CS 315: Computer Networks Lab
Spring 2024-25, IIT Dharwad
Assignment-3
Wireshark Lab: HTTP
January 26, 2025

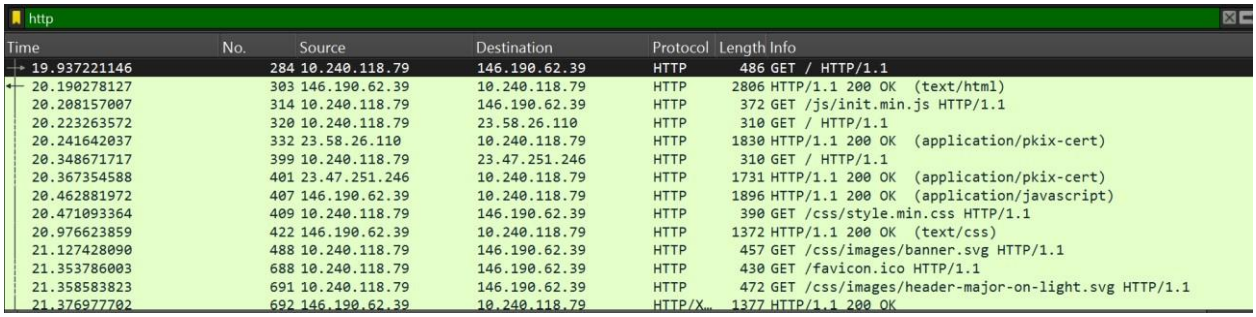
Part-1: The Basic HTTP GET/response interaction

Answer the following questions based on the information you observed when tracing the above HTTP requests and responses.

Enter these links on your browser one after another:

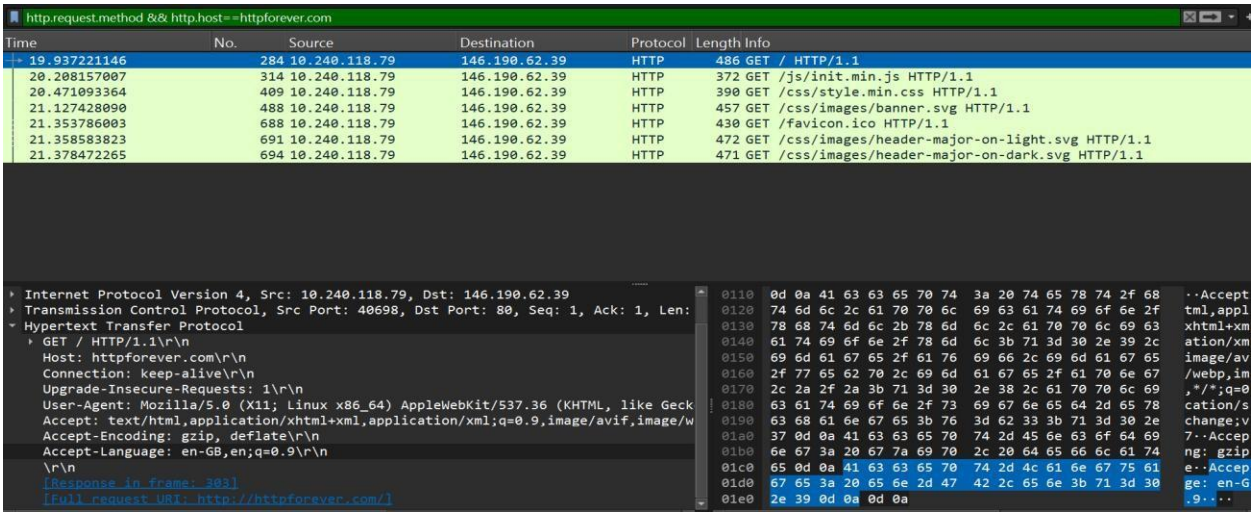
- http://httpforever.com/
- http://web.simmons.edu/
- http://www.vulnweb.com/
- http://www.testingmcafeesites.com/

1. **What type of HTTP version do you observe in the above trace?**

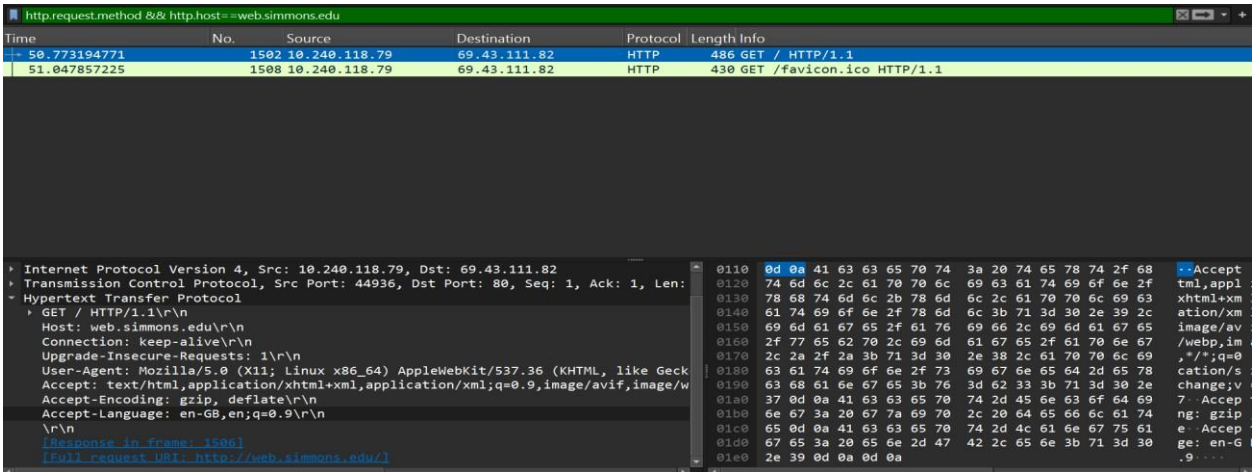


The HTTP version observed in the trace is **HTTP/1.1**.This can be deduced from the snippet: GET / HTTP/1.1, where the HTTP/1.1 explicitly states the version of the HTTP protocol being used for this request.

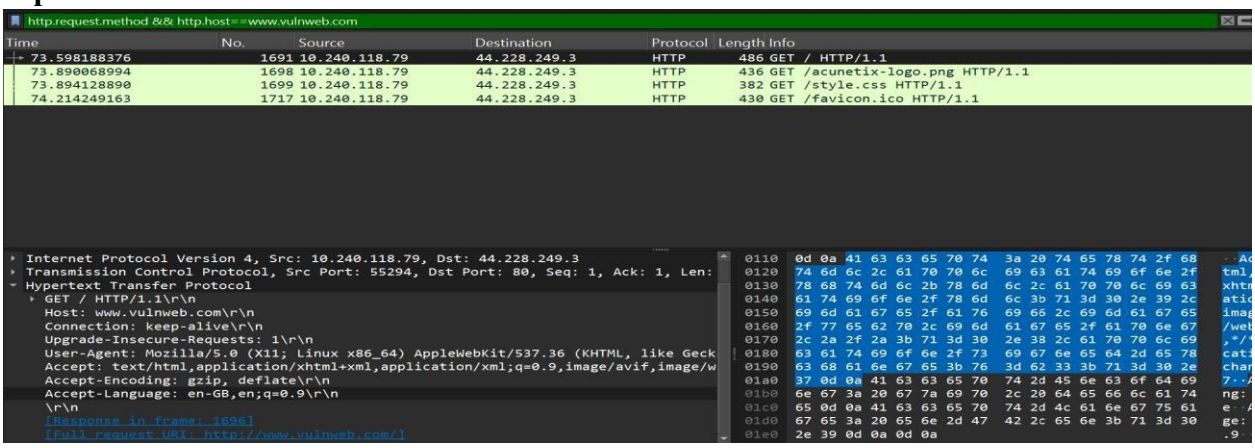
2. **Mention the HTTP request method used to request these four websites.**



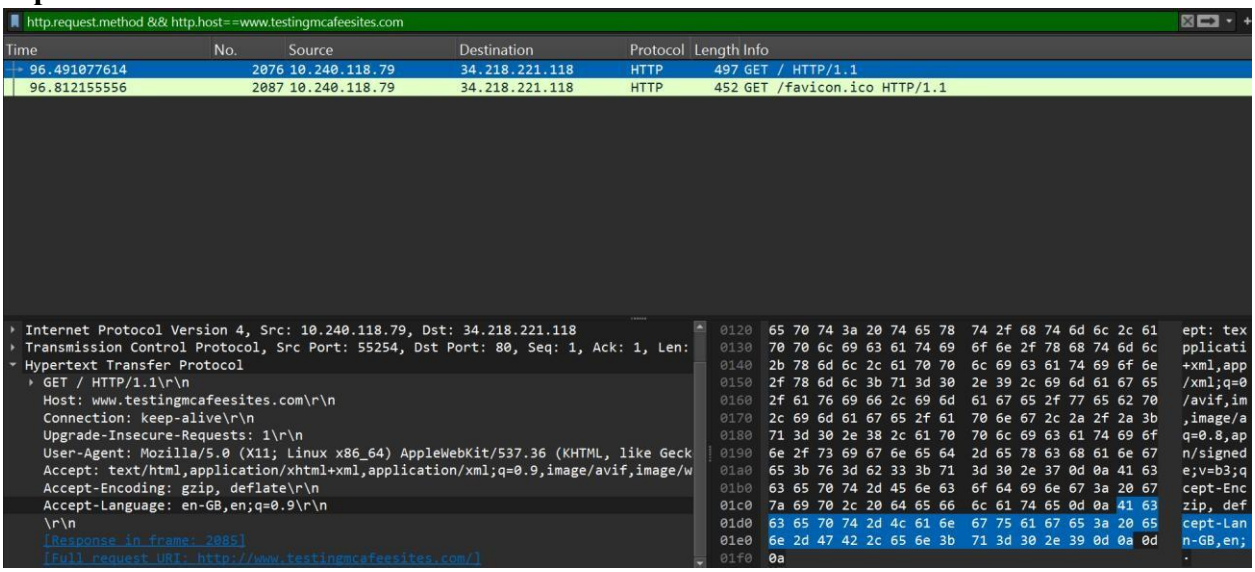
In the provided trace (GET / HTTP/1.1), the HTTP request method is **GET** for **http://httpforever.com/.**



In the provided trace (GET / HTTP/1.1), the HTTP request method is **GET** for **http://web.simmons.edu/.**

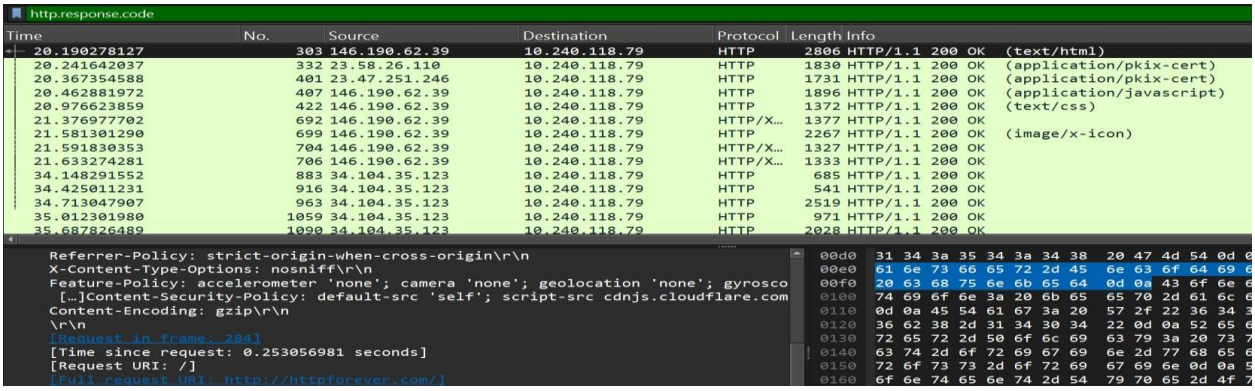


In the provided trace (GET / HTTP/1.1), the HTTP request method is **GET** for **http://www.vulnweb.com/.**

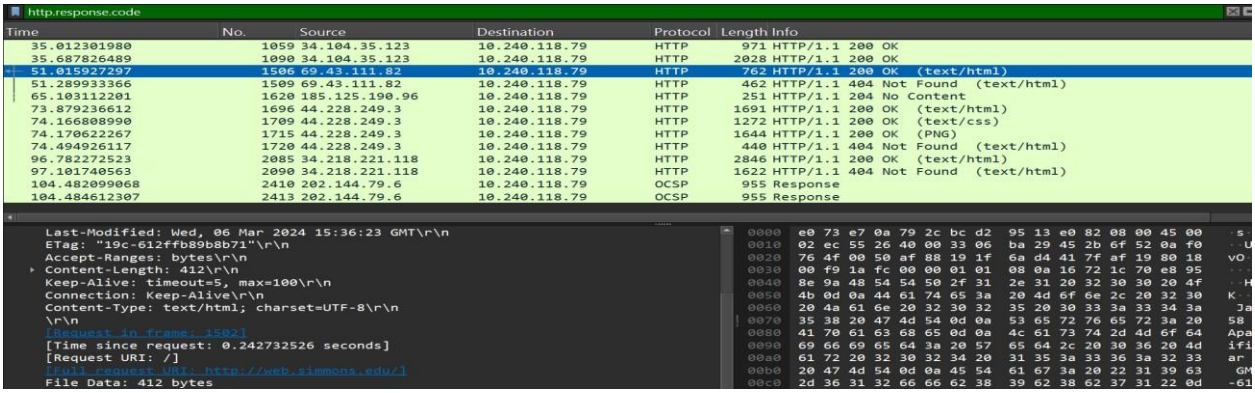


In the provided trace (GET / HTTP/1.1), the HTTP request method is **GET** for **http://www.testingmcafeesites.com/.**

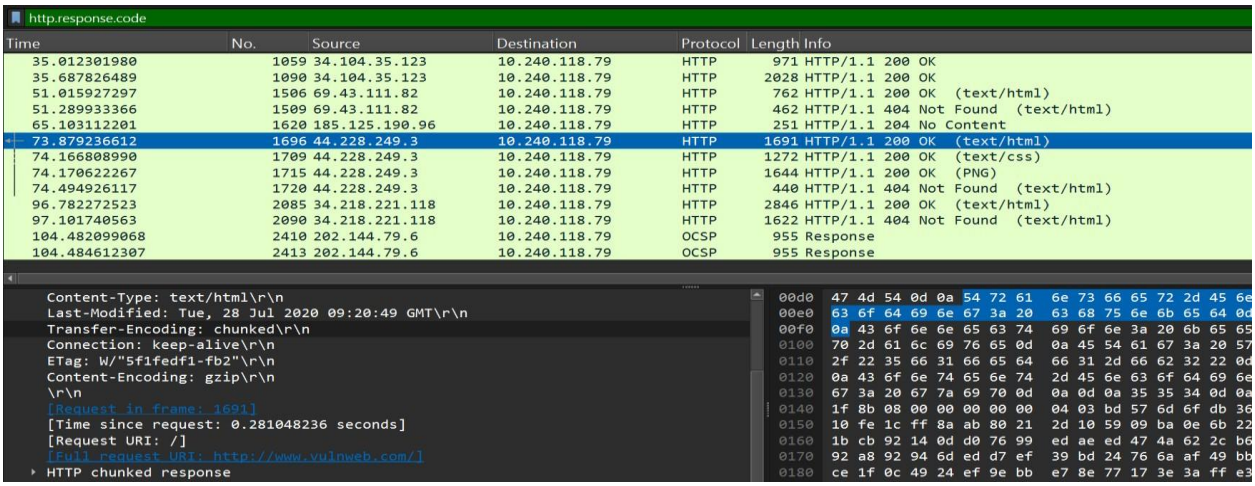
3. **What are the status codes returned by the server to the browser?**



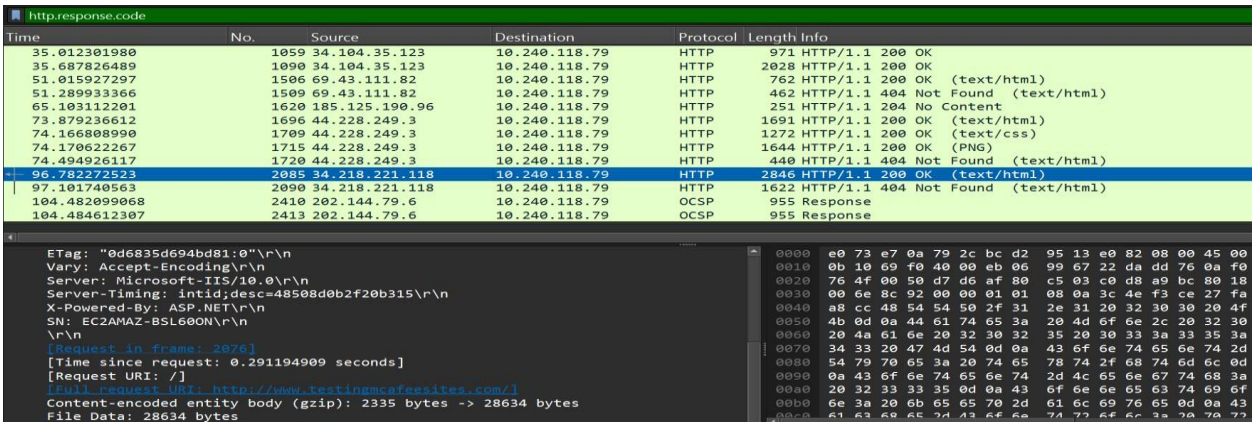
In the provided trace (HTTP/1.1 200 OK) for **http://httpforever.com**/, the status codes returned by the server to the browser is **200 OK** signifies that the server successfully processed the request and returned the requested resource.



In the provided trace (HTTP/1.1 200 OK) for **http://web.simmons.edu**/, the status codes returned by the server to the browser is **200 OK** signifies that the server successfully processed the request and returned the requested resource.

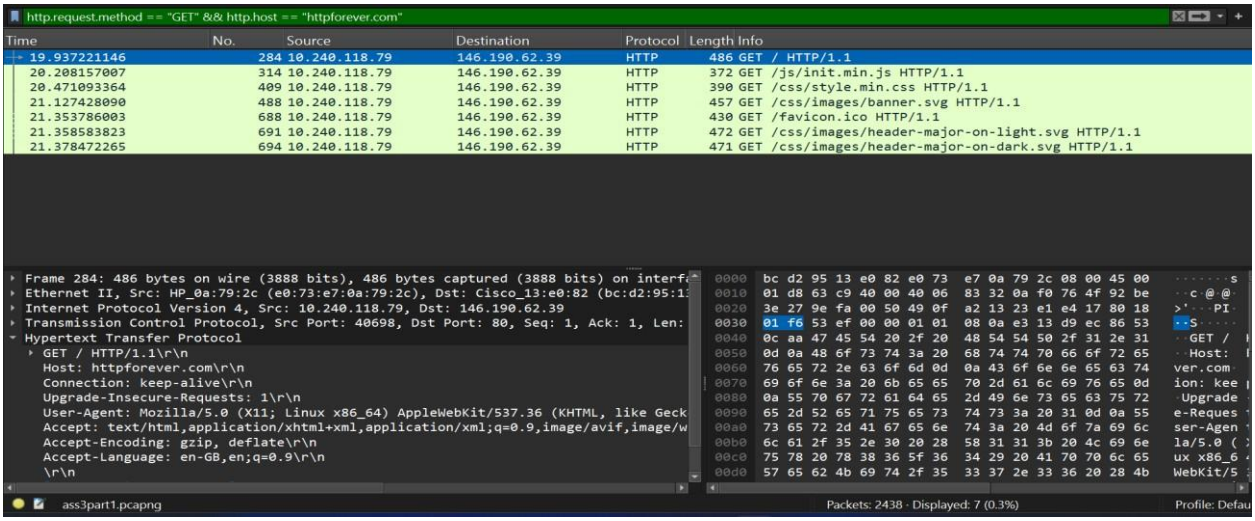


In the provided trace (HTTP/1.1 200 OK) for **http://www.vulnweb.com**/, the status codes returned by the server to the browser is **200 OK** signifies that the server successfully processed the request and returned the requested resource.



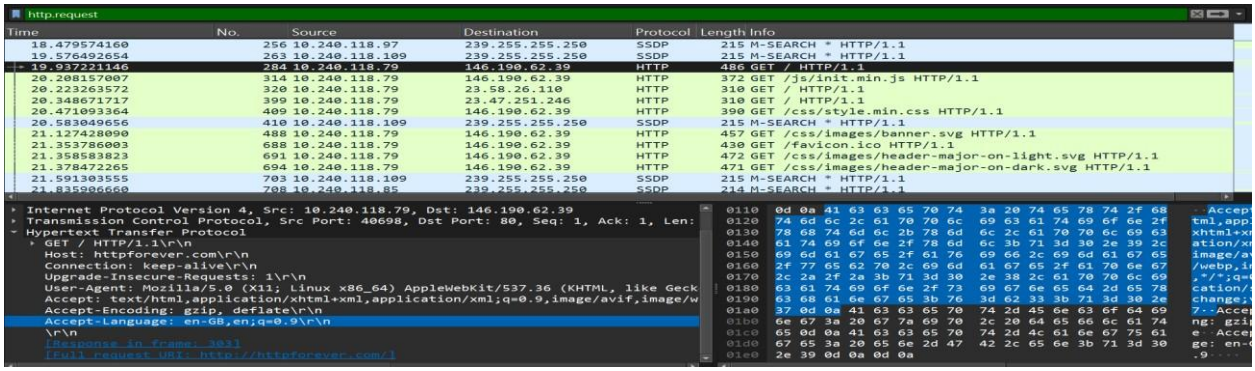
In the provided trace (HTTP/1.1 200 OK) for **http://www.testingmcafeesites.com/**, the status codes returned by the server to the browser is **200 OK** signifies that the server successfully processed the request and returned the requested resource.

4. **List the number of HTTP GET requests and frame number for "http://httpforever.com/" request.**



- Number of HTTP GET Requests: There are 7 GET requests to **http://httpforever.com/.**
- Frame Numbers for Each Request: Frame 284, Frame 314, Frame 409, Frame 488, Frame 688, Frame 691 and Frame 694.

5. **What languages does your browser indicate that it can accept to the server?**

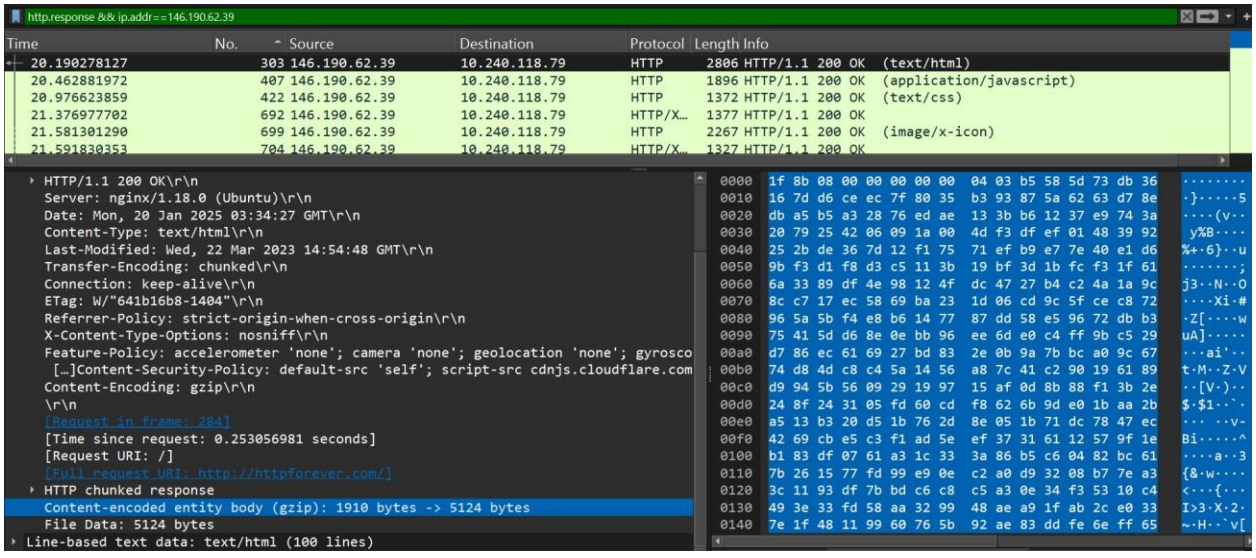


Accept-Language: en-GB,en;q=0.9\r\n

6. **List the source and the destination IP address details for all the above HTTP requests.**
   Using question 2 images of 4 websites we can fill this-

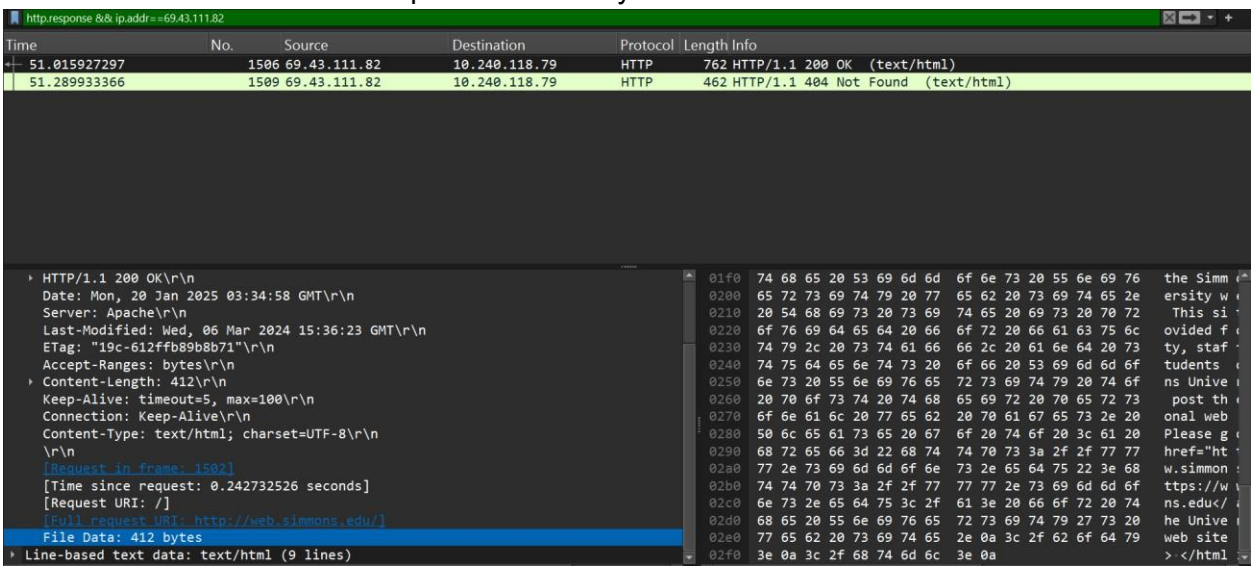
Domain	Source IP	Destination IP
http://httpforever.com/	10.240.118.79	146.190.62.39
http://web.simmons.edu/	10.240.118.79	69.43.111.82
http://www.vulnweb.com/	10.240.118.79	44.228.249.3
http://www.testingmcafeesites.com/	10.240.118.79	34.218.221.118

7. **By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.**
   No, the raw data displayed in the packet content window corresponds exactly to what is shown in the packet-listing window. There are no additional headers visible in the packet content window that are not already displayed in the packet-listing window.
8. **For each url requested, provide the number of bytes returned by the server corresponding to the first HTTP GET request.**

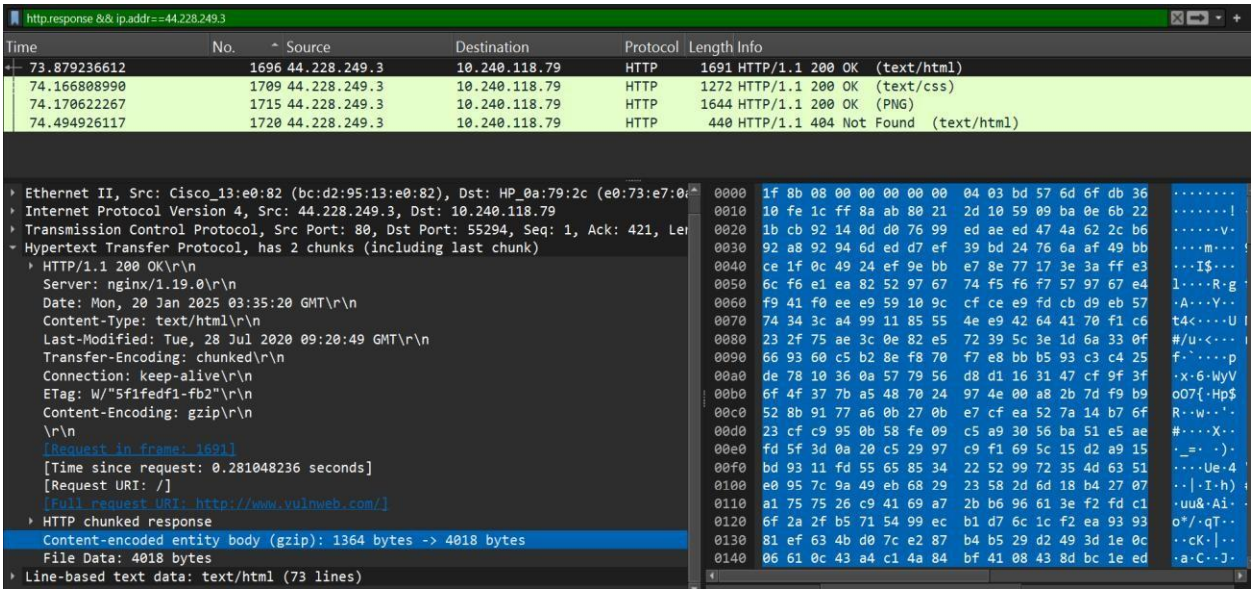


for **http://httpforever.com/** , number of bytes returned by the server corresponding to the first HTTP GET request 1910 bytes, which is the size of the compressed content (gzip).This is the amount of data transmitted over the network in response to the HTTP GET request. The decompressed size (5124 bytes) represents the expanded size of the content after it is processed

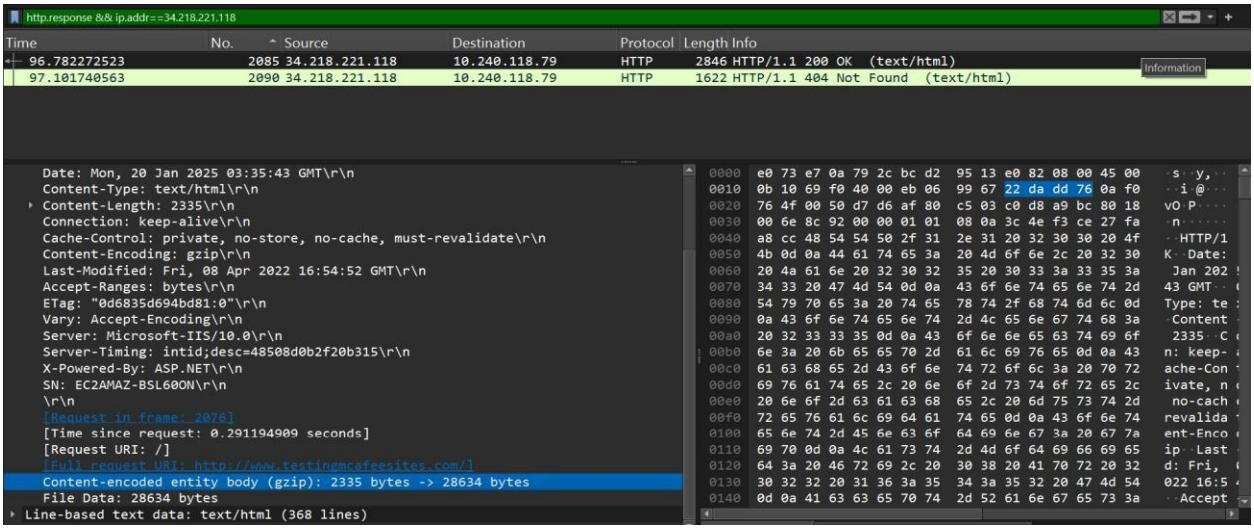
by the client, but it is not what the server physically transmitted. Therefore, the correct value to consider for the server's response is 1910 bytes.



for **http://web.simmons.edu/**, number of bytes returned by the server corresponding to the first HTTP GET request = 412 bytes.

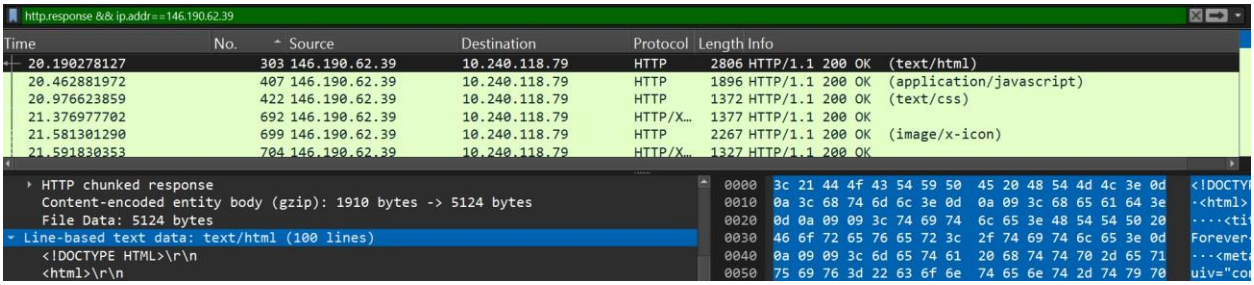


for **http://www.vulnweb.com/,** number of bytes returned by the server corresponding to the first HTTP GET request 1364 bytes, which is the size of the compressed content (gzip).This is the amount of data transmitted over the network in response to the HTTP GET request. The decompressed size (4018 bytes) represents the expanded size of the content after it is processed by the client, but it is not what the server physically transmitted. Therefore, the correct value to consider for the server's response is 1364 bytes.

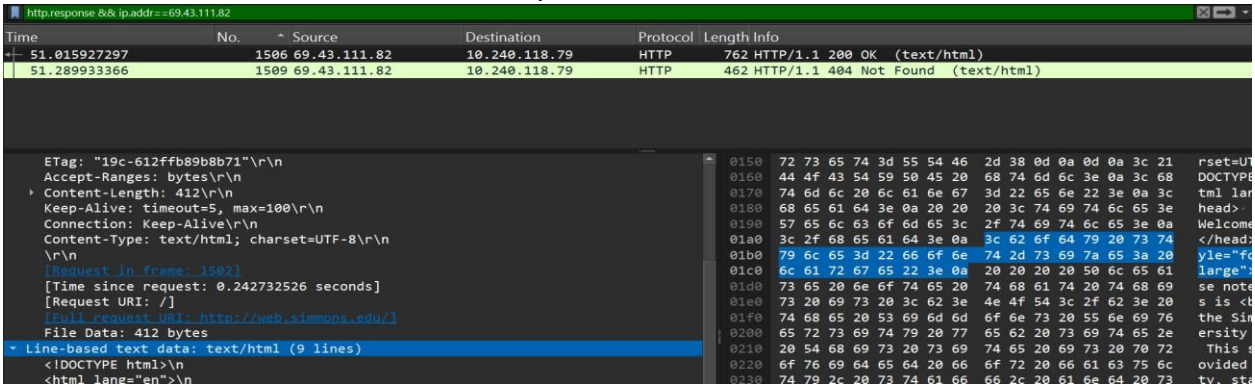


for **http://www.testingmcafeesites.com/**, number of bytes returned by the server corresponding to the first HTTP GET request 2335 bytes, which is the size of the compressed content (gzip). This is the amount of data transmitted over the network in response to the HTTP GET request. The decompressed size (28634 bytes) represents the expanded size of the content after it is processed by the client, but it is not what the server physically transmitted. Therefore, the correct value to consider for the server's response is 2335 bytes.

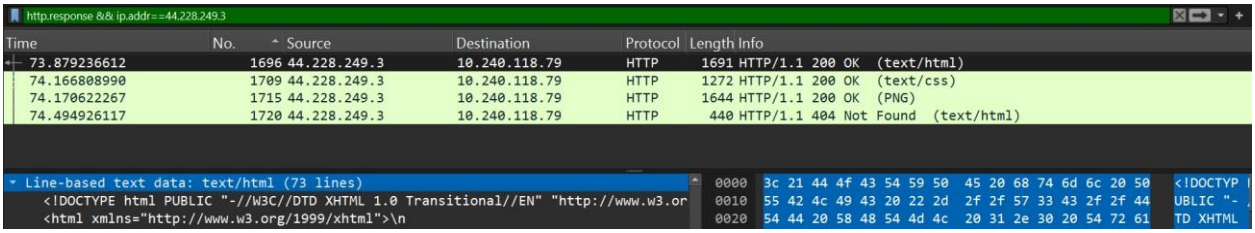
9. **Obtain the number of lines of data and the content being received by your browser for the above first HTTP GET requests.**



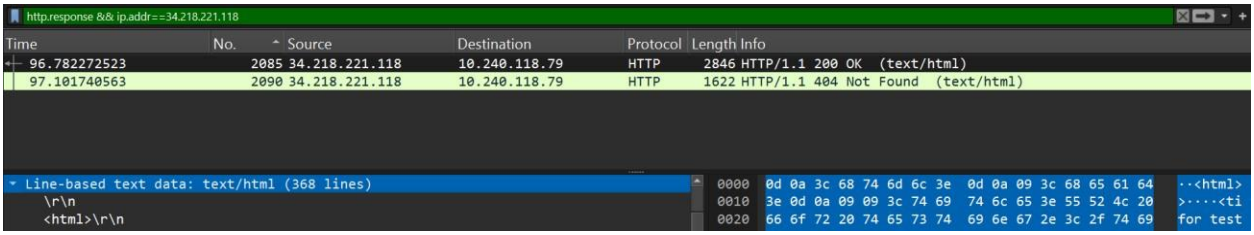
for **http://httpforever.com/,** number of lines of data and the content being received by your browser for the above first HTTP GET requests= 100 lines.



for **http://web.simmons.edu/,** number of lines of data and the content being received by your browser for the above first HTTP GET requests= 9 lines.



for **http://www.vulnweb.com/,** number of lines of data and the content being received by your browser for the above first HTTP GET requests= 73 lines.



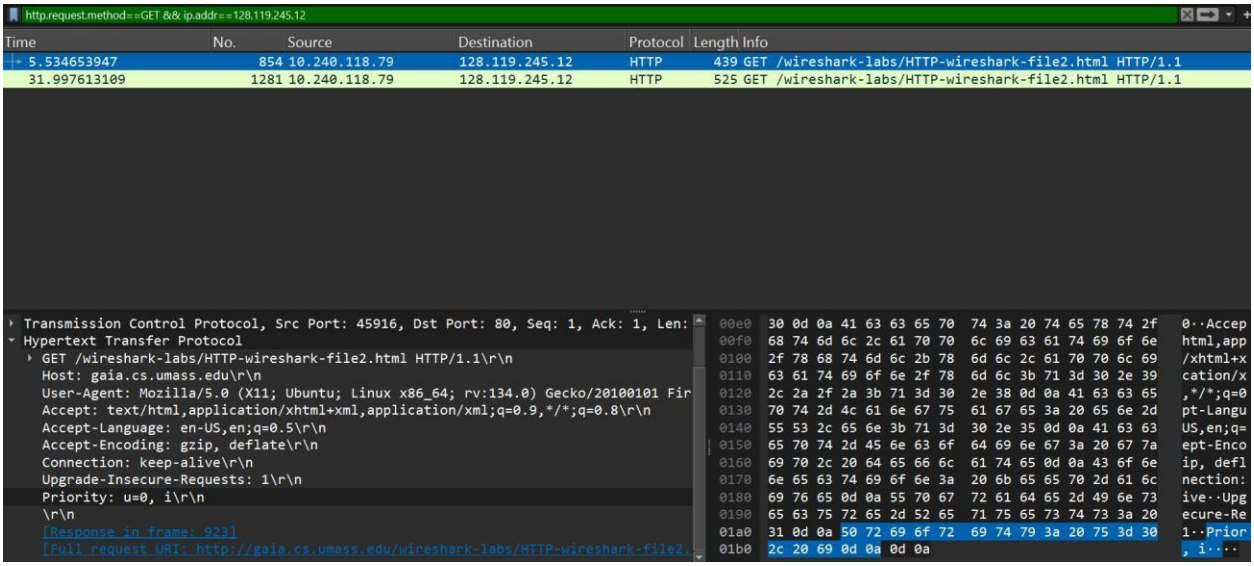
for **http://www.testingmcafeesites.com/,** number of lines of data and the content being received by your browser for the above first HTTP GET requests= 368 lines.

Part 2: The HTTP CONDITIONAL GET/response interaction

- **Enter the following URL into your browser http://gaia.cs.umass.edu/wiresharklabs/HTTP-wireshark-file2.html Your browser should display a very simple five-line HTML file.**

Answer the following questions:

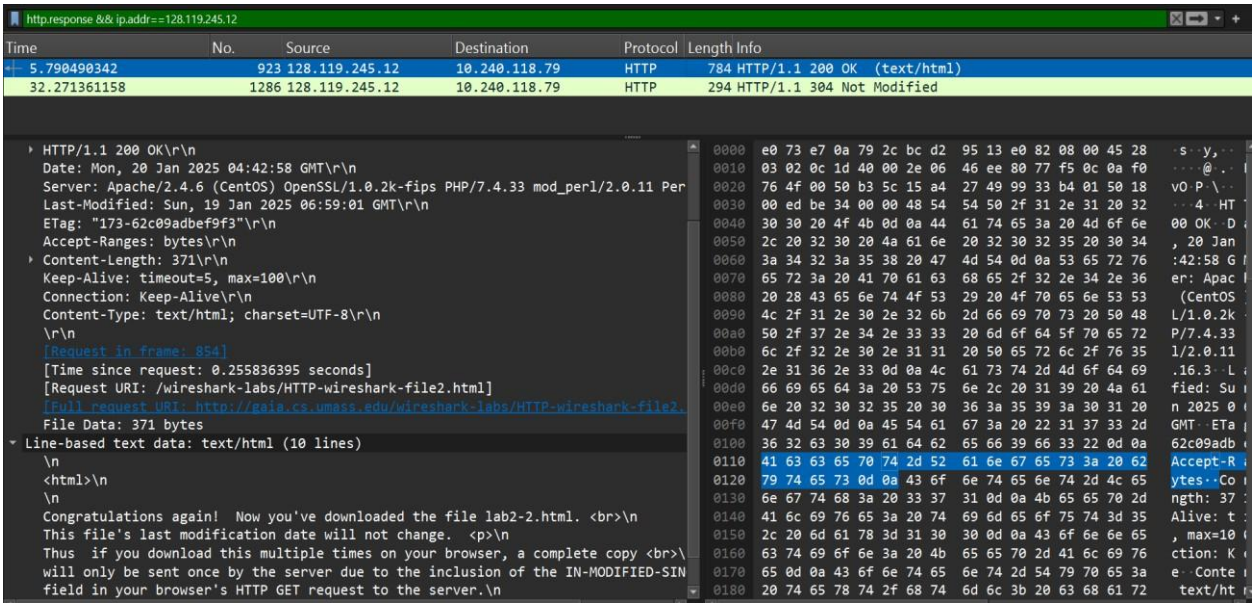
1. **Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?**



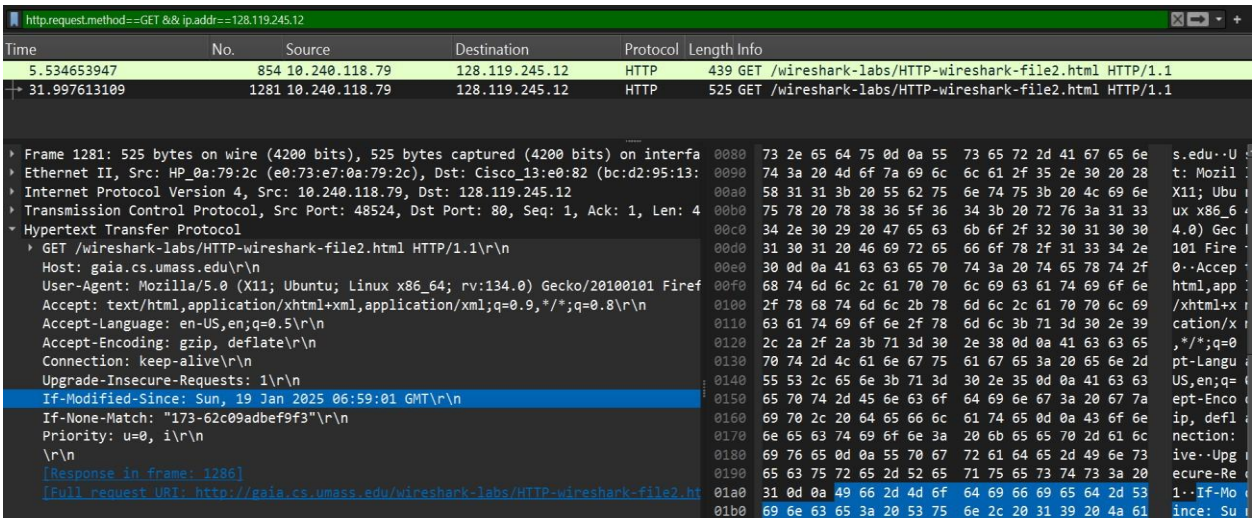
The **"IF-MODIFIED-SINCE"** line is not seen inside Hypertext Transfer Protocol, for the first HTTP GET request, indicating the browser has no cached copy yet.

2. **Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?**

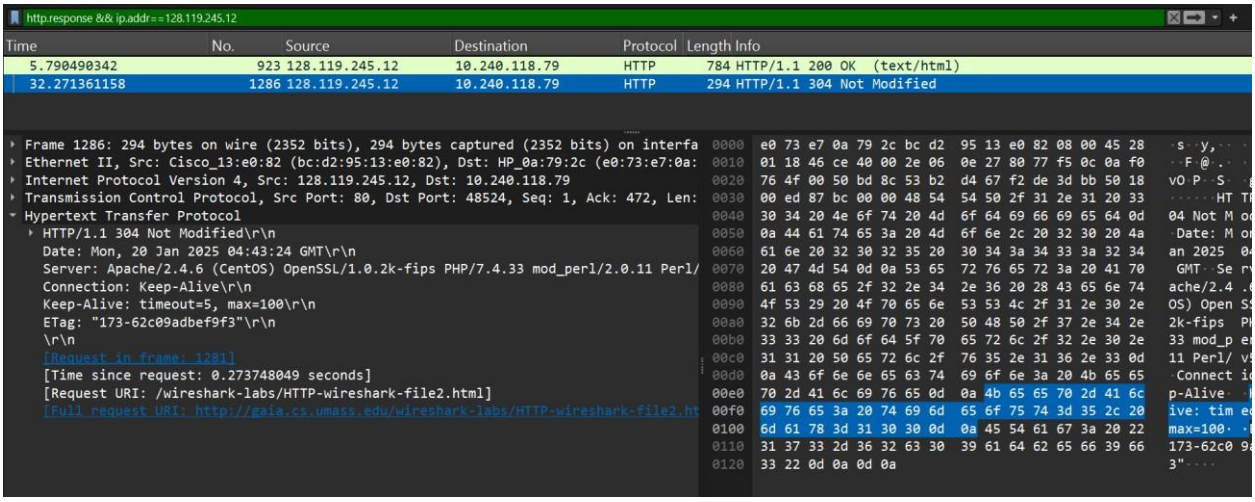
For the first HTTP GET request, the server explicitly returned the file contents because the response had a **200 OK** status and included the file's text data. This confirms the file was sent by the server.



3. **Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header?**



In the second HTTP GET request, the **IF-MODIFIED-SINCE** header is present inside Hypertext Transfer Protocol, indicating the timestamp when the resource was last downloaded by the browser. This header helps the server determine whether the resource has changed since the specified time.

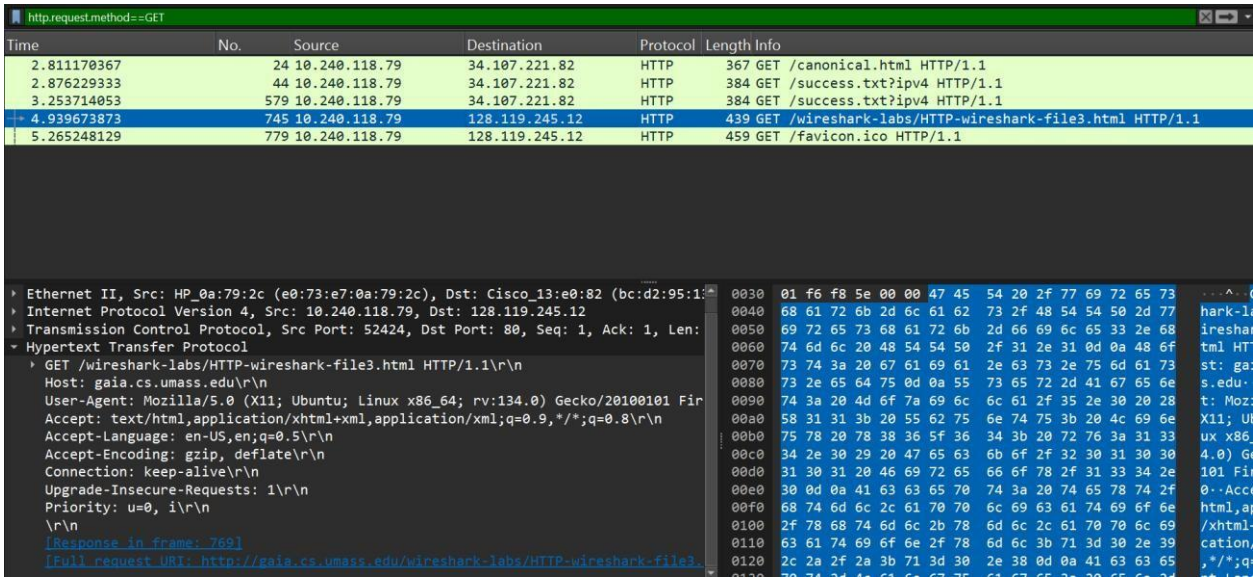


If the resource is unchanged, the server responds with a **304 Not Modified** status, signaling the browser to use the cached version without downloading it again. This mechanism improves performance by reducing unnecessary data transfer and saving bandwidth. The **IF-MODIFIED-SINCE** header typically appears when refreshing or reentering a URL.

4. **What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.**
From above image, the HTTP status code for the **second GET response** is **304**, and the phrase is **Not Modified**. The server did not explicitly return the file contents, as this status indicates the resource has not changed since the client last requested it. Instead, the browser uses the cached version of the resource. This mechanism reduces bandwidth usage and improves performance by avoiding redundant downloads.

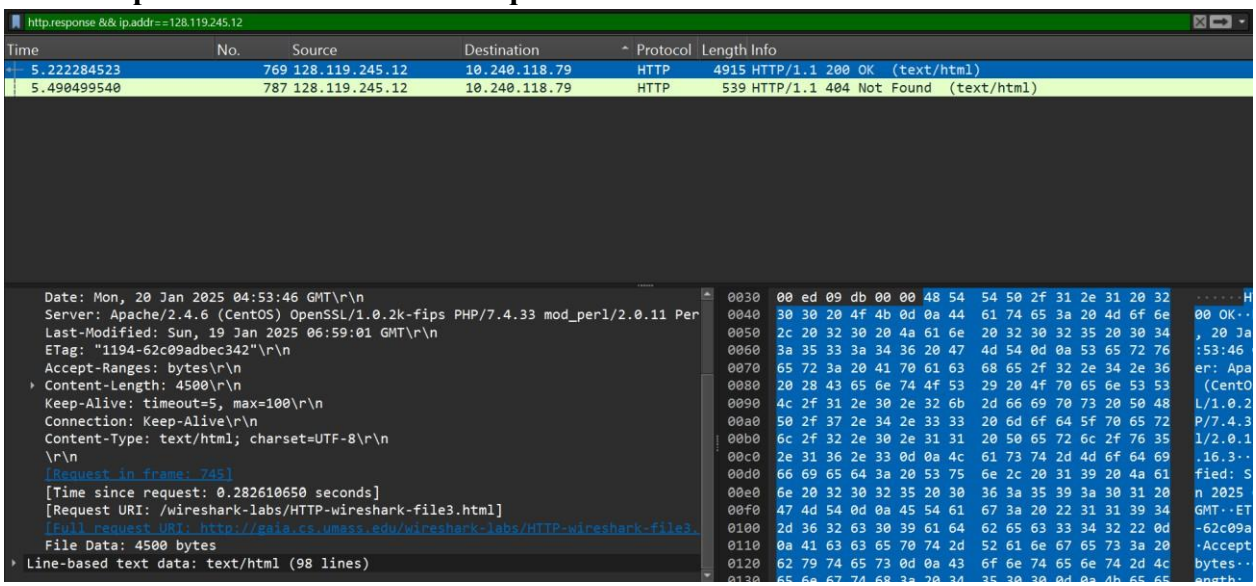
Part 3: Retrieving Long Documents

1. **How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill of Rights?**



My browser sent **5** GET request messages and **745th** packet contains the GET request for the Bill of Rights.

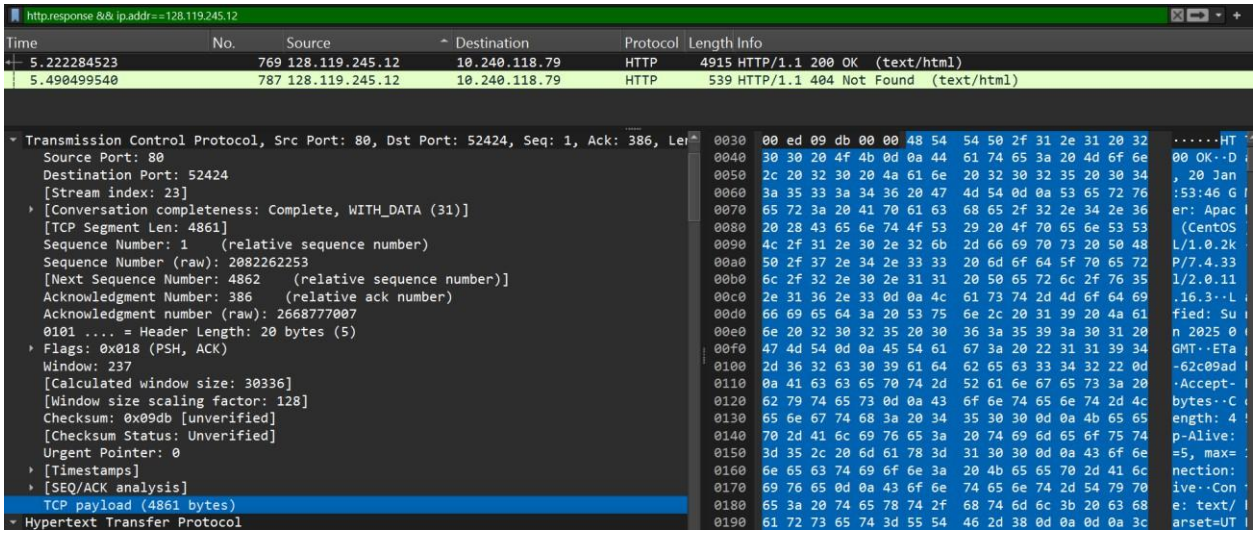
2. **Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?**



769th packet contains the status code and phrase associated with the response to the HTTP GET request which is **200 OK.**

3. **What is the status code and phrase in the response?**
   The status code is **200** and the phrase is **OK**.

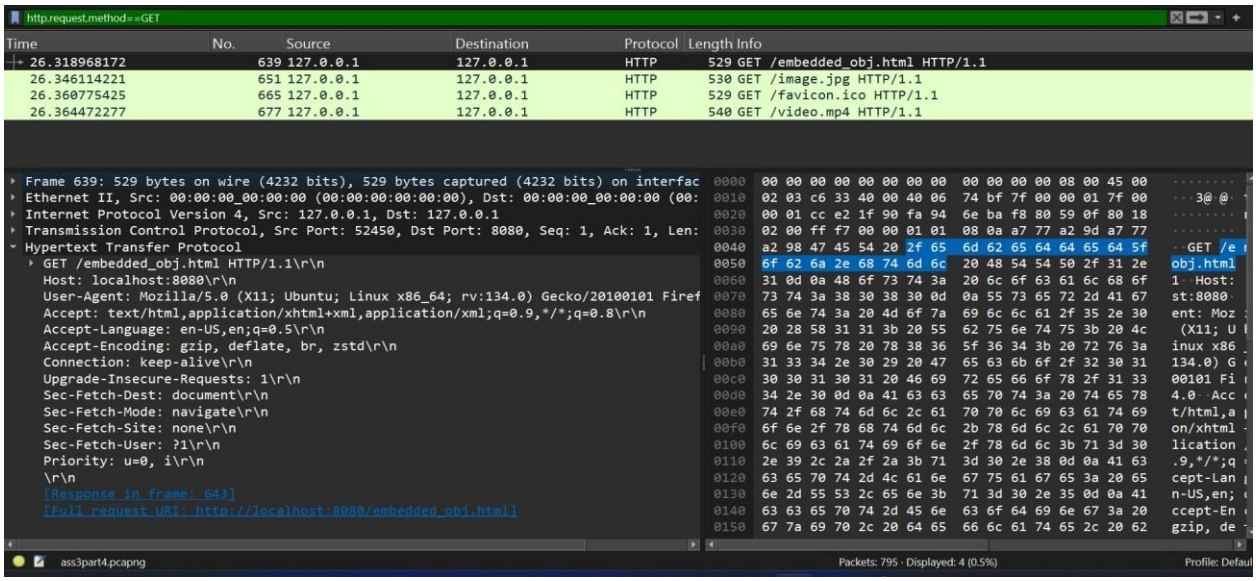
4. **How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?**



TCP payload is **4861 bytes** and one data containing TCP segment was needed to carry the single HTTP response and the text of the Bill of Rights.

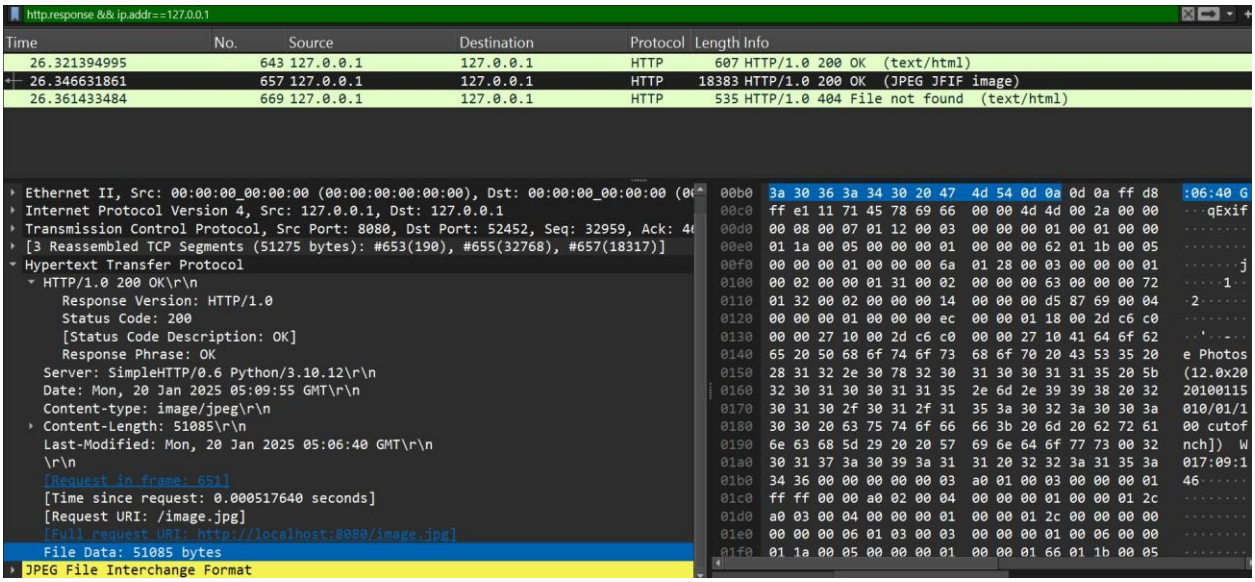
Part-4: HTML Documents with Embedded Objects

1. **How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?**



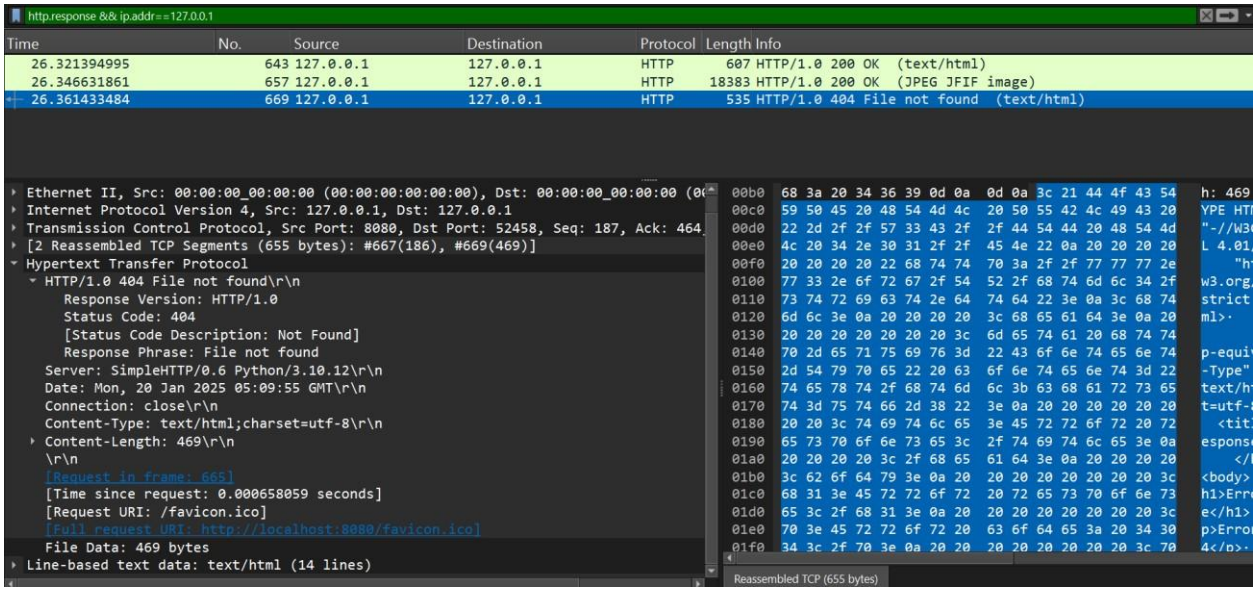
Browser sent **4** HTTP GET request messages. These GET requests are sent to the Internet address **127.0.0.1**

2. **Is the size of the image file matching with the size of the file in the Wireshark? Provide the image file size in bytes.**



The size of the image file we downloaded is of 49.8 KB which is of 50995.2 bytes whereas the size of the file in wireshark is 51085 bytes. Therefore, both are not matching.

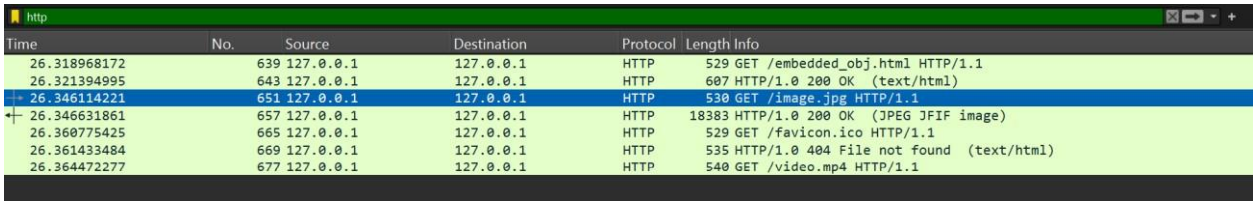
3. **Were there any HTTP response codes indicating errors (e.g., 4xx or 5xx)? If so, what do they indicate?**



Yes, there was an HTTP response code **404** with the phrase **File Not Found**. This indicates that the server could not find the requested resource. It typically occurs when the URL is incorrect, the resource has been moved or deleted, or there is a restriction on accessing the resource. The 404 code means the server was unable to locate the file or page, often due to issues with the request's path or the server configuration.

4. **Mention the source and destination IP addresses for the HTTP requests made to the image and video files. Explain.**

When hosting a local server and accessing resources via http://localhost, the communication happens on the loopback interface (lo).



For each HTTP request made for the HTML, image, and video files, the IP addresses are:

- Source IP: **127.0.0.1** (the local machine initiating the request).
- Destination IP: **127.0.0.1** (the server on the same machine).

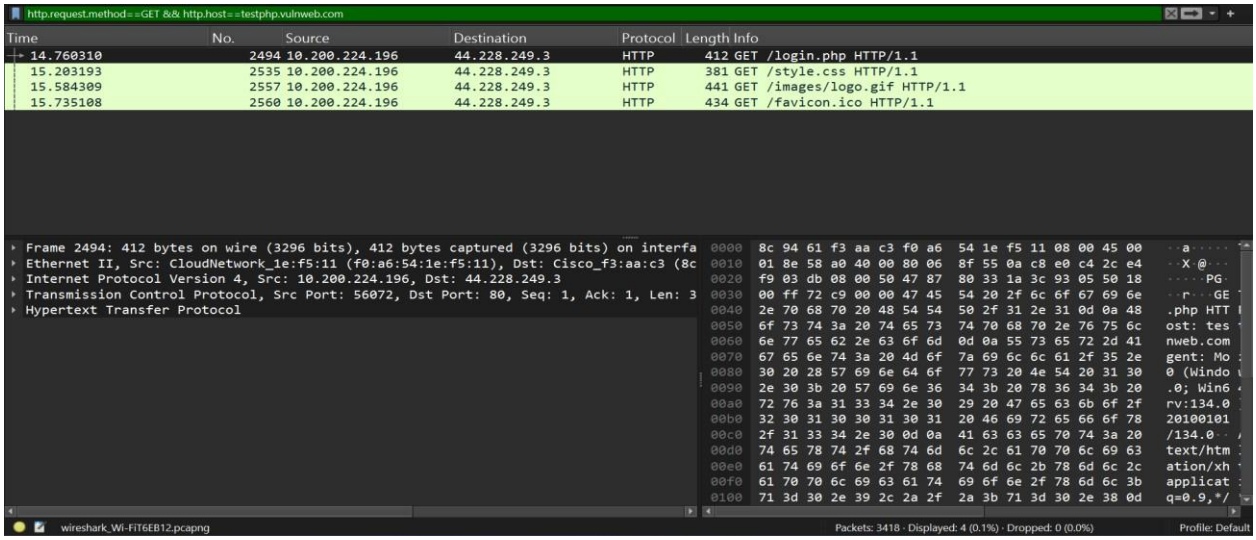
This is because the loopback interface is used for local communication, which doesn't require external network traffic.

Part-5: HTTP Authentication

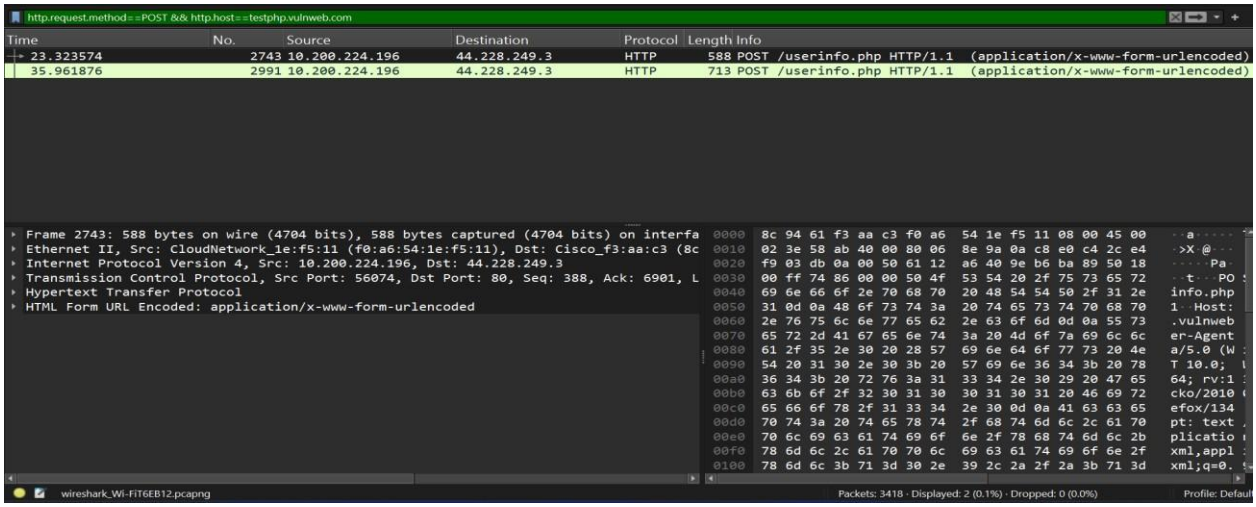
Do the following:

- Open the browser and paste the URL: http://testphp.vulnweb.com/login.php
- Add the username and password as “test” and click on the “login “button
- Change any one of the field's values of any of the information presented on the screen and click on the “update” button

1. How many GET and POST packets do you observe in the trace?



The number of GET packets is **4.**

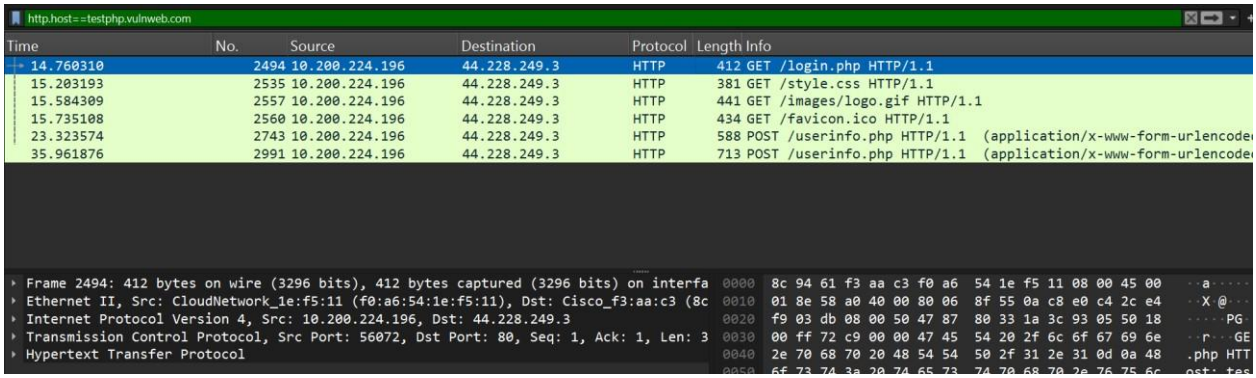


The number of POST packets is **2.**

2. Write the domain and the corresponding IP address that you visited.

The domain **testphp.vulnweb.com** resolves to the IP address **44.228.249.3.**

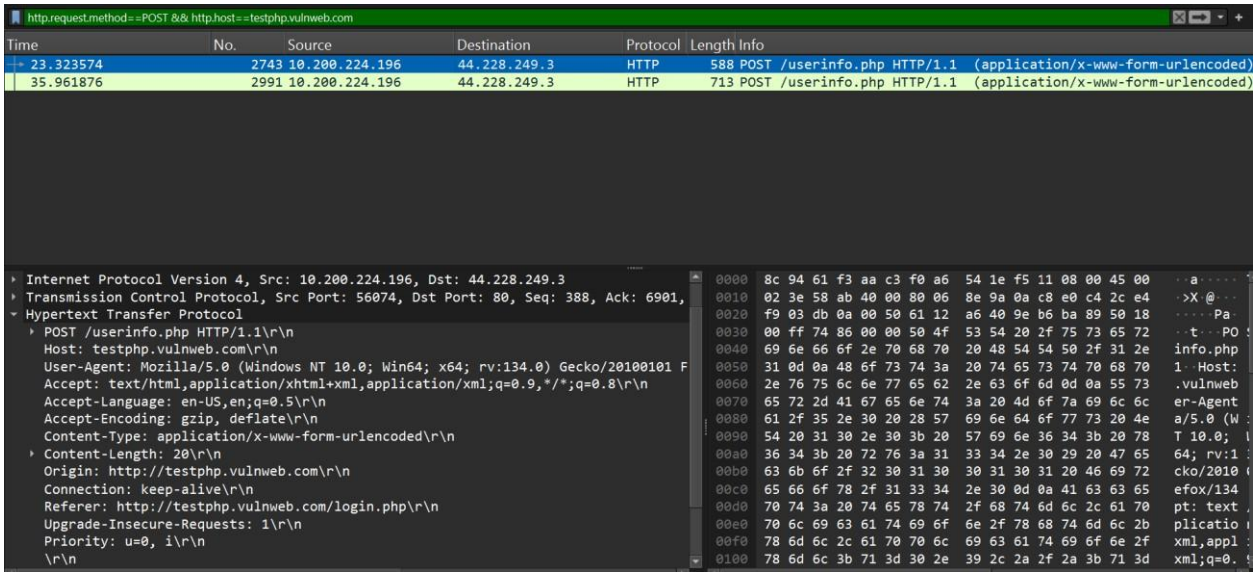
3. **What is the web server's destination port you have requested, and is it a standard port? If yes, then which protocol?**



The web server's destination port is **port 80**, which is a standard port for **HTTP (Hypertext Transfer Protocol)**.

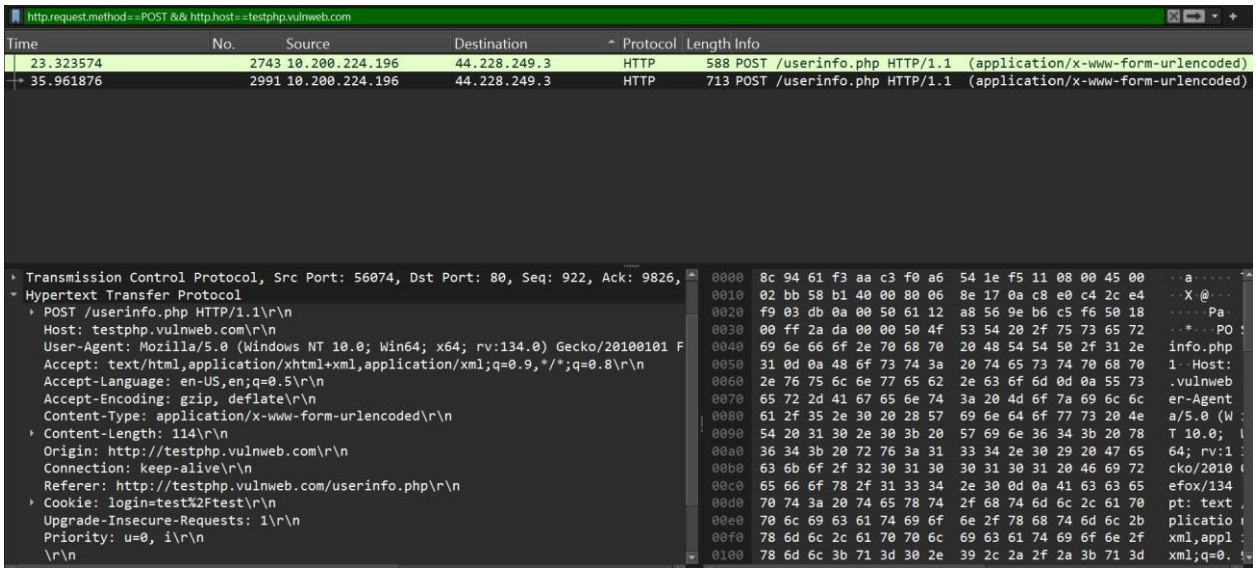
- **Port 80** is the default port used by web servers to handle HTTP requests, meaning all regular web traffic is routed through this port unless specified otherwise.
- If no port is explicitly mentioned in the URL, the browser automatically assumes **port 80** for HTTP requests.
- This is the standard, unencrypted protocol used for most web browsing, in contrast to HTTPS, which operates on **port 443** for encrypted communication.

4. **Inspect the contents of the first HTTP POST request sent from your browser to the server. Do you notice an If-Modified-Since line in the HTTP POST request? If so, what is its significance?**



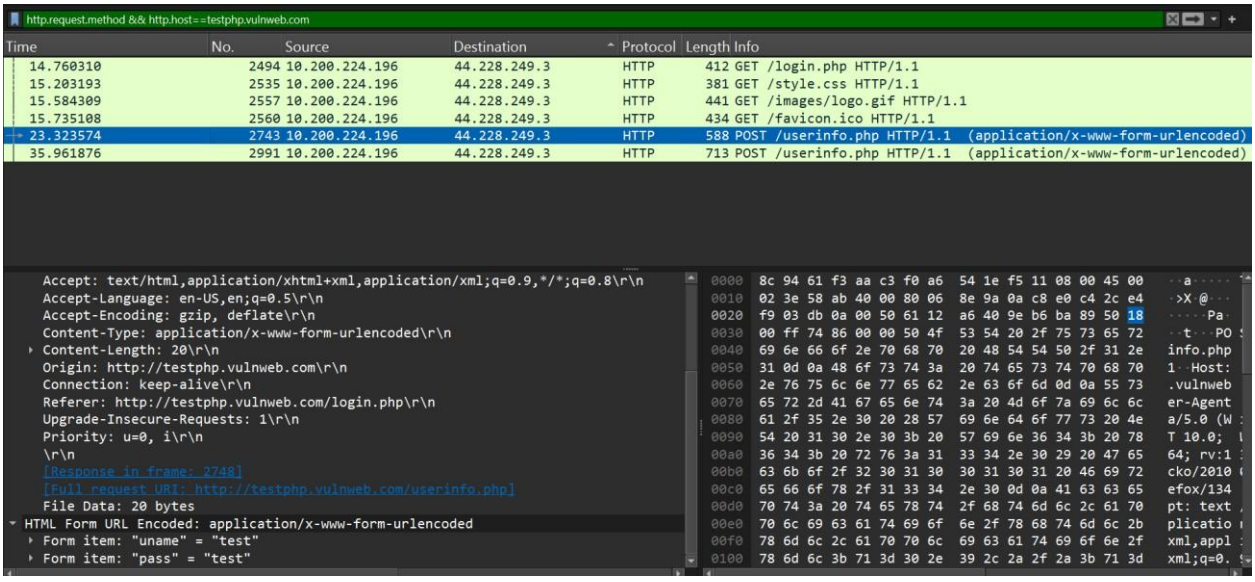
The **If-Modified-Since** line is not present in the first HTTP POST request because POST requests are used to send data to the server, not for validating cached resources like GET requests.

5. **Now inspect the contents of the second HTTP POST request from your browser to the server. What information follows the "IF-MODIFIED-SINCE:" header?**



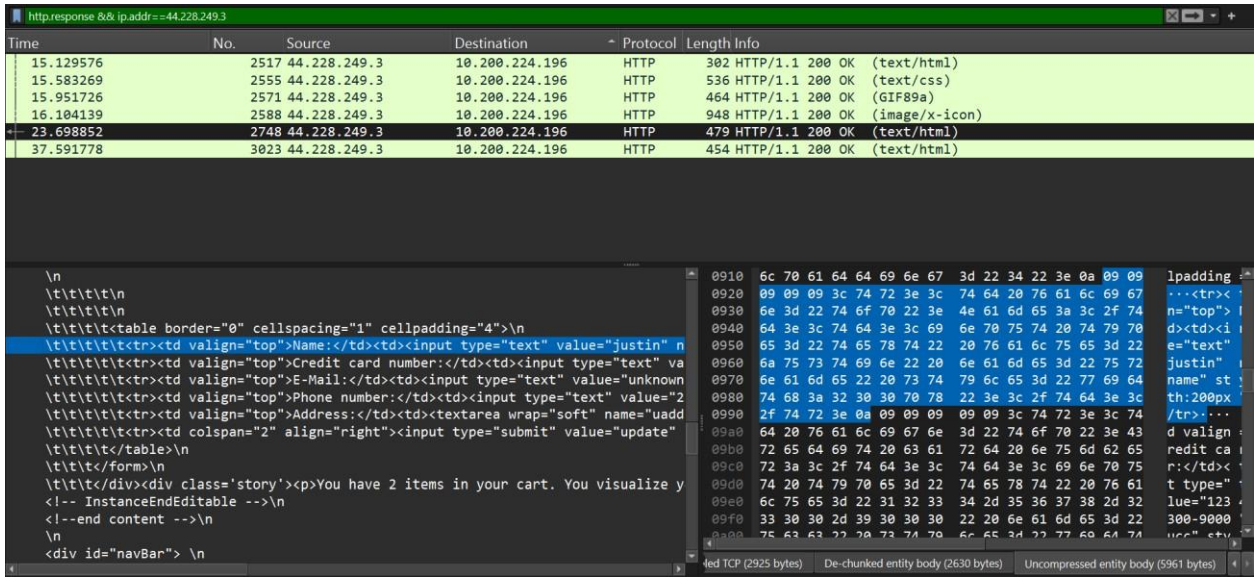
In the second HTTP POST request, the **IF-MODIFIED-SINCE:** line is not present. This is because **POST requests** are used to send data to the server (such as form submissions or file uploads) rather than retrieve or validate cached resources. The **IF-MODIFIED-SINCE:** line is relevant for **GET requests** to check if the resource has changed, while POST requests focus on submitting new data, making the caching validation unnecessary.

6. **Trace the HTTP communication when accessing the authentication server for the website http://testphp.vulnweb.com/login.php. Analyze the following:**
    a. **Request Details: What information (e.g., username and password) is included in the HTTP request when you attempt to log in?**



From the HTTP POST request for http://testphp.vulnweb.com/login.php inside Hypertext Transfer Protocol we can get the information of username and password entered during login, sent as form data in the request body (here, username=test &password=test).

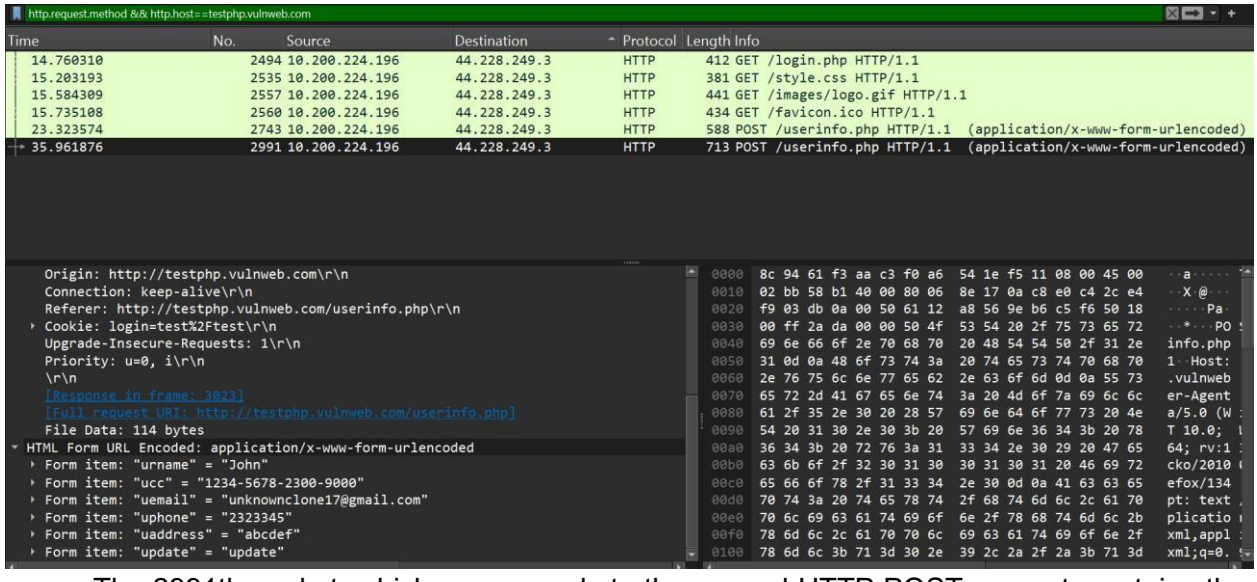
b. Response Behavior: After submitting the login details, does the server respond with a new webpage? If so, what information is displayed on this page?



After submitting the login details, the server responds with a new webpage that contains user-specific information. This information is displayed as part of the HTML content returned in response to the first **GET** request following a successful login. From, the content we get the information displayed on the page which may include details like the Name, Credit card number, E-Mail, Phone number, Address.

7. Update information

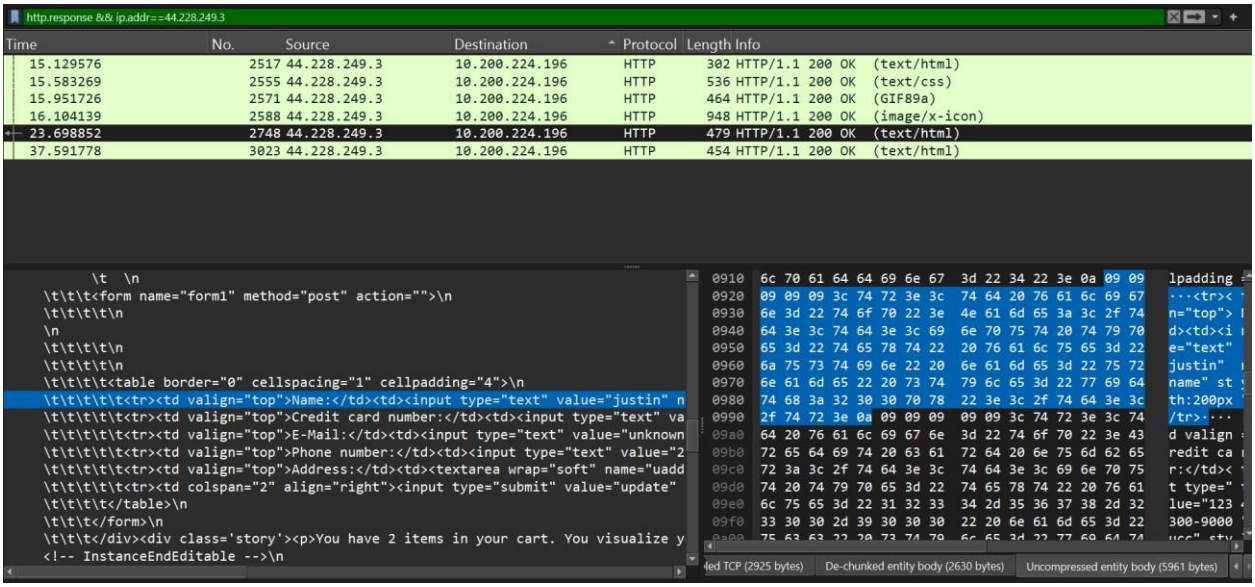
a. Request Packet: Which packet in the HTTP trace contains the updated information you entered, and can you observe this information in the request?

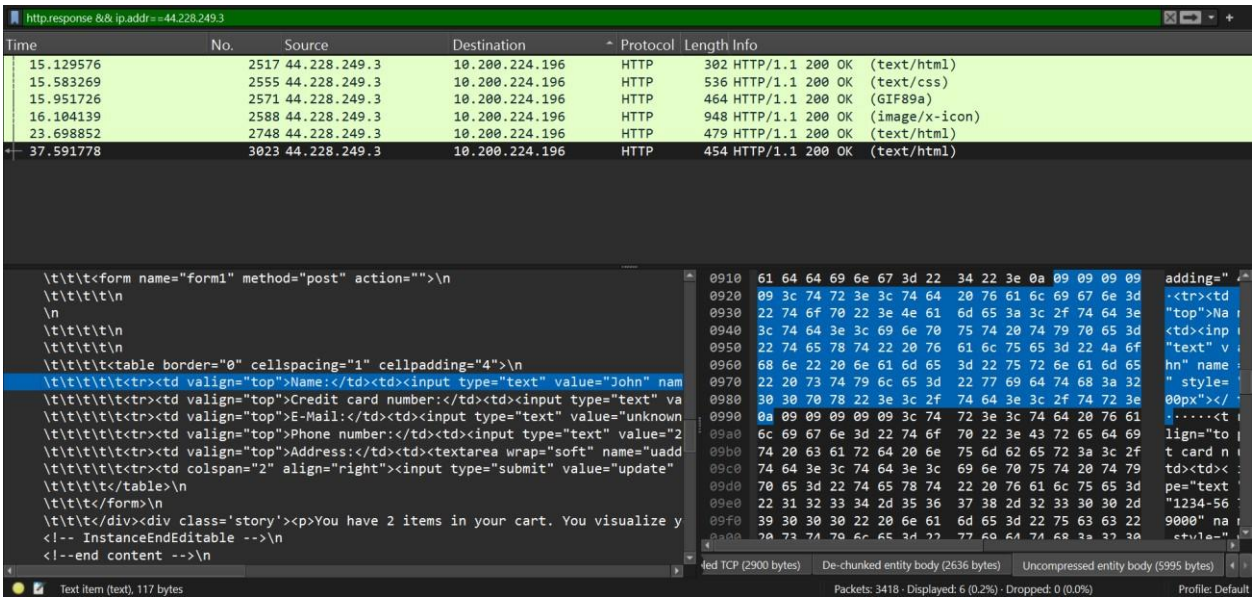


The 2991th packet, which corresponds to the second HTTP POST request, contains the updated information I entered. This packet includes the data submitted in the request, such as the Name, Credit card number, E-Mail, Phone number, Address which can be

observed in the request body. The details are visible as form data within the HTTP request, confirming that the updated information was sent to the server.

b. Response Packet: Do you see the updated information reflected in the response packet for the corresponding HTTP request?





Yes, we can see the updated information is reflected in the response packet. After submitting the updated data, the server's response, which is in the form of HTML code, shows the changes. Here, the name previously set to "Justin" is now updated to "John" in the server's response, confirming that the new information was processed successfully.