

CNS LAB

LAB - 7

Firewall Evasion Lab: Bypassing Firewalls using VPN

NAME : SIRI S

SEMESTER : 5

SECTION :H

SRN : PES1UG19CS485

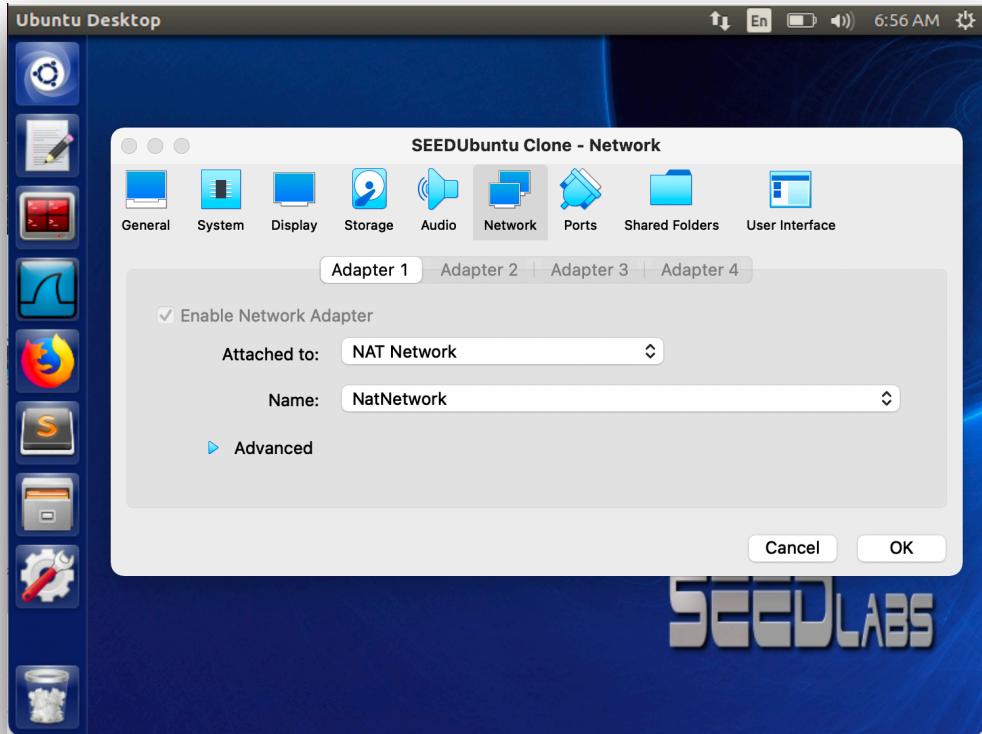
LAB SETUP:

VPN SERVER: 10. 0. 2. 5

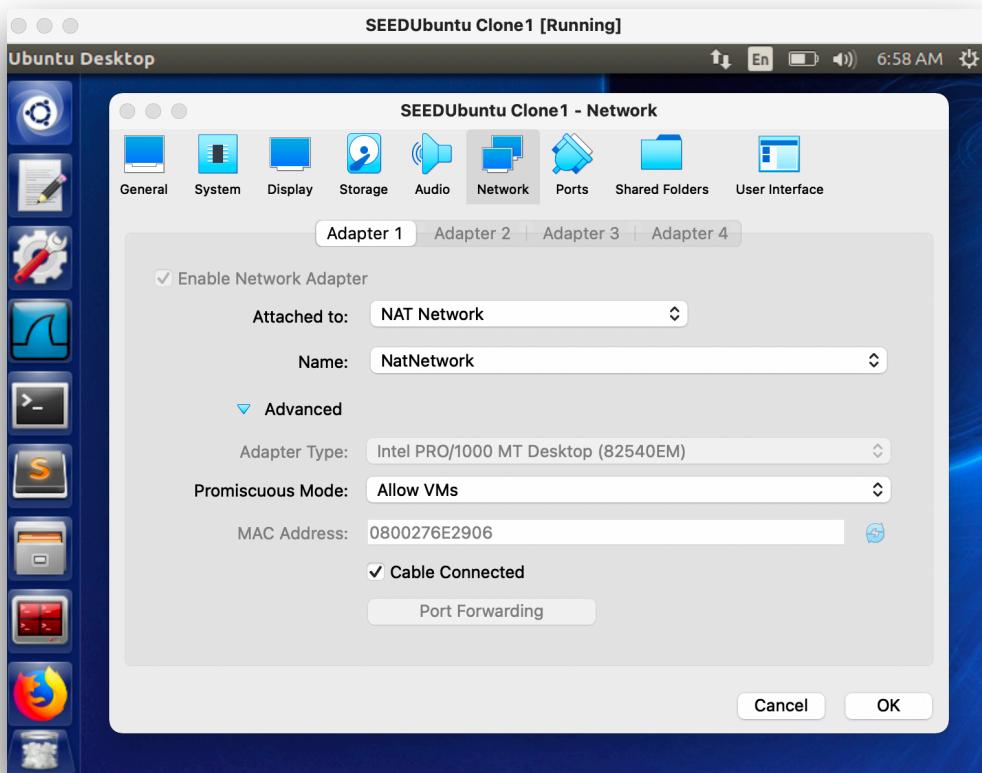
VPN CLIENT: 10. 0. 2. 6

Task 1: VM Setup

Both the VMs need to have the “NAT Network” Adapter enabled on them:



Server machine network configuration



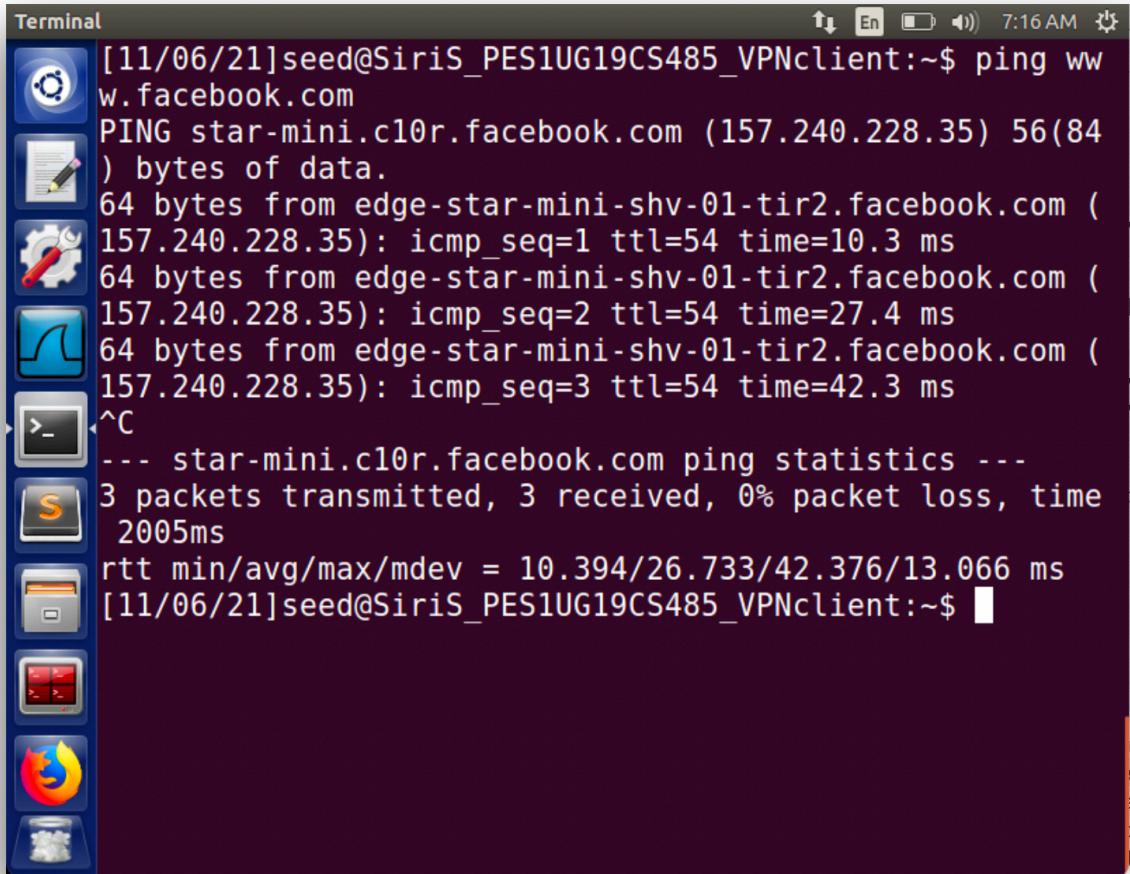
Client machine network configuration

Task 2: Set up Firewall

We have identified '**www.facebook.com**' as a website to be blocked on the client machine. The client machine is inside the firewall.

The website is able to be pinged from the client machine:

IP Address: 157. 240. 228. 35



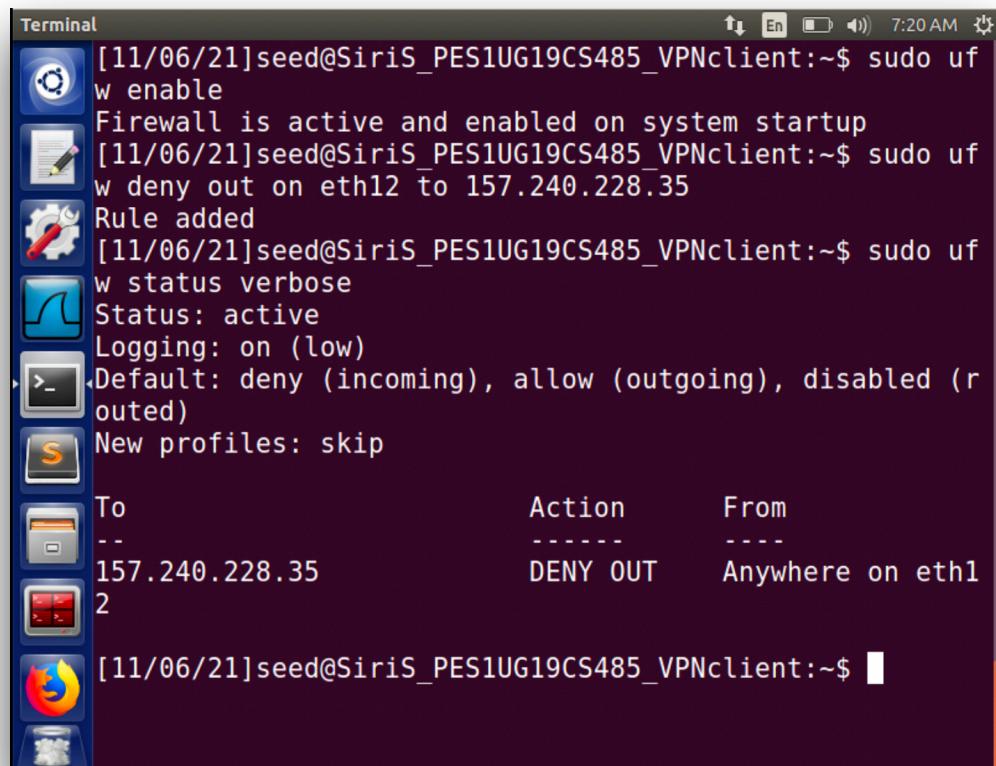
A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal". The terminal content shows the output of a ping command to www.facebook.com. The output includes several lines of ICMP echo requests being sent to the IP address 157.240.228.35, with responses from edge-star-mini-shv-01-tir2.facebook.com. The terminal window has a dark background and light-colored text. The desktop interface includes a dock with various icons on the left side of the terminal window.

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ ping www.facebook.com
PING star-mini.c10r.facebook.com (157.240.228.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=1 ttl=54 time=10.3 ms
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=2 ttl=54 time=27.4 ms
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=3 ttl=54 time=42.3 ms
^C
--- star-mini.c10r.facebook.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 10.394/26.733/42.376/13.066 ms
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$
```

Therefore we can see that Facebook has a fixed IP Address.

Now, we will set up a firewall to block this website from being accessed.

We can see that www.facebook.com is blocked:

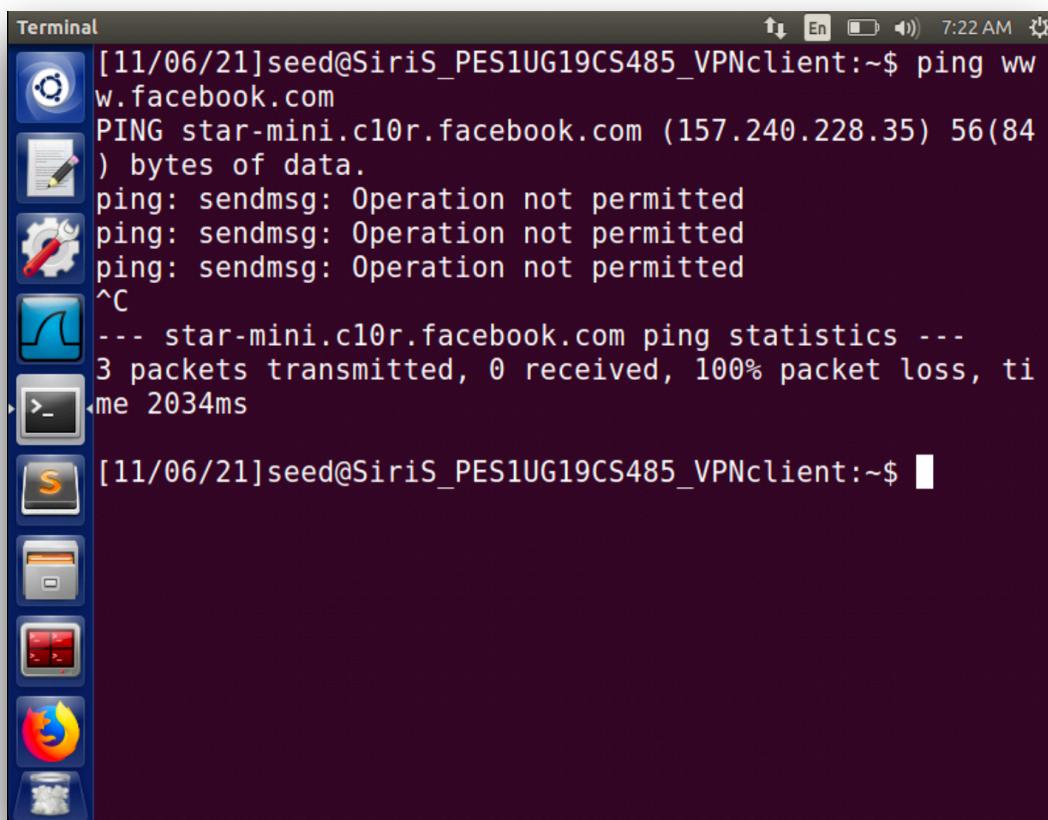


A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal displays the output of several commands related to the Uncomplicated Firewall (ufw). The commands and their outputs are as follows:

- [11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~\$ sudo ufw enable
Firewall is active and enabled on system startup
- [11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~\$ sudo ufw deny out on eth12 to 157.240.228.35
Rule added
- [11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~\$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
- To Action From

157.240.228.35 DENY OUT Anywhere on eth1
2

Now, we are unable to ping the website:



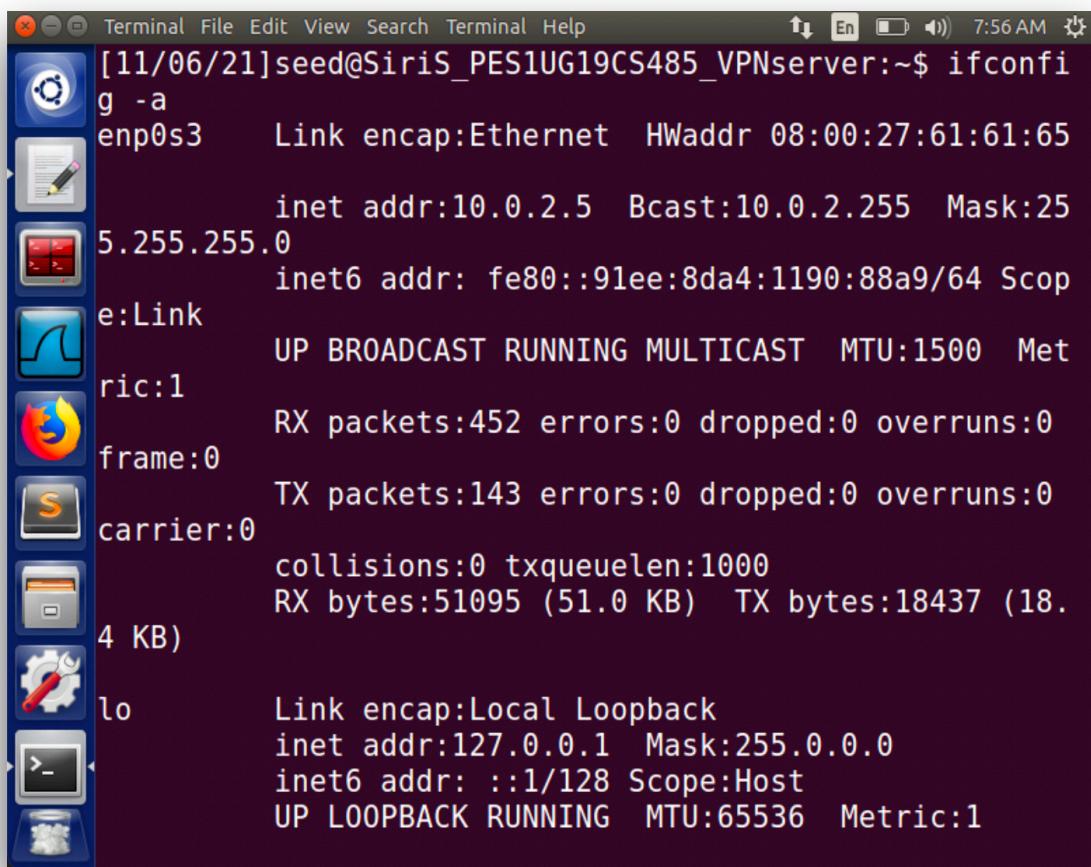
A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal displays the output of a ping command to the website www.facebook.com. The output shows that the ping failed due to an operation not permitted error, indicating network restrictions.

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ ping www.facebook.com  
PING star-mini.c10r.facebook.com (157.240.228.35) 56(84) bytes of data.  
ping: sendmsg: Operation not permitted  
ping: sendmsg: Operation not permitted  
ping: sendmsg: Operation not permitted  
^C  
--- star-mini.c10r.facebook.com ping statistics ---  
3 packets transmitted, 0 received, 100% packet loss, time 2034ms
```

Task 3: Bypassing Firewall using VPN

Step 1: Run VPN Server:

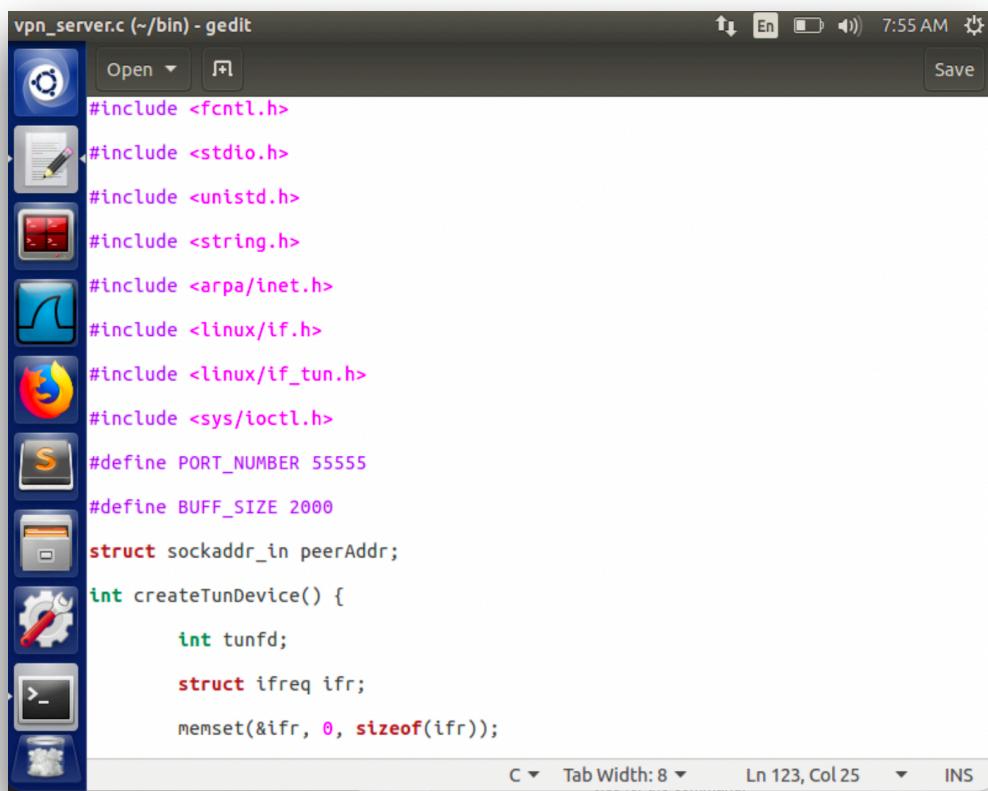
Before running the server code we can see that when we run *ifconfig -a*, tun0 (the name of the interface that establishes a VPN tunnel) Is inactive:



```
[11/06/21]seed@Siris_PES1UG19CS485_VPNserver:~$ ifconfig -a
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:61:61:65
              inet  addr:10.0.2.5    Bcast:10.0.2.255  Mask:25
                           5.255.255.0
              inet6 addr: fe80::91ee:8da4:1190:88a9/64  Scop
                           e:Link
                           UP BROADCAST RUNNING MULTICAST  MTU:1500  Met
                           ric:1
                           RX packets:452 errors:0 dropped:0 overruns:0
                           frame:0
                           TX packets:143 errors:0 dropped:0 overruns:0
                           carrier:0
                           collisions:0 txqueuelen:1000
                           RX bytes:51095 (51.0 KB)  TX bytes:18437 (18.
                           4 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
```

There is no mention of *tun0*.

The code that is run on the server to establish the tunnel:



A screenshot of a Gedit text editor window titled "vpn_server.c (~/.bin) - gedit". The code in the editor is as follows:

```
#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 PORT_NUMBER 55555
#define BUFF_SIZE 2000
struct sockaddr_in peerAddr;
int createTunDevice() {
    int tunfd;
    struct ifreq ifr;
    memset(&ifr, 0, sizeof(ifr));
```

The status bar at the bottom shows "C" with a dropdown arrow, "Tab Width: 8", "Ln 123, Col 25", and "INS".

Now, we run the above code:



A screenshot of a terminal window titled "Terminal". The command history shows:

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~/bin/code
$ gcc -o vpn_server vpn_server.c -Wall
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~/bin/code
$ sudo ./vpn_server
```

Then we assign an IP address to the tun0 interface and activate it. IP Address assigned: 192.168.53.1/25

Upon checking ifconfig -a : we have an established tunnel:

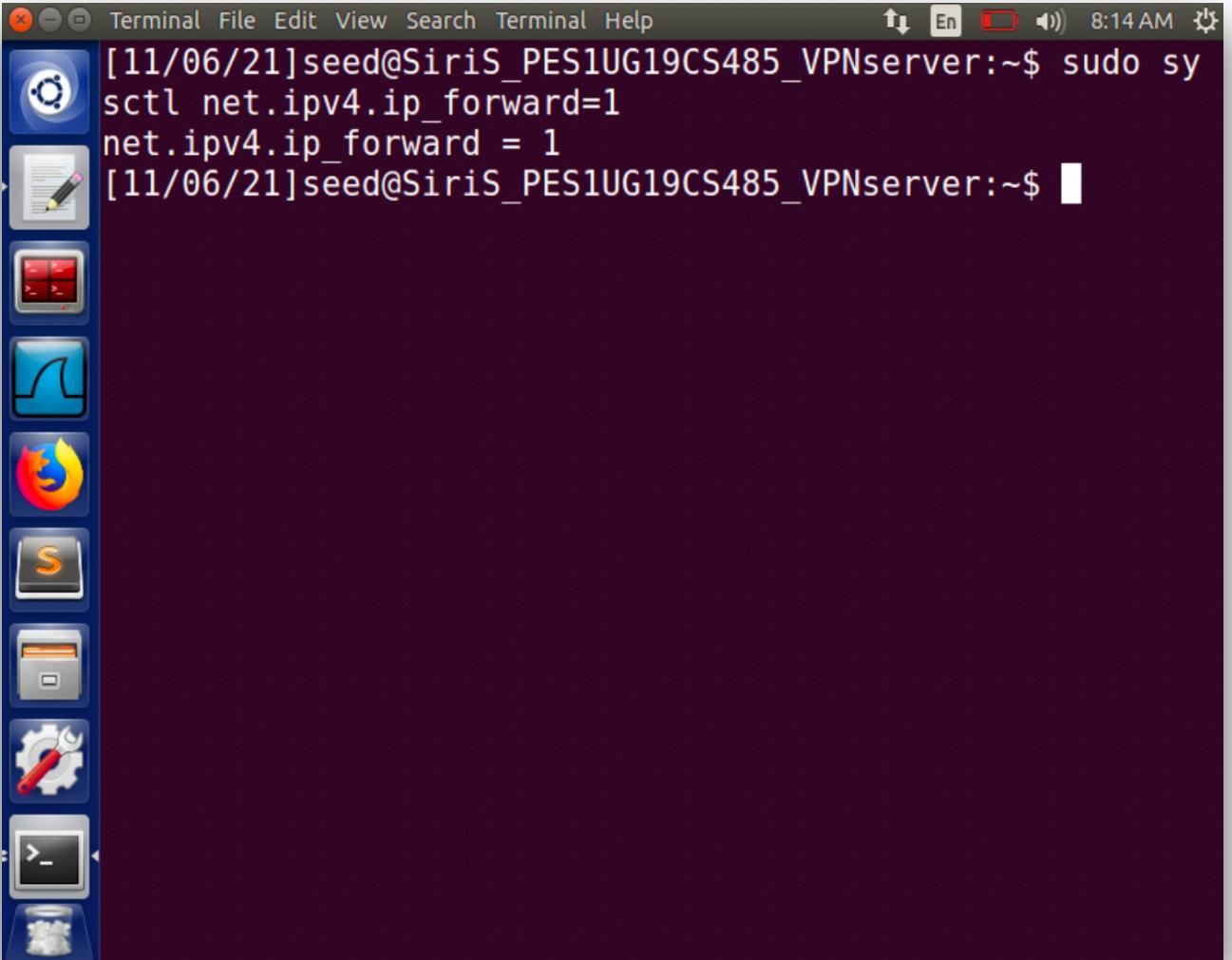
```
Terminal [11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$ sudo if config tun0 192.168.53.1/24 up
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$ ifconfig -a
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:61:61:65
              inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:25
              inet6 addr: fe80::91ee:8da4:1190:88a9/64 Scop
              e:Link
              UP BROADCAST RUNNING MULTICAST  MTU:1500  Met
              ric:1
              RX packets:462 errors:0 dropped:0 overruns:0
              frame:0
              TX packets:162 errors:0 dropped:0 overruns:0
              carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:53525 (53.5 KB)  TX bytes:21280 (21.
              2 KB)
lo          Link encap:Local Loopback
              inet addr:127.0.0.1  Mask:255.0.0.0
```

```
Terminal
carrier:0
              TX packets:327 errors:0 dropped:0 overruns:0
              collisions:0 txqueuelen:1
              RX bytes:35897 (35.8 KB)  TX bytes:35897 (35.
              8 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-0
0-00-00-00-00-00-00-00
              inet addr:192.168.53.1  P-t-P:192.168.53.1  M
              ask:255.255.255.0
              inet6 addr: fe80::dfd9:c59f:a8a8:ef3a/64 Scop
              e: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)

[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$
```

Now the VPN Server needs to forward packets to other destinations, so it needs to function as a gateway. We need to enable the IP forwarding for a computer to behave like a gateway.

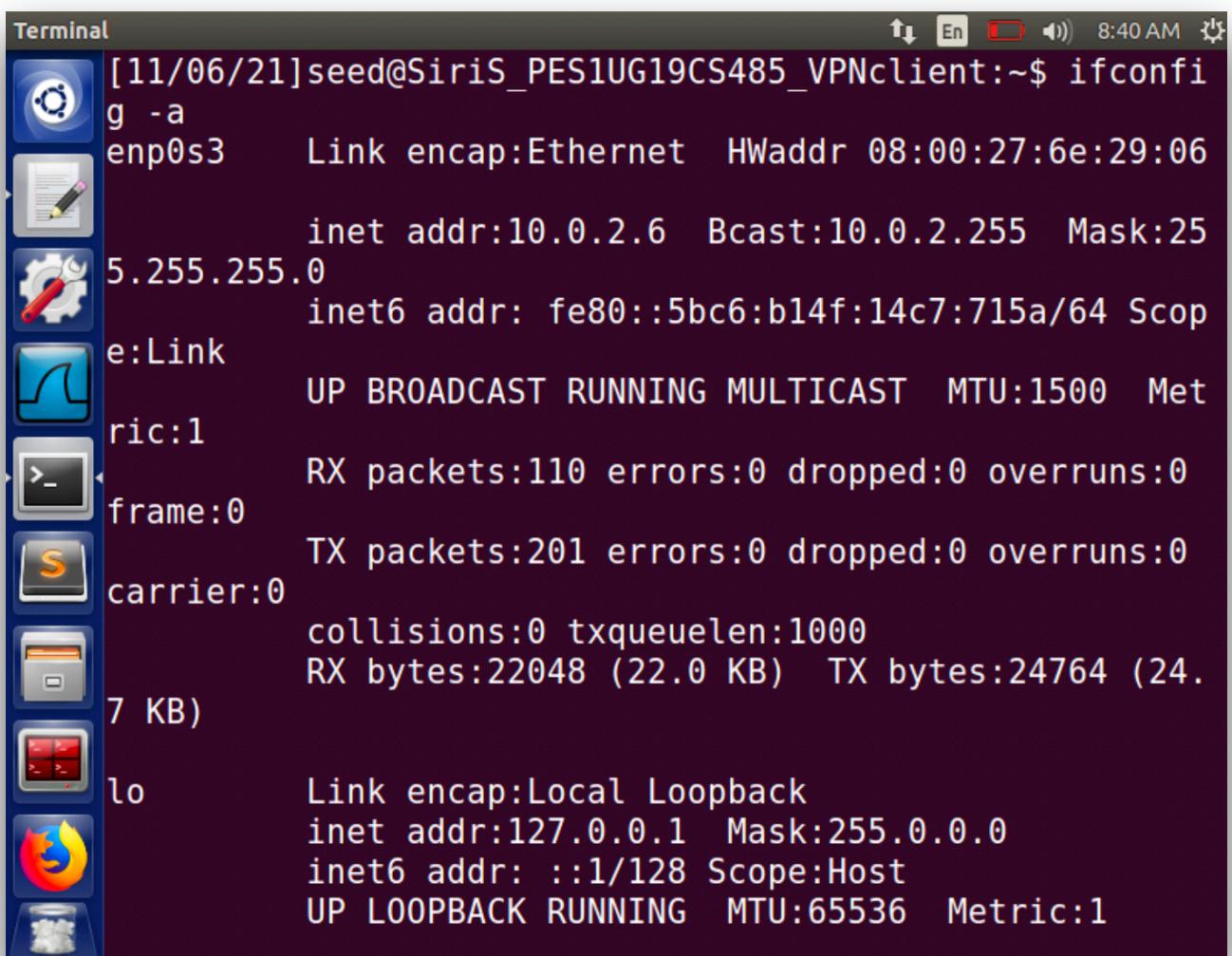


The screenshot shows a Linux desktop environment with a dark blue background. On the left, there is a vertical dock containing icons for various applications: a terminal window, a file manager, a text editor, a browser, a system settings icon with an orange 'S', a file browser, a gear and wrench icon, a terminal icon, and a trash can icon. The main window is a terminal window titled 'Terminal' with the command line 'Terminal File Edit View Search Terminal Help'. The status bar at the top right shows the date and time as '[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~\$' and the time '8:14 AM'. The terminal window displays the following command and its output:

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$
```

Step 2: Run VPN Client

Before running the client code we can see that when we run `ifconfig -a`, tun0 (the name of the interface that establishes a VPN tunnel) Is inactive:

A screenshot of a Linux desktop environment, specifically Ubuntu, showing a terminal window. The terminal window title is "Terminal". The command `ifconfig -a` has been run, displaying the configuration for various network interfaces. The output shows:

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ ifconfig -a
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:6e:29:06
              inet  addr:10.0.2.6    Bcast:10.0.2.255  Mask:25
                        5.255.255.0
              inet6 addr: fe80::5bc6:b14f:14c7:715a/64  Scop
e:Link
              UP BROADCAST RUNNING MULTICAST  MTU:1500  Met
ric:1
              RX packets:110 errors:0 dropped:0 overruns:0
              TX packets:201 errors:0 dropped:0 overruns:0
              carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:22048 (22.0 KB)   TX bytes:24764 (24.
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
```

The terminal window is located in the bottom right corner of the desktop, with other application icons visible in the dock.

There is no mention of `tun0`.

The code that is run on the server to establish the tunnel:

```
vpn_client.c (~/bin) - gedit
Open ▾ Save
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define PORT_NUMBER 55555
#define SERVER_IP "10.0.2.5"
#define BUFF_SIZE 2000
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);

    return tunfd;
}

int connectToUDPServer(){
```

Now, we run the above code:

```
Terminal
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~/bin$ gcc
-o vpn_client vpn_client.c -Wall
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~/bin$ sud
o ./vpn_client
```

Then we assign an IP address to the tun0 interface and activate it. IP Address assigned: 192.168.53.1/25

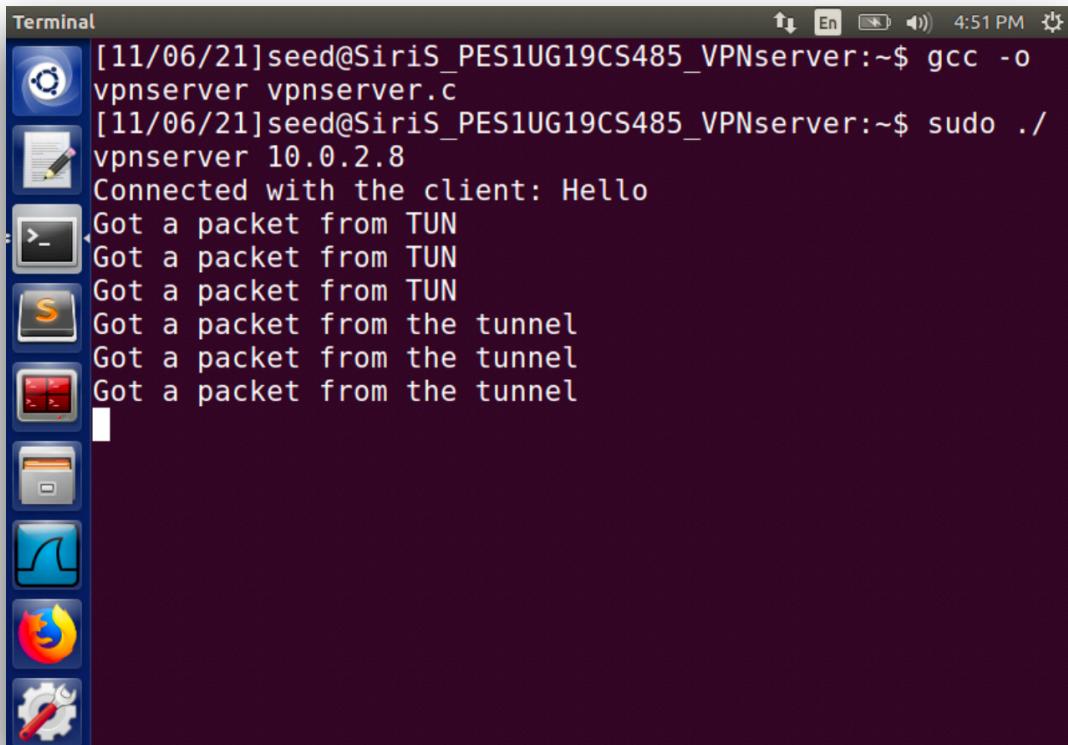
Upon checking ifconfig -a : we have an established tunnel:

```
Terminal [11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ sudo if config tun0 192.168.53.5/24 up
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ ifconfig -a
enp0s3      Link encap:Ethernet HWaddr 08:00:27:6e:29:06
              inet addr:10.0.2.6 Bcast:10.0.2.255 Mask:25
              5.255.255.0
              inet6 addr: fe80::5bc6:b14f:14c7:715a/64 Scop
              e:Link
              Metric:1
              UP BROADCAST RUNNING MULTICAST MTU:1500 Met
              RX packets:117 errors:0 dropped:0 overruns:0
              frame:0
              TX packets:214 errors:0 dropped:0 overruns:0
              carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:23158 (23.1 KB) TX bytes:26146 (26.
              1 KB)
lo          Link encap:Local Loopback
              inet addr:127.0.0.1 Mask:255.0.0.0
```

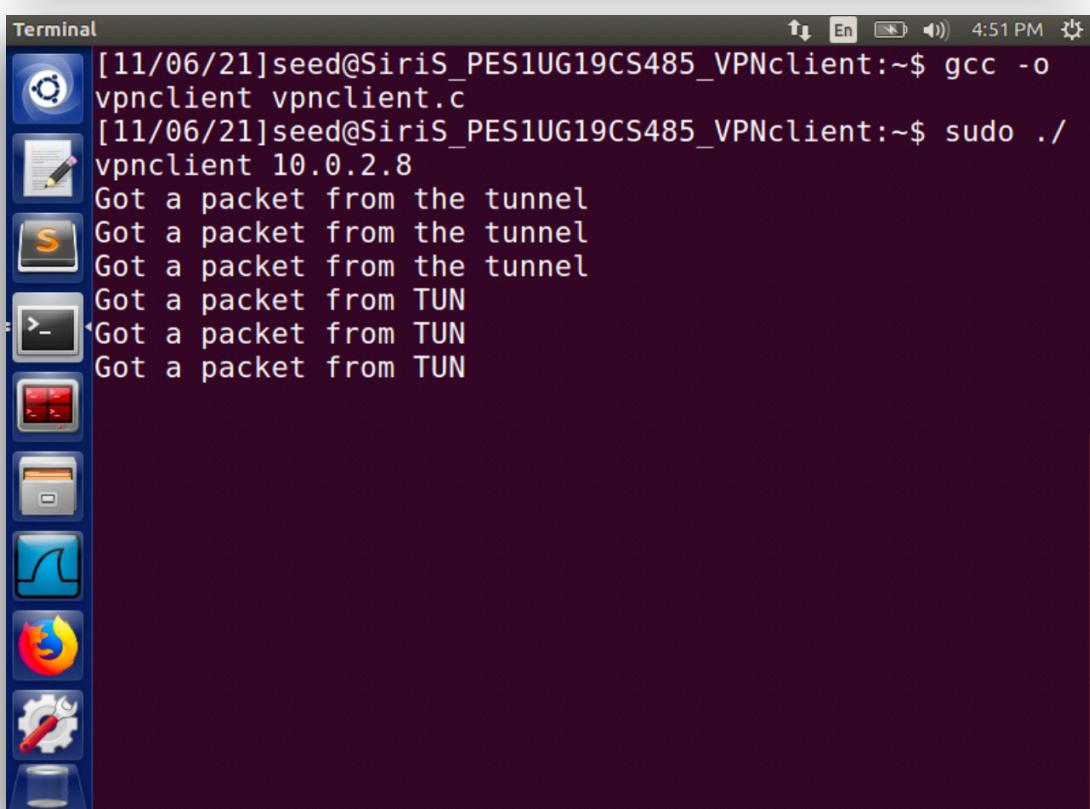
```
Terminal TX packets:384 errors:0 dropped:0 overruns:0
carrier:0
              collisions:0 txqueuelen:1
              RX bytes:38100 (38.1 KB) TX bytes:38100 (38.
              1 KB)
tun0      Link encap:UNSPEC HWaddr 00-00-00-00-00-00-0
0-00-00-00-00-00-00-00
              inet addr:192.168.53.5 P-t-P:192.168.53.5 M
              ask:255.255.255.0
              inet6 addr: fe80::5f0a:14c0:bd5b:105b/64 Scop
              e: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:3 errors:0 dropped:0 overruns:0 ca
              rrier:0
              collisions:0 txqueuelen:500
              RX bytes:0 (0.0 B) TX bytes:144 (144.0 B)
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$
```

Step 3: Set Up Routing on Client and Server VMs

We can now see that the tunnel is established because of the messages displayed like ‘Connected with client: Hello’:



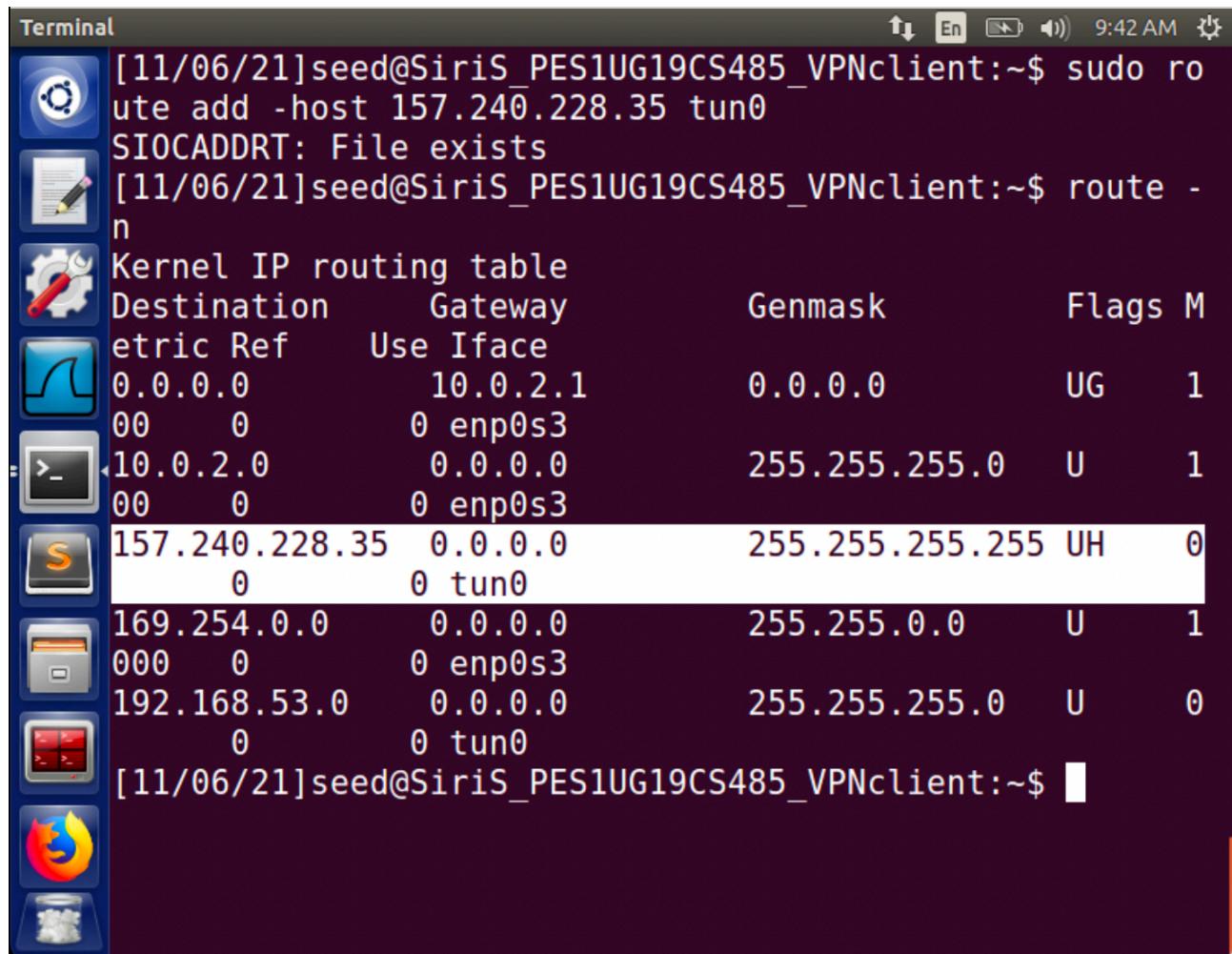
```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$ gcc -o vpnserver vpnserver.c
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~$ sudo ./vpnserver 10.0.2.8
Connected with the client: Hello
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```



```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ gcc -o vpnclient vpnclient.c
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ sudo ./vpnclient 10.0.2.8
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
```

We need to set up routing paths on both client and server machines to direct the intended traffic through the tunnel. This is done on the client machine as follows:

IP Address of Facebook: 157.240.228.35

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal displays the following command and its output:

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ sudo route add -host 157.240.228.35 tun0
SIOCADDRT: File exists
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ route -n
Kernel IP routing table
Destination      Gateway        Genmask        Flags M
Metric Ref      Use Iface
0.0.0.0          10.0.2.1     0.0.0.0       UG    1
00      0          0 enp0s3
10.0.2.0          0.0.0.0      255.255.255.0  U     1
00      0          0 enp0s3
157.240.228.35  0.0.0.0      255.255.255.255 UH    0
          0          0 tun0
169.254.0.0      0.0.0.0      255.255.0.0   U     1
000     0          0 enp0s3
192.168.53.0     0.0.0.0      255.255.255.0   U     0
          0          0 tun0
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$
```

The terminal window is part of a desktop interface with icons for various applications like Dash, Home, and System Monitor visible on the left.

This ensures that all the packets from the IP address 157.240.228.0/24 (facebook's IP) will be routed to tun0 interface.

Step 4: Set Up NAT on Server VM

We make NAT to believe that the MAC address of 192.168.53.5 is the VPN server's MAC address. The following commands can enable the NAT on the Server VM:

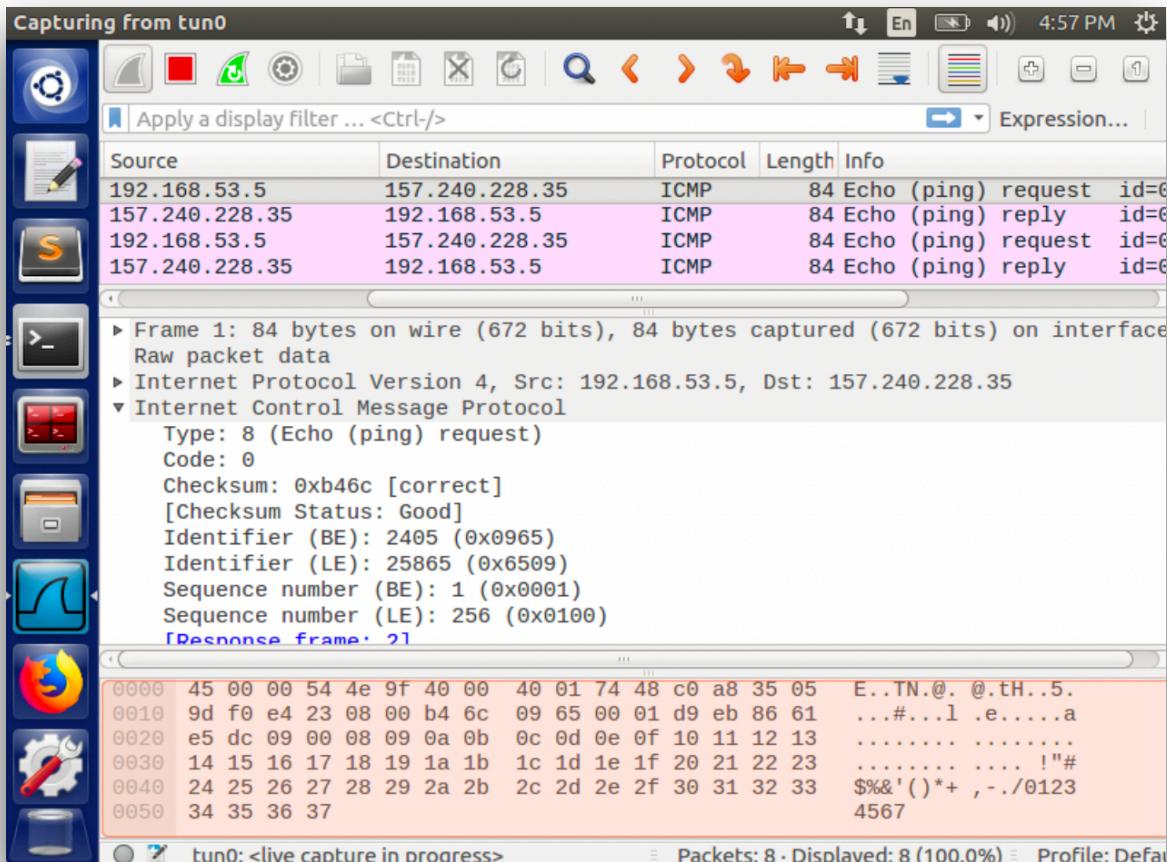
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~\$ sudo ip
tables -F
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~\$ sudo ip
tables -t nat -F
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~\$ sudo ip
tables -t nat -A POSTROUTING -j MASQUERADE -o enp0s3
[11/06/21]seed@SiriS_PES1UG19CS485_VPNserver:~\$ █

Task 4: Demonstration

Client machine can now access facebook.com through the tunnel established. We can try ping command. Therefore the task has been completed successfully.

```
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$ ping ww  
w.facebook.com  
PING star-mini.c10r.facebook.com (157.240.228.35) 56(84)  
bytes of data.  
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=1 ttl=53 time=10.6 ms  
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=2 ttl=53 time=10.6 ms  
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=3 ttl=53 time=21.0 ms  
64 bytes from edge-star-mini-shv-01-tir2.facebook.com (157.240.228.35): icmp_seq=4 ttl=53 time=77.7 ms  
^C  
--- star-mini.c10r.facebook.com ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
3892ms  
rtt min/avg/max/mdev = 10.686/30.038/77.722/27.854 ms  
[11/06/21]seed@SiriS_PES1UG19CS485_VPNclient:~$
```

Wireshark screenshot that explains the route traversed by the packet:



We can observe that the ICMP request packet from tunnel interface (192.168.53.5) is created to facebook's IP(157.240.228.35), the tunnel writes the packet to the UDP socket which sends the packet to server machine.(10.0.2.5). The ping reply is received back on tun0 interface.Hence, ping works bypassing firewall through the created tunnel.

