Principles of Information Security

HW4 Using Wireshark

Anonymous

日期: 2024/03/27

2024-2025 春 学期

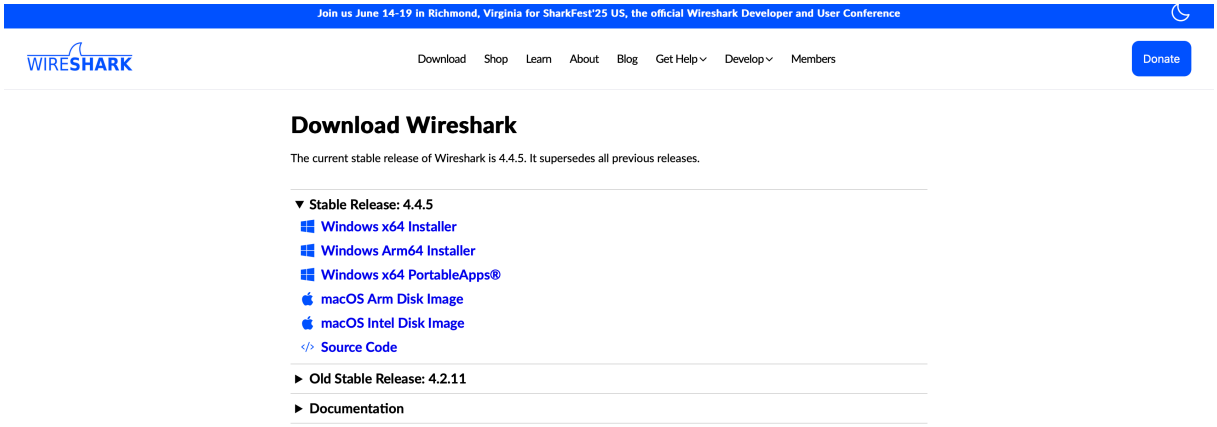
Table of Contents

1 Download & Install Wireshark

1.1 Download Wireshark

- Operating System: MacOS

Download from the official website of Wireshark. According to your specific operating system, choose different versions to download:



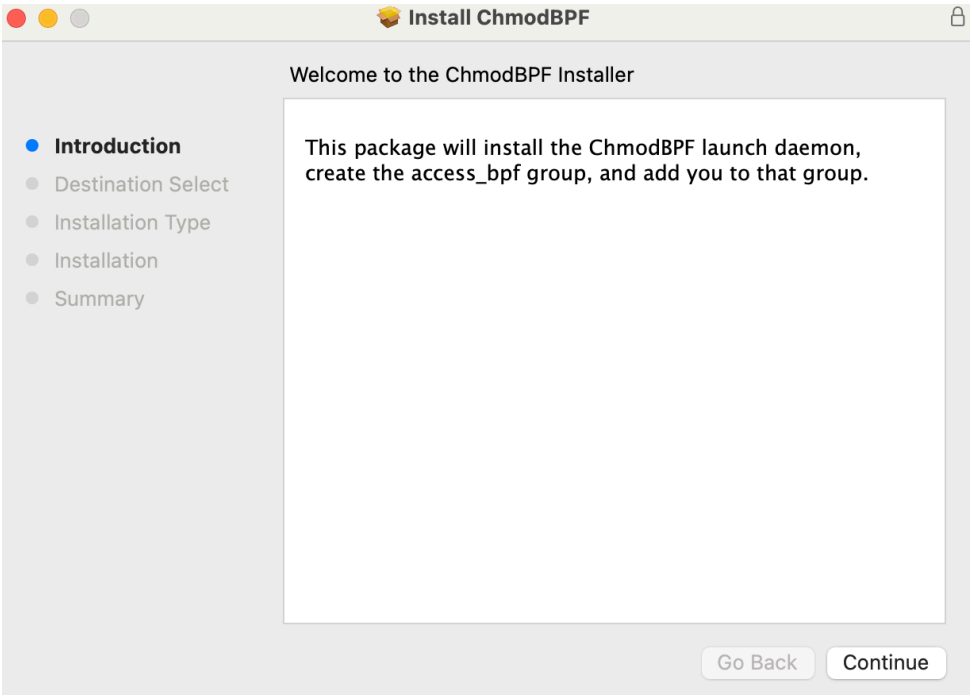
For MacOS users, simply click `MacOS Arm Disk Image`, and then a `dmg` file will be downloaded.

1.2 Install Wireshark

After successfully downloading the `dmg` file, double click to start the installation, an interface like the following will appear:

1. Drag the icon Wireshark.app to the folder Application.
2. Double click Install ChmodBPF.pkg to install ChmodBPF. ChmodBPF is a utility associated with Wireshark on macOS. It is used to manage permissions for capturing network packets.

Follow the instructions of Install ChmodBPF, and the installation process will finish successfully.



2 Capture all packages when you access https://www.zju.edu.cn (using filter)

2.1 Get the IP Address of https://www.zju.edu.cn

Open the terminal (zsh for instance), input ping www.zju.edu.cn and press enter.

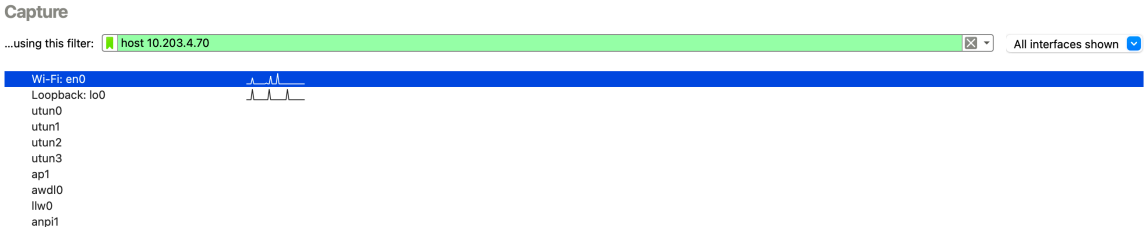
Result like the following will be acquired:

```
~/ ping www.zju.edu.cn
PING www.zju.edu.cn (10.203.4.70): 56 data bytes
64 bytes from 10.203.4.70: icmp_seq=0 ttl=60 time=18.706 ms
64 bytes from 10.203.4.70: icmp_seq=1 ttl=60 time=13.100 ms
64 bytes from 10.203.4.70: icmp_seq=2 ttl=60 time=17.235 ms
64 bytes from 10.203.4.70: icmp_seq=3 ttl=60 time=17.862 ms
64 bytes from 10.203.4.70: icmp_seq=4 ttl=60 time=18.854 ms
64 bytes from 10.203.4.70: icmp_seq=5 ttl=60 time=18.222 ms
64 bytes from 10.203.4.70: icmp_seq=6 ttl=60 time=13.622 ms
64 bytes from 10.203.4.70: icmp_seq=7 ttl=60 time=13.365 ms
64 bytes from 10.203.4.70: icmp_seq=8 ttl=60 time=15.261 ms
^C
--- www.zju.edu.cn ping statistics ---
9 packets transmitted, 9 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 13.100/16.247/18.854/2.271 ms
```

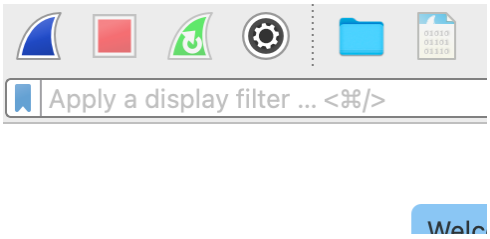
Notice that there is an IP address 10.203.4.70, remember the IP address which will be used in Wireshark app to capture packages.

2.2 Capture Packages when accessing the website

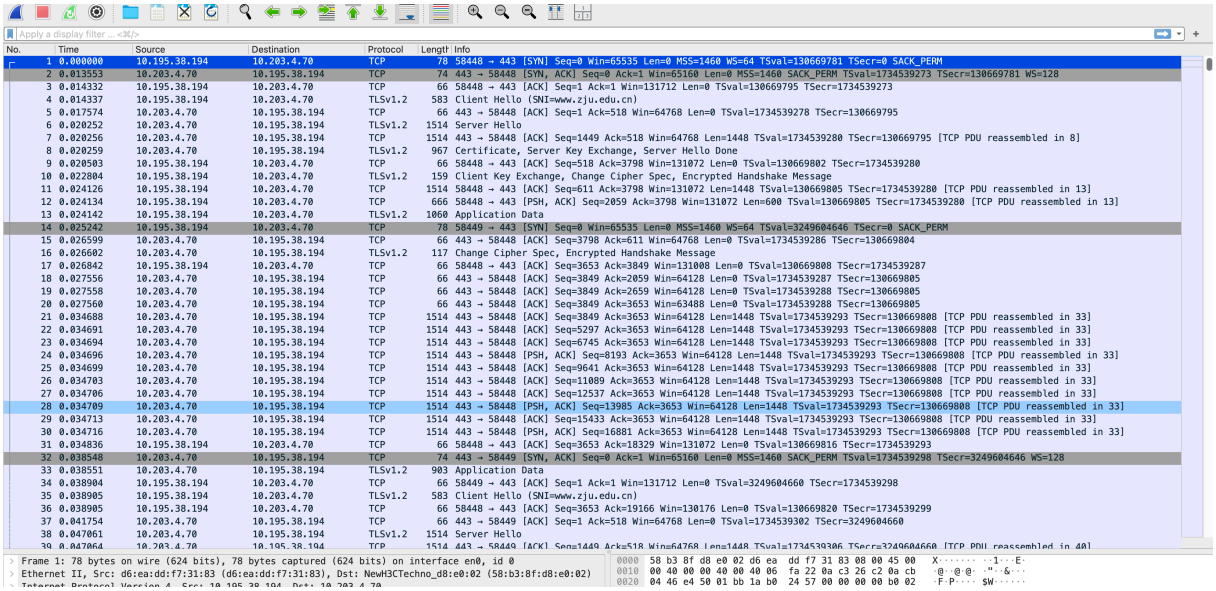
Open the Wireshark app, input the filter condition host 10.203.4.70 and select Wi-Fi: en0.



Press the blue shark fin on the left upper corner to start capturing:



Then open the website https://www.zju.edu.cn in the browser (Google Chrome for instance), Wireshark will capture all the packages when accessing the website:



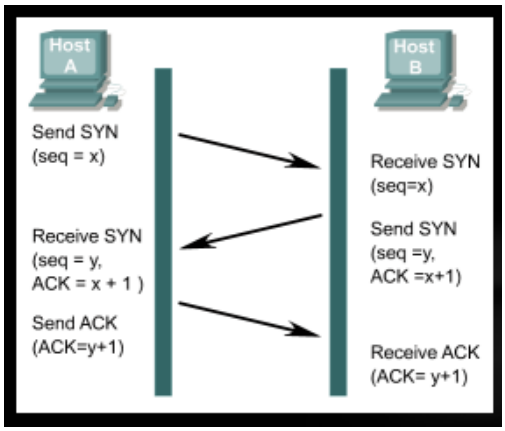
Click the red square button to stop capturing, and then click the form-like button to save the captured file.

3 Using Wireshark to analyze the packets

Http connection is based on TCP, so in order to construct an HTTP connection, TCP connection has to be established first.

3.1 Analyzing TCP 3-Way Handshake

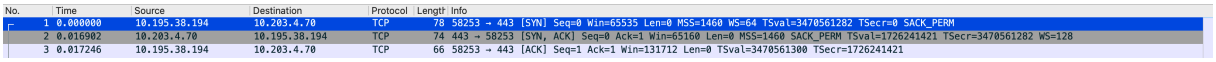
The establishment of TCP can be concluded as 3-way handshake:



In order to find the packets corresponding to 3-way handshake, simply look for three consecutive packets with the following pattern:

1. SYN: Client → Server with [SYN] flag
2. SYN-ACK: Server → Client with [SYN, ACK] flags
3. ACK: Client → Server with [ACK] flag

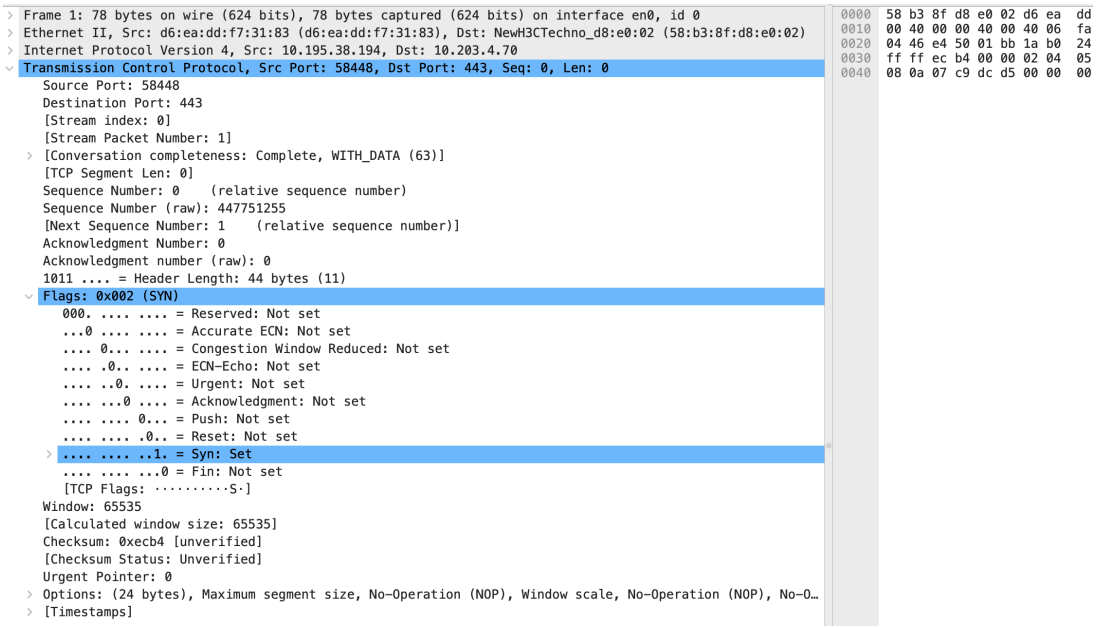
As shown in the following:



3.1.1 First Stage: SYN (Synchronize) –> Client -> Server

The client sends a SYN packet to the server, indicating it wants to establish a connection. The packet includes:

- SYN flag = 1 (indicating a new connection request)
- A randomly generated initial sequence number (ISN) (e.g., Seq=0)

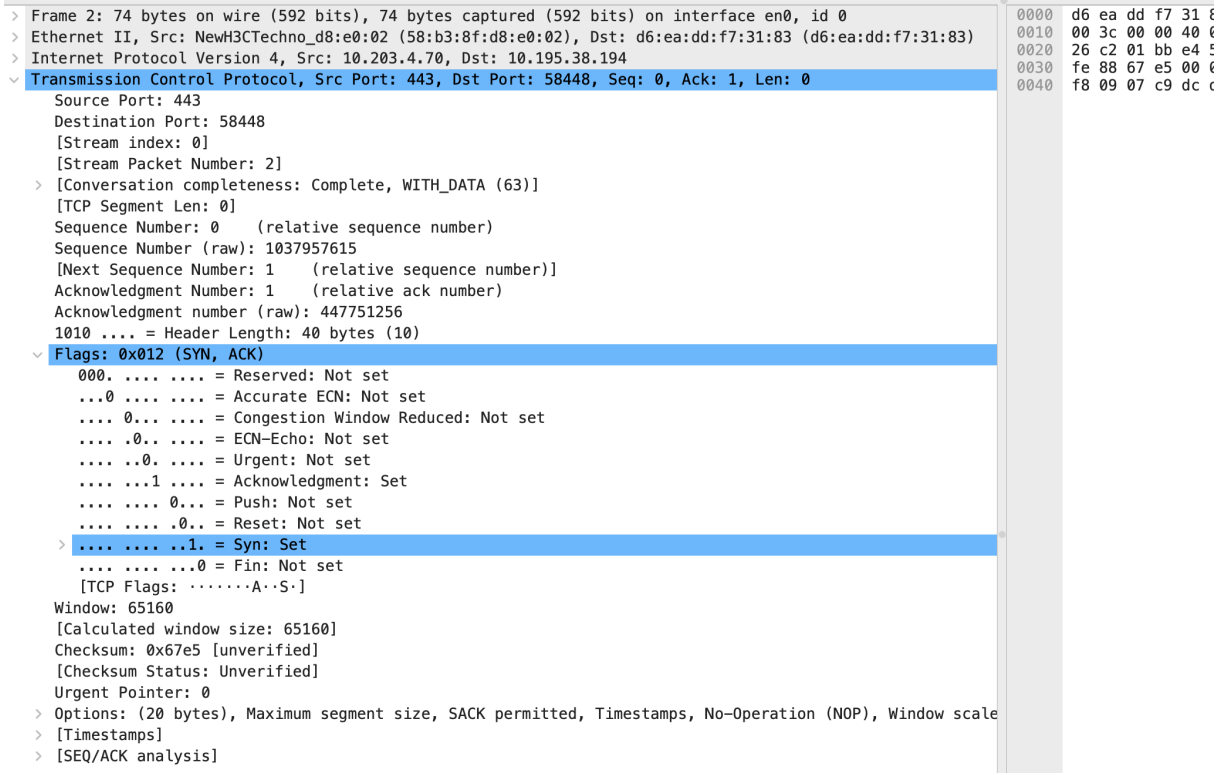


As is shown in the figure above, SYN flag is set to 1 and the randomly generated number is 447751255, while the source port is 58448, the destination port is 443.

3.1.2 Second Stage: SYN-ACK (Synchronize-Acknowledge) –> Server -> Client

The server acknowledges the client's request by sending:

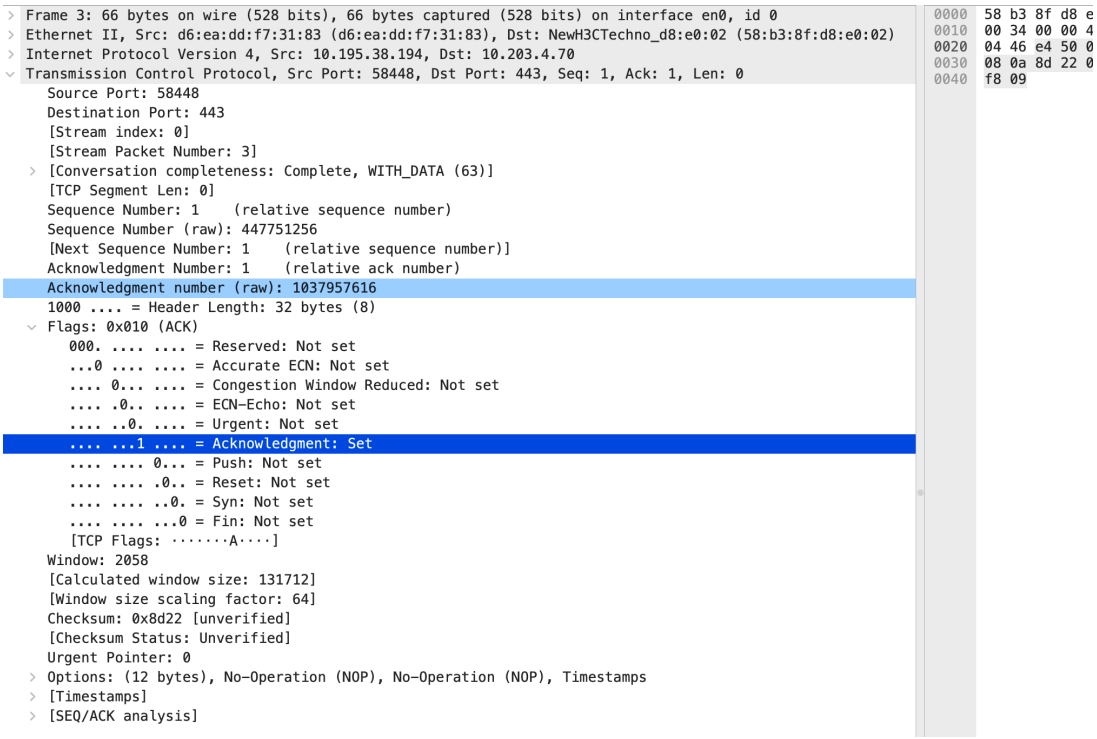
- SYN flag = 1 (its own synchronization request)
- ACK flag = 1 (acknowledging the client's SYN)
- Acknowledgment number = Client's ISN + 1 (e.g., Ack=1)
- Its own initial sequence number (ISN) (e.g., Seq=0)



The source port is 443, the destination port is 58448, reverse of the previous one. As is shown, SYN flag is set to 1, so does ACK. Simultaneously, Acknowledge number (447751256) is set equal to Client's ISN (447751255) + 1, and it generated its own ISN 1037957615.

3.1.3 Third Stage: ACK (Acknowledge) –> Client -> Server

- The client confirms the server's SYN-ACK by sending:
  1. ACK flag = 1 (acknowledging the server's SYN)
  2. Acknowledgment number = Server's ISN + 1 (e.g., Ack=1)

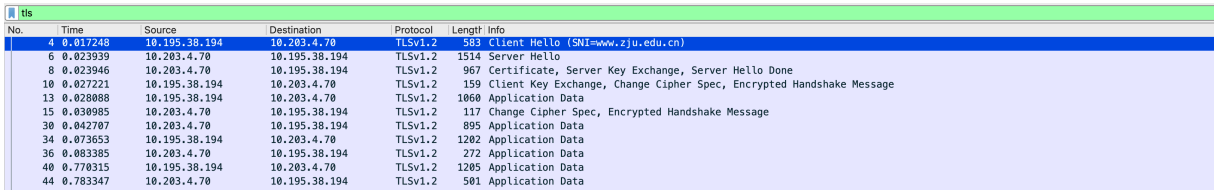


Same as the first stage, the source port is 58448 while the destination port is 443. Additionally, ACK flag (Acknowledgement) is set to 1, while the acknowledge number is set to 1037957616 = 1037957615 + 1.

At this point, the connection is established, and data transfer can begin.

3.2 Examining HTTPS Request/Response

In order to get HTTPS (TLS) traffic, use the filter tls in the display filter:



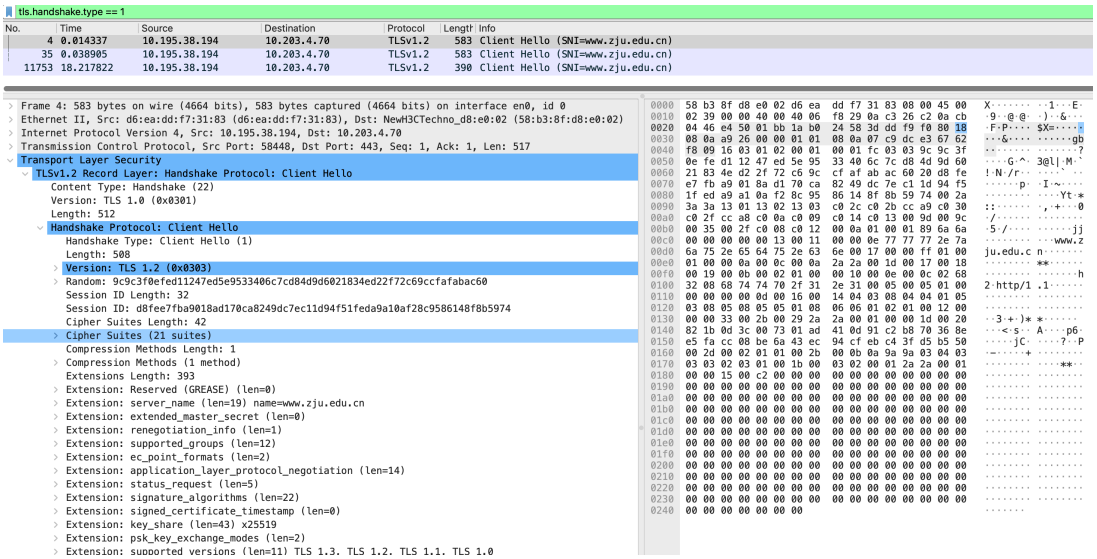
This will show all TLS-encrypted traffic between local machine and the server.

3.2.1 TLS Handshake Step 1: Client Hello

Filter: tls.handshake.type == 1.

The client (browser) sends a Client Hello, which includes:

- Supported TLS versions (e.g., TLS 1.2, TLS 1.3)
- Cipher suites (encryption methods it can use)
- Server Name Indication (SNI) → Reveals the domain (www.zju.edu.cn) even though the traffic is encrypted.



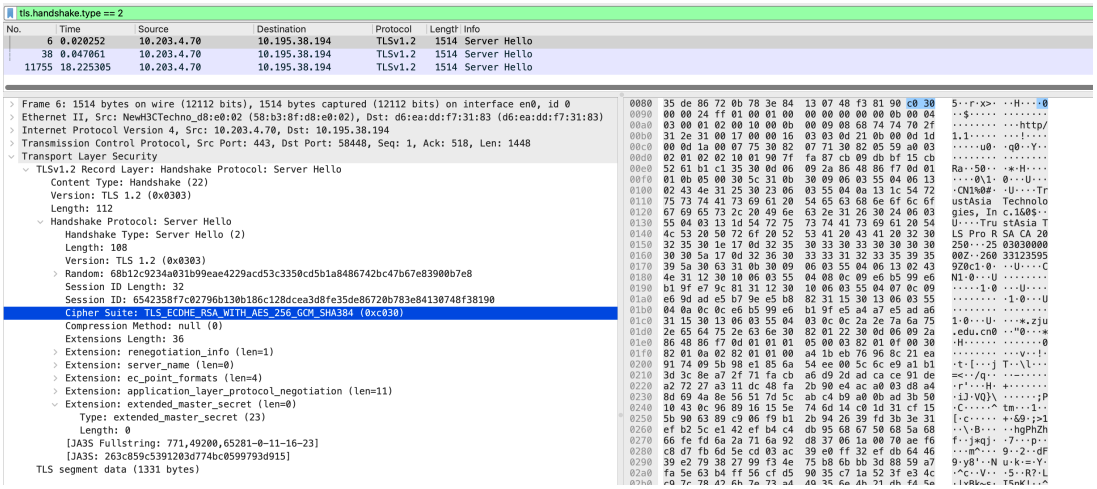
As is shown in the figure above, TLS version is 1.2, cipher suites is 21 and SNI is www.zju.edu.cn.

3.2.2 TLS Handshake Step 2: Server Hello

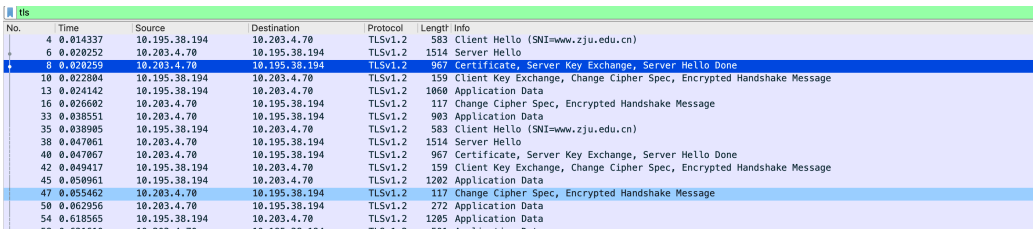
Filter: tls.handshake.type == 2

The server responds with:

- Selected TLS version & cipher suite
- Its SSL certificate



3.2.3 TLS Handshake Step 3: Key Exchange & Encryption Setup



As shown, No.8 to 47 are the corresponding key exchange and encryption setup steps. The client and server exchange keys (Diffie-Hellman, RSA, etc.) to establish encryption.

After this, all HTTP traffic is encrypted and appears as "Application Data" in Wireshark.

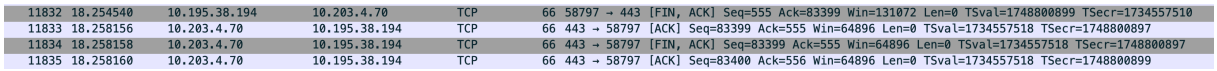
3.3 Analyzing TCP Connection Termination (4-Way Handshake)

A proper TCP connection termination involves 4 steps (though sometimes it appears as 3 packets if FIN and ACK are combined).

- `curl -v https://www.zju.edu.cn` to terminate the connection.

Step	Direction	Packet Type	Description
1	Client → Server	FIN + ACK	The client (or server) initiates closure by sending FIN (Finish)
2	Server → Client	ACK	The other side acknowledges the FIN but may still send remaining data
3	Server → Client	FIN + ACK	The server (or client) sends its own FIN to fully close the connection
4	Client → Server	ACK	Final acknowledgment; the connection is now closed

Here is the Screenshot from Wireshark that corresponds to the above procedure:



1. Packet 11832: [FIN, ACK] (Client → Server)
   - Seq=555, Ack=83399
   - Indicates the client wants to close the connection.
2. Packet 11833: [ACK] (Server → Client)
   - Seq=83399, Ack=555
   - Server acknowledges the FIN but may still send data.
3. Packet 11834: [FIN, ACK] (Server → Client)
   - Seq=83399, Ack=555
   - Server initiates its own closure.
4. Packet 11835: [ACK] (Client → Server)
   - Seq=83400, Ack=556
   - Final acknowledgment; connection fully closed.