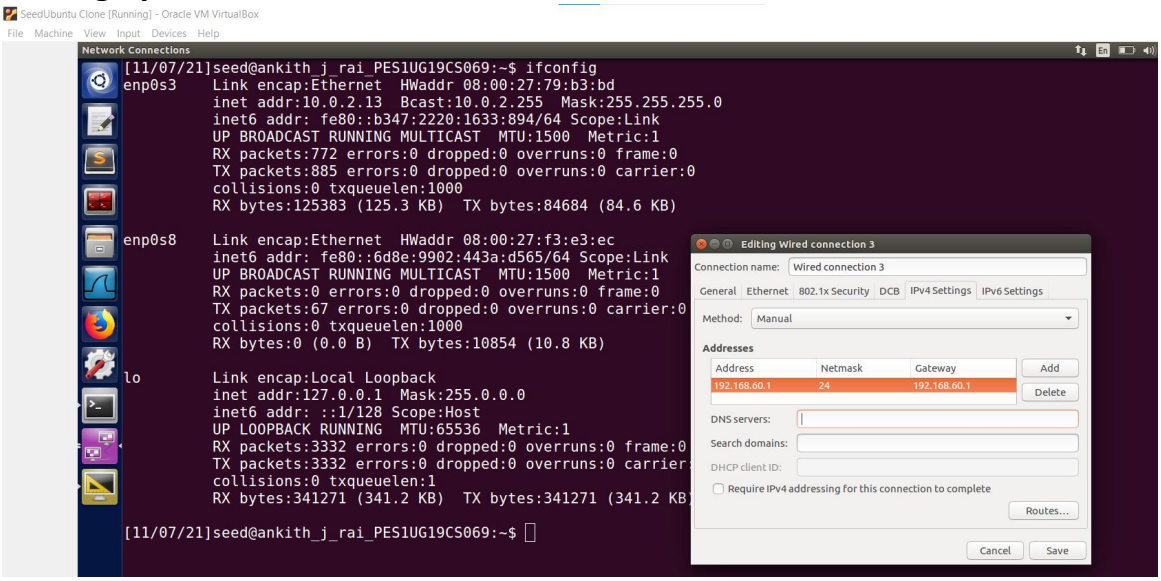
Virtual Private Network Lab

Name : Ankith J Rai
SRN : PES1UG19CS069
SEC : B

Machine	IP address
VPN Client	10.0.2.12
VPN Server	10.0.2.13(NAT network) 192.168.60.1(Internal network)
Host V	192.168.60.101(Internal network)

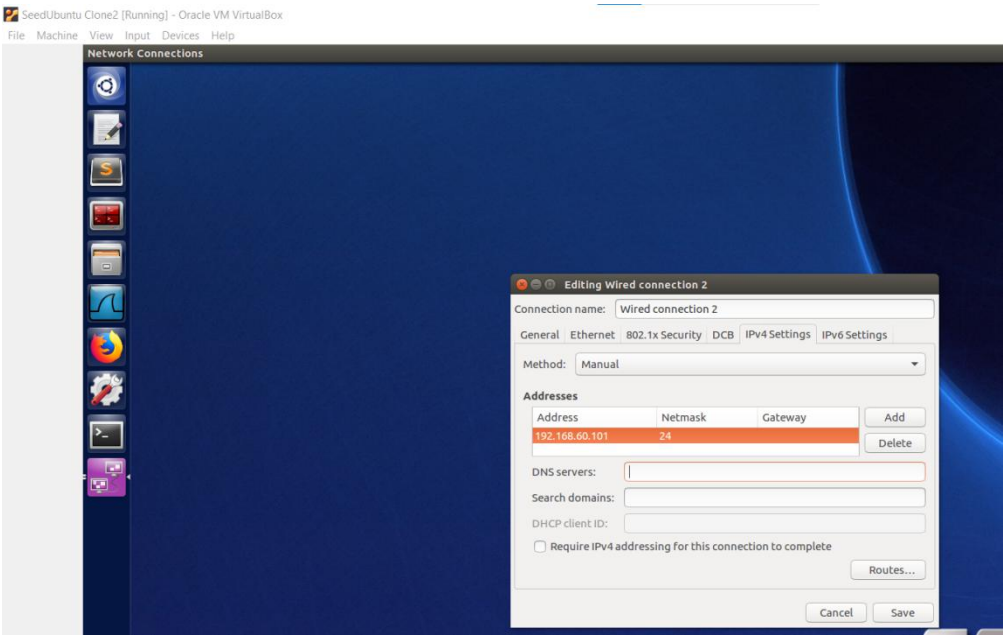
Task 1: VM Setup

Setting up for VM server:



We can see that we have VM Server has been configured.

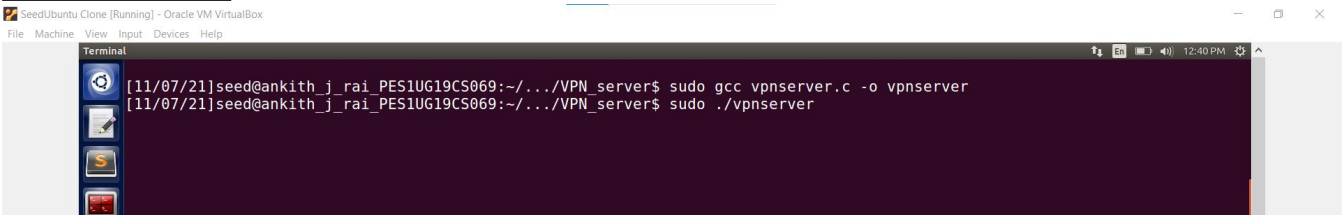
Setting up for Host V:



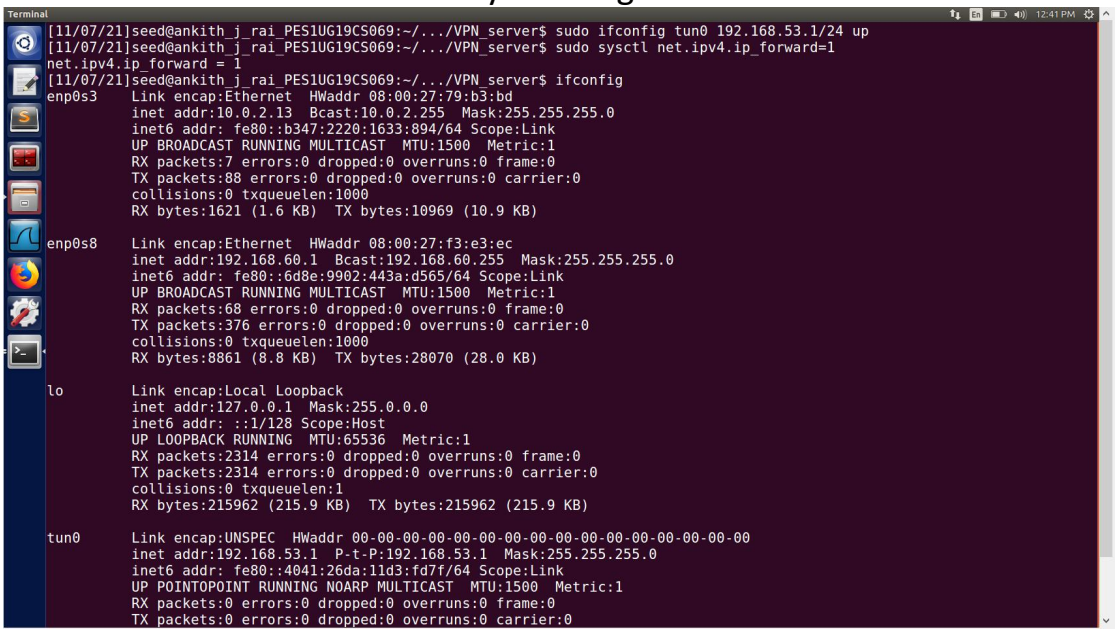
We can see that we have Host V has been configured.

Task 2: Creating a VPN Tunnel using TUN/TA:

Step 1: Run VPN server and set it's IP address of the interface – (Run on VPNServer VM



We can see that the vpnserver program is running.By running this tun0 interface is created but it is not yet configured.

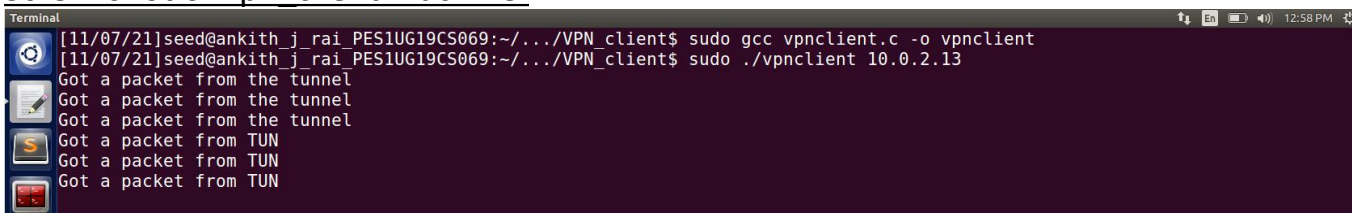


From the above screenshot we can see that a new interface tun0 has been configured and it's ip address is 192.168.53.1 .

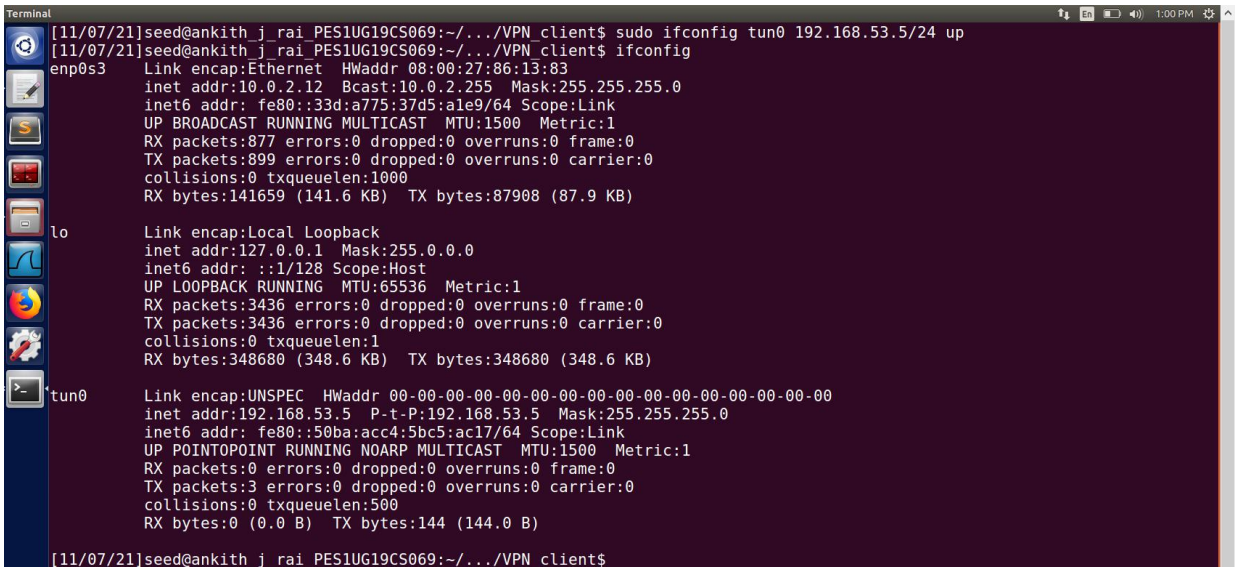
From the above screenshot we can also conclude that enp0s3 is the NAT network interface and enp0s8 is the internal network interface.

Step 2: Run VPN Client and set IP address of the interface - (Run on VPNClient VM)

Screnshot of vpn_client machine:

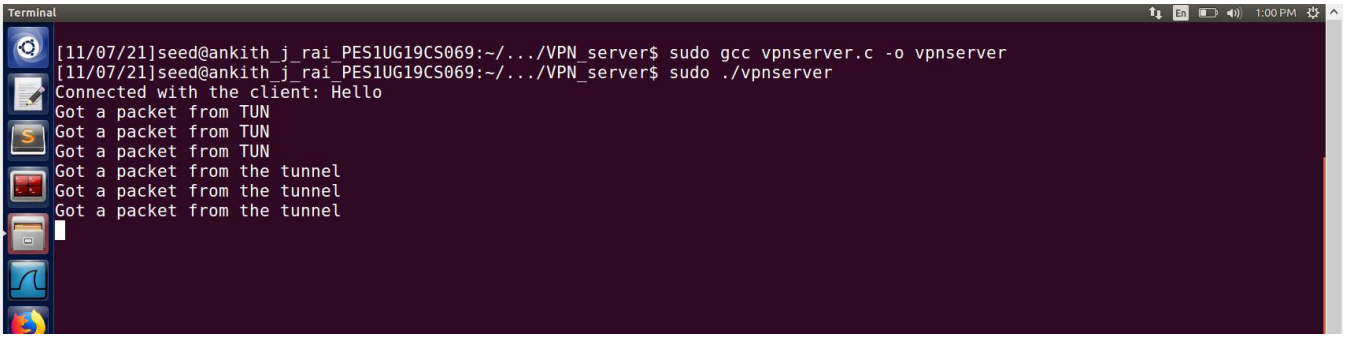


We can see from above screenshot that vpnclient.c program is running and we can see that the tunnel is created between the VPN client and VPN server. Due to the running of vpcclient.c program a new interface called tun0 has been created but it is not yet configured.



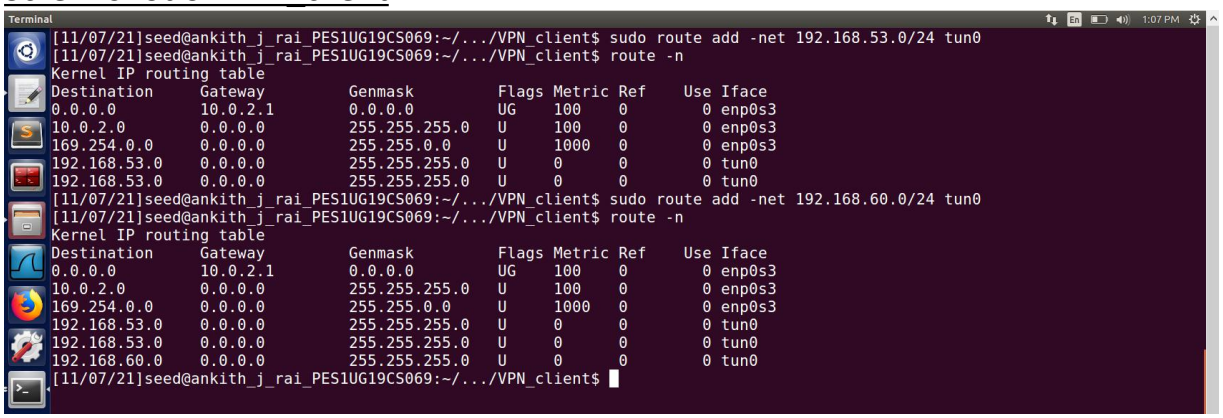
We can see that the new interface tun0 which got created when the vpnclient.c was run has been configured and it's ip address is 192.168.53.5 .

Screnshot of vpn_server machine:



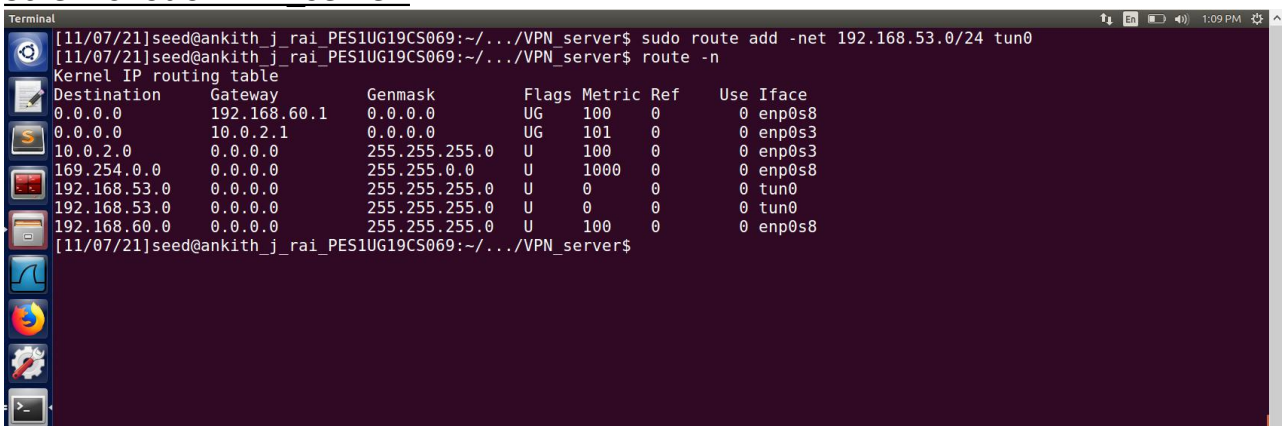
Step 3: Set up routing on Client and Server VMs

Screnshot of VPN_client



The routes have been successfully added and the now we can route the packets from local tunnel to internal network.

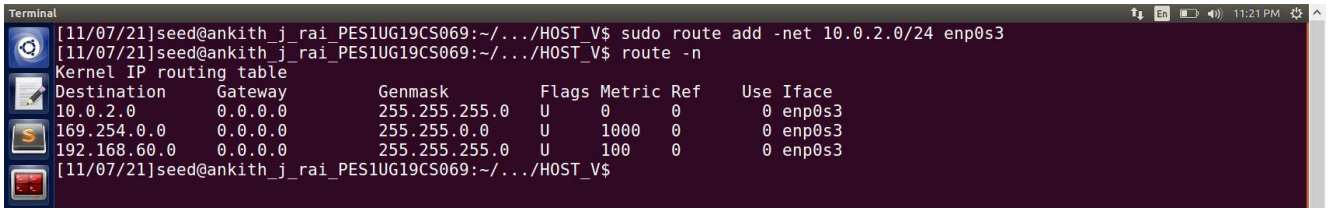
Screnshot of VPN_server:

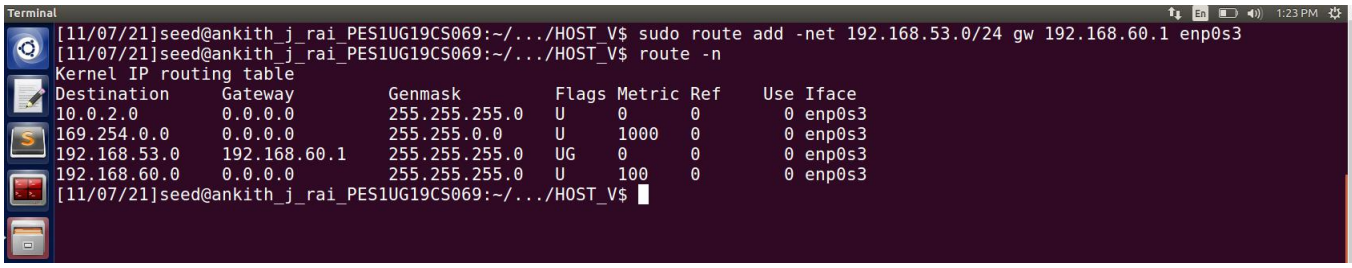


The routes have been successfully added on the VPN server and the now we can route the packets from local tunnel to internal network.

Step 4: Set up routing on HOST V

Screnshot of Host_V:

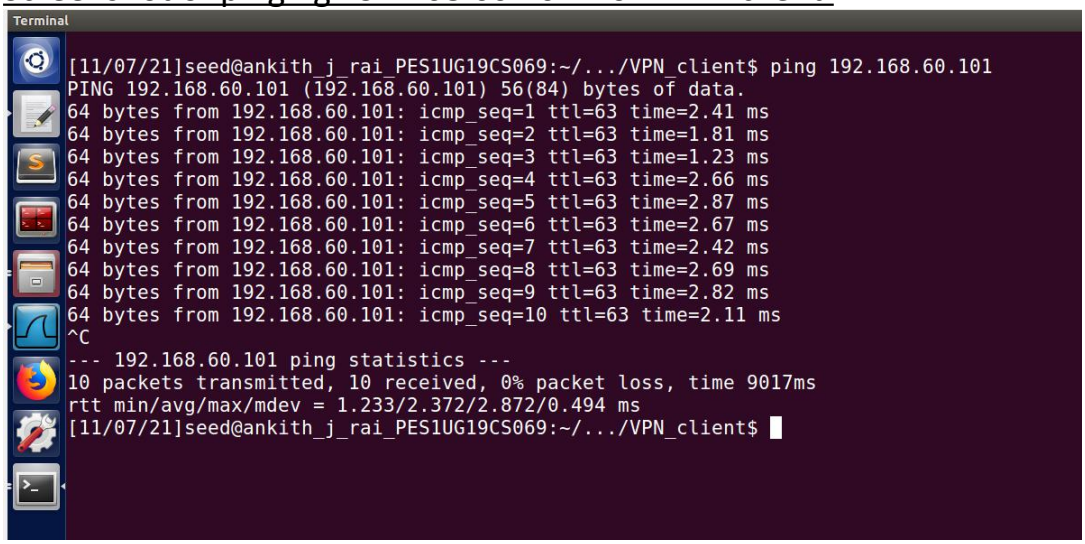




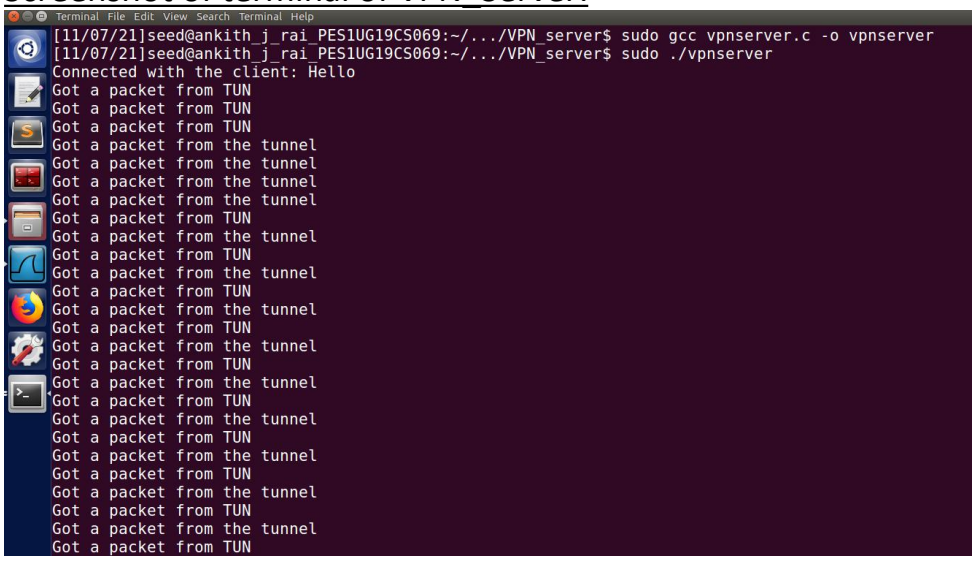
The routes have been successfully add to the routing tables of Host V.

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

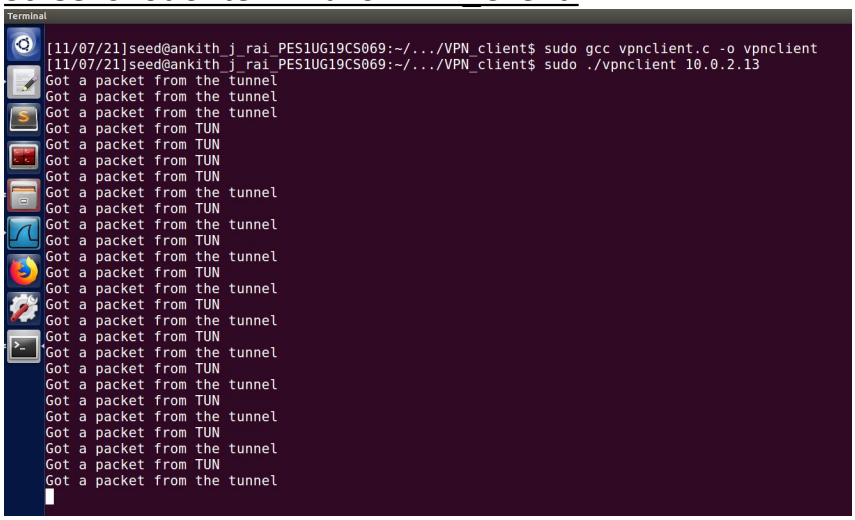
Screenshot of pinging 192.168.60.101 from VPN client:



Screenshot of terminal of VPN_Server:

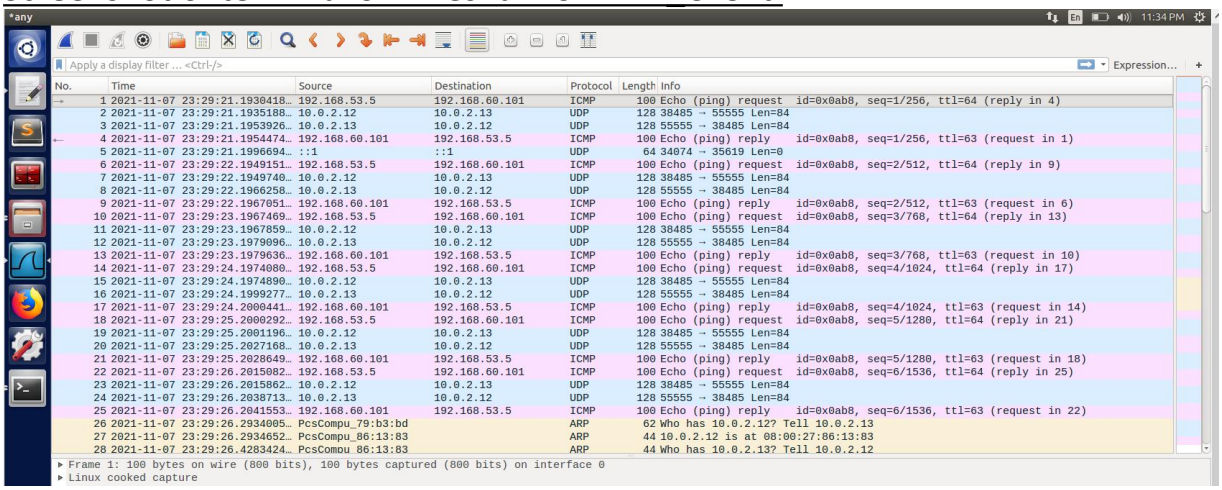


Screenshot of terminal of VPN_Client:

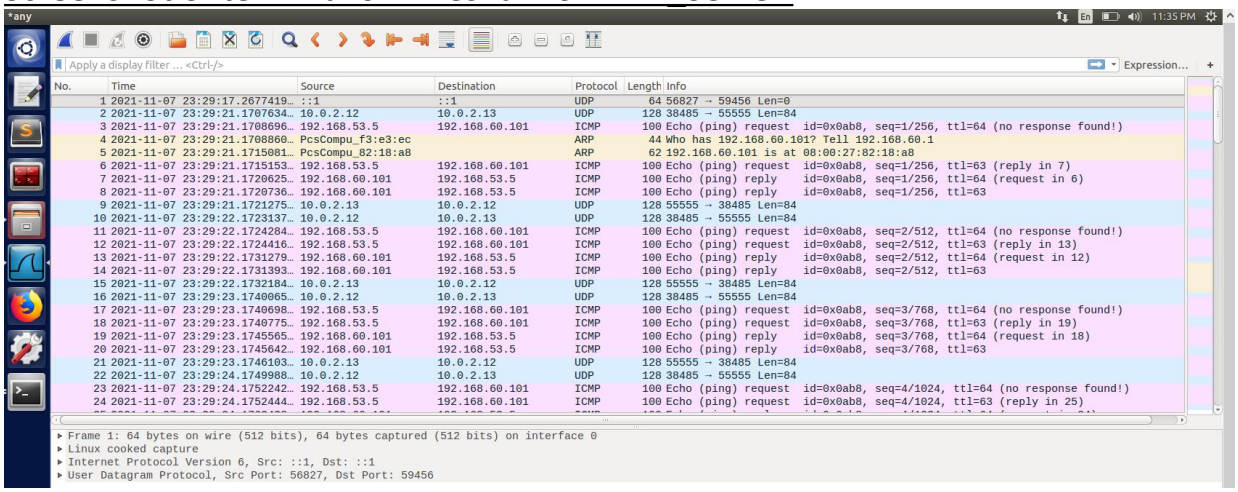


From the above screenshot's we can see that the tunnel has been established successfully and the packets are sent through the tunnel.

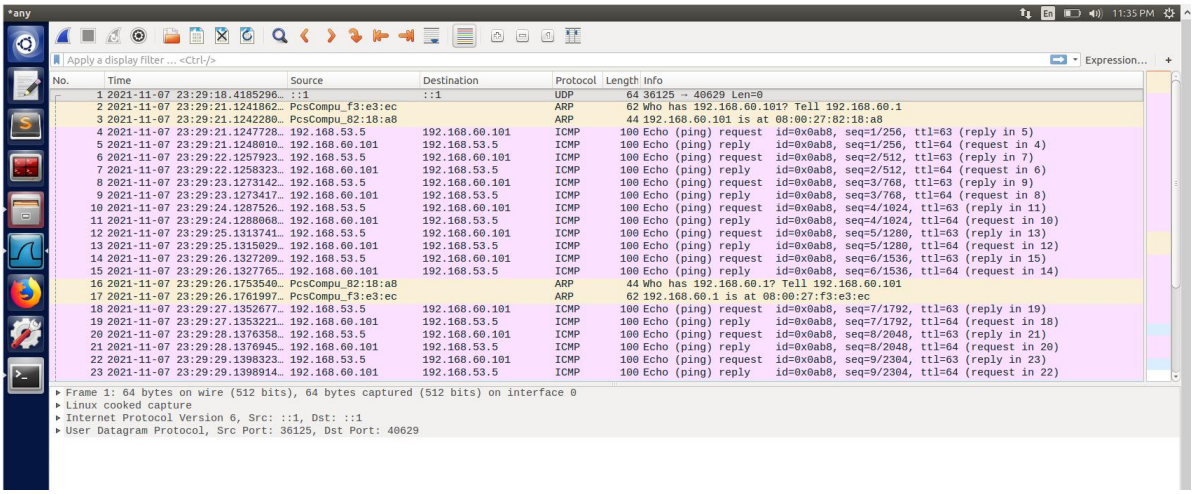
Screenshot of terminal of wireshark on VPN_Client:



Screenshot of terminal of wireshark on VPN_Server:



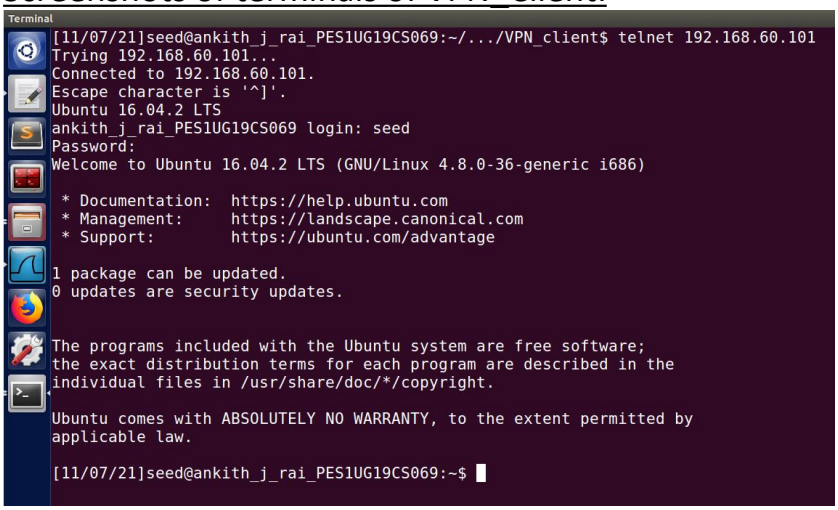
Screenshot of terminal of wireshark on Host V:



From the above screenshots we can tell that VPN client's ping request goes to VPN server and VPN server it is redirected to internal network and from there it is sent to Host V.Host V receives the packet and send back the reply to VPN server and the VPN server machine sends back the packet to VPN Client through the tunnel.

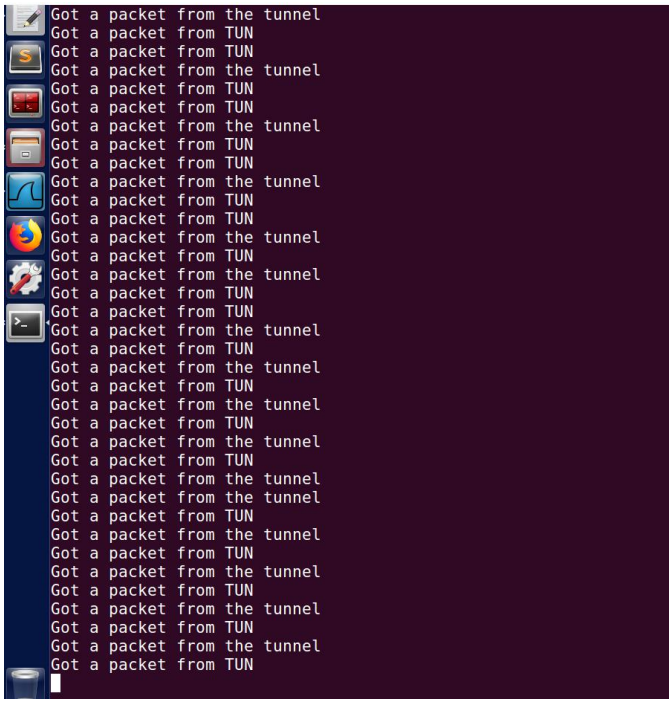
Telnet connection:

Screenshots of terminals of VPN_Client:



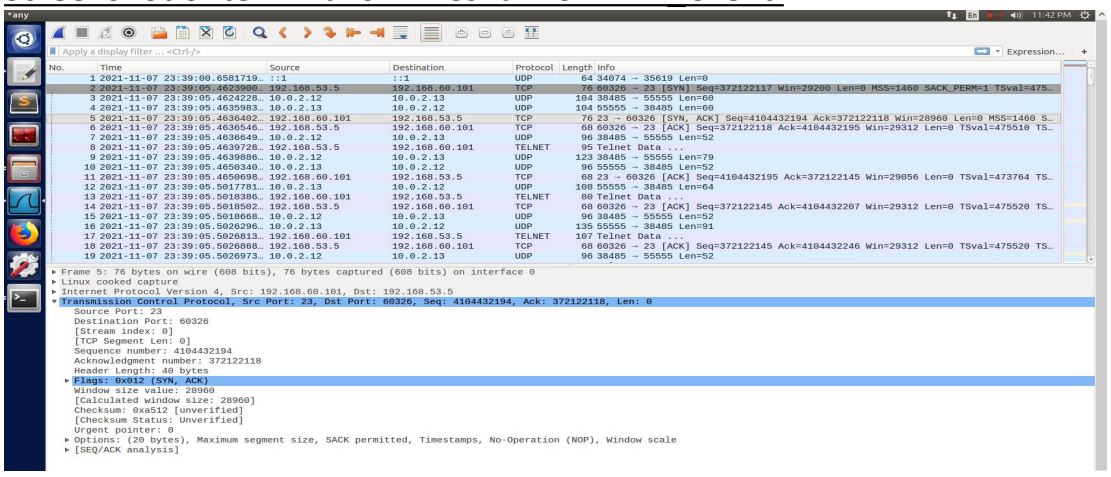
The telnet connection has been successfully established.

Screenshot of VPN client:



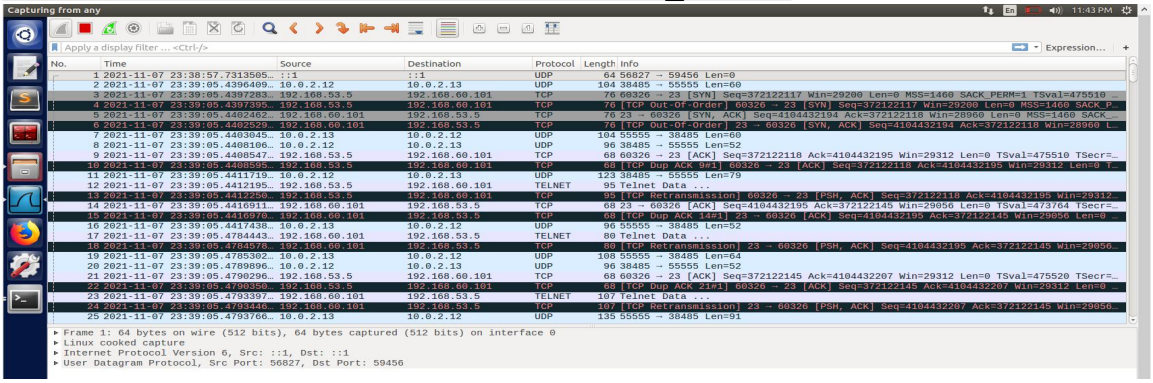
We can see that the packets have been successfully sent through the VPN tunnel.

Screenshot of terminal of wireshark on VPN_Client:

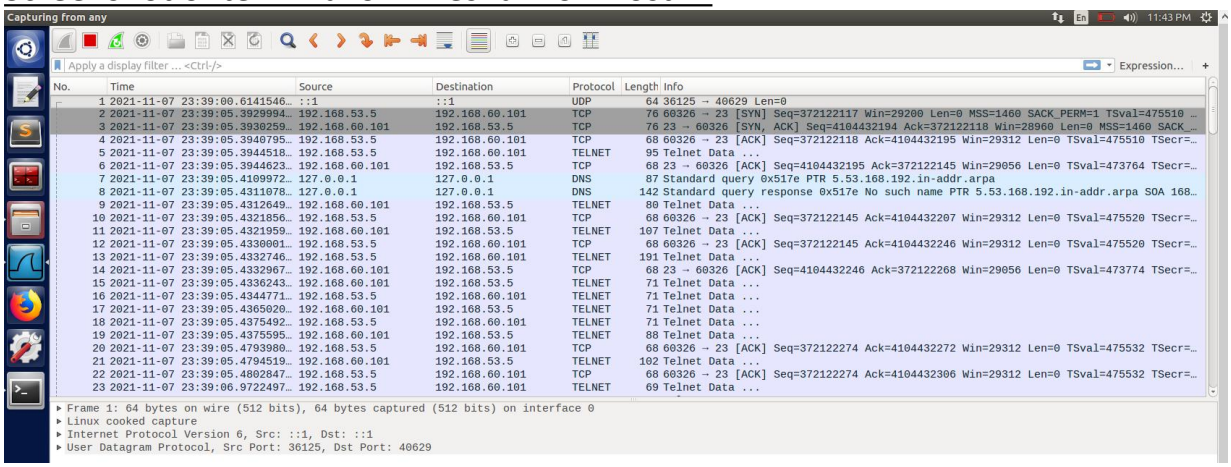


Here we observe that it is TCP instead of ICMP this is because telnet uses TCP.

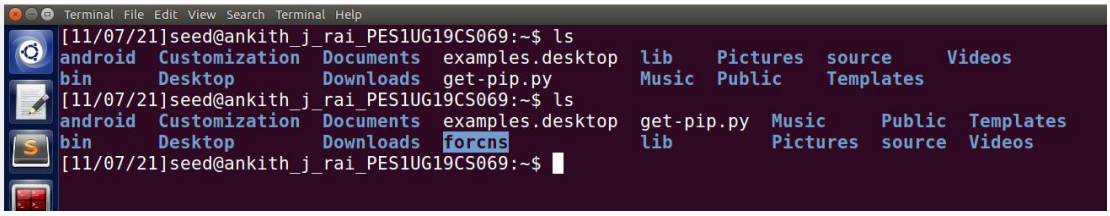
Screenshot of terminal of wireshark on VPN_Server:



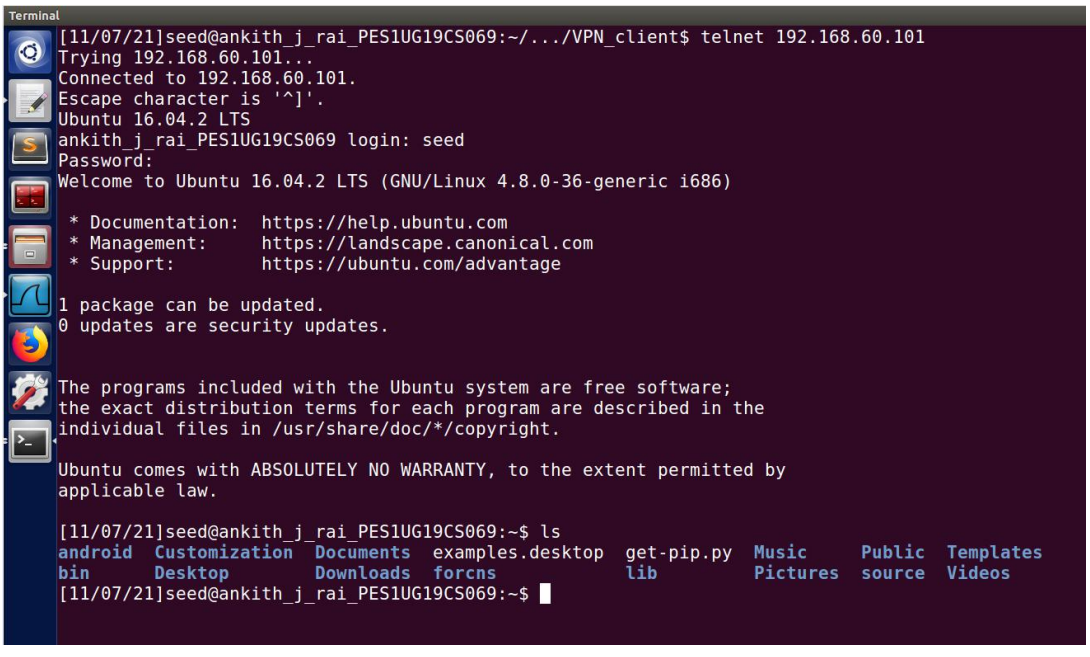
Screenshot of terminal of wireshark on Host V:



Now using ls command



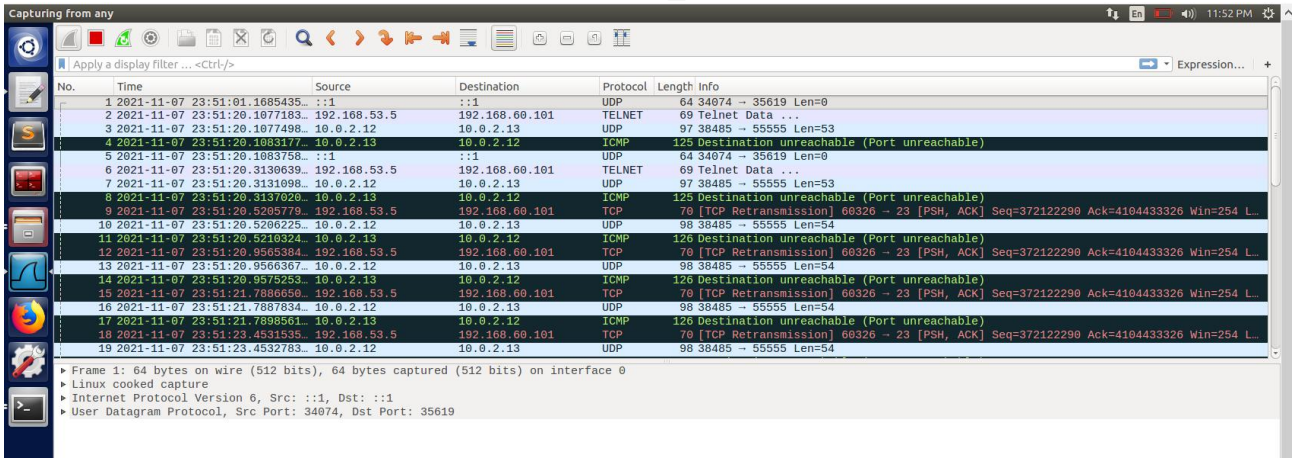
We can see that forcns has been created on Host V.



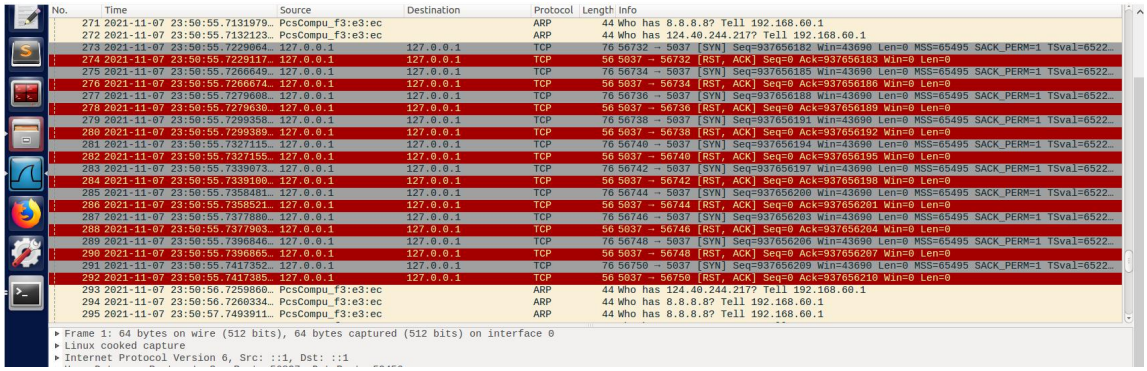
From above screenshot we can see that forcns is also visible on the telnet connection.

Step 6: Tunnel-Breaking Test:

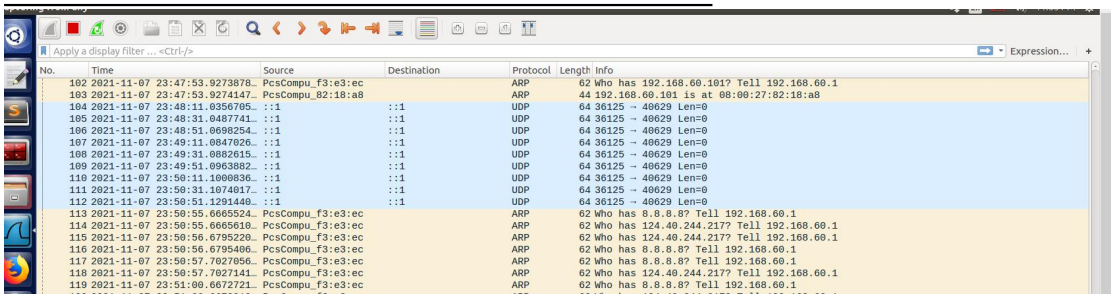
Screenshot of terminal of wireshark on VPN_Client:



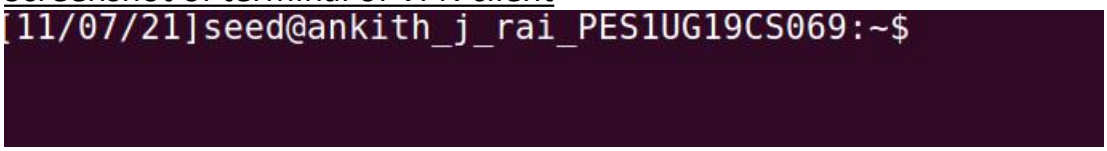
Screenshot of terminal of wireshark on VPN_Server:



Screenshot of terminal of wireshark on Host V:



Screenshot of terminal of VPN client



We can see that we are not able to type anything on vpn client machine after breaking the tunnel.But from wireshark screenshot's we can see that the tcp connection's are still there, hence everything we type are stored in the buffer.

On restarting the vpnserver.c program we can see that the connection has been already lost.

Screenshot of terminal of wireshark on VPN Server: