COMPUTER NETWORKS LAB – WEEK 3

Name: TUSHAR Y S

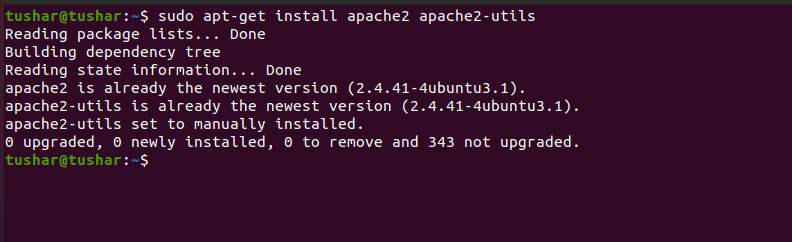
SRN: PES1UG19CS545

**1. Password Authentication:**

1.1. Setting the username and password:

- The apache utilities are installed using the following command:

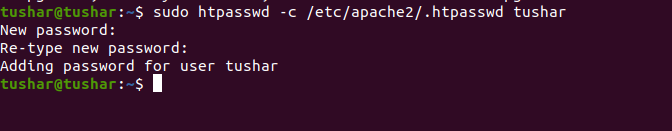
sudo apt-get install apache2 apache2-utils



- We can set a password for a given user and write it into .htpasswd file after running the following command:

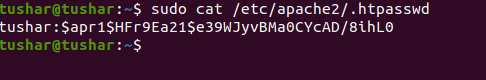
sudo htpasswd -c/etc/apache2/.htpasswd tushar

Here username is ‘tushar’



- To view the authentication , following command is used:

sudo cat /etc/apache2/.htpasswd



1.2. Setting up the Authentication phase:

- The file is opened for authentication by using the command:

sudo nano /etc/apache2/sites-available/000-default.conf

The file is modified by writing the following:

<Directory "/var/www/html">

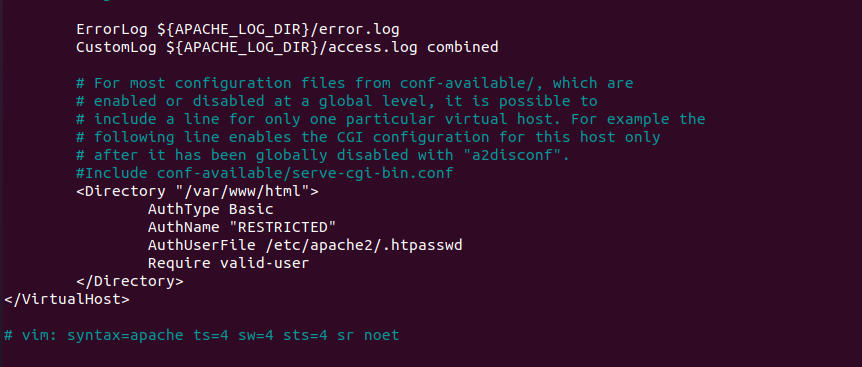
AuthType Basic

AuthName "RESTRICTED"

AuthUserFile /etc/apache2/.htpasswd

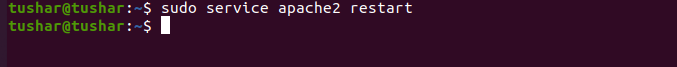
Require valid-user

</Directory>



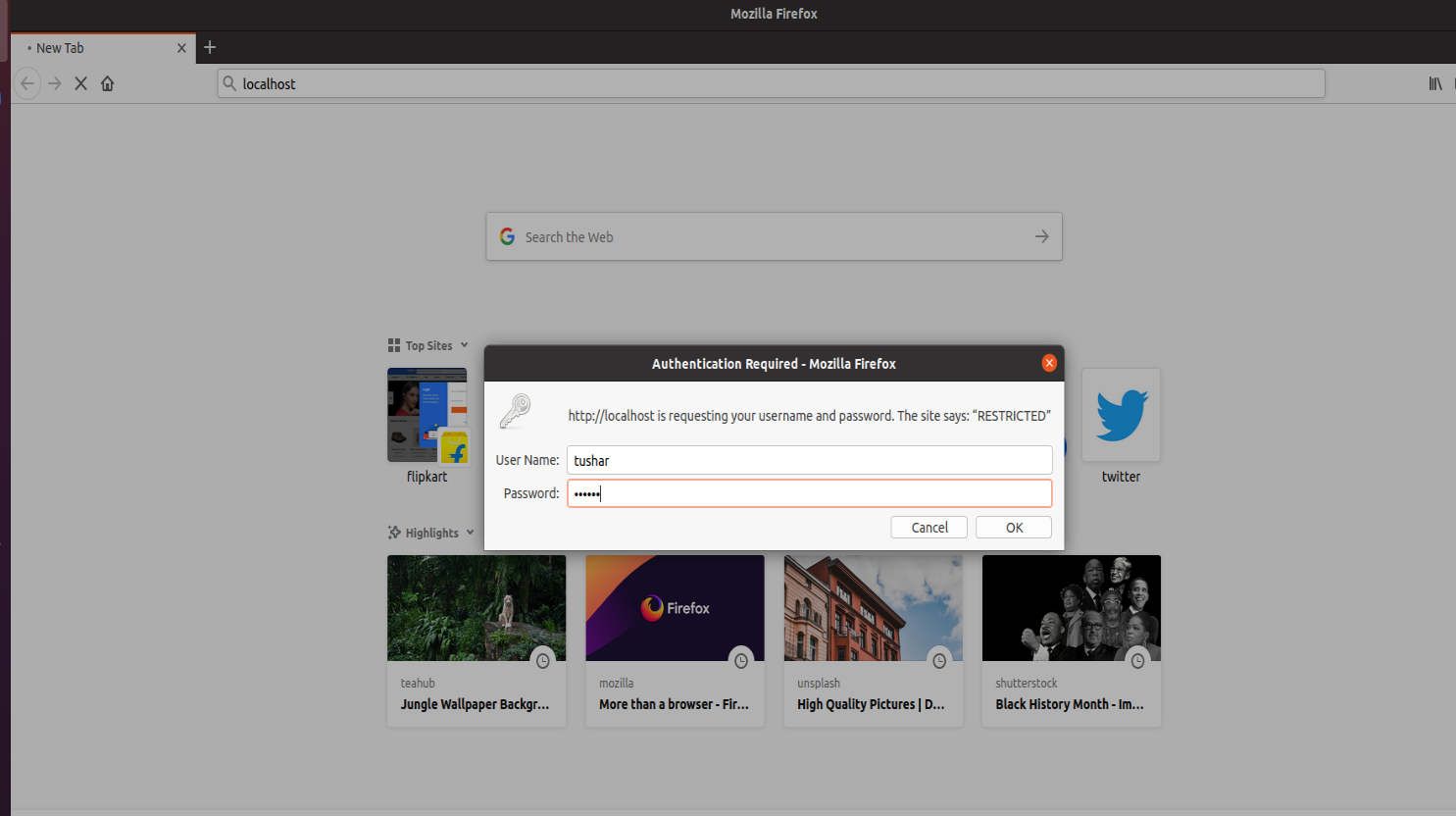
- Password policy implementation is done by restarting the apache server by using the command:

sudo service apache2 restart



1.3. Accessing the localhost:

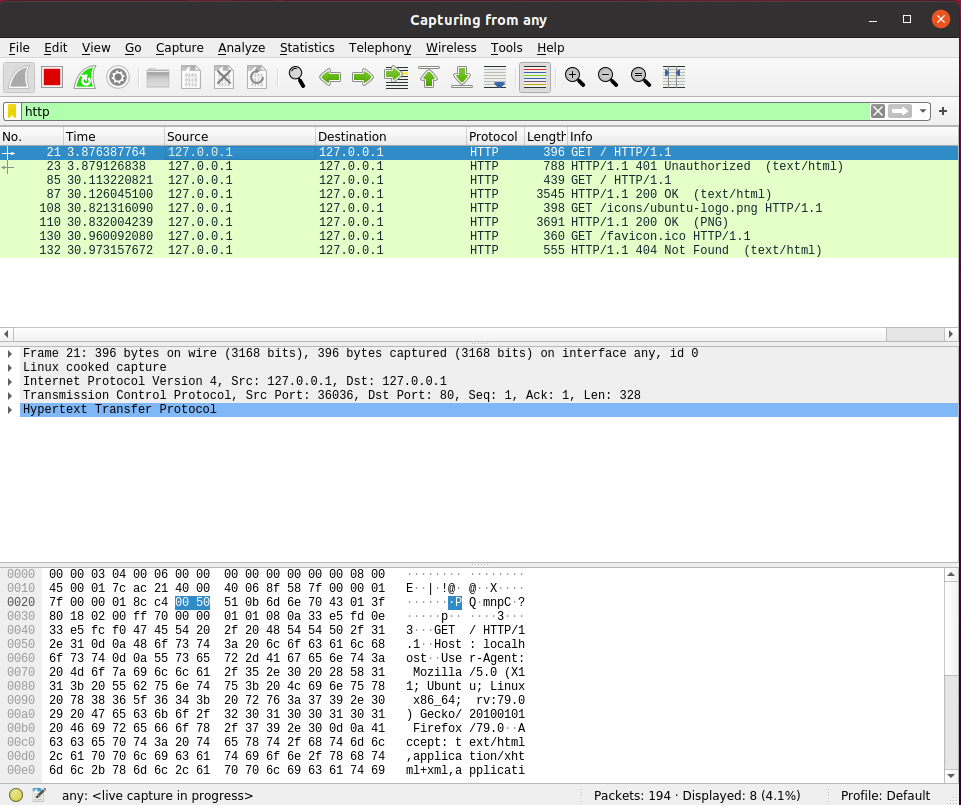
- The local host is accessed using the browser by entering the username and password.



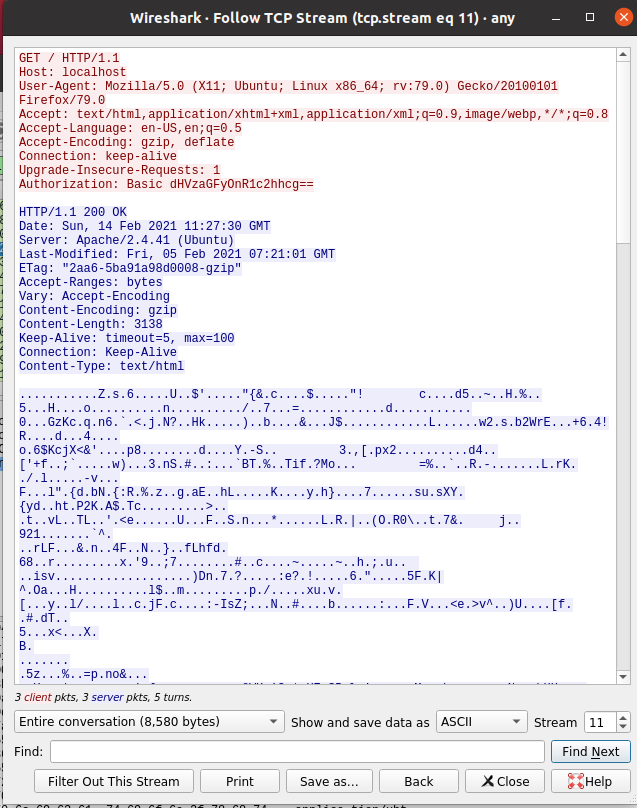
1.4. Packets capture using Wireshark

- Wireshark is used to capture the packets sent upon the network.

Wireshark packet capture:



- Using the ‘follow TCP stream’ on HTTP message segment the password is retrieved which is being encrypted by the base64 algorithm.



As we can observe above, the encrypted content is :

“dHVzaGFy0nR1c2hhcg”

1.5. Decrypting the encryption (base64 algorithm):

|  |  |  |
| --- | --- | --- |
| Encrypted Character | Index | Binary Equivalent |
| d | 29 | 011101 |
| H | 7 | 000111 |
| V | 21 | 010101 |
| z | 51 | 110011 |
| a | 26 | 011010 |
| G | 6 | 000110 |
| F | 5 | 000101 |
| y | 50 | 110010 |
| O | 14 | 001110 |
| n | 39 | 100111 |
| R | 17 | 010001 |
| 1 | 53 | 110101 |
| c | 28 | 011100 |
| 2 | 54 | 110110 |
| h | 33 | 100001 |
| h | 33 | 100001 |
| c | 28 | 011100 |
| g | 32 | 100000 |

|  |  |  |
| --- | --- | --- |
| Combined 8 bit word | ASCII Value | Decrypted Character |
| 01110100 | 116 | t |
| 01110101 | 117 | u |
| 01110011 | 115 | s |
| 01101000 | 104 | h |
| 01100001 | 97 | a |
| 01110010 | 114 | r |
| 00111010 | 58 | : |
| 01110100 | 116 | t |
| 01110101 | 117 | u |
| 01110011 | 115 | s |
| 01101000 | 104 | h |
| 01100001 | 97 | a |
| 01110010 | 114 | r |

Therefore, the decrypted content is:

tushar:tushar

The word before the semi colon is the username.

The word after the semi colon is the password corresponding to that username.

**2. Cookie Setting:**

2.1. Cookie setting using PHP:

- A PHP file is created to set the cookie using the following code:

<html>

<?php

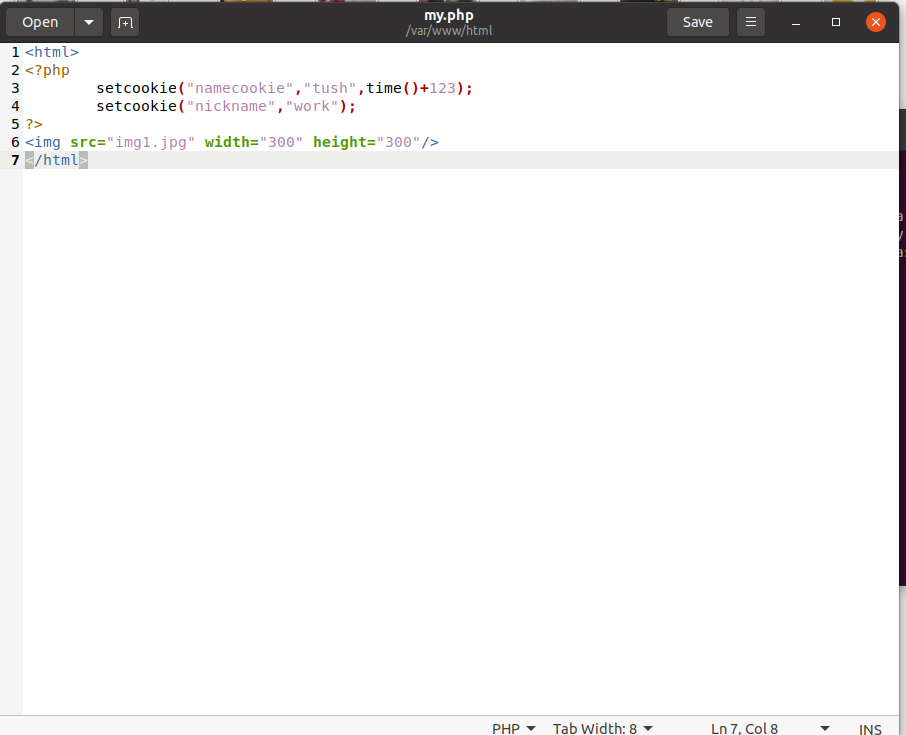
setcookie("namecookie","tush",time()+123);

setcookie("nickname","work");

?>

<img src= “img1.jpg” width= “300” height= “300” />

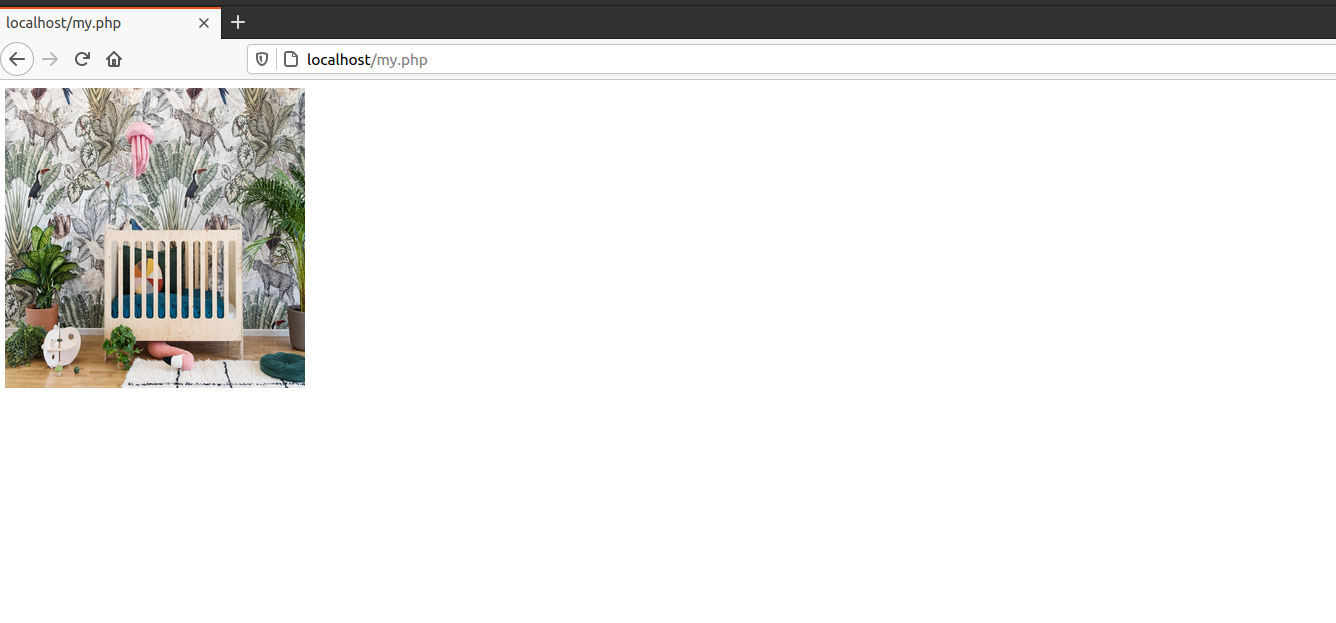
</html>

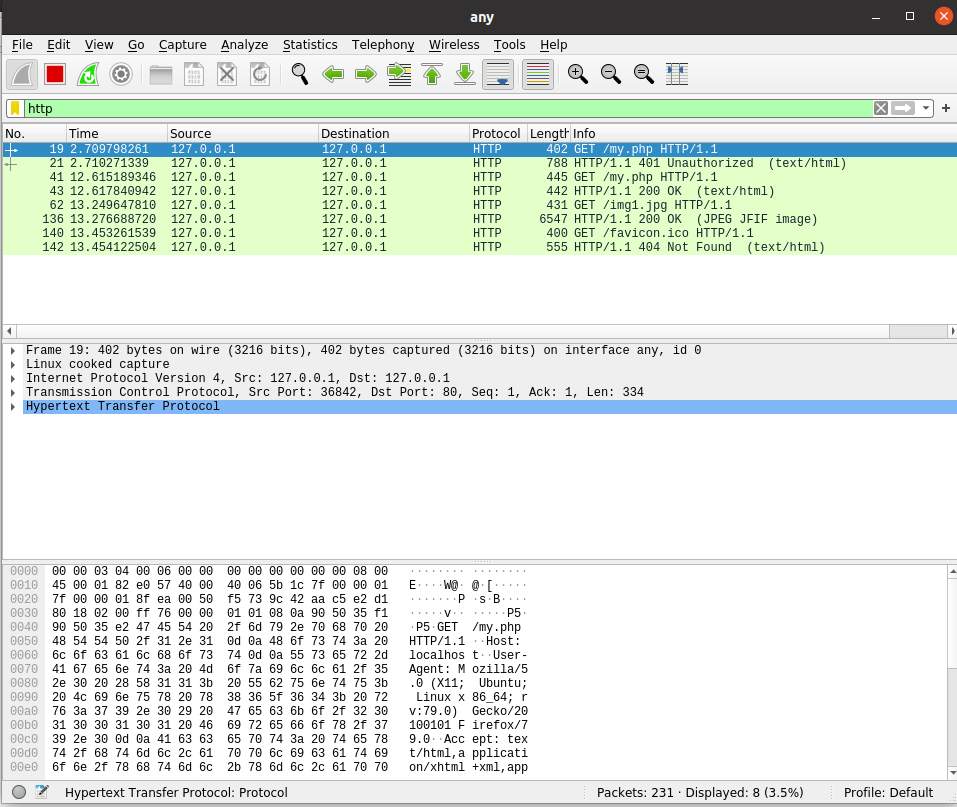


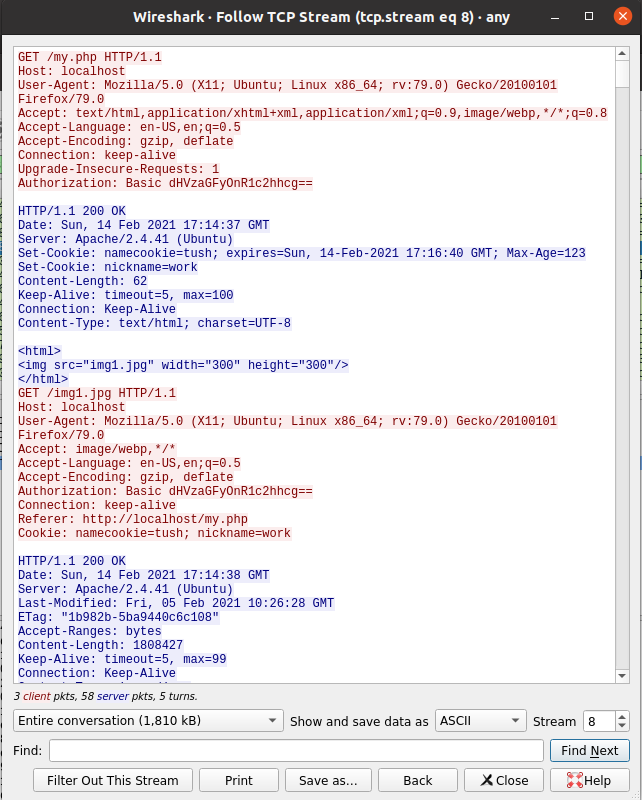
This ‘my.php’ is palced under /var/www/html.

2.2. Wireshark Packet Capture:

- The php file is accessed using the browser.

- - - At the same time, packets are captured using wireshark.



- The packet of GET request for the php file is analyzed and its follow TCP stream is shown below:  


From the above screenshot, it can be seen that the cookie has been set.

The cookie name, value and the expiry time of the cookie are some of the parameters associated with the cookie which can be observed above.

**3. Conditional Get: If-Modified-Since**

3.1. Conditional Get for plain HTML:

- The browser cache and browser history is cleared.

- Wireshark is launched.

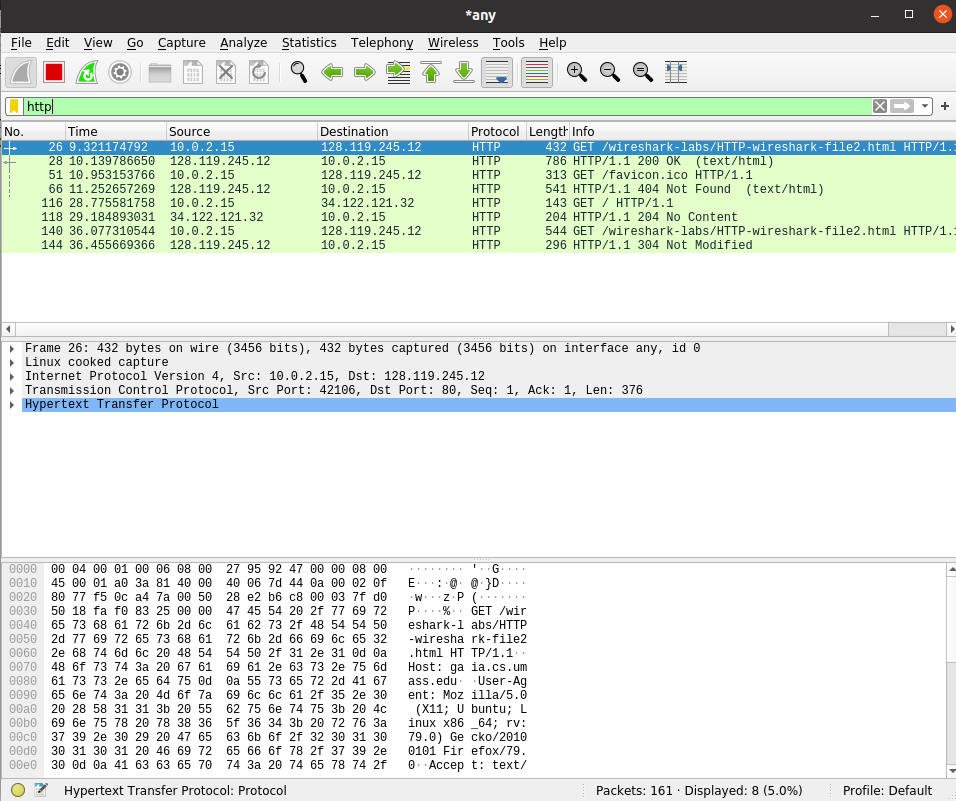
- The following URL is entered in the browser:

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>



Next, the website is refreshed.

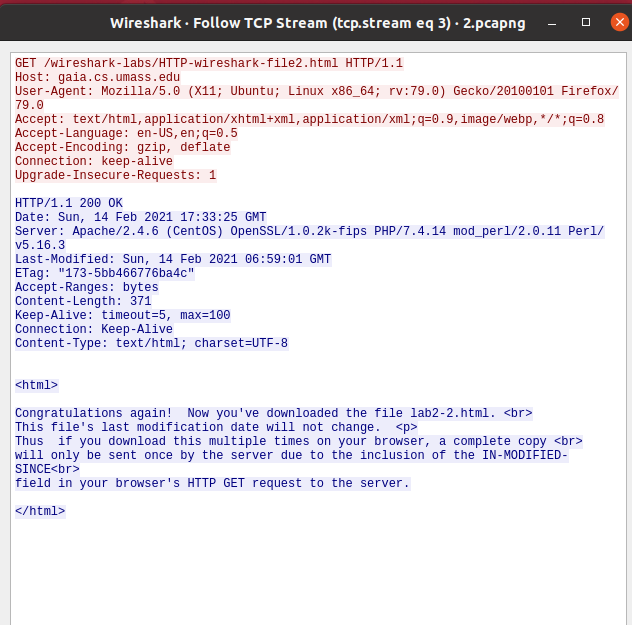
- Wireshark packet capture:



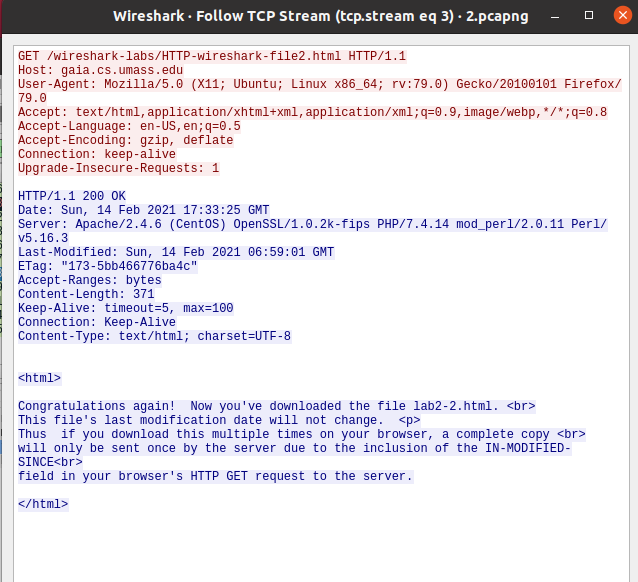
As can be observed above, the response for the first GET request is having a 200 OK response status. The response from the server for the second request is obtained as ‘304 Not Modified’. This is because we just refreshed the page and while doing so there wasn’t any change in the content of the resource(it wasn’t modified).

- Follow TCP Stream:

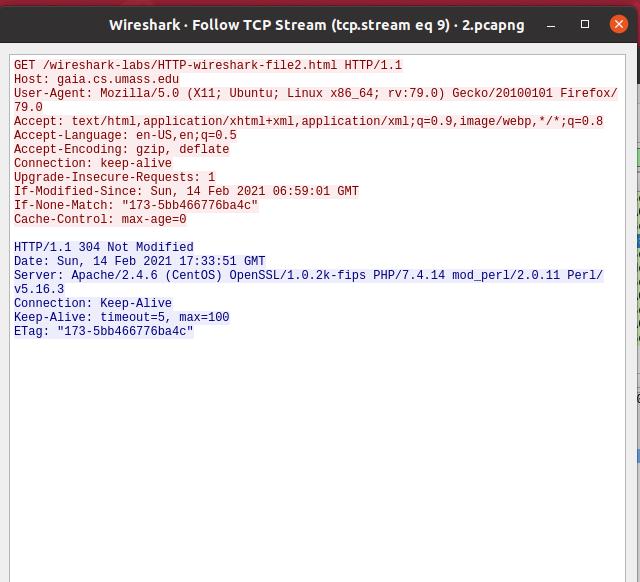
First GET request:



As can be observed from the above screenshot, there is no “If-Modified-Since” line for the first GET request.

Server Response for first GET request:  


As can be observed above, the server has explicitly returned the content of the file. The HTML file contents can be seen above.

Second GET request:  


From the above screenshot , we can see the “If-Modified-Since” field for the second GET request( if the resource remains unmodified).

The “If-Modified-Since” header indicates the time for which a browser first downloaded a resource from the server.

Server response for second GET request:

  
The server response for the second GET request is having a response status code and phrase of “304 Not Modified”.

The server hasn’t explicitly returned the contents of the file(as the request is made for the second time and the resource remains unchanged).

3.2. Conditional GET in Local machine(with some image):

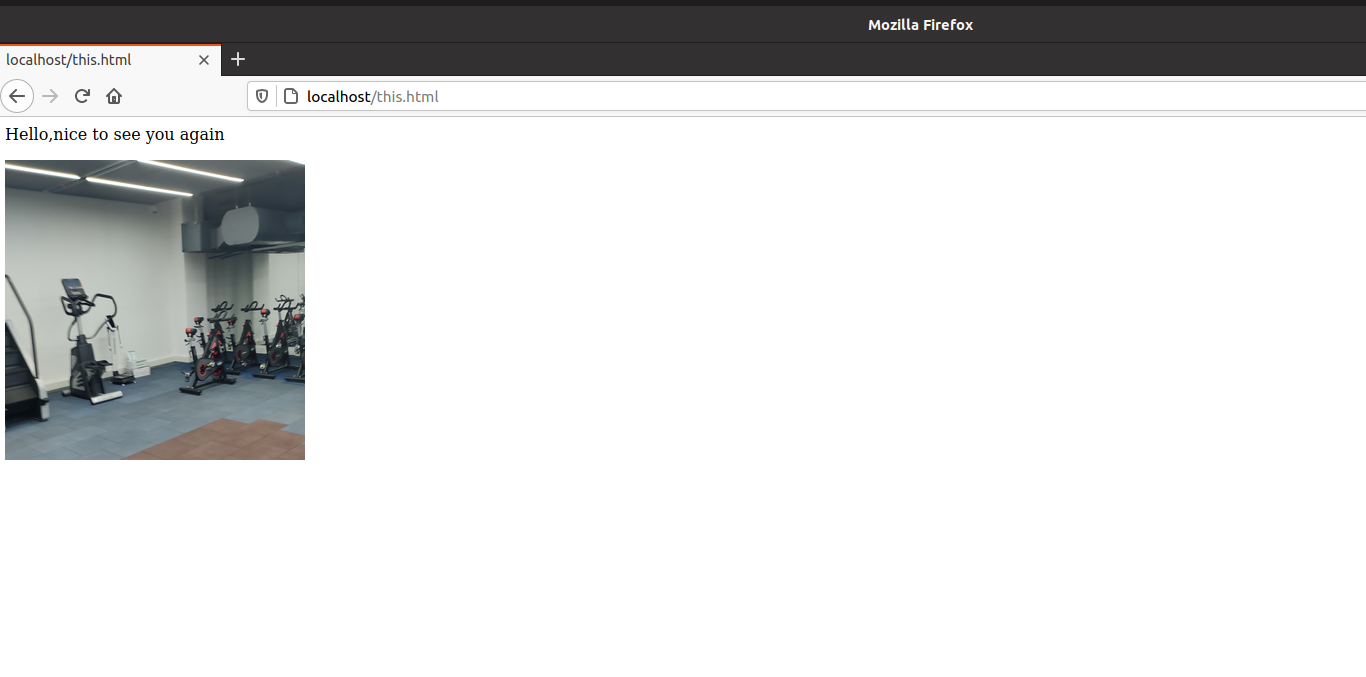
- The browser cache and browser history is cleared.

- Wireshark is launched.

- A html file named ‘this.html’ is placed under the html directory.

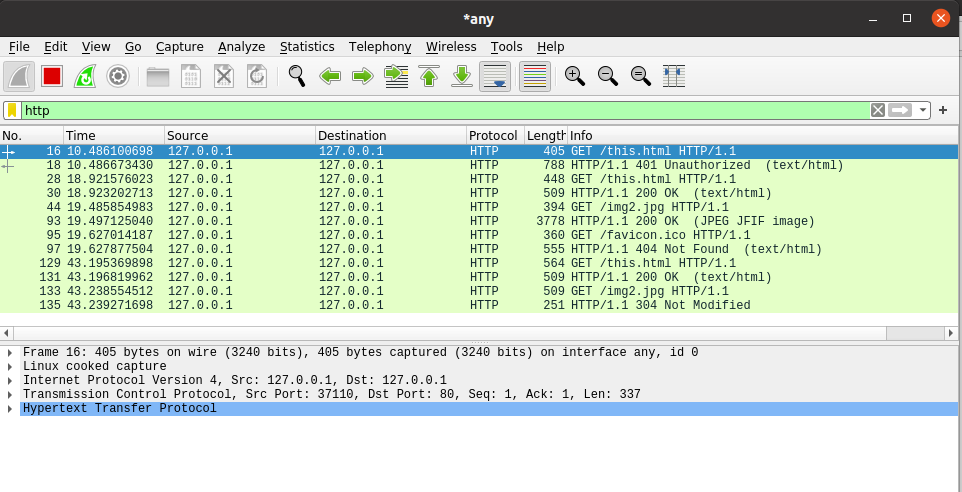


- This html file accessed with the help of a web browser



Now the website is refreshed.

- Wireshark Packet Capture:



Again, the response for the first GET request of image is having a 200 OK response status. The response from the server for the second request of image is obtained as ‘304 Not Modified’. Since the image wasn’t modified , we get ‘304 Not Modified’ server response.

- Follow TCP Stream:

First GET request:

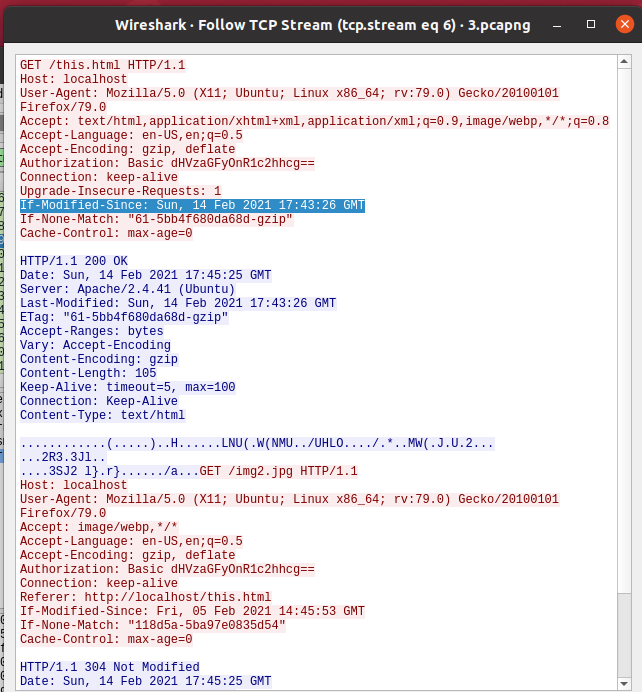


As can be observed from the above screenshot, there is no “If-Modified-Since” line for the first GET request.

Server response for first GET request:  


As can be observed above, the server has explicitly returned the content of the file. GET request is made further for obtaining the image from the server.

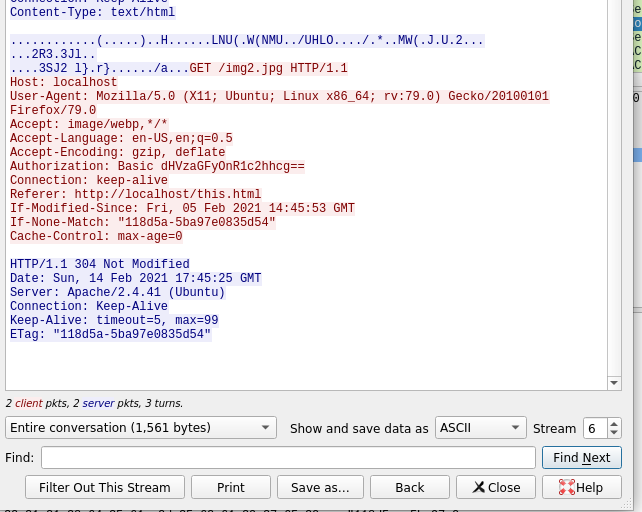
Second GET request:



From the above screenshot , we can see the “If-Modified-Since” field (highlighted in blue color)for the second GET request( if the resource remains unchanged).

The “If-Modified-Since” header indicates the time for which a browser first downloaded a resource from the server.

Server response for second GET request:



The server response for the second GET request is having a response status code and phrase of “304 Not Modified”. This is because the resource which is being requested has not been modified.