Computer Network Lab – WEEK 3

PES1UG20CS806

Divyanshu Sharma

1. Password Authentication

1.1 Password Generation:

- To enable basic authentication for HTTP, we need to generate a password file. This file can be generated using the **htpasswd** command.
- Using **sudo htpasswd -c /etc/apache2/.htpasswd username** we can set a password for the given user username and write it into the .htpasswd configuration file
- The **cat command** can be used to view the encrypted password file, which is encrypted using the Data Encryption Standard algorithm

```
@CSELAB: ~/Desktop/CN LAB/WEEK3
student@CSELAB:~/Desktop/CN LAB/WEEK3$ sudo htpasswd -c /etc/apache2/.htpasswd divyanshu
[sudo] password for student:
New password:
Re-type new password:
Adding password for user divyanshu
student@CSELAB:~/Desktop/CN LAB/WEEK3$ sudo cat  /etc/apache2/.htpasswd
divyanshu:$apr1$LGQLYov9$6B0YVy0vjjK5wEeMTNrEI/
student@CSELAB:~/Desktop/CN LAB/WEEK3$
```

1.2 Apache Server Authentication

- To enable password authentication in the server, we need to modify the Apache configuration file.
- This can be done using **sudo nano /etc/apache2/sites-available/000default.conf**
- Password authentication is added to **the /var/www/html directory** which is the localhost home directory so that all files hosted here will require authentication to access.
- To activate the authentication and policy, we need to restart the server using **sudo service apache2 restart**

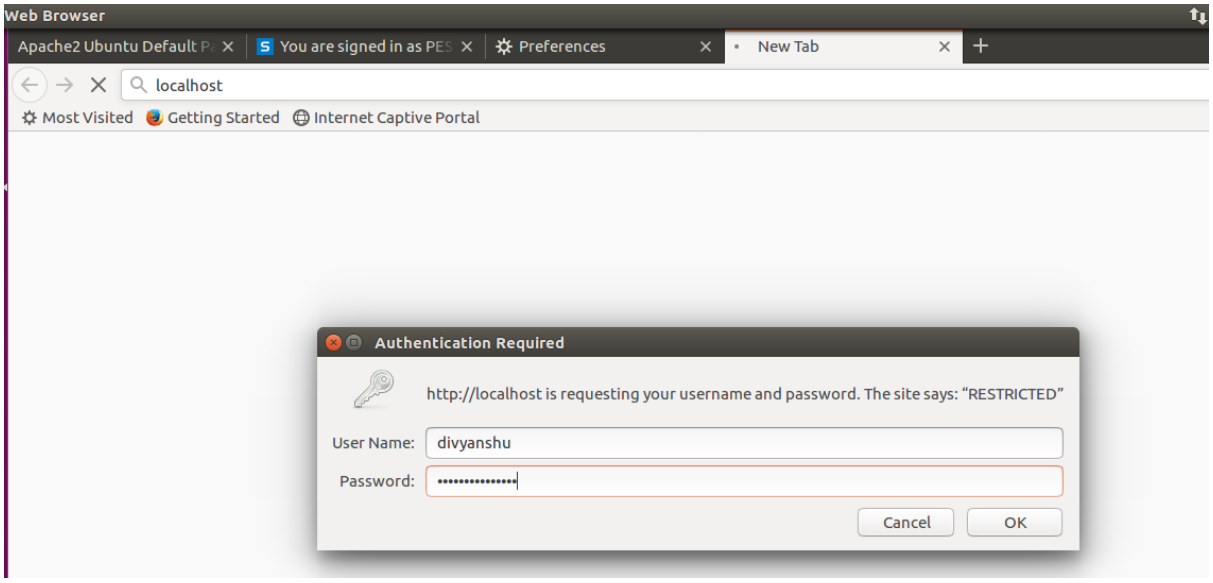
```
student@CSELAB:~$ sudo cat /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        <Directory "/var/www/html">
                AuthType Basic
                AuthName "RESTRICTED"
                AuthUserFile /etc/apache2/.htpasswd
                require valid-user >
        </Directory>
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- Password policy implementation is done by restarting the server as: sudo service apache2 restart

```
student@CSELAB:~/Desktop/CN LAB/WEEK3$ sudo service apache2 restart
student@CSELAB:~/Desktop/CN LAB/WEEK3$
```

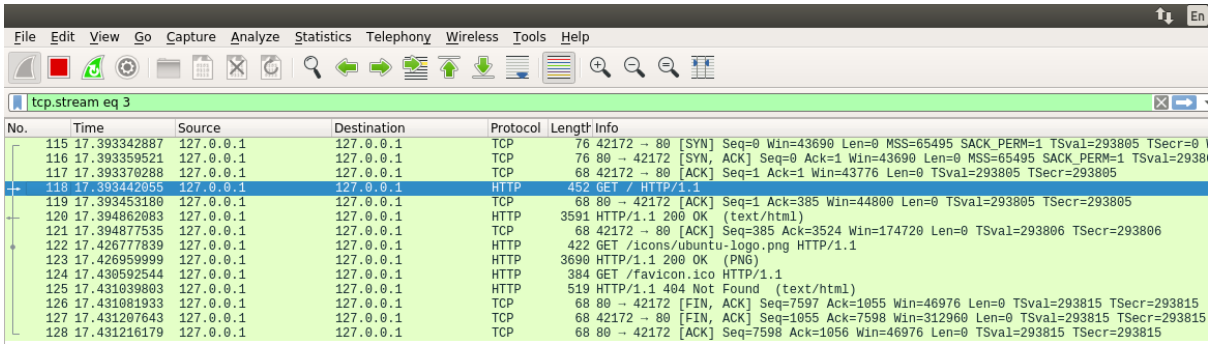
1.3 Accessing Localhost:

- We can now access localhost only after entering the username and password set earlier
- These credentials are entered on the browser window.



1.4 Wireshark Packet Capture:

- Wireshark can be used to capture the packets sent on the network. The f**irst GET request** corresponding to the HTML file is analyzed and its TCP Stream is expanded, and parameters examined.



Wireshark · Follow TCP Stream (tcp.stream eq 3) · any

```
GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic ZGl2eWFuc2h1OmRpdnlhbnNodXNoYXJtYQ==

HTTP/1.1 200 OK
Date: Wed, 10 Feb 2021 08:20:37 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Tue, 31 Jul 2018 04:05:26 GMT
ETag: "2c39-57243b01f87c0-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3186
Keep-Alive: timeout=5, max=2
Connection: Keep-Alive
Content-Type: text/html
```

1.5 <u>Decrypting Base64 Encryption:</u>

- We can observe that the Authorization field stores the password we had entered to access localhost.
- This password is encrypted using the Base64 algorithm before it is transmitted along the network.
    - Each character is **converted into 8-bit binary ASCII representation**
    - Group these bits into **chunks of 6-bits**.
    - **Convert these chunks into their decimal equivalent** and assign the corresponding Base64 character
    - The Base64 algorithm supports the use of lowercase as well as uppercase alphabets, all digits from 0 to 9 and the special characters + and / only.
- Similarly, Base64 is decoded by obtaining the 6-bit binary chunks for each character, grouping them into chunks of 8-bits and then converting into their corresponding character.

ZGl2eWFuc2h10mRpdnlhbnNodXNoYXJtYQ== can be first converted to a 6-bit binary equivalent

Z	011001
G	000110
l	100101
2	110110
e	011110
W	010110
F	000101
u	101110
c	011100

2	110110
h	100001
1	110101
0	110100
m	100110
R	010001
p	101001
d	011101
n	100111
l	100101
h	100001
b	011011
n	100111
N	001101
o	101000
d	011101
X	010111
N	001101
o	101000
Y	011000
X	010111
J	001001
t	101101
Y	011000
Q	010000

- **These binary equivalents can then be grouped together and then decoded to ASCI**

01100100	**d**
01101001	**i**
01110110	**v**
01111001	**y**
01100001	**a**
01101110	**n**
01110011	**s**
01101000	**h**
01110101	**u**
01110011	**s**
01101000	**h**
01100001	**a**

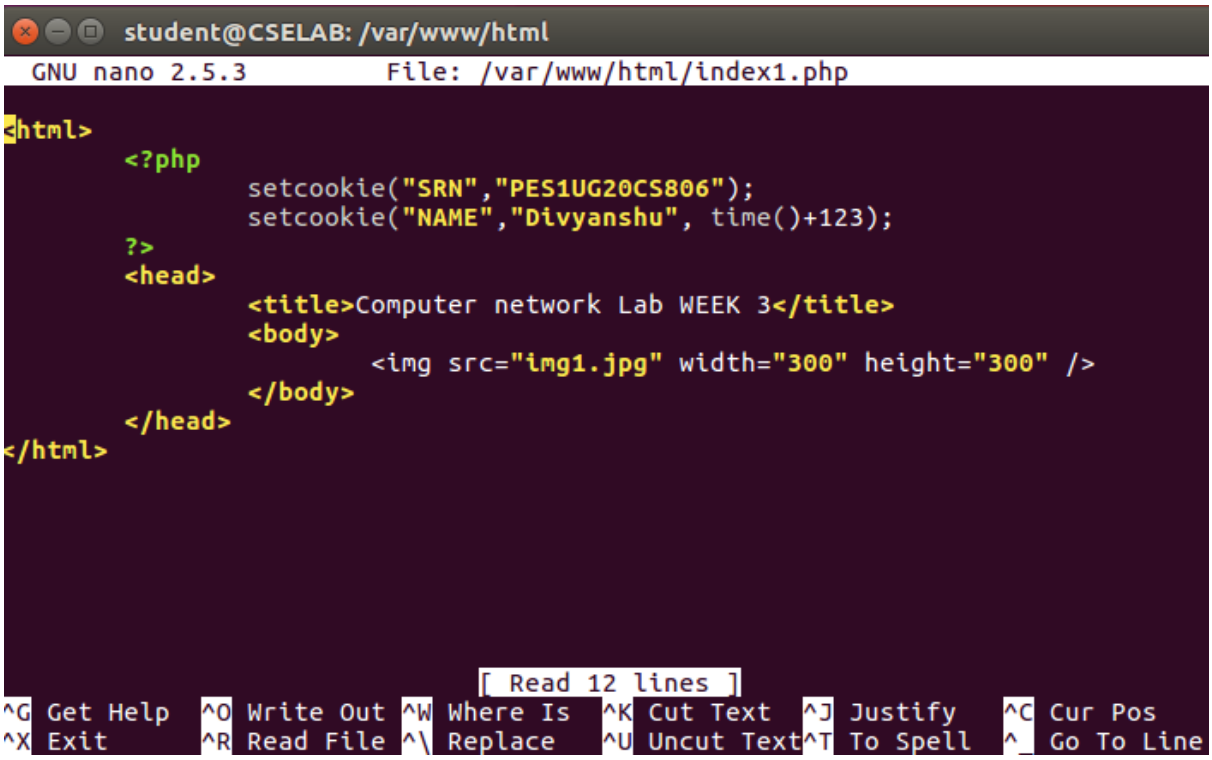
01110010	**r**
01101101	**m**
01100001	**a**

2. Setting Cookies

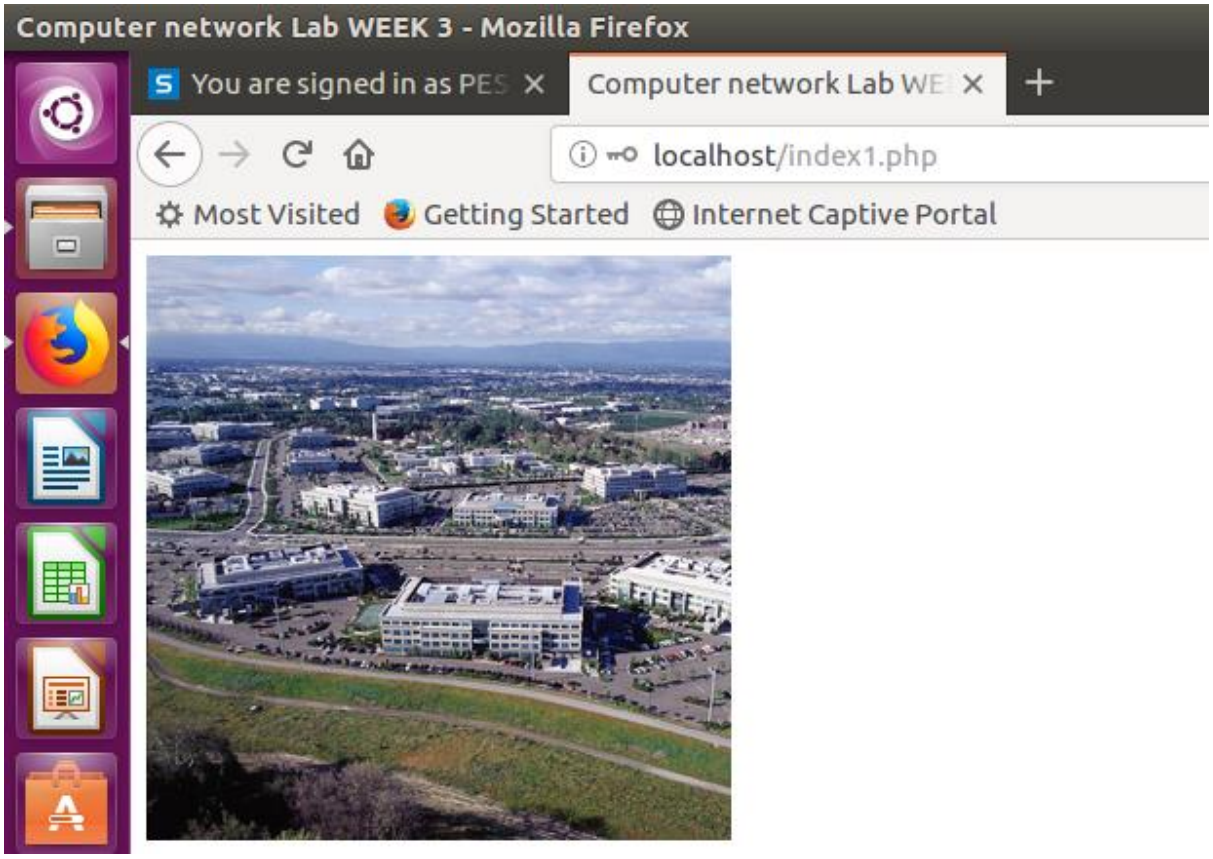
2.1 Setting Cookies with PHP:

- We can set cookies using a PHP script and the **setcookie(name, value, expire_time) function**
- When this file is requested by the browser a cookie will be set

```
<html>
 <?php
        setcookie("SRN","PES1UG20CS806”);
        setcookie("NAME","Divyanshu", time()+123);
?>
    <img src= “img1.jpg” width= “300” height= “300”  />
</html>
```

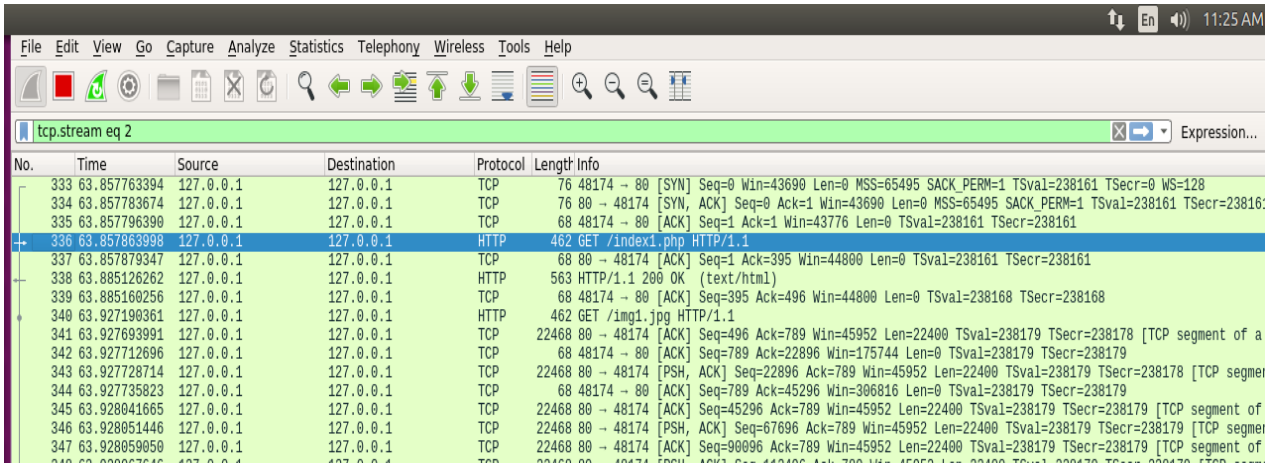


- The combined file saved with a .php extension is placed under /var/www/html for accessing.



2.2 Wireshark Capture

- Wireshark can be used to capture the packets sent on the network. The first GET request corresponding to the PHP file is analyzed and its TCP Stream is expanded and examined.
- The Cookie name, value and the associated parameters can be viewed under the HTTP header Set-Cookie.
- We can observe the name, value, and the expiry time of the set cookie, if the cookie has not already expired.



Wireshark · Follow TCP Stream (tcp.stream eq 2) · any

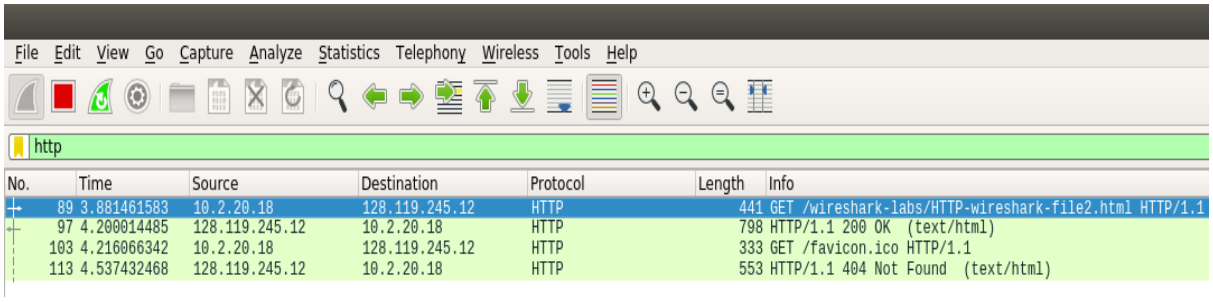
```
GET /index1.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic ZGl2eWFuc2h1OmRpdnlhbnNodXNoYXJtYQ==

HTTP/1.1 200 OK
Date: Thu, 11 Feb 2021 05:53:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Set-Cookie: SRN=PES1UG20CS806
Set-Cookie: NAME=Divyanshu; expires=Thu, 11-Feb-2021 05:55:27 GMT; Max-Age=123
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 134
Keep-Alive: timeout=5, max=2
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

............(........HML.3J2KrR...s.JKR...RK.....|....]]...m.!J@...S*A.N...t...d[% .P/. ]
#53=........j....
.....
|F.....GET /img1.jpg HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Authorization: Basic ZGl2eWFuc2h1OmRpdnlhbnNodXNoYXJtYQ==
Connection: keep-alive
Referer: http://localhost/index1.php
Cookie: SRN=PES1UG20CS806; NAME=Divyanshu

HTTP/1.1 200 OK
```

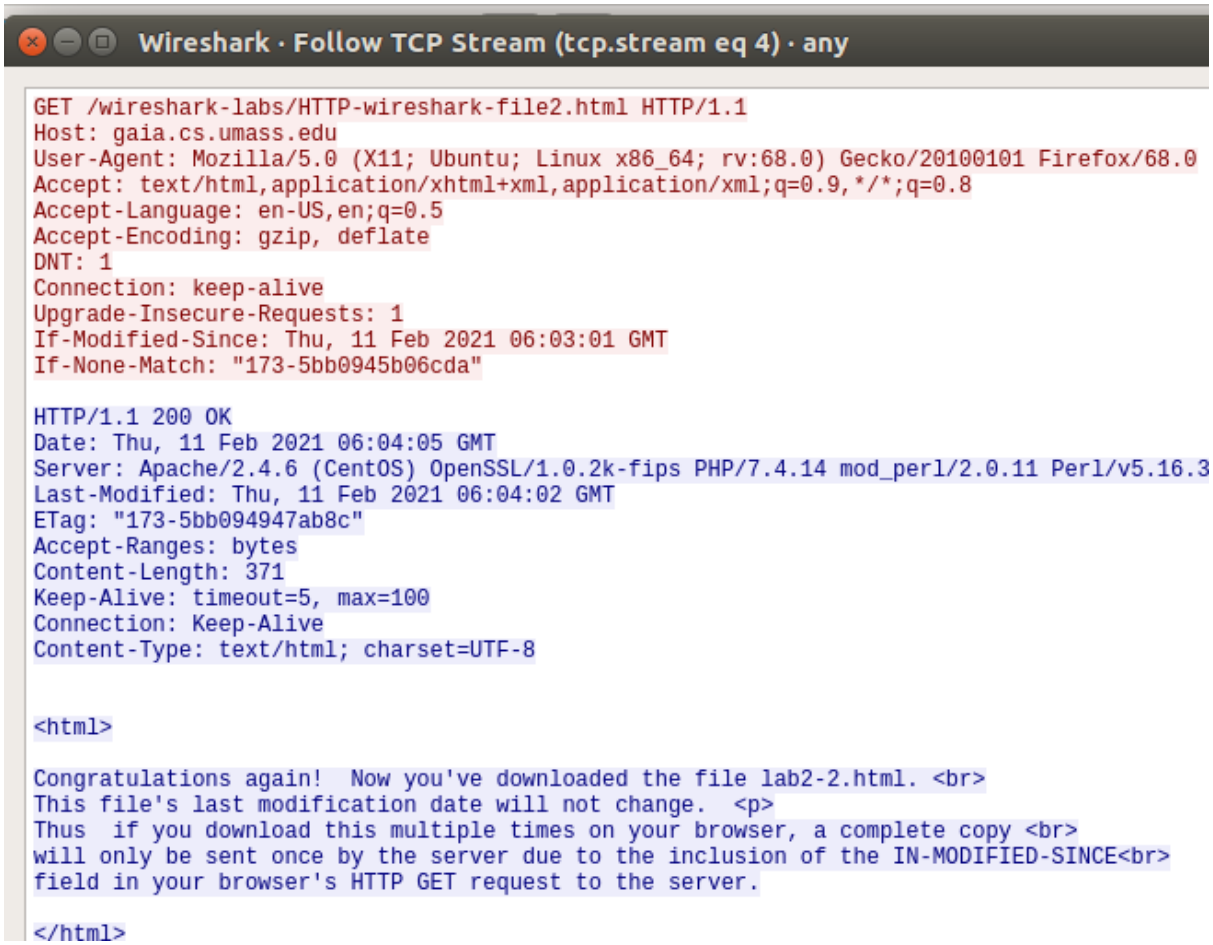
3. Conditional GET

- A conditional HTTP response is one that carries the resource only it had been modified since the last GET request by the client.
- The HTTP header **If-Modified**-Since is one way to implement Conditional GET
- The server checks the If-Modified-Since header value and resends the resource only if it has been modified since the timestamp in the header
- If it has not been modified, **a 304 Not Modified** status code is sent back.

3.1 Repeat Requests for HTML Page

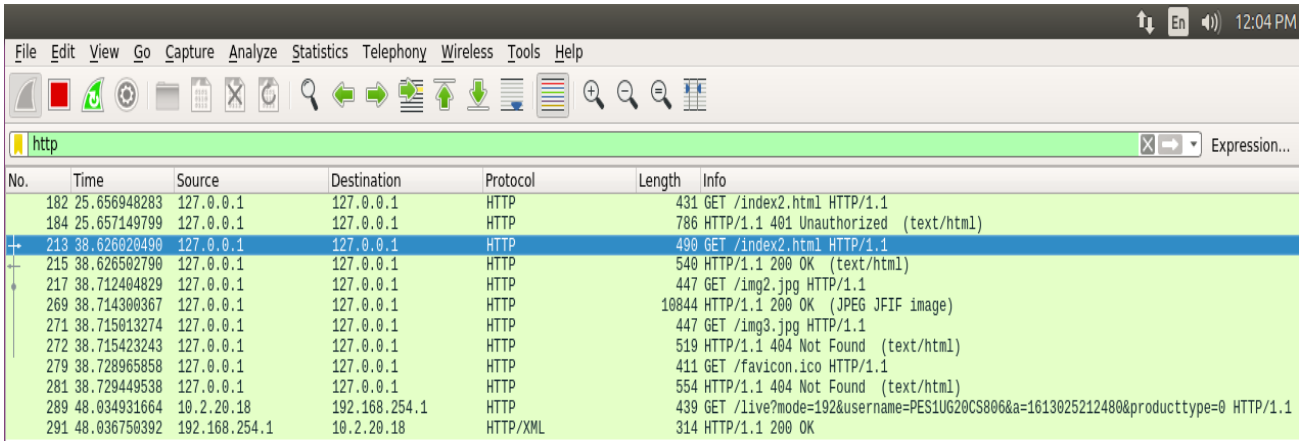
- An HTML page is requested by the client and the HTML file is obtained along with a 200 OK response status
- Immediately, the request is made again either by refreshing or accessing it via a browser tab
- The second response from the server is obtained as 304 Not Modified since the resource has not been modified since the last GET.





3.2 <u>Conditional GET on Localhost</u>

- A simple HTML file with 2 images is placed in the localhost home directory.
- From a browser, a request is made for the file, which receives a response of 200 OK with both images being sent by the server.
- When the request is sent again, the 304 Not Modified status code is sent and images are not sent back.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
182	25.656948283	127.0.0.1	127.0.0.1	HTTP	431	GET /index2.html HTTP/1.1
184	25.657149799	127.0.0.1	127.0.0.1	HTTP	786	HTTP/1.1 401 Unauthorized (text/html)
213	38.626020490	127.0.0.1	127.0.0.1	HTTP	490	GET /index2.html HTTP/1.1
215	38.626502790	127.0.0.1	127.0.0.1	HTTP	540	HTTP/1.1 200 OK (text/html)
217	38.712404829	127.0.0.1	127.0.0.1	HTTP	447	GET /img2.jpg HTTP/1.1
269	38.714300367	127.0.0.1	127.0.0.1	HTTP	10844	HTTP/1.1 200 OK (JPEG JFIF image)
271	38.715013274	127.0.0.1	127.0.0.1	HTTP	447	GET /img3.jpg HTTP/1.1
272	38.715423243	127.0.0.1	127.0.0.1	HTTP	519	HTTP/1.1 404 Not Found (text/html)
279	38.728965858	127.0.0.1	127.0.0.1	HTTP	411	GET /favicon.ico HTTP/1.1
281	38.729449538	127.0.0.1	127.0.0.1	HTTP	554	HTTP/1.1 404 Not Found (text/html)
289	48.034931664	10.2.20.18	192.168.254.1	HTTP	439	GET /live?mode=192&username=PES1UG20CS806&a=1613025212480&producttype=0 HTTP/1.1
291	48.036750392	192.168.254.1	10.2.20.18	HTTP/XML	314	HTTP/1.1 200 OK

Wireshark · Follow TCP Stream (tcp.stream eq 1) · any

```
GET /index2.html HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Cookie: SRN=PES1UG20CS806
Upgrade-Insecure-Requests: 1
Authorization: Basic ZGl2eWFuc2h1OmRpdnlhbnNodXNoYXJtYQ==

HTTP/1.1 200 OK
Date: Thu, 11 Feb 2021 06:33:23 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Thu, 11 Feb 2021 06:25:53 GMT
ETag: "c4-5bb09976ff621-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 138
Keep-Alive: timeout=5, max=2
Connection: Keep-Alive
Content-Type: text/html

............(.......HML...6%.%9.v.....%.E
y.%..E.
>.I
....
.6..%6.p=I.). ..Mfn.BqQ....a..U...P..R.a.dl`.......Q...cj1&F..>.>.............GET /img2.jpg HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Authorization: Basic ZGl2eWFuc2h1OmRpdnlhbnNodXNoYXJtYQ==
Connection: keep-alive
Referer: http://localhost/index2.html
Cookie: SRN=PES1UG20CS806
```

```
4.B....4.y..z..\.\.d)"..(.W...+Oz....95.o.?.?..5.......GET /img3.jpg HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Authorization: Basic ZGl2eWFuc2h1OmRpdnlhbnNodXNoYXJtYQ==
Connection: keep-alive
Referer: http://localhost/index2.html
Cookie: SRN=PES1UG20CS806

HTTP/1.1 404 Not Found
Date: Thu, 11 Feb 2021 06:33:23 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 271
Connection: close
Content-Type: text/html; charset=iso-8859-1
```