The Laboratory of Computer Networks Security (UE19CS326)

Documented by Anurag.R.Simha

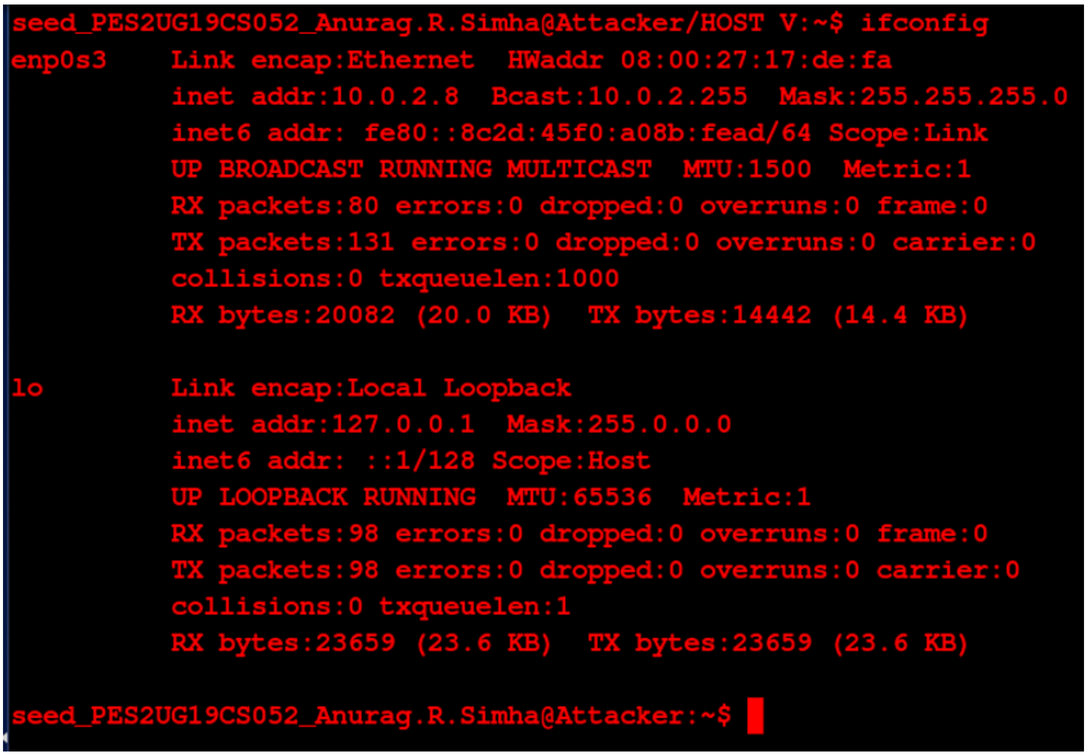
SRN	:	PES2UG19CS052
Name	:	Anurag.R.Simha
Date	:	13/11/2021
Section	:	A
Week	:	6

The Table of Contents

The Setup

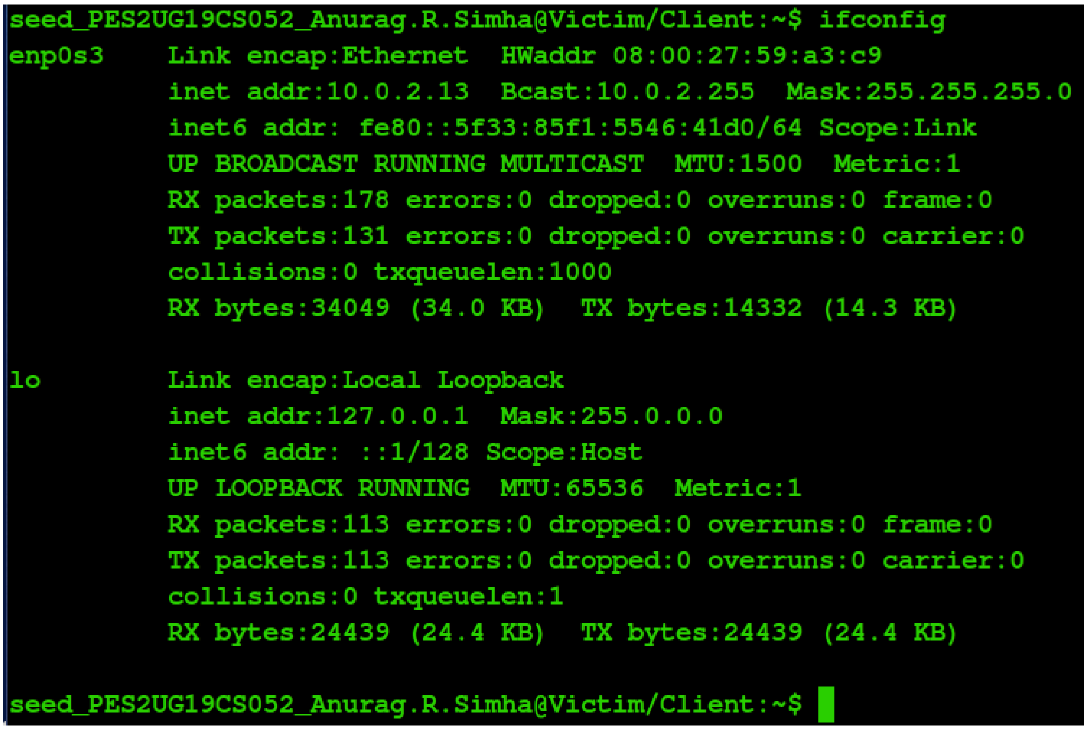
For the experimentation of various attacks, three virtual machines were employed.

1. The host machine (10.0.2.8)



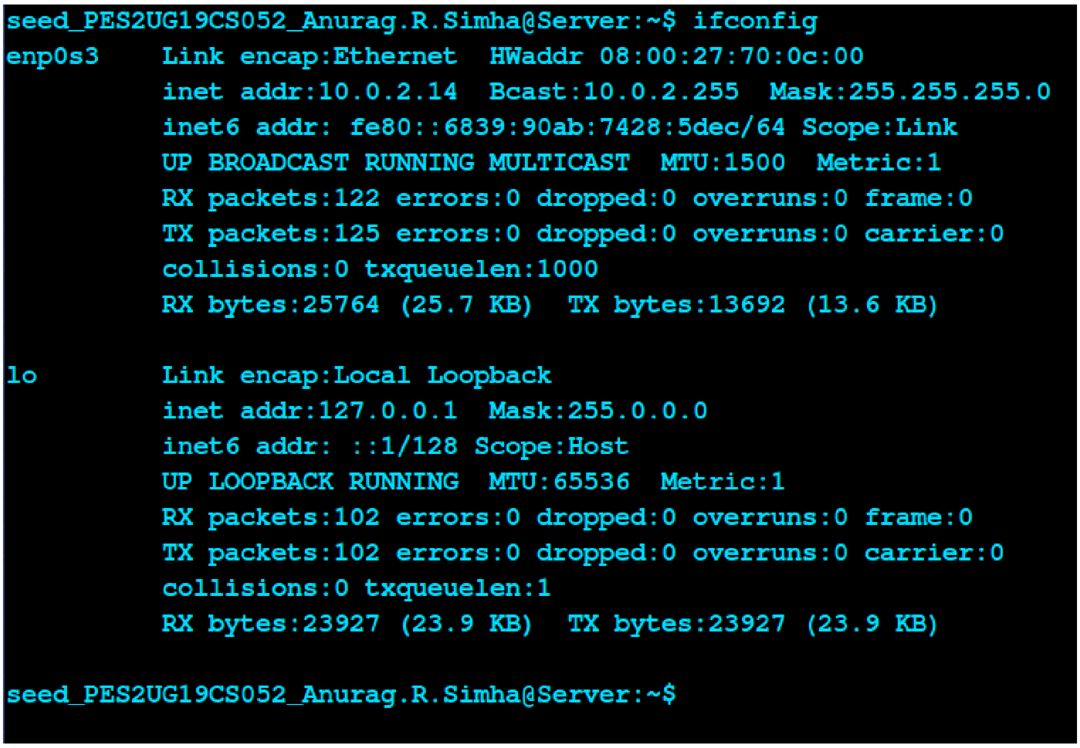
```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:17:de:fa
          inet addr:10.0.2.8  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::8c2d:45f0:a08b:fead/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20082 (20.0 KB)  TX bytes:14442 (14.4 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
          RX packets:98 errors:0 dropped:0 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:23659 (23.6 KB)  TX bytes:23659 (23.6 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Attacker:~$
```

2. The VPN Client machine (10.0.2.13)



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:59:a3:c9
          inet addr:10.0.2.13  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::5f33:85f1:5546:41d0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:178 errors:0 dropped:0 overruns:0 frame:0
          TX packets:131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34049 (34.0 KB)  TX bytes:14332 (14.3 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:113 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:24439 (24.4 KB)  TX bytes:24439 (24.4 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$
```

3. The VPN Server machine (10.0.2.14)



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:70:0c:00
          inet addr:10.0.2.14  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6839:90ab:7428:5dec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:122 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:25764 (25.7 KB)  TX bytes:13692 (13.6 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:102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:23927 (23.9 KB)  TX bytes:23927 (23.9 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Server:~$
```

Task 1: VM Setup

The Setup is performed in conformity with the diagram below.

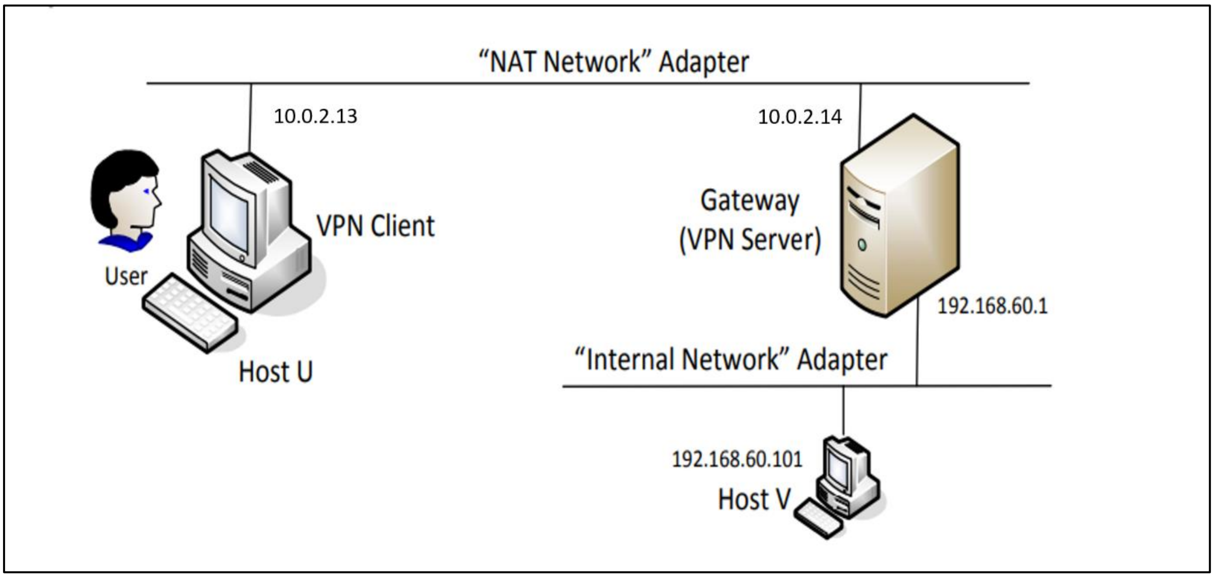
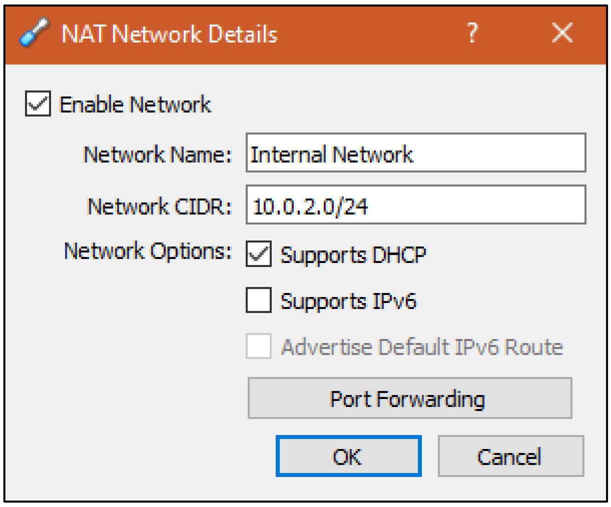


Fig. 1(a): The VM Setup

The following steps are abided:

1. The network, 'Internal Network' is created following the path, File → Preferences → Network → .



NAT Network Details

☑ Enable Network

Network Name: Internal Network

Network CIDR: 10.0.2.0/24

Network Options: ☑ Supports DHCP

□ Supports IPv6

□ Advertise Default IPv6 Route

Fig. 1(b): Setup of the 'Internal Network'.

NOTICE:

Here, the connections are configured according to the scheme below:

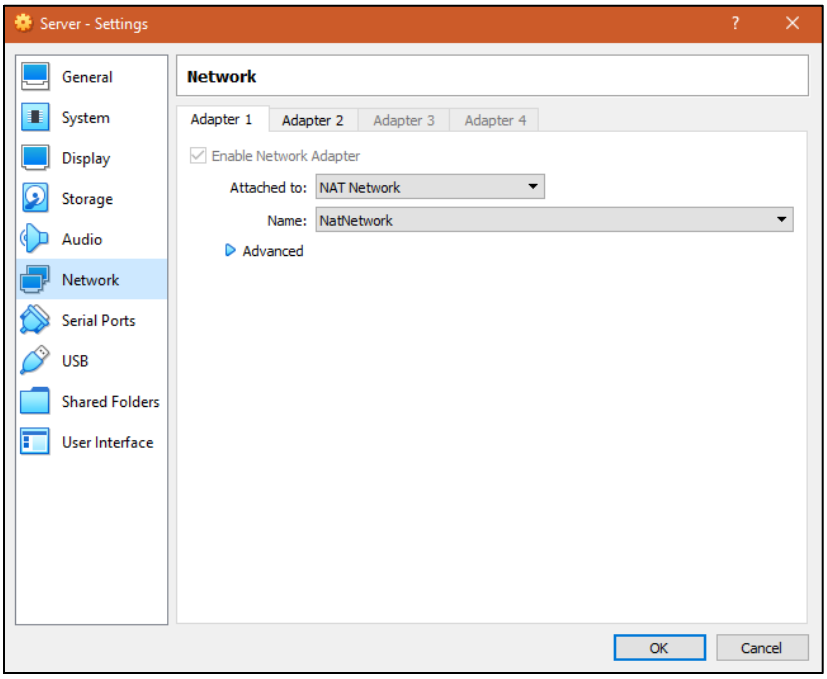
VPN client (10.0.2.13) – Adapter 1 (10.0.2.13) – NAT Network

VPN Server (10.0.2.14) – Adapter 1 (10.0.2.14) – NAT network, Adapter 2 (192.168.60.1) – Internal Network

HOST V (10.0.2.8) – Adapter 1 (192.168.60.101) – Internal Network

2. The network adapters are configured.

Server machine (10.0.2.14 ↔ 192.168.60.1)



Server - Settings

Network

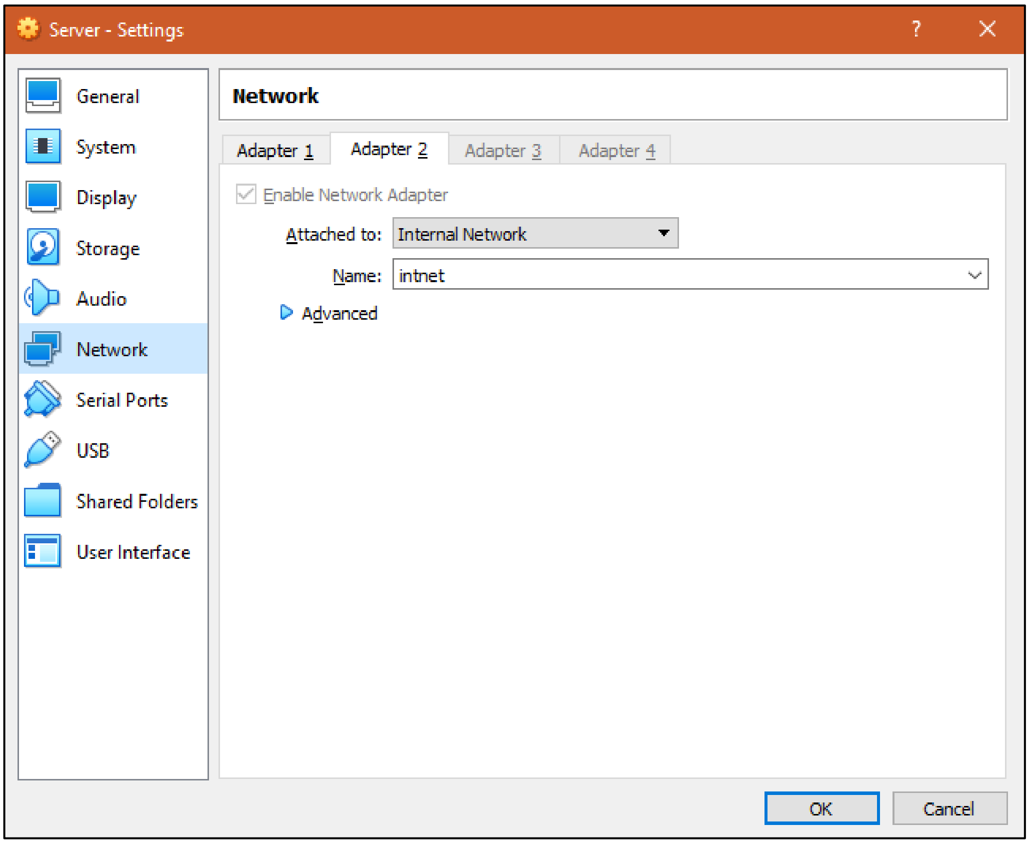
Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

Fig. 1(c): The first adapter on the server machine is the NAT Network.



Server - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

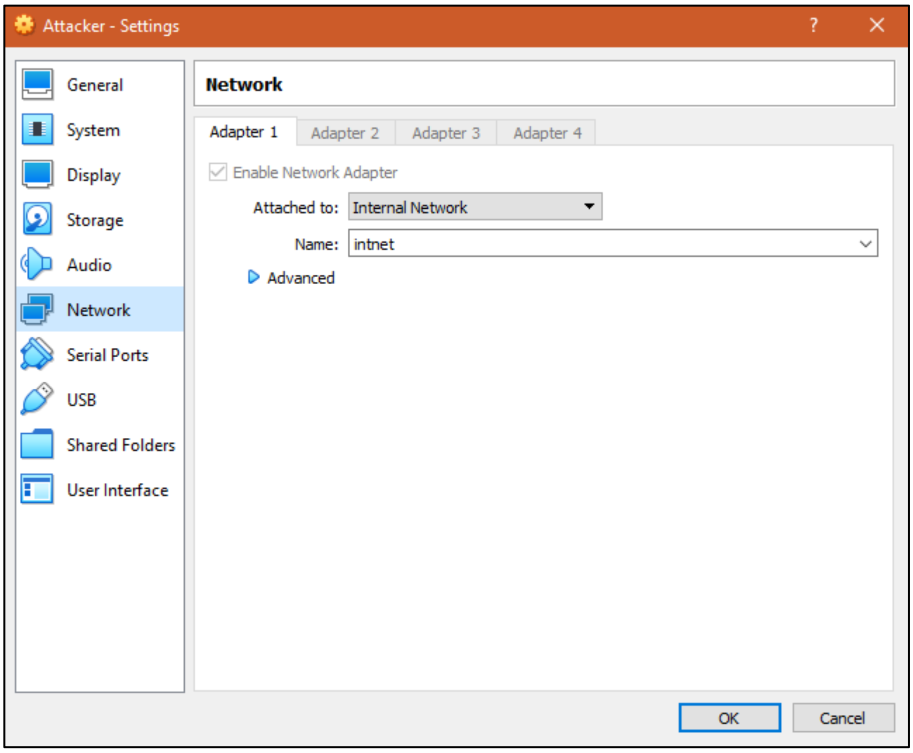
Attached to: Internal Network

Name: intnet

▷ Advanced

Fig. 1(d): The second adapter on the server machine is the Internal Network.

Host V machine (192.168.60.101):



Attacker - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Internal Network

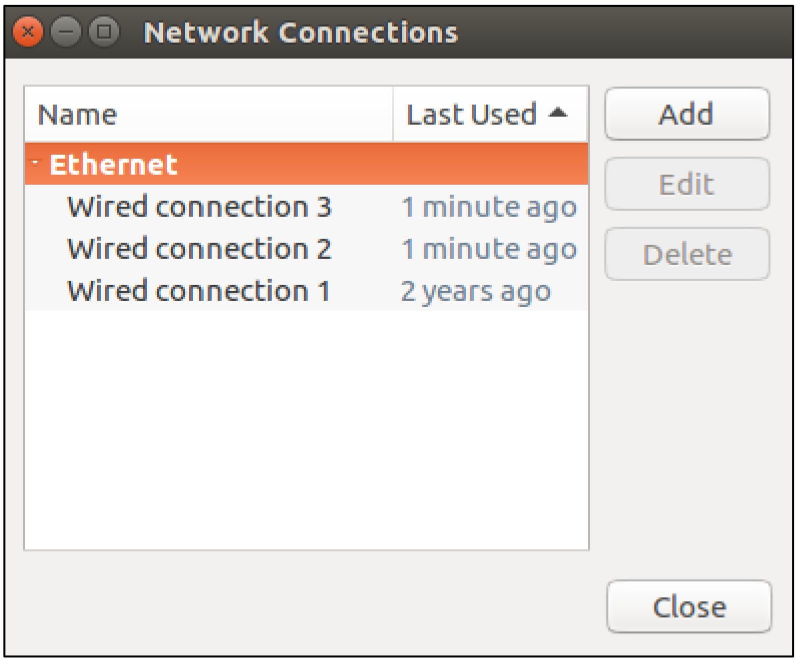
Name: intnet

▷ Advanced

Fig. 1(e): The only network adapter of the host machine is configured to the ‘Internal Network’.

3. Configuring the connections (under the edit section) and examining them.

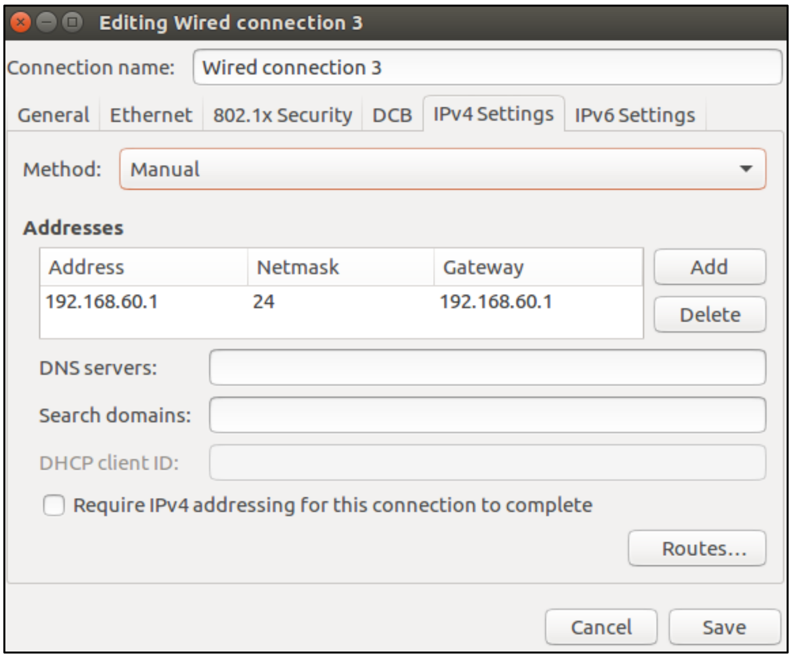
On the server machine:



Name	Last Used
Ethernet	
Wired connection 3	1 minute ago
Wired connection 2	1 minute ago
Wired connection 1	2 years ago

Fig. 1(f): The wired connections.

Of the three networks, ‘Wired connection 3’ is responsible to get a connection established to the internal network, and ‘Wired connection 2’ to the NAT network. They are then configured.



Connection name: Wired connection 3

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.60.1	24	192.168.60.1

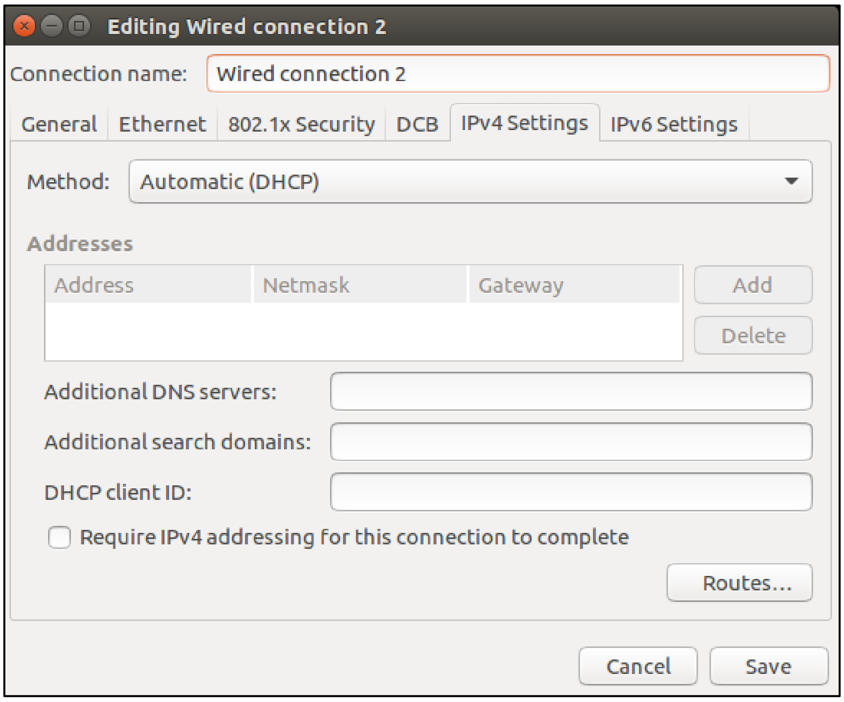
DNS servers:

Search domains:

DHCP client ID:

□ Require IPv4 addressing for this connection to complete

Fig. 1(g): The address and gateway are configured to the internal network (Wired Connection 3).



Editing Wired connection 2

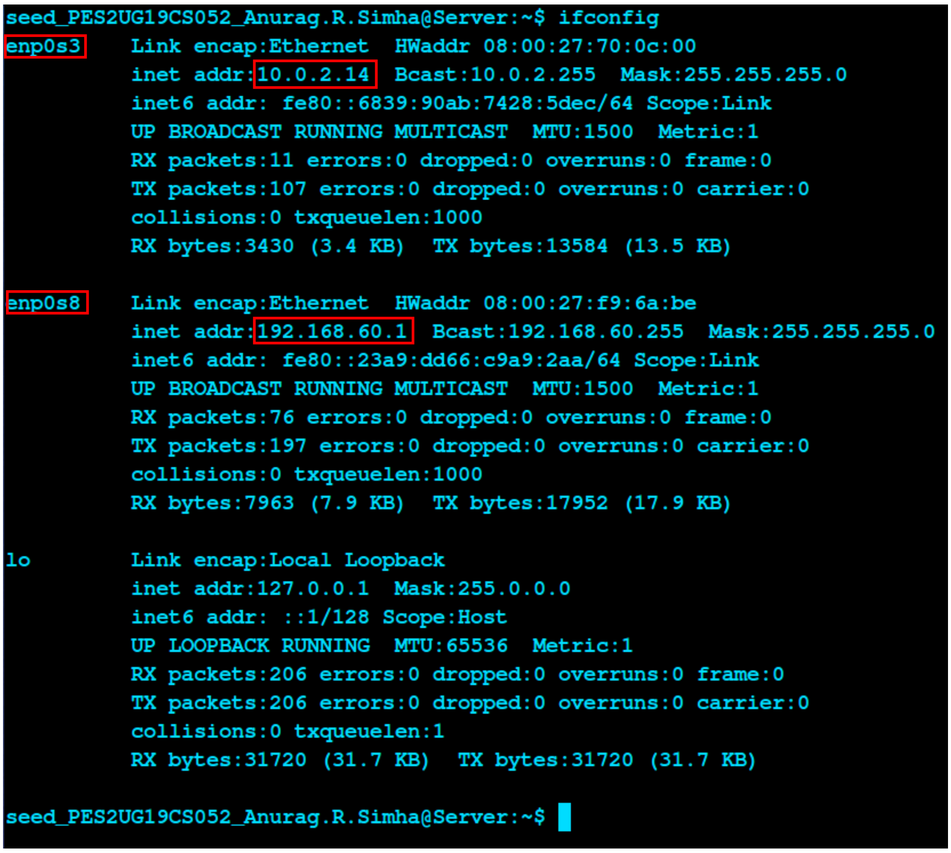
Connection name: Wired connection 2

Method: Automatic (DHCP)

☐ Require IPv4 addressing for this connection to complete

Fig. 1(h): The address and gateway are automatically configured to the NAT network (Wired Connection 2).

For inspection’s sake, the command ‘ifconfig’ is used.

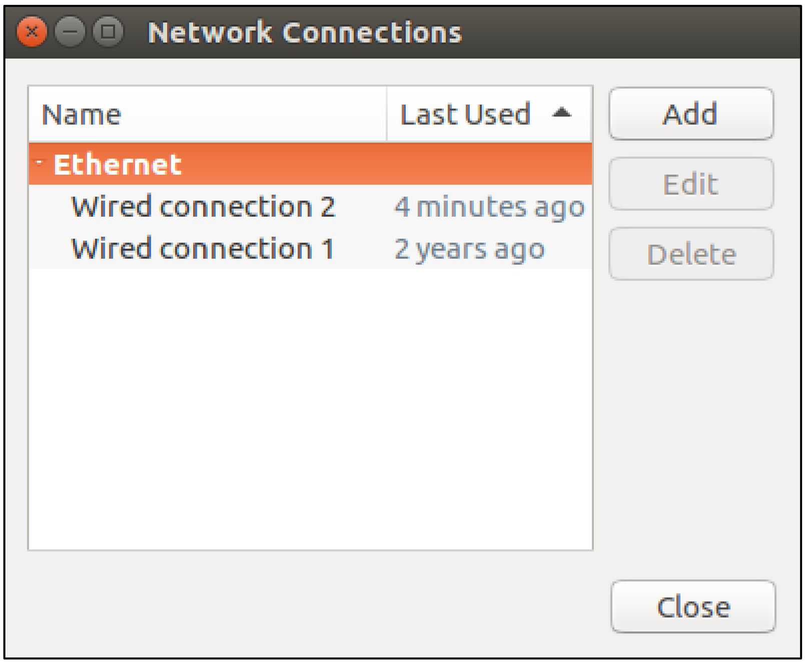


```
seed_PES2UG19CS052_Anurag.R.Simha@Server:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:70:0c:00
          inet addr:10.0.2.14  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6839:90ab:7428:5dec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:107 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3430 (3.4 KB)  TX bytes:13584 (13.5 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:f9:6a:be
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::23a9:dd66:c9a9:2aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:76 errors:0 dropped:0 overruns:0 frame:0
          TX packets:197 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7963 (7.9 KB)  TX bytes:17952 (17.9 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:206 errors:0 dropped:0 overruns:0 frame:0
          TX packets:206 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:31720 (31.7 KB)  TX bytes:31720 (31.7 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Server:~$
```

Fig. 1(i): Testing the triumph of the connections.

Henceforth, the test yields a triumphant result in the connections.

On the host/isolated machine:

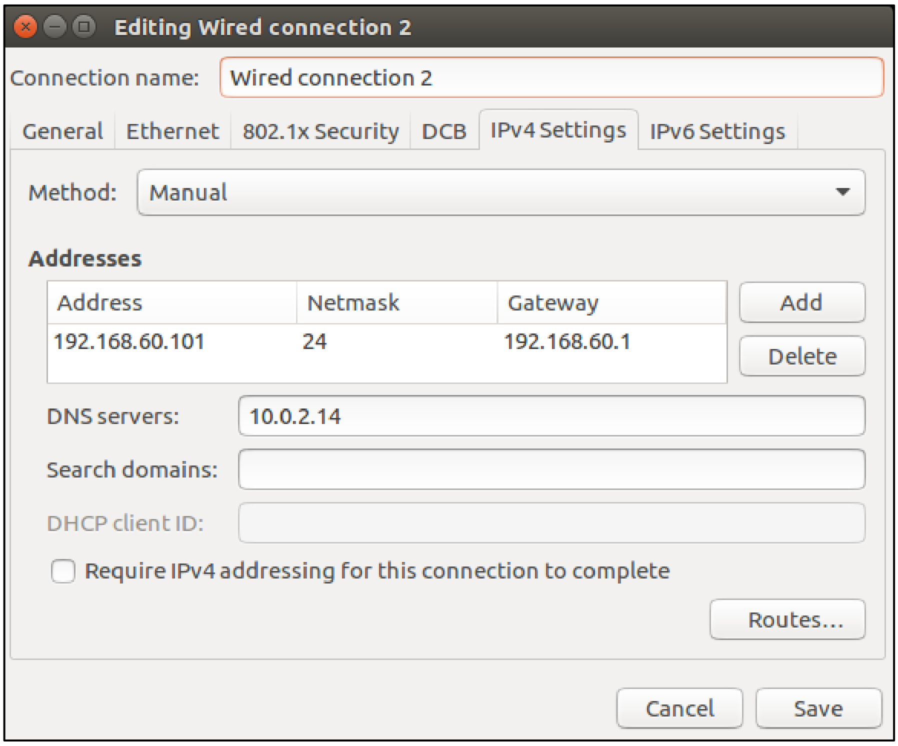


Network Connections

Name	Last Used
Ethernet	
Wired connection 2	4 minutes ago
Wired connection 1	2 years ago

Fig. 1(j): The wired connections on the isolated machine.

The second wired connection is where the eyes are upon.



Editing Wired connection 2

Connection name: Wired connection 2

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.60.101	24	192.168.60.1

DNS servers: 10.0.2.14

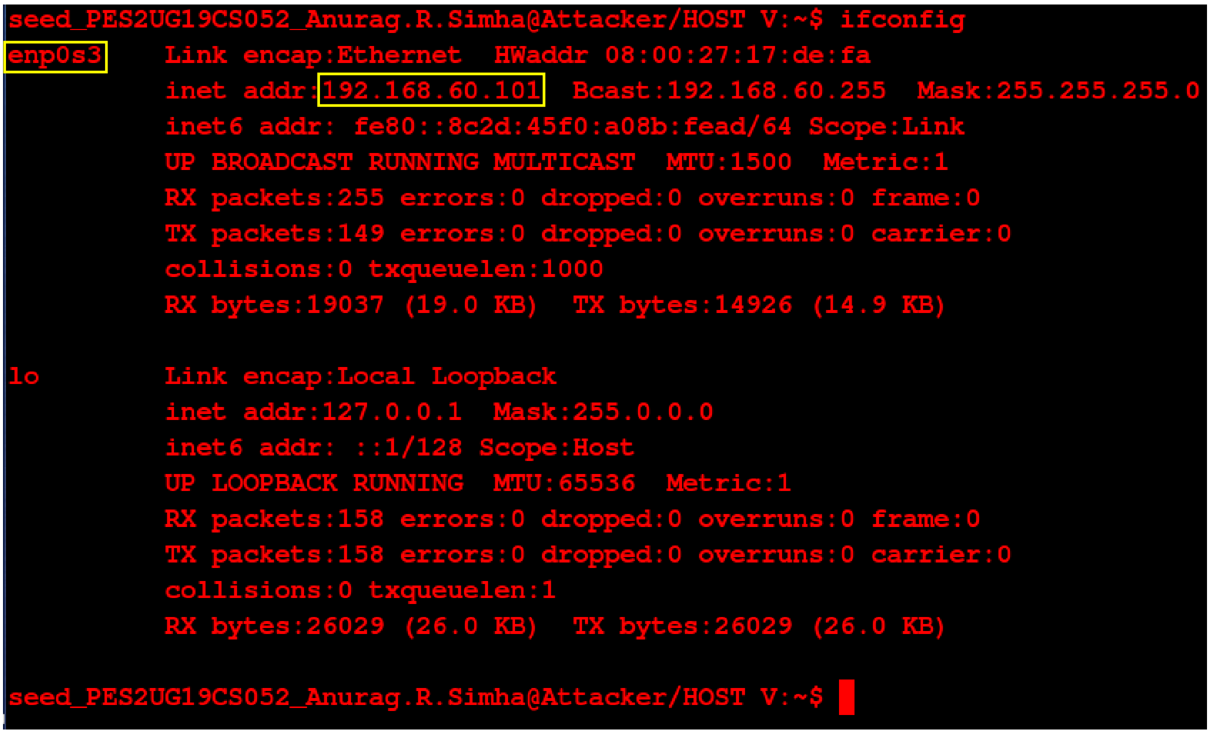
Search domains:

DHCP client ID:

□ Require IPv4 addressing for this connection to complete

Fig. 1(k): The address and gateway are configured to the internal network.

For inspection’s sake, the command ‘ifconfig’ is used.

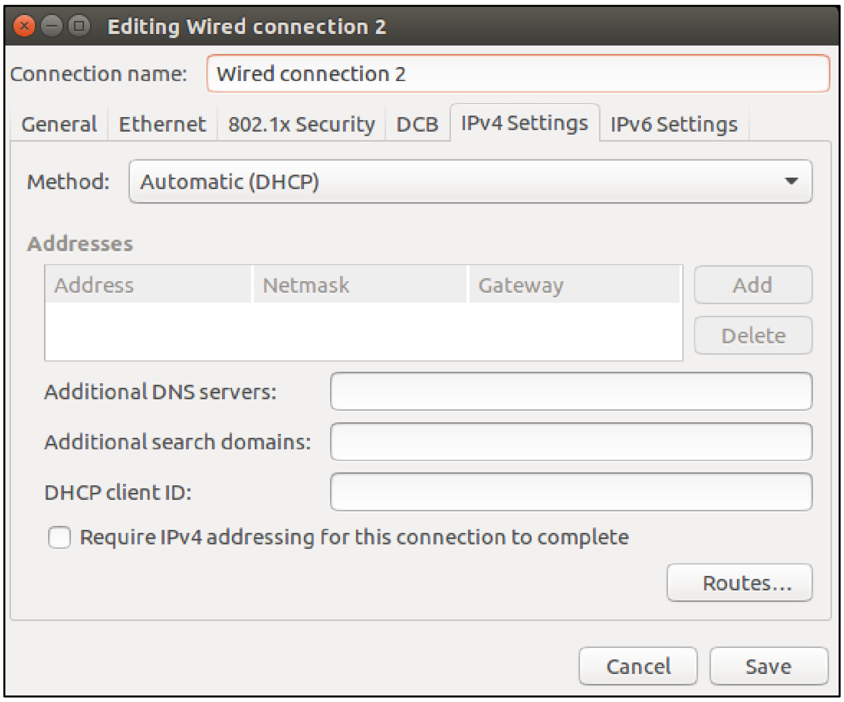


```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:17:de:fa
          inet addr:192.168.60.101  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::8c2d:45f0:a08b:fead/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:255 errors:0 dropped:0 overruns:0 frame:0
          TX packets:149 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19037 (19.0 KB)  TX bytes:14926 (14.9 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:158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:26029 (26.0 KB)  TX bytes:26029 (26.0 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$
```

Fig. 1(*l*): Testing the triumph of the connections.

Henceforth, the test yields a triumphant result in the connections.

On the client machine, the IP address is set by DHCP (Automatic):



Editing Wired connection 2

Connection name: Wired connection 2

Method: Automatic (DHCP)

Addresses

Additional DNS servers:

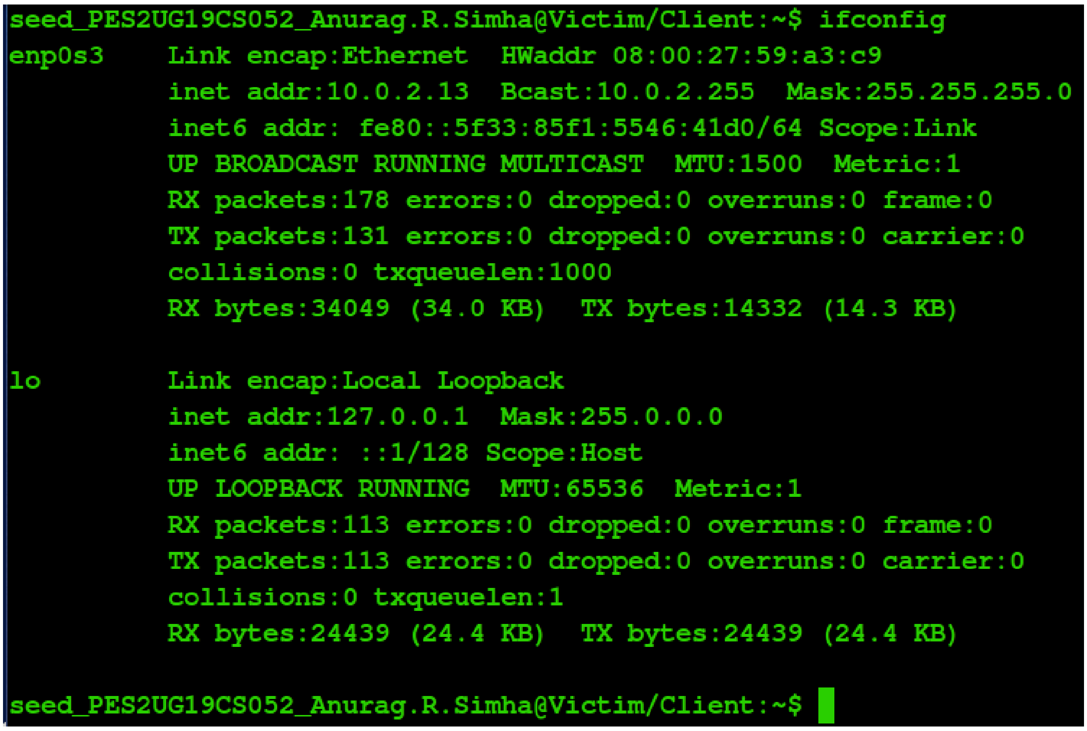
Additional search domains:

DHCP client ID:

□ Require IPv4 addressing for this connection to complete

Fig. 1(m): The address and gateway are automatically configured by the DHCP (Wired Connection 2).

For inspection's sake, the command 'ifconfig' is used.



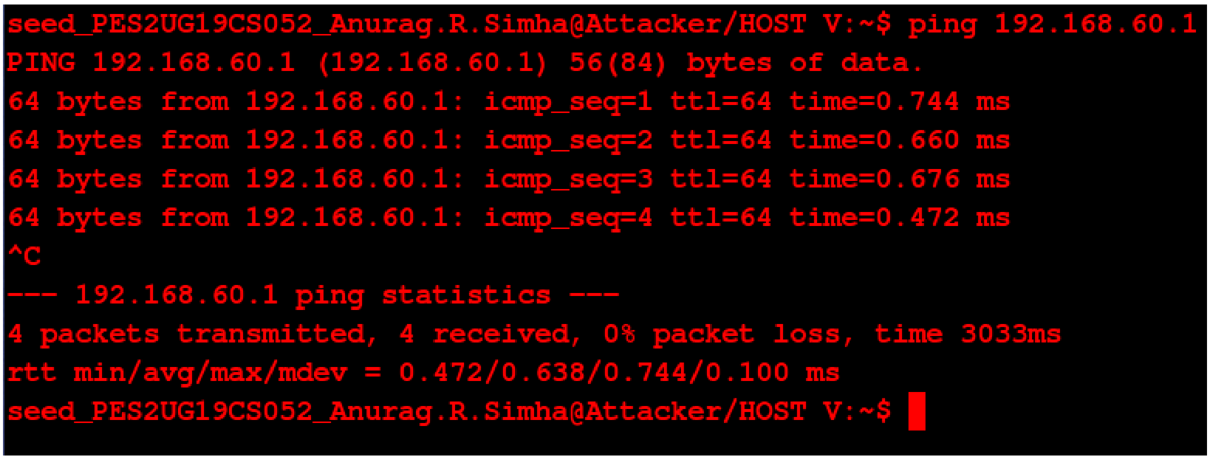
```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:59:a3:c9
          inet addr:10.0.2.13  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::5f33:85f1:5546:41d0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:178 errors:0 dropped:0 overruns:0 frame:0
          TX packets:131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34049 (34.0 KB)  TX bytes:14332 (14.3 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:113 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:24439 (24.4 KB)  TX bytes:24439 (24.4 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$
```

Fig. 1(n): Testing the triumph of the connections.

Henceforth, the test yields a triumphant result in the connections.

4. Testing the reachability of the machines.

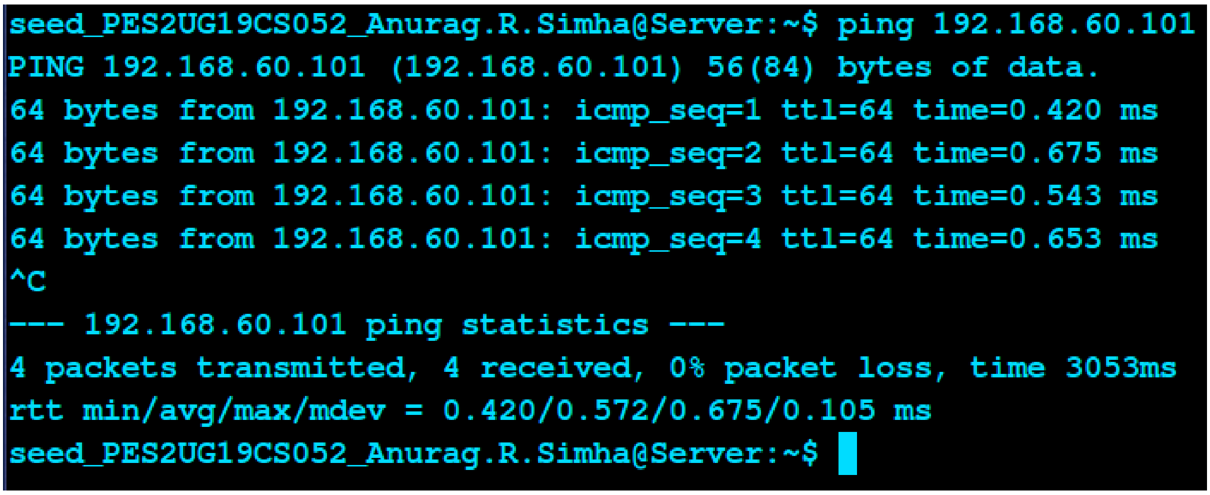
Host V → VPN Server (Expected result: Pass)



```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ping 192.168.60.1
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
64 bytes from 192.168.60.1: icmp_seq=1 ttl=64 time=0.744 ms
64 bytes from 192.168.60.1: icmp_seq=2 ttl=64 time=0.660 ms
64 bytes from 192.168.60.1: icmp_seq=3 ttl=64 time=0.676 ms
64 bytes from 192.168.60.1: icmp_seq=4 ttl=64 time=0.472 ms
^C
--- 192.168.60.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3033ms
rtt min/avg/max/mdev = 0.472/0.638/0.744/0.100 ms
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$
```

Fig. 1(o): A successful ping between the host and server machine.

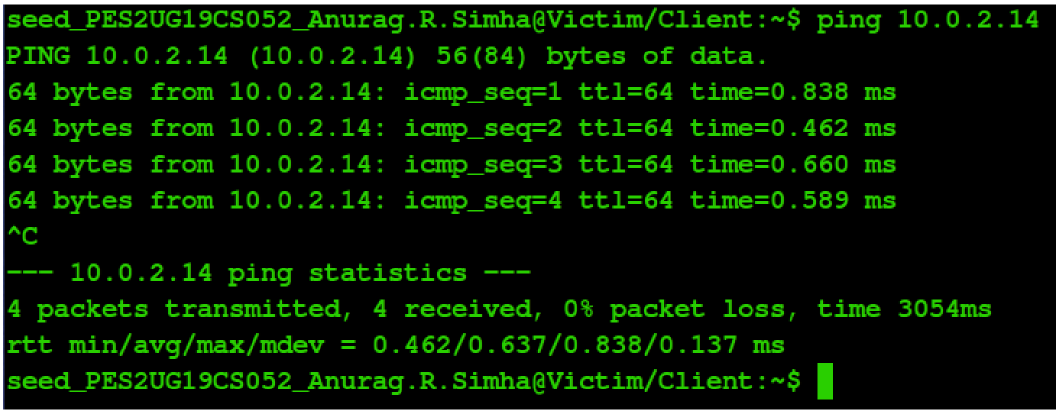
VPN Server → Host V (Expected result: Pass)



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:~$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=64 time=0.420 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=64 time=0.675 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=64 time=0.543 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=64 time=0.653 ms
^C
--- 192.168.60.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.420/0.572/0.675/0.105 ms
seed_PES2UG19CS052_Anurag.R.Simha@Server:~$
```

Fig. 1(p): A successful ping between the server and host machine.

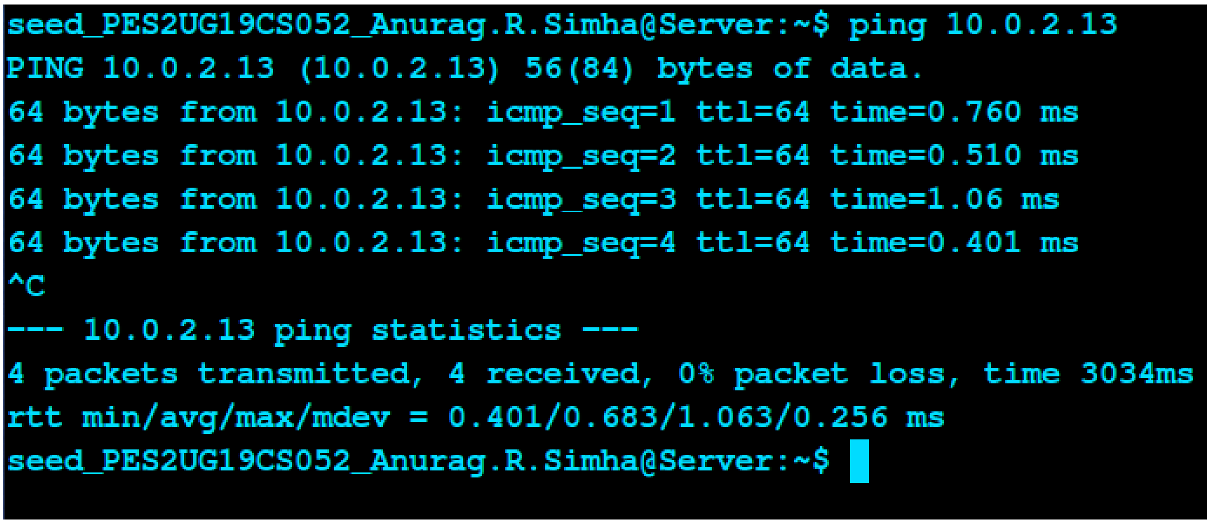
VPN Client → VPN Server (Expected result on enp0s3: Pass)



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ ping 10.0.2.14
PING 10.0.2.14 (10.0.2.14) 56(84) bytes of data.
64 bytes from 10.0.2.14: icmp_seq=1 ttl=64 time=0.838 ms
64 bytes from 10.0.2.14: icmp_seq=2 ttl=64 time=0.462 ms
64 bytes from 10.0.2.14: icmp_seq=3 ttl=64 time=0.660 ms
64 bytes from 10.0.2.14: icmp_seq=4 ttl=64 time=0.589 ms
^C
--- 10.0.2.14 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.462/0.637/0.838/0.137 ms
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$
```

Fig. 1(q): A successful ping between the client and server machine.

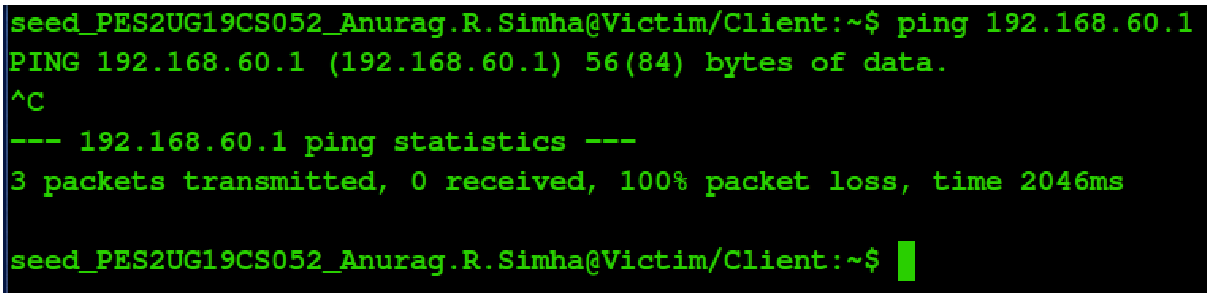
VPN Server → VPN Client (Expected result on enp0s3: Pass)



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:~$ ping 10.0.2.13
PING 10.0.2.13 (10.0.2.13) 56(84) bytes of data.
64 bytes from 10.0.2.13: icmp_seq=1 ttl=64 time=0.760 ms
64 bytes from 10.0.2.13: icmp_seq=2 ttl=64 time=0.510 ms
64 bytes from 10.0.2.13: icmp_seq=3 ttl=64 time=1.06 ms
64 bytes from 10.0.2.13: icmp_seq=4 ttl=64 time=0.401 ms
^C
--- 10.0.2.13 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.401/0.683/1.063/0.256 ms
seed_PES2UG19CS052_Anurag.R.Simha@Server:~$
```

Fig. 1(r): A successful ping between the server and client machine.

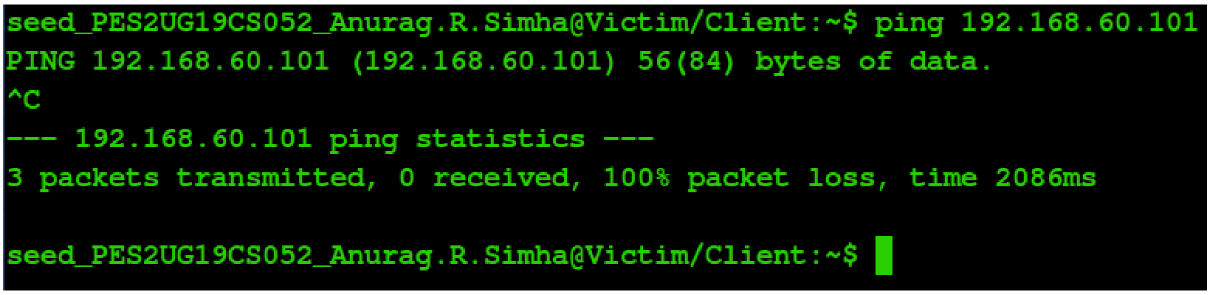
VPN Client → VPN Server (Expected result on enp0s8: Fail)



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ ping 192.168.60.1
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
^C
--- 192.168.60.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2046ms

seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ 
```

Fig. 1(s): The connection failed.

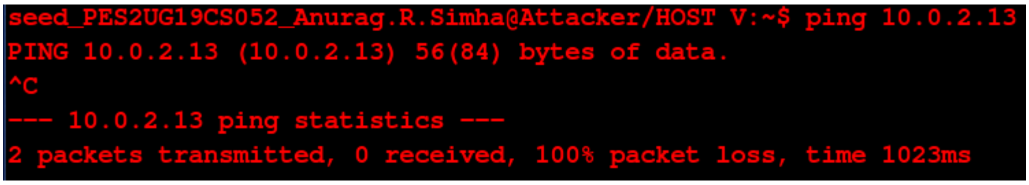
VPN Client → Host V (Expected result: Fail)



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
^C
--- 192.168.60.101 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2086ms

seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:~$ 
```

Fig. 1(t): The connection fails.

Host V → VPN Client (Expected result: Fail)



```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ping 10.0.2.13
PING 10.0.2.13 (10.0.2.13) 56(84) bytes of data.
^C
--- 10.0.2.13 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1023ms
```

Fig. 1(u): The connection fails.

Task 2: Creating a VPN Tunnel using TUN/TAP

The vpnclient and vpnserver programmes are the two ends of a VPN tunnel. They communicate with each other using either TCP or UDP via the sockets depicted in Figure 2(a). In the sample code, it’s chosen to use UDP for the sake of simplicity. The dotted line between the client and server depicts the path for the VPN tunnel. The VPN client and server programmes connect to the hosting system via a TUN interface, through which they do two things:

1. Get IP packets from the hosting system, so the packets can be sent through the tunnel.

2. Get IP packets from the tunnel, and then forward it to the hosting system, which will forward the packet to its final destination.

Below is the procedure to create a VPN tunnel using the vpnclient and vpnserver programmes.

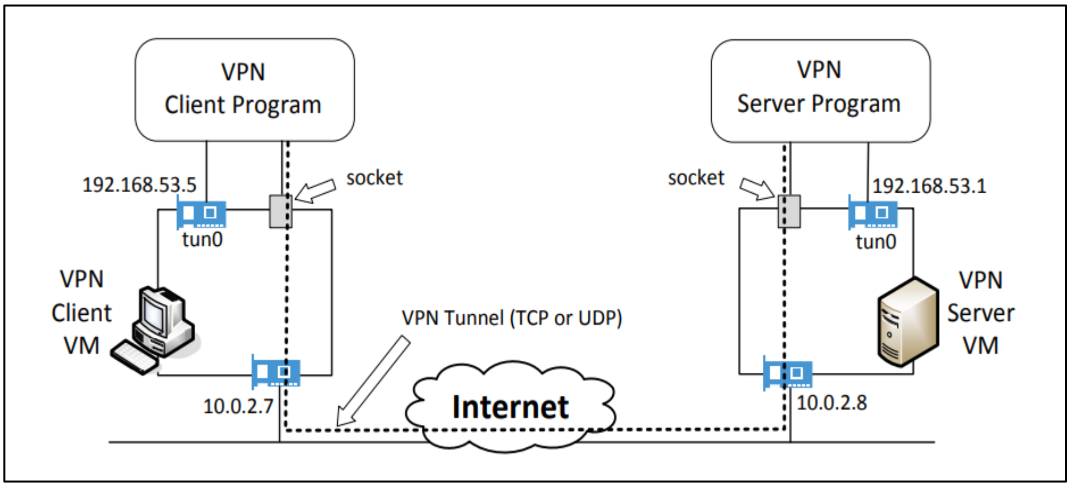
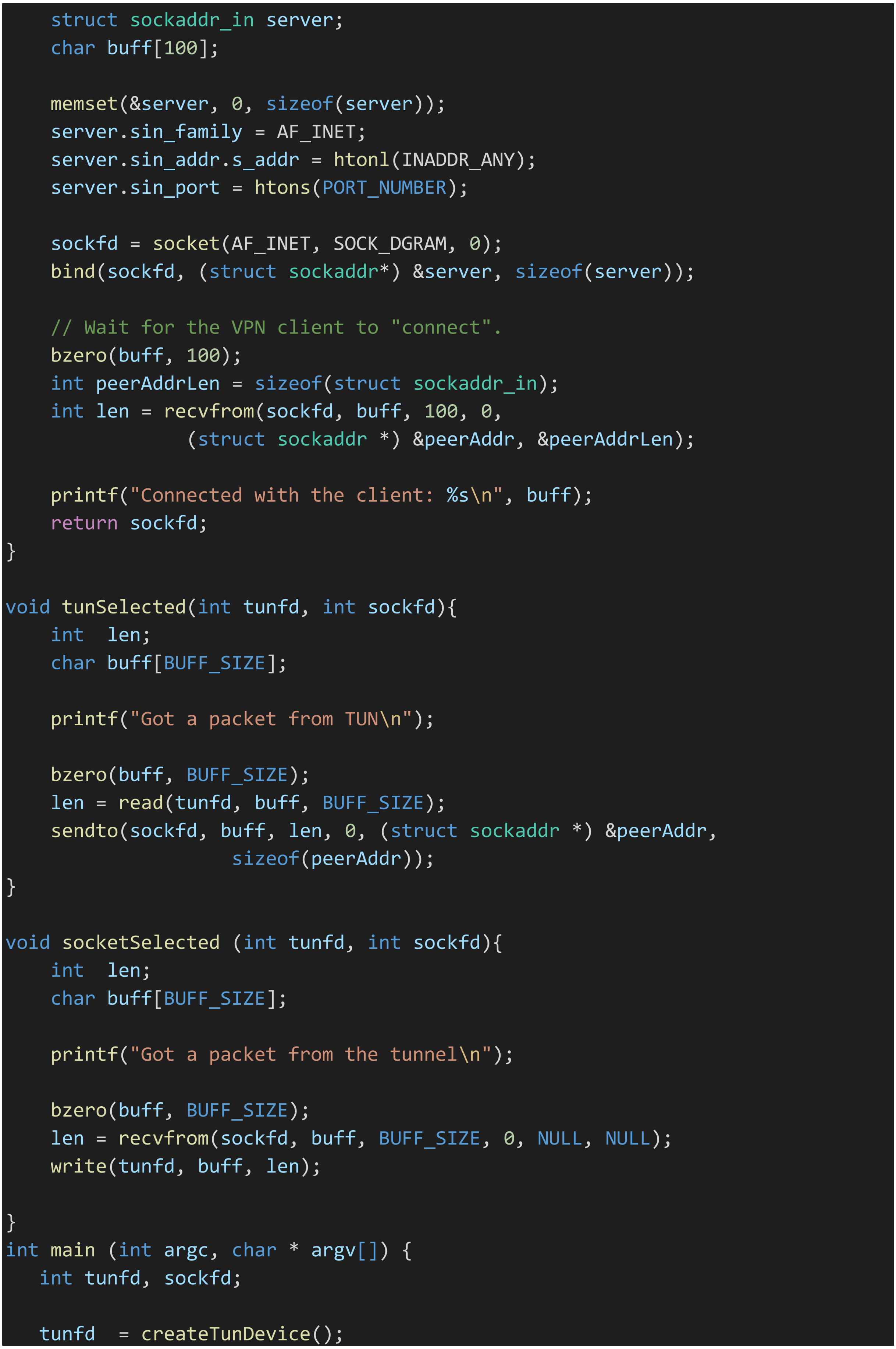


Fig. 2(a): VPN Client and Server.

Step 1: Running the VPN server and setting it's IP address of the interface – (On the VPN Server VM)

The programme:

```
#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));

   ifr.ifr_flags = IFF_TUN | IFF_NO_PI;

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

   return tunfd;
}

int initUDPServer() {
    int sockfd;
```

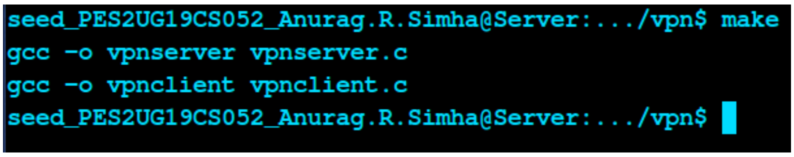


```
    struct sockaddr_in server;
    char buff[100];

    memset(&server, 0, sizeof(server));
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl(INADDR_ANY);
    server.sin_port = htons(PORT_NUMBER);

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    bind(sockfd, (struct sockaddr*) &server, sizeof(server));

    // Wait for the VPN client to "connect".
    bzero(buff, 100);
    int peerAddrLen = sizeof(struct sockaddr_in);
    int len = recvfrom(sockfd, buff, 100, 0,
                (struct sockaddr *) &peerAddr, &peerAddrLen);

    printf("Connected with the client: %s\n", buff);
    return sockfd;
}

void tunSelected(int tunfd, int sockfd){
    int  len;
    char buff[BUFF_SIZE];

    printf("Got a packet from TUN\n");

    bzero(buff, BUFF_SIZE);
    len = read(tunfd, buff, BUFF_SIZE);
    sendto(sockfd, buff, len, 0, (struct sockaddr *) &peerAddr,
                    sizeof(peerAddr));
}

void socketSelected (int tunfd, int sockfd){
    int  len;
    char buff[BUFF_SIZE];

    printf("Got a packet from the tunnel\n");

    bzero(buff, BUFF_SIZE);
    len = recvfrom(sockfd, buff, BUFF_SIZE, 0, NULL, NULL);
    write(tunfd, buff, len);

}
int main (int argc, char * argv[]) {
   int tunfd, sockfd;

   tunfd  = createTunDevice();
   sockfd = initUDPServer();

   // Enter the main loop
   while (1) {
     fd_set readFDSet;

     FD_ZERO(&readFDSet);
     FD_SET(sockfd, &readFDSet);
     FD_SET(tunfd, &readFDSet);
     select(FD_SETSIZE, &readFDSet, NULL, NULL, NULL);

     if (FD_ISSET(tunfd,  &readFDSet)) tunSelected(tunfd, sockfd);
     if (FD_ISSET(sockfd, &readFDSet)) socketSelected(tunfd, sockfd);
  }
}
```

In the programme, the VPN server is configured where a port is opened with the aid of a socket, awaiting a connection from the client.

The commands:

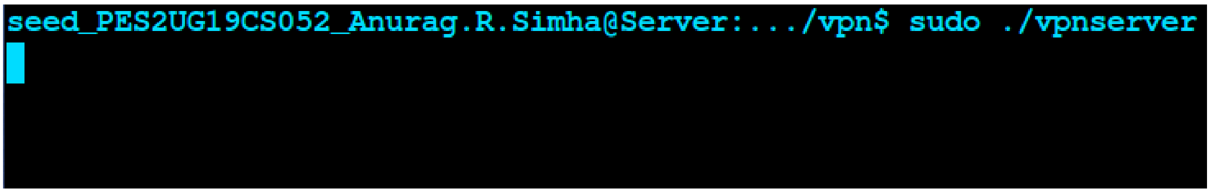
`make`



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$
```

Fig. 2(b): Generating the executables.

`sudo ./vpnserver`



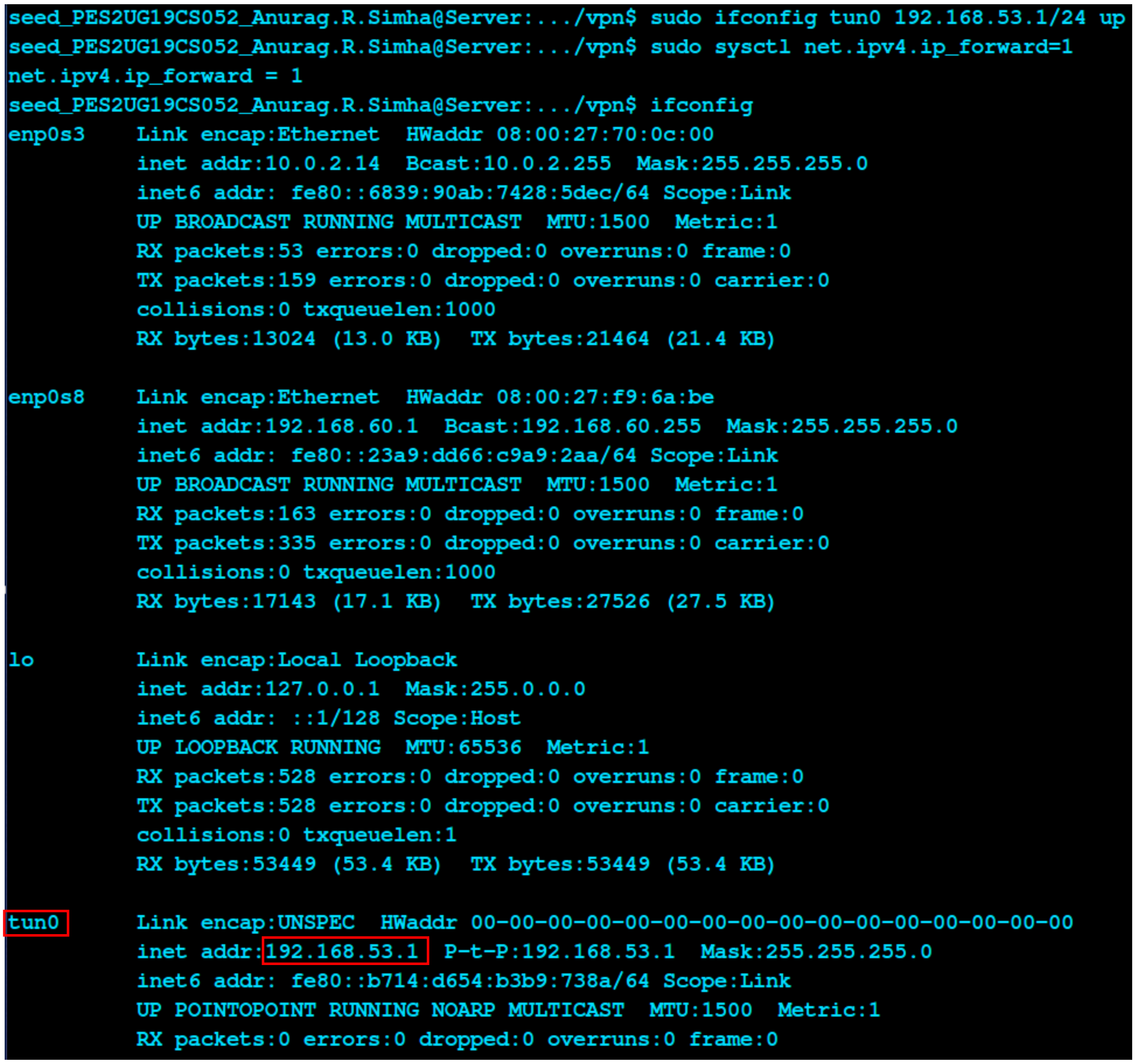
```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo ./vpnserver
```

Fig. 2(c): Instigating the VPN server programme.

Another terminal is opened and the command below is run:

```
sudo ifconfig tun0 192.168.53.1/24 up
sudo sysctl net.ipv4.ip_forward=1
```

After performing this, the command, `ifconfig` is put into effect.



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo ifconfig tun0 192.168.53.1/24 up
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:70:0c:00
          inet addr:10.0.2.14  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6839:90ab:7428:5dec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:53 errors:0 dropped:0 overruns:0 frame:0
          TX packets:159 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13024 (13.0 KB)  TX bytes:21464 (21.4 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:f9:6a:be
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::23a9:dd66:c9a9:2aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:163 errors:0 dropped:0 overruns:0 frame:0
          TX packets:335 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17143 (17.1 KB)  TX bytes:27526 (27.5 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:528 errors:0 dropped:0 overruns:0 frame:0
          TX packets:528 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:53449 (53.4 KB)  TX bytes:53449 (53.4 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::b714:d654:b3b9:738a/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

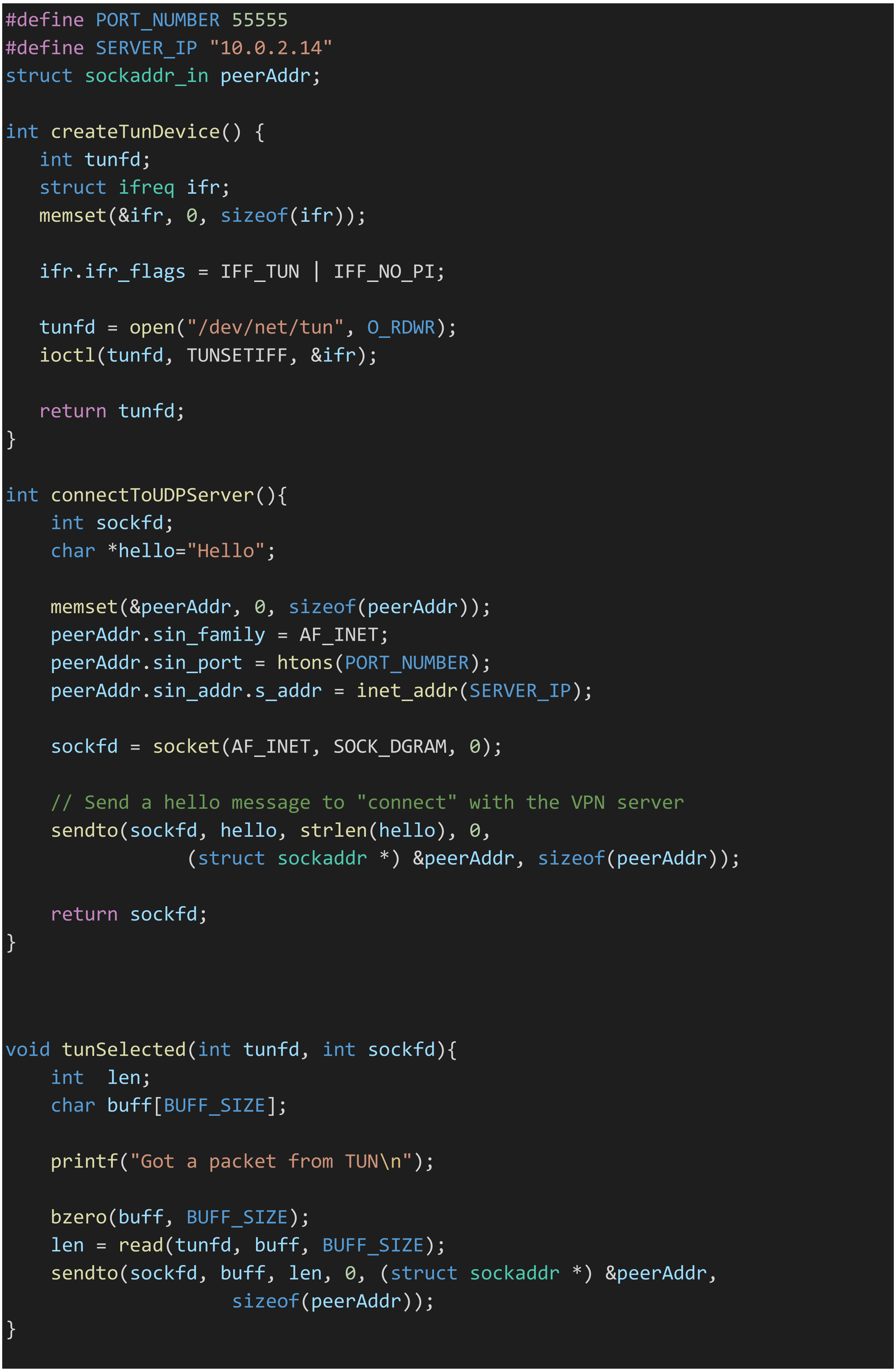
Fig. 2(d): Configuring the VPN tunnel.

From the figure above, it’s manifested that the tunnel interface is configured.

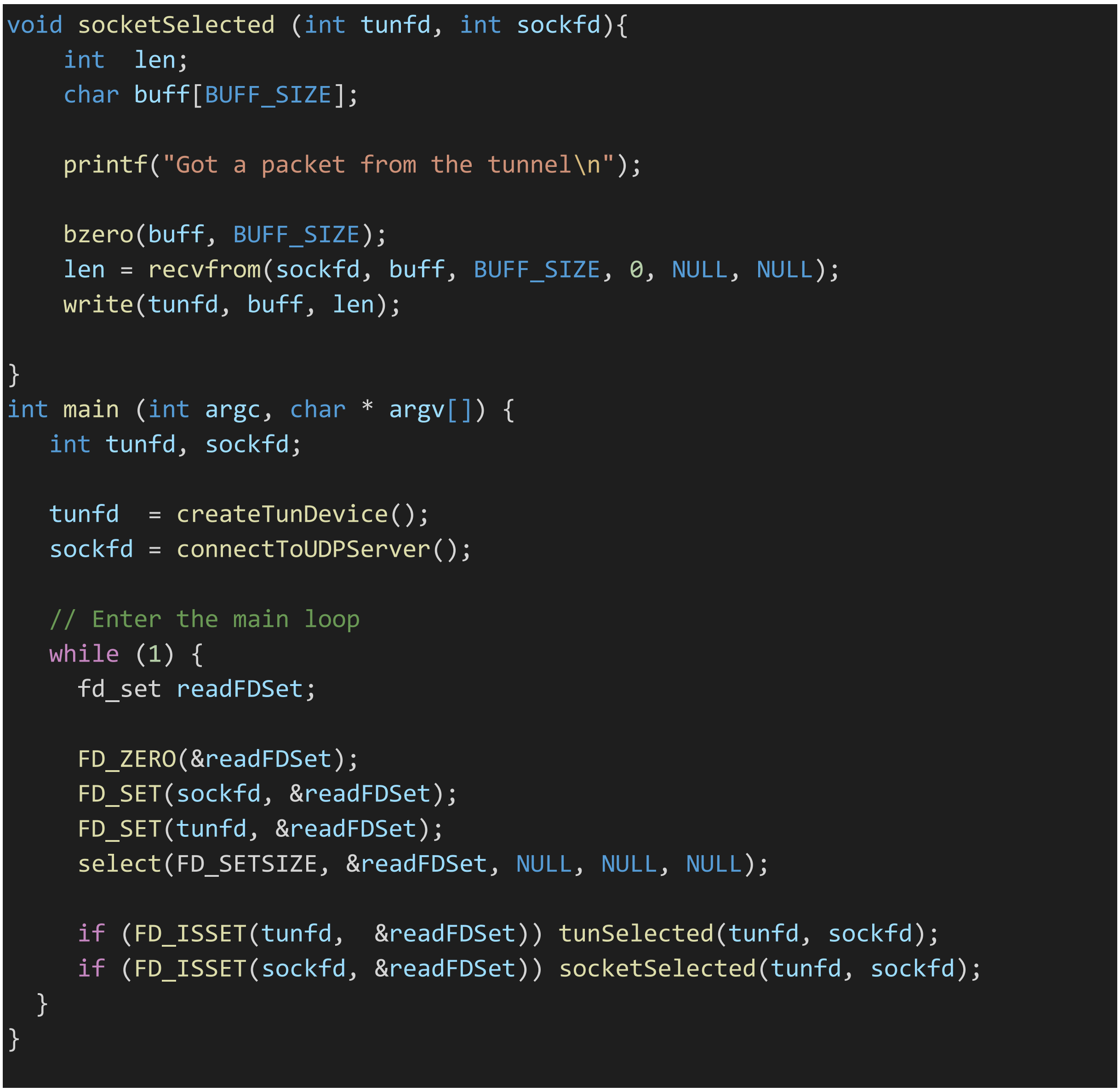
Step 2: Running VPN Client and setting IP address of the interface - (On the VPN Client VM)

The programme:

```
#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 "10.0.2.14"
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(){
    int sockfd;
    char *hello="Hello";

    memset(&peerAddr, 0, sizeof(peerAddr));
    peerAddr.sin_family = AF_INET;
    peerAddr.sin_port = htons(PORT_NUMBER);
    peerAddr.sin_addr.s_addr = inet_addr(SERVER_IP);

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    // Send a hello message to "connect" with the VPN server
    sendto(sockfd, hello, strlen(hello), 0,
                (struct sockaddr *) &peerAddr, sizeof(peerAddr));

    return sockfd;
}



void tunSelected(int tunfd, int sockfd){
    int  len;
    char buff[BUFF_SIZE];

    printf("Got a packet from TUN\n");

    bzero(buff, BUFF_SIZE);
    len = read(tunfd, buff, BUFF_SIZE);
    sendto(sockfd, buff, len, 0, (struct sockaddr *) &peerAddr,
                    sizeof(peerAddr));
}
```

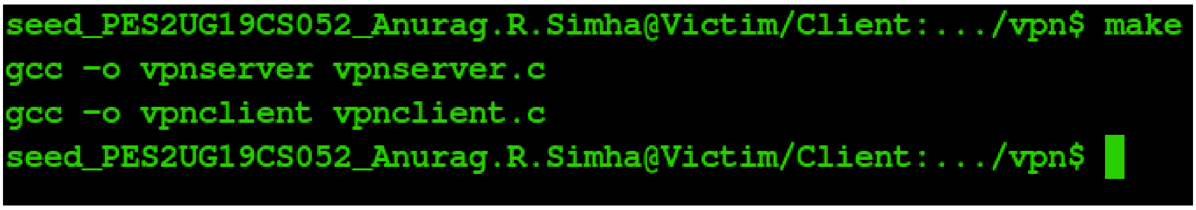


```
void socketSelected (int tunfd, int sockfd){
    int  len;
    char buff[BUFF_SIZE];

    printf("Got a packet from the tunnel\n");

    bzero(buff, BUFF_SIZE);
    len = recvfrom(sockfd, buff, BUFF_SIZE, 0, NULL, NULL);
    write(tunfd, buff, len);

}
int main (int argc, char * argv[]) {
   int tunfd, sockfd;

   tunfd  = createTunDevice();
   sockfd = connectToUDPServer();

   // Enter the main loop
   while (1) {
     fd_set readFDSet;

     FD_ZERO(&readFDSet);
     FD_SET(sockfd, &readFDSet);
     FD_SET(tunfd, &readFDSet);
     select(FD_SETSIZE, &readFDSet, NULL, NULL, NULL);

     if (FD_ISSET(tunfd,  &readFDSet)) tunSelected(tunfd, sockfd);
     if (FD_ISSET(sockfd, &readFDSet)) socketSelected(tunfd, sockfd);
  }
}
```

In the programme, the client machine is made to connect to the server.

The commands:

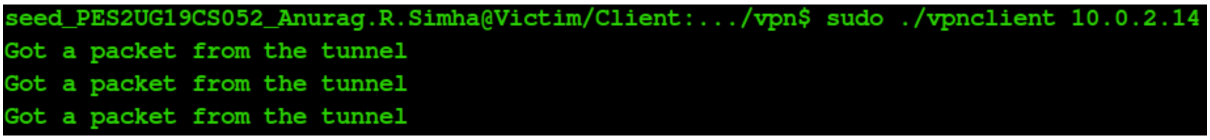
```
make
```



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$
```

Fig. 2(e): The object files are created with the make command.

```
sudo ./vpnclient 10.0.2.14
```



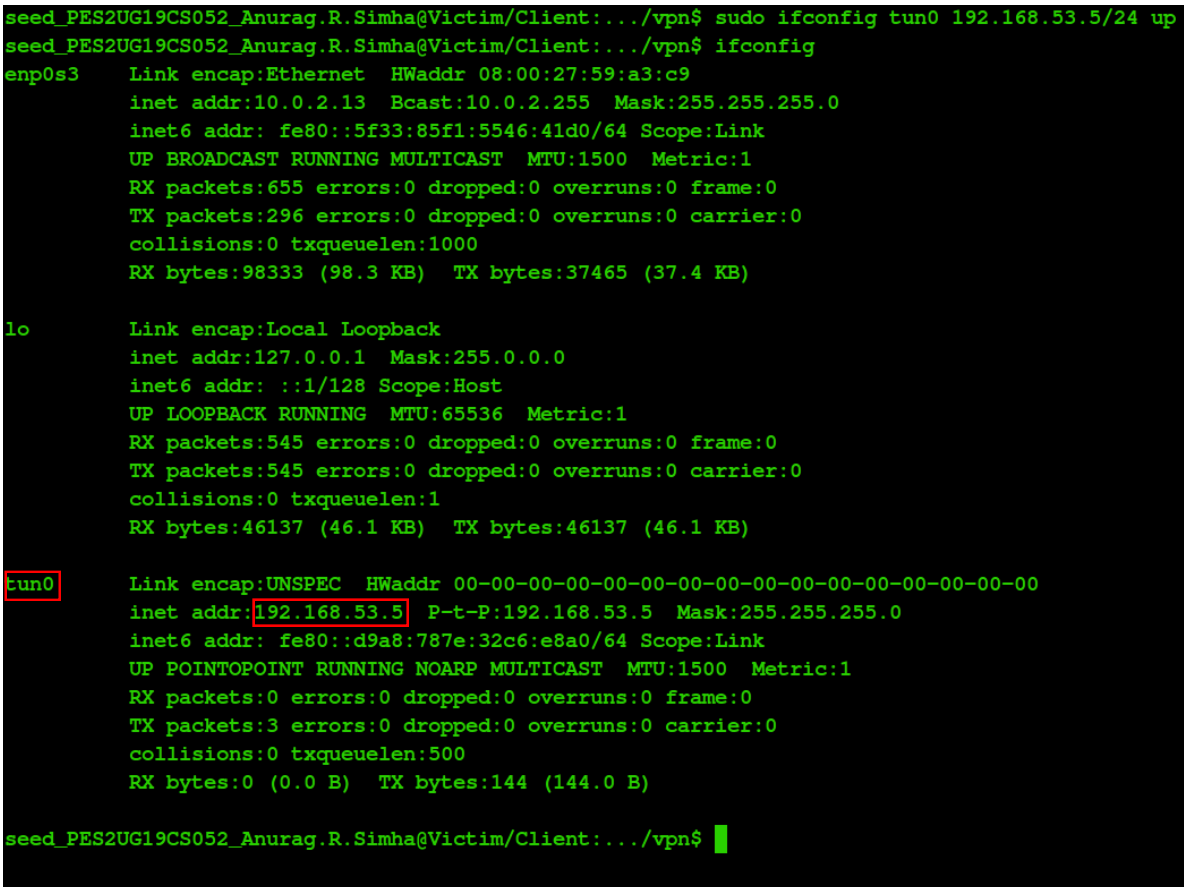
```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ sudo ./vpnclient 10.0.2.14
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```

Fig. 2(f): The tunnel seems to be opened.

Another terminal is opened and the command below is run:

`sudo ifconfig tun0 192.168.53.5/24 up`

After performing this, the command, `ifconfig` is put into effect.

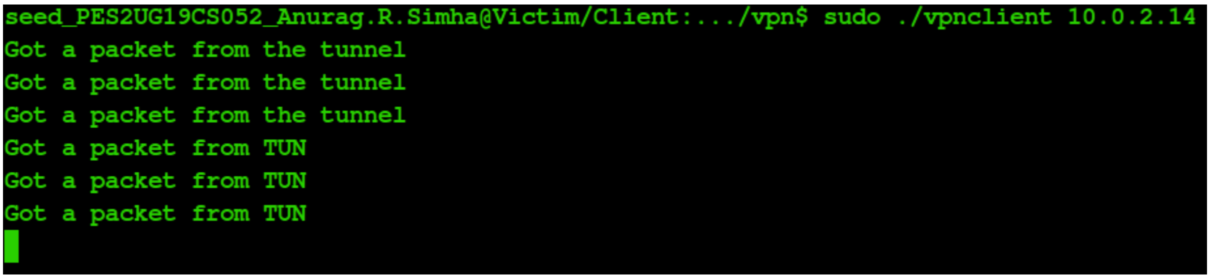


```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ sudo ifconfig tun0 192.168.53.5/24 up
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:59:a3:c9
          inet addr:10.0.2.13  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::5f33:85f1:5546:41d0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:655 errors:0 dropped:0 overruns:0 frame:0
          TX packets:296 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:98333 (98.3 KB)  TX bytes:37465 (37.4 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
          RX packets:545 errors:0 dropped:0 overruns:0 frame:0
          TX packets:545 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:46137 (46.1 KB)  TX bytes:46137 (46.1 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::d9a8:787e:32c6:e8a0/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:144 (144.0 B)

seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$
```

Fig. 2(g): The tunnel interface is hence configured.

The observation:

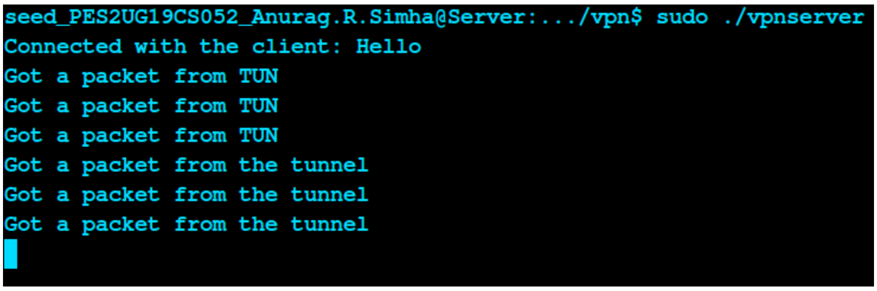
On the client machine:



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ sudo ./vpnclient 10.0.2.14
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
```

Fig. 2(h): Packets are being transmitted within the sockets.

On the server machine:



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../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
```

Fig. 2(i): Packets are being transmitted within the sockets.

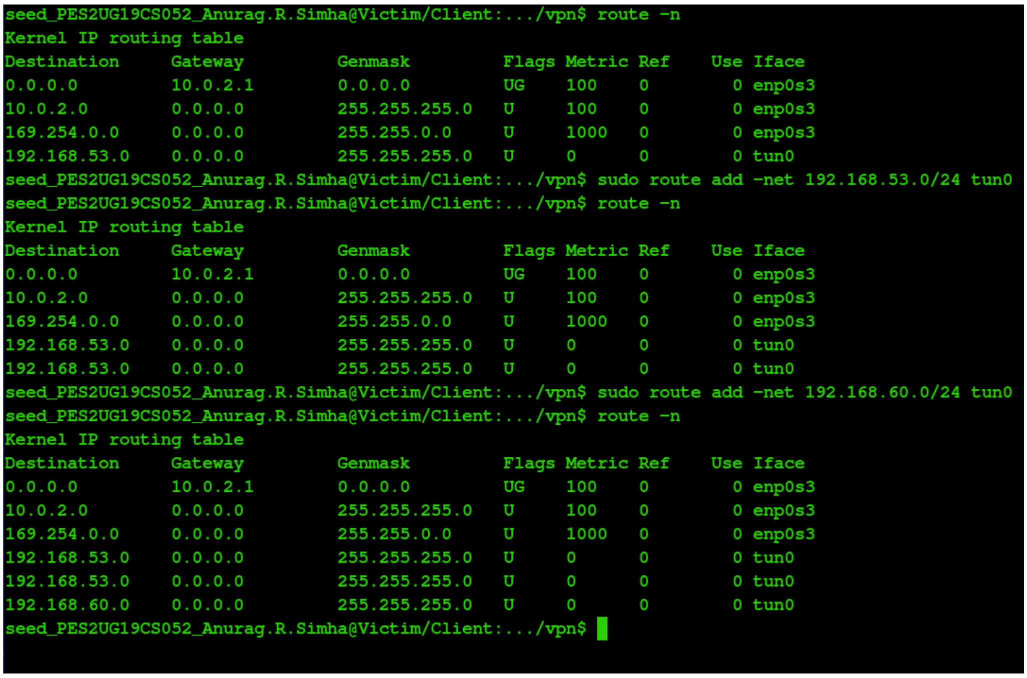
It’s observed that, before configuring the tunnel on the client machine, there’s no packet received from TUN. But, on a triumphant configuration of the tunnel interface, there’re packets received over both the machines. Thus, both sides communication is (fractionally) achieved.

Step 3: Setting up routing on Client and Server VMs

On the client VM:

The commands:

```
sudo route add -net 192.168.53.0/24 tun0

route -n

sudo route add -net 192.168.60.0/24 tun0

route -n
```



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ 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    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
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ sudo route add -net 192.168.53.0/24 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ 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    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
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ sudo route add -net 192.168.60.0/24 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ 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    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
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.60.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$
```

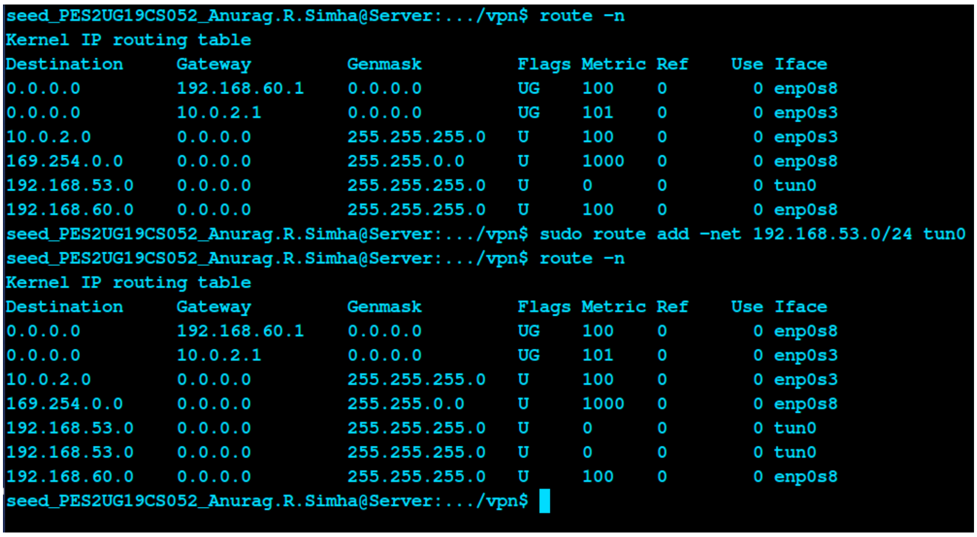
Fig. 2(j): The routing table is setup.

The routing table is hence setup.

On the server VM:

The commands:

```
sudo route add -net 192.168.53.0/24 tun0
route -n
```



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.60.1    0.0.0.0         UG    100    0        0 enp0s8
0.0.0.0         10.0.2.1        0.0.0.0         UG    101    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 enp0s8
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.60.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s8
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo route add -net 192.168.53.0/24 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.60.1    0.0.0.0         UG    100    0        0 enp0s8
0.0.0.0         10.0.2.1        0.0.0.0         UG    101    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 enp0s8
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.60.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s8
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$
```

Fig. 2(k): The routing table is setup

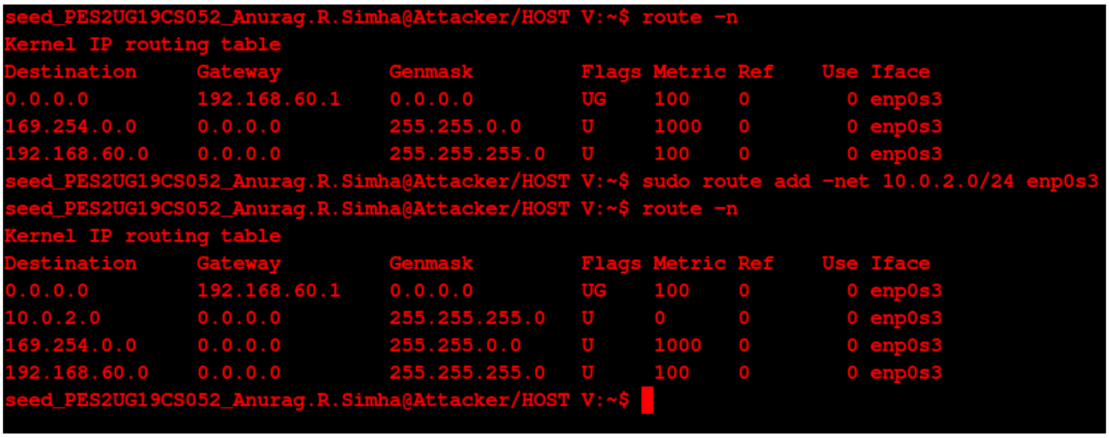
The routing table is hence setup.

Step 4: Set up routing on HOST V

On Host V,

The commands:

```
sudo route add -net 10.0.2.0/24 enp0s3
route -n
```



```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.60.1    0.0.0.0         UG    100    0        0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
192.168.60.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ sudo route add -net 10.0.2.0/24 enp0s3
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.60.1    0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
192.168.60.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$
```

Fig. 2(*l*): The routing table is setup

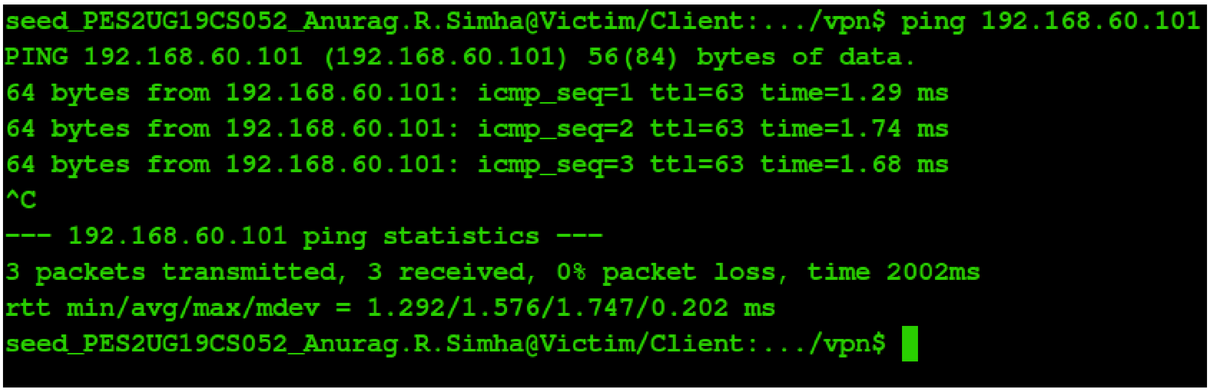
The routing table is hence setup.

Step 5: Testing the VPN tunnel (ping and telnet)

This step is performed over the client machine.

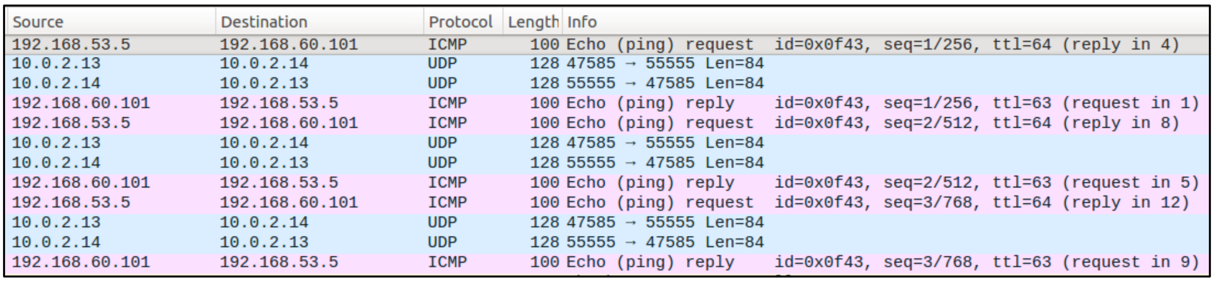
The command:

```
ping 192.168.60.101
```



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=63 time=1.29 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=63 time=1.74 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=63 time=1.68 ms
^C
--- 192.168.60.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.292/1.576/1.747/0.202 ms
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ 
```

Fig. 2(m): Performing the successful ping operation on the isolated machine.



Source	Destination	Protocol	Length	Info
192.168.53.5	192.168.60.101	ICMP	100	Echo (ping) request id=0x0f43, seq=1/256, ttl=64 (reply in 4)
10.0.2.13	10.0.2.14	UDP	128	47585 → 55555 Len=84
10.0.2.14	10.0.2.13	UDP	128	55555 → 47585 Len=84
192.168.60.101	192.168.53.5	ICMP	100	Echo (ping) reply id=0x0f43, seq=1/256, ttl=63 (request in 1)
192.168.53.5	192.168.60.101	ICMP	100	Echo (ping) request id=0x0f43, seq=2/512, ttl=64 (reply in 8)
10.0.2.13	10.0.2.14	UDP	128	47585 → 55555 Len=84
10.0.2.14	10.0.2.13	UDP	128	55555 → 47585 Len=84
192.168.60.101	192.168.53.5	ICMP	100	Echo (ping) reply id=0x0f43, seq=2/512, ttl=63 (request in 5)
192.168.53.5	192.168.60.101	ICMP	100	Echo (ping) request id=0x0f43, seq=3/768, ttl=64 (reply in 12)
10.0.2.13	10.0.2.14	UDP	128	47585 → 55555 Len=84
10.0.2.14	10.0.2.13	UDP	128	55555 → 47585 Len=84
192.168.60.101	192.168.53.5	ICMP	100	Echo (ping) reply id=0x0f43, seq=3/768, ttl=63 (request in 9)

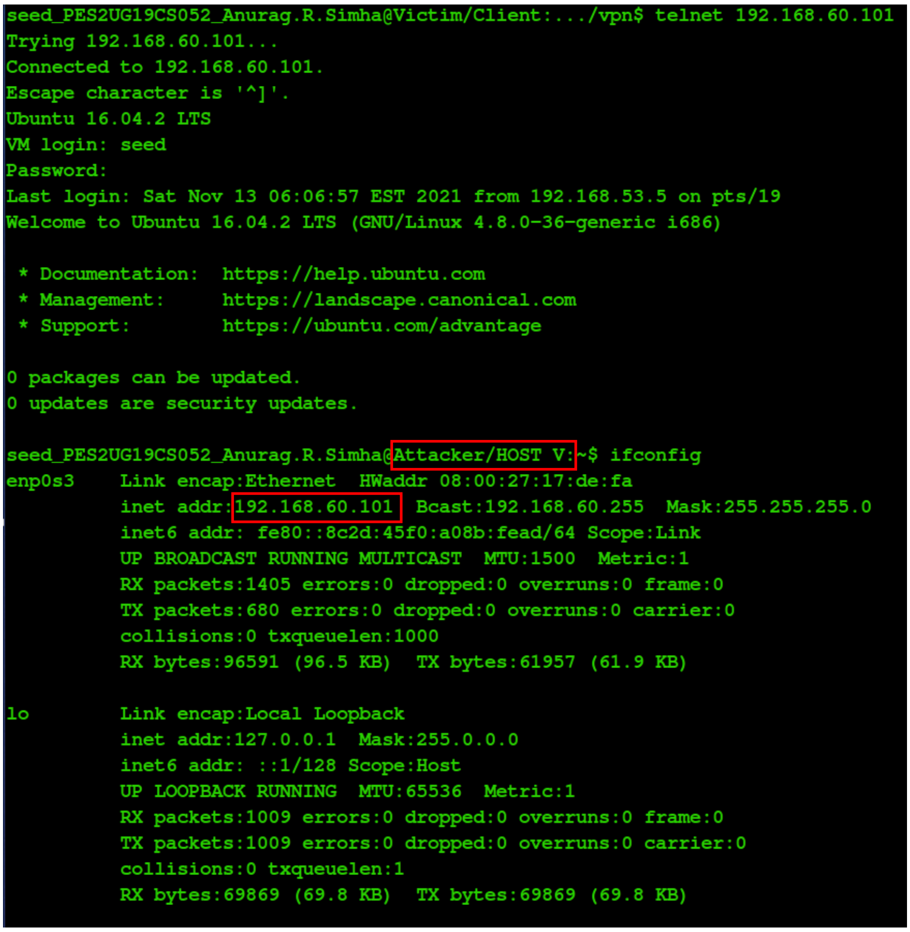
Fig. 2(n): The Wireshark capture result.

It's observed that, on performing the ping operation, there are UDP packets transferred between the two machines (client and server). Then, due to the activation of IP forwarding upon the server machine (step 1), the isolated machine is triumphantly contacted via the tunnel interface. The packets coloured in blue are the UDP packets and the packets in pink, including the grey are the ICMP packets.

Next, a telnet connection to the host machine is performed.

The command:

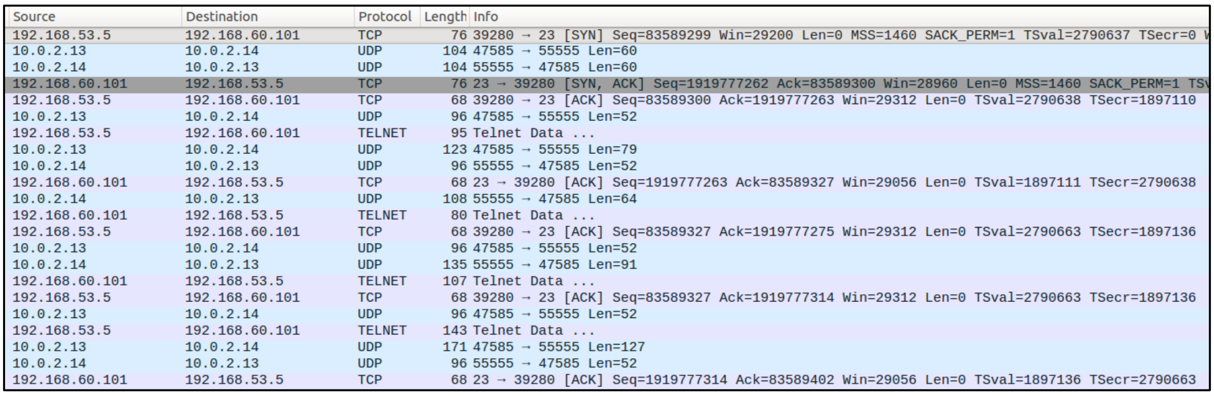
```
telnet 192.168.60.101
```



```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Nov 13 06:06:57 EST 2021 from 192.168.53.5 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:17:de:fa
          inet addr:192.168.60.101  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::8c2d:45f0:a08b:fead/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1405 errors:0 dropped:0 overruns:0 frame:0
          TX packets:680 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:96591 (96.5 KB)  TX bytes:61957 (61.9 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:1009 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1009 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:69869 (69.8 KB)  TX bytes:69869 (69.8 KB)
```

Fig. 2(o): There’s a successful connection to the host machine.

The Wireshark packet capture:

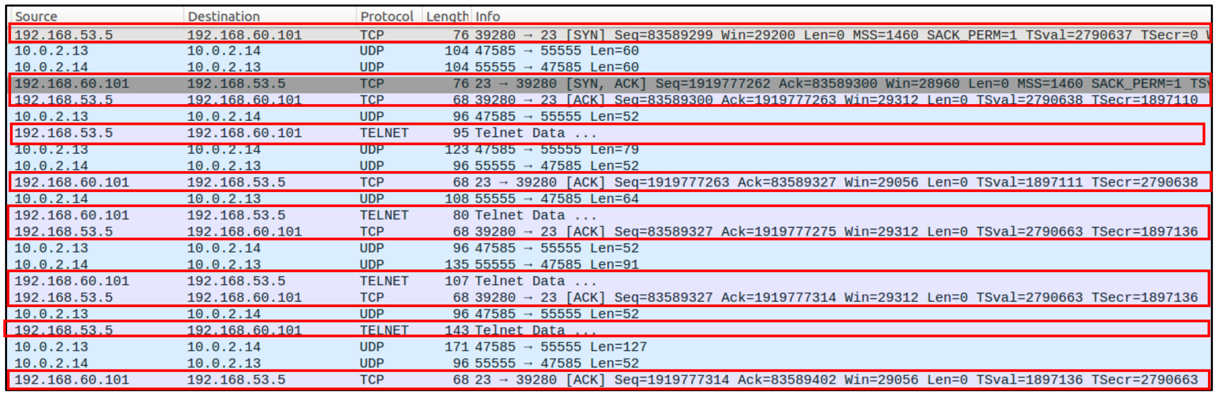


Source	Destination	Protocol	Length	Info
192.168.53.5	192.168.60.101	TCP	76	39280 → 23 [SYN] Seq=83589299 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2790637 TSecr=0 W
10.0.2.13	10.0.2.14	UDP	104	47585 → 55555 Len=60
10.0.2.14	10.0.2.13	UDP	104	55555 → 47585 Len=60
192.168.60.101	192.168.53.5	TCP	76	23 → 39280 [SYN, ACK] Seq=1919777262 Ack=83589300 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSv
192.168.53.5	192.168.60.101	TCP	68	39280 → 23 [ACK] Seq=83589300 Ack=1919777263 Win=29312 Len=0 TSval=2790638 TSecr=1897110
10.0.2.13	10.0.2.14	UDP	96	47585 → 55555 Len=52
192.168.53.5	192.168.60.101	TELNET	95	Telnet Data ...
10.0.2.13	10.0.2.14	UDP	123	47585 → 55555 Len=79
10.0.2.14	10.0.2.13	UDP	96	55555 → 47585 Len=52
192.168.60.101	192.168.53.5	TCP	68	23 → 39280 [ACK] Seq=1919777263 Ack=83589327 Win=29056 Len=0 TSval=1897111 TSecr=2790638
10.0.2.14	10.0.2.13	UDP	108	55555 → 47585 Len=64
192.168.60.101	192.168.53.5	TELNET	80	Telnet Data ...
192.168.53.5	192.168.60.101	TCP	68	39280 → 23 [ACK] Seq=83589327 Ack=1919777275 Win=29312 Len=0 TSval=2790663 TSecr=1897136
10.0.2.13	10.0.2.14	UDP	96	47585 → 55555 Len=52
10.0.2.14	10.0.2.13	UDP	135	55555 → 47585 Len=91
192.168.60.101	192.168.53.5	TELNET	107	Telnet Data ...
192.168.53.5	192.168.60.101	TCP	68	39280 → 23 [ACK] Seq=83589327 Ack=1919777314 Win=29312 Len=0 TSval=2790663 TSecr=1897136
10.0.2.13	10.0.2.14	UDP	96	47585 → 55555 Len=52
192.168.53.5	192.168.60.101	TELNET	143	Telnet Data ...
10.0.2.13	10.0.2.14	UDP	171	47585 → 55555 Len=127
10.0.2.14	10.0.2.13	UDP	96	55555 → 47585 Len=52
192.168.60.101	192.168.53.5	TCP	68	23 → 39280 [ACK] Seq=1919777314 Ack=83589402 Win=29056 Len=0 TSval=1897136 TSecr=2790663

Fig. 2(p): The Wireshark packet capture results.

From figures 2(o) and 2(p), it can be deemed that there’s a triumphant connection established through the VPN tunnel to the host machine.

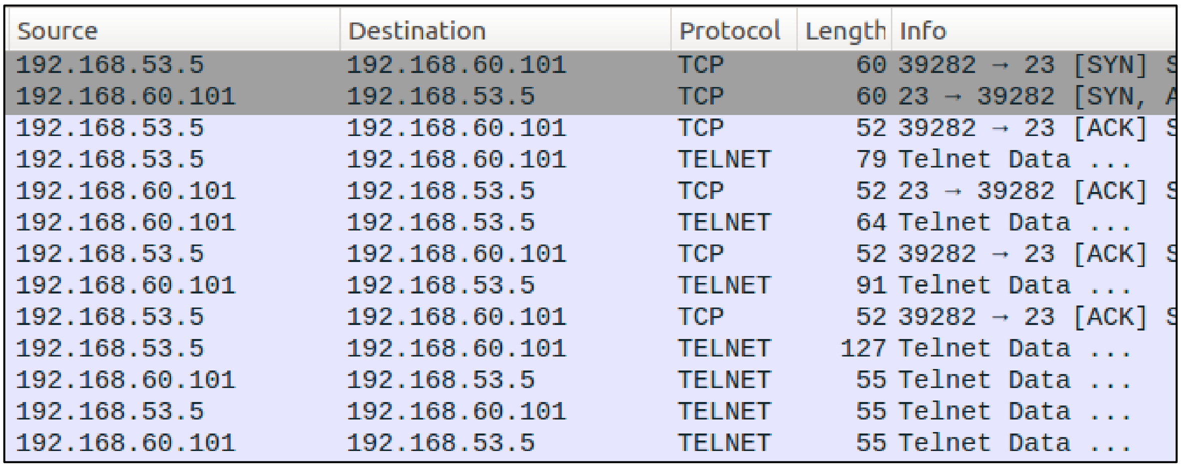
The figure(s) below shows those packets that are the traffic not generated in the tunnel.



```
Source          Destination     Protocol Length Info
192.168.53.5    192.168.60.101  TCP        76 39280 → 23 [SYN] Seq=83589299 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2790637 TSecr=0
10.0.2.13       10.0.2.14       UDP       104 47585 → 55555 Len=60
10.0.2.14       10.0.2.13       UDP       104 55555 → 47585 Len=60
192.168.60.101  192.168.53.5    TCP        76 23 → 39280 [SYN, ACK] Seq=1919777262 Ack=83589300 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TS
192.168.53.5    192.168.60.101  TCP        68 39280 → 23 [ACK] Seq=83589300 Ack=1919777263 Win=29312 Len=0 TSval=2790638 TSecr=1897110
10.0.2.13       10.0.2.14       UDP        96 47585 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET     95 Telnet Data ...
10.0.2.13       10.0.2.14       UDP       123 47585 → 55555 Len=79
10.0.2.14       10.0.2.13       UDP        96 55555 → 47585 Len=52
192.168.60.101  192.168.53.5    TCP        68 23 → 39280 [ACK] Seq=1919777263 Ack=83589327 Win=29056 Len=0 TSval=1897111 TSecr=2790638
10.0.2.14       10.0.2.13       UDP       108 55555 → 47585 Len=64
192.168.60.101  192.168.53.5    TELNET     80 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39280 → 23 [ACK] Seq=83589327 Ack=1919777275 Win=29312 Len=0 TSval=2790663 TSecr=1897136
10.0.2.13       10.0.2.14       UDP        96 47585 → 55555 Len=52
10.0.2.14       10.0.2.13       UDP       135 55555 → 47585 Len=91
192.168.60.101  192.168.53.5    TELNET    107 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39280 → 23 [ACK] Seq=83589327 Ack=1919777314 Win=29312 Len=0 TSval=2790663 TSecr=1897136
10.0.2.13       10.0.2.14       UDP        96 47585 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET    143 Telnet Data ...
10.0.2.13       10.0.2.14       UDP       171 47585 → 55555 Len=127
10.0.2.14       10.0.2.13       UDP        96 55555 → 47585 Len=52
192.168.60.101  192.168.53.5    TCP        68 23 → 39280 [ACK] Seq=1919777314 Ack=83589402 Win=29056 Len=0 TSval=1897136 TSecr=2790663
```

Fig. 2(q): The packets that are not in the tunnel traffic.

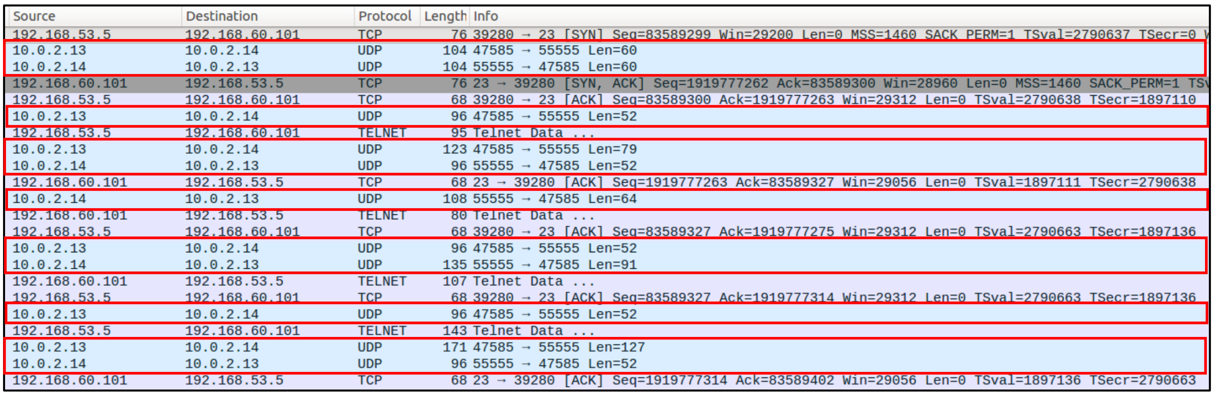
The packets when captured over the interface ‘tun0’:



```
Source          Destination     Protocol Length Info
192.168.53.5    192.168.60.101  TCP        60 39282 → 23 [SYN] S
192.168.60.101  192.168.53.5    TCP        60 23 → 39282 [SYN, A
192.168.53.5    192.168.60.101  TCP        52 39282 → 23 [ACK] S
192.168.53.5    192.168.60.101  TELNET     79 Telnet Data ...
192.168.60.101  192.168.53.5    TCP        52 23 → 39282 [ACK] S
192.168.60.101  192.168.53.5    TELNET     64 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        52 39282 → 23 [ACK] S
192.168.60.101  192.168.53.5    TELNET     91 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        52 39282 → 23 [ACK] S
192.168.53.5    192.168.60.101  TELNET    127 Telnet Data ...
192.168.60.101  192.168.53.5    TELNET     55 Telnet Data ...
192.168.53.5    192.168.60.101  TELNET     55 Telnet Data ...
192.168.60.101  192.168.53.5    TELNET     55 Telnet Data ...
```

Fig. 2(r): The traffic over the interface, tun0.

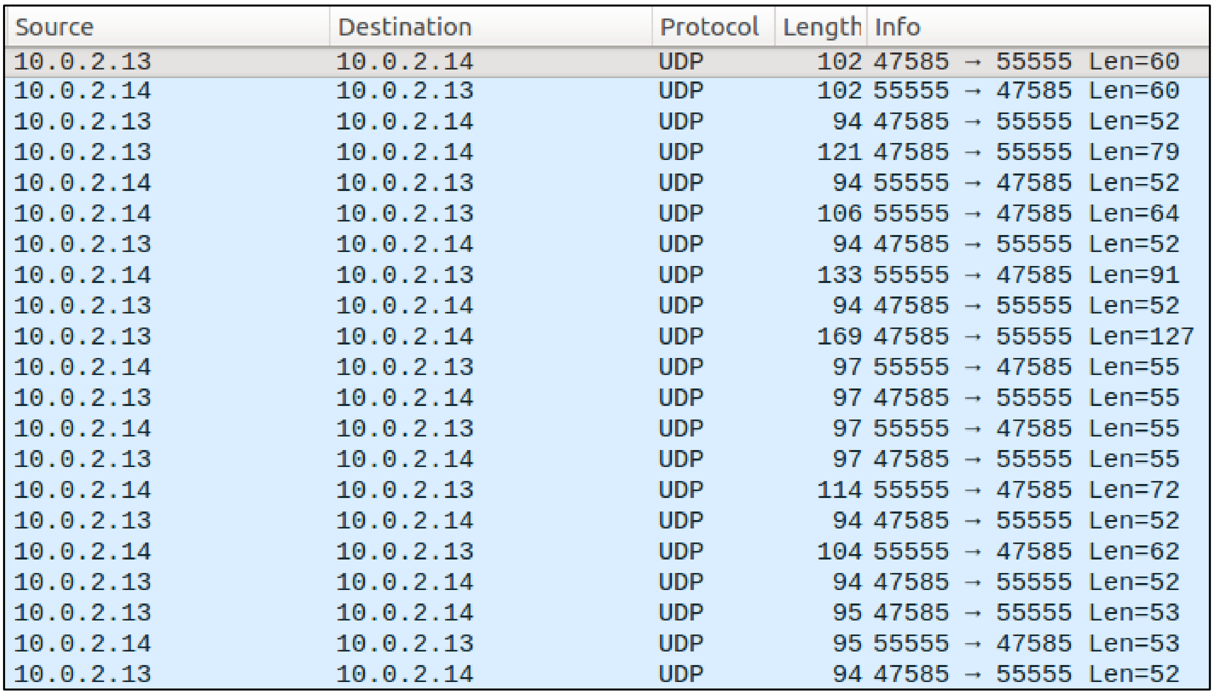
The figure(s) below shows those packets that are the traffic generated in the tunnel.



```
Source          Destination     Protocol Length Info
192.168.53.5    192.168.60.101  TCP        76 39280 → 23 [SYN] Seq=83589299 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2790637 TSecr=0
10.0.2.13       10.0.2.14       UDP       104 47585 → 55555 Len=60
10.0.2.14       10.0.2.13       UDP       104 55555 → 47585 Len=60
192.168.60.101  192.168.53.5    TCP        76 23 → 39280 [SYN, ACK] Seq=1919777262 Ack=83589300 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TS
192.168.53.5    192.168.60.101  TCP        68 39280 → 23 [ACK] Seq=83589300 Ack=1919777263 Win=29312 Len=0 TSval=2790638 TSecr=1897110
10.0.2.13       10.0.2.14       UDP        96 47585 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET     95 Telnet Data ...
10.0.2.13       10.0.2.14       UDP       123 47585 → 55555 Len=79
10.0.2.14       10.0.2.13       UDP        96 55555 → 47585 Len=52
192.168.60.101  192.168.53.5    TCP        68 23 → 39280 [ACK] Seq=1919777263 Ack=83589327 Win=29056 Len=0 TSval=1897111 TSecr=2790638
10.0.2.14       10.0.2.13       UDP       108 55555 → 47585 Len=64
192.168.60.101  192.168.53.5    TELNET     80 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39280 → 23 [ACK] Seq=83589327 Ack=1919777275 Win=29312 Len=0 TSval=2790663 TSecr=1897136
10.0.2.13       10.0.2.14       UDP        96 47585 → 55555 Len=52
10.0.2.14       10.0.2.13       UDP       135 55555 → 47585 Len=91
192.168.60.101  192.168.53.5    TELNET    107 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39280 → 23 [ACK] Seq=83589327 Ack=1919777314 Win=29312 Len=0 TSval=2790663 TSecr=1897136
10.0.2.13       10.0.2.14       UDP        96 47585 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET    143 Telnet Data ...
10.0.2.13       10.0.2.14       UDP       171 47585 → 55555 Len=127
10.0.2.14       10.0.2.13       UDP        96 55555 → 47585 Len=52
192.168.60.101  192.168.53.5    TCP        68 23 → 39280 [ACK] Seq=1919777314 Ack=83589402 Win=29056 Len=0 TSval=1897136 TSecr=2790663
```

Fig. 2(s): The packets that are in the tunnel traffic.

The packets highlighted are those that appear under the interface, ‘enp0s3’.



Source	Destination	Protocol	Length	Info
10.0.2.13	10.0.2.14	UDP	102	47585 → 55555 Len=60
10.0.2.14	10.0.2.13	UDP	102	55555 → 47585 Len=60
10.0.2.13	10.0.2.14	UDP	94	47585 → 55555 Len=52
10.0.2.13	10.0.2.14	UDP	121	47585 → 55555 Len=79
10.0.2.14	10.0.2.13	UDP	94	55555 → 47585 Len=52
10.0.2.14	10.0.2.13	UDP	106	55555 → 47585 Len=64
10.0.2.13	10.0.2.14	UDP	94	47585 → 55555 Len=52
10.0.2.14	10.0.2.13	UDP	133	55555 → 47585 Len=91
10.0.2.13	10.0.2.14	UDP	94	47585 → 55555 Len=52
10.0.2.13	10.0.2.14	UDP	169	47585 → 55555 Len=127
10.0.2.14	10.0.2.13	UDP	97	55555 → 47585 Len=55
10.0.2.13	10.0.2.14	UDP	97	47585 → 55555 Len=55
10.0.2.14	10.0.2.13	UDP	97	55555 → 47585 Len=55
10.0.2.13	10.0.2.14	UDP	97	47585 → 55555 Len=55
10.0.2.14	10.0.2.13	UDP	114	55555 → 47585 Len=72
10.0.2.13	10.0.2.14	UDP	94	47585 → 55555 Len=52
10.0.2.14	10.0.2.13	UDP	104	55555 → 47585 Len=62
10.0.2.13	10.0.2.14	UDP	94	47585 → 55555 Len=52
10.0.2.13	10.0.2.14	UDP	95	47585 → 55555 Len=53
10.0.2.14	10.0.2.13	UDP	95	55555 → 47585 Len=53
10.0.2.13	10.0.2.14	UDP	94	47585 → 55555 Len=52

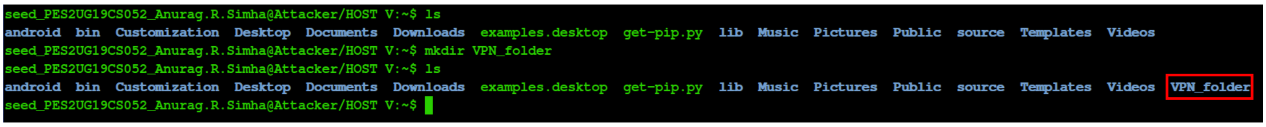
Fig. 2(t): The traffic over the interface, enp0s3.

Next, on the host machine, a folder is created.

The commands:

```
mkdir VPN_folder
```

```
ls
```

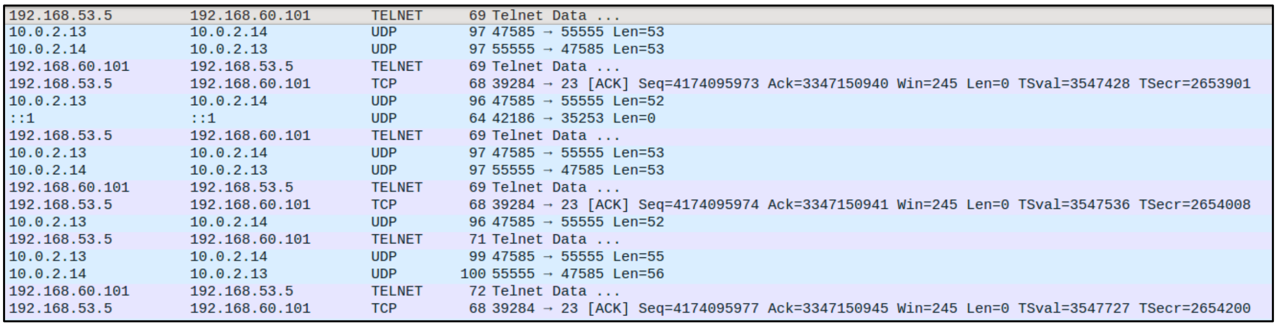


```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ls
android  bin  Customization  Desktop  Documents  Downloads  examples.desktop  get-pip.py  lib  Music  Pictures  Public  source  Templates  Videos
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ mkdir VPN_folder
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ls
android  bin  Customization  Desktop  Documents  Downloads  examples.desktop  get-pip.py  lib  Music  Pictures  Public  source  Templates  Videos  VPN_folder
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ 
```

Fig. 2(u): Creating the folder.

Figure 2(u) is the manifestation of the triumph in establishing the connection. The folder titled, 'VPN_folder' is crystal clearly visible.

Packets are captured on Wireshark for this action.



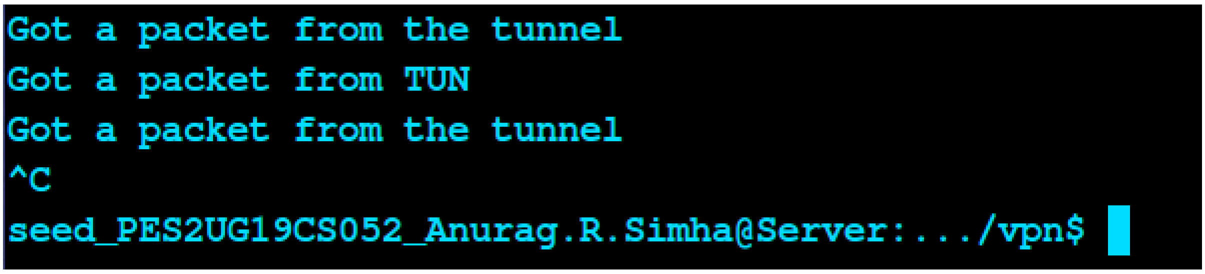
```
192.168.53.5     192.168.60.101   TELNET    69 Telnet Data ...
10.0.2.13        10.0.2.14        UDP       97 47585 → 55555 Len=53
10.0.2.14        10.0.2.13        UDP       97 55555 → 47585 Len=53
192.168.60.101   192.168.53.5     TELNET    69 Telnet Data ...
192.168.53.5     192.168.60.101   TCP       68 39284 → 23 [ACK] Seq=4174095973 Ack=3347150940 Win=245 Len=0 TSval=3547428 TSecr=2653901
10.0.2.13        10.0.2.14        UDP       96 47585 → 55555 Len=52
::1              ::1              UDP       64 42186 → 35253 Len=0
192.168.53.5     192.168.60.101   TELNET    69 Telnet Data ...
10.0.2.13        10.0.2.14        UDP       97 47585 → 55555 Len=53
10.0.2.14        10.0.2.13        UDP       97 55555 → 47585 Len=53
192.168.60.101   192.168.53.5     TELNET    69 Telnet Data ...
192.168.53.5     192.168.60.101   TCP       68 39284 → 23 [ACK] Seq=4174095974 Ack=3347150941 Win=245 Len=0 TSval=3547536 TSecr=2654008
10.0.2.13        10.0.2.14        UDP       96 47585 → 55555 Len=52
192.168.53.5     192.168.60.101   TELNET    71 Telnet Data ...
10.0.2.13        10.0.2.14        UDP       99 47585 → 55555 Len=55
10.0.2.14        10.0.2.13        UDP      100 55555 → 47585 Len=56
192.168.60.101   192.168.53.5     TELNET    72 Telnet Data ...
192.168.53.5     192.168.60.101   TCP       68 39284 → 23 [ACK] Seq=4174095977 Ack=3347150945 Win=245 Len=0 TSval=3547727 TSecr=2654200
```

Fig. 2(v): The Wireshark packet capture on creating the directory.

Step 6: Tunnel-Breaking Test

The established tunnel is then broken for testing purpose, and then reconnected.

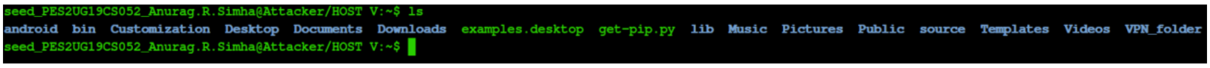
Running on the server machine, the vpnserver.c programme is ceased to function impermanently.



```
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
^C
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$
```

Fig. 2(w): The server programme is ceased.

The command, ‘ls’ is typed on the client machine.

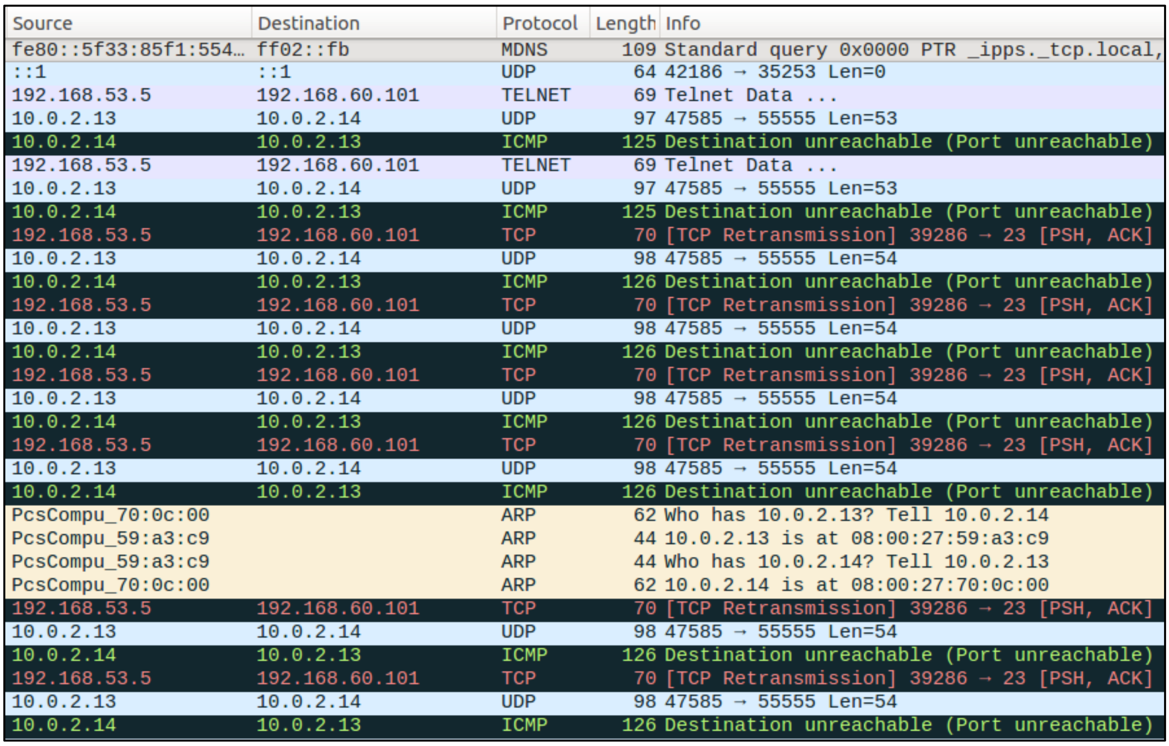


```
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ls
android  bin  Customization  Desktop  Documents  Downloads  examples.desktop  get-pip.py  lib  Music  Pictures  Public  source  Templates  Videos  VPN_folder
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$
```

Fig. 2(x): ls does not function.

In the figure above, the ‘ls’ command in the first line is the old output. When it’s once again typed after interrupting the connection, ‘ls’ does not appear on the window. For, the tunnel remains fractured.

The packets captured on Wireshark yield flabbergasting results.



Source	Destination	Protocol	Length	Info
fe80::5f33:85f1:554…	ff02::fb	MDNS	109	Standard query 0x0000 PTR _ipps._tcp.local,
::1	::1	UDP	64	42186 → 35253 Len=0
192.168.53.5	192.168.60.101	TELNET	69	Telnet Data ...
10.0.2.13	10.0.2.14	UDP	97	47585 → 55555 Len=53
10.0.2.14	10.0.2.13	ICMP	125	Destination unreachable (Port unreachable)
192.168.53.5	192.168.60.101	TELNET	69	Telnet Data ...
10.0.2.13	10.0.2.14	UDP	97	47585 → 55555 Len=53
10.0.2.14	10.0.2.13	ICMP	125	Destination unreachable (Port unreachable)
192.168.53.5	192.168.60.101	TCP	70	[TCP Retransmission] 39286 → 23 [PSH, ACK]
10.0.2.13	10.0.2.14	UDP	98	47585 → 55555 Len=54
10.0.2.14	10.0.2.13	ICMP	126	Destination unreachable (Port unreachable)
192.168.53.5	192.168.60.101	TCP	70	[TCP Retransmission] 39286 → 23 [PSH, ACK]
10.0.2.13	10.0.2.14	UDP	98	47585 → 55555 Len=54
10.0.2.14	10.0.2.13	ICMP	126	Destination unreachable (Port unreachable)
192.168.53.5	192.168.60.101	TCP	70	[TCP Retransmission] 39286 → 23 [PSH, ACK]
10.0.2.13	10.0.2.14	UDP	98	47585 → 55555 Len=54
10.0.2.14	10.0.2.13	ICMP	126	Destination unreachable (Port unreachable)
192.168.53.5	192.168.60.101	TCP	70	[TCP Retransmission] 39286 → 23 [PSH, ACK]
10.0.2.13	10.0.2.14	UDP	98	47585 → 55555 Len=54
10.0.2.14	10.0.2.13	ICMP	126	Destination unreachable (Port unreachable)
PcsCompu_70:0c:00		ARP	62	Who has 10.0.2.13? Tell 10.0.2.14
PcsCompu_59:a3:c9		ARP	44	10.0.2.13 is at 08:00:27:59:a3:c9
PcsCompu_59:a3:c9		ARP	44	Who has 10.0.2.14? Tell 10.0.2.13
PcsCompu_70:0c:00		ARP	62	10.0.2.14 is at 08:00:27:70:0c:00
192.168.53.5	192.168.60.101	TCP	70	[TCP Retransmission] 39286 → 23 [PSH, ACK]
10.0.2.13	10.0.2.14	UDP	98	47585 → 55555 Len=54
10.0.2.14	10.0.2.13	ICMP	126	Destination unreachable (Port unreachable)
192.168.53.5	192.168.60.101	TCP	70	[TCP Retransmission] 39286 → 23 [PSH, ACK]
10.0.2.13	10.0.2.14	UDP	98	47585 → 55555 Len=54
10.0.2.14	10.0.2.13	ICMP	126	Destination unreachable (Port unreachable)

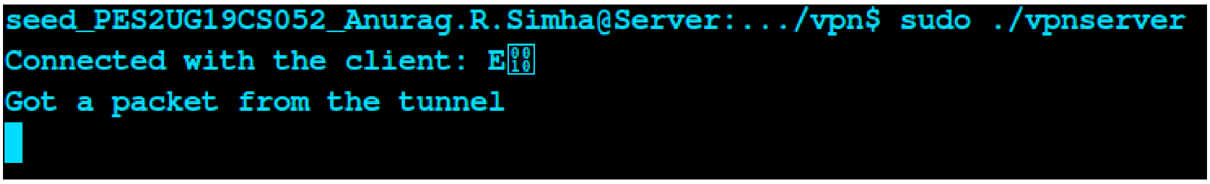
Fig. 2(y): The Wireshark packet capture.

It’s observed that, when ‘ls’ is typed, the server’s unable to reach the client (10.0.2.14 → 10.0.2.13, Destination unreachable). Since this attempt fails, the tunnel interface, too, is unable to reach the host machine. 10.0.2.13 can contact 10.0.2.14. But, for 10.0.2.13 to contact 192.168.60.101, the existence of the tunnel is a vital desideratum. This occurs when 10.0.2.14 contacts 192.168.60.101. While returning or fetching the data, it must travel through 192.168.53.5, which is the broken tunnel. Henceforth, there’s nothing displayed on the terminal.

Next, the tunnel is to be reconnected.

Henceforth, steps 1 and 2 are repeated.

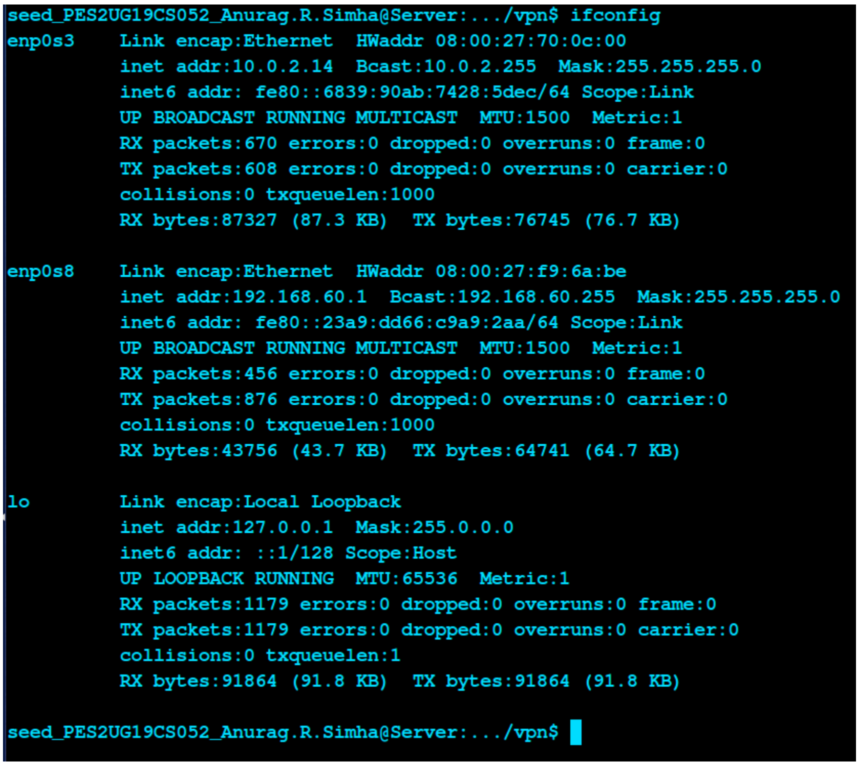
To re-establish the connection, the programme is instigated once again and a couple of steps are followed.



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo ./vpnserver
Connected with the client: E
Got a packet from the tunnel
```

Fig. 2(z): Reconnecting to the tunnel.

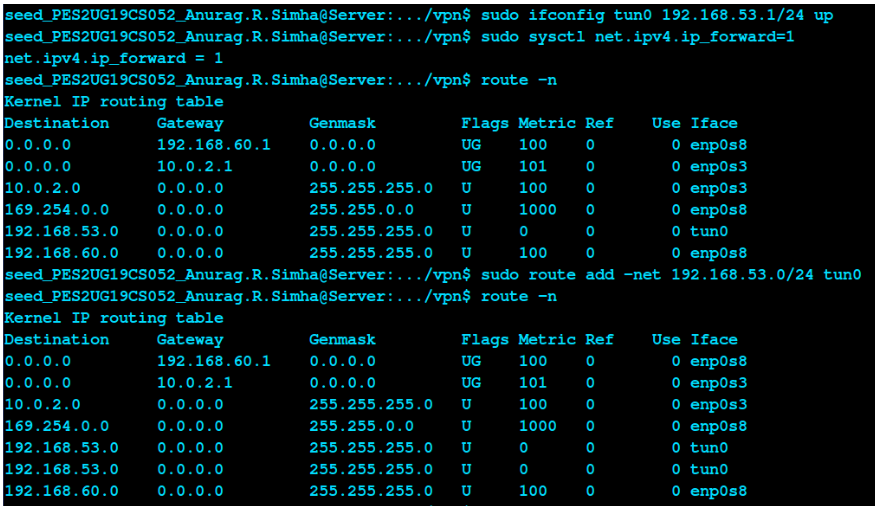
The tunnel interface seems to be down.



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:70:0c:00
          inet addr:10.0.2.14  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6839:90ab:7428:5dec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:670 errors:0 dropped:0 overruns:0 frame:0
          TX packets:608 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:87327 (87.3 KB)  TX bytes:76745 (76.7 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:f9:6a:be
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::23a9:dd66:c9a9:2aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:456 errors:0 dropped:0 overruns:0 frame:0
          TX packets:876 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43756 (43.7 KB)  TX bytes:64741 (64.7 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:1179 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1179 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:91864 (91.8 KB)  TX bytes:91864 (91.8 KB)

seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$
```

Fig. 2(A): The tun0 interface is down.

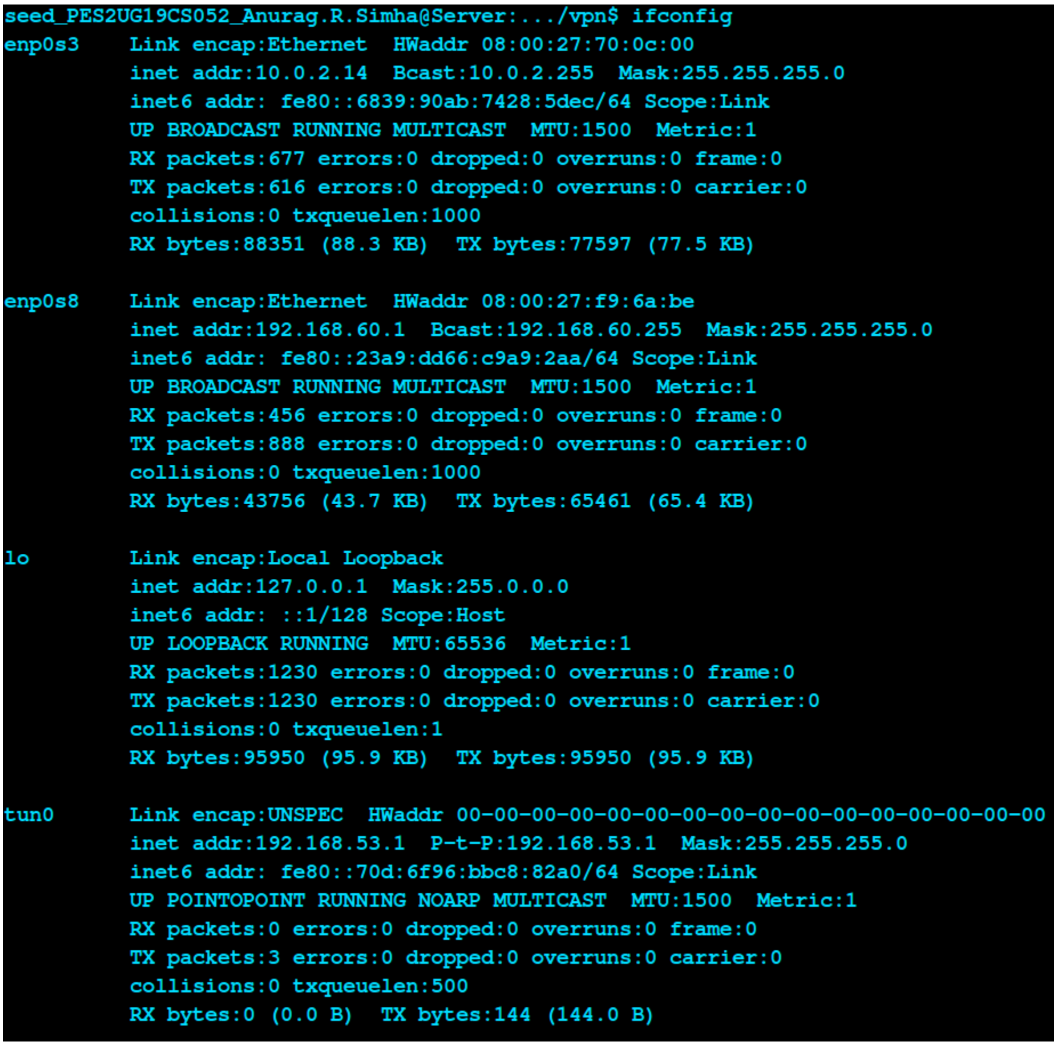
Henceforth, steps 1 and 2 are repeated (only for the server machine).



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo ifconfig tun0 192.168.53.1/24 up
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.60.1    0.0.0.0         UG    100    0        0 enp0s8
0.0.0.0         10.0.2.1        0.0.0.0         UG    101    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 enp0s8
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.60.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s8
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ sudo route add -net 192.168.53.0/24 tun0
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.60.1    0.0.0.0         UG    100    0        0 enp0s8
0.0.0.0         10.0.2.1        0.0.0.0         UG    101    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 enp0s8
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.53.0    0.0.0.0         255.255.255.0   U     0      0        0 tun0
192.168.60.0    0.0.0.0         255.255.255.0   U     100    0        0 enp0s8
```

Fig. 2(B): The tunnel is reconfigured.

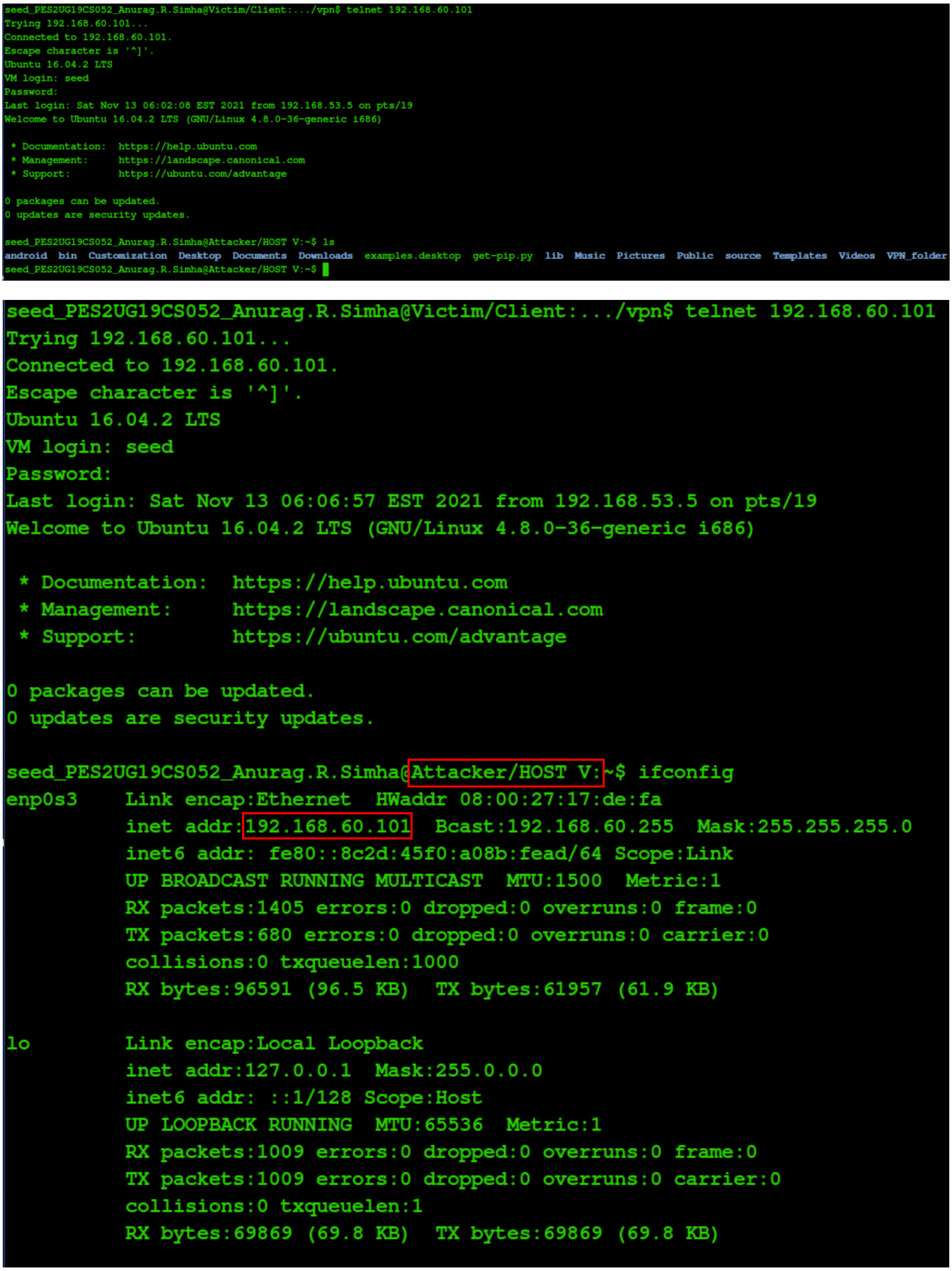
The tunnel interface is once again up and running.



```
seed_PES2UG19CS052_Anurag.R.Simha@Server:.../vpn$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:70:0c:00
          inet addr:10.0.2.14  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6839:90ab:7428:5dec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:677 errors:0 dropped:0 overruns:0 frame:0
          TX packets:616 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88351 (88.3 KB)  TX bytes:77597 (77.5 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:f9:6a:be
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::23a9:dd66:c9a9:2aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:456 errors:0 dropped:0 overruns:0 frame:0
          TX packets:888 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:43756 (43.7 KB)  TX bytes:65461 (65.4 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
          RX packets:1230 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1230 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:95950 (95.9 KB)  TX bytes:95950 (95.9 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::70d:6f96:bbc8:82a0/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:144 (144.0 B)
```

Fig. 2(C): The tunnel interface is finally reconfigured.

Once again, a telnet connection is made.



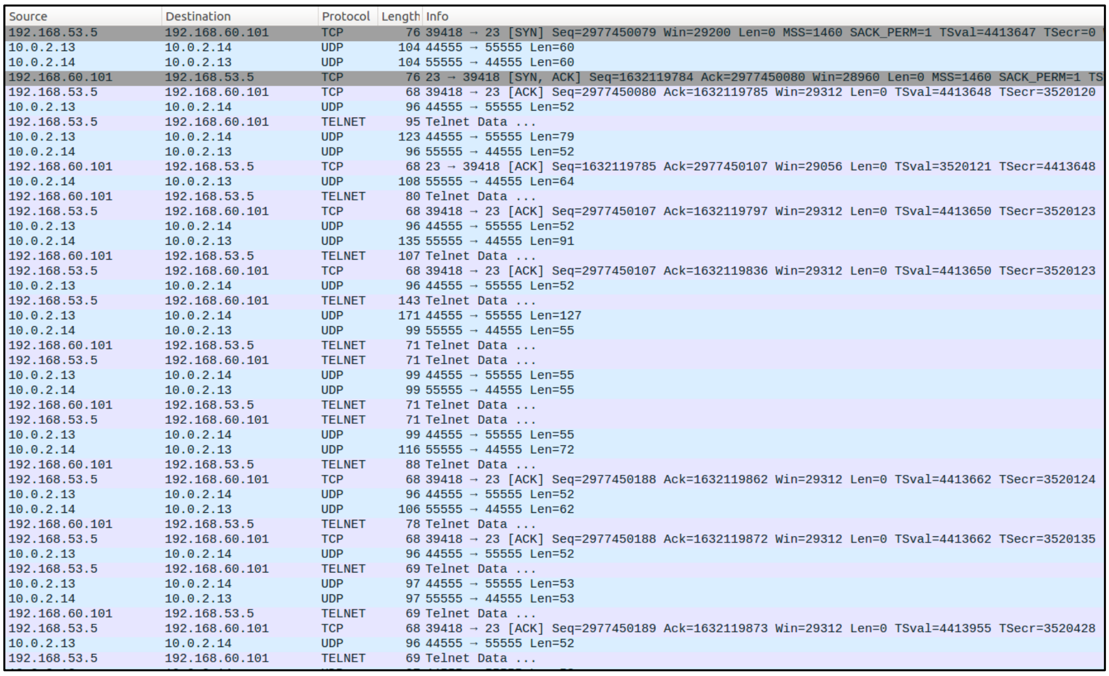
```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Nov 13 06:02:08 EST 2021 from 192.168.53.5 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ls
android  bin  Customization  Desktop  Documents  Downloads  examples.desktop  get-pip.py  lib  Music  Pictures  Public  source  Templates  Videos  VPN_folder
seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ 
```

```
seed_PES2UG19CS052_Anurag.R.Simha@Victim/Client:.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Nov 13 06:06:57 EST 2021 from 192.168.53.5 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

seed_PES2UG19CS052_Anurag.R.Simha@Attacker/HOST V:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:17:de:fa
          inet addr:192.168.60.101  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::8c2d:45f0:a08b:fead/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1405 errors:0 dropped:0 overruns:0 frame:0
          TX packets:680 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:96591 (96.5 KB)  TX bytes:61957 (61.9 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:1009 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1009 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:69869 (69.8 KB)  TX bytes:69869 (69.8 KB)
```

Fig. 2(D): Re-connecting to the tunnel.

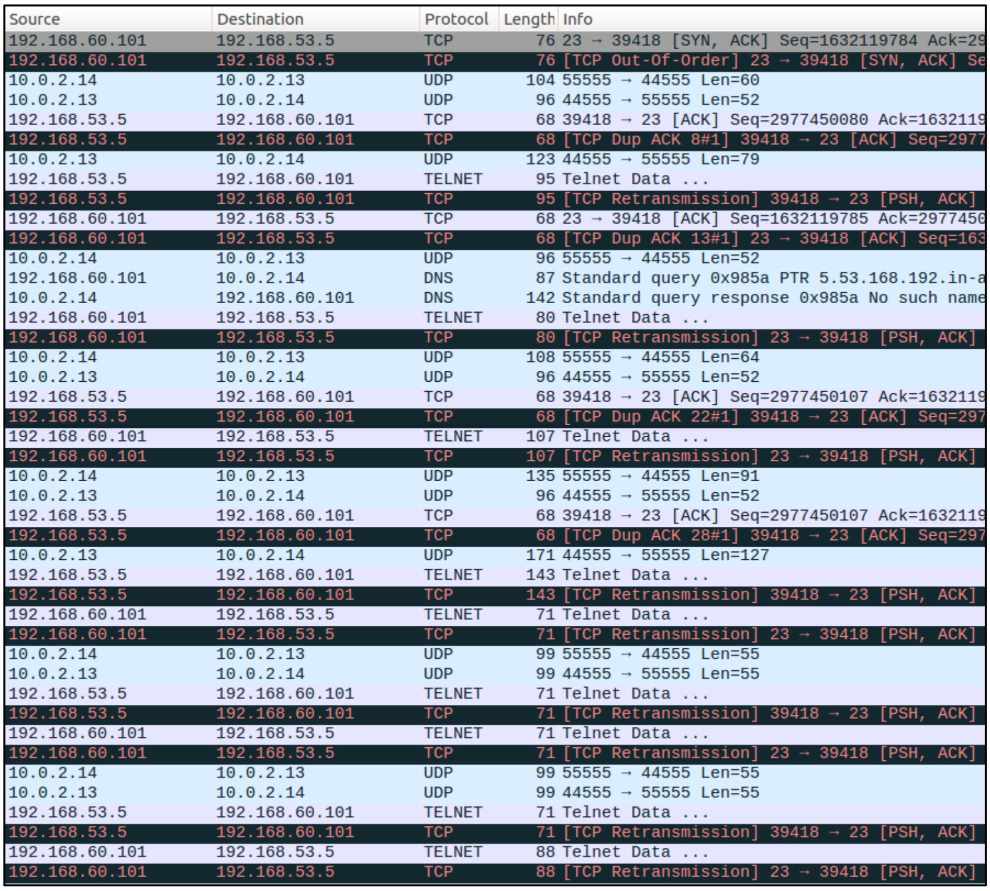
The observations are noted on Wireshark.

On the client machine:



```
Source          Destination     Protocol Length Info
192.168.53.5    192.168.60.101  TCP        76 39418 → 23 [SYN] Seq=2977450079 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=4413647 TSecr=0
10.0.2.13       10.0.2.14       UDP       104 44555 → 55555 Len=60
10.0.2.14       10.0.2.13       UDP       104 55555 → 44555 Len=60
192.168.60.101  192.168.53.5    TCP        76 23 → 39418 [SYN, ACK] Seq=1632119784 Ack=2977450080 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TS
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450080 Ack=1632119785 Win=29312 Len=0 TSval=4413648 TSecr=3520120
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET     95 Telnet Data ...
10.0.2.13       10.0.2.14       UDP       123 44555 → 55555 Len=79
10.0.2.14       10.0.2.13       UDP        96 55555 → 44555 Len=52
192.168.60.101  192.168.53.5    TCP        68 23 → 39418 [ACK] Seq=1632119785 Ack=2977450107 Win=29056 Len=0 TSval=3520121 TSecr=4413648
10.0.2.14       10.0.2.13       UDP       108 55555 → 44555 Len=64
192.168.60.101  192.168.53.5    TELNET     80 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450107 Ack=1632119797 Win=29312 Len=0 TSval=4413650 TSecr=3520123
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
10.0.2.14       10.0.2.13       UDP       135 55555 → 44555 Len=91
192.168.60.101  192.168.53.5    TELNET    107 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450107 Ack=1632119836 Win=29312 Len=0 TSval=4413650 TSecr=3520123
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET    143 Telnet Data ...
10.0.2.13       10.0.2.14       UDP       171 44555 → 55555 Len=127
10.0.2.14       10.0.2.13       UDP        99 55555 → 44555 Len=55
192.168.60.101  192.168.53.5    TELNET     71 Telnet Data ...
192.168.53.5    192.168.60.101  TELNET     71 Telnet Data ...
10.0.2.13       10.0.2.14       UDP        99 44555 → 55555 Len=55
10.0.2.14       10.0.2.13       UDP        99 55555 → 44555 Len=55
192.168.60.101  192.168.53.5    TELNET     71 Telnet Data ...
192.168.53.5    192.168.60.101  TELNET     71 Telnet Data ...
10.0.2.13       10.0.2.14       UDP        99 44555 → 55555 Len=55
10.0.2.14       10.0.2.13       UDP       116 55555 → 44555 Len=72
192.168.60.101  192.168.53.5    TELNET     88 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450188 Ack=1632119862 Win=29312 Len=0 TSval=4413662 TSecr=3520124
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
10.0.2.14       10.0.2.13       UDP       106 55555 → 44555 Len=62
192.168.60.101  192.168.53.5    TELNET     78 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450188 Ack=1632119872 Win=29312 Len=0 TSval=4413662 TSecr=3520135
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET     69 Telnet Data ...
10.0.2.13       10.0.2.14       UDP        97 44555 → 55555 Len=53
10.0.2.14       10.0.2.13       UDP        97 55555 → 44555 Len=53
192.168.60.101  192.168.53.5    TELNET     69 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450189 Ack=1632119873 Win=29312 Len=0 TSval=4413955 TSecr=3520428
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TELNET     69 Telnet Data ...
```

Fig. 2(E): The Wireshark results (on VPN client).



```
Source          Destination     Protocol Length Info
192.168.60.101  192.168.53.5    TCP        76 23 → 39418 [SYN, ACK] Seq=1632119784 Ack=29
192.168.60.101  192.168.53.5    TCP        76 [TCP Out-Of-Order] 23 → 39418 [SYN, ACK] Se
10.0.2.14       10.0.2.13       UDP       104 55555 → 44555 Len=60
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450080 Ack=1632119
192.168.53.5    192.168.60.101  TCP        68 [TCP Dup ACK 8#1] 39418 → 23 [ACK] Seq=2977
10.0.2.13       10.0.2.14       UDP       123 44555 → 55555 Len=79
192.168.53.5    192.168.60.101  TELNET     95 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        95 [TCP Retransmission] 39418 → 23 [PSH, ACK]
192.168.60.101  192.168.53.5    TCP        68 23 → 39418 [ACK] Seq=1632119785 Ack=2977450
192.168.60.101  192.168.53.5    TCP        68 [TCP Dup ACK 13#1] 23 → 39418 [ACK] Seq=163
10.0.2.14       10.0.2.13       UDP        96 55555 → 44555 Len=52
192.168.60.101  10.0.2.14       DNS        87 Standard query 0x985a PTR 5.53.168.192.in-a
10.0.2.14       192.168.60.101  DNS       142 Standard query response 0x985a No such name
192.168.60.101  192.168.53.5    TELNET     80 Telnet Data ...
192.168.60.101  192.168.53.5    TCP        80 [TCP Retransmission] 23 → 39418 [PSH, ACK]
10.0.2.14       10.0.2.13       UDP       108 55555 → 44555 Len=64
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450107 Ack=1632119
192.168.53.5    192.168.60.101  TCP        68 [TCP Dup ACK 22#1] 39418 → 23 [ACK] Seq=297
192.168.60.101  192.168.53.5    TELNET    107 Telnet Data ...
192.168.60.101  192.168.53.5    TCP       107 [TCP Retransmission] 23 → 39418 [PSH, ACK]
10.0.2.14       10.0.2.13       UDP       135 55555 → 44555 Len=91
10.0.2.13       10.0.2.14       UDP        96 44555 → 55555 Len=52
192.168.53.5    192.168.60.101  TCP        68 39418 → 23 [ACK] Seq=2977450107 Ack=1632119
192.168.53.5    192.168.60.101  TCP        68 [TCP Dup ACK 28#1] 39418 → 23 [ACK] Seq=297
10.0.2.13       10.0.2.14       UDP       171 44555 → 55555 Len=127
192.168.53.5    192.168.60.101  TELNET    143 Telnet Data ...
192.168.53.5    192.168.60.101  TCP       143 [TCP Retransmission] 39418 → 23 [PSH, ACK]
192.168.60.101  192.168.53.5    TELNET     71 Telnet Data ...
192.168.60.101  192.168.53.5    TCP        71 [TCP Retransmission] 23 → 39418 [PSH, ACK]
10.0.2.14       10.0.2.13       UDP        99 55555 → 44555 Len=55
10.0.2.13       10.0.2.14       UDP        99 44555 → 55555 Len=55
192.168.53.5    192.168.60.101  TELNET     71 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        71 [TCP Retransmission] 39418 → 23 [PSH, ACK]
192.168.60.101  192.168.53.5    TELNET     71 Telnet Data ...
192.168.60.101  192.168.53.5    TCP        71 [TCP Retransmission] 23 → 39418 [PSH, ACK]
10.0.2.14       10.0.2.13       UDP        99 55555 → 44555 Len=55
10.0.2.13       10.0.2.14       UDP        99 44555 → 55555 Len=55
192.168.53.5    192.168.60.101  TELNET     71 Telnet Data ...
192.168.53.5    192.168.60.101  TCP        71 [TCP Retransmission] 39418 → 23 [PSH, ACK]
192.168.60.101  192.168.53.5    TELNET     88 Telnet Data ...
192.168.60.101  192.168.53.5    TCP        88 [TCP Retransmission] 23 → 39418 [PSH, ACK]
```

Fig. 2(F): The Wireshark results (on VPN server).

It's observed that the Wireshark packet capture results on the client machine are unchanged and resemble the outcome obtained in figure 2(p). But there are a plethora of retransmissions and duplicate packets transferred in the tunnel. Henceforth, the results on the server machine show retransmitted and duplicate packets.

Without repeating steps 1 and 2, the reconnection is insuperable.

****