2.8 Three-Way Handshake

Three-Way Handshake

Before we get into how the Three-Way Handshake works, here is a short explanation of a botnet. Compare it to your answer in the previous activity.

A botnet is a logical collection of Internet-connected devices (computers, smartphones or IoT devices) that have been infected and are controlled by a third party.

A controller software (known as 'command & control') is able to direct the activities of each compromised device (known as a 'bot') using network protocols like HTTP. Botnets are used for distributed denial-of-service (DDoS) attacks and other types of attacks (spam, data theft, etc).

Now let's look at another DoS attack which exploits how the communications in a network are established.

A **Three-Way Handshake**, also known as a TCP 3-Way Handshake, is a process which is used to establish a connection between two parties on a network. It is a three-step process that requires both the parties to exchange synchronisation and acknowledgement packets before the real data communication process starts.

This is what the process looks like visually with a text explanation following.

Step 1: A connection between server and client is established

First, a connection between server and client is established, so the target server must have open ports that can accept and initiate new connections. The client node sends a SYN (Synchronise Sequence Number) data packet to a server on the same or an external network. This SYN packet is a random sequence number that the client wants to use for the communication (for example, X). The objective of this packet is to ask if the server is open for new connections.

Step 2: The server receives the SYN packet from the client node and responds

When the server receives the SYN packet from the client node, it returns a confirmation receipt – the SYN/ACK (Acknowledgement Sequence Number) packet. This packet includes two sequence numbers. The first one is the SYN sent by the server itself, which is another random sequence number (for example, Y). The second one is the ACK, which is set by the server to be the sequence number it received from the client plus one (eg, X+1). This sequence indicates that the server correctly acknowledged the client's packet, and that it is sending its own packet to be acknowledged as well.

Step 3: Client node receives the SYN/ACK from the server and responds with an ACK packet

The client node receives the SYN/ACK from the server and responds with an ACK packet. Once again, each side must acknowledge the sequence number received by incrementing it by one. So now it's the turn of the client to acknowledge the server's packet by adding one to the sequence number (in this case, Y+1), and resend it to the server. Upon completion of this process, the connection is created and the host and server can communicate.

All these steps are necessary to verify the sequence numbers originated by both sides, guaranteeing the stability of the connection.

SYN flooding

A SYN flood attack works by not responding to the server with the expected ACK code. The malicious client can either not send the expected ACK, or more effectively spoof the source IP address in the SYN, causing the server to send the SYN-ACK to a falsified IP address – which will not send an ACK because it 'knows' that it never sent a SYN. The server will wait for the missing ACK for a bit. However, the resources bound on the server may eventually exceed the resources available and the server cannot connect to any client.A SYN flooding attack can be broken down as follows:

1. The attacker sends a high volume of SYN packets to the targeted server, often with spoofed IP addresses.
2. The server then responds to each one of the connection requests and leaves an open port ready to receive the response.
3. While the server waits for the final ACK packet, which never arrives, the attacker continues to send more SYN packets. The arrival of each new SYN packet causes the server to temporarily maintain a new open port connection for a certain length of time, and once all the available ports have been utilised the server is unable to function normally.
4. A legitimate client, who is trying to connect to the targeted server, won't be able to connect with it as the server's resources are consumed.