

CSE4074 Homework 1

Wireshark Lab: HTTP Solutions

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

Yes, my browser is running HTTP version 1.1.

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

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

Accept-Language: en-US,en;q=0.9,tr-TR;q=0.8,tr;q=0.7\r\n

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

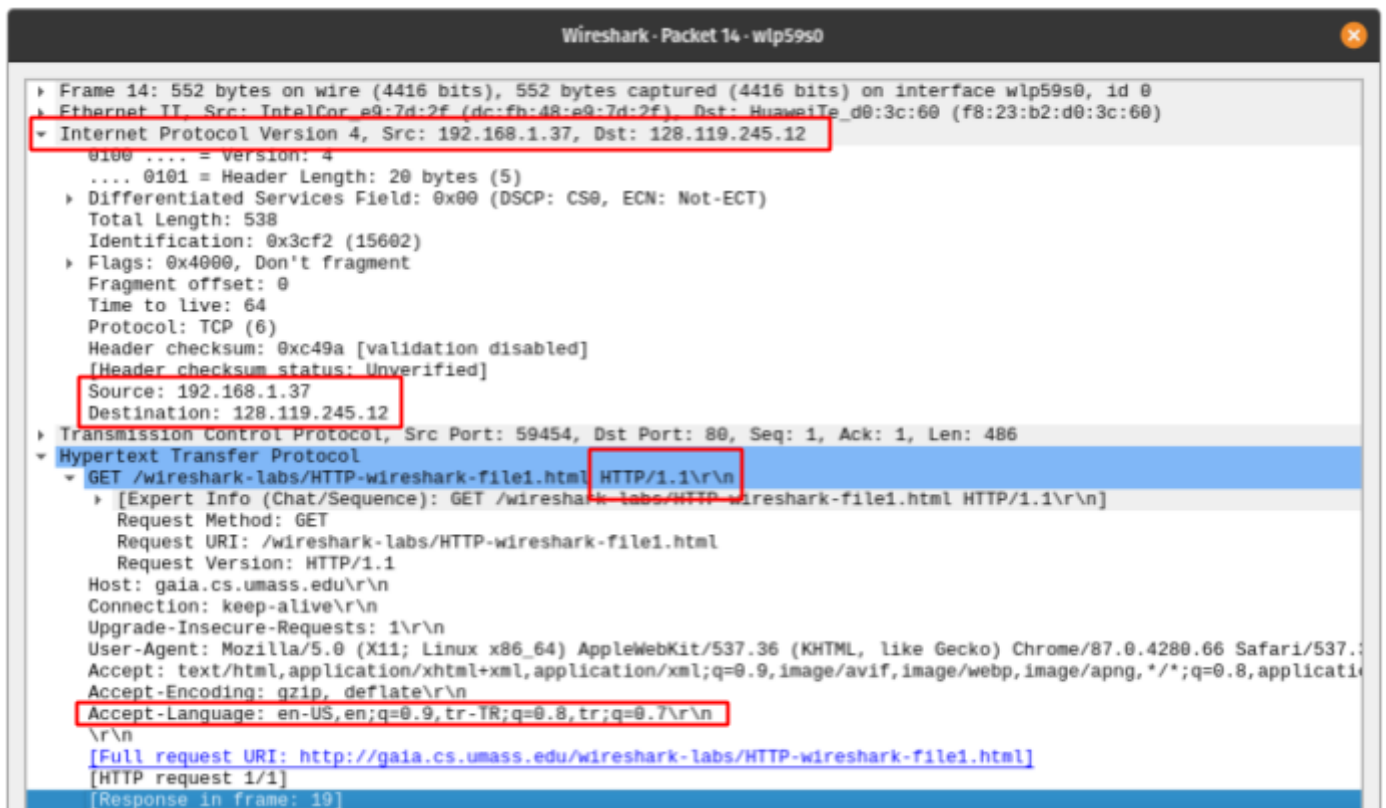
My IP address is 192.168.1.37.

IP address of gaia.cs.umass.edu.server is 128.119.245.12

Internet Protocol Version 4, Src:192.168.1.37, Dst:128.119.245.12

Source: 192.168.1.37

Destination: 128.119.245.12



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

Status code returned 200 from server to my browser.

Status Code: 200

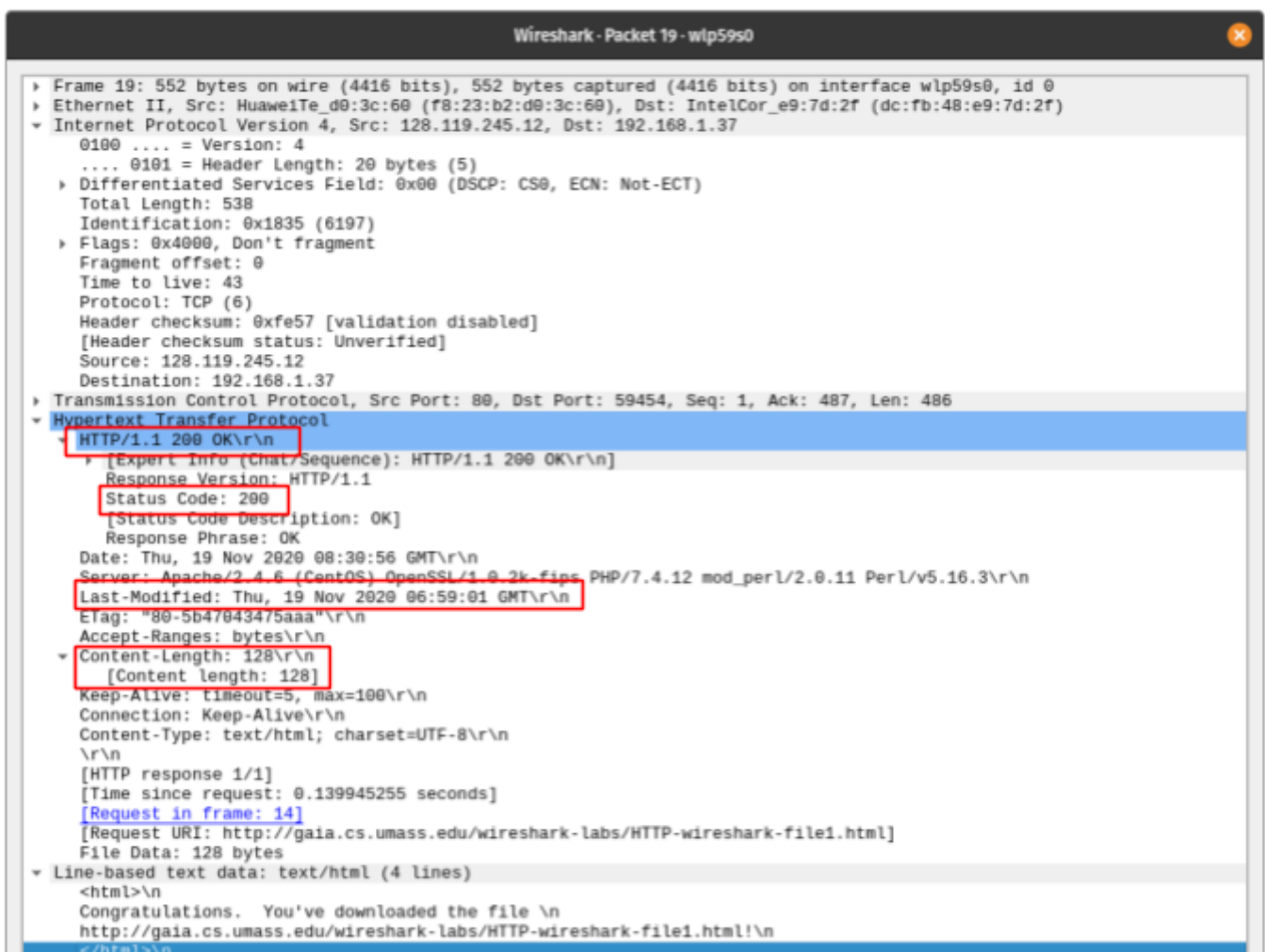
5. When was the HTML file that you are retrieving last modified at the server?

Last-Modified: Thu, 19 Nov 2020 06:59:01 GMT\r\n

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

Content-Length: 128\r\n

[Content length: 128]



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. I don't see any in the HTTP Message.

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?

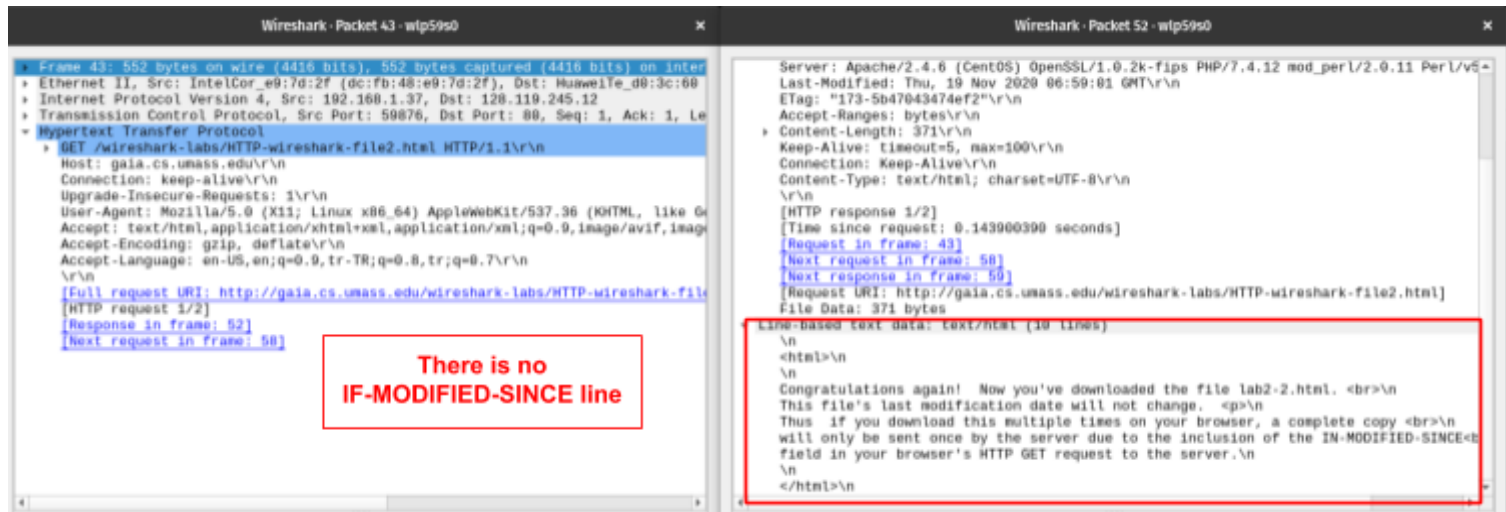
No, I don't see the “IF-MODIFIED-SINCE” line in the contents of the first HTTP GET request.

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

Text returned in response to the first GET.

Line-based text data: text/html (10 lines)

```
\n
<html>\n
\n
Congratulations again! Now you've downloaded the file lab2-2.html.<br>\n
This file's last modification date will not change.<p>\n
Thus if you download this multiple times on your browser, a complete copy<br>\n
will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
field in your browser's HTTP GET request to the server.\n
\n
</html>\n
```



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?

Yes, I see now.

If-Modified-Since: Thu, 19 Nov 2020 06:59:01 GMT\r\n

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 file has not been modified. So the text of the file is not returned in the HTTP message.



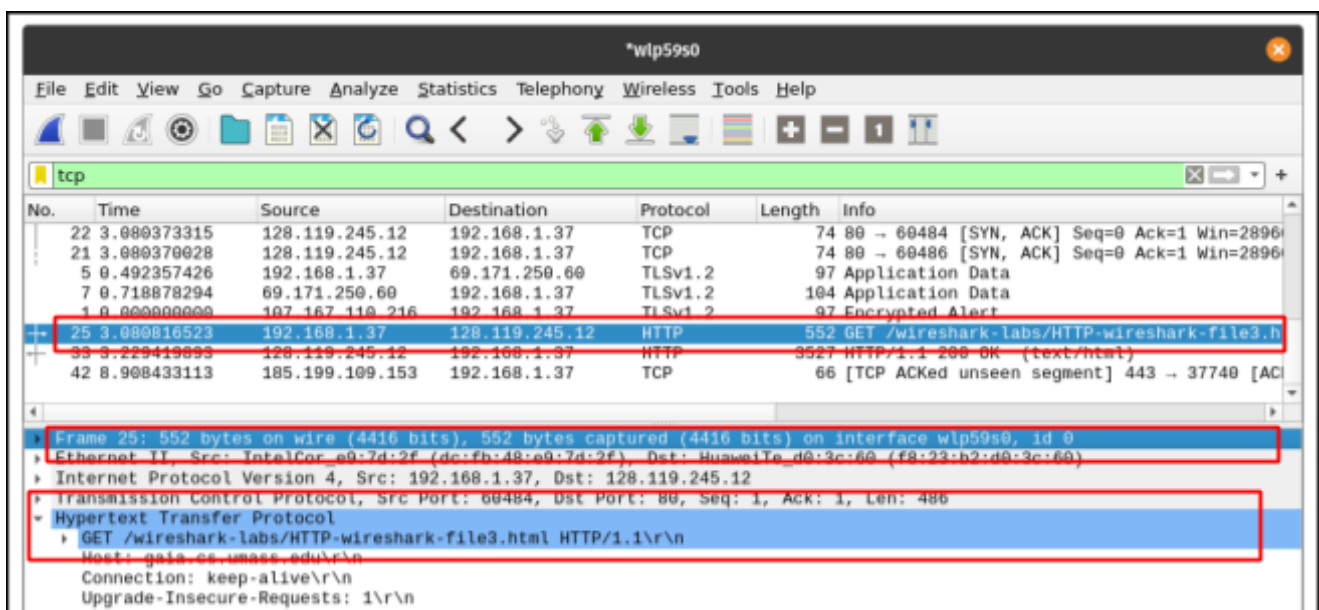
12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

There is one HTTP GET request. Its packet number is 25.

Frame 25:

Hypertext Transfer Protocol

GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1\r\n



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

Packet 31.

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

HTTP/1.1 200 OK

The image shows a Wireshark packet capture window titled "wlp59s0". The packet list pane shows several packets, with packet 31 highlighted in blue. Packet 31 is a TCP ACK from 128.119.245.12 to 192.168.1.37, sequence number 1466, acknowledgment number 60484, and window size 30080. The packet details pane shows the "HTTP" section with the status line "HTTP/1.1 200 OK". The packet bytes pane shows the raw data of the packet, with the status line "HTTP/1.1 200 OK" highlighted in red. The packet list pane also shows a red box around packet 31. The packet details pane shows the status line "HTTP/1.1 200 OK" and the response body "Date: Thu, 19 Nov 2020 06:59:01 GMT". The packet bytes pane shows the raw data of the packet, with the status line "HTTP/1.1 200 OK" highlighted in red.

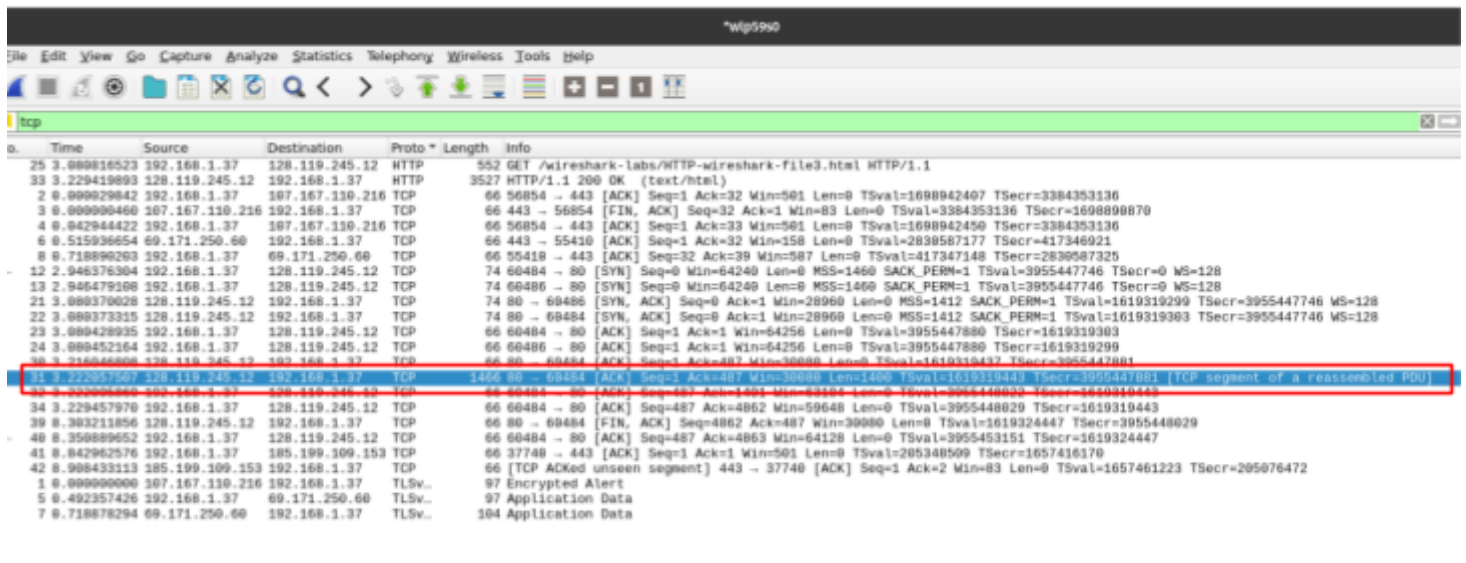
Wireshark - Follow TCP Stream (tcp.stream eq 2) - wlp59s0

GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
Host: gaia.cs.umass.edu
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36
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=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,tr-TR;q=0.8,tr;q=0.7

HTTP/1.1 200 OK
Date: Thu, 19 Nov 2020 06:59:01 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.12 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Thu, 19 Nov 2020 06:59:01 GMT
ETag: "1194-5b4704346ed49"

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

There is one packet. Packet 31.



The image shows a Wireshark packet capture window titled '*wlp5950'. The packet list on the left shows packet 31 selected, which is an HTTP GET request. The packet details pane on the right shows the structure of the packet: Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
25	3.088816523	192.168.1.37	128.119.245.12	HTTP	552	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
33	3.229419893	128.119.245.12	192.168.1.37	HTTP	3527	HTTP/1.1 200 OK (text/html)
2	0.000029842	192.168.1.37	107.167.130.216	TCP	66	56854 → 443 [ACK] Seq=1 Ack=32 Win=501 Len=0 TSval=1698942407 TSecr=3384353136
3	0.000000460	107.167.130.216	192.168.1.37	TCP	66	443 → 56854 [FIN, ACK] Seq=32 Ack=1 Win=83 Len=0 TSval=3384353136 TSecr=1698989870
4	0.042944422	192.168.1.37	107.167.130.216	TCP	66	56854 → 443 [ACK] Seq=1 Ack=32 Win=501 Len=0 TSval=1698942450 TSecr=3384353136
6	0.515936654	69.171.250.60	192.168.1.37	TCP	66	443 → 55410 [ACK] Seq=1 Ack=32 Win=158 Len=0 TSval=2838587177 TSecr=417346921
8	0.718890263	192.168.1.37	69.171.250.60	TCP	66	55410 → 443 [ACK] Seq=32 Ack=39 Win=587 Len=0 TSval=417347148 TSecr=2838587325
12	2.946376304	192.168.1.37	128.119.245.12	TCP	74	60484 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3955447746 TSecr=0 WS=128
13	2.946479108	192.168.1.37	128.119.245.12	TCP	74	80 → 60484 [SYN, ACK] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3955447746 TSecr=0 WS=128
21	3.088370028	128.119.245.12	192.168.1.37	TCP	74	80 → 60484 [SYN, ACK] Seq=0 Ack=1 Min=28960 Len=0 MSS=1412 SACK_PERM=1 TSval=1619319299 TSecr=3955447746 WS=128
22	3.088373315	128.119.245.12	192.168.1.37	TCP	74	80 → 60484 [SYN, ACK] Seq=0 Ack=1 Min=28960 Len=0 MSS=1412 SACK_PERM=1 TSval=1619319299 TSecr=3955447746 WS=128
23	3.088428935	192.168.1.37	128.119.245.12	TCP	66	60484 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3955447880 TSecr=1619319303
24	3.088452164	192.168.1.37	128.119.245.12	TCP	66	60484 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3955447880 TSecr=1619319299
26	3.216846808	128.119.245.12	192.168.1.37	TCP	66	80 → 60484 [ACK] Seq=1 Ack=487 Min=30080 Len=0 TSval=1619319437 TSecr=3955447880
31	3.229419893	128.119.245.12	192.168.1.37	HTTP	1000	200 OK (text/html) [ACK] Seq=1 Ack=487 Min=30080 Len=0 TSval=1619319437 TSecr=3955447880 [TCP segment of a reassembled PDU]
33	3.229419893	128.119.245.12	192.168.1.37	TCP	66	60484 → 80 [ACK] Seq=1 Ack=487 Min=30080 Len=0 TSval=1619319437 TSecr=3955447880
34	3.229457976	192.168.1.37	128.119.245.12	TCP	66	60484 → 80 [ACK] Seq=487 Ack=4862 Min=59648 Len=0 TSval=3955448829 TSecr=1619319443
39	8.363211856	128.119.245.12	192.168.1.37	TCP	66	80 → 60484 [FIN, ACK] Seq=4862 Ack=487 Win=30080 Len=0 TSval=1619324447 TSecr=3955448829
40	8.358889652	192.168.1.37	128.119.245.12	TCP	66	60484 → 80 [ACK] Seq=487 Ack=4863 Win=64128 Len=0 TSval=3955453151 TSecr=1619324447
41	8.842962576	192.168.1.37	185.199.309.153	TCP	66	37748 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=205348509 TSecr=1657416170
42	8.908433113	185.199.309.153	192.168.1.37	TCP	66	[TCP ACKed unseen segment] 443 → 37748 [ACK] Seq=1 Ack=2 Min=83 Len=0 TSval=1657461223 TSecr=205076472
1	0.000000000	107.167.130.216	192.168.1.37	TLSv...	97	Encrypted Alert
5	0.492357426	192.168.1.37	69.171.250.60	TLSv...	97	Application Data
7	0.718878294	69.171.250.60	192.168.1.37	TLSv...	104	Application Data

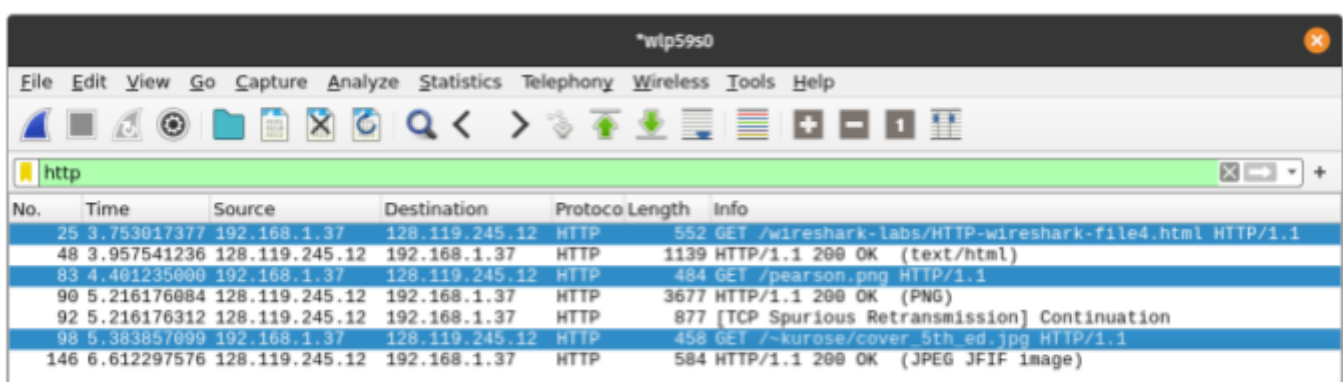
16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

There are three HTTP GET messages. Each of these three GET messages are sent to the same IP addresses.

25 - 128.119.245.12 - HTTP GET - /wireshark-labs/HTTP-wireshark-file4.html
83 - 128.119.245.12 - HTTP GET - /pearson.png
25 - 128.119.245.12 - HTTP GET - /~kurose/cover_5th_ed.jpg

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

The downloads occurred in parallel. Note that the two GET messages for the images are in packets 83 and 98.



The image shows a Wireshark packet capture window titled '*wlp5950'. The packet list on the left shows several HTTP GET requests. The packet details pane on the right shows the structure of the packet: Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
25	3.753017377	192.168.1.37	128.119.245.12	HTTP	552	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
48	3.957541236	128.119.245.12	192.168.1.37	HTTP	1139	HTTP/1.1 200 OK (text/html)
83	4.401235000	192.168.1.37	128.119.245.12	HTTP	484	GET /pearson.png HTTP/1.1
90	5.216176084	128.119.245.12	192.168.1.37	HTTP	3677	HTTP/1.1 200 OK (PNG)
92	5.216176312	128.119.245.12	192.168.1.37	HTTP	877	[TCP Spurious Retransmission] Continuation
98	5.383857099	192.168.1.37	128.119.245.12	HTTP	458	GET /~kurose/cover_5th_ed.jpg HTTP/1.1
146	6.612297576	128.119.245.12	192.168.1.37	HTTP	584	HTTP/1.1 200 OK (JPEG JFIF image)

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

Packet 804 in the trace contains the first GET and packet 825 contains the REPLY. The server's in packet 825 is: 401 Authorization Required

HTTP/1.1 401 Unauthorized\r\n

Status Code: 401

[Status Code Description: Unauthorized]

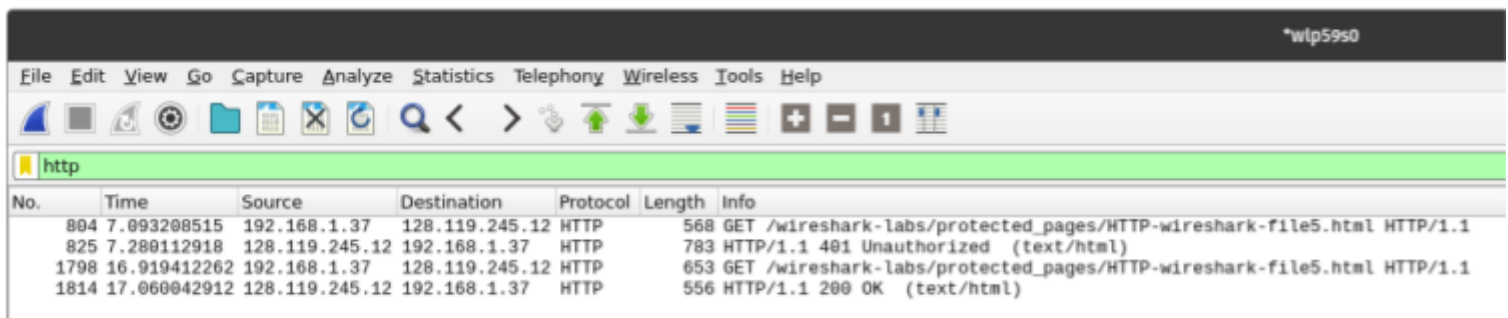
Response Phrase: Unauthorized

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

The HTTP GET includes the Authorization.

Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcm5=\r\n

Credentials: wireshark-students:network



The screenshot shows the Wireshark interface with a packet capture of an HTTP transaction. The packet list pane shows four packets. Packet 825 is selected, showing an HTTP 401 Unauthorized response from 128.119.245.12 to 192.168.1.37. The packet details pane shows the HTTP response structure with status code 401 and phrase 'Unauthorized'.

No.	Time	Source	Destination	Protocol	Length	Info
804	7.093208515	192.168.1.37	128.119.245.12	HTTP	568	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
825	7.280112918	128.119.245.12	192.168.1.37	HTTP	783	HTTP/1.1 401 Unauthorized (text/html)
1798	16.919412262	192.168.1.37	128.119.245.12	HTTP	653	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
1814	17.060042912	128.119.245.12	192.168.1.37	HTTP	556	HTTP/1.1 200 OK (text/html)

150116884
Esra Polat