# COMPUTER NETWORKS SECURITY LABORATORY

# LAB 2


# BY: RAJDEEP SENGUPTA

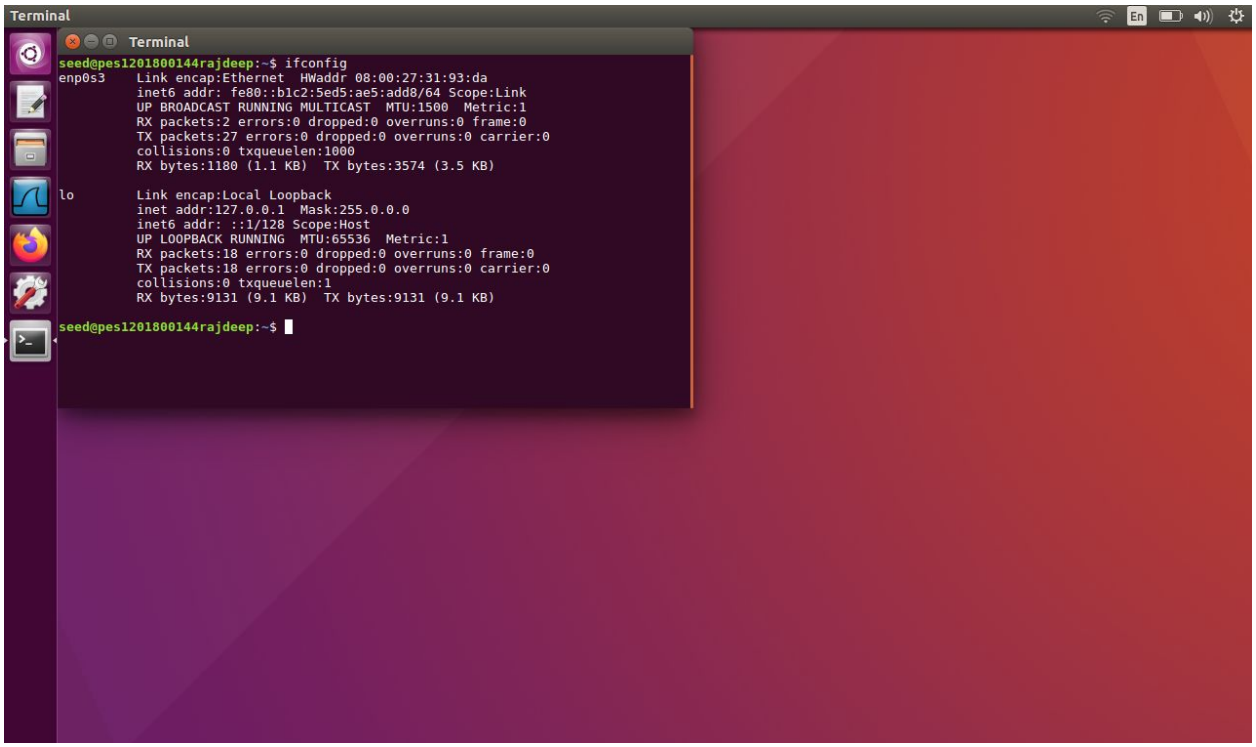
# SRN: PES1201800144

# SECTION: C

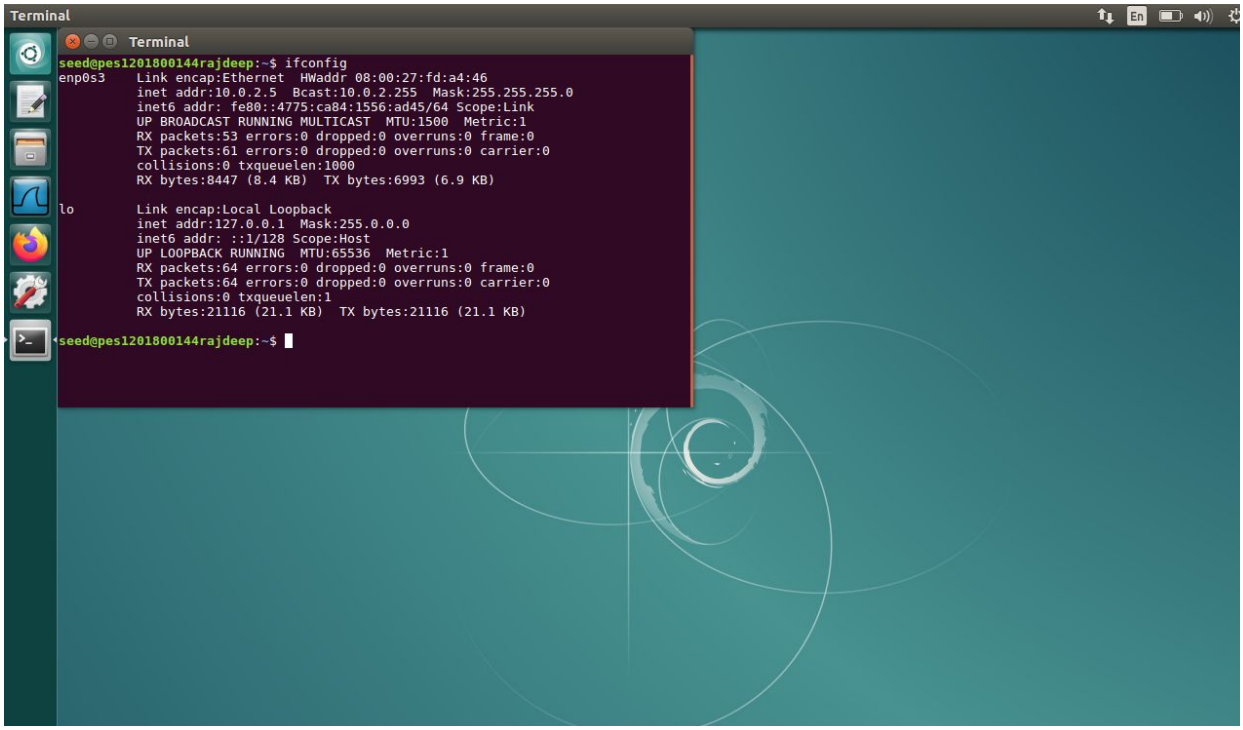

**NOTE: Please find my SRN ‘PES1201800144rajdeep’ as the terminal username.**

## MY CONFIGURATIONS:

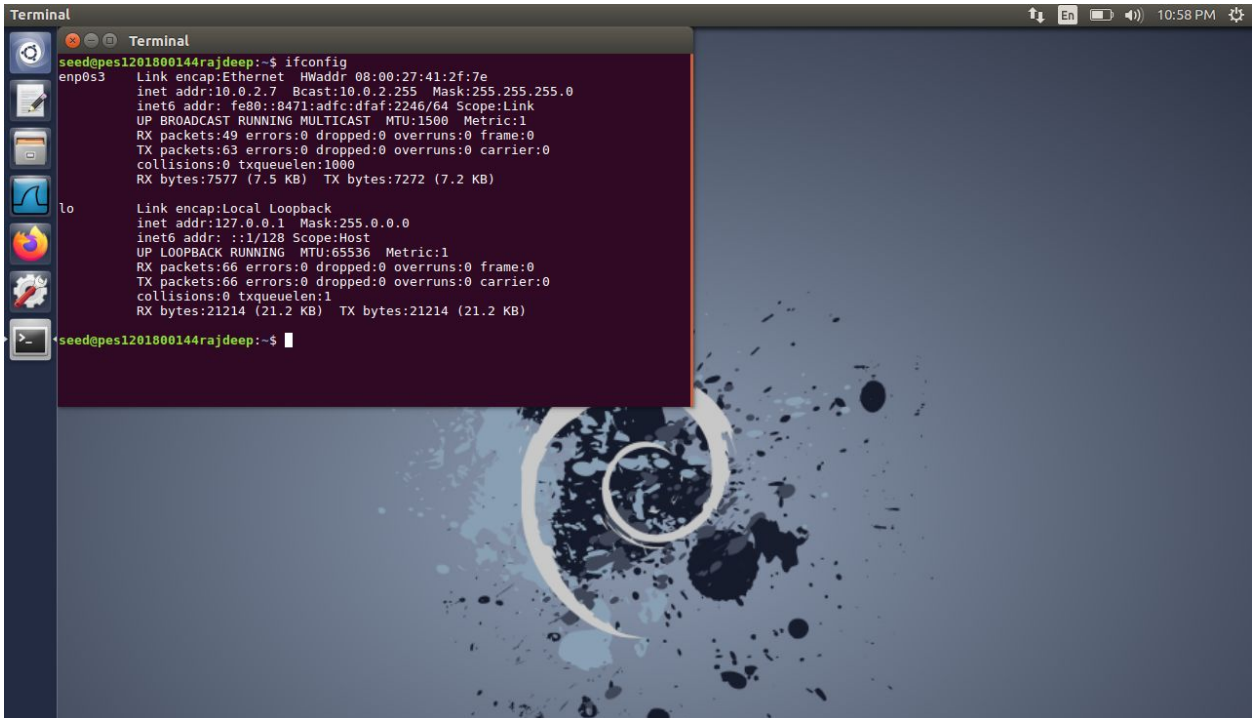
## VM1 (Attacker) => IP Address: 10.0.2.9




## VM2 (Victim/Client) => IP Address: 10.0.2.5



## VM3 (Observer/Server) => IP Address: 10.0.2.7




**In task 1 :**

- **Attacker → 10.0.2.9**
- **Victim → 10.0.2.5**
- **Observer → 10.0.2.7**

**In task 2,4,5 :**

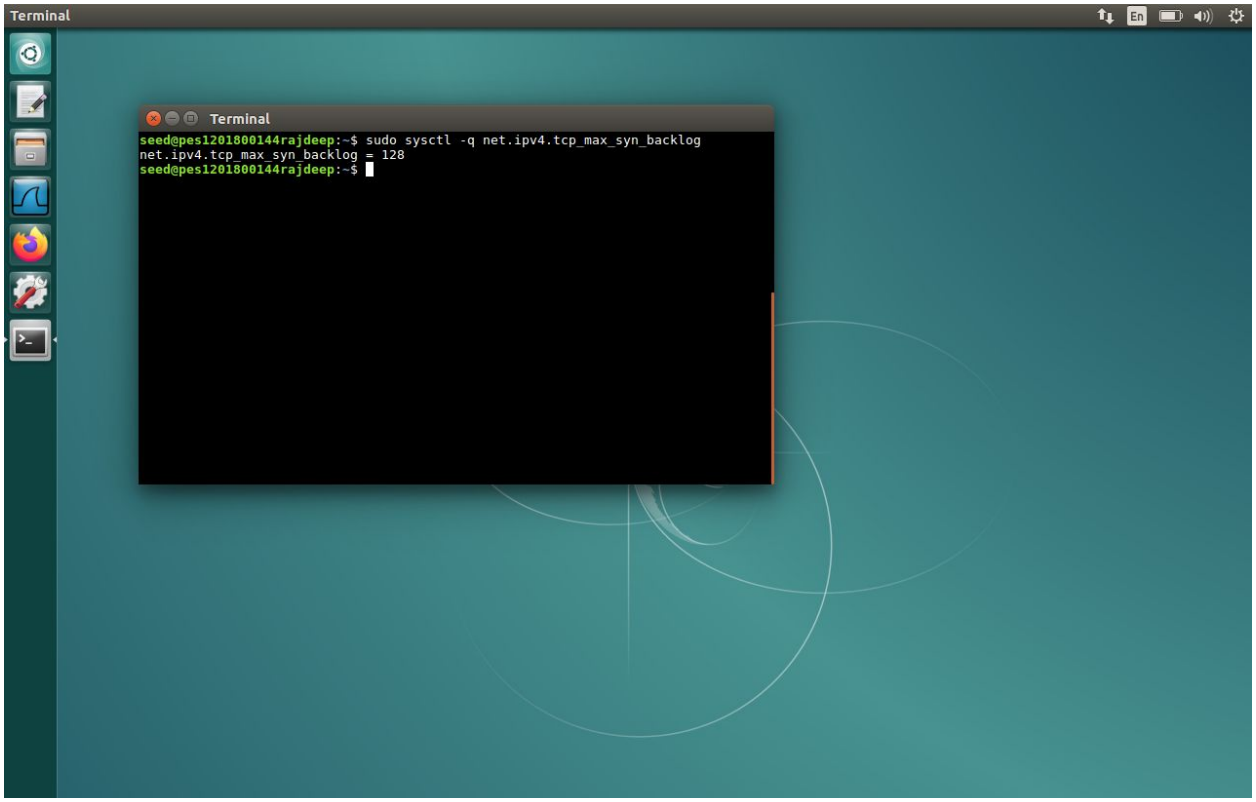
- **Attacker → 10.0.2.9**
- **Client → 10.0.2.5**
- **Server → 10.0.2.7**

**In task 3 :**

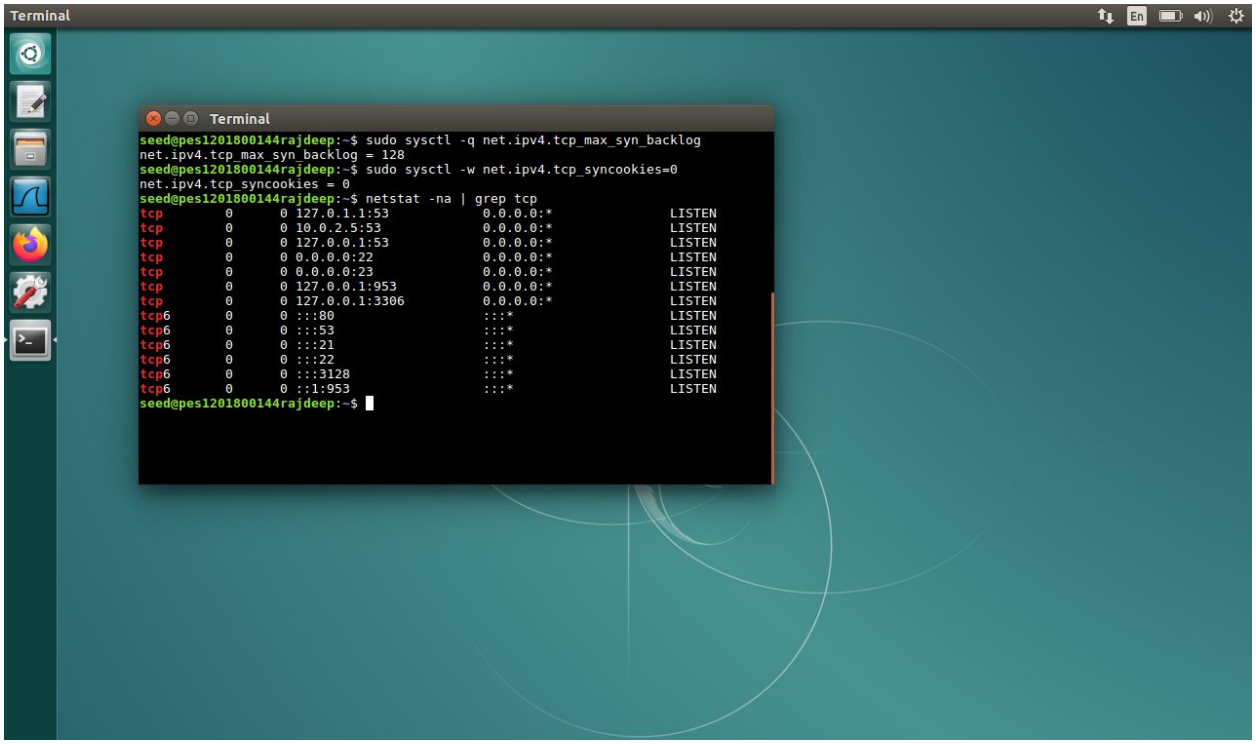
- **Attacker → 10.0.2.9**
- **Victim → 10.0.2.5**

==========

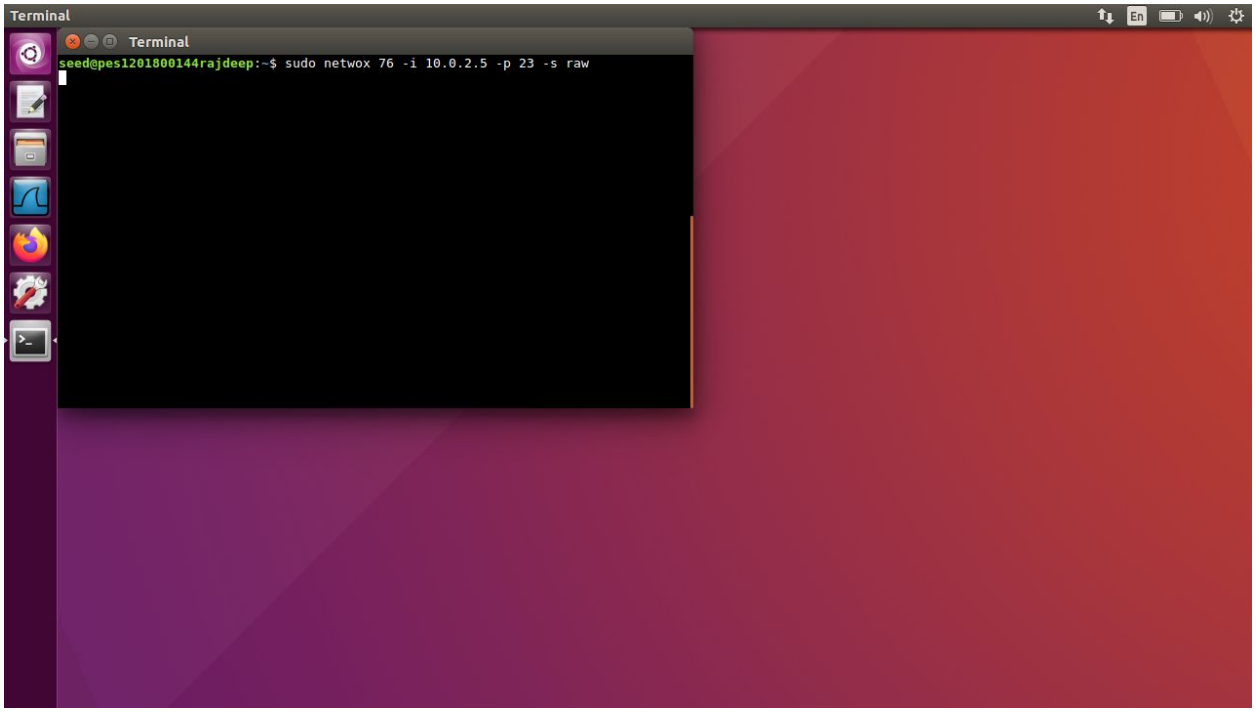
# TASK 1:



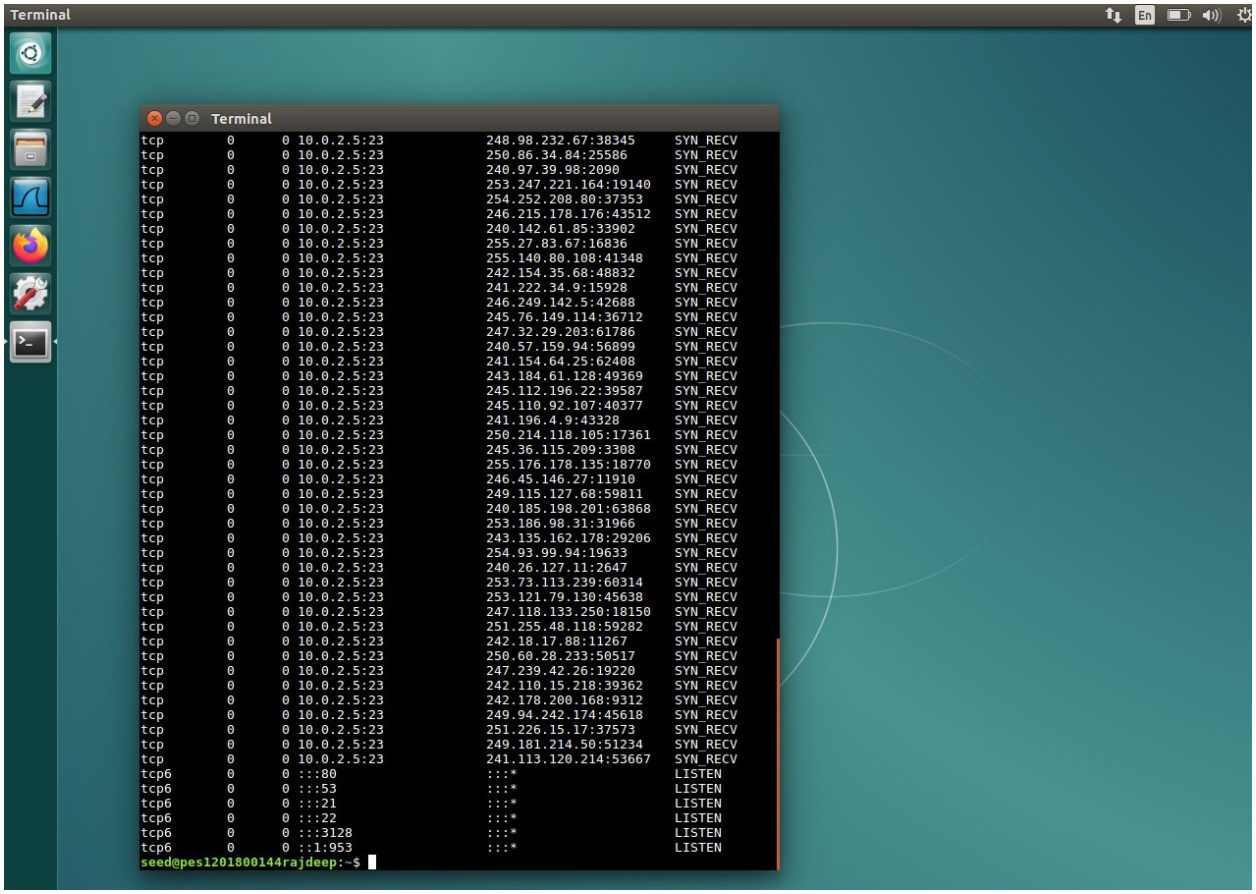

Screenshot 1.1: Here we can see that the SYN queue can take max 128 open connections on the victim machine.



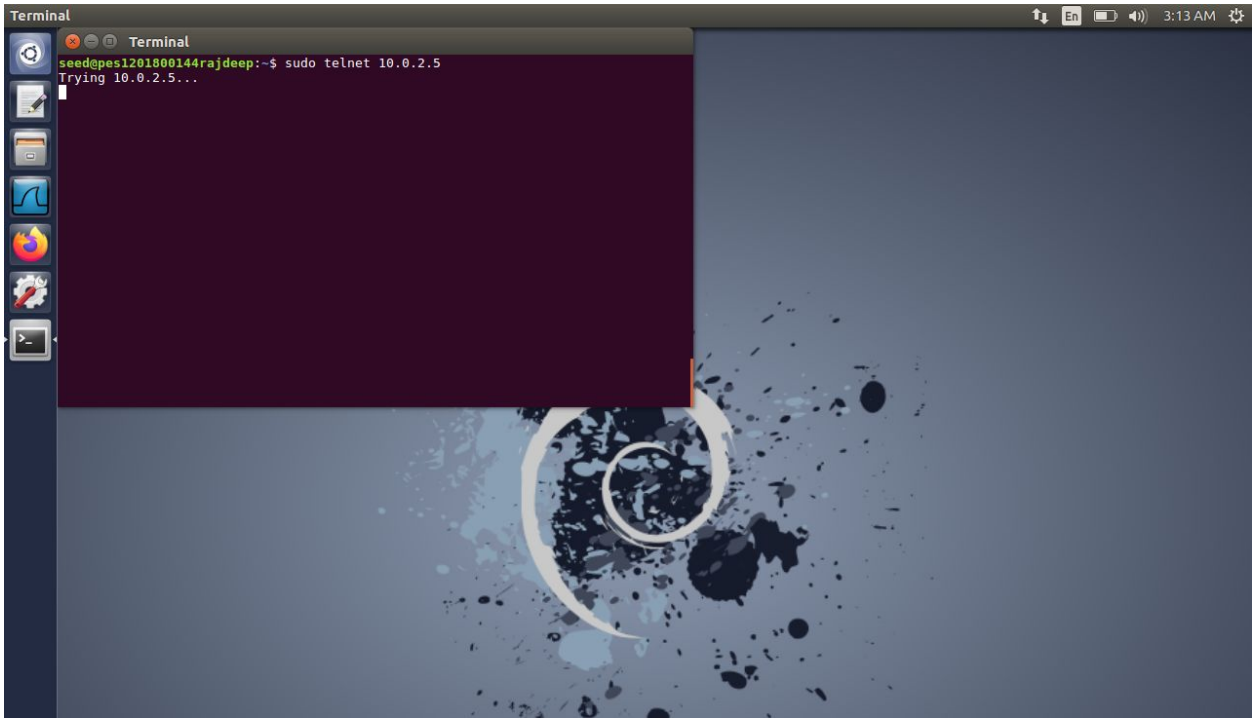

Screenshot 1.2: On the victim machine, tcp_syncookies is set to 0 so that SYN flooding can be performed. Also it can be seen that the connections in the queue are in LISTEN mode waiting for connections.



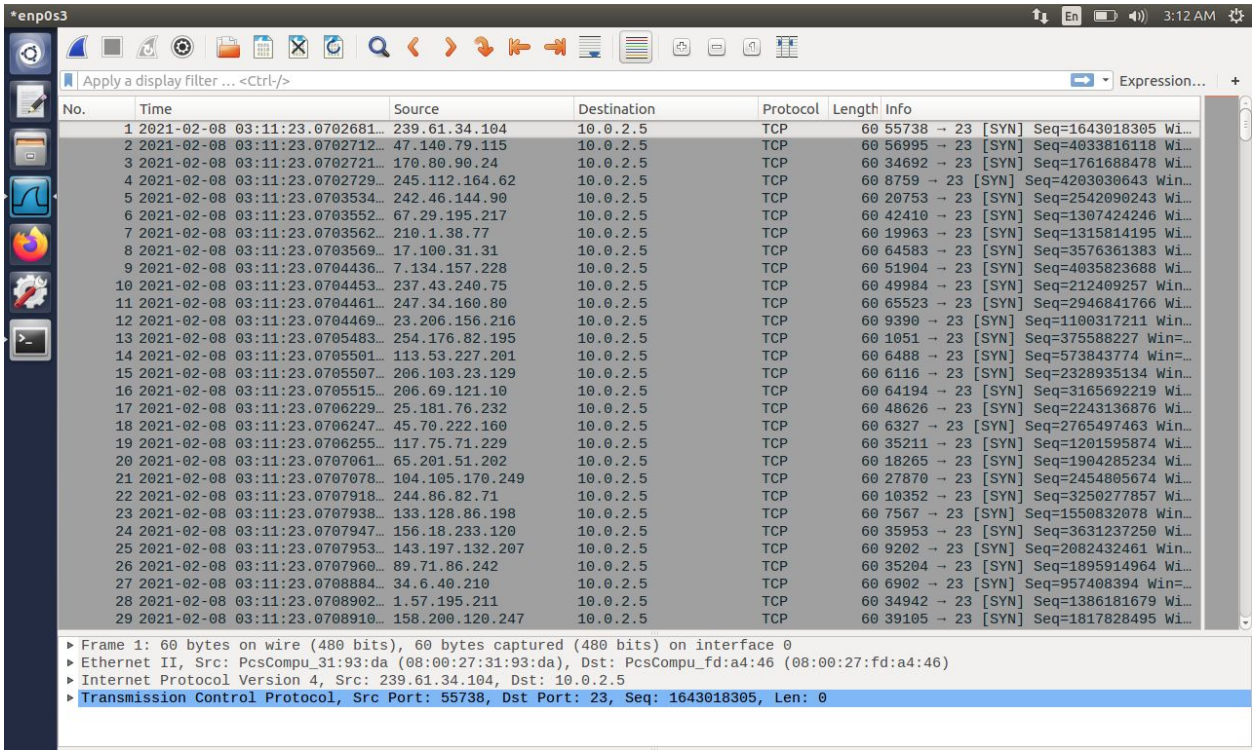

Screenshot 1.3: On the attacker machine, we execute the SYN flood attack.



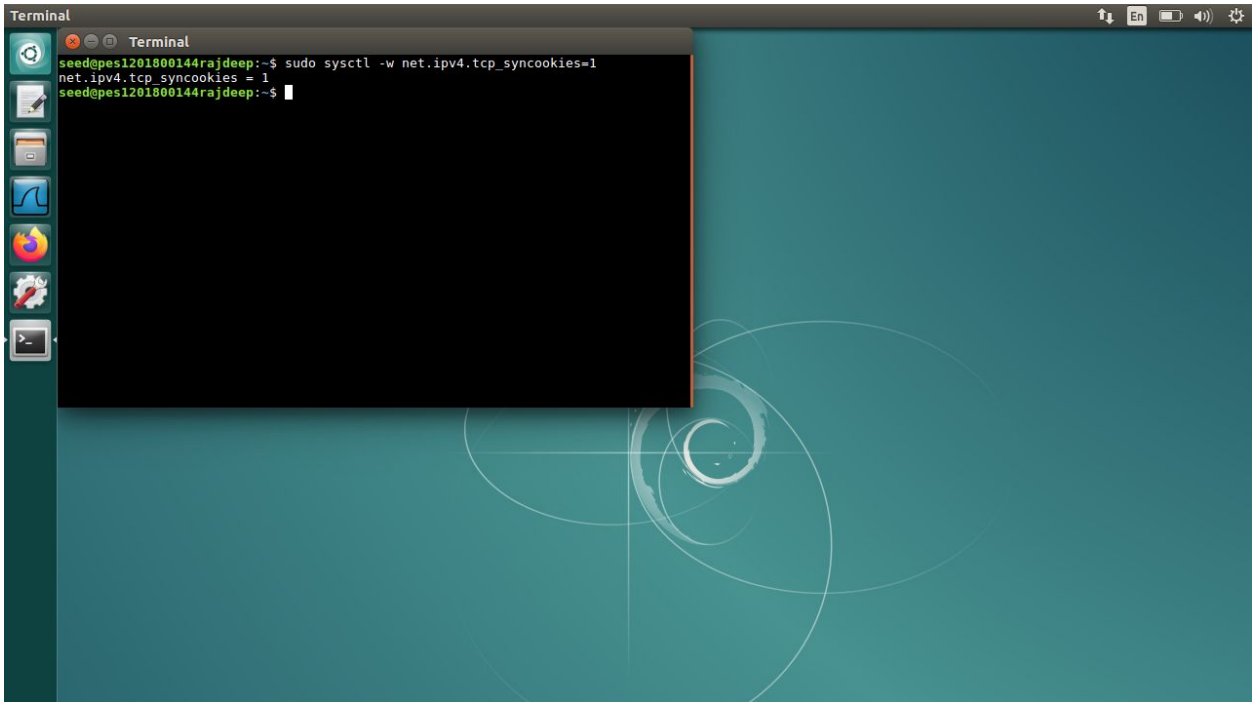

Screenshot 1.4: On the victim machine, the SYN flood attack has been executed and SYN connections can be seen from random IPs. Many connections can be seen as SYN_RECV. When all the connections in the queue turn from LISTEN mode to SYN_RECV mode, the machine will no longer be able to accept new connections.



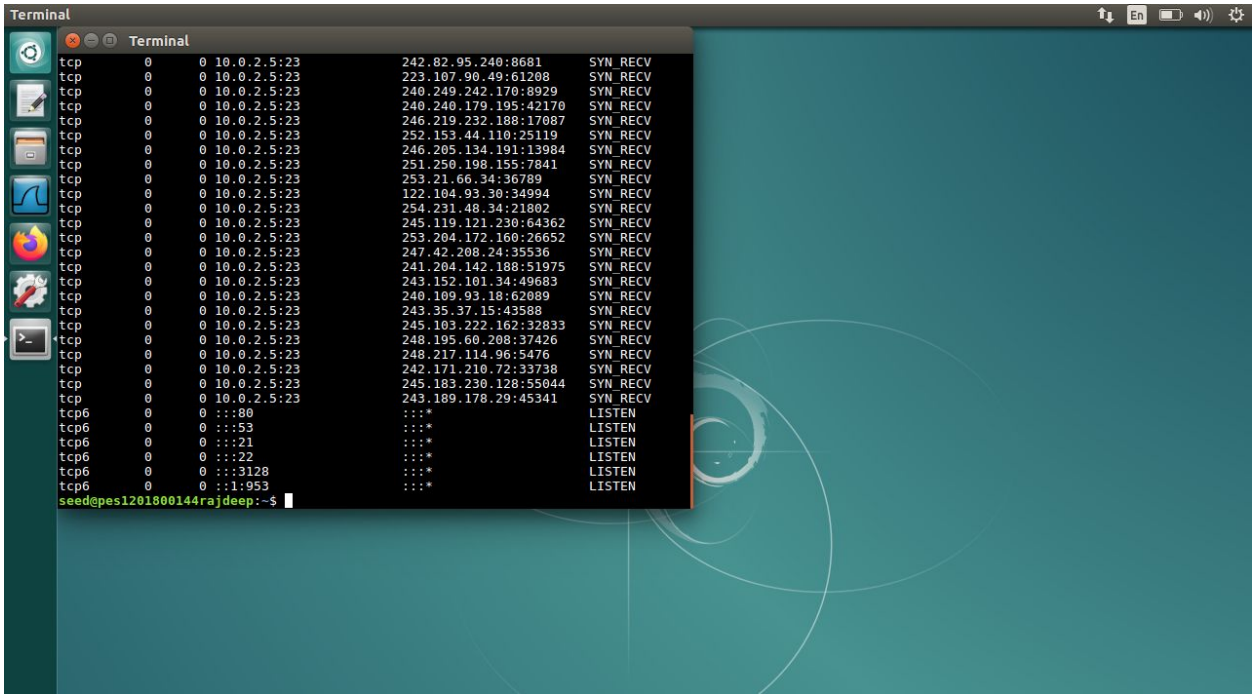

Screenshot 1.5_1: On the observer machine, we try to connect to the victim machine using telnet protocol which uses TCP port 23 but the connection cannot be made since the SYN queue is flooded by an attacker on the victim machine.



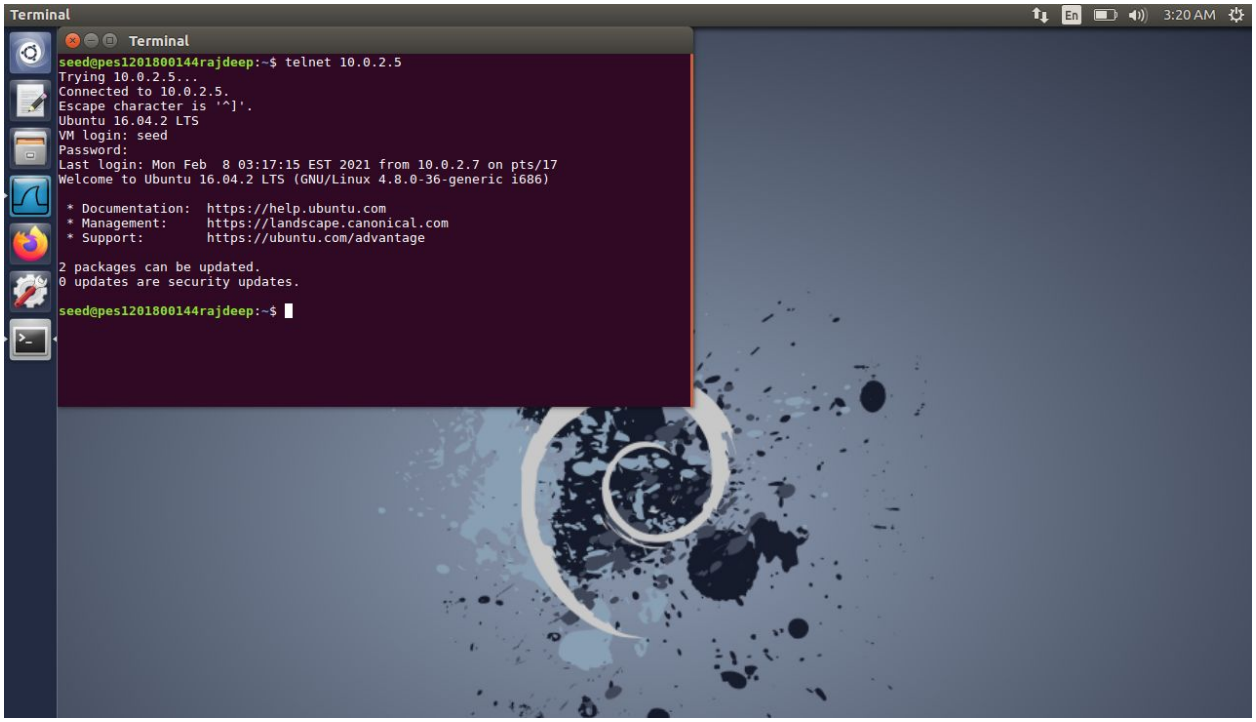

Screenshot 1.5_2: Using wireshark, we can find out that the victim machine is flooded with SYN packets from random IPs.




Screenshot 1.6: Now we set the tcp_syncookies to 1 on the victim machine.




Screenshot 1.7: We again check the SYN queue after executing the attack from the attacker machine while the tcp_syncookies=1 this time. Also, it can be seen that the victim machine is still flooded by the attacker.



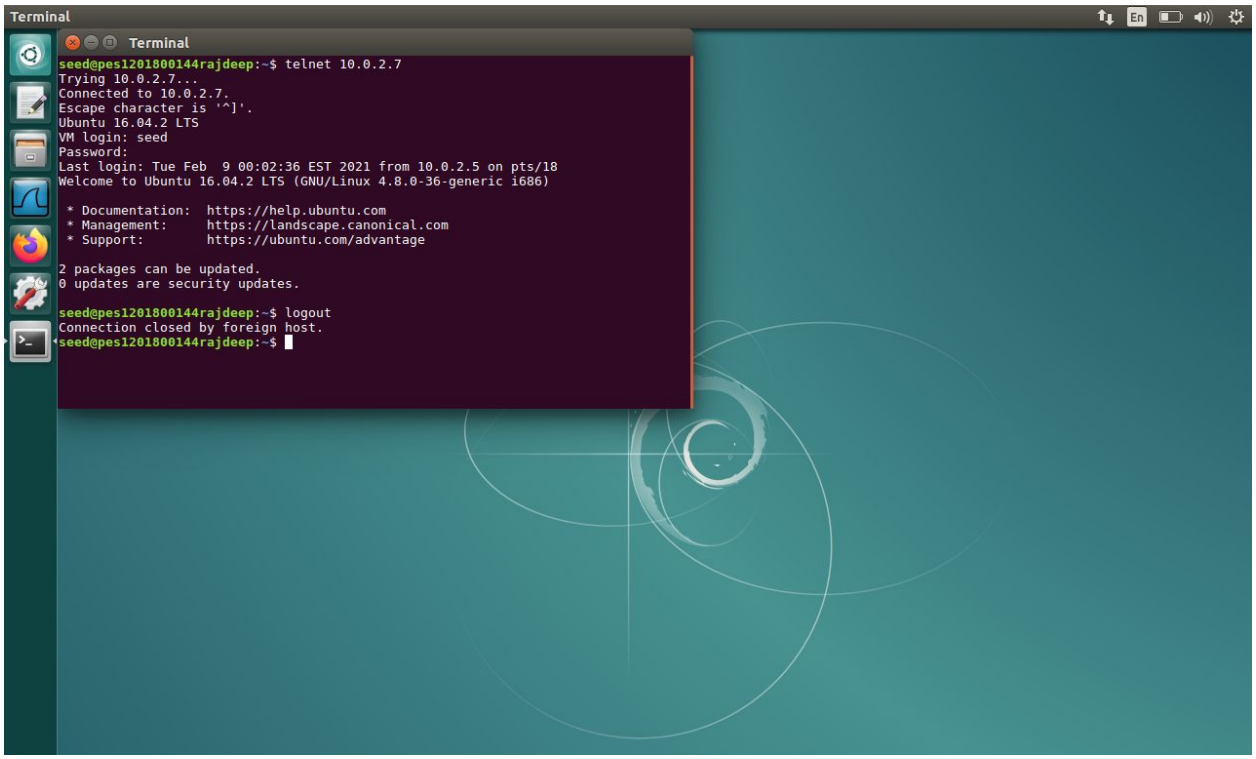

Screenshot 1.8: But this time, telnet connection is established from the observer machine to the victim machine.

**This is because the SYN cookies don't allow the SYN queue to be filled until the ACK is received. So the SYN flood packets are ignored and the telnet connection can successfully be established.**

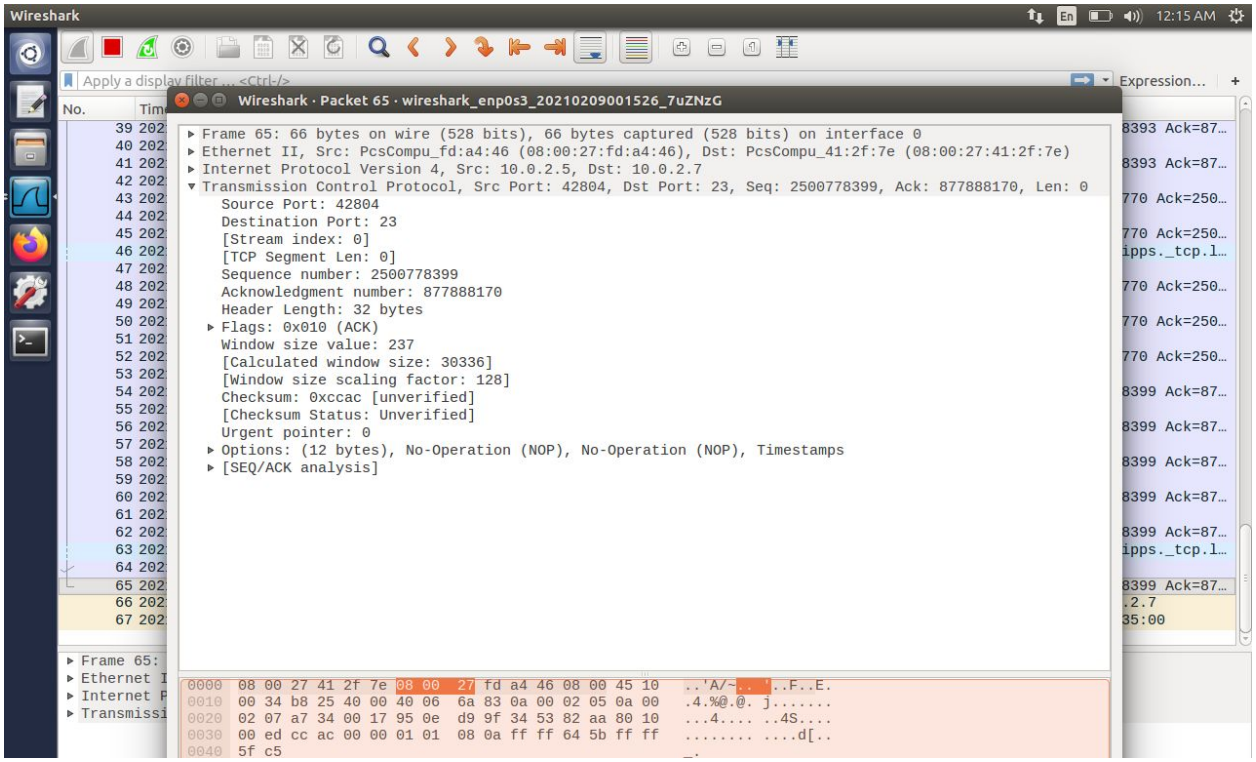
==========

# TASK 2:

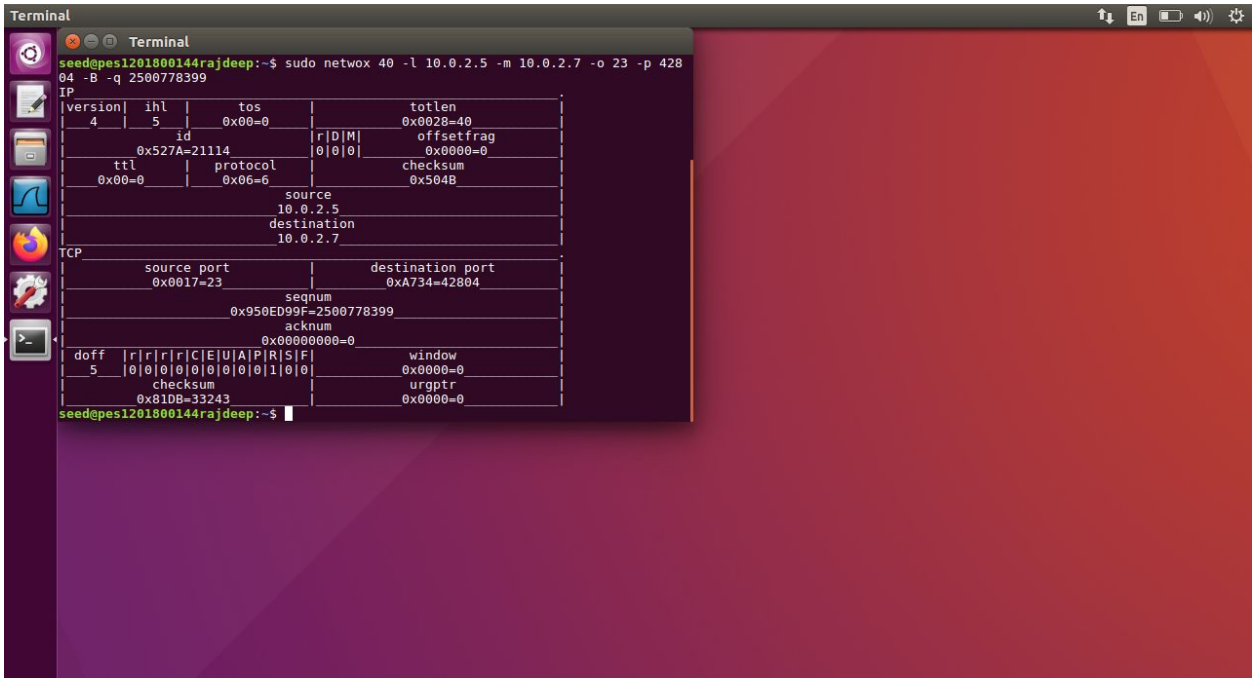
## 2.1 TELNET CONNECTION



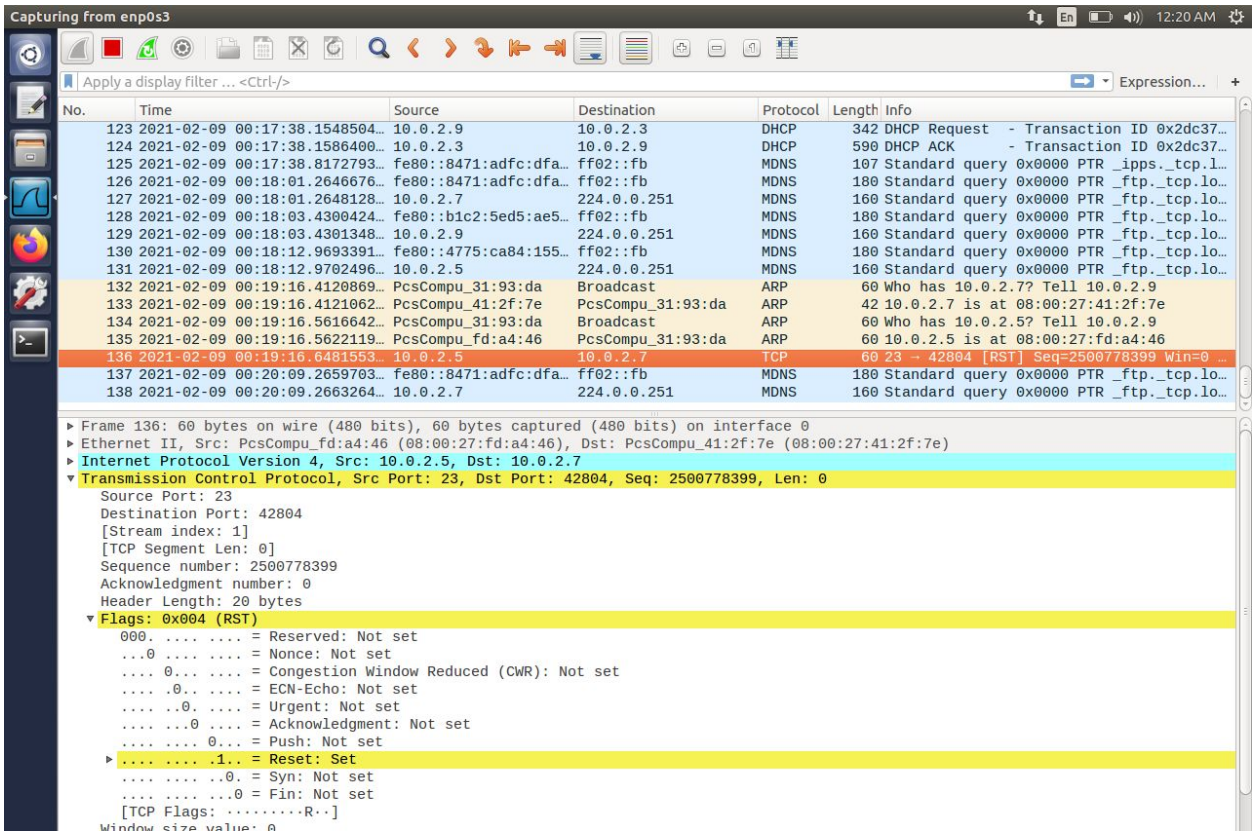

Screenshot 2.1.1: Connected telnet from client to server machine




Screenshot 2.1.2: Last packet details in wireshark for telnet connection

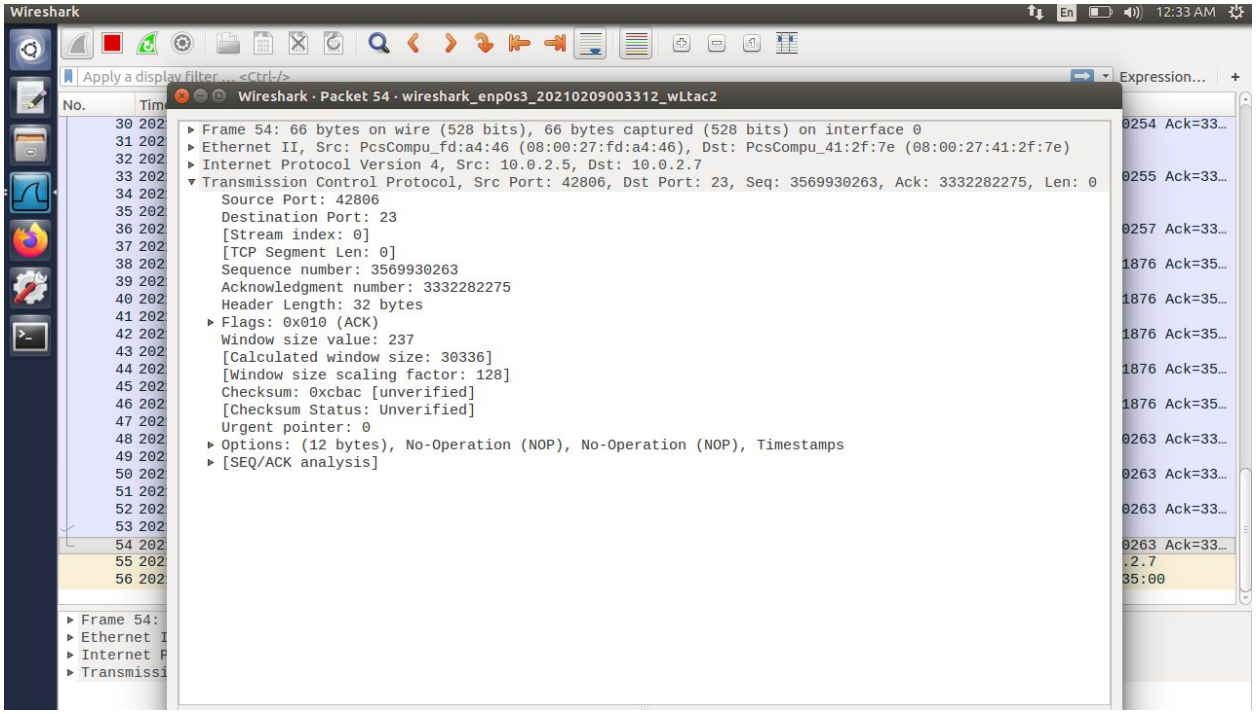



Screenshot 2.1.3: On attacker machine providing port(42804) and sequence number(2500778399) of the last packet as of above wireshark screenshot

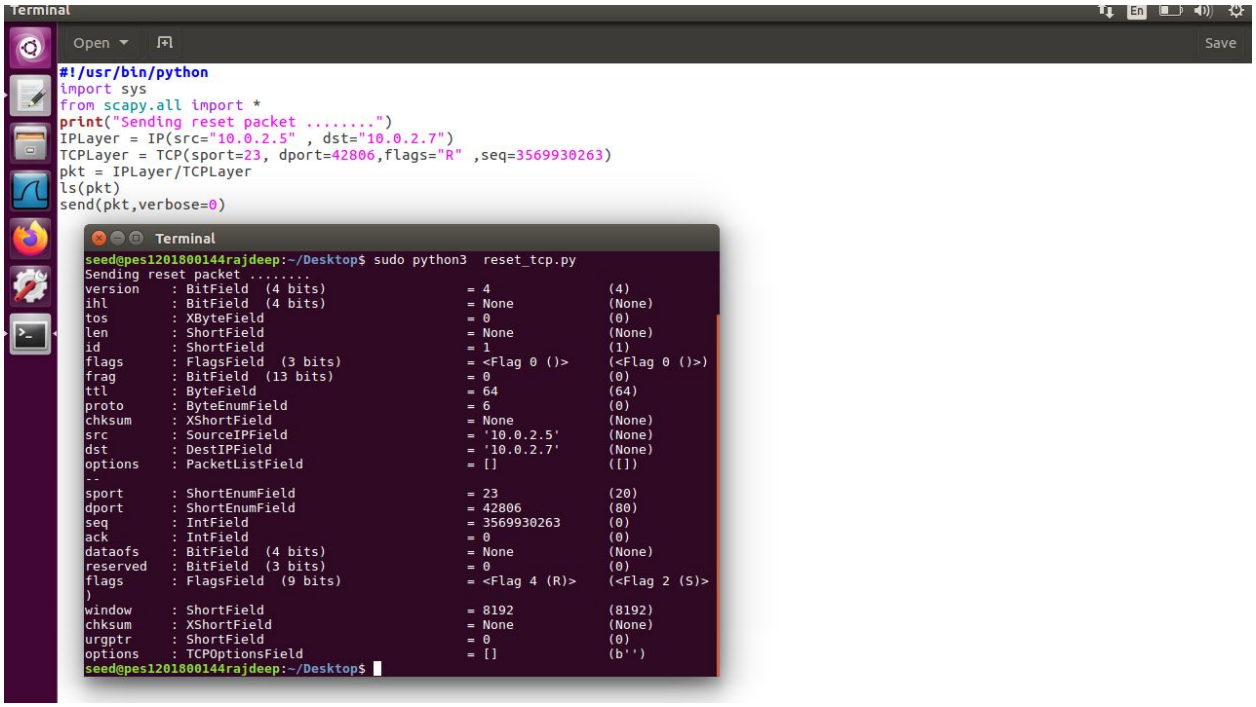



Screenshot 2.1.4: Telnet connection getting disconnecting as the RST packet is sent(reset bit set to 1).

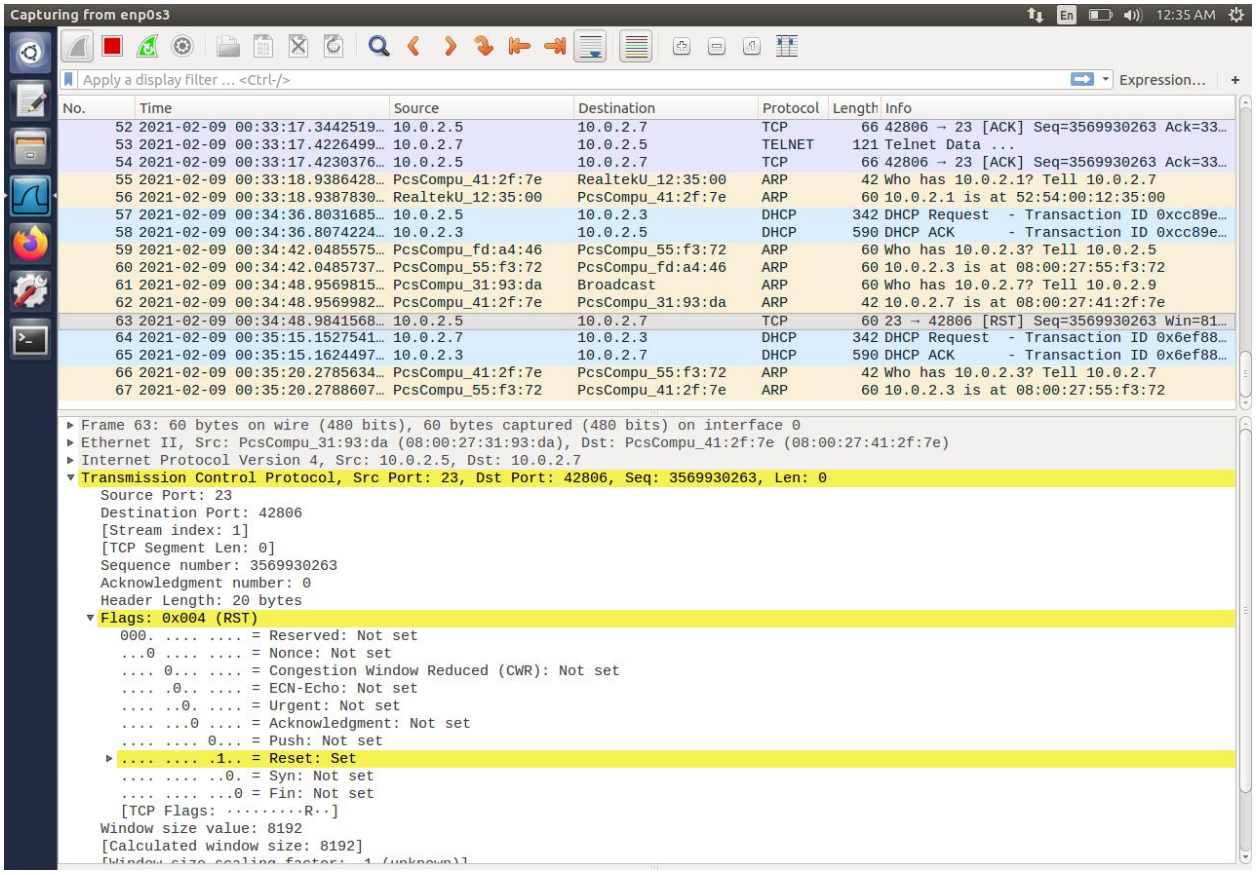
## Using Scapy:




Screenshot 2.1.5: Last packet details for new connection of telnet from client to server

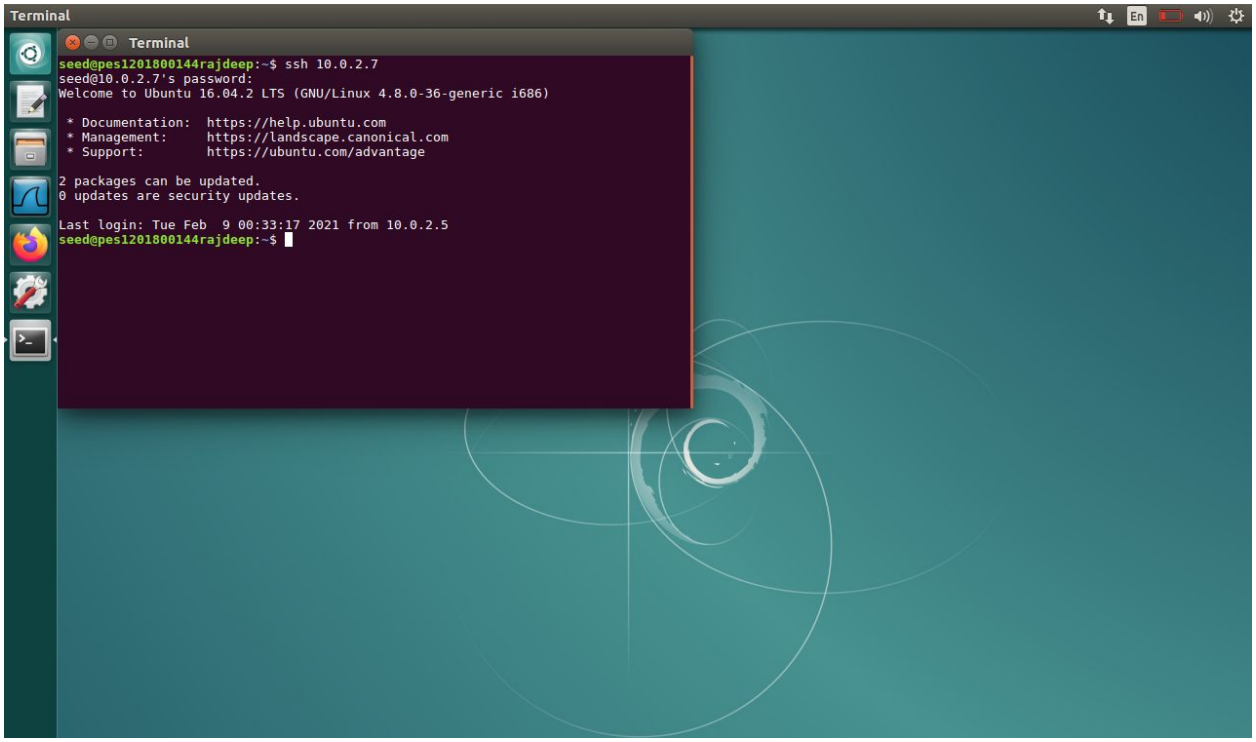



Screenshot 2.1.6: Code Snippet along with terminal output of execution of code.

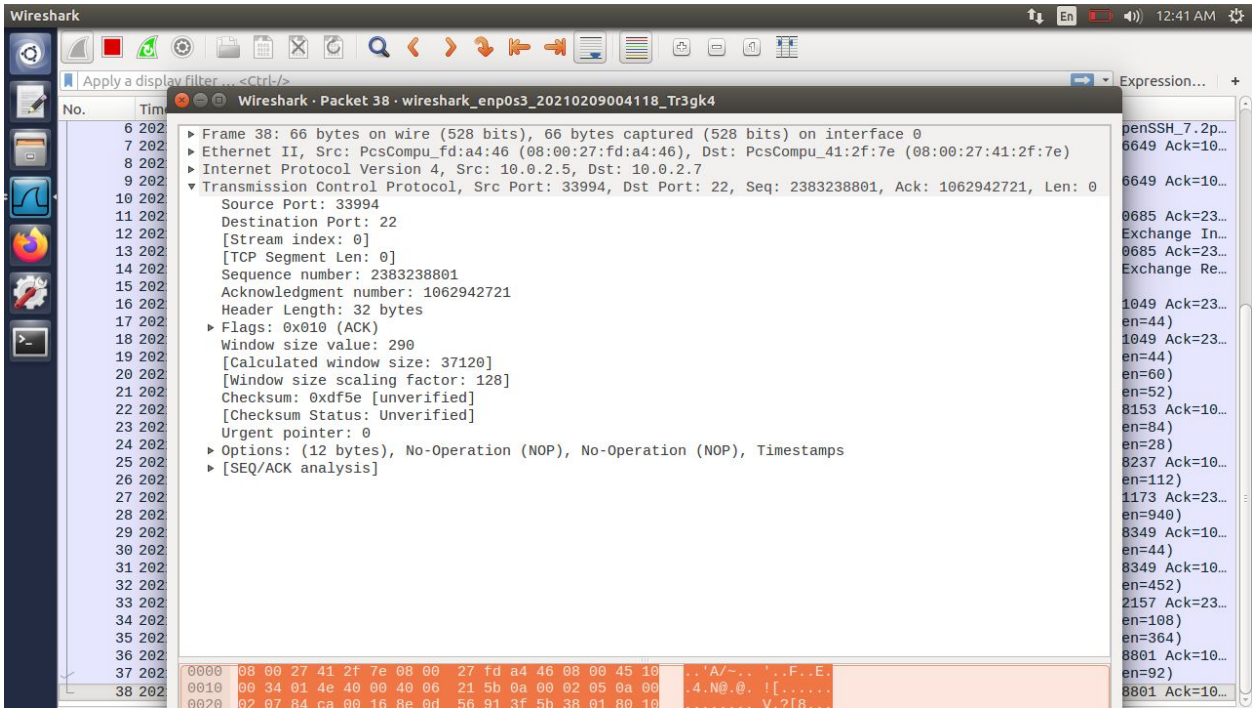



Screenshot 2.1.7: Connection reset since the packet can be seen with reset bit set to 1(RST packet sent).

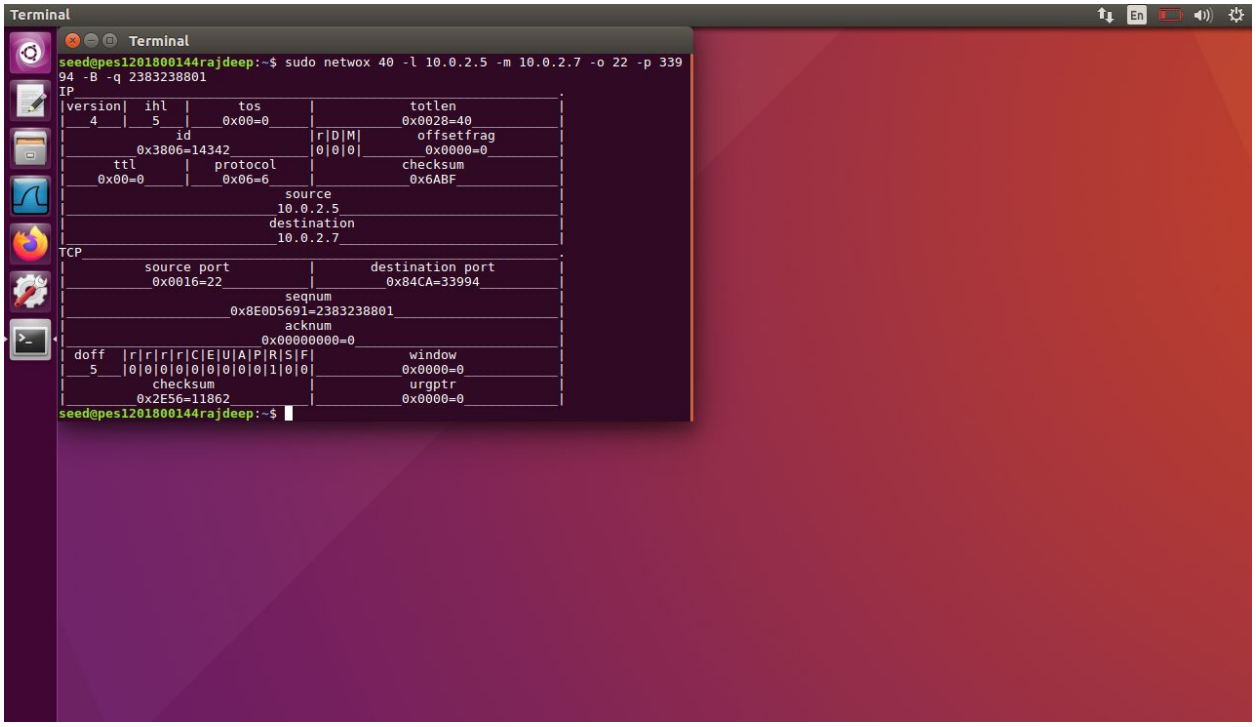
## 2.2 SSH CONNECTION



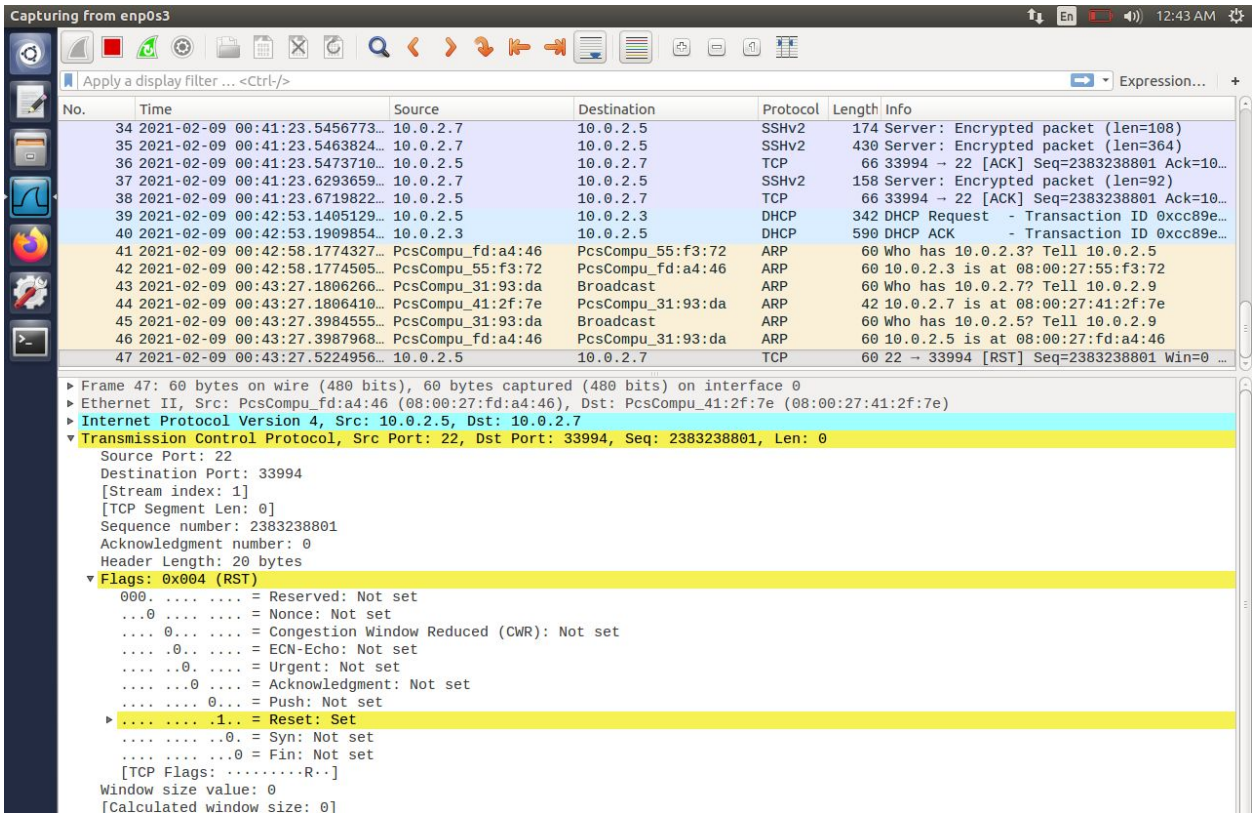

Screenshot 2.2.1: SSH connection successful from client to server




Screenshot 2.2.2: Last TCP packet in wireshark with sequence number(2383238801) and port(33994)

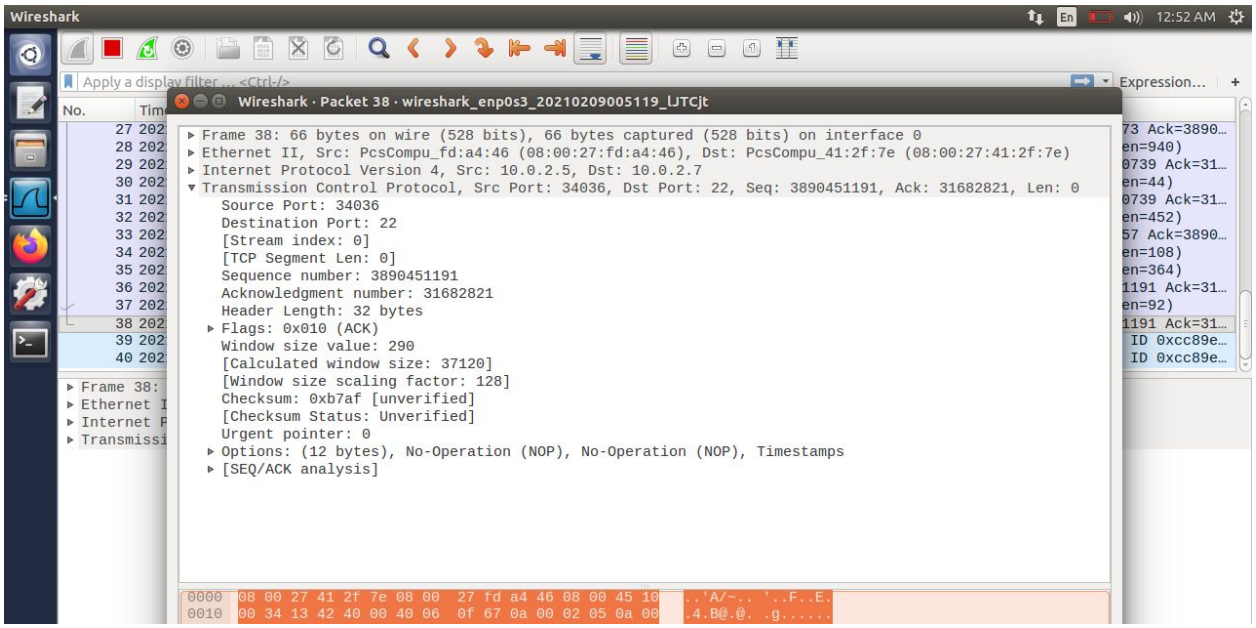



Screenshot 2.2.3: Attacker machine executing netwox command

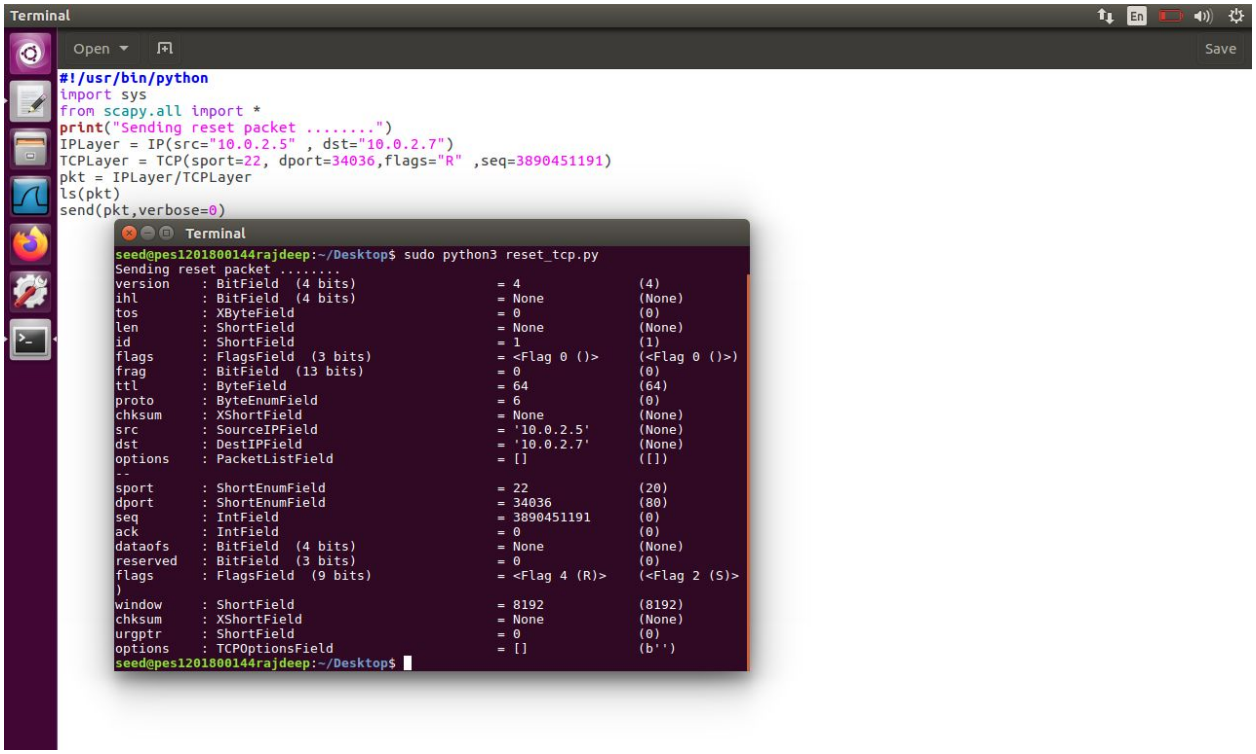



Screenshot 2.2.4: Connection reset as the RST packet is sent(reset bit of last packet set to 1).

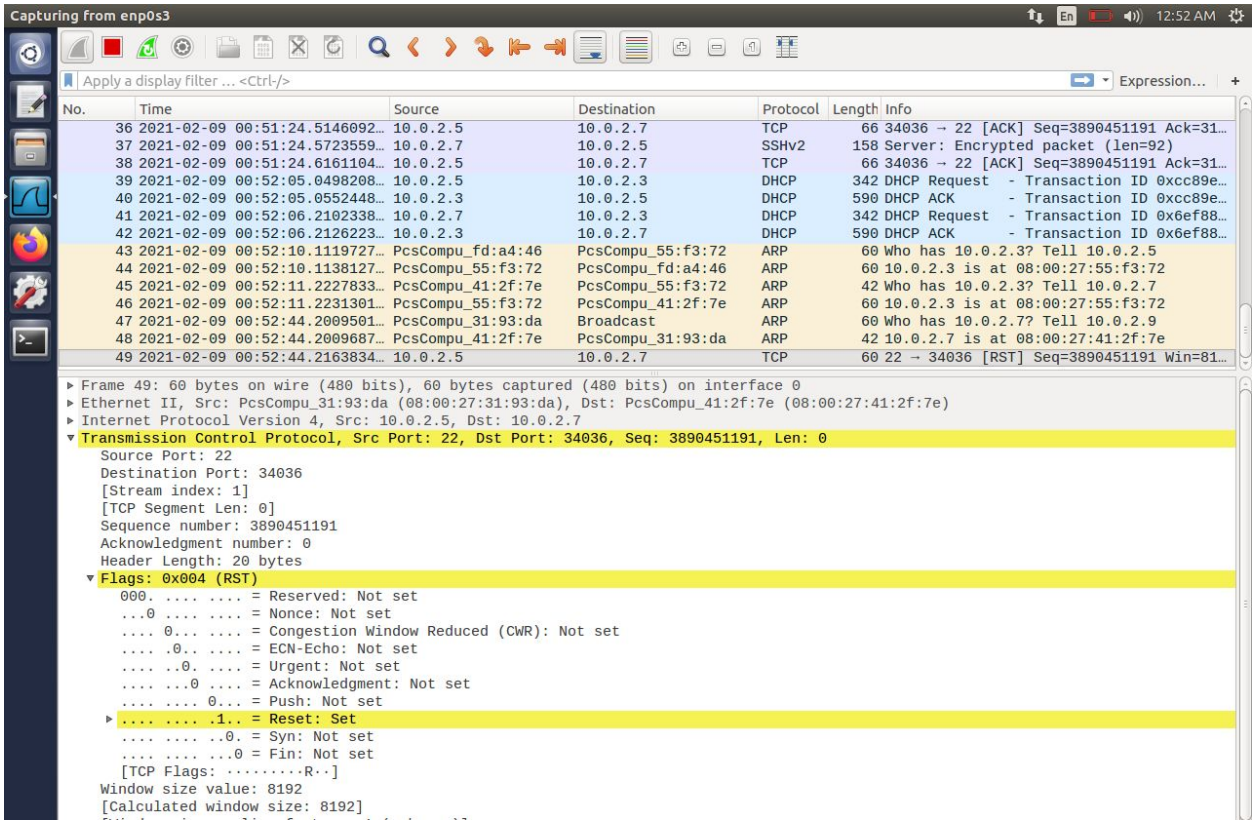
## Using Scapy:




Screenshot 2.2.5: Last packet details with port(34036) and sequence number(3890451191)




Screenshot 2.2.6: Code snippet along with the execution of code on terminal



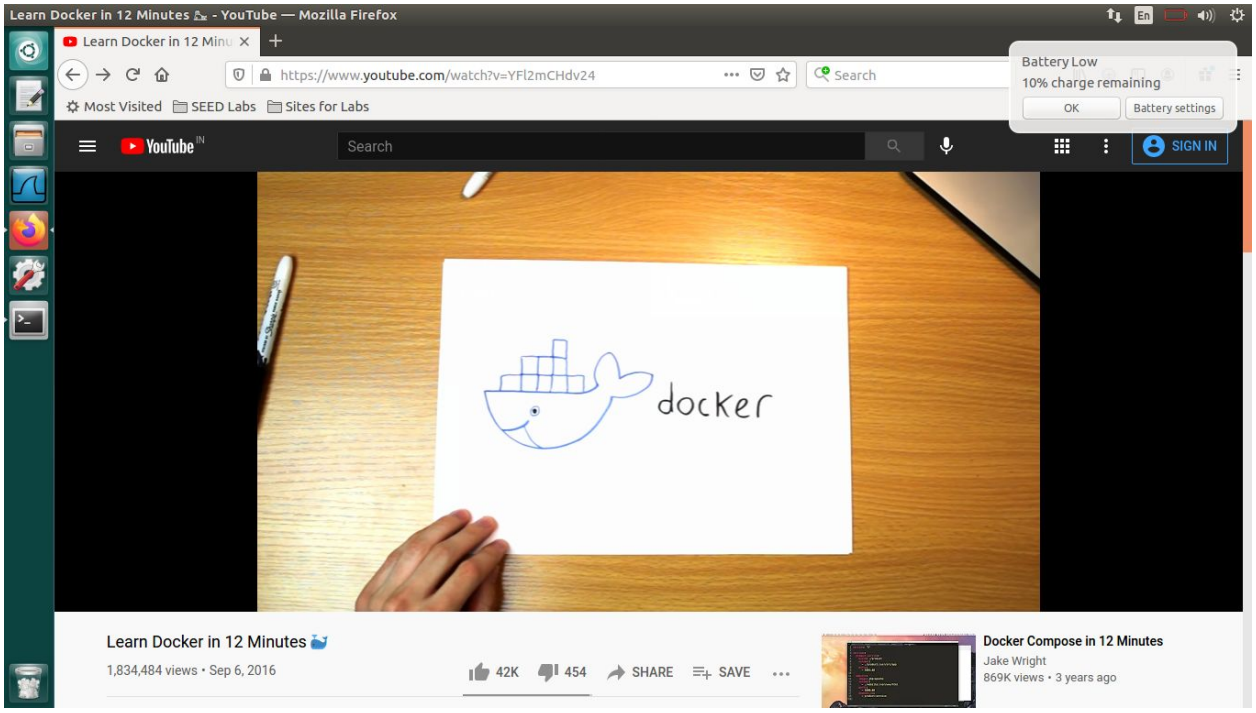

Screenshot 2.2.7: Connection closed since the last packet's reset bit is set to 1(RST packet sent).

**ADDITIONAL OBSERVATION:**

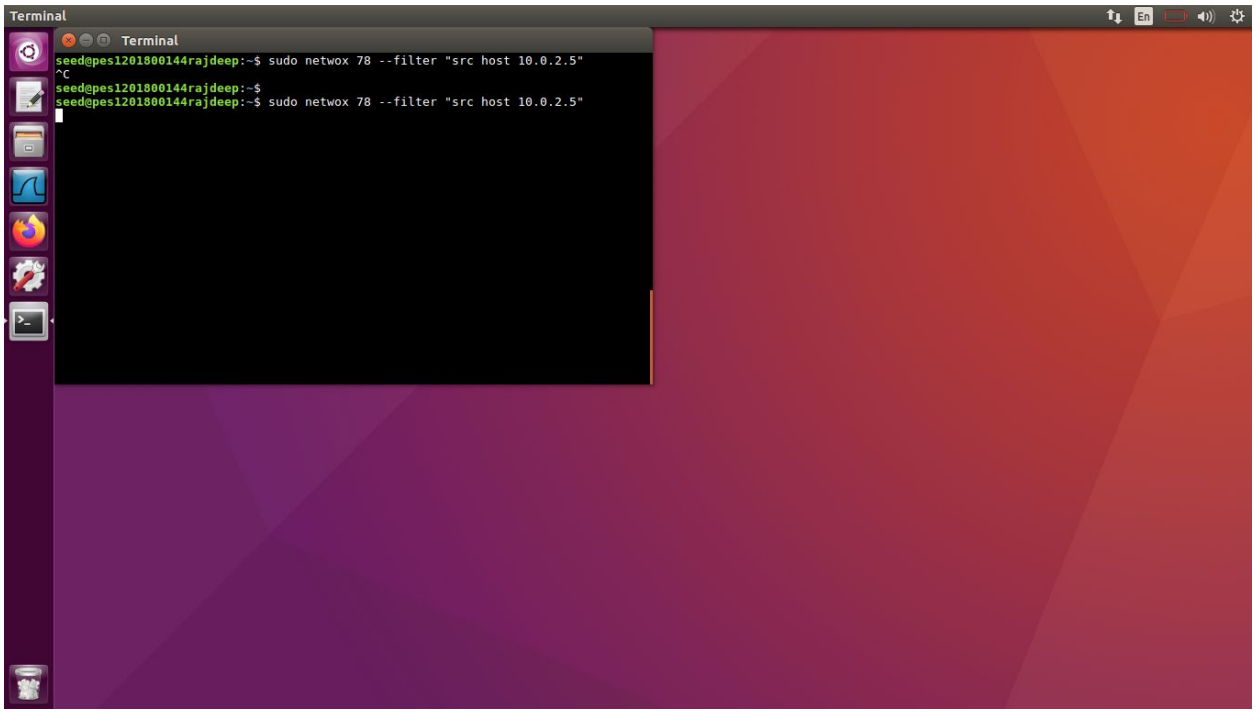
**This is a Denial-of-Service(DoS) attack since the attacker tries to disrupt the ssh or telnet connection between the client and server.**
**This is done by:**
**The attacker sending RST(sets the reset bit to 1) for the last TCP packet of the telnet or ssh connection**

==========

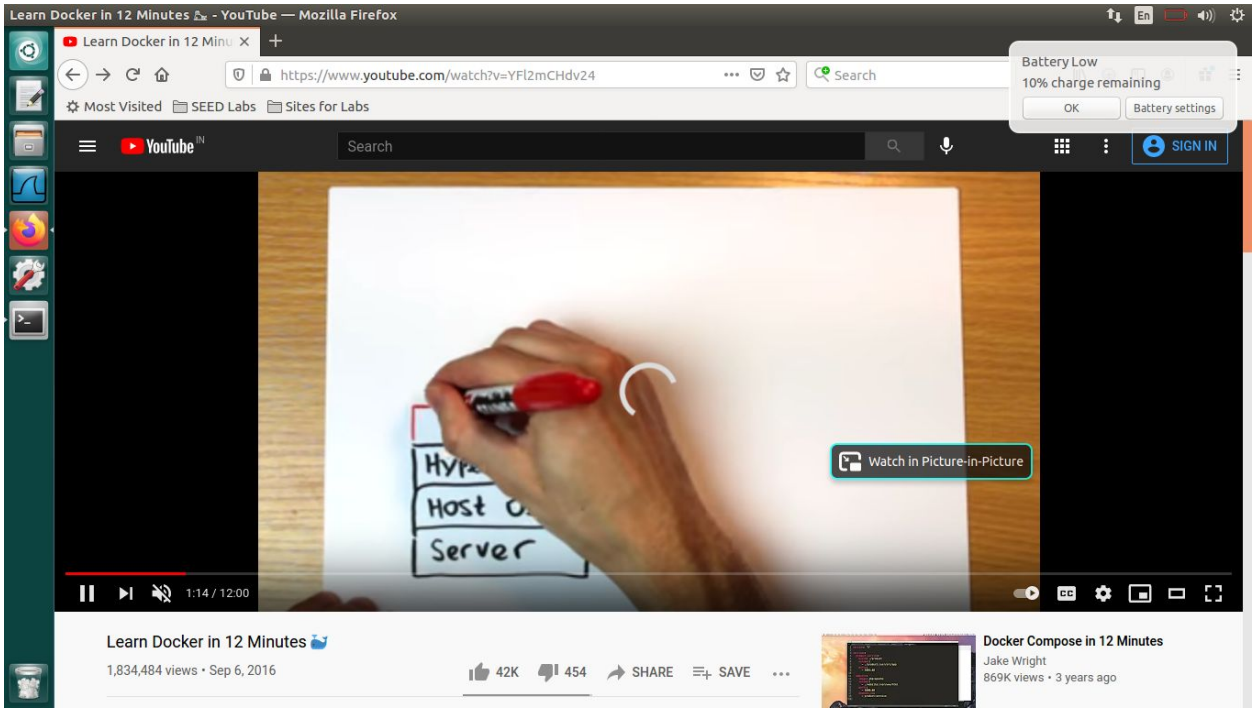
# TASK 3:




Screenshot 3.1: Youtube video playing normally on victim machine




Screenshot 3.2: Attacker executes the command




Screenshot 3.3: Youtube video crashes and keeps on buffering.

**ADDITIONAL OBSERVATION:**

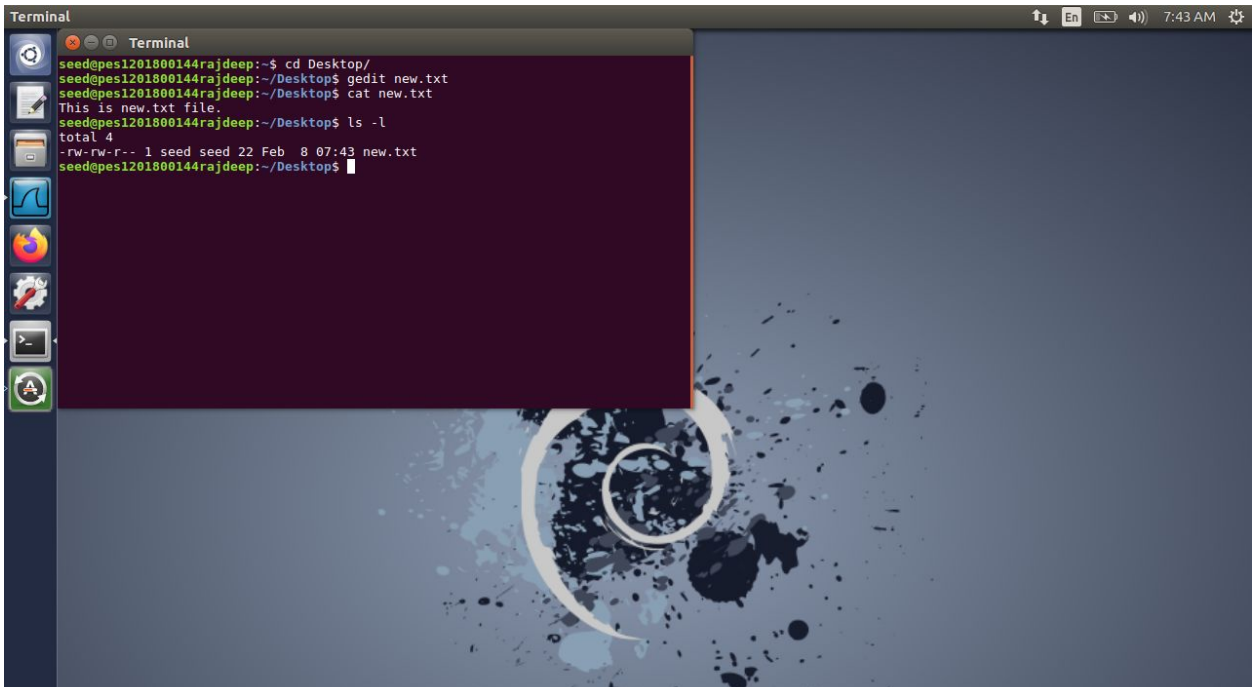
**This is a Denial-of-Service(DoS) attack since the attacker tries to disrupt the connection between the client and server.**

**This happens because the attacker resets all the TCP packets between victim and the youtube server using netwox 78 tool.**

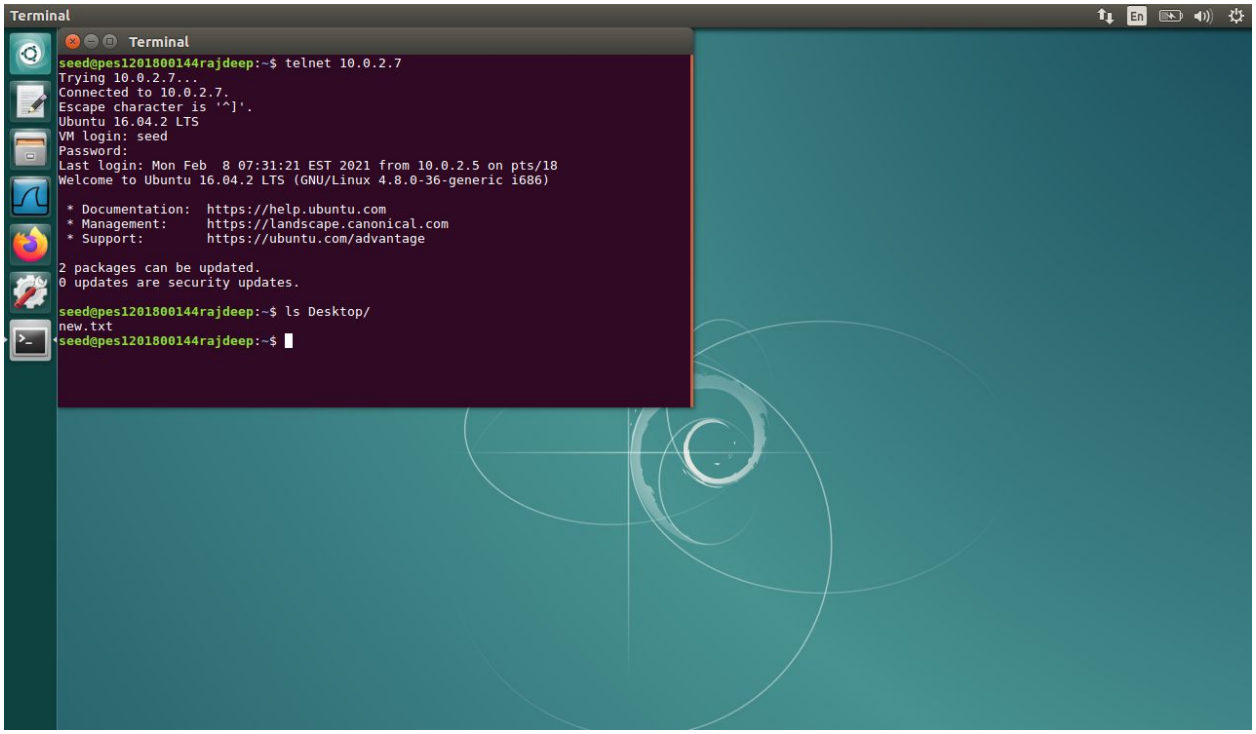
==========

# TASK 4:

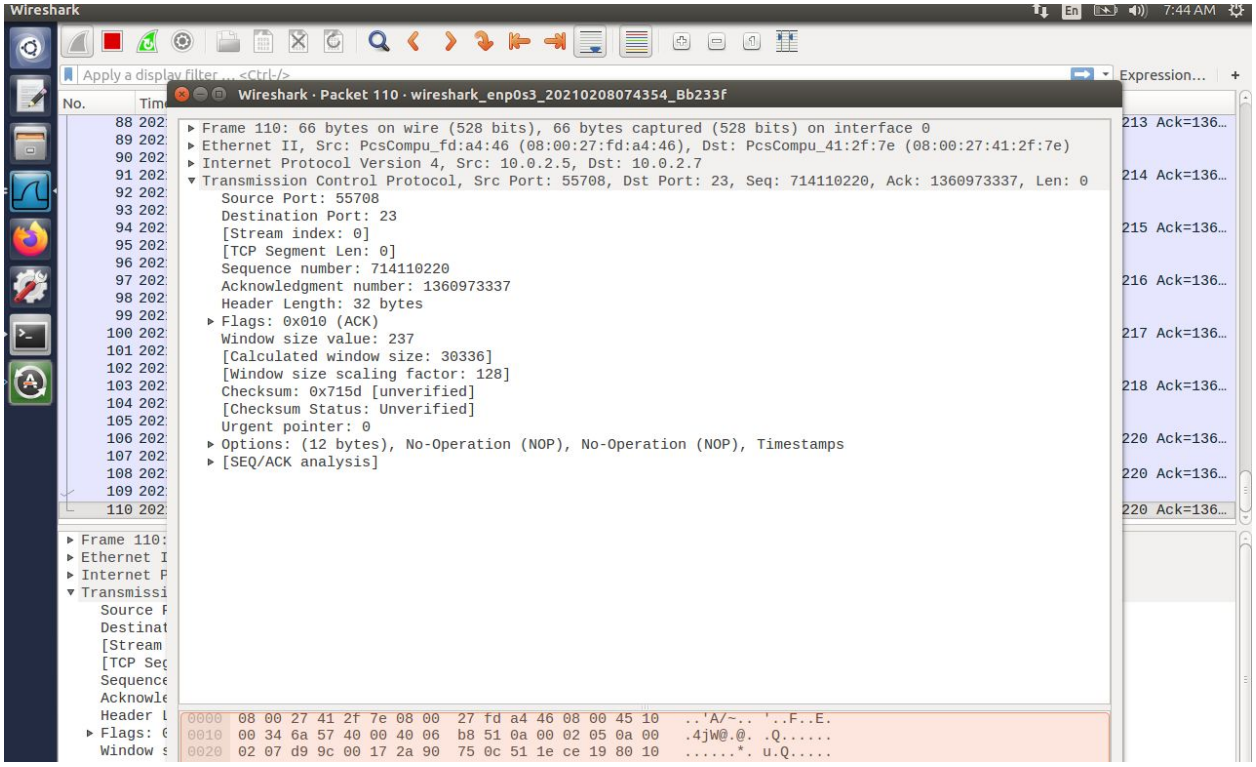
## 4.1 Using netwox command:



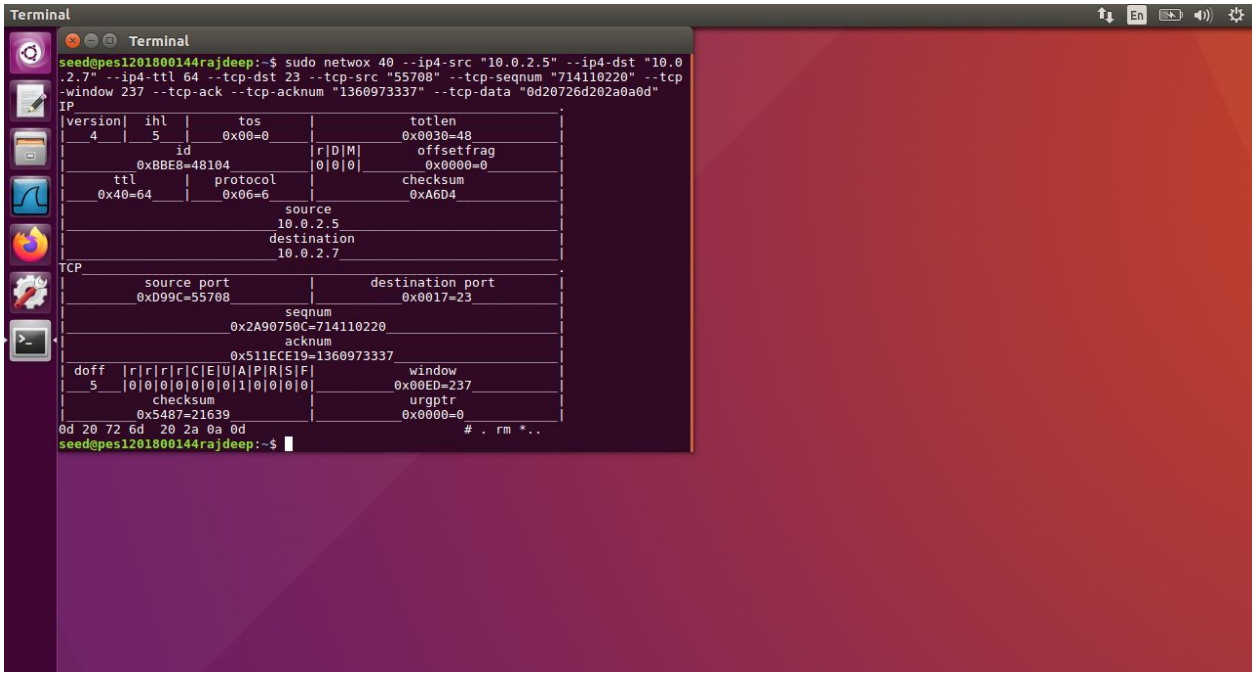

Screenshot 4.1.1: Creating a new.txt file in the desktop directory of server machine.



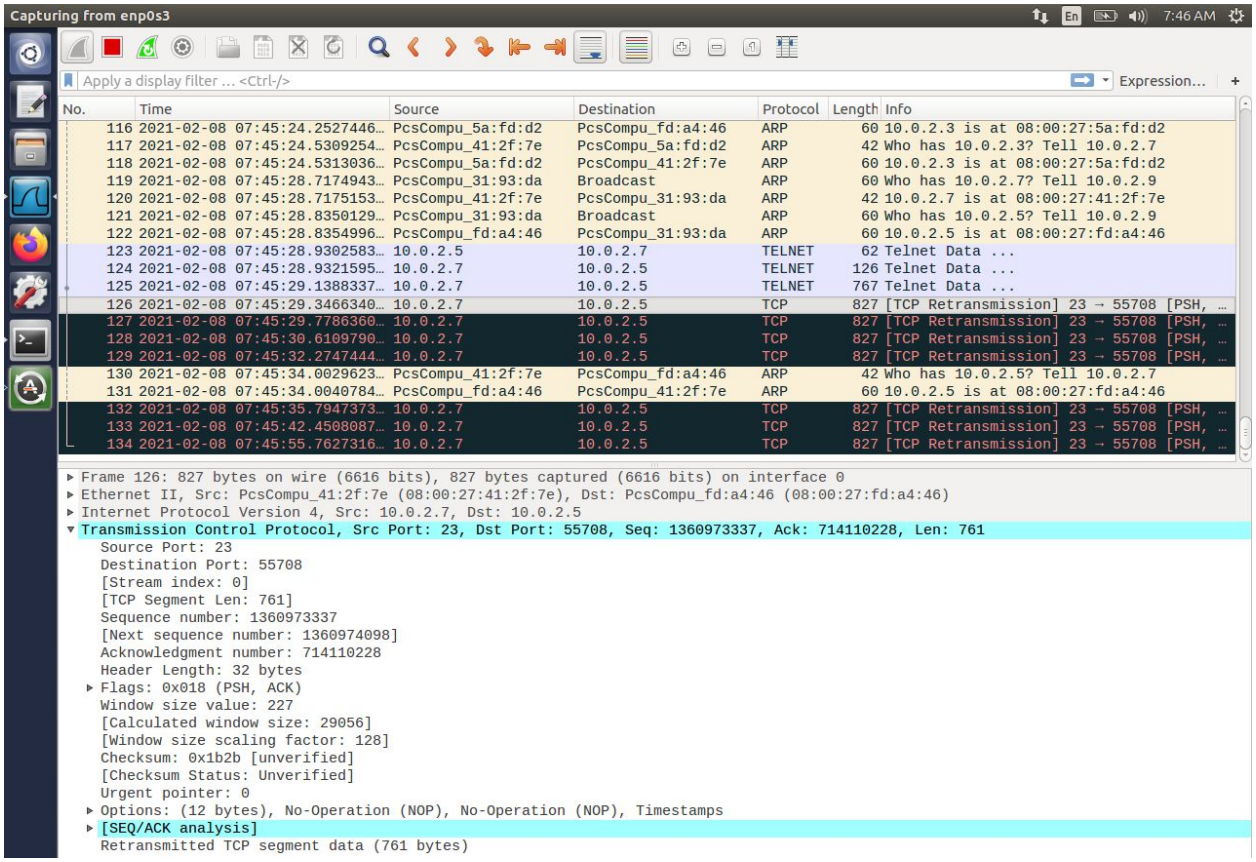

Screenshot 4.1.2: Telnet connection from client to server and server can access the new.txt file in server's desktop directory.



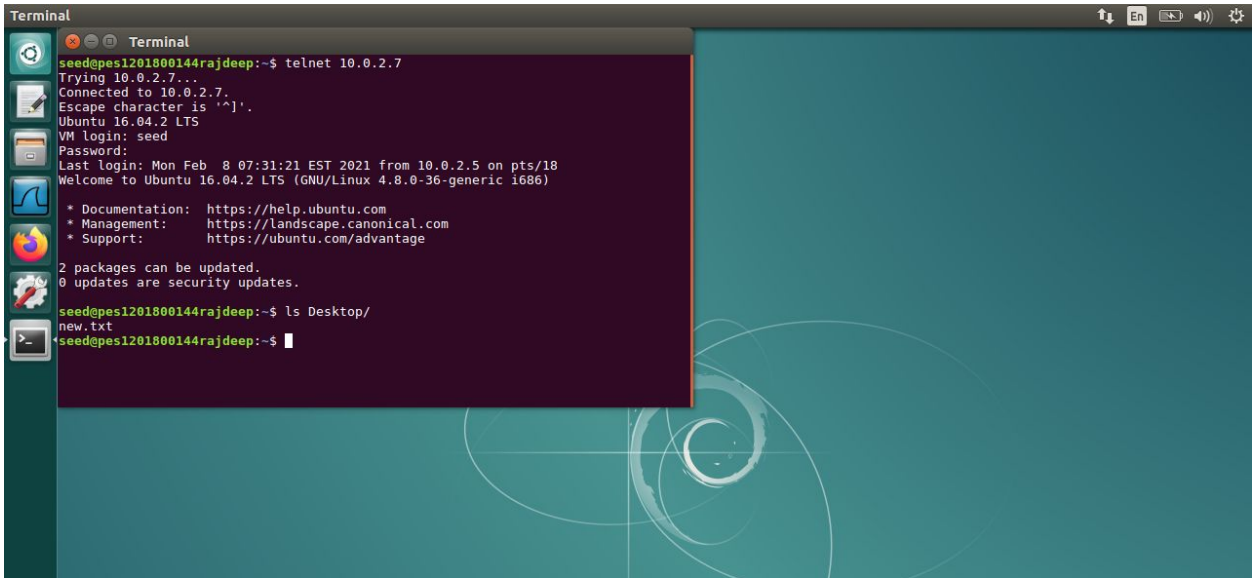

Screenshot 4.1.3: Last TCP packet in wireshark for the connection of client to server.




Screenshot 4.1.4: Attacker using the sequence number, port and acknowledgement number.

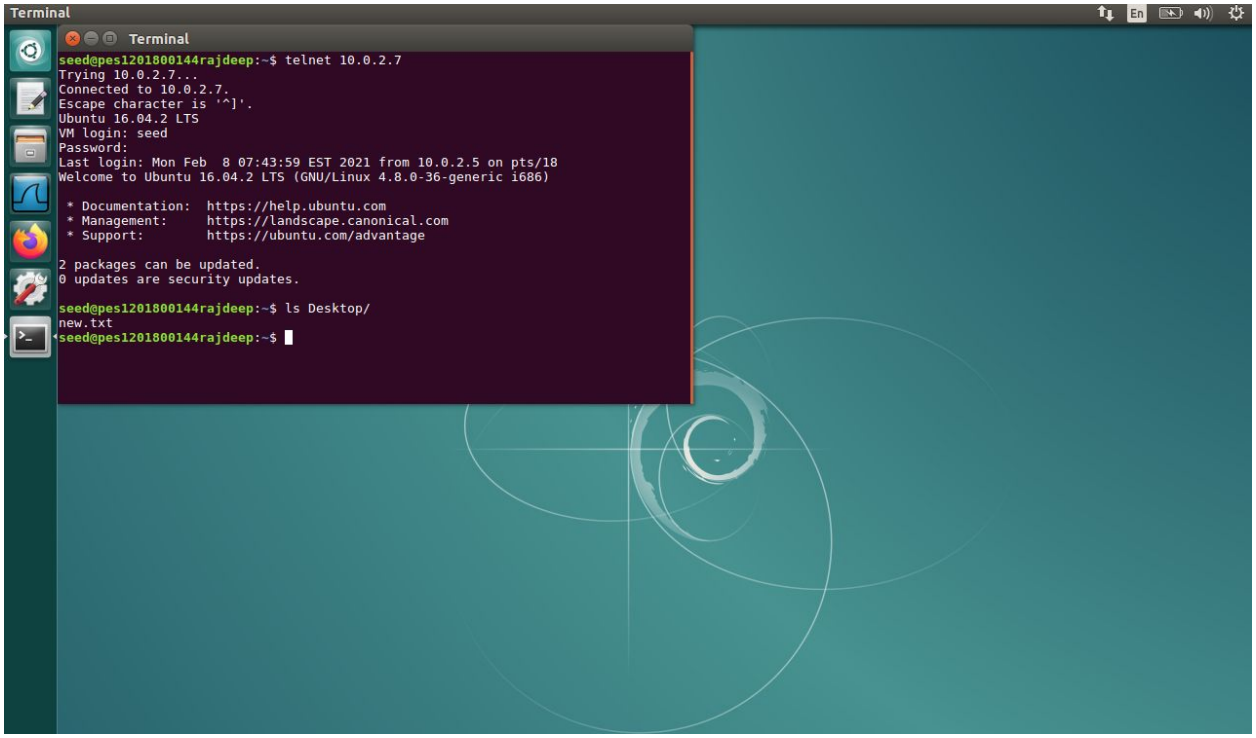



Screenshot 4.1.5: When the attacker executes the command, multiple retransmissions happen. The server thinks that the connection is lost and retransmits packets but is continuously dropped by the client.

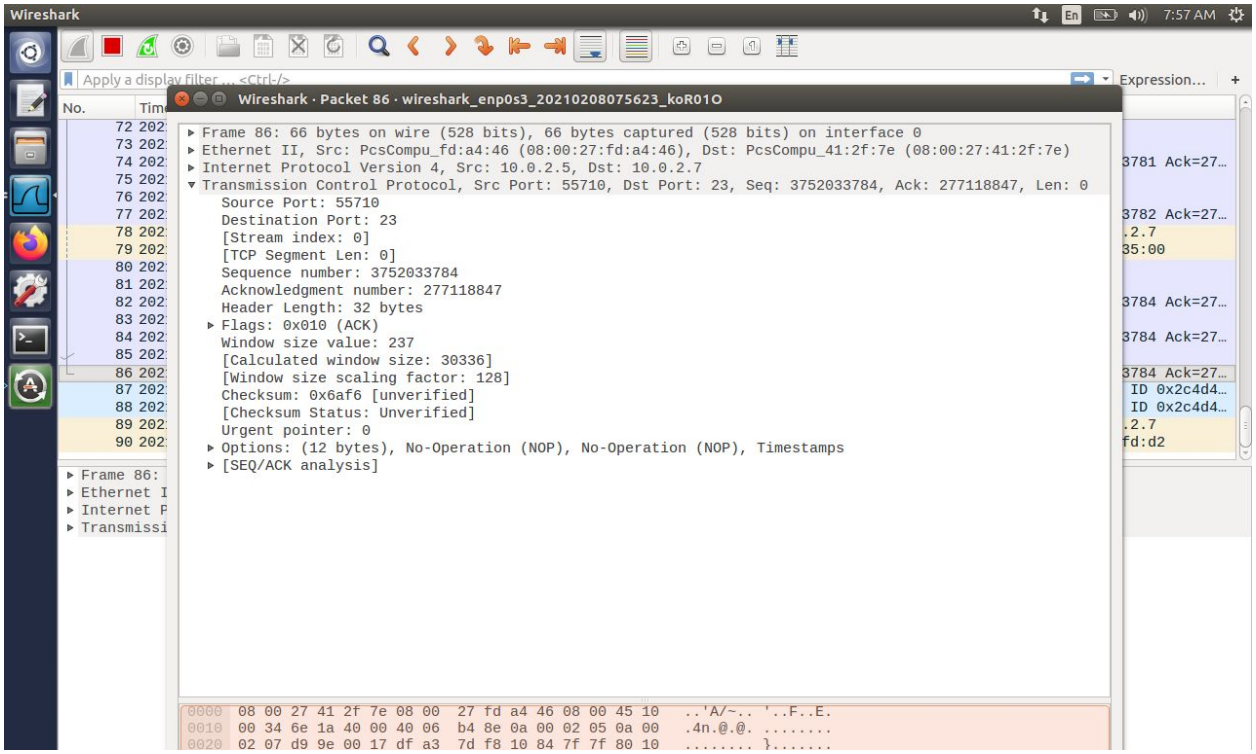



Screenshot 4.1.6: Also, in the client machine, the connection is frozen and the only way to exit the terminal is to forcibly exit.

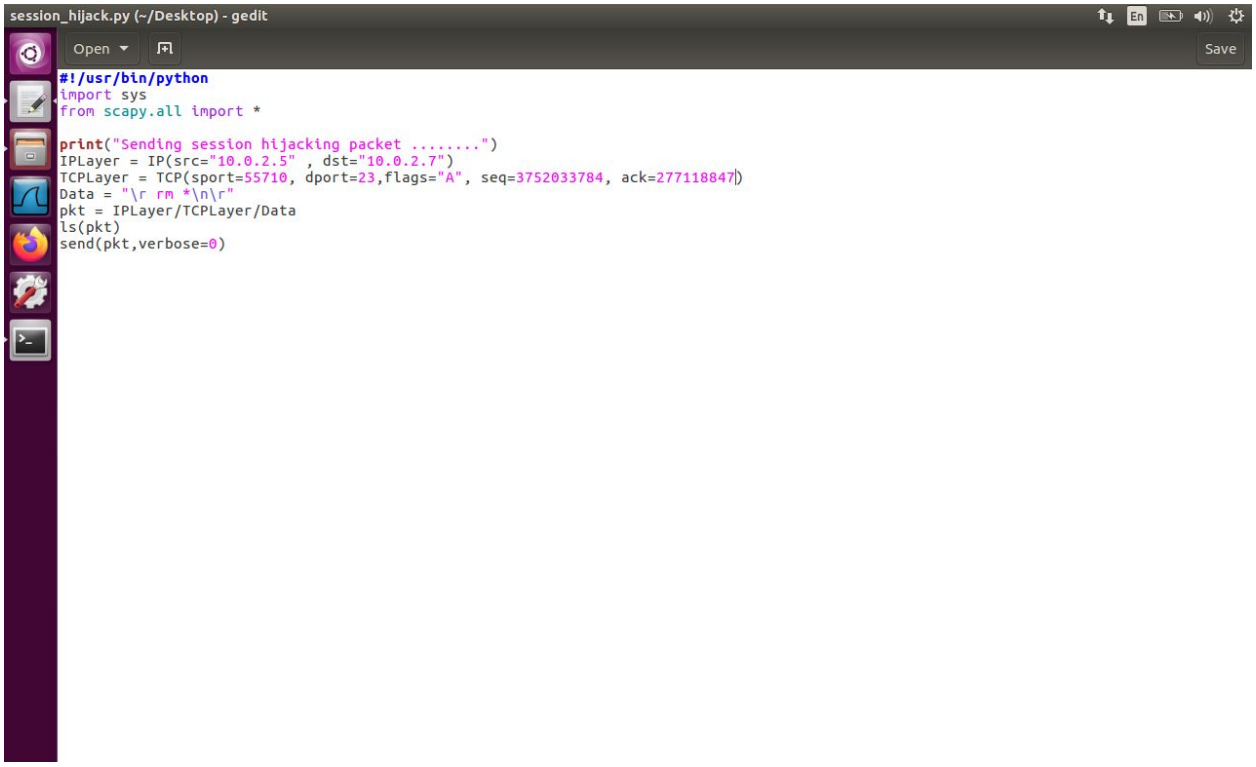
## 4.2 Using scapy python script:



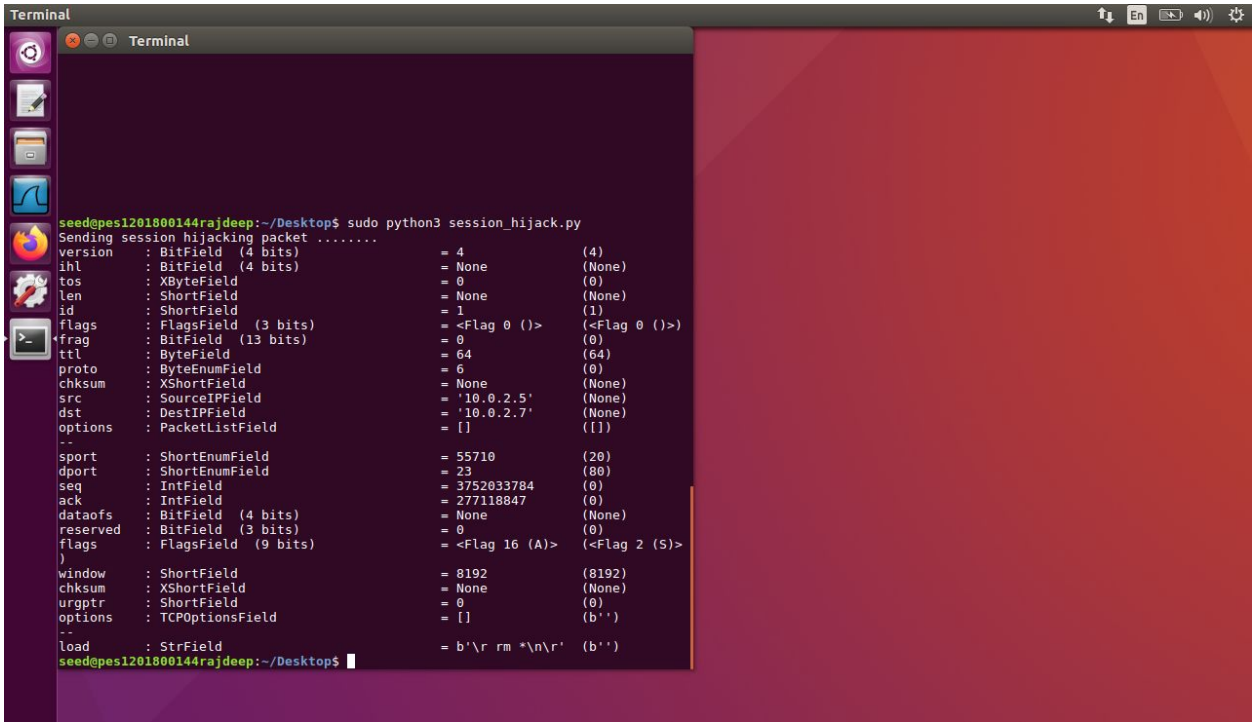

Screenshot 4.2.1: Client connects to server using telnet



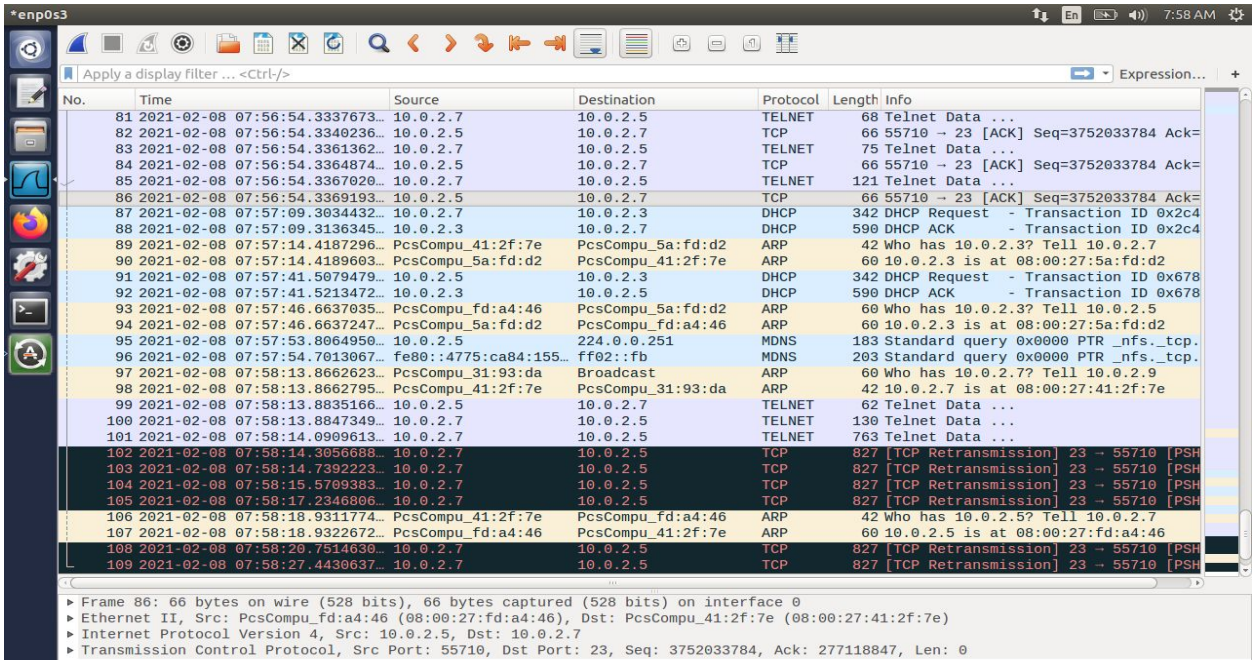

Screenshot 4.2.2: Last packet details in wireshark for using port, sequence number and acknowledgement number.



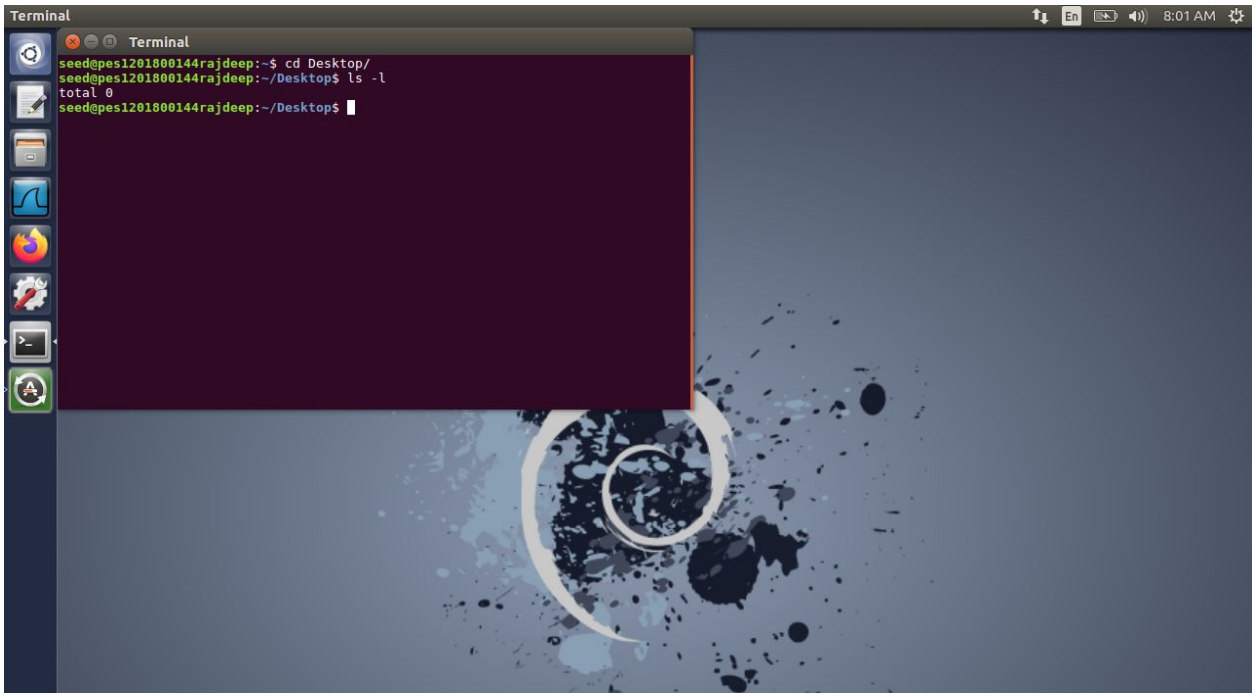

Screenshot 4.2.3: Attacker code snippet with port, sequence number and acknowledgement number.




Screenshot 4.2.4: Attacker executes the code




Screenshot 4.2.5: When the attacker executes the command, multiple retransmissions happen. The server thinks that the connection is lost and retransmits packets but is continuously dropped by the client.

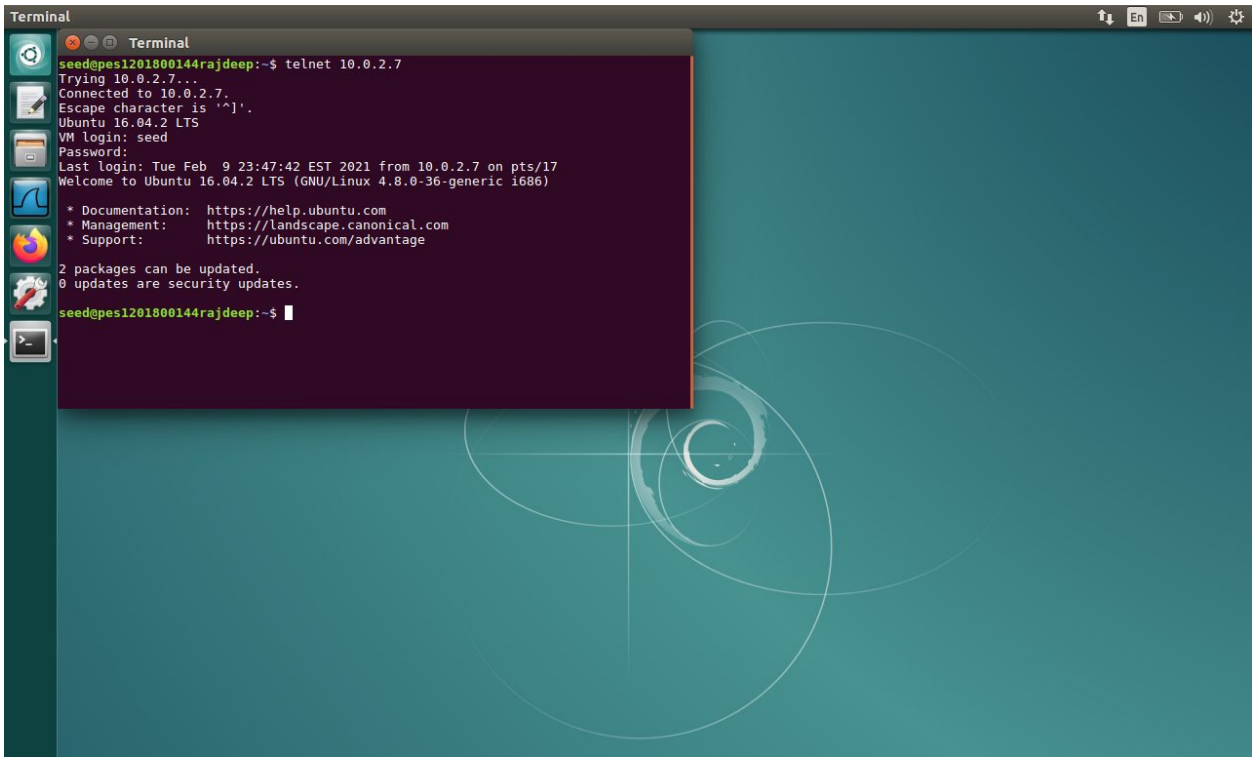



Screenshot 4.2.6: The file(new.txt) in the server machine gets deleted

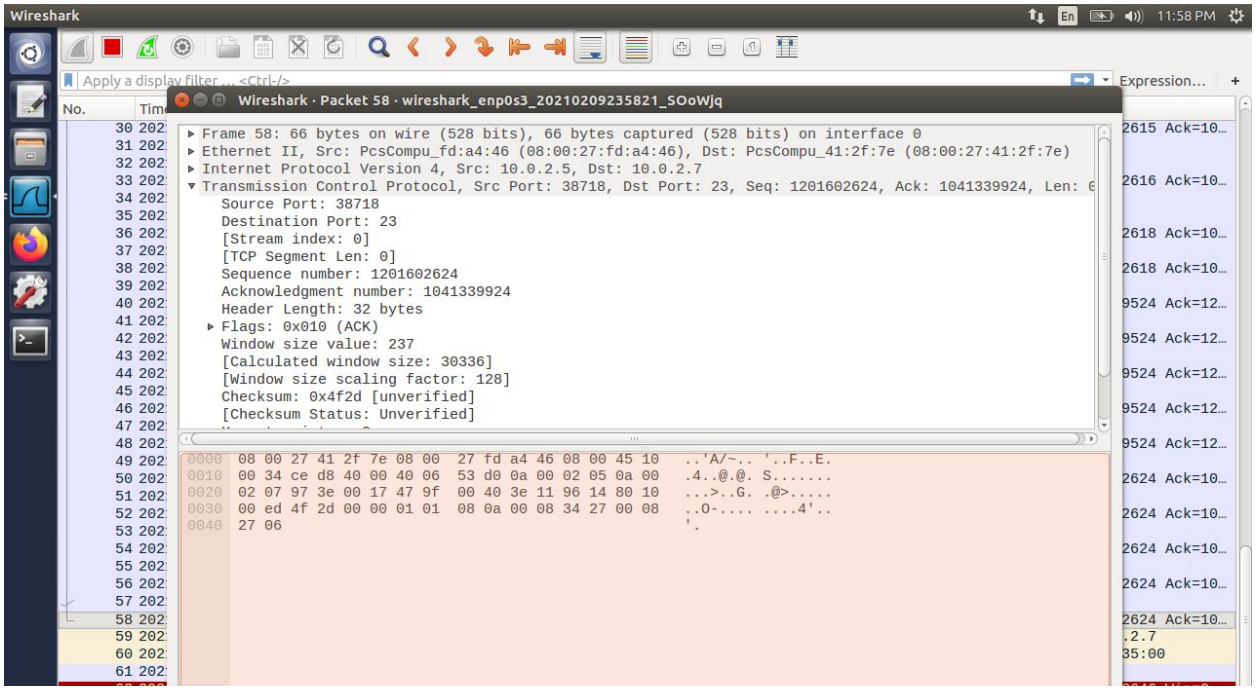
==========

# TASK 5:

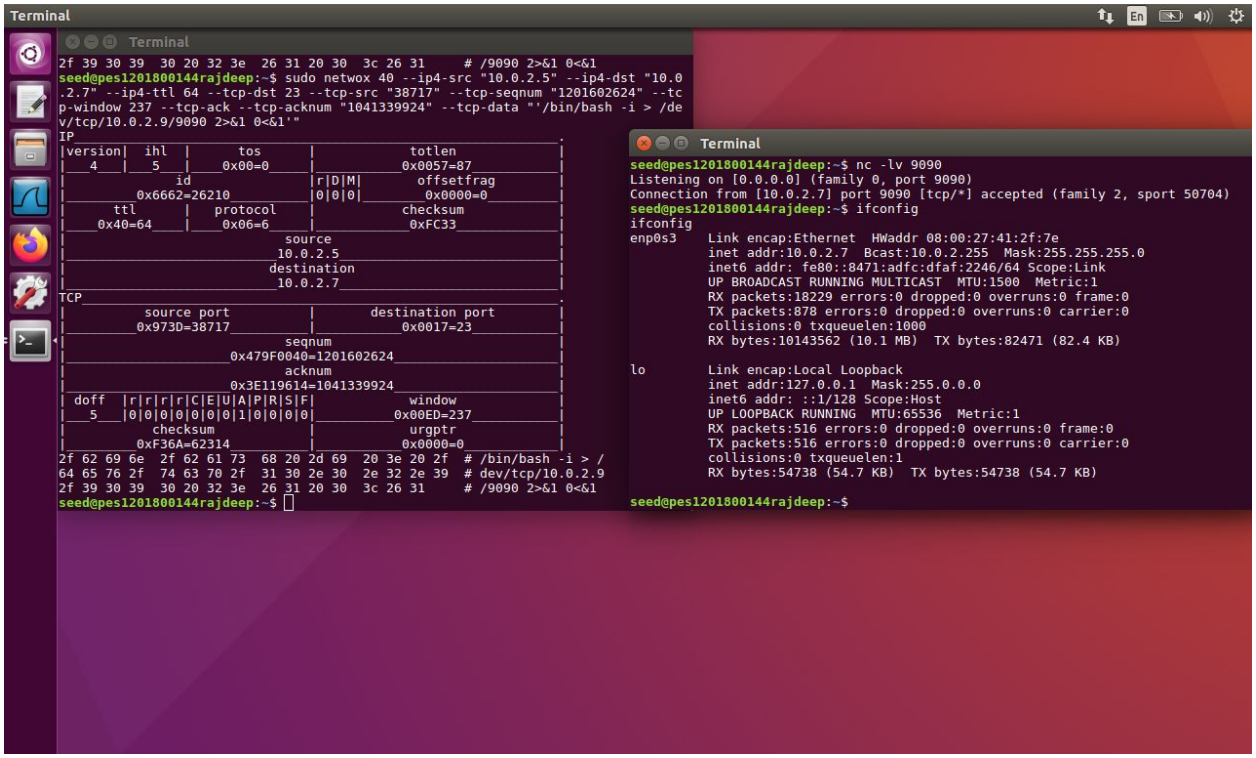
## 5.1 Using netwox command:




Screenshot 5.1.1: Client connects to server using telnet

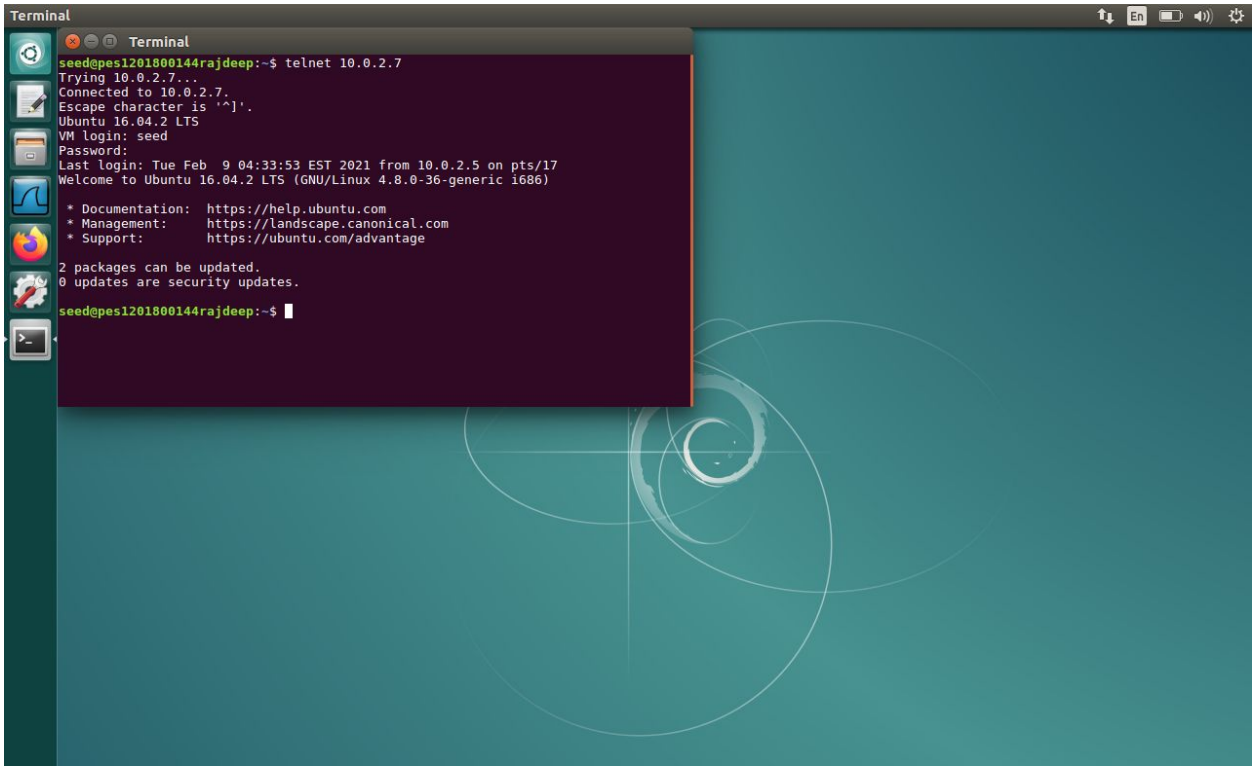



Screenshot 5.1.2: Last packet details for the above telnet connection

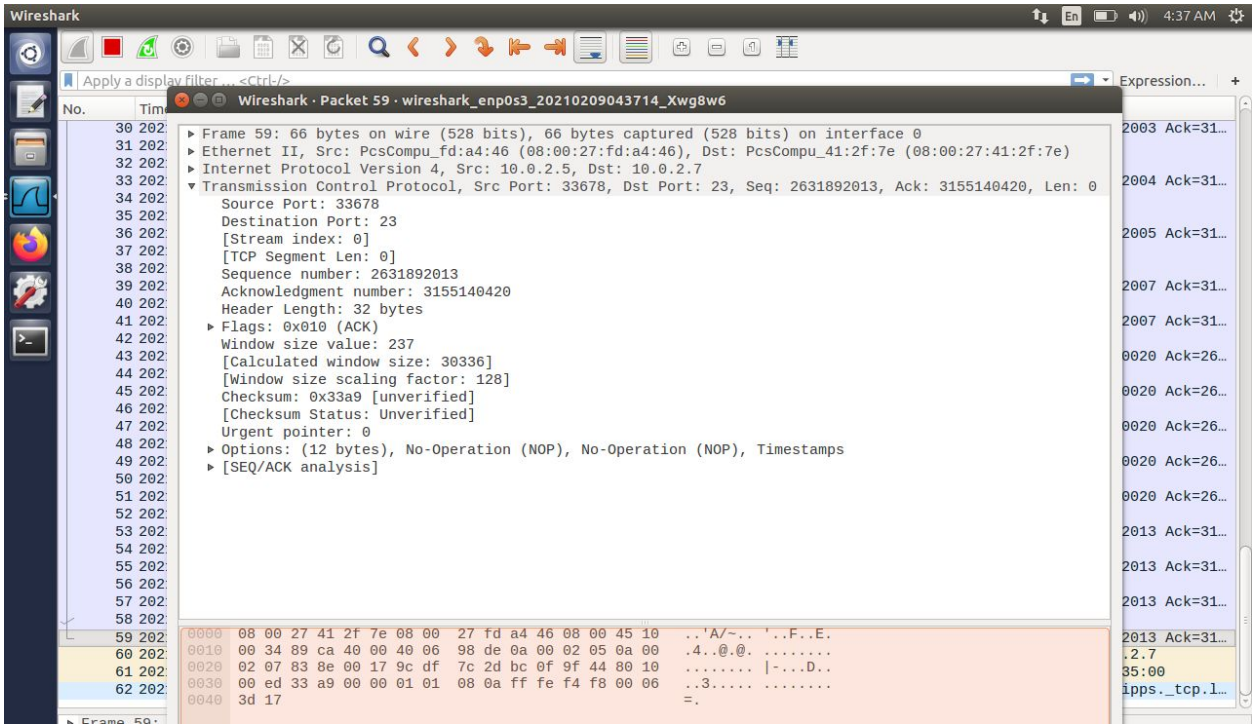



Screenshot 5.1.3: This screenshot shows that the attacker executes the netwox command and listens on another terminal to netcat on port 9090. The connection is successful as we can see that the ifconfig command shows the IP address of the server.

## 5.2 Using Scapy:

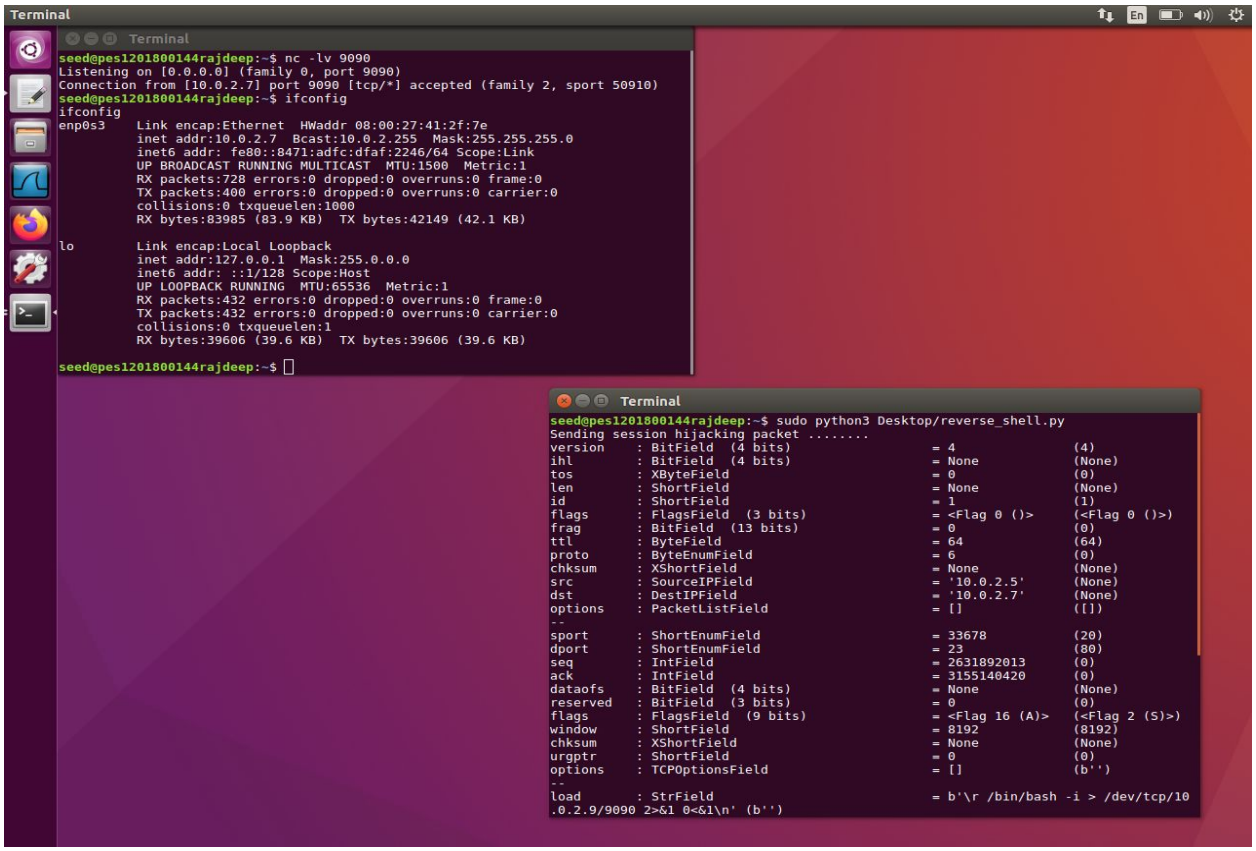



Screenshot 5.2.1: Client machine connects to server machine via telnet




Screenshot 5.2.2: Last TCP packet in wireshark for telnet between client and server.


Screenshot 5.2.3: Attacker machine scapy code with sequence number, acknowledgement number and port




Screenshot 5.2.4: Attacker executes code while listening on port 9090. Reverse shell successful and it can be seen that the ifconfig command shows server's IP.

**ADDITIONAL OBSERVATION:**

**This is a spoofing attack since the attacker sends some payload malicious data to the client which is sent to the server through client-server telnet connection.**
**In this attack, the TCP packet with payload as the reverse shell command is sent from attacker to client machine which goes to the server machine through the telnet connection. Hence, the attacker gets remote shell access to the server's shell. In part 1 of task 5, we use netwox to execute this whereas we use a scapy python script for part 2.**

==========