

Part 1: Capturing HTTP Traffic.

Task 1: Start Wireshark and capture packets.

Step 1: Open Wireshark.

Step 2: Select the network interface connected to the internet (e.g., Ethernet or Wi-Fi).

Step 3: Click the "Start Capturing Packets" button (the shark fin icon).

Step 4: Open your favorite web browser and navigate to (<http://neverssl.com/>) website.

Step 5: After the website has fully loaded, stop capturing packets by clicking the red stop button in Wireshark.

Task 2: Filter HTTP packets and analyze them.

Step 1: In the filter bar, type http and press Enter. This filters out only the HTTP packets from the capture.

Step 2: Select any HTTP packet to view its details.

Step 3: Observe the HTTP request and response messages. Note the method (GET, POST), URL, response codes (200 OK, 404 Not Found), etc.

The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for various functions. The packet list pane shows a list of captured packets, with the following details visible:

No.	Time	Source	Destination	Protocol	Length	Info
26	0.928244	172.20.10.9	149.154.165.111	HTTP	198	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
1289	15.059231	2001:16a2:c062:a779...	2600:1f13:37c:1400:...	HTTP	598	GET /online HTTP/1.1
1291	15.338079	2600:1f13:37c:1400:...	2001:16a2:c062:a779...	HTTP	617	HTTP/1.1 301 Moved Permanently (text/html)
1299	15.358799	2001:16a2:c062:a779...	2600:1f13:37c:1400:...	HTTP	599	GET /online/ HTTP/1.1
1339	15.661889	2600:1f13:37c:1400:...	2001:16a2:c062:a779...	HTTP	373	HTTP/1.1 200 OK (text/html)
1361	15.835894	2001:16a2:c062:a779...	2600:1f13:37c:1400:...	HTTP	508	GET /favicon.ico HTTP/1.1
1379	16.092278	2600:1f13:37c:1400:...	2001:16a2:c062:a779...	HTTP	490	HTTP/1.1 200 OK (PNG)

The packet details pane for the selected packet (No. 1291) shows the following information:

- Frame 1289: Packet, 598 bytes on wire (4784 bits), 598 bytes captured (4784 b)
- Ethernet II, Src: ChongqingFug_c4:21:8d (a8:93:4a:c4:21:8d), Dst: b6:fa:48:3c:52:64
- Internet Protocol Version 6, Src: 2001:16a2:c062:a779:10f7:9885:db79:59f7, Dst: 2001:16a2:c062:a779:10f7:9885:db79:59f7
- Transmission Control Protocol, Src Port: 63771, Dst Port: 80, Seq: 1, Ack: 1, Win: 0, Len: 0
- Hypertext Transfer Protocol
 - GET /online HTTP/1.1\r\n
 - Host: beautifulquietclearlight.neverssl.com\r\n
 - Connection: keep-alive\r\n
 - Cache-Control: max-age=0\r\n
 - Upgrade-Insecure-Requests: 1\r\n
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36\r\n
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
 - Accept-Encoding: gzip, deflate\r\n
 - Accept-Language: ar,en-US;q=0.9,en;q=0.8\r\n
 - \r\n
 - [Response in frame: 1291]
 - [Full request URI: http://beautifulquietclearlight.neverssl.com/online]

The packet bytes pane shows the raw data of the packet, including the HTTP request line and headers.

Part 2: Analyzing TCP/IP Traffic.

Task 1: Filter TCP packets

Step 1: Clear the previous filter and type TCP to focus on TCP packets.

Step 2: Select a TCP packet related to your HTTP request/response.

Step 3: Right-click on the packet and select "Follow" -> "TCP Stream".

Step 4: This shows the entire conversation between the client and server.

Task 2: Analyze TCP handshake and investigate Data Transfer and Termination

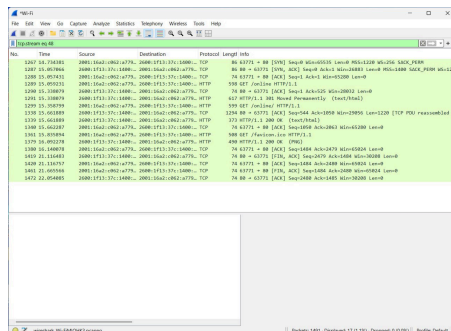
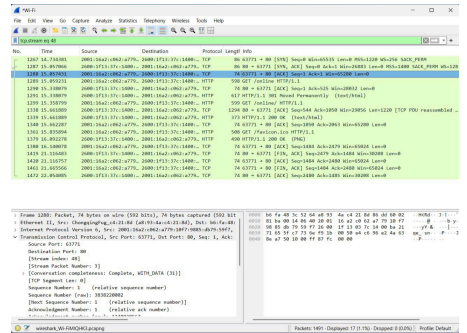
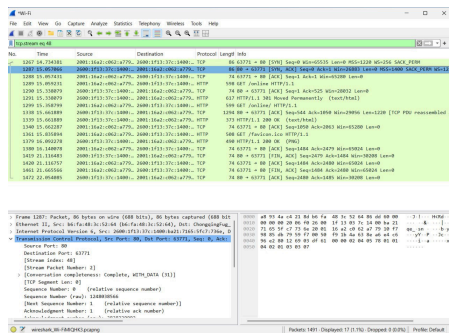
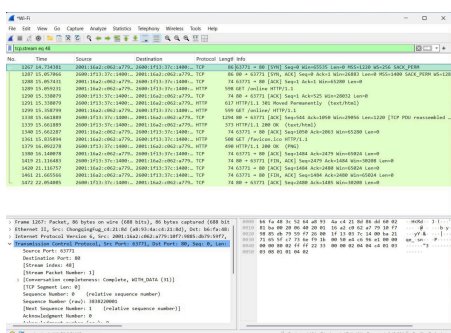
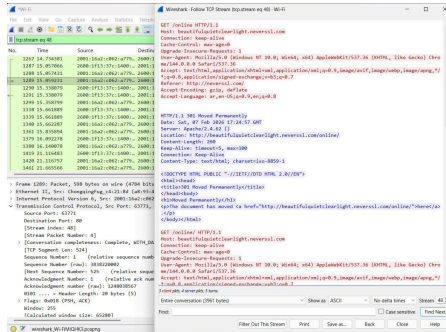
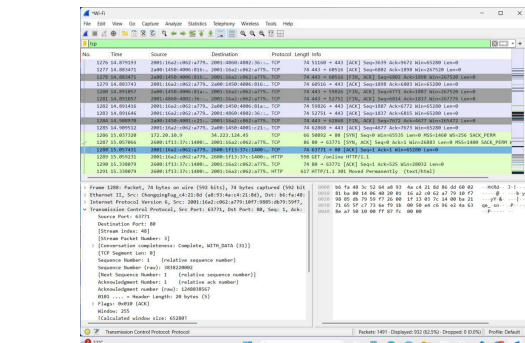
Step 1: Find and select packets related to the TCP three-way handshake:

- SYN: Initiates a connection.
- SYN-ACK: Acknowledges and responds to the SYN.
- ACK: Acknowledges the SYN-ACK and establishes the connection.

Step 2: Note the sequence and acknowledgment numbers. Screenshot and upload your image to your online git repository.

Step 3: Observe the data packets exchanged between the client and server. Take a screenshot and upload it to your online git repo.

Step 4: Look at the TCP termination process (FIN, ACK packets).



Part 3: Capturing and Analyzing UDP Traffic

Task 1: Generate UDP traffic and capture packets

Step 1: Open a network application that uses UDP (e.g., streaming video, VoIP software, or custom script).

Step 2: Start the application to generate UDP traffic.

Step 3: Start capturing packets in Wireshark while the UDP application is running.

Step 4: After sufficient traffic is generated, stop capturing packets.

Task 2: Filter and analysis UDP Packets

Step 1: In the filter bar, type UDP and press Enter.

Step 2: This filters out only the UDP packets from the capture.

Step 3: Select any UDP packet to view its details.

Step 4: Observe the source and destination ports, length, and data.

Step 5: Compare the simplicity of UDP headers with TCP headers.

The screenshot shows the Wireshark interface with a packet capture of UDP traffic. The packet list at the top shows several UDP packets. Packet 2940 is selected, and its details are expanded in the bottom pane. The details pane shows the User Datagram Protocol (UDP) section with the following information:

- Source Port: 443
- Destination Port: 58661
- Length: 1238
- Checksum: 0xa9c7 [unverified]
- [Checksum Status: Unverified]
- [Stream index: 5]
- [Stream Packet Number: 1196]
- [Timestamps]
- UDP payload (1230 bytes)
- Data (1230 bytes)

The raw data section at the bottom shows the hexadecimal and ASCII representation of the packet data.