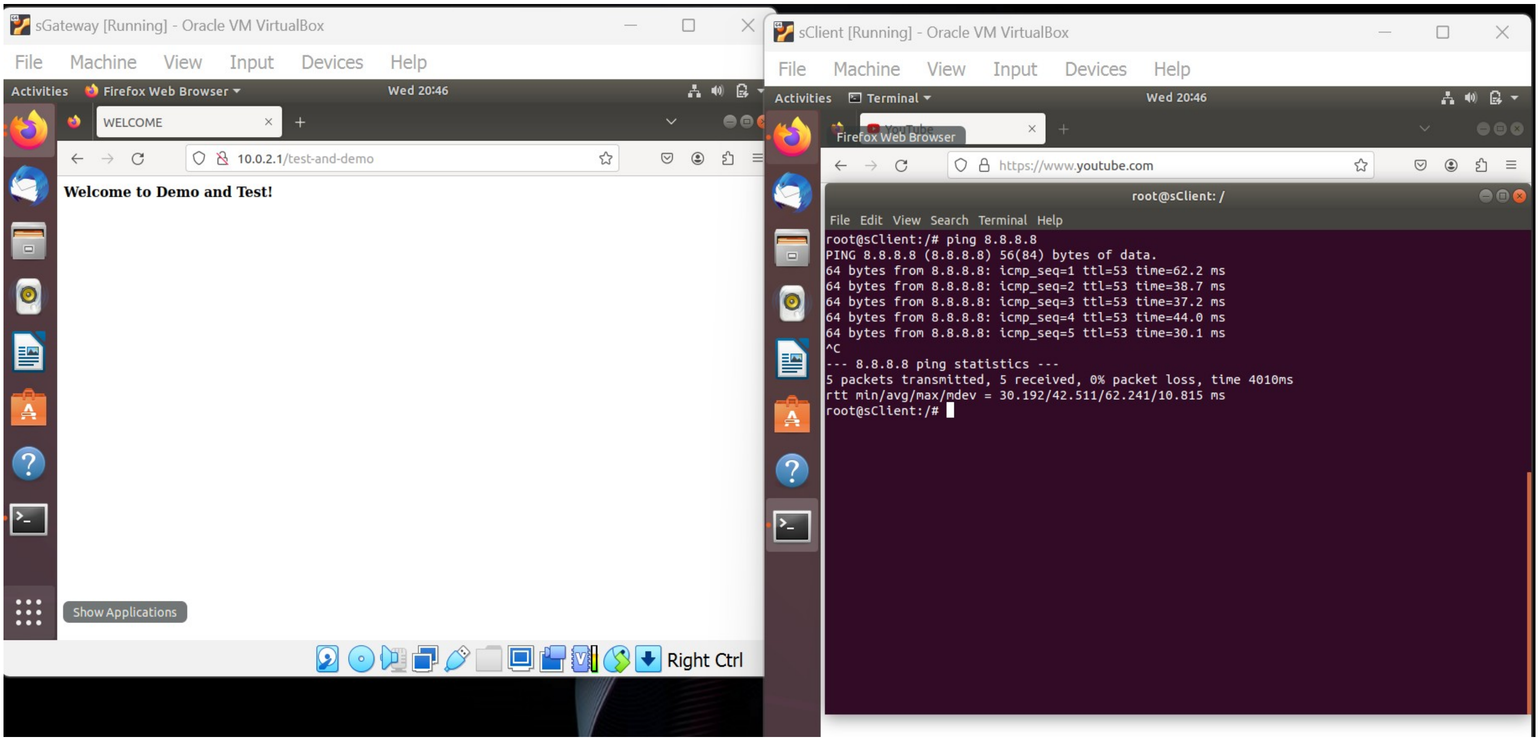


Welcome to Demo and Test!

```
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data.
^X^C
--- 10.0.2.1 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time
9204ms

root@sClient:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=118 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=40.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=38.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=40.6 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 302
4ms
rtt min/avg/max/mdev = 38.723/59.536/118.721/34.178 ms
root@sClient:/#
```

Figure 38-Assigment Task 1

N. The Gateway VM can set up http(webpage) service to its own IP address (with the demo page available). • enable POSTROUTING to allow client to access outside network (8.8.8.8) and change their source IP addresses. The are the commands you add to the script.
# Replace with your external network interface
EXTERNAL_IFACE=enp0s3
# Add POSTROUTING rule to change the source IP address of outgoing packets
sudo iptables -t nat -A POSTROUTING -o $EXTERNAL_IFACE -j MASQUERADE
#Allow HTTP traffic to the web server
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
#Drop ICMP echo requests (ping) to the server IP address:
sudo iptables -A INPUT -p icmp --icmp-type echo-request -d $SERVER_IP -j DROP
#Allow ping to 8.8.8.8
sudo iptables -A OUTPUT -p icmp --icmp-type echo-request -d 8.8.8.8 -j ACCEPT
sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 8.8.8.8 -j ACCEPT


sGateway [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Wed 20:46
WELCOME
10.0.2.1/test-and-demo
Show Applications
Right Ctrl
sClient [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Wed 20:46
Firefox Web Browser
https://www.youtube.com
root@sClient: /
File Edit View Search Terminal Help

Welcome to Demo and Test!

```
root@sClient:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=62.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=53 time=38.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=37.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=53 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=53 time=30.1 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 30.192/42.511/62.241/10.815 ms
root@sClient:/# 
```

Figure 39-Assignment Task 2

V. Conclusion

Describe lessons learned from this project, e.g., any interesting discoveries, tips, and tricks. Provide a self-assessment about your project and provide comments on this project.

1. To manage network traffic, providing ways for source and destination address translation which are very necessary for tasks like internet connection and port forwarding by using the NAT table in iptables. This helps us to control and secure network traffic effectively.

2. Packet filters also help to set rules that define what is allowed and not allowed based on the rules that are set in order to protect networks by effectively controlling traffic. I understood how networks are ensured by enabling firewalls, managing traffic, and enforcing access control policies.

3. Initial Network IP addresses set-up must be done correctly to avoid network issues like I.P Address conflicts and ensuring smooth network operation. Incorrectly I.P addresses might cause intermittent connection problems.

VI. Appendix B: Attached files

Provide a list of used configurations and developed source files (or gitlab/github links). In your configuration file, please with well-marked comments. A good practice is to provide comments where you made changes, something like:
The comment format depends on your used system files and programs.

Use this link https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C- for all developed source files on github .

01-network-manager-all-Client.yaml	https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C-/blob/main/01-network-manager-all-Client.yaml
01-network-manager-all.yaml	https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C-/blob/main/01-network-manager-all.yaml

ports.conf	https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C-/blob/main/ports.conf
rc.firewall	https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C-/blob/main/rc.firewall
test-and-demo	https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C-/blob/main/test-and-demo
test-and-demo.conf	https://github.com/nananyamedia/CSE-548-Adv-Computer-Network-Security-2024-Summer-C-/blob/main/test-and-demo.conf

VII. References

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

a. ChatGpt 4.0
b. https://www.netfilter.org/
c. Cisco Packet Filter: https://www.cisco.com/c/en/us/td/docs/wireless/access_point/15-3-3/configuration/guide/cg15-3-3/cg15-3-3-chap16-filters.pdf

TABLE OF FIGURES

Figure 1- Network Set-up........1
Figure 2-Client Configurations........2
Figure 3-Gateway Configurations........2
Figure 4- Gateway Configurations........2
Figure 5-Network Manager for client........2
Figure 6-NatNetwork for Client........2
Figure 7-Client Adapter Set-up........2
Figure 8-Settin up Client Adapter........2
Figure 9-Set-Up for Gateway........3
Figure 10-Gateway Interface........3
Figure 11-Client Interface........3
Figure 12-Gateway initial forward routing and iptable rules........3
Figure 13-Initial network situation-Client........3
Figure 14-Client Initial Routing Table........4
Figure 15-DNS Resolution........4
Figure 16-Ensure Internet Reachability on Gateway........5
Figure 17-Internet Reachability from Gateway to Client........5
Figure 18- Enable Packet Forwarding........6
Figure 19-Routing on Client........6
Figure 20-Enable NAT table on Gateway........6
Figure 21-Testing Other IP address by using Ping........7
Figure 22-Update Softwares on Gateway........7
Figure 23-Update Softwares on Client........7
Figure 24- Update Softwares to current versions on Both Machines........8
Figure 25-Install tcpdump........8
Figure 26-Install Apache2 on Gateway........8
Figure 27-Disabling Firewalls........9
Figure 28-Port.cong........9
Figure 29-Configure the test-and-demo.conf file........9
Figure 30-Add the hostname “www.test-and-demo.com........9
Figure 31-Html File for Webserver........10
Figure 32-Connecting to the web server........10
Figure 33-Client Reachability to the web Server........11
Figure 34-Packet Filter Set up........11
Figure 35-Reachability from the Client to the Gateway........12
Figure 36-Gateway Sniffing Traffic to Client........13
Figure 37-Using WireShark for a better View........13
Figure 38-Changing Rules on Gateway........14
Figure 39- Blocking Internet Access........14