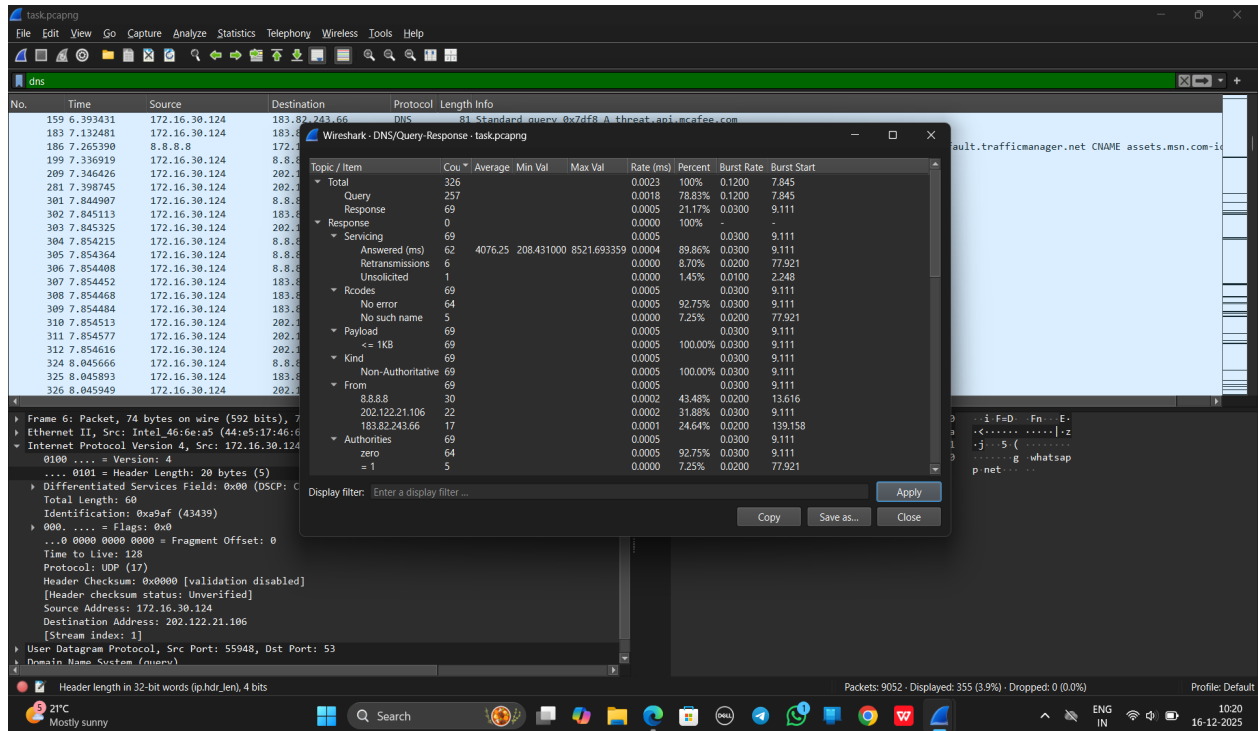


# Wireshark Investigation of Application Layer Protocols

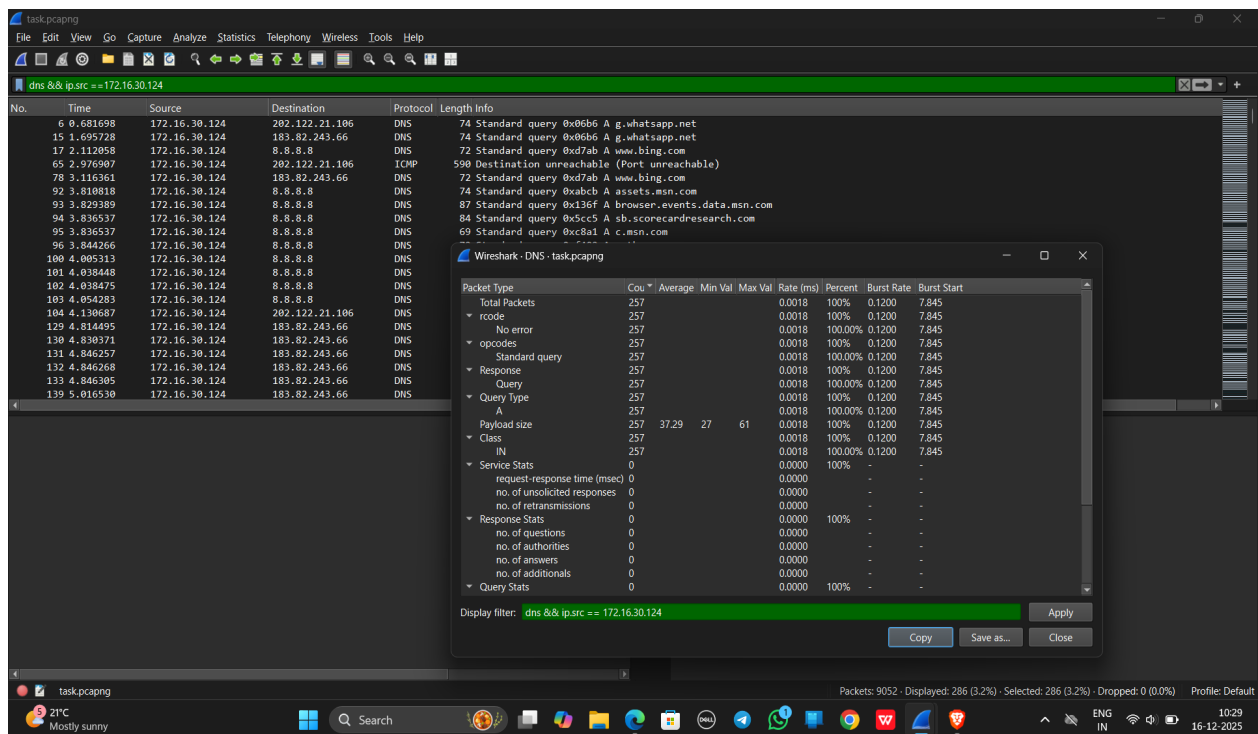
This report presents a detailed Wireshark-based analysis of DNS, TCP, HTTP/1.1, and HTTP/2 protocols while accessing <http://testphp.vulnweb.com/login.php>. Each screenshot below directly corresponds to a specific task or sub-question and serves as primary evidence for the analysis.

# Task 1: DNS, TCP, and HTTP Analysis

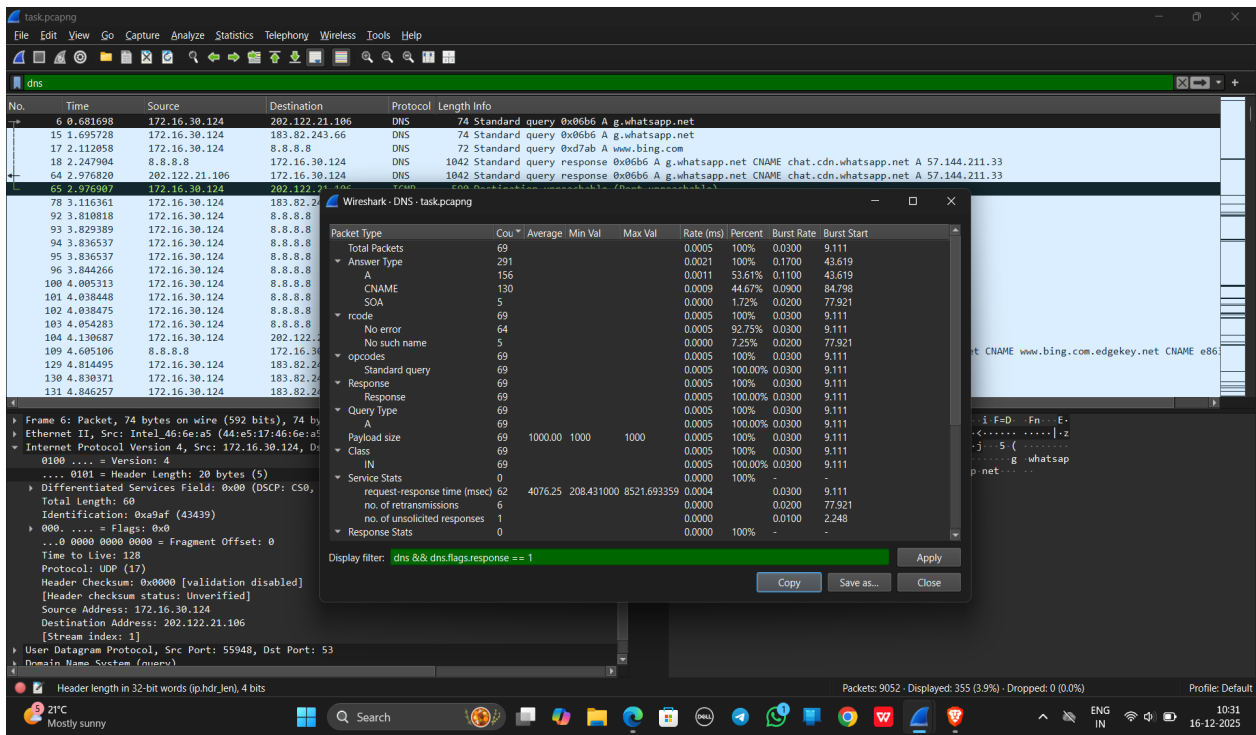
## DNS Resolution Process



DNS query packets generated by the browser while resolving testphp.vulnweb.com.

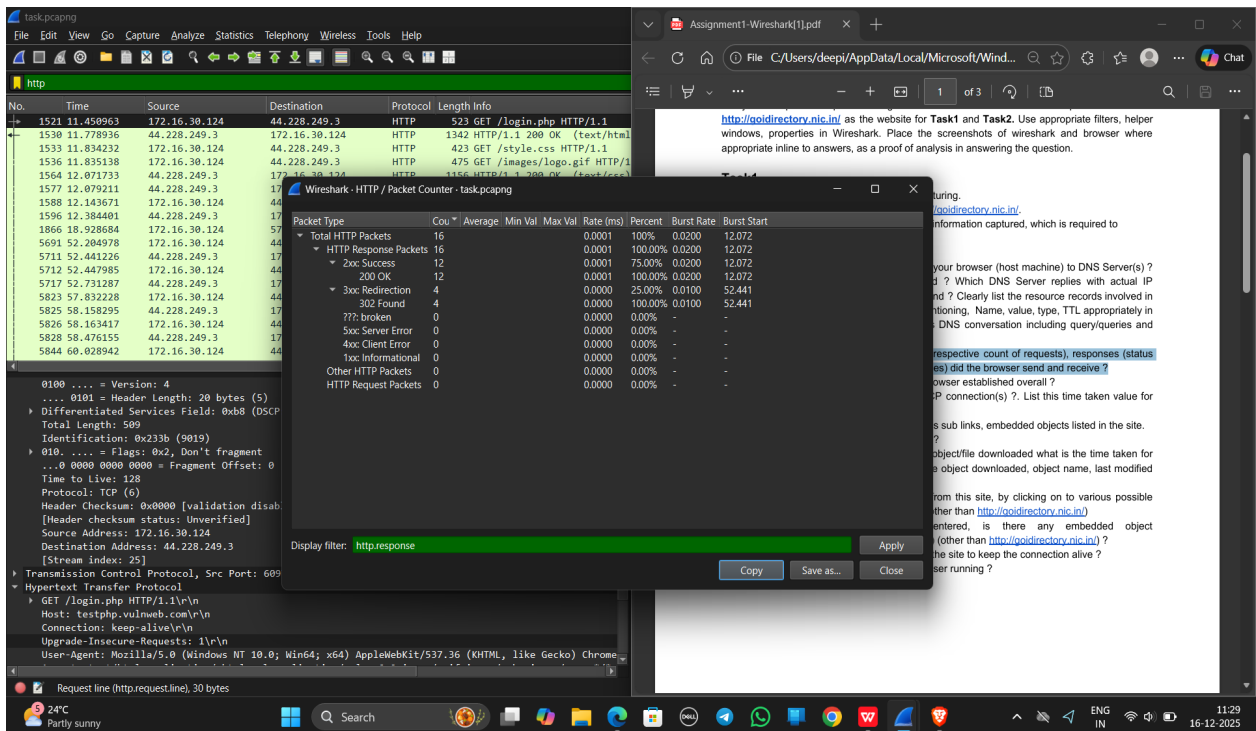


DNS response packets showing resolved IP address records returned by DNS servers.

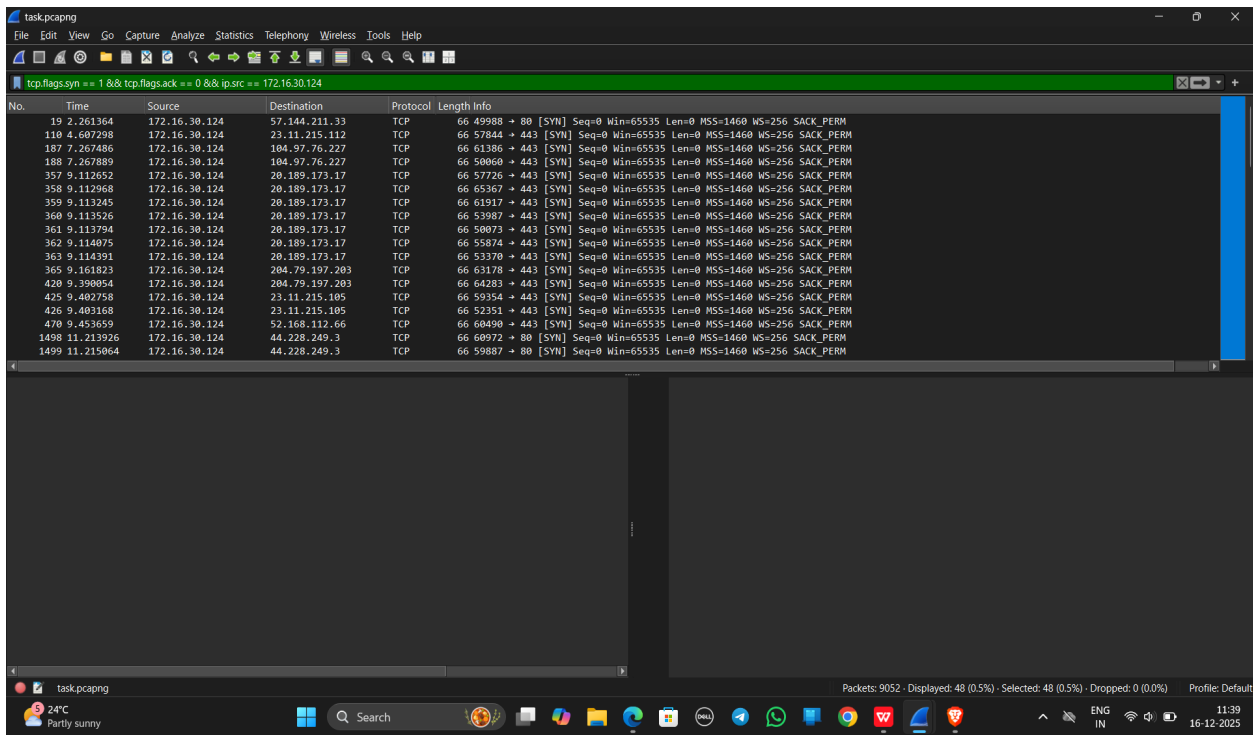


Expanded DNS answer section displaying resource record type, value, and TTL.

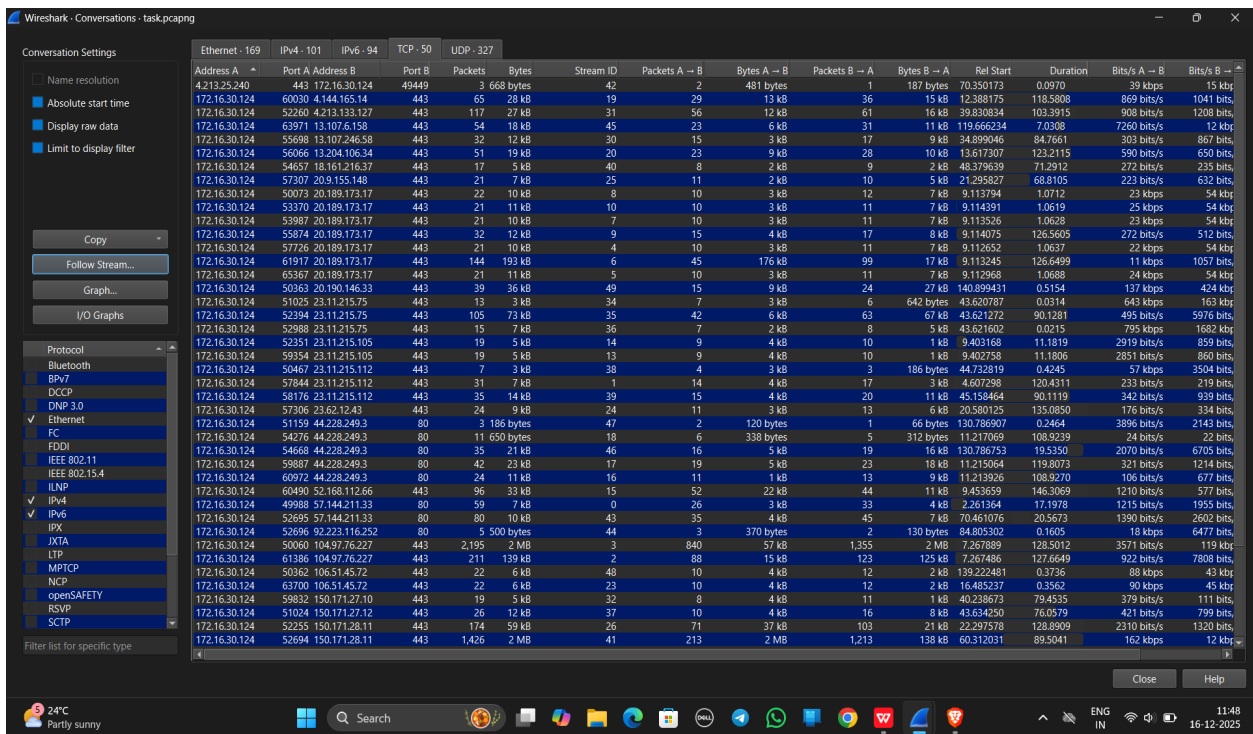
## HTTP Requests and Responses



HTTP GET requests sent by the browser for login.php and embedded objects.

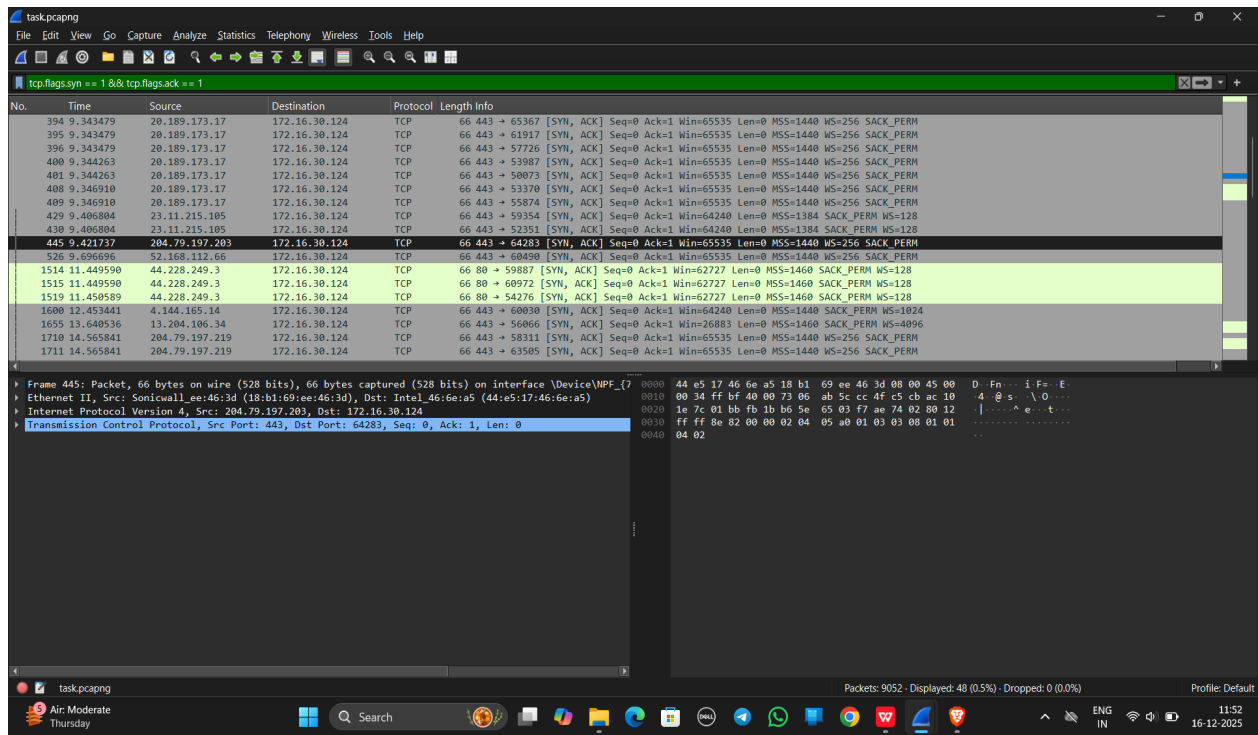


HTTP response packets indicating successful and redirected responses from the server.

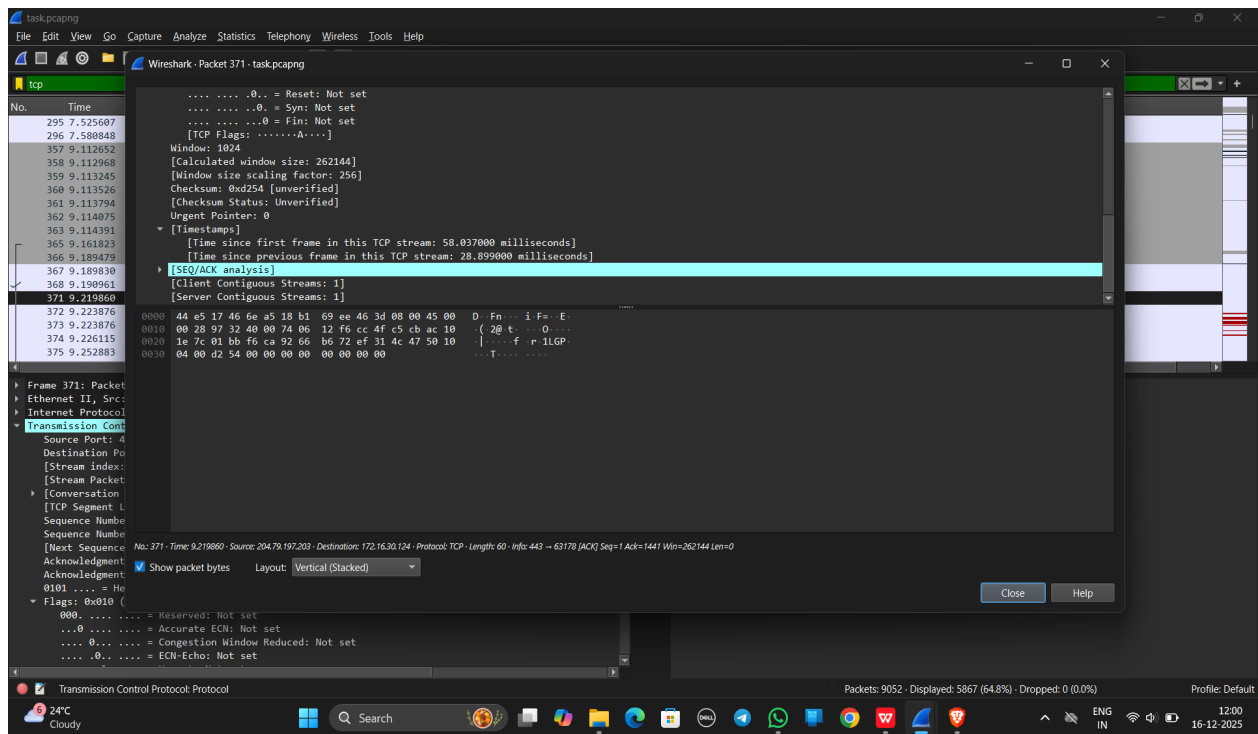


HTTP statistics window summarizing request types and response codes.

## TCP Connections and Timing



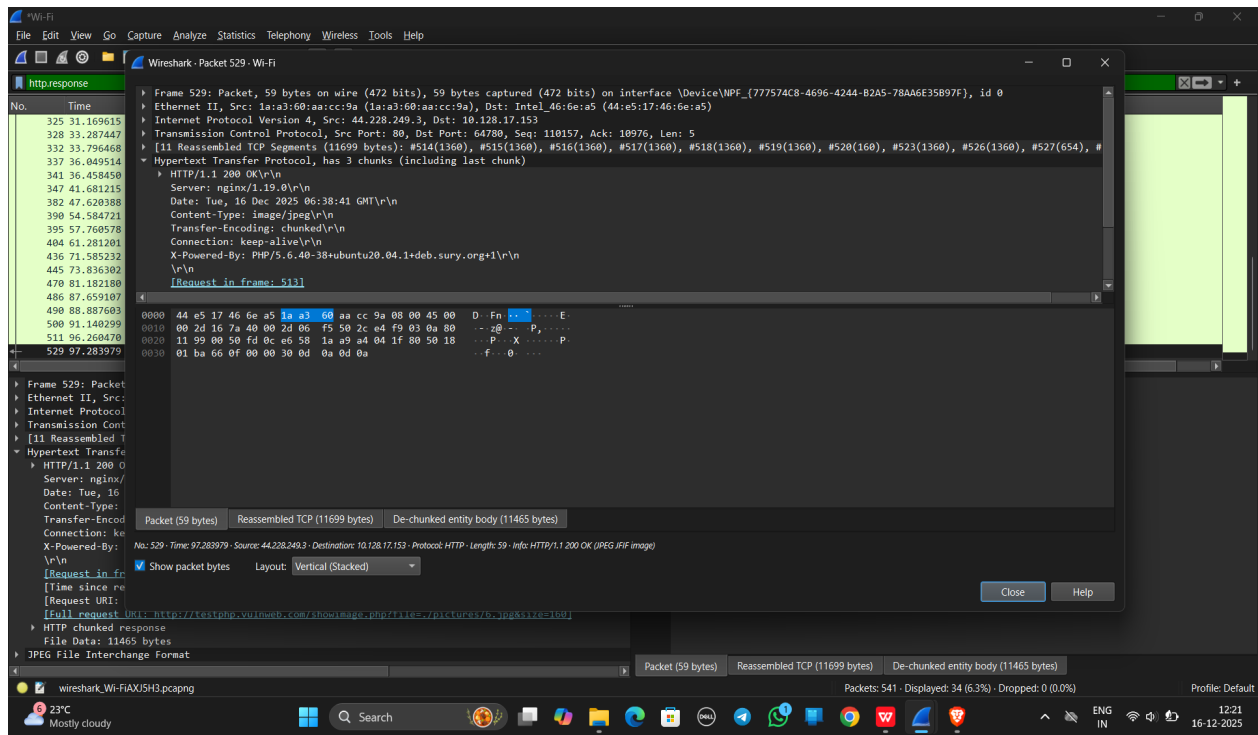
TCP SYN packets initiating new connections between client and web server.



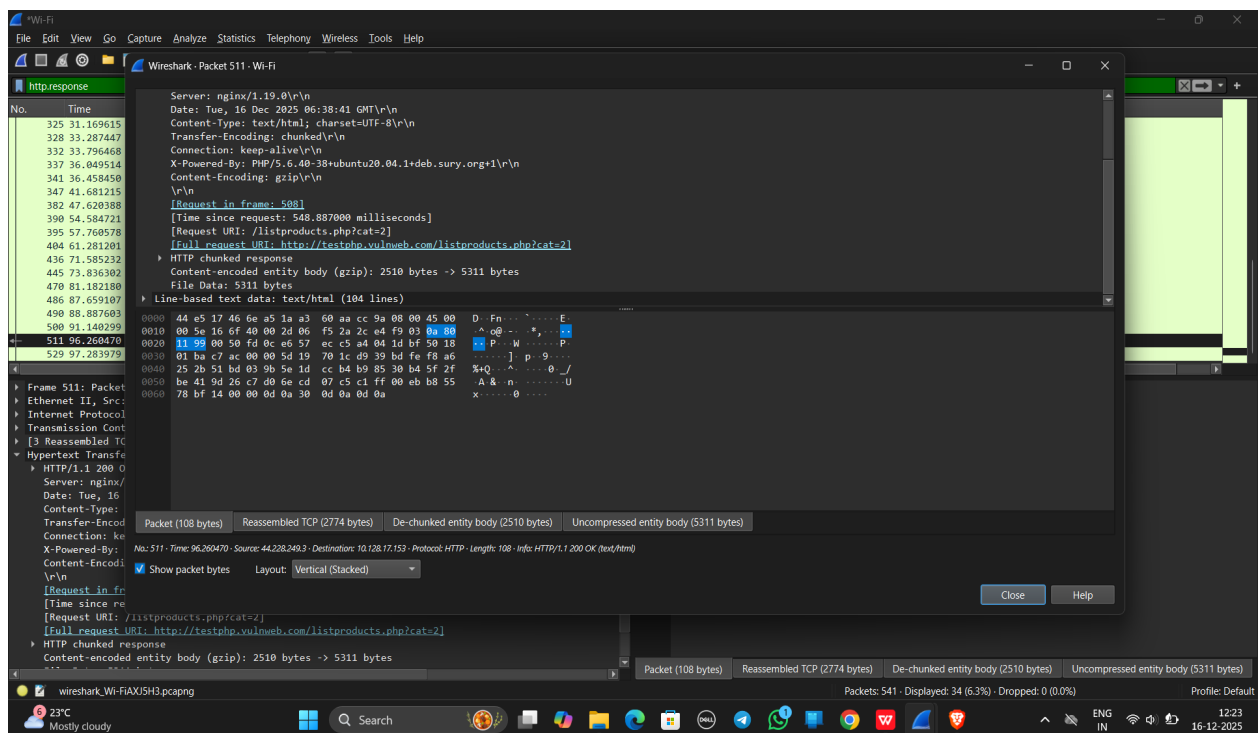
TCP three-way handshake completion confirming successful connection establishment.







HTTP response headers showing Content-Length and Last-Modified fields for downloaded objects.



Persistent TCP connection usage demonstrating HTTP keep-alive behavior.

# Task 2: Browser Cache and Conditional GET Analysis

Wireshark packet capture showing an initial HTTP GET request for /userinfo.php. The packet list shows a GET request from 172.16.30.124 to 172.16.30.124. The packet details show the request line: GET /userinfo.php HTTP/1.1. The packet bytes show the raw data of the request.

Initial HTTP request after clearing browser cache, resulting in full object download.

Wireshark packet capture showing a subsequent HTTP GET request for /userinfo.php. The packet list shows a GET request from 172.16.30.124 to 172.16.30.124. The packet details show the request line: GET /userinfo.php HTTP/1.1. The packet bytes show the raw data of the request. A Wireshark - HTTP / Packet Counter - Wi-Fi window is open, showing statistics for the captured packets.

Subsequent HTTP request containing conditional headers such as If-Modified-Since.



The image shows a Wireshark packet capture window titled 'http2-h2c.pcap'. The packet list pane displays two packets:

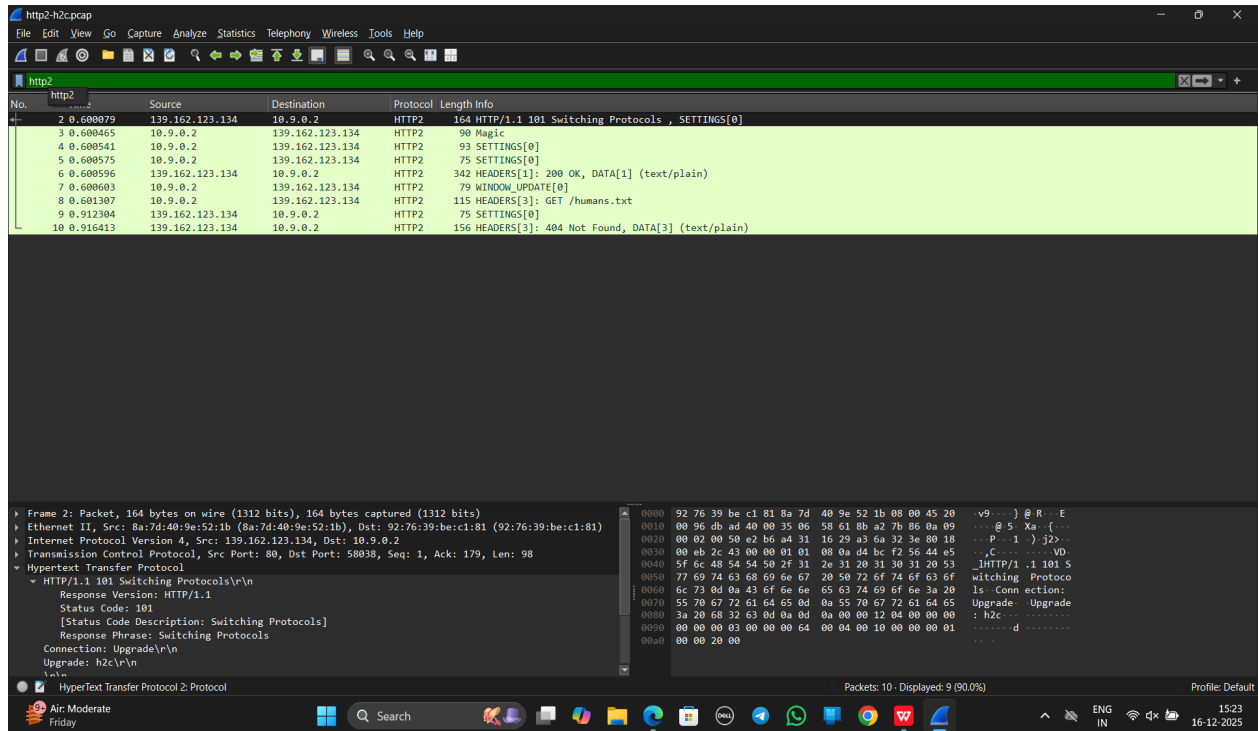
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.9.0.2	139.162.123.134	HTTP	244	GET /robots.txt HTTP/1.1
2	0.600079	139.162.123.134	10.9.0.2	HTTP2	164	HTTP/1.1 304 Switching Protocols , SETTINGS[0]

The packet details pane shows the 'Hypertext Transfer Protocol: Protocol' section. The status bar at the bottom indicates 'Packets: 10 · Displayed: 2 (20.0%)' and 'Profile: Default'. The system tray shows the date and time as '15:23 16-12-2025'.

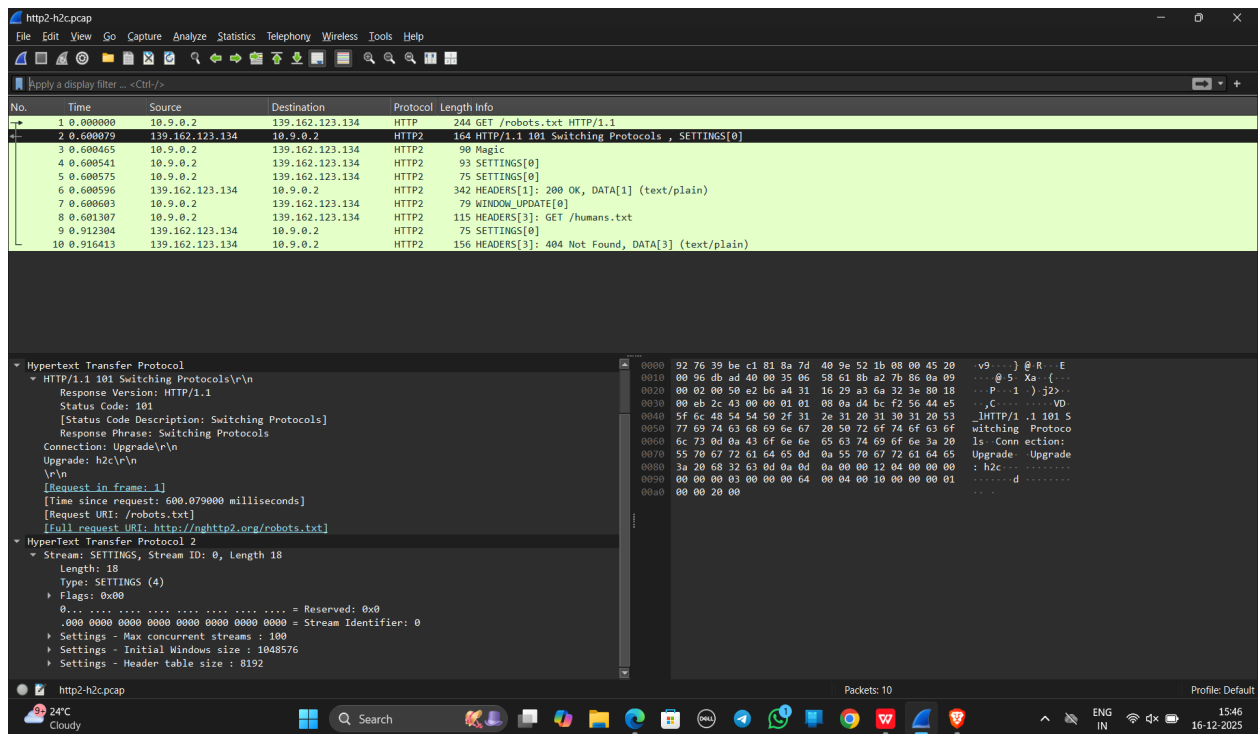
Server response indicating cache validation behavior during repeated access.

The difference between first and subsequent requests is governed by HTTP cache-control mechanisms. Conditional GET requests allow the server to respond with reduced data or status codes such as 304 Not Modified, improving performance.

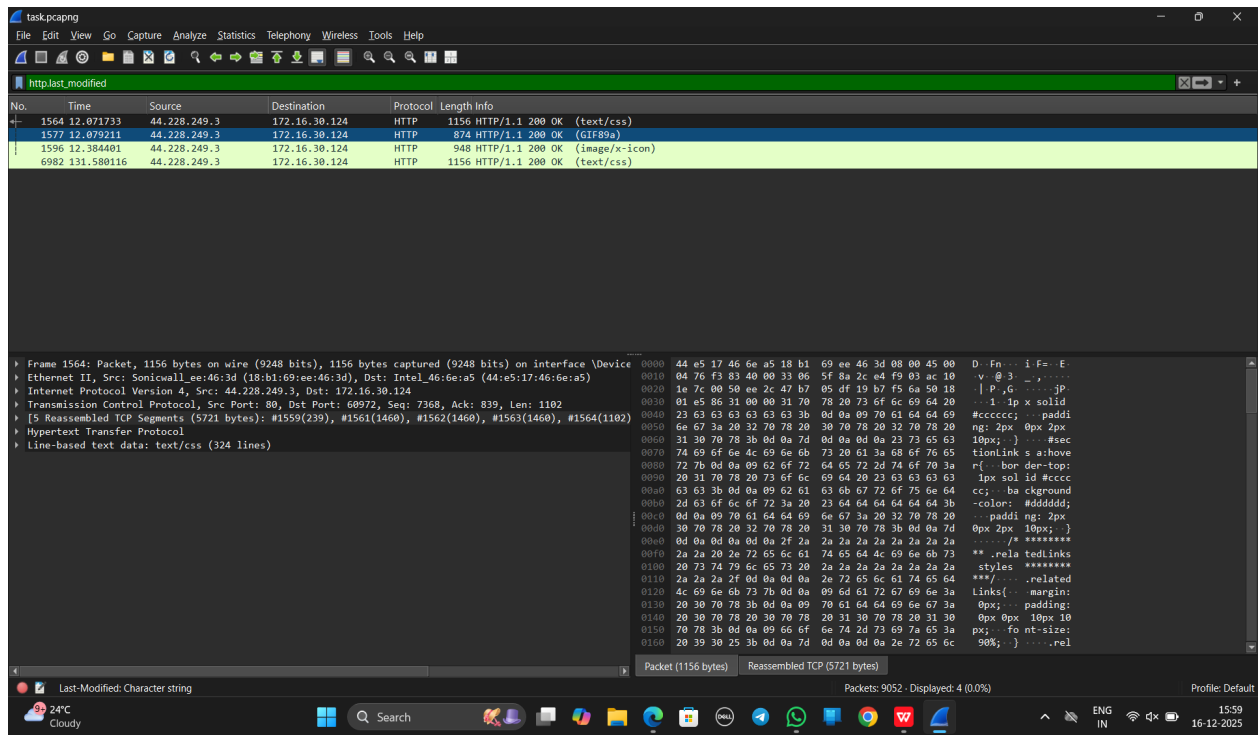
# Task 3: HTTP/2 Protocol Analysis



HTTP/1.1 to HTTP/2 protocol upgrade using 101 Switching Protocols (h2c).



HTTP/2 SETTINGS and HEADERS frames exchanged between client and server.



HTTP/2 GET request and response demonstrating multiplexed data transfer.

HTTP/2 differs from HTTP/1.1 by using binary framing, header compression, and multiplexed streams, resulting in improved efficiency and reduced latency.