CSE 406 Report

TCP reset attack on video streaming

Course: CSE 406
Lab Group: B2
Submitted By:

Meher Afroz
(1605114)

Department Of Computer Science and Engineering
Bangladesh University of Engineering and Technology
(BUET)

Contents

1.Introduction

1.1 What is TCP?

The Transmission Control Protocol (TCP) is a transport protocol that is used on top of IP to ensure reliable transmission of packets. TCP includes mechanisms to solve many of the problems that arise from packet-based messaging, such as lost packets, out of order packets, duplicate packets, and corrupted packets. Since TCP is the protocol used most commonly on top of IP, the Internet protocol stack is sometimes referred to as TCP/IP.
Although TCP is widely used in major internet applications, it introduces a few vulnerabilities too. The most common of these vulnerabilities are: Denial of Service (DOS), Connection Hijacking, TCP Reset attack etc.

1.2 TCP Connection Establishment

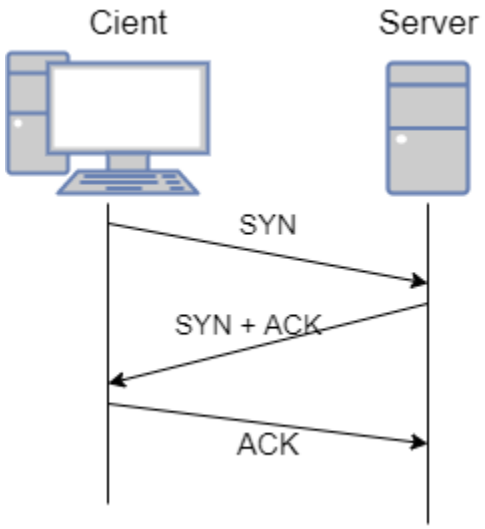

Figure 1: TCP 3 Way Handshake

TCP uses a three-way handshake to establish a reliable connection. The connection is full duplex, and both sides synchronize (SYN) and acknowledge (ACK) each other. The exchange of these four flags is performed in three steps: SYN, SYN-ACK, ACK, as shown in Figure 1. The client chooses an initial sequence number, set in the first SYN packet. The server also chooses its own initial sequence number, set in the SYN/ACK packet shown in Figure 1. Each side acknowledges each other's sequence number by incrementing it, this is the

acknowledgement number. The use of sequence and acknowledgement numbers allows both sides to detect missing or out-of-order segments.

1.3 TCP Connection Termination

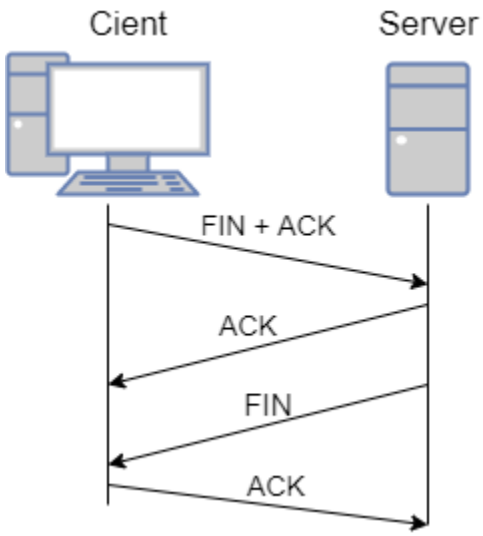

Figure 2: TCP Connection Termination

The connection termination phase uses a four-way handshake, with each side of the connection terminating independently. When an endpoint wishes to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical tear-down requires a pair of FIN and ACK segments from each TCP endpoint. After the side that sent the first FIN has responded with the final ACK, it waits for a timeout before finally closing the connection, during which time the local port is unavailable for new connections; this prevents confusion due to delayed packets being delivered during subsequent connections.

2. TCP Reset Attack on Video Streaming

TCP reset attack does not target the protocol's typical method of closing a connection which uses a 4-way handshake method. Rather it uses the protocol's method of immediately terminating an unwanted, unexpected or erroneous connection. Reset (RST) must be sent whenever a segment arrives which apparently is not intended for the current connection. It is an important property of TCP for ensuring robustness, but at the same time it has opened up a scope of exploitation. In TCP reset attack on video streaming, an attacker forges a spoofed RST packet that pretends to be the one coming from the original video streaming server. As a result, the victim immediately closes the TCP connection and goes to CLOSED state. In addition to that, upon receiving additional packets from the original server, the victim itself sends RST packet to the server terminating the connection at the remote end. In this way the attacker can successfully disrupt video streaming on its victim's machine.

3. Strategy

RESET is a flag in TCP packets to indicate that the connection is no longer working. So, if any of the two participants in a TCP connection send a packet contains such a RESET flag, the connection will be closed immediately. Using the concept of RESET flag reset on video streaming attack is done.

The strategy of our attack can be described in 3 steps. Those are:
1. First, we need to find out the IP address of either the victim machine or the video server.

2. Then we have to sniff packets in the network to discover the other IP address, the TCP port numbers and the correct sequence number. For this purpose, the tool will require additional feature of ARP Spoofing.

3. Finally, we forge a TCP packet with correct IP addresses, TCP Port numbers and sequence number and with RST bit set and send it to the victim machine.

4. Timing Diagrams

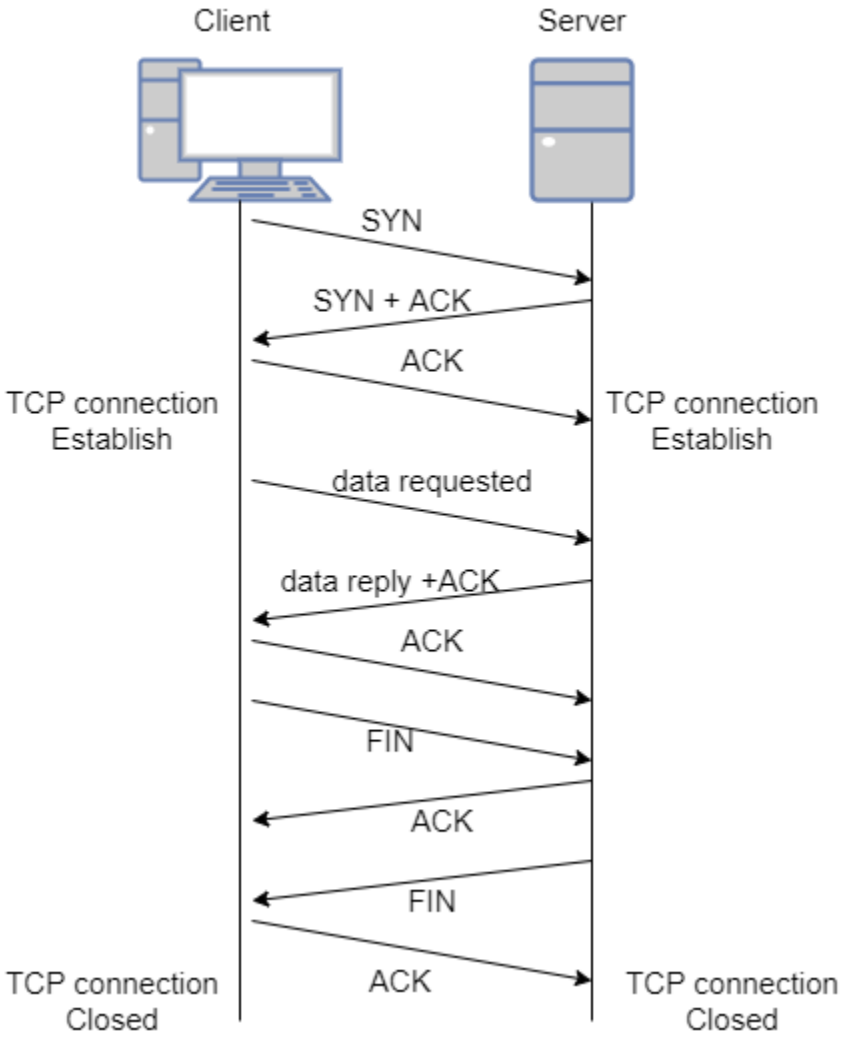

Figure 3: TCP Timing Diagram

In figure 3, the timing diagram of typical communication for video streaming between the streaming server and victim is shown. Here connection can be closed by either of the parties and in both cases a 4-way handshake will occur.
We will perform the TCP reset attack by sending forged RST packet from the attacker machine. The timing diagram of the attack is shown in figure 4. In this

case the victim will assume the video server has unexpectedly terminated the TCP connection, and upon receiving additional messages it will send RST back to the server. Finally, the server will also close the connection receiving the RST packet.

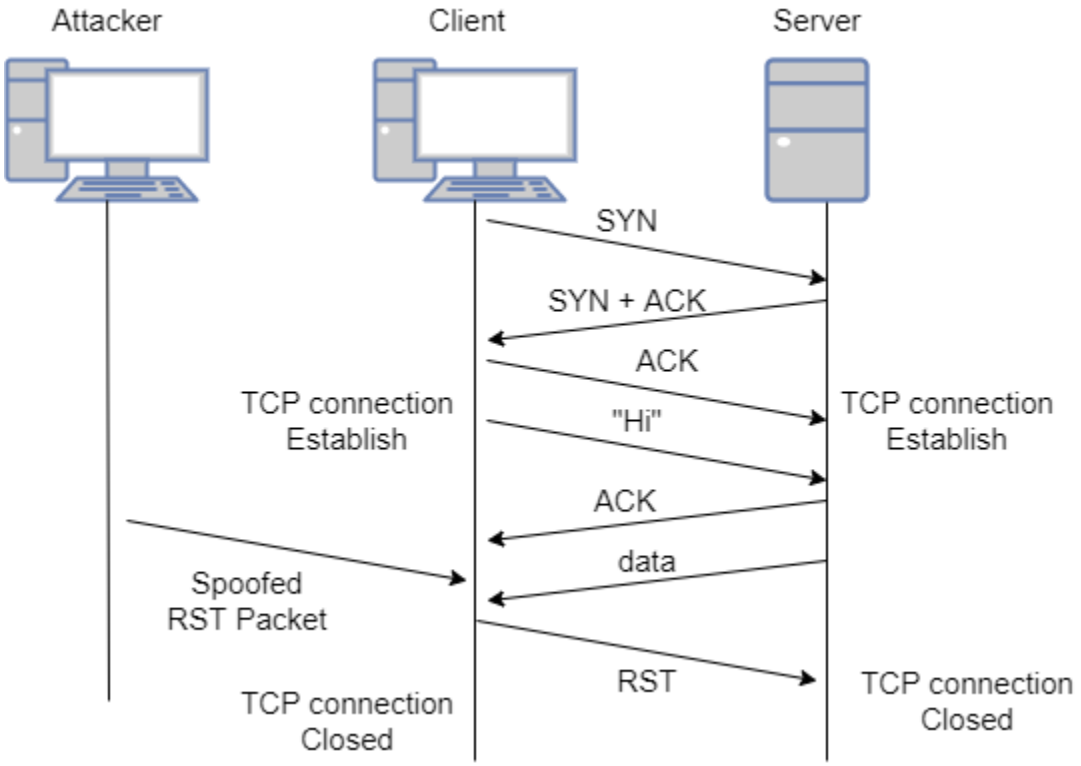

Figure 4: TCP Reset Attack Timing Diagram

5. Packet Details

In Figure 5, we show the specific fields in the TCP/IP header that need to be handled in order to perform TCP reset attack. Here source IP will be the video streaming server's IP address, Destination IP will be victim's IP address, and the port numbers will be set correspondingly. Sequence number must be correctly discovered through sniffing. Finally, RST bit must be set to 1. Also, payload to this header is not really important in this case as it is merely an RST packet.

Reset Packet

A RST packet is made by setting the RST bit in the flags section of the TCP header. In normal connection, this bit always 0.
This property of RST flag is very important and is used in emergency situations.

- When there is no time for the FIN protocol.
- To close half open connections of TCP SYN FLOOD attack.

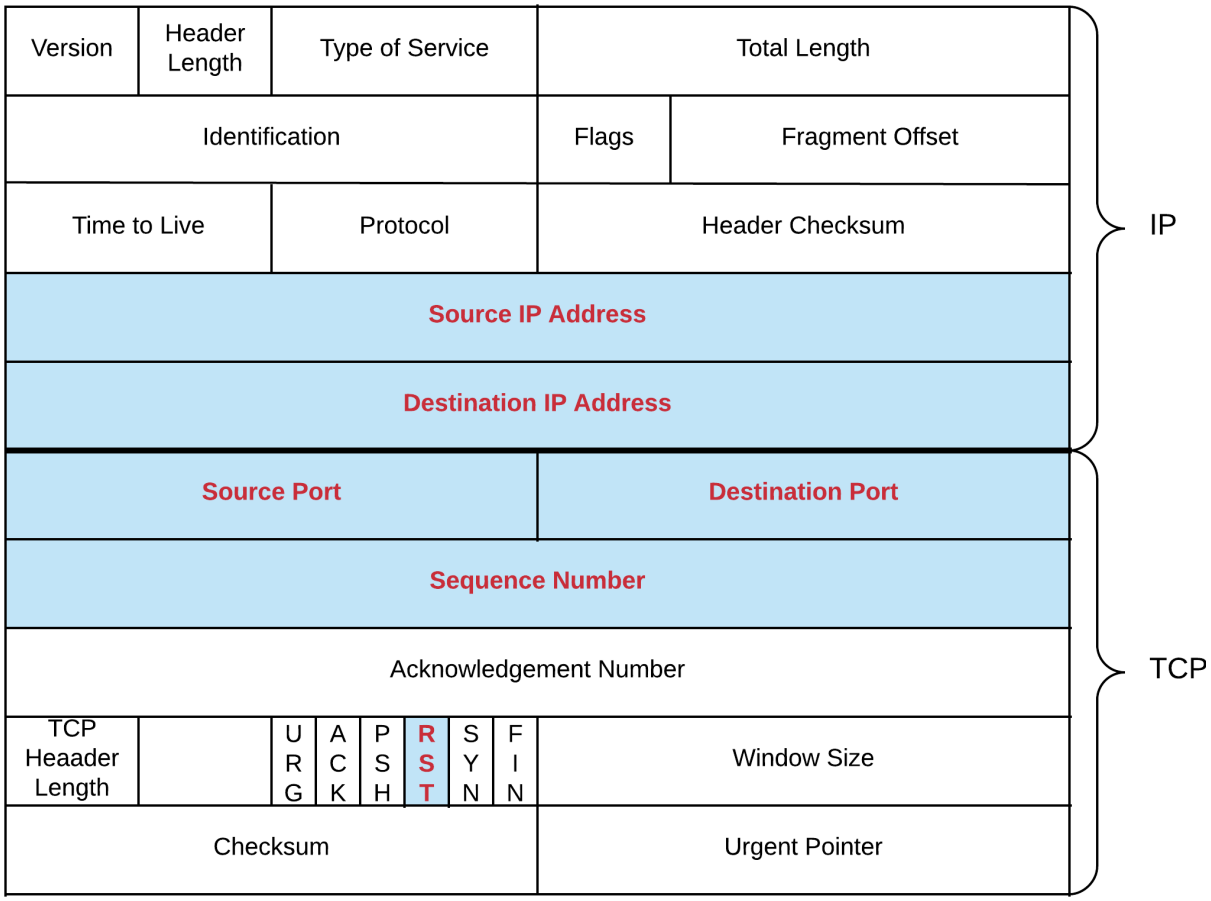

Figure 5: Attack Packet Header

6. Topology

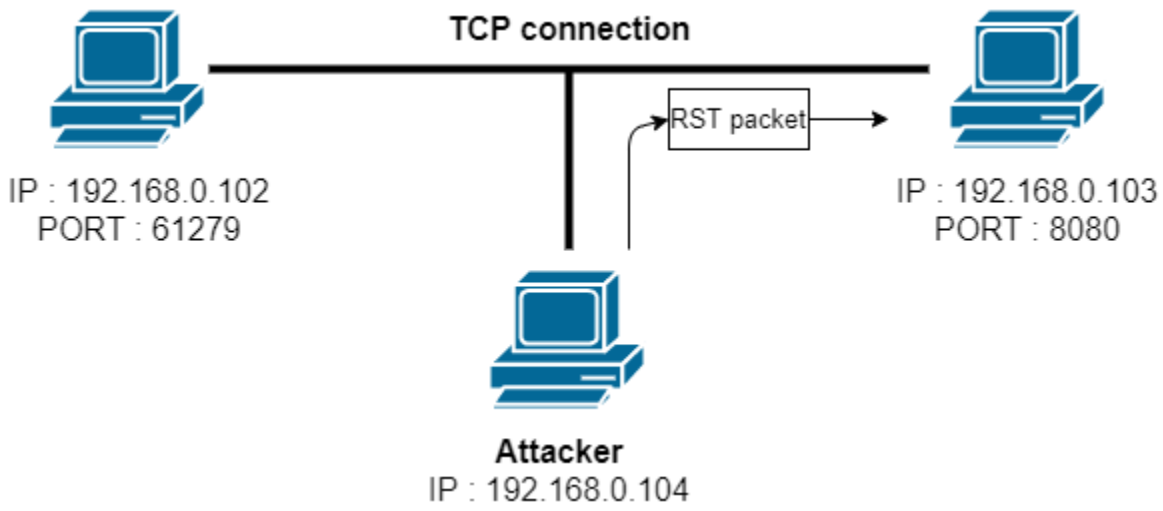

Figure 6: Topology of the Network

7. Implementation

7.1 Machines and tools

- We will use Wireshark for packer sniffing.
- Windows as a client.
- One Ubuntu as the video streaming server site.
- Another Ubuntu as attacker.
- 2 PC are connected in the same WIFI or LAN. One is the attacker and other is the victim machine.
- VLC Media Player for streaming.

7.2 Sniffing

On a LAN, after the stream has started, server and victim will be able to communicate with each other because the TCP connection has taken place. Attacker will not be able to listen that communication because the packets are not intended for him/her. In order to sniff the packets, attacker will need to carry out an ARP spoofing attack.

7.3 ARP Spoofing Man in the Middle Attack

ARP spoofing attack will update the mac table of both the server and the victim. Both the server and the victim will send their packets to the attacker. So, the attacker needs to turn on IP forwarding. On a Linux based system, we can turn on IP forwarding by the following command:

```
$ sudo sysctl net.ipv4.ip_forward=1
```

IP forwarding ensures that when the server and the victim send packet to the attacker, the attacker will redirect the packets to their originally intended destination. ARP Spoofing helps to capture the packets between the two parties, i.e., to sniff them in order to extract meaningful information.

In order to carry out the ARP Spoofing attack, we run a C++ program “Spoof.cpp” on attacker's machine which will continuously send out ARP packets to both server and victim and update Arp table.
We can check Arp table using the following command:

```
$ arp -a
```

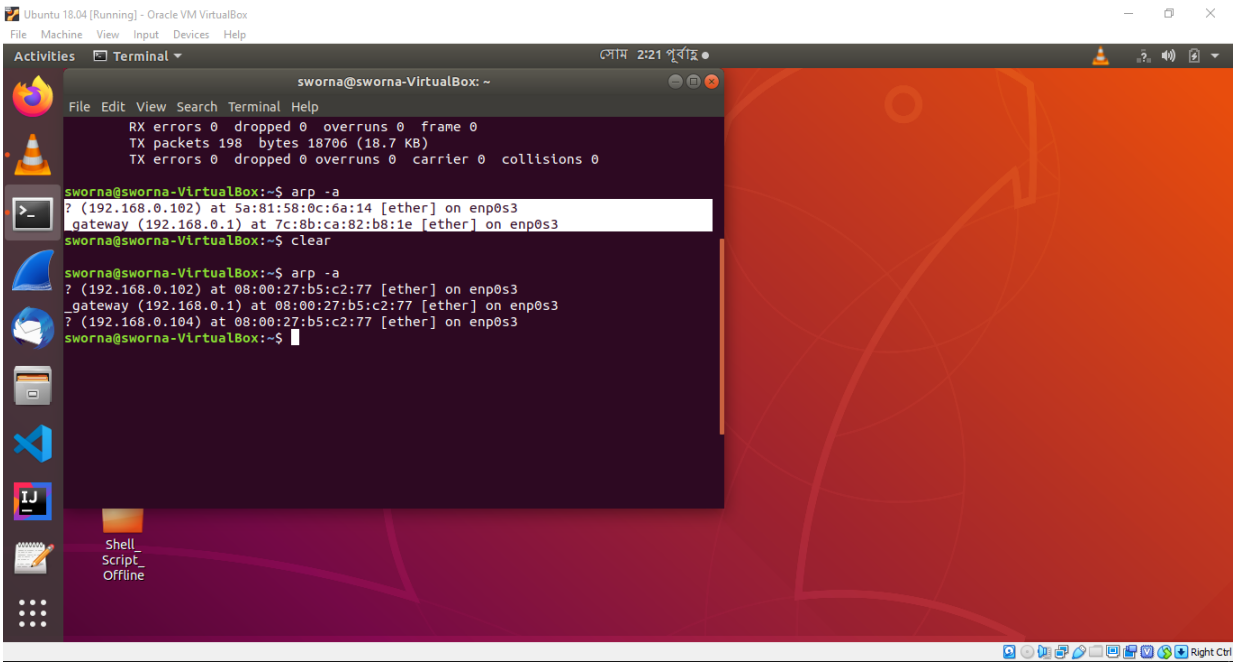

Figure 7: Mac table before Arp spoofing.

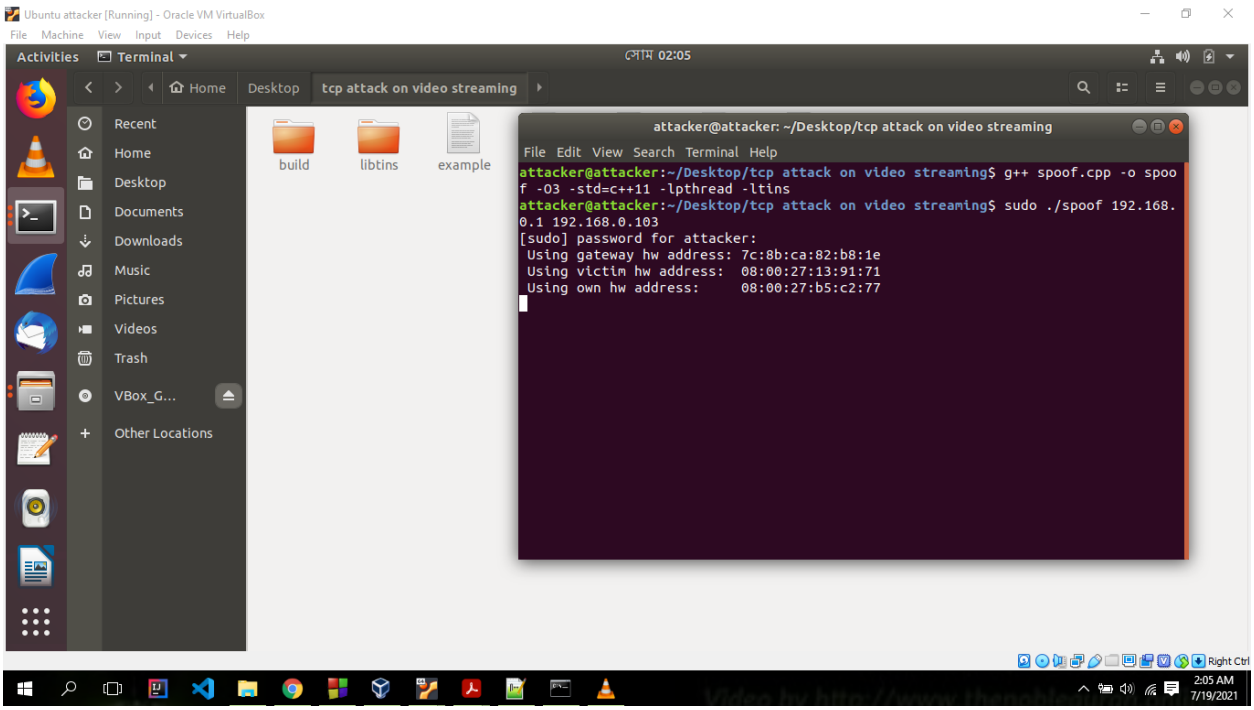

Figure 8: Arp spoofing attack.

In our case, the attacker (192.168.0.104) claims to be both the server (192.168.0.102) and the victim (192.168.0.103). So, the server will think the attacker as the victim and send the packets intended for the victim to the

attacker. The victim will do the same. So, all the packets will pass through the network interface of the attacker.

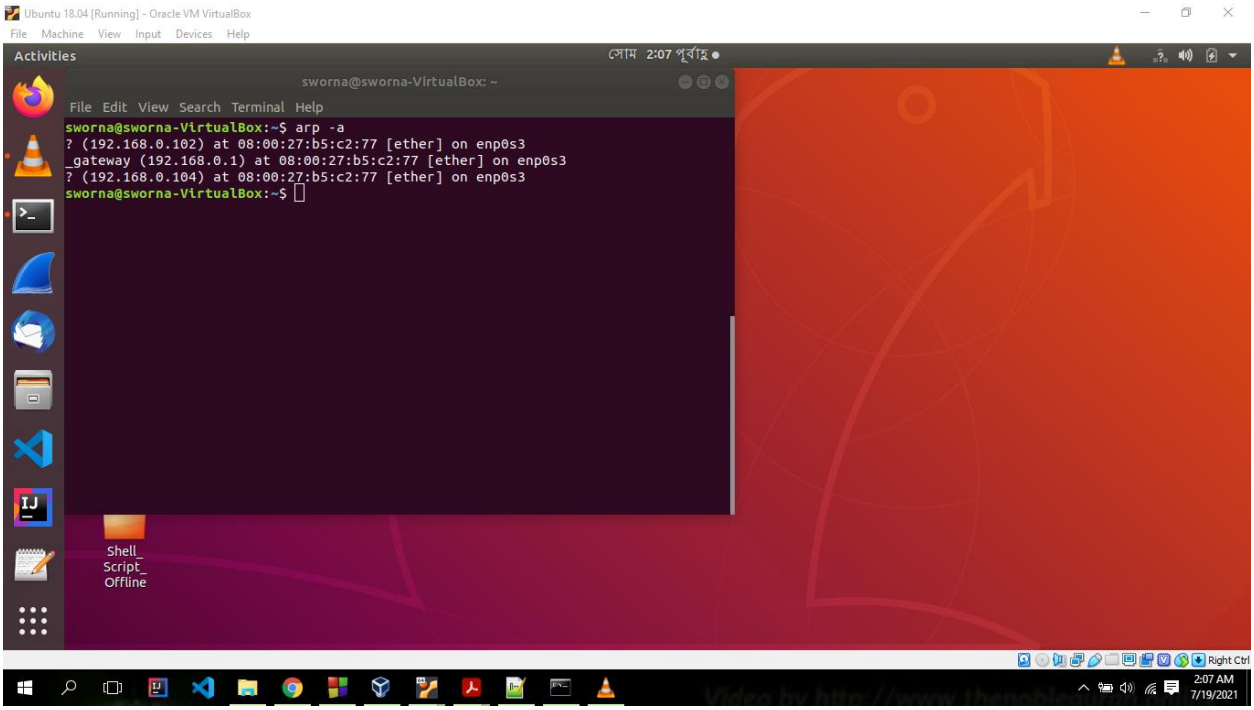

Figure 9: Incorrect Mac in Mac Table after Arp spoofing attack.

7.4 Building RST Packet

We will get source IP, destination IP, source Port, destination Port from the sniffed packet. And make a spoofed RST packet. We may use C or C++ for this purpose.

7.5 RST Packet Spoofing

Server (192.168.0.103) start streaming at port 8080 using VLC media player. Client (192.168.0.102) plays the video using VLC media player. At this moment, server have incorrect mac table. So, all the packets between server and client pass through attacker.

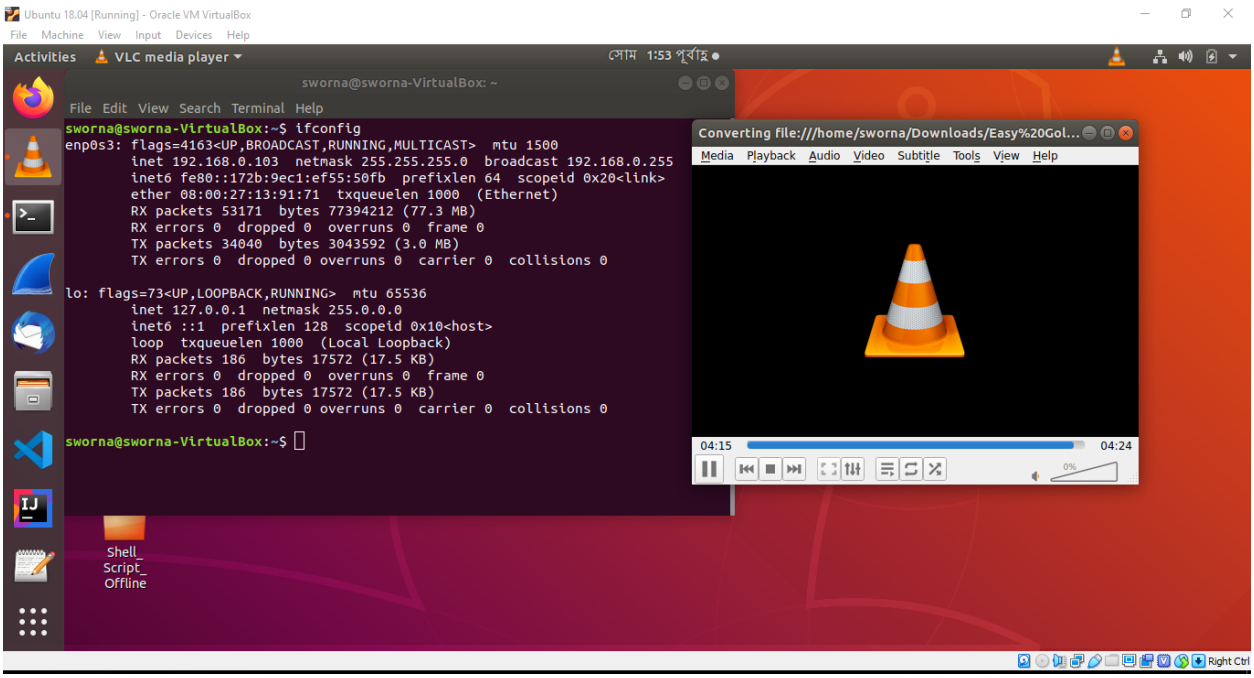

Figure 10: Server streaming video using VLC media player.

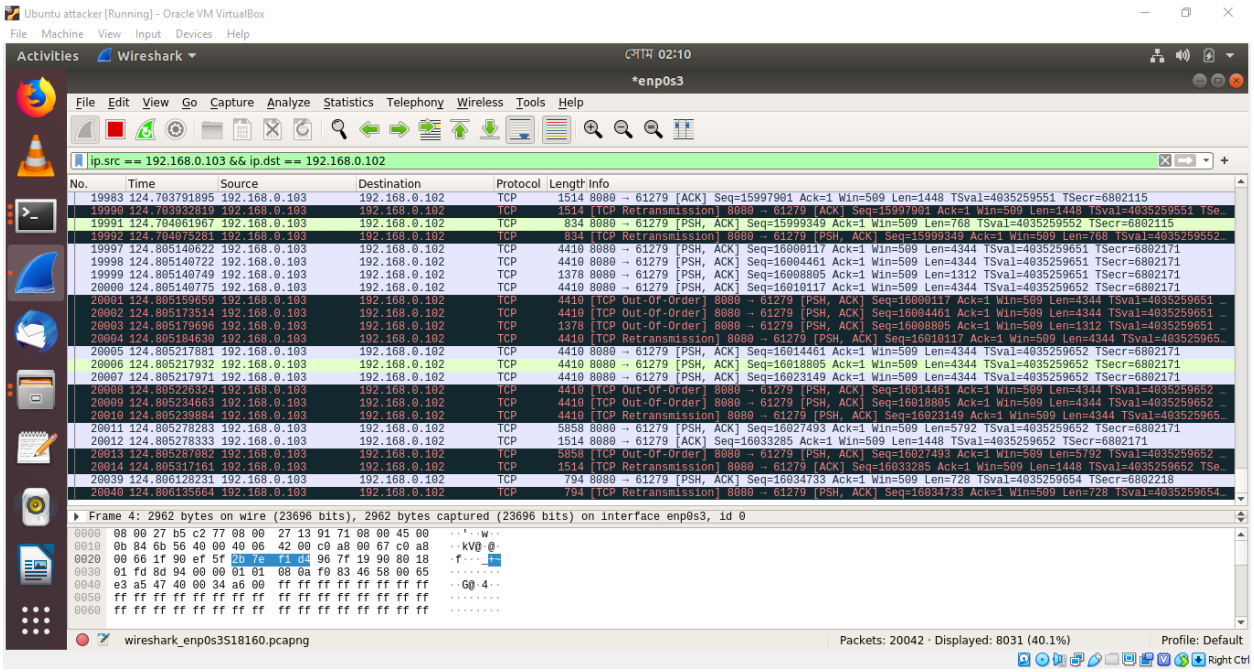

Figure 11: Packets are passing through attacker machine.

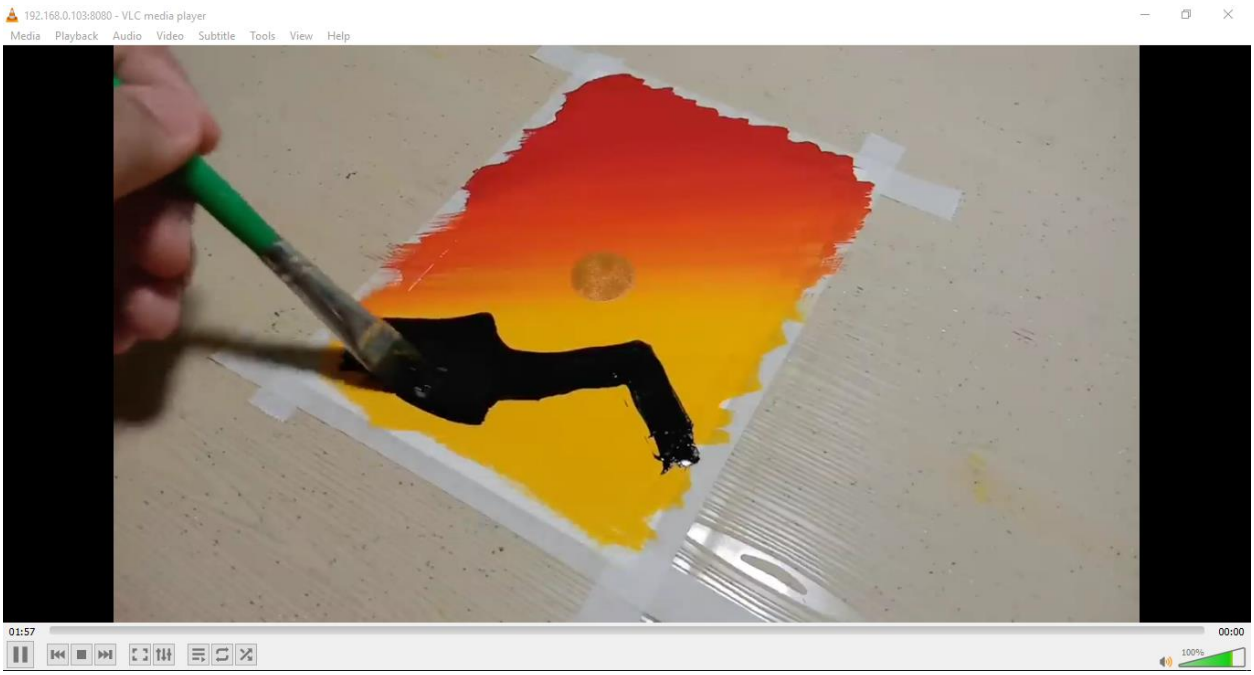

Figure 12: Video playing in client side.

Now the attacker starts “sniff_spoof.c” program. The program intercepts any TCP packet on attacker's network interface. It does so successfully in the context of the server and the victim because our ARP spoofing Man in the Middle attack is in place. Once it catches a packet, a callback function got packet is invoked. The callback function extract packet information and construct RST packet and send the spoofed packet to the victim.

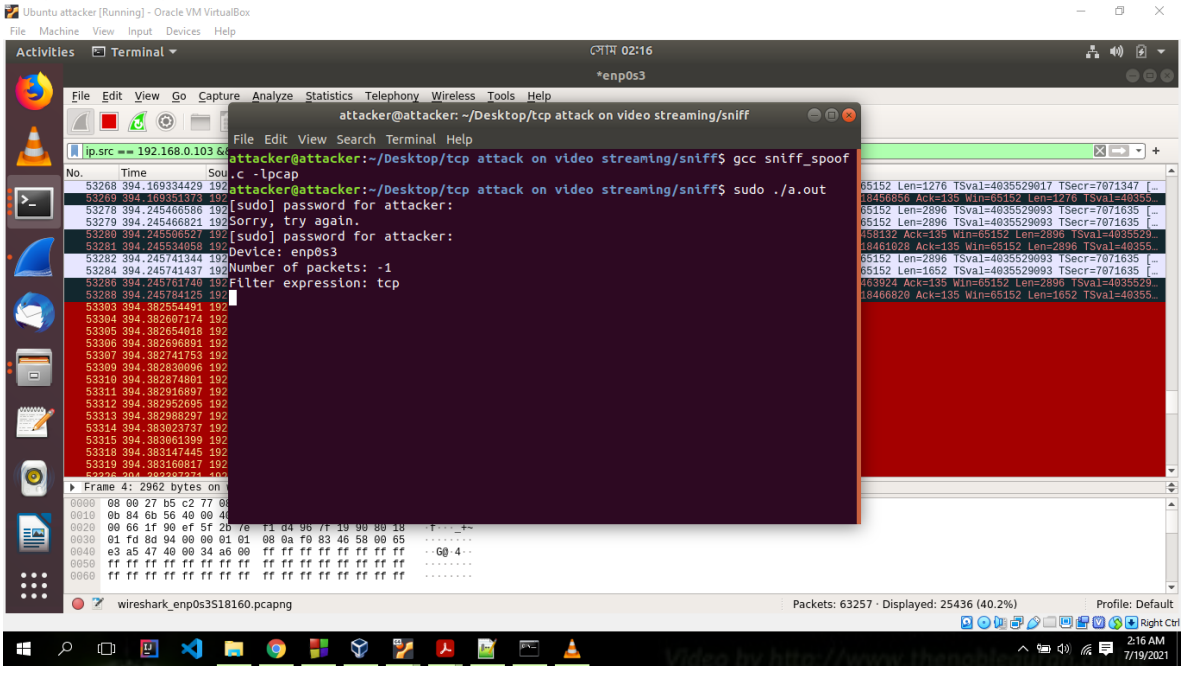

Figure 13: Attacker running his program to send RST packet.

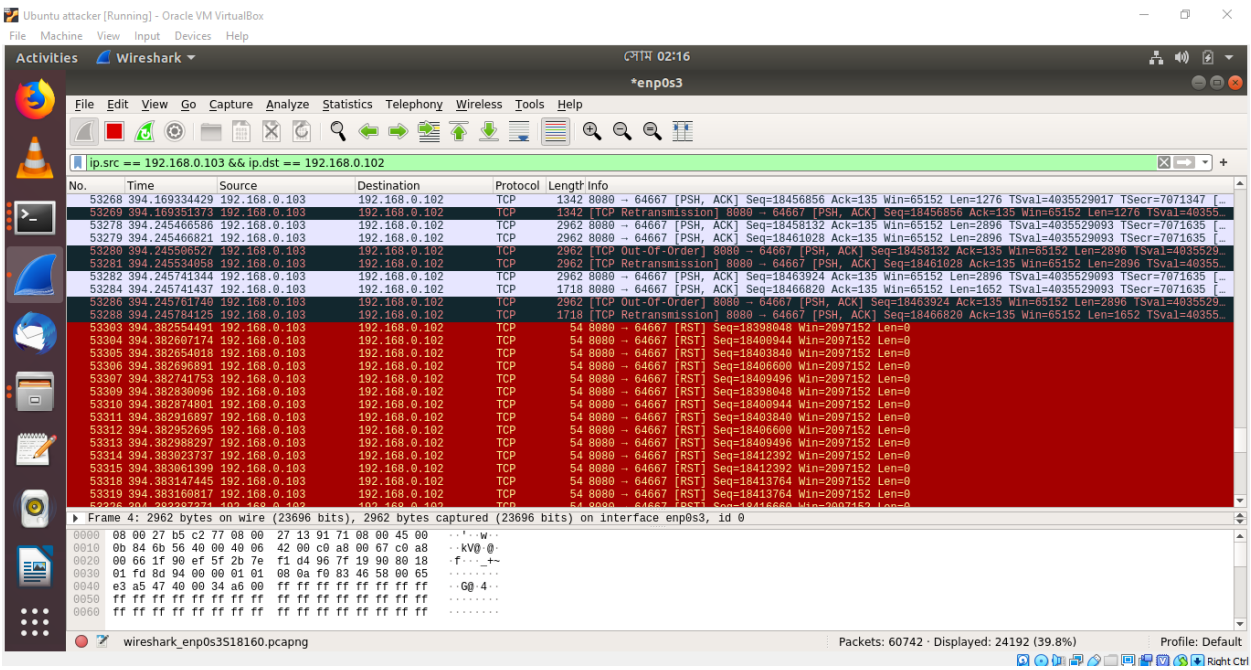

Figure 14: RST packet sent to victim captured in Wireshark.

The attacker successfully sends RST packet and video stopped running in the victim side.

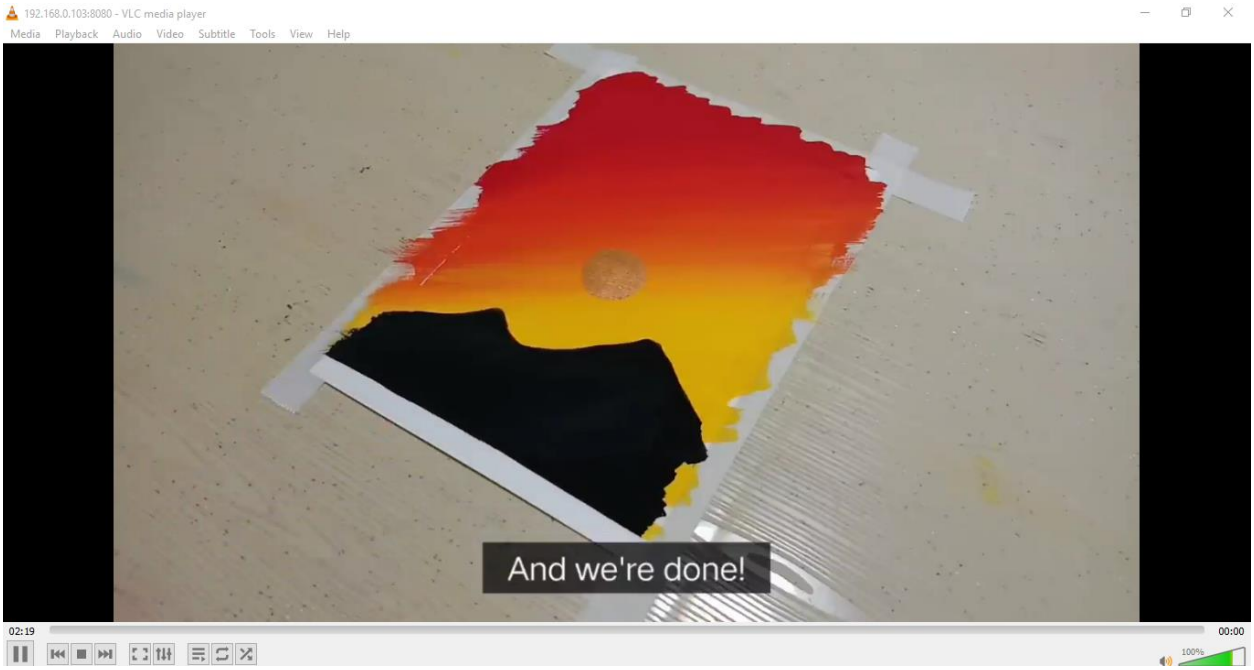

Figure 15: Video Stopped Running.

8. Justification

We have included both Man-in-the-Middle and TCP reset mechanism in our implementation. It can be inferred from above discussion that with the help of MITM mechanism, TCP reset can successfully find out correct IP, port numbers and the sequence number, and consequently our tool can exactly mimic a RST packet from the original server. When the victim machine receives the RST packet, it does not have any idea about the packet's actual origin. So, the victim machine has no other option but to terminate its TCP connection.

9. Attack Prevention

In a blind TCP reset attack using the RST bit, we attempted to guess the RST segments to prematurely terminate an active TCP session. To prevent a user from using the RST bit to reset a TCP connection, the RST bit is subject to the following rules when receiving TCP segments:

- If the RST bit is set and the sequence number is outside the expected window, the device silently drops the segment.
- If the RST bit is exactly the next expected sequence number, the device resets the connection.
- If the RST bit is set and the sequence number does not exactly match the next expected sequence value, but is within the acceptable window, the device sends an acknowledgement (ACK).

In this way, we are planning to prevent our TCP reset attack on video streaming.