

# CN Lab Report – Week 3

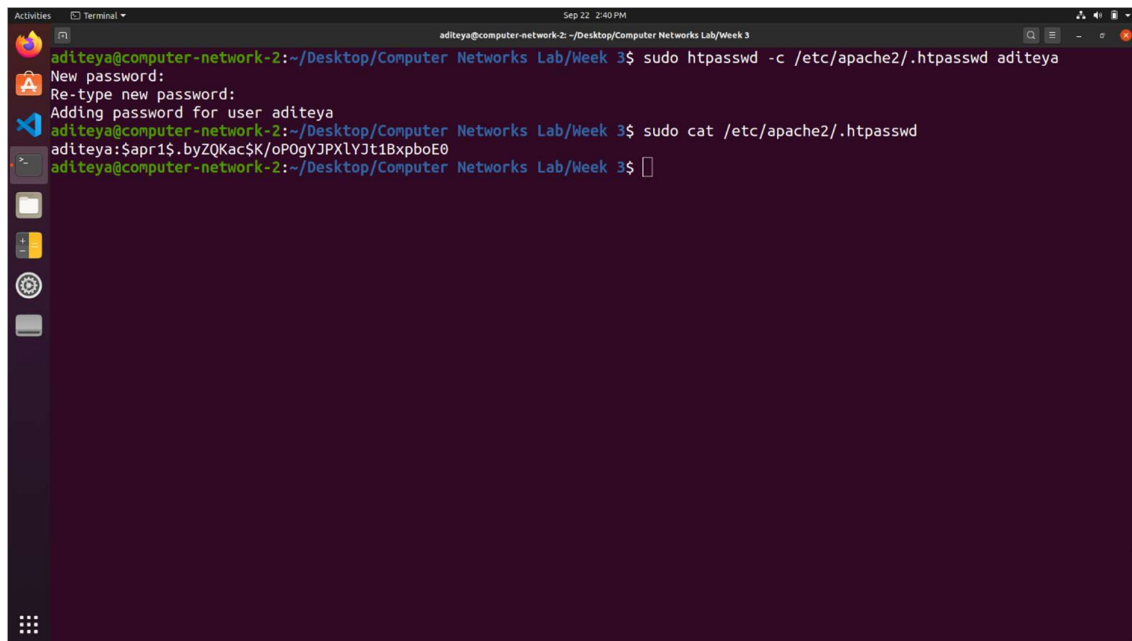
PES1201800366

Aditeya Baral

## 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.

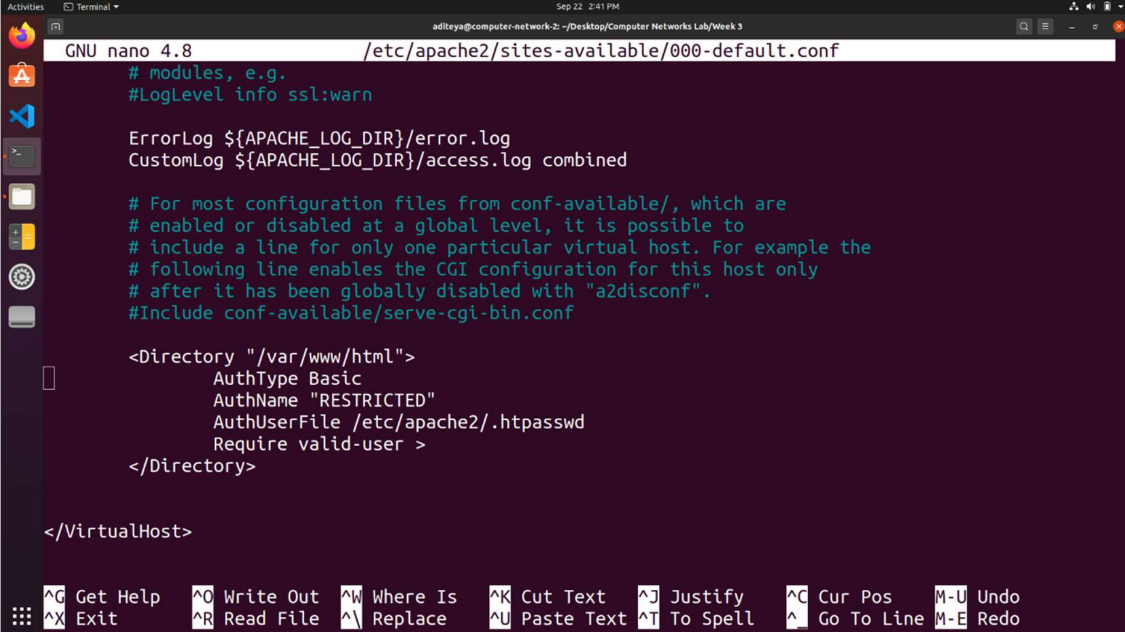


```
aditeya@computer-network-2: ~/Desktop/Computer Networks Lab/Week 3
aditeya@computer-network-2:~/Desktop/Computer Networks Lab/Week 3$ sudo htpasswd -c /etc/apache2/.htpasswd aditeya
New password:
Re-type new password:
Adding password for user aditeya
aditeya@computer-network-2:~/Desktop/Computer Networks Lab/Week 3$ sudo cat /etc/apache2/.htpasswd
aditeya:$apr1$.byZQKac$K/oP0gYJPXLYJt1BxpboE0
aditeya@computer-network-2:~/Desktop/Computer Networks Lab/Week 3$
```

### 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/000-default.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**.



```

GNU nano 4.8 /etc/apache2/sites-available/000-default.conf
# 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

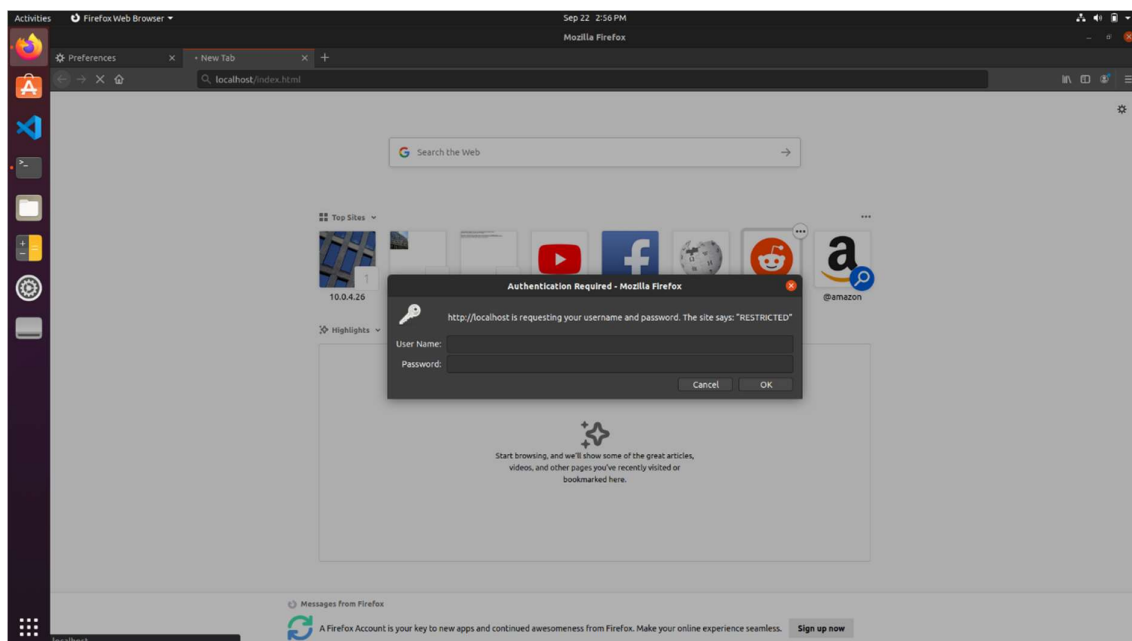
<Directory "/var/www/html">
    AuthType Basic
    AuthName "RESTRICTED"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user >
</Directory>

</VirtualHost>

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^_ Replace   ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo
  
```

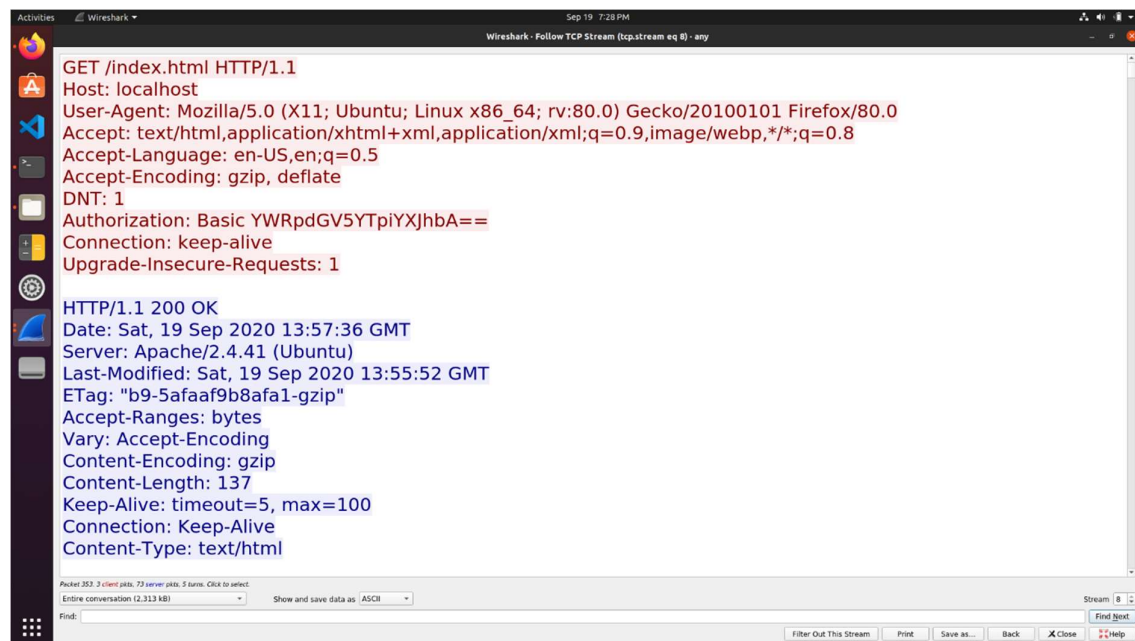
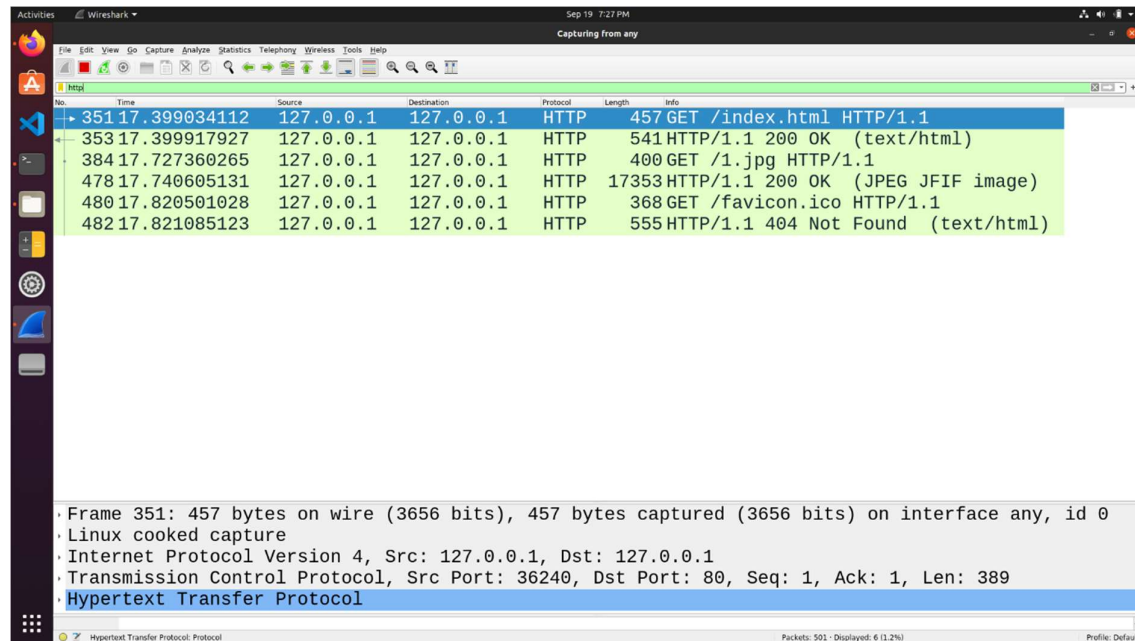
### 1.3 Accessing Localhost

- 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 first GET request corresponding to the HTML file is analysed and its TCP Stream is expanded, and parameters examined.



## 1.5 Decrypting Base64 Encryption

- 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
  - **YWRpdGV5YTpiYXJhbA==** can be first converted to a 6-bit binary equivalent

Y	011000
W	010110
R	010001
p	101001
d	011101
G	000110
V	010101
5	111001
Y	011000
T	010011
p	101001
i	100010
Y	011000
X	010111
J	001001
h	100001
b	011011
A	000000

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

01100001	a
01100100	d
01101001	i
01110100	t
01100101	e
01111001	y
01100001	a
00111010	:
01100010	b
01100001	a
01110010	r
01100001	a
01101100	l

## 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

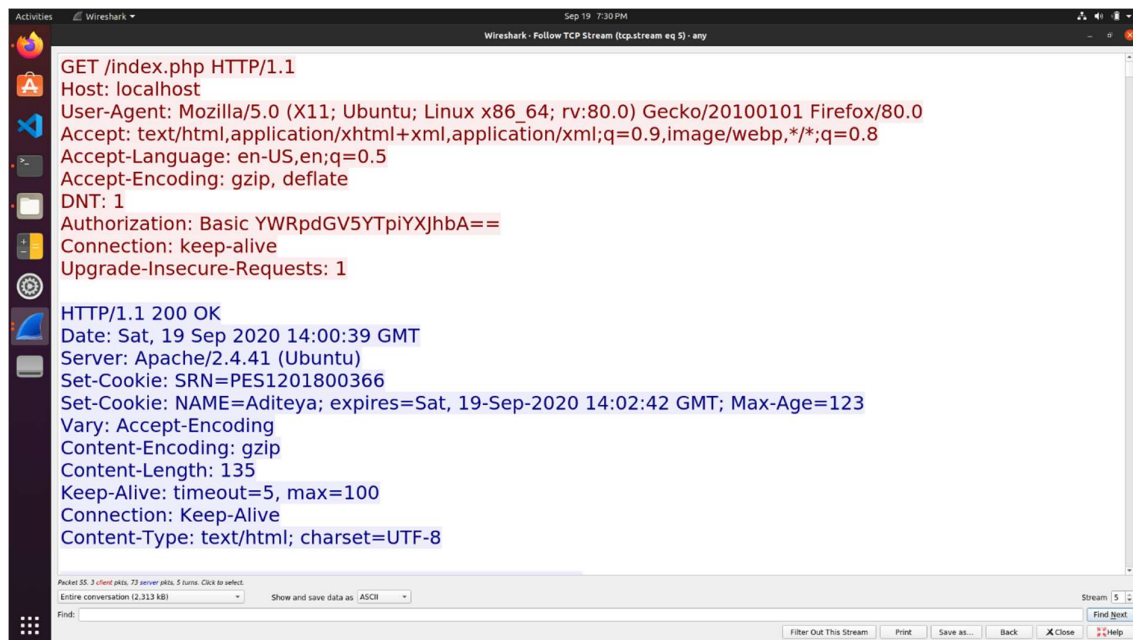
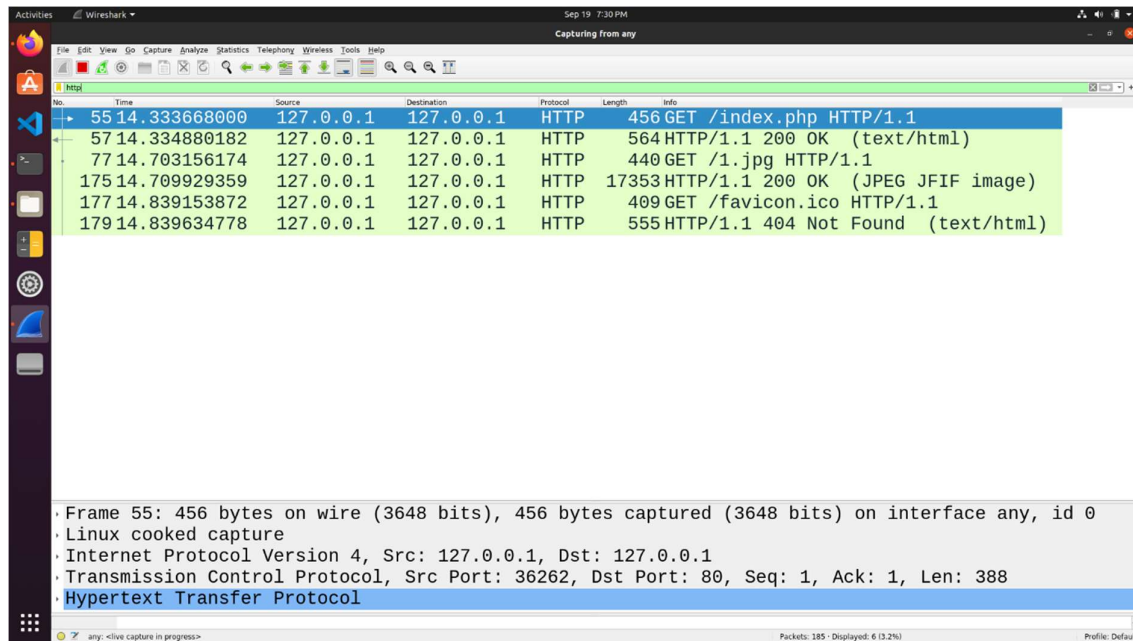
```

1. <html>
2.     <?php
3.         setcookie("SRN", "PES1201800366");
4.         setcookie("NAME", "Aditeya", time()+123);
5.     ?>
6.     <head>
7.         <title>Computer Networks Lab Week 3</title>
8.         <body>
9.             
10.        </body>
11.    </head>
12. </html>

```

### 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 analysed 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.



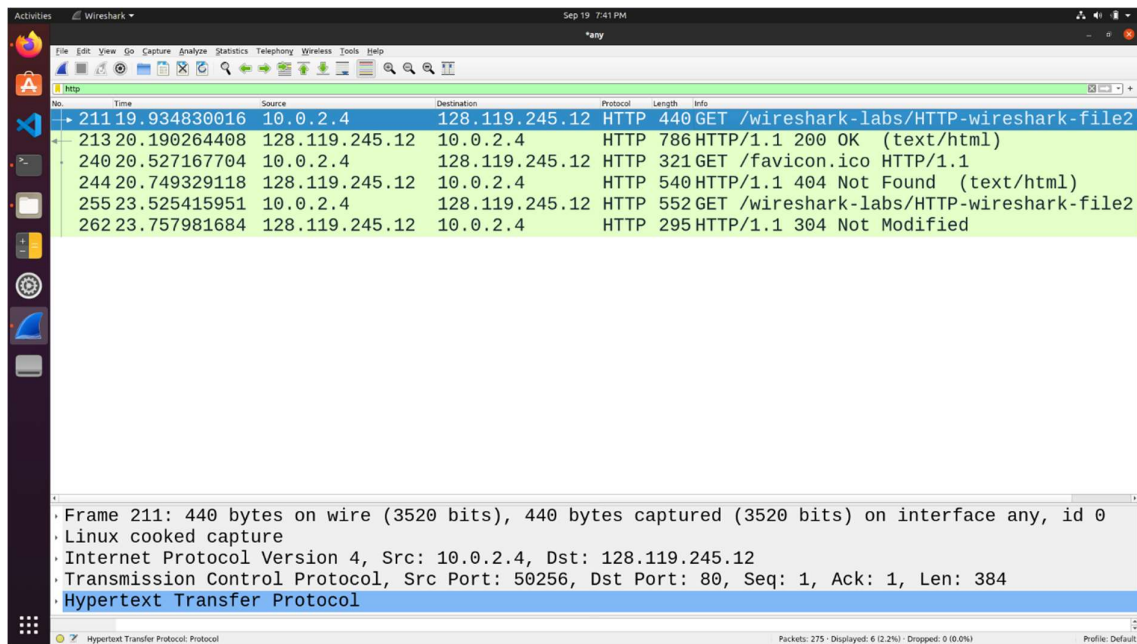
### 3. Conditional GET

- A conditional HTTP response is one that carries the resource only if 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.

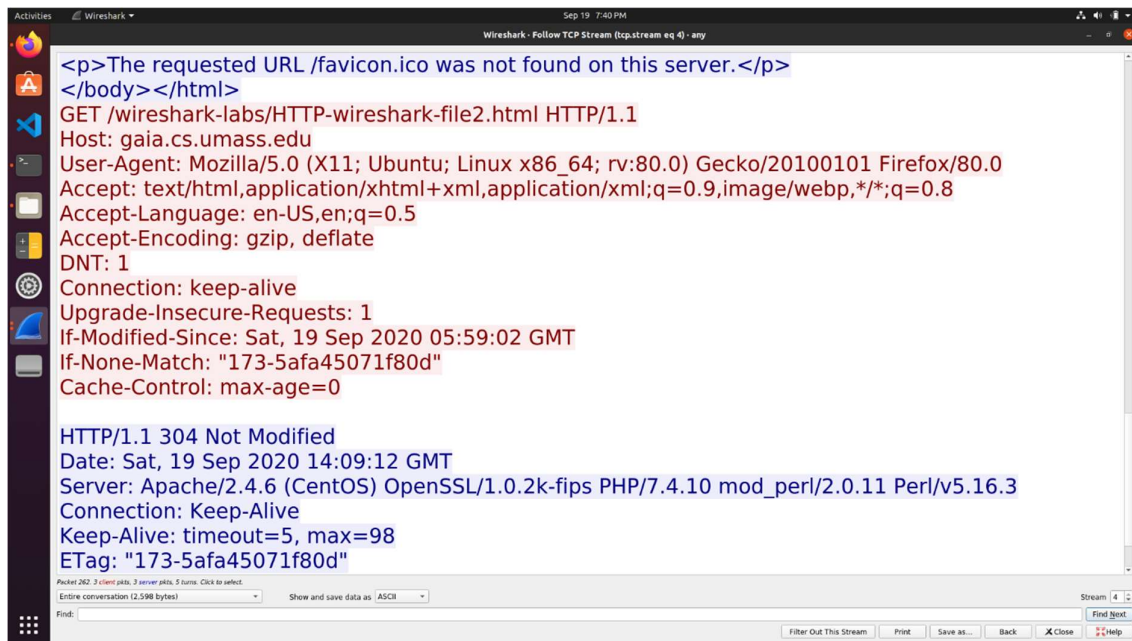
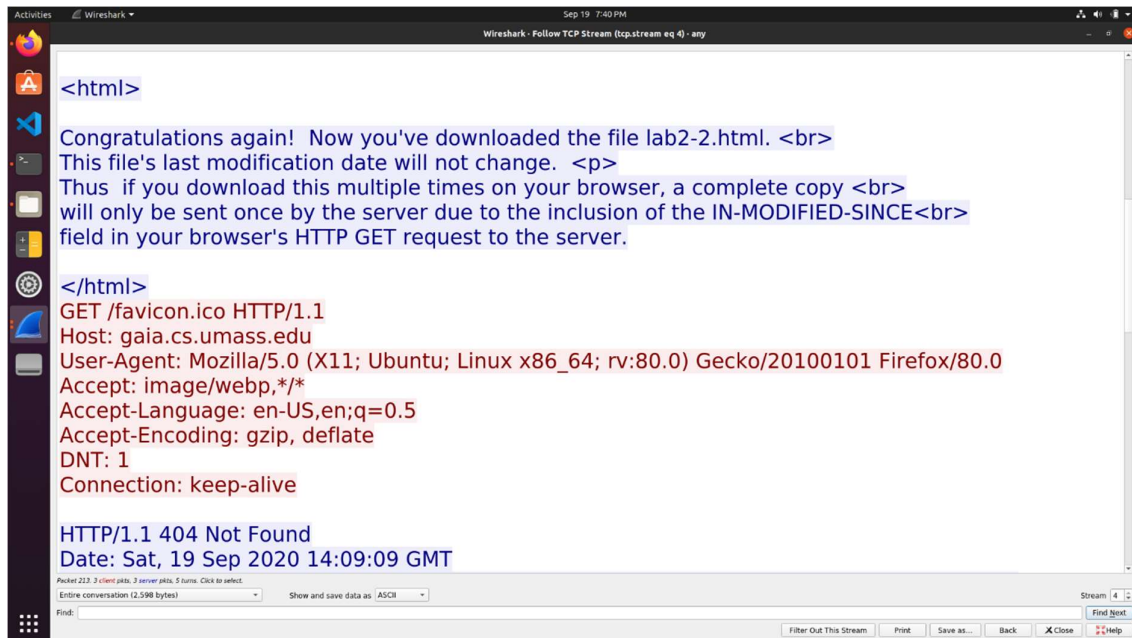


### 3.1 Repeat Requests for HTML Page

- 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.



## First Request from Server – 200 OK



## Second Request from Server – 304 Not Modified

### 3.2 Conditional GET on Localhost

- 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.

