NETWORK SECURITY

HOMEWORK-5

BYPASSING FIREWALLS USING VPN

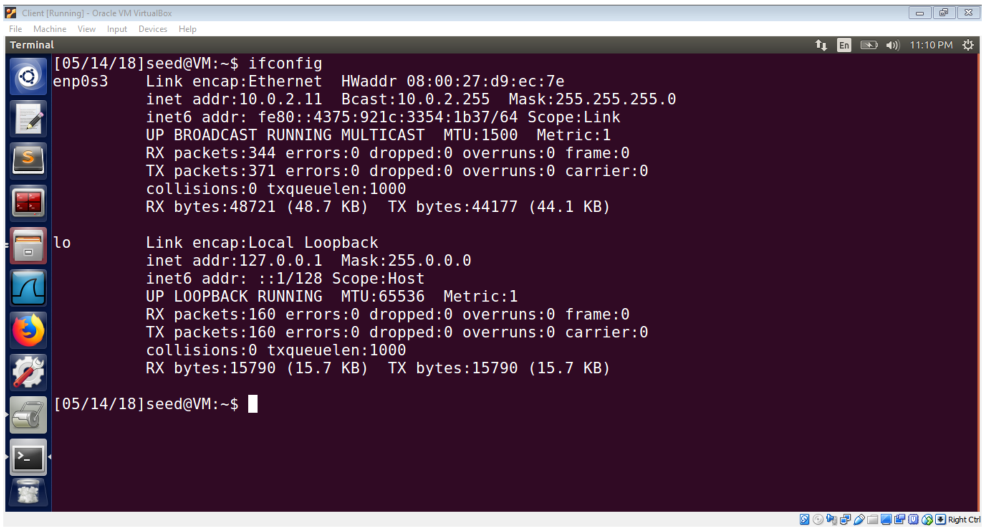
ANIMESH JAIN

UIN – 669653208

ajain65@uic.edu

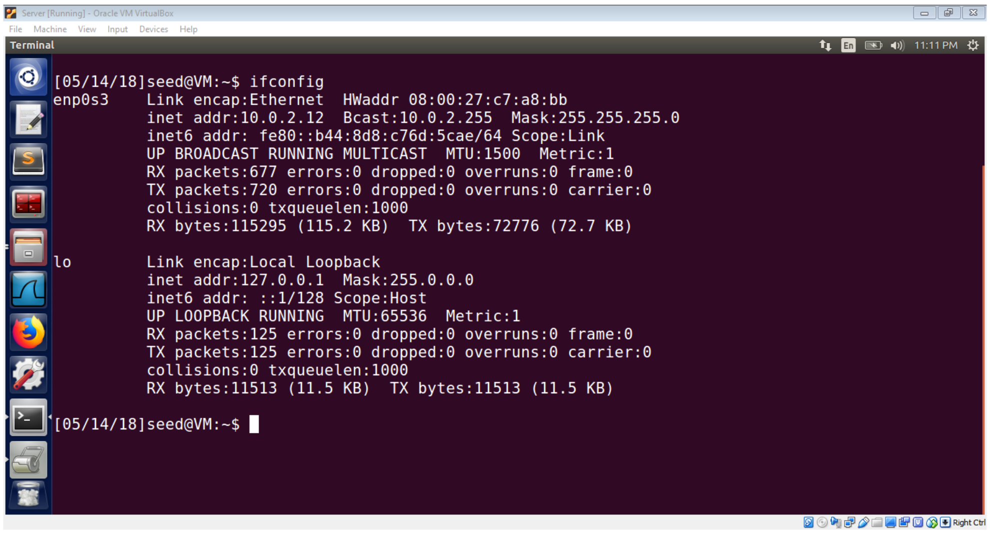
2.1 TASK 1: VM SETUP

VM1 = CLIENT MACHINE (inside the firewall)


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
En
11:10 PM
Right Ctrl

```
[05/14/18]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:d9:ec:7e
          inet addr:10.0.2.11  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::4375:921c:3354:1b37/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:344 errors:0 dropped:0 overruns:0 frame:0
          TX packets:371 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48721 (48.7 KB)  TX bytes:44177 (44.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15790 (15.7 KB)  TX bytes:15790 (15.7 KB)

[05/14/18]seed@VM:~$
```

VM2 = SERVER MACHINE (outside the firewall)

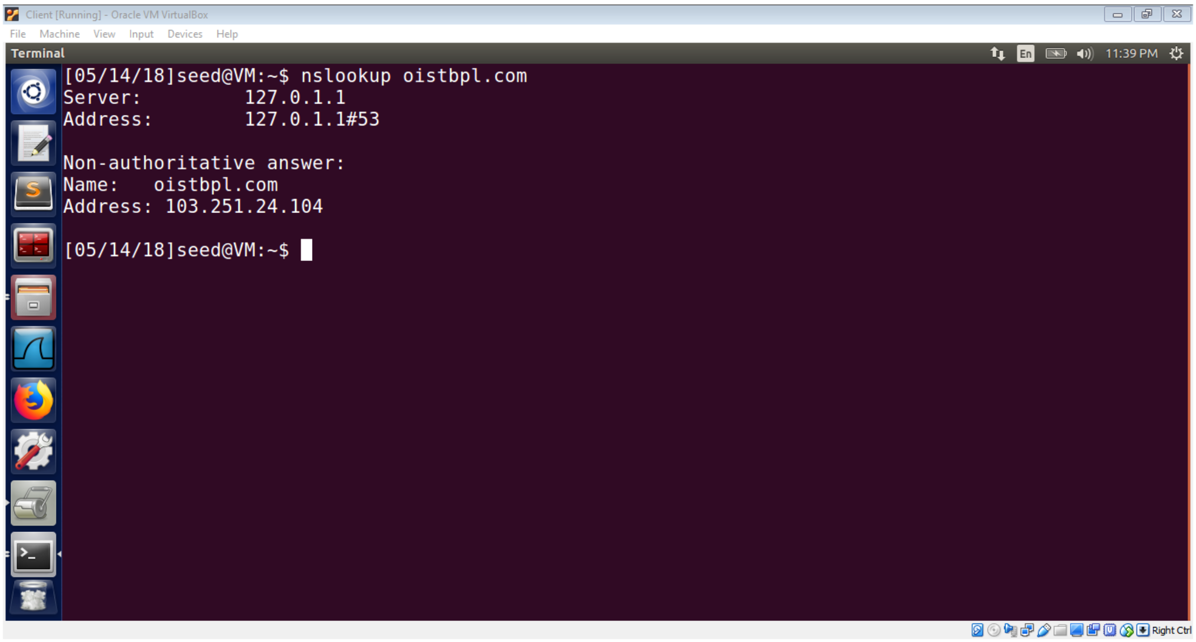

Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
En
11:11 PM
Right Ctrl

```
[05/14/18]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:c7:a8:bb
          inet addr:10.0.2.12  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::b44:8d8:c76d:5cae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:677 errors:0 dropped:0 overruns:0 frame:0
          TX packets:720 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:115295 (115.2 KB)  TX bytes:72776 (72.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:125 errors:0 dropped:0 overruns:0 frame:0
          TX packets:125 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11513 (11.5 KB)  TX bytes:11513 (11.5 KB)

[05/14/18]seed@VM:~$
```

2.2 TASK 2: SET UP FIREWALL

In this task we will setup a firewall on the “Client” machine to block the access to a website. The target website in this lab is **“oistbpl.com”**, which is the webpage of “ORIENTAL INSTITUTE OF SCIENCE AND TECHNOLOGY, Bhopal, India. In the below screenshot we have given a command to look for the IP address of our target website. Command that we will use for this is

- nslookup oistbpl.com



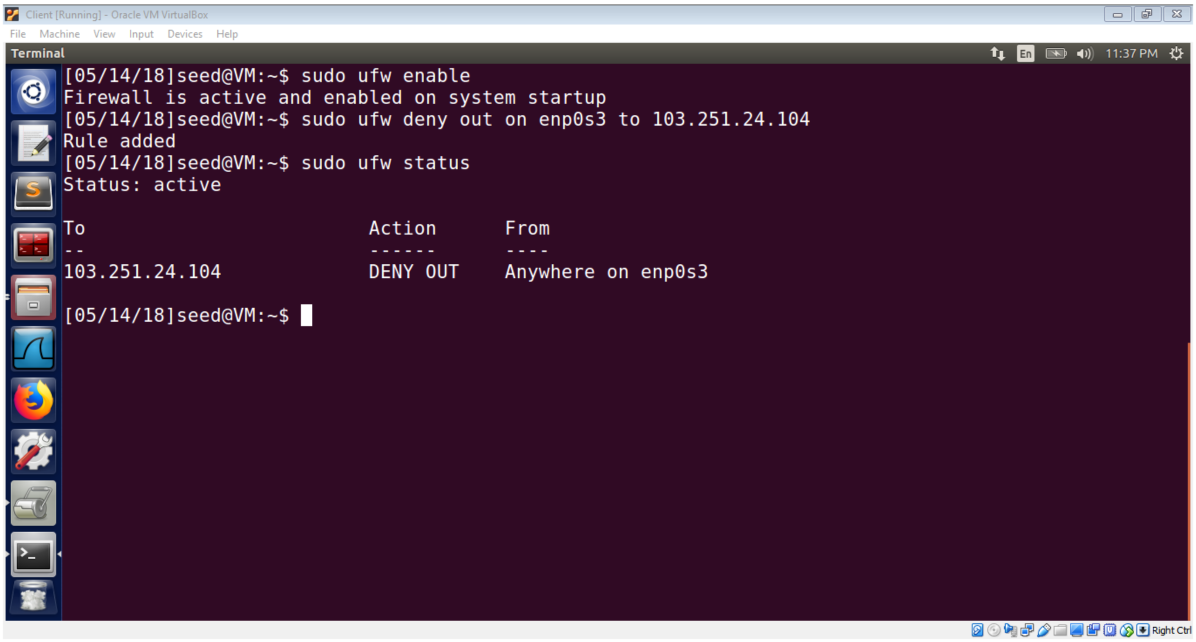
Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:39 PM
Right Ctrl

```
[05/14/18]seed@VM:~$ nslookup oistbpl.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
Name:   oistbpl.com
Address: 103.251.24.104

[05/14/18]seed@VM:~$ 
```

Now for setting up the firewall rules we will use the **ufw** program. The command that we will use to block “oistbpl.com” is

- sudo ufw deny out on enp0s3 to 103.251.24.104

All the commands that we use to setup firewall are mentioned in the below screenshot.

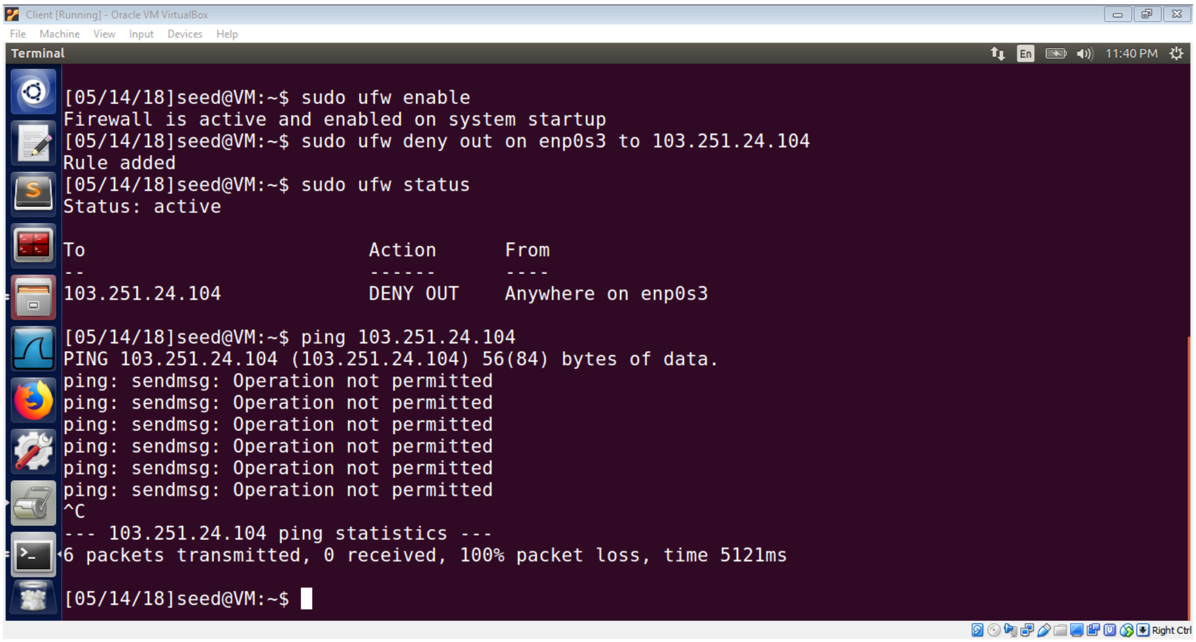


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:37 PM
Right Ctrl

```
[05/14/18]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[05/14/18]seed@VM:~$ sudo ufw deny out on enp0s3 to 103.251.24.104
Rule added
[05/14/18]seed@VM:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
103.251.24.104             DENY OUT    Anywhere on enp0s3

[05/14/18]seed@VM:~$ 
```

Now to check whether the firewall is working properly or not we will make use of ping command.

- ping 103.251.24.104



Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:40 PM
Right Ctrl

```
[05/14/18]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[05/14/18]seed@VM:~$ sudo ufw deny out on enp0s3 to 103.251.24.104
Rule added
[05/14/18]seed@VM:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
103.251.24.104             DENY OUT    Anywhere on enp0s3

[05/14/18]seed@VM:~$ ping 103.251.24.104
PING 103.251.24.104 (103.251.24.104) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 103.251.24.104 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5121ms

[05/14/18]seed@VM:~$ 
```

On pinging the “oistbpl.com” website we are getting the following response

- **ping: sendmsg: Operation not permitted** (in the above screenshot)

Also, few packets are transmitted but none of them were received. So, this confirms that our **firewall setup was successful**.

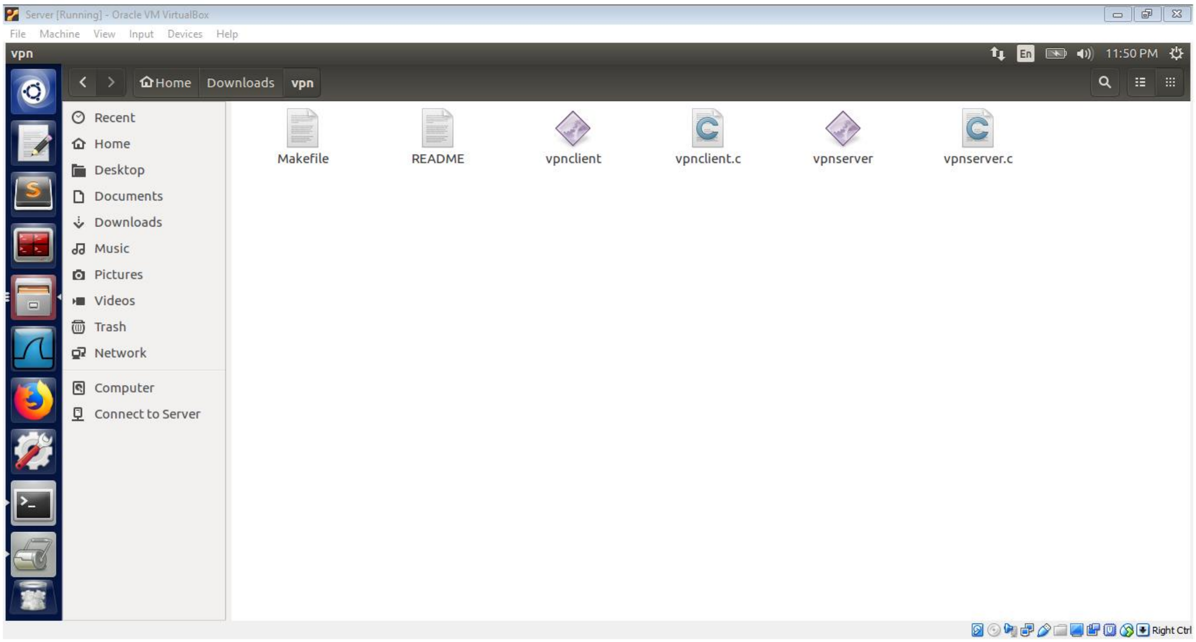
2.3 TASK 3: BYPASSING FIREWALL USING VPN

Earlier we saw that we are no longer able to access “oistbpl.com” because we have blocked it using firewall. Now if we want to access the blocked site then we must first establish a VPN tunnel through which the packets from “Client” machine will reach the “Server” machine. Now from the “Server” machine the packet will be directed to the destination i.e. the target website. The response packets also follow the similar process.

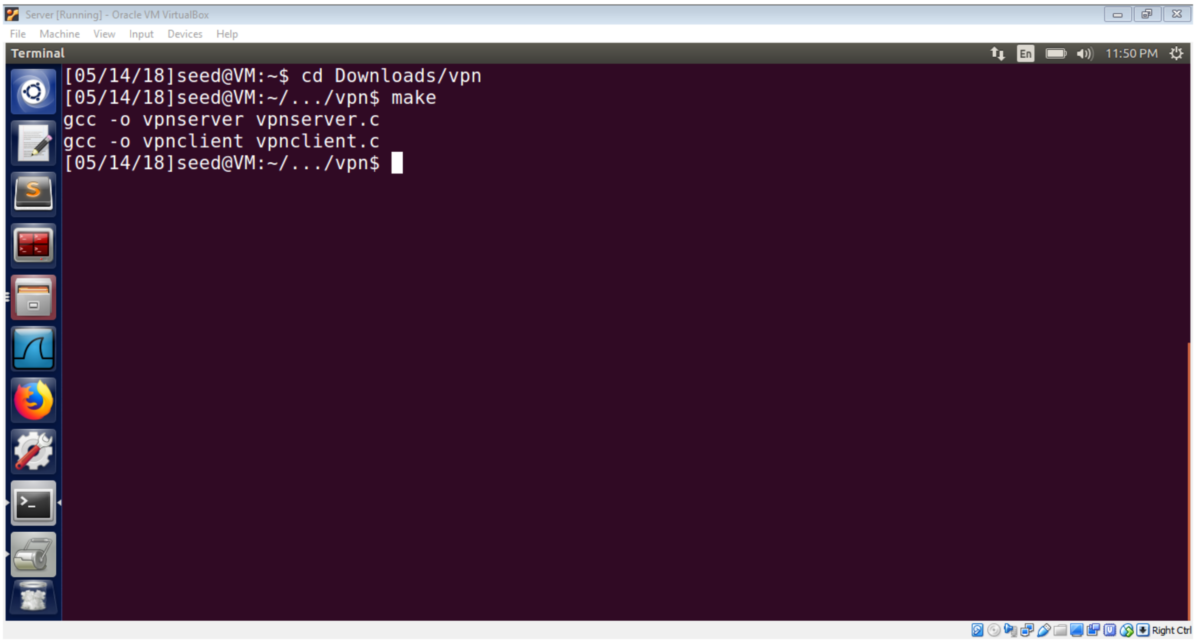
Now we will run the VPN server and VPN client program, both of which can be downloaded from the SEED labs website. **After running the server and client program a tunnel will be established between the “Client” and the “server” machine.**

Step 1: Run VPN server

After downloading the VPN server and client program from SEED labs website, we will extract all the files from the ZIP folder.

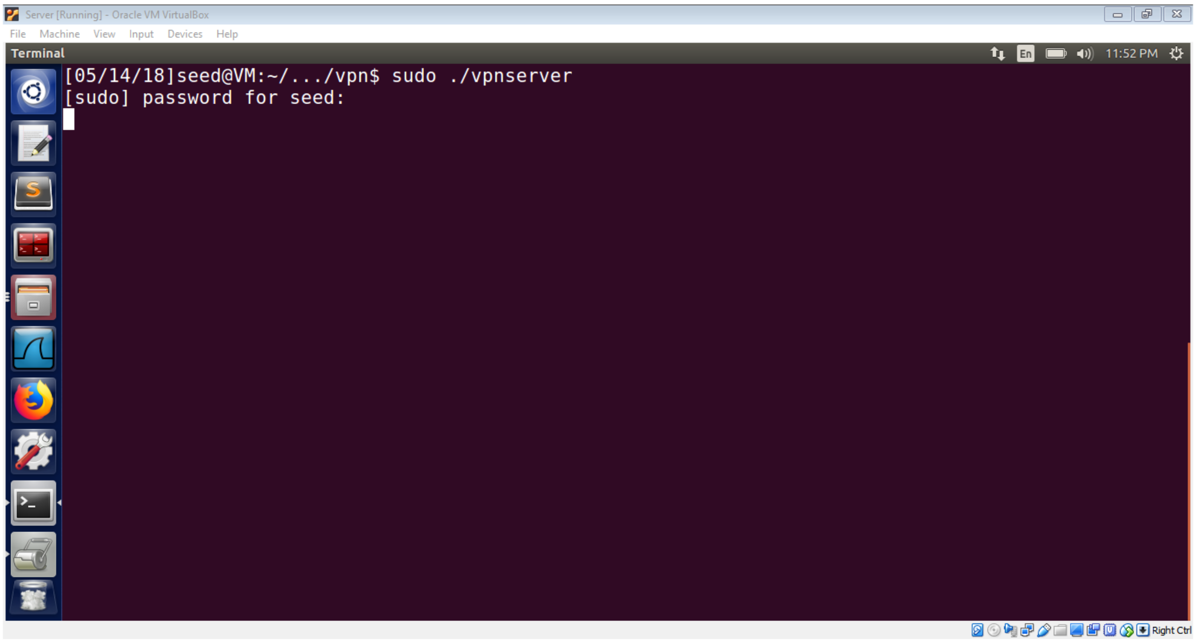


Now, to compile the VPN server and client program we will give the make command to run the make file. It is demonstrated in the below screenshot.


Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:50 PM
Right Ctrl

```
[05/14/18]seed@VM:~$ cd Downloads/vpn
[05/14/18]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[05/14/18]seed@VM:~/.../vpn$ 
```

Now we will start the server program.


Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:52 PM
Right Ctrl

```
[05/14/18]seed@VM:~/.../vpn$ sudo ./vpnserver
[sudo] password for seed:
```

After the program runs we will give the following command

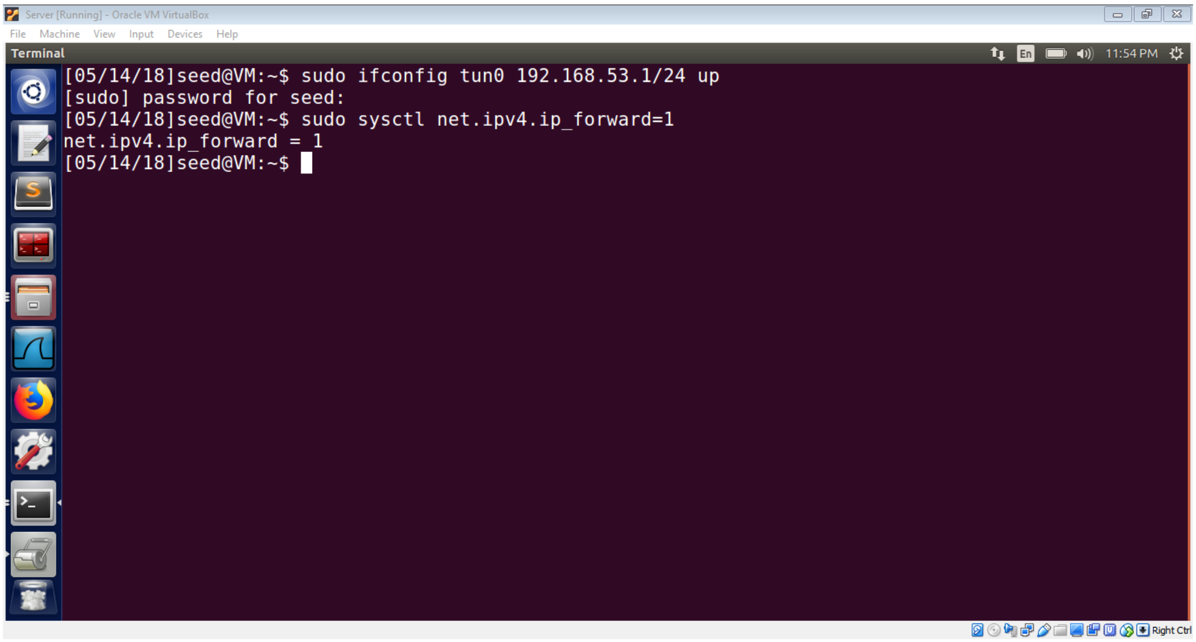
- ifconfig -a

We will observe that a virtual interface is created called “tun 0”. Now we will move onto another terminal and will give a command to assign an IP address to the “tun0” interface on the server side and to initiate the tunnel. Command that we use is

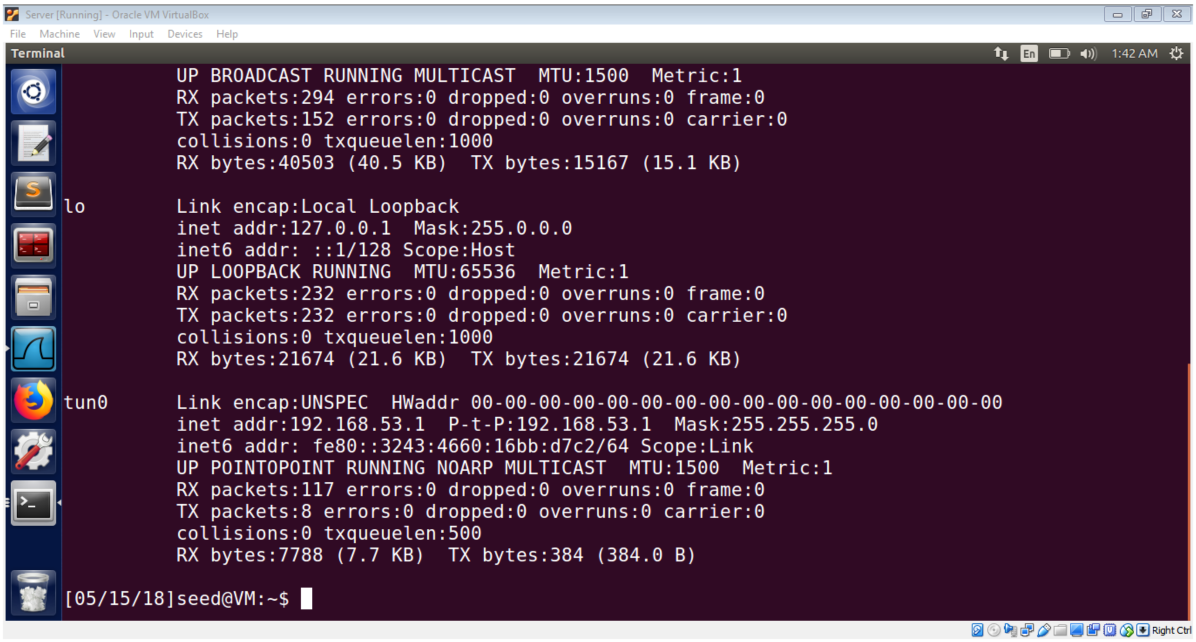
- sudo ifconfig tun0 192.168.53.1/24 up

Now since the server also must forward the packets to other destination we will enable the IPv4 packet forwarding on the server side. To do this we will use the following command

- sudo sysctl net.IPv4.IP_forward=1


Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:54 PM
Right Ctrl

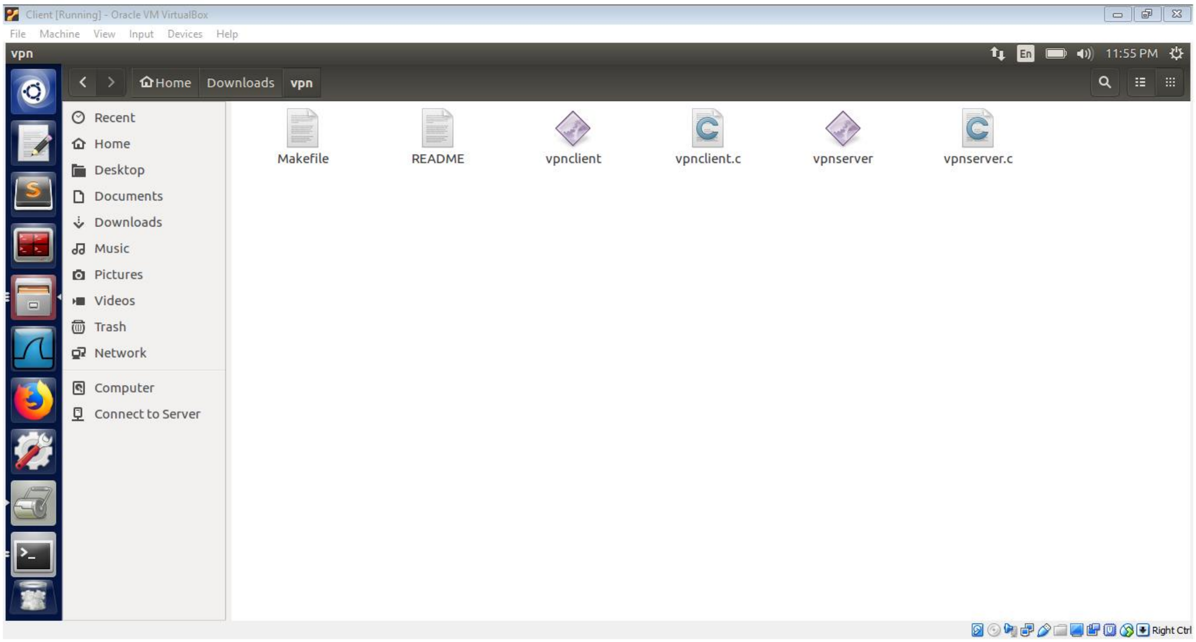
```
[05/14/18]seed@VM:~$ sudo ifconfig tun0 192.168.53.1/24 up
[sudo] password for seed:
[05/14/18]seed@VM:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[05/14/18]seed@VM:~$
```


Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
1:42 AM
Right Ctrl

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:294 errors:0 dropped:0 overruns:0 frame:0
          TX packets:152 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40503 (40.5 KB)  TX bytes:15167 (15.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:232 errors:0 dropped:0 overruns:0 frame:0
          TX packets:232 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21674 (21.6 KB)  TX bytes:21674 (21.6 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.1  P-t-P:192.168.53.1  Mask:255.255.255.0
          inet6 addr: fe80::3243:4660:16bb:d7c2/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:117 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:7788 (7.7 KB)  TX bytes:384 (384.0 B)

[05/15/18]seed@VM:~$ 
```

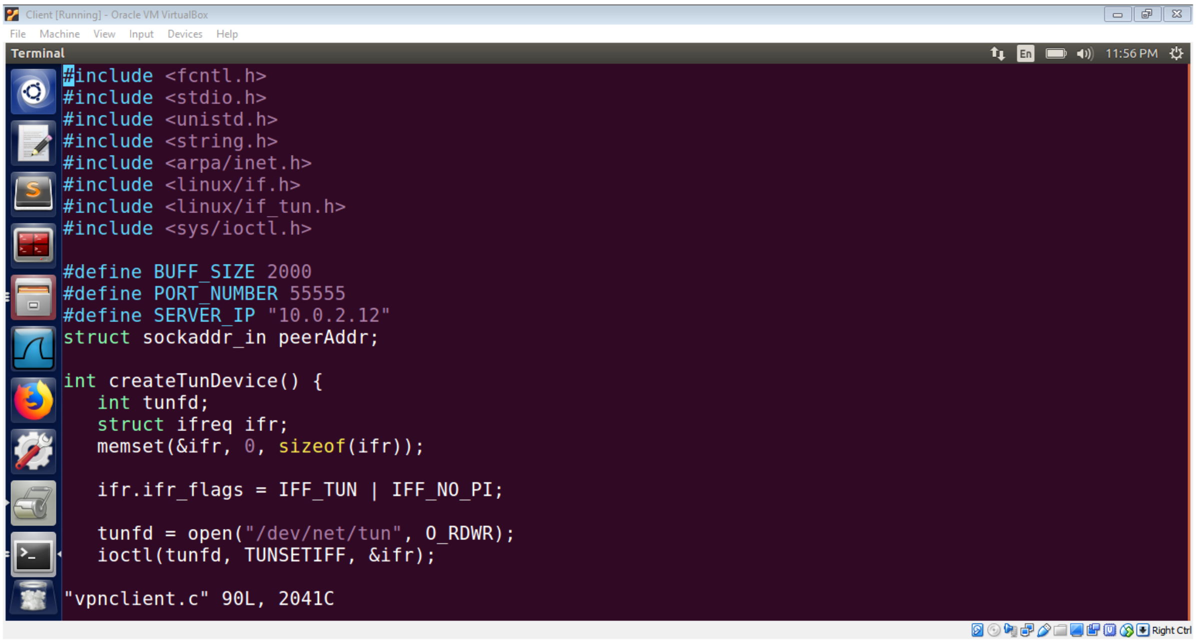
Step 2: Run the client side

After downloading the VPN server and client program from SEED labs website, we will extract all the files from the ZIP folder.



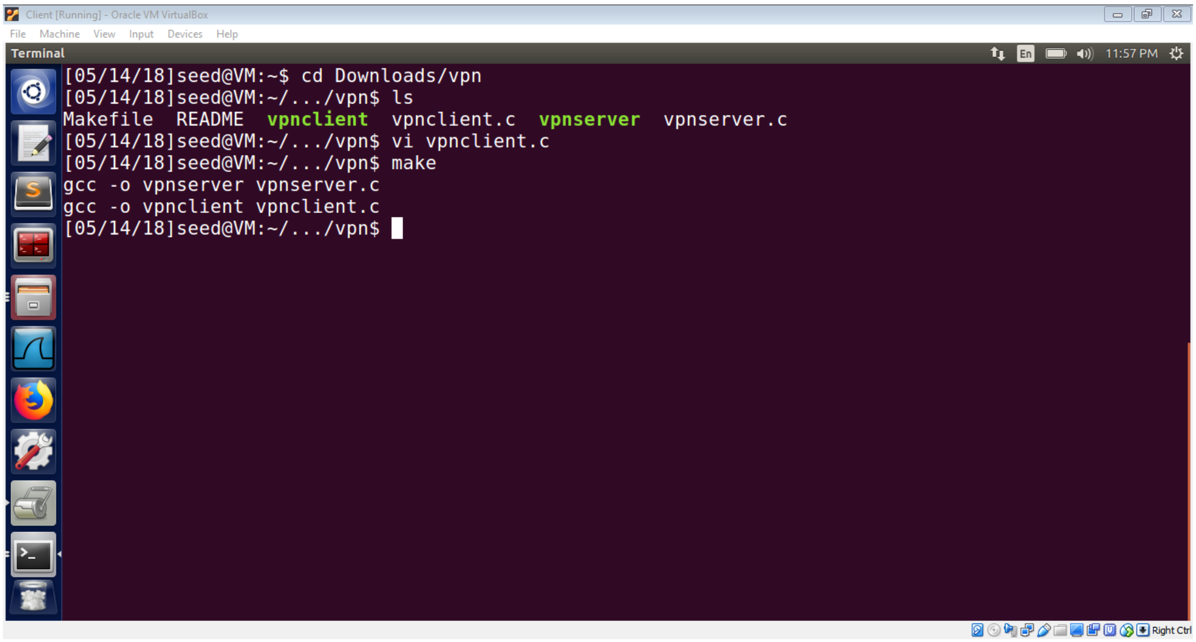
Now we will run the VPN client program which in turn will connect this VPN client program with VPN server program running on “10.0.2.12”.

But before that we will open the vpnclient.c file. Once we get access to the file we will update the SERVER_IP as 10.0.2.12 which is our “Server” machine IP address. Once we are done with this we will save and close the file.


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:56 PM
Right Ctrl

```
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define BUFF_SIZE 2000
#define PORT_NUMBER 55555
#define SERVER_IP "10.0.2.12"
struct sockaddr_in peerAddr;

int createTunDevice() {
   int tunfd;
   struct ifreq ifr;
   memset(&ifr, 0, sizeof(ifr));

   ifr.ifr_flags = IFF_TUN | IFF_NO_PI;

   tunfd = open("/dev/net/tun", O_RDWR);
   ioctl(tunfd, TUNSETIFF, &ifr);

"vpnclient.c" 90L, 2041C
```

Now to compile the VPN client program we will first run the make file.


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:57 PM
Right Ctrl

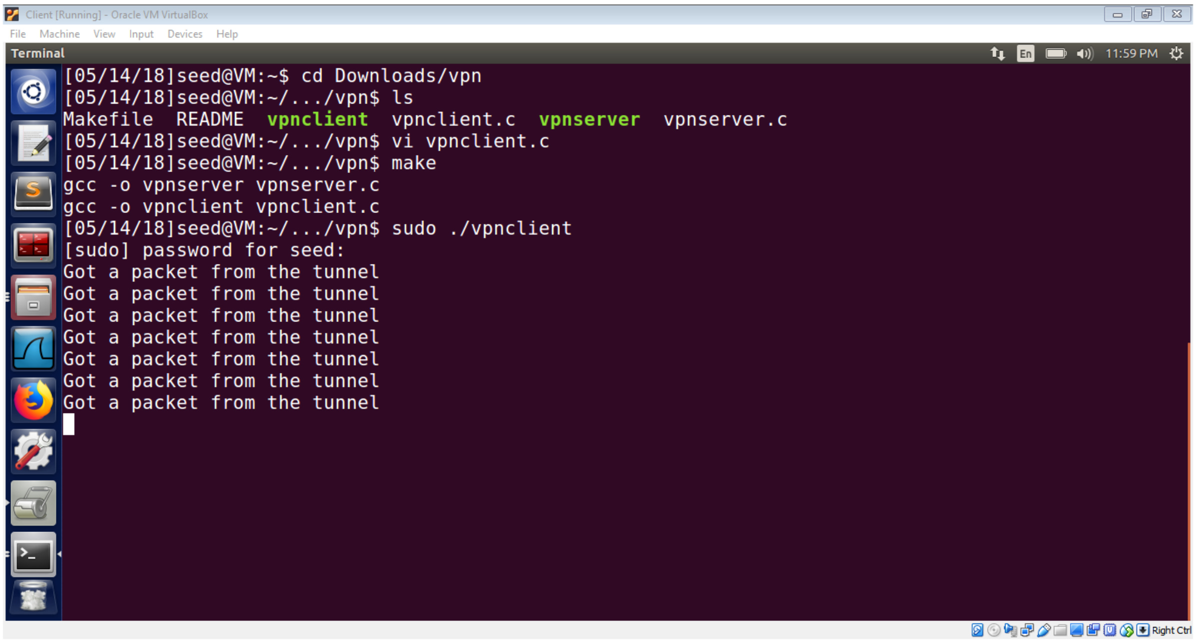
```
[05/14/18]seed@VM:~$ cd Downloads/vpn
[05/14/18]seed@VM:~/.../vpn$ ls
Makefile  README  vpnclient  vpnclient.c  vpnserver  vpnserver.c
[05/14/18]seed@VM:~/.../vpn$ vi vpnclient.c
[05/14/18]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[05/14/18]seed@VM:~/.../vpn$ 
```

Now we will run the client program. To do this we will give the following command

- sudo ./vpnclient

And again, in another terminal we will give a command to assign an IP address to “tun0” interface.

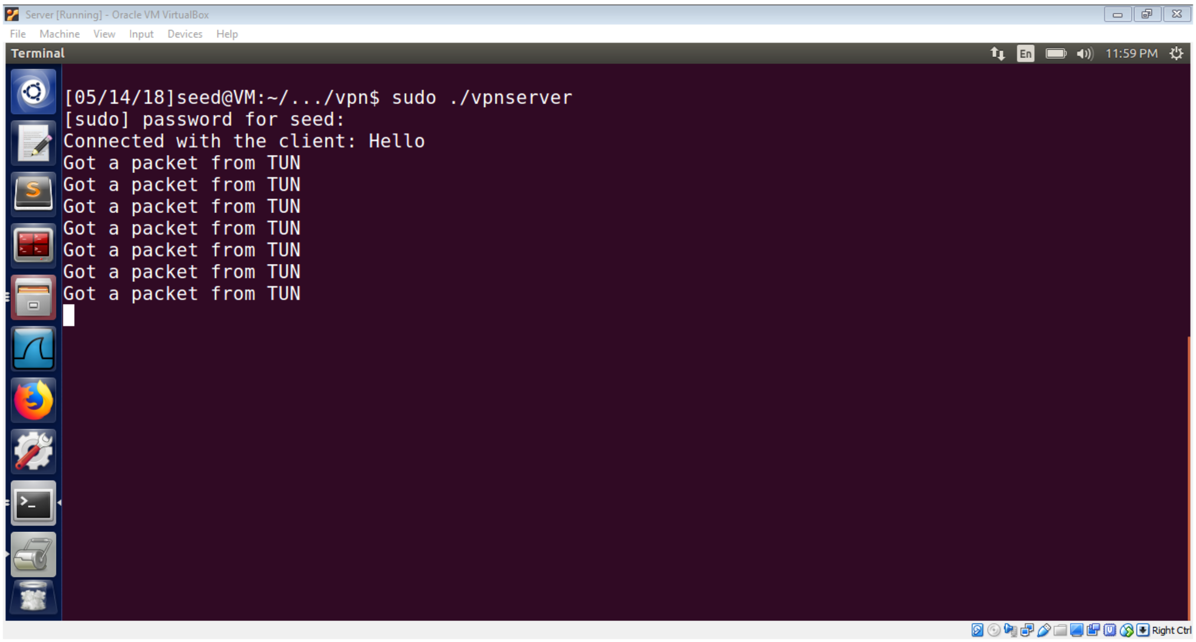
- Sudo ifconfig tun0 192.168.53.5/24 up


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:59 PM
Right Ctrl

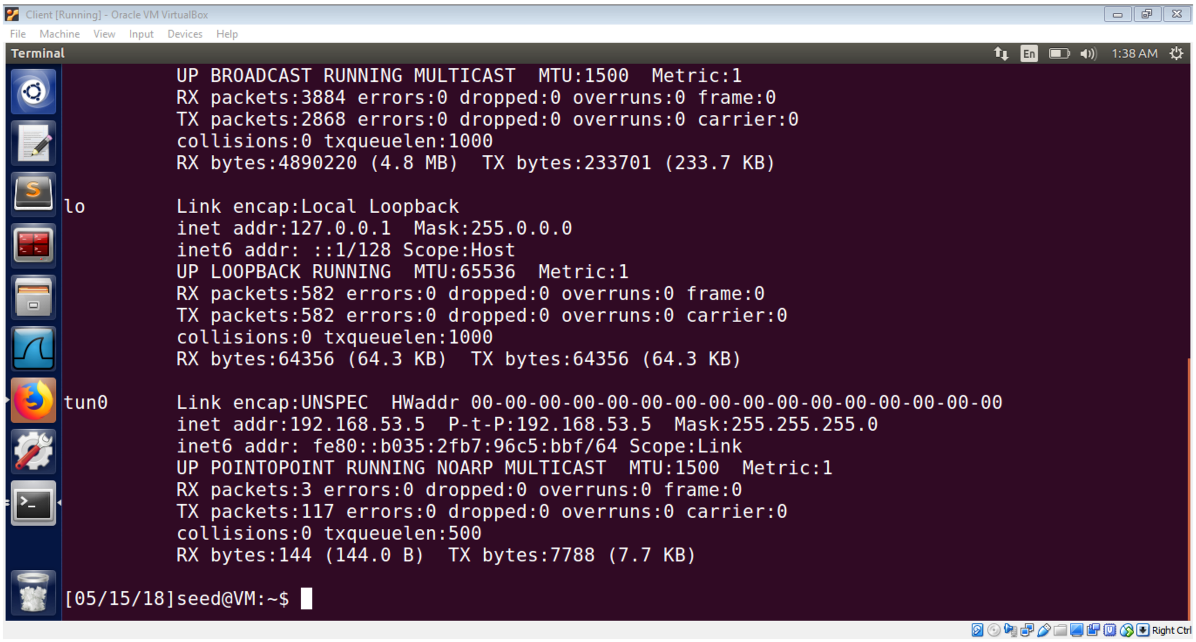
```
[05/14/18]seed@VM:~$ cd Downloads/vpn
[05/14/18]seed@VM:~/.../vpn$ ls
Makefile  README  vpnclient  vpnclient.c  vpnserver  vpnserver.c
[05/14/18]seed@VM:~/.../vpn$ vi vpnclient.c
[05/14/18]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[05/14/18]seed@VM:~/.../vpn$ sudo ./vpnclient
[sudo] password for seed:
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```

From the above screenshot we can observe that as soon as we run the VPN client program**, we see that the tunnel is established, and few packets are going from the client to the server side.**

On the server side (see the below screenshot), we see that it shows the message **"Connected with the client: Hello", and it is receiving packets from the server. It confirms that the tunnel is created.**

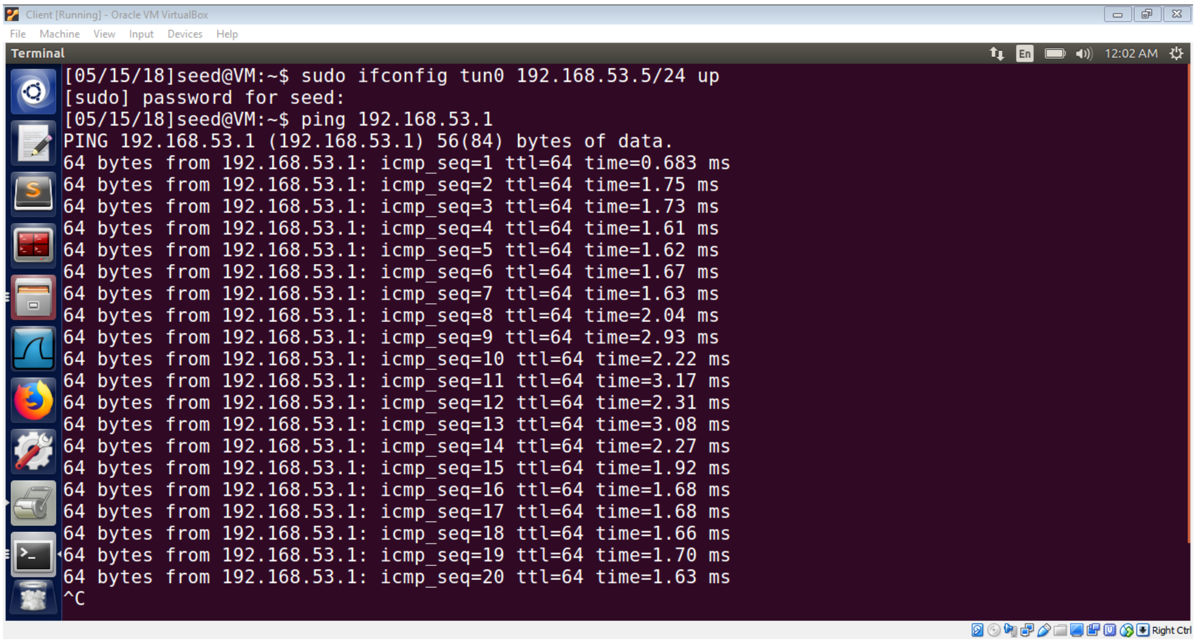

Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
11:59 PM
Right Ctrl

```
[05/14/18]seed@VM:~/.../vpn$ sudo ./vpnserver
[sudo] password for seed:
Connected with the client: Hello
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
```



Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
1:38 AM
Right Ctrl

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3884 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2868 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4890220 (4.8 MB)  TX bytes:233701 (233.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:582 errors:0 dropped:0 overruns:0 frame:0
          TX packets:582 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:64356 (64.3 KB)  TX bytes:64356 (64.3 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.5  P-t-P:192.168.53.5  Mask:255.255.255.0
          inet6 addr: fe80::b035:2fb7:96c5:bbf/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:117 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:144 (144.0 B)  TX bytes:7788 (7.7 KB)

[05/15/18]seed@VM:~$ 
```



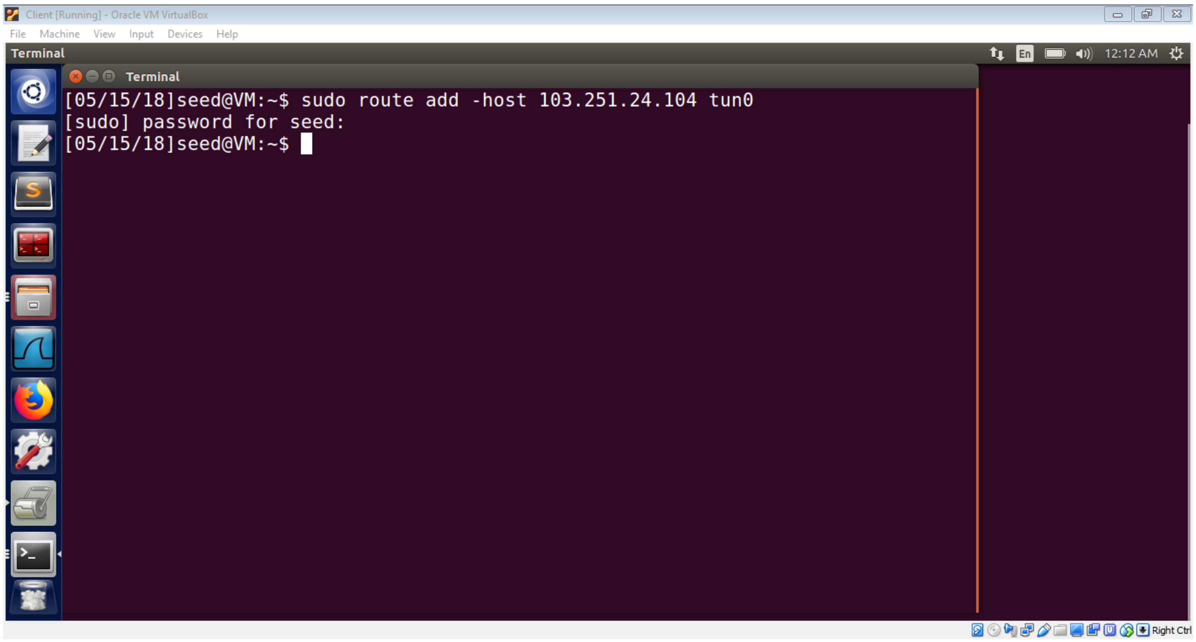
Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
12:02 AM
Right Ctrl

```
[05/15/18]seed@VM:~$ sudo ifconfig tun0 192.168.53.5/24 up
[sudo] password for seed:
[05/15/18]seed@VM:~$ ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.1: icmp_seq=1 ttl=64 time=0.683 ms
64 bytes from 192.168.53.1: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 192.168.53.1: icmp_seq=3 ttl=64 time=1.73 ms
64 bytes from 192.168.53.1: icmp_seq=4 ttl=64 time=1.61 ms
64 bytes from 192.168.53.1: icmp_seq=5 ttl=64 time=1.62 ms
64 bytes from 192.168.53.1: icmp_seq=6 ttl=64 time=1.67 ms
64 bytes from 192.168.53.1: icmp_seq=7 ttl=64 time=1.63 ms
64 bytes from 192.168.53.1: icmp_seq=8 ttl=64 time=2.04 ms
64 bytes from 192.168.53.1: icmp_seq=9 ttl=64 time=2.93 ms
64 bytes from 192.168.53.1: icmp_seq=10 ttl=64 time=2.22 ms
64 bytes from 192.168.53.1: icmp_seq=11 ttl=64 time=3.17 ms
64 bytes from 192.168.53.1: icmp_seq=12 ttl=64 time=2.31 ms
64 bytes from 192.168.53.1: icmp_seq=13 ttl=64 time=3.08 ms
64 bytes from 192.168.53.1: icmp_seq=14 ttl=64 time=2.27 ms
64 bytes from 192.168.53.1: icmp_seq=15 ttl=64 time=1.92 ms
64 bytes from 192.168.53.1: icmp_seq=16 ttl=64 time=1.68 ms
64 bytes from 192.168.53.1: icmp_seq=17 ttl=64 time=1.68 ms
64 bytes from 192.168.53.1: icmp_seq=18 ttl=64 time=1.66 ms
64 bytes from 192.168.53.1: icmp_seq=19 ttl=64 time=1.70 ms
64 bytes from 192.168.53.1: icmp_seq=20 ttl=64 time=1.63 ms
^C
```

The above screenshot shows commands that we use to assign IP address to the tunnel interface on the client side. Also**, we will use the ping command** to check whether the packets are going through tunnel or not. After pinging we see that packet are being transmitted that means **the tunnel is successfully established between the VPN client and VPN server**.

Step 3: Set up Routing on Client VM

After the tunnel is established we will set up a routing path on the Client VM so that the traffic we want to pass through the tunnel can pass through it**. We want the packet destined to “103.251.24.104” to pass through the interface “tun0”**. The below screenshot demonstrates the command we use to do this.


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
En
12:12 AM
Terminal
Right Ctrl

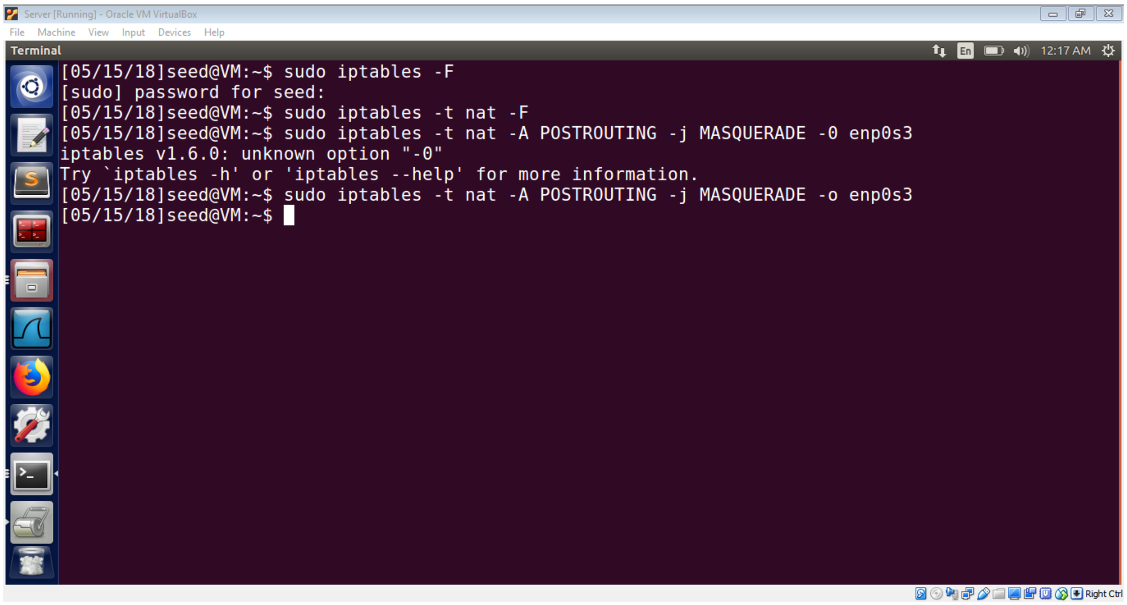
```
[05/15/18]seed@VM:~$ sudo route add -host 103.251.24.104 tun0
[sudo] password for seed:
[05/15/18]seed@VM:~$ 
```

Step 4: Setup NAT on server VM

Why the packet will reach the VPN server first?

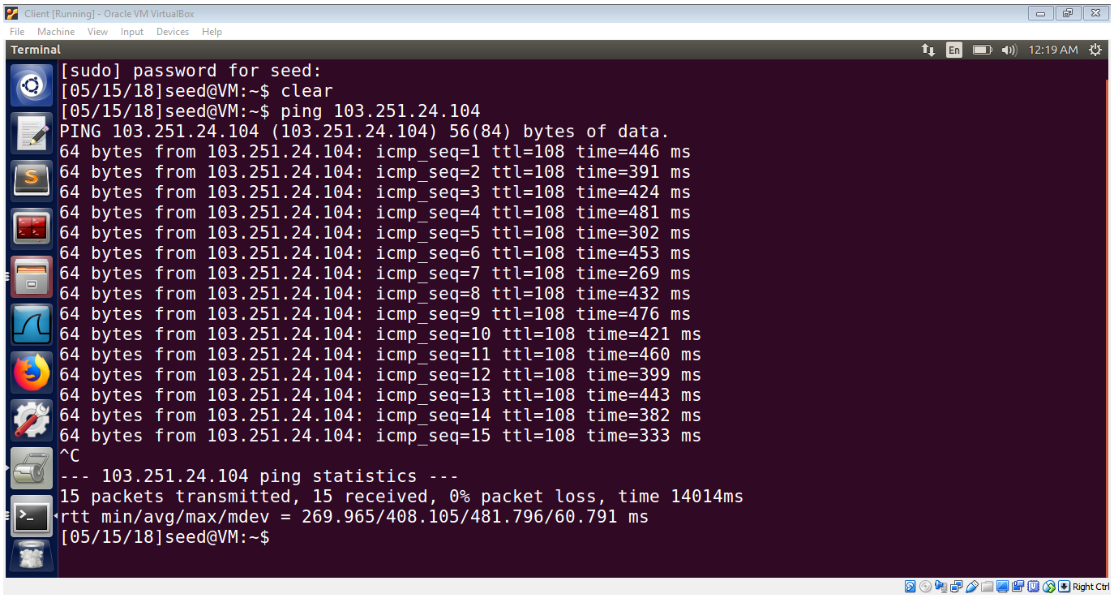
Because we will enable the NAT on the “Server” machine. So, when the packet is sent to destination from the server as it leaves the server because of NAT, its private IP is changed to Public IP address, which is the IP address of the server. So now whenever the response from the website will come then it will be directed to the public IP address of the “Server” machine. Now as it passes the NAT again the response packet will be directed to the private address of VPN server.

Now we will give the following commands (shown in the below screenshot) to enable the NAT on the server VM. Using this we won’t require to fool the NAT anymore by using the ARP cache poisoning.

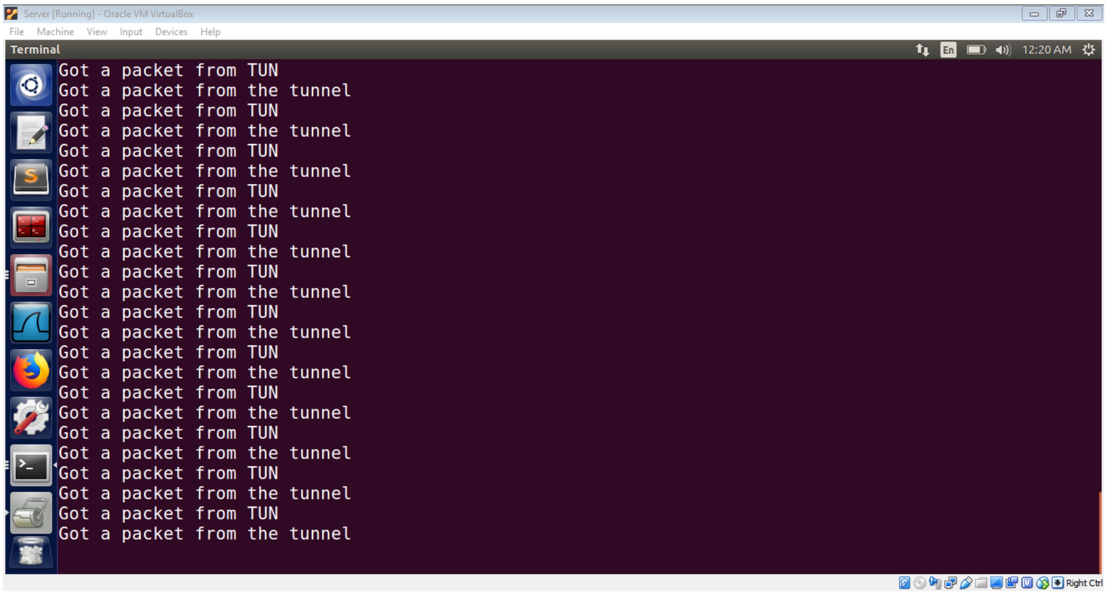

Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
En
12:17 AM
Right Ctrl

```
[05/15/18]seed@VM:~$ sudo iptables -F
[sudo] password for seed:
[05/15/18]seed@VM:~$ sudo iptables -t nat -F
[05/15/18]seed@VM:~$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE -0 enp0s3
iptables v1.6.0: unknown option "-0"
Try `iptables -h' or 'iptables --help' for more information.
[05/15/18]seed@VM:~$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE -o enp0s3
[05/15/18]seed@VM:~$ 
```

Now, the tunnel is successfully created, and all the required routes are added. Now we will check whether we are able to bypass the firewall by using VPN server. For this we will ping the “oistbpl.com” again


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
12:19 AM
Right Ctrl

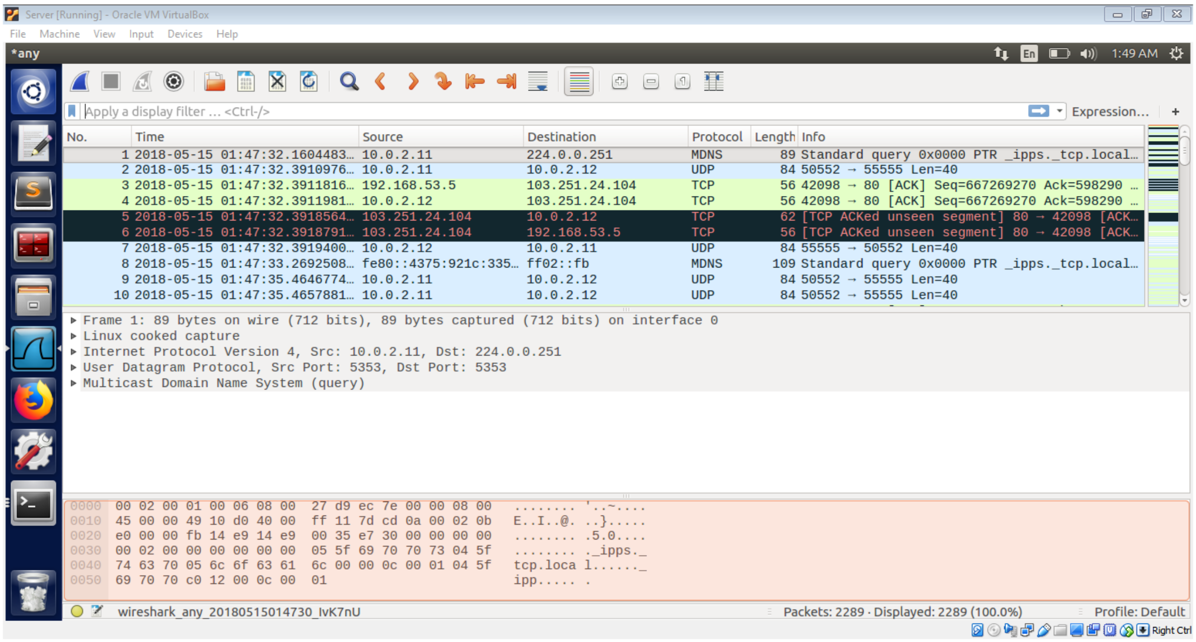
```
[sudo] password for seed:
[05/15/18]seed@VM:~$ clear
[05/15/18]seed@VM:~$ ping 103.251.24.104
PING 103.251.24.104 (103.251.24.104) 56(84) bytes of data.
64 bytes from 103.251.24.104: icmp_seq=1 ttl=108 time=446 ms
64 bytes from 103.251.24.104: icmp_seq=2 ttl=108 time=391 ms
64 bytes from 103.251.24.104: icmp_seq=3 ttl=108 time=424 ms
64 bytes from 103.251.24.104: icmp_seq=4 ttl=108 time=481 ms
64 bytes from 103.251.24.104: icmp_seq=5 ttl=108 time=302 ms
64 bytes from 103.251.24.104: icmp_seq=6 ttl=108 time=453 ms
64 bytes from 103.251.24.104: icmp_seq=7 ttl=108 time=269 ms
64 bytes from 103.251.24.104: icmp_seq=8 ttl=108 time=432 ms
64 bytes from 103.251.24.104: icmp_seq=9 ttl=108 time=476 ms
64 bytes from 103.251.24.104: icmp_seq=10 ttl=108 time=421 ms
64 bytes from 103.251.24.104: icmp_seq=11 ttl=108 time=460 ms
64 bytes from 103.251.24.104: icmp_seq=12 ttl=108 time=399 ms
64 bytes from 103.251.24.104: icmp_seq=13 ttl=108 time=443 ms
64 bytes from 103.251.24.104: icmp_seq=14 ttl=108 time=382 ms
64 bytes from 103.251.24.104: icmp_seq=15 ttl=108 time=333 ms
^C
--- 103.251.24.104 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14014ms
rtt min/avg/max/mdev = 269.965/408.105/481.796/60.791 ms
[05/15/18]seed@VM:~$
```


Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
12:20 AM
Right Ctrl

```
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
```

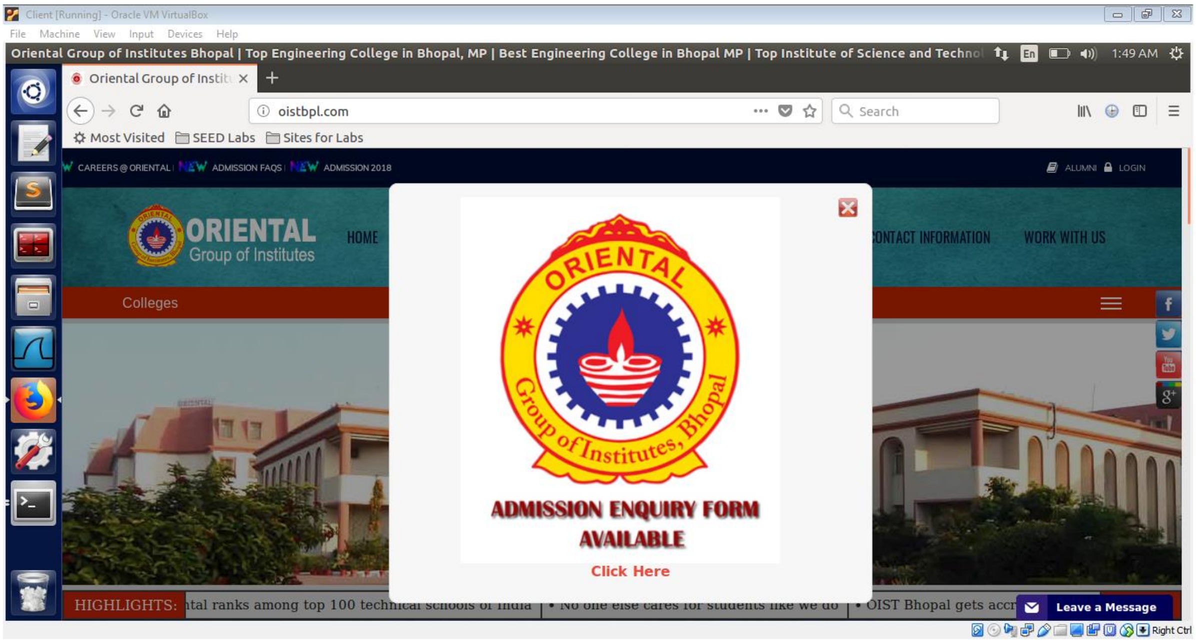
From the above two screenshot we can say that the firewall has been bypassed successfully through the VPN tunnel, as on the “Client” machine when we ping the target website, on the “Server” machine we are receiving the packets from the tunnel.

We can also showcase that the traffic went through the tunnel only and not through any side door by capturing the packets through Wireshark.


Server [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
*any
1:49 AM
Apply a display filter … <Ctrl-/>
Expression…

```
No.  Time                          Source                 Destination      Protocol Length Info
   1 2018-05-15 01:47:32.1604483… 10.0.2.11              224.0.0.251      MDNS       89 Standard query 0x0000 PTR _ipps._tcp.local…
   2 2018-05-15 01:47:32.3910976… 10.0.2.11              10.0.2.12        UDP        84 50552 → 55555 Len=40
   3 2018-05-15 01:47:32.3911816… 192.168.53.5           103.251.24.104   TCP        56 42098 → 80 [ACK] Seq=667269270 Ack=598290 …
   4 2018-05-15 01:47:32.3911981… 10.0.2.12              103.251.24.104   TCP        56 42098 → 80 [ACK] Seq=667269270 Ack=598290 …
   5 2018-05-15 01:47:32.3918564… 103.251.24.104         10.0.2.12        TCP        62 [TCP ACKed unseen segment] 80 → 42098 [ACK…
   6 2018-05-15 01:47:32.3918791… 103.251.24.104         192.168.53.5     TCP        56 [TCP ACKed unseen segment] 80 → 42098 [ACK…
   7 2018-05-15 01:47:32.3919400… 10.0.2.12              10.0.2.11        UDP        84 55555 → 50552 Len=40
   8 2018-05-15 01:47:33.2692508… fe80::4375:921c:335…  ff02::fb         MDNS      109 Standard query 0x0000 PTR _ipps._tcp.local…
   9 2018-05-15 01:47:35.4646774… 10.0.2.11              10.0.2.12        UDP        84 50552 → 55555 Len=40
  10 2018-05-15 01:47:35.4657881… 10.0.2.11              10.0.2.12        UDP        84 50552 → 55555 Len=40

▸ Frame 1: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 10.0.2.11, Dst: 224.0.0.251
▸ User Datagram Protocol, Src Port: 5353, Dst Port: 5353
▸ Multicast Domain Name System (query)

0000  00 02 00 01 00 06 08 00  27 d9 ec 7e 00 00 08 00   ........ '..~....
0010  45 00 00 49 10 d0 40 00  ff 11 7d cd 0a 00 02 0b   E..I..@. ..}.....
0020  e0 00 00 fb 14 e9 14 e9  00 35 e7 30 00 00 00 00   ........ .5.0....
0030  00 02 00 00 00 00 00 00  05 5f 69 70 70 73 04 5f   ........ ._ipps._
0040  74 63 70 05 6c 6f 63 61  6c 00 00 0c 00 01 04 5f   tcp.loca l......_
0050  69 70 70 c0 12 00 0c 00  01                        ipp..... .

wireshark_any_20180515014730_IvK7nU    Packets: 2289 · Displayed: 2289 (100.0%)    Profile: Default
```

From the above screenshot we can see that the packet is going from “192.168.53.5” i.e., from client to “oistbpl.com”. First it goes to the “Server” VM and then to the target website. The response packet from the website first goes to the “Server” VM and from there it comes to the Client tunnel interface. So, the firewall is bypassed using tunnel itself and not through any side door.


Client [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Oriental Group of Institutes Bhopal | Top Engineering College in Bhopal, MP | Best Engineering College in Bhopal MP | Top Institute of Science and Techno
1:49 AM
Oriental Group of Instit
oistbpl.com
Search
Most Visited
SEED Labs
Sites for Labs
CAREERS @ ORIENTAL
NEW ADMISSION FAQS
NEW ADMISSION 2018
ALUMNI
LOGIN
ORIENTAL
Group of Institutes
HOME
CONTACT INFORMATION
WORK WITH US
Colleges
ORIENTAL
Group of Institutes, Bhopal
ADMISSION ENQUIRY FORM
AVAILABLE
Click Here
HIGHLIGHTS:
ntal ranks among top 100 technical schools of India
• No one else cares for students like we do
• OIST Bhopal gets accr
Leave a Message

Now, when we try to access the webpage of “oistbpl.com” we see that we can access it. Although the firewall rules are still enabled on the “Client” machine which blocks the access to the webpage we requested in the above screenshot, but we successfully evaded the firewall by using the IP tunneling technology of VPN and therefore we are able to access the target website.