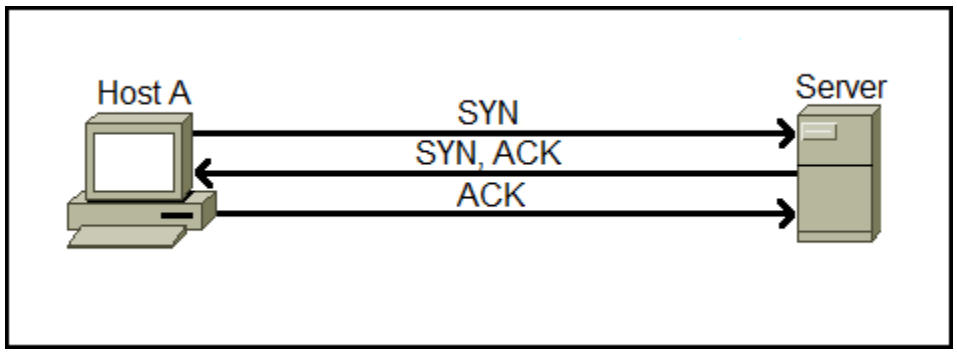
Lab - Capturing a 3-way TCP Handshake Using Wireshark

Overview

A Three-way handshake or a TCP 3-way handshake is a process that is used in a TCP/IP network to make a connection between the server and client. It is a three-step process that requires both the client and server to exchange synchronization and acknowledgment packets before the actual communication process starts.

TCP message types

Message	Description
Syn	Used to initiate and establish a connection. It also helps you to synchronize sequence numbers between devices.
ACK	Helps to confirm to the other side that it has received the SYN.
SYN-ACK	SYN message from local device and ACK of the earlier packet.
FIN	Used to terminate a connection.



- Host A begins the connection by sending the TCP SYN packet to its host destination. The packets contain a random sequence number (For example, 4321) that indicates the beginning of the sequence numbers for data that the Host X should transmit.
- After that, the server will receive the packet, and it responds with its sequence number. The server's response also includes the acknowledgment number, that is, Host A's sequence number incremented with 1 (Here, it is 4322).
- Host A responds to the server by sending the acknowledgment number that is mostly the server's sequence number that is incremented by 1.
- After the data transmission process is over, TCP automatically terminates the connection between two separate endpoints.

Key Points

Malicious network traffic begins the same way as legitimate traffic, using a TCP 3-way handshake. If a client or server has been taken over and is now part of a botnet, the traffic coming and going must still adhere to the rules of how devices communicate. The difference would be a large amount of network traffic the zombie machine would be generating as part of the botnet.

Lab Requirements

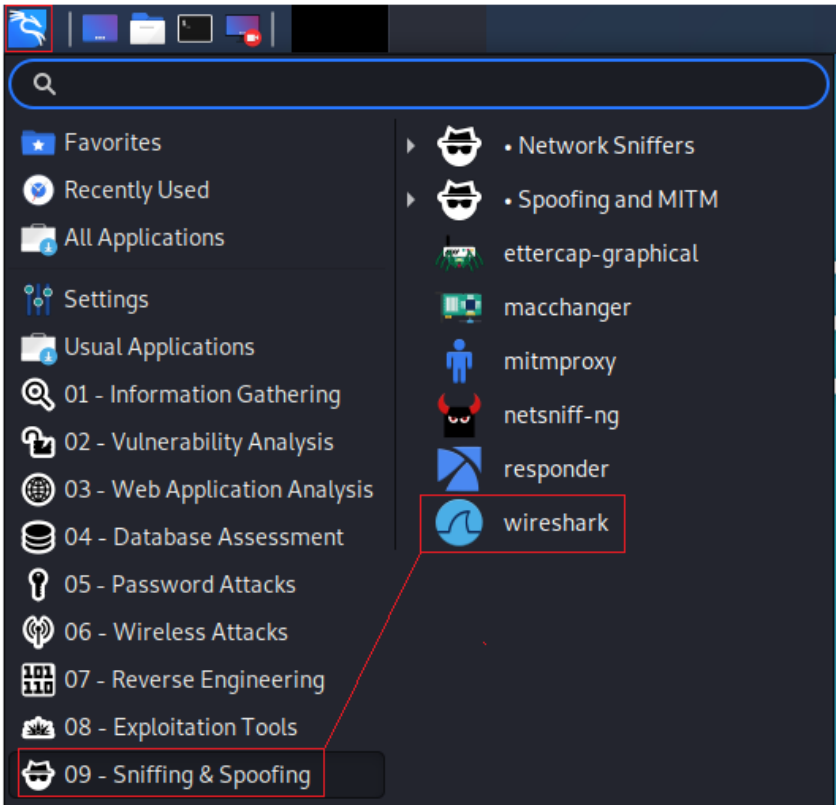
- One virtual install of Kali Linux
- Wireshark
- Internet connection

Lab Scenario

In this lab, we will look at a normal 3-way handshake using Wireshark. We will also learn how to filter just the information we wish to see.

Begin the lab

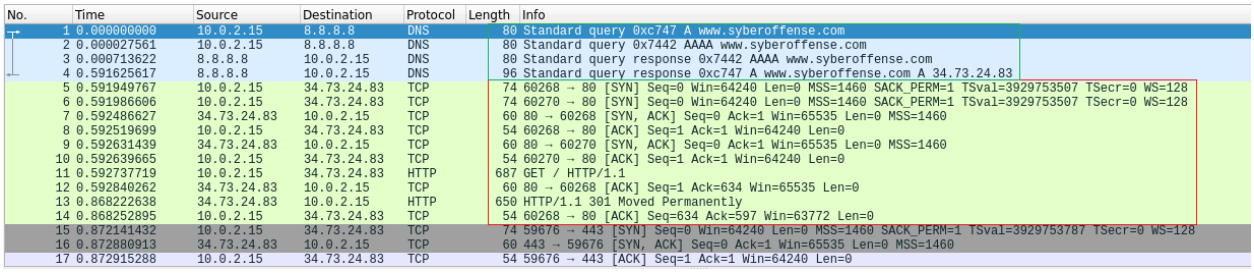
Using the Kali Quick Launch toolbar, under Sniffing and Spoofing, find and launch Wireshark.



For this lab, I will be using my wired adapter, but you are free to use a wireless or wired adapter. Ensure that your Kali machine has Internet access.

- Start a packet capture with Wireshark.
- Next, open a browser and navigate to www.syberoffense.com
- Once the website has loaded, stop the Wireshark capture.

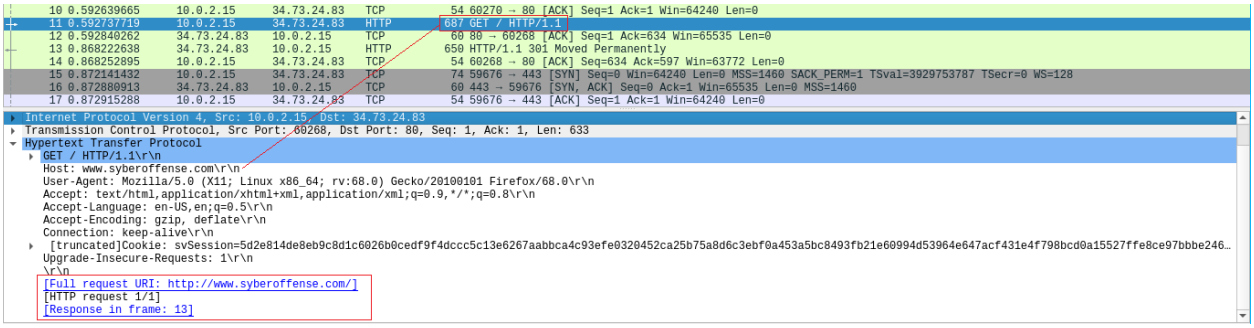
We can now examine the Wireshark capture and see the TCP 3 Way Handshake as it happened.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	DNS	80	Standard query 0xc747 A www.syberoffense.com
2	0.000027561	10.0.2.15	8.8.8.8	DNS	80	Standard query 0x7442 AAAA www.syberoffense.com
3	0.000713622	8.8.8.8	10.0.2.15	DNS	80	Standard query response 0x7442 AAAA www.syberoffense.com
4	0.591625617	8.8.8.8	10.0.2.15	DNS	96	Standard query response 0xc747 A www.syberoffense.com A 34.73.24.83
5	0.591949767	10.0.2.15	34.73.24.83	TCP	74	60268 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3929753507 TSecr=0 WS=128
6	0.591986606	10.0.2.15	34.73.24.83	TCP	74	60270 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3929753507 TSecr=0 WS=128
7	0.592486627	34.73.24.83	10.0.2.15	TCP	60	80 → 60268 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
8	0.592519699	10.0.2.15	34.73.24.83	TCP	54	60268 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
9	0.592631439	34.73.24.83	10.0.2.15	TCP	60	80 → 60270 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
10	0.592639665	10.0.2.15	34.73.24.83	TCP	54	60270 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
11	0.592737719	10.0.2.15	34.73.24.83	HTTP	687	GET / HTTP/1.1
12	0.592840262	34.73.24.83	10.0.2.15	TCP	60	80 → 60268 [ACK] Seq=1 Ack=634 Win=65535 Len=0
13	0.868222638	34.73.24.83	10.0.2.15	HTTP	650	HTTP/1.1 301 Moved Permanently
14	0.868252895	10.0.2.15	34.73.24.83	TCP	54	60268 → 80 [ACK] Seq=634 Ack=597 Win=63772 Len=0
15	0.872141432	10.0.2.15	34.73.24.83	TCP	74	59676 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3929753787 TSecr=0 WS=128
16	0.872880913	34.73.24.83	10.0.2.15	TCP	60	443 → 59676 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
17	0.872915288	10.0.2.15	34.73.24.83	TCP	54	59676 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0

The first three lines show that my website request for www.sybersoffence.com was converted to an IP address or machine language (Line 5). On the same line, you can see the beginning of the 3-way handshake. The three-way is concluded by line 10, and on line 11, we see the GET / HTTP:1.1 request, which is a direct request for a web page from the source IP address.

If we examine the contents of the GET / HTTP:1.1 packet by double-clicking it, in the second windows pane, we can see what web page we were requesting.



```
    10 0.592639665    10.0.2.15     34.73.24.83   TCP      54 60270 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
    11 0.592737719    10.0.2.15     34.73.24.83   HTTP    687 GET / HTTP/1.1
    12 0.592840262    34.73.24.83   10.0.2.15     TCP      60 80 → 60268 [ACK] Seq=1 Ack=634 Win=65535 Len=0
    13 0.868222638    34.73.24.83   10.0.2.15     HTTP    650 HTTP/1.1 301 Moved Permanently
    14 0.868252895    10.0.2.15     34.73.24.83   TCP      54 60268 → 80 [ACK] Seq=634 Ack=597 Win=63772 Len=0
    15 0.872141432    10.0.2.15     34.73.24.83   TCP      74 59676 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3929753787 TSecr=0 WS=128
    16 0.872880913    34.73.24.83   10.0.2.15     TCP      60 443 → 59676 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
    17 0.872915288    10.0.2.15     34.73.24.83   TCP      54 59676 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.73.24.83
▸ Transmission Control Protocol, Src Port: 60268, Dst Port: 80, Seq: 1, Ack: 1, Len: 633
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Host: www.syberoffense.com\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
  ▸ [truncated]Cookie: svSession=5d2e814de8eb9c8d1c6026b0cedf9f4dccc5c13e6267aabbca4c93efe0320452ca25b75a8d6c3ebf0a453a5bc8493fb21e60994d53964e647acf431e4f798bcd0a15527ffe8ce97bbbe246…
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.syberoffense.com/]
    [HTTP request 1/1]
    [Response in frame: 13]
```

Summary –

As a network administrator, pentester, or digital forensic investigator, it is important to discern malicious traffic from legitimate traffic. This begins with being able to look at and analyze a three-way handshake.