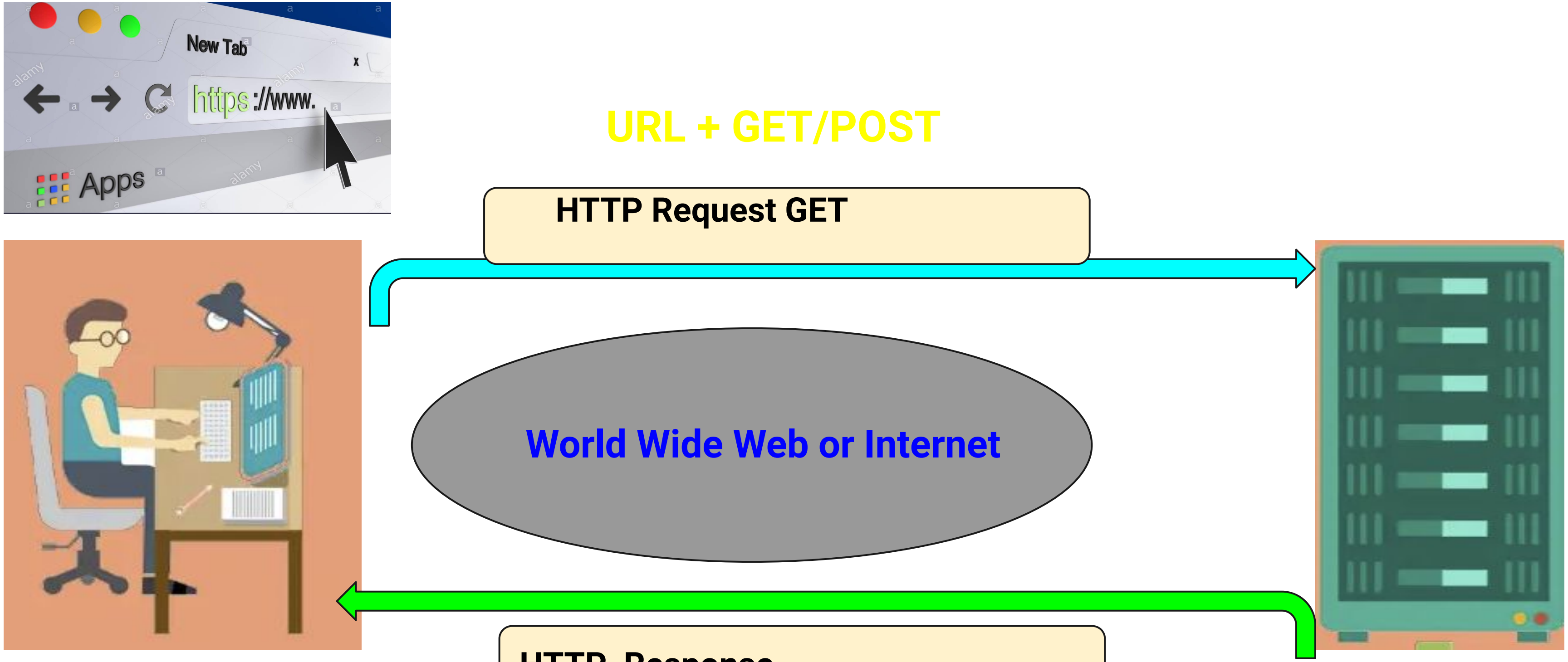
COMPUTER NETWORKS (CS315) LAB 3

Reference to Assignment 3

Agenda

1. The Basic HTTP/GET response interaction.
2. The HTTP CONDITIONAL GET/response interaction.
3. Retrieving Long Documents.
4. HTML Documents with Embedded Objects.
5. HTTP Authentication.

The Basic HTTP/GET response interaction



URL + GET/POST

HTTP Request GET

World Wide Web or Internet

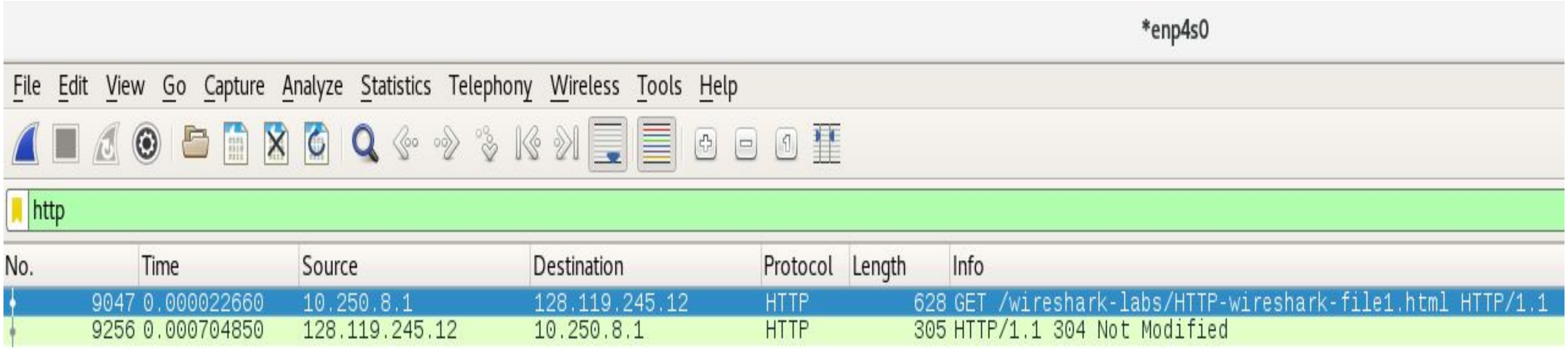
HTTP Response

Status code + Message

Steps

HTTP GET/response

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

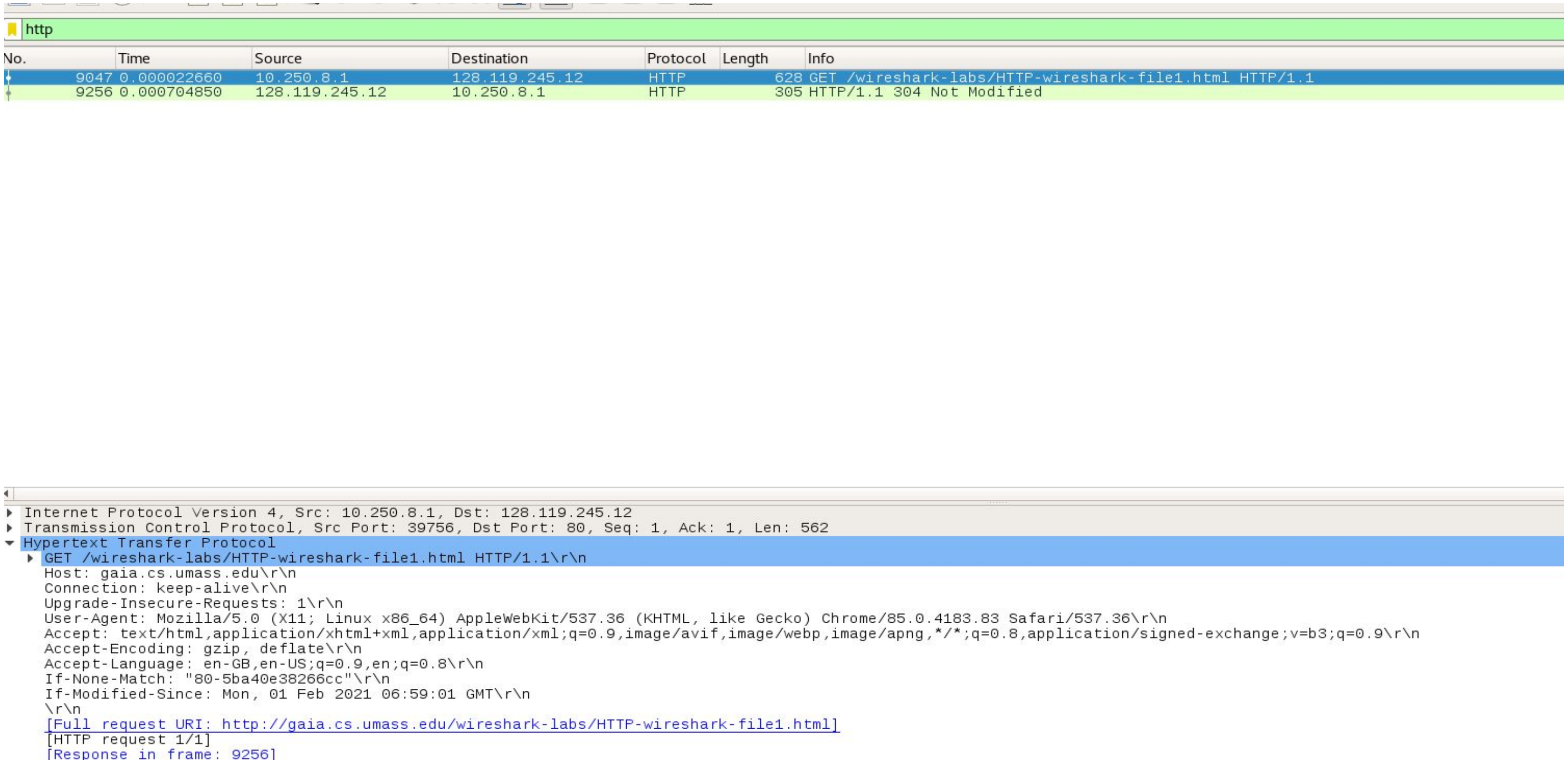


*enp4s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
9047	0.000022660	10.250.8.1	128.119.245.12	HTTP	628	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
9256	0.000704850	128.119.245.12	10.250.8.1	HTTP	305	HTTP/1.1 304 Not Modified

HTTP GET



```
http
```

```
No.          Time           Source           Destination      Protocol  Length  Info
        9047 0.000022660    10.250.8.1       128.119.245.12   HTTP         628  GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
        9256 0.000704850    128.119.245.12   10.250.8.1       HTTP         305  HTTP/1.1 304 Not Modified
```

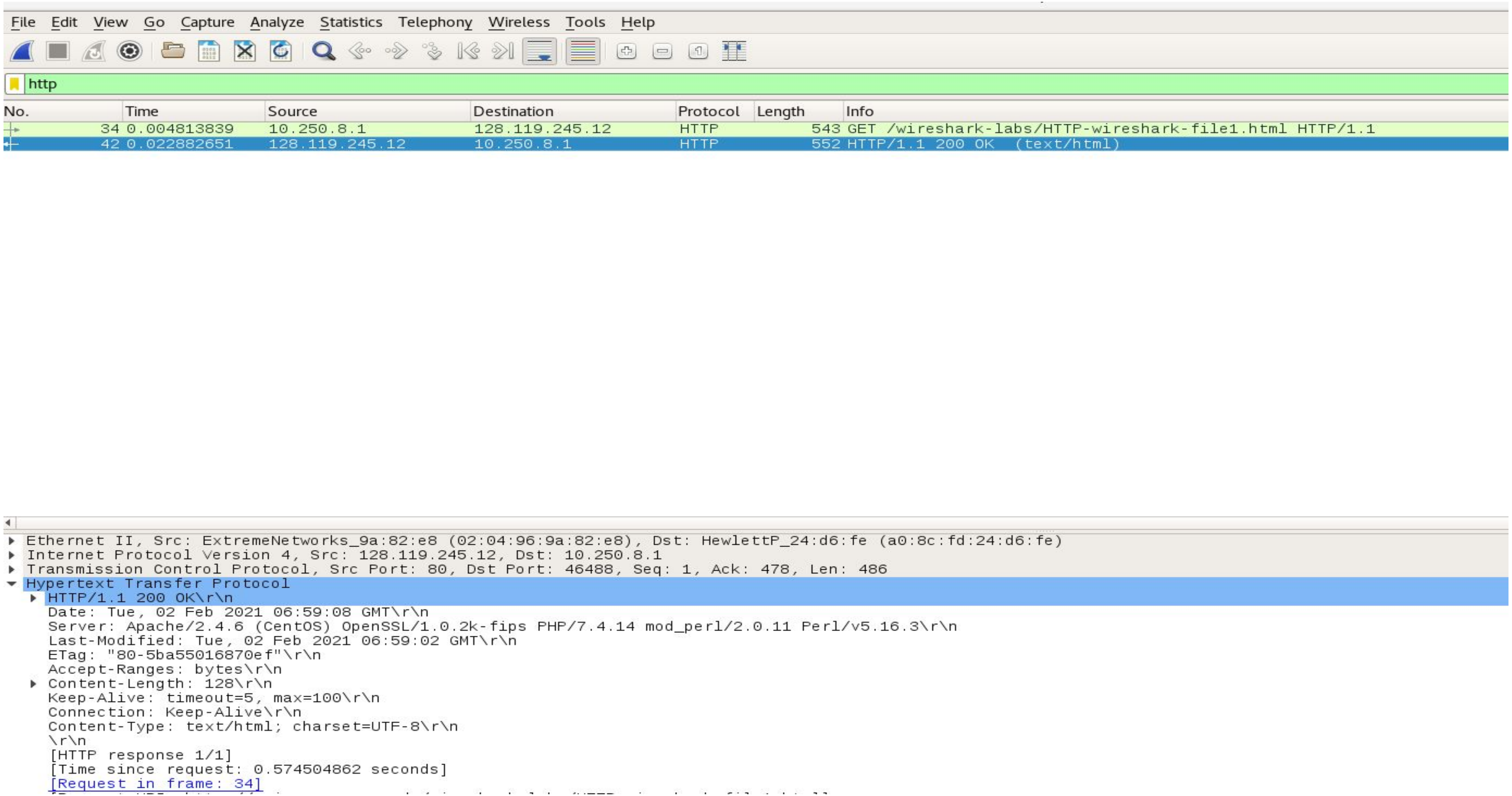
```
▸ Internet Protocol Version 4, Src: 10.250.8.1, Dst: 128.119.245.12
▸ Transmission Control Protocol, Src Port: 39756, Dst Port: 80, Seq: 1, Ack: 1, Len: 562
▾ Hypertext Transfer Protocol
  ▸ GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 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=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en-US;q=0.9,en;q=0.8\r\n
    If-None-Match: "80-5ba40e38266cc"\r\n
    If-Modified-Since: Mon, 01 Feb 2021 06:59:01 GMT\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
    [HTTP request 1/1]
    [Response in frame: 9256]
```

HTTP Response

1. HTTP Response is the packet of information sent by Server to the Client in response to an earlier Request made by Client.
2. HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:
3. Informational responses (100–199)
4. Successful responses (200–299)
5. Redirects (300–399)
6. Client errors (400–499)
7. Server errors (500–599)

3. HTTP response also displays date,last modified,content length,content type,file date,etc.

4. HTTP response message consists of a status line, followed by header lines, followed by a blank line, followed by the entity body.



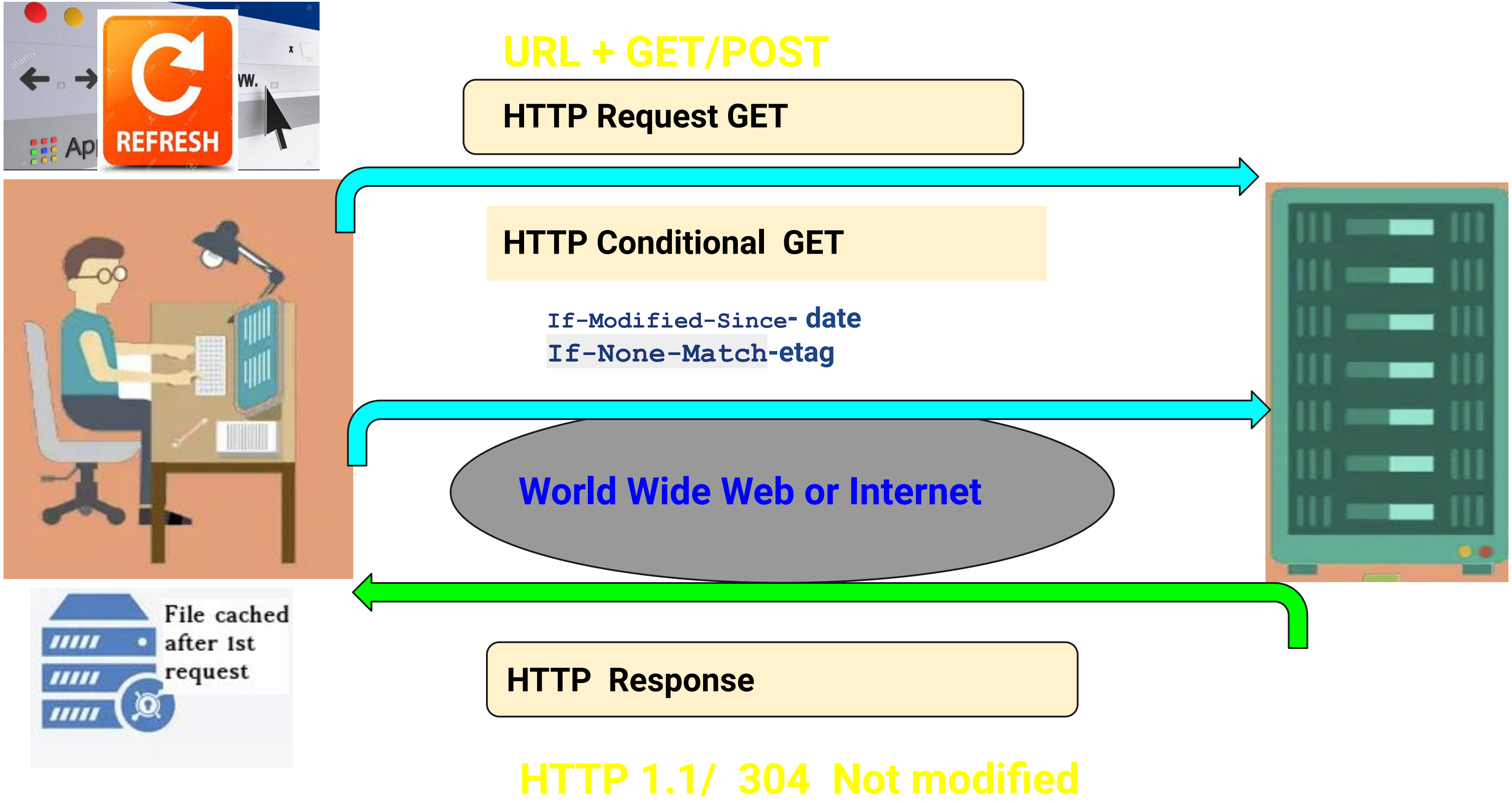
```
No.       Time           Source            Destination       Protocol  Length  Info
       34 0.004813839    10.250.8.1        128.119.245.12    HTTP         543  GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
       42 0.022882651    128.119.245.12    10.250.8.1        HTTP         552  HTTP/1.1 200 OK  (text/html)
```

```
▶ Ethernet II, Src: ExtremeNetworks_9a:82:e8 (02:04:96:9a:82:e8), Dst: HewlettP_24:d6:fe (a0:8c:fd:24:d6:fe)
▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.250.8.1
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 46488, Seq: 1, Ack: 478, Len: 486
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\r\n
    Date: Tue, 02 Feb 2021 06:59:08 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Tue, 02 Feb 2021 06:59:02 GMT\r\n
    ETag: "80-5ba55016870ef"\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: 0.574504862 seconds]
    [Request in frame: 34]
```

2. The HTTP CONDITIONAL GET/response interaction

The HTTP CONDITIONAL GET/response interaction

The HTTP CONDITIONAL GET/response interaction



URL + GET/POST

HTTP Request GET

HTTP Conditional GET

`If-Modified-Since`- date

`If-None-Match`-etag

World Wide Web or Internet

HTTP Response

HTTP 1.1/ 304 Not modified

Steps

HTTP CONDITIONAL GET/response

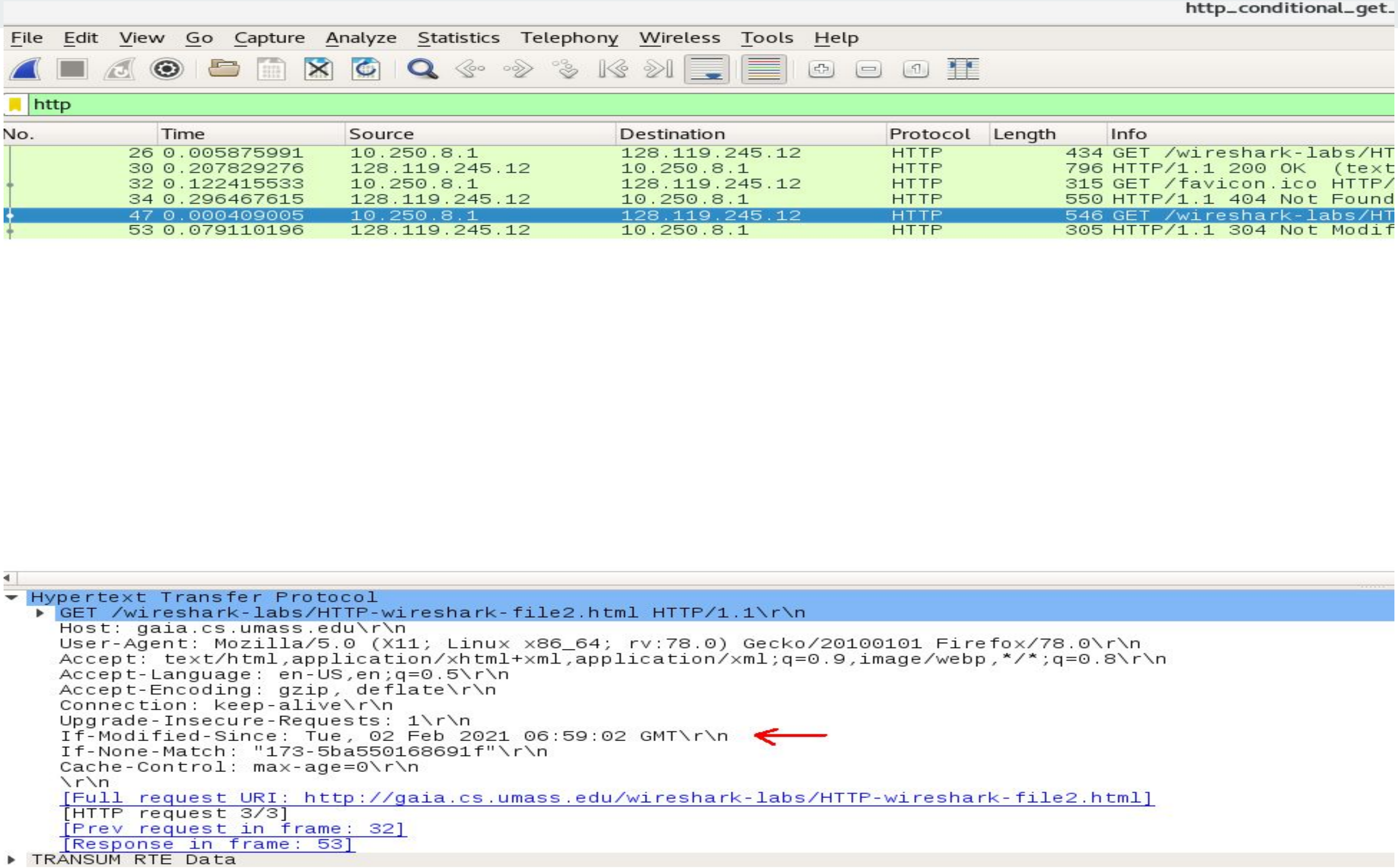
1.Start up your web browser, and make sure your browser's cache is cleared.

2.Start up the Wireshark packet sniffer

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

4. Quickly enter the same URL into your browser again (or simply select the refresh button on your browser).

5. Stop Wireshark packet capture, and enter "http" in the display-filter-specification window,



```
http
```

```
No.          Time          Source            Destination       Protocol  Length Info
        26 0.005875991   10.250.8.1        128.119.245.12    HTTP         434 GET /wireshark-labs/HT
        30 0.207829276   128.119.245.12    10.250.8.1        HTTP         796 HTTP/1.1 200 OK  (text
        32 0.122415533   10.250.8.1        128.119.245.12    HTTP         315 GET /favicon.ico HTTP/
        34 0.296467615   128.119.245.12    10.250.8.1        HTTP         550 HTTP/1.1 404 Not Found
        47 0.000409005   10.250.8.1        128.119.245.12    HTTP         546 GET /wireshark-labs/HT
        53 0.079110196   128.119.245.12    10.250.8.1        HTTP         305 HTTP/1.1 304 Not Modif
```

```
▼ Hypertext Transfer Protocol
  ▶ GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Tue, 02 Feb 2021 06:59:02 GMT\r\n
    If-None-Match: "173-5ba550168691f"\r\n
    Cache-Control: max-age=0\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 3/3]
    [Prev request in frame: 32]
    [Response in frame: 53]
▶ TRANSUM RTE Data
```

- **If-Modified-Since HTTP header**

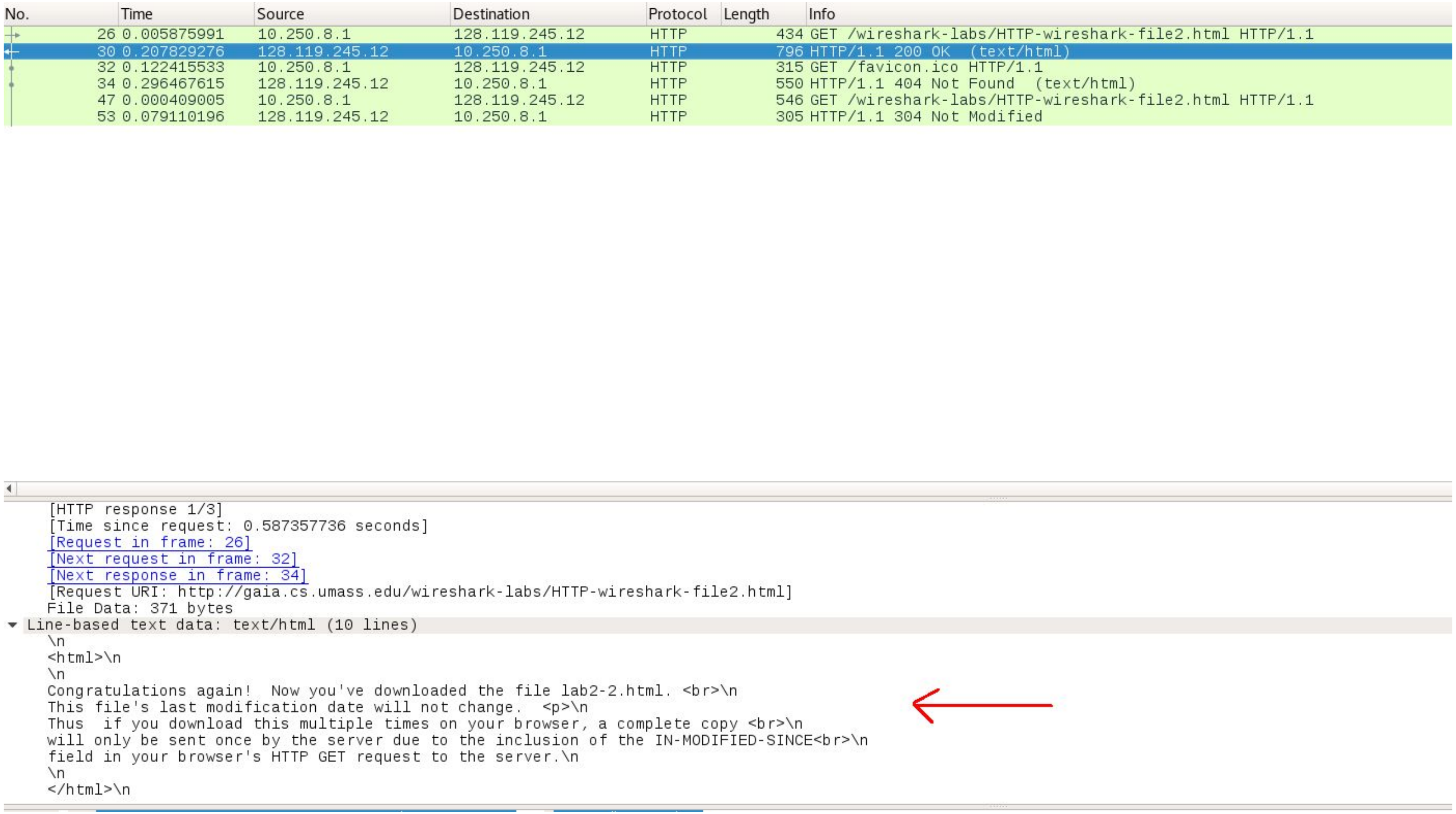
indicates the time for which a browser first downloaded a resource from the server. This helps to determine whether the resource has changed or not, since the last time it was accessed. If the status of a particular resource is 304 Not Modified, this means that the file has not changed and there is no need to download it again.

- **If-None-Match HTTP Header**

- **Cache-control HTTP Header**

It is used to specify browser caching policies in both client requests and server responses. Policies include how a resource is cached, where it's cached and its maximum age before expiring (i.e., time to live).

Server returning content of file explicitly



```
No.       Time          Source            Destination       Protocol  Length  Info
       26 0.005875991   10.250.8.1        128.119.245.12    HTTP         434 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
       30 0.207829276   128.119.245.12    10.250.8.1        HTTP         796 HTTP/1.1 200 OK  (text/html)
       32 0.122415533   10.250.8.1        128.119.245.12    HTTP         315 GET /favicon.ico HTTP/1.1
       34 0.296467615   128.119.245.12    10.250.8.1        HTTP         550 HTTP/1.1 404 Not Found  (text/html)
       47 0.000409005   10.250.8.1        128.119.245.12    HTTP         546 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
       53 0.079110196   128.119.245.12    10.250.8.1        HTTP         305 HTTP/1.1 304 Not Modified
```

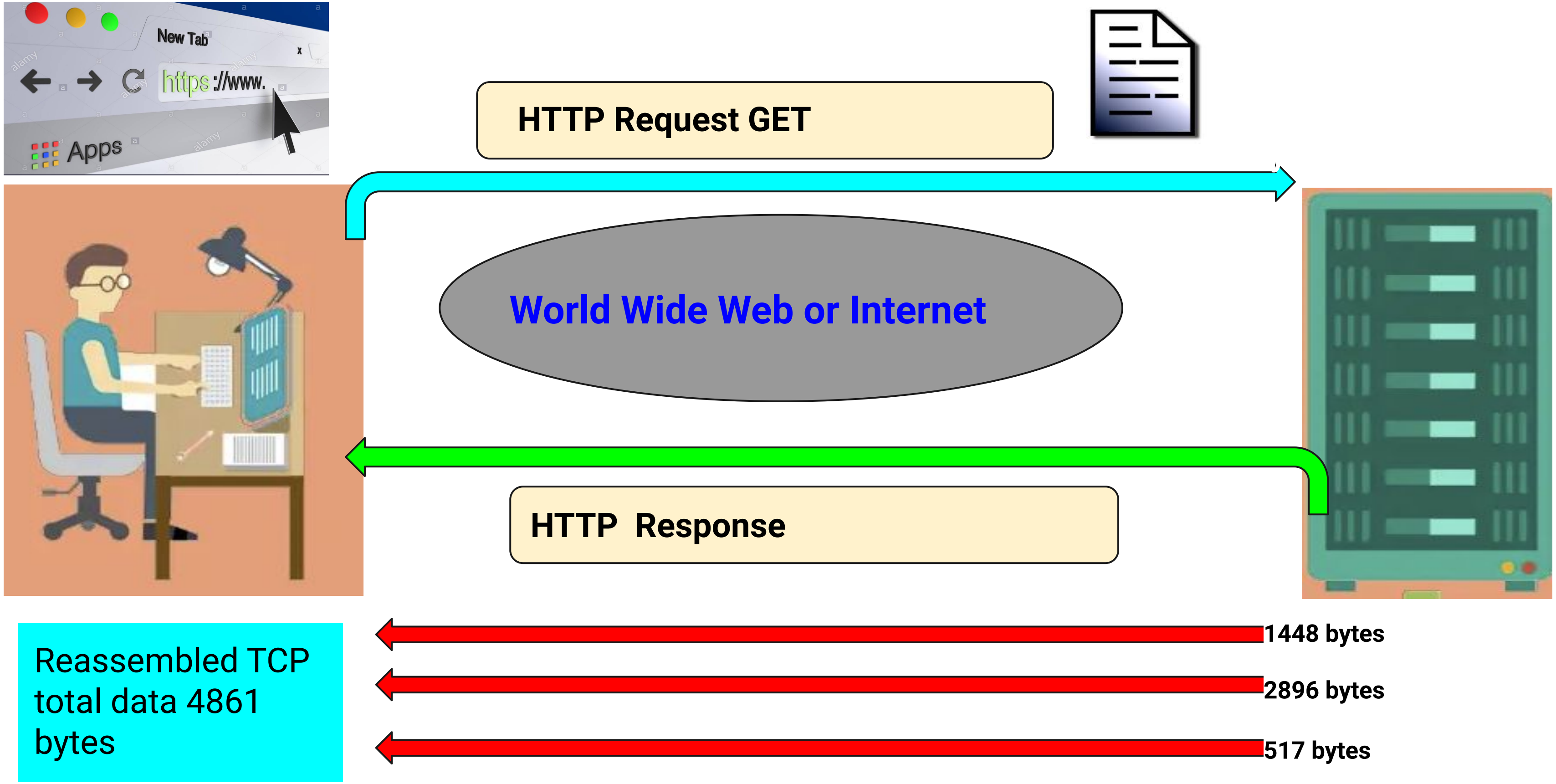
```
     [HTTP response 1/3]
     [Time since request: 0.587357736 seconds]
     [Request in frame: 26]
     [Next request in frame: 32]
     [Next response in frame: 34]
     [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
     File Data: 371 bytes
 ▾ Line-based text 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
```

3. Retrieving Long Documents

Retrieving Long Documents

- We are downloading a large document which is around 4500 bytes in size.
- But 4500 bytes is too large to fit in one TCP packet.
- So the single HTTP response message is thus broken into several pieces by TCP, with each piece being contained within a separate TCP segment.
- Each TCP segment is recorded as a separate packet by Wireshark, and the fact that the single HTTP response was fragmented across multiple TCP packets.

Retrieving Long Documents



HTTP Request GET

World Wide Web or Internet

HTTP Response

1448 bytes

2896 bytes

517 bytes

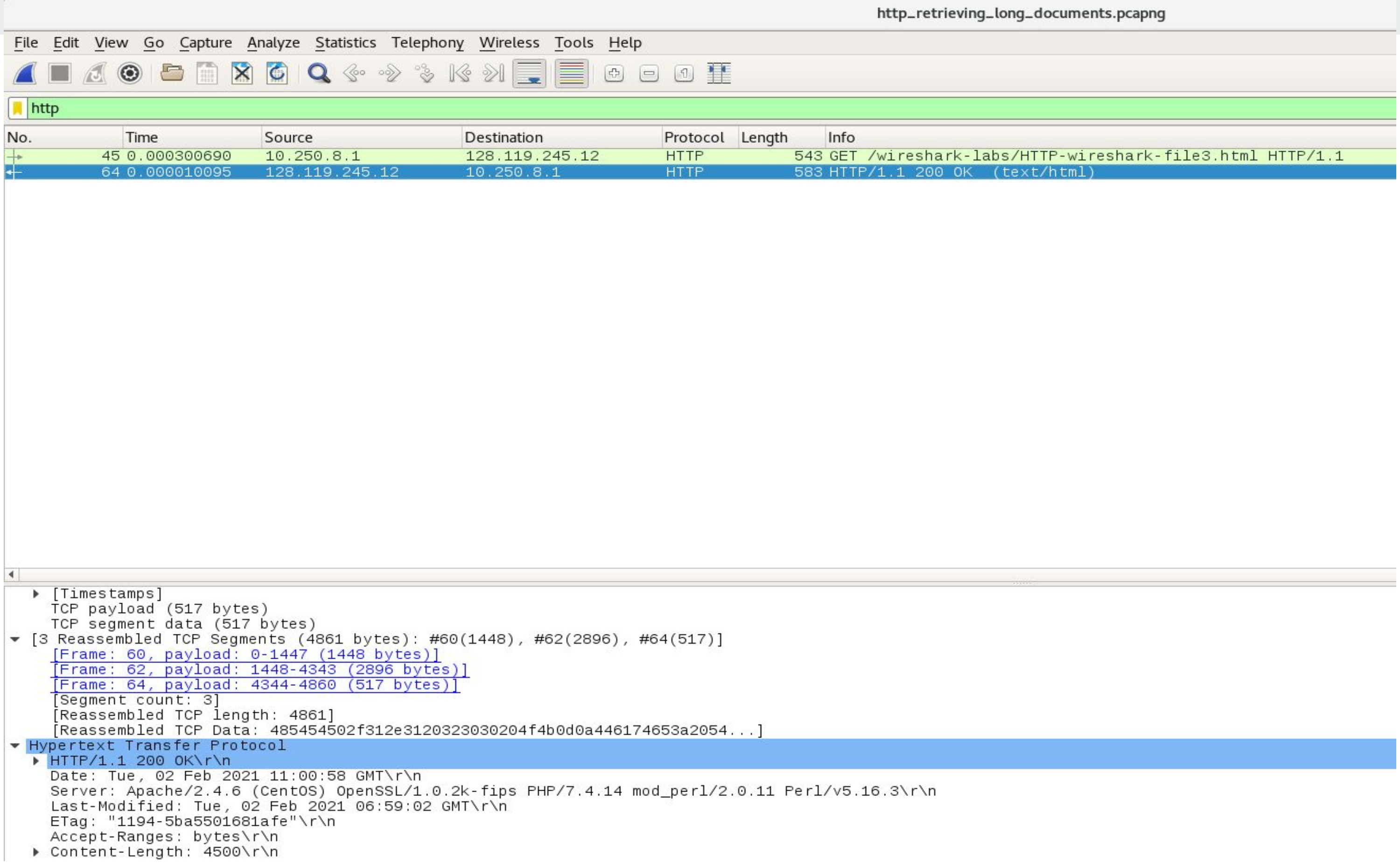
Reassembled TCP total data 4861 bytes

Retrieving Long Documents

Steps

What does TCP segment of reassembled PDU means?

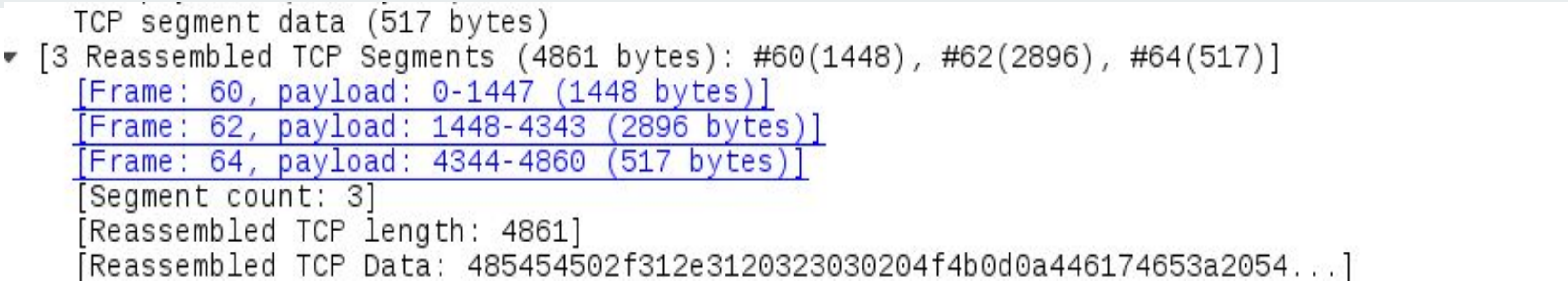
What is Payload?



```
http
```

```
No.         Time         Source           Destination       Protocol  Length  Info
      45 0.000300690  10.250.8.1       128.119.245.12    HTTP         543 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
      64 0.000010095  128.119.245.12   10.250.8.1        HTTP         583 HTTP/1.1 200 OK  (text/html)
```

```
  ▸ [Timestamps]
    TCP payload (517 bytes)
    TCP segment data (517 bytes)
▾ [3 Reassembled TCP Segments (4861 bytes): #60(1448), #62(2896), #64(517)]
    [Frame: 60, payload: 0-1447 (1448 bytes)]
    [Frame: 62, payload: 1448-4343 (2896 bytes)]
    [Frame: 64, payload: 4344-4860 (517 bytes)]
    [Segment count: 3]
    [Reassembled TCP length: 4861]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a2054...]
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 OK\r\n
    Date: Tue, 02 Feb 2021 11:00:58 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Tue, 02 Feb 2021 06:59:02 GMT\r\n
    ETag: "1194-5ba5501681afe"\r\n
    Accept-Ranges: bytes\r\n
  ▸ Content-Length: 4500\r\n
```



```
  TCP segment data (517 bytes)
▾ [3 Reassembled TCP Segments (4861 bytes): #60(1448), #62(2896), #64(517)]
    [Frame: 60, payload: 0-1447 (1448 bytes)]
    [Frame: 62, payload: 1448-4343 (2896 bytes)]
    [Frame: 64, payload: 4344-4860 (517 bytes)]
    [Segment count: 3]
    [Reassembled TCP length: 4861]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a2054...]
```

Here we see three TCP segments as the segment count is 3.

1. Frame : 60 with 1448 bytes
2. Frame : 62 with 2896 bytes
3. Frame : 64 with 517 bytes

Total data transmitted = 1448+2896+517
= 4861 bytes

4. HTML Documents with Embedded Objects

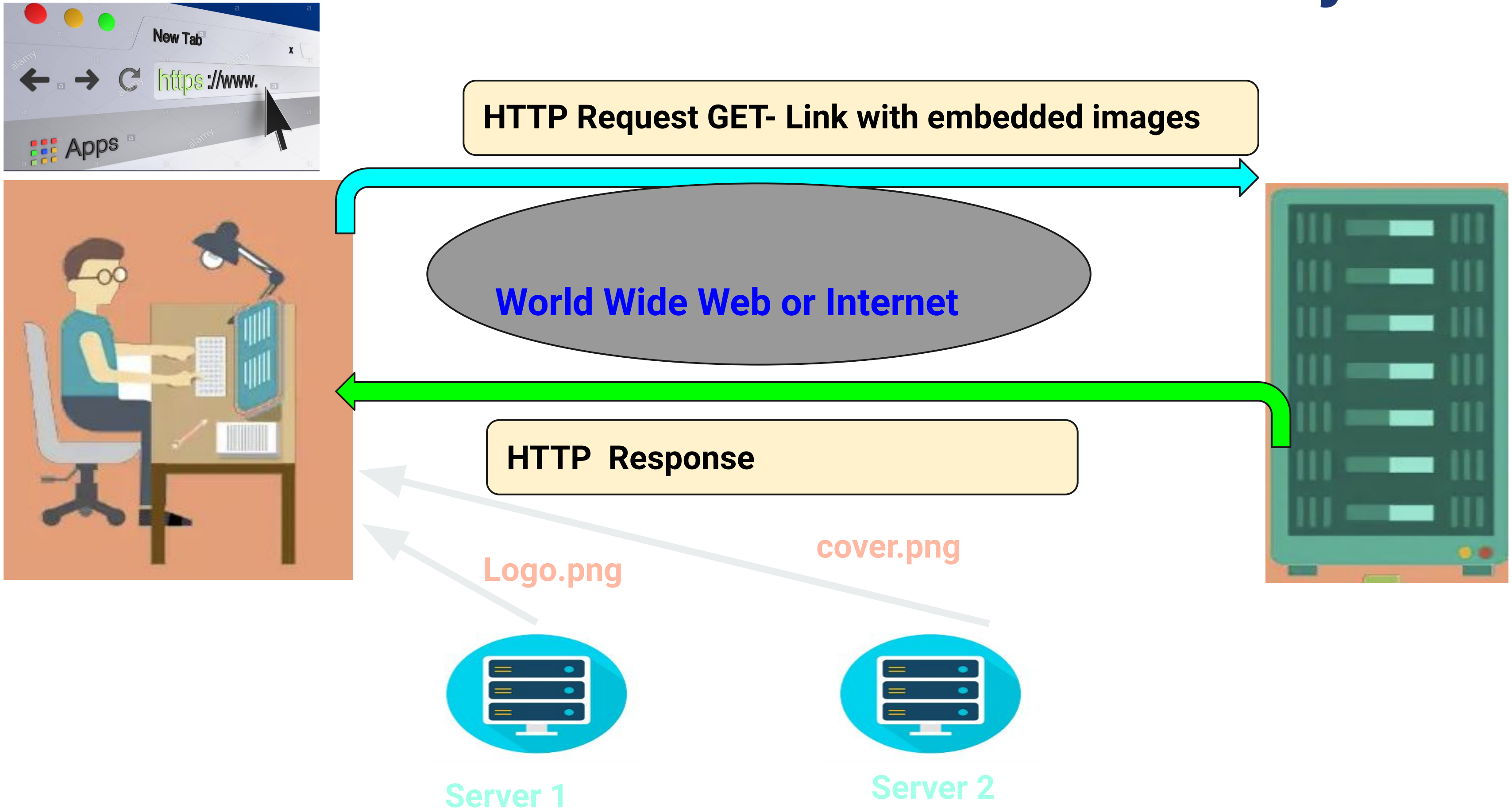
HTML Documents with Embedded Objects

- We are downloading the file with embedded objects i.e., a file that includes other objects (here image files) that are stored on another server(s).

- The images themselves are not contained in the HTML; instead the URLs for the images are contained in the downloaded HTML file.

- The browser will retrieve these images from the indicated web sites.

HTML Documents with Embedded Objects



HTTP Request GET- Link with embedded images

World Wide Web or Internet

HTTP Response

Logo.png

cover.png

Server 1

Server 2

HTML Documents with Embedded Objects

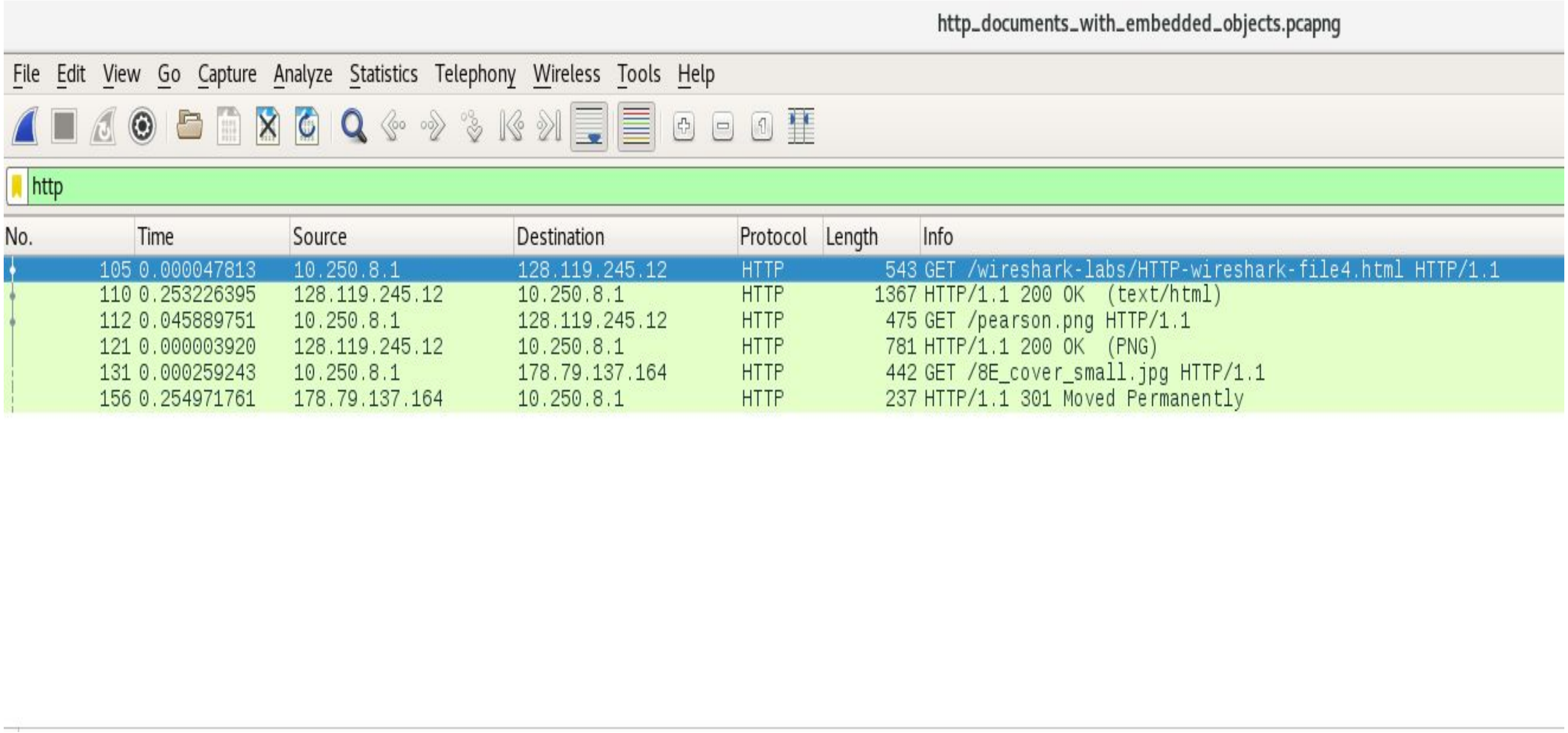
Steps

1.Start up your web browser, and make sure your browser's cache is cleared, as discussed above.

2. Start up the Wireshark packet sniffer.

3. Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
Your browser should display a short HTML file with two images.

4. Stop Wireshark packet capture, and enter "http" in the display-filter-specification window, so that only captured HTTP messages will be displayed.



http

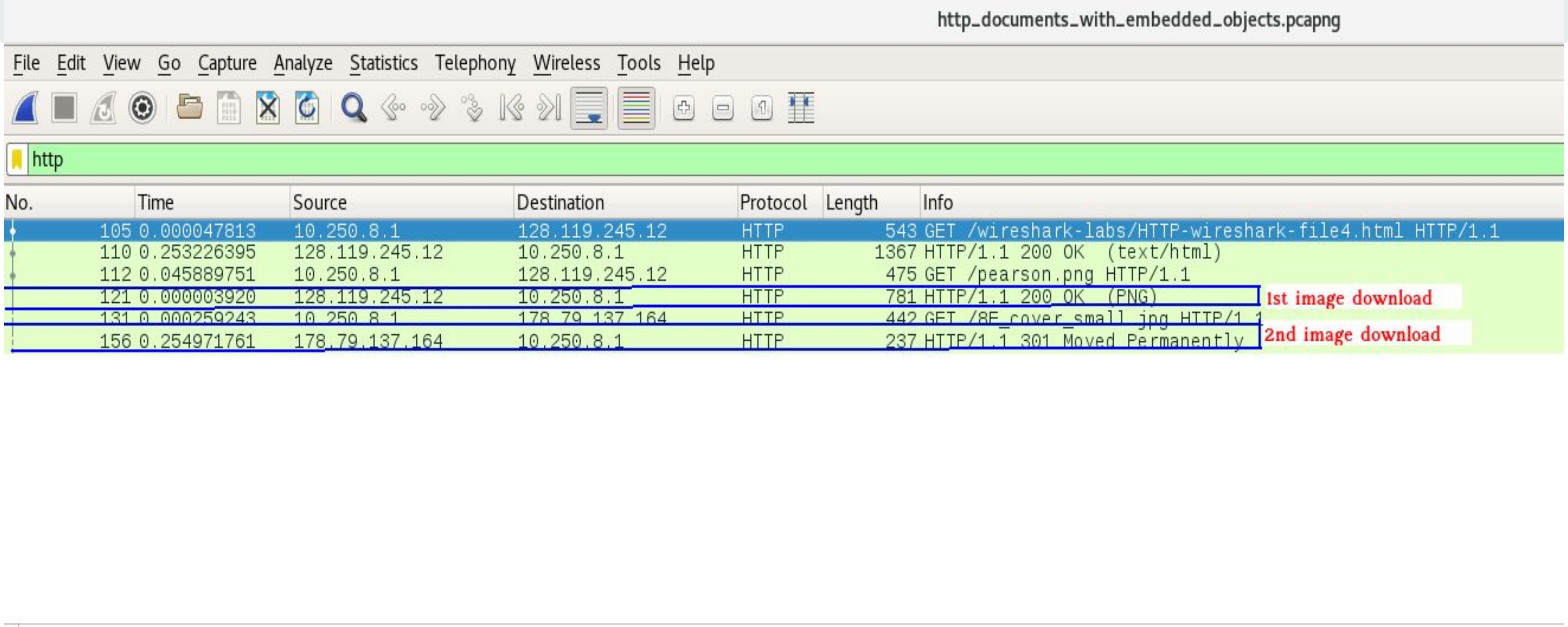
```
No.       Time         Source           Destination      Protocol Length Info
      105 0.000047813  10.250.8.1       128.119.245.12   HTTP        543 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
      110 0.253226395  128.119.245.12   10.250.8.1       HTTP       1367 HTTP/1.1 200 OK  (text/html)
      112 0.045889751  10.250.8.1       128.119.245.12   HTTP        475 GET /pearson.png HTTP/1.1
      121 0.000003920  128.119.245.12   10.250.8.1       HTTP        781 HTTP/1.1 200 OK  (PNG)
      131 0.000259243  10.250.8.1       178.79.137.164   HTTP        442 GET /8E_cover_small.jpg HTTP/1.1
      156 0.254971761  178.79.137.164   10.250.8.1       HTTP        237 HTTP/1.1 301 Moved Permanently
```

- In the screenshot,we can see two 3 HTTP GET requests.One for when we enter the url.Other two are when the images are downloaded.

- The publisher’s logo is retrieved from the gaia.cs.umass.edu web site. The image of the cover for the 5th edition is stored at the caite.cs.umass.edu server. (These are two different web servers inside cs.umass.edu).

We see two web servers here

1. 128.119.245.12 - for downloading pearson.png image (logo image)
2. 178.79.137.164 - for downloading cover page image



http

```
No.         Time          Source          Destination     Protocol  Length Info
        105 0.000047813   10.250.8.1      128.119.245.12  HTTP         543 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
        110 0.253226395   128.119.245.12  10.250.8.1      HTTP        1367 HTTP/1.1 200 OK  (text/html)
        112 0.045889751   10.250.8.1      128.119.245.12  HTTP         475 GET /pearson.png HTTP/1.1
        121 0.000003920   128.119.245.12  10.250.8.1      HTTP         781 HTTP/1.1 200 OK  (PNG)
        131 0.000259243   10.250.8.1      178.79.137.164  HTTP         442 GET /8E_cover_small.jpg HTTP/1.1
        156 0.254971761   178.79.137.164  10.250.8.1      HTTP         237 HTTP/1.1 301 Moved Permanently
```

1st image download

2nd image download

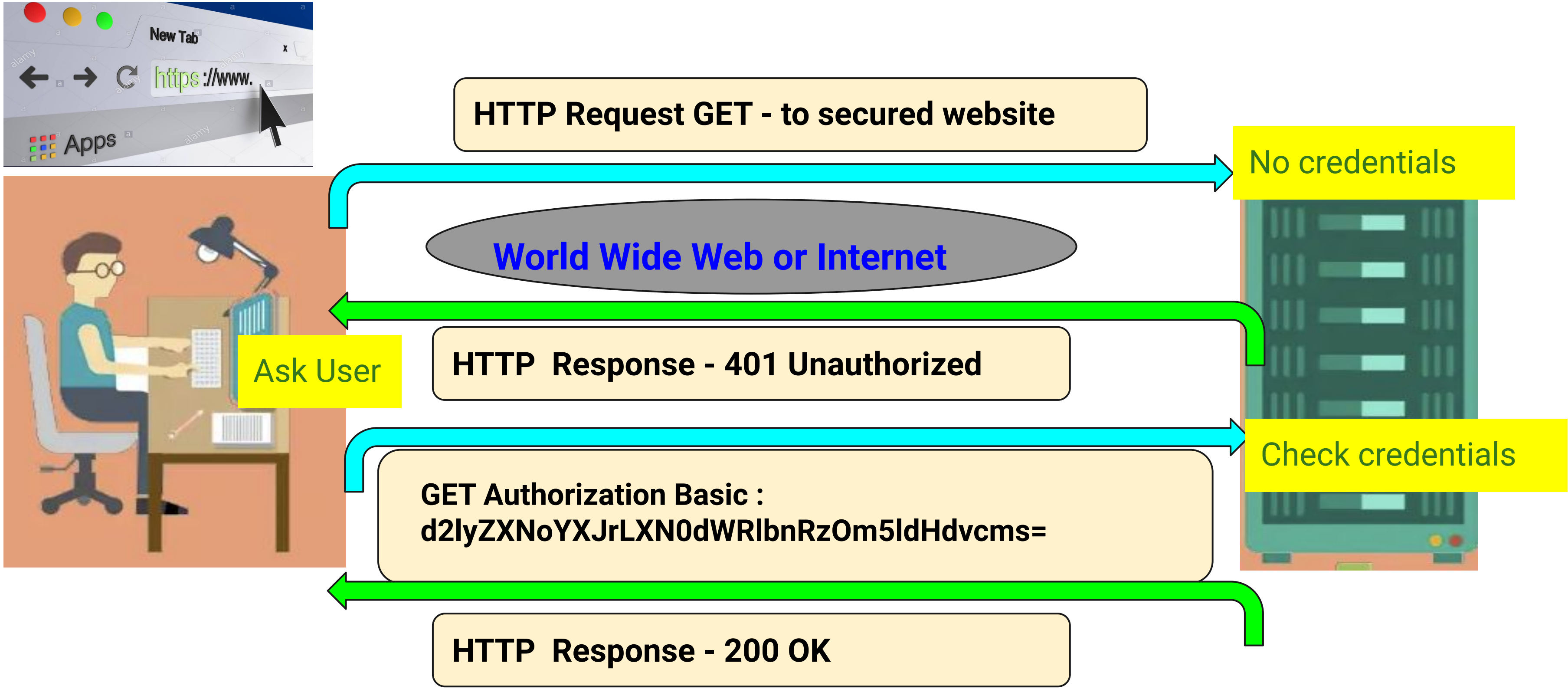
5. HTTP Authentication

HTTP Authentication

The HTTP authentication scheme works as follows:

- The client sends a request to the server for a specific page or an API resource.
- The server responds to the client with a 401 (Unauthorized) status code and provides information on how to authorize with the WWW-Authenticate header.
- The client then sends another request, including the Authorization header with credentials.
- If the credentials are valid, the server responds with the requested page or an API resource or with the 403 (Forbidden) status code if the credentials are invalid.

HTTP Authentication



HTTP Request GET - to secured website

No credentials

World Wide Web or Internet

Ask User

HTTP Response - 401 Unauthorized

Check credentials

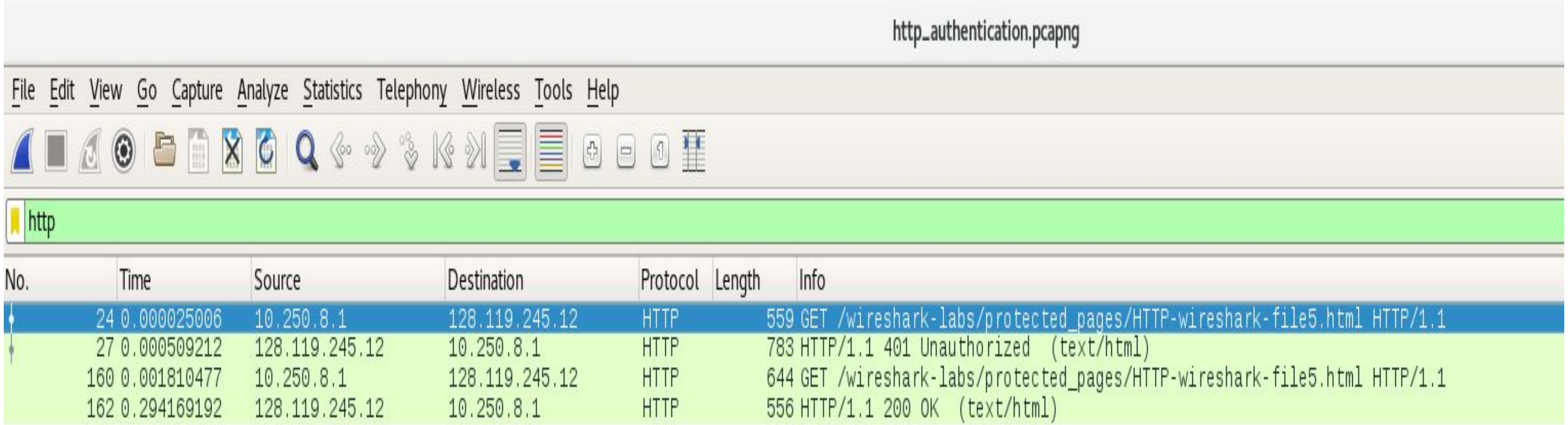
GET Authorization Basic :
d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=

HTTP Response - 200 OK

HTTP Authentication

Steps

1. Make sure your browser's cache is cleared, as discussed earlier, and close down your browser.
2. Then, start up your browser.
3. Start up the Wireshark packet sniffer • Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html Type the requested user name and password into the pop up box.
4. Stop Wireshark packet capture, and enter "http" in the display-filter-specification window



http_authentication.pcapng

http

```
No.     Time           Source           Destination      Protocol  Length  Info
     24 0.000025006    10.250.8.1       128.119.245.12   HTTP         559  GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
     27 0.000509212    128.119.245.12   10.250.8.1       HTTP         783  HTTP/1.1 401 Unauthorized  (text/html)
    160 0.001810477    10.250.8.1       128.119.245.12   HTTP         644  GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
    162 0.294169192    128.119.245.12   10.250.8.1       HTTP         556  HTTP/1.1 200 OK  (text/html)
```

We see two HTTP GET and HTTP responses.

1. The first HTTP GET loads the entered URL in the browser.

1. The second HTTP response is sent as a response to first HTTP GET.The status code of the server response to the initial **HTTP GET** request was **401 Unauthorized**.The **401 Unauthorized Error** is HTTP status code error that represented the request sent by the client to the server lacks **valid authentication** credentials.

1. The second HTTP GET loads the URL again after entering username and password.The second HTTP GET request include the field “**Authorization: Basic**” with the **username** and **password** that was entered.

1. The second HTTP response has status code 200 OK.

- The username (**wireshark-students**) and password (**network**) that you entered are encoded in the string of characters (**d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=**) in the "**Authorization: Basic**" header of client's HTTP GET message.

- The string highlighted in red is username and highlighted in blue is password.

- We can decode the encrypted username and password with the following website :https://www.motobit.com/util/base64-decoder-encoder.asp.

THANK YOU