

CS315: Lab Assignment 3

B Siddharth Prabhu

200010003@iitdh.ac.in

17 January 2023

1 Answers for Task 1: The Basic HTTP GET/response interaction

The following screenshot depicts the observed output records. Below it, some questions regarding the same are answered.

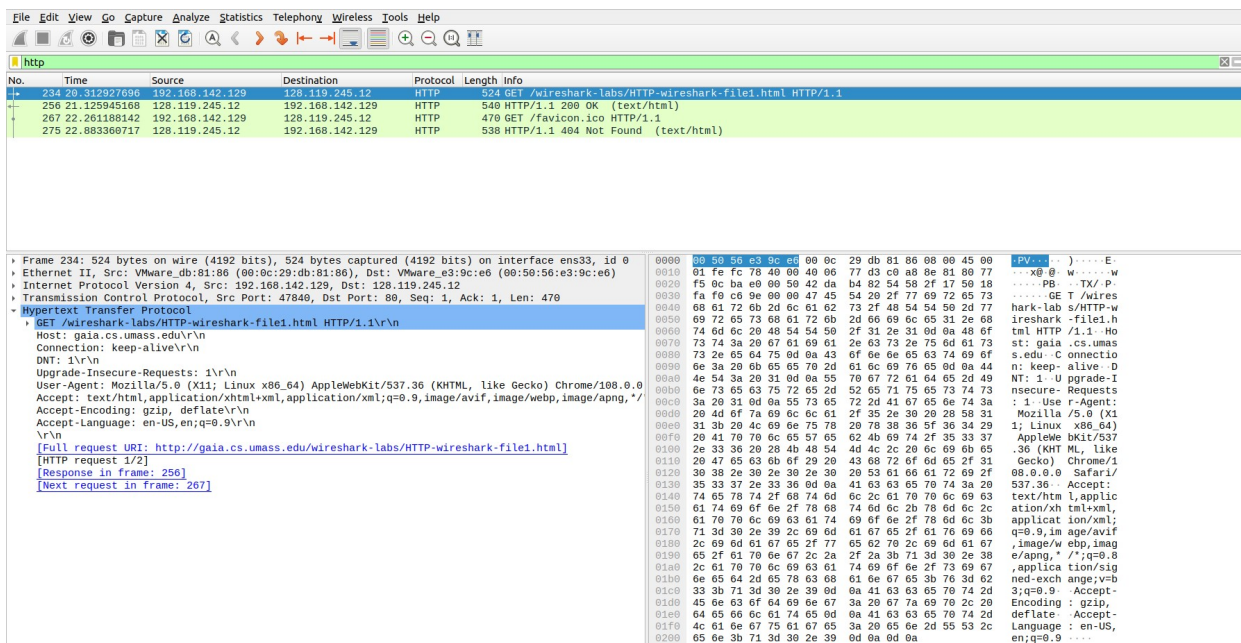


Figure 1: Screenshot of records obtained

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

- The browser is running HTTP version 1.1. This is observable in the record with ID 234.
- The server is running HTTP version 1.1. This is observable in the record with ID 256.

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

In the screenshot, within the description of HTTP, we find the accepted languages in the `Accept-Language` attribute. It is `en-US`, which is American English. This could be due to browser settings.

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

- The IP address of my computer (the VM) is **192.168.142.129**. The reason for this is that I'm running the browser (and Wireshark) on an Ubuntu VM with Network settings set to NAT (Network Address Translation). IP addresses beginning with 192.168 are for private networks, and in this case the private network is formed by the VM and the host system, via a network adapter. Hence, to communicate with the internet, this would take 1 hop more, than if this was run on the host.
- The IP address of the gaia.cs.umass.edu server is **128.119.245.12**.

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

As observed in the below screenshot, the status code returned from the server to the browser is **200** (which means status OK). This can be found in the header part of the HTTP Response packet.

```
▶ Frame 256: 540 bytes on wire (4320 bits), 540 bytes captured (4320 bits) on interface ens33, id 0
▶ Ethernet II, Src: VMware_e3:9c:e6 (00:50:56:e3:9c:e6), Dst: VMware_db:81:86 (00:0c:29:db:81:86)
▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.142.129
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 47840, Seq: 1, Ack: 471, Len: 486
▼ 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: Tue, 17 Jan 2023 04:27:19 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.30 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Mon, 16 Jan 2023 06:59:01 GMT\r\n
    ETag: "80-5f25c1e3c8396"\r\n
    Accept-Ranges: bytes\r\n
```

Figure 2: Screenshot of status code in response

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

As visible in the above figure, the **Last-Modified** field reads: **Mon, 16 Jan 2023 06:59:01 GMT**. This is possible due to the packet being modified along the way. Sometimes, the Last-Modified field comes to be a date in 2016, and this may occur if the file is not modified at all along the way.

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

As observed in the below screenshot, **128 bytes** of content are being returned to the browser.

```
▼ Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    Date: Tue, 17 Jan 2023 04:27:19 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.30 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Mon, 16 Jan 2023 06:59:01 GMT\r\n
    ETag: "80-5f25c1e3c8396"\r\n
    Accept-Ranges: bytes\r\n
  Content-Length: 128\r\n
    [Content length: 128]
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
```

Figure 3: Screenshot of content bytes in response

(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 seems to exactly match up with what is shown in the packet-listing window.

2 Answers for Task 2: The HTTP CONDITIONAL GET/response interaction

The following screenshot depicts the observed output records on loading and refreshing the page. Below it, some questions regarding the same are answered.

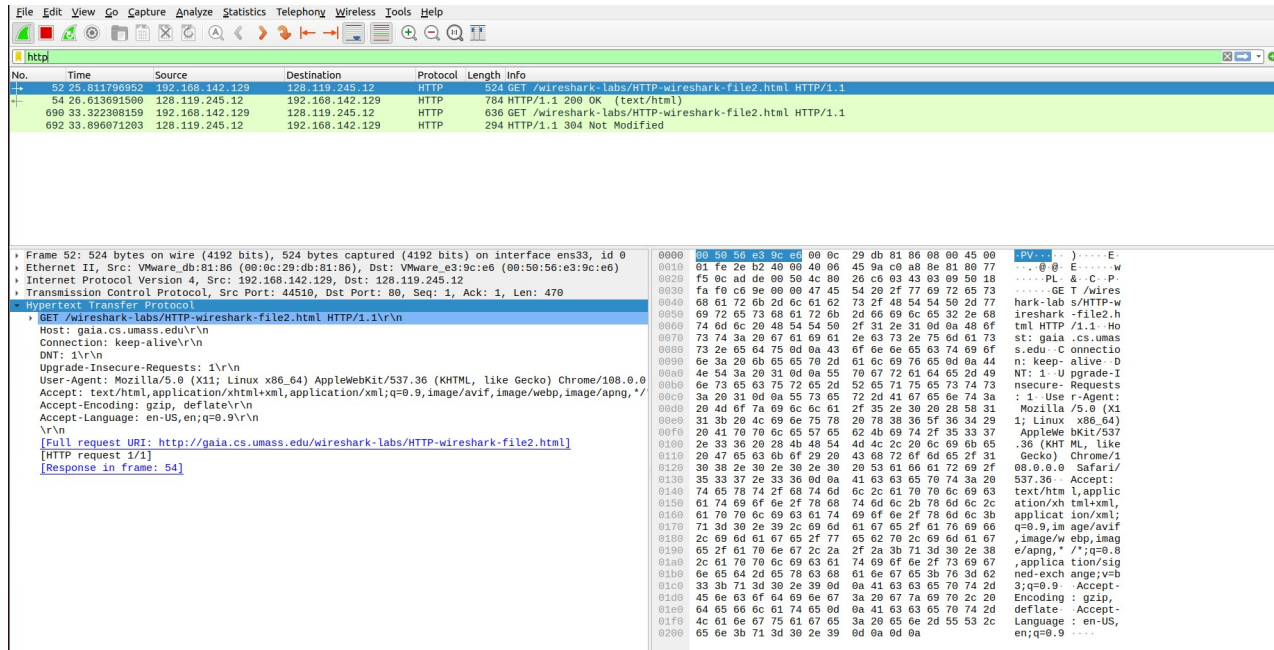


Figure 4: Screenshot of records obtained

(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?

No. IF-MODIFIED-SINCE: line is not present in the contents of the first HTTP GET request from the browser to the server, as seen in the below screenshot.

```
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  DNT: 1\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
  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/HTTP-wireshark-file2.html]
  [HTTP request 1/1]
  [Response in frame: 54]
```

Figure 5: Lack of IF-MODIFIED-SINCE field

(2) Inspect the contents of the server 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. We observe the section titled “Line-Based Text Data”, which shows what the server sent back to the browser (which is exactly what the website displays). The contents of this section are shown in the below screenshot.

```
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
```

Figure 6: Contents returned by 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?

Yes. **IF-MODIFIED-SINCE:** line is present in the contents of the second HTTP GET request from the browser to the server, as seen in the below screenshot. It is followed by the time when the page was last accessed/modified by the browser.

```
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.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
  DNT: 1\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9\r\n
  If-None-Match: "173-5f25c1e3c77de"\r\n
  If-Modified-Since: Mon, 16 Jan 2023 06:59:01 GMT\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
  [HTTP request 1/1]
  [Response in frame: 692]
```

Figure 7: Presence of IF-MODIFIED-SINCE field

(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.

- The HTTP status code and phrase returned from the server in response to the second HTTP GET are **304** and **Not Modified** respectively. The same can be observed in the screenshot present at the top of the next page.
- Also, there is no section titled “Line-Based Text Data” here, so the server doesn’t explicitly return the file contents in this case.

```

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: Tue, 17 Jan 2023 05:24:48 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.30 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Mon, 16 Jan 2023 06:59:01 GMT\r\n
    ETag: "173-5f25c1e3c77de"\r\n
    Accept-Ranges: bytes\r\n

```

Figure 8: Status Code and Phrase from second HTTP GET

3 Answers for Task 3: Retrieving Long Documents

The following screenshot depicts the observed output records on retrieving a long document. Below it, some questions regarding the same are answered.

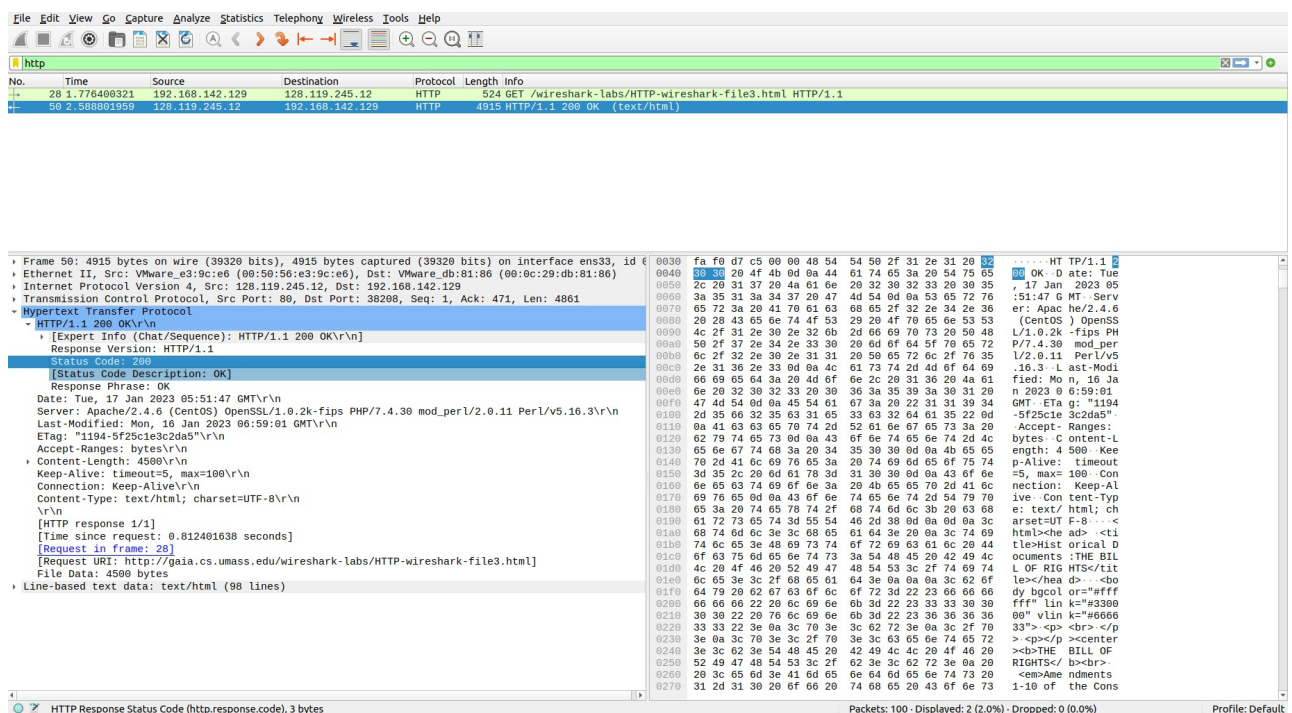


Figure 9: Screenshot of records obtained

(1) 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?

- The browser sent ONE HTTP GET request message, as visible in the above screenshot.
- The packet number 28 in the trace contains the GET message 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?

The packet number 50 in the trace contains the status code and phrase associated with the response to the HTTP GET request

(3) What is the status code and phrase in the response?

As visible in the second image on the previous page, the HTTP status code and phrase in the response are **200** and **OK** respectively.

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

As visible in the below screenshot, there are 2 TCP segments needed to carry the single HTTP response. The number of such segments seems to vary in each packet capturing (even after clearing cache). This would be based on congestion of the network.

```
[2 Reassembled TCP Segments (4861 bytes): #40(4380), #42(481)]
[Frame: 40, payload: 0-4379 (4380 bytes)]
[Frame: 42, payload: 4380-4860 (481 bytes)]
[Segment count: 2]
[Reassembled TCP length: 4861]
[Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a4461746553a205475652c203137204a616e2032...
```

Figure 10: Screenshot of TCP segments

4 Answers for Task 4: HTML Documents with Embedded Objects

The following screenshot depicts the observed output records on retrieving HTML documents with embedded objects. Below it, some questions regarding the same are answered.

No.	Time	Source	Destination	Protocol	Length	Info
93	14.885473938	192.168.142.129	128.119.245.12	HTTP	524	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
184	15.781065436	128.119.245.12	192.168.142.129	HTTP	1355	HTTP/1.1 200 OK (text/html)
186	16.02225533	178.79.137.164	128.119.245.12	HTTP	437	GET /8E_cover_small.jpg HTTP/1.1
121	16.545729635	192.168.142.129	178.79.137.164	HTTP	437	GET /8E_cover_small.jpg HTTP/1.1
129	16.644579178	128.119.245.12	192.168.142.129	HTTP	3665	HTTP/1.1 200 OK (PNG)
521	17.145859587	178.79.137.164	192.168.142.129	HTTP	225	HTTP/1.1 301 Moved Permanently

Frame 186: 470 bytes on wire (3760 bits), 470 bytes captured (3760 bits) on interface ens33, id 0
Ethernet II, Src: VMware_b8:18:86 (00:0c:29:db:81:86), Dst: VMware_e3:9c:e6 (00:50:56:e3:9c:e6)
Internet Protocol Version 4, Src: 192.168.142.129, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 45192, Dst Port: 80, Seq: 471, Ack: 1302, Len: 416
Hypertext Transfer Protocol
0000 00 50 56 e3 9c e6 00 0c 29 db 81 86 08 00 45 00 PV.....):...E.
0010 01 c8 27 20 40 00 40 06 4d 62 c0 a8 0e 81 80 77 ...' @ @: Mb...W
0020 f5 0c b0 88 00 50 58 3b e4 70 47 56 0e 23 50 18PX; pGVn#P
0030 f9 05 c0 08 00 00 47 45 54 20 2f 70 05 61 72 73 ...h GE T /pears
0040 6f 6e 2e 70 6e 07 20 48 54 54 50 2f 31 2e 31 8d on.png H TTP/1.1
0050 0a 48 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75 Host: g aia.cs.u
0060 6d 61 73 73 2e 65 64 75 0d 0a 43 0f 6e 0e 05 63 mass.edu Connec
0070 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 tion: ke ep-alive
0080 6d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f User-A gent: No
0090 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 zilla/5. 0 (X11;
00a0 4c 69 6e 75 78 20 78 38 36 5f 36 34 29 20 41 70 Linux x8 6.64) Ap
00b0 70 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 pleWebKi t/537.36
00c0 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 (KHTML, like Ge
00d0 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 31 30 38 2e cko) Chr ome/108.
00e0 30 2e 30 2e 30 29 53 61 66 61 72 69 2f 35 33 37 0.0.0 Sa fari/537
00f0 2e 33 36 0d 0a 44 4e 54 3a 20 31 0d 0a 41 63 63 36-DNT : 1 Acc
0100 65 70 74 3a 20 69 6d 61 67 65 2f 61 76 69 66 2c ept: ima ge/avif,
0110 69 6d 61 67 65 2f 77 65 62 70 2c 69 6d 61 67 65 image/we bp,image
0120 2f 61 70 6e 67 2c 69 6d 61 67 65 2f 73 76 67 2b /apng,im age/svg+
0130 78 6d 6c 2c 69 6d 61 67 65 2f 2a 2c 2e 2f 2a 3b xml,imag e/*,/*;
0140 71 3d 30 2e 38 0d 0a 52 65 66 65 72 65 72 3a 20 q=0.8 R eferer:
0150 68 74 74 70 3a 2f 2f 67 61 69 61 2e 63 73 2e 75 http://g aia.cs.u
0160 6d 61 73 73 2e 65 64 75 2f 77 69 72 65 73 68 61 mass.edu /wiresha
0170 72 6b 2d 6c 61 62 73 2f 48 54 54 50 2d 77 69 72 rk-labs/ HTTP-wir
0180 65 73 68 61 72 6b 2d 66 69 6c 65 34 2e 68 74 6d eshark-f ile4.htm
0190 6c 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 l-Accep t-Encodi
01a0 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 ng: gzip, deflat
01b0 65 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 e-Accep t-Langua
01c0 67 65 3a 20 65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 ge: en-U S,en;q=0
01d0 2e 39 0d 0a 0d 0a .9:...

Figure 11: Screenshot of records obtained

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

In total, the browser sent **THREE** HTTP GET request messages. The first two GET requests were sent to IP address **128.119.245.12**, while the third one was sent to IP **178.79.137.164**. This is shown in the above screenshot.

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

It can be concluded that the browser downloaded the two images parallelly. Here, both files have been requested, and then have returned in the same time period. Since the second image's GET request doesn't wait for the response of the first image to return, we could come to the conclusion that it is **parallel**. Had it been serial, the second image's GET request would have been sent after the arrival of the response of the first image. Evidence for all this is visible in the second image on the previous page.

5 Answers for Task 5: HTTP Authentication

The following screenshot depicts the observed output records on accessing a page that requires authentication. Below it, some questions regarding the same are answered.

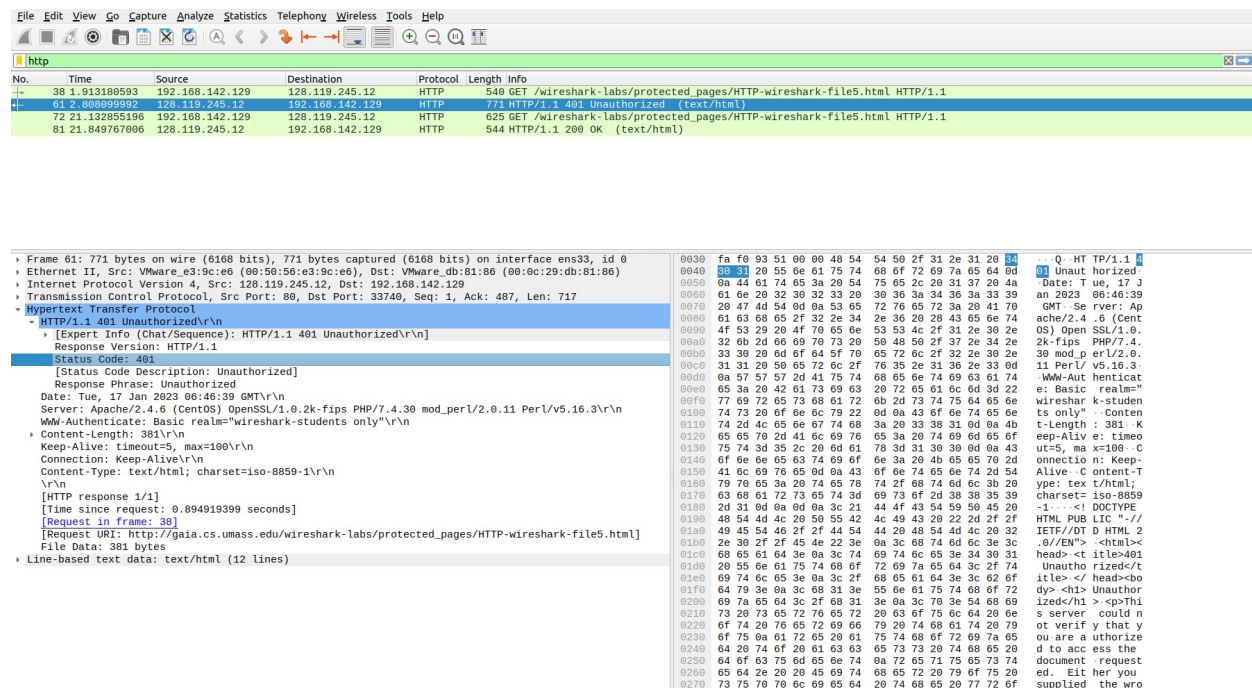


Figure 12: Screenshot of records obtained

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

As visible in the above screenshot, the HTTP status code and phrase returned from the server in response to the second HTTP GET are **401** and **Unauthorized** respectively.

(2) When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

As visible in the screenshot on the next page, a new field called "Authorization" is included when the browser sends the second HTTP GET message.

```

> Frame 72: 625 bytes on wire (5000 bits), 625 bytes captured (5000 bits) on interface ens33, id 0
> Ethernet II, Src: VMware_db:81:86 (00:0c:29:db:81:86), Dst: VMware_e3:9c:e6 (00:50:56:e3:9c:e6)
> Internet Protocol Version 4, Src: 192.168.142.129, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 33750, Dst Port: 80, Seq: 1, Ack: 1, Len: 571
< Hypertext Transfer Protocol
  < GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
    < [Expert Info (Chat/Sequence): GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1
      Request Method: GET
      Request URI: /wireshark-labs/protected_pages/HTTP-wireshark-file5.html
      Request Version: HTTP/1.1
      Host: gaia.cs.umass.edu\r\n
      Connection: keep-alive\r\n
      Cache-Control: max-age=0\r\n
    < Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzM5ldHdvcms=\r\n
      Credentials: wireshark-students:network
      DNT: 1\r\n
      Upgrade-Insecure-Requests: 1\r\n
      User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 s
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
      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: 81]

```

Figure 13: Presence of Authentication field