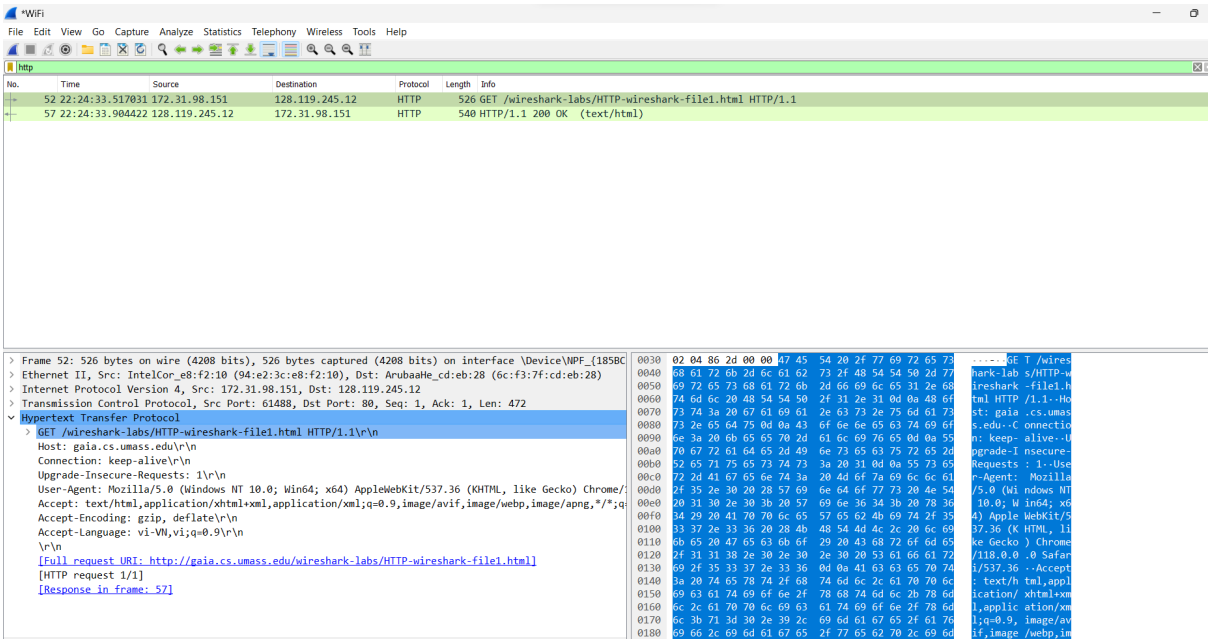
Lab 02

1. The Basic HTTP GET/response interaction

Do the following:

1. Start up your web browser.
2. Start up the Wireshark packet sniffer, as described in the Introductory lab (but don't yet begin packet capture). Enter "http" (just the letters, not the quotation marks, and in lower case) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).
3. Wait a bit more than one minute (we'll see why shortly), and then begin Wireshark packet capture.
4. Enter the following to your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html Your browser should display the very simple, one-line HTML file.
5. Stop Wireshark packet capture.



Answer the following questions:

1. Is your browser running HTTP version 1.0, 1.1, or 2? 1.1

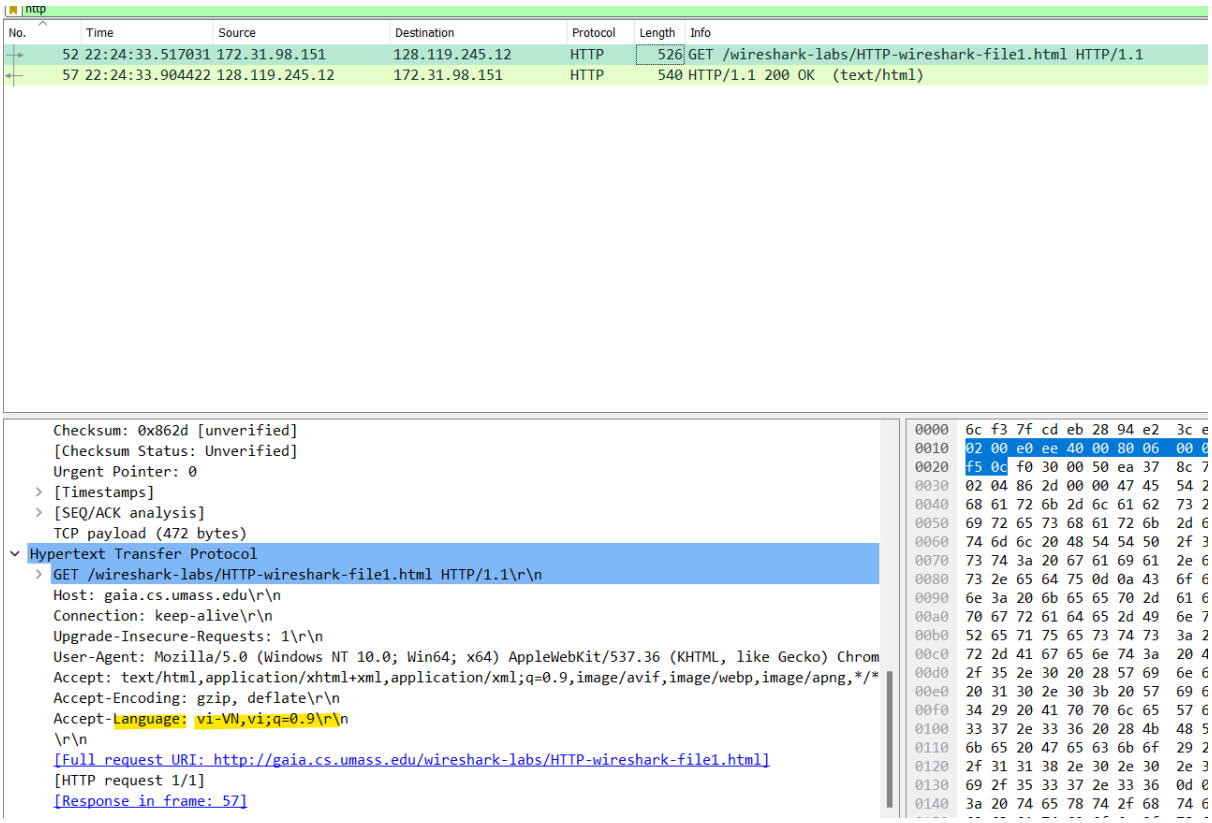
```
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
```

   What version of HTTP is the server running? 1.1

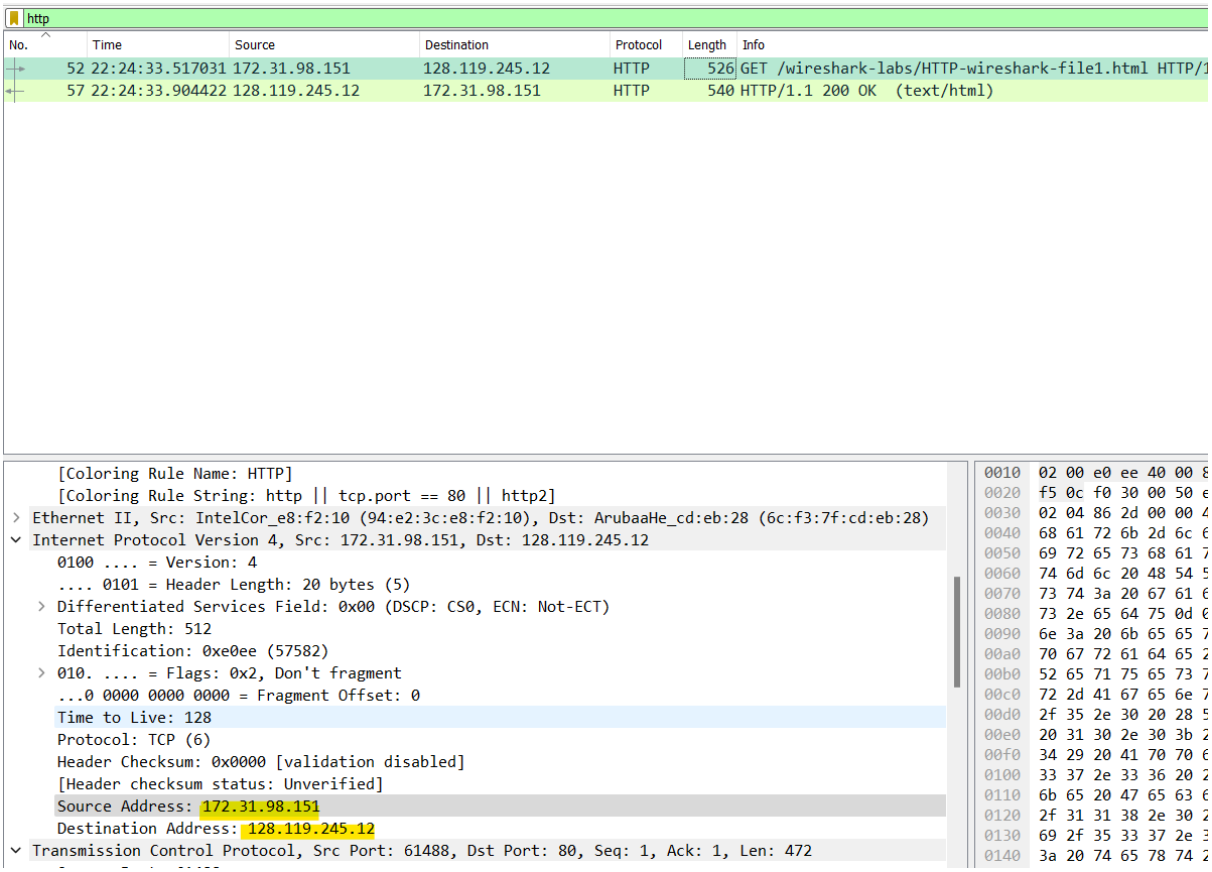
```
HTTP/1.1 200 OK\r\n
```

2. What languages (if any) does your browser indicate that it can accept to the server?

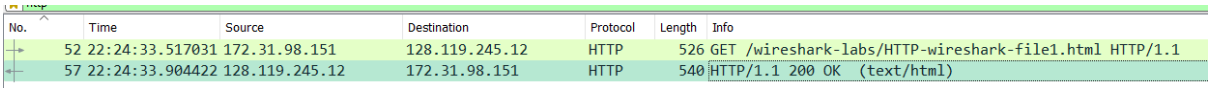
vi-VN and vi



3. What is the IP address of your computer? 172.31.98.151
   What is the IP address of the gaia.cs.umass.edu server? 128.119.245.12

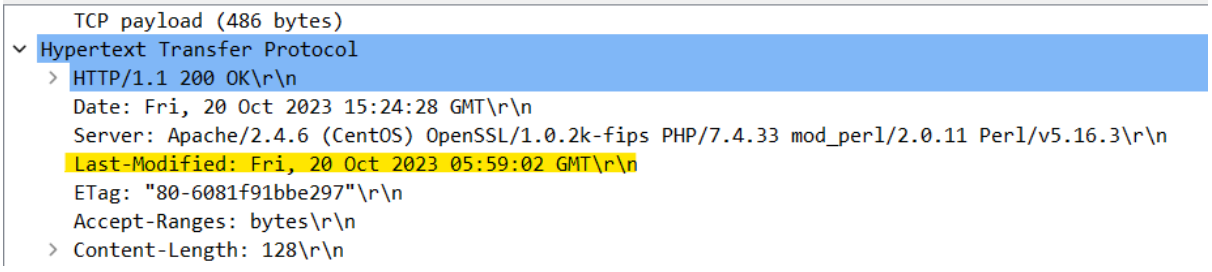


4. What is the status code returned from the server to your browser? 200 OK



5. When was the HTML file that you are retrieving last modified at the server?

Fri, 20 Oct 2023 05:59:02 GMT\r\n



6. How many bytes of content are being returned to your browser? 128 bytes

```
TCP payload (486 bytes)
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Date: Fri, 20 Oct 2023 15:24:28 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod
  Last-Modified: Fri, 20 Oct 2023 05:59:02 GMT\r\n
  ETag: "80-6081f91bbe297"\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 128\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.387391000 seconds]
  [Request in frame: 52]
  [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wires
  File Data: 128 bytes
```

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.

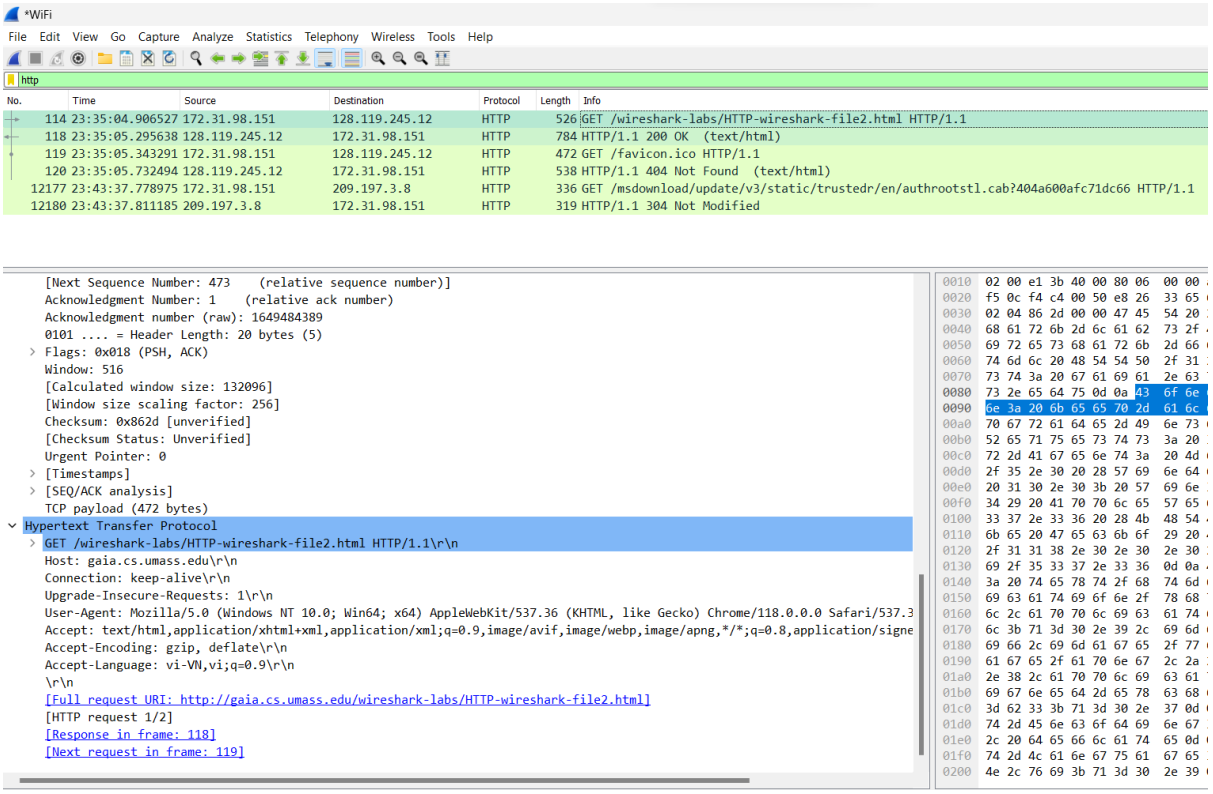
Nothing.

2. The HTTP CONDITIONAL GET/response interaction

Answer the following questions:

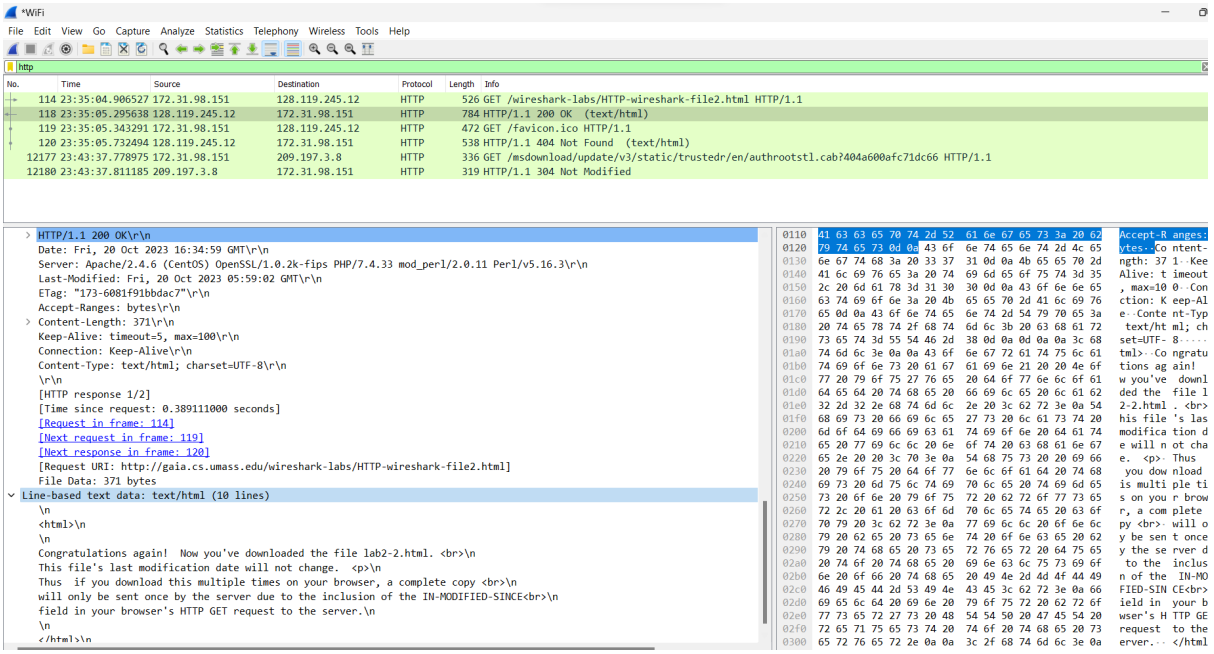
8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

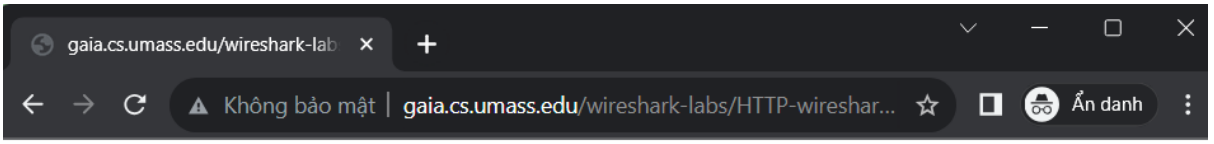
There is no "IF-MODIFIED-SINCE" line in the first HTTP GET.



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

   The server did explicitly return the contents of the file. Wireshark includes a section titled “Line-Based Text Data” which shows what the server sent back to my browser which is specifically what the website showed when I brought it up on my browser



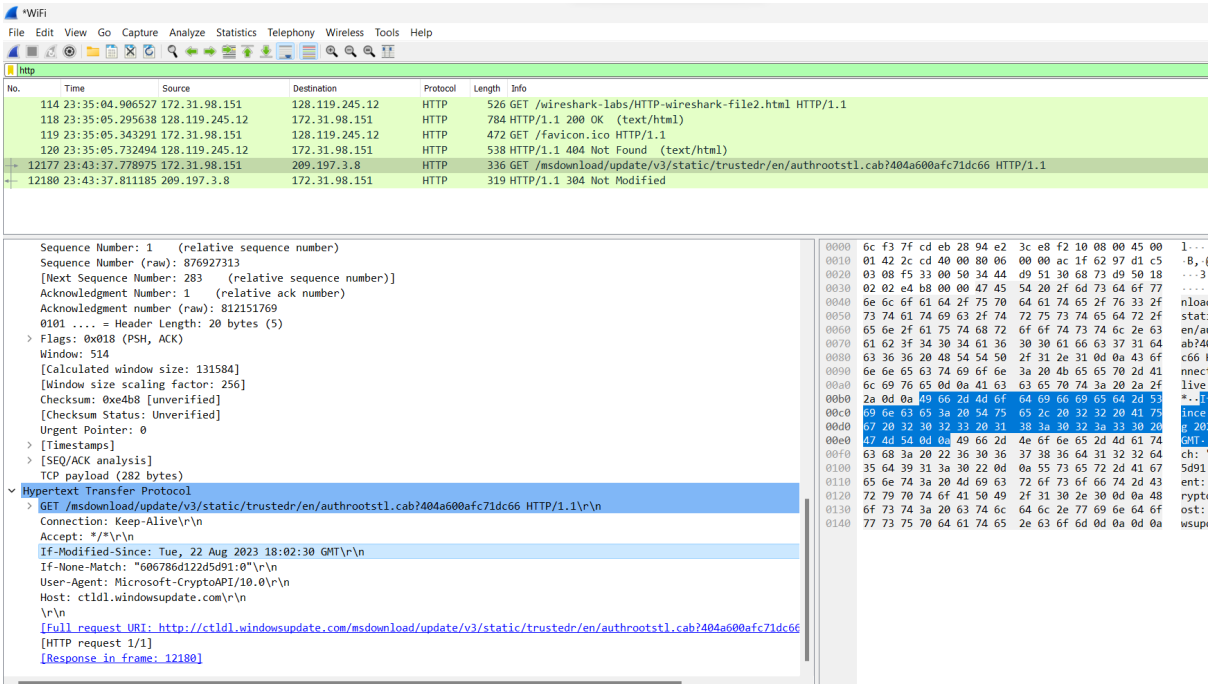


Congratulations again! Now you've downloaded the file lab2-2.html.
This file's last modification date will not change.

Thus if you download this multiple times on your browser, a complete copy
will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE
field in your browser's HTTP GET request to the server.

10. 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[4]? If so, what information follows the "IF-MODIFIED-SINCE:" header?

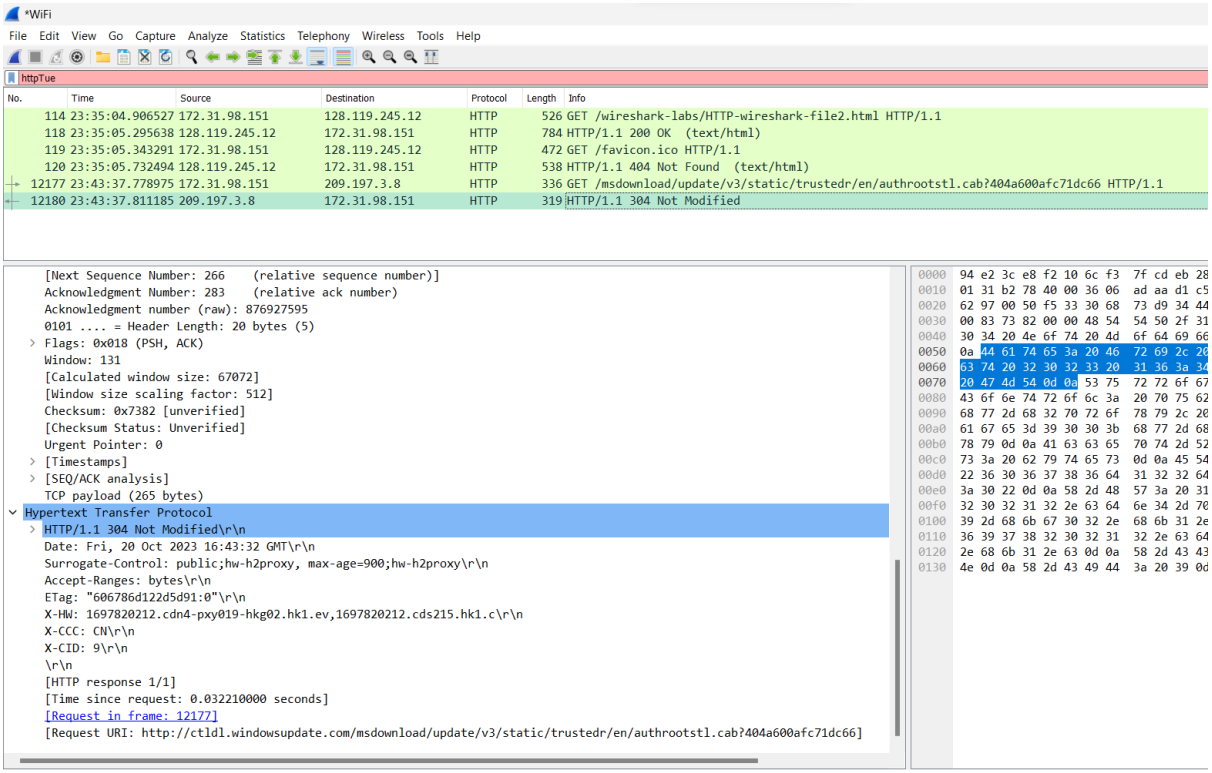
There is an "IF-MODIFIED-SINCE" line in the second HTTP GET. The information follows are "Tue, 22 Aug 2023 18:02:30 GMT".



11. 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.

    The HTTP status code is “304: Not Modified”.

    This is because it has not received the content explicitly since the content was not modified in the first time requested, but in the second time there is no need to download the file again.

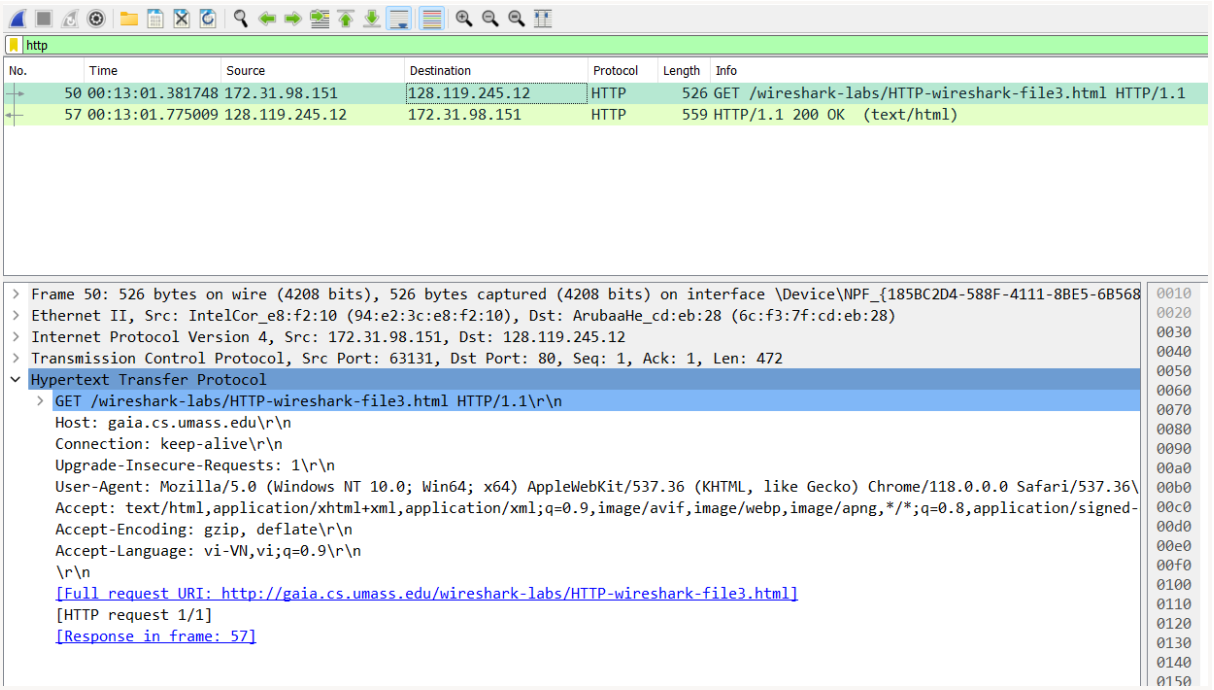


3. Retrieving Long Documents

Answer the following questions:

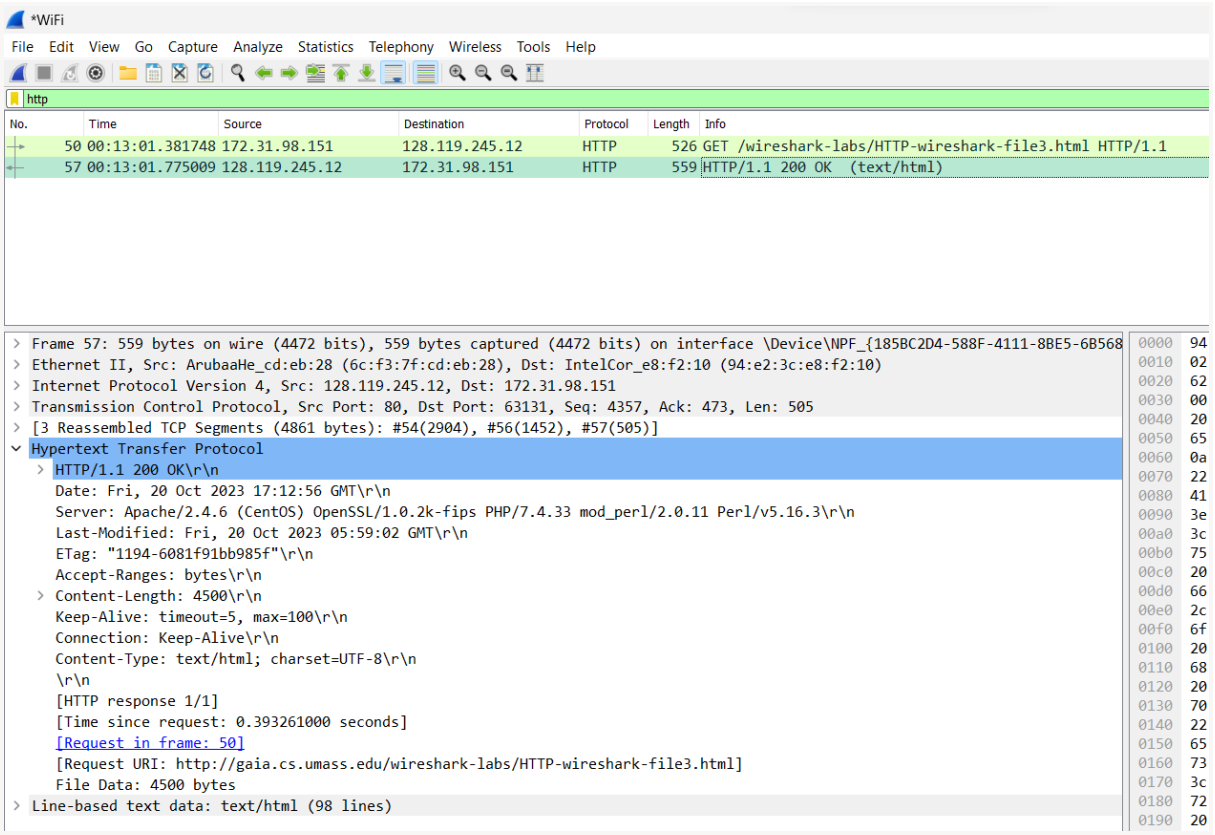
12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

My browser only sent 1 HTTP GET request to the server.
Packet number 50.



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

Packet number 57.

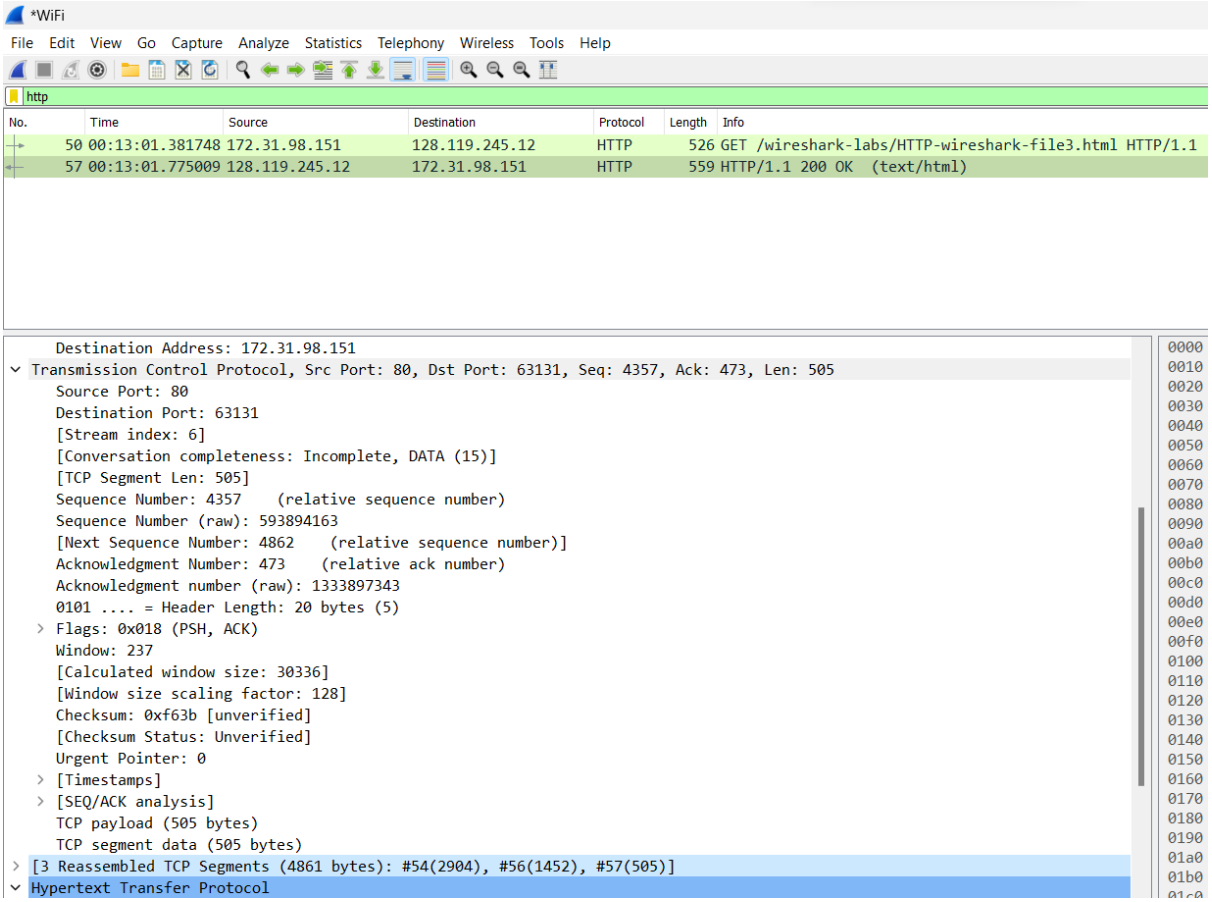


14. What is the status code and phrase in the response?

The code and phrase in the response was 200 OK

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

3 Reassembled TCP Segments (4861 bytes): #54(2904), #56(1452), #57(505)



4. HTML Documents with Embedded Objects

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html

Answer the following questions:

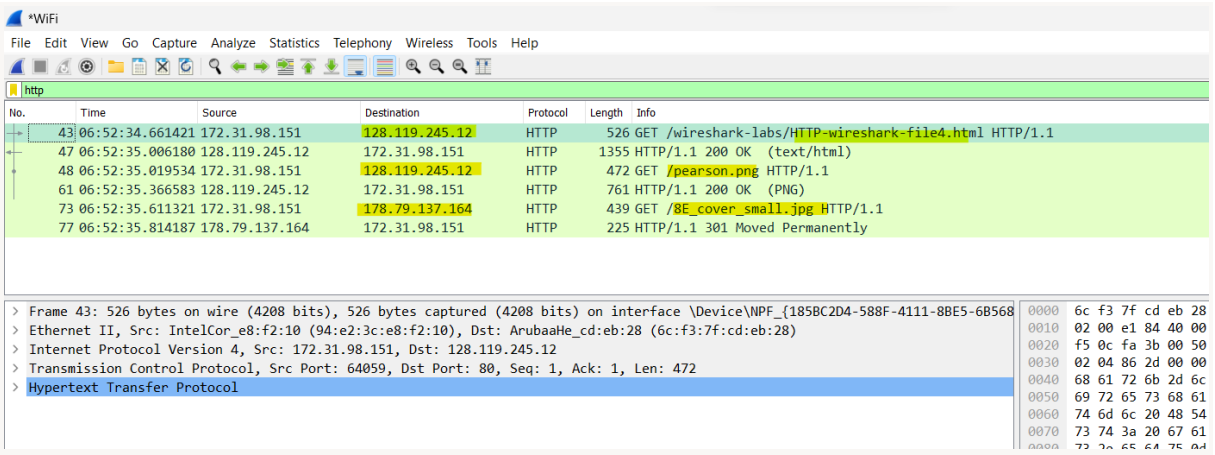
16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

My browser sent 3 http GET message requests. It sent to the internet address of the main html page and the location of the images:

Initial Page address: 128.119.245.12

Pearson Logo: 128.199.245.12

Pearson book, 5th Edition: 172.31.98.151



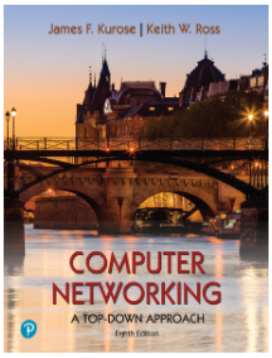
17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

The two photos were downloaded sequentially by the browser.

The first image "pearson.png" was requested and retrieved the 200 OK status. The second image was requested and first retrieved failed with "301 moved permanently". I believe this is the situation since the first image was requested and transmitted by the browser before the second image. If they had been operating concurrently, both files would have been requested and returned in the same time period. However, in this example, the second image was requested only after the first image was returned.

Pearson

This little HTML file is being served by gaia.cs.umass.edu. It contains two embedded images. The image above, also served from the gaia.cs.umass.edu web site, is the logo of our publisher, Pearson. The image of our 8th edition book cover below is stored at, and served from, a WWW server kurose.cslash.net in France:



And while we have your attention, you might want to take time to check out the available open resources for this book at http://gaia.cs.umass.edu/kurose_ross.

5 HTTP Authentication

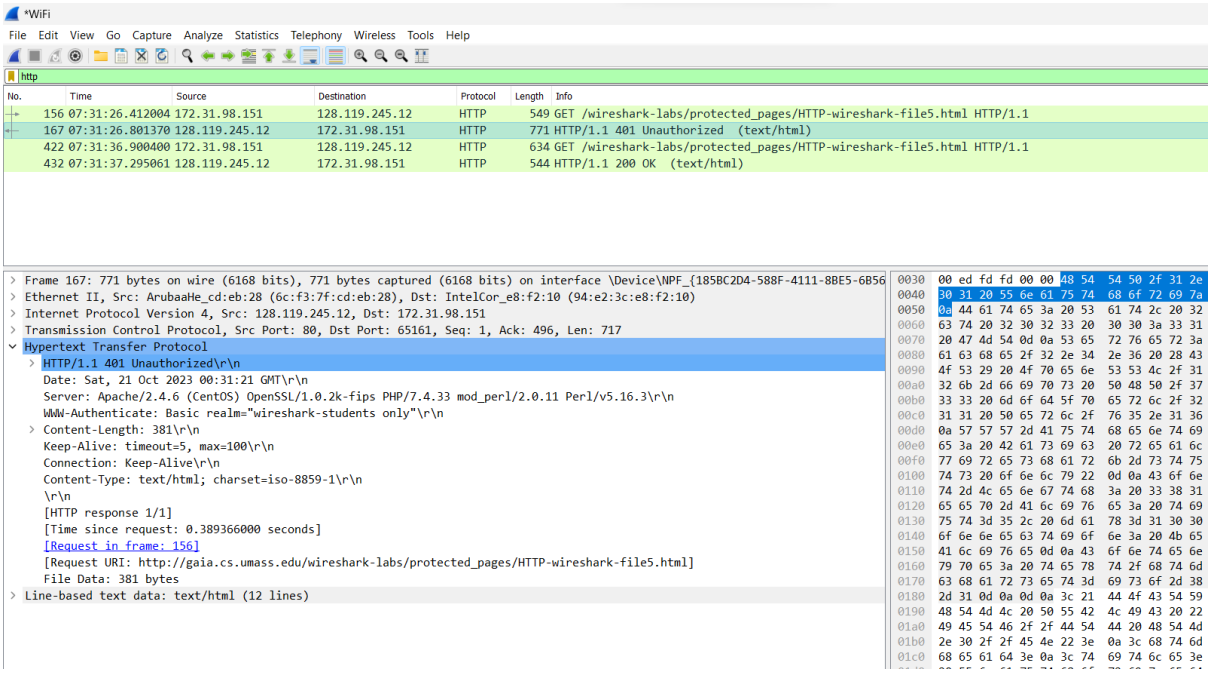
http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
The username is “wireshark-students” (without the quotes), and the password is “network”

Answer the following questions[5]:

18. What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?

State code: 401

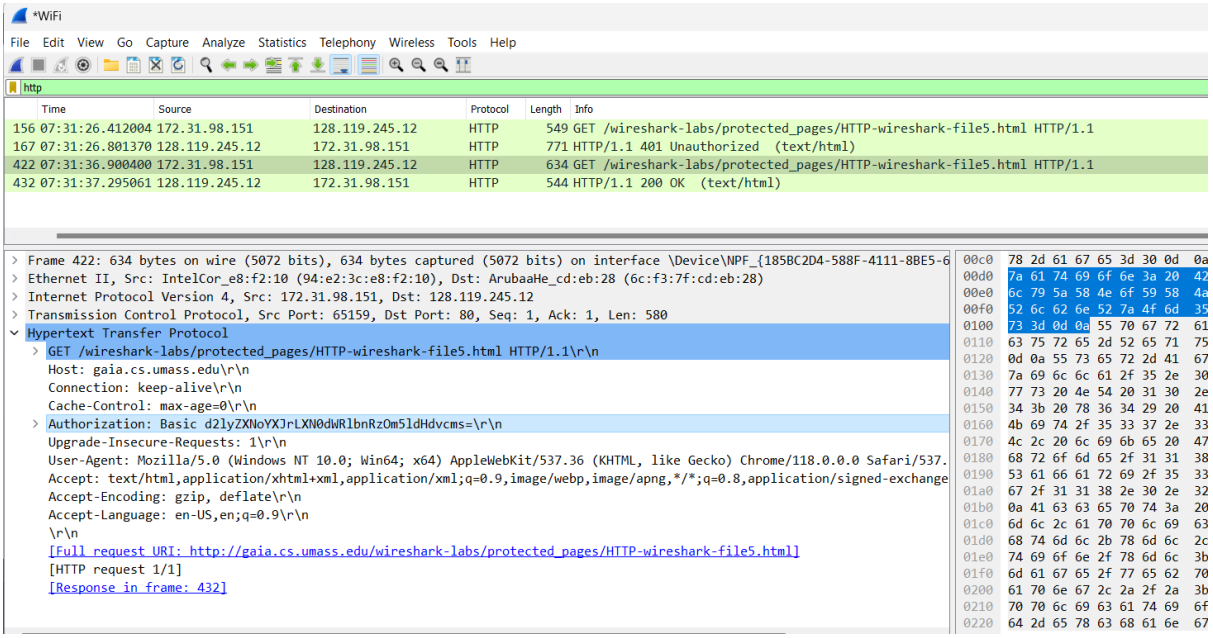
Phase: Unauthorized



19. When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms =\r\n

The permission field has been added as a new field. This is included because, along with our request, we supplied the server a username and password indicating that we were permitted to see the page.


*WiFi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Time Source Destination Protocol Length Info
156 07:31:26.412004 172.31.98.151 128.119.245.12 HTTP 549 GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
167 07:31:26.801370 128.119.245.12 172.31.98.151 HTTP 771 HTTP/1.1 401 Unauthorized (text/html)
422 07:31:36.900400 172.31.98.151 128.119.245.12 HTTP 634 GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
432 07:31:37.295061 128.119.245.12 172.31.98.151 HTTP 544 HTTP/1.1 200 OK (text/html)
Frame 422: 634 bytes on wire (5072 bits), 634 bytes captured (5072 bits) on interface \Device\NPF_{185BC2D4-588F-4111-8BE5-6
Ethernet II, Src: IntelCor_e8:f2:10 (94:e2:3c:e8:f2:10), Dst: ArubaaHe_cd:eb:28 (6c:f3:7f:cd:eb:28)
Internet Protocol Version 4, Src: 172.31.98.151, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 65159, Dst Port: 80, Seq: 1, Ack: 1, Len: 580
Hypertext Transfer Protocol
GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
Host: gaia.cs.umass.edu\r\n
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\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/118.0.0.0 Safari/537.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html]
[HTTP request 1/1]
[Response in frame: 432]
00c0 78 2d 61 67 65 3d 30 0d 0a
00d0 7a 61 74 69 6f 6e 3a 20 42
00e0 6c 79 5a 58 4e 6f 59 58 4a
00f0 52 6c 62 6e 52 7a 4f 6d 35
0100 73 3d 0d 0a 55 70 67 72 61
0110 63 75 72 65 2d 52 65 71 75
0120 0d 0a 55 73 65 72 2d 41 67
0130 7a 69 6c 6c 61 2f 35 2e 30
0140 77 73 20 4e 54 20 31 30 2e
0150 34 3b 20 78 36 34 29 20 41
0160 4b 69 74 2f 35 33 37 2e 33
0170 4c 2c 20 6c 69 6b 65 20 47
0180 68 72 6f 6d 65 2f 31 31 38
0190 53 61 66 61 72 69 2f 35 33
01a0 67 2f 31 31 38 2e 30 2e 32
01b0 0a 41 63 63 65 70 74 3a 20
01c0 6d 6c 2c 61 70 70 6c 69 63
01d0 68 74 6d 6c 2b 78 6d 6c 2c
01e0 74 69 6f 6e 2f 78 6d 6c 3b
01f0 6d 61 67 65 2f 77 65 62 70
0200 61 70 6e 67 2c 2a 2f 2a 3b
0210 70 70 6c 69 63 61 74 69 6f
0220 64 2d 65 78 63 68 61 6e 67