CSE 406 Report

TCP Reset Attack
On Video Streaming

Course: CSE 406
Lab Group: B1

Submitted By:
Rishov Paul
(1605084)

Submitted To:
Dr. Md. Shohrab Hossain
Abu Wasif
Md. Toufikuzzaman

TCP Reset Attack on Video Streaming

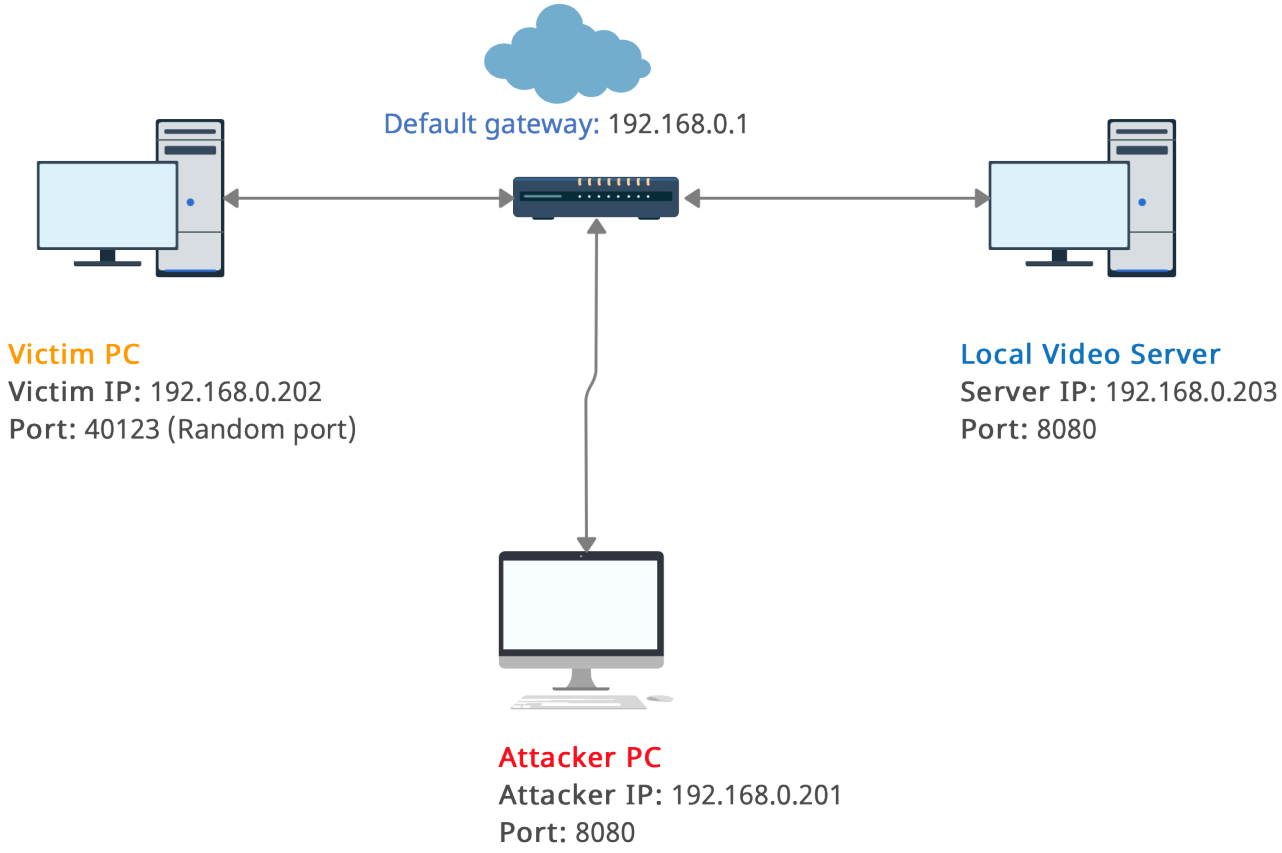
TCP reset attack, also known as "forged TCP resets", "spoofed TCP reset packets" or "TCP reset attacks", is a way to tamper and terminate the Internet connection by sending a forged TCP reset packet. This tampering technique can be used by a firewall in goodwill, or abused by a malicious attacker to interrupt Internet connections.

In a stream of packets of a TCP connection, each packet contains a TCP header. Each of these headers contains a bit known as the "reset" (**RST**) flag.In most packets this bit is set to 0 and has no effect; however, if this bit is set to 1, it indicates to the receiving computer that the computer should immediately stop using the TCP connection; it should not send any more packets using the connection's identifying numbers, called ports, and discard any further packets it receives with headers indicating they belong to that connection. A TCP reset basically kills a TCP connection instantly.

Normally, the TCP reset bit is sent by a computer that was one of the connection endpoints. It is possible for a 3rd computer to monitor the TCP packets on the connection and then send a "forged" packet containing a TCP reset to one or both endpoints. The headers in the forged packet must indicate, falsely, that it came from an endpoint, not the forger. This information includes the endpoint IP addresses and port numbers. Every field in the IP and TCP headers must be set to a convincing forged value for the fake reset to trick the endpoint into closing the TCP connection. Properly formatted forged TCP resets can be a very effective way to disrupt any TCP connection that the forger can monitor.

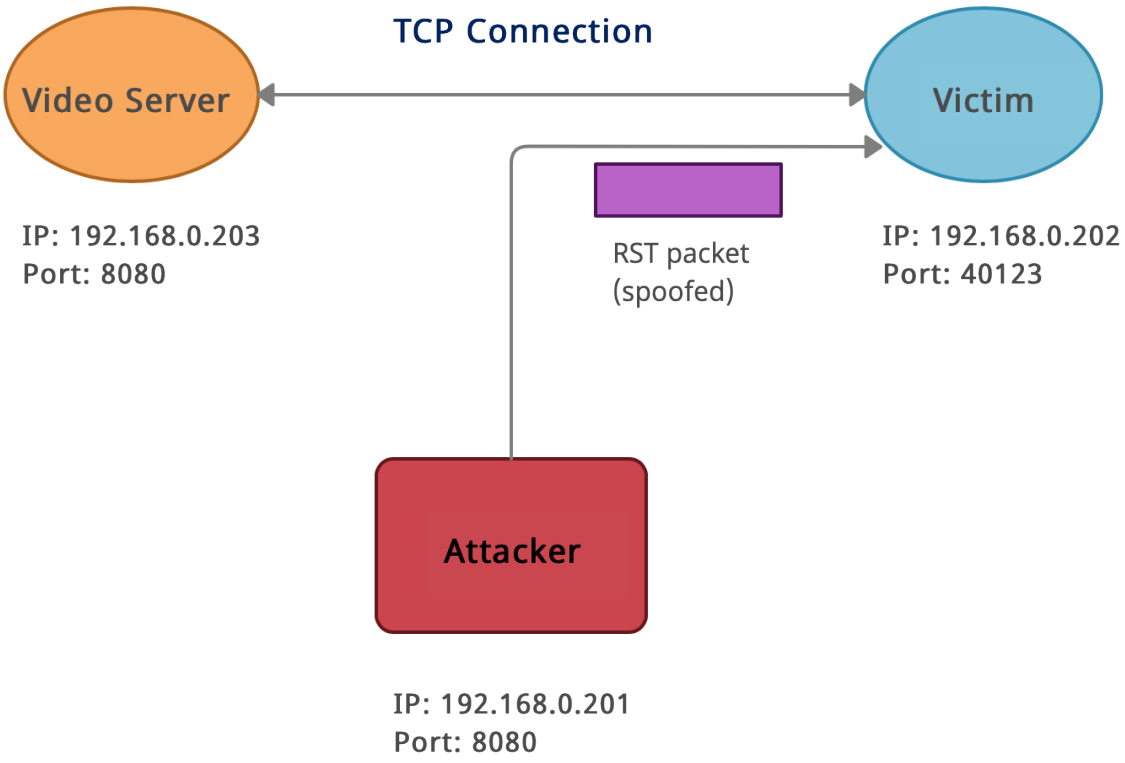
Topology Diagram

The network topology is pretty straightforward. The victim, the attacker and the video streaming server - all are on the same LAN. Server will stream the video, and the victim will connect to the server to watch that video. Attacker will perform a man in the middle attack between this connection to interrupt the video streaming on the victim's PC.



Attacker, Server and Victim machine are on the same LAN

Attack Diagram



Attack Diagram of TCP Reset Attack On Video Streaming

Attack Strategies

There are mainly 2 types of variation of the **TCP RST** attack:

1. **RST attack on Telnet/SSH**
2. **RST attack on video streaming applications**

In this report, we will be looking at **TCP RST** attack on video streaming applications in detail.

In the case of Telnet/SSH, the victim establishes a connection with another host/server and browses their directory. As it requires the victim to type commands in the command line interface, the sequence number does not vary much. We can simply look up the sequence number using a tool like **Wireshark**.

But in the case of video streaming applications, the scenario is a bit complicated. There is a unique challenge for resetting video-streaming connections. The challenge is the sequence number. In our attack against telnet connection, we sniff the packet, get the sequence number, and then type it in our command. While doing this manually, we will not type anything in the telnet terminal, or that will increase the sequence number, making the one that we get from **Wireshark** invalid. In video-streaming connections, we have no way to stop the packets between the client and the server, so the sequence number increases very fast, making manual efforts very difficult. We have to automate our attack, so instead of using the manual sniff-and-type approach, we want to do it automatically, i.e., we would like to run a program that sniffs the video-streaming packets, gets the sequence numbers and the other essential parameters, and then automatically sends out spoofed **TCP RST** packets. This is called sniff-and-spoof.

Packet Details

The attacker will first sniff the packets transferred between the video server and the victim machine in order to gain information about the IP addresses, Port numbers and Sequence numbers. There has to be some sort of automation involved, as in this case the sequence numbers can rise very quickly. After acquiring the information, the attacker will carefully construct a TCP packet with the correct source port, destination port, sequence number and the set **RST** bit field. The attacker will encapsulate it with an IP datagram after putting the correct source IP and destination IP. The relevant fields are visible from the figures below.

Modifications in the header

TCP header

- Source Port
- Destination Port
- Sequence Number
- RST bit

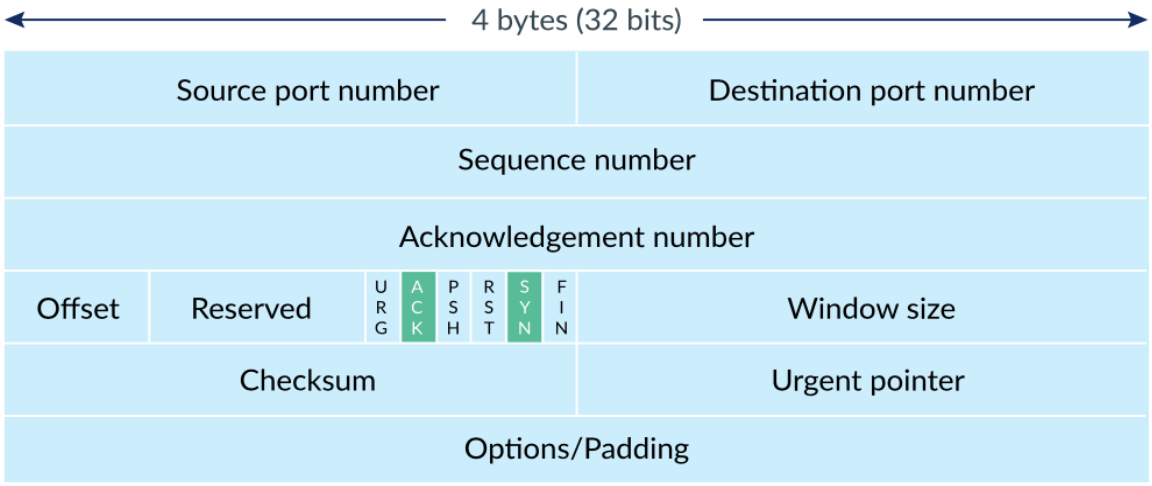


Fig: TCP Header

IP header

- Source IP address
- Destination IP address

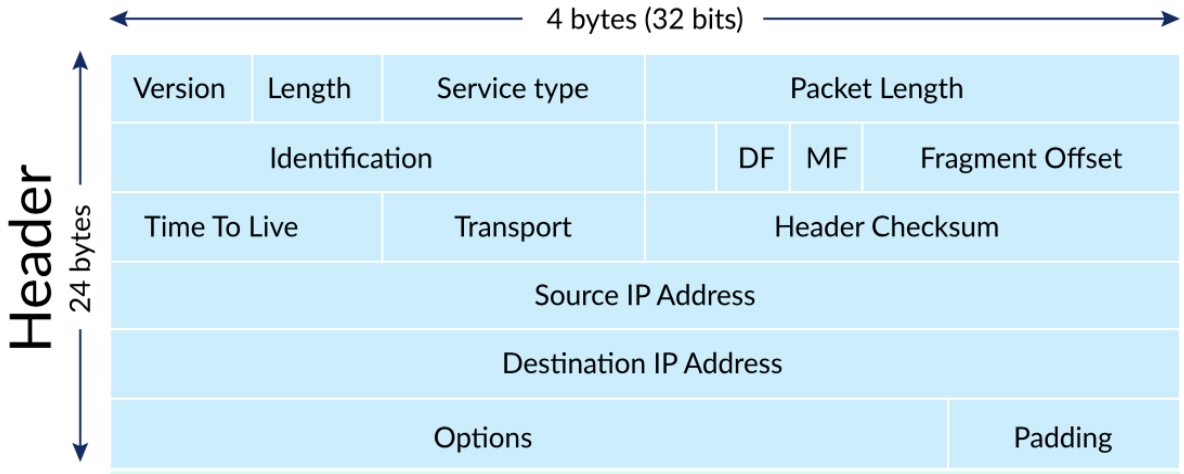


Fig: IP header

Justification

- Sending spoofed TCP packets is comparatively easy, since neither TCP nor IP comes with any built-in way to verify a sender's identity. There is an extension to IP which does provide authentication, called IPSec. However, it is not widely used. Internet service providers are supposed to refuse to transit IP packets that claim to have come from a clearly-spoofed IP address, but such verification is said to be patchy. All a receiver can do is to take the source IP address and port inside a packet at face value, and where possible use higher-level protocols, such as TLS, to verify the sender's identity. However, since TCP reset packets are part of the TCP protocol itself, they cannot be validated using these higher-level protocols.

- The TCP RST flag doesn't wait for any ACK. So it should work.