Ronn Joseph
IT-620

Semester Project- Firewall

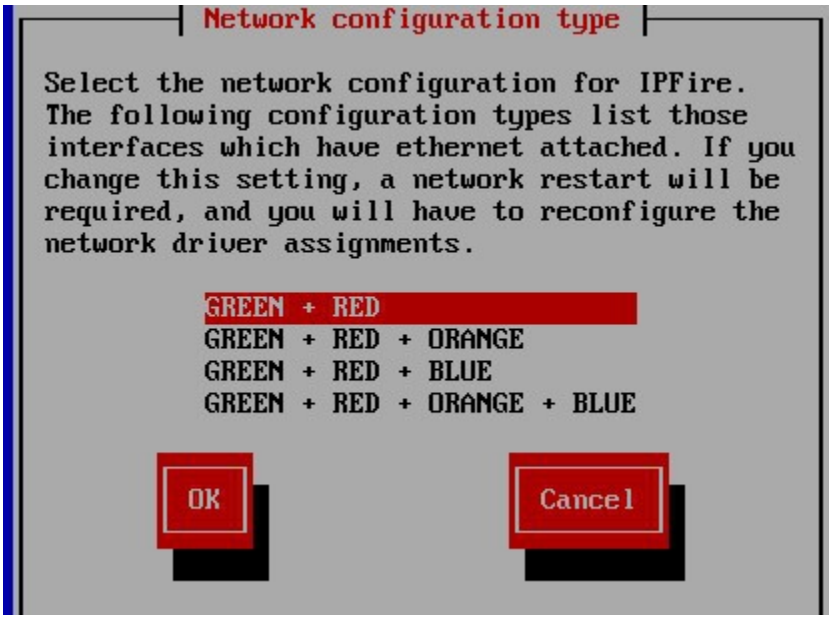
For the semester project, I am going to implement the project in virtualization through VirtualBox. I decided to implement IPFire as my Firewall VM. The operating system I used for client VM (exterior) is Debian 9 and server (interior) VM is the Ubuntu.

Configuring Firewall VM

I named the firewall VM: Firewall (IPFire) with the recommended default settings: 512MB RAM, 8GB VDI. I went through installation with mostly default settings.

First, I changed the network adapters to the following setting:

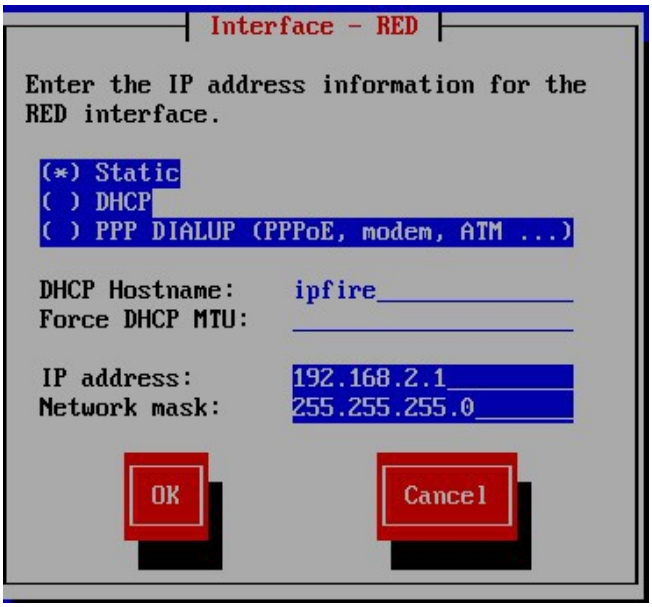
Next, I started the firewall VM and ran the default installation after which I had to reboot. Then, I changed to network configuration settings as follows:



The colors have the following meanings:

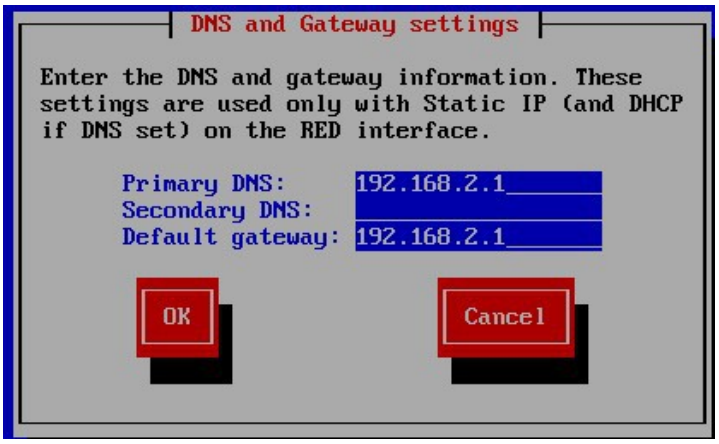
	item	description
Red	WAN	External network, Connected to the Internet (typically a connection to your ISP)
Green	LAN	Internal/Private network, connected locally
Orange	DMZ	The DeMilitarized Zone, an unprotected/Server network accessible from the internet
Blue	WLAN	Wireless Network, A separate network for wireless clients

The RED Interface was configured as follows:



The GREEN Interface was configured as follows:

Lastly, the DNS and Gateway settings was configured as follows:

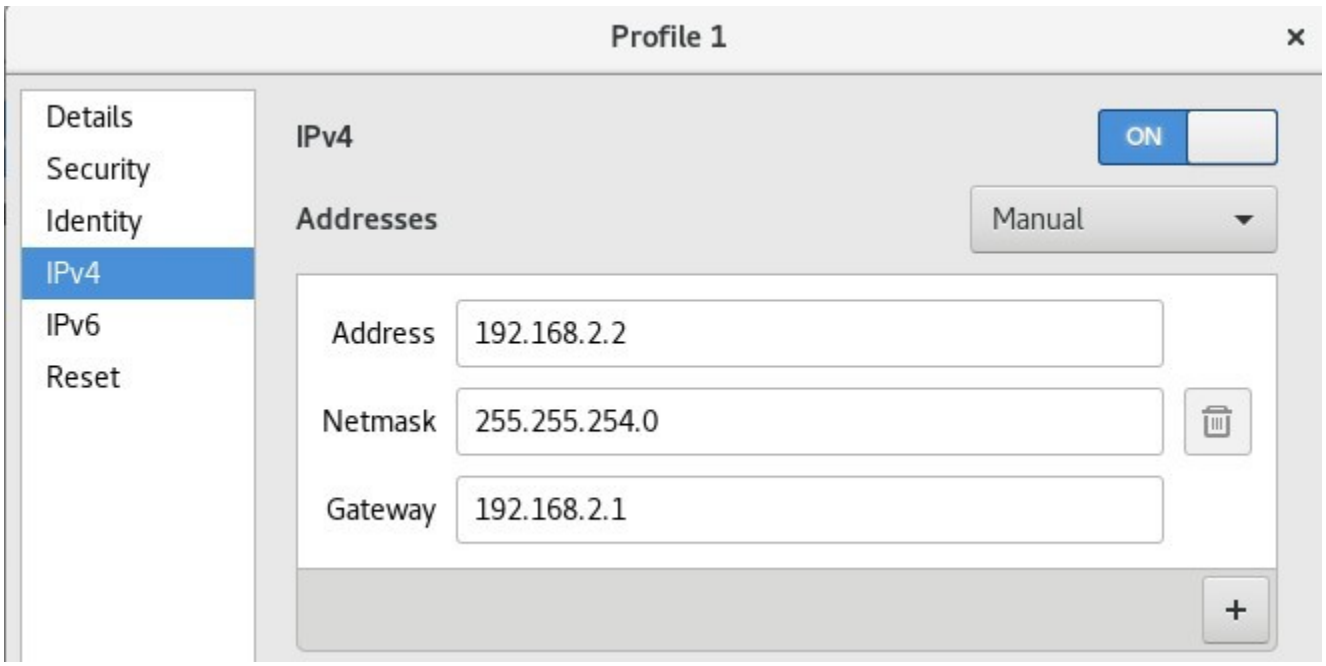


Configuring Client/Exterior VM 192.168.2.1

I named the exterior VM: Debian with the recommended default settings: 1GB RAM, 8GB VDI. I went through graphical installation with mostly default settings. I used debian as username.

First, I changed the network adapter to the following setting:

Then I started the VM and logged on and changed the connection settings to follow Profile 1:



Then, to confirm my changes, I went to terminal and I installed net-tools in order to use ifconfig to view in terminal:
sudo apt-get install net-tools

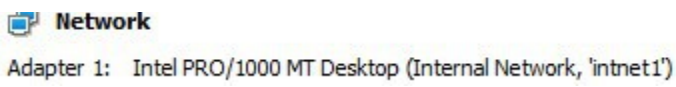
Next, I ran *ifconfig:*

```
debian@debian:~$ sudo ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.2  netmask 255.255.254.0  broadcast 192.168.3.255
        inet6 fe80::34de:78:442b:62b1  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1d:23:62  txqueuelen 1000  (Ethernet)
        RX packets 1185  bytes 1563541 (1.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 433  bytes 38674 (37.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Configuring Server/Interior VM 192.168.1.1

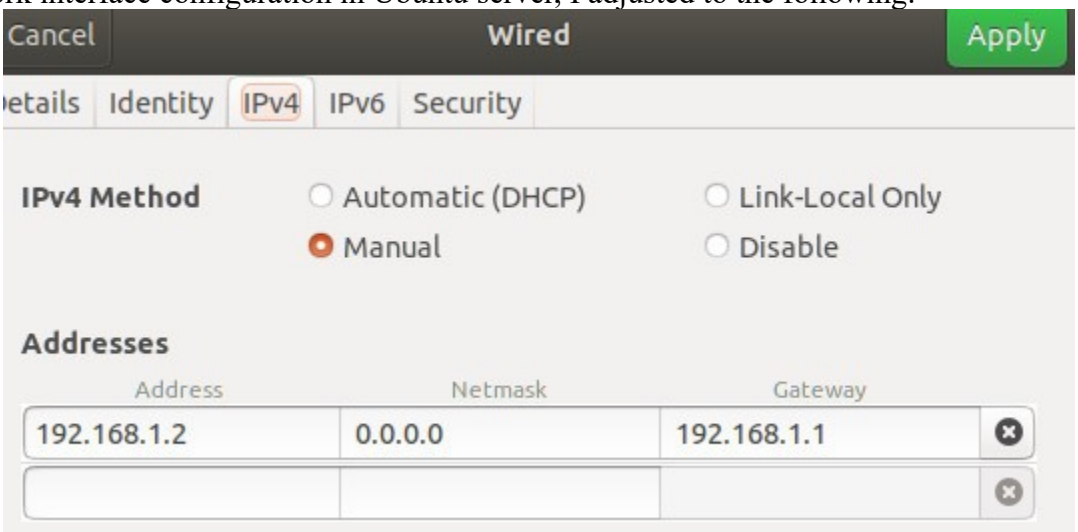
I named the exterior VM: Ubuntu Server with the recommended default settings: 1GB RAM, 10GB VDI. I went through installation with mostly default settings.

First, I changed the network adapter to the following setting:

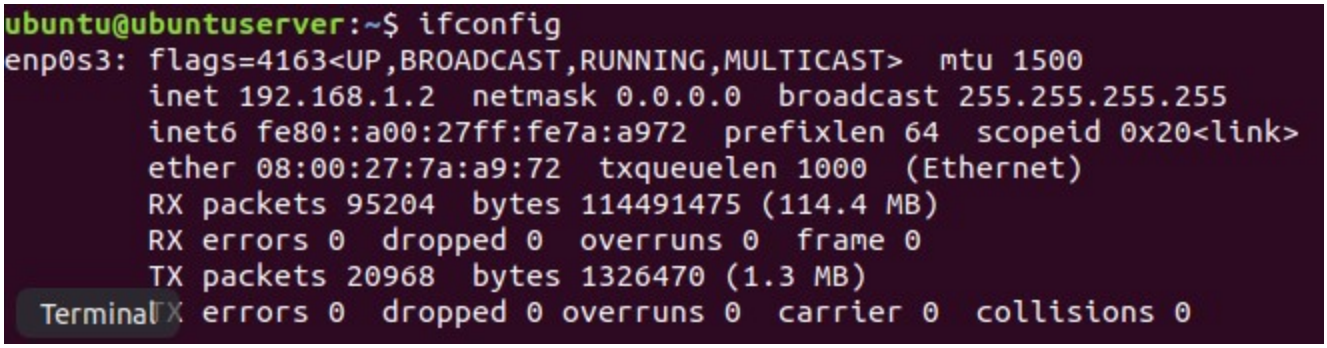


During installation, I created the profile for Ubuntu and give the username ubuntu with device name ubuntuserver.

For network interface configuration in Ubuntu server, I adjusted to the following:



After completing installation of firewall, the system rebooted after which I arrived at the terminal. I entered *ifconfig* to confirm configuration.



```
ubuntu@ubuntuserver:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.2  netmask 0.0.0.0  broadcast 255.255.255.255
        inet6 fe80::a00:27ff:fe7a:a972  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7a:a9:72  txqueuelen 1000  (Ethernet)
        RX packets 95204  bytes 114491475 (114.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20968  bytes 1326470 (1.3 MB)
Terminal TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Testing Connectivity

Currently the networks are configured as follows:

192.168.2.0/24 ---->|(Red-WAN port) Firewall (Green-LAN port)|<----192.168.1.0/24

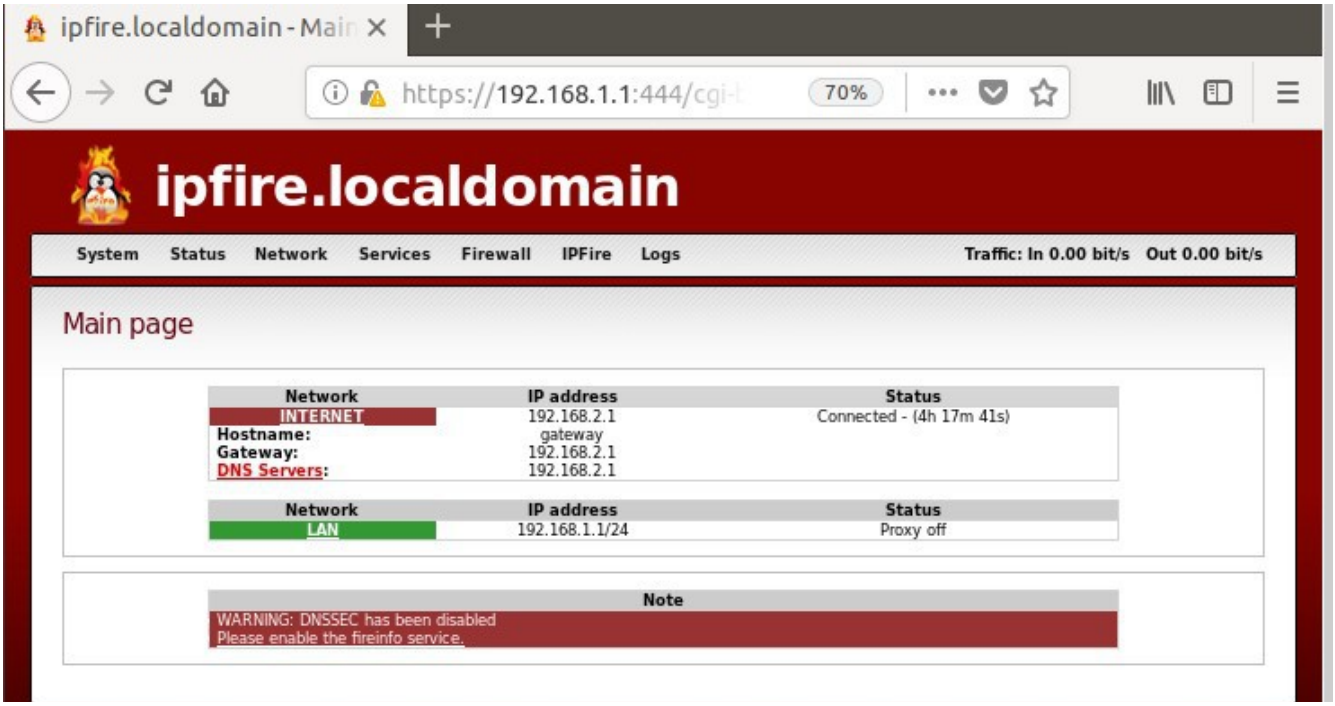
After all VM instances are setup and running, I tested connectivity by pinging from client to firewall.

```
debian@debian:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.732 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.265 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.744 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.265/0.503/0.744/0.235 ms
```

I also tested connectivity by pinging from server to firewall.

```
ubuntu@ubuntuserver:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.367 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.274 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.330 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.333 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.274/0.326/0.367/0.033 ms
```

I can access the firewall through the server's web interface via firefox at: https://192.168.1.1:444 and create security exception and certificate for the firewall admin page.



Implementing Firewall properties and Verfication

Rule 1: Block external ICMP messages (ping, tracerout, etc), but should allow these from interior clients

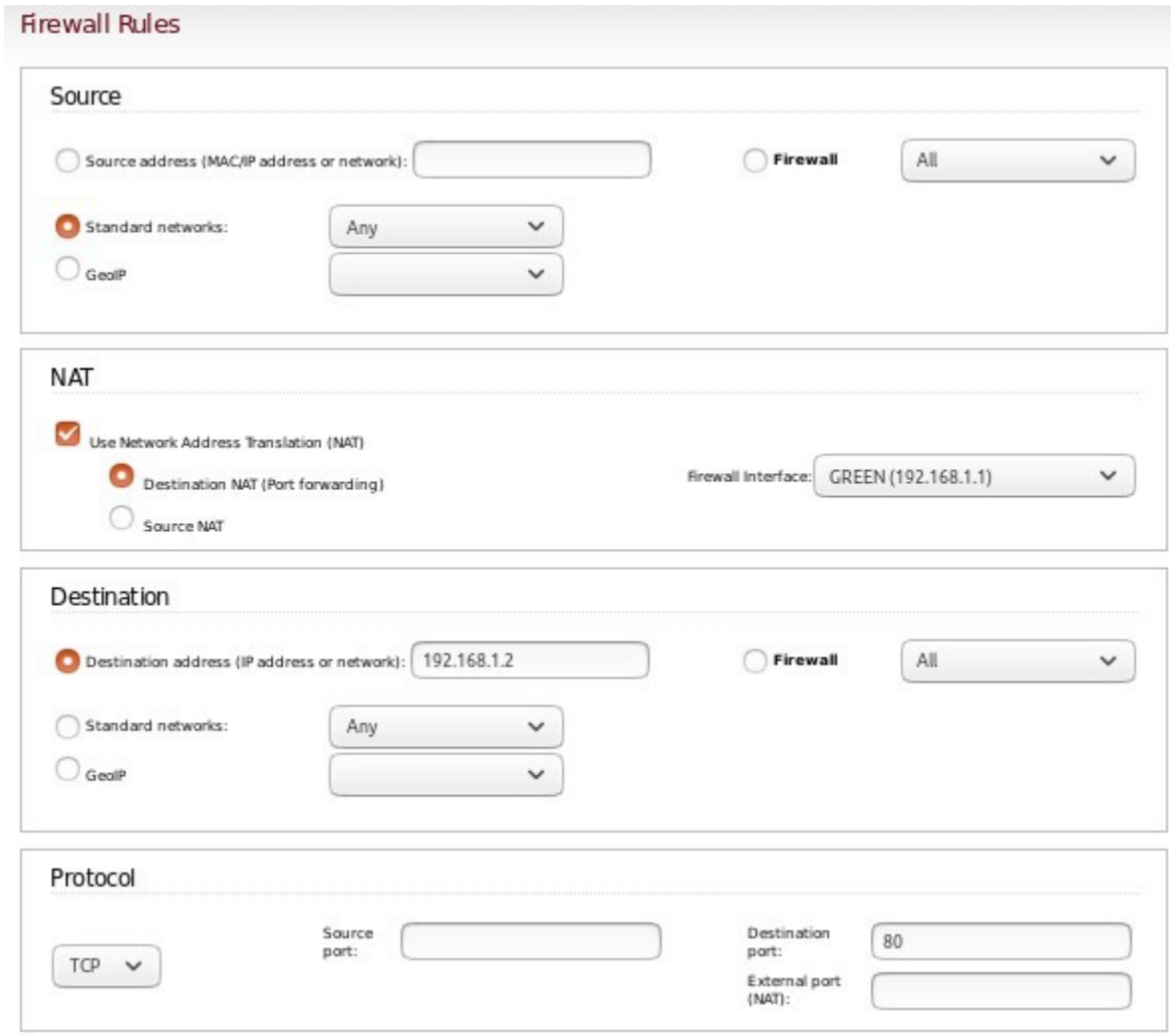
After rule 1 was applied, I attempted to ping server and received the following:

```
debian@debian:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Destination Port Unreachable
From 192.168.2.1 icmp_seq=2 Destination Port Unreachable
From 192.168.2.1 icmp_seq=3 Destination Port Unreachable
From 192.168.2.1 icmp_seq=4 Destination Port Unreachable
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3011ms
```

The server is still allowed to ping client as follows:

```
ubuntu@ubuntuserver:~$ ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=63 time=0.513 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=63 time=1.39 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=63 time=1.44 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=63 time=1.36 ms
^C
--- 192.168.2.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3021ms
rtt min/avg/max/mdev = 0.513/1.178/1.441/0.387 ms
```

Rule 2: Allow port 80 requests to the interior client

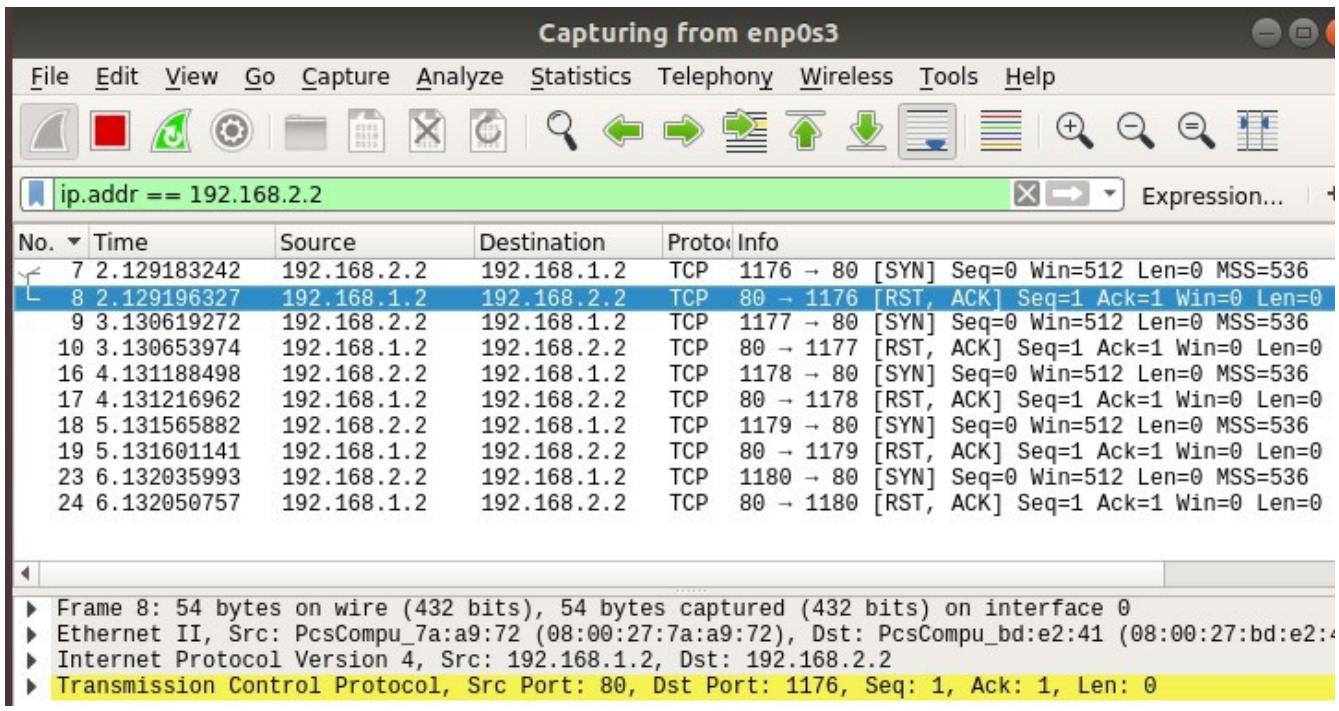


On client side, I installed hping3 to send SYN packets through port 80.

```
debian@debian:~$ sudo hping3 192.168.1.2 -c 5 -p 80 -S
HPING 192.168.1.2 (enp0s3 192.168.1.2): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.2 ttl=63 DF id=29250 sport=80 flags=RA seq=0 win=0 rtt=14.6 ms
len=46 ip=192.168.1.2 ttl=63 DF id=29402 sport=80 flags=RA seq=1 win=0 rtt=5.9 ms
len=46 ip=192.168.1.2 ttl=63 DF id=29623 sport=80 flags=RA seq=2 win=0 rtt=5.0 ms
len=46 ip=192.168.1.2 ttl=63 DF id=29746 sport=80 flags=RA seq=3 win=0 rtt=8.9 ms
len=46 ip=192.168.1.2 ttl=63 DF id=29908 sport=80 flags=RA seq=4 win=0 rtt=7.9 ms

--- 192.168.1.2 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5.0/8.5/14.6 ms
```

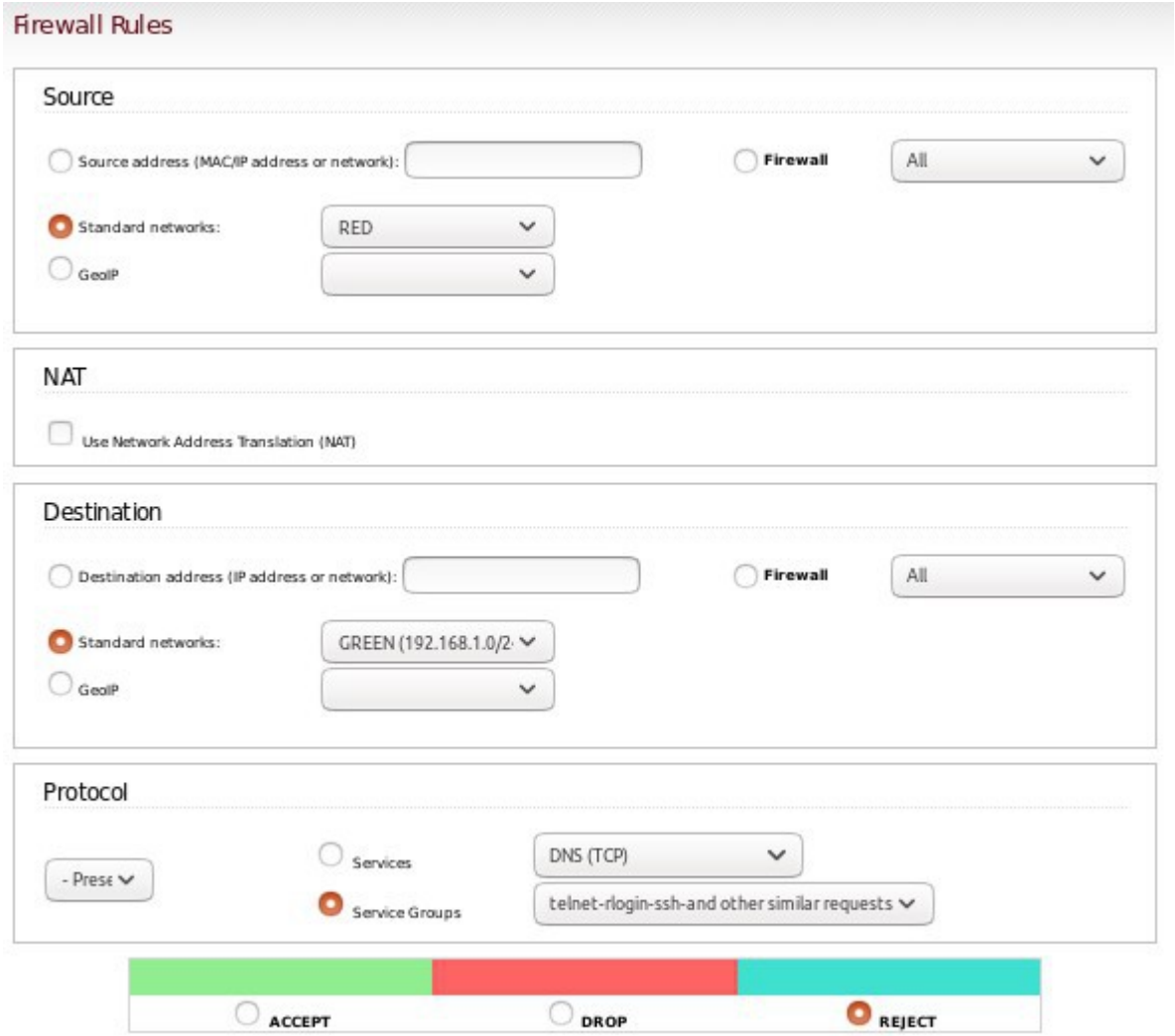
On the server side, I installed wireshark to see if packets from the client came through port 80.



Rule 3: Block external telnet, rlogin, and other similar requests

In the firewall interface, I created Service Group: "telnet, rlogin, ssh and other similar requests" to add all the requested protocols to be added to one rule as follows:

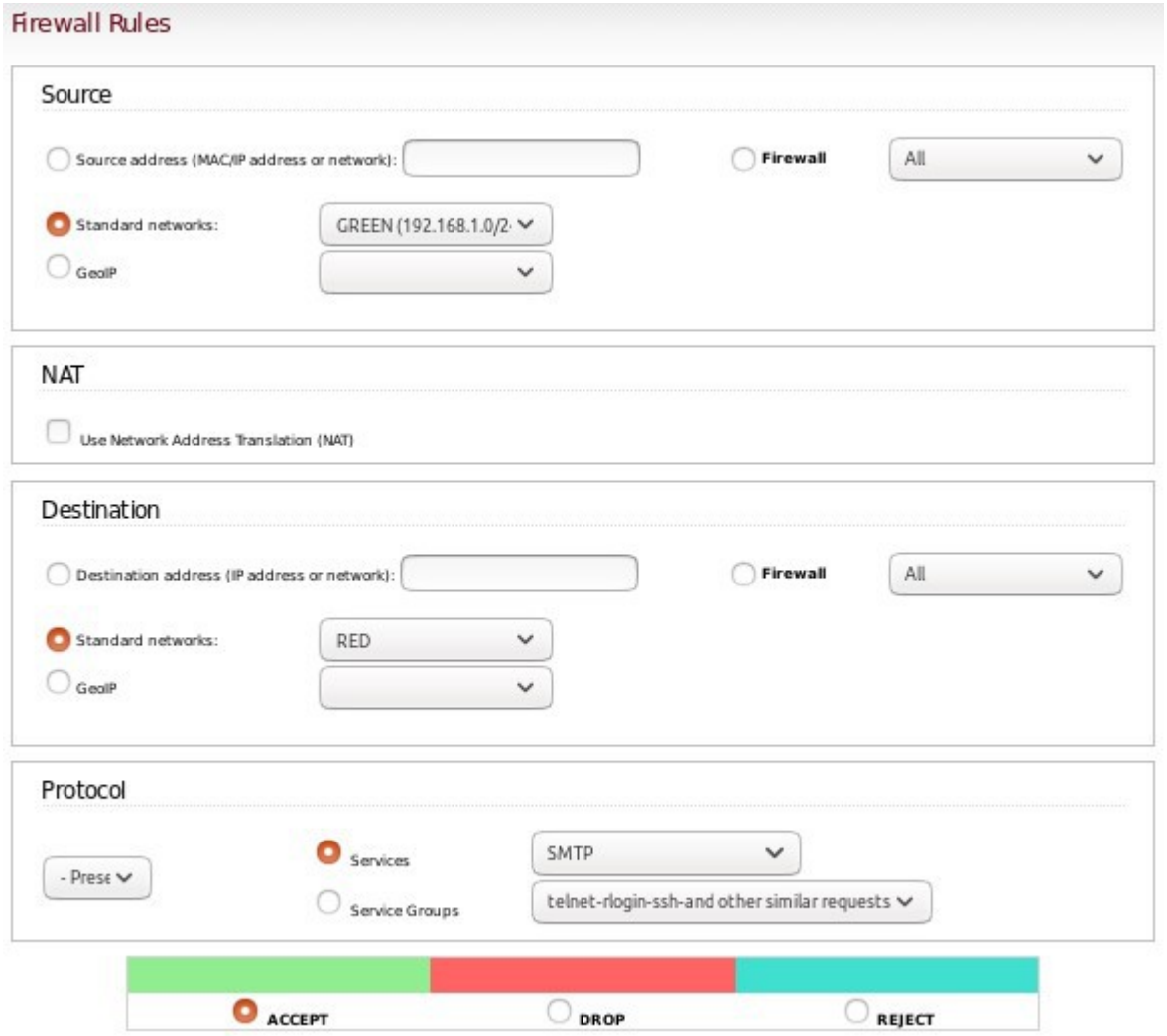
The following would be applied to rule 3:



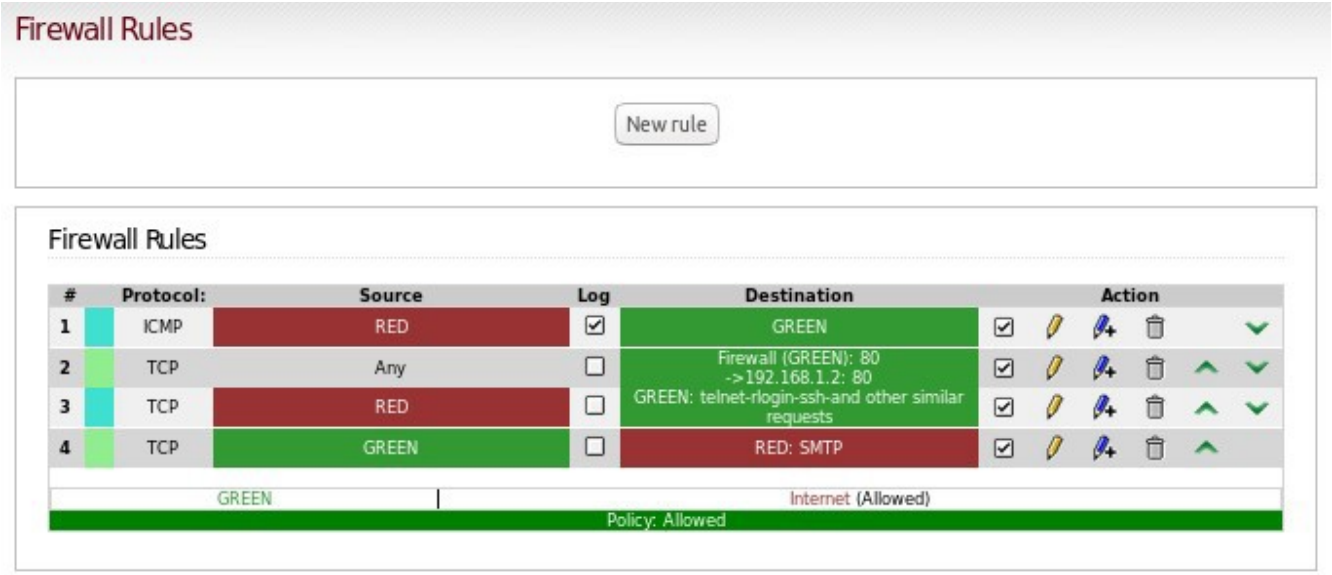
After application, I tried to connect from client to server:

```
debian@debian:~$ ssh 192.168.1.2
ssh: connect to host 192.168.1.2 port 22: Connection refused
debian@debian:~$ telnet 192.168.1.2
Trying 192.168.1.2...
telnet: Unable to connect to remote host: Connection refused
debian@debian:~$ rsh 192.168.1.2
ssh: connect to host 192.168.1.2 port 22: Connection refused
debian@debian:~$ rlogin 192.168.1.2
ssh: connect to host 192.168.1.2 port 22: Connection refused
debian@debian:~$
```

Rule 4: Allow internal messages using SMTP to be sent through the firewall



Firewall Rules List in conclusion:



Problems faced

Initially, I attempted to make the server VM ubuntu server which was completely command line interface which made it difficult for me to access web interface of the firewall through the server VM.

When I tried to apply rule 2 to practice, I was sending hping3 tcp packets with no tags which gave me no response. I then realized I need to apply SYN tag in order to get a ACK response.