**CSE 406 : COMPUTER SECURITY**

**DESIGN REPORT ON TCP SYN FLOOD ATTACK**

**MRITUNJOY ROY**

**GROUP NO.:3**

**STUDENT ID:1505071**

TCP SYN FLOOD ATTACK

**Definition:**

1. Client --SYN Packet--> Server  
2. Server --SYN/ACK Packet --> Client  
3. Client --ACK Packet --> Server

The above 3 steps are followed to establish a TCP connection between source and destination.

SYN Flood DOS attacks involves sending too many SYN packets (with a bad or random source ip) to the destination server. These SYN requests get queued up on the server's buffer and use up the resources and memory of the server. This can lead to a crash or hang of the server machine.

It is a half-open connection and it takes up resources on the server machine. So if an attacker sends SYN packets faster than memory is being freed up on the server then it would be an overflow situation. Since the server's resources are used the response to legitimate users is slowed down resulting in Denial of Service.

TOPOLOGY DIAGRAM :

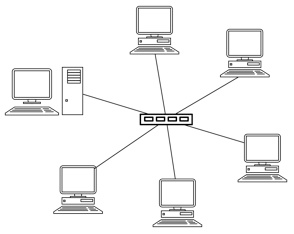


Fig. Topology diagram

Here one of the pc is attacker and one of the pc is victim.

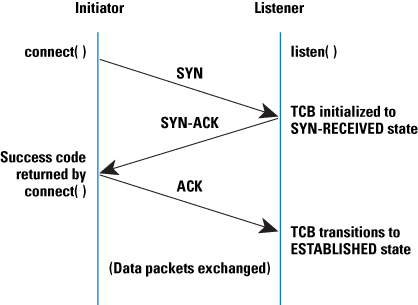


Fig. Timing diagram of original protocol

**SYN Flood Attack Timing Diagram:**

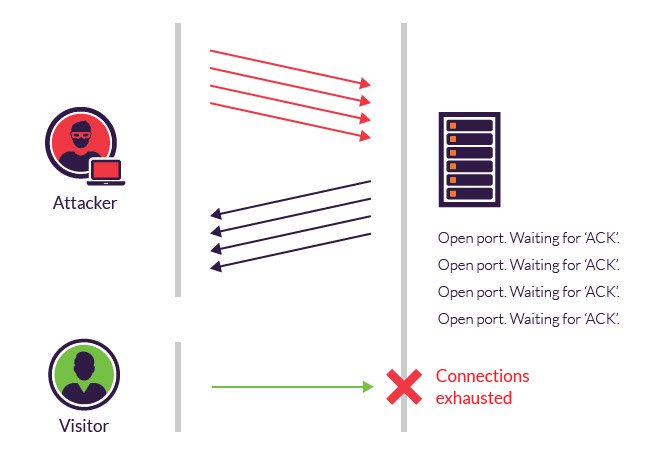


Fig. Timing diagram of TCP protocol at the time of attacking

**Attack strategy:**

A SYN flood program works by creating Syn packets which need raw socket support. Linux has raw socket support natively and hence I shall work on a Linux system. Moreover, I shall code in C/Python depending on the resource from web I get. Though both of them are platform independent, this is because the underlying socket libraries are different on Windows and Linux. Unlike Java, which doesn’t support raw socket programming, Both Python & C support it.

The theory behind my strategy is quite simple. I’ll just create a raw socket and a structure for TCP SYN packet and send the packet over the raw socket in a loop. That is all that needs to be done.

**Packet and Header details:**

Raw sockets allow a program or application to provide custom headers for the specific protocol (TCP IP) which are otherwise provided by the kernel/OS network stack. In more simple terms it’s for adding custom headers instead of headers provided by the underlying operating system.

Raw socket support is available natively in the socket API in LINUX. we are going to create raw TCP/IP packets. For this we need to know how to make proper IP header and TCP headers. A packet = IP header + TCP header + data.

IP header:

According to [RFC 791](http://www.ietf.org/rfc/rfc791.txt)

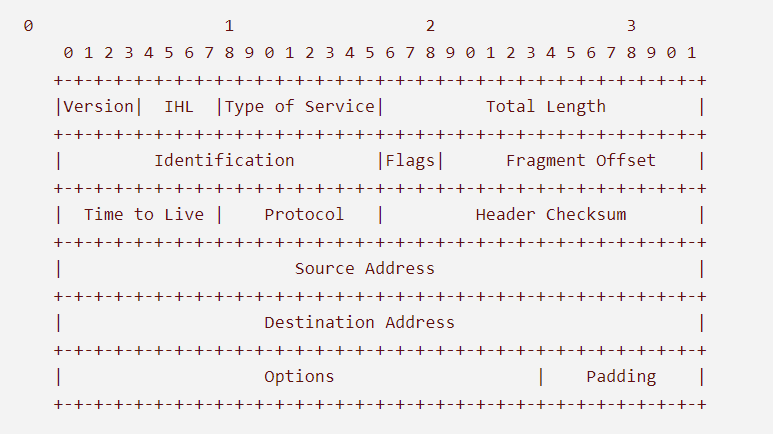


Fig. IP header format

Every single number is 1 bit. So for example the Version field is 4 bit. The header must be constructed exactly like shown.

TCP header:

According to RFC 793

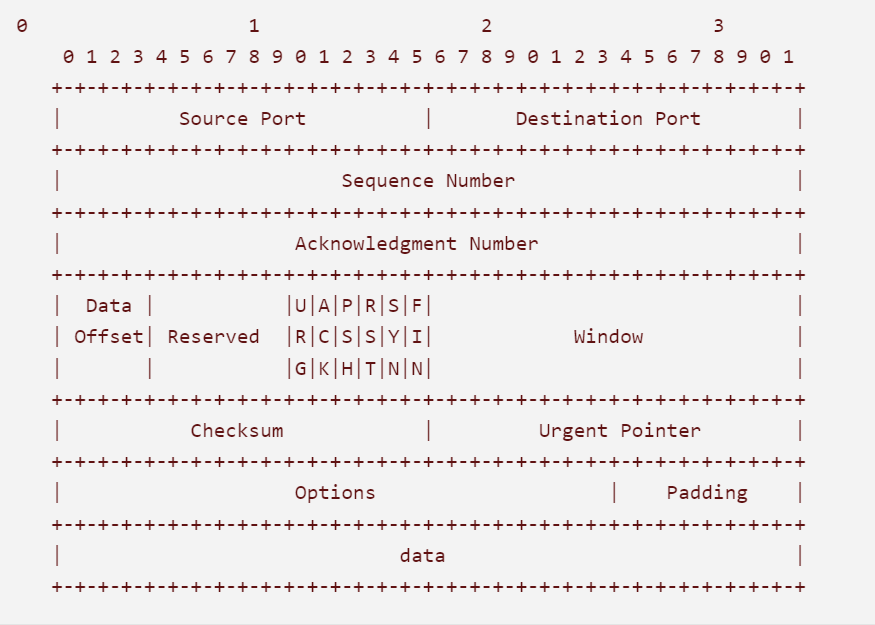


Fig. TCP header format

TCP flags

URG=0

ACK=0

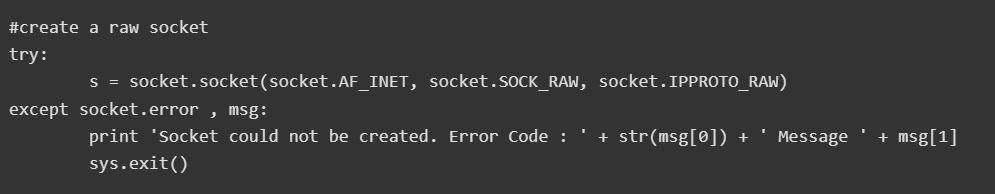
PSH=0

RST=0

SYN=1

FIN=0

Raw packet can be created like this in python



Justification:

For implementing the attack I’m going to create a raw socket through which numerous Syn packet will be sent at the victim’s side. As the victim PC has no way to stop receiving Syn packet during its active session in internet, in a few moments all of its resources will be consumed storing Syn information. Hence the denial of service will take place.