**LAB ASSIGNMENT 1** **SJSU ID**: 012018487 **NAME**: NIVEDHITHA RAMARATHNAM KRISHNA

## SYN FLOOD ATTACK:

- A SYN flood attack is the kind of attack where in the attacker continuously sends SYN packets to the server in an attempt to consume all of the servers resources thereby leaving the server unavailable to other users on the network.
- In this task following systems are on the same LAN:
  VM0 – client
  VM1 – server
  VM2 – attacker
- Initially a telnet connection was established between vm0 and vm1 by using telnet command. There was smooth flow of traffic and there was no interruption. In this case the TCP_syncookies were disabled. The following images show the client and server before the attack.
- The attack was initiated from vm2 using netwox tool and as syn flooding started the connection was disrupted at the vm0's end and the following message was displayed.
- On observing the wireshark tool to see the server's end, all the TCP traffic was shown in red color indicating irregularity in TCP session. Also the following image shows that server was attacked with so many SYN packets that no ACK was sent.

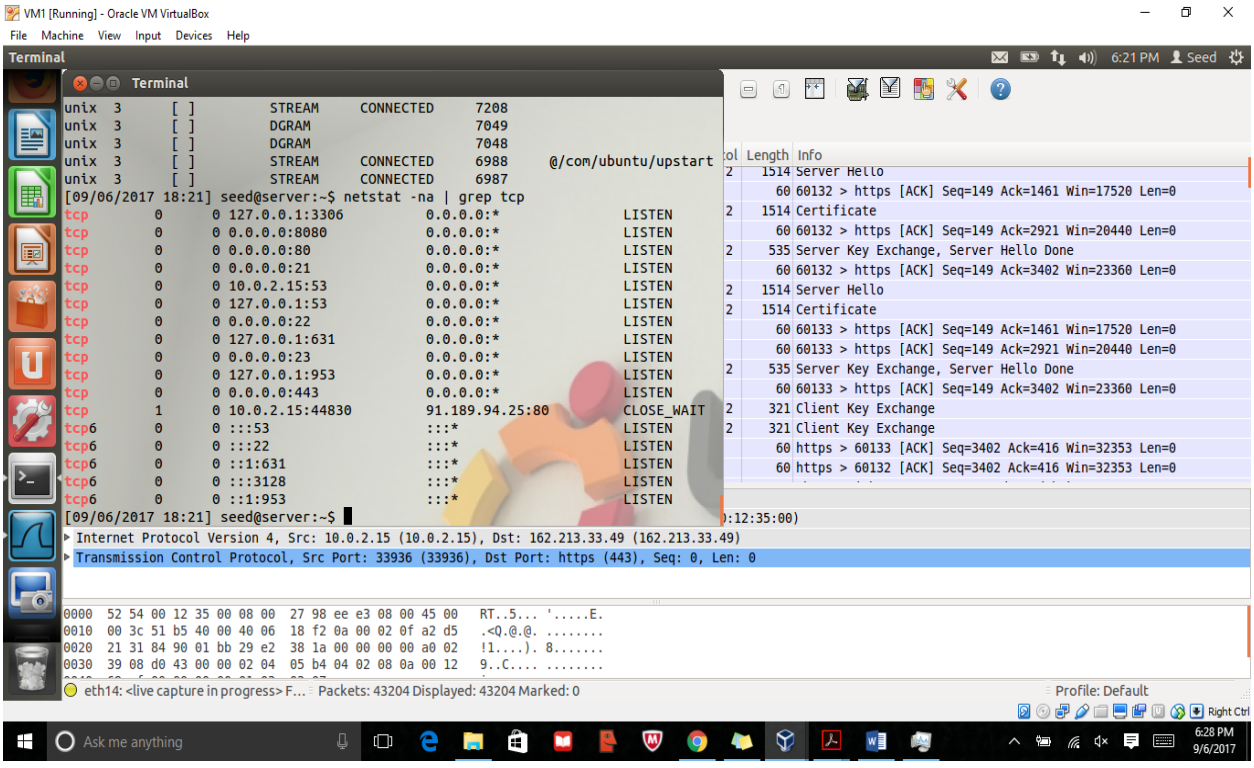


Figure shows that the server's TCP port 23 is listening for a connection to be established.

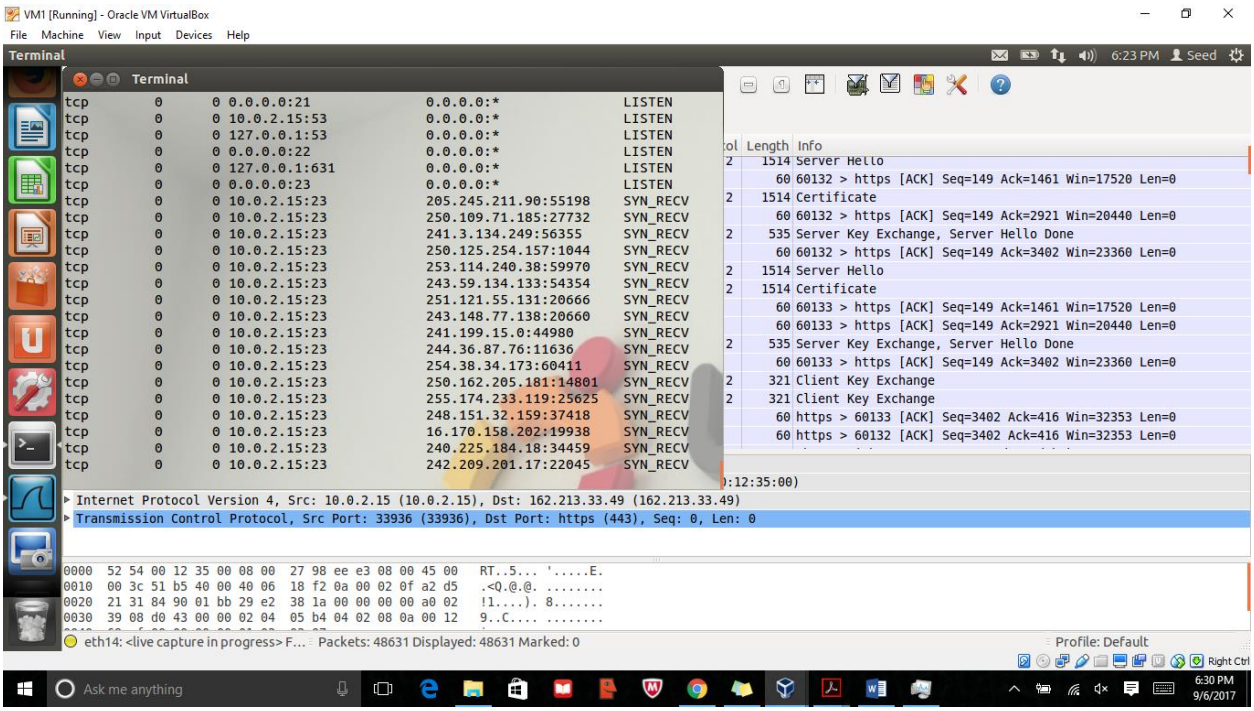


Figure shows the status of the connection after the attack as SYN_RECV.

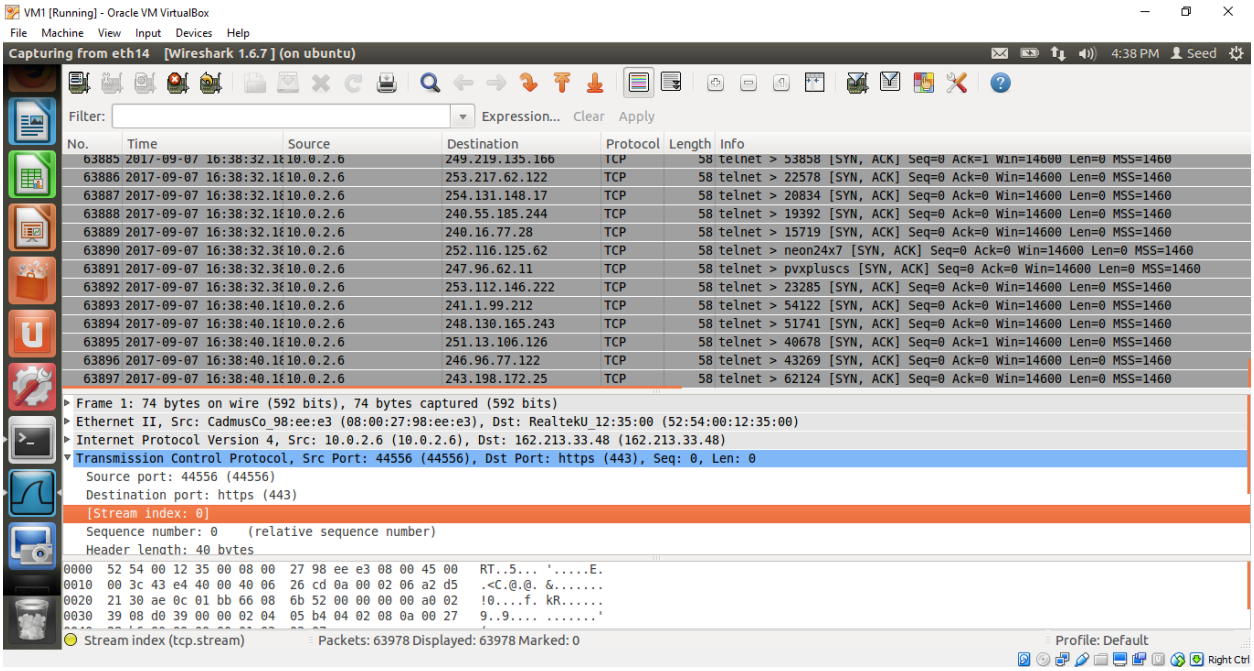


Figure shows wire shark analysis of server after the attack where in continuous SYN packets have been sent.

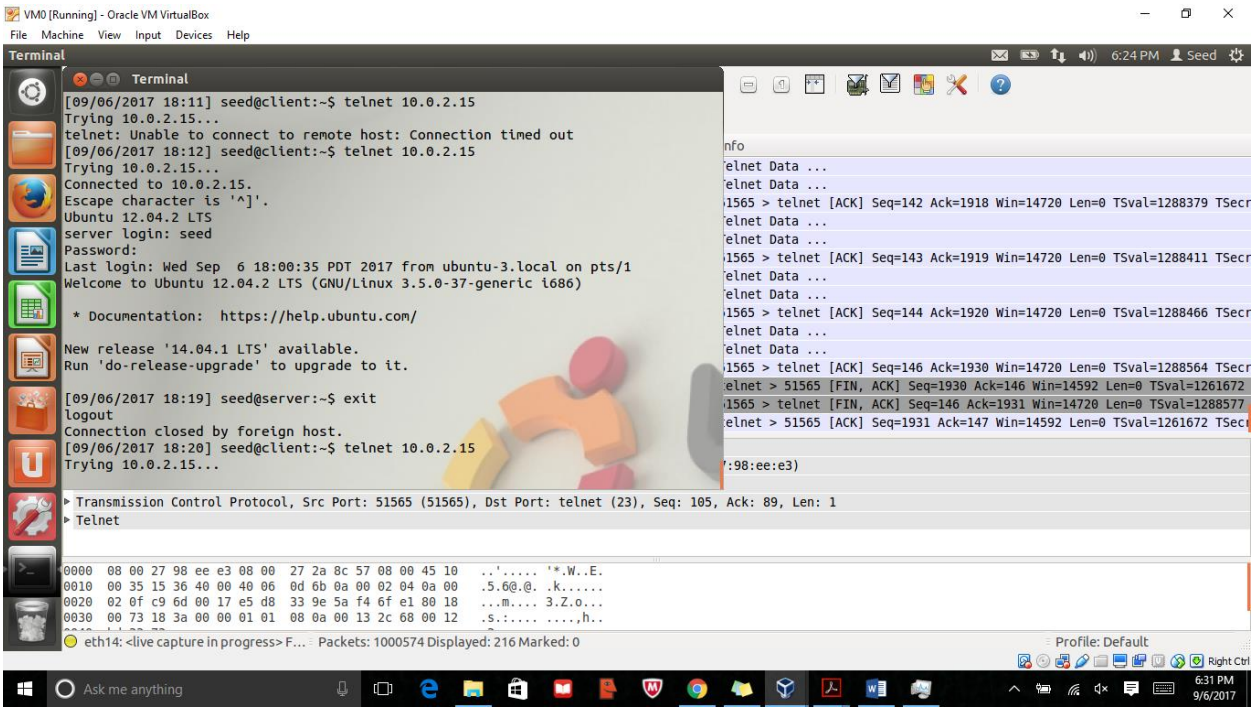


Figure shows client's connection being disrupted due to syn flood attack.

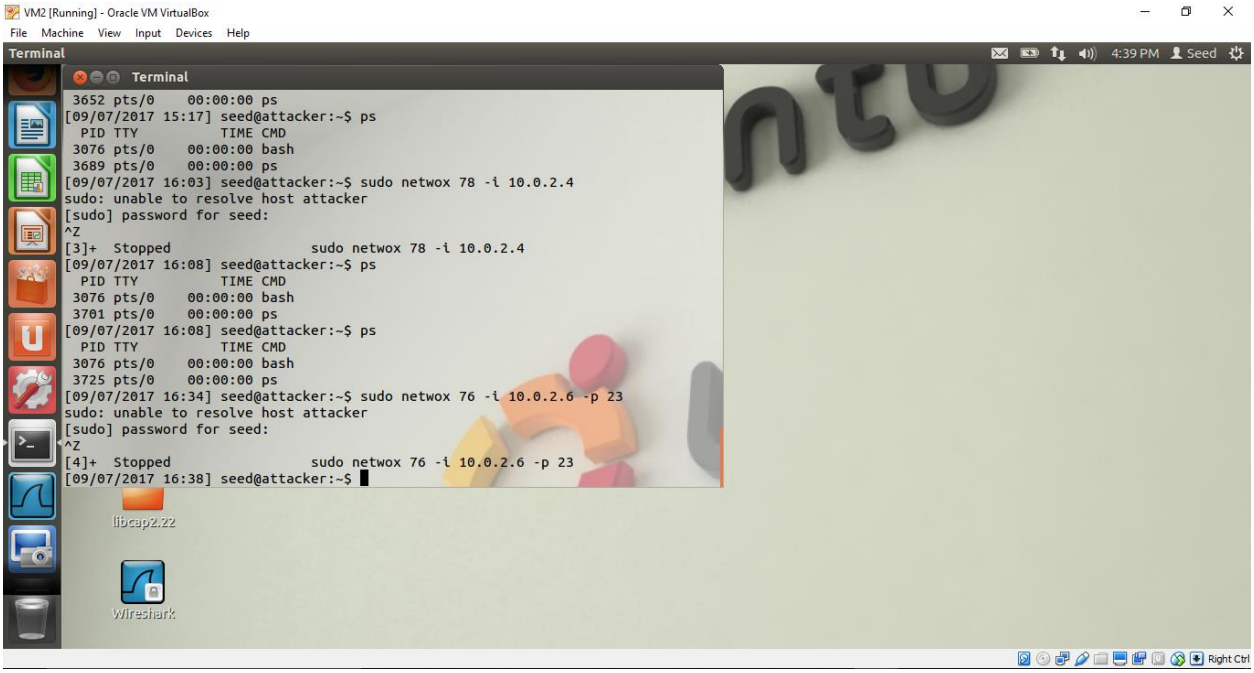


Figure shows attacker initiating the syn flood attack.

- In the second case TCP_syncookies are being enabled and then attack is observed.
- In this case there is not much effect of the attack, because the server responds back with an ACK packet with initial sequence number called cookie.

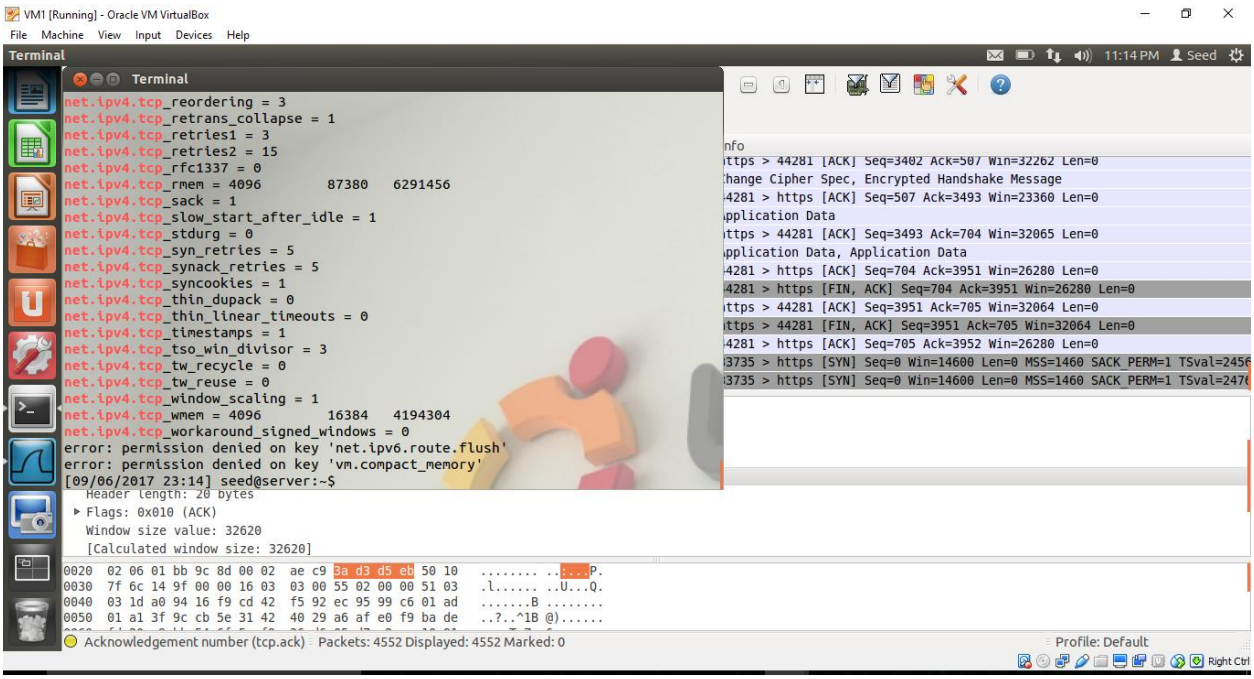


Figure shows setting the cookie to enabled state.

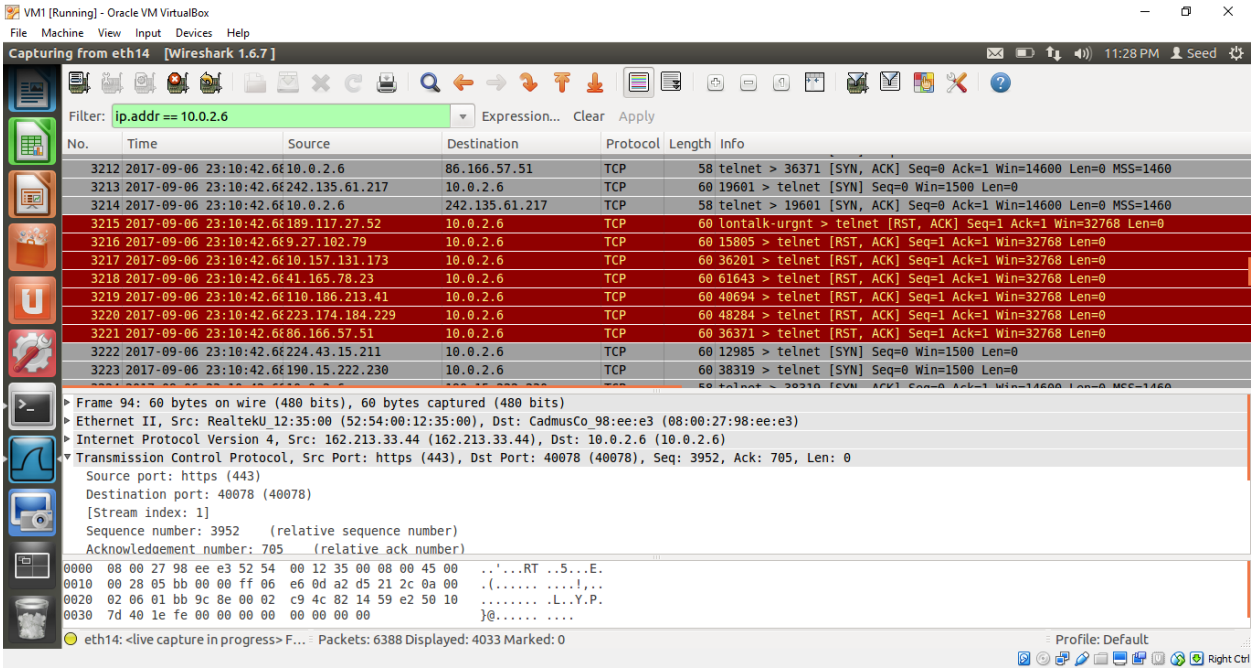


Figure shows wireshark analysis of server with the cookie enabled.

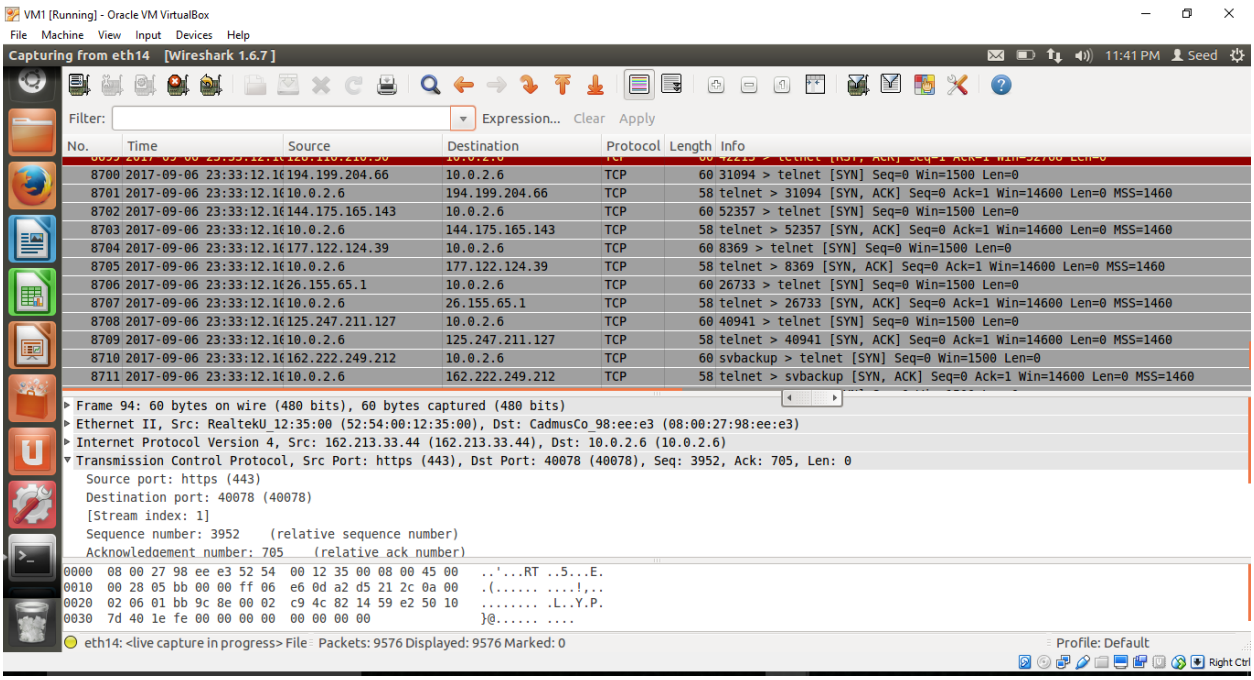


Figure shows server sending back the acknowledgment.

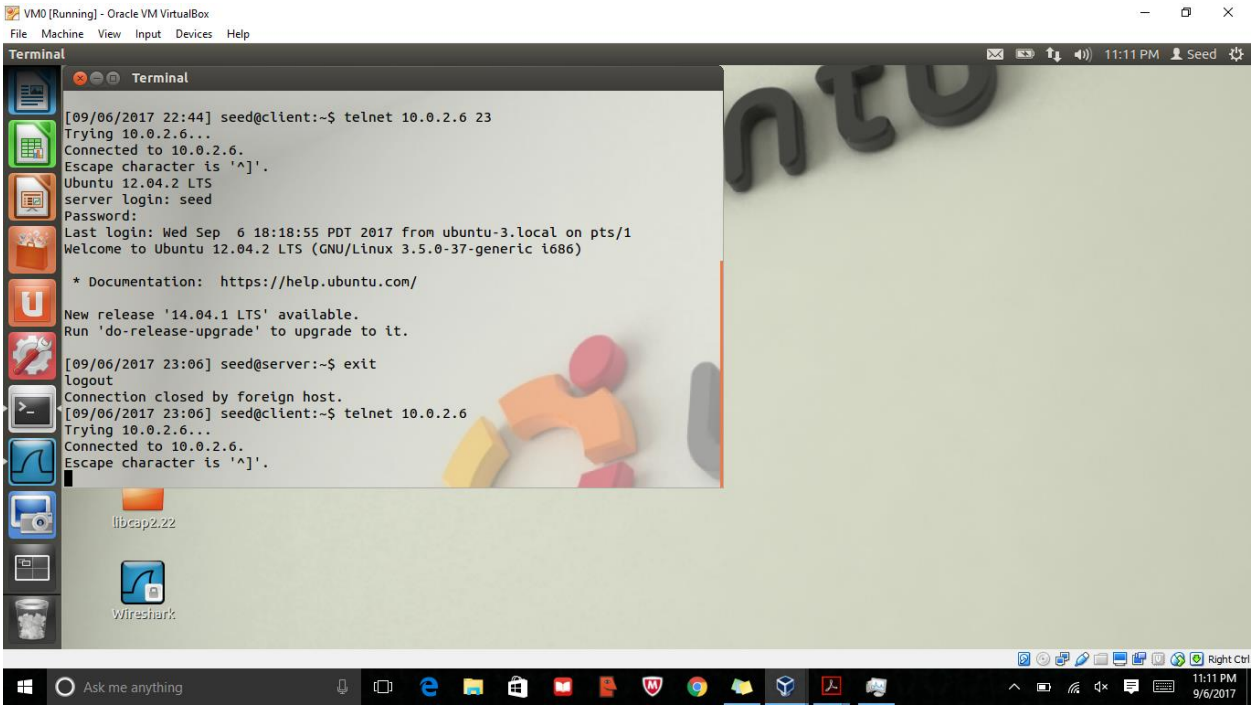


Figure shows connection at client's end when cookie is enabled.

## TCP RST ATTACK 1:

- In a TCP RST attack the connection simply breaks between the client and the server.
- The following images show the client, server and attacker's state before and after the attack.


Figure shows client before the attack where-in it is connected to the server through telnet and is able to view to files of server.

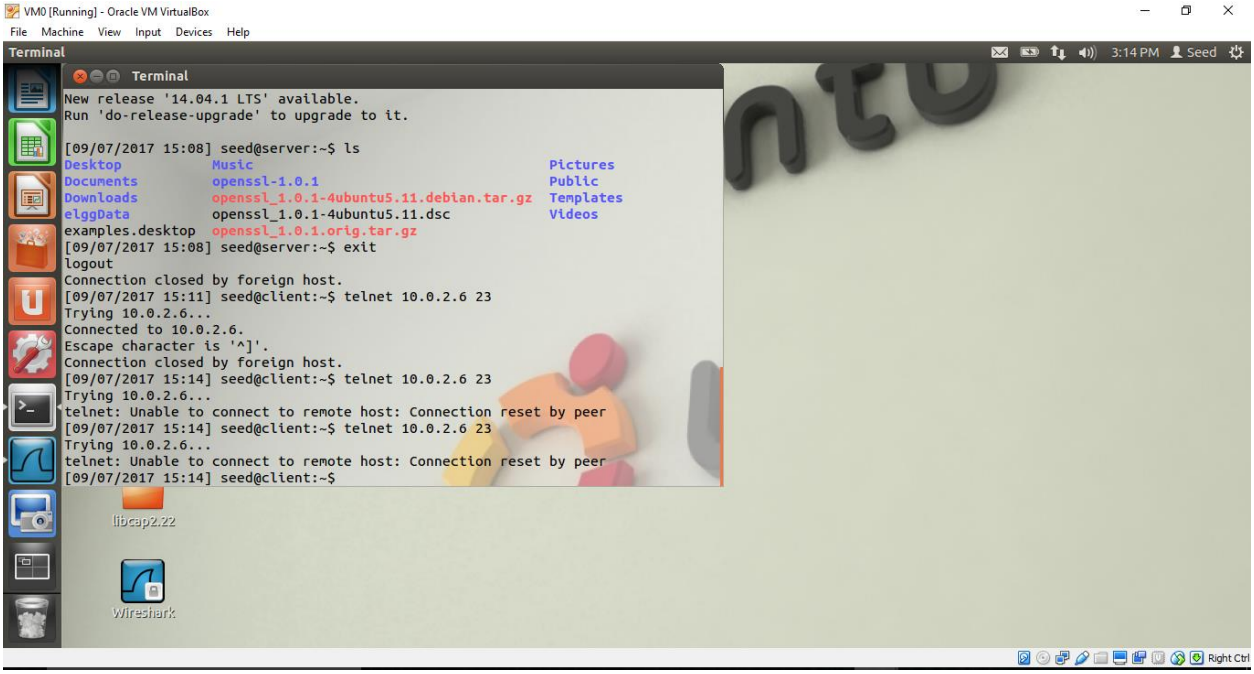


Figure shows client after the attack.

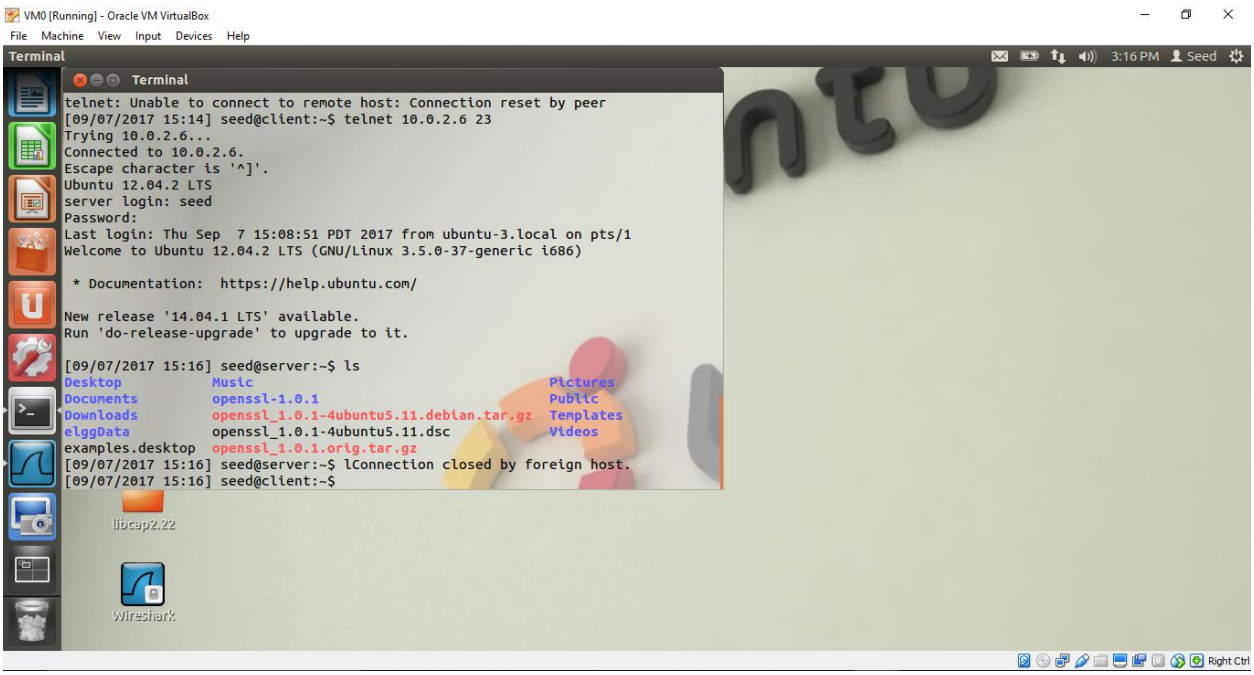


Figure shows client after the attack where-in the connection is closed by foreign host.

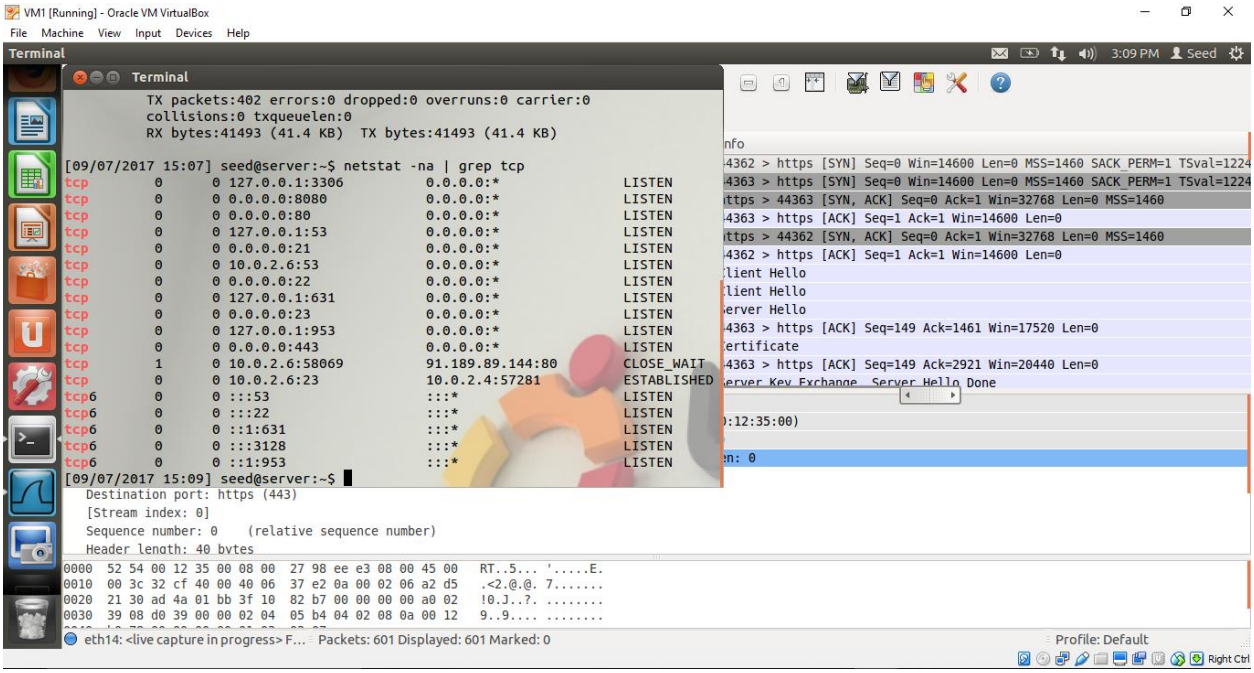


Figure shows server before the attack where-in connection is established at port 23 with the client.

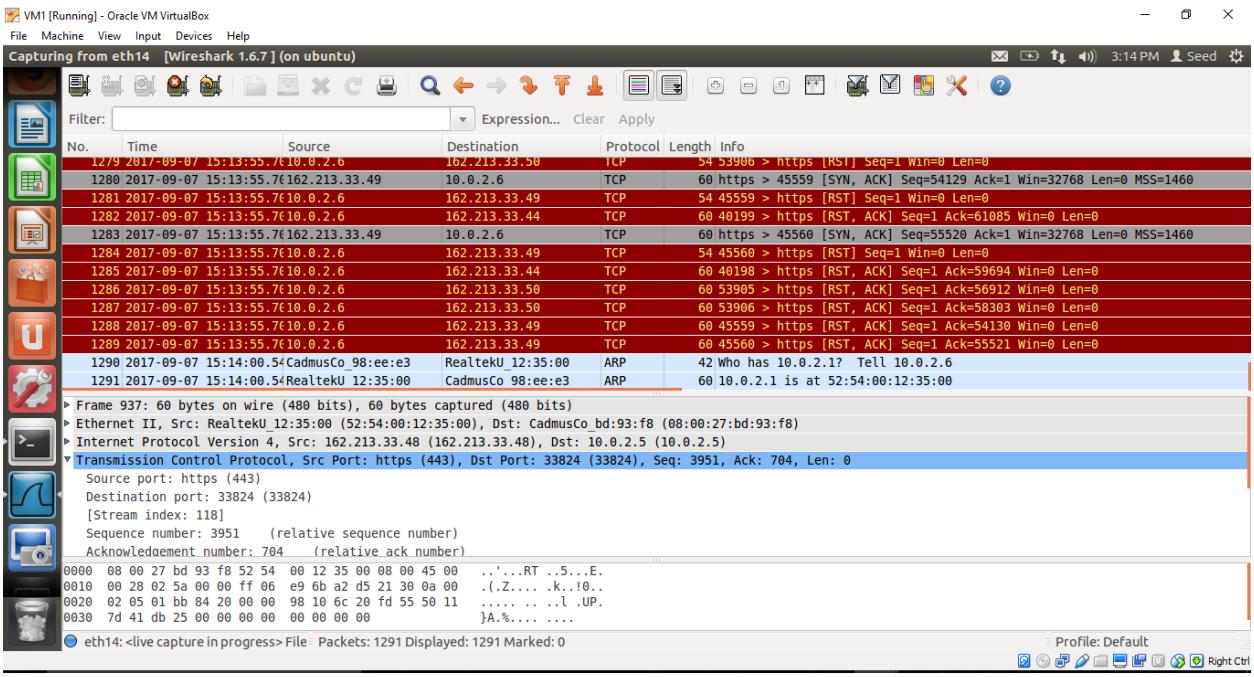


Figure shows wireshark analysis of server after the attack where-in RST flags are set and packets are sent to the server.

## TCP RST ATTACK 2:

- In this kind of RST attack the video connection to the URL (video streaming site) is disabled.

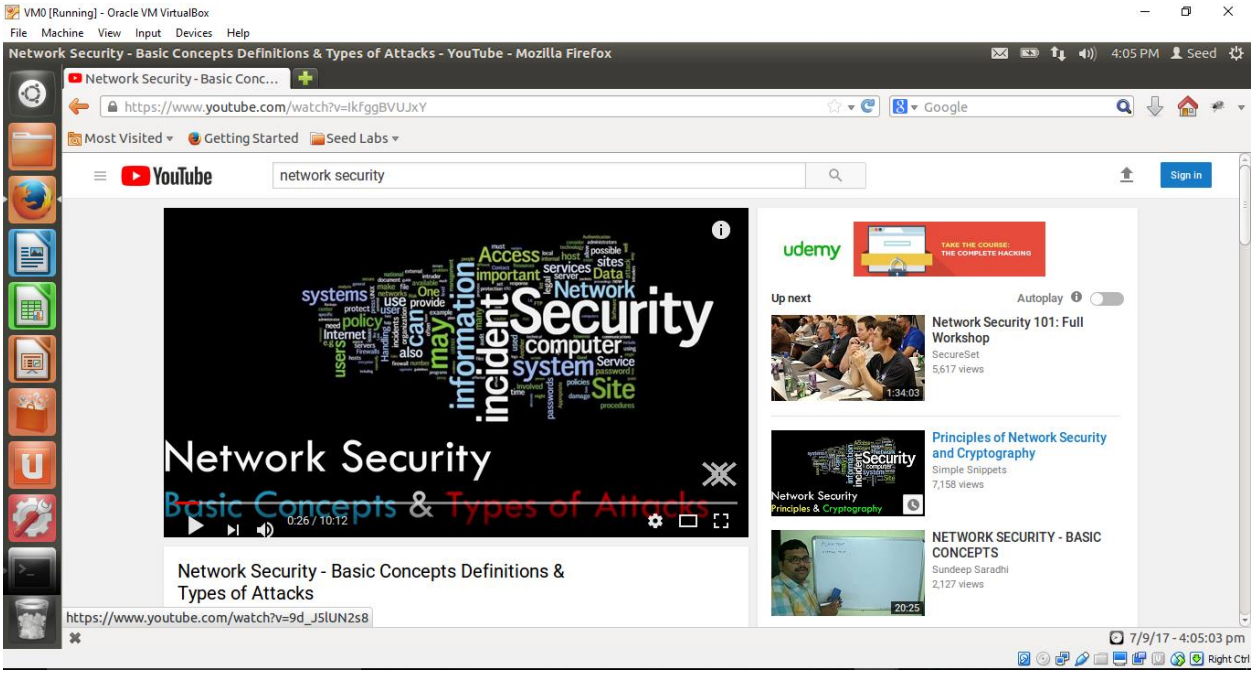


Figure shows client video streaming before the attack.

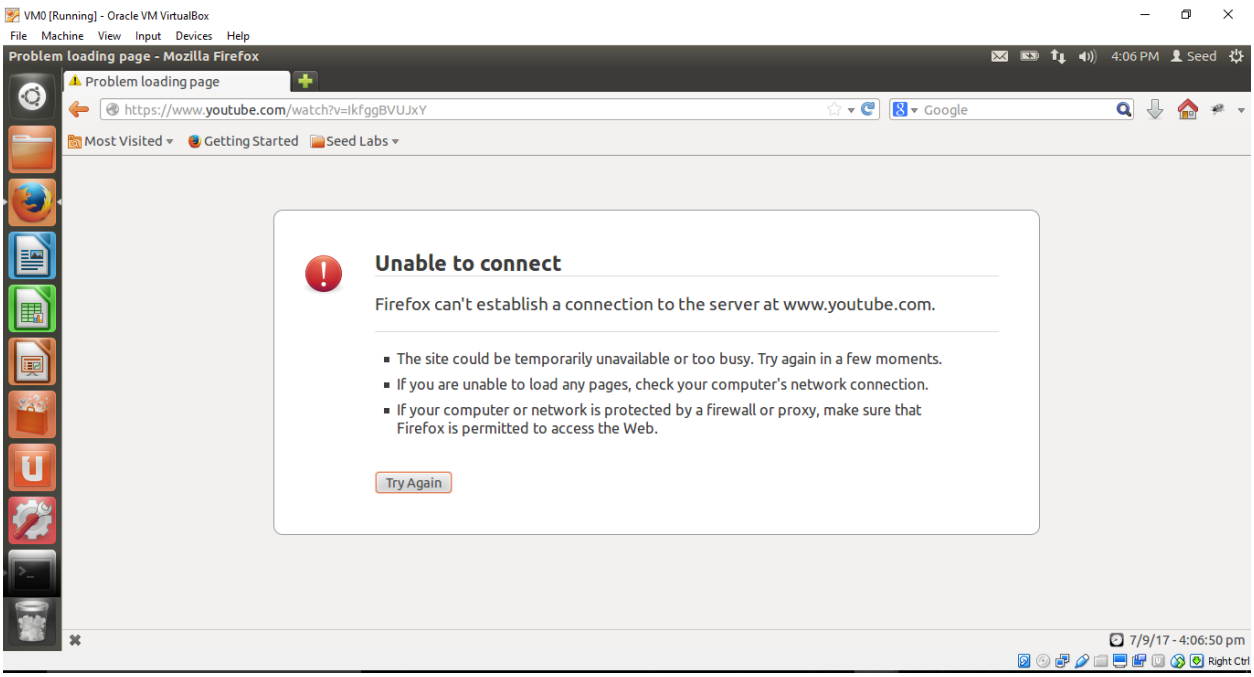


Figure shows client after attack.

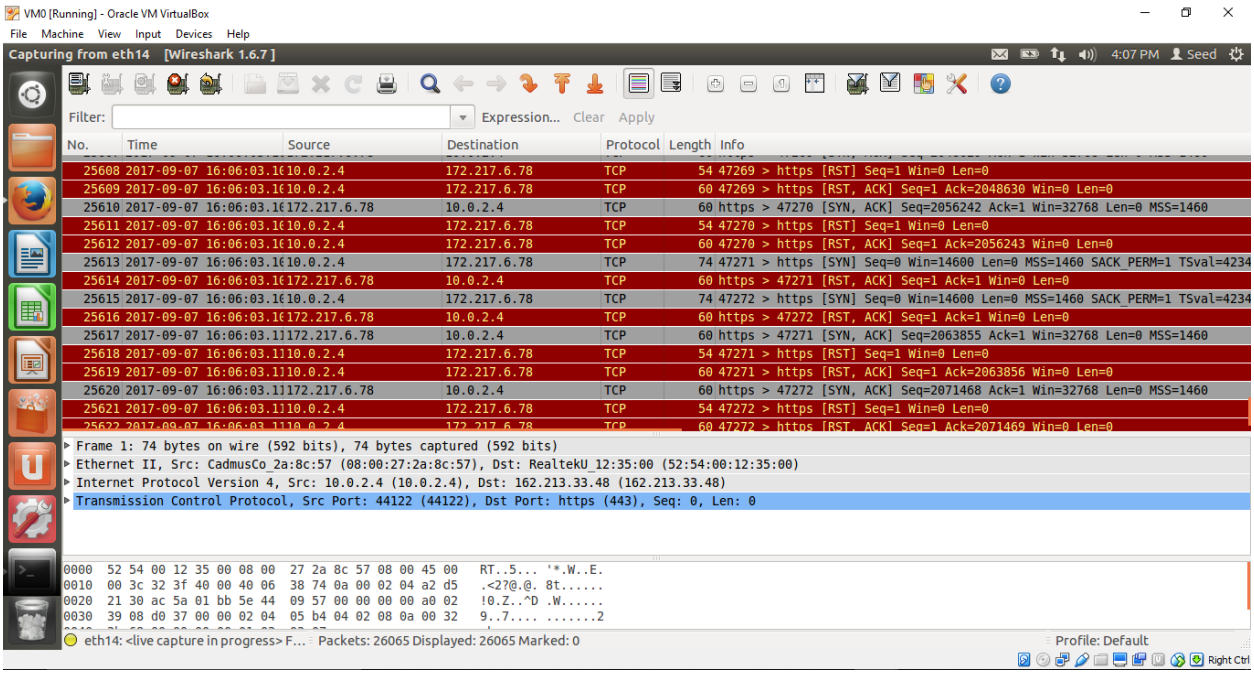


Figure shows client's wire-shark analysis after attack.

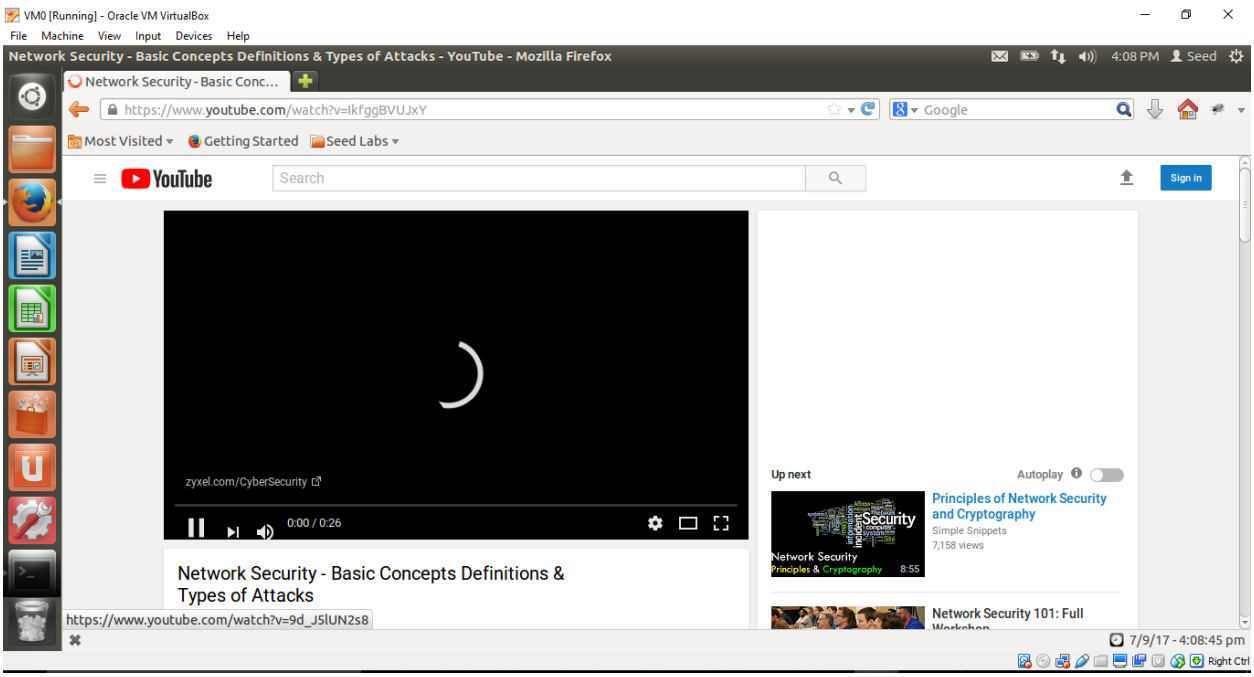


Figure shows client after attack has been stopped, the video starts reloading.

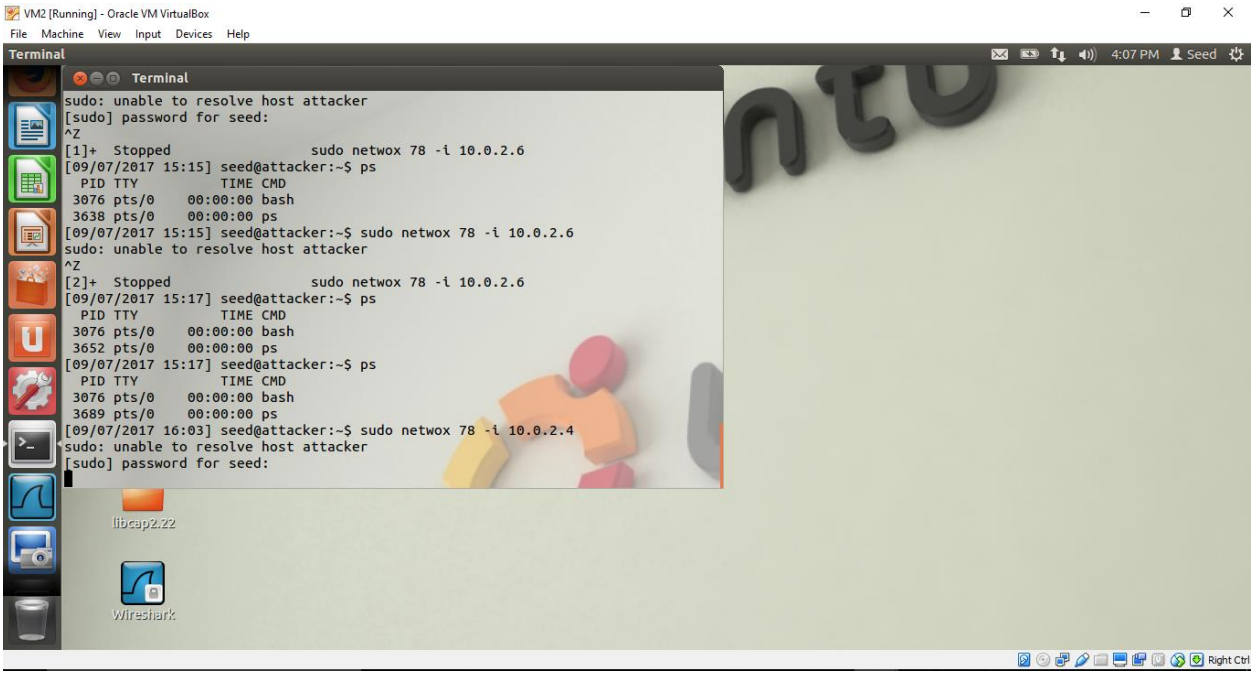


Figure shows attacker, initiating the RST attack.


Figure shows attacker stopping the attack.

## TCP Session hijacking:

- In this process the sniffing takes place first on an already existing TCP connection, where-in the attacker analyzes the packets and flow of network and then constructs a packet with the sequence number, ack number and port number appropriately.
- Server acknowledges and returns the result to the attacker under the assumption that it is from the client.

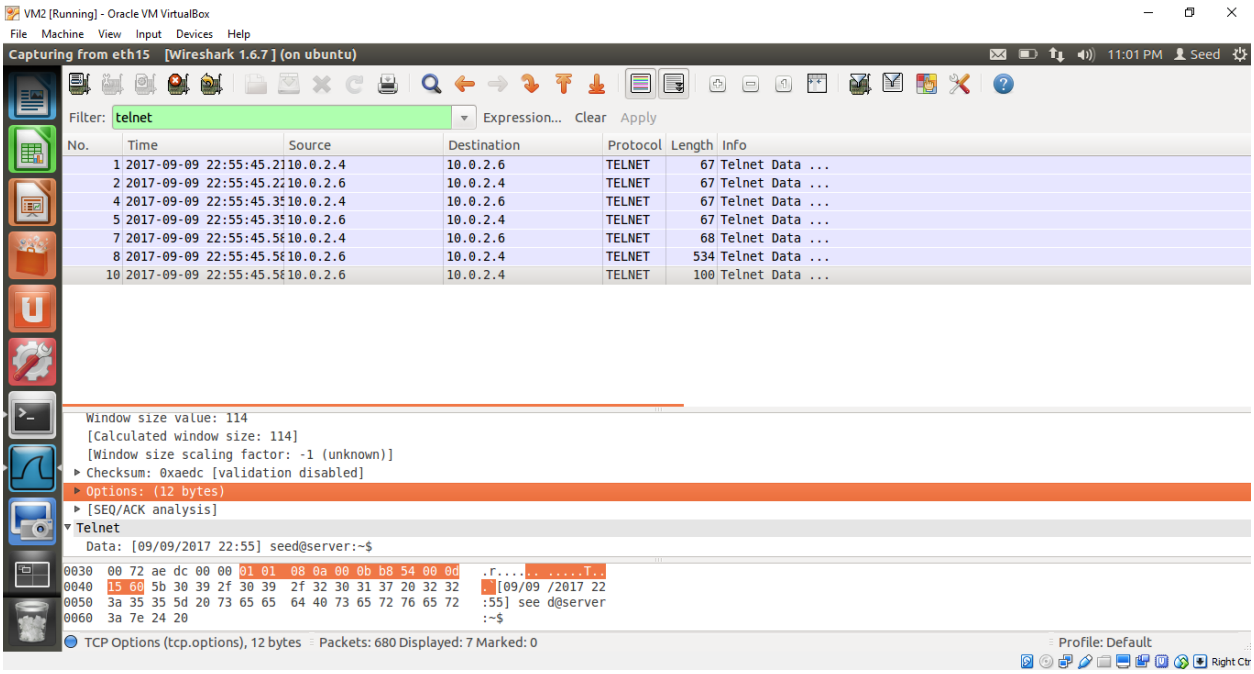

Figure shows existing connection between client and server on attacker's wireshark analysis.

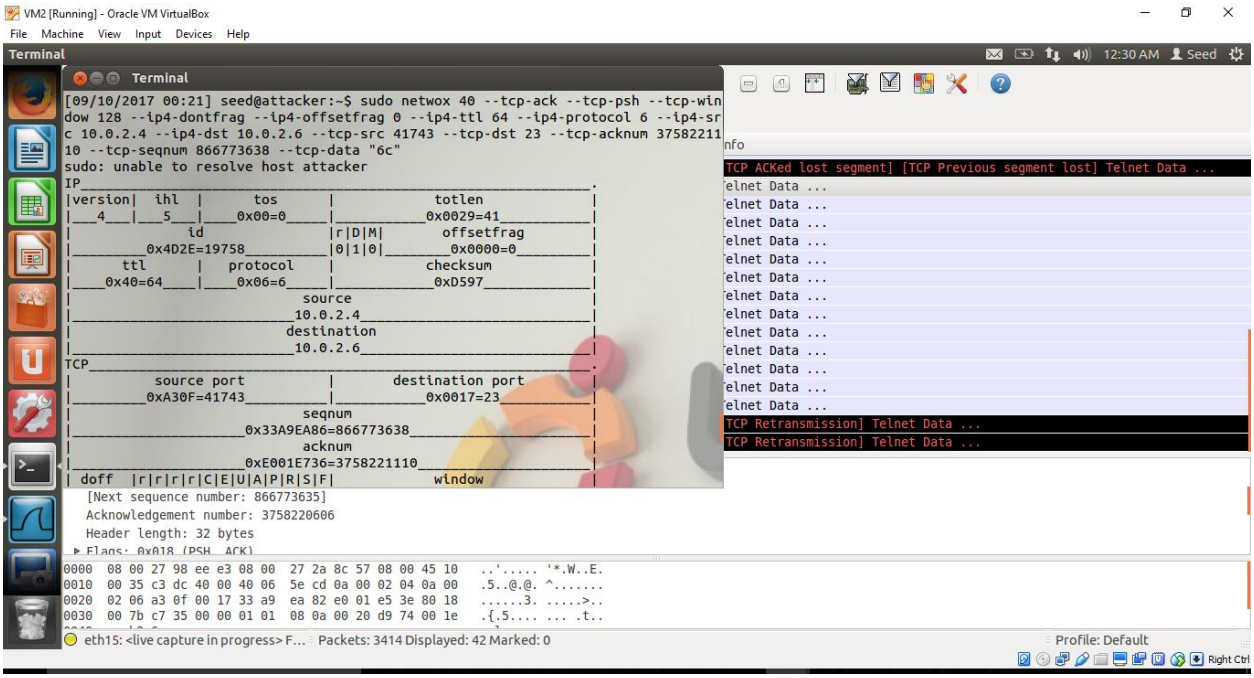


Figure shows attacker sending "ls" command part 1.

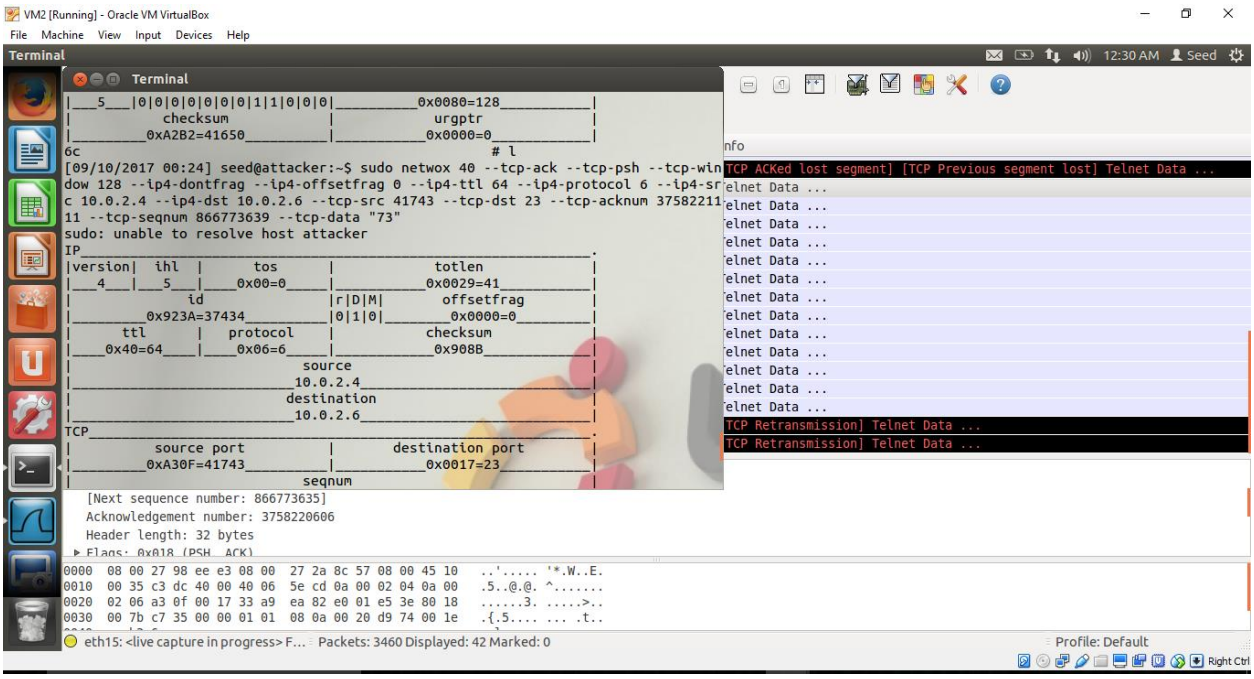


Figure shows attacker sending "ls" command part 2.

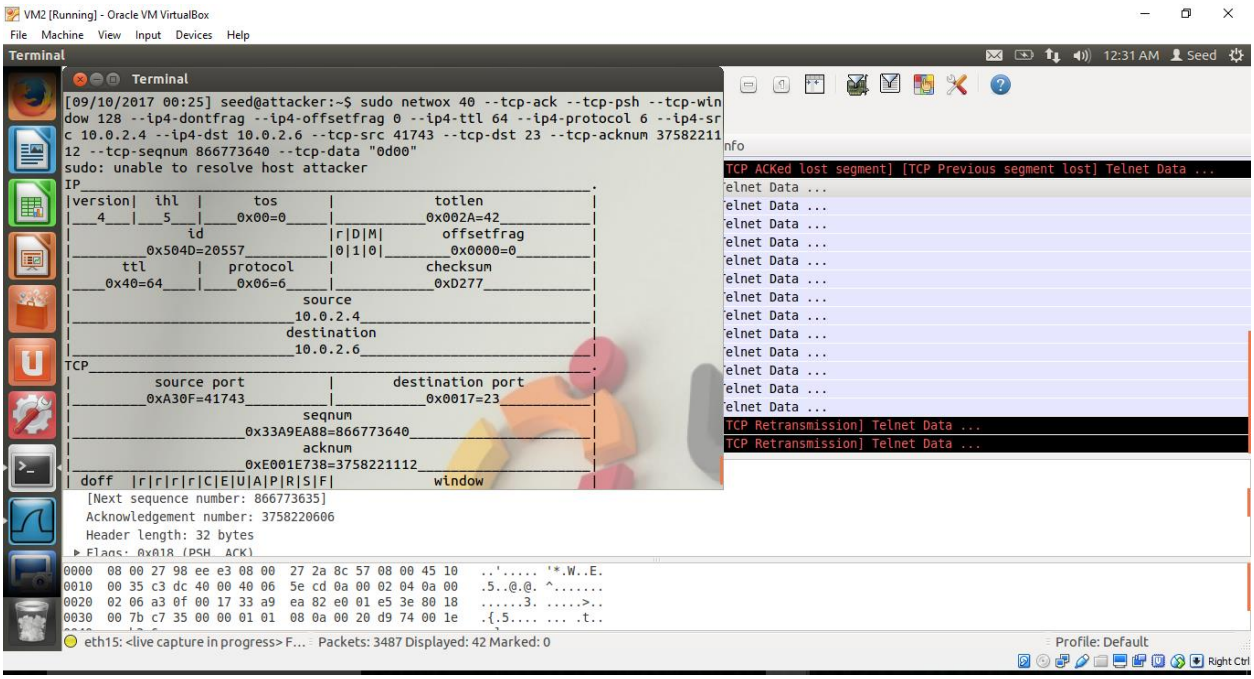


Figure shows attacker sending “ls” command part 3.

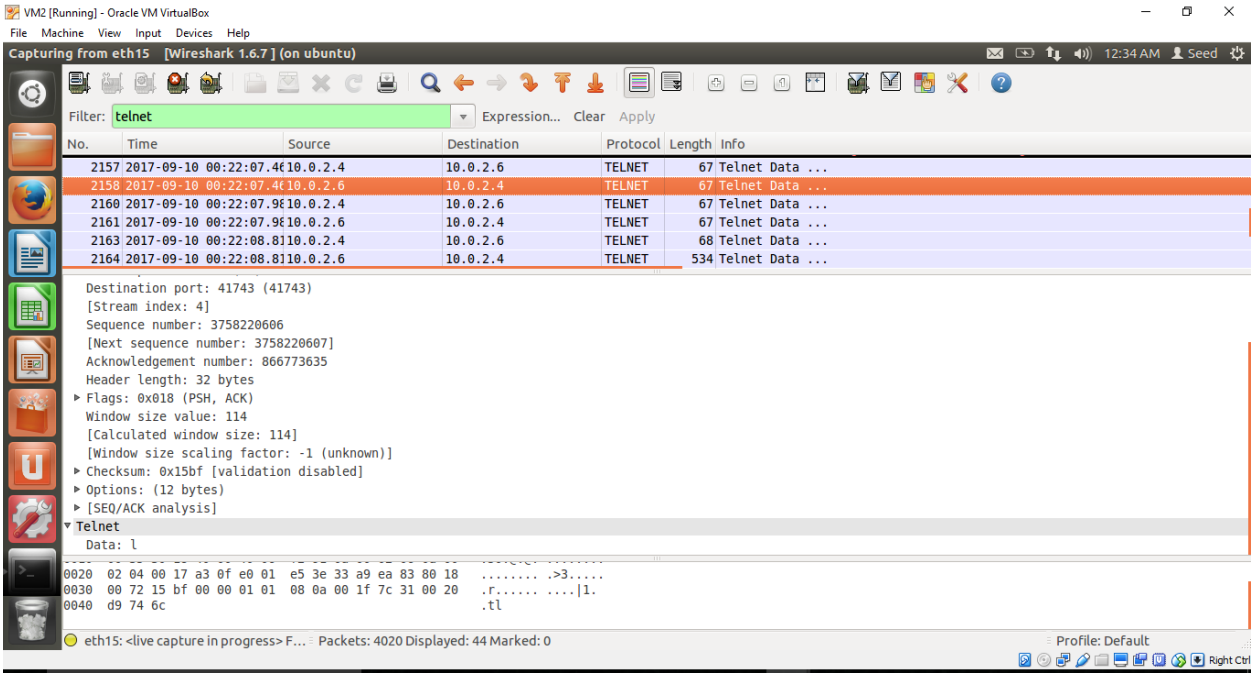


Figure shows server responding to “l”.

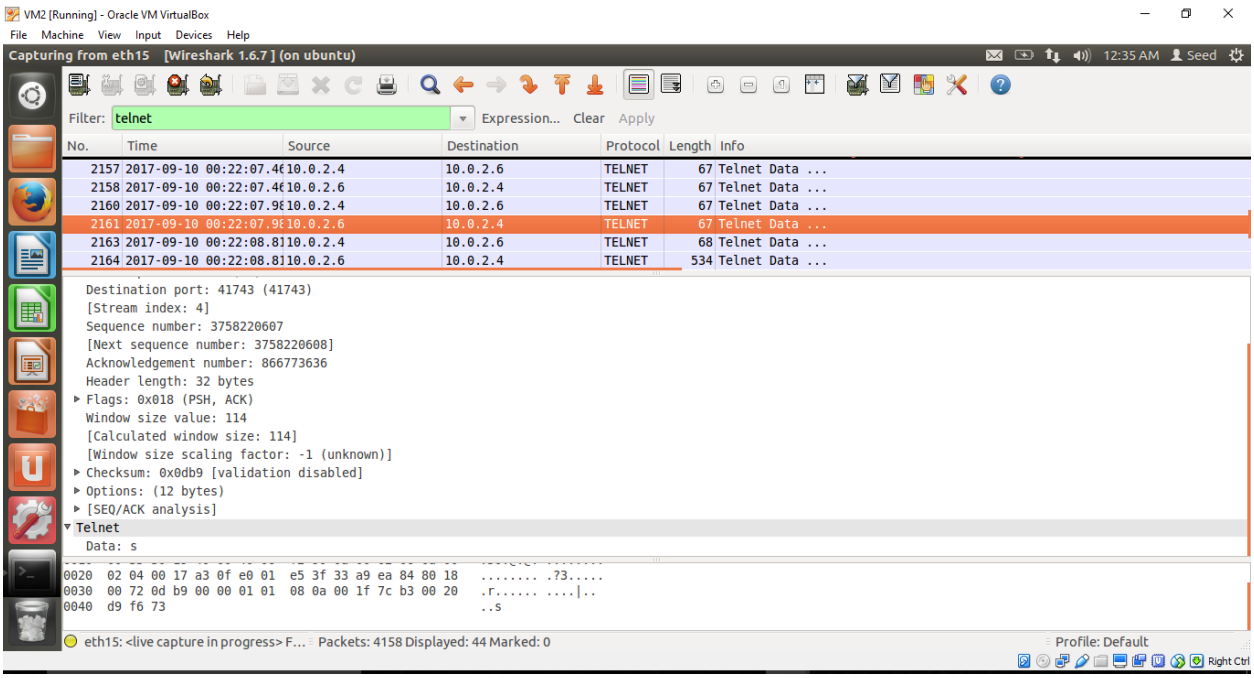


Figure shows server responding to “s”.

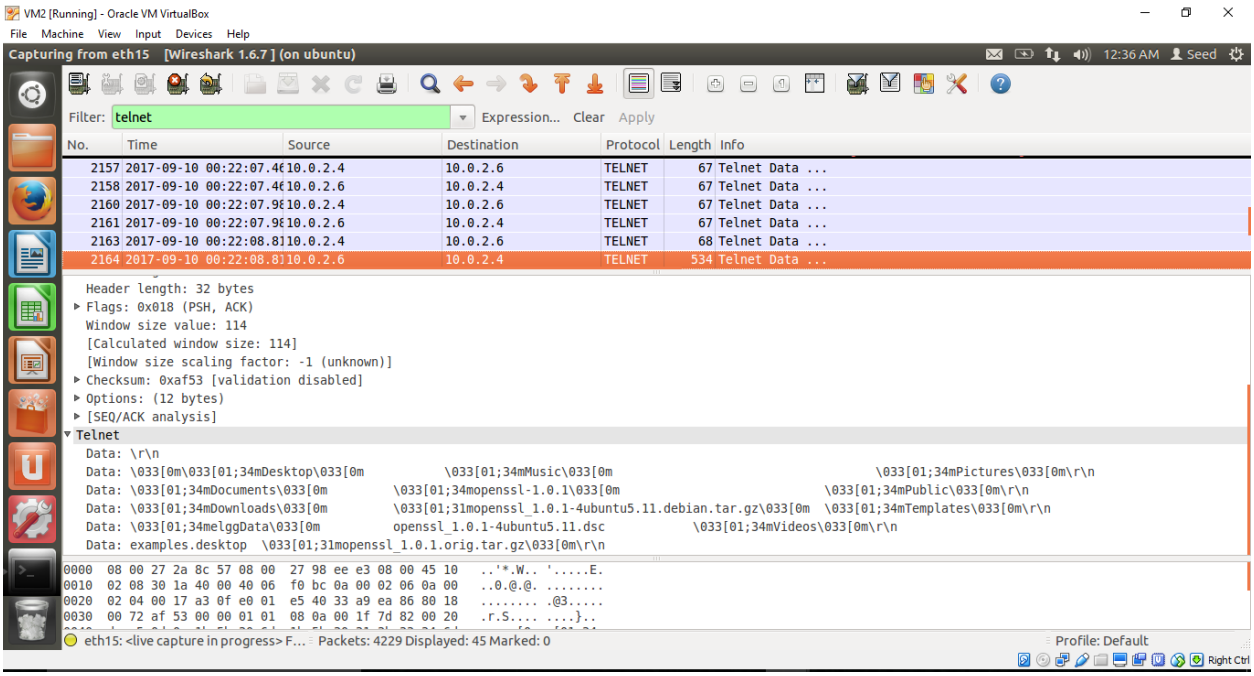


Figure shows server’s response to the attacker’s “ls” command.

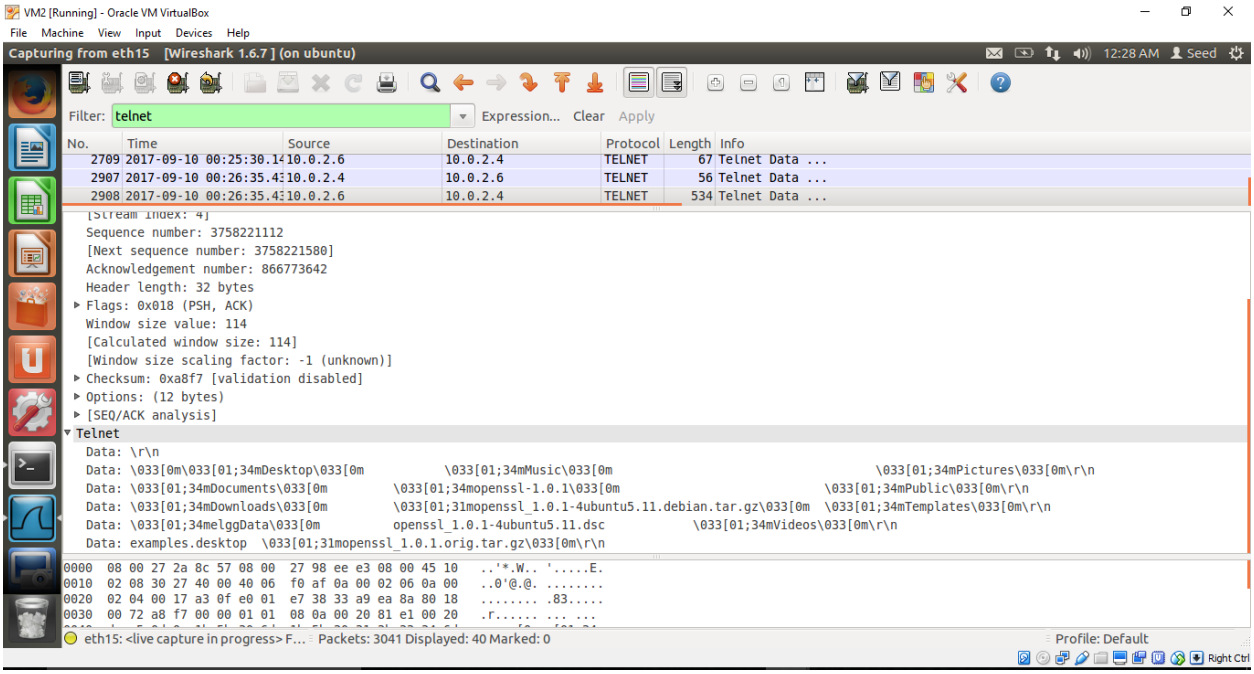


Figure shows payload information.

## Reversed shell:

- In this case there is a command where in a backdoor is created for the attacker on server's shell.
- To make this happen the attacker must use sniffing attack to hijack TCP connection between any client and server. After hijacking the TCP session the attacker must run the backdoor command which would create a reverse shell connection and redirect standard input and output to the TCP connection.
- Now that the attacker has access to the server's shell, attacker can send malicious commands.

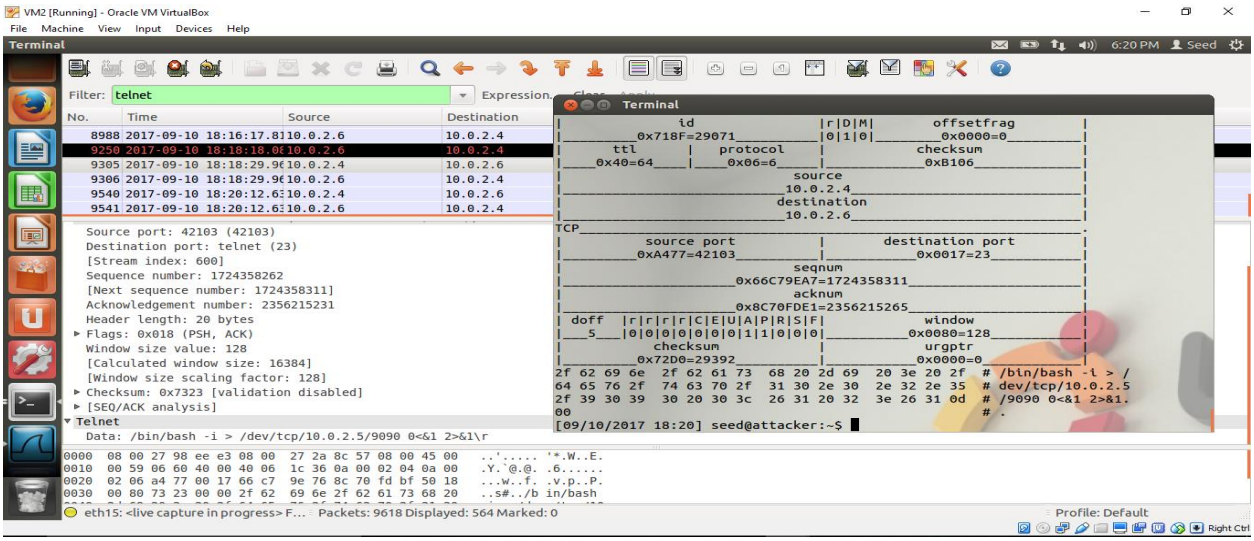


Figure shows attacker's hijacking command.

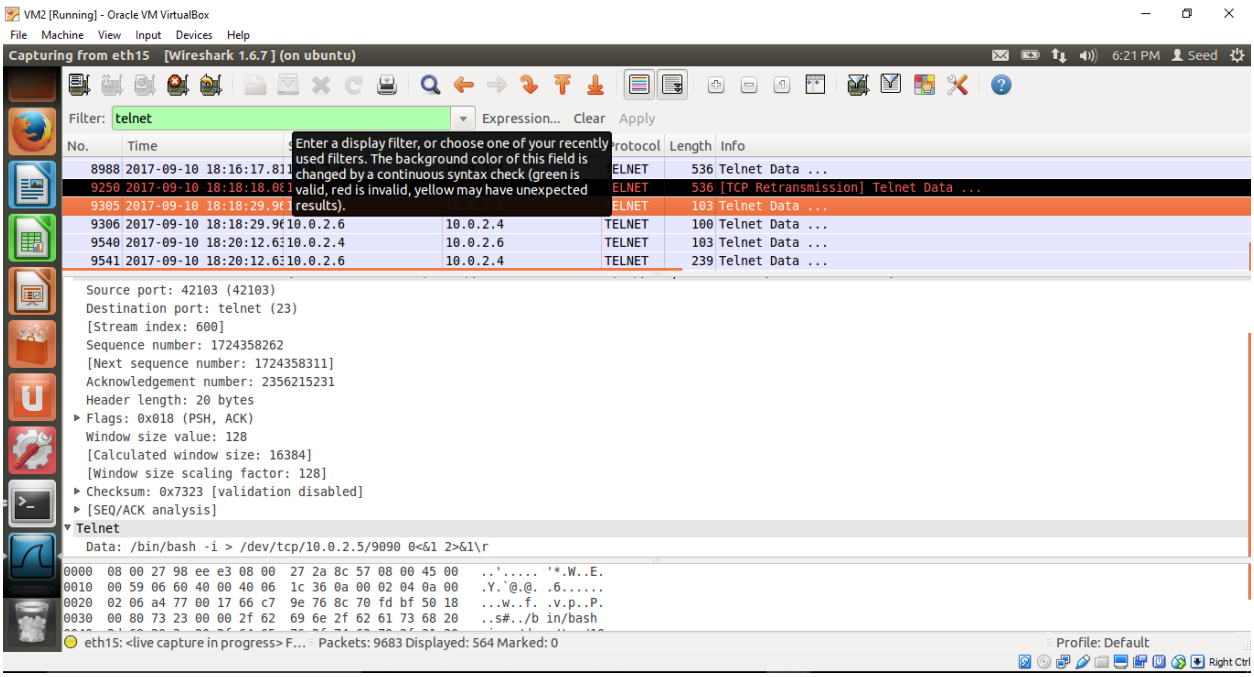


Figure shows wireshark analysis of attacker where-in the reverse shell command is sent to the server through TCP hijacking session.

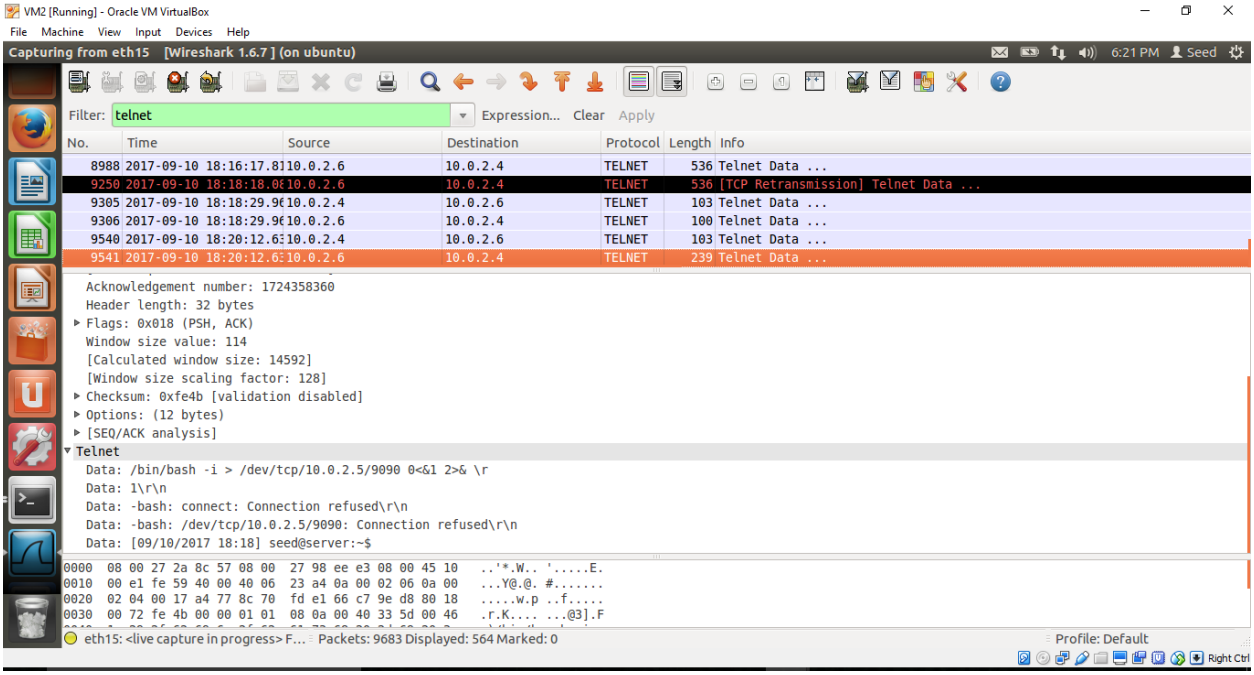


Figure shows output from server to attacker after TCP session hijacking.

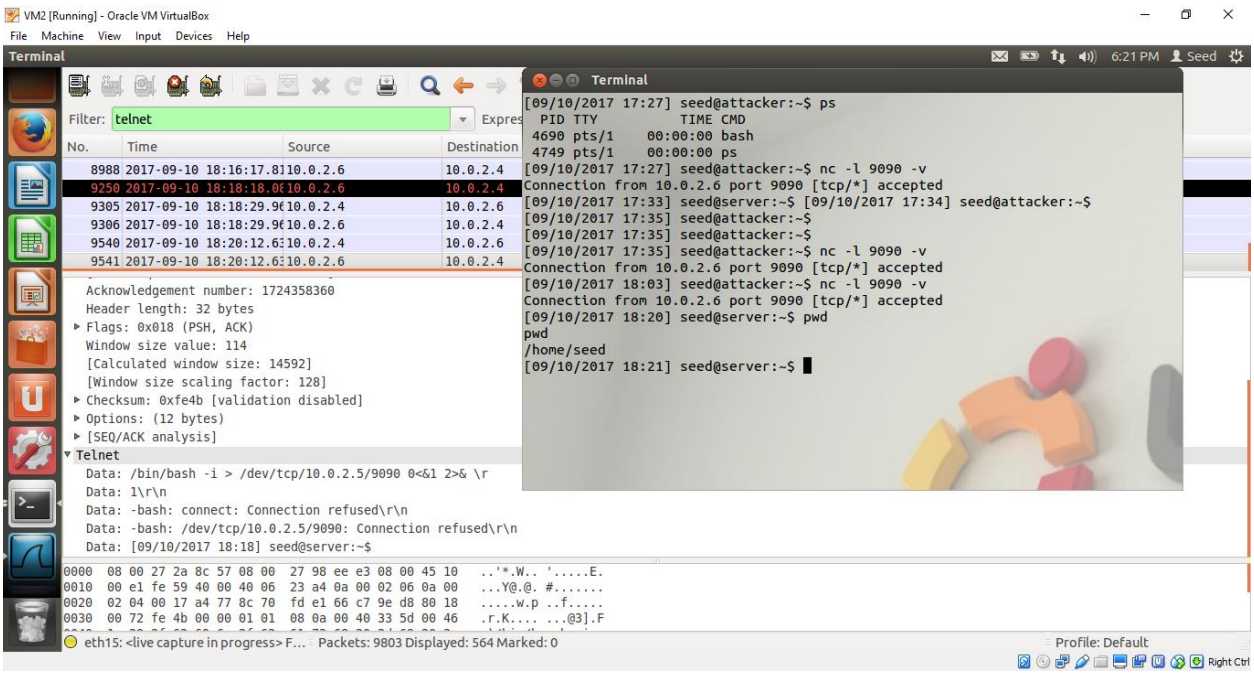


Figure shows successful reverse shell process.

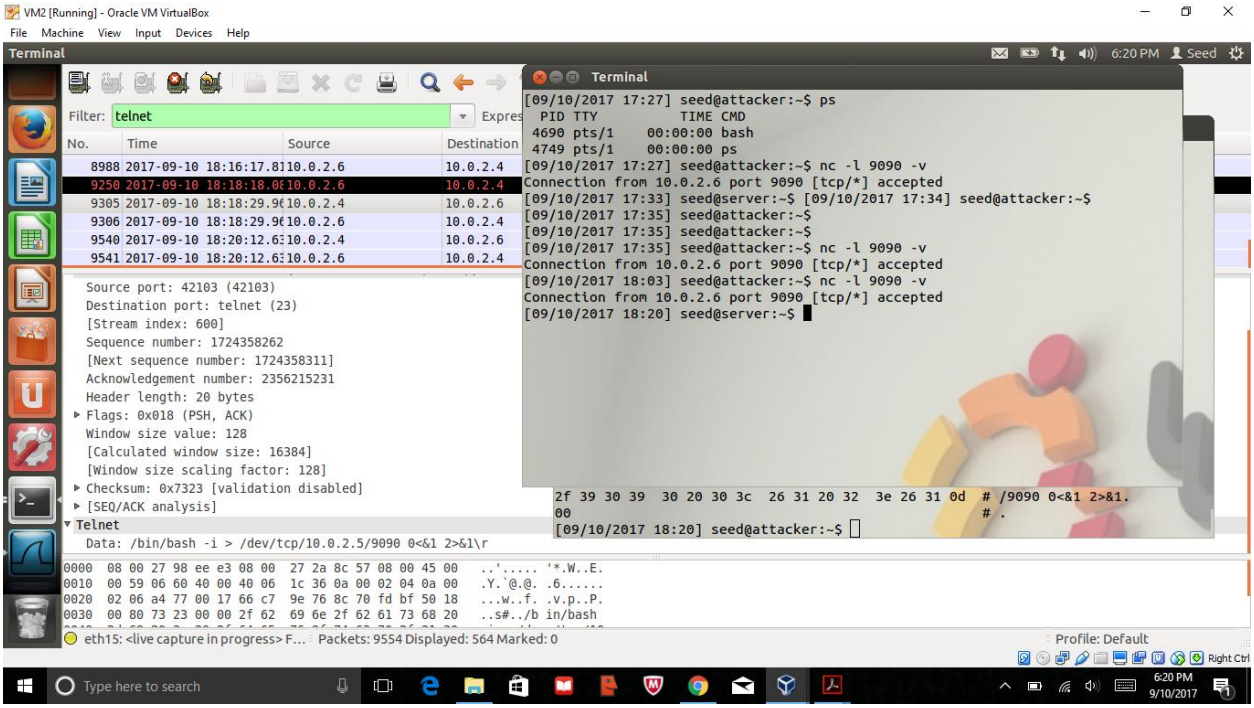


Figure shows attacker's side where connection from server is accepted.