

Name: GAURAV MAHAJAN	SRN: PES1UG20CS150	Section: C
	Date: 09-02-2022	WEEK 3

3A. Understanding Working of HTTP Headers

Steps of Execution (for Password Authentication) 1.

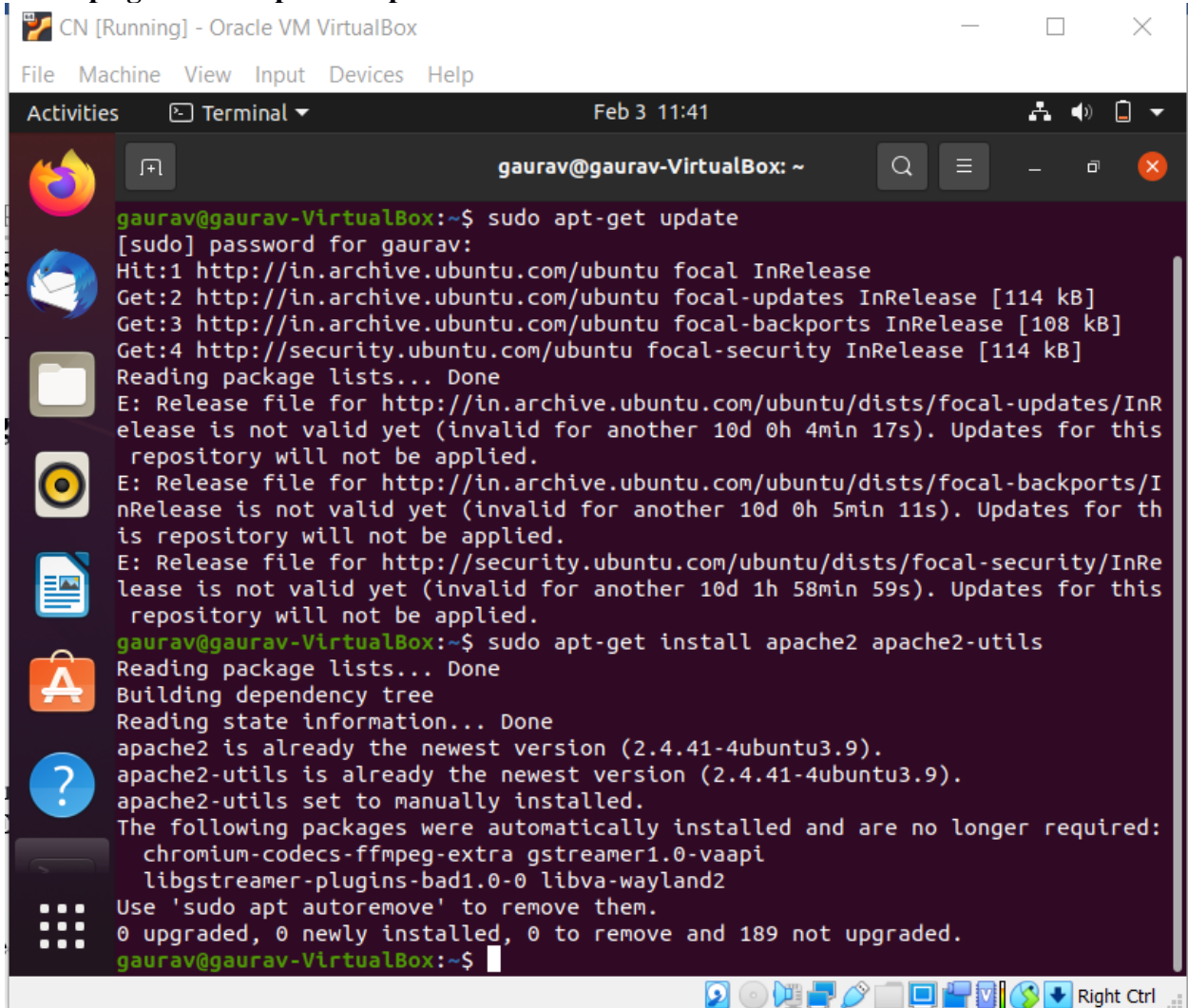
Executing the below commands on the terminal.

--> To update and integrate the existing softwares

sudo apt-get update

--> To install the apache utility

sudo apt-get install apache2 apache2-utils

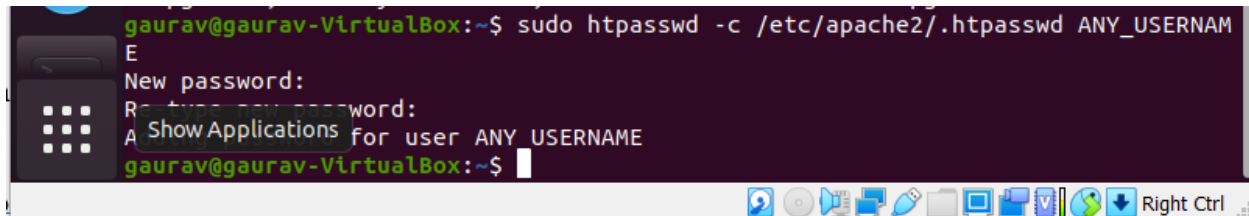


```

CN [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 3 11:41
gaurav@gaurav-VirtualBox: ~
gaurav@gaurav-VirtualBox:~$ sudo apt-get update
[sudo] password for gaurav:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Reading package lists... Done
E: Release file for http://in.archive.ubuntu.com/ubuntu/dists/focal-updates/InRelease is not valid yet (invalid for another 10d 0h 4min 17s). Updates for this repository will not be applied.
E: Release file for http://in.archive.ubuntu.com/ubuntu/dists/focal-backports/InRelease is not valid yet (invalid for another 10d 0h 5min 11s). Updates for this repository will not be applied.
E: Release file for http://security.ubuntu.com/ubuntu/dists/focal-security/InRelease is not valid yet (invalid for another 10d 1h 58min 59s). Updates for this repository will not be applied.
gaurav@gaurav-VirtualBox:~$ sudo apt-get install apache2 apache2-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.41-4ubuntu3.9).
apache2-utils is already the newest version (2.4.41-4ubuntu3.9).
apache2-utils set to manually installed.
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 189 not upgraded.
gaurav@gaurav-VirtualBox:~$
  
```

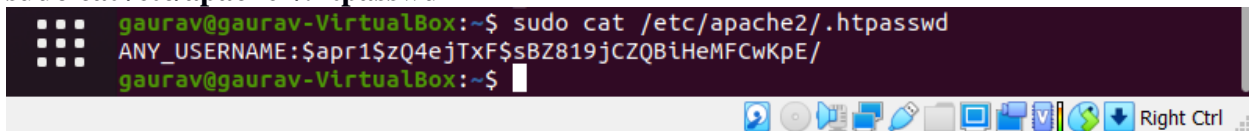
--> Provide username and password to set authentication

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



--> View the authentication

sudo cat /etc/apache2/.htpasswd



2. To setup the authentication phase, execute the following commands. Configuring Access control within the Virtual Host Definition.

--> Opening the file for setting authentication

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

```
<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>
Activities Terminal Feb 3 12:03
gaaurav@gaaurav-VirtualBox: ~
GNU nano 4.8 /etc/apache2/sites-available/000-default.conf

# Available loglevels: trace8, ..., trace1, debug, info, notice, v
# 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
<Directory "/var/www/html">
    AuthType Basic
    AuthName "RESTRICTED"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
# 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
# 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
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
[ Wrote 36 lines ]
Show Applications Write Out Where Is Cut Text Justify
Exit Read File Replace Paste Text To Spell

```

3. Password policy implementation is done by restarting the server as:

sudo service apache2 restart

```

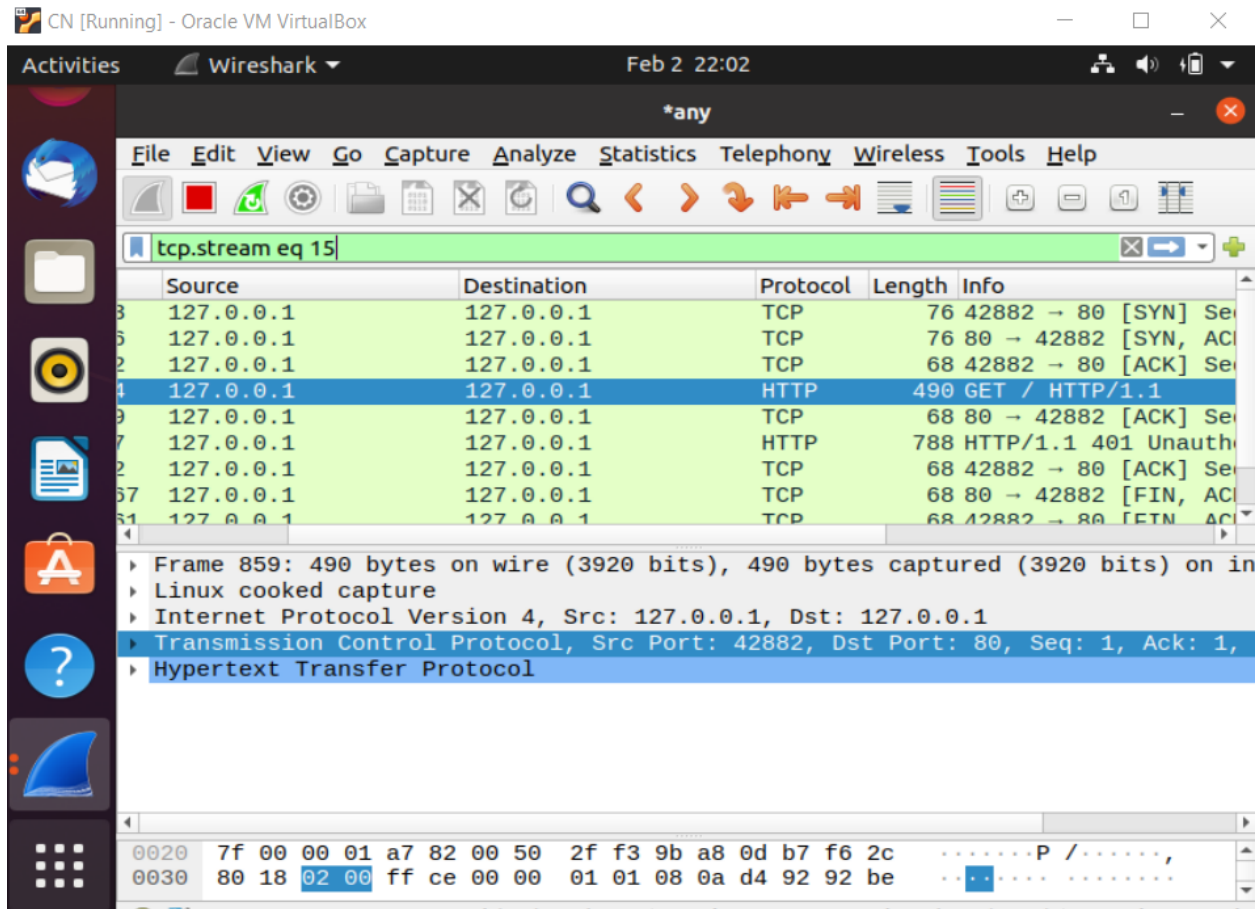
gaaurav@gaaurav-VirtualBox:~$ sudo service apache2 restart
gaaurav@gaaurav-VirtualBox:~$

```

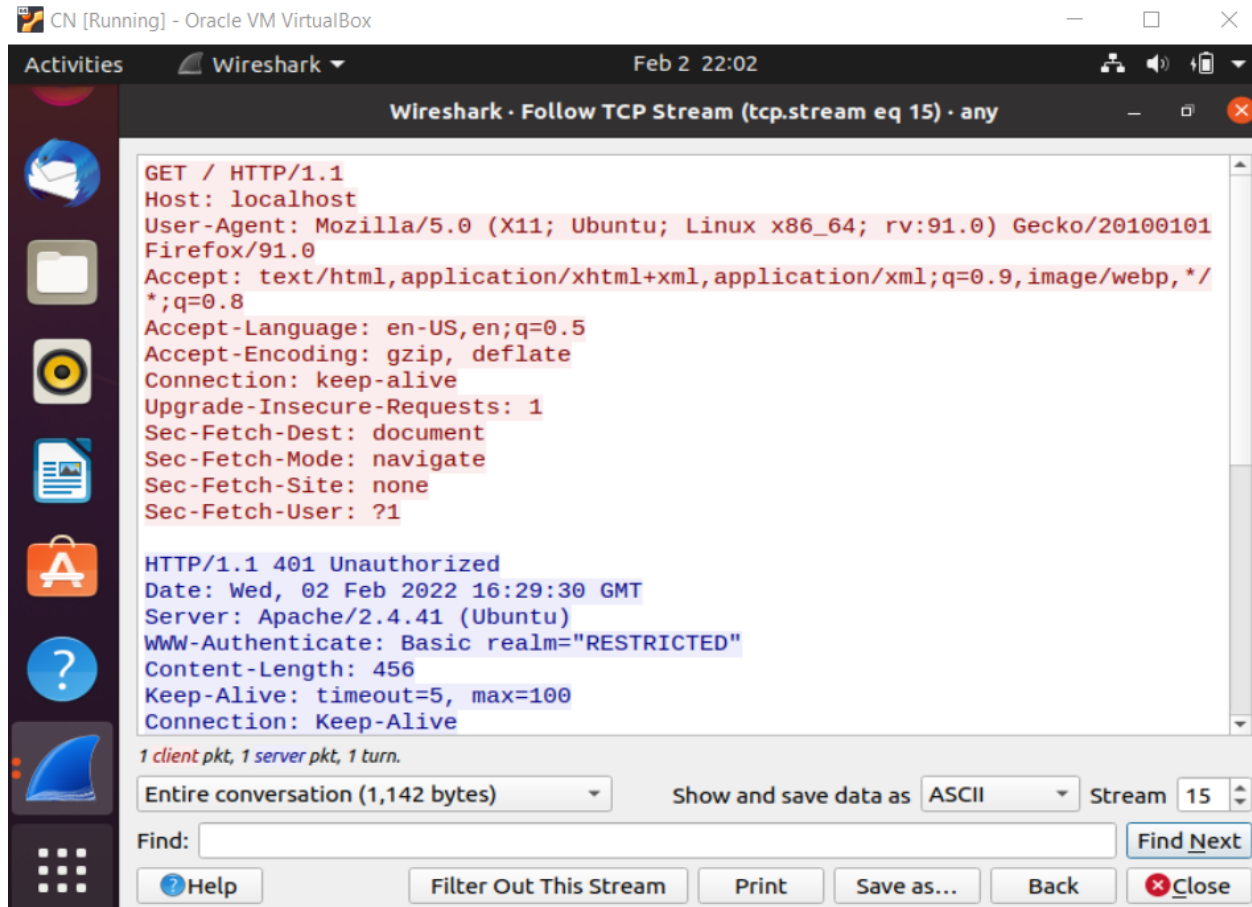
4. The localhost is then accessed using the Firefox browser requiring a username and a password set during the authentication phase.



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



6. Using the “follow TCP stream” on the HTTP message segment the password was retrieved which was encrypted by the base64 algorithm and decryption could be done with same algorithm.



Steps of Execution (Cookie Setting)

1. A PHP file to set the cookie is created which also contains an image in it (placed under the HTML directory) to be accessed once the cookie is set. The following code helped to set the cookie:

```
<html>
<?php
```

```
setcookie("namecookie","netqwerty",time()+123);
setcookie("nickname","work"); ?>
<img src= "highres.png" width= "300" height= "300" title= "password" />
</html>
```



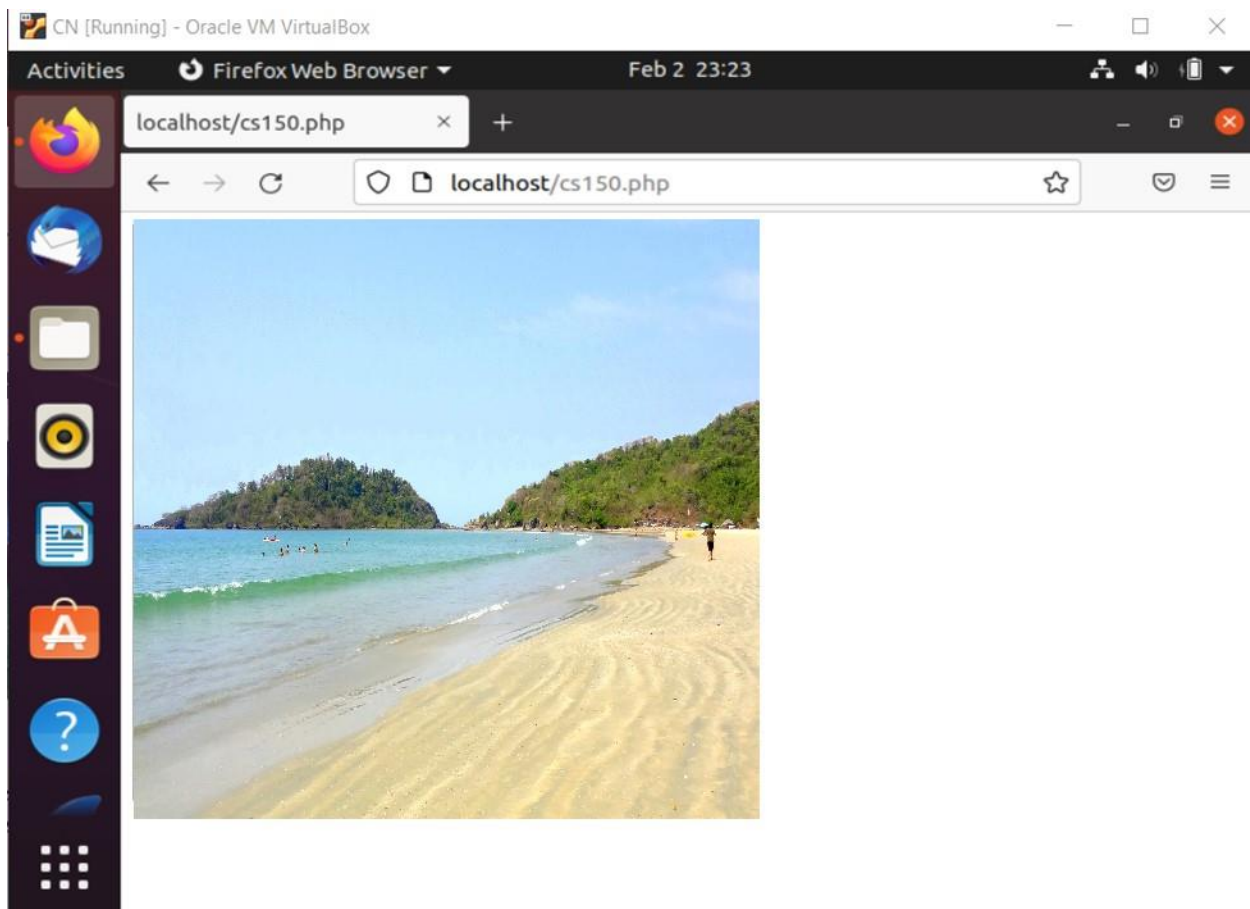

```
gaurav@gaurav-VirtualBox: /var/www/html
GNU nano 4.8 cs150.php
<html>
<?php
setcookie("namecookie","netcookie",time()+123);
setcookie("nickname","cn");
?>

</html>
```

Note: Here you can add any image if required

Note: You can capture Cookies mostly during the first time of web access. Hence keep wireshark capture ready before executing the task for the first time.

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



3. The packets are captured using Wireshark and using the “follow TCP stream” which checks for the set-cookie field whether the cookie is set or not set.

CN [Running] - Oracle VM VirtualBox

Feb 2 22:53

Activities Wireshark

*any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.1	TCP	76	42914 → 80 [SYN] Seq=0
127.0.0.1	127.0.0.1	TCP	76	80 → 42914 [SYN, ACK] S
127.0.0.1	127.0.0.1	TCP	68	42914 → 80 [ACK] Seq=1
127.0.0.1	127.0.0.1	HTTP	619	GET /cs150.php HTTP/1.1
127.0.0.1	127.0.0.1	TCP	68	80 → 42914 [ACK] Seq=1
127.0.0.1	127.0.0.1	HTTP	529	HTTP/1.1 200 OK (text/
127.0.0.1	127.0.0.1	TCP	68	42914 → 80 [ACK] Seq=55
127.0.0.1	127.0.0.1	HTTP	562	GET /home/gaurav/index.
127.0.0.1	127.0.0.1	TCP	68	80 → 42914 [ACK] Seq=46

Frame 23: 619 bytes on wire (4952 bits), 619 bytes captured (4952 bits) on int

Linux cooked capture

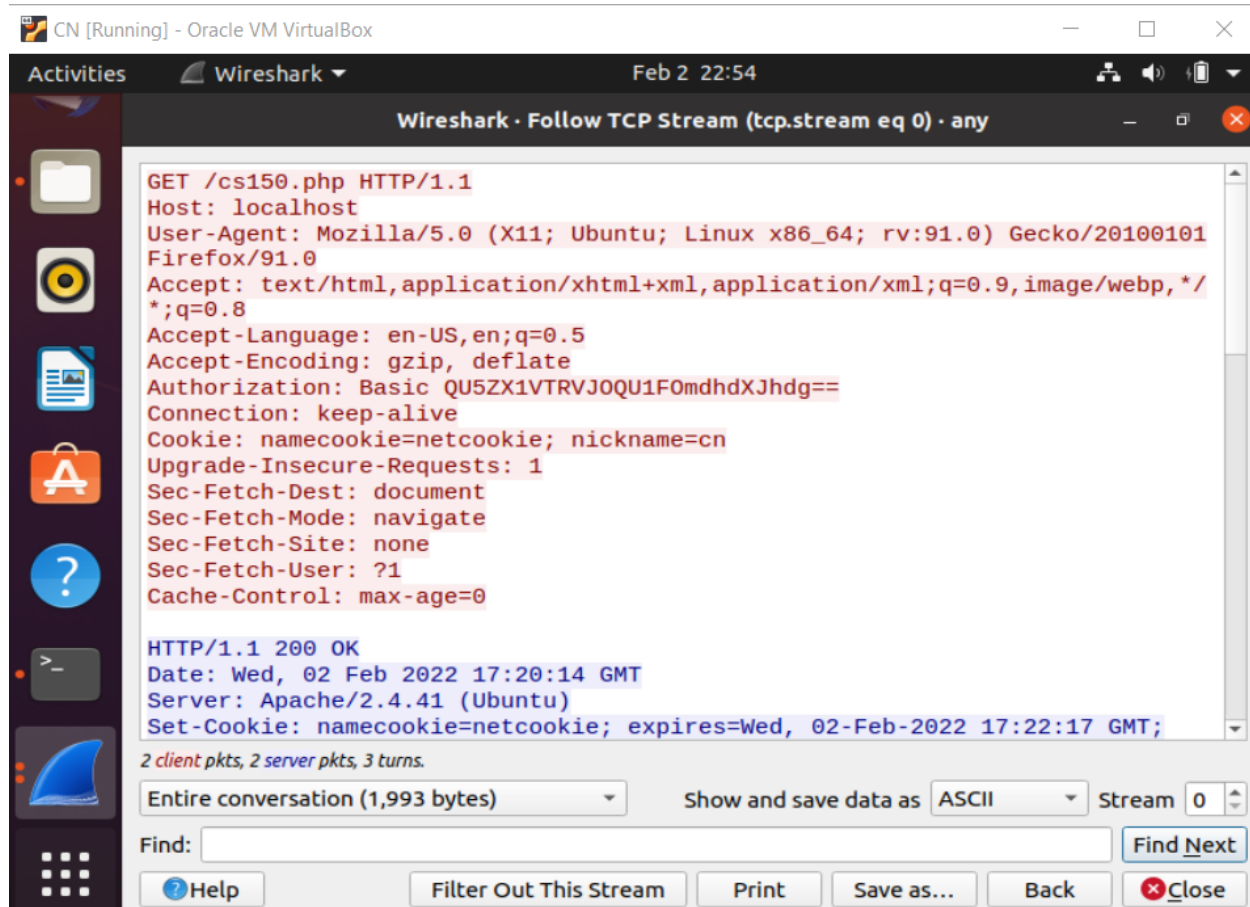
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 42914, Dst Port: 80, Seq: 1, Ack: 1,

Hypertext Transfer Protocol

0020 7f 00 00 01 a7 a2 00 50 3c da 6a 82 31 b3 57 ceP <·j·1·W·

0030 80 18 02 00 00 50 00 00 01 01 08 0a d4 c1 06 e0P·



The cookie is set as shown in the above screenshot.

Observation: Understand and work out base 64 algorithm and write in your observation. Observe various parameters associated with Cookie in the wireshark capture.

Conditional Get: If-Modified-Since

Before performing the steps below, make sure your browser's cache is empty. (To do this under Firefox, select Tools -> Clear Recent History and check the Cache box). Now do the following:

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

- Start up the Wireshark packet sniffer.
- Enter the following URL into your browser
<http://gaia.cs.umass.edu/wiresharklabs/HTTP-wireshark-file2.html>
- Your browser should display a very simple five-line HTML file.

- Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
- Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packetlisting window.

Wireshark · Follow TCP Stream (tcp.stream eq 2) · Wi-Fi

```
GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
Host: gaia.cs.umass.edu
Connection: keep-alive
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36
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
Accept-Encoding: gzip, deflate
Accept-Language: en-IN,en;q=0.9,fr-FR;q=0.8,fr;q=0.7,hi-IN;q=0.6,hi;q=0.5,en-GB;q=0.4,be-BY;q=0.3,be;q=0.2,en-US;q=0.1
If-None-Match: "173-5d7e0d5c000cb"
If-Modified-Since: Sun, 13 Feb 2022 06:59:02 GMT

HTTP/1.1 304 Not Modified
Date: Sun, 13 Feb 2022 11:34:08 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.27 mod_perl/2.0.11 Perl/v5.16.3
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
ETag: "173-5d7e0d5c000cb"
```

Observations:

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?

→ YES

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

→ YES BECAUSE WE CAN SEE THE CONTENTS IN THE LINE-BASED TEXT DATA FIELD

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. THE INFORMATION FOLLOWED IS: SUN, 13 FEB 2022 06:59:02 GMT\r\n WHICH IS THE DATE OF THE LAST MODIFICATION OF THE FILE FROM THE PREVIOUS GET REQUEST.

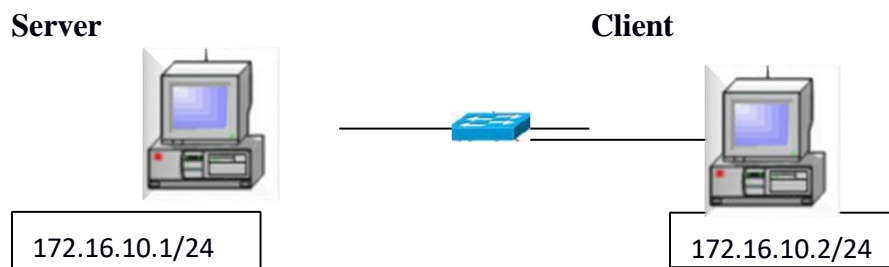
What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

→ THE STATUS CODE AND PHRASE RETURNED FROM THE SERVER IS HTTP/1.1 304 NOT MODIFIED. THE SERVER

DIDN'T RETURN THE CONTENTS OF THE FILE SINCE THE BROWSER LOADED IT FROM ITS CACHE.

3B) Understanding Persistent and Non-persistent HTTP Connections

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



Server Side:

Step 2: Check your Web Server

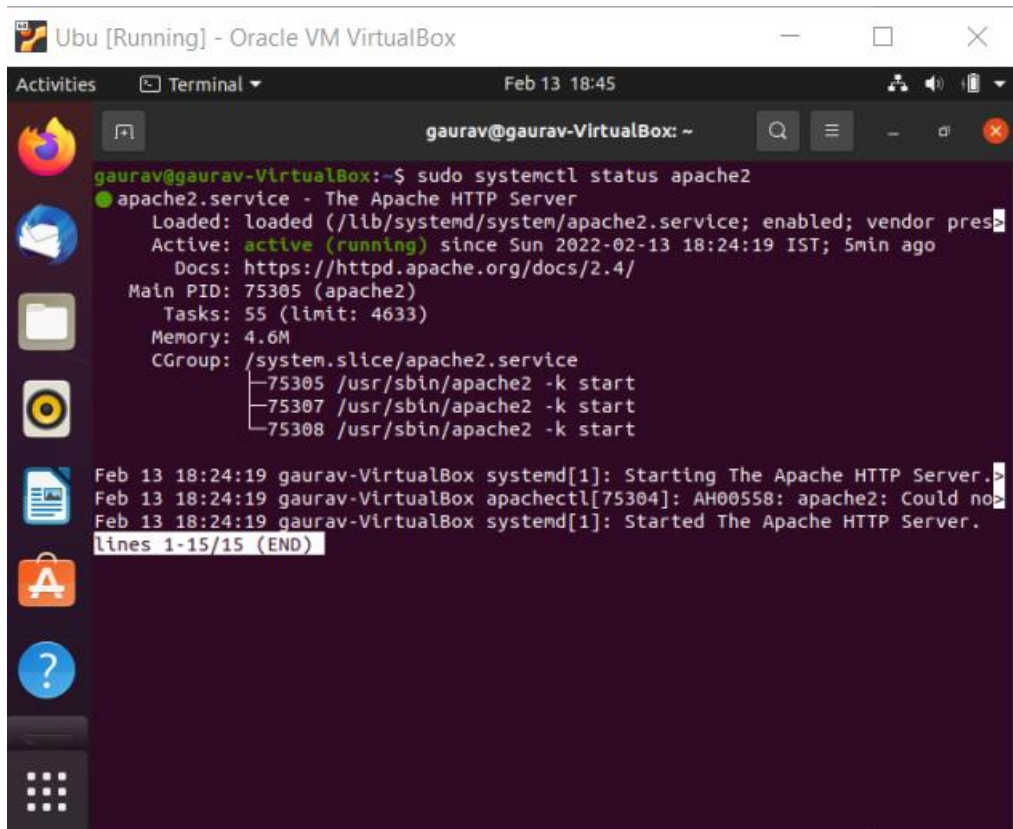
At the end of the installation process, Ubuntu 16.04 starts Apache. The web server should already be up and running. We can check with the `systemctl` command to make sure the service is running by typing:

`sudo systemctl status apache2`

or

`sudo service apache2 status`

As you can see above, the service appears to have started successfully. However, the best way to test this is to actually request a page from Apache. You can access the default Apache landing page to confirm that the software is running properly. You can access this through your server's domain name or IP address.



```

Ubu [Running] - Oracle VM VirtualBox
Feb 13 18:45
gaurav@gaurav-VirtualBox: ~
gaurav@gaurav-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor pres
   Active: active (running) since Sun 2022-02-13 18:24:19 IST; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 75305 (apache2)
      Tasks: 55 (limit: 4633)
     Memory: 4.6M
    CGroup: /system.slice/apache2.service
            └─75305 /usr/sbin/apache2 -k start
              └─75307 /usr/sbin/apache2 -k start
                └─75308 /usr/sbin/apache2 -k start

Feb 13 18:24:19 gaurav-VirtualBox systemd[1]: Starting The Apache HTTP Server.
Feb 13 18:24:19 gaurav-VirtualBox apachectl[75304]: AH00558: apache2: Could no
Feb 13 18:24:19 gaurav-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)

```

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
```

Note: If IP address fluctuates, kindly setup the IP address manually using 'Edit connections'.

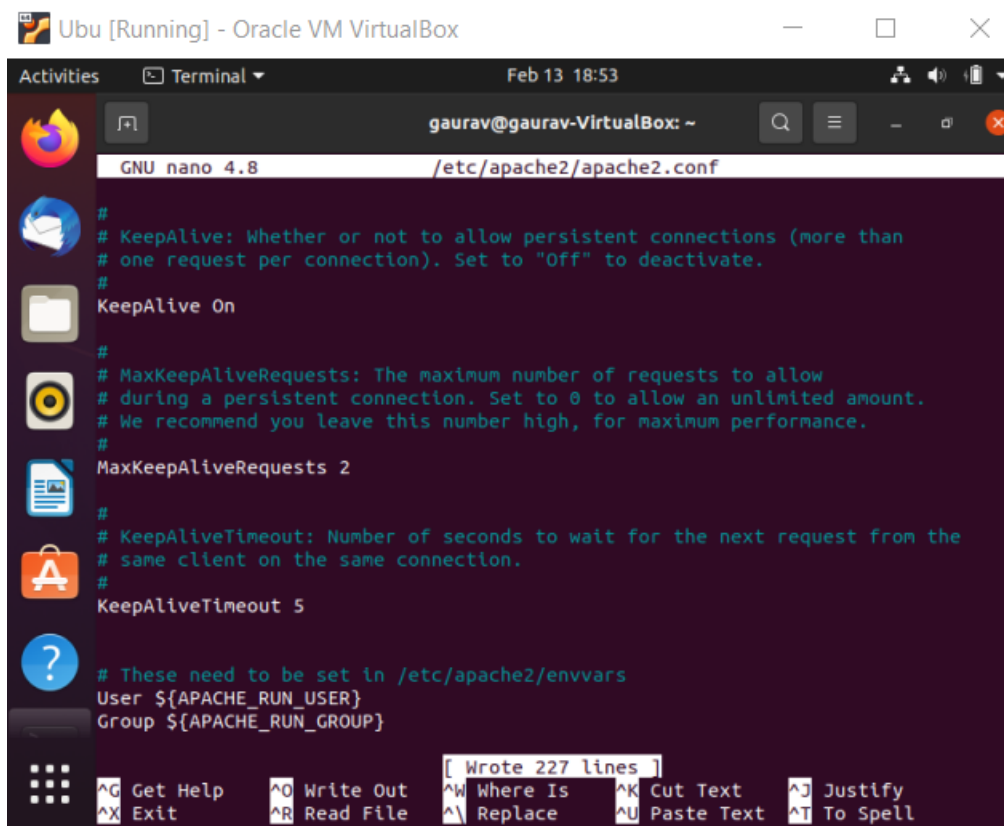
```

gaurav@gaurav-VirtualBox:/var/www/html$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enps3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e2:67:1a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic noprefixroute enps3
        valid_lft 501sec preferred_lft 501sec
    inet6 fe80::2488:7f22:7a0:78d3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
gaurav@gaurav-VirtualBox:/var/www/html$

```

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

- The **keep-alive** option was set (i.e. value was made **ON**)
- The **MaximumKeepAliveRequests** were set to **2**



```

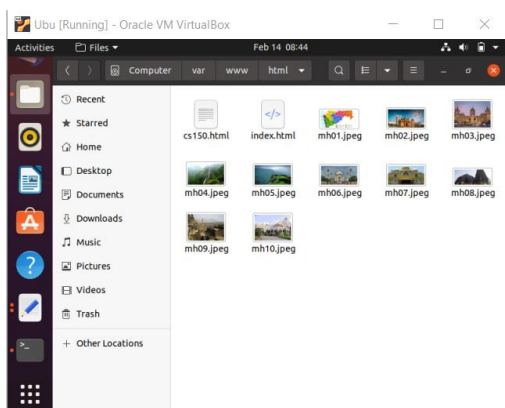
gaurav@gaurav-VirtualBox: ~
GNU nano 4.8 /etc/apache2/apache2.conf

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 2
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5

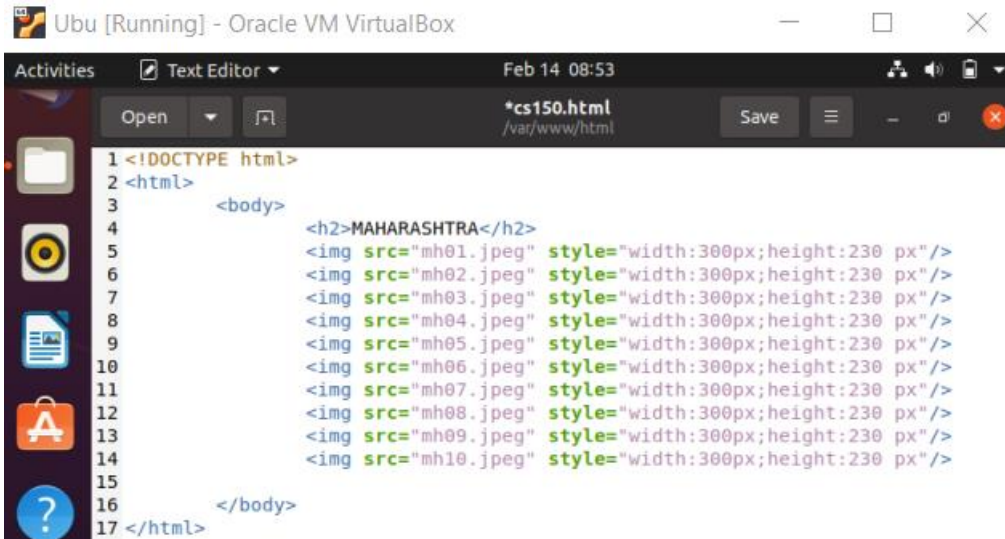
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

[ Wrote 227 lines ]
Get Help  Write Out  Where Is  Cut Text  Justify
Exit      Read File  Replace  Paste Text To Spell
  
```

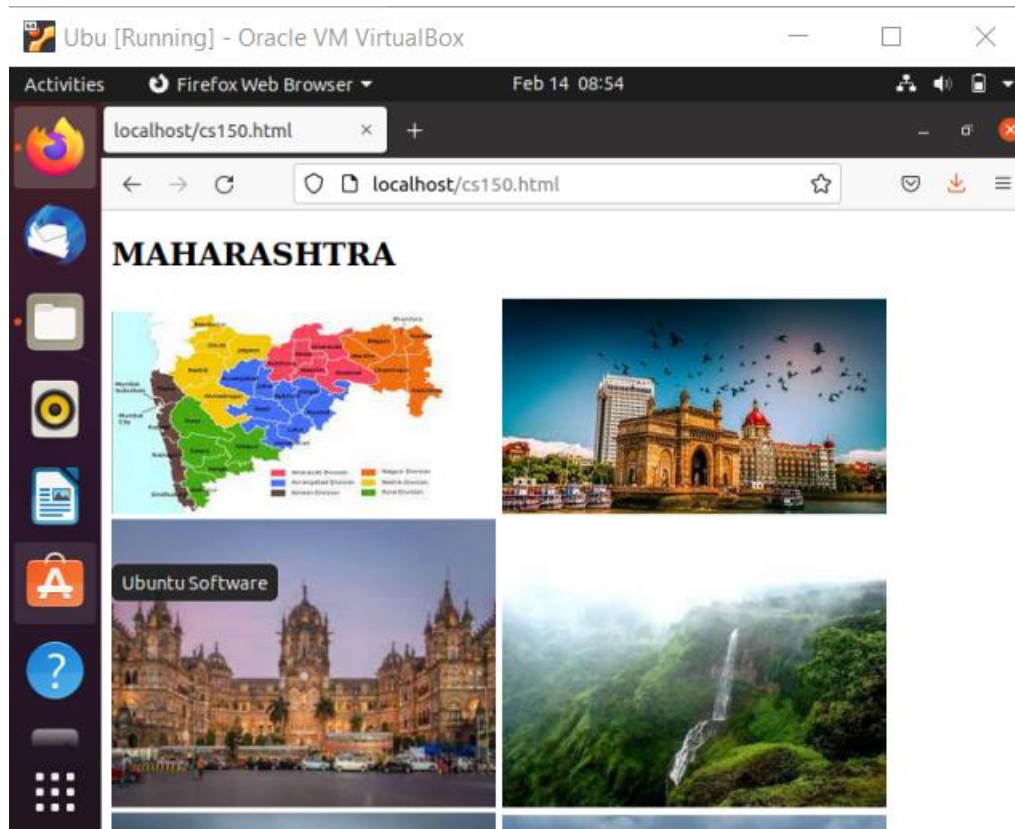
Step 5: Store images in the server path. A html page consisting of 10 images having size > 2MB were placed and accessed by the client. This html page is stored in the location - **/var/www/html/file_name.html**.



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



```
1 <!DOCTYPE html>
2 <html>
3     <body>
4         <h2>MAHARASHTRA</h2>
5         
6         
7         
8         
9         
10        
11        
12        
13        
14        
15
16    </body>
17 </html>
```

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
```

Note: If IP address fluctuates, kindly setup the IP address manually using 'Edit connections'.

```

gaurav@gaurav-VirtualBox: ~
gaurav@gaurav-VirtualBox: ~
gaurav@gaurav-VirtualBox:~$ ip addr
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gro
p default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
te UP group default qlen 1000
    link/ether 08:00:27:99:d1:77 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute
enp0s3
        valid_lft 461sec preferred_lft 461sec
    inet6 fe80::5caf:29c4:2149:8ba3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
gaurav@gaurav-VirtualBox:~$

```

There are broadly two parts of execution:

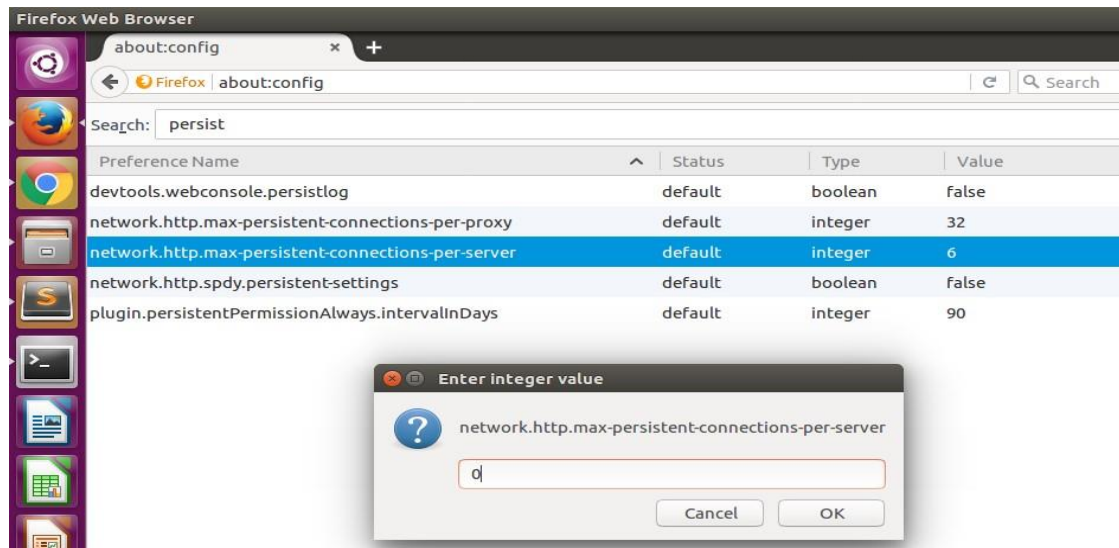
1. Dealing with non-persistent connections
2. Dealing with persistent connections

Open Firefox browser to configure for persistent option. Go to browser and type **about:config** and search for the term ‘**persistent**’

- While using non-persistent connection experiment, the **max-persistent-connectionsper-server** has the value set to **0** and **persistent-settings** value set to false.
- While using persistent connection experiment, the **max-persistent-connections-perserver** should have value greater than 0 (depending on the number of persistent connections needed) and **persistent-settings** value set to true.

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)

The client could access the file as:

172.16.10.1/file_name.html where--> **172.16.10.1** is Server's IP

Here the file name is **a.html** present in server. So, by typing **172.16.10.1/a.html** in client browser, we will be able to open the requested web page.

Note 1: The wireshark should capture the packets between the client and the server while the file is accessed.

Note 2: The images in the HTML page should have all the permissions specified through the server for the proper access.

Step 3: Use wireshark. Open wireshark in the server computer while client is trying to access the server's local host webpage. Apply 'http' filter and note the time to capture all the 10 images.

Wireshark Feb 25 17:11

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
19	9.367303710	10.0.2.4	10.0.2.5	HTTP	405	GET /cs150.html HTTP/1.1
21	9.367777268	10.0.2.5	10.0.2.4	HTTP	569	HTTP/1.1 200 OK (text/html)
23	9.396491115	10.0.2.4	10.0.2.5	HTTP	351	GET /mh01.jpeg HTTP/1.1
26	9.396877560	10.0.2.5	10.0.2.4	HTTP	3881	HTTP/1.1 200 OK (JPEG JFIF image)
29	9.397532804	10.0.2.4	10.0.2.5	HTTP	351	GET /mh02.jpeg HTTP/1.1
31	9.397790217	10.0.2.5	10.0.2.4	HTTP	11070	HTTP/1.1 200 OK (JPEG JFIF image)
39	9.399020744	10.0.2.4	10.0.2.5	HTTP	351	GET /mh03.jpeg HTTP/1.1
42	9.399410949	10.0.2.5	10.0.2.4	HTTP	2446	HTTP/1.1 200 OK (JPEG JFIF image)
45	9.399928867	10.0.2.4	10.0.2.5	HTTP	351	GET /mh04.jpeg HTTP/1.1
47	9.400143762	10.0.2.5	10.0.2.4	HTTP	8741	HTTP/1.1 200 OK (JPEG JFIF image)
49	9.400686674	10.0.2.4	10.0.2.5	HTTP	351	GET /mh05.jpeg HTTP/1.1
51	9.400944080	10.0.2.5	10.0.2.4	HTTP	7344	HTTP/1.1 200 OK (JPEG JFIF image)
59	9.404791353	10.0.2.4	10.0.2.5	HTTP	351	GET /mh06.jpeg HTTP/1.1
62	9.405107470	10.0.2.5	10.0.2.4	HTTP	3516	HTTP/1.1 200 OK (JPEG JFIF image)
65	9.405895328	10.0.2.4	10.0.2.5	HTTP	351	GET /mh07.jpeg HTTP/1.1
67	9.406074337	10.0.2.5	10.0.2.4	HTTP	9218	HTTP/1.1 200 OK (JPEG JFIF image)
69	9.407321819	10.0.2.4	10.0.2.5	HTTP	351	GET /mh08.jpeg HTTP/1.1
71	9.407676040	10.0.2.5	10.0.2.4	HTTP	8710	HTTP/1.1 200 OK (JPEG JFIF image)
79	9.409933147	10.0.2.4	10.0.2.5	HTTP	351	GET /mh09.jpeg HTTP/1.1
82	9.410315224	10.0.2.5	10.0.2.4	HTTP	3222	HTTP/1.1 200 OK (JPEG JFIF image)
85	9.411323584	10.0.2.4	10.0.2.5	HTTP	351	GET /mh10.jpeg HTTP/1.1
87	9.411619440	10.0.2.5	10.0.2.4	HTTP	8983	HTTP/1.1 200 OK (JPEG JFIF image)
90	9.433471186	10.0.2.4	10.0.2.5	HTTP	353	GET /favicon.ico HTTP/1.1

Here it is $9.411619440 - 9.367303710 = 0.04431573$

PART 2: PERSISTENT CONNECTIONS

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

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

Running] - Oracle VM VirtualBox

File View Input Devices Help

Wireshark Feb 25 16:57

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

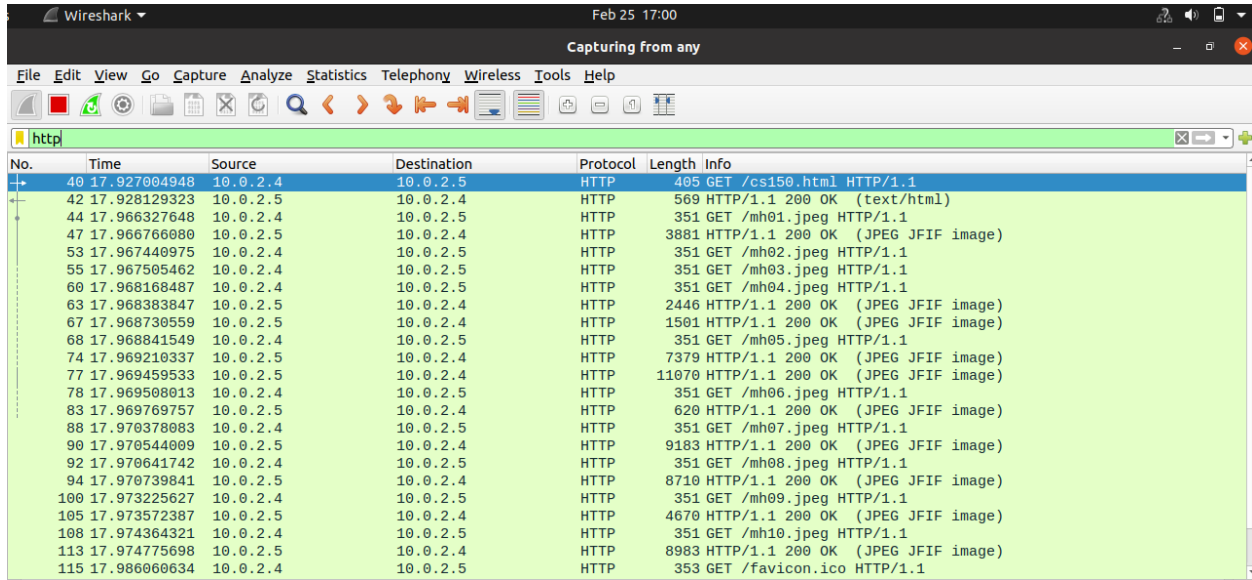
http

No.	Time	Source	Destination	Protocol	Length	Info
33	15.617435494	10.0.2.4	10.0.2.5	HTTP	405	GET /cs150.html HTTP/1.1
35	15.618318492	10.0.2.5	10.0.2.4	HTTP	569	HTTP/1.1 200 OK (text/html)
37	15.650720302	10.0.2.4	10.0.2.5	HTTP	351	GET /mh01.jpeg HTTP/1.1
40	15.651249449	10.0.2.5	10.0.2.4	HTTP	3881	HTTP/1.1 200 OK (JPEG JFIF image)
43	15.652896550	10.0.2.4	10.0.2.5	HTTP	351	GET /mh02.jpeg HTTP/1.1
45	15.653236662	10.0.2.5	10.0.2.4	HTTP	11070	HTTP/1.1 200 OK (JPEG JFIF image)
53	15.654601628	10.0.2.4	10.0.2.5	HTTP	351	GET /mh03.jpeg HTTP/1.1
56	15.654846846	10.0.2.5	10.0.2.4	HTTP	2446	HTTP/1.1 200 OK (JPEG JFIF image)
62	15.655429876	10.0.2.4	10.0.2.5	HTTP	351	GET /mh04.jpeg HTTP/1.1
64	15.655597225	10.0.2.5	10.0.2.4	HTTP	8741	HTTP/1.1 200 OK (JPEG JFIF image)
66	15.658358398	10.0.2.4	10.0.2.5	HTTP	351	GET /mh05.jpeg HTTP/1.1
68	15.658728170	10.0.2.5	10.0.2.4	HTTP	7344	HTTP/1.1 200 OK (JPEG JFIF image)
69	15.658803058	10.0.2.4	10.0.2.5	HTTP	351	GET /mh06.jpeg HTTP/1.1
77	15.659320087	10.0.2.5	10.0.2.4	HTTP	3516	HTTP/1.1 200 OK (JPEG JFIF image)
82	15.659677552	10.0.2.4	10.0.2.5	HTTP	351	GET /mh07.jpeg HTTP/1.1
85	15.659893167	10.0.2.5	10.0.2.4	HTTP	9218	HTTP/1.1 200 OK (JPEG JFIF image)
86	15.659976357	10.0.2.4	10.0.2.5	HTTP	351	GET /mh08.jpeg HTTP/1.1
90	15.660307745	10.0.2.5	10.0.2.4	HTTP	1505	HTTP/1.1 200 OK (JPEG JFIF image)
91	15.660415732	10.0.2.4	10.0.2.5	HTTP	351	GET /mh09.jpeg HTTP/1.1
95	15.660705145	10.0.2.5	10.0.2.4	HTTP	10427	HTTP/1.1 200 OK (JPEG JFIF image)
96	15.660797266	10.0.2.4	10.0.2.5	HTTP	351	GET /mh10.jpeg HTTP/1.1
100	15.661123386	10.0.2.5	10.0.2.4	HTTP	8983	HTTP/1.1 200 OK (JPEG JFIF image)
105	15.681955779	10.0.2.4	10.0.2.5	HTTP	353	GET /favicon.ico HTTP/1.1

Here it is $15.661123386 - 15.617435494 = 0.043687892$

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

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



No.	Time	Source	Destination	Protocol	Length	Info
40	17.927004948	10.0.2.4	10.0.2.5	HTTP	405	GET /cs150.html HTTP/1.1
42	17.928129323	10.0.2.5	10.0.2.4	HTTP	569	HTTP/1.1 200 OK (text/html)
44	17.966327648	10.0.2.4	10.0.2.5	HTTP	351	GET /mh01.jpeg HTTP/1.1
47	17.966766080	10.0.2.5	10.0.2.4	HTTP	3881	HTTP/1.1 200 OK (JPEG JFIF image)
53	17.967440975	10.0.2.4	10.0.2.5	HTTP	351	GET /mh02.jpeg HTTP/1.1
55	17.967505462	10.0.2.4	10.0.2.5	HTTP	351	GET /mh03.jpeg HTTP/1.1
60	17.968168487	10.0.2.4	10.0.2.5	HTTP	351	GET /mh04.jpeg HTTP/1.1
63	17.968383847	10.0.2.5	10.0.2.4	HTTP	2446	HTTP/1.1 200 OK (JPEG JFIF image)
67	17.968730559	10.0.2.5	10.0.2.4	HTTP	1501	HTTP/1.1 200 OK (JPEG JFIF image)
68	17.968841549	10.0.2.4	10.0.2.5	HTTP	351	GET /mh05.jpeg HTTP/1.1
74	17.969210337	10.0.2.5	10.0.2.4	HTTP	7379	HTTP/1.1 200 OK (JPEG JFIF image)
77	17.969459533	10.0.2.5	10.0.2.4	HTTP	11070	HTTP/1.1 200 OK (JPEG JFIF image)
78	17.969508013	10.0.2.4	10.0.2.5	HTTP	351	GET /mh06.jpeg HTTP/1.1
83	17.969769757	10.0.2.5	10.0.2.4	HTTP	620	HTTP/1.1 200 OK (JPEG JFIF image)
88	17.970378083	10.0.2.4	10.0.2.5	HTTP	351	GET /mh07.jpeg HTTP/1.1
90	17.970544009	10.0.2.5	10.0.2.4	HTTP	9183	HTTP/1.1 200 OK (JPEG JFIF image)
92	17.970641742	10.0.2.4	10.0.2.5	HTTP	351	GET /mh08.jpeg HTTP/1.1
94	17.970739841	10.0.2.5	10.0.2.4	HTTP	8710	HTTP/1.1 200 OK (JPEG JFIF image)
100	17.973225627	10.0.2.4	10.0.2.5	HTTP	351	GET /mh09.jpeg HTTP/1.1
105	17.973572387	10.0.2.5	10.0.2.4	HTTP	4670	HTTP/1.1 200 OK (JPEG JFIF image)
108	17.974364321	10.0.2.4	10.0.2.5	HTTP	351	GET /mh10.jpeg HTTP/1.1
113	17.974775698	10.0.2.5	10.0.2.4	HTTP	8983	HTTP/1.1 200 OK (JPEG JFIF image)
115	17.986060634	10.0.2.4	10.0.2.5	HTTP	353	GET /favicon.ico HTTP/1.1

Here is it $17.974775698 - 17.927004948 = 0.04777075$

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

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

Running] - Oracle VM VirtualBox

hine View Input Devices Help

Wireshark Feb 25 17:04

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
10	2.910497892	10.0.2.4	10.0.2.5	HTTP	405	GET /cs150.html HTTP/1.1
12	2.911567117	10.0.2.5	10.0.2.4	HTTP	569	HTTP/1.1 200 OK (text/html)
14	2.951888821	10.0.2.4	10.0.2.5	HTTP	351	GET /mh01.jpeg HTTP/1.1
17	2.952326709	10.0.2.5	10.0.2.4	HTTP	3881	HTTP/1.1 200 OK (JPEG JFIF image)
22	2.952689268	10.0.2.4	10.0.2.5	HTTP	351	GET /mh02.jpeg HTTP/1.1
25	2.952966444	10.0.2.4	10.0.2.5	HTTP	351	GET /mh03.jpeg HTTP/1.1
27	2.953035375	10.0.2.5	10.0.2.4	HTTP	11070	HTTP/1.1 200 OK (JPEG JFIF image)
31	2.953302960	10.0.2.5	10.0.2.4	HTTP	2446	HTTP/1.1 200 OK (JPEG JFIF image)
36	2.953827230	10.0.2.4	10.0.2.5	HTTP	351	GET /mh04.jpeg HTTP/1.1
40	2.954129199	10.0.2.5	10.0.2.4	HTTP	8741	HTTP/1.1 200 OK (JPEG JFIF image)
42	2.954430300	10.0.2.4	10.0.2.5	HTTP	351	GET /mh05.jpeg HTTP/1.1
46	2.954584952	10.0.2.5	10.0.2.4	HTTP	139	HTTP/1.1 200 OK (JPEG JFIF image)
47	2.954885625	10.0.2.4	10.0.2.5	HTTP	351	GET /mh06.jpeg HTTP/1.1
52	2.955086698	10.0.2.5	10.0.2.4	HTTP	585	HTTP/1.1 200 OK (JPEG JFIF image)
54	2.955238438	10.0.2.4	10.0.2.5	HTTP	351	GET /mh07.jpeg HTTP/1.1
59	2.955874113	10.0.2.5	10.0.2.4	HTTP	9218	HTTP/1.1 200 OK (JPEG JFIF image)
65	2.956340365	10.0.2.4	10.0.2.5	HTTP	351	GET /mh08.jpeg HTTP/1.1
68	2.956402347	10.0.2.4	10.0.2.5	HTTP	351	GET /mh09.jpeg HTTP/1.1
70	2.956571978	10.0.2.5	10.0.2.4	HTTP	8710	HTTP/1.1 200 OK (JPEG JFIF image)
77	2.956978753	10.0.2.5	10.0.2.4	HTTP	3222	HTTP/1.1 200 OK (JPEG JFIF image)
80	2.957179390	10.0.2.4	10.0.2.5	HTTP	351	GET /mh10.jpeg HTTP/1.1
86	2.957388938	10.0.2.5	10.0.2.4	HTTP	1743	HTTP/1.1 200 OK (JPEG JFIF image)
89	2.970288018	10.0.2.4	10.0.2.5	HTTP	353	GET /favicon.ico HTTP/1.1

Here it is $2.957388938 - 2.910497892 = 0.046891046$

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

Step 8: Repeat the **steps 1-3** in the previous section.

Wireshark Feb 25 17:06

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
23	6.089407174	10.0.2.4	10.0.2.5	HTTP	405	GET /cs150.html HTTP/1.1
25	6.090246456	10.0.2.5	10.0.2.4	HTTP	569	HTTP/1.1 200 OK (text/html)
27	6.135078598	10.0.2.4	10.0.2.5	HTTP	351	GET /mh01.jpeg HTTP/1.1
30	6.135527510	10.0.2.5	10.0.2.4	HTTP	3881	HTTP/1.1 200 OK (JPEG JFIF image)
38	6.136139447	10.0.2.4	10.0.2.5	HTTP	351	GET /mh02.jpeg HTTP/1.1
42	6.136256528	10.0.2.4	10.0.2.5	HTTP	351	GET /mh03.jpeg HTTP/1.1
44	6.136391829	10.0.2.4	10.0.2.5	HTTP	351	GET /mh04.jpeg HTTP/1.1
46	6.136603004	10.0.2.5	10.0.2.4	HTTP	11070	HTTP/1.1 200 OK (JPEG JFIF image)
50	6.137143608	10.0.2.5	10.0.2.4	HTTP	2446	HTTP/1.1 200 OK (JPEG JFIF image)
56	6.137441575	10.0.2.5	10.0.2.4	HTTP	1501	HTTP/1.1 200 OK (JPEG JFIF image)
62	6.137962131	10.0.2.4	10.0.2.5	HTTP	351	GET /mh05.jpeg HTTP/1.1
64	6.138201669	10.0.2.5	10.0.2.4	HTTP	7379	HTTP/1.1 200 OK (JPEG JFIF image)
65	6.138273384	10.0.2.4	10.0.2.5	HTTP	351	GET /mh06.jpeg HTTP/1.1
68	6.138375120	10.0.2.4	10.0.2.5	HTTP	351	GET /mh07.jpeg HTTP/1.1
74	6.138776956	10.0.2.5	10.0.2.4	HTTP	620	HTTP/1.1 200 OK (JPEG JFIF image)
78	6.139081366	10.0.2.5	10.0.2.4	HTTP	1978	HTTP/1.1 200 OK (JPEG JFIF image)
79	6.139177654	10.0.2.4	10.0.2.5	HTTP	351	GET /mh08.jpeg HTTP/1.1
83	6.139393818	10.0.2.5	10.0.2.4	HTTP	8710	HTTP/1.1 200 OK (JPEG JFIF image)
88	6.141089388	10.0.2.4	10.0.2.5	HTTP	351	GET /mh09.jpeg HTTP/1.1
90	6.141369669	10.0.2.5	10.0.2.4	HTTP	10427	HTTP/1.1 200 OK (JPEG JFIF image)
95	6.142289778	10.0.2.4	10.0.2.5	HTTP	351	GET /mh10.jpeg HTTP/1.1
97	6.142637920	10.0.2.5	10.0.2.4	HTTP	8983	HTTP/1.1 200 OK (JPEG JFIF image)
99	6.152228124	10.0.2.4	10.0.2.5	HTTP	353	GET /favicon.ico HTTP/1.1

Here it is $6.142637920 - 6.089407174 = 0.053230746$



WEEK 3:COMPUTER NETWORKS LAB

JAN-
MAY
2022