INFORMATION SECURITY ASSURANCE

Project Design Vamshi Rajarikam

Denial of Service Attacks

Attack Description:

DoS is a cyber-attack where we make a server or machine unavailable to its users, such as to temporarily suspend the service of a host connected to the internet. DoS attack is achieved by overloading the machine or server with huge number of requests in order to flood the systems with traffic and prevent its users from accessing.

DDOS(Distributed Denial of service) attack is a type of DoS where it overloads the server or machine with traffic from multiple sources and make it unable to access.

SYN FLOOD:

It is a form of Denial-of-service attack in which client/attacker repeatedly send SYN (synchronization) packets to a server/target's system in an attempt to consume enough server resources to make the system unresponsive to other client's/users.

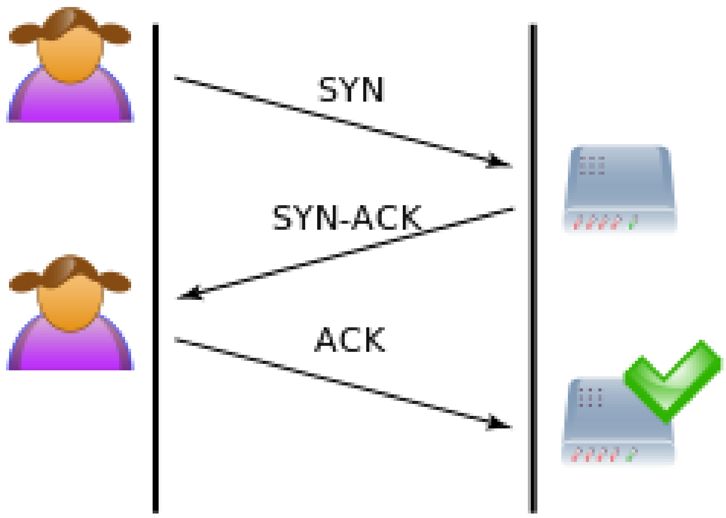

Fig. Normal connection where 3-way handshake is correctly performed.

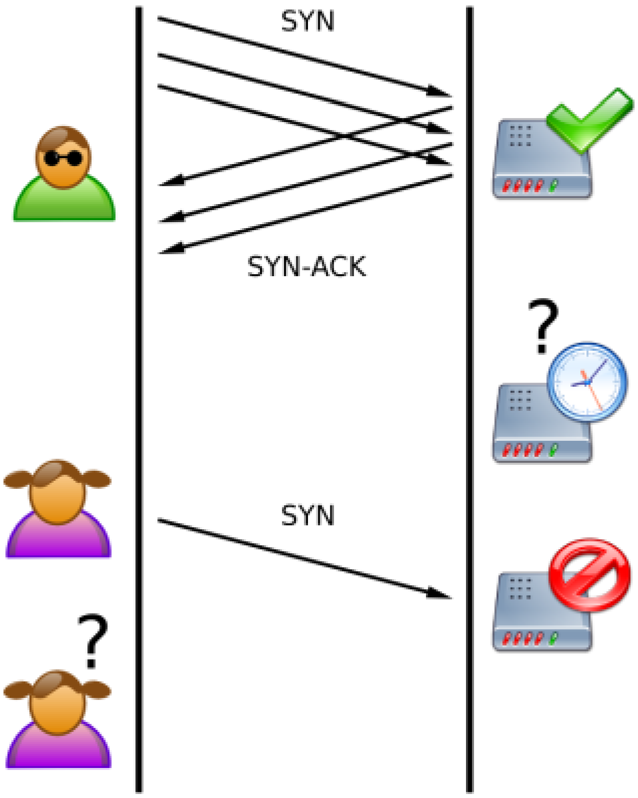

Fig. When SYN Flood attack happens. Attacker does not send the “ACK” back to the server.

Detailed Description:

Normally when a client/user attempts to start a TCP connection with server/machine, they exchange the series of protocols called TCP three-way handshake, series of steps are:

1. Client/user sends a SYN (synchronize) message to the server/machine.
2. Server/machine acknowledges the request and sends SYN-ACK to the client.
3. Client will respond with an ACK and the connection is established.

A SYN flood attack works by not responding the server/machine with **ACK.** If server does not get the respond from the client/user, it will send the SYN-ACK again. This process repeats until fully utilizing the server/machine resources. Now the server/machine is busy in sending the SYN-ACK to the client, so it couldn’t give responds to other clients/users. Denial-of-Service attack is now happened.

So, I’m going to create a program in which stops sending the ACK back to the server/machine.

SNORT RULE:

```
alert tcp !$HOME_NET any -> $HOME_NET 80 (flags: S; msg:"Possible TCP DoS"; flow: stateless;
threshold: type both, track by_dst, count 70, seconds 10; sid:10001;rev:1;)
```