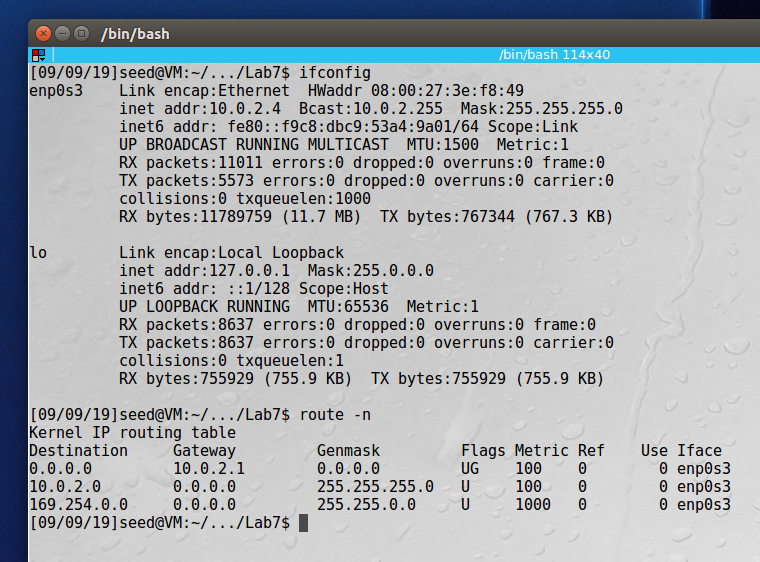
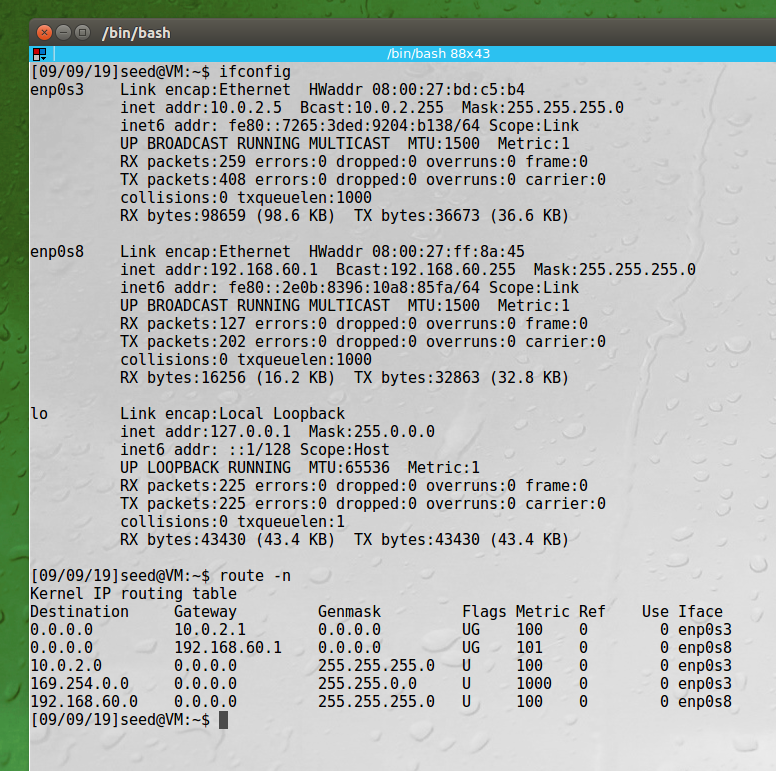
Internet Security - Lab 7 Report

# Task 1: VM Setup

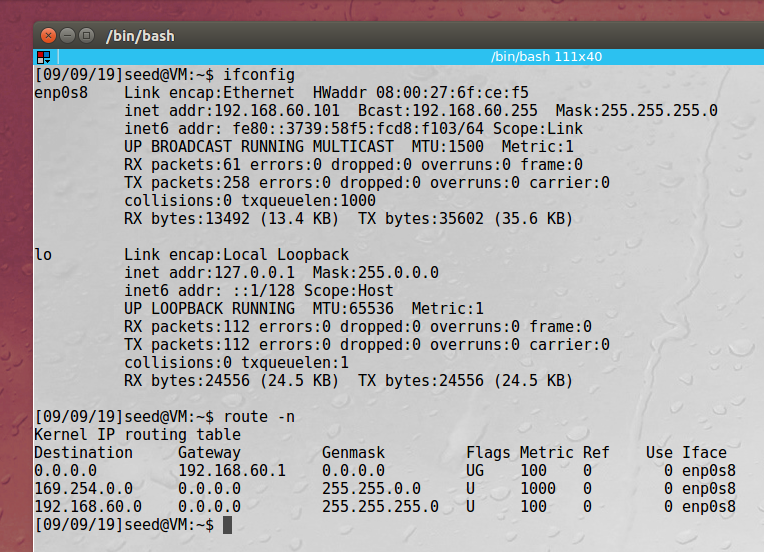
Host U, on NAT Network as 10.0.2.4:



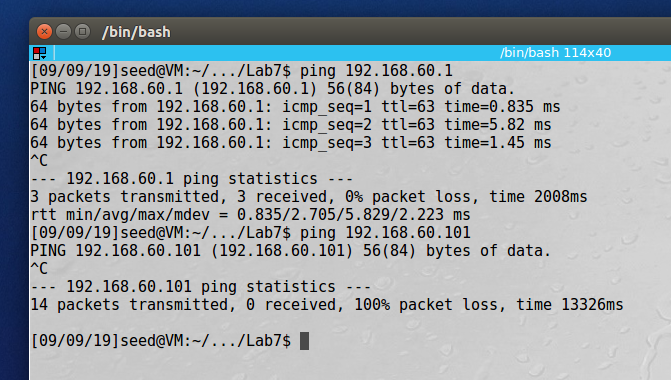
VPN Server/Gateway, on NAT Network at 10.0.2.5 and Internal network as 192.168.60.1:



Host V is on internal network only, at 192.168.60.101:



From Host U we can ping VPN Server/Gateway but not Host V:

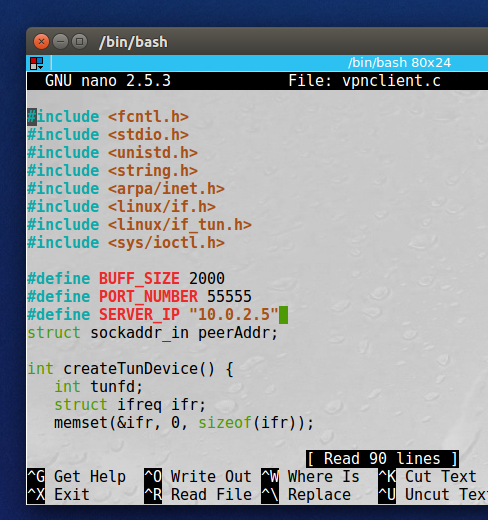


**Observation**: In this task we set up the network in preparation for the VPN tunnel. We have three machines running: Host U (10.0.2.4), VPN Server/Gateway (10.0.2.5/192.168.60.1) and Host V (192.168.60.101). Currently Host U and Host V cannot communicate with each other as seen in the ping messages above.

**Explanation**: The VPN Server/Gateway machine has two network interfaces configured so that we will be able to connect Host U to Host V after we establish our VPN tunnel. We are simulating that the hosts and the VPN server are connected over the internet by keeping them on separate networks.

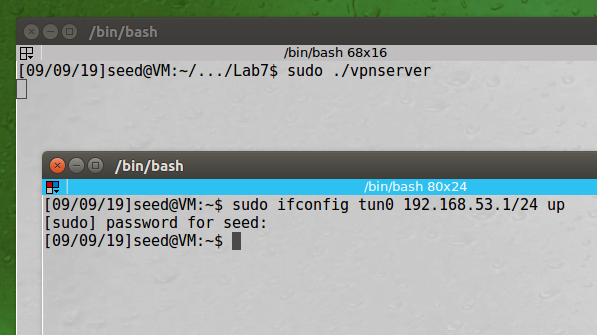
# Task 2: Creating a VPN Tunnel using TUN/TAP

Before running the client/server applications we must update the IP address of our VPN Server in the program:

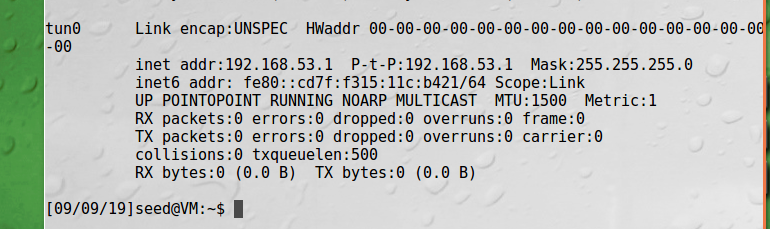


## Step 1: Run VPN Server

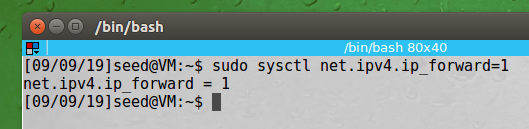
First, we run the VPN server program on our Server VM, then configure tun0 which will be our VPN interface:



We can see our new tun0 interface in ifconfig:

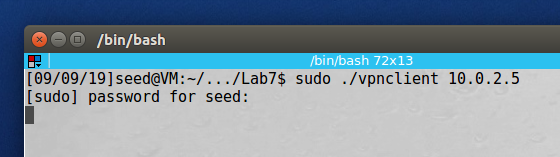


Next we enable forwarding since our server will act as a gateway:

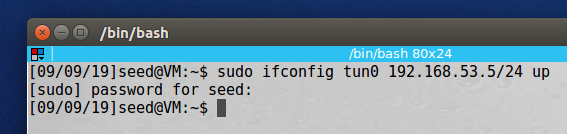


## Step 2: Run the VPN Client

On Host U, we run the VPN Client program and point it to our server at 10.0.2.5:



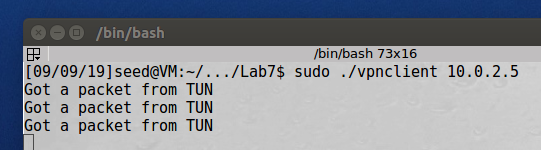
Then we configure our tun0 interface that will be used in the VPN:



We can see on our Server machine that we have connected by the “Hello” message:

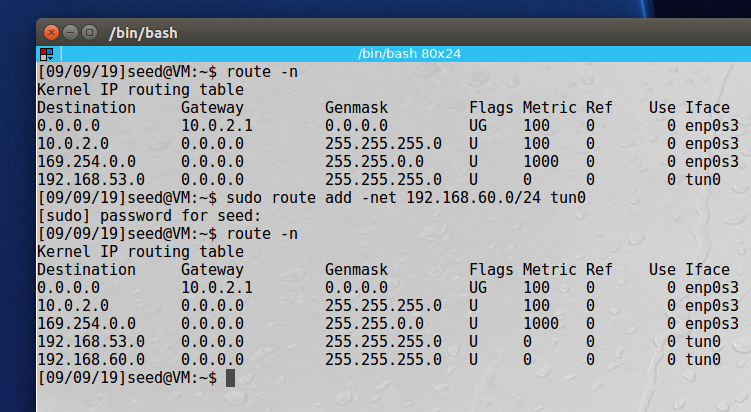


And we can see on our Host U, that we have received packets on tun0:

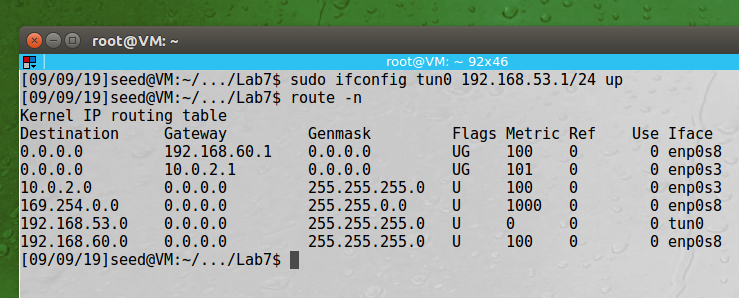


## Step 3: Routing on the Client and Server VMs

In this screenshot from Host U, we first look at the routing table, then we add the route to 192.168.60.0/24 network via our tun0 interface. We again look at our routing table and see the new route:

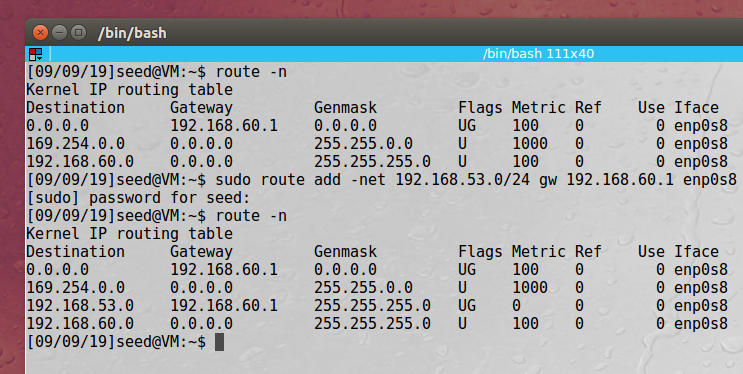


We can also see on our sever that our route is set up for the 192.168.60.0/24 network:



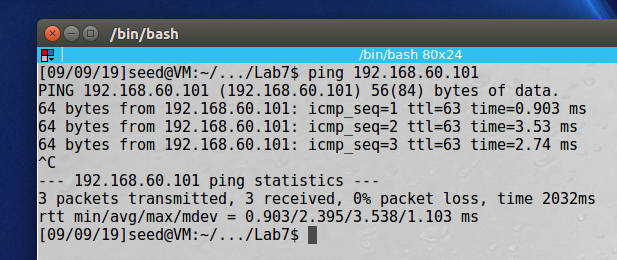
## Step 4: Set Up Routing Host V

In the screenshot below we first see the Host V routing table, then we add the route for our VPN, 192.168.53.0/24 network:

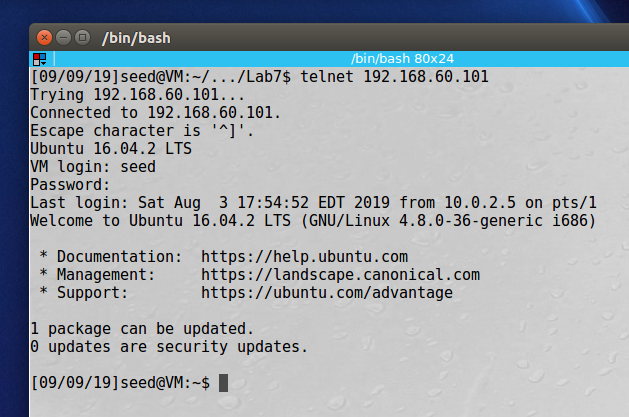


## Step 5: Test the VPN

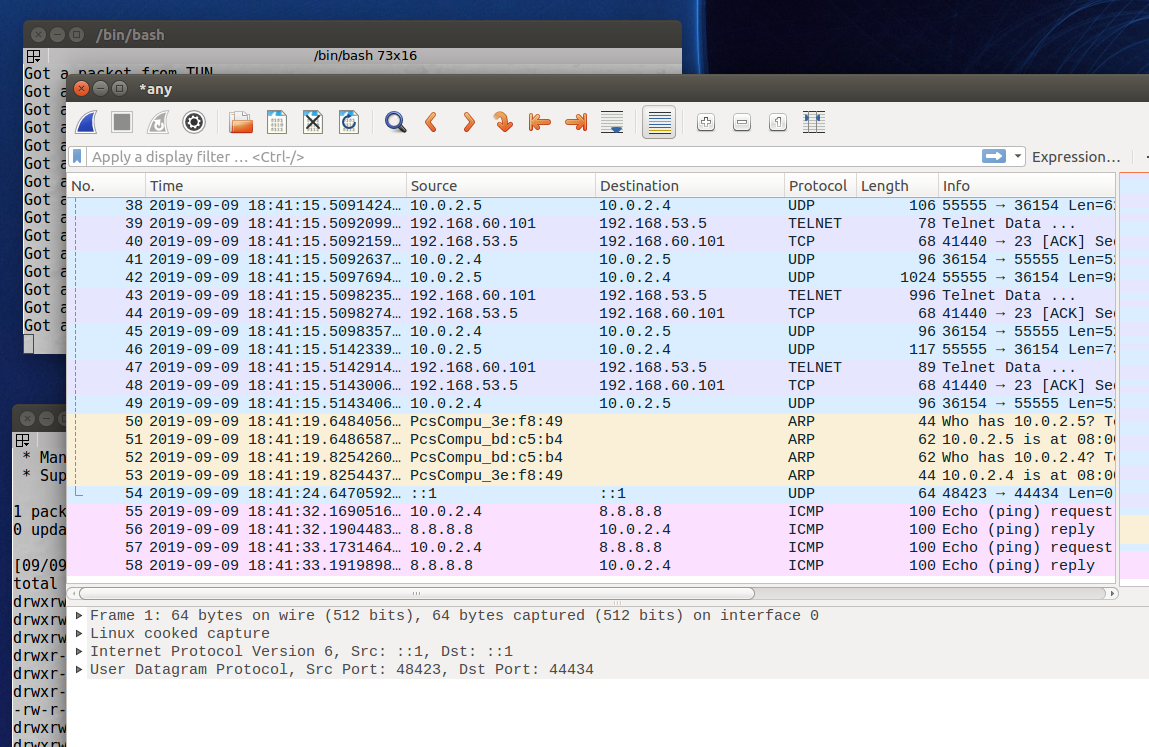
Now from Host U we test the connection by pinging Host V, and see we are connected:



And now we can connect via telnet as well:

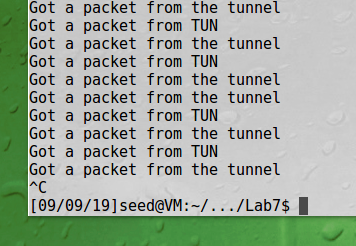
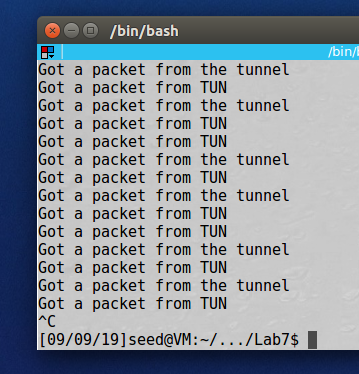


Looking at Wireshark, we can see that traffic routed between Host U and Host V uses the 192.168.53.0 VPN source, and the other traffic like when I ping 8.8.8.8 doesn’t go through the VPN interface:

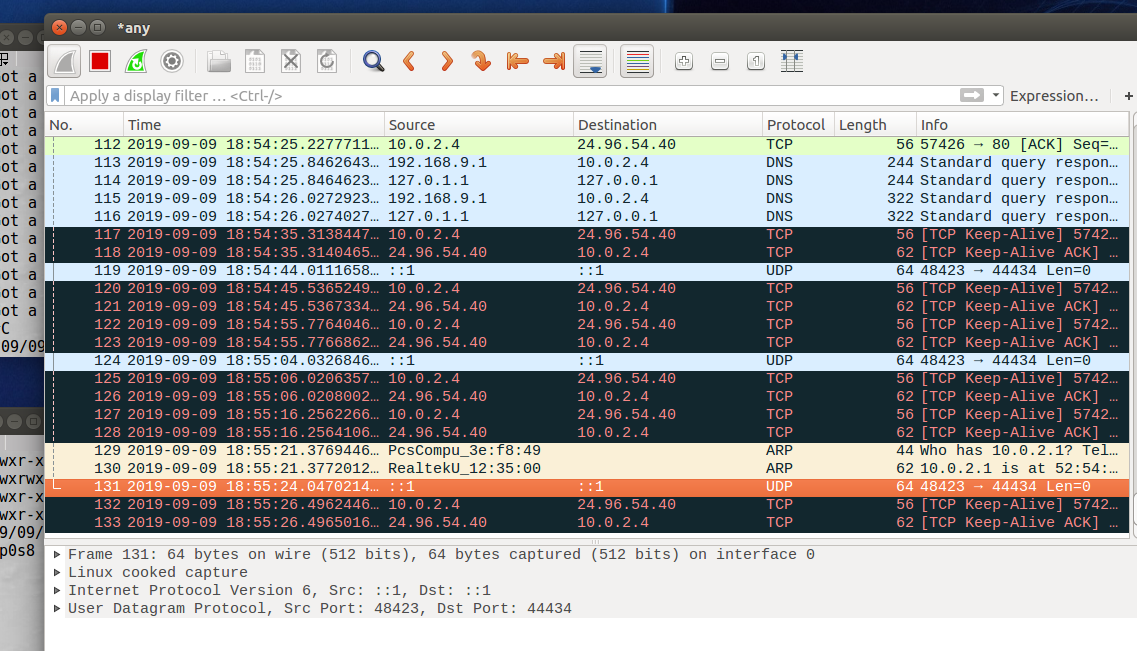


## Step 6: Tunnel-Breaking Test

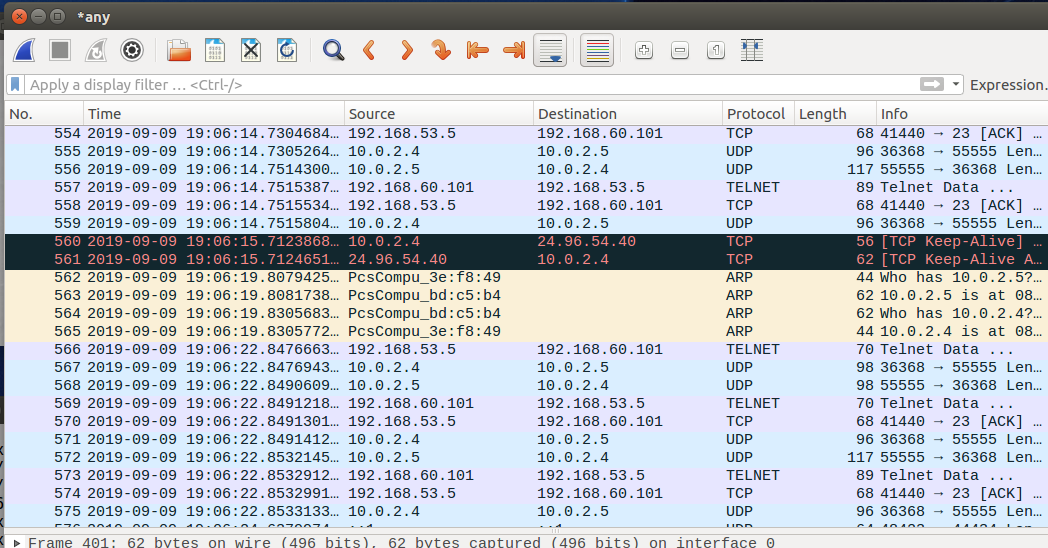
While Telnet is still active, we kill our client and server VPN programs:

After breaking the tunnel, we attempt to type commands in the Telent window, and we can see that the packets are not delivered but being put in a buffer:



Next we restart our VPN on Host U and Server, reconfigure the tun0 interfaces, and routing tables, and we see our buffered Telnet commands are delivered and the buffer is cleared:



**Explanation**: In this task we setup our VPN tunnel, tested it, broke the tunnel, and reestablished it, allowing us to communicate between Host U and Host V. We started by configuring our Server, we first run the VPN server program, then we configure the tun0 interface, and allow traffic forwarding, so that it can act as a gateway. The second step was to run the VPN Client program on Host U, and again we had to configure traffic destined to the 192.168.60.0/24 network to use our tun0 interface. Next we configure the Host V; we need to add the route so when it responds to Host U it knows to go through the VPN server. Then we can ping and telnet from Host U to Host V. As an experiment we tried to break the VPN to observe what happened with telnet packets. We saw that they were buffered and when we reestablished the VPN the packets that were buffered were sent.