

Please edit the highlighted portion.

Packet Filter Firewall (iptables) Project

Student Name: Jeremy Escobar

Email: jgescoba@asu.edu

Submission Date: May 23, 2024

Class Name and Term: CSE548 Summer 2024, Session B

I. PROJECT OVERVIEW

The goal of the project is to implement two virtual machines: a Client virtual machine and a Gateway virtual machine that should be able to send packets to each other. Using iptables, we can configure a packet filtering firewall that permits or denies network traffic. Basic networking diagnostic for a pair of VMs (Client and Gateway) on an Ubuntu using the setup of two NatNetworks (10.0.2.0/24 and 10.0.1.0/24): ifconfig, route, ping and traceroute. Next, we setup the network configuration to implement firewall filtering rules to permit or deny network traffic (rc.firewall). Finally, install Apache2 web service to test filtering rules on the Gateway vm machine managing the Client's vm traffic.

II. NETWORK SETUP

Optional: Not completed by Jeremy Escobar for this lab.

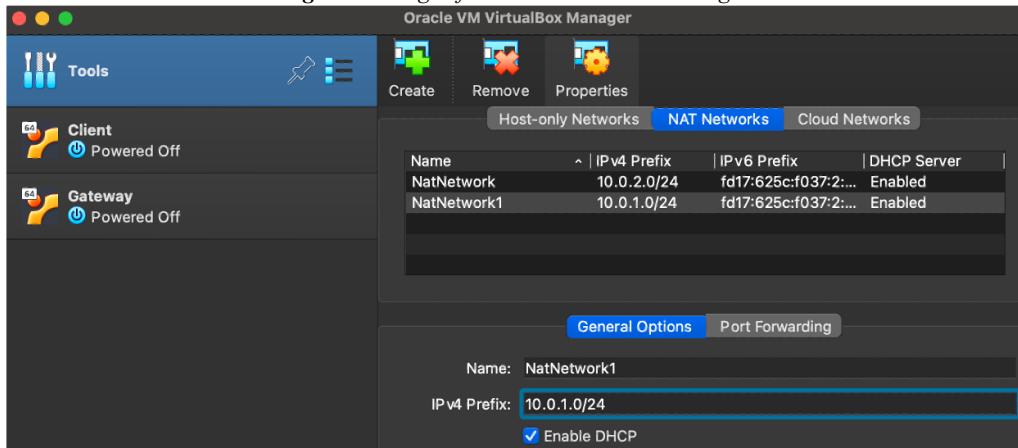
III. SOFTWARE

In this lab, we've utilized VirtualBox to set up two virtual machines (Client and Gateway). An Ubuntu VDI image, supplied by our instructor, was loaded onto these VMs. To monitor network traffic and check for packet filtering, we installed Wireshark, a packet analysis tool. Additionally, we're employing several network management tools: iptables for establishing routing protocols, Ping for testing connectivity between the client and the gateway server, and Route -n to view the current network routing configurations. Lastly, we used nmap, apache2, index.html, Firefox (loading webpages).

IV. PROJECT DESCRIPTION

Step 1: First step in the process was setting up both VM's (Client and Gateway) according to CS-SYS-00101. Next, I found the settings based on the instructions were different and found that I needed to click "file" the "network manage" to add the two require NatNetworks.

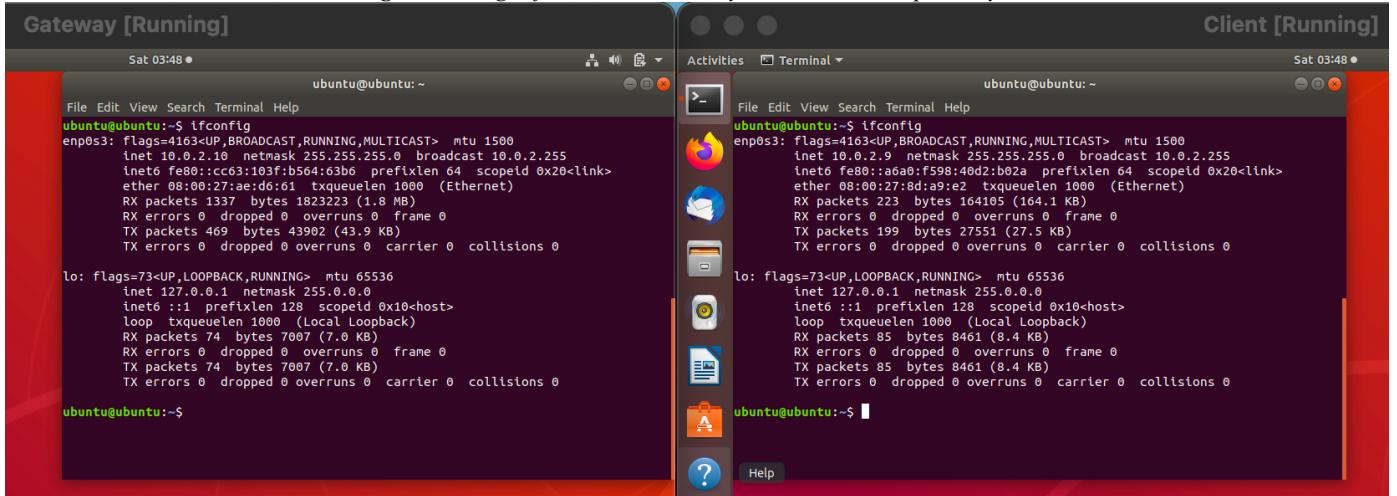
Figure 1: Image of virtual box network manager.



Please edit the highlighted portion.

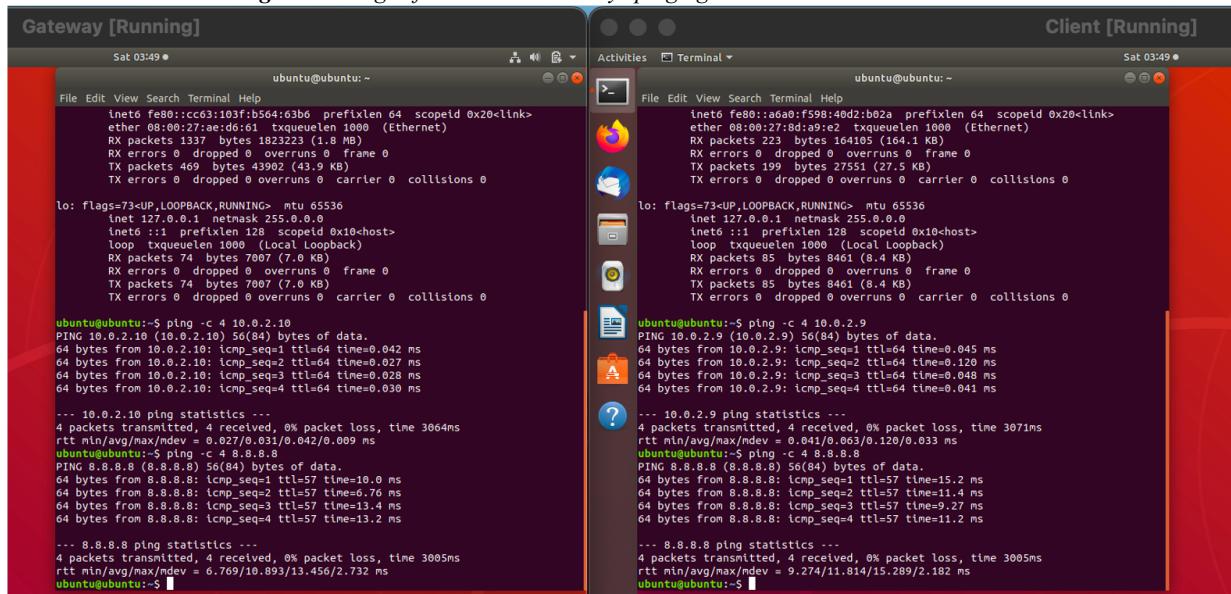
Step 2: Using CS-SYS-00101, I start by verifying the Client and Gateway IP addresses (10.0.2.9 as Client) and (10.0.2.10 as Gateway).

Figure 2: Image of Client and Gateways IP addresses respectively.



Step 3: Next tested that both Client and Gateway can *ping* 8.8.8.8 and their perspective IP addresses successfully.

Figure 3: Image of Client and Gateways pinging IP addresses and 8.8.8.8.



Please edit the highlighted portion.

Step 4: Next I checked both routing prior to creating the default gw route.

Figure 4: Image executing `route -n` on Client and Gateway verifying default settings.

```

Gateway [Running]
Sat 03:50 ●
File Edit View Search Terminal Help
ubuntu@ubuntu: ~
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 74 bytes 7007 (7.0 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 74 bytes 7007 (7.0 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ubuntu:~$ ping -c 4 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=64 time=0.028 ms
64 bytes from 10.0.2.10: icmp_seq=4 ttl=64 time=0.030 ms

--- 10.0.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.027/0.031/0.042/0.009 ms
ubuntu@ubuntu:~$ ping -c 4 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=57 time=10.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=6.76 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=13.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=57 time=13.2 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 6.769/10.893/13.456/2.732 ms
ubuntu@ubuntu:~$ route -n
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   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

Client [Running]
Sat 03:50 ●
File Edit View Search Terminal Help
ubuntu@ubuntu: ~
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 85 bytes 8461 (8.4 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 85 bytes 8461 (8.4 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ubuntu:~$ ping -c 4 10.0.2.9
PING 10.0.2.9 (10.0.2.9) 56(84) bytes of data.
64 bytes from 10.0.2.9: icmp_seq=1 ttl=64 time=0.045 ms
64 bytes from 10.0.2.9: icmp_seq=2 ttl=64 time=0.120 ms
64 bytes from 10.0.2.9: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 10.0.2.9: icmp_seq=4 ttl=64 time=0.041 ms

--- 10.0.2.9 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.041/0.063/0.120/0.033 ms
ubuntu@ubuntu:~$ ping -c 4 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=57 time=15.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=11.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=9.27 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=57 time=11.2 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 9.274/11.814/15.289/2.182 ms
ubuntu@ubuntu:~$ route -n
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   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

```

Step 5: Next I will setup the default gateway to point to Gateway VM from Client VM.

Figure 5: Image executing `sudo route add default gw 10.0.2.10` as route for Client VM.

```

ubuntu@ubuntu:~$ sudo route add default gw 10.0.2.10 enp0s3
[sudo] password for ubuntu:
ubuntu@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         10.0.2.10     0.0.0.0        UG   0      0      0 enp0s3
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

```

Step 6: Next, After setting up the default gateway, I checked path with `route -n`, then tried to `ping 8.8.8.8` and failed.

Figure 6: Showing that after setting up default gateway to 10.0.2.10 cannot ping 8.8.8.8 on Client VM.

```

ubuntu@ubuntu:~$ sudo route add default gw 10.0.2.10 enp0s3
[sudo] password for ubuntu:
ubuntu@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         10.0.2.10     0.0.0.0        UG   0      0      0 enp0s3
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
ubuntu@ubuntu:~$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3062ms

ubuntu@ubuntu:~$ 

```

Please edit the highlighted portion.

Step 7: Next I `sudo nano /etc/sysctl.conf` the conf file to uncomment `net.ipv4.ip_forward=1`

Figure 7: Uncommented the net.ipv4.ip_froward below on Gateway VM.

```

Gateway [Running]
Sat 03:58 ●
ubuntu@ubuntu: ~
File Edit View Search Terminal Help
GNU nano 2.9.3          /etc/sysctl.conf

#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Step 8: Next checked to make sure that on Gateway ip_forward was setup with `cat` command.

Figure 8: Image of command being used to verify forwarding is now 1 not 0.

```

ubuntu@ubuntu:~$ cat /proc/sys/net/ipv4/ip_forward
1
ubuntu@ubuntu:~$
```

Please edit the highlighted portion.

Step 9: Next setup in the gateway the iptables to allow Client vm to access the internet fully again for the time being CS-NET-00002.

Figure 9: Client is on the right side now able to ping 8.8.8.8 again, and on the left Gateway VM enabling iptables.

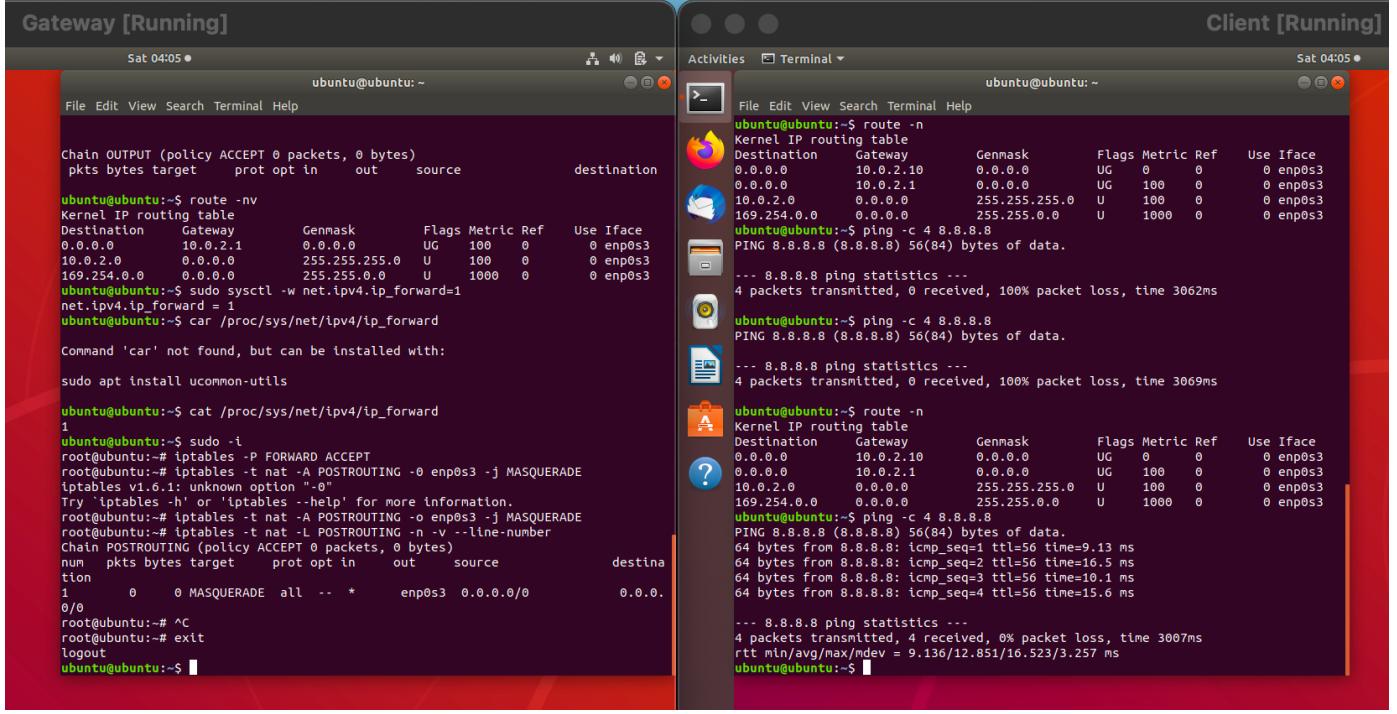
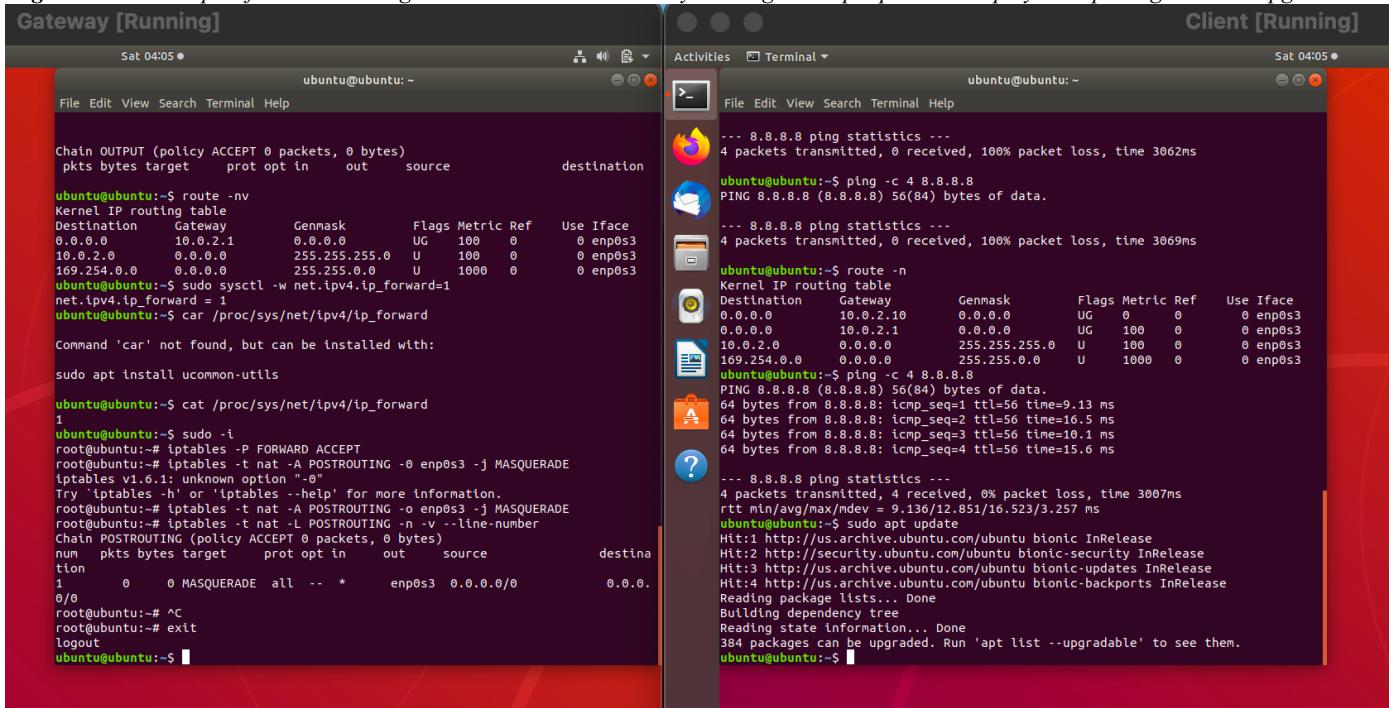


Figure 10: Further proof Client VM is again able to access internet by running sudo apt update. It displays 384 packages can be upgraded.



Please edit the highlighted portion.

Step 10: Next I installed apache2 on the Gateway Vm for later use.

Figure 11: Installing Apache2 on Gateway VM on left side terminal.

```

Gateway [Running]                                         Client [Running]
Sat 04:10 ●                                         Sat 04:10 ●
ubuntu@ubuntu:~                                         ubuntu@ubuntu:~
File Edit View Search Terminal Help                   File Edit View Search Terminal Help
Reading package lists... Done                         --- 8.8.8.8 ping statistics ---
Building dependency tree                           4 packets transmitted, 0 received, 100% packet loss, time 3062ms
Reading state information... Done                    ubuntu@ubuntu:~$ ping -c 4 8.8.8.8
The following packages were automatically installed and are no longer required:
  distro-info python3-click python3-colorama
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  tcpdump
1 upgraded, 0 newly installed, 0 to remove and 383 not upgraded.
Need to get 364 kB of archives.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 tcpdump amd64 4.9.3-0ubuntu0.18.04.3 [364 kB]
Fetched 364 kB in 5s (67.9 kB/s)
(Reading database... 166148 files and directories currently installed.)
Preparing to unpack .../tcpdump_4.9.3-0ubuntu0.18.04.3_amd64.deb ...
Unpacking tcpdump (4.9.3-0ubuntu0.18.04.3) over (4.9.3-0ubuntu0.18.04.1) ...
Setting up tcpdump (4.9.3-0ubuntu0.18.04.3) ...
Installing new version of config file /etc/apparmor.d/usr.sbin.tcpdump ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ubuntu:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: drop-in)
  Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2.conf
  Active: active (running) since Sat 2024-05-11 03:45:40 MST; 24min ago
    Main PID: 2268 (apache2)
      Tasks: 55 (limit: 2327)
     CGroup: /system.slice/apache2.service
             ├─2268 /usr/sbin/apache2 -k start
             ├─2269 /usr/sbin/apache2 -k start
             ├─2270 /usr/sbin/apache2 -k start
             └─2271 /usr/sbin/apache2 -k start

May 11 03:45:40 ubuntu systemd[1]: Starting The Apache HTTP Server...
May 11 03:45:40 ubuntu apachectl[2257]: AH00558: apache2: Could not reliably det
May 11 03:45:40 ubuntu systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)

ubuntu@ubuntu:~$ route -n
Kernel IP routing table
Destination      Gateway        Genmask        Flags Metric Ref  Use Iface
0.0.0.0          10.0.2.10    0.0.0.0        UG   0      0      0 enp0s3
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
ubuntu@ubuntu:~$ ping -c 4 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=56 time=9.13 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=16.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=10.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=15.6 ms

ubuntu@ubuntu:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security 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
Building dependency tree
Reading state information... Done
384 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ubuntu:~$ 
```

Step 11: Then I reviewed the default listen settings for Apache2.

Figure 12: Confirmed in left side terminal Gateway VM, listen is 80 (port).

```

Gateway [Running]                                         Client [Running]
Sat 04:13 ●                                         Sat 04:13 ●
ubuntu@ubuntu:~                                         ubuntu@ubuntu:~
File Edit View Search Terminal Help                   File Edit View Search Terminal Help
GNU nano 2.9.3                                         --- 8.8.8.8 ping statistics ---
                                         4 packets transmitted, 0 received, 100% packet loss, time 3062ms
/etc/apache2/ports.conf                               ubuntu@ubuntu:~$ ping -c 4 8.8.8.8
# 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 80                                           PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

<IfModule ssl_module>
  Listen 443
</IfModule>

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

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

                                         --- 8.8.8.8 ping statistics ---
                                         4 packets transmitted, 0 received, 100% packet loss, time 3069ms

ubuntu@ubuntu:~$ route -n
Kernel IP routing table
Destination      Gateway        Genmask        Flags Metric Ref  Use Iface
0.0.0.0          10.0.2.10    0.0.0.0        UG   0      0      0 enp0s3
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
ubuntu@ubuntu:~$ ping -c 4 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=56 time=9.13 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=16.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=10.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=15.6 ms

ubuntu@ubuntu:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security 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
Building dependency tree
Reading state information... Done
384 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ubuntu:~$ 
```

Please edit the highlighted portion.

Step 12: I then updated default port IP addresses to the correct IP addresses on port 80.

Figure 13: I updated the defaults in the ports.conf file below (127.0.0.1:80 and 10.0.2.10:80).

```

Gateway [Running]
Sat 04:14 •
ubuntu@ubuntu: ~
File Edit View Search Terminal Help
GNU nano 2.9.3      /etc/apache2/ports.conf      Modified
# 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 10.0.2.10:80
<IfModule ssl_module>
    Listen 443
</IfModule>
<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

Client [Running]
Sat 04:15 •
ubuntu@ubuntu: ~
File Edit View Search Terminal Help
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3062ms
ubuntu@ubuntu:~$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3069ms
ubuntu@ubuntu:~$ route -n
Kernel IP routing table
Destination      Gateway      Genmask      Flags Metric Ref  Use Iface
0.0.0.0          10.0.2.10   0.0.0.0      UG        0      0      0 enp0s3
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
ubuntu@ubuntu:~$ ping -c 4 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=56 time=9.13 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=16.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=10.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=15.6 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 9.136/12.851/16.523/3.257 ms
ubuntu@ubuntu:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security 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
Building dependency tree
Reading state information... Done
384 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ubuntu:~$ 
```

Step 13: After doing that I used sudo -i to go in and update the default index.html file for the Apache2 web server and visited the sites from **localhost** on the Gateway VM, then visited **10.0.2.9** on the Client VM side see below:

Figure 14: Below is use sudo -I to go to the default location of the index.html file and create the file..

```

ubuntu@ubuntu:~$ sudo -i
[sudo] password for ubuntu:
root@ubuntu:~# cd /usr/www/html
-bash: cd: /usr/www/html: No such file or directory
root@ubuntu:~# ls
root@ubuntu:~# cd /var/www/html
root@ubuntu:/var/www/html# ls
index.html
root@ubuntu:/var/www/html# nano index.html
root@ubuntu:/var/www/html# 
```

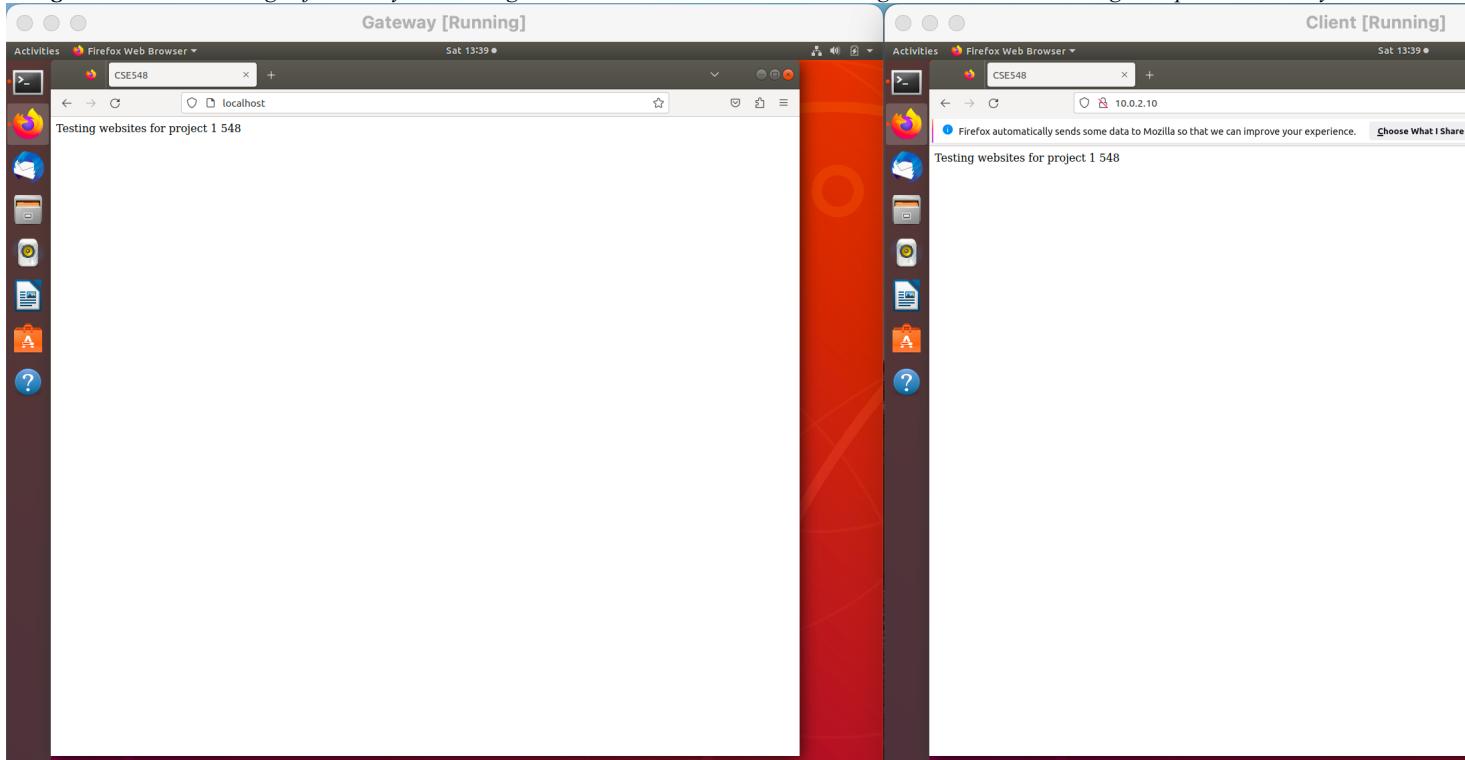
Figure 15: Below is image of the index.html I updated for use in Gateway and Client Vm's.

```

Gateway [Running]
Sat 19:18 •
root@ubuntu: /var/www/html
File Edit View Search Terminal Help
GNU nano 2.9.3      index.html
<html>
<head>
    <title> CSE548</title>
</head>
<body>
    <p> Testing websites for project 1 548 </p>
</body>
</html>
```

Please edit the highlighted portion.

Figure 16: Below is image of Gateway Vm visiting localhost, and then Client VM visiting 10.0.2.10 and behaving as expected correctly.



Step 14: Then I disabled access to the internet to begin setting up for the firewall settings.

Figure 17: Below enabling DROP on FORWARD, INPUT and OUTPUT.

```
ubuntu@ubuntu:~$ sudo systemctl restart apache2.service
ubuntu@ubuntu:~$ sudo -i
root@ubuntu:~# iptables -p INPUT DROP
iptables v1.6.1: unknown protocol "input" specified
Try `iptables -h` or `iptables --help` for more information.
root@ubuntu:~# iptables -P INPUT DROP
root@ubuntu:~# iptables -P OUTPUT DROP
root@ubuntu:~# iptables -P FORWARD DROP
root@ubuntu:~# iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target     prot opt in     out     source               destination
                                         destination

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

Chain OUTPUT (policy DROP 2 packets, 656 bytes)
pkts bytes target     prot opt in     out     source               destination
root@ubuntu:~#
```

Please edit the highlighted portion.

Step 15: After that the Client VM can no longer ping or visit the web see below.

Figure 18: Client VM unable to visit/ping 8.8.8.8, 10.0.2.10 and web.

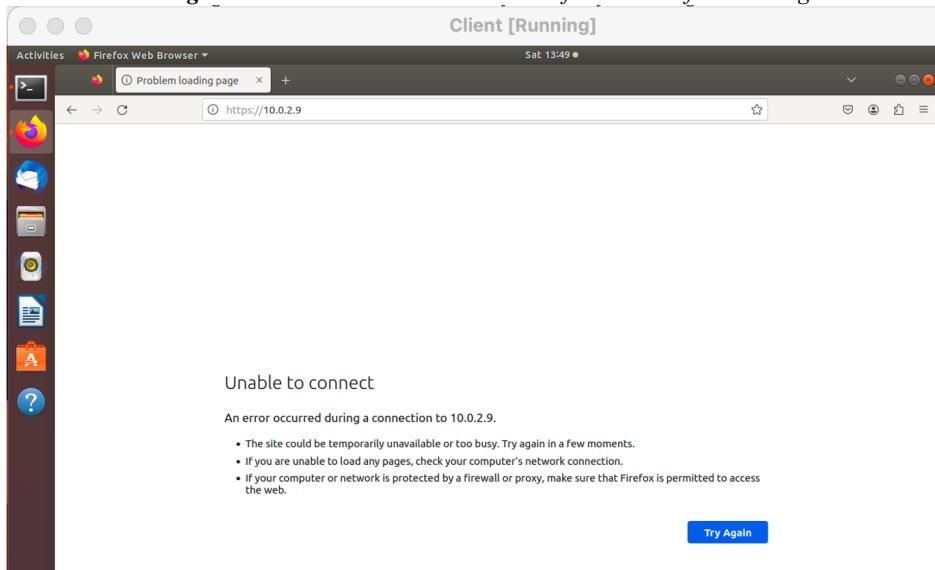
```
ubuntu@ubuntu:~$ ping -c 4 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
^C
--- 10.0.2.10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

ubuntu@ubuntu:~$ 
ubuntu@ubuntu:~$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2039ms

ubuntu@ubuntu:~$ 
```

Figure 19: Cannot access the web either from client after enabling DROP.



Step 16: Next locally I updated the firewall given to us rc.firewall and emailed it to Gateway VM via google so I could download it on the vm machine Gateway.

Figure 20: Setting up OUTPUT chain rule below.

```
# Allowed ping from client and server
$IPTABLES -A INPUT -p icmp -j ACCEPT
#####
# 4.3 OUTPUT chain
# Jeremy Rule below
$IPTABLES -A OUTPUT -p TCP --sport 80 -o $Client_NET_IFACE -d $WEB_IP_ADDRESS -j ACCEPT
```

Figure 21: Setting up the default FORWARD rules below:

```
#####
# 4.2 FORWARD chain
#
#
# Provide your forwarding rules below
#
# Jeremy Updated example of checking bad tcp packets
$IPTABLES -A FORWARD -p icmp -d $Client_NET_IP -o $Client_NET_IFACE -s 8.8.8.8 -j ACCEPT
$IPTABLES -A FORWARD -p ICMP -s $Client_NET_IP -i $Client_NET_IFACE -d 8.8.8.8 -j ACCEPT
```

Please edit the highlighted portion.

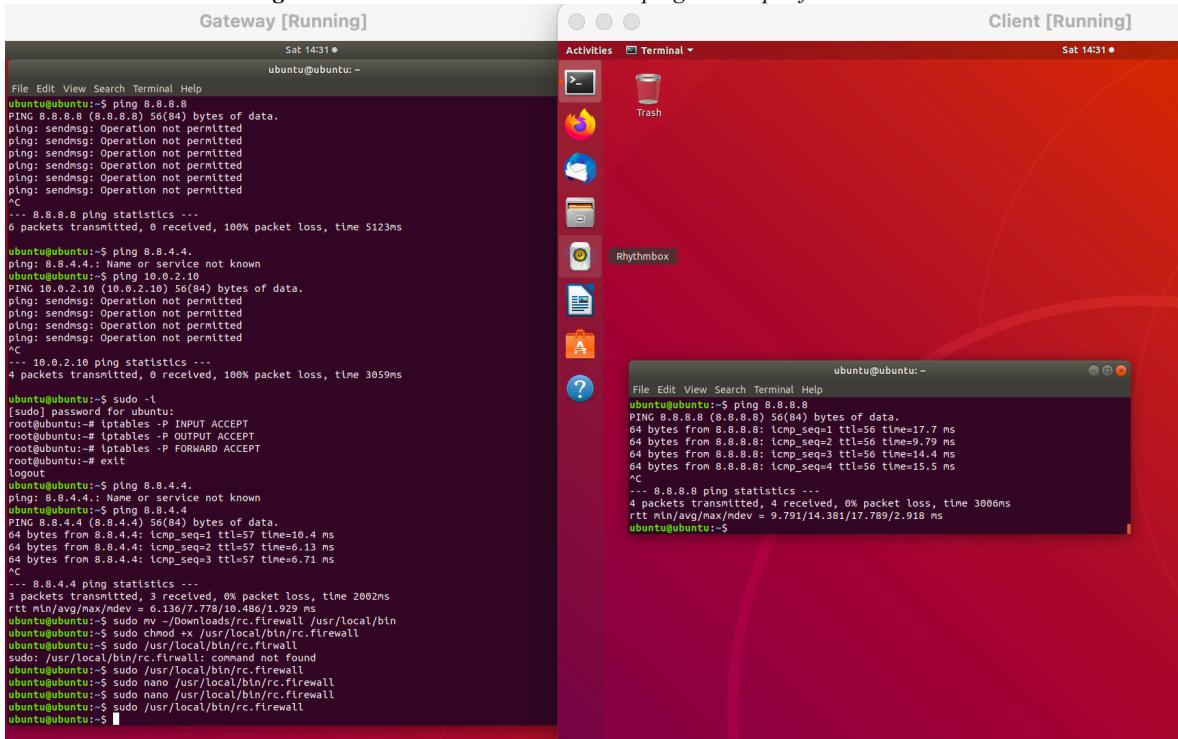
Figure 22: Setting up the default POSTROUTING rules below:

```
#####
# 5.2 POSTROUTING chain.
#
#
# Provide your NAT PREROUTING rules (packets go to the internet domain)
# Add your own rule below to only allow ping from client to 8.8.8.8 on internet

# Example: Allow client node to access to all Internet using masquerade
$IPTABLES -t nat -A POSTROUTING -p icmp -o $Internet_IFACE -d 8.8.8.8 -j MASQUERADE
```

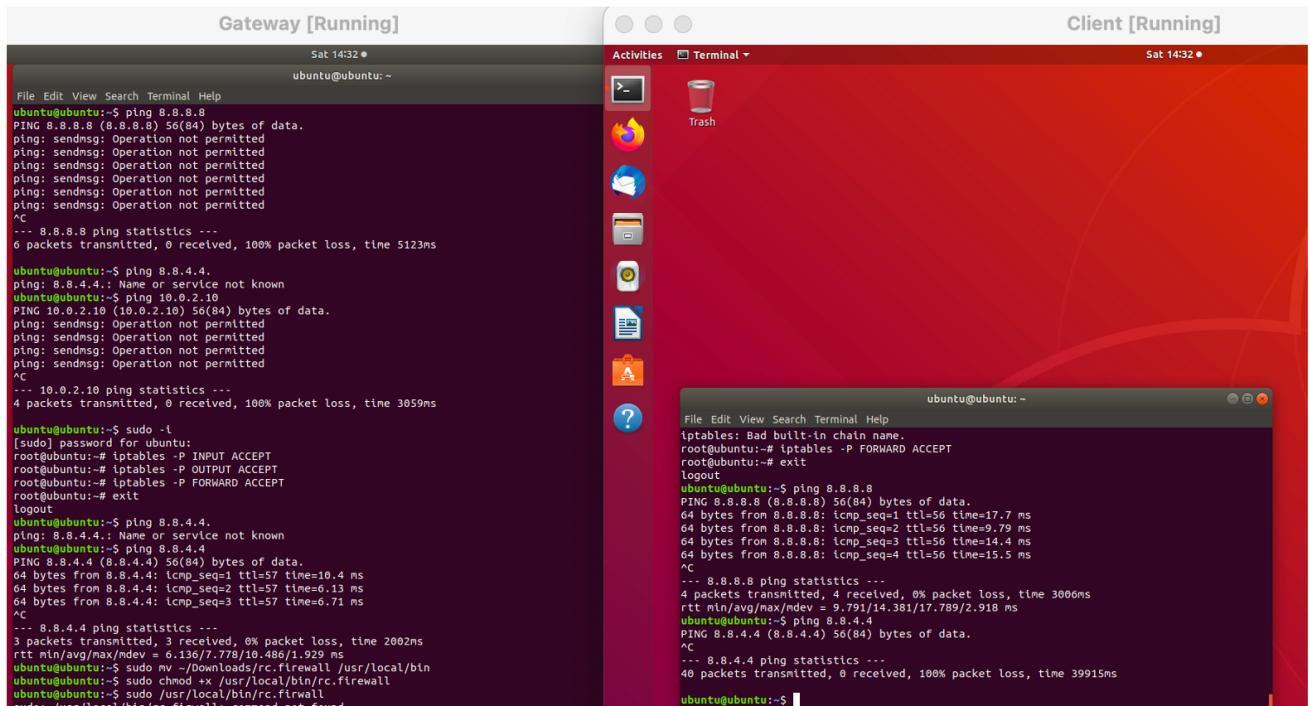
Step 17: Next when trying to *ping* 8.8.8.8 it worked as expected see below in the Client VM now.

Figure 23: Client VM is now allowed to ping 8.8.8.8 per firewall rules.



Please edit the highlighted portion.

Figure 24: Attempts to ping 8.8.4.4. do not work with 100% packet loss and all other sites fail as well.



Step 18: Next, I did a UDP and TCP checks below. All worked as expected for both, TCP was good and UDP all ports where open and filtered.

Figure 25: TCP nmap scan on Gateway VM.

```

ubuntu@ubuntu:~$ sudo nmap -sT -p- 10.0.2.10

Starting Nmap 7.60 ( https://nmap.org ) at 2024-05-11 16:13 MST
Nmap scan report for ubuntu (10.0.2.10)
Host is up (0.00012s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 2.83 seconds
ubuntu@ubuntu:~$ sudo nmap -sU -p- -T4 10.0.2.10

```

Figure 26: UDP nmap scan below on Gateway VM

```

ubuntu@ubuntu:~$ sudo nmap -sU -p- -T4 10.0.2.10
[sudo] password for ubuntu:

Starting Nmap 7.60 ( https://nmap.org ) at 2024-05-11 15:09 MST
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 47.98% done; ETC: 15:09 (0:00:05 remaining)
Stats: 0:00:12 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 75.66% done; ETC: 15:09 (0:00:04 remaining)
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 76.39% done; ETC: 15:09 (0:00:04 remaining)
Stats: 0:00:26 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 77.01% done; ETC: 15:09 (0:00:04 remaining)
Stats: 0:00:33 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 78.02% done; ETC: 15:09 (0:00:04 remaining)
Stats: 0:01:49 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 86.22% done; ETC: 15:11 (0:00:18 remaining)
Stats: 0:02:21 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 89.62% done; ETC: 15:11 (0:00:16 remaining)
Stats: 0:03:12 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 95.41% done; ETC: 15:12 (0:00:09 remaining)
Stats: 0:03:32 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 97.93% done; ETC: 15:12 (0:00:05 remaining)
Stats: 0:03:52 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 15:12 (0:00:00 remaining)
Stats: 0:04:26 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 15:13 (0:00:00 remaining)
Nmap scan report for ubuntu (10.0.2.10)
Host is up (0.0000070s latency).
Not shown: 65531 closed ports
PORT      STATE      SERVICE
68/udp    open|filtered  dhcpc
631/udp   open|filtered  ipp
5353/udp  open|filtered  zeroconf
54174/udp open|filtered  unknown

Nmap done: 1 IP address (1 host up) scanned in 293.53 seconds
ubuntu@ubuntu:~$ 

```

Please edit the highlighted portion.

V. CONCLUSION

This lab proved to be quite challenging and difficult, particularly in how it pushed my understanding of iptables. I gained hands-on experience in configuring a gateway to route all traffic from a designated host and in tweaking the iptables rules to observe the flow of packets under the conditions we established. The network tools I discussed in the Software section of my report were instrumental in analyzing the behavior of both the Client and Gateway VMs. Notably, Wireshark was invaluable for scrutinizing the packet exchanges from the client to the gateway. Personally, I encountered few hurdles in setting up the lab. However, I did face challenges with the provided Project1 VDI. I eventually discovered that cloning the Client VM would allow me to create two functional VMs. Another snag I hit involved running the modified rc.firewall script, which initially returned a 'command not found' error.

Appendix B: Attached files

VI. REFERENCES