Computer Networks **Lab #3**

Name : Laxmikant Bhujang Gurav
Section : K
Roll no : 55
SRN : PES1UG20CS658

Understanding Persistent and Non-persistent HTTP Connections

Experiment: Create a web page with N (e.g. 10) embedded images. Each image should be ofminimum 2 MB size. Configure your browser (Firefox) with following settings (each setting requires repeat of experiment)

a) Non-persistent connection
b) 2 persistent connections
c) 4 persistent connections
d) 6 persistent connections
e) 10 persistent connections

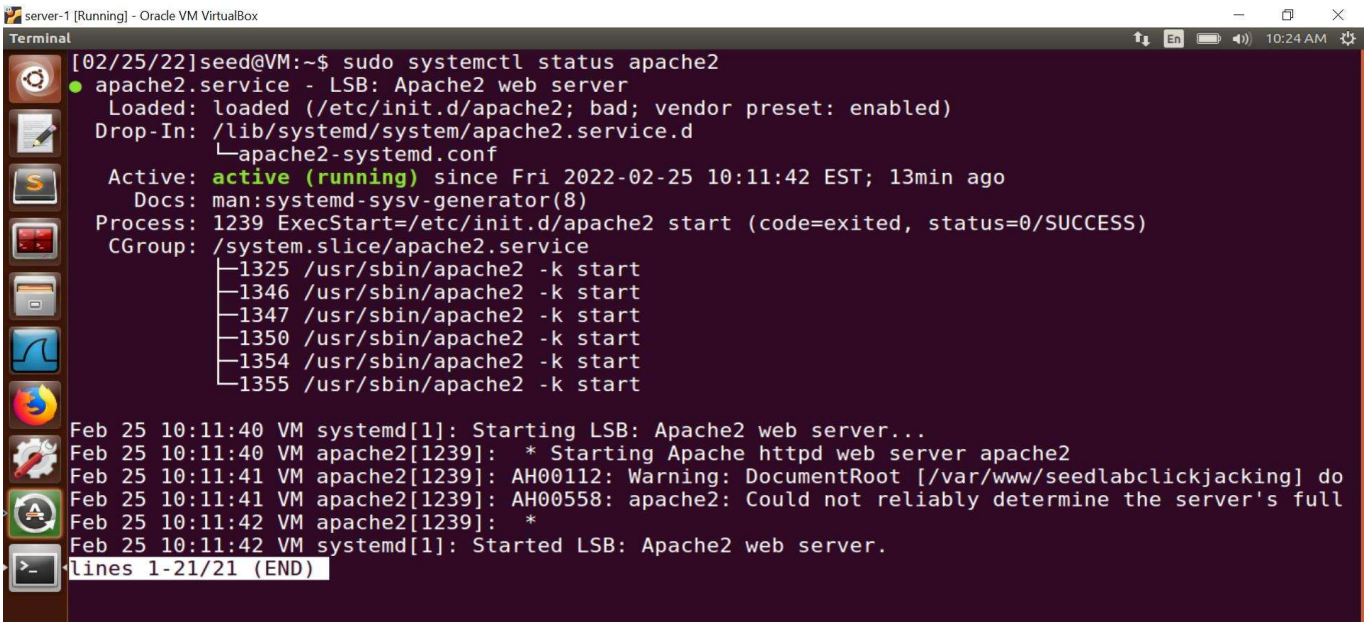
EXECUTION STEPS

Step 1: Connect 2 desktops using switch and cables as shown below. (Use 2 VMs onVirtualbox or VMware instead of physical connections.)

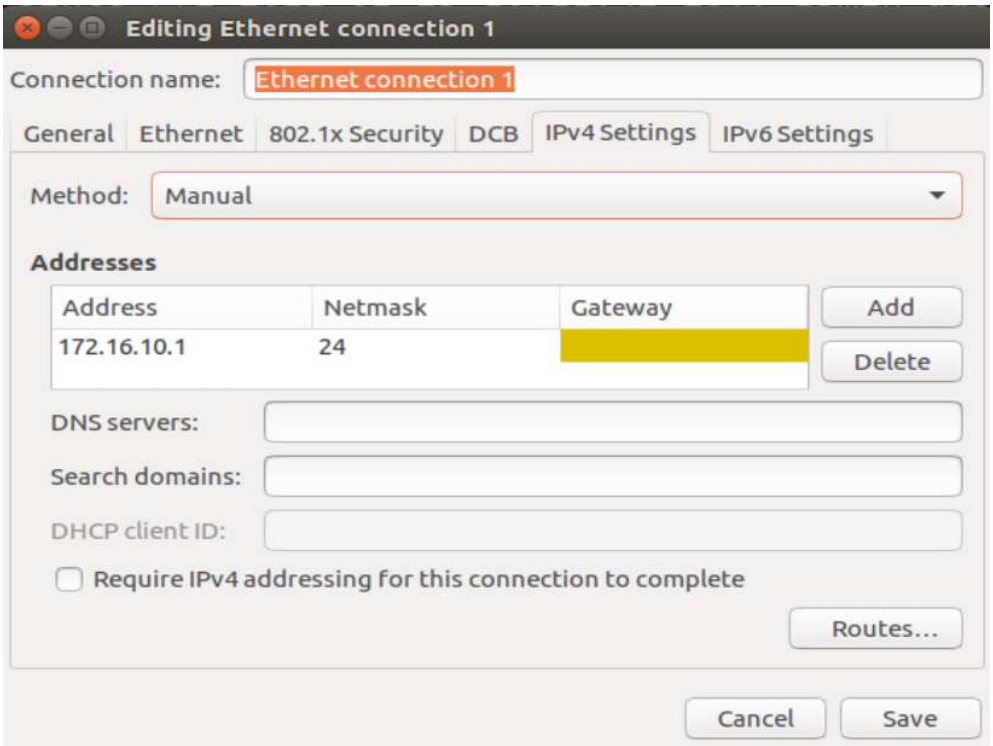
<u>Server **Side:**</u>

Step 2: Check your Web Server

sudo systemctl status apache2



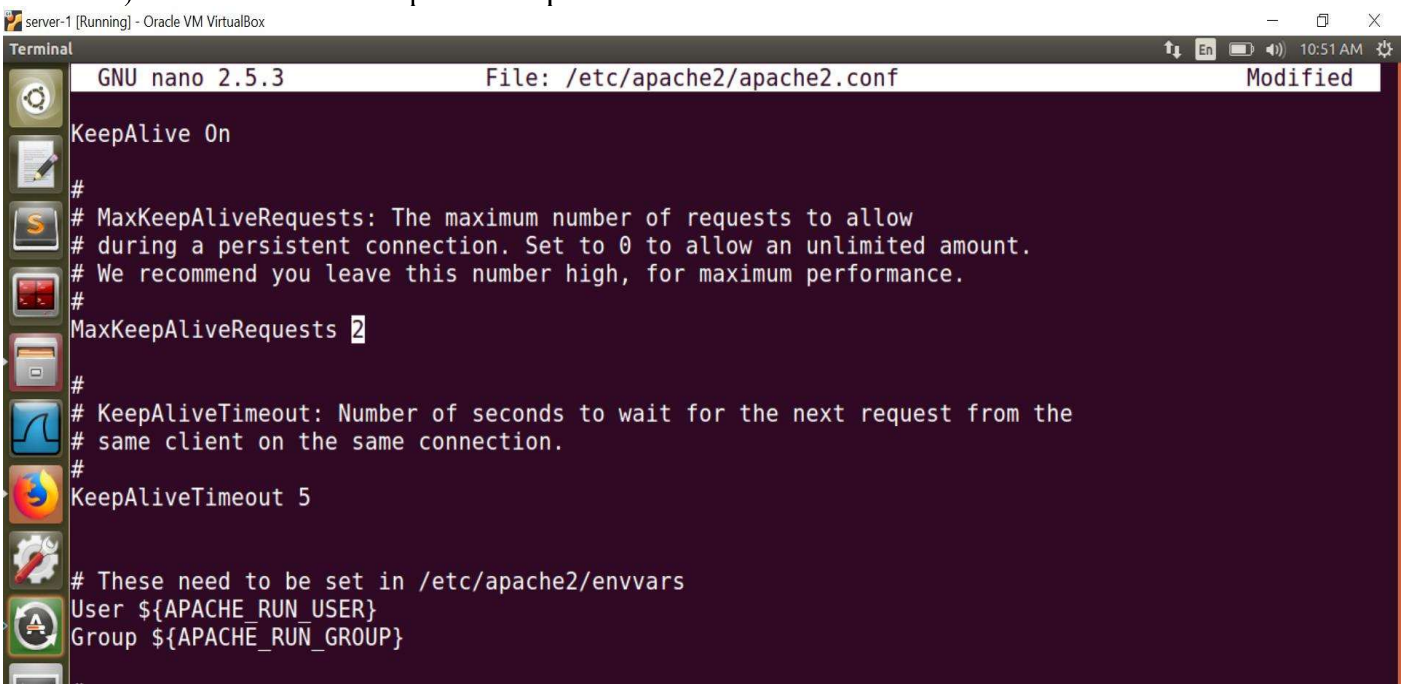
Step 3: Server IP address can be set by the following command
$sudo ip addr add 172.16.10.1/24 dev enps0
$sudo ip addr



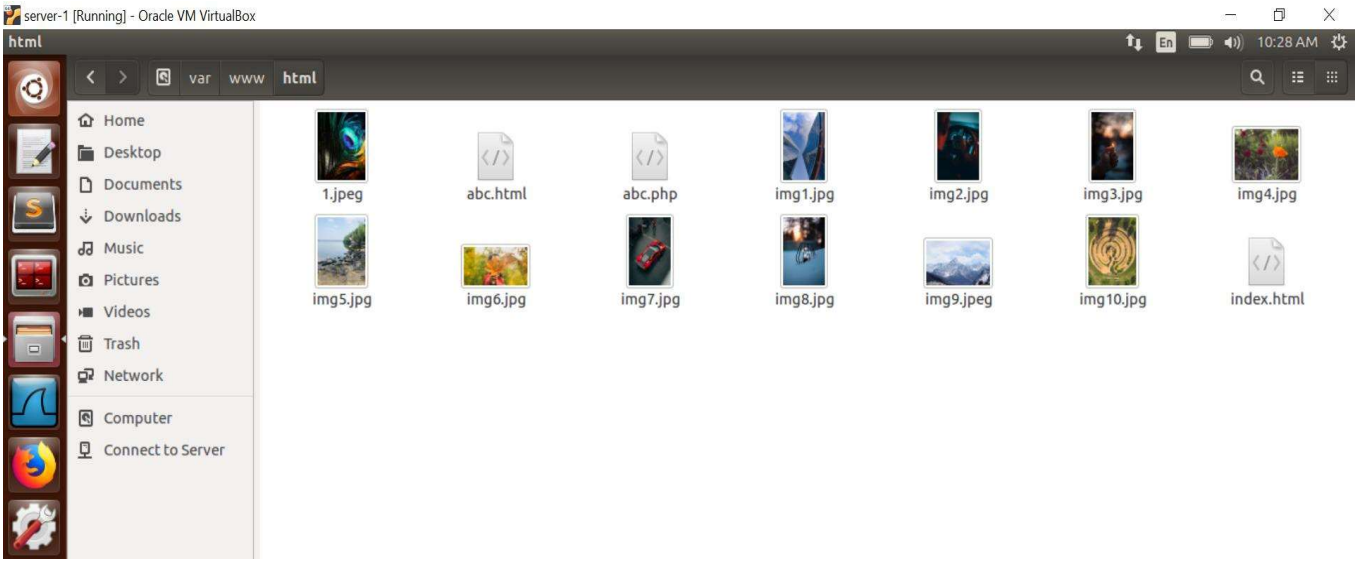
Step 4: The apache2.conf file present in the etc/apache2 directory is modified as:

a) The keep-alive option was set (i.e. value was made ON)

b) The MaximumKeepAliveRequests were set to 2



Step 5: Store images in the server path

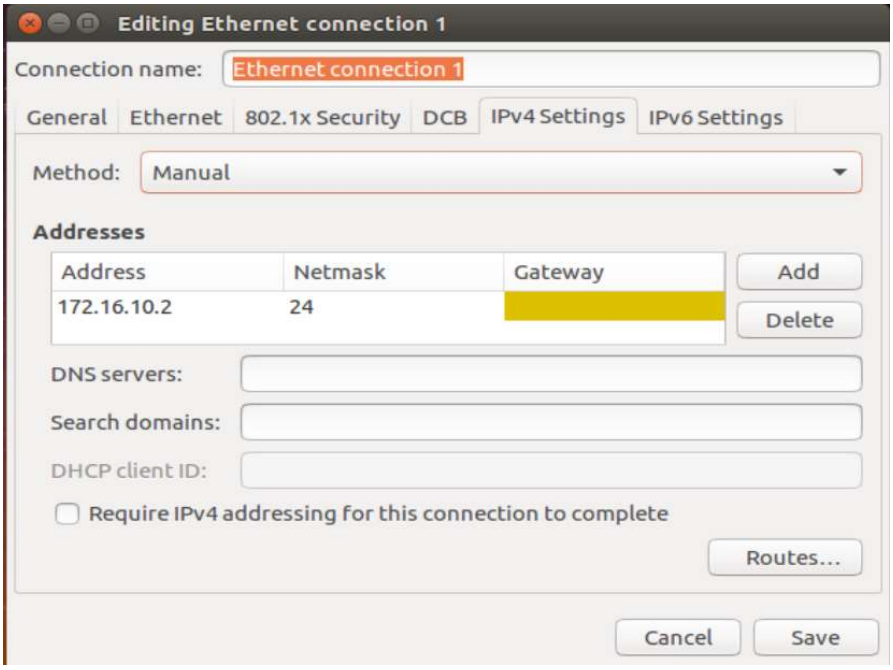


Step 6: Prepare a web page as shown below. The html file needs to add 10 images. (Kindlyskip the style attribute in the below image)

Client side:

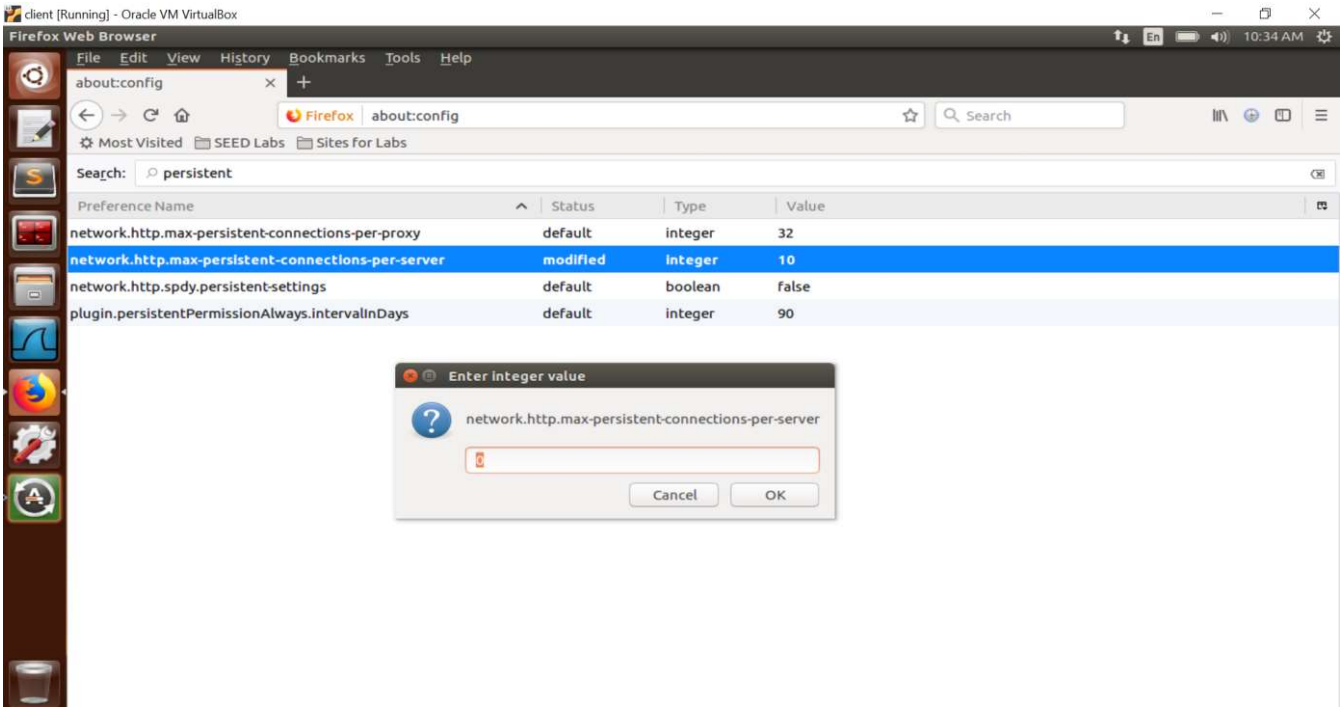
Client IP address can be set by the following command.

```
$sudo ip addr add 172.16.10.2/24 dev enps0
$sudo ip addr
```

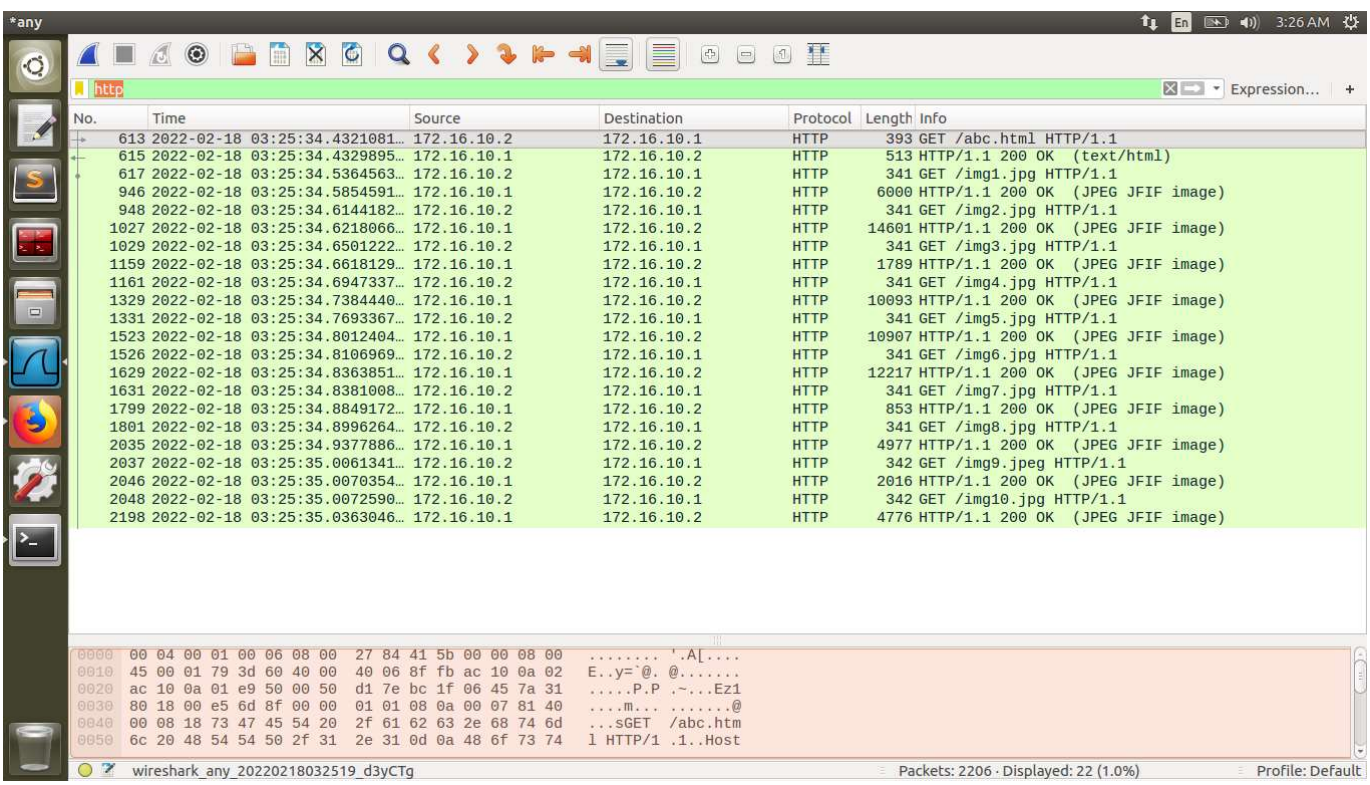


PART 1: NON-PERSISTENT CONNECTION

Step 1: This is done by setting the value of max-persistent-connection-per-server to 0 in the client computer.



Step 2: Access web page on client-side browser (Firefox)

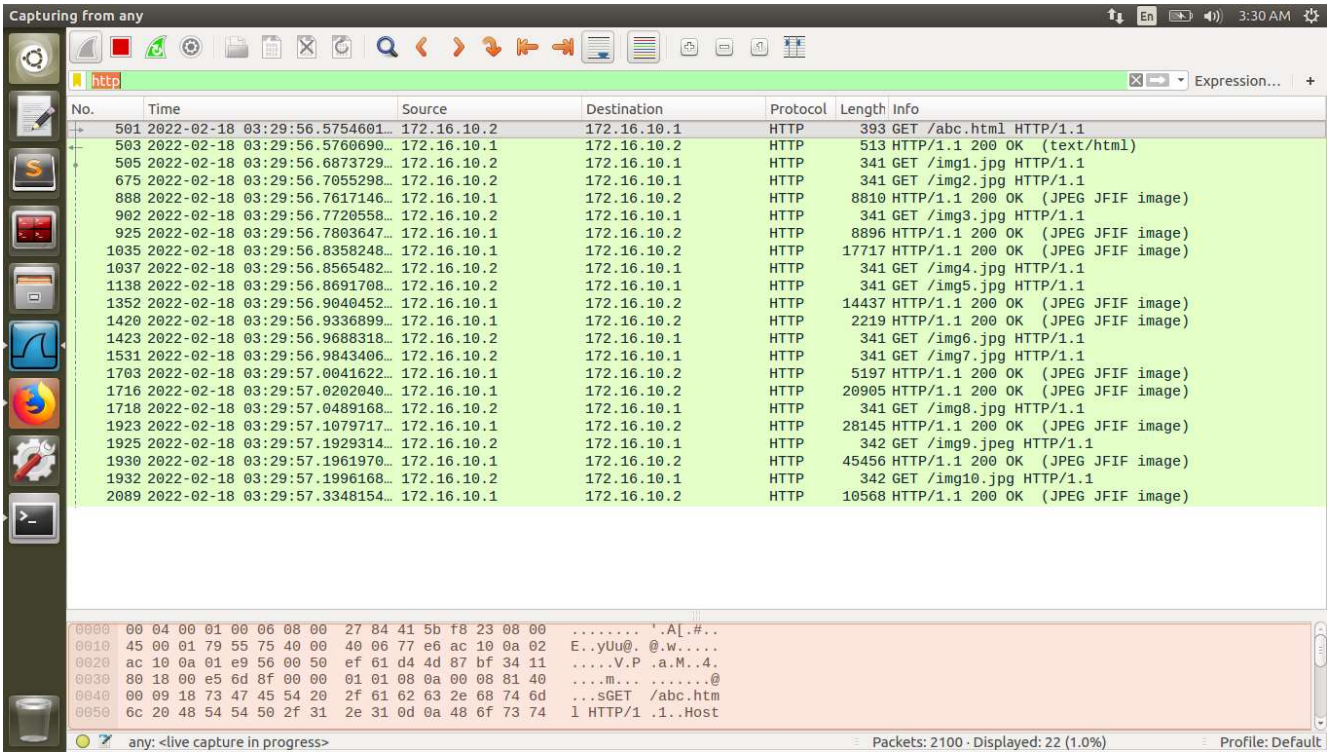


Here Time difference is = 35.0363046 – 34.4321081 = 0.6041965

PART 2: PERSISTENT CONNECTIONS

Step 1: For 2 persistent connections, set the value of max-persistent-connection-per-serverto 2 in the client computer.

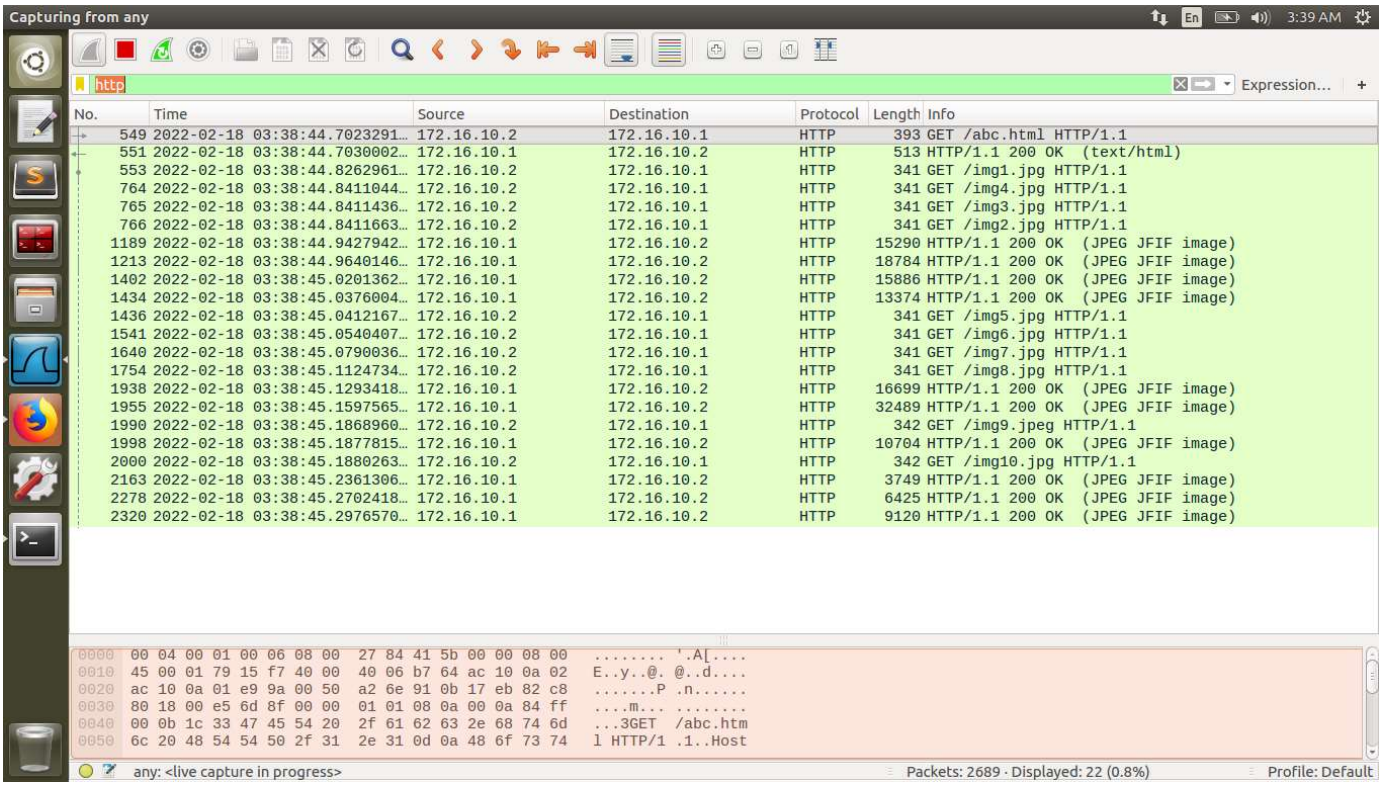
Step 2: Repeat the steps 1-3 in the previous section.



Here Time difference = 57.3348154 – 56.5754601 = 0.7593553

Step 3: For 4 persistent connections, Set the value of max-persistent-connection-per-serverto 4 in the client computer.

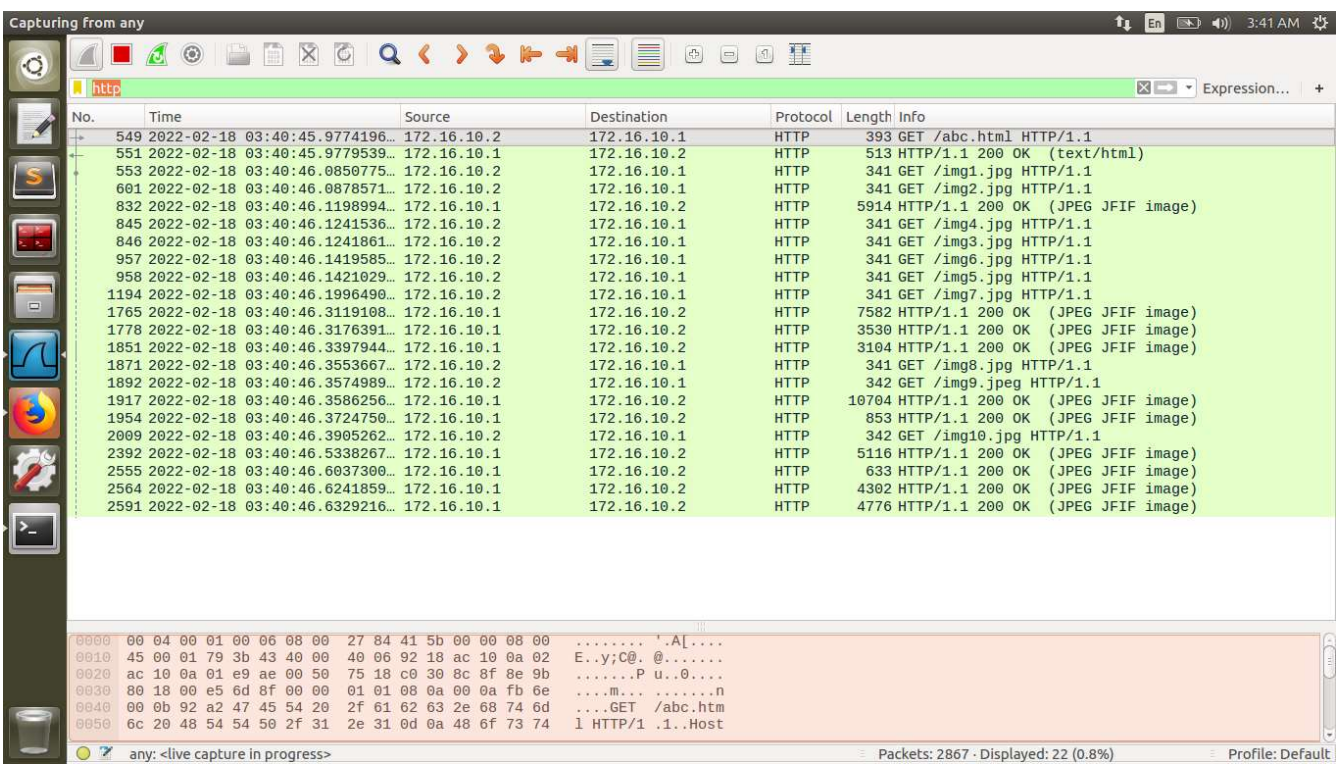
Step 4: Repeat the steps 1-3 in the previous section



Here Time difference = 45.2976570 – 44.7023291 = 0.5953279

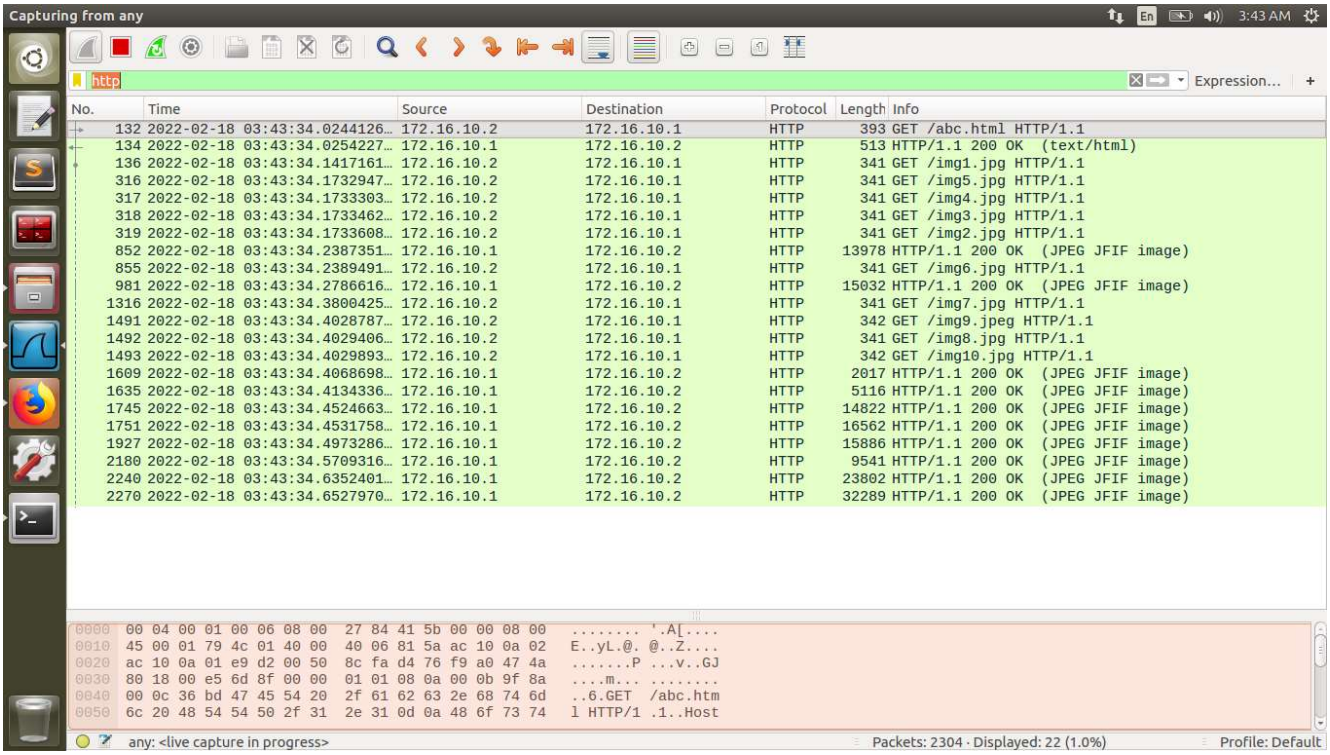
Step 5: For 6 persistent connections, set the value of max-persistent-connection-per-serverto 6 in the server computer.

Step 6: Repeat the steps 1-3 in the previous section.



Here Time difference = 46.6329216 – 45.9774196 = 0.6555020

Step 7: For 10 persistent connections, set the value of max-persistent-connection-per-server to 10 in the client computer.



Here Time difference = 34.6527970– 34.0244126 = 0.6283844

Sl.no	Max-Persistent-Connection-per-server	Load Time
1	0	0.6041965
2	2	0.7593553
3	4	0.5953279
4	6	0.6555020
5	10	0.6283844

Conclusion Remarks : It can be observered the load time is minimum for the 4 -persistent-connection. Hence it can be considered as the optimum persistent connection.

Understand working of HTTP Headers

Authentication: Auth-Basic

Steps of Execution (for Password Authentication)

1. Executing the below commands on the terminal.

→Provide username and password to set authentication
→View the Authentication

```
laxmikant@MyPC:~$ sudo htpasswd -c /etc/apache2/.htpasswd nobi
New password:
Re-type new password:
Adding password for user nobi
laxmikant@MyPC:~$ sudo cat /etc/apache2/.htpasswd
nobi:$apr1$tpqHVSti$Av3SfWF9YCFqmUNO8VNkT.
laxmikant@MyPC:~$
```

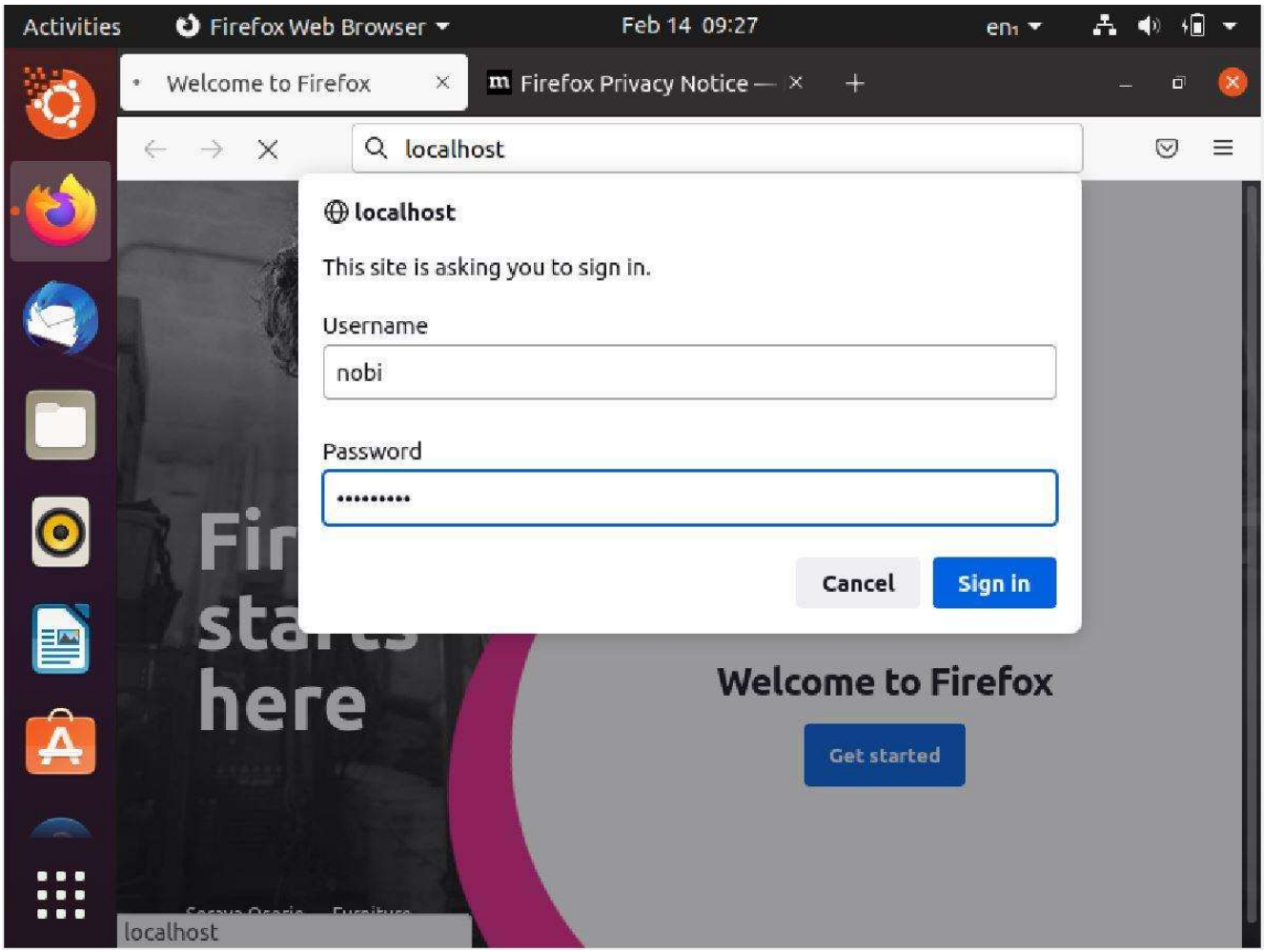
→Opening the file for setting authentication

```
server-1 [Running] - Oracle VM VirtualBox
Terminal                                                                    12:21 PM
  GNU nano 2.5.3        File: /etc/apache2/sites-available/000-default.conf        Modified

<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        <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
<VirtualHost *:80>
        ServerName http://www.SeedLabSQLInjection.com
        DocumentRoot /var/www/SQLInjection
</VirtualHost>
```

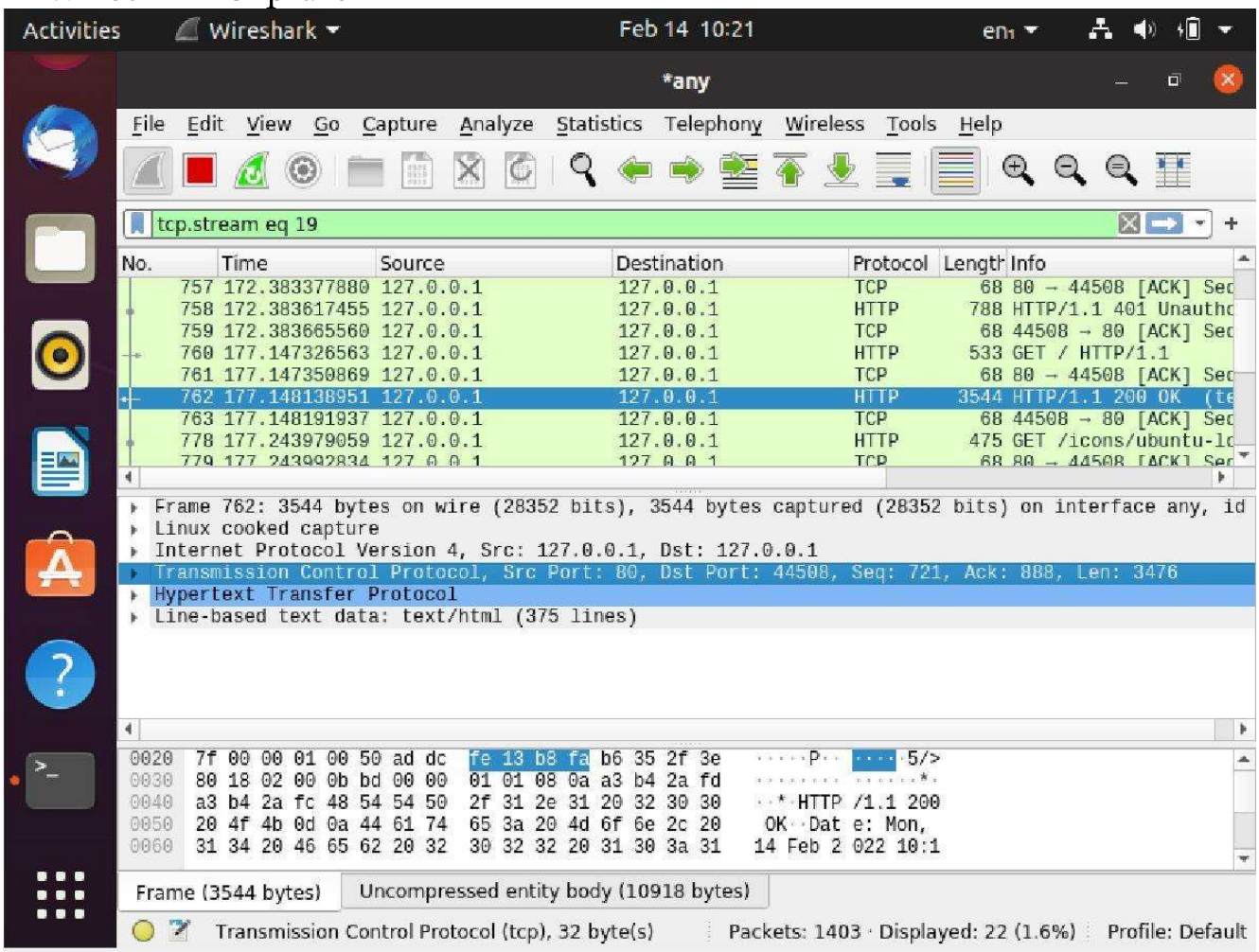
→Restarting the Server

```
laxmikant@MyPC:~$ sudo service apache2 restart
[sudo] password for laxmikant:
laxmikant@MyPC:~$
```

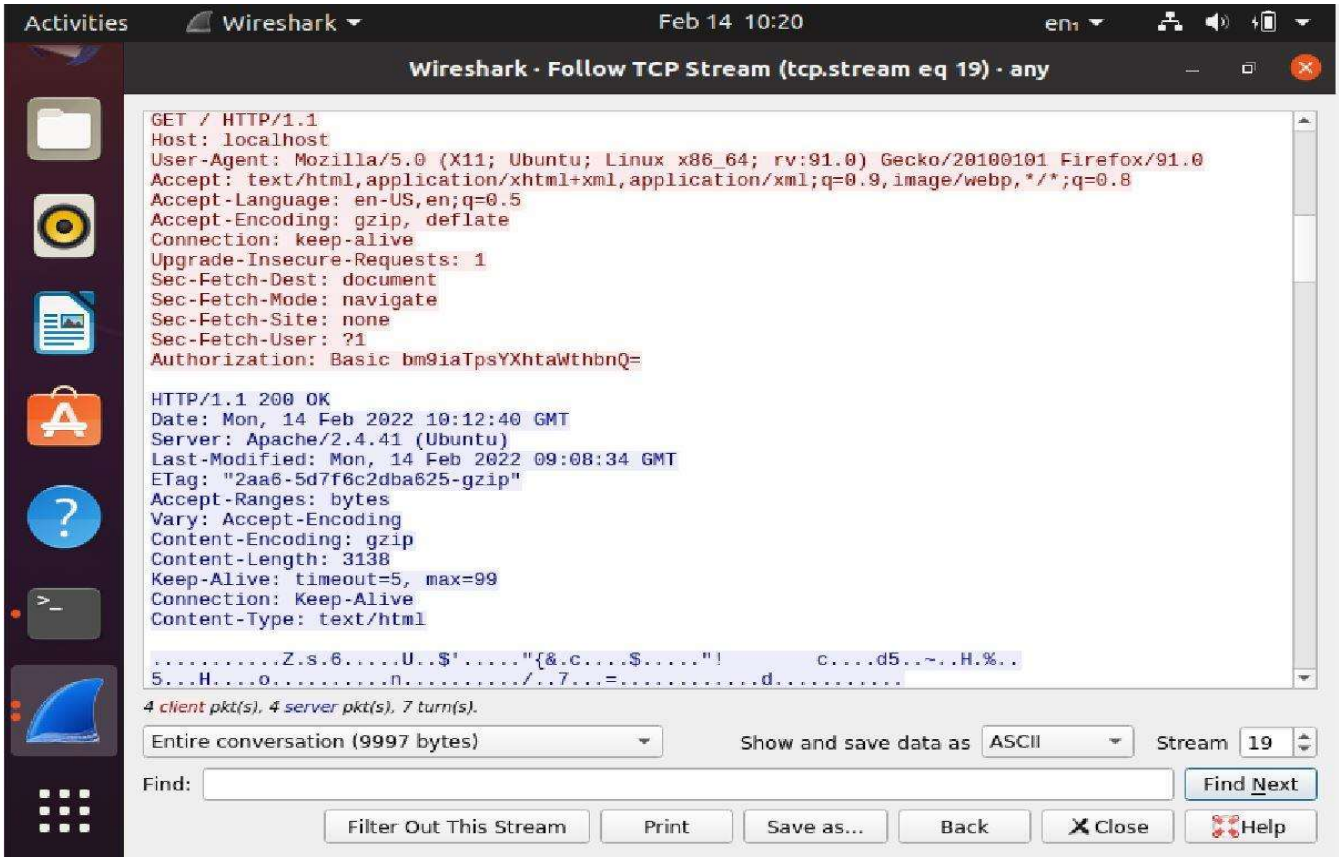
→Access Localhost using Firefox Browser



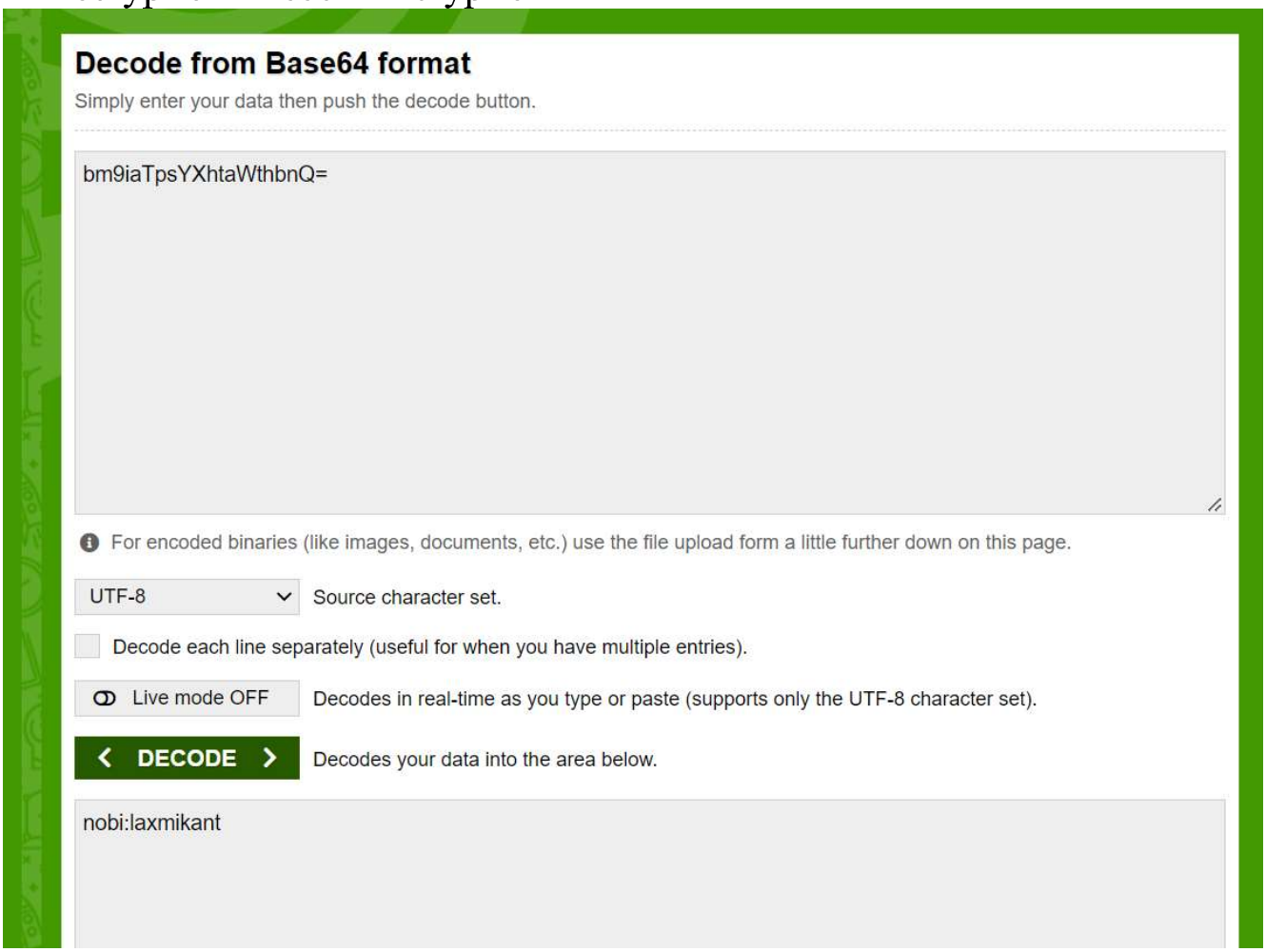
→Wireshark Capture



→Follow TCP Stream using HTTP message


Activities
Wireshark
Feb 14 10:20
Wireshark · Follow TCP Stream (tcp.stream eq 19) · any
GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Authorization: Basic bm9iaTpsYXhtaWthbnQ=
HTTP/1.1 200 OK
Date: Mon, 14 Feb 2022 10:12:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Mon, 14 Feb 2022 09:08:34 GMT
ETag: "2aa6-5d7f6c2dba625-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3138
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html
..........Z.s.6.....U..$'....."{&.c....$....."! c....d5..~..H.%..
5...H....o.........n........../..7...=...........d..........
4 client pkt(s), 4 server pkt(s), 7 turn(s).
Entire conversation (9997 bytes)
Show and save data as
ASCII
Stream
19
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help

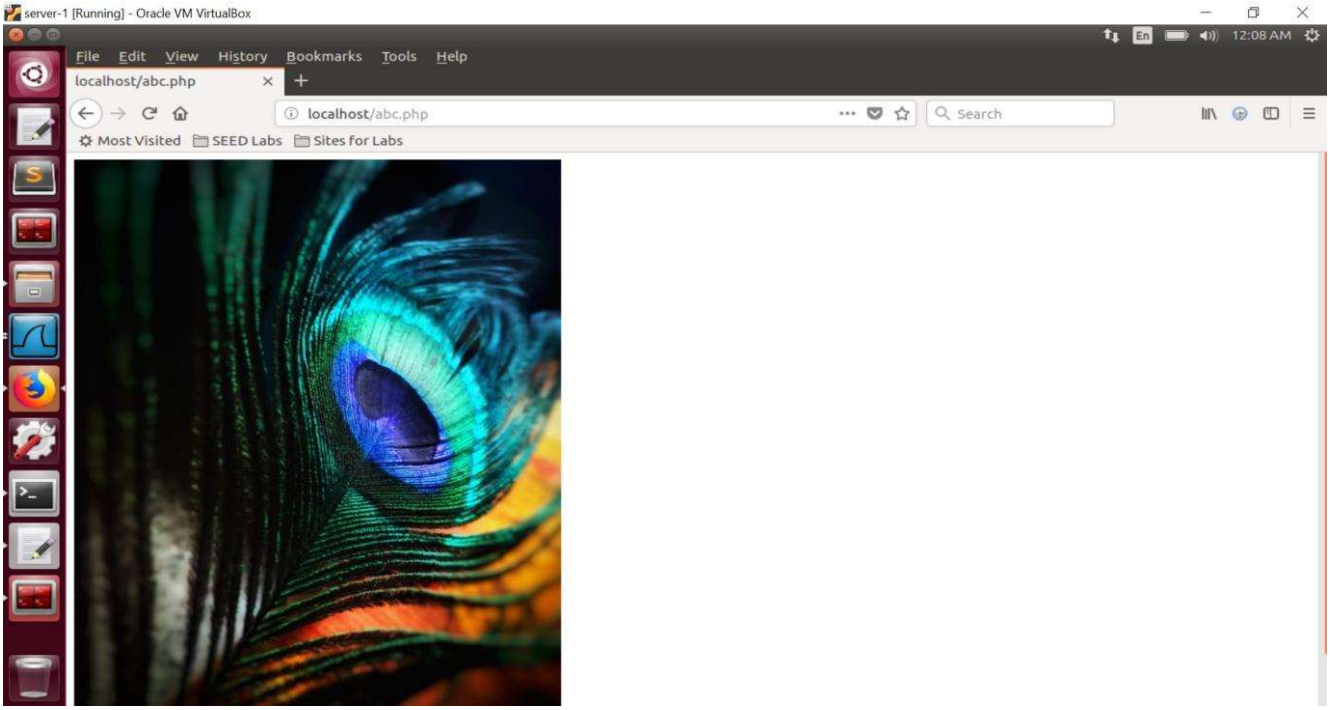
→Decryption Base64 Encryption



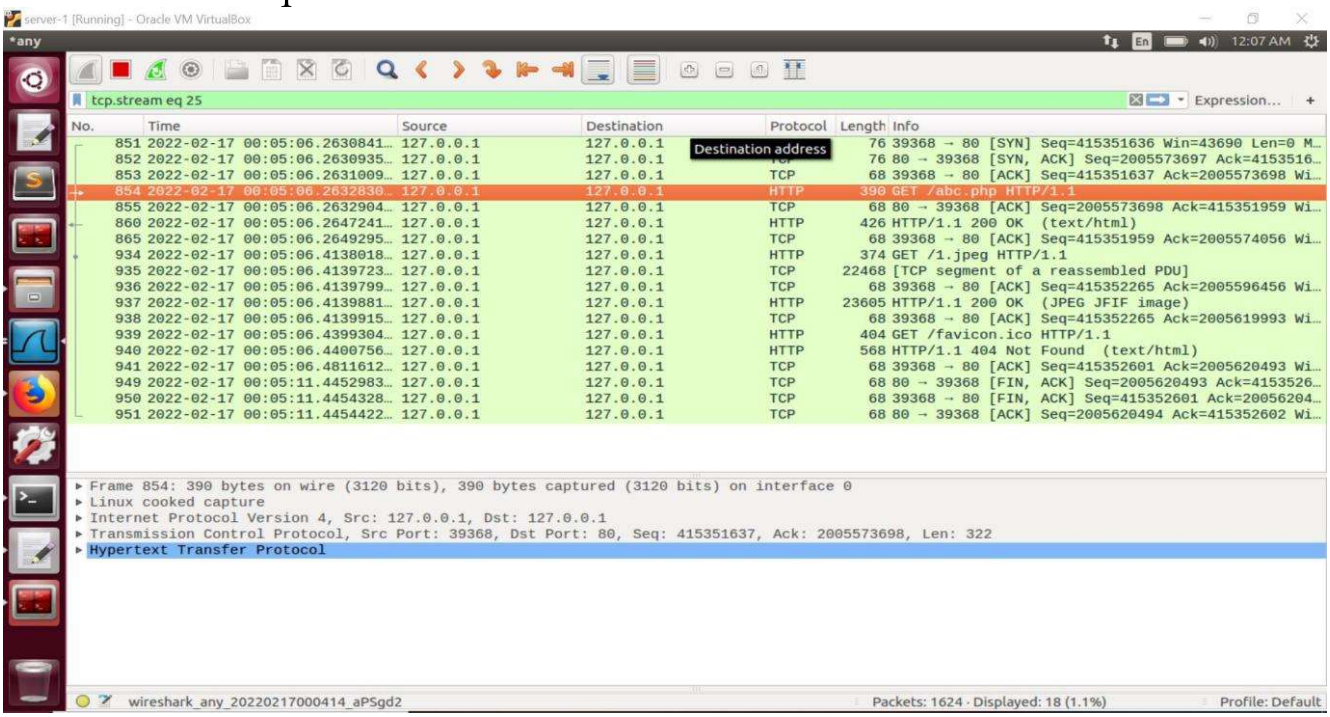
Steps of Execution (Cookie Setting)

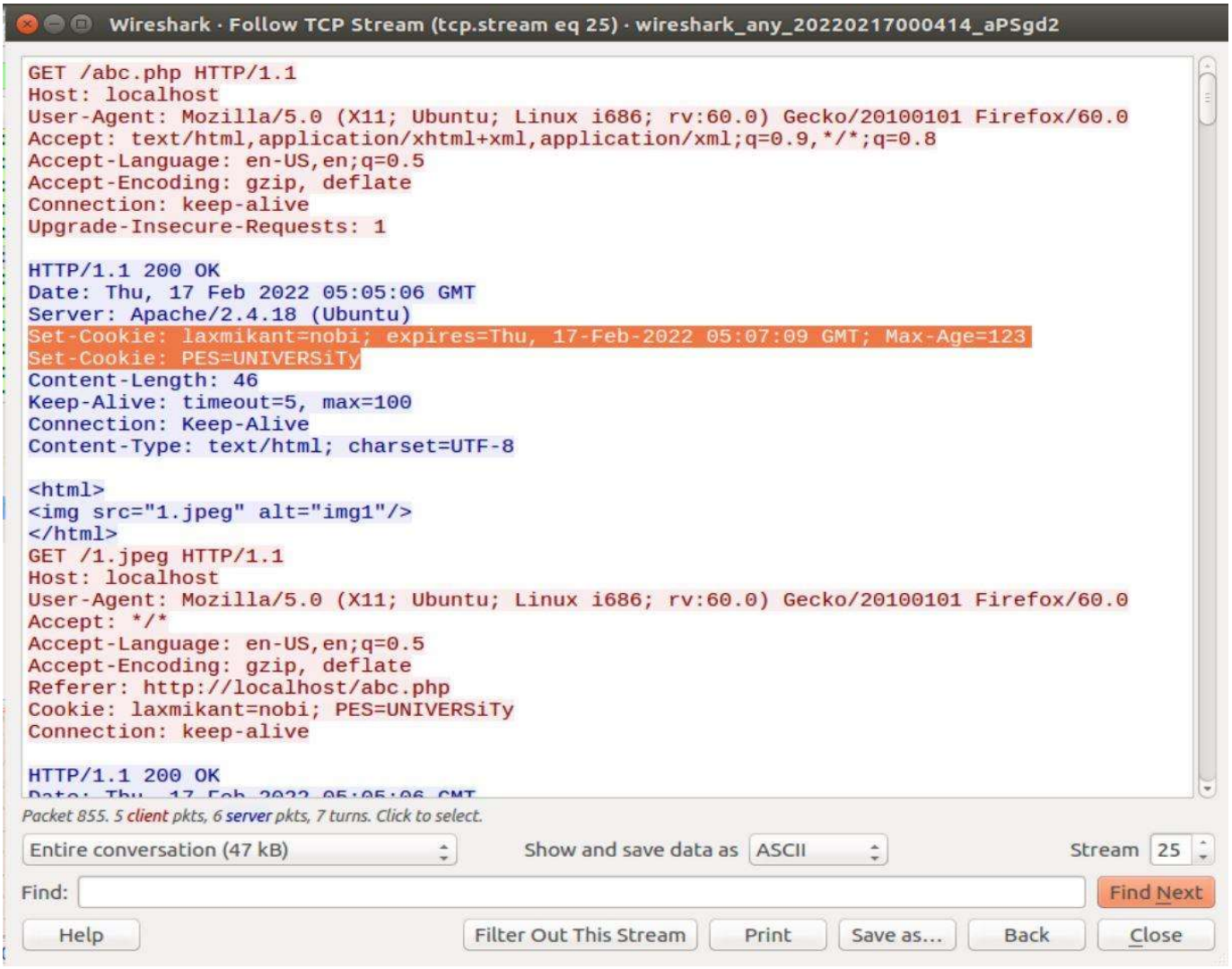
→PHP File

→Access the Php file using Firefox Browser



→Wireshark Capture And follow TCP Stream





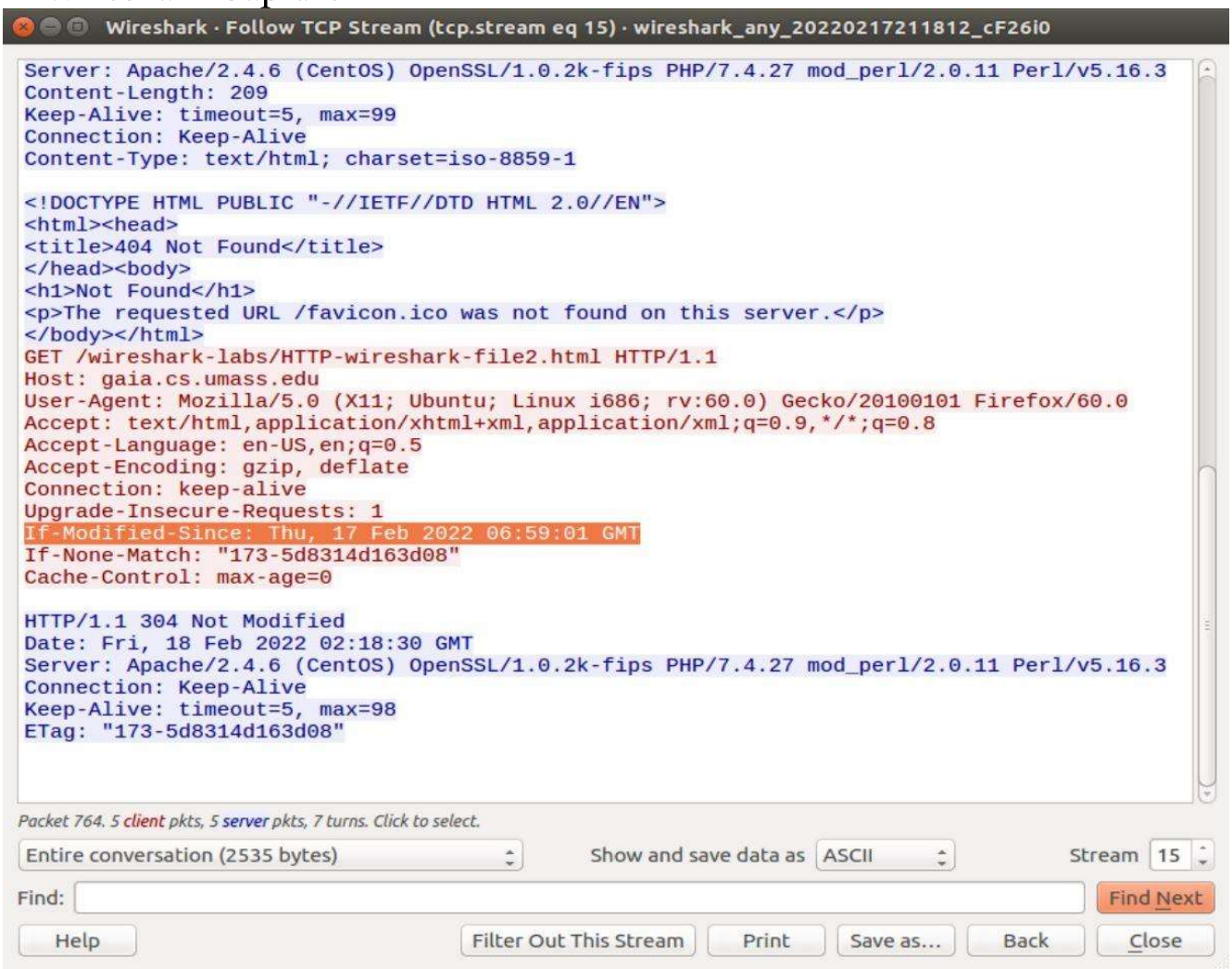
Conditional Get: If-Modified-Since

→Open the Browser And follow the link

Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-

→Before following the link clear the cache.

→Wireshark Capture



Question and Answers :

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?

Ans : No, there is no “IF-MODIFIED-SINCE” line in the first HTTP GET.

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

Ans : Yes, the server explicitly returns the contents of the file. we can confirm from header line-based text data.

3) Do you see an “IF-MODIFIED-SINCE:” line in the second HTTP GET request from your browser to the server? If so, what information follows the “IF-MODIFIED-SINCE:” header?
Ans : Yes, there is an “IF-MODIFIED-SINCE:” line in the second HTTP GET request. It contains the time at which 1st response send by a server to the client.

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.

Ans : 304 is the status code and Not modified is the phrase returned from the server in response. The server doesn’t explicitly return the contents of the file so we can’t find a line-based text data line in the header and the file is loaded from the cache.