Wireshark 3-Way Handshake Tutorial

Eoin Fitzsimons

A 3-way handshake is the process used in a TCP (Transmission Control Protocol) connection to establish communication between a client and server. This handshake ensures a reliable connection before actual data transmission begins.

**Step 1: Install Wireshark**

If you haven't installed Wireshark, you can download it from [Wireshark's official website](https://www.wireshark.org/download.html). It's available for Windows, macOS, and Linux.

**Step 2: Capture Network Traffic**

1. Start Wireshark: Open Wireshark, and you'll see a list of network interfaces.

2. Select Network Interface: Choose the network interface (like Ethernet, Wi-Fi) from which you want to capture traffic. For example, if you're connected via Wi-Fi, select your wireless interface.

3. Start Capturing: Click on the blue shark fin icon at the top left to start capturing packets.

**Step 3: Establish a TCP Connection (Generate the Handshake)**

To generate a TCP 3-way handshake, you can use a tool that establishes a TCP connection, such as:

* Open a web browser: Visit any website (e.g., http://example.com).
* Command Line Tools: Use telnet, curl, or any application that makes a TCP connection.

Example command:

telnet example.com 80

This command initiates a TCP connection to port 80 (HTTP) of example.com.

Step 4: Stop Capture

Once you've initiated a connection, stop the capture by clicking on the red square at the top left.

**Step 5: Filter the Capture for TCP Handshake**

To focus only on the TCP handshake, apply the following filter in the display filter bar at the top of Wireshark:

**tcp.flags.syn == 1**

This filter will show packets involved in the SYN phase, which is part of the TCP handshake.

A screenshot of a computer

Description automatically generated

**Step 6: Analyse the TCP 3-Way Handshake**

The 3-way handshake involves three main steps:

1. SYN (Synchronize):

- The client sends a TCP SYN packet to the server to initiate the connection.

- Look for a packet with SYN set in the "Flags" field.

2. SYN-ACK (Synchronize-Acknowledge):

- The server responds to the client's SYN request with a SYN-ACK packet.

- This packet will have both SYN and ACK flags set in the TCP header.

3. ACK (Acknowledge):

- The client responds to the server's SYN-ACK with an ACK packet to finalize the handshake.

- The final packet will have only the ACK flag set.

# Sample Handshake in Wireshark

1. Client SYN:

- Source: Client IP ip.src == 10.3.226.134 Source: AzureWaveTec\_22:ce:af (70:66:55:22:ce:af)

- Destination: Server IP ip.dst == 93.184.215.14 Destination: JuniperNetwo\_ff:10:04 (00:10:db:ff:10:04)

- Flags: SYN

- Sequence number: Seq=0

88 8.115902 10.3.226.134 93.184.215.14 TCP 66 60481 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK\_PERM

2. Server SYN-ACK:

- Source: Server IP ip.src == 93.184.215.14

- Destination: Client IP ip.dst == 10.3.226.134

- Flags: SYN, ACK

- Sequence number: Seq=0, Acknowledgment number: Ack=1

89 8.211278 93.184.215.14 10.3.226.134 TCP 66 80 → 60481 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK\_PERM WS=512

3. Client ACK:

- Source: Client IP

- Destination: Server IP

- Flags: ACK

- Sequence number: Seq=1, Acknowledgment number: Ack=1

90 8.211375 10.3.226.134 93.184.215.14 TCP 54 60481 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0

**Step 7: Interpret the Capture**

- TCP Handshake Complete: Once you see these three packets (SYN, SYN-ACK, ACK), the TCP connection is established.

- Relative Sequence Numbers: Wireshark displays relative sequence numbers by default, which simplifies the view.

- Flags: The flags in the TCP header (SYN, ACK) help in identifying each step of the handshake.