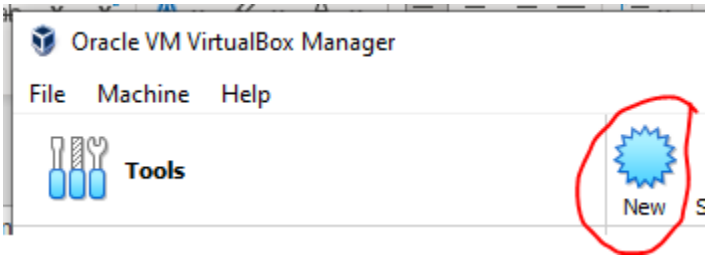
VPN Project

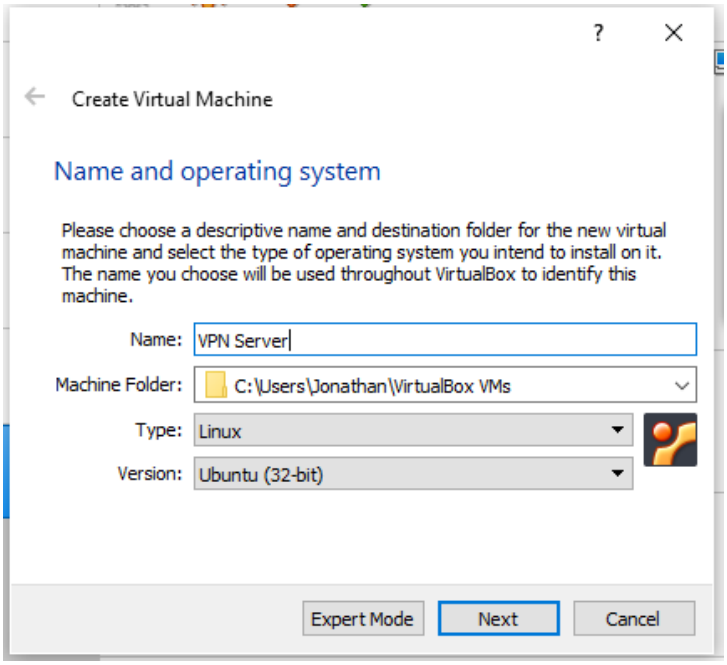
Part 1: Setting up the SEED virtual machines and the network:

Step 1: Go to link https://seedsecuritylabs.org/lab_env.html , then go to the "SEED Ubuntu16.04 VM (32-bit)" section and click on the zip file next to DigitalOcean to download the file (this will take some time to download). Once you are done downloading the zip file, extract it to a location you can remember.

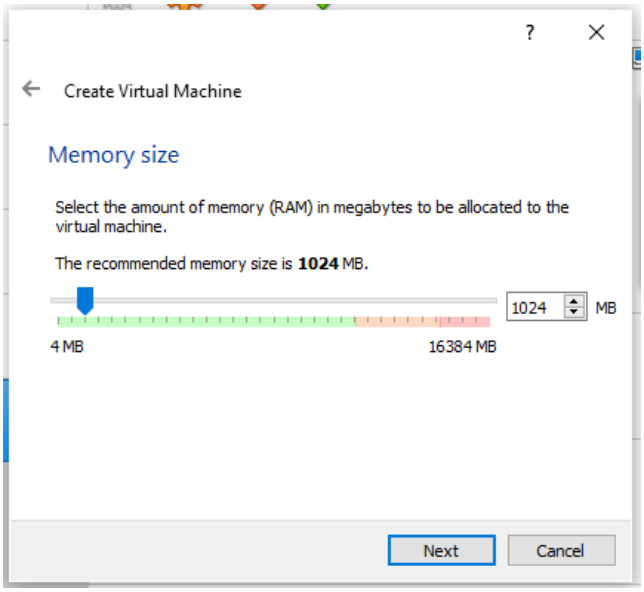
Step 2: Open VirtualBox Manager, and click New:



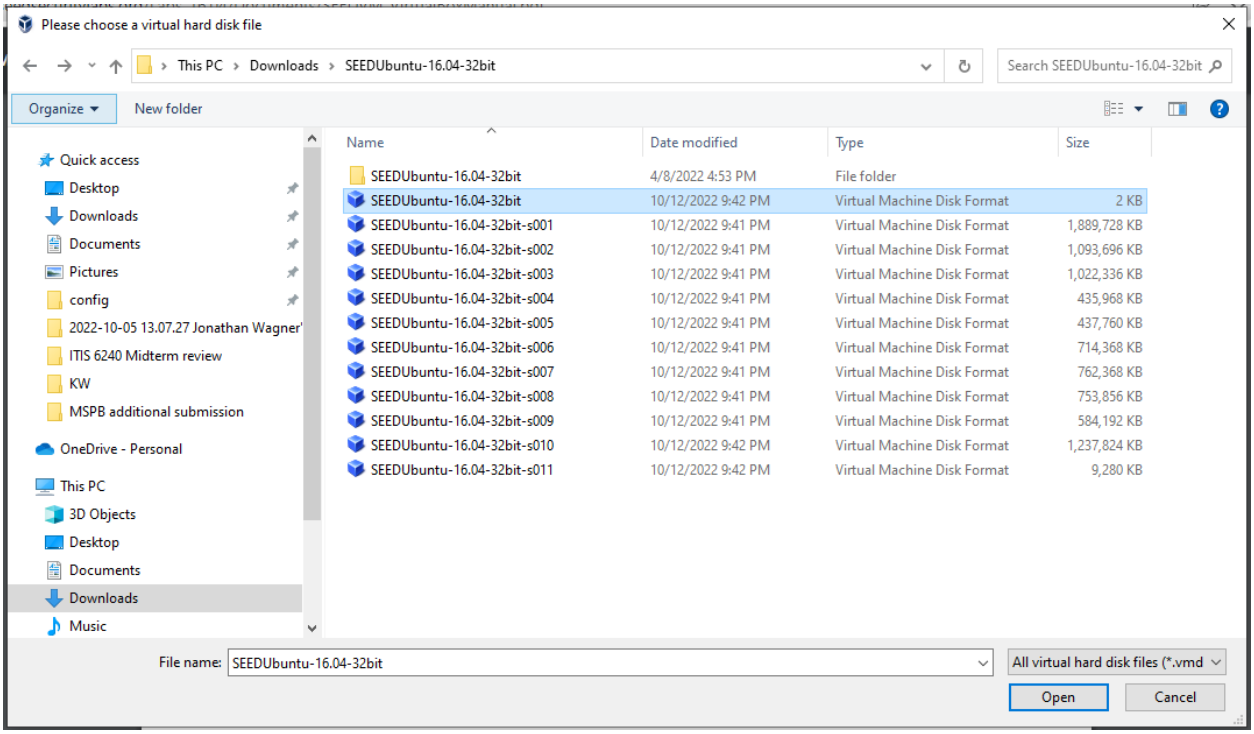
Step 3: Choose a name such as VPN Server, but choose for Type "Linux" and Version "Ubuntu (32-bit)". Then click Next:



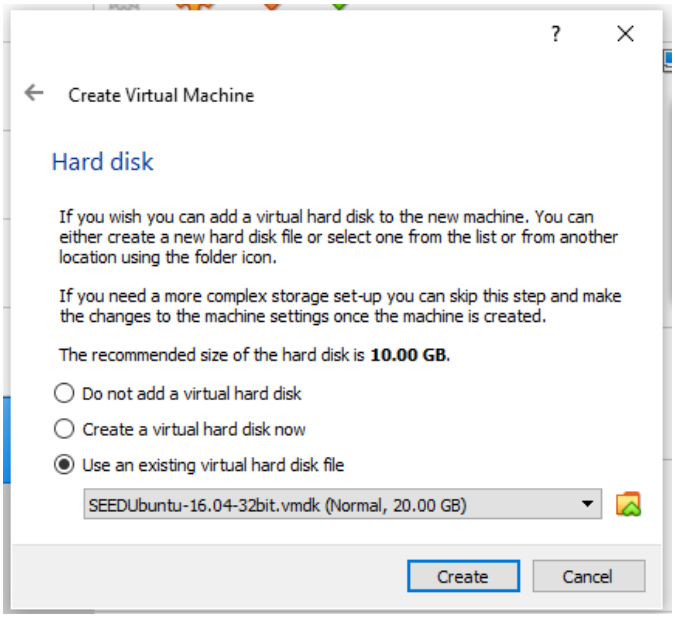
Step 4: Click Next:



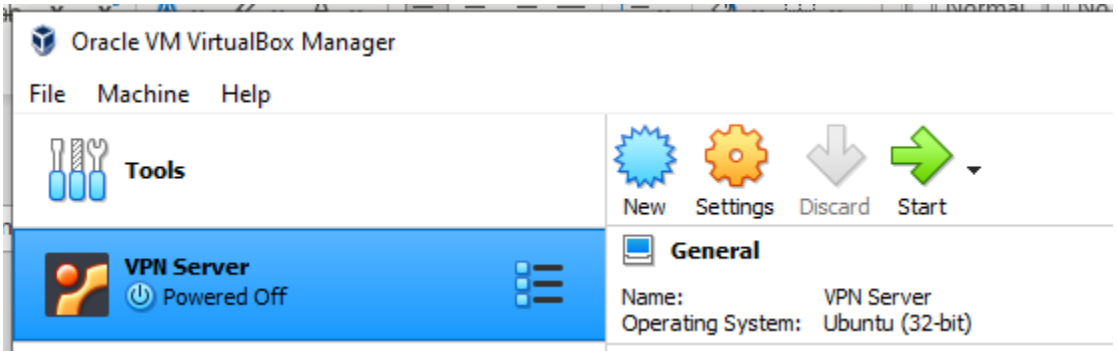
Step 5: Choose for your virtual hard disk file the second option shown below from your extracted folder, when you click the folder icon in the “Use an existing virtual hard disk file” option in Step 6:



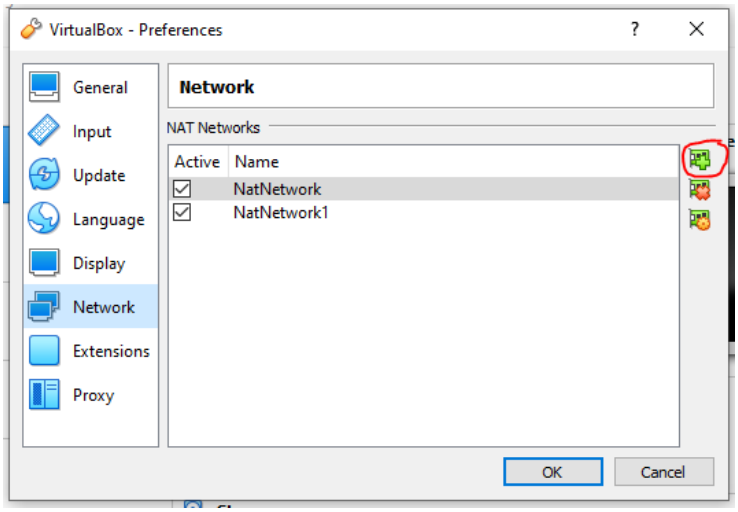
Step 6: Click Create:



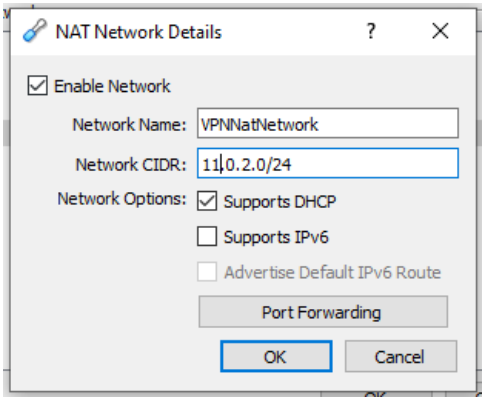
What should appear on your VirtualBox Manager:



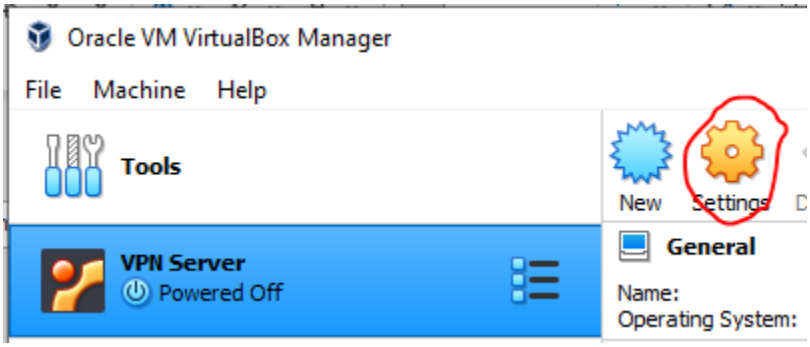
Step 7: Next go to File, then Preferences. Then click on the Network tab on the left. Then the green plus icon in the right (if you don't have a NAT Network):



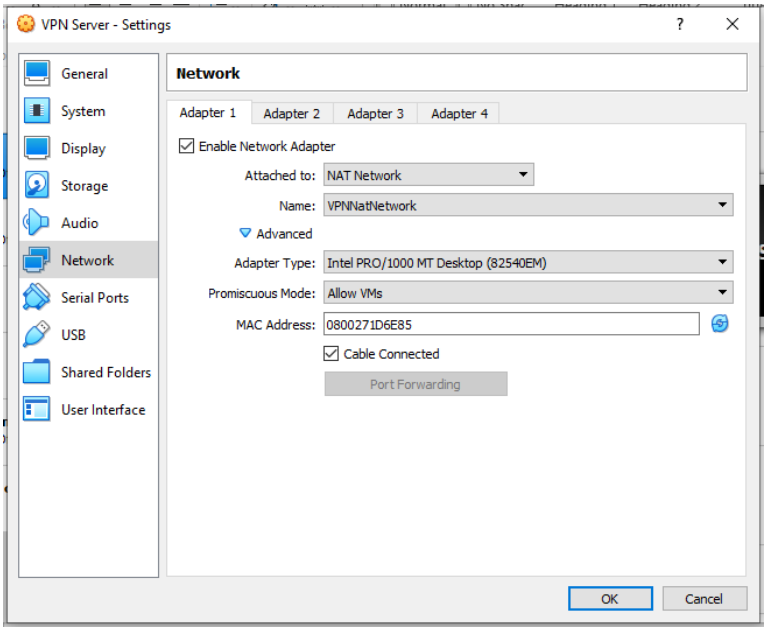
Step 8: Click on the new Nat Network shown, then choose a Network Name, then a Network CIDR. Then click OK, then OK again:



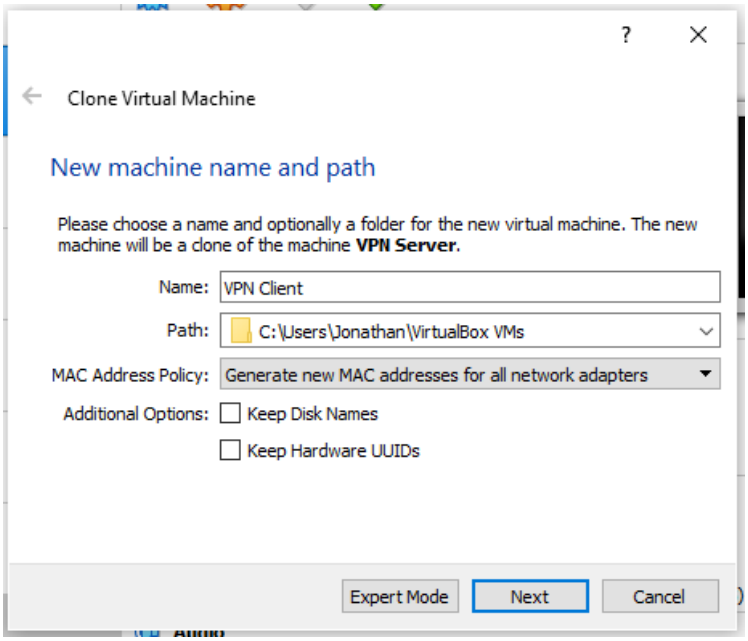
Step 9: Go to your Settings for the VPN Server:



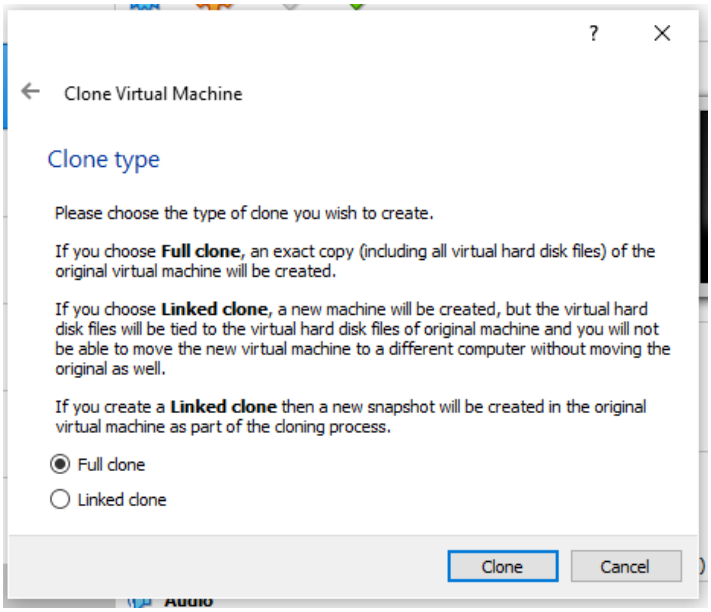
Step 10: Choose the Network tab, then for “Attached to:” choose “NAT Network”, then the name of your new NAT Network (mine is VPNNatNetwork). Then for Promiscuous Mode choose “Allow VMS”. Then Click OK:



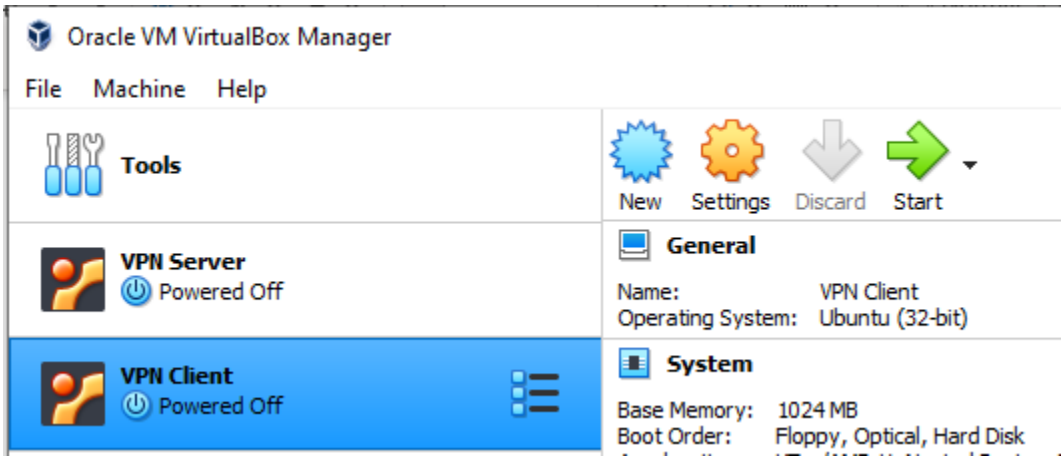
Step 11: Right click on your Server VM in VirtualBox Manager and choose Clone. Then choose new name for your cloned VM (below is name VPN Client), then choose “Generate new MAC addresses for all network adapters” for your MAC Address Policy. Then click Next:



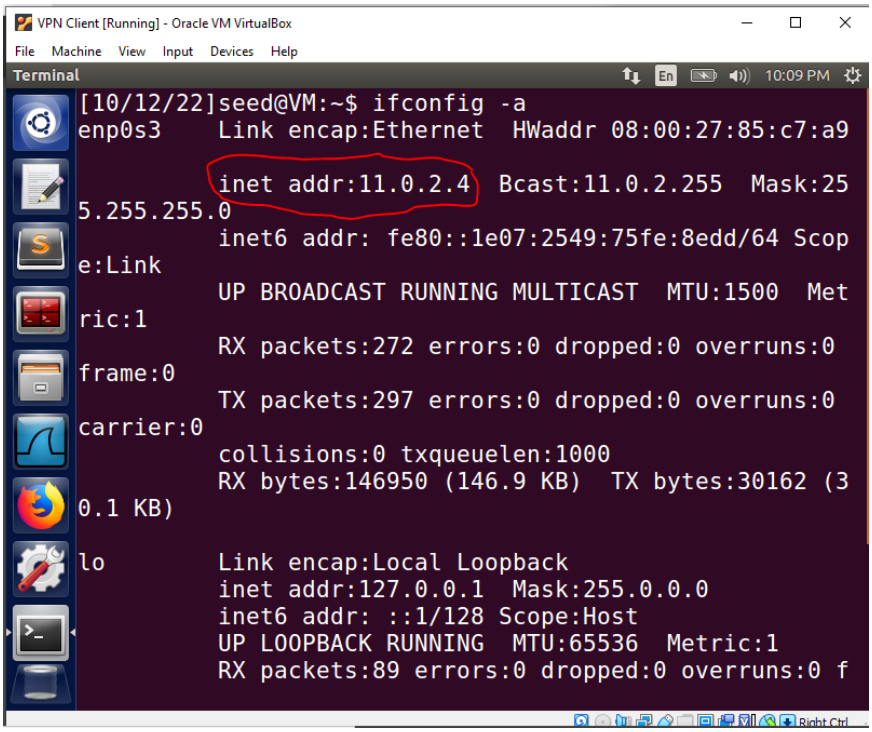
Step 12: Choose “Full clone”, then click Clone:



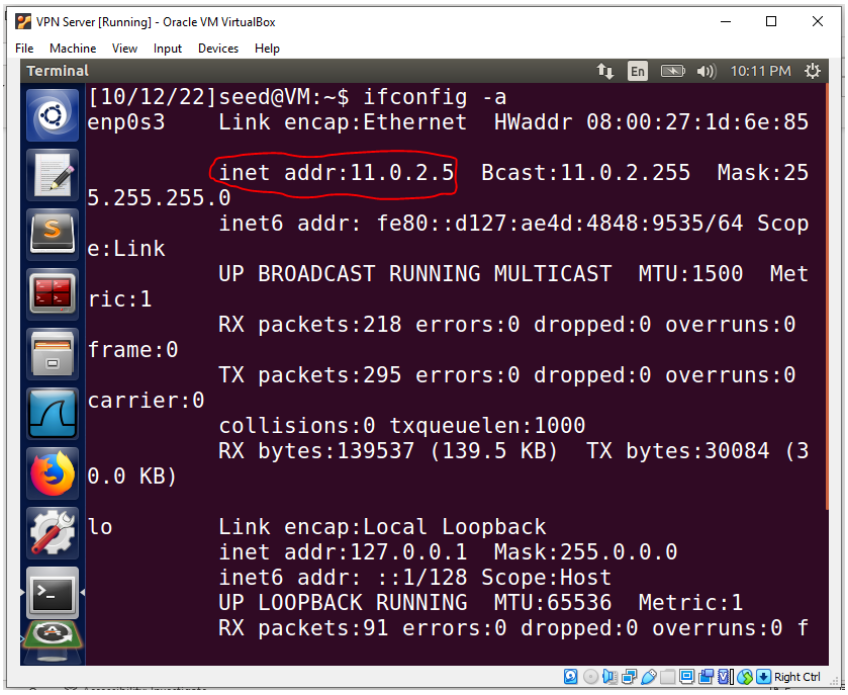
What should appear (you should have the same Network settings for your VPN Client as you do with the VPN Server):



Step 13: Start both your VPN Server and VPN Client virtual machines, and open Terminal for both. Then execute ifconfig -a to find the IP addresses for both machines:

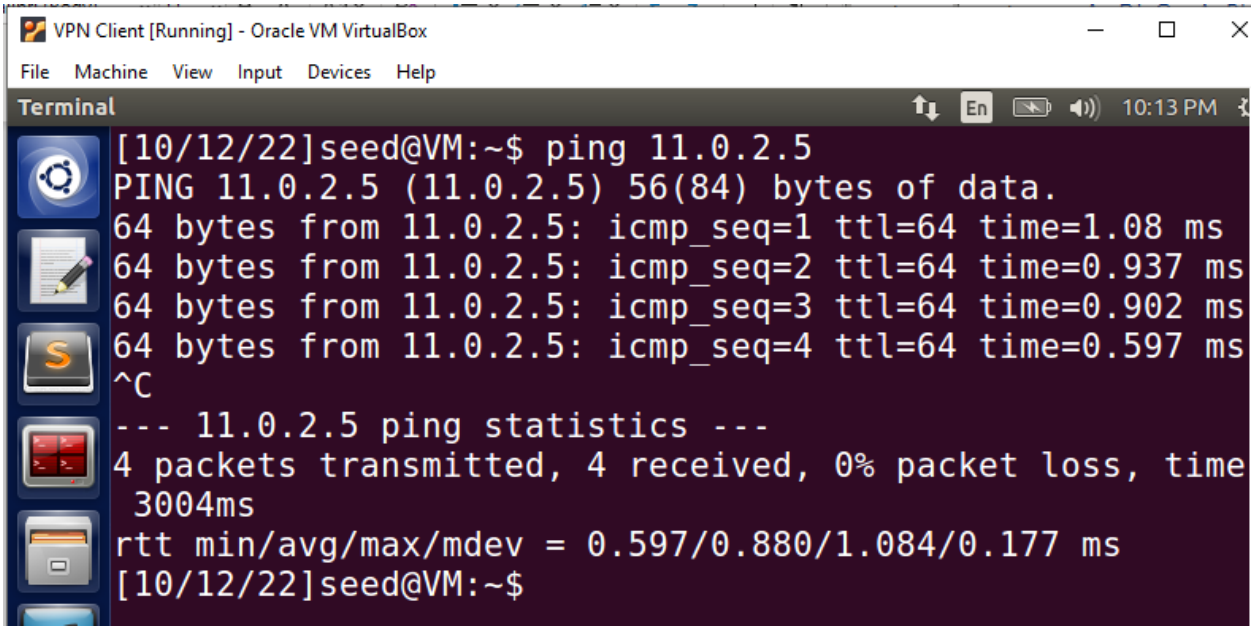


VPN Client's IP address is 11.0.2.4

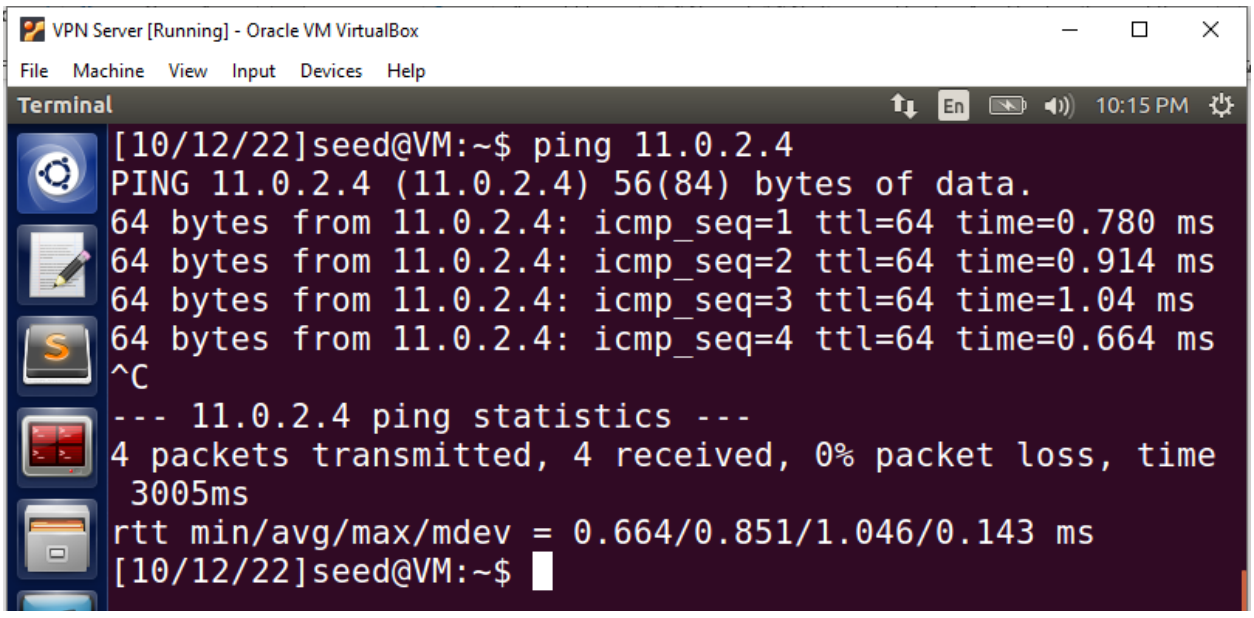


VPN Server's IP address is 11.0.2.5

Step 14: Ping from one VM to another, and vice versa **(take screenshots for your report)**:

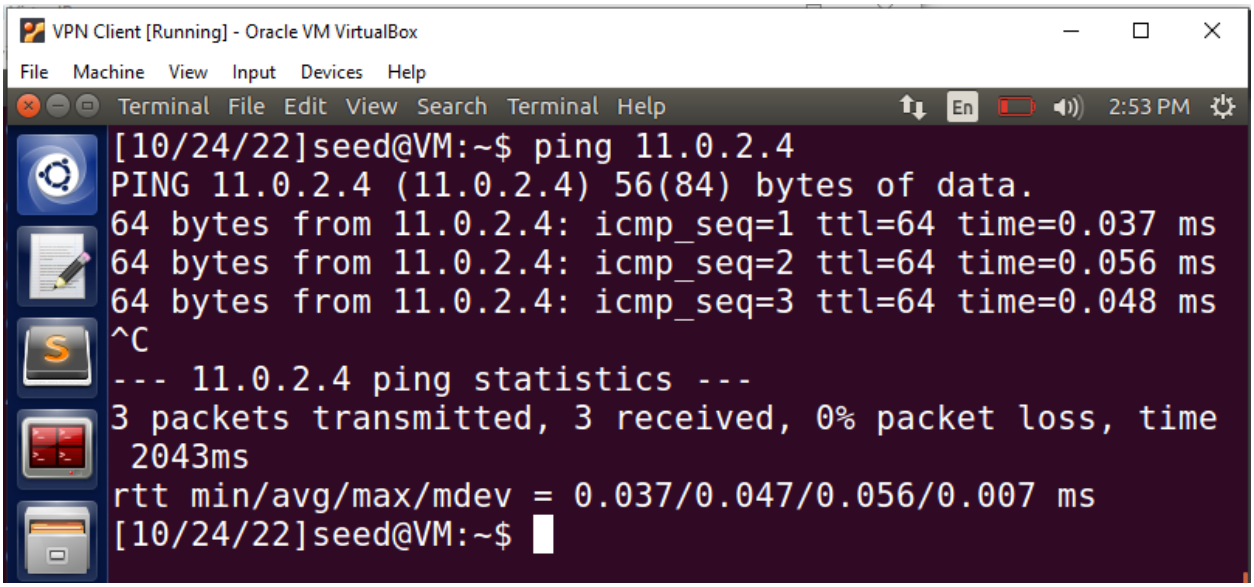


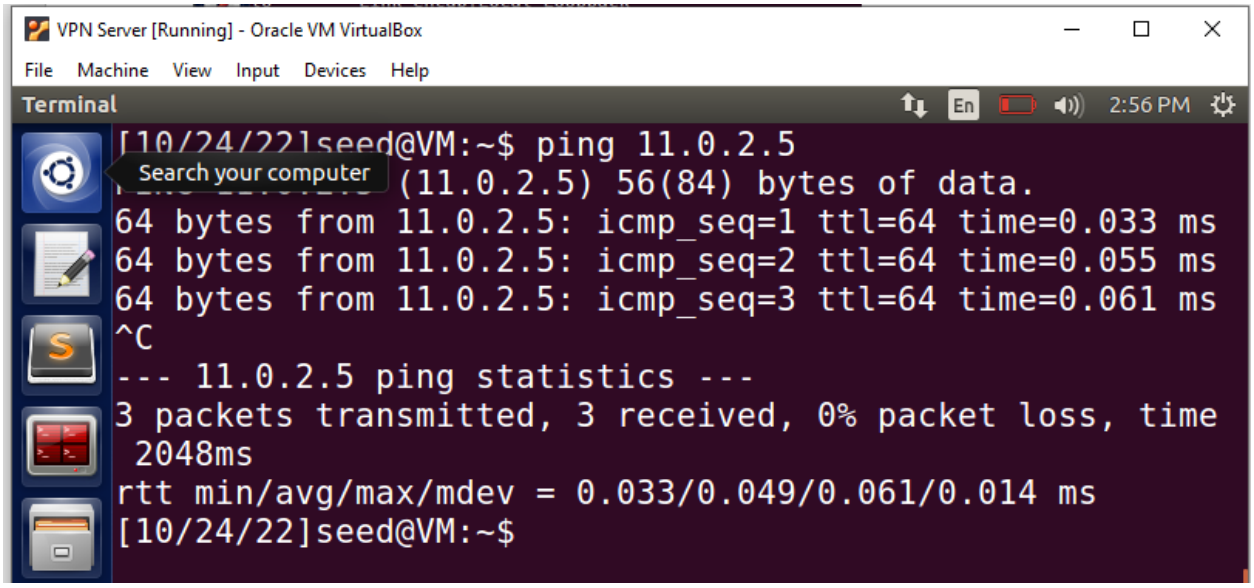
Pinged VPN Server from VPN Client



Pinged VPN Client from VPN Server

Step 15: Make sure that VPN Client VM is able to ping itself, and do the same for the VPN Server VM:

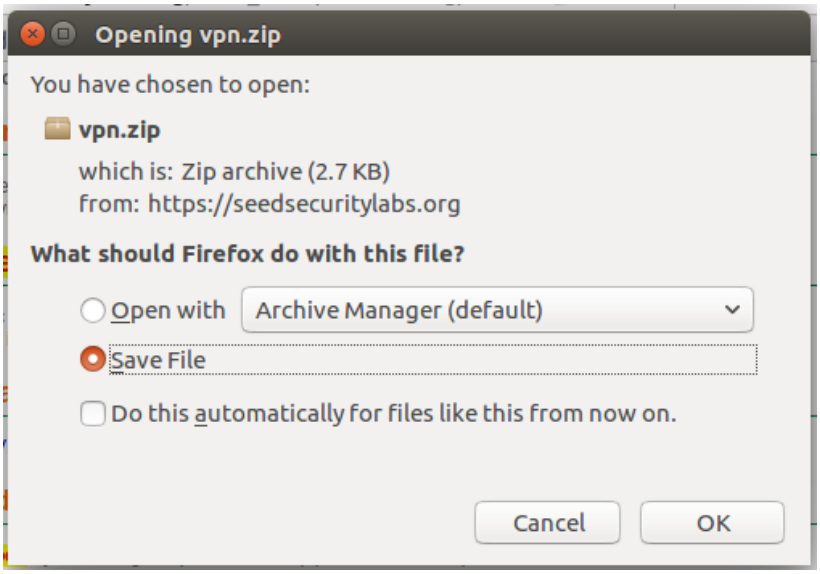




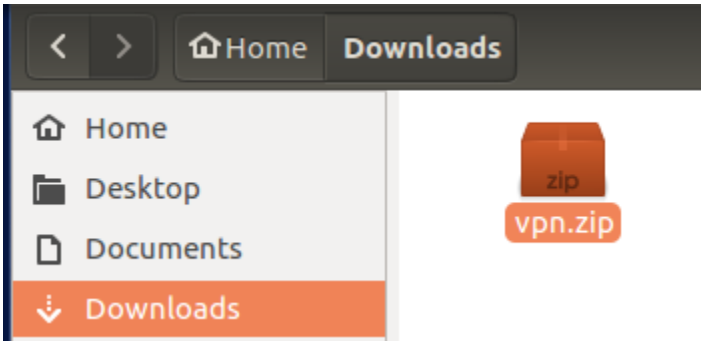
Part 2: Downloading and compiling vpnserver and vpnclient programs

Step 16: Visit https://seedsecuritylabs.org/Labs_16.04/Networking/Firewall_VPN/ on both of your virtual machines using Firefox. Then scroll down the page to click on “Sample VPN client and server programs (without encryption)” on each virtual machine:

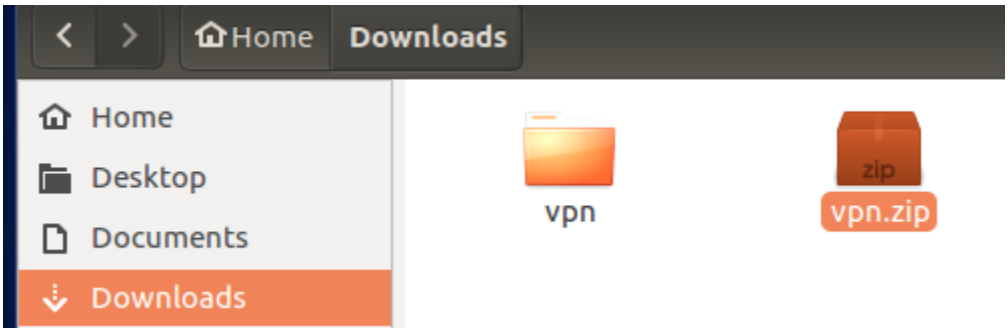
Step 17: Next click “Save File”, the OK:



You will see the following in Downloads

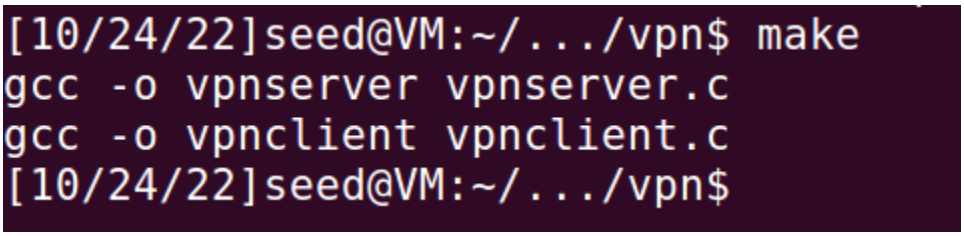


Step 18: Right click on the file and click “Extract Here”, then you will see the following:

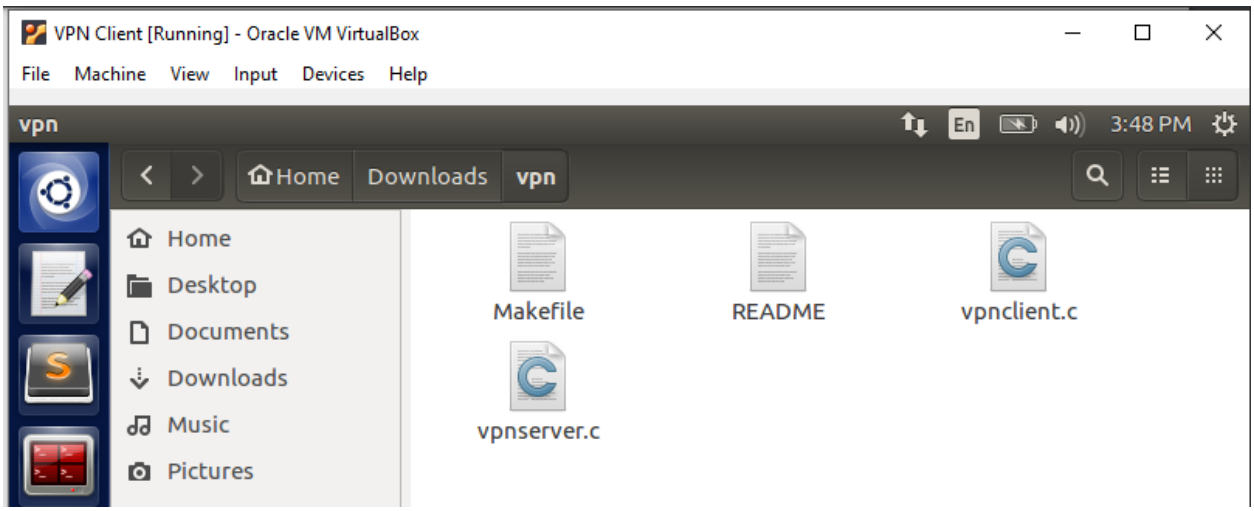


Note: Please make sure that you do Steps 16 – 18 for both virtual machines

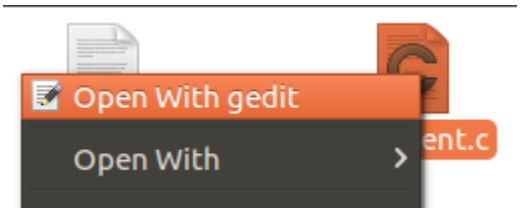
Step 19: On you VPN server virtual machine, open the Terminal. Then go to your extracted “vpn” folder (see above), then execute “make”:



Step 20: On your VPN Client virtual machine, open your extracted vpn folder, and right click on vpnclient.c and click on Gedit:



Click “Open with gedit”:



Step 21: Toward the top of the file shown be a line of code that states, #define SERVER_IP “127.0.0.1”. Replace this IP address with the VPN Server’s IP address

Before:

```
#define SERVER_IP "127.0.0.1"
```

After:

```
#define SERVER_IP "11.0.2.5"
```

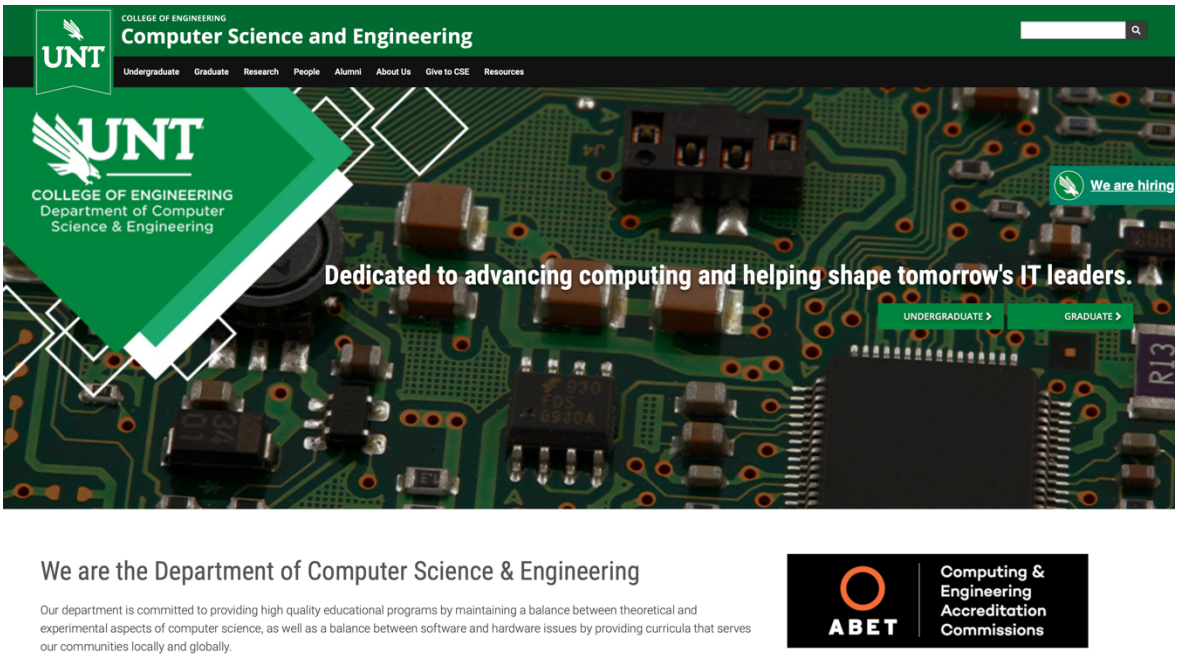
Step 22: Then click Save in the top right:

Step 23: Open Terminal on your VPN Client virtual machine, then go to the location of your extracted vpn folder and execute “make”:

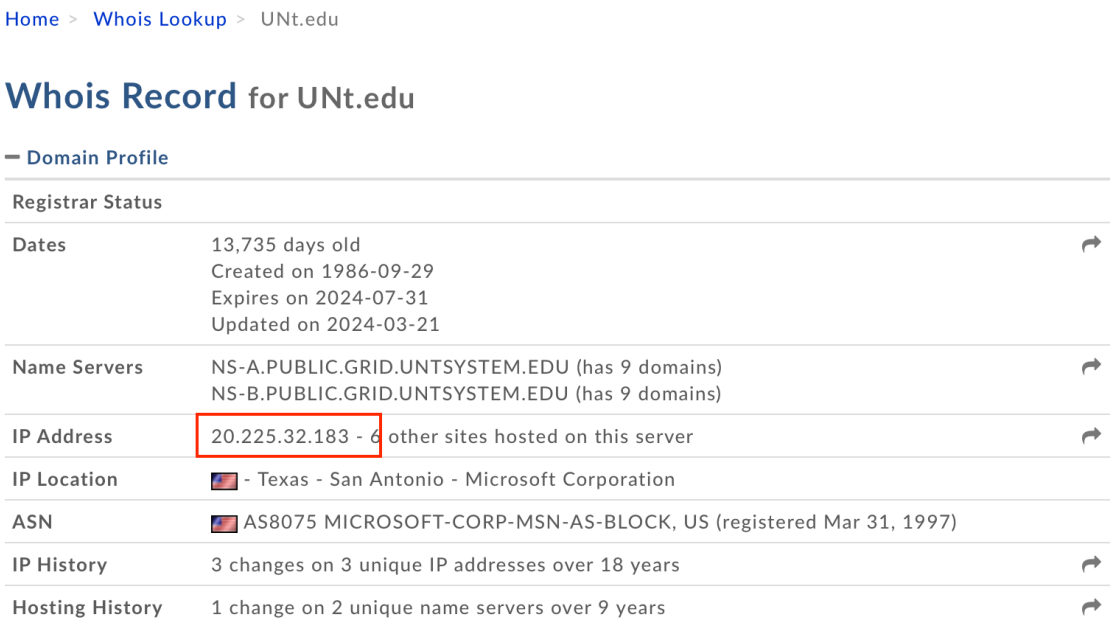
```
[10/24/22]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[10/24/22]seed@VM:~/.../vpn$
```

Part 3: Setting up the firewall

Step 24: On your VPN Client virtual machine, visit any webpage using your Firefox browser (this will be the webpage you will be blocking your client from accessing with a firewall). For example, I am accessing UNT CSE page:



Step 25: Find the IP address of that link. For example, I am using whois.domaintools.com where I am able to find the IP address for https://computerscience.engineering.unt.edu (see yellow highlighted below):



Step 26: Then execute on your VPN Client “sudo ufw enable”:

```
[10/24/22]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[10/24/22]seed@VM:~$
```

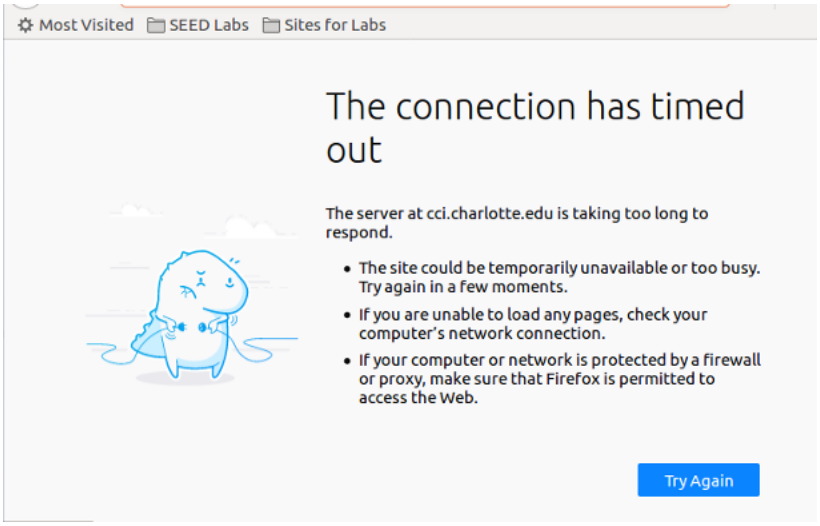
Step 27: Use your enp0s3 interface (recall Step 13) and the IP address of the webpage you visited (in my case, its 20.225.32.183) to execute the following command:1

```
[10/24/22]seed@VM:~$ sudo ufw deny out on enp0s3 to 20
.225.32.183
Rule added
[10/24/22]seed@VM:~$
```

Step 28: Then execute “sudo ufw status”:

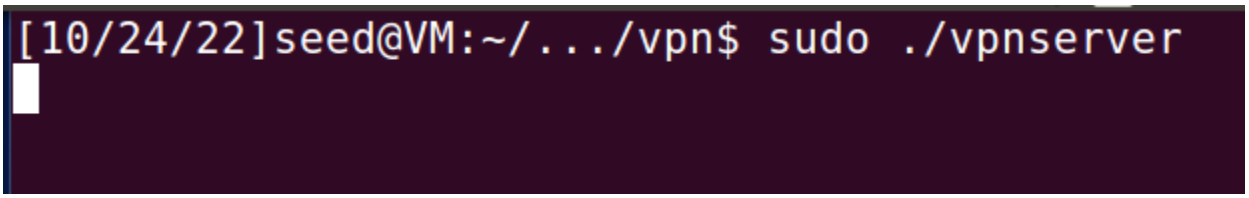
```
[10/24/22]seed@VM:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
142.251.15.0/24            DENY OUT    Anywhere on enp0
s3
20.225.32.183              DENY OUT    Anywhere on enp0
s3

[10/24/22]seed@VM:~$
```

Step 29: When I try to access cci.charlotte.edu again on my VPN Client, I get the following:

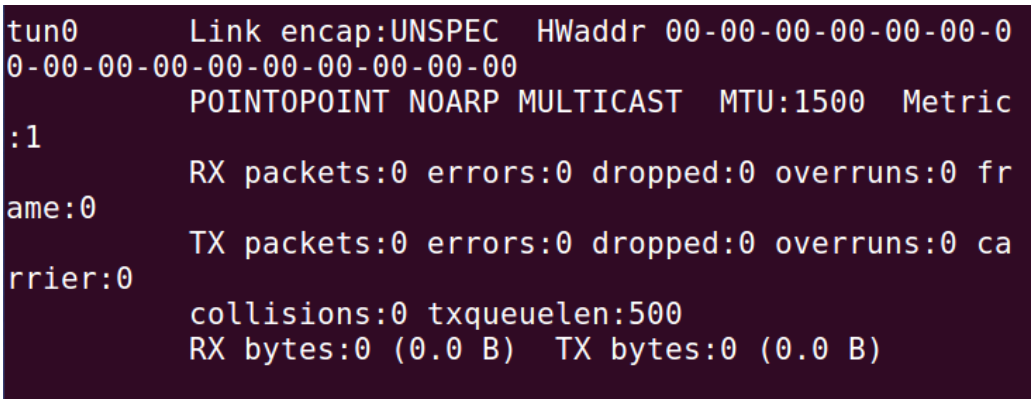


Part 4: Setting up VPN tunnel

Step 30: On your VPN server virtual machine, go to your extracted vpn folder on Terminal, and execute the following:



Step 31: In another terminal, execute “ifconfig -a”. You should then see a tun0 interface:



Step 32: Next execute “sudo ifconfig tun0 192.168.53.1/24 up”:

```
[10/24/22]seed@VM:~$ sudo ifconfig tun0 192.168.53.1/24
 up
[10/24/22]seed@VM:~$
```

Step 33: When you execute “ifconfig -a”, you will then see the following for tun0 where tun0 now has IP address of 192.168.53.1:

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-0
0-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.1  P-t-P:192.168.53.1  M
ask:255.255.255.0
          inet6 addr: fe80::c9d:d324:fe29:c2bb/64 Scope
:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1
500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 fr
ame:0
          TX packets:0 errors:0 dropped:0 overruns:0 ca
rrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

[10/24/22]seed@VM:~$
```

Step 34: Next execute “sudo sysctl net.ipv4.ip_forward=1” in order for your VPN server to act as a gateway:

```
[10/24/22]seed@VM:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[10/24/22]seed@VM:~$
```

Step 35: On your VPN Client virtual machine, navigate to your extracted vpn folder and execute “sudo ./vpnclient” using the VPN Server’s IP address (recall Step 13).

```
[10/24/22]seed@VM:~/.../vpn$ sudo ./vpnclient 11.0.2.5
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```

What you will now see on the VPN server:

```
[10/24/22]seed@VM:~/.../vpn$ sudo ./vpnserver
Connected with the client: Hello
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
```

Step 36: Open another Terminal window in your VPN Client, and execute “sudo ifconfig tun0 192.168.53.5/24 up”:

```
[10/24/22]seed@VM:~$ sudo ifconfig tun0 192.168.53.5/24
 up
[10/24/22]seed@VM:~$
```

What now appears on VPN Server:

```
[10/24/22]seed@VM:~/.../vpn$ sudo ./vpnserver
Connected with the client: Hello
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```

Part 5: Add routing to VPN tunnel

Step 37: On VPN server virtual machine, execute “sudo route add -net 192.168.53.0/24 tun0” (192.168.53.0 is my VPN network address, but you may have something different):

```
[10/24/22]seed@VM:~$ sudo route add -net 192.168.53.0/2
4 tun0
[10/24/22]seed@VM:~$
```

Step 38: On VPN client virtual machine, execute the same command of “sudo route add -net 192.168.53.0/24 tun0”:

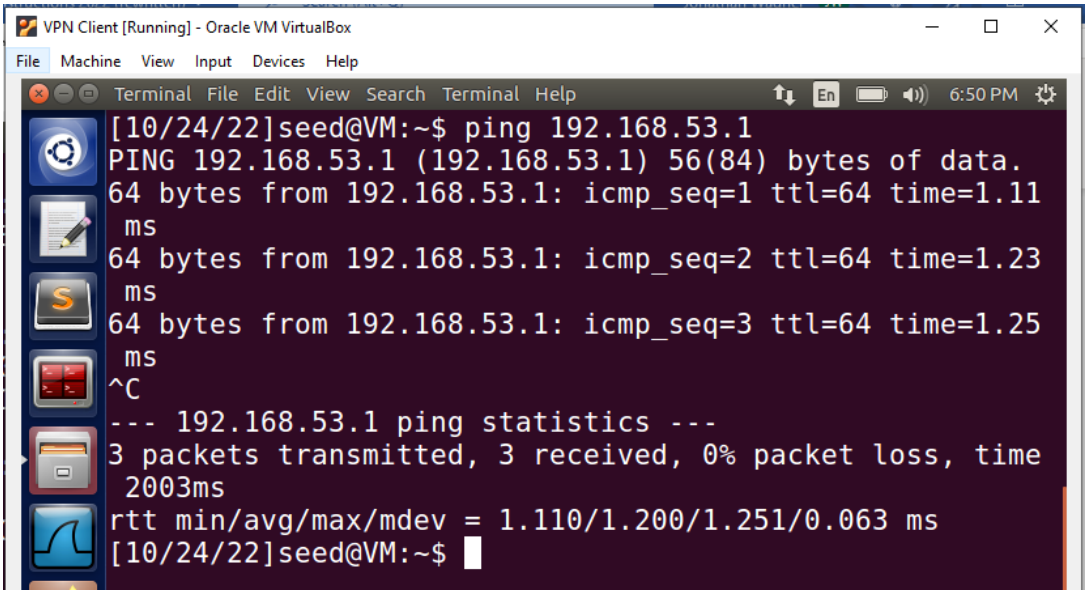
```
[10/24/22]seed@VM:~$ sudo route add -net 192.168.53.0/2
4 tun0
[10/24/22]seed@VM:~$
```

Step 39: Next execute on your VPN client the same command, but this time you are using the IP address of the site you blocked (I have mine listed for range below, so instead of listing the IP address I blocked of 20.225.32.183, I list 20.225.32.0/24. Otherwise, the below command won’t work):

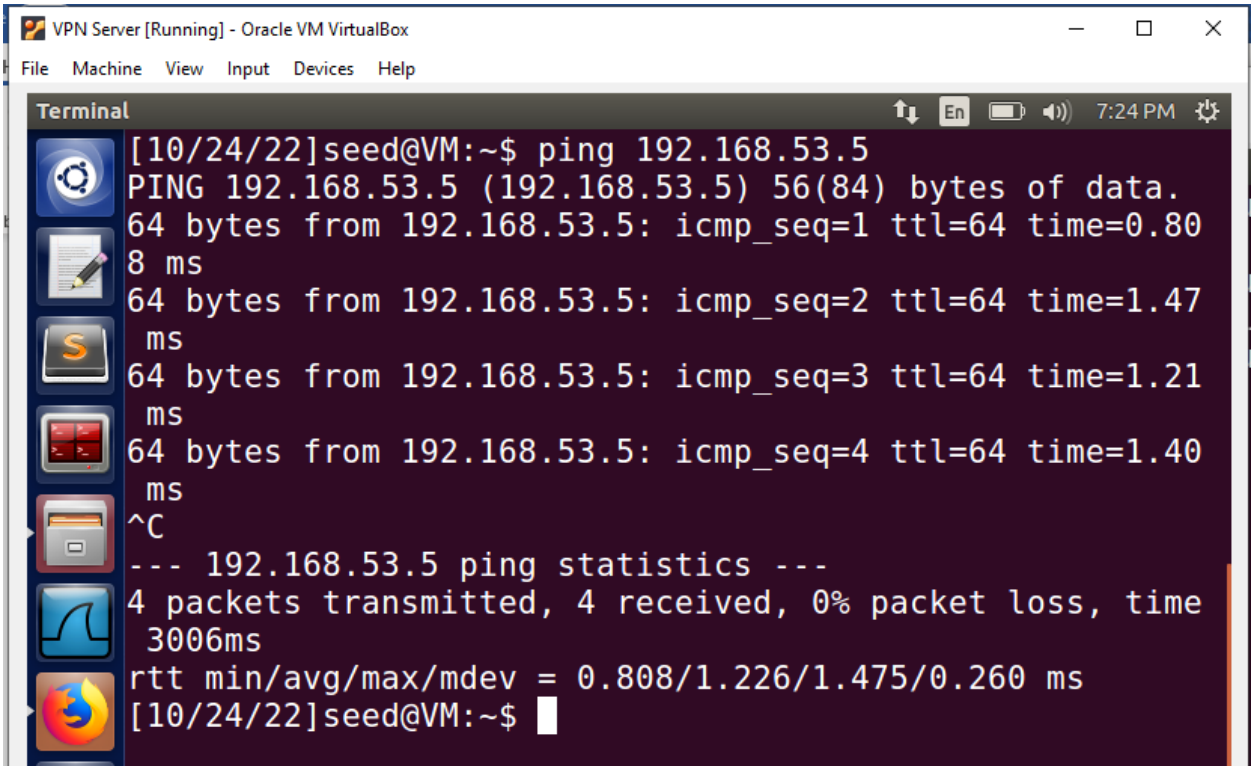
```
[10/24/22]seed@VM:~$ sudo route add -net 20.225.32.0/24
 tun0
[10/24/22]seed@VM:~$
```

Step 40: You should now be able to have the VPN Client and VPN Server ping each other's tun0 interfaces **(take screenshots for your report)**:

Note: If you are having trouble pinging the VPN Client, try disabling the VPN Client's firewall with "sudo ufw disable" and try pinging the VPN Client from the VPN Server. Then reenable VPN Client's firewall with "sudo ufw enable", then try pinging the VPN Client from the VPN Server.



Pinging VPN Server's tun0 interface from VPN Client



Pinging VPN Client's tun0 interface from VPN Server

Part 6: Enabling NAT on your VPN Server:

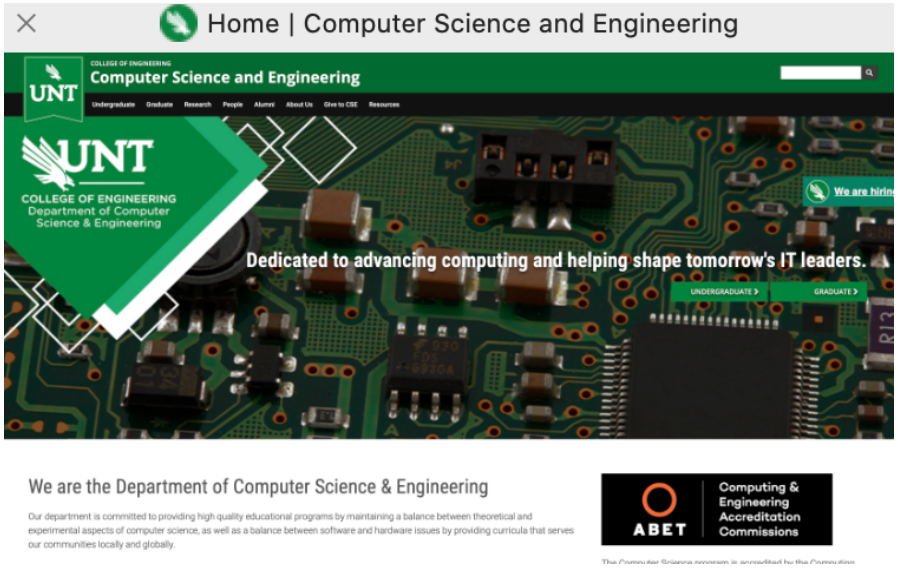
Step 41: Go to your VPN server virtual machine. For cleaning iptables rules, execute the following two commands:

```
[10/24/22]seed@VM:~$ sudo iptables -F
[10/24/22]seed@VM:~$ sudo iptables -t nat -F
[10/24/22]seed@VM:~$
```

Step 42: Then execute "sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o enp0s3" (for this command, use the interface that is linked to your VPN Server's IP address from Step 13, in my case its "enp0s3"):

```
[10/24/22]seed@VM:~$ sudo iptables -t nat -A POSTROUTIN
G -j MASQUERADE -o enp0s3
[10/24/22]seed@VM:~$
```

Step 43: Now try accessing the blocked link on your VPN Client (I was able to reach my blocked link below):



What you need to turn in:

A) Take screenshots of the following:
   a. The VPN Server pinging the VPN Client's enp0s3 interface (recall Step 14)
   b. The VPN Client pinging the VPN Server's enp0s3 interface (recall Step 14)
   c. The VPN Server pinging the VPN Client's tun0 interface (recall Step 40)
   d. The VPN Client pinging the VPN Server's tun0 interface (recall Step 40)

B) Follow the below steps, then answer the question that follows:
   a. Open Wireshark on both the VPN Server and VPN Client.
   b. On the VPN Server's Wireshark, listen to the tun0 interface
   c. On the VPN Client's Wireshark, listen to the enps0s3 interface
   d. Have the VPN Server ping the VPN Client's tun0 interface's IP address.
   e. Take screenshots of what you see on Wireshark on both the VPN Server and VPN Client

   Based on what you see on Wireshark on both virtual machines, how does VPN tunneling hide an IP packet within another IP packet? Please explain using the screenshots you took.

C) Follow the below steps, then answer the question that follows:
   a. Open Wireshark on your VPN Client
   b. Listen to the enps0s3 interface
   c. Visit the blocked webpage
   d. Take screenshot(s) of what you see on Wireshark
   e. Next listen to the tun0 interface
   f. Visit the blocked webpage again
   g. Take screenshot(s) of what you see on Wireshark

   How is the VPN Client able to access the webpage that's blocked by its firewall? Please explain using the screenshots you took.

Each screenshot should directly show your names as the account running the OS. Otherwise you cannot get the bonus points. Deadline for bonus project is May 10 11:59 pm. **Deadline is firm. Late submission will not be accepted.**

References:

https://seedsecuritylabs.org/Labs_16.04/Documents/SEEDVM_VirtualBoxManual.pdf

https://seedsecuritylabs.org/Labs_16.04/PDF/Firewall_VPN.pdf