## 



CSE-406

Optimistic TCP Ack attack

30.07.2019

**─**

Md. Asif Tanvir

1505088

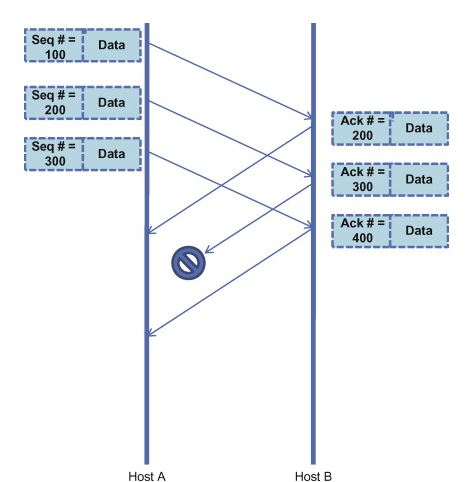
Group - 1

Section - B

# Introduction

Optimistic TCP-Ack acknowledges a packet before even it is received. In effect, the connection’s round trip time is reduced and the total throughput increased. An attack can be initiated on this technique where attacker keeps acknowledging the sender without getting the information to make the system flooded with packets.

In general TCP connection ack works as following:

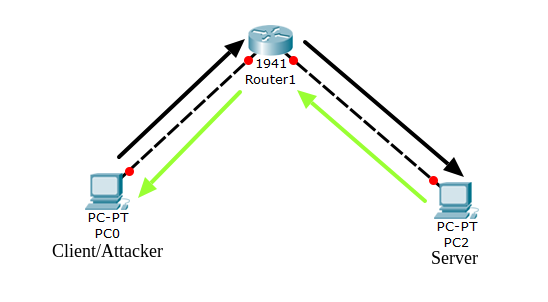


Here Host A gets Ack for Seq#200 , so it will guess as Seq#200 acknowledged.

For **TCP ack attack**, upon receiving the first data packet, the receiver sends a stream of ACKs to the sender for data which it has not yet received. The sender, confused by these ACKs, will put more data into the network before the previous data has left the network.This will force the network to increase its bandwidth. Eventually after a certain time the network will run out of bandwidth and it will create a **Denial of Service.** It means that other clients will not be able to get access to that server.

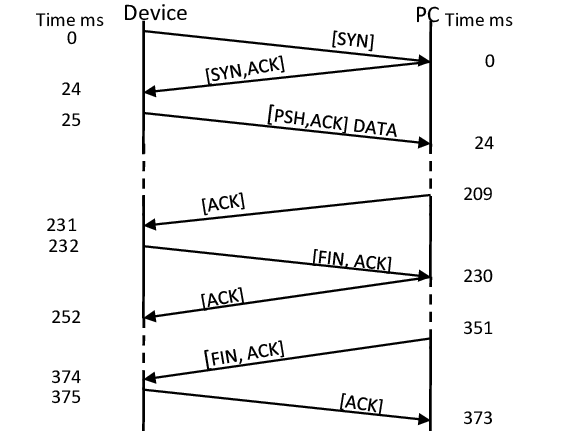
# Topology

Packet will go from server to client. After getting the first packet the main attacking code will start from the client side.

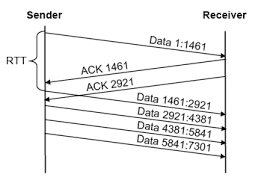


# Timing Diagram

Original diagram of TCP works like following

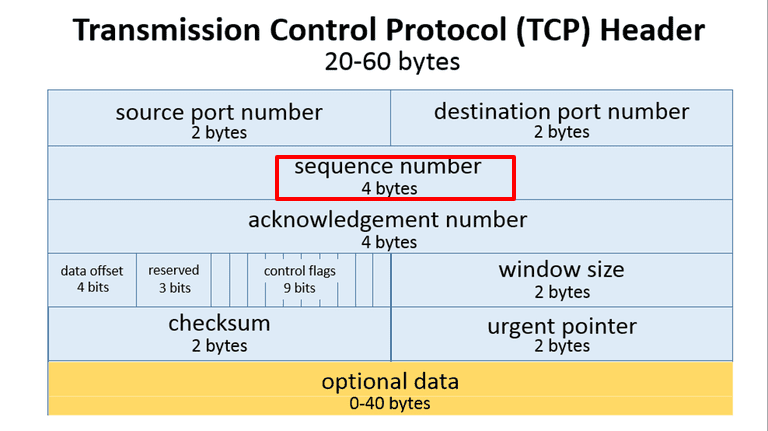


After the attack is implemented then the diagram will look as following



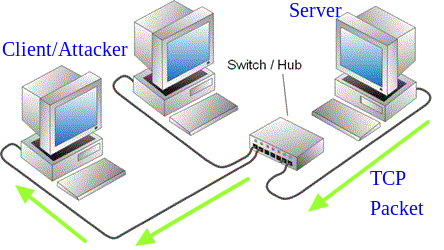
# Implementation

1. First we need to know about the **TCP Header** details.

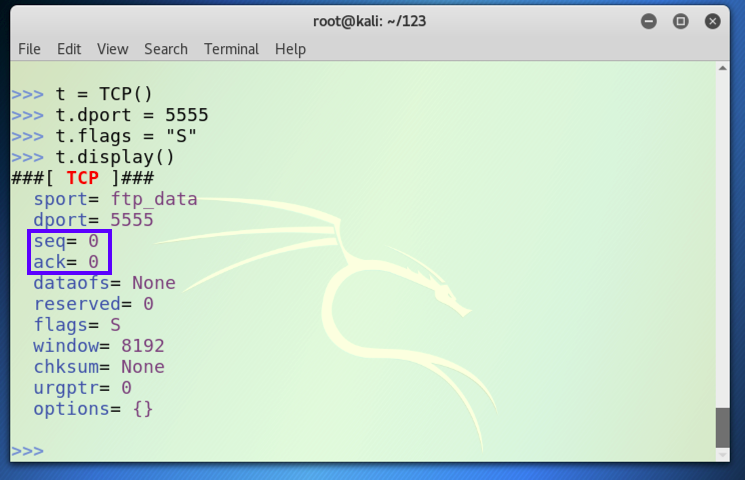


TCP has 4 bytes of **Sequence Number,** 4 bytes of **ACK Number** and 2 bytes of **Window Size.** For implementing this attack we will need to extract these two parameters from the incoming packet and change them accordingly.

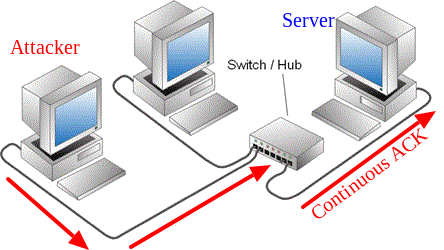
2. We need to build the client side of the program. Using LAN connection we can connect two PCs. One of them will work as the server and the other as the client. **The client will be the attacker**. The server will send normal TCP packets to the client. The server will be built using regular socket programming in **PYTHON.**



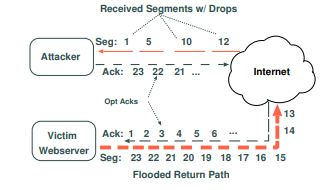
3. Now the **attacker** need to get the **sequence number**  and the **data size**  from the header of the packet. We can do this using a packet manipulation tool called **SCAPY**. It is a **PYTHON** tool that creates all the necessary headers for TCP packets. Using this header we can fetch all the parameters from the header that we need.



4.Now we need to write the socket programming code for the attacker. We need to send continuous ACK to the sender even before receiving the packets. As we have the SEQ number from the last received packet, we can calculate the ACK number that we will send to the server. The server upon receiving an ACK of a high number will assume that all the packet before that ACK number has been received and increase the window size.



The attacker sends this ACKs continuously and the server eventually finishes up all is bandwidth by increasing its window size.



# Tools & Language

1. Router for LAN

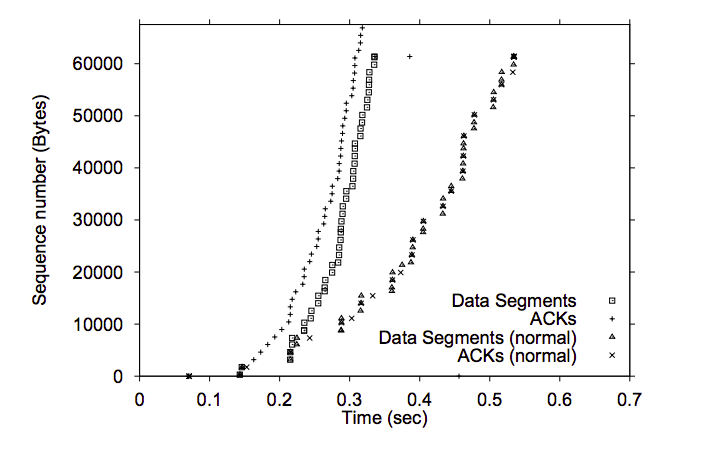
2. PYTHON for socket programming

3. SCAPY for packet sniffing

# 

# Justification

This attack will work because of the fault of the implementation of tcp server. A TCP sender varies its transmission rate based on receiving ACKs of the packets it sends. An *optimistic* ACK is an ACK sent by a client for a data segment that it has not yet received. Here the client sends continuous ACKs starting from the SEQ number of the packet it received at the very first handshake with the server. These ACKs correspond to a legitimate packet that the sender has sent in the network called **in-Flight packets**. As a result, the sender believes that the transfer is progressing better than it actually is and may increase the rate at which it sends packets.



This diagram shows how by only sending more and more ACKs we can get more data segments.