Week: #4 Understand working of HTTP Headers

	Understand working of HTTP headers: Conditional Get: If-Modified-Since HTTP Cookies: Cookie and Set-Cookie Authentication: Auth-Basic Design a web page that has one embedded page (e.g. image) and sets a cookie and enables authentication. You are required to configure the web server (e.g. apache)with authentication mechanism. Show the behavior of conditional get when embedded objects is modified and when it is not (you can just change the create date of the embedded object). Decode the Basic-Auth header using Base64 mechanism as per the password setup. **Observation**: Show the behavior of browser when is cookie is set and when cookie is removed.

NAME :NAGAVENI L G

SEC :4F

SRN:PES2UG21CS315

Week: 3, b

Understanding Working of HTTP Headers

Question: Understand working of HTTP headers

Conditional Get: If-Modified-Since

HTTP Cookies: Cookie and Set-Cookie

Authentication: Auth-Basic

Design a web page that has one embedded page (e.g. image) and sets a cookie and enables authentication. You are required to configure the web server (e.g. apache) with authentication mechanism. Show the behavior of conditional get when embedded objects are modified and when it is not (you can just change the create date of the embedded object). Decode the Basic-Auth header using Base64 mechanism as per the password setup.

Observation: Show the behavior of browser when is cookie is set and when cookie is removed.

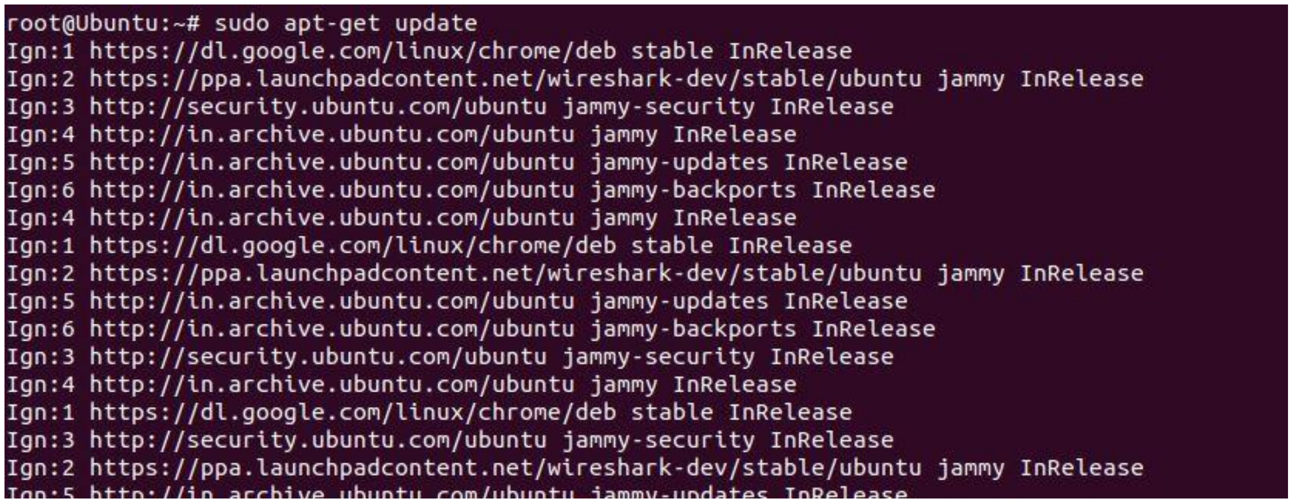
Solution: Analyzing Basic Authentication and Cookies

The three parts of experiment are:

1. Password Authentication
2. Cookie Setting
3. Conditional get

Steps of Execution (for Password Authentication)
1. Executing the below commands on the terminal.

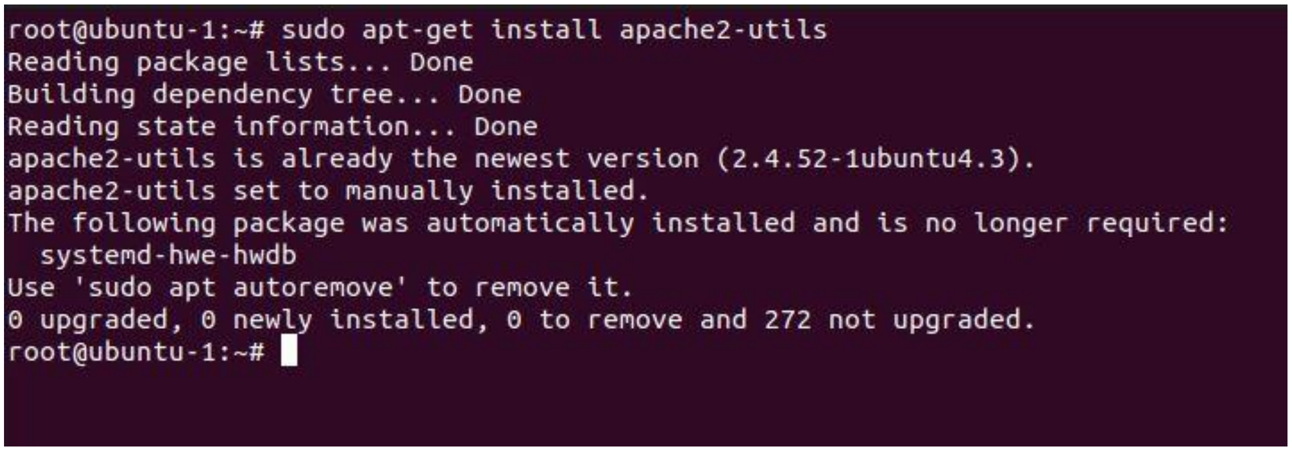
--> To update and integrate the existing softwares
sudo apt-get update



```
root@Ubuntu:~# sudo apt-get update
Ign:1 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:2 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy InRelease
Ign:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Ign:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Ign:1 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:2 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy InRelease
Ign:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Ign:1 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:2 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy InRelease
Ign:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
```

--> To install the apache utility

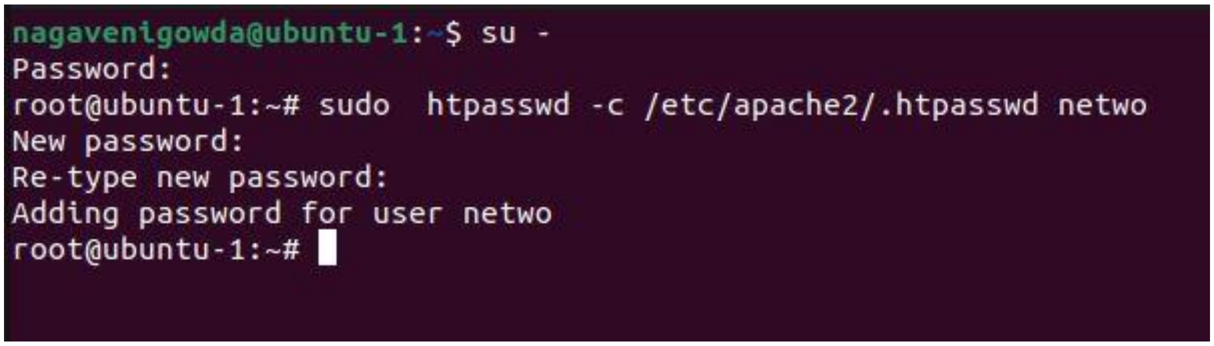
sudo apt-get install apache2 apache2-utils



```
root@ubuntu-1:~# sudo apt-get install apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2-utils is already the newest version (2.4.52-1ubuntu4.3).
apache2-utils set to manually installed.
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 272 not upgraded.
root@ubuntu-1:~#
```

--> Provide username and password to set authentication

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

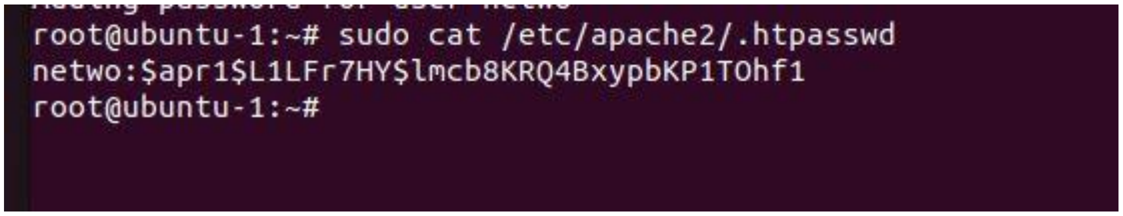


```
nagavenigowda@ubuntu-1:~$ su -
Password:
root@ubuntu-1:~# sudo  htpasswd -c /etc/apache2/.htpasswd netwo
New password:
Re-type new password:
Adding password for user netwo
root@ubuntu-1:~#
```

Here “netwo” is the username. Also, password is entered twice.

--> View the authentication

sudo cat /etc/apache2/.htpasswd



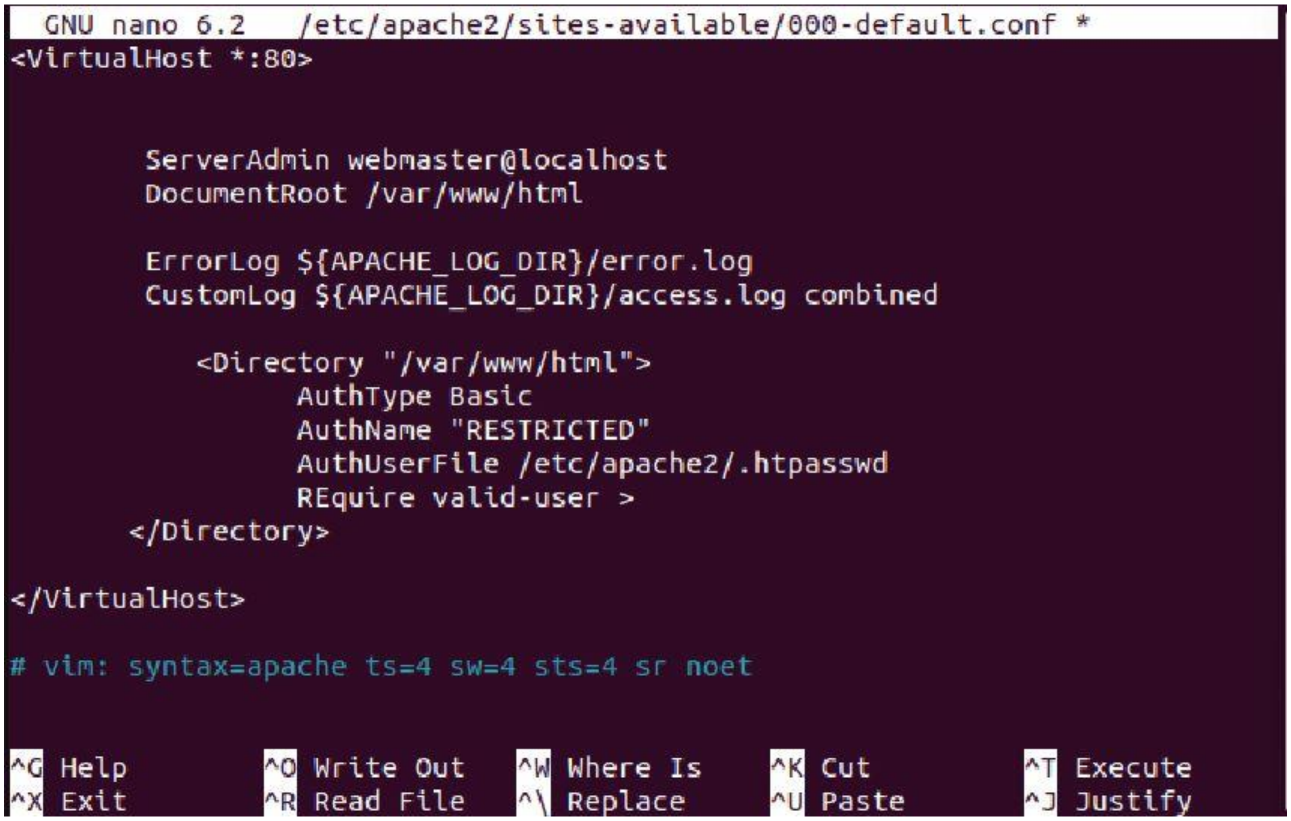
```
root@ubuntu-1:~# sudo cat /etc/apache2/.htpasswd
netwo:$apr1$L1LFr7HY$lmcb8KRQ4BxypbKP1TOhf1
root@ubuntu-1:~#
```

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

--> Opening the file for setting authentication

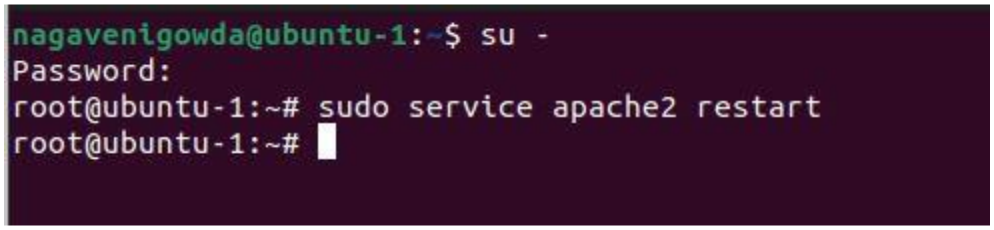
sudo nano /etc/apache2/sites-available/000-default.conf

```
<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>
```



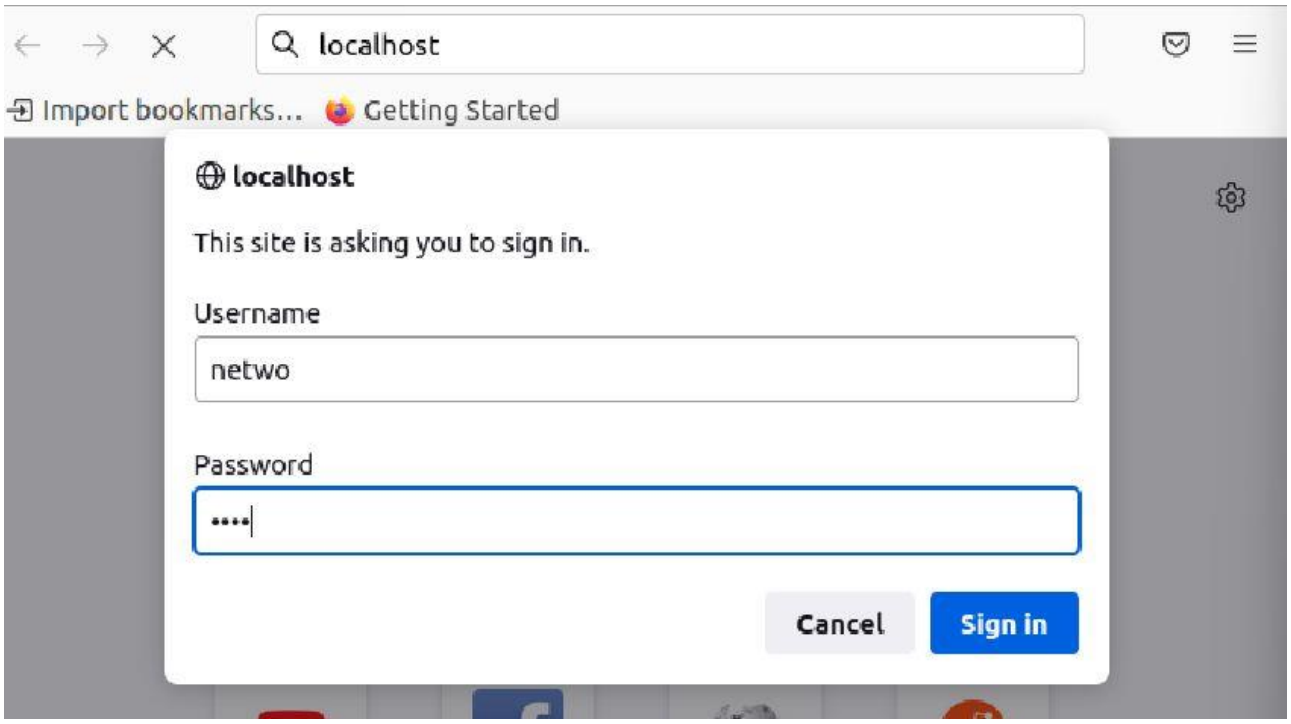
```
  GNU nano 6.2    /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:80>


        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>

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


^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify
```

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



```
nagavenigowda@ubuntu-1:~$ su -
Password:
root@ubuntu-1:~# sudo service apache2 restart
root@ubuntu-1:~#
```

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



localhost

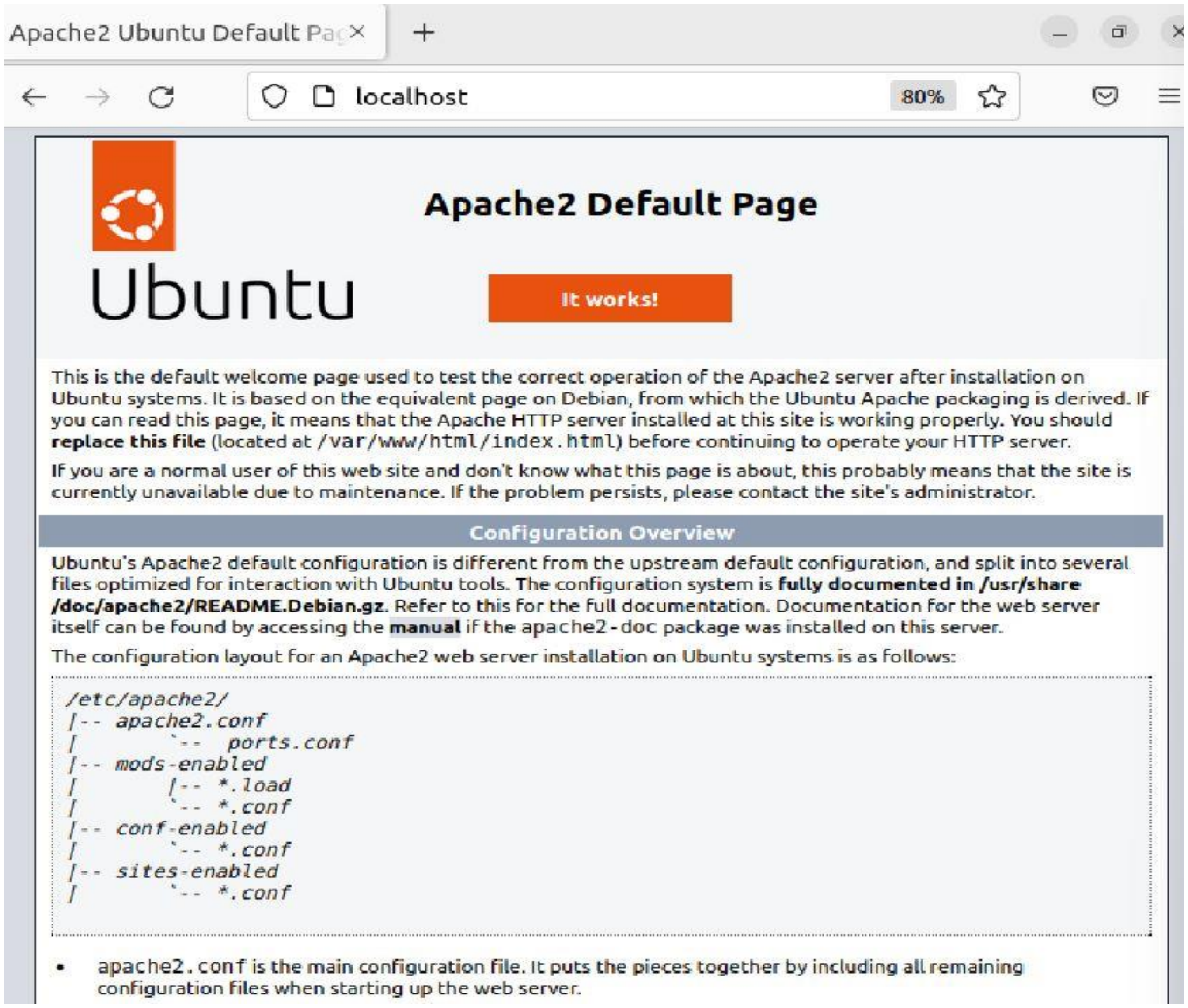
This site is asking you to sign in.

Username

netwo

Password

••••


Apache2 Ubuntu Default Pa

localhost

80%

Ubuntu

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

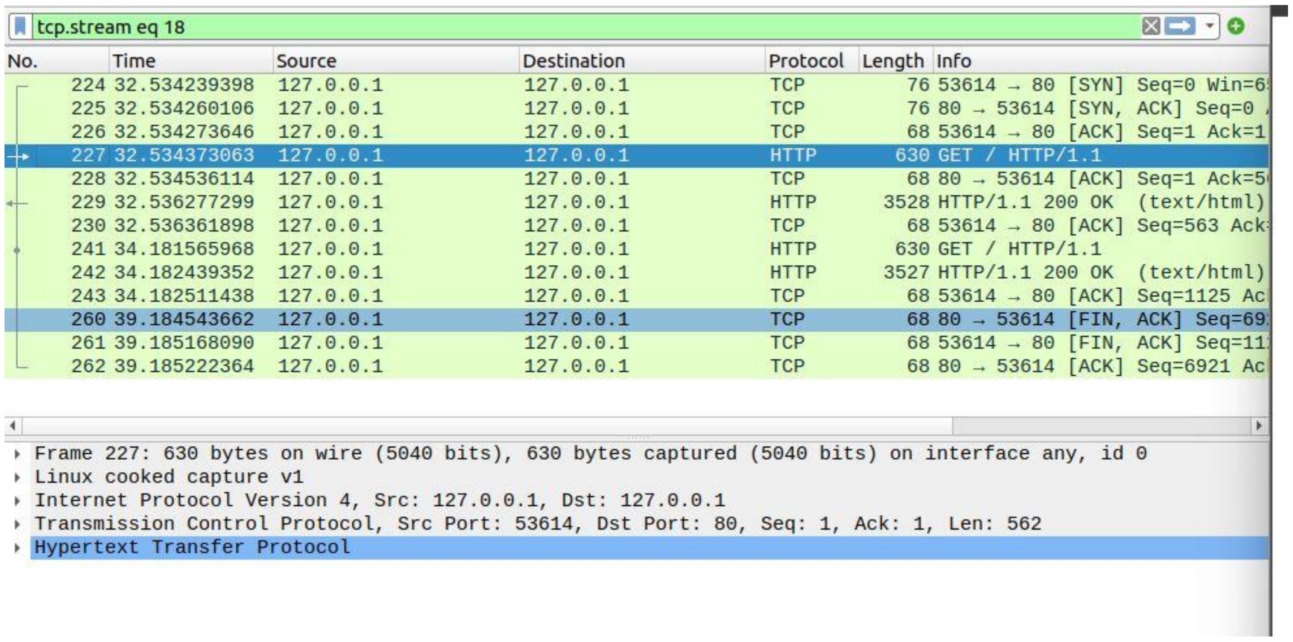
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share /doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

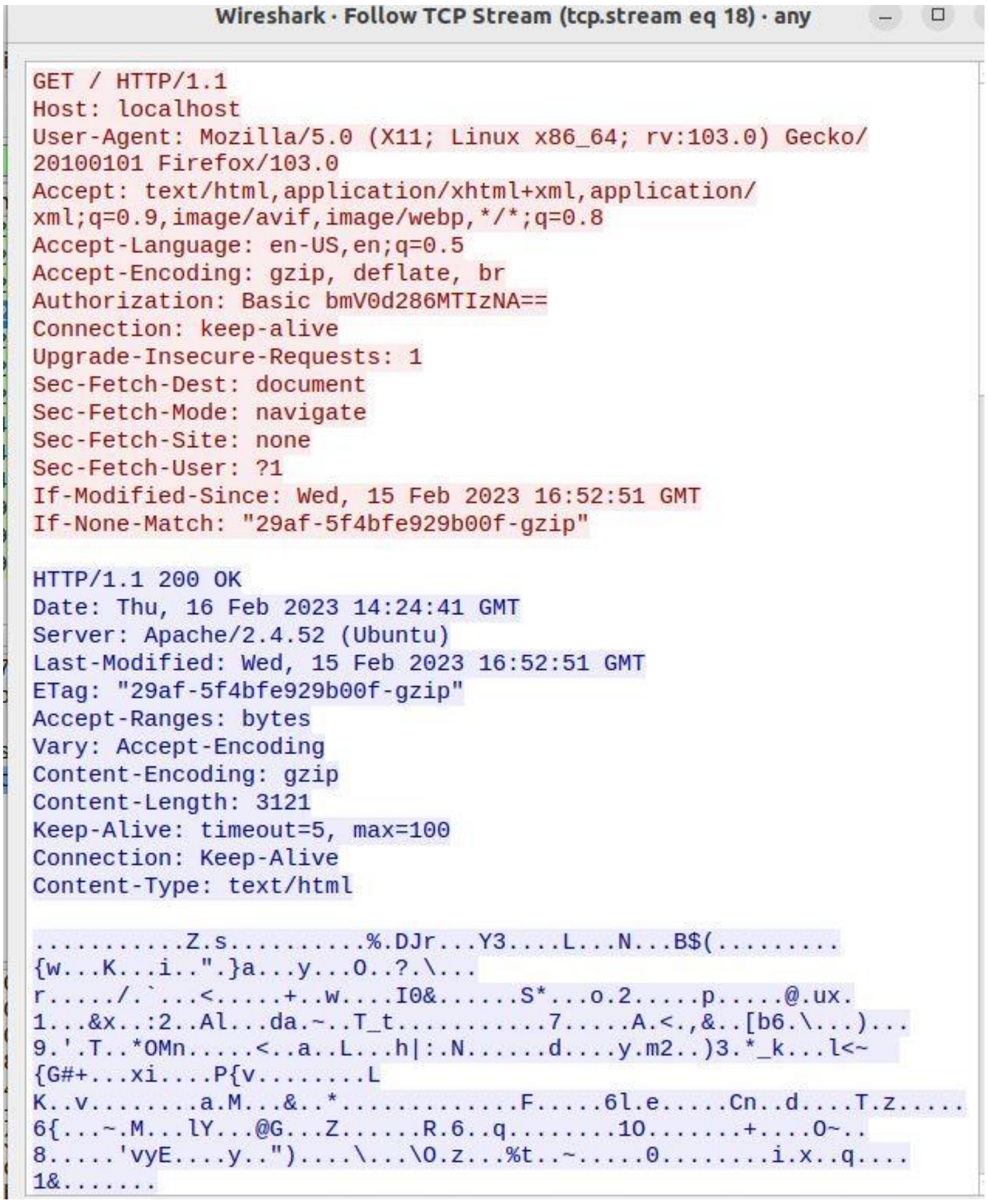
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

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



```
tcp.stream eq 18

No.   Time          Source      Destination  Protocol  Length Info
  224 32.534239398  127.0.0.1   127.0.0.1    TCP           76 53614 → 80 [SYN] Seq=0 Win=6
  225 32.534260106  127.0.0.1   127.0.0.1    TCP           76 80 → 53614 [SYN, ACK] Seq=0
  226 32.534273646  127.0.0.1   127.0.0.1    TCP           68 53614 → 80 [ACK] Seq=1 Ack=1
  227 32.534373063  127.0.0.1   127.0.0.1    HTTP         630 GET / HTTP/1.1
  228 32.534536114  127.0.0.1   127.0.0.1    TCP           68 80 → 53614 [ACK] Seq=1 Ack=5
  229 32.536277299  127.0.0.1   127.0.0.1    HTTP        3528 HTTP/1.1 200 OK  (text/html)
  230 32.536361898  127.0.0.1   127.0.0.1    TCP           68 53614 → 80 [ACK] Seq=563 Ack
  241 34.181565968  127.0.0.1   127.0.0.1    HTTP         630 GET / HTTP/1.1
  242 34.182439352  127.0.0.1   127.0.0.1    HTTP        3527 HTTP/1.1 200 OK  (text/html)
  243 34.182511438  127.0.0.1   127.0.0.1    TCP           68 53614 → 80 [ACK] Seq=1125 Ac
  260 39.184543662  127.0.0.1   127.0.0.1    TCP           68 80 → 53614 [FIN, ACK] Seq=69
  261 39.185168090  127.0.0.1   127.0.0.1    TCP           68 53614 → 80 [FIN, ACK] Seq=11
  262 39.185222364  127.0.0.1   127.0.0.1    TCP           68 80 → 53614 [ACK] Seq=6921 Ac

› Frame 227: 630 bytes on wire (5040 bits), 630 bytes captured (5040 bits) on interface any, id 0
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 53614, Dst Port: 80, Seq: 1, Ack: 1, Len: 562
› Hypertext Transfer Protocol
```

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.

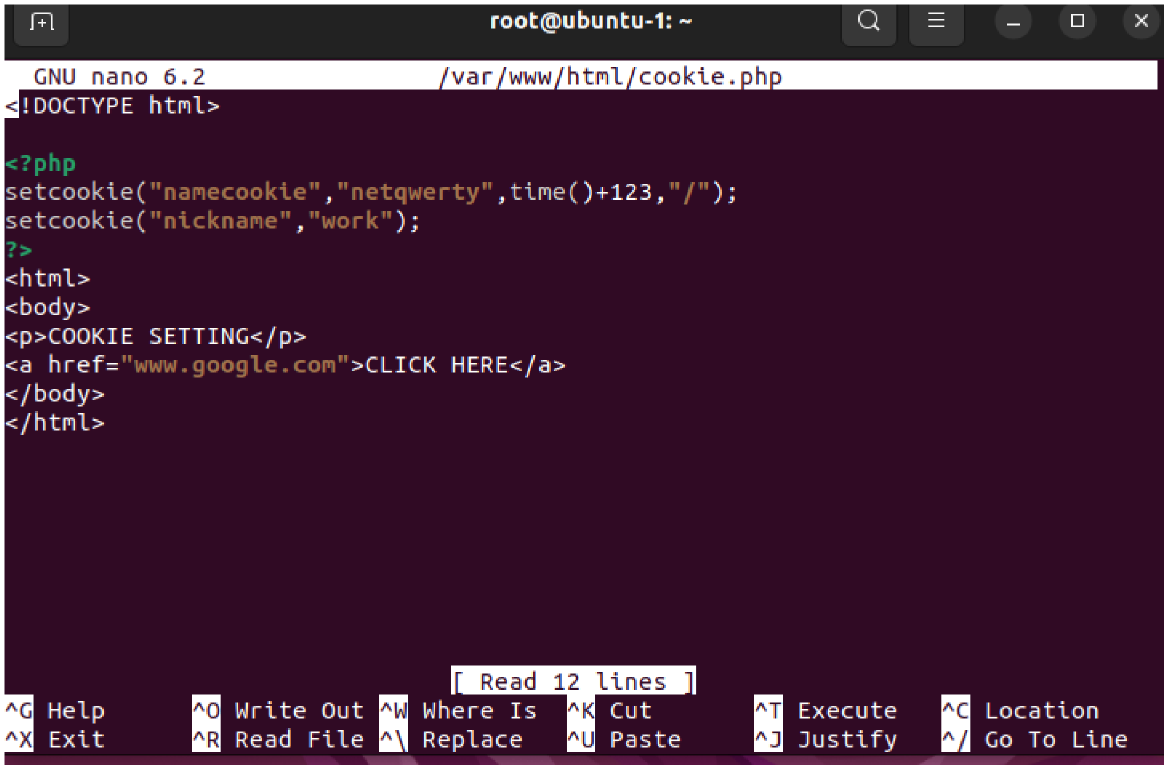


```
GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/
20100101 Firefox/103.0
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Basic bmV0d286MTIzNA==
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
If-Modified-Since: Wed, 15 Feb 2023 16:52:51 GMT
If-None-Match: "29af-5f4bfe929b00f-gzip"

HTTP/1.1 200 OK
Date: Thu, 16 Feb 2023 14:24:41 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Wed, 15 Feb 2023 16:52:51 GMT
ETag: "29af-5f4bfe929b00f-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3121
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

...........Z.s..........%.DJr...Y3....L...N...B$(.........
{w...K...i..".}a...y...O..?.\...
r...../.`...<.....+..w....I0&......S*...o.2.....p.....@.ux.
1...&x..:2..Al...da.~..T_t...........7.....A.<.,&..[b6.\...)...
9.'.T..*OMn.....<..a..L...h|:.N......d....y.m2..)3.*_k...l<~
{G#+...xi....P{v........L
K..v........a.M...&..*.............F.....6l.e.....Cn..d....T.z.....
6{...~.M...lY...@G...Z......R.6..q........1O.......+....O~..
8.....'vyE....y..")....\...\O.z...%t..~.....0.........i.x..q....
1&.......
```

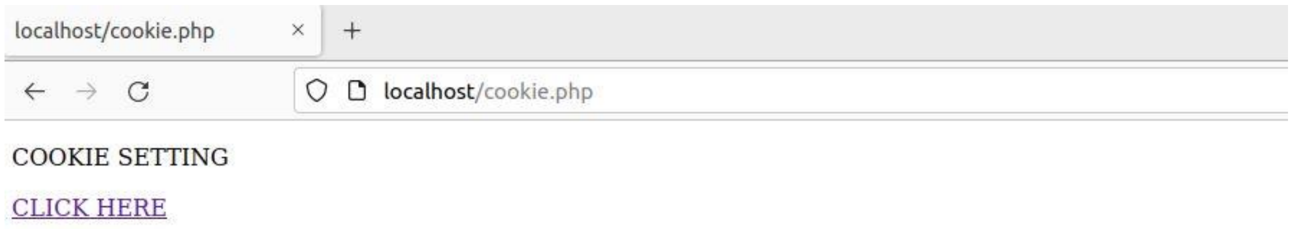
Steps of Execution (Cookie Setting)

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>
```



```
root@ubuntu-1: ~

  GNU nano 6.2                    /var/www/html/cookie.php
<!DOCTYPE html>

<?php
setcookie("namecookie","netqwerty",time()+123,"/");
setcookie("nickname","work");
?>
<html>
<body>
<p>COOKIE SETTING</p>
<a href="www.google.com">CLICK HERE</a>
</body>
</html>

                               [ Read 12 lines ]
^G Help       ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit       ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



COOKIE SETTING

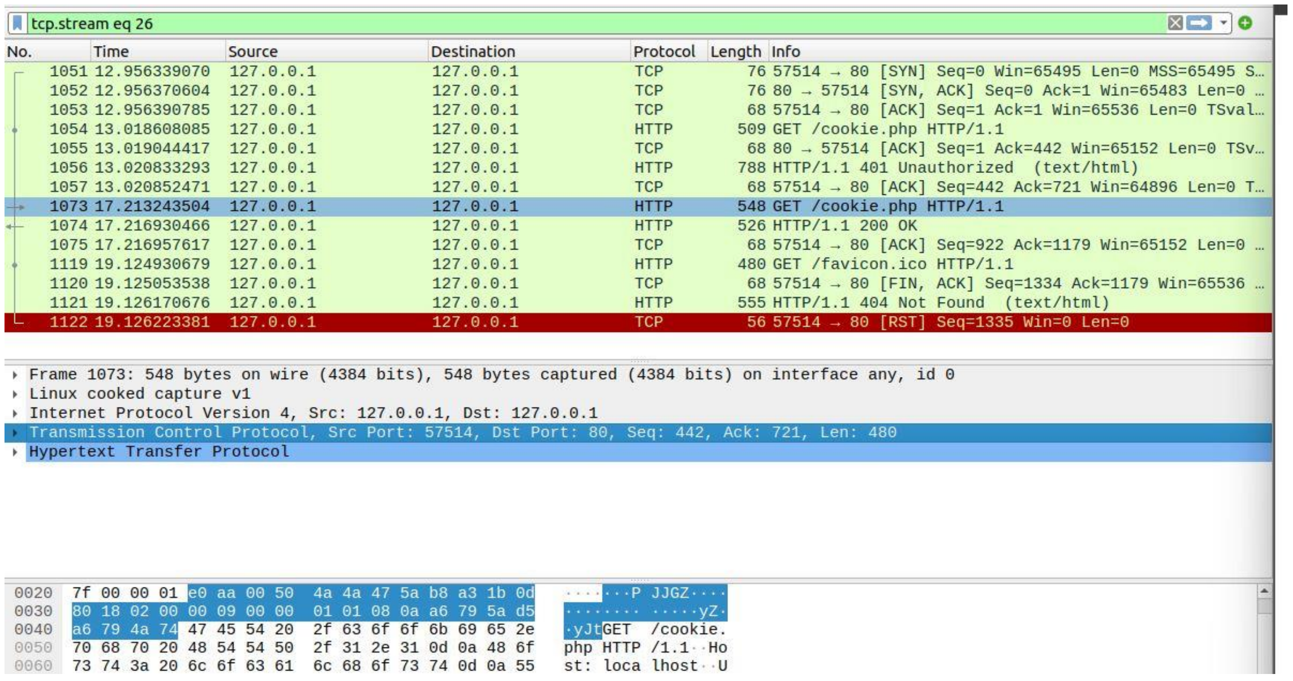
CLICK HERE

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.

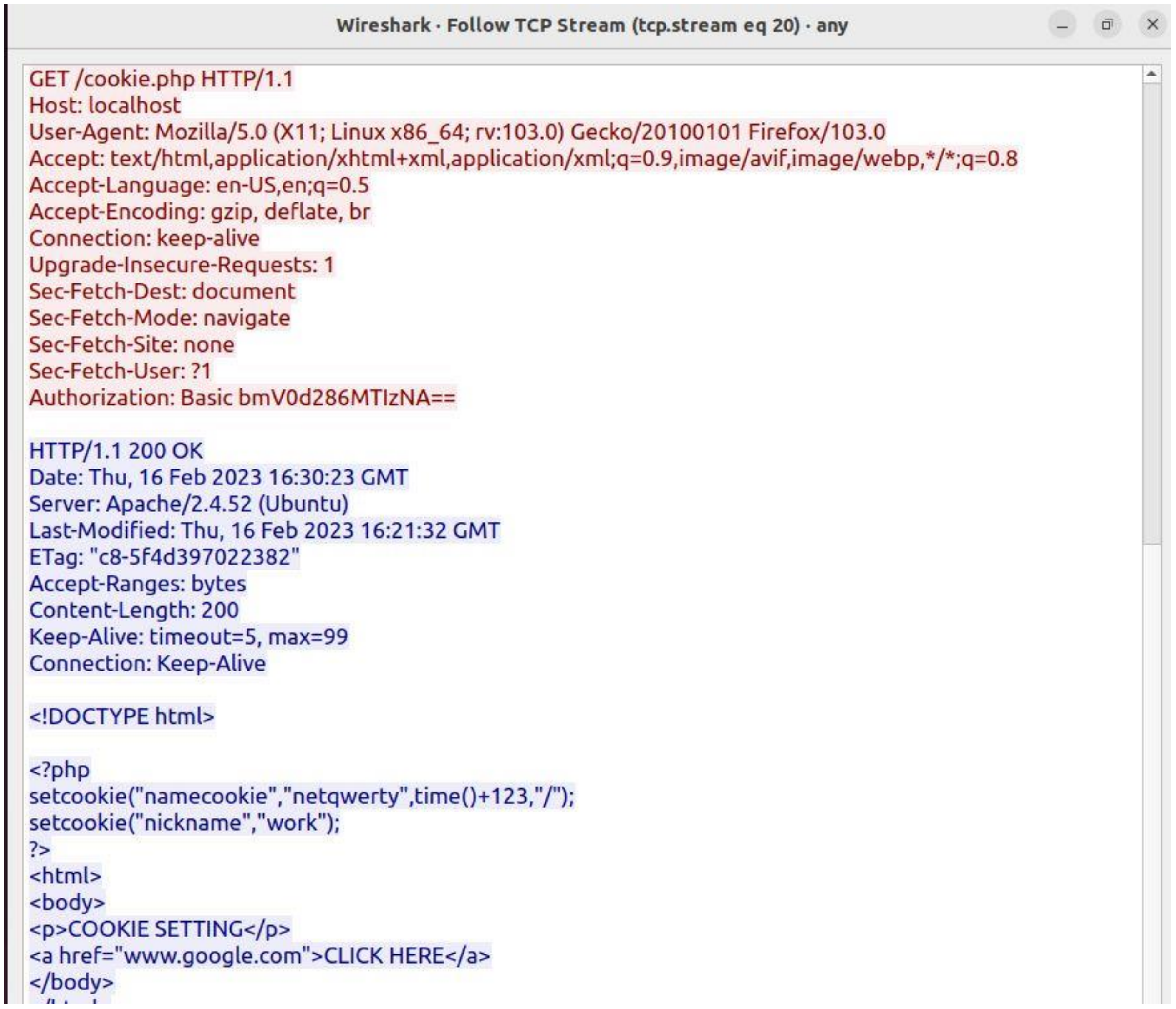


```
No.   Time          Source      Destination  Protocol Length Info
 1051 12.956339070  127.0.0.1   127.0.0.1    TCP        76 57514 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 S…
 1052 12.956370604  127.0.0.1   127.0.0.1    TCP        76 80 → 57514 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 …
 1053 12.956390785  127.0.0.1   127.0.0.1    TCP        68 57514 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval…
 1054 13.018608085  127.0.0.1   127.0.0.1    HTTP      509 GET /cookie.php HTTP/1.1
 1055 13.019044417  127.0.0.1   127.0.0.1    TCP        68 80 → 57514 [ACK] Seq=1 Ack=442 Win=65152 Len=0 TSv…
 1056 13.020833293  127.0.0.1   127.0.0.1    HTTP      788 HTTP/1.1 401 Unauthorized  (text/html)
 1057 13.020852471  127.0.0.1   127.0.0.1    TCP        68 57514 → 80 [ACK] Seq=442 Ack=721 Win=64896 Len=0 T…
 1073 17.213243504  127.0.0.1   127.0.0.1    HTTP      548 GET /cookie.php HTTP/1.1
 1074 17.216930466  127.0.0.1   127.0.0.1    HTTP      526 HTTP/1.1 200 OK
 1075 17.216957617  127.0.0.1   127.0.0.1    TCP        68 57514 → 80 [ACK] Seq=922 Ack=1179 Win=65152 Len=0 …
 1119 19.124930679  127.0.0.1   127.0.0.1    HTTP      480 GET /favicon.ico HTTP/1.1
 1120 19.125053538  127.0.0.1   127.0.0.1    TCP        68 57514 → 80 [FIN, ACK] Seq=1334 Ack=1179 Win=65536 …
 1121 19.126170676  127.0.0.1   127.0.0.1    HTTP      555 HTTP/1.1 404 Not Found  (text/html)
 1122 19.126223381  127.0.0.1   127.0.0.1    TCP        56 57514 → 80 [RST] Seq=1335 Win=0 Len=0
```

```
› Frame 1073: 548 bytes on wire (4384 bits), 548 bytes captured (4384 bits) on interface any, id 0
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
› Transmission Control Protocol, Src Port: 57514, Dst Port: 80, Seq: 442, Ack: 721, Len: 480
› Hypertext Transfer Protocol
```

```
0020  7f 00 00 01 e0 aa 00 50  4a 4a 47 5a b8 a3 1b 0d   ·······P JJGZ····
0030  80 18 02 00 00 09 00 00  01 01 08 0a a6 79 5a d5   ········ ·····yZ·
0040  a6 79 4a 74 47 45 54 20  2f 63 6f 6f 6b 69 65 2e   ·yJtGET  /cookie.
0050  70 68 70 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   php HTTP /1.1··Ho
0060  73 74 3a 20 6c 6f 63 61  6c 68 6f 73 74 0d 0a 55   st: loca lhost··U
```

The cookie is set as shown in the above screenshot.



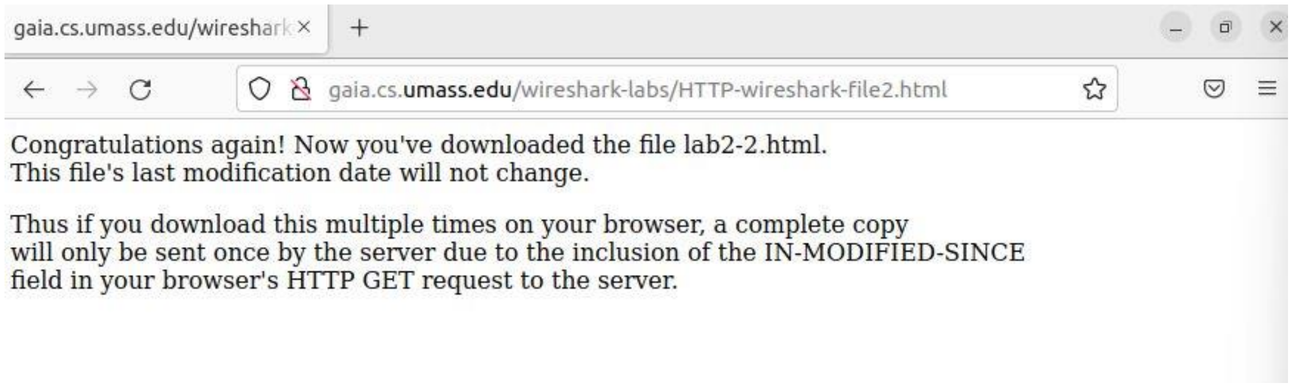
```
GET /cookie.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/20100101 Firefox/103.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Authorization: Basic bmV0d286MTIzNA==

HTTP/1.1 200 OK
Date: Thu, 16 Feb 2023 16:30:23 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Thu, 16 Feb 2023 16:21:32 GMT
ETag: "c8-5f4d397022382"
Accept-Ranges: bytes
Content-Length: 200
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive

<!DOCTYPE html>

<?php
setcookie("namecookie","netqwerty",time()+123,"/");
setcookie("nickname","work");
?>
<html>
<body>
<p>COOKIE SETTING</p>
<a href="www.google.com">CLICK HERE</a>
</body>
```

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

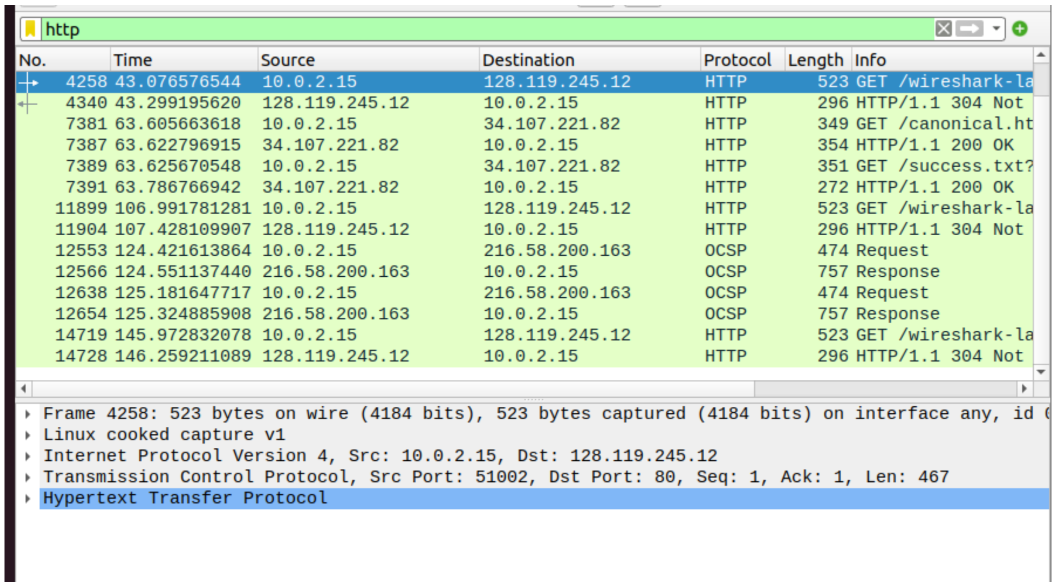
Before performing the steps below, make sure your browser's cache is empty. (To do this under Firefox, select Tools -> Clear Recent History and check the Cache box). Now do the following:

- 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/wireshark-labs/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 packet-listing window.

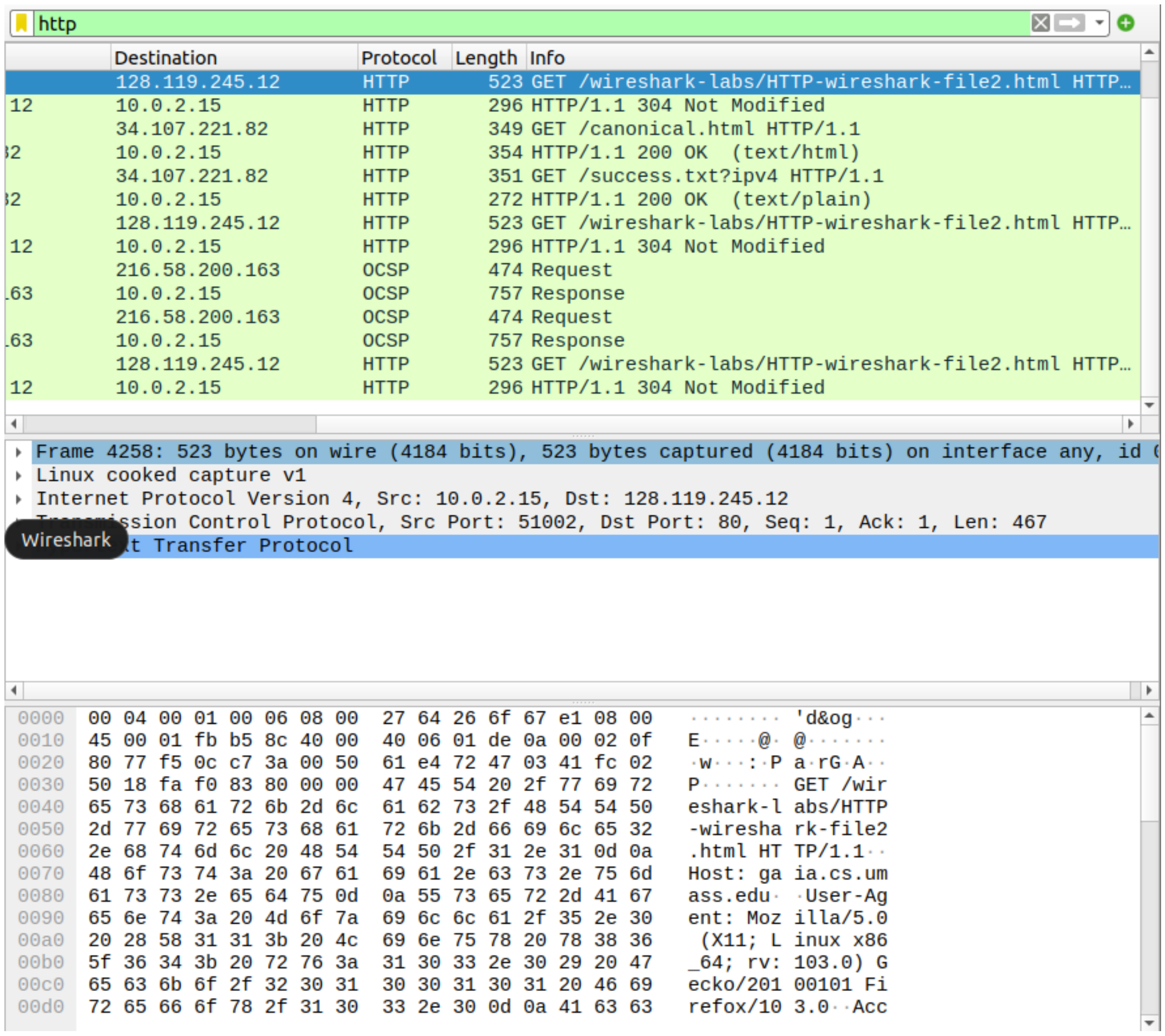

gaia.cs.umass.edu/wireshark
gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

Congratulations again! Now you've downloaded the file lab2-2.html.
This file's last modification date will not change.

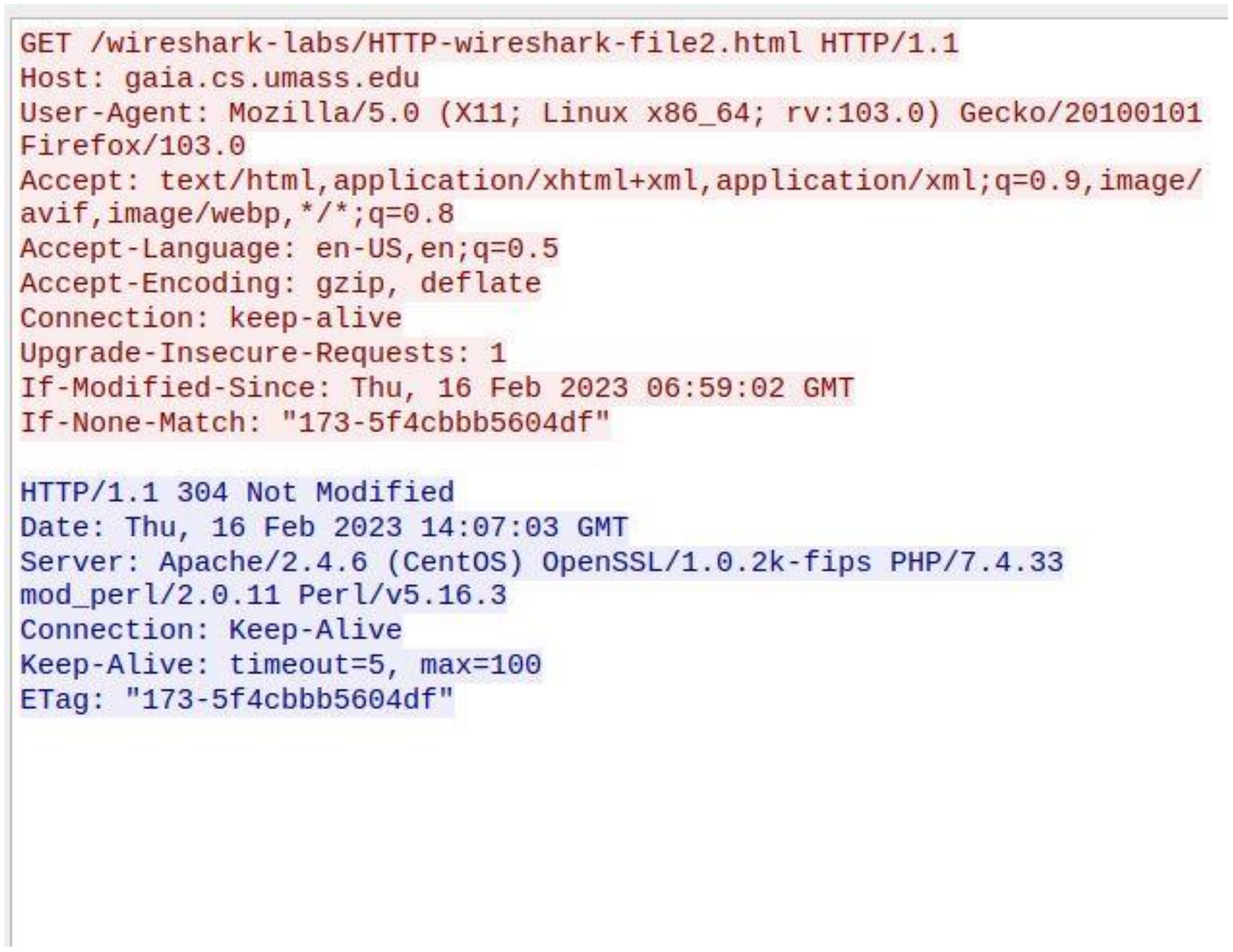
Thus if you download this multiple times on your browser, a complete copy
will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE
field in your browser's HTTP GET request to the server.



```
http
No.     Time            Source           Destination      Protocol Length Info
   4258 43.076576544    10.0.2.15        128.119.245.12   HTTP     523 GET /wireshark-la
   4340 43.299195620    128.119.245.12   10.0.2.15        HTTP     296 HTTP/1.1 304 Not
   7381 63.605663618    10.0.2.15        34.107.221.82    HTTP     349 GET /canonical.ht
   7387 63.622796915    34.107.221.82    10.0.2.15        HTTP     354 HTTP/1.1 200 OK
   7389 63.625670548    10.0.2.15        34.107.221.82    HTTP     351 GET /success.txt?
   7391 63.786766942    34.107.221.82    10.0.2.15        HTTP     272 HTTP/1.1 200 OK
  11899 106.991781281   10.0.2.15        128.119.245.12   HTTP     523 GET /wireshark-la
  11904 107.428109907   128.119.245.12   10.0.2.15        HTTP     296 HTTP/1.1 304 Not
  12553 124.421613864   10.0.2.15        216.58.200.163   OCSP     474 Request
  12566 124.551137440   216.58.200.163   10.0.2.15        OCSP     757 Response
  12638 125.181647717   10.0.2.15        216.58.200.163   OCSP     474 Request
  12654 125.324885908   216.58.200.163   10.0.2.15        OCSP     757 Response
  14719 145.972832078   10.0.2.15        128.119.245.12   HTTP     523 GET /wireshark-la
  14728 146.259211089   128.119.245.12   10.0.2.15        HTTP     296 HTTP/1.1 304 Not

Frame 4258: 523 bytes on wire (4184 bits), 523 bytes captured (4184 bits) on interface any, id (
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 51002, Dst Port: 80, Seq: 1, Ack: 1, Len: 467
Hypertext Transfer Protocol
```



```
http
      Destination       Protocol Length Info
      128.119.245.12    HTTP        523 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP…
12    10.0.2.15         HTTP        296 HTTP/1.1 304 Not Modified
      34.107.221.82     HTTP        349 GET /canonical.html HTTP/1.1
32    10.0.2.15         HTTP        354 HTTP/1.1 200 OK  (text/html)
      34.107.221.82     HTTP        351 GET /success.txt?ipv4 HTTP/1.1
32    10.0.2.15         HTTP        272 HTTP/1.1 200 OK  (text/plain)
      128.119.245.12    HTTP        523 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP…
12    10.0.2.15         HTTP        296 HTTP/1.1 304 Not Modified
      216.58.200.163    OCSP        474 Request
.63   10.0.2.15         OCSP        757 Response
      216.58.200.163    OCSP        474 Request
.63   10.0.2.15         OCSP        757 Response
      128.119.245.12    HTTP        523 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP…
12    10.0.2.15         HTTP        296 HTTP/1.1 304 Not Modified

Frame 4258: 523 bytes on wire (4184 bits), 523 bytes captured (4184 bits) on interface any, id (
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 51002, Dst Port: 80, Seq: 1, Ack: 1, Len: 467
Hypertext Transfer Protocol

0000  00 04 00 01 00 06 08 00  27 64 26 6f 67 e1 08 00   ········ 'd&og···
0010  45 00 01 fb b5 8c 40 00  40 06 01 de 0a 00 02 0f   E·····@· @·······
0020  80 77 f5 0c c7 3a 00 50  61 e4 72 47 03 41 fc 02   ·w···:·P a·rG·A··
0030  50 18 fa f0 83 80 00 00  47 45 54 20 2f 77 69 72   P······· GET /wir
0040  65 73 68 61 72 6b 2d 6c  61 62 73 2f 48 54 54 50   eshark-l abs/HTTP
0050  2d 77 69 72 65 73 68 61  72 6b 2d 66 69 6c 65 32   -wiresha rk-file2
0060  2e 68 74 6d 6c 20 48 54  54 50 2f 31 2e 31 0d 0a   .html HT TP/1.1··
0070  48 6f 73 74 3a 20 67 61  69 61 2e 63 73 2e 75 6d   Host: ga ia.cs.um
0080  61 73 73 2e 65 64 75 0d  0a 55 73 65 72 2d 41 67   ass.edu· ·User-Ag
0090  65 6e 74 3a 20 4d 6f 7a  69 6c 6c 61 2f 35 2e 30   ent: Moz illa/5.0
00a0  20 28 58 31 31 3b 20 4c  69 6e 75 78 20 78 38 36    (X11; L inux x86
00b0  5f 36 34 3b 20 72 76 3a  31 30 33 2e 30 29 20 47   _64; rv: 103.0) G
00c0  65 63 6b 6f 2f 32 30 31  30 30 31 30 31 20 46 69   ecko/201 00101 Fi
00d0  72 65 66 6f 78 2f 31 30  33 2e 30 0d 0a 41 63 63   refox/10 3.0··Acc
```



```
GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/20100101
Firefox/103.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Thu, 16 Feb 2023 06:59:02 GMT
If-None-Match: "173-5f4cbbb5604df"

HTTP/1.1 304 Not Modified
Date: Thu, 16 Feb 2023 14:07:03 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33
mod_perl/2.0.11 Perl/v5.16.3
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
ETag: "173-5f4cbbb5604df"
```

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

Repeat the above task with some images on the server.

Attach screenshots wherever necessary.