LAB -1

DATA COMMUNICATIONS NETWORKS

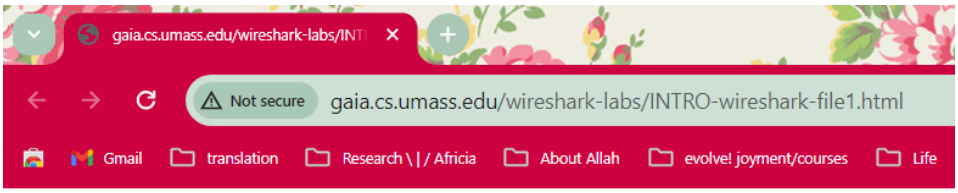
Prof. Dr. Karim Banawan - Prof. Dr. Noha ElKorany
Communication & Electronics Department

Asmaa Gamal Abdel-Halem Mabrouk Nagy
أسماء جمال عبد الحليم مبروك ناجي
15010473 - section 8

Introduction To “Wireshark” Packet Sniffer Program

Section 0 : Taking Wireshark for a Test Run

The Task:



Congratulations! You've downloaded the first Wireshark lab file!

The Questions & Answers:

0. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.

From the Wireshark capture below, we can see:

- TCP protocols
- QUIC protocols which are a new multiplexed transport built on top of UDP protocol
- HTTP protocols
- ARP (Address Resolution Protocol).
- DNS (Domain Name System protocol).

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
628	5.662137	172.217.18.234	192.168.1.5	QUIC	66	Protected Payload (KP0)
629	5.682454	142.251.37.46	192.168.1.5	QUIC	289	Protected Payload (KP0)
630	5.682691	192.168.1.5	142.251.37.46	QUIC	77	Protected Payload (KP0), DCID=ea0723a926dbf1b9
631	5.684865	142.251.37.46	192.168.1.5	QUIC	64	Protected Payload (KP0)
632	5.712419	192.168.1.5	142.251.37.46	QUIC	74	Protected Payload (KP0), DCID=ea0723a926dbf1b9
633	5.746979	142.251.37.46	192.168.1.5	QUIC	66	Protected Payload (KP0)
634	5.778132	128.119.245.12	192.168.1.5	TCP	54	80 → 1811 [ACK] Seq=1 Ack=567 Win=30336 Len=0
635	5.779670	128.119.245.12	192.168.1.5	HTTP	492	HTTP/1.1 200 OK (text/html)
636	5.823101	192.168.1.5	128.119.245.12	TCP	54	1811 → 80 [ACK] Seq=567 Ack=439 Win=131072 Len=0
637	5.984856	192.168.1.5	128.119.245.12	HTTP	505	GET /favicon.ico HTTP/1.1
638	6.058829	192.168.1.5	8.8.8.8	QUIC	251	Protected Payload (KP0), DCID=eca5f3fd7b57cebc
639	6.059056	192.168.1.5	8.8.8.8	QUIC	251	Protected Payload (KP0), DCID=eca5f3fd7b57cebc
640	6.096929	8.8.8.8	192.168.1.5	QUIC	69	Protected Payload (KP0)
641	6.098209	8.8.8.8	192.168.1.5	QUIC	69	Protected Payload (KP0)
642	6.098329	192.168.1.5	8.8.8.8	QUIC	73	Protected Payload (KP0), DCID=eca5f3fd7b57cebc
545	3.695685	192.168.1.5	142.250.201.4	QUIC	73	Protected Payload (KP0), DCID=fd5ef668f860c4e5
546	3.756685	142.250.201.4	192.168.1.5	QUIC	66	Protected Payload (KP0)
547	4.135227	Intel_db:e3:45	Broadcast	ARP	42	Who has 192.168.1.200? Tell 192.168.1.5
548	5.396256	192.168.1.5	8.8.8.8	QUIC	252	Protected Payload (KP0), DCID=eca5f3fd7b57cebc
549	5.396504	192.168.1.5	8.8.8.8	QUIC	252	Protected Payload (KP0), DCID=eca5f3fd7b57cebc
550	5.404372	192.168.1.5	8.8.8.8	DNS	70	Standard query 0x61fe A dns.google
551	5.404829	192.168.1.5	8.8.8.8	DNS	70	Standard query 0xfa7d HTTPS dns.google

2. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

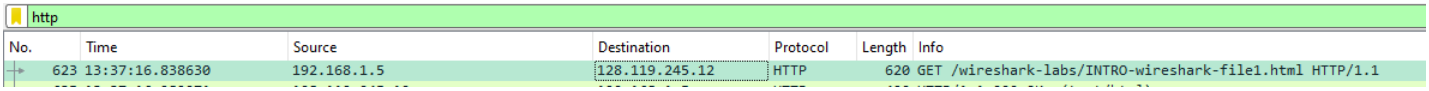
http

No.	Time	Source	Destination	Protocol	Length	Info
623	13:37:16.838630	192.168.1.5	128.119.245.12	HTTP	620	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
635	13:37:16.989271	128.119.245.12	192.168.1.5	HTTP	492	HTTP/1.1 200 OK (text/html)

To determine the time it took, we look at the get request and the HTTP OK arrival times and subtract the GET from the OK.

So, 16.989271 seconds - 16.838630 seconds = 0.150641 seconds.

3. What is the Internet address of the gaia.cs.umass.edu (also known as www-net.cs.umass.edu)? What is the Internet address of your computer?



As shown in the source and destination of the above screenshot:

- The Internet address of gaia.cs.umass.edu is 128.119.245.12
- The Internet address of my laptop computer is 192.168.1.5

4. Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select Print from the Wireshark File command menu, and select the "Selected Packet Only" and "Print as displayed" radial buttons, and then click OK.

Get:

```
C:\Users\DELL\AppData\Local\Temp\wireshark_Wi-FiLQN6J2.pcapng 698 total packets, 4 shown

No.     Time                 Source                Destination           Protocol Length Info
    623 13:37:16.838630      192.168.1.5           128.119.245.12        HTTP     620    GET /wireshark-labs/INTRO-wireshark-
file1.html HTTP/1.1
Frame 623: 620 bytes on wire (4960 bits), 620 bytes captured (4960 bits) on interface \Device\NPF_{1FE64003-
B61C-472C-8851-98EB49C717F6}, id 0
    Section number: 1
    Interface id: 0 (\Device\NPF_{1FE64003-B61C-472C-8851-98EB49C717F6})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar  1, 2024 13:37:16.838630000 Egypt Standard Time
    UTC Arrival Time: Mar  1, 2024 11:37:16.838630000 UTC
    Epoch Arrival Time: 1709293036.838630000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000198000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 5.629029000 seconds]
    Frame Number: 623
    Frame Length: 620 bytes (4960 bits)
    Capture Length: 620 bytes (4960 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: Intel_db:e3:45 (80:00:0b:db:e3:45), Dst: TpLinkTechno_6c:42:46 (cc:32:e5:6c:42:46)
    Destination: TpLinkTechno_6c:42:46 (cc:32:e5:6c:42:46)
    Source: Intel_db:e3:45 (80:00:0b:db:e3:45)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.1.5, Dst: 128.119.245.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 606
    Identification: 0xa345 (41797)
    010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
Protocol: TCP (6)
Header Checksum: 0x1e23 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.5
Destination Address: 128.119.245.12
Transmission Control Protocol, Src Port: 1811, Dst Port: 80, Seq: 1, Ack: 1, Len: 566
    Source Port: 1811
    Destination Port: 80
    [Stream index: 4]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 566]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 188178890
    [Next Sequence Number: 567    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 4210148017
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 514
    [Calculated window size: 131584]
    [Window size scaling factor: 256]
    Checksum: 0xba06 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    TCP payload (566 bytes)
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\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/122.0.0.0 Safari/
537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed
exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    dnt: 1\r\n
    If-None-Match: "51-6127fcb50c915"\r\n
```

```
If-Modified-Since: Thu, 29 Feb 2024 06:59:01 GMT\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
[HTTP request 1/2]
[Response in frame: 635]
[Next request in frame: 637]
```

OK:

C:\Users\DELL\AppData\Local\Temp\wireshark_Wi-FiLQN6J2.pcapng 698 total packets, 4 shown

```
No.     Time                 Source                Destination           Protocol Length Info
    635 13:37:16.989271      128.119.245.12        192.168.1.5           HTTP     492    HTTP/1.1 200 OK  (text/html)
Frame 635: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface \Device\NPF_{1FE64003-
861C-472C-8851-98EB49C717F6}, id 0
    Section number: 1
    Interface id: 0 (\Device\NPF_{1FE64003-861C-472C-8851-98EB49C717F6})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar  1, 2024 13:37:16.989271000 Egypt Standard Time
    UTC Arrival Time: Mar  1, 2024 11:37:16.989271000 UTC
    Epoch Arrival Time: 1709293036.989271000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.001538000 seconds]
    [Time delta from previous displayed frame: 0.150641000 seconds]
    [Time since reference or first frame: 5.779670000 seconds]
    Frame Number: 635
    Frame Length: 492 bytes (3936 bits)
    Capture Length: 492 bytes (3936 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: TpLinkTechno_6c:42:46 (cc:32:e5:6c:42:46), Dst: Intel_db:e3:45 (80:00:0b:db:e3:45)
    Destination: Intel_db:e3:45 (80:00:0b:db:e3:45)
    Source: TpLinkTechno_6c:42:46 (cc:32:e5:6c:42:46)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.5
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 478
    Identification: 0xf0e0 (61664)
    010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 47
    Protocol: TCP (6)
    Header Checksum: 0x2208 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 128.119.245.12
    Destination Address: 192.168.1.5
Transmission Control Protocol, Src Port: 80, Dst Port: 1811, Seq: 1, Ack: 567, Len: 438
    Source Port: 80
    Destination Port: 1811
    [Stream index: 4]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 438]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 4210148017
    [Next Sequence Number: 439    (relative sequence number)]
    Acknowledgment Number: 567    (relative ack number)
    Acknowledgment number (raw): 188179456
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 237
    [Calculated window size: 30336]

    [Window size scaling factor: 128]
    Checksum: 0xd31c [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    TCP payload (438 bytes)
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
        [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        Response Version: HTTP/1.1
        Status Code: 200
        [Status Code Description: OK]
        Response Phrase: OK
    Date: Fri, 01 Mar 2024 11:37:16 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Fri, 01 Mar 2024 06:59:02 GMT\r\n
    ETag: "51-61293e93040a6"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 81\r\n
    Keep-Alive: timeout=5, max=100\r\n
```

C:\Users\DELL\AppData\Local\Temp\wireshark_Wi-FiLQN6J2.pcapng 698 total packets, 4 shown

```
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/2]
    [Time since request: 0.150641000 seconds]
    [Request in frame: 623]
    [Next request in frame: 637]
    [Next response in frame: 650]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    File Data: 81 bytes
Line-based text data: text/html (3 lines)
    <html>\n
    Congratulations!  You've downloaded the first Wireshark lab file!\n
    </html>\n
```

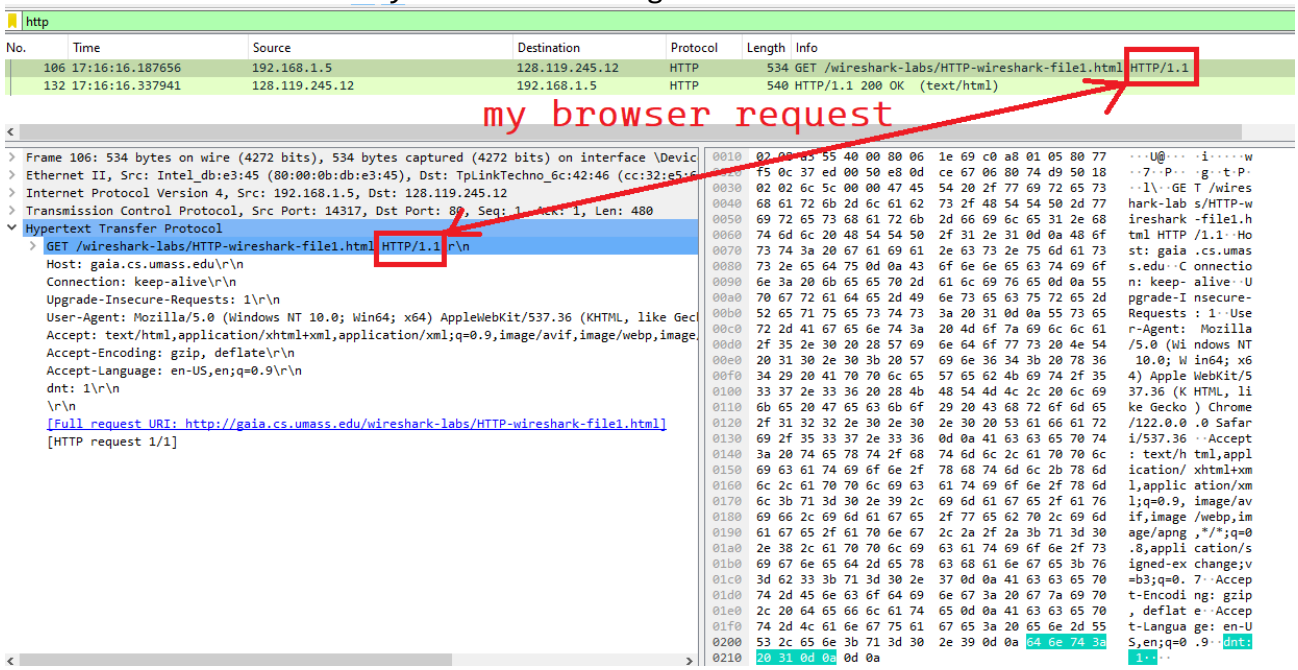
Section1: The Basic HTTP GET/response interaction

The Task:

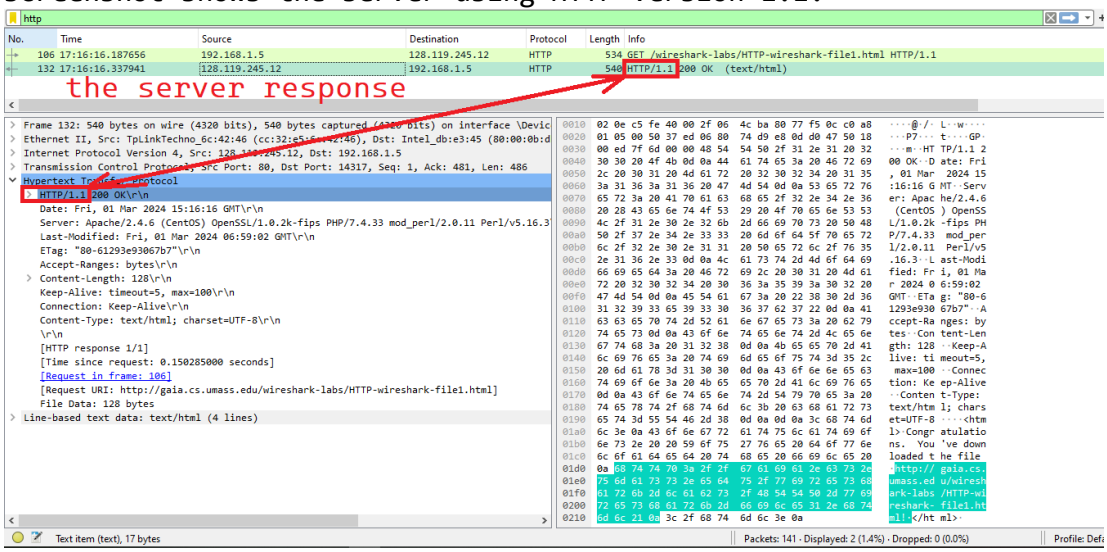
The Questions & Answers:

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

This below screenshot shows my browser running HTTP version 1.1.

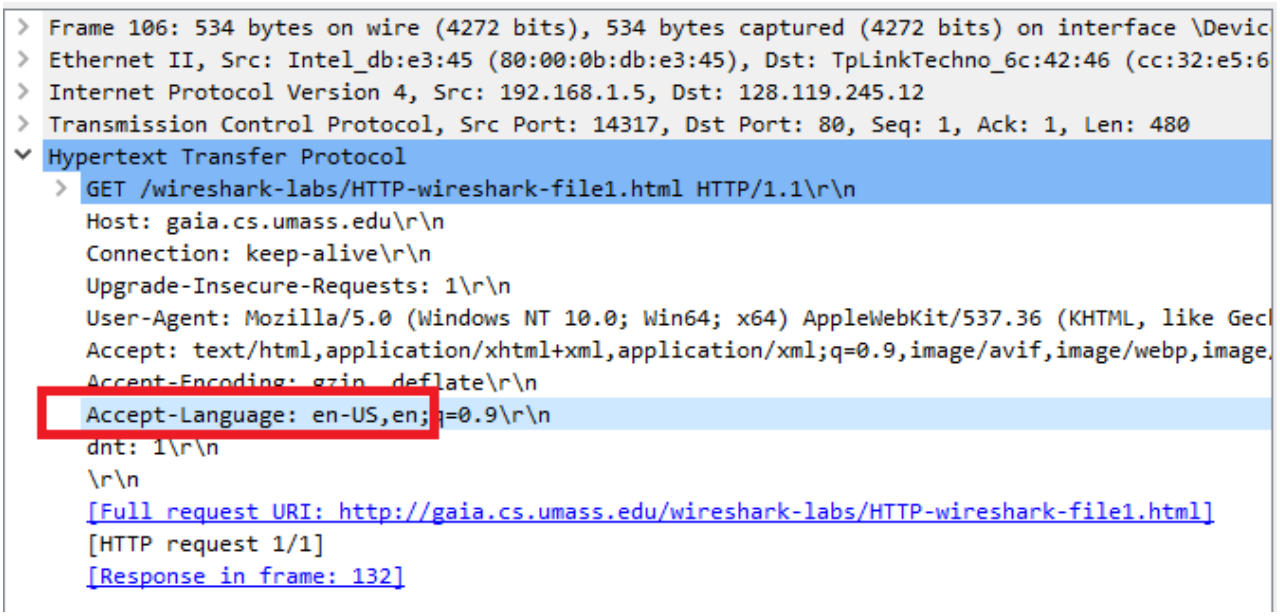


This below screenshot shows the server using HTTP version 1.1.



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

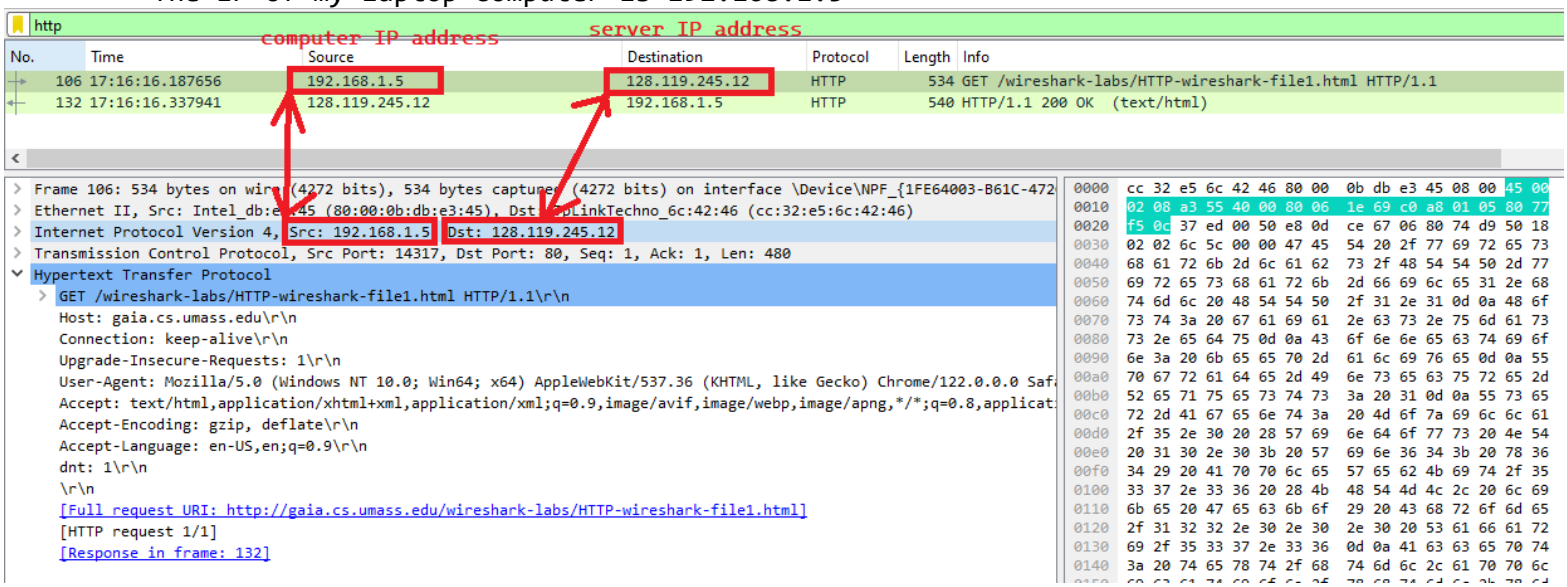
This below screenshot is where it says “Accept-Language”, it lists the US English as its accepted language:



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

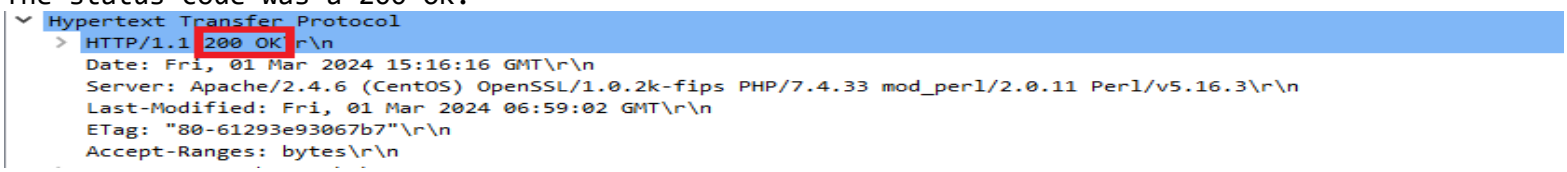
As shown in the below screenshot:

- The IP of gaia.cs.umass.edu is 128.119.245.12
- The IP of my laptop computer is 192.168.1.5



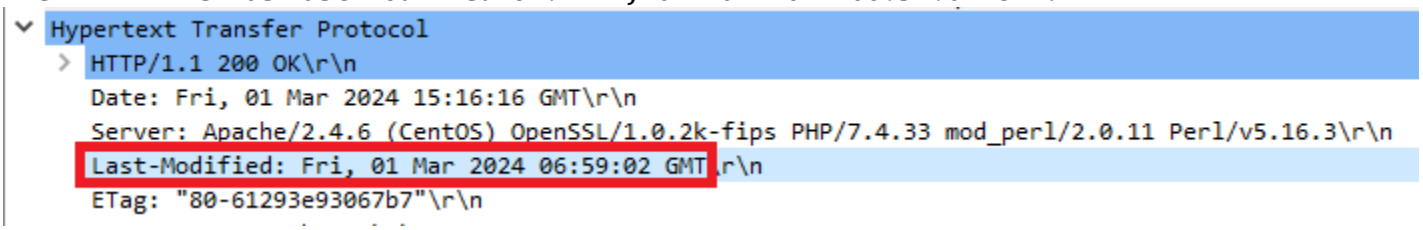
4. What is the status code returned from the server to your browser?

The status code was a 200 OK.



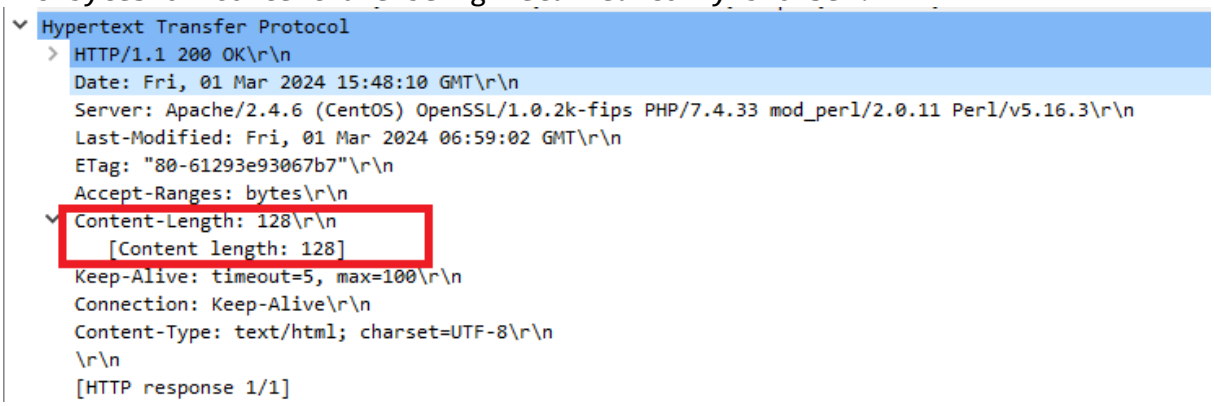
5. When was the HTML file that you are retrieving last modified at the server?

The HTML file was Last-Modified on: Fri, 01 Mar 2024 06:59:02 GMT.



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

128 bytes of content are being returned to my browser.

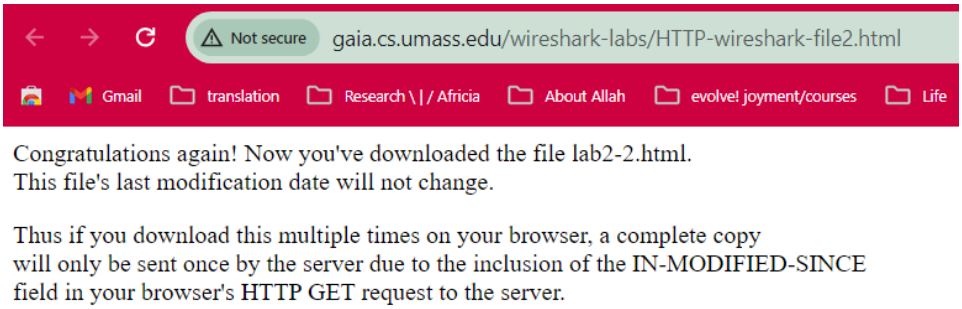


7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

No, all packet headers are completely displayed in the packet-content window by an encoded format: (hexadecimal-ASCII). So,I do not see any headers that are not displayed in the packet window.

Section 2: The HTTP CONDITIONAL GET/response interaction

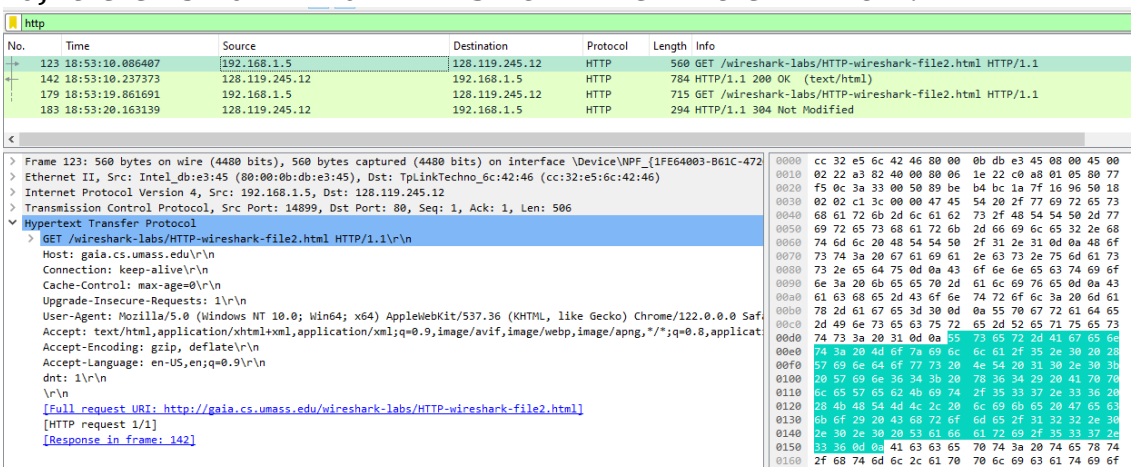
The Task:



The Questions & Answers:

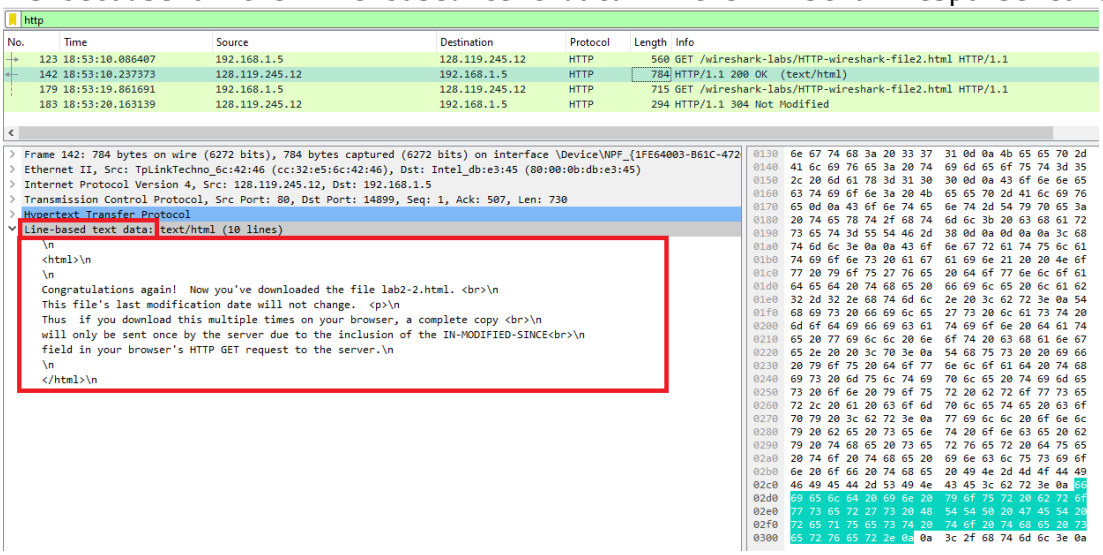
8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

No, there is no "IF-MODIFIED-SINCE" line in the HTTP GET.



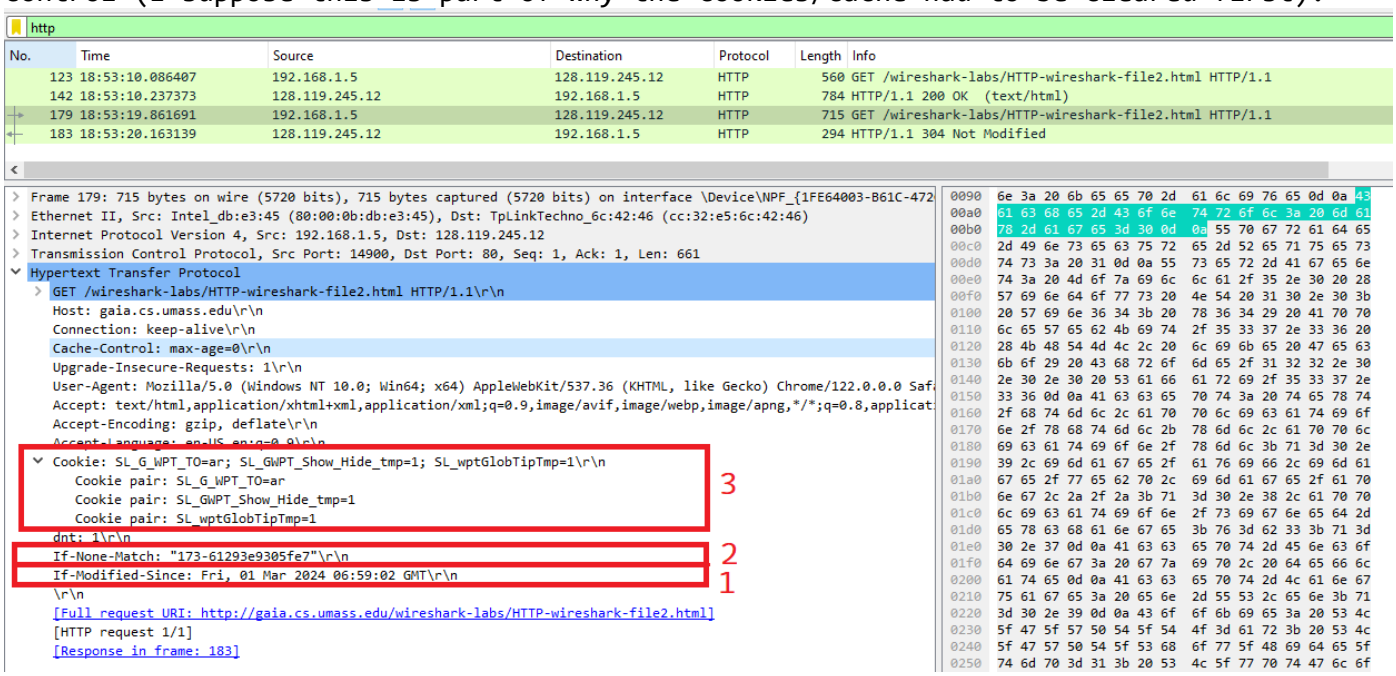
9. Inspect the contents of the server first response. Did the server explicitly return the contents of the file? How can you tell?

Yes, the server explicitly returned the contents of the file after the first HTTP request. I am able to tell this because of the Line-based text data in the first OK response to the GET.



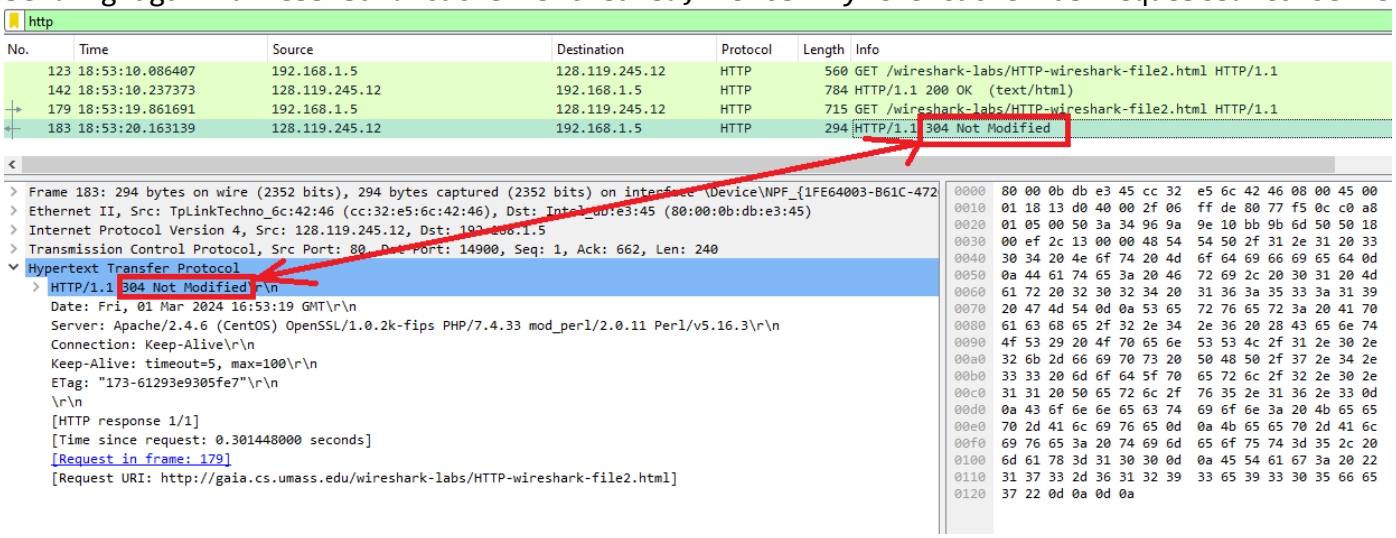
10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

There is an “IF-MODIFIED-SINCE:” line in the second HTTP GET request. The information that followed the “IF-MODIFIED-SINCE:” header represents the date it was modified that was not present in the first HTTP GET. Also, the information that are before the “IF-MODIFIED-SINCE” (and aren’t exist in the first http get request) are: a match query, cookies and cache-control (I suppose this is part of why the cookies/cache had to be cleared first).



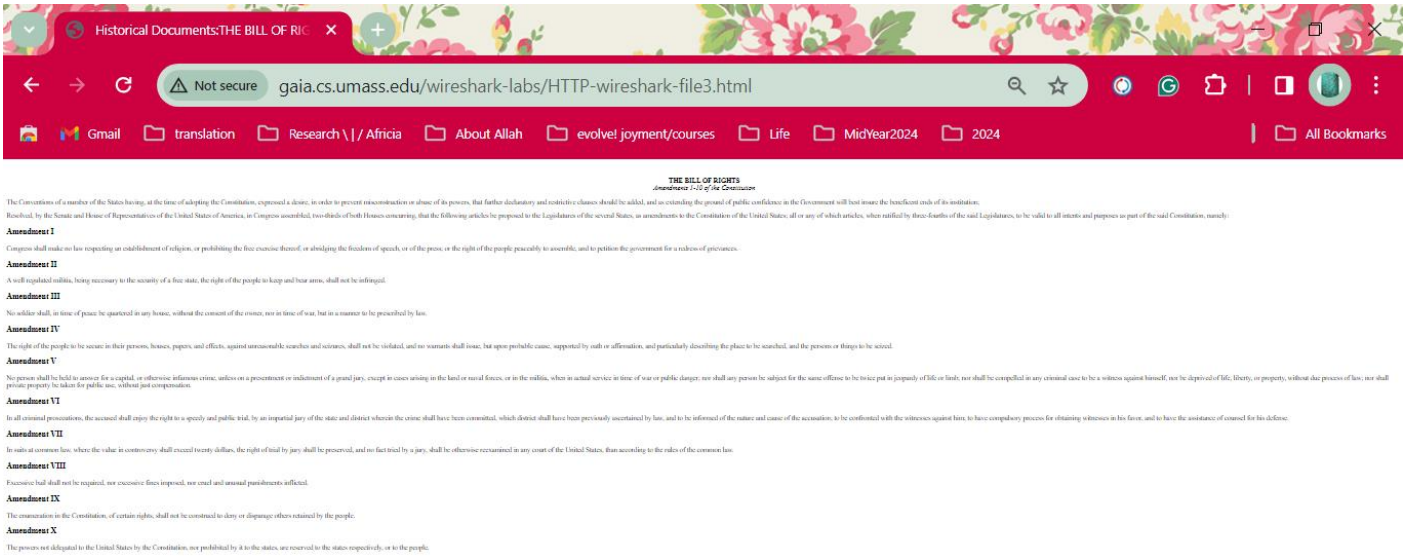
11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

304 Not Modified. No, the server didn’t return the content of the data file again. The 304 Not Modified HTTP server response code (after a conditional GET) indicates that the requested resource has not been modified since the last time it was loaded, and there's no need to transfer it again. Since some data packets is already stored in the cache, it does not need sending again unless said cache is cleared, hence why the cache was requested to be removed.



Section 3: Retrieving Long Documents

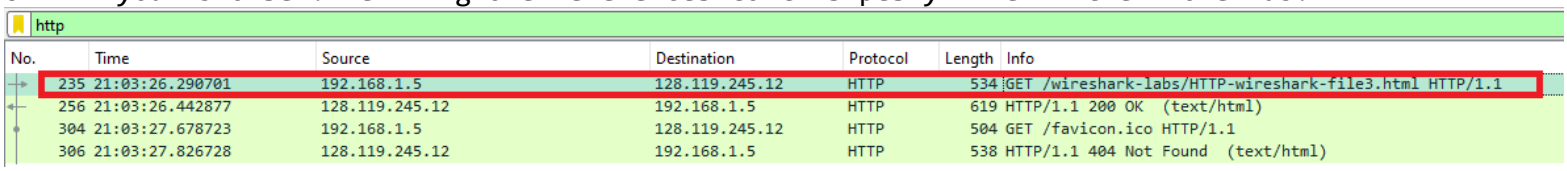
The Task:



The Questions & Answers:

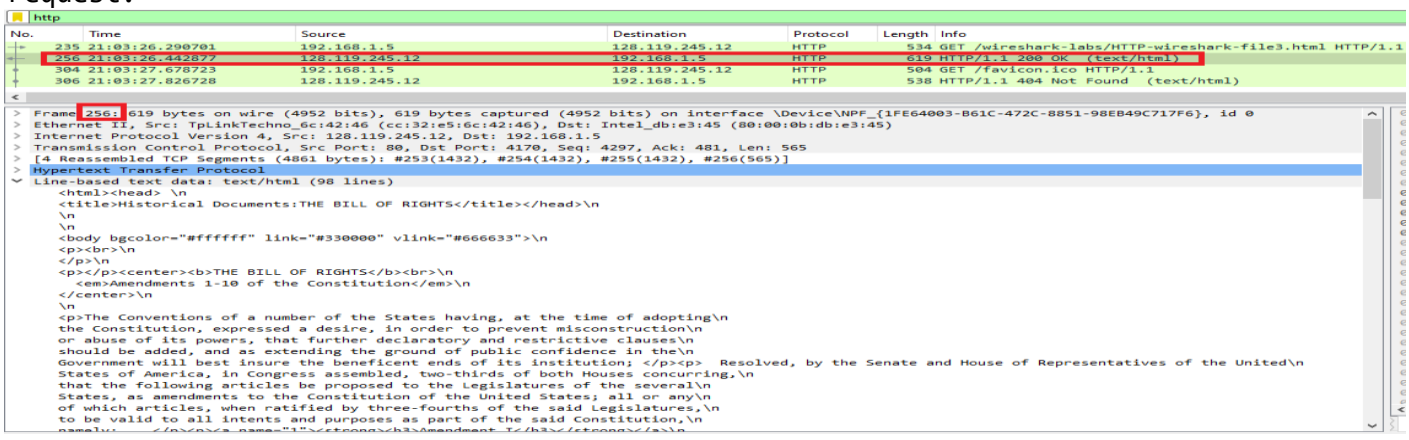
12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

My browser sent only one HTTP GET request message. Packet number 235 contained the GET HTTP request for the Bill or Rights. Because we should ignore any HTTP GET and response for favicon.ico. If we see a reference to this file, it is the browser automatically asking the server if it (the server) has a small icon file that should be displayed next to the displayed URL in your browser. We'll ignore references to this pesky file in the whole lab.



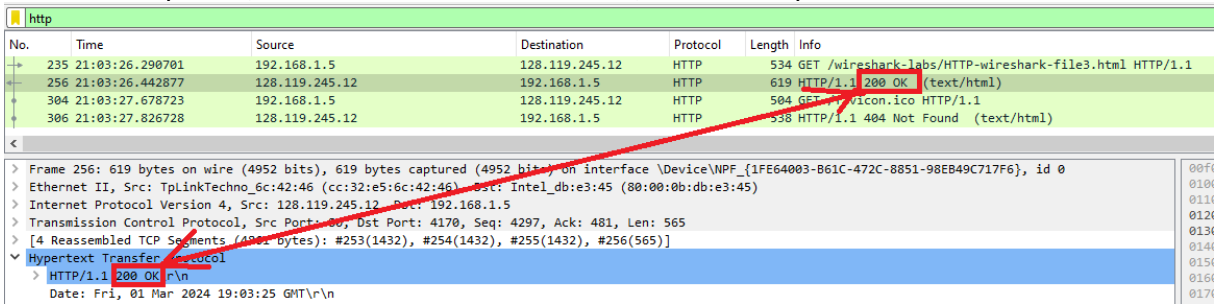
13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

Packet 256 contains the status code and phrase associated with the response to the HTTP GET request.



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

In the response, the status code is 200, and the phrase is an OK.

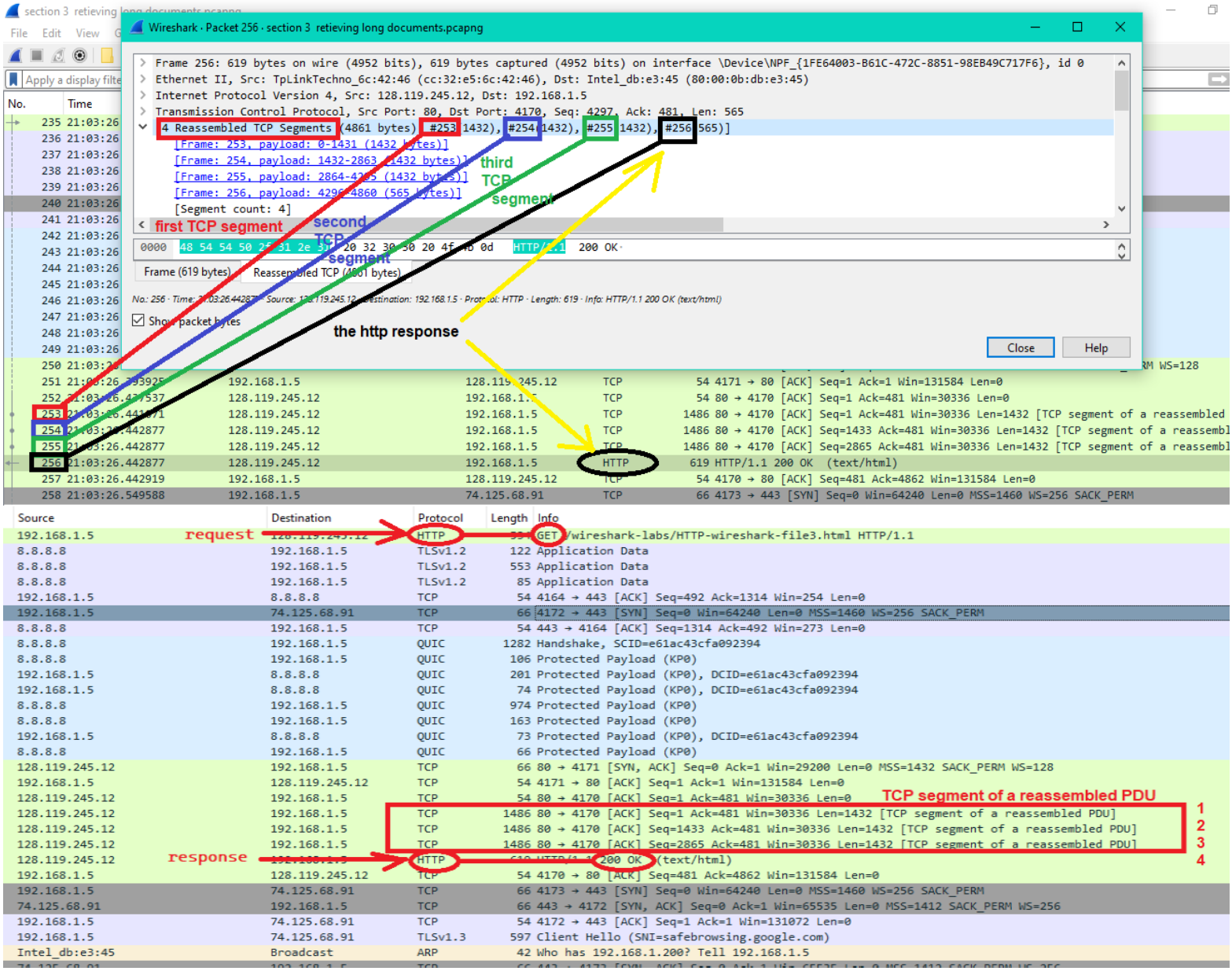


15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

There are:

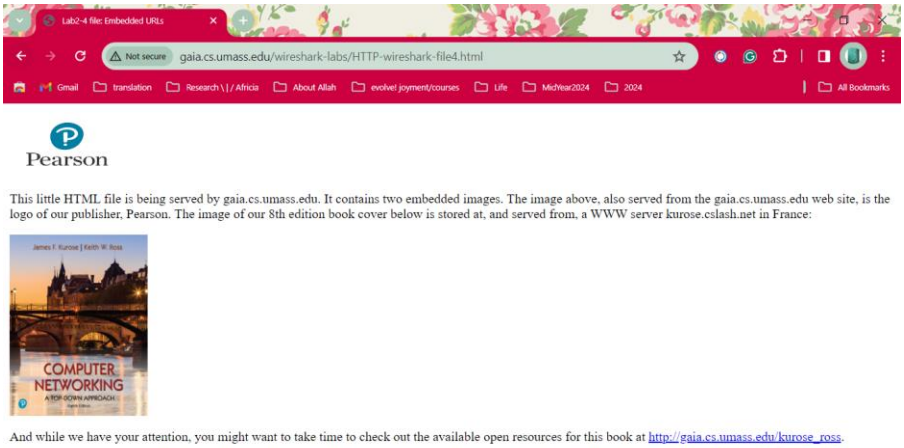
- 3 data-containing TCP segments “**TCP segment of a reassembled PDU** “.
- The fourth one is the http response itself.

So, the total count is 4 that are needed to carry the single HTTP response and the text of the Bill of Rights. As shown in the below screenshots:



Section 4: HTML Documents with Embedded Objects

The Task:



The Questions & Answers:

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

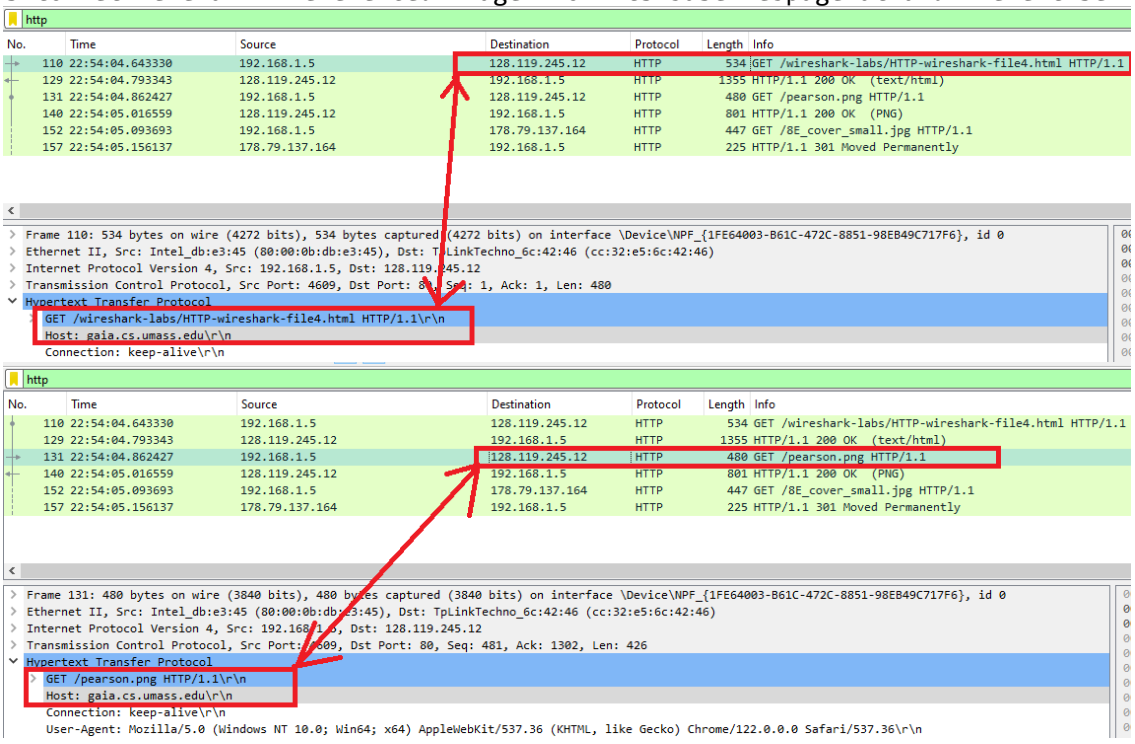
There are 3 HTTP GET request messages that were sent by my browser (whose IP address is = 192.168.1.15) and were sent to 2 different IP addresses which are 128.119.245.12 (Host: gaia.cs.umass.edu), 178.79.137.164 (Host: kurose.cslash.net).

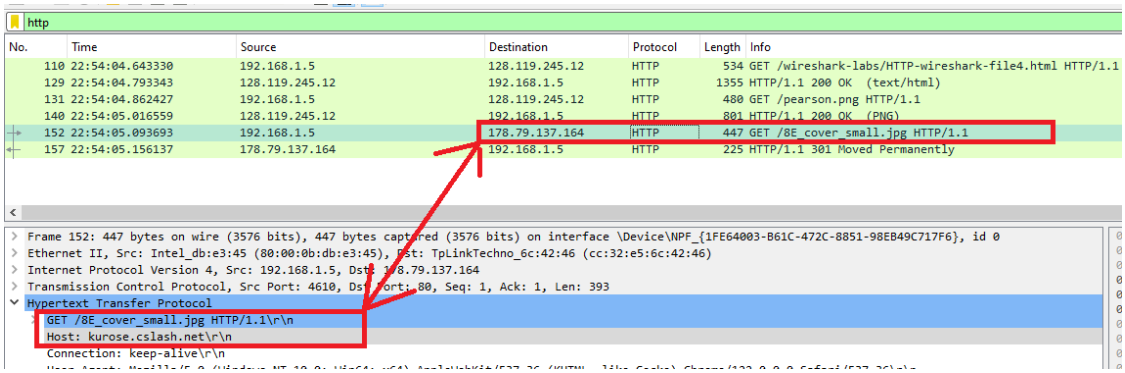
http

No.	Time	Source	Destination	Protocol	Length	Info
110	22:54:04.643330	192.168.1.5	128.119.245.12	HTTP	534	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
129	22:54:04.793343	128.119.245.12	192.168.1.5	HTTP	1355	HTTP/1.1 200 OK (text/html)
131	22:54:04.862427	192.168.1.5	128.119.245.12	HTTP	480	GET /pearson.png HTTP/1.1
140	22:54:05.016559	128.119.245.12	192.168.1.5	HTTP	801	HTTP/1.1 200 OK (PNG)
152	22:54:05.093693	192.168.1.5	178.79.137.164	HTTP	447	GET /8E_cover_small.jpg HTTP/1.1
157	22:54:05.156137	178.79.137.164	192.168.1.5	HTTP	225	HTTP/1.1 301 Moved Permanently

As following:

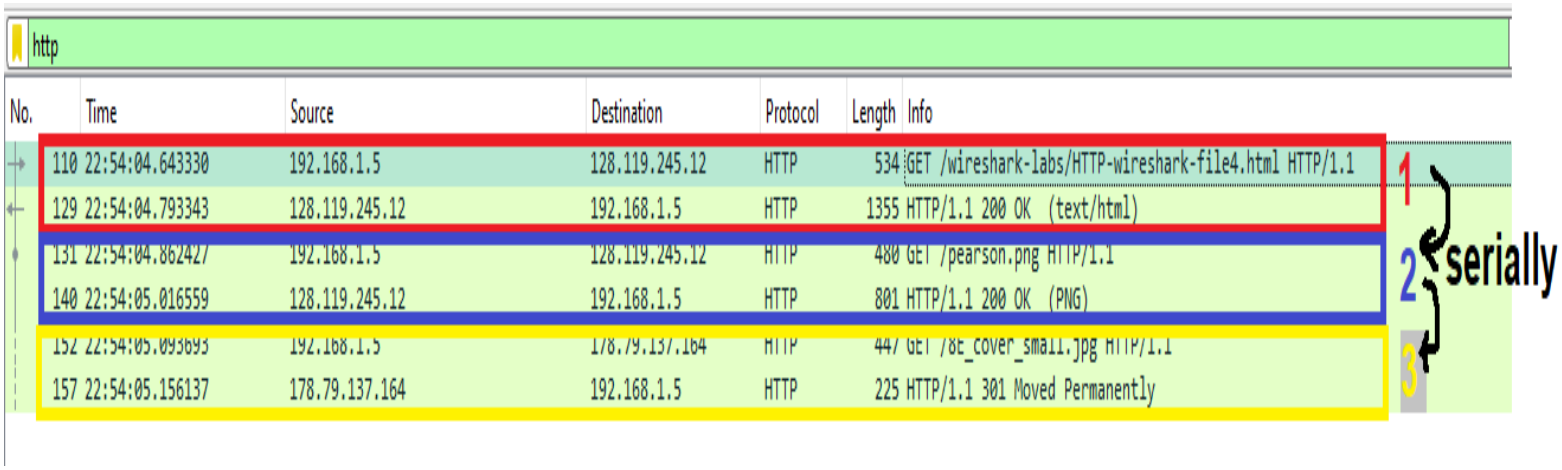
- 1st GET is to retrieve the initial webpage.
- 2nd GET is to retrieve a referenced image from its base webpage at the same server.
- 3rd GET is to retrieve a 2nd referenced image from its base webpage at a different server.





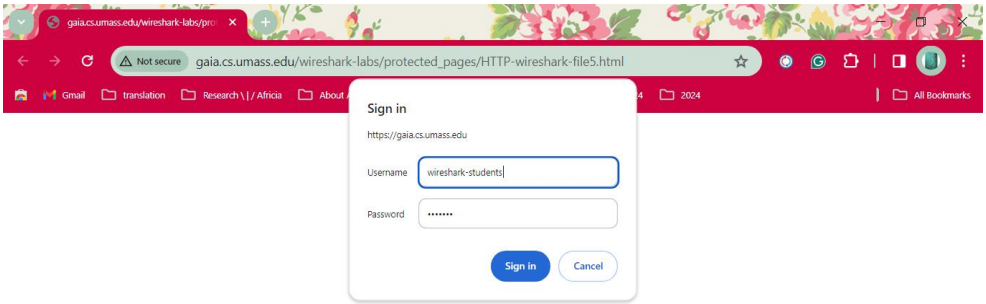
17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

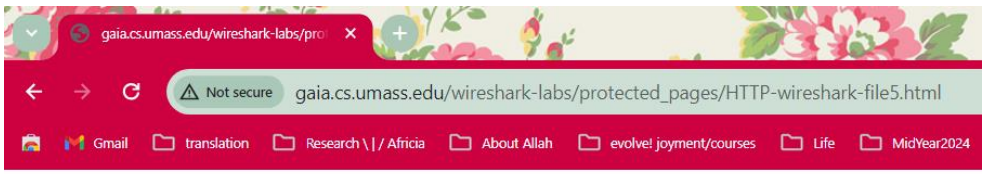
they were downloaded serially. Serially operation because they're on different time stamps as the GET is sent out after an response is seen. So, the first image appeared, then the second image started to load, although the second image was hosted in another website!



Section 5: HTTP Authentication

The Task:





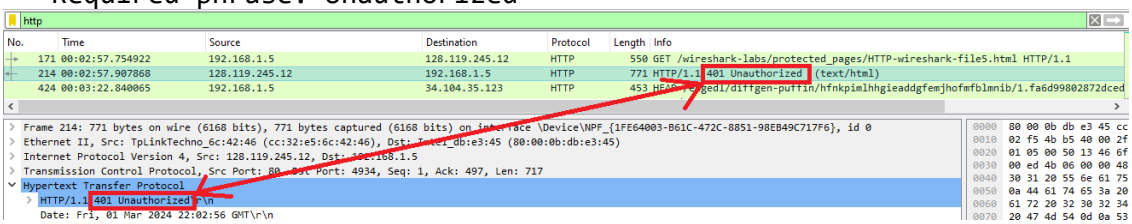
This page is password protected! If you're seeing this, you've downloaded the page correctly
Congratulations!

The Questions & Answers:

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

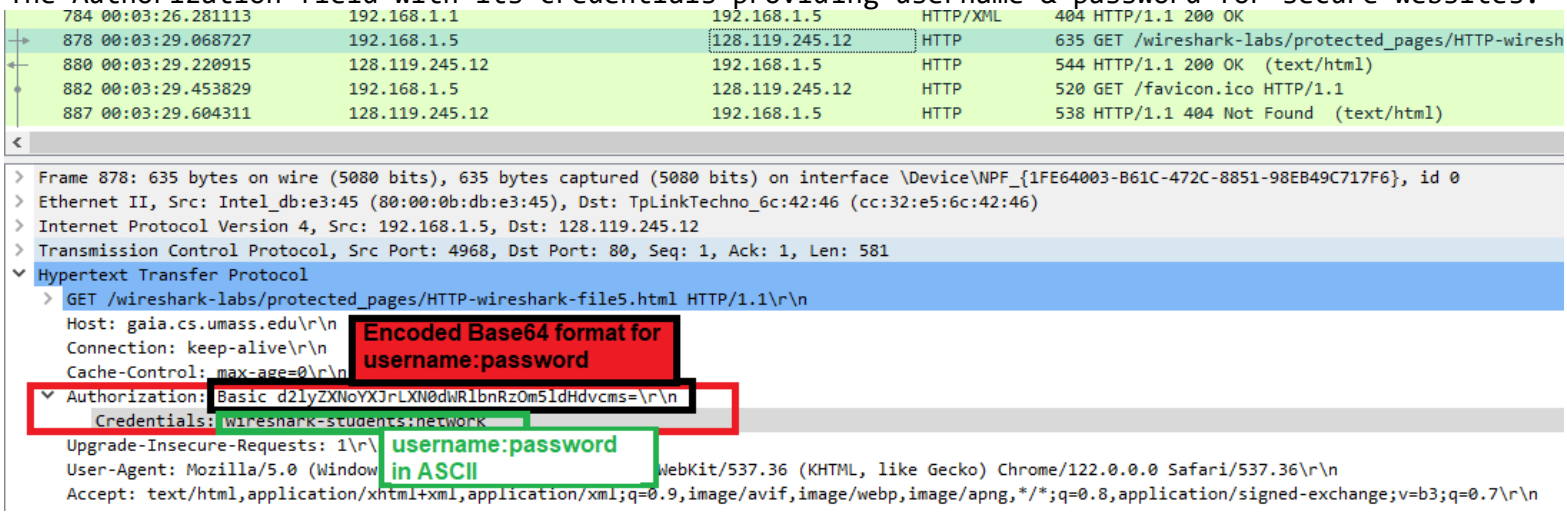
The server's response to my HTTP GET message is a 401 Unauthorized :

- Required status code:401
- Required phrase: 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?

The Authorization field with its credentials providing username & password for secure websites.




thank you


BYE