PRACTICAL 5 A

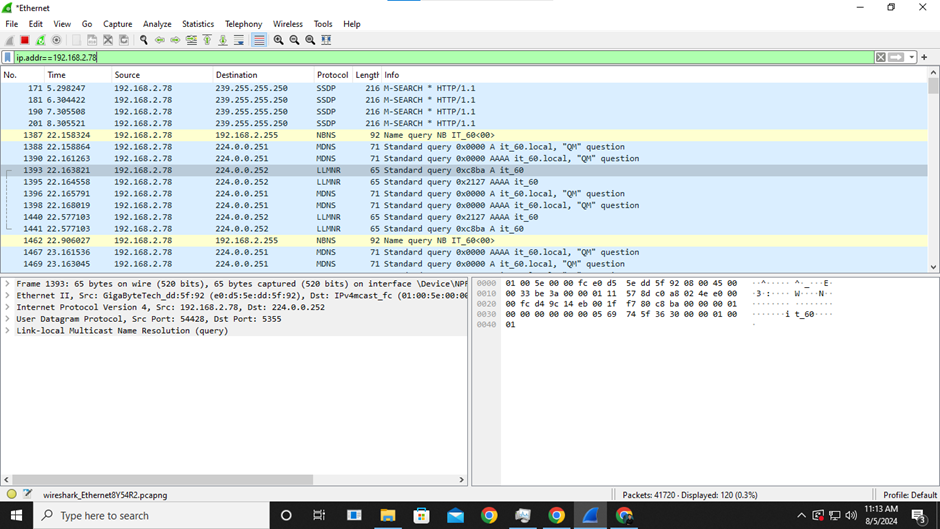
Wireshark(Sniffer)

**Aim :**  Use Wireshark (Sniffer) to capture network traffic and analyze and thus check vulnerability of network systems.

1. Inspect traffic for a specific IP address.

**Filter :** ip.addr==192.168.2.78

(Check the IP address of machine using ipconfig command in cmd line)



**Comments :**

~ The first image shows all the packets and network traffic being captured.

~ Second image shows the terminal.

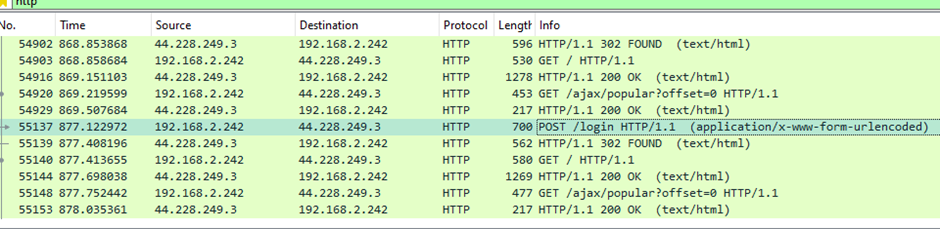
~ Third image shows the Frame 1’s (transmission no. 1) all the details like – source, destination, arrival time, epoch time. Similarly if any other transmission is selected (Ex : Frame 15) then the details regarding that frame/transmission are displayed.

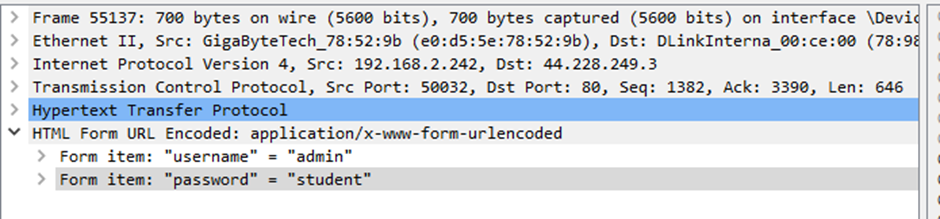
~ Last image shows the output in third window i.e. some hexadecimal digits marked in blue. These are the actual raw data carried by the frame from the source to the destination but in hexadecimal format.

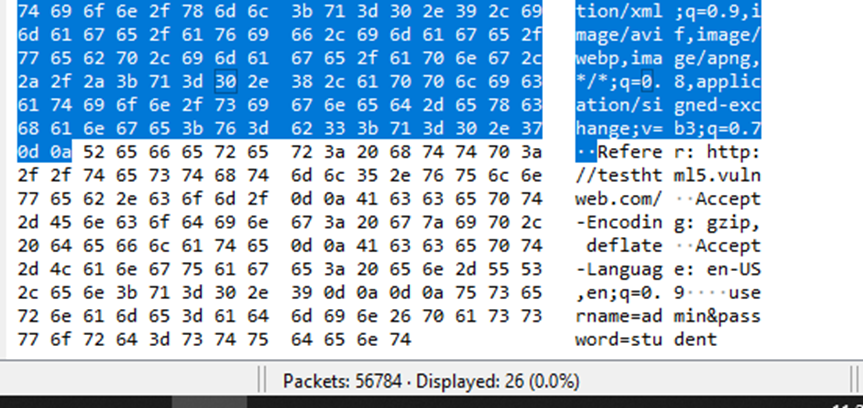
1. Sniff the credentials details from a vulnerable website:security tweets.com.

**Filter :** Keep scanning first. Goto website and then stop scanning and put filter “http”. Click the blue arrow at the side.

**Output :**

****

****

****

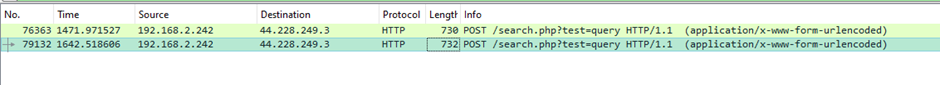
**Comments :** The new area in console Hypertext Transfer Protocol has appeared. It shows the host i.e. the website we logged in into, the user agent i.e. the browser that it works on and other details. While logging in, get method was used. That too is seen.

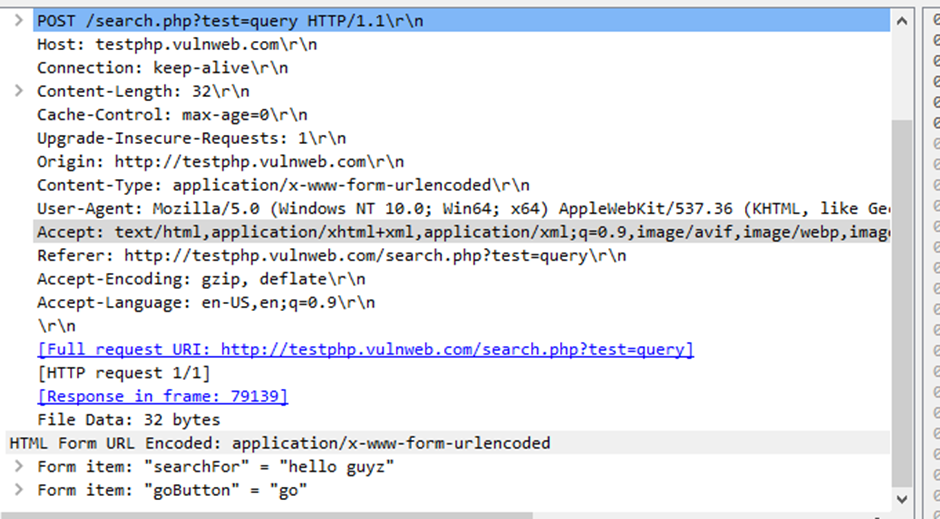
1. Check what user has typed in browser

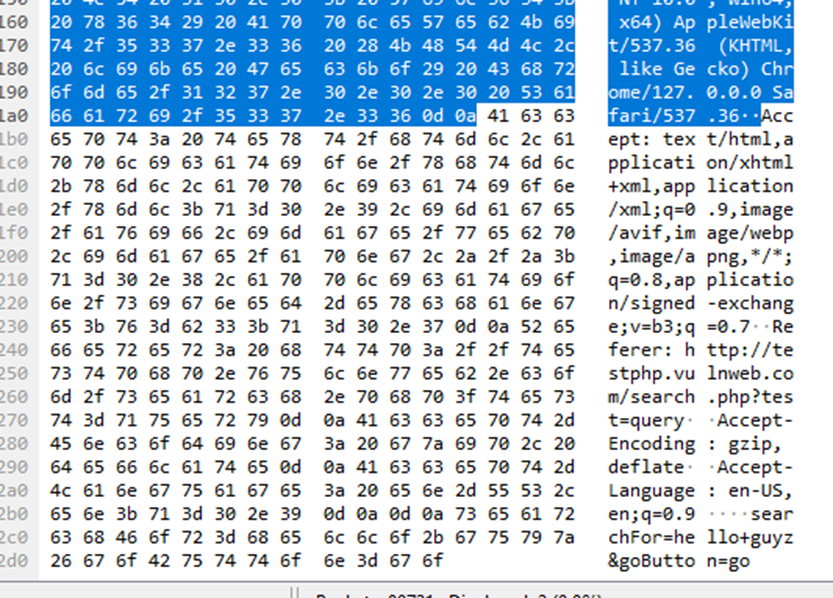
-http://test.vulnweb.com/search.php/

**Filter :** http.referer contains "http://test.vulnweb.com/search.php/"

**Output :**

****

****

****

**Comments :** The details for the search appears in the console

1. Check what string user has passed in google search engine.

**Filter :** http.referer contains “search”

(here I used directly http as filter and then checked the console to get the details…see image 2 in output)

(First scan, then search on browser with word search in URL, then stop scanning and add filter)

**Output :**

**Comments :** Details regarding the search wherein “search” word was used during query put into browser, appears on wireshark. Sniffing is successful.

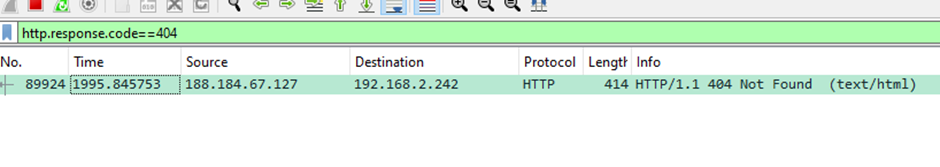
1. Find out packets which are getting the response as PAGE

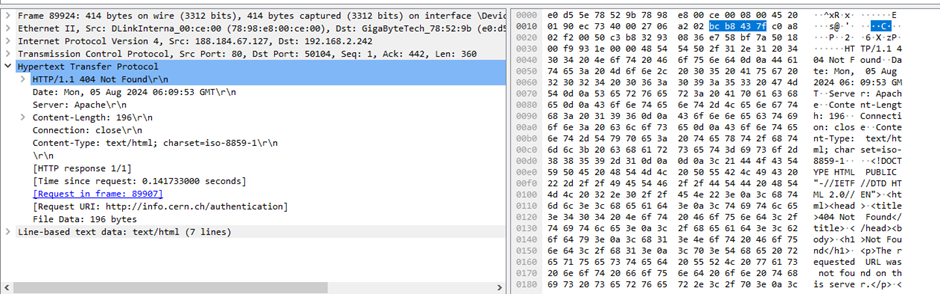
NOT FOUND.

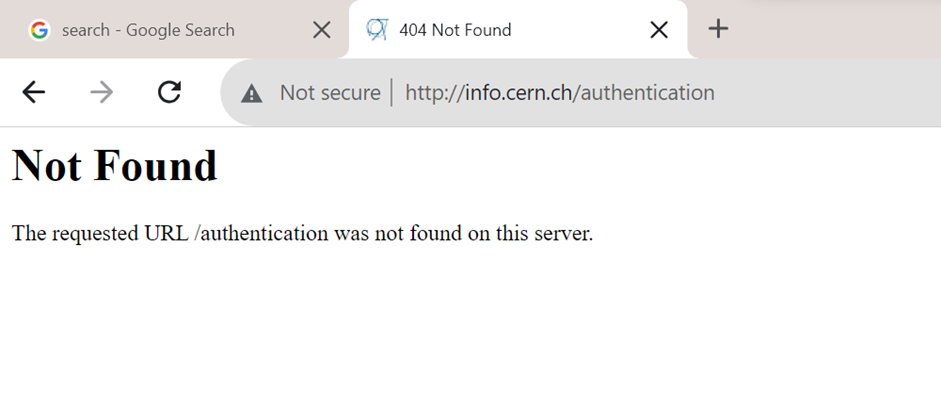
**URL used :** http://info.cern.ch/authentication

**Filter :** http.response.code==404

**Output :**







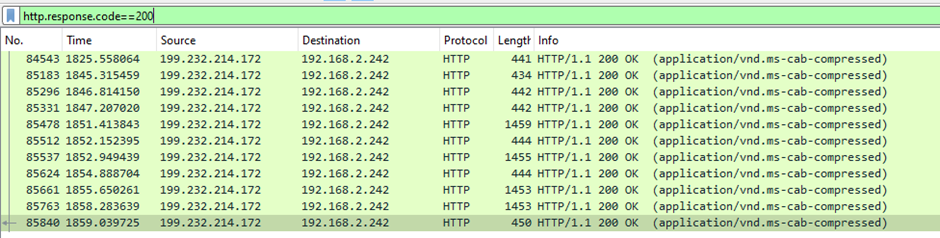
**Comments :**

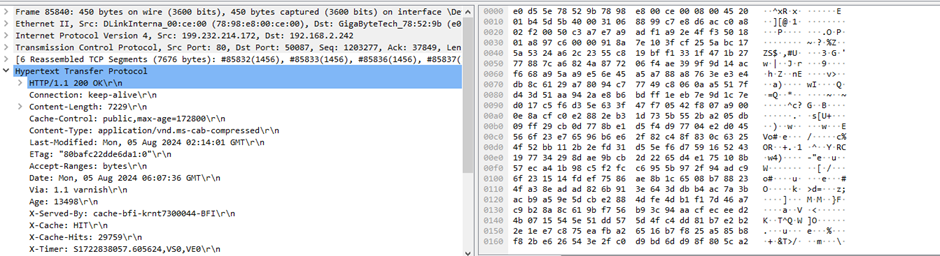
404 code is for page not found. So thus when you search for the page, the browser throws 404 not found error. Wireshark sniffs this as well and when we put the above filter details regarding the search appears.

1. Find out those packets which are getting response.

**Filter :** http.response.code==200

**Output :**

****

****

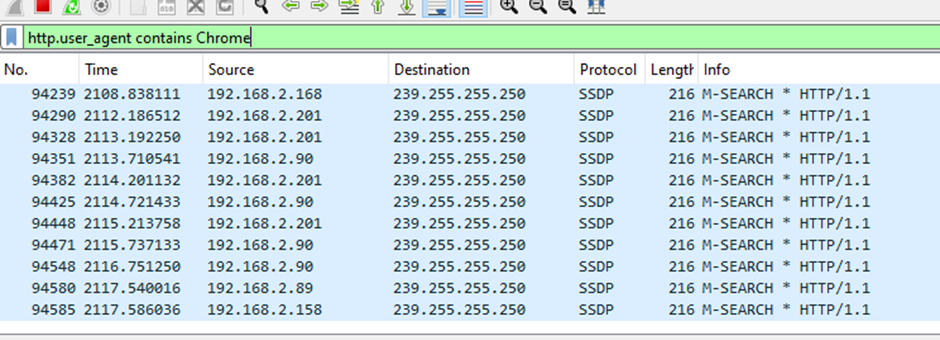
**Comments :**

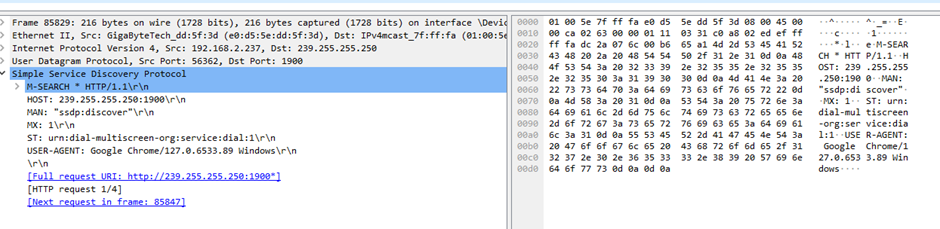
200 code is for page found/all ok. So thus when you search for the page, the browser returns status of page as 200 i.e. found the searched page. Wireshark sniffs this as well and when we put the above filter details regarding the search appears.

1. Find Packets-: Browser Specific.

**Filter :** http.user\_agent contains Chrome

**Output :**





**Browser search :**

**Codes for server.py and client.py :**

**Server.py :**

import socket

s=socket.socket()

print("Socket created successfully !")

port=8080

s.bind(('192.168.2.78',port))

print("Socket binded to %s"%(port))

s.listen(5)

print("Socket is listening...")

while True:

c,addr=s.accept()

print("Got connection from ",addr)

c.sendall('Thank you for connectiong! '.encode())

c.close()

**Client.py :**

import socket

s=socket.socket()

port=8080

s.connect(('192.168.2.78',port))

print(s.recv(1024).decode())

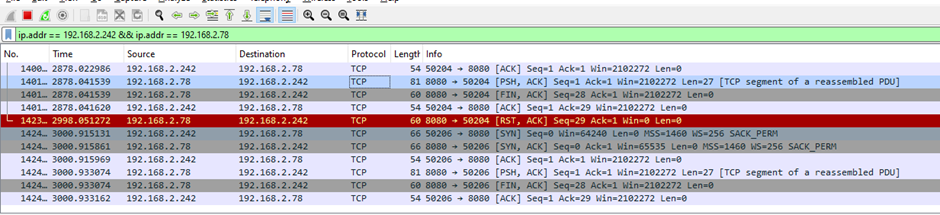
**Comments :**

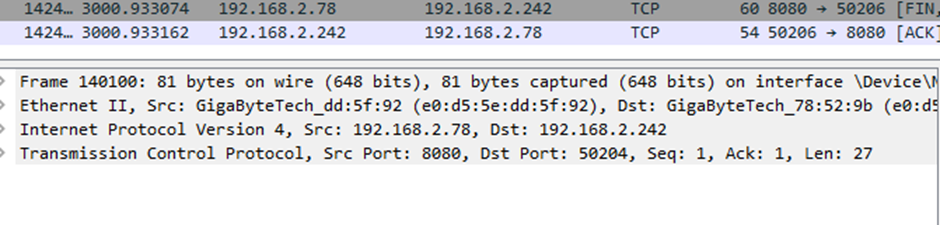
Above filter finds browser specific packets. The request made to browser and the browser’s response both are sniffed and captured by wireshark. In the above case it is seen that the request is captured in Frame (transmission no.) 546 and its response is given in frame no. 548. So these details are also given for easy understanding.

1. Inspect traffic between specific source and destination

**Filter :** ip.aadr==192.168.2.78 && ip.addr==192.168.2.242

**Output :**





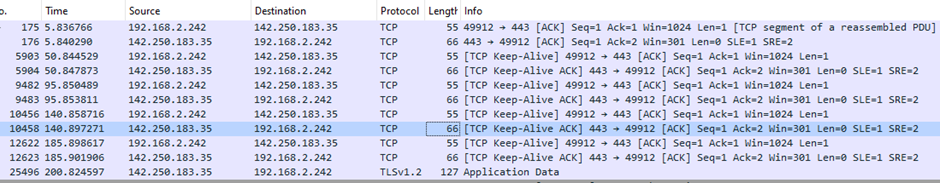
**Comments :**

The traffic i.e. data packets being sent between the two IPs is sniffed.

1. To sniff communication between client and server

**Filter :**

**Output :**

****

**Comments :**

If we see the wireshark output, the 7th frame shows ACK flag data packet being sent from server to client (443 to 23303) and then in 15th frame the other way round. Communication then happens. This is sniffed using wireshark. In Internet Protocol the source and destination also changes accordingly.