**Name**: Khushei Meghana Meda

**SRN**: PES1201800416

**Week number**: 3

**Name of experiment**: Understand working of HTTP Headers

**Date**: 19-09-2020

**Objectives of the experiment**: To understand working of HTTP headers, Conditional Get: If-Modified-Since, HTTP Cookies: Cookie and Set-Cookie, Authentication: Auth-Basic

**Steps of Execution (for Password Authentication)**

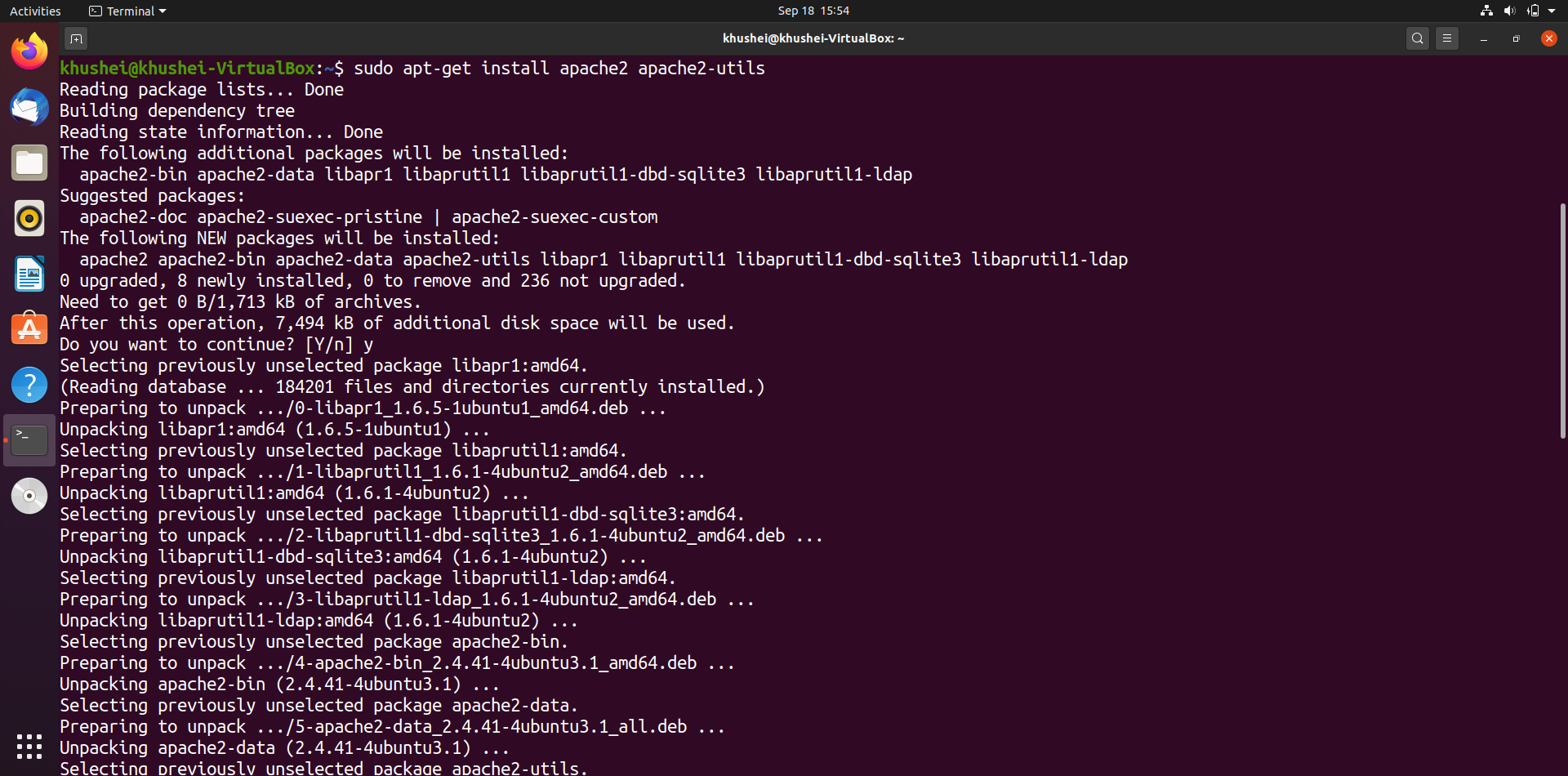
1. Executing the below commands on the terminal.

--> To update and integrate the existing softwares

**sudo apt-get update**

--> To install the apache utility

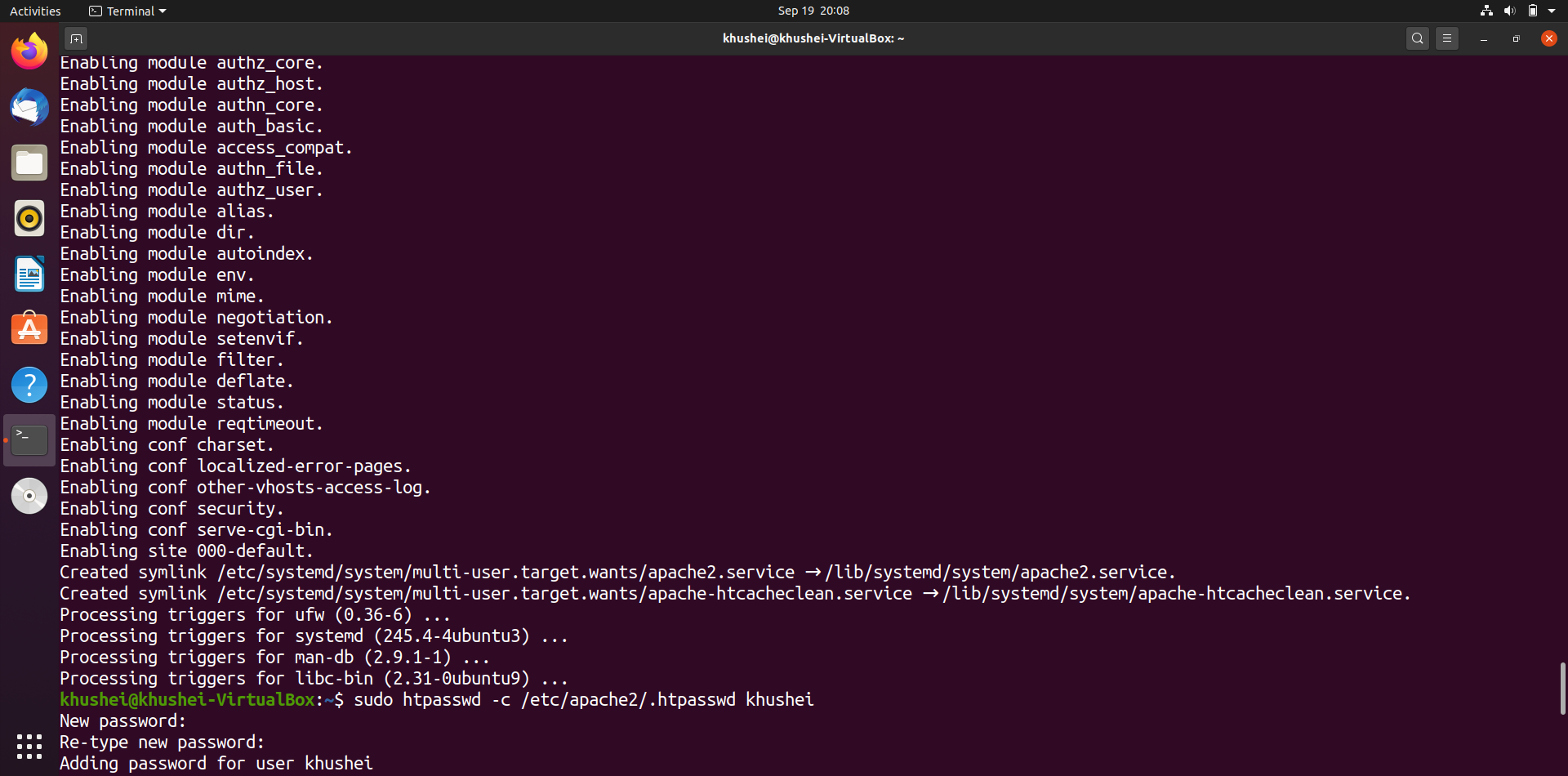
**sudo apt-get install apache2 apache2-utils**



--> Provide username and password to set authentication

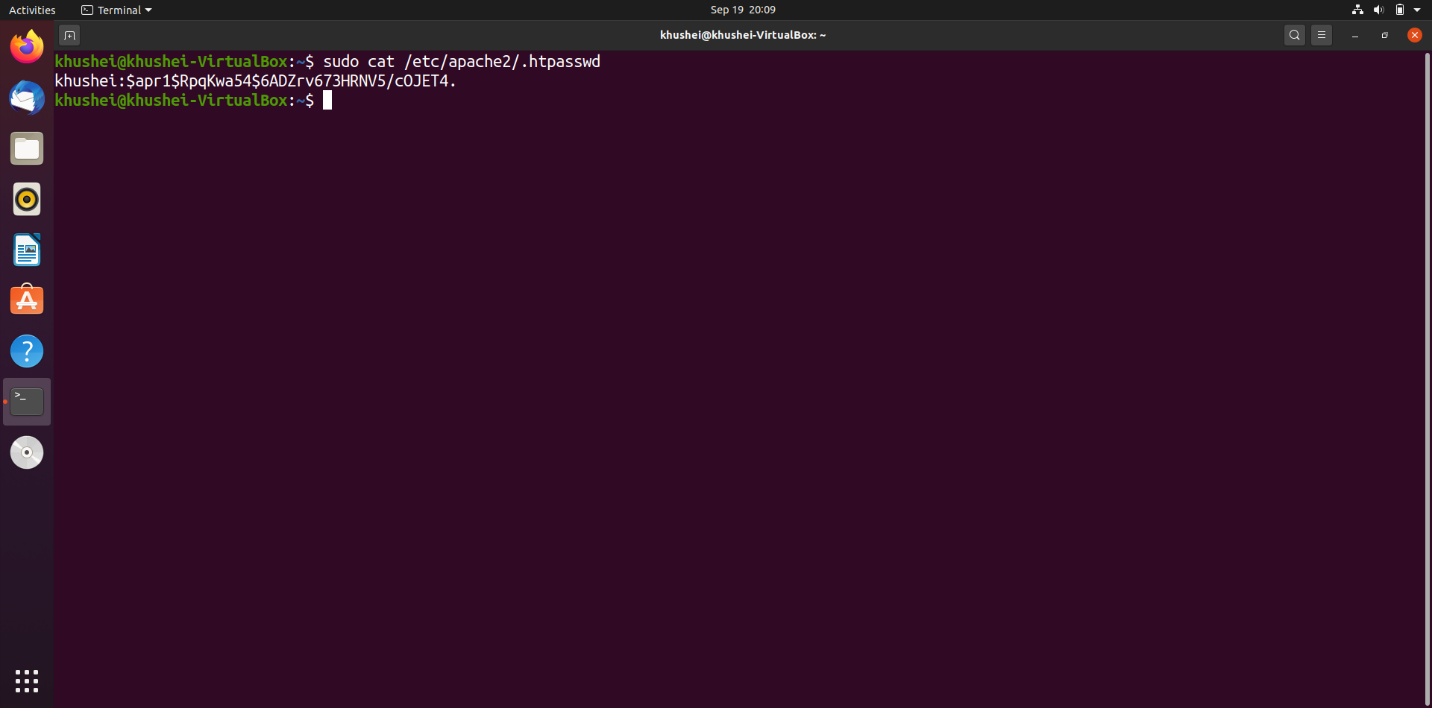
**sudo htpasswd -c /etc/apache2/.htpasswd ANY\_USERNAME**

I have given the username as khushei



--> View the authentication

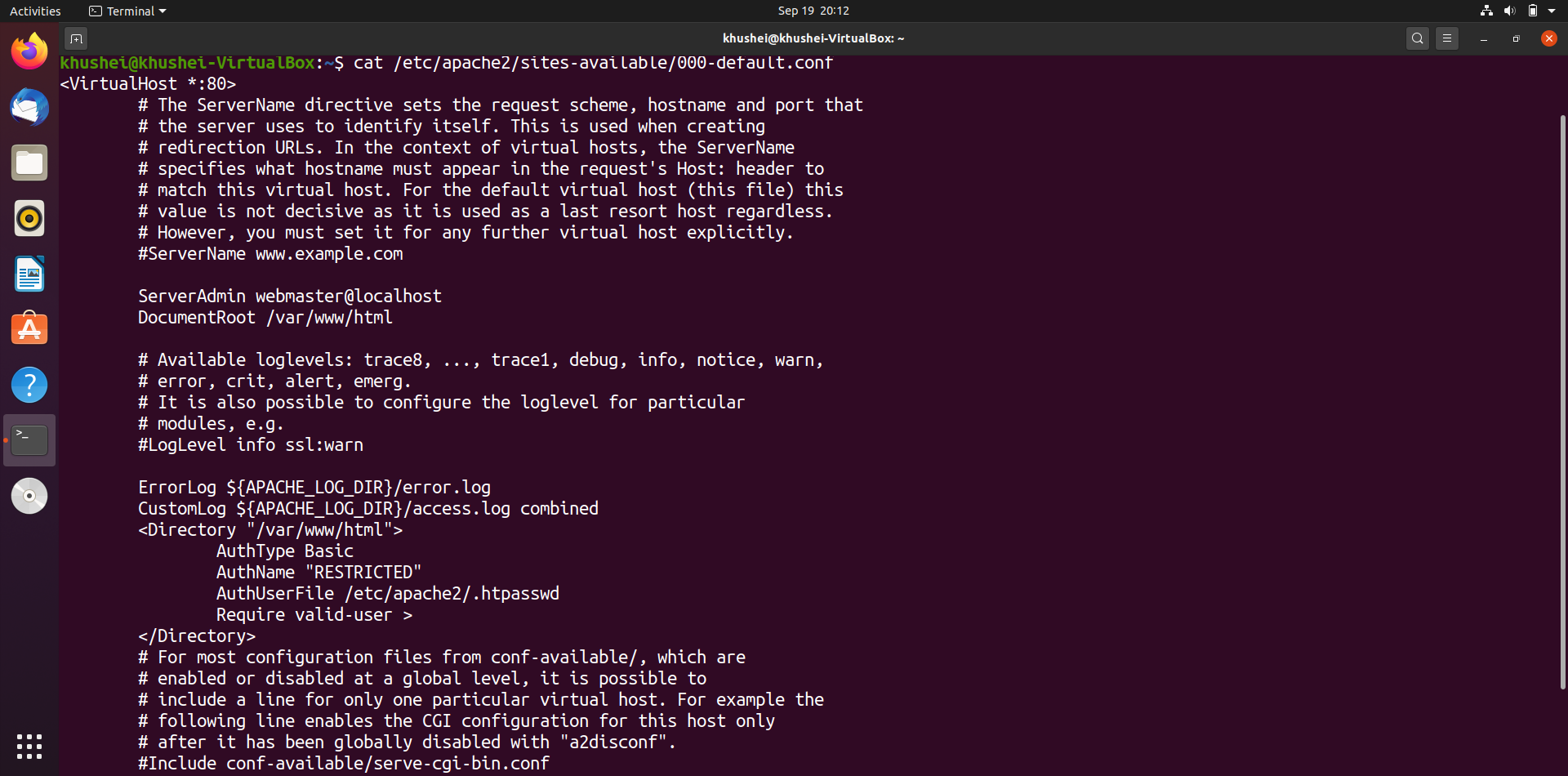
**sudo cat /etc/apache2/.htpasswd**



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

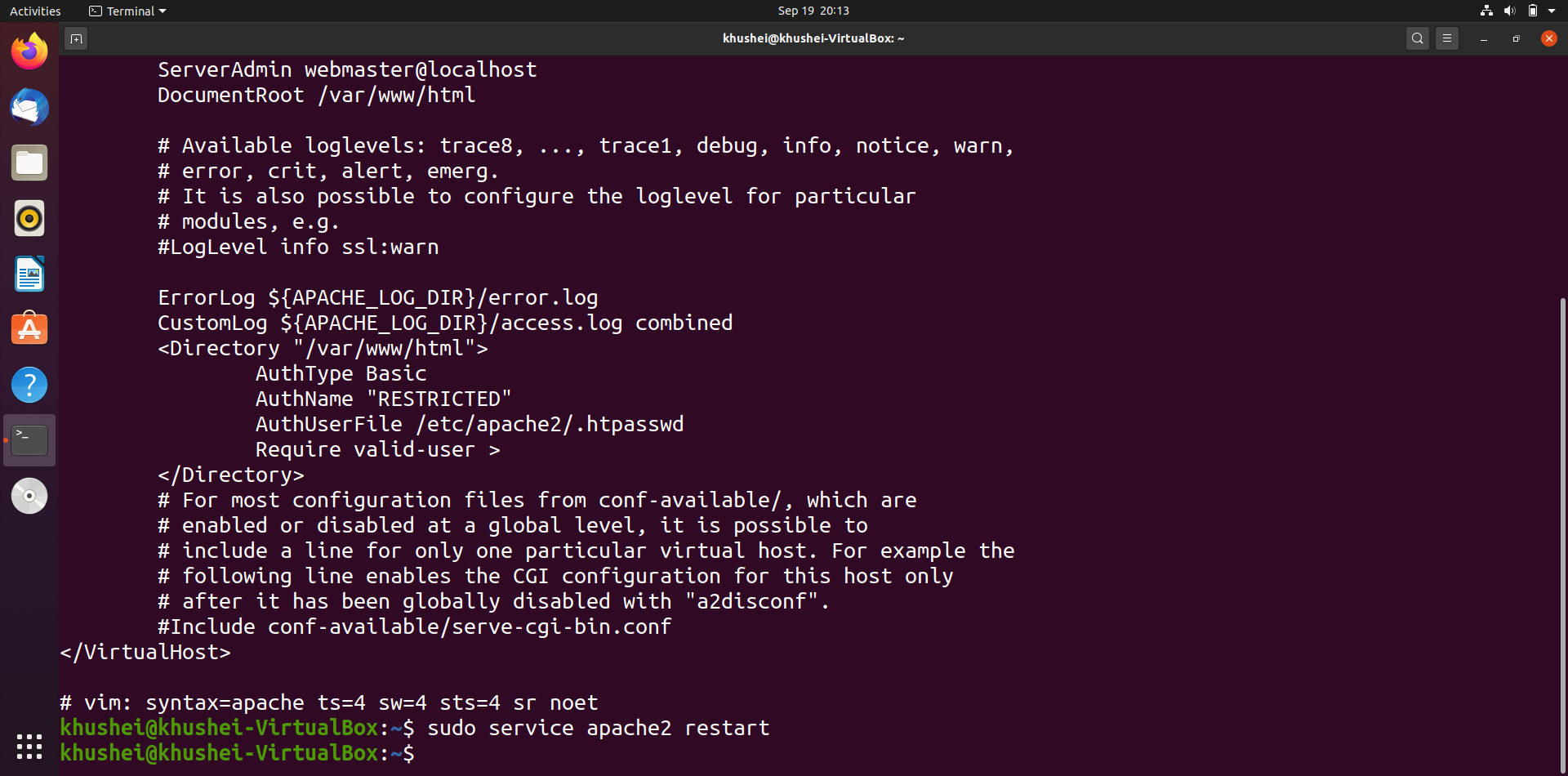
--> Opening the file for setting authentication

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

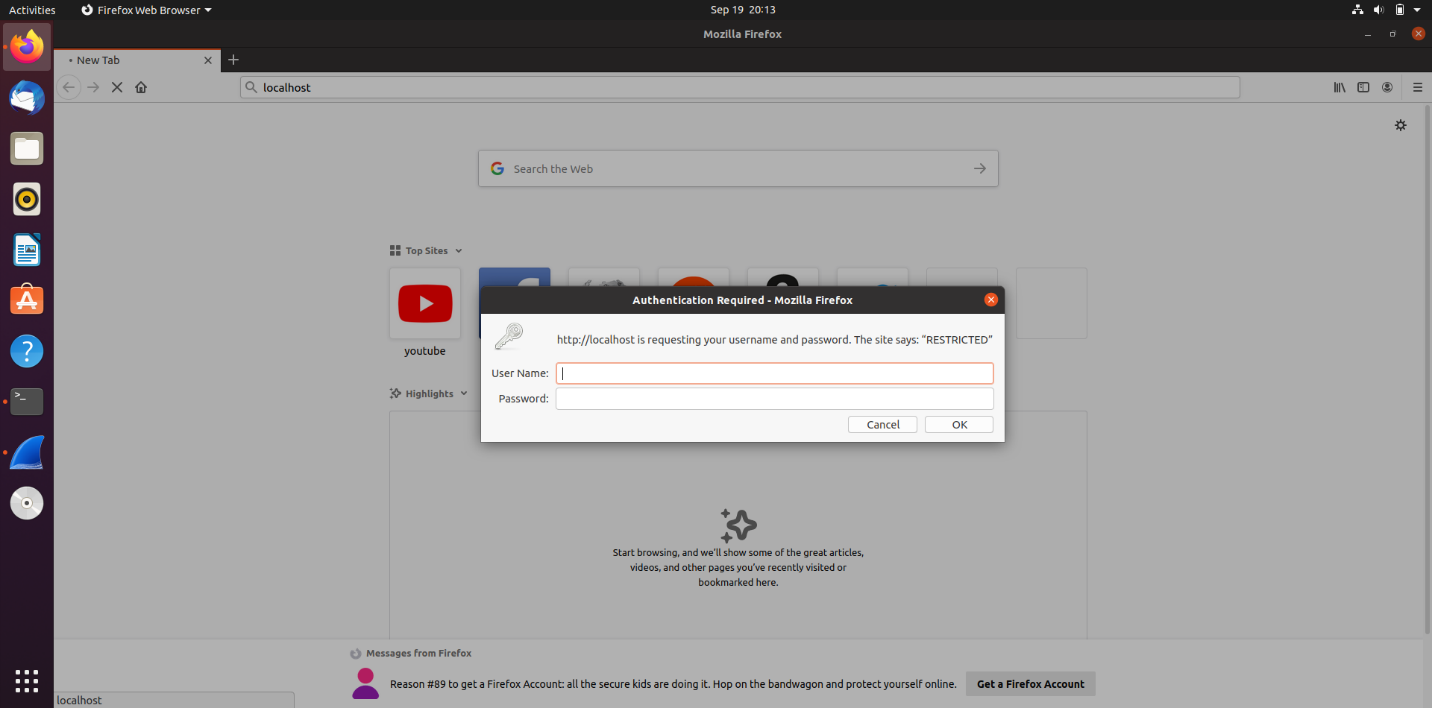


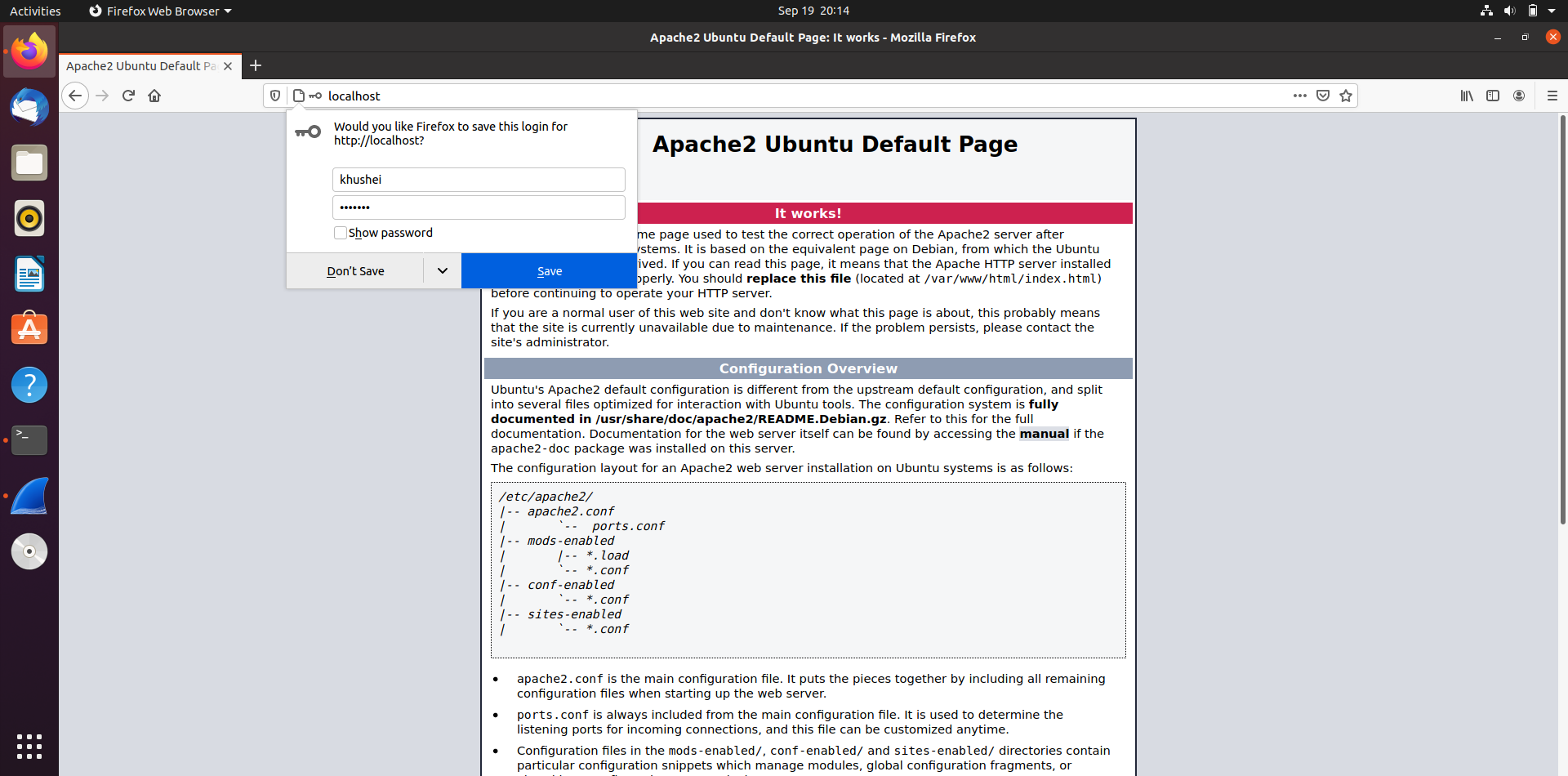
3. Password policy implementation is done by restarting the server as:

**sudo service apache2 restart**

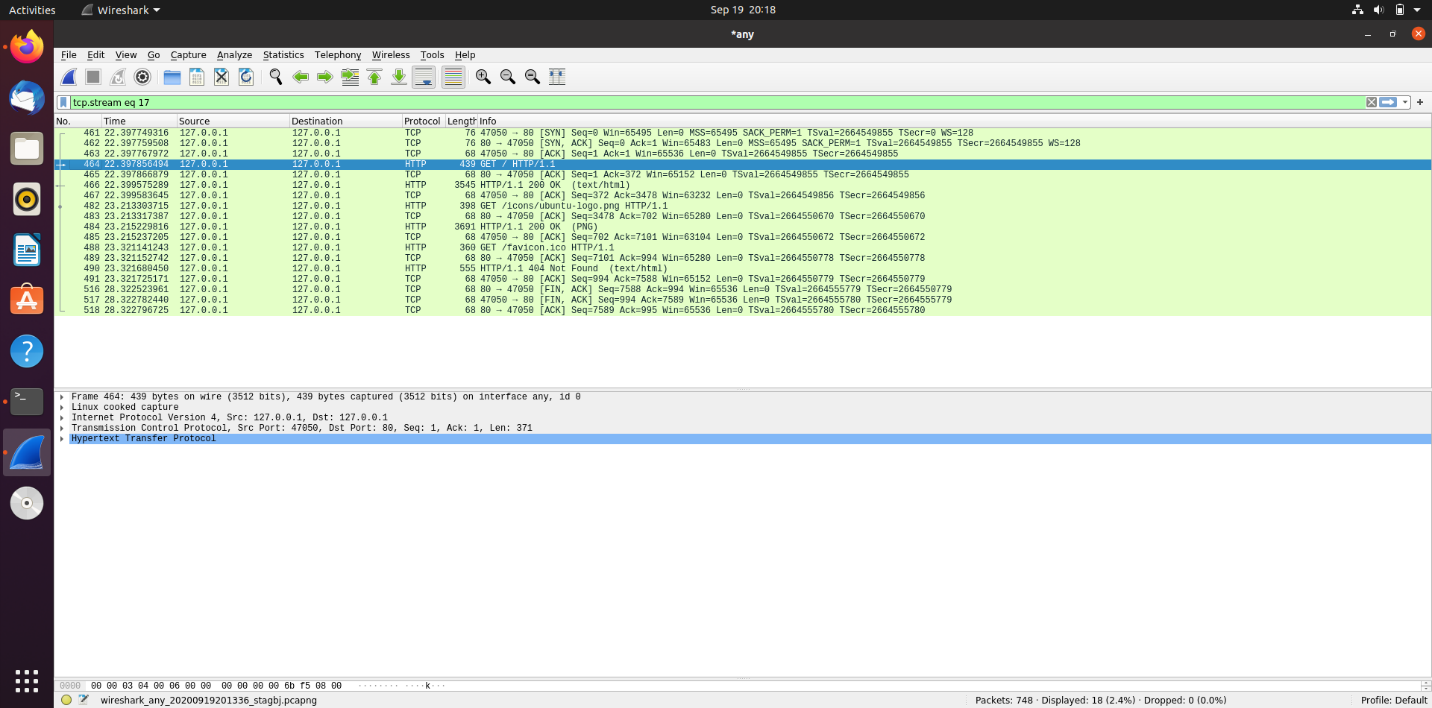


4. The localhost is then accessed using the Firefox browser requiring a username and a password set during the authentication phase.





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

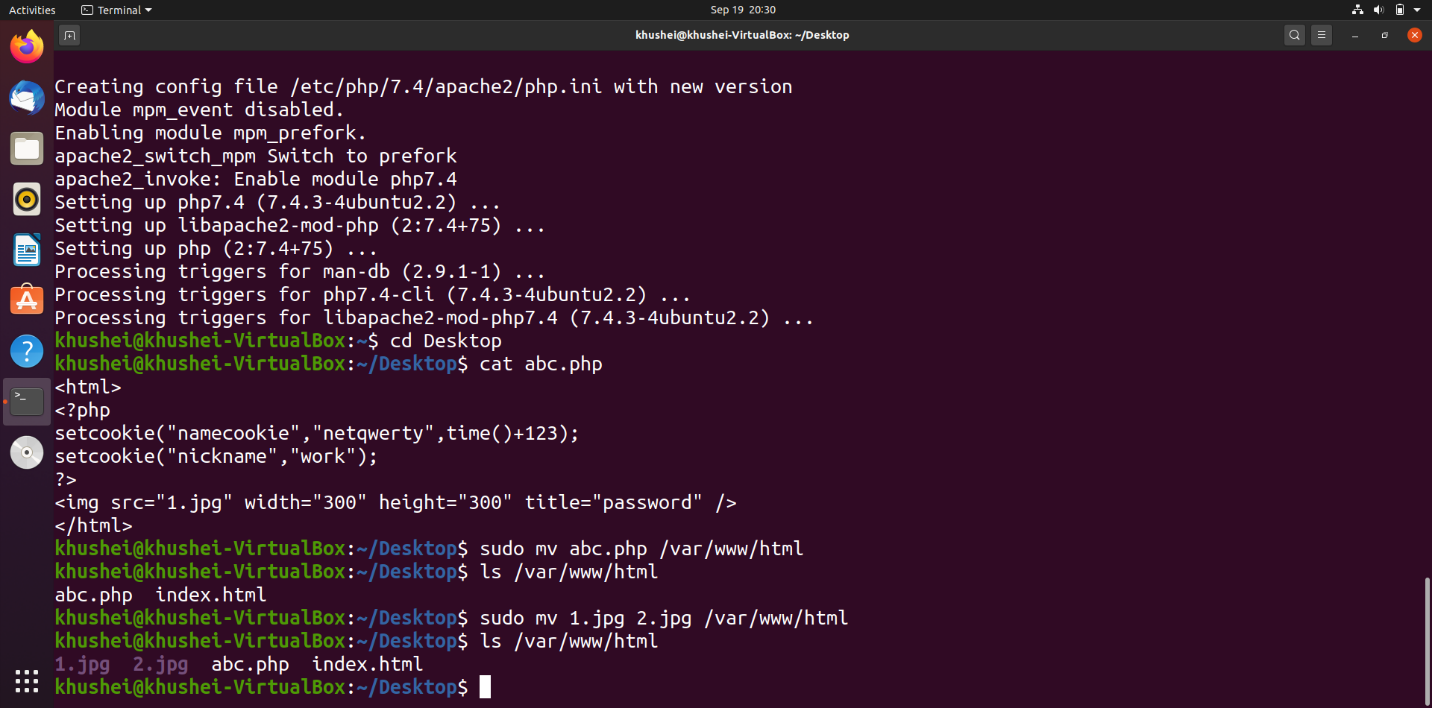


6. Using the “follow TCP stream” on the HTTP message segment the password was retrieved which was encrypted by the base64 algorithm and decryption could be done with same algorithm.

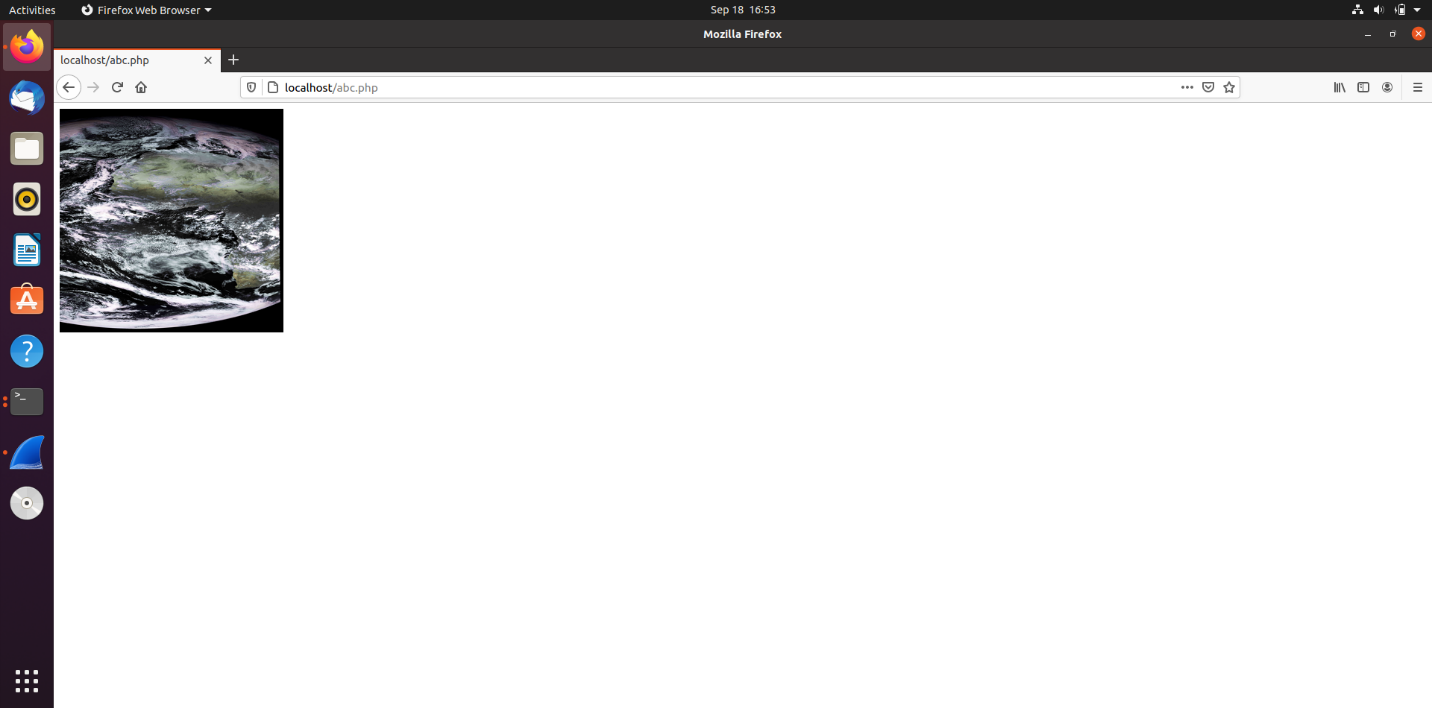


**Steps of Execution (Cookie Setting)**

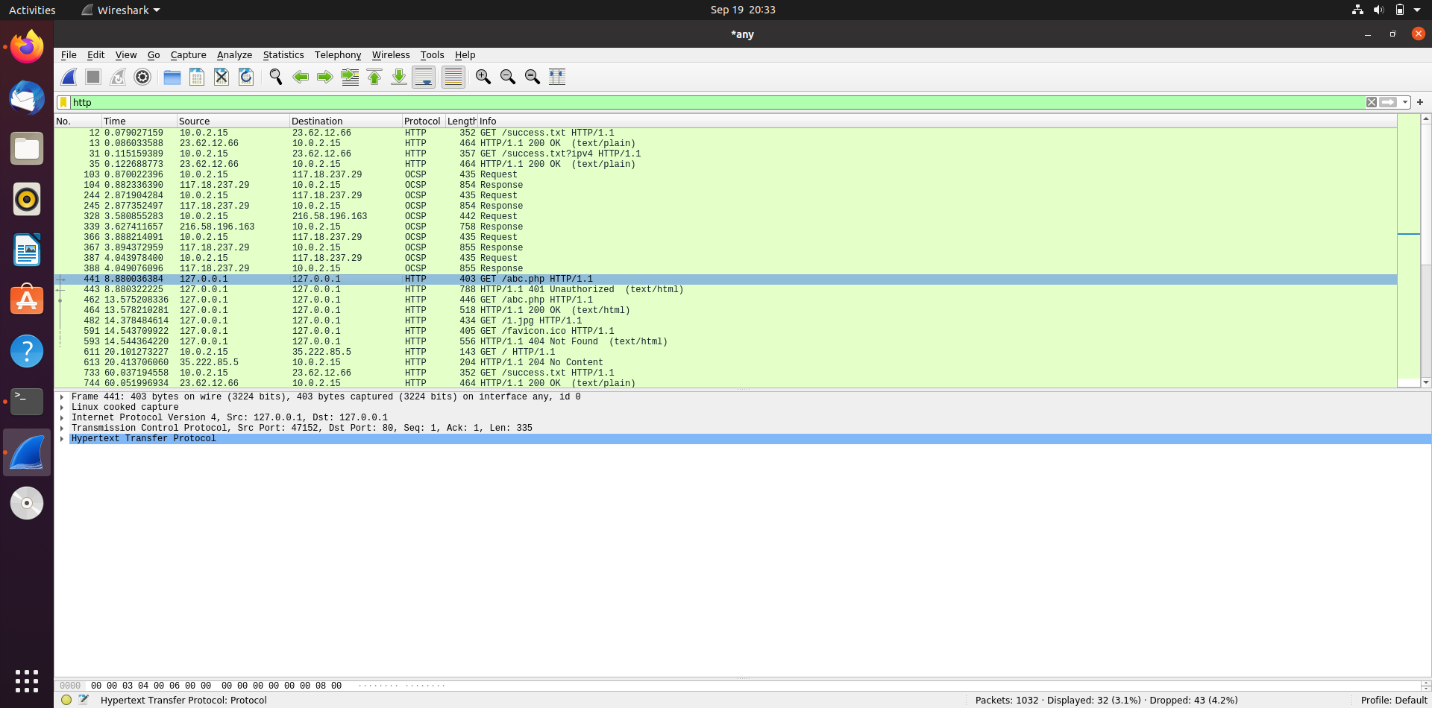
1. A PHP file to set the cookie is created which also contains an image in it (placed under the HTML directory) to be accessed once the cookie is set. The following code helped to set the cookie:



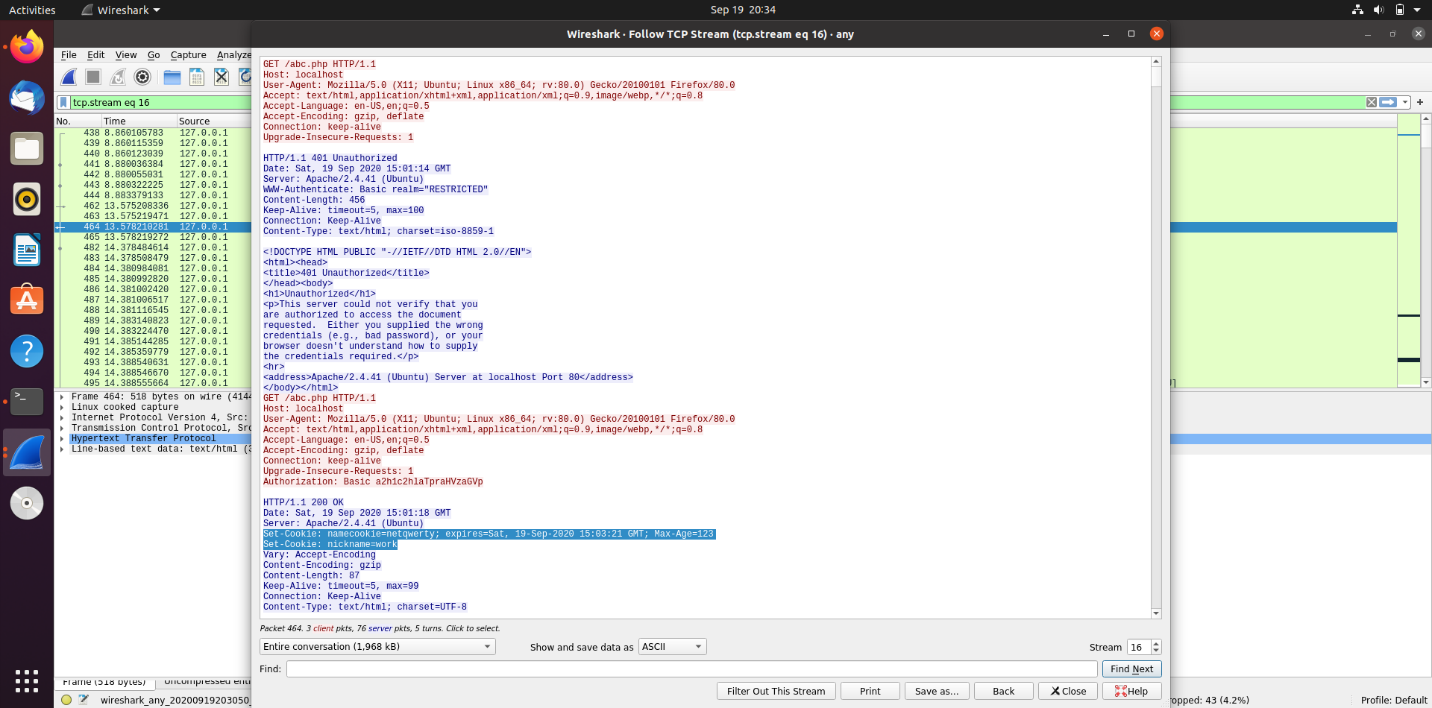
2. The combined file saved with a .php extension is placed under **/var/www/html** for accessing.



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.



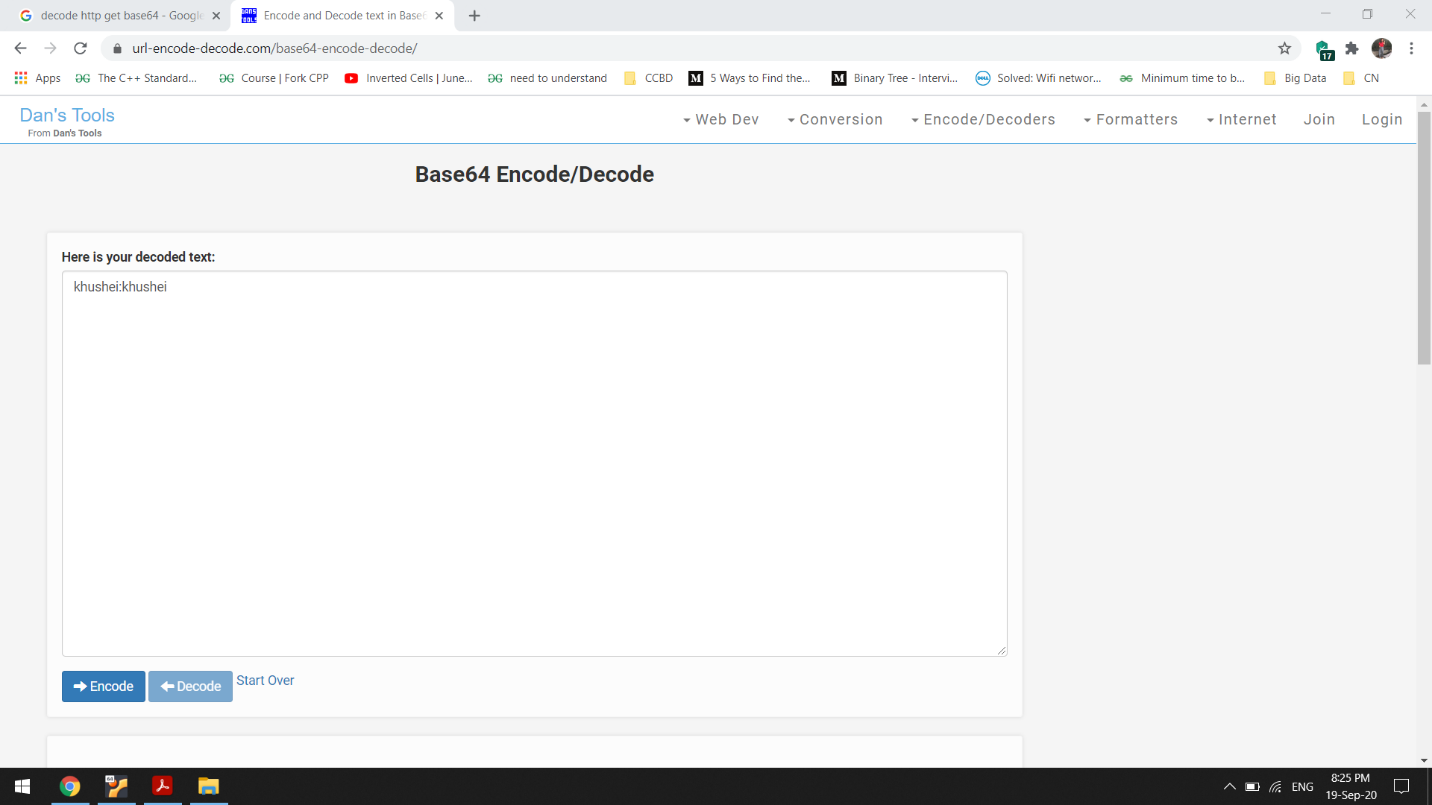
The cookie is set as shown in the below screenshot.



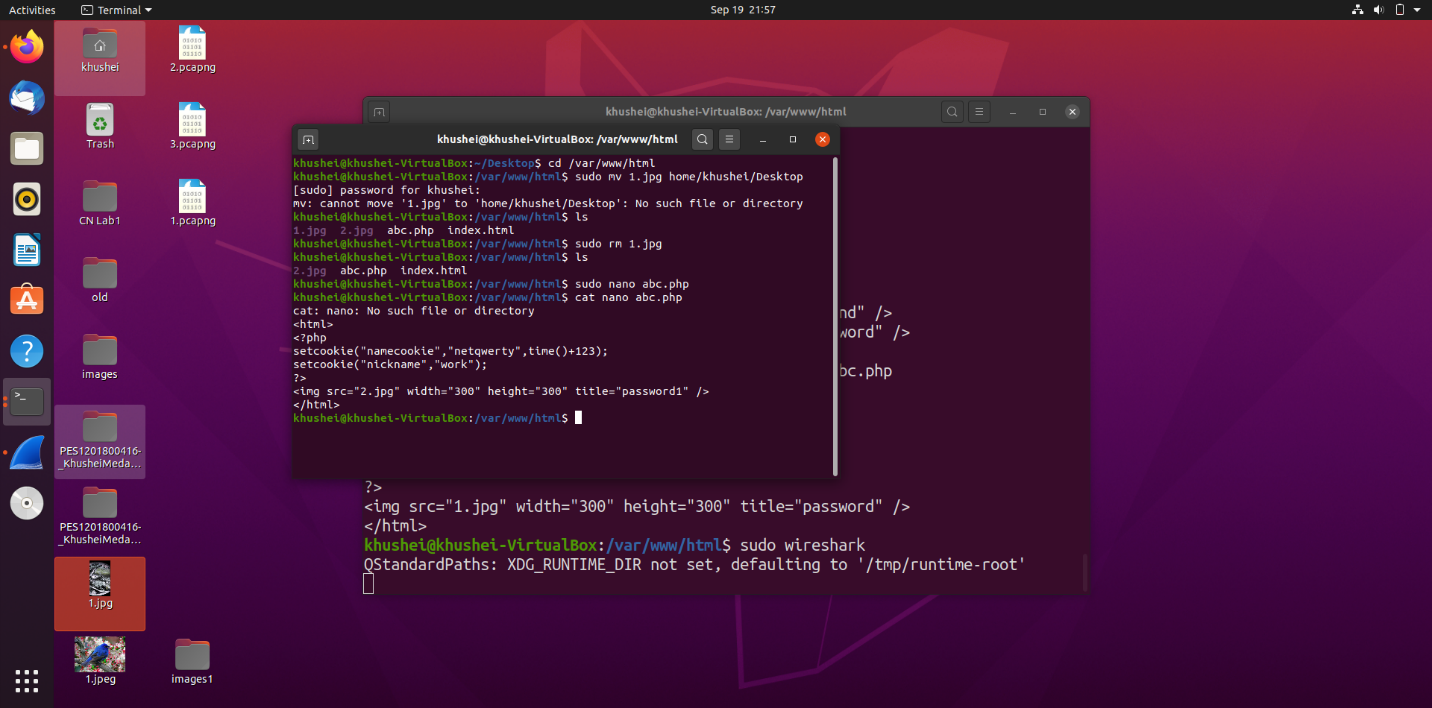
**Observation**: Understand and work out base 64 algorithm and write in your observation. Observe various parameters associated with Cookie in the wireshark capture.

In the above screenshot notice Authorization: Basic a2h1c2hlaTpraHVzaGVp

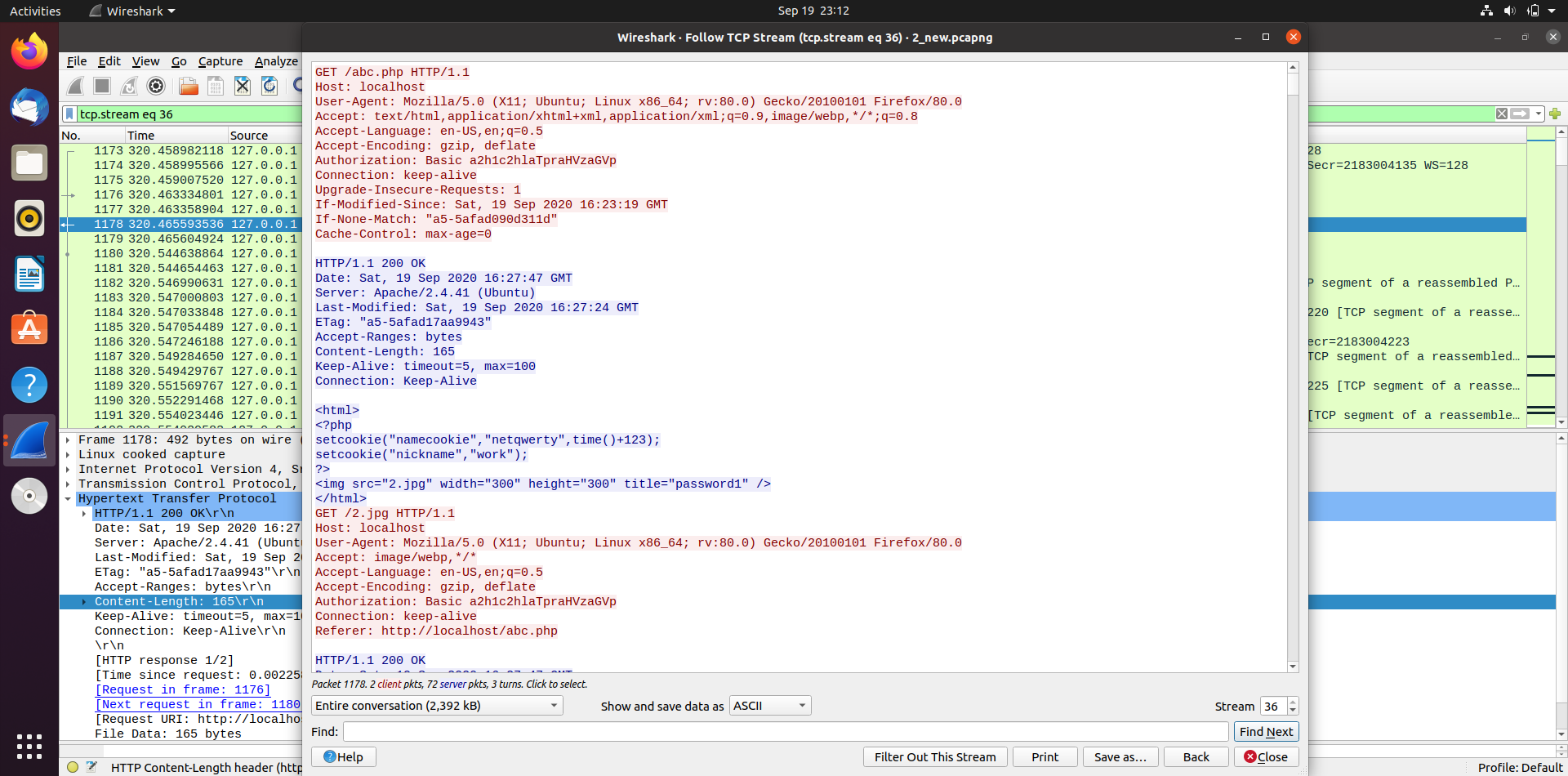
We decode this encoded text using an online decoder. We get the correct authentication id and password that we had set earlier, i.e, khushei:khushei



We now try to modify abc.php by adding another image to see if the Modified since parameter in the HTTP GET request changes. While Wireshark is capturing the packets, we open a new terminal and rename the image to 2.jpg. Note that we haven’t stopped Wireshark when we do this, so it continues to capture the packets.



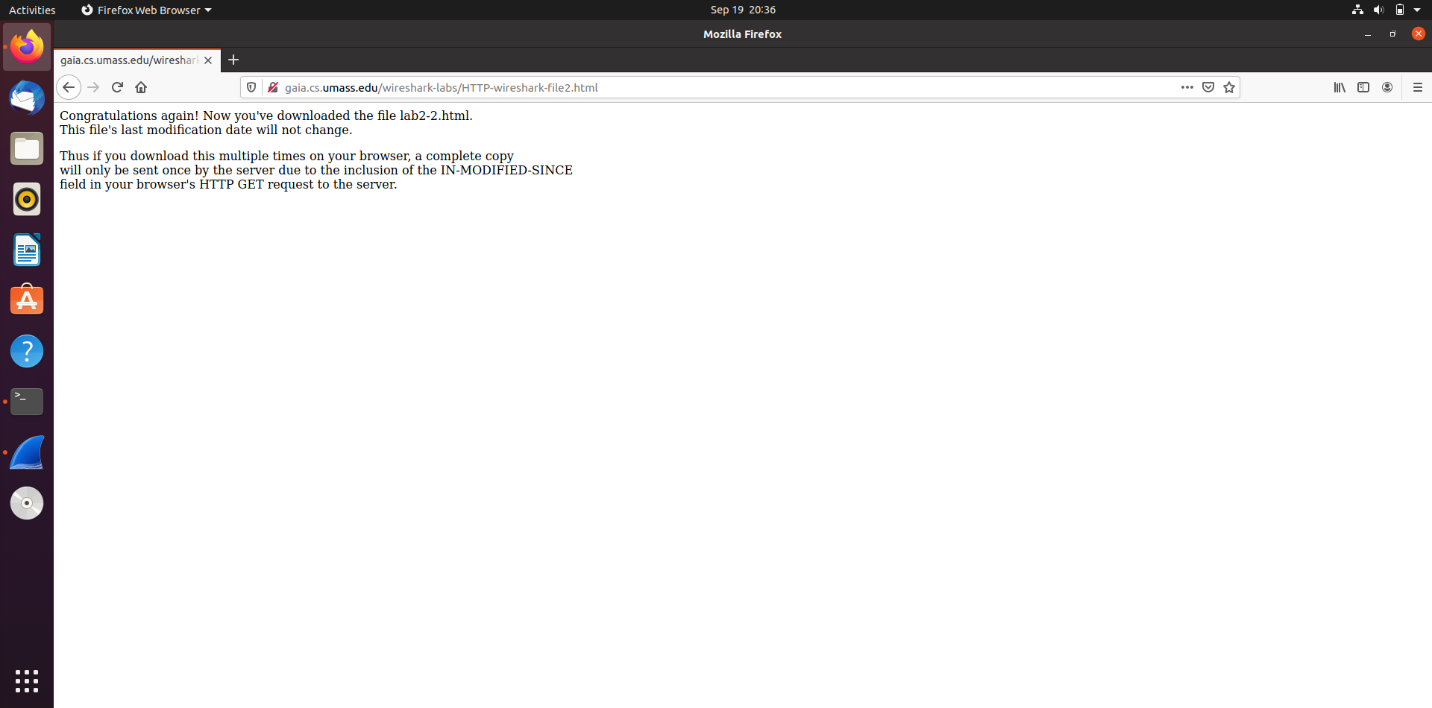
Notice that on refreshing the page and then following TCP stream of the HTTP 200 OK message after this change, we see an if-modified-since parameter and notice that since there has been a change, the image is sent by the server.





**Conditional Get: If-Modified-Since**

Enter the following URL into your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

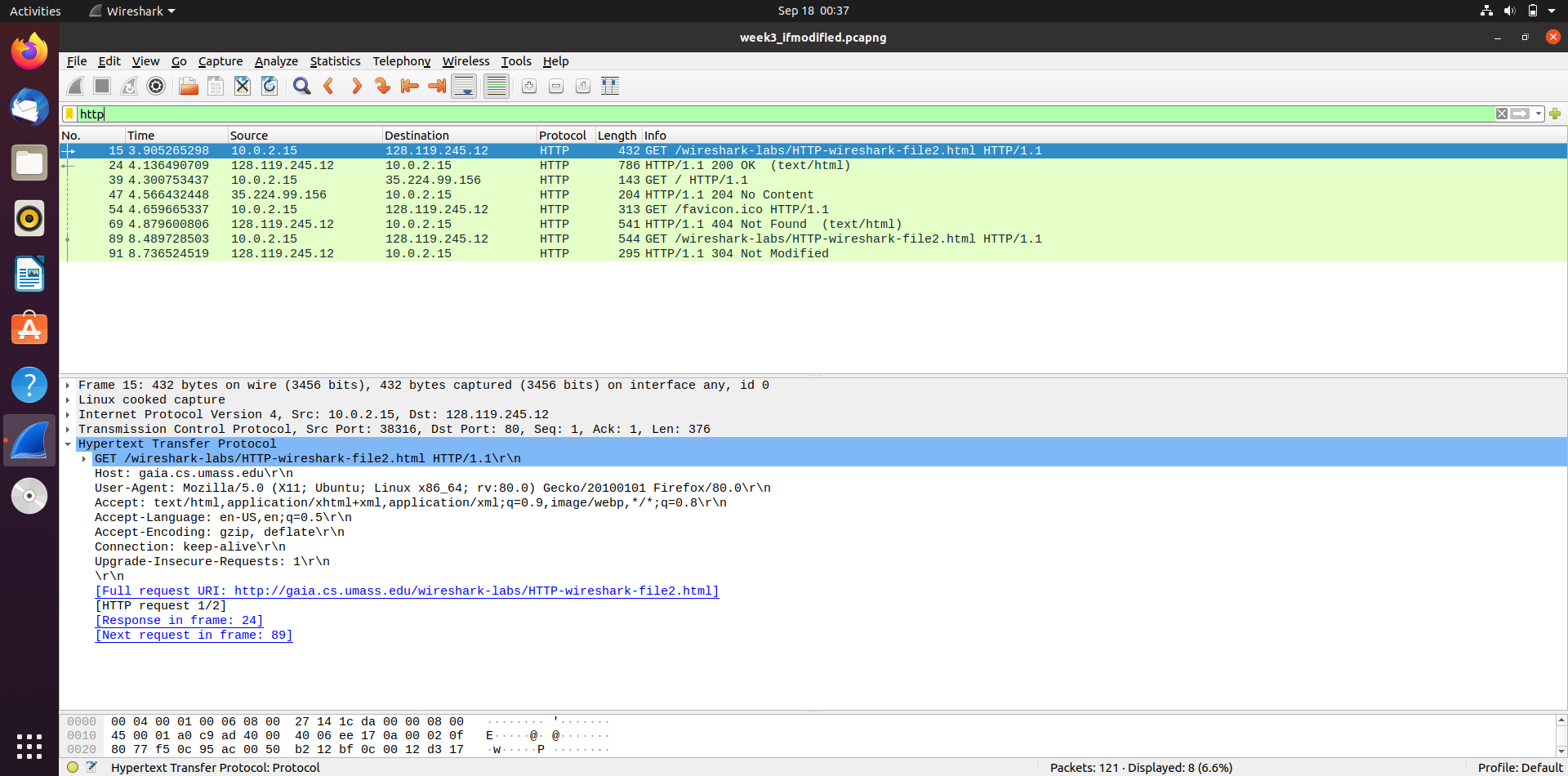


**Observations:**

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?

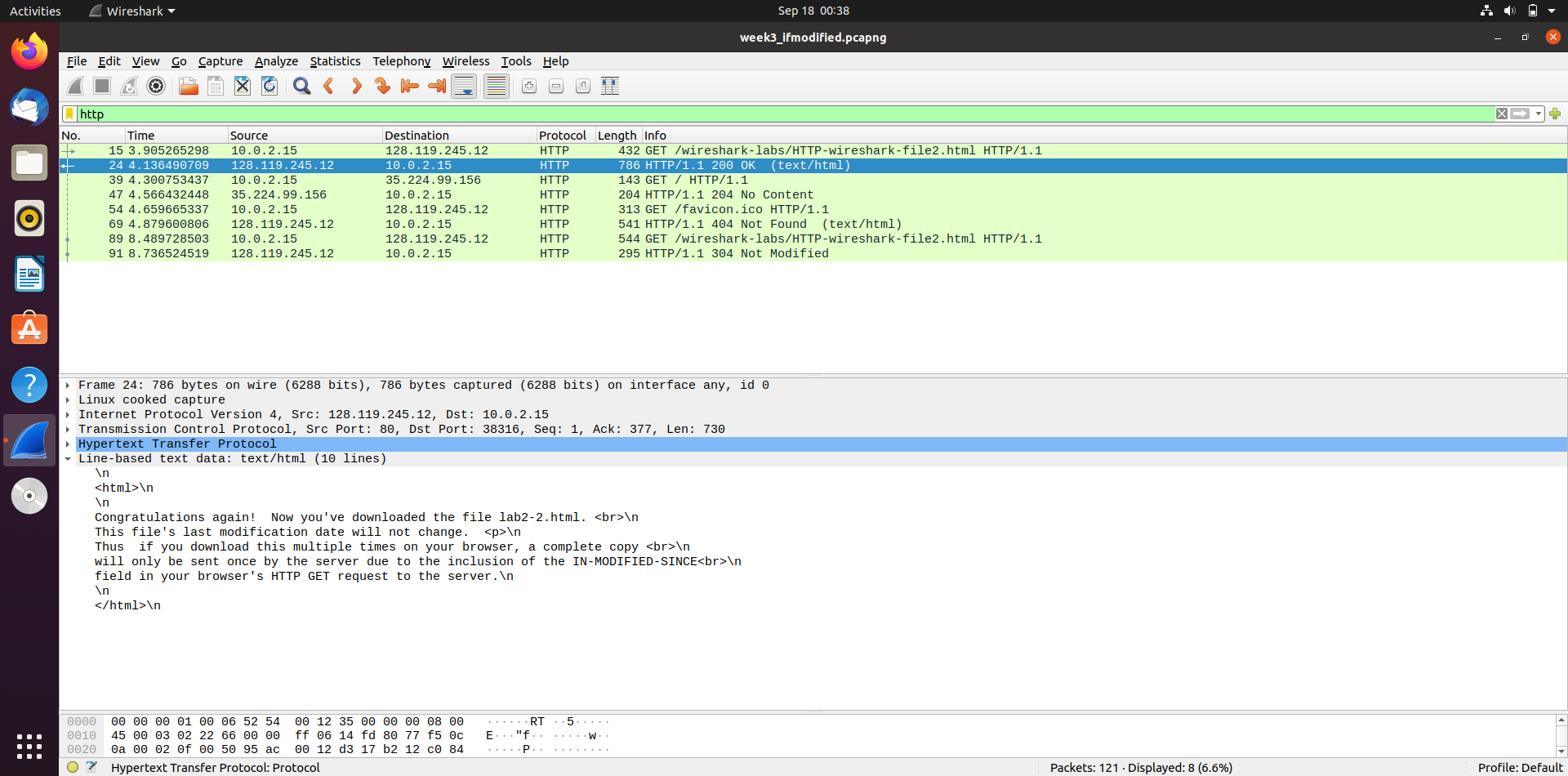
No, there is no “IF-MODIFIED-SINCE” line in the HTTP GET in the first GET request. Obviously, this is understandable because this is the first time that we are accessing this page of the website.

Notice that under the HTTP GET details, there is no line about last modification.



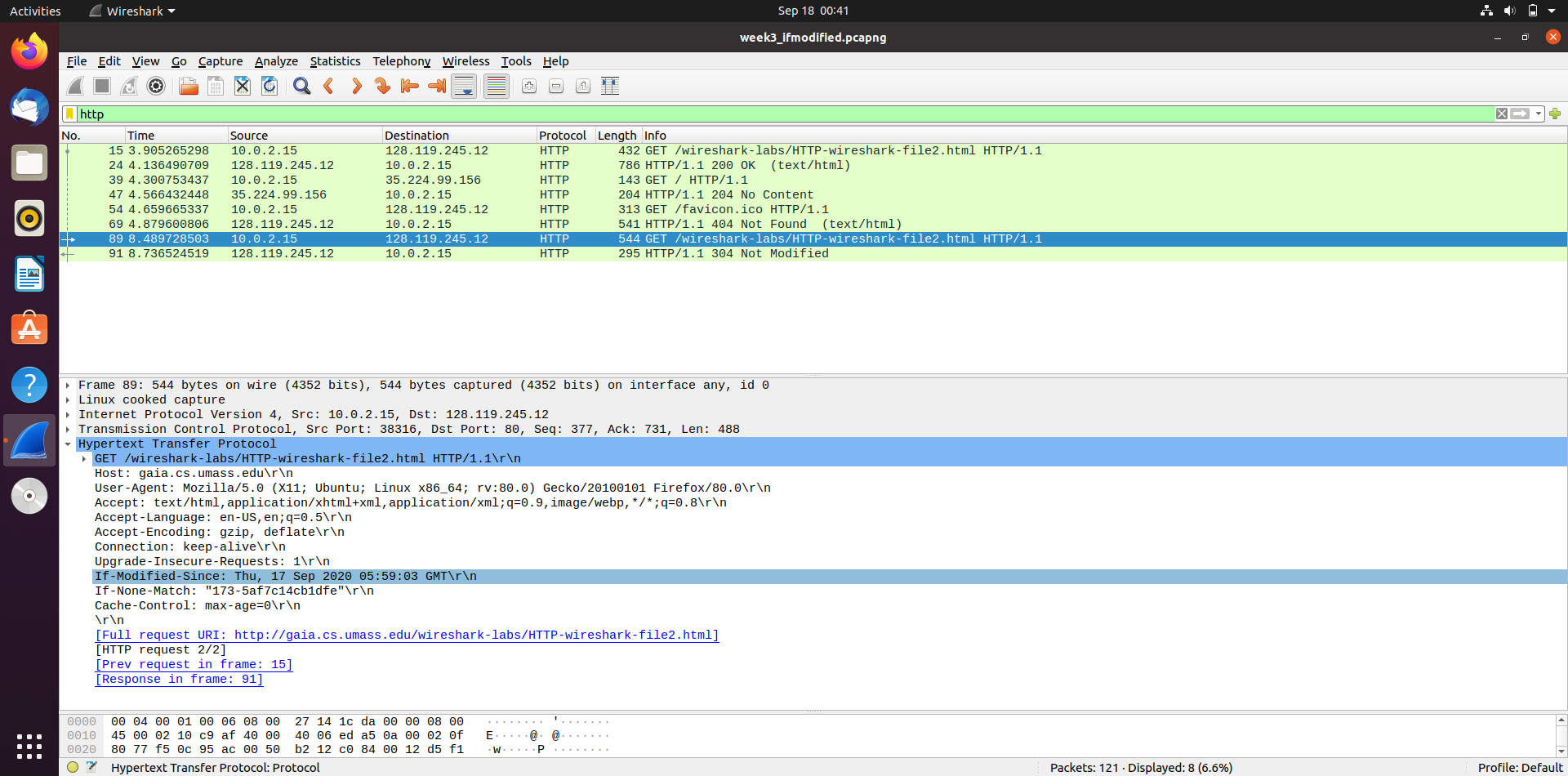
2. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

Yes, the server did explicitly return the contents of the file. We can see under Line-Based Text Data the text that the server sent back to the client browser.



3. 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?

In the second HTTP request, we do see the “IF-MODIFIED-SINCE:” line. The information that follows it is the date and timestamp of when my browser last accessed the webpage.



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.

Status code: 304.

Response Phrase: Not Modified

The server didn’t return the contents of the file since the browser loaded it from its cache.

