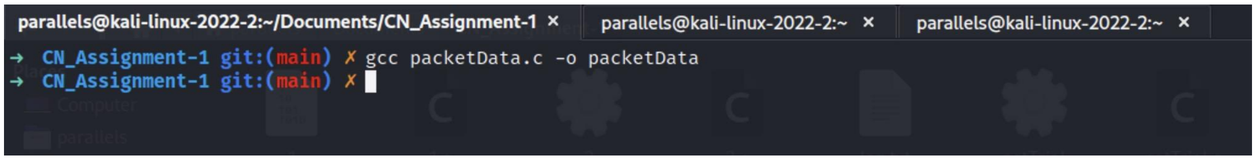
CTF:

Pcap file: 1.pcap

Step1 : Compile the packetData.c program.

Command: `gcc <path to packetData.c> -o <name of output file>`

Here, we have directly referred to packetData.c as it is present in the current directory itself.



parallels@kali-linux-2022-2:~/Documents/CN_Assignment-1
parallels@kali-linux-2022-2:~
parallels@kali-linux-2022-2:~

```
➜  CN_Assignment-1 git:(main) ✗ gcc packetData.c -o packetData
➜  CN_Assignment-1 git:(main) ✗
```

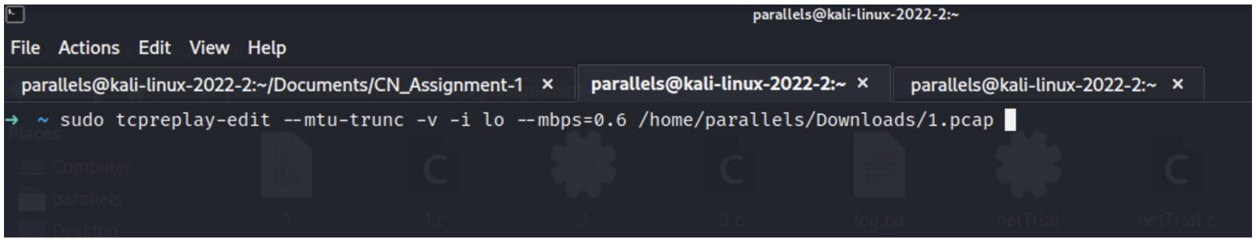
The packetData program is an extension of the program from Question 1 where the packet data (or payload) is written to a new file 'output.txt' using fopen to open the file and fprintf to write to the file.

Step 2: Parallelly open another terminal window to run the tcpreplay to emulate the packet transfers.

```
[For x86] sudo tcpreplay -v -i lo --mbps=1 <path_to_pcap_file>
[For Arm based Mac] sudo tcpreplay-edit –mtu-trunc -v -i lo --mbps=1 <path_to_pcap_file>
```

The -v flag is optional, its just to make it verbose, and here we are choosing lo as the network interface.

Lo represents the loopback interface which is a virtual interface unlike other interfaces.



parallels@kali-linux-2022-2:~
File Actions Edit View Help
parallels@kali-linux-2022-2:~/Documents/CN_Assignment-1
parallels@kali-linux-2022-2:~
parallels@kali-linux-2022-2:~

```
➜  ~ sudo tcpreplay-edit --mtu-trunc -v -i lo --mbps=0.6 /home/parallels/Downloads/1.pcap
```

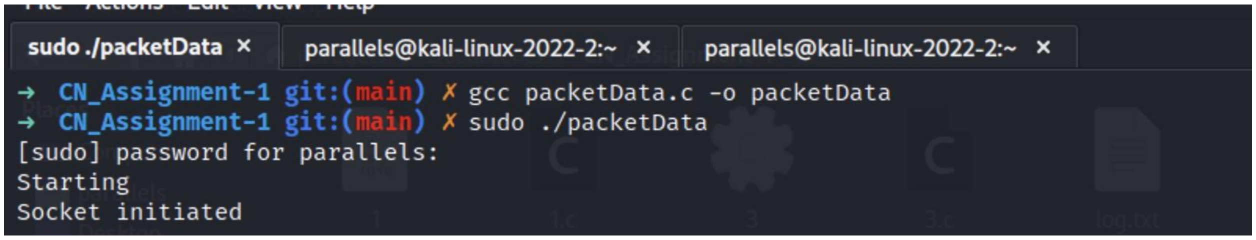
Illustration of command kali linux running on parallels[Mac M1]

Step 3: Keep the command to run the compiled packetData.c file read. Also, keep both commands ready to execute in different windows. Also run the compiled file on sudo.
Command: `sudo  <path to output file after compilation>`

Sudo here is important as opening a raw socket requires super user permissions.

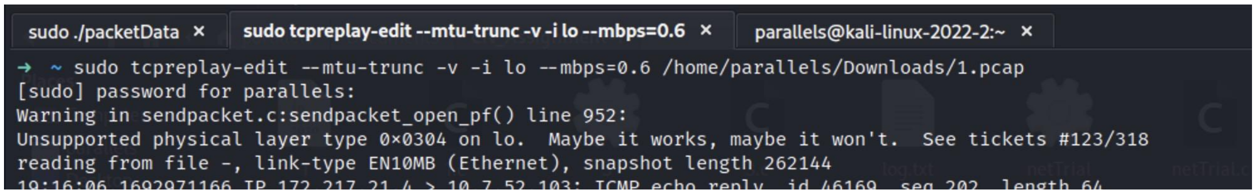
Step 4: Disconnect the ethernet/wifi network as the packetData program will sniff all the packets passing through. Otherwise, the output file would contain excess packets apart from the ones in the pcap file.

Step 4: [Very Imp]Run the packetData executable file first to start capturing the packets and then run the tcpreplay command.



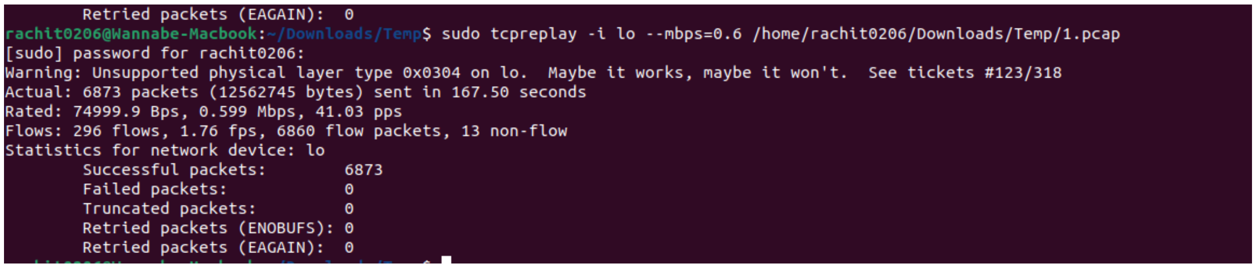
```
→ CN_Assignment-1 git:(main) ✗ gcc packetData.c -o packetData
→ CN_Assignment-1 git:(main) ✗ sudo ./packetData
[sudo] password for parallels:
Starting
Socket initiated
```

Illustration after running packetData file.


sudo ./packetData
sudo tcpreplay-edit --mtu-trunc -v -i lo --mbps=0.6
parallels@kali-linux-2022-2:~

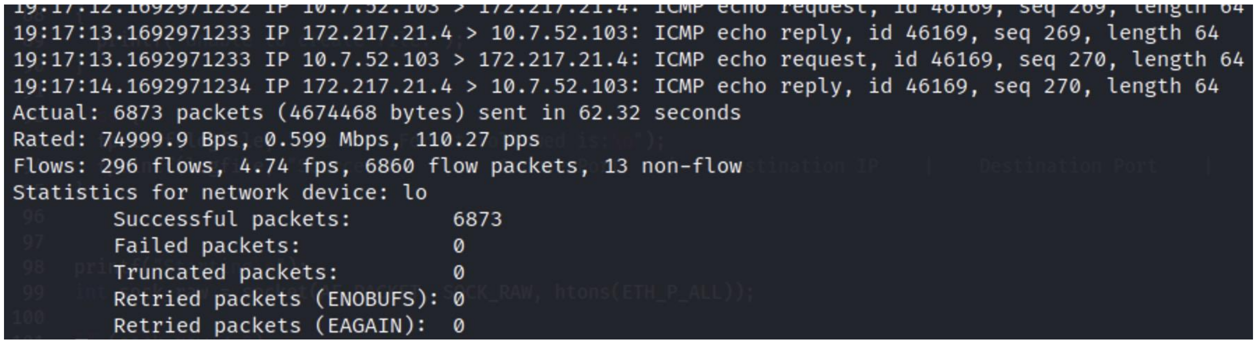
```
→ ~ sudo tcpreplay-edit --mtu-trunc -v -i lo --mbps=0.6 /home/parallels/Downloads/1.pcap
[sudo] password for parallels:
Warning in sendpacket.c:sendpacket_open_pf() line 952:
Unsupported physical layer type 0×0304 on lo.  Maybe it works, maybe it won't.  See tickets #123/318
reading from file -, link-type EN10MB (Ethernet), snapshot length 262144
19:16:06.1692971166 IP 172.217.21.4 > 10.7.52.103: ICMP echo reply, id 46169, seq 202, length 64
```

Illustration depicting the start of execution of tcp replay.



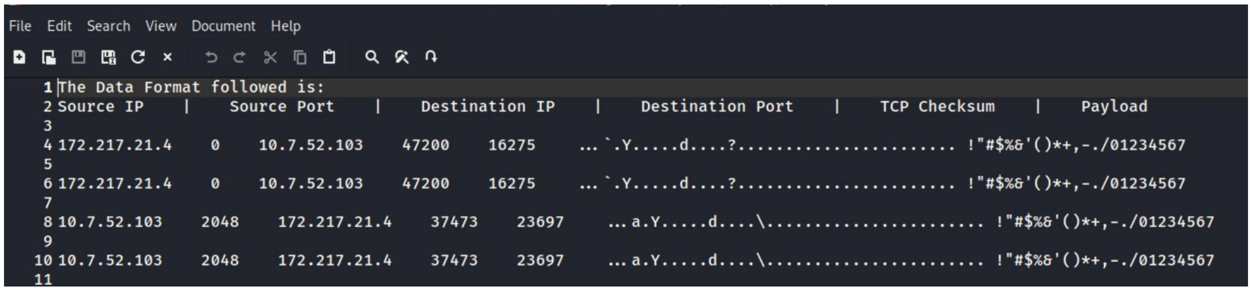
```
        Retried packets (EAGAIN):  0
rachit0206@Wannabe-Macbook:~/Downloads/Temp$ sudo tcpreplay -i lo --mbps=0.6 /home/rachit0206/Downloads/Temp/1.pcap
[sudo] password for rachit0206:
Warning: Unsupported physical layer type 0x0304 on lo.  Maybe it works, maybe it won't.  See tickets #123/318
Actual: 6873 packets (12562745 bytes) sent in 167.50 seconds
Rated: 74999.9 Bps, 0.599 Mbps, 41.03 pps
Flows: 296 flows, 1.76 fps, 6860 flow packets, 13 non-flow
Statistics for network device: lo
        Successful packets:        6873
        Failed packets:            0
        Truncated packets:         0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

Output screen after completion of TCP replay on x86.



```
19:17:12.1692971232 IP 10.7.52.103 > 172.217.21.4: ICMP echo request, id 46169, seq 269, length 64
19:17:13.1692971233 IP 172.217.21.4 > 10.7.52.103: ICMP echo reply, id 46169, seq 269, length 64
19:17:13.1692971233 IP 10.7.52.103 > 172.217.21.4: ICMP echo request, id 46169, seq 270, length 64
19:17:14.1692971234 IP 172.217.21.4 > 10.7.52.103: ICMP echo reply, id 46169, seq 270, length 64
Actual: 6873 packets (4674468 bytes) sent in 62.32 seconds
Rated: 74999.9 Bps, 0.599 Mbps, 110.27 pps
Flows: 296 flows, 4.74 fps, 6860 flow packets, 13 non-flow
Statistics for network device: lo
        Successful packets:        6873
        Failed packets:            0
        Truncated packets:         0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
```

Output screen after completion of TCP replay on M1 Mac.


```
The Data Format followed is:
Source IP    |    Source Port    |    Destination IP    |    Destination Port    |    TCP Checksum    |    Payload

172.217.21.4    0    10.7.52.103    47200    16275    ...`.Y.....d....?...................... !"#$%&'()*+,-./01234567

172.217.21.4    0    10.7.52.103    47200    16275    ...`.Y.....d....?...................... !"#$%&'()*+,-./01234567

10.7.52.103    2048    172.217.21.4    37473    23697    ...a.Y.....d....\...................... !"#$%&'()*+,-./01234567

10.7.52.103    2048    172.217.21.4    37473    23697    ...a.Y.....d....\...................... !"#$%&'()*+,-./01234567

```
The output.txt file after successful completion of both commands.

Step 5: After the completion of tcpreplay Run the following commands to obtain respective flags .

Question wise answers:

1. **Romeo**

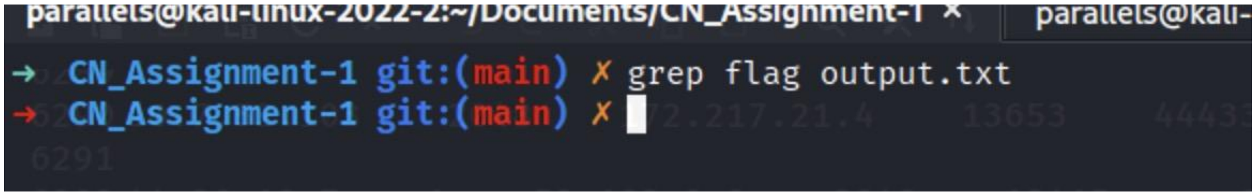
Main command:

```
grep -i flag output.txt | grep -v skip
[For generic] grep -i <path to output file> | grep -v skip
```

Explanation:

It was hinted that the payload or data might contain the flag keyword and we could search for the same using grep in the output.txt file.

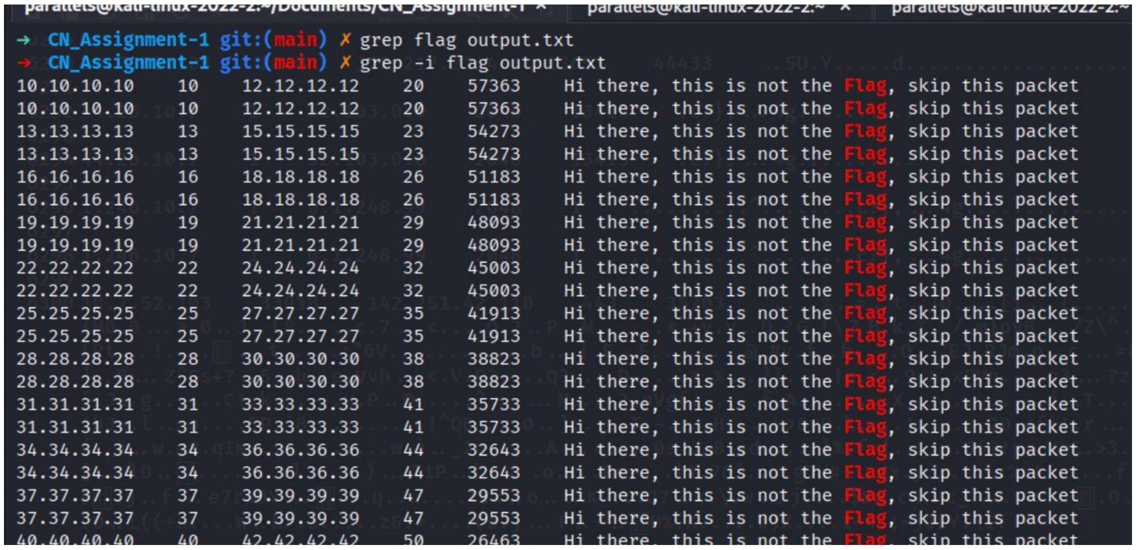
```
grep flag output.txt
[For generic] grep flag <path to output file>
```



```
parallels@kali-linux-2022-2:~/Documents/CN_Assignment-1
→ CN_Assignment-1 git:(main) ✗ grep flag output.txt
→ CN_Assignment-1 git:(main) ✗ 
```

Using this we get no output, so there's a chance that its case sensitive then we use the -i flag to make the search case insensitive.

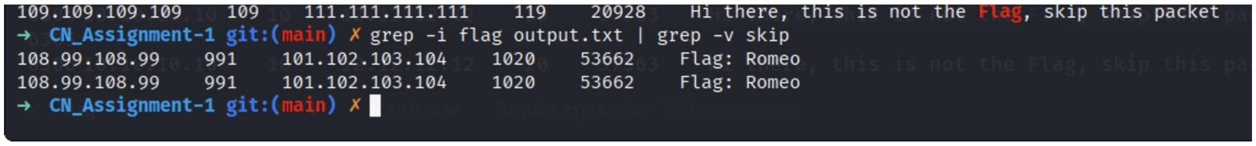
```
grep -i flag output.txt
[For generic] grep -i flag  <path to output file>
```



```
parallels@kali-linux-2022-2:~/Documents/CN_Assignment-1 ×   parallels@kali-linux-2022-2:~ ×   parallels@kali-linux-2022-2:~
→ CN_Assignment-1 git:(main) ✗ grep flag output.txt
→ CN_Assignment-1 git:(main) ✗ grep -i flag output.txt
10.10.10.10     10    12.12.12.12    20    57363    Hi there, this is not the Flag, skip this packet
10.10.10.10     10    12.12.12.12    20    57363    Hi there, this is not the Flag, skip this packet
13.13.13.13     13    15.15.15.15    23    54273    Hi there, this is not the Flag, skip this packet
13.13.13.13     13    15.15.15.15    23    54273    Hi there, this is not the Flag, skip this packet
16.16.16.16     16    18.18.18.18    26    51183    Hi there, this is not the Flag, skip this packet
16.16.16.16     16    18.18.18.18    26    51183    Hi there, this is not the Flag, skip this packet
19.19.19.19     19    21.21.21.21    29    48093    Hi there, this is not the Flag, skip this packet
19.19.19.19     19    21.21.21.21    29    48093    Hi there, this is not the Flag, skip this packet
22.22.22.22     22    24.24.24.24    32    45003    Hi there, this is not the Flag, skip this packet
22.22.22.22     22    24.24.24.24    32    45003    Hi there, this is not the Flag, skip this packet
25.25.25.25     25    27.27.27.27    35    41913    Hi there, this is not the Flag, skip this packet
25.25.25.25     25    27.27.27.27    35    41913    Hi there, this is not the Flag, skip this packet
28.28.28.28     28    30.30.30.30    38    38823    Hi there, this is not the Flag, skip this packet
28.28.28.28     28    30.30.30.30    38    38823    Hi there, this is not the Flag, skip this packet
31.31.31.31     31    33.33.33.33    41    35733    Hi there, this is not the Flag, skip this packet
31.31.31.31     31    33.33.33.33    41    35733    Hi there, this is not the Flag, skip this packet
34.34.34.34     34    36.36.36.36    44    32643    Hi there, this is not the Flag, skip this packet
34.34.34.34     34    36.36.36.36    44    32643    Hi there, this is not the Flag, skip this packet
37.37.37.37     37    39.39.39.39    47    29553    Hi there, this is not the Flag, skip this packet
37.37.37.37     37    39.39.39.39    47    29553    Hi there, this is not the Flag, skip this packet
40.40.40.40     40    42.42.42.42    50    26463    Hi there, this is not the Flag, skip this packet
```

Here, we see a lot of packets indicating to skip those packets. So, we could use -v to exclude the lines with a certain keyword. We could use any word(except for flag) in the recurring sentence. Here, we have chosen to exclude lines with 'skip'.

```
grep -i flag output.txt | grep -v skip
[For generic] grep -i <path to output file> | grep -v skip
```



```
109.109.109.109     109     111.111.111.111     119     20928     Hi there, this is not the Flag, skip this packet
→ CN_Assignment-1 git:(main) ✗ grep -i flag output.txt | grep -v skip
108.99.108.99      991     101.102.103.104     1020     53662     Flag: Romeo
108.99.108.99      991     101.102.103.104     1020     53662     Flag: Romeo
→ CN_Assignment-1 git:(main) ✗
```

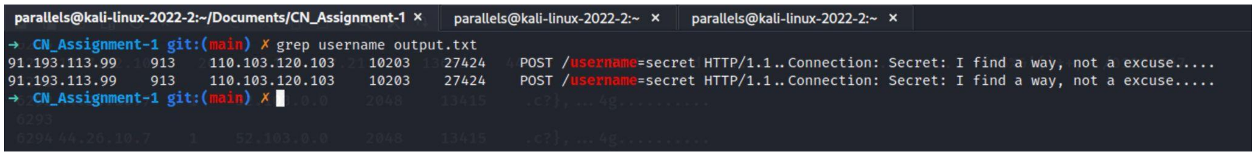
The final output depicting the flags.

2. **I find a way, not a excuse**

Main command:

```
grep username output.txt
[For generic] grep username <path to output file>
```

Here, it was mentioned that the username is secret. So, the most trivial idea would be to directly search for the data containing the 'username' keyword.



```
parallels@kali-linux-2022-2:~/Documents/CN_Assignment-1 ×   parallels@kali-linux-2022-2:~ ×   parallels@kali-linux-2022-2:~ ×
→ CN_Assignment-1 git:(main) ✗ grep username output.txt
91.193.113.99     913     110.103.120.103     10203     27424     POST /username=secret HTTP/1.1..Connection: Secret: I find a way, not a excuse.....
91.193.113.99     913     110.103.120.103     10203     27424     POST /username=secret HTTP/1.1..Connection: Secret: I find a way, not a excuse.....
→ CN_Assignment-1 git:(main) ✗
```

3. **Berlin**

Main command:

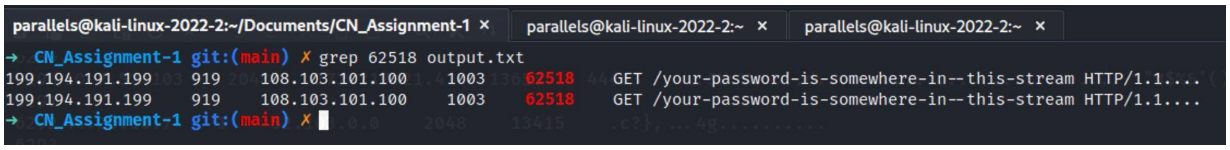
```
grep 199.194.191.199 output.txt | grep -i password
```

```
[For generic] grep 199.194.191.199 <path to output file> | grep -i password
```

Explanation:

So, we have also added TCP checksum(in decimal format) in the outout.txt file.
We need the packet with TCP 0xf436 converting it to decimal is : 62518

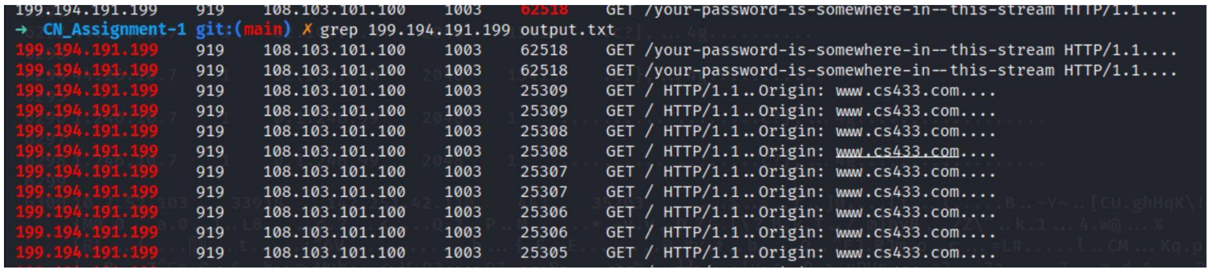
```
grep 62518 output.txt
[For generic] grep 62518 <path to output file>
```



```
parallels@kali-linux-2022-2:~/Documents/CN_Assignment-1 ×   parallels@kali-linux-2022-2:~ ×   parallels@kali-linux-2022-2:~ ×
→ CN_Assignment-1 git:(main) ✗ grep 62518 output.txt
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
→ CN_Assignment-1 git:(main) ✗
```

As mentioned that the password would be in the same stream then the packets will share the same 4 tuple defined by TCP flow.
So we can try to find a match for the source IP. (which is 199.194.191.199).

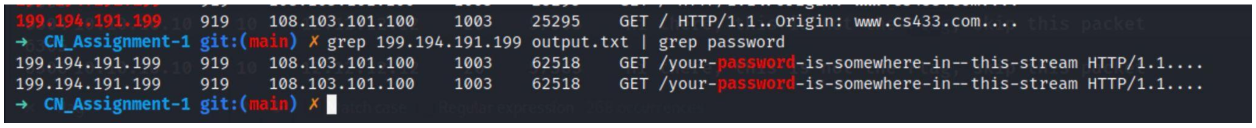
```
grep 199.194.191.199 output.txt
[For generic] grep 199.194.191.199 <path to output file>
```



```
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
→ CN_Assignment-1 git:(main) ✗ grep 199.194.191.199 output.txt
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
199.194.191.199    919    108.103.101.100    1003    25309    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25309    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25308    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25308    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25307    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25307    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25306    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25306    GET / HTTP/1.1..Origin: www.cs433.com....
199.194.191.199    919    108.103.101.100    1003    25305    GET / HTTP/1.1..Origin: www.cs433.com....
```

We still have a lot of packets from the same source IP. So we could search for the 'password' keyword from these packets.

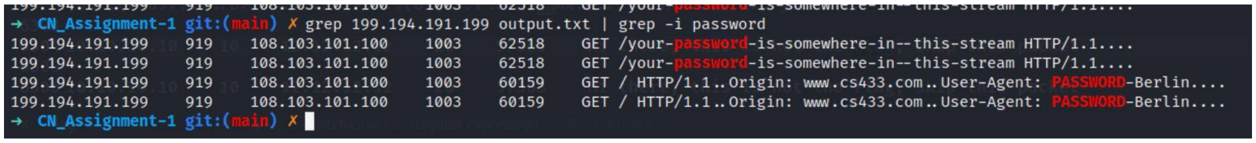
```
grep 199.194.191.199 output.txt | grep password
[For generic] grep 199.194.191.199 <path to output file> | grep password
```



```
199.194.191.199    919    108.103.101.100    1003    25295    GET / HTTP/1.1..Origin: www.cs433.com....
→ CN_Assignment-1 git:(main) ✗ grep 199.194.191.199 output.txt | grep password
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
→ CN_Assignment-1 git:(main) ✗
```

There could be a chance that it could be case sensitive.

```
grep 199.194.191.199 output.txt | grep -i password
[For generic] grep 199.194.191.199 <path to output file> | grep -i password
```



```
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
→ CN_Assignment-1 git:(main) ✗ grep 199.194.191.199 output.txt | grep -i password
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
199.194.191.199    919    108.103.101.100    1003    62518    GET /your-password-is-somewhere-in--this-stream HTTP/1.1....
199.194.191.199    919    108.103.101.100    1003    60159    GET / HTTP/1.1..Origin: www.cs433.com..User-Agent: PASSWORD-Berlin....
199.194.191.199    919    108.103.101.100    1003    60159    GET / HTTP/1.1..Origin: www.cs433.com..User-Agent: PASSWORD-Berlin....
→ CN_Assignment-1 git:(main) ✗
```

4. **Rabindranath Tagore**

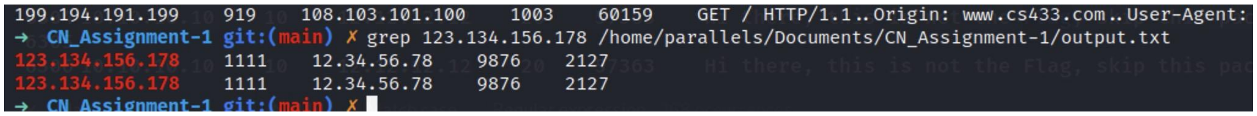
Main command:

```
grep 10987 output.txt
[For generic] grep 10987 <path to output file>
```

Explanation:

As mentioned we need to find the sum of ports based on the IP, we grep with respect to IP.

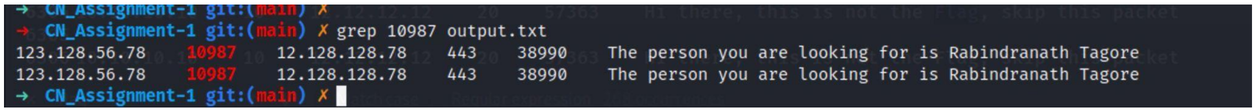
```
grep 123.134.156.178 /home/parallels/Documents/CN_Assignment-1/output.txt
[For generic] grep 123.134.156.178 <path to output file>
```



```
199.194.191.199    919    108.103.101.100    1003    60159    GET / HTTP/1.1..Origin: www.cs433.com..User-Agent:
→ CN_Assignment-1 git:(main) ✗ grep 123.134.156.178 /home/parallels/Documents/CN_Assignment-1/output.txt
123.134.156.178    1111    12.34.56.78    9876    2127
123.134.156.178    1111    12.34.56.78    9876    2127
→ CN_Assignment-1 git:(main) ✗
```

Sum of ports = 1111 + 9876 = 10987

```
grep 10987 output.txt
[For generic] grep 10987 <path to output file>
```



```
→ CN_Assignment-1 git:(main) ✗
→ CN_Assignment-1 git:(main) ✗ grep 10987 output.txt
123.128.56.78    10987    12.128.128.78    443    38990    The person you are looking for is Rabindranath Tagore
123.128.56.78    10987    12.128.128.78    443    38990    The person you are looking for is Rabindranath Tagore
→ CN_Assignment-1 git:(main) ✗
```

5. **Strawberry**

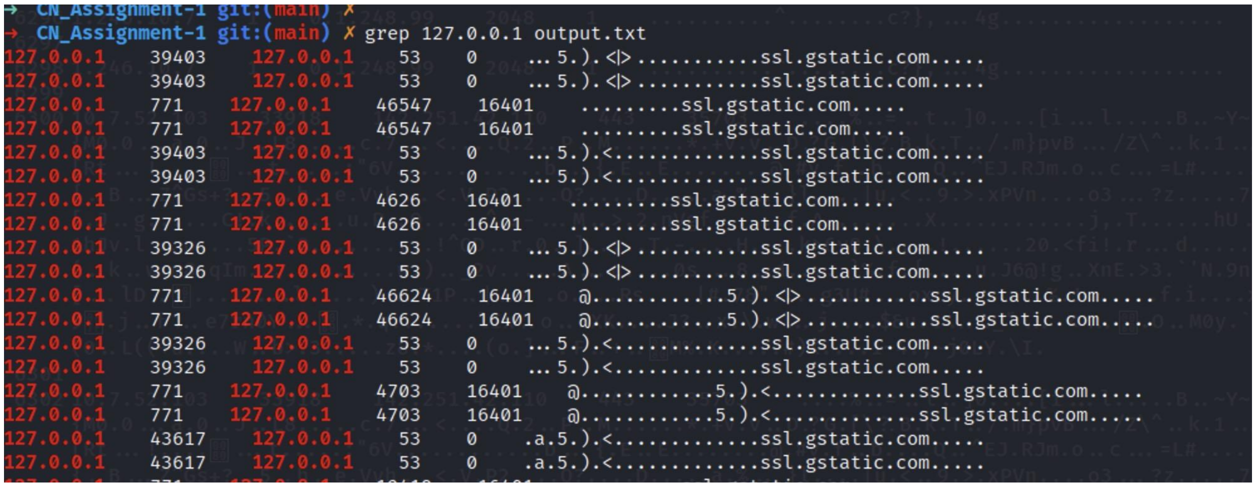
Main command:

```
grep 127.0.0.1 output.txt | grep milkshake
[For generic] grep 127.0.0.1 <path to output file> | grep milkshake
```

Explanation:

As localhost has an IP address of '127.0.0.1', we will use that to filter the packets.

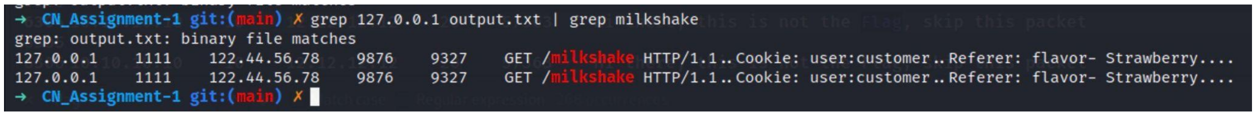
```
grep 127.0.0.1 output.txt
[For generic] grep 127.0.0.1 <path to output file>
```



```
→ CN_Assignment-1 git:(main) ✗
→ CN_Assignment-1 git:(main) ✗ grep 127.0.0.1 output.txt
127.0.0.1    39403    127.0.0.1    53    0    ...5.).<|>...........ssl.gstatic.com.....
127.0.0.1    39403    127.0.0.1    53    0    ...5.).<|>...........ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    46547    16401    .........ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    46547    16401    .........ssl.gstatic.com.....
127.0.0.1    39403    127.0.0.1    53    0    ...5.).<............ssl.gstatic.com.....
127.0.0.1    39403    127.0.0.1    53    0    ...5.).<............ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    4626    16401    .........ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    4626    16401    .........ssl.gstatic.com.....
127.0.0.1    39326    127.0.0.1    53    0    ...5.).<|>...........ssl.gstatic.com.....
127.0.0.1    39326    127.0.0.1    53    0    ...5.).<|>...........ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    46624    16401    @...........5.).<|>...........ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    46624    16401    @...........5.).<|>...........ssl.gstatic.com.....
127.0.0.1    39326    127.0.0.1    53    0    ...5.).<............ssl.gstatic.com.....
127.0.0.1    39326    127.0.0.1    53    0    ...5.).<............ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    4703    16401    @...........5.).<............ssl.gstatic.com.....
127.0.0.1    771    127.0.0.1    4703    16401    @...........5.).<............ssl.gstatic.com.....
127.0.0.1    43617    127.0.0.1    53    0    .a.5.).<............ssl.gstatic.com.....
127.0.0.1    43617    127.0.0.1    53    0    .a.5.).<............ssl.gstatic.com.....
```

As we have a lot of packets from this IP, we could again filter it with respect to milkshake.

```
grep 127.0.0.1 output.txt | grep milkshake
[For generic] grep 127.0.0.1 <path to output file> | grep milkshake
```

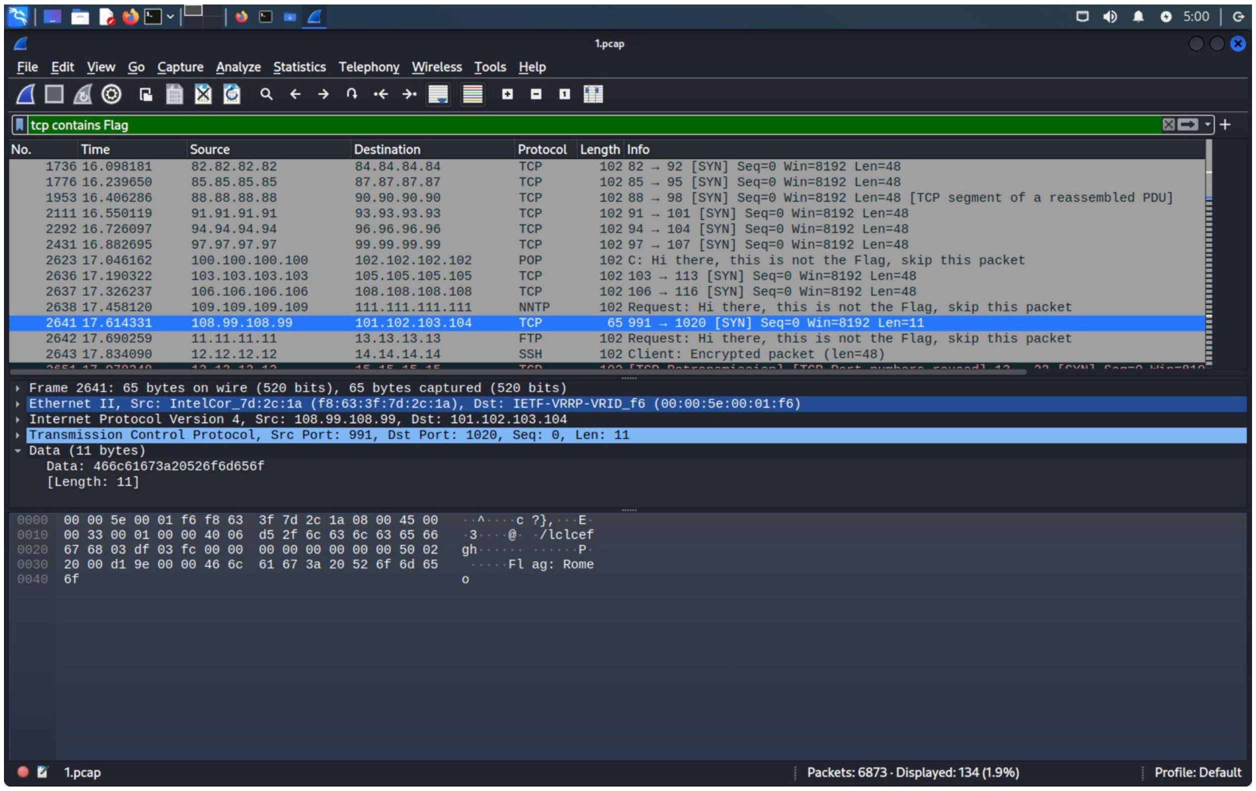


```
→ CN_Assignment-1 git:(main) ✗ grep 127.0.0.1 output.txt | grep milkshake
grep: output.txt: binary file matches
127.0.0.1    1111    122.44.56.78    9876    9327    GET /milkshake HTTP/1.1..Cookie: user:customer..Referer: flavor- Strawberry....
127.0.0.1    1111    122.44.56.78    9876    9327    GET /milkshake HTTP/1.1..Cookie: user:customer..Referer: flavor- Strawberry....
→ CN_Assignment-1 git:(main) ✗
```

Verification:

Using Wireshark Tool to verify the answers.

1. **Romeo**

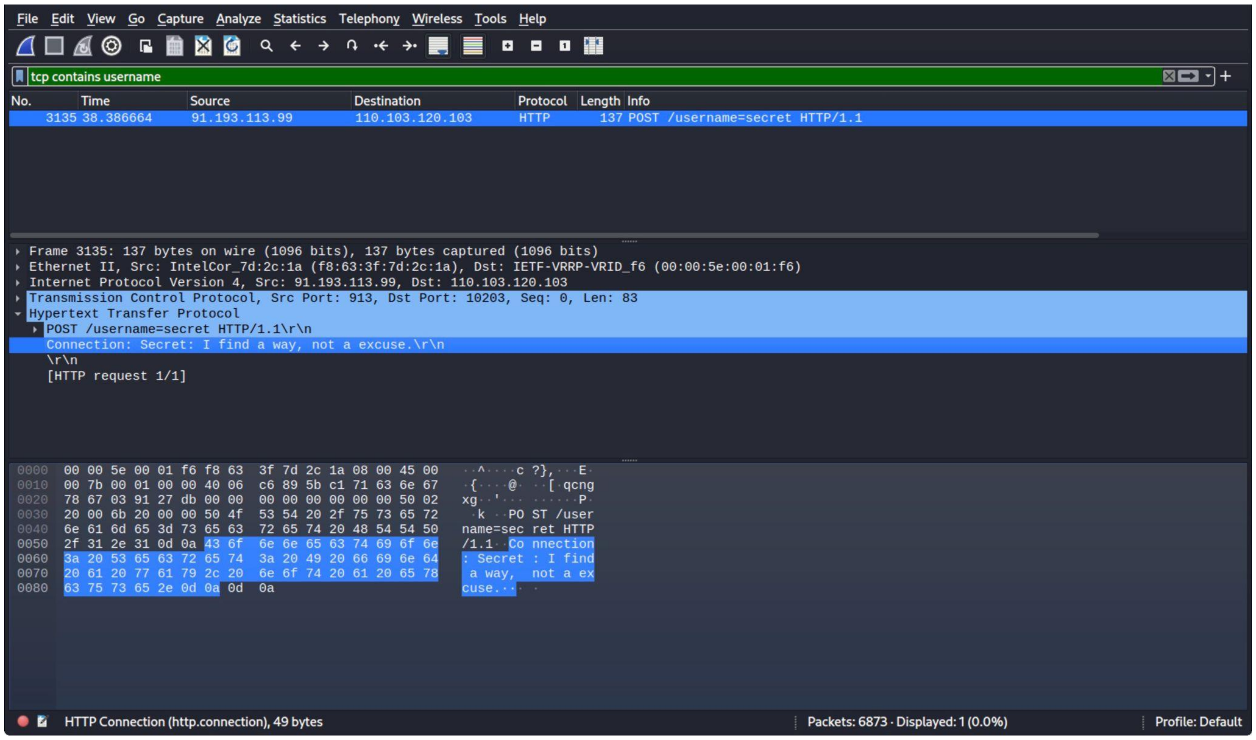


```
tcp contains Flag

No.   Time      Source           Destination      Protocol Length Info
 1736 16.098181 82.82.82.82      84.84.84.84      TCP       102 82 → 92 [SYN] Seq=0 Win=8192 Len=48
 1776 16.239650 85.85.85.85      87.87.87.87      TCP       102 85 → 95 [SYN] Seq=0 Win=8192 Len=48
 1953 16.406286 88.88.88.88      90.90.90.90      TCP       102 88 → 98 [SYN] Seq=0 Win=8192 Len=48 [TCP segment of a reassembled PDU]
 2111 16.550119 91.91.91.91      93.93.93.93      TCP       102 91 → 101 [SYN] Seq=0 Win=8192 Len=48
 2292 16.726097 94.94.94.94      96.96.96.96      TCP       102 94 → 104 [SYN] Seq=0 Win=8192 Len=48
 2431 16.882695 97.97.97.97      99.99.99.99      TCP       102 97 → 107 [SYN] Seq=0 Win=8192 Len=48
 2623 17.046162 100.100.100.100  102.102.102.102  POP       102 C: Hi there, this is not the Flag, skip this packet
 2636 17.190322 103.103.103.103  105.105.105.105  TCP       102 103 → 113 [SYN] Seq=0 Win=8192 Len=48
 2637 17.326237 106.106.106.106  108.108.108.108  TCP       102 106 → 116 [SYN] Seq=0 Win=8192 Len=48
 2638 17.458120 109.109.109.109  111.111.111.111  NNTP      102 Request: Hi there, this is not the Flag, skip this packet
 2641 17.614331 108.99.108.99    101.102.103.104  TCP        65 991 → 1020 [SYN] Seq=0 Win=8192 Len=11
 2642 17.690259 11.11.11.11      13.13.13.13      FTP       102 Request: Hi there, this is not the Flag, skip this packet
 2643 17.834090 12.12.12.12      14.14.14.14      SSH       102 Client: Encrypted packet (len=48)

› Frame 2641: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 108.99.108.99, Dst: 101.102.103.104
› Transmission Control Protocol, Src Port: 991, Dst Port: 1020, Seq: 0, Len: 11
⌄ Data (11 bytes)
    Data: 466c61673a20526f6d656f
    [Length: 11]

0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 33 00 01 00 00 40 06  d5 2f 6c 63 6c 63 65 66   ·3····@· ·/lclcef
0020  67 68 03 df 03 fc 00 00  00 00 00 00 00 00 50 02   gh······ ······P·
0030  20 00 d1 9e 00 00 46 6c  61 67 3a 20 52 6f 6d 65    ·····Fl ag: Rome
0040  6f                                                 o

1.pcap                                    Packets: 6873 · Displayed: 134 (1.9%)    Profile: Default
```

As shown above we used a filter[tcp contains Flag] which filters out all the packets containing the keyword Flag and there were several such false packets which were supposed to be skipped and we finally found the correct packet.

2. username=secret

Connection: **Secret: I find a way, not a excuse.**



tcp contains username

```
No.   Time       Source          Destination      Protocol Length Info
 3135 38.386664  91.193.113.99   110.103.120.103  HTTP        137 POST /username=secret HTTP/1.1
```

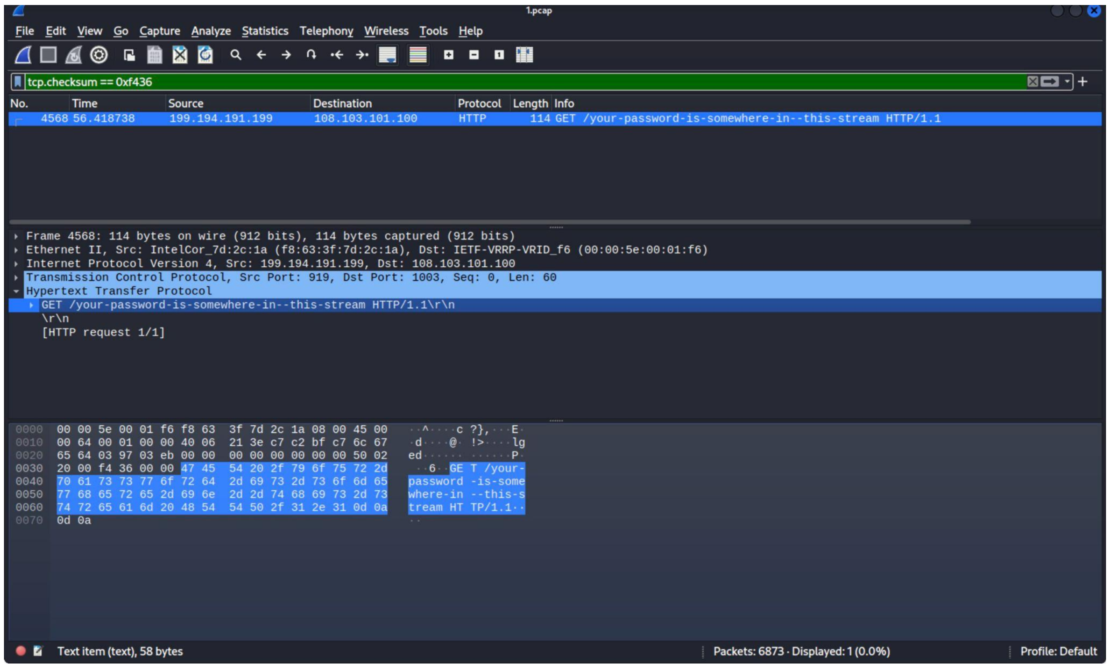
```
› Frame 3135: 137 bytes on wire (1096 bits), 137 bytes captured (1096 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 91.193.113.99, Dst: 110.103.120.103
› Transmission Control Protocol, Src Port: 913, Dst Port: 10203, Seq: 0, Len: 83
⌄ Hypertext Transfer Protocol
  › POST /username=secret HTTP/1.1\r\n
    Connection: Secret: I find a way, not a excuse.\r\n
    \r\n
    [HTTP request 1/1]
```

```
0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 7b 00 01 00 00 40 06  c6 89 5b c1 71 63 6e 67   ·{····@· ··[·qcng
0020  78 67 03 91 27 db 00 00  00 00 00 00 00 00 50 02   xg··'··· ······P·
0030  20 00 6b 20 00 00 50 4f  53 54 20 2f 75 73 65 72    ·k ··PO ST /user
0040  6e 61 6d 65 3d 73 65 63  72 65 74 20 48 54 54 50   name=sec ret HTTP
0050  2f 31 2e 31 0d 0a 43 6f  6e 6e 65 63 74 69 6f 6e   /1.1··Co nnection
0060  3a 20 53 65 63 72 65 74  3a 20 49 20 66 69 6e 64   : Secret : I find
0070  20 61 20 77 61 79 2c 20  6e 6f 74 20 61 20 65 78    a way,  not a ex
0080  63 75 73 65 2e 0d 0a 0d  0a                        cuse.··· ·
```

HTTP Connection (http.connection), 49 bytes | Packets: 6873 · Displayed: 1 (0.0%) | Profile: Default

Filtered out all the packets containing the word username and we were able to find the secret.

3. **Berlin** (for www.cs433.com) [in form of user-agent: Password]



1.pcap

tcp.checksum == 0xf436

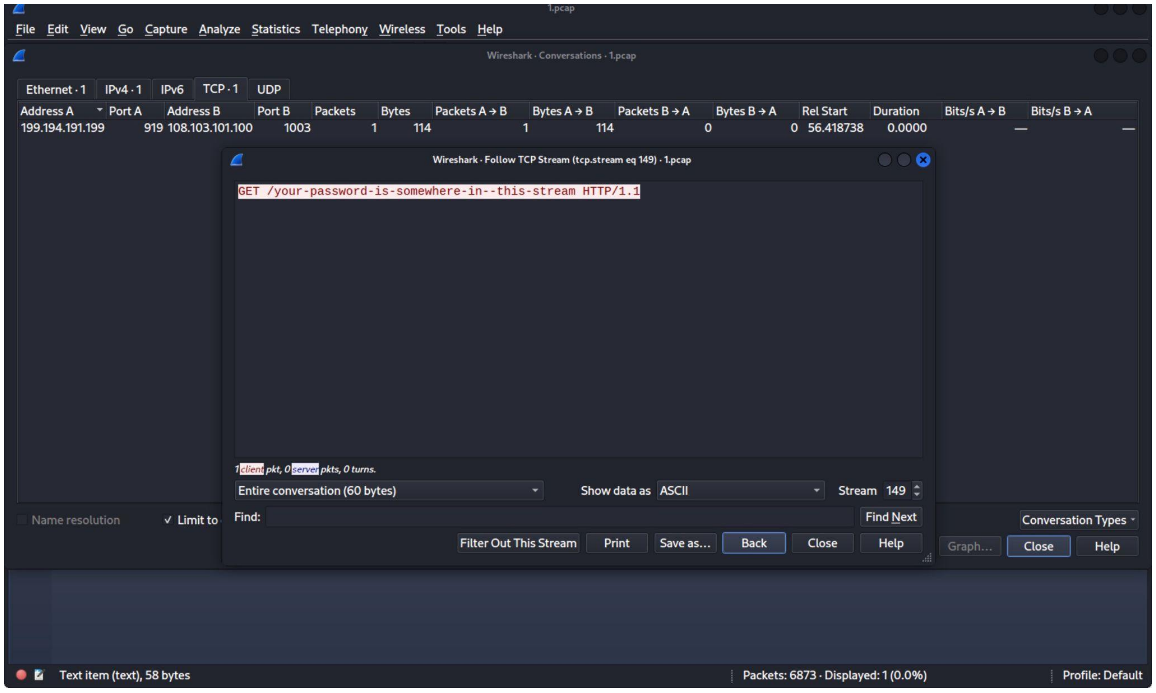
```
No.   Time       Source           Destination      Protocol Length Info
 4568 56.418738  199.194.191.199  108.103.101.100  HTTP        114 GET /your-password-is-somewhere-in--this-stream HTTP/1.1
```

```
› Frame 4568: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 199.194.191.199, Dst: 108.103.101.100
› Transmission Control Protocol, Src Port: 919, Dst Port: 1003, Seq: 0, Len: 60
⌄ Hypertext Transfer Protocol
  › GET /your-password-is-somewhere-in--this-stream HTTP/1.1\r\n
    \r\n
    [HTTP request 1/1]
```

```
0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 64 00 01 00 00 40 06  21 3e c7 c2 bf c7 6c 67   ·d····@· !>····lg
0020  65 64 03 97 03 eb 00 00  00 00 00 00 00 00 50 02   ed······ ······P·
0030  20 00 f4 36 00 00 47 45  54 20 2f 79 6f 75 72 2d    ··6··GE T /your-
0040  70 61 73 73 77 6f 72 64  2d 69 73 2d 73 6f 6d 65   password -is-some
0050  77 68 65 72 65 2d 69 6e  2d 2d 74 68 69 73 2d 73   where-in --this-s
0060  74 72 65 61 6d 20 48 54  54 50 2f 31 2e 31 0d 0a   tream HT TP/1.1··
0070  0d 0a                                              ··
```

Text item (text), 58 bytes | Packets: 6873 · Displayed: 1 (0.0%) | Profile: Default

Filtered with respect to the tcp checksum and it gave the instruction to follow the stream. So, we opened the 'Conversations' option in the Statistics menu and limited the conversation to filter used. After selecting the conversation we had an option to follow the stream.

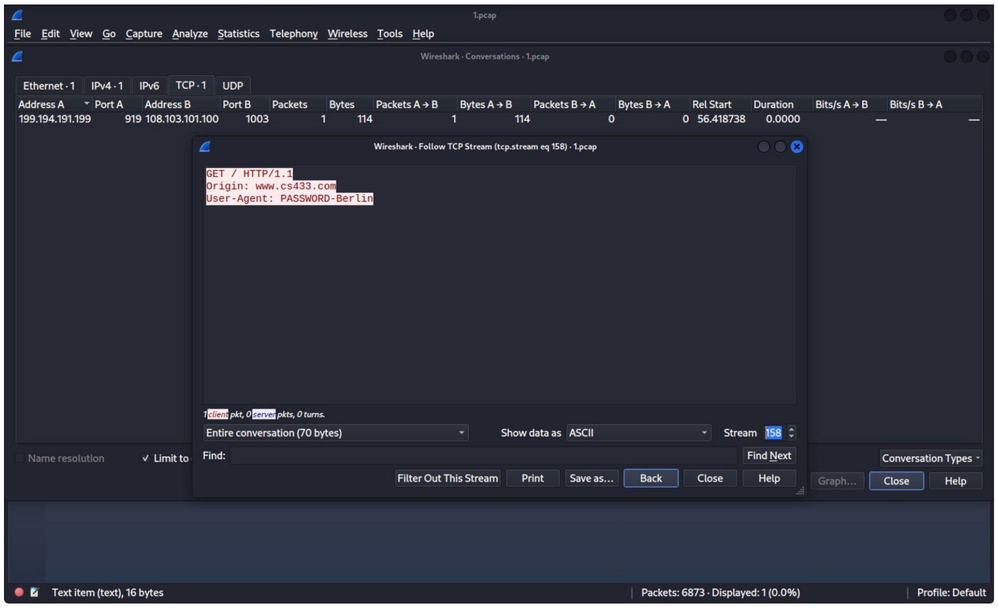

1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · 1.pcap
Ethernet · 1 IPv4 · 1 IPv6 TCP · 1 UDP
Wireshark · Follow TCP Stream (tcp.stream eq 149) · 1.pcap
Name resolution
Limit to
Conversation Types
Filter Out This Stream
Print
Save as…
Back
Close
Help
Graph…
Find Next

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
199.194.191.199	919	108.103.101.100	1003	1	114	1	114	0	0	56.418738	0.0000	—	—

```
GET /your-password-is-somewhere-in--this-stream HTTP/1.1
```

1 client pkt, 0 server pkts, 0 turns.
Entire conversation (60 bytes)
Show data as ASCII
Stream 149
Find:
Text item (text), 58 bytes
Packets: 6873 · Displayed: 1 (0.0%)
Profile: Default

There was an option to increase the stream value and we kept on going front and back to find the flag, and we were able to find the password in the stream: 158.


1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · 1.pcap
Ethernet · 1 IPv4 · 1 IPv6 TCP · 1 UDP
Wireshark · Follow TCP Stream (tcp.stream eq 158) · 1.pcap
Name resolution
Limit to
Conversation Types
Filter Out This Stream
Print
Save as…
Back
Close
Help
Graph…
Find Next

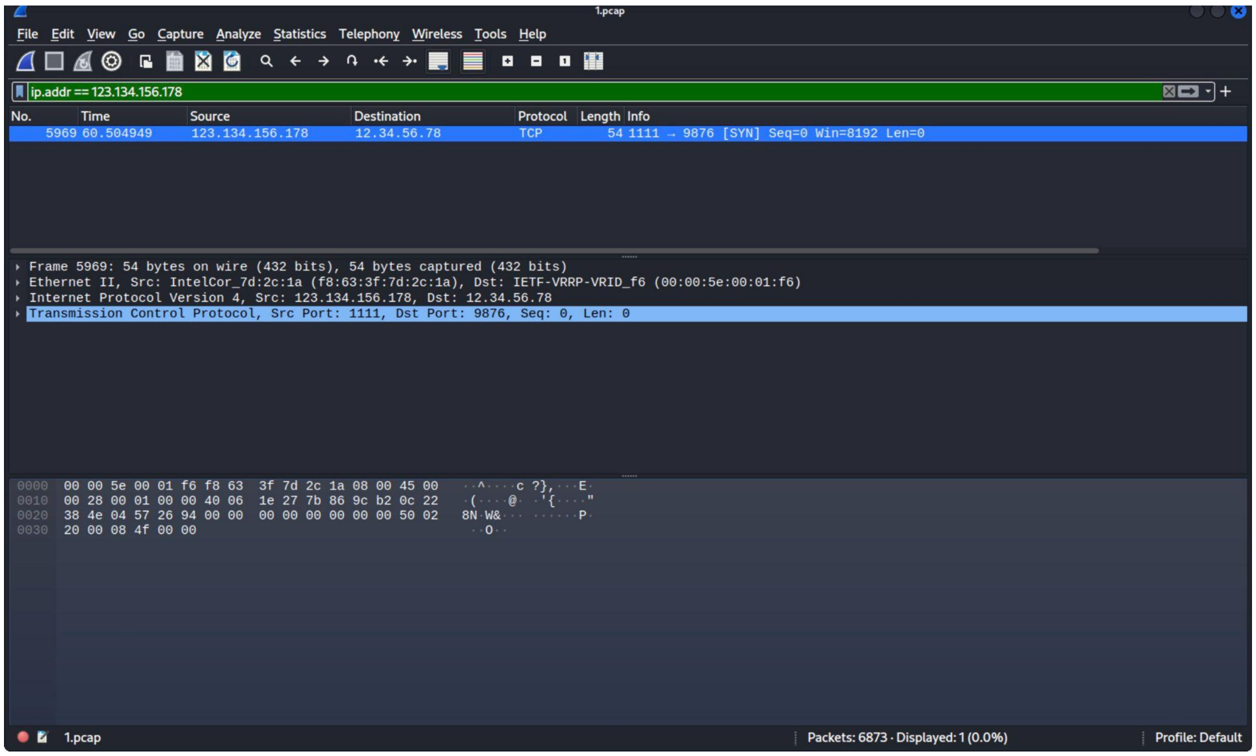
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
199.194.191.199	919	108.103.101.100	1003	1	114	1	114	0	0	56.418738	0.0000	—	—

```
GET / HTTP/1.1
Origin: www.cs433.com
User-Agent: PASSWORD-Berlin
```

1 client pkt, 0 server pkts, 0 turns.
Entire conversation (70 bytes)
Show data as ASCII
Stream 158
Find:
Text item (text), 16 bytes
Packets: 6873 · Displayed: 1 (0.0%)
Profile: Default

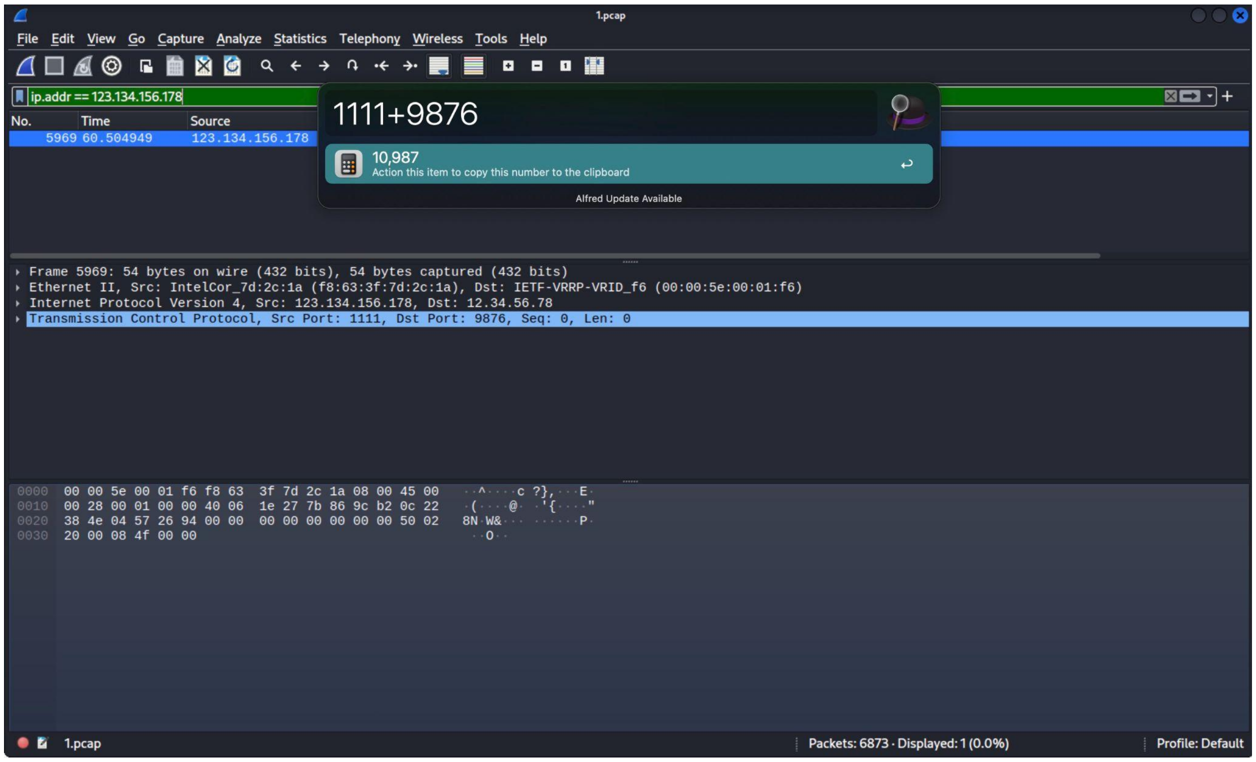
4. **Rabindranath Tagore**

Filtering with respect to the ip address provided.


1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

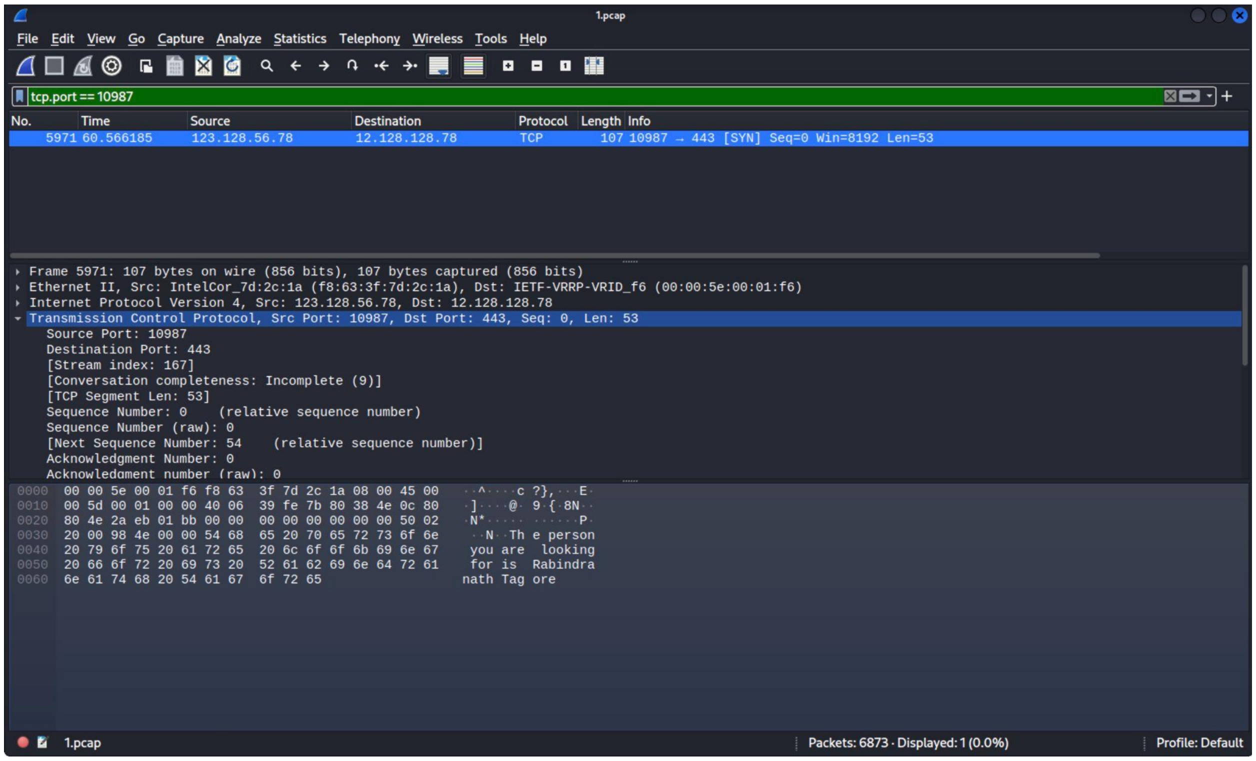
```
ip.addr == 123.134.156.178
No.   Time          Source            Destination      Protocol  Length  Info
 5969 60.504949     123.134.156.178   12.34.56.78      TCP           54  1111 → 9876 [SYN] Seq=0 Win=8192 Len=0

› Frame 5969: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 123.134.156.178, Dst: 12.34.56.78
› Transmission Control Protocol, Src Port: 1111, Dst Port: 9876, Seq: 0, Len: 0

0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 28 00 01 00 00 40 06  1e 27 7b 86 9c b2 0c 22   ·(····@· ·'{····"
0020  38 4e 04 57 26 94 00 00  00 00 00 00 00 00 50 02   8N·W&··· ······P·
0030  20 00 08 4f 00 00                                   ··O··

1.pcap                                   Packets: 6873 · Displayed: 1 (0.0%)      Profile: Default
```

The port values are: 1111 and 9876 and their sum is 10987.


1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Alfred Update Available

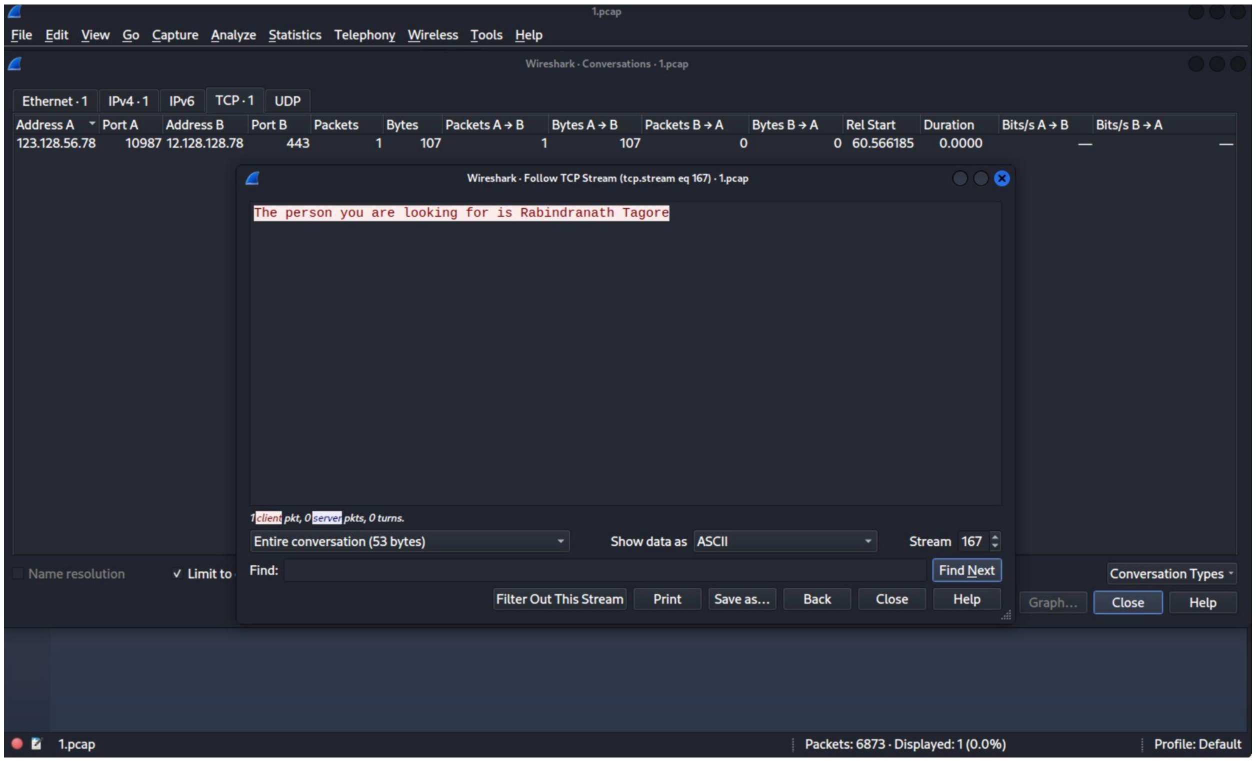
```
ip.addr == 123.134.156.178
1111+9876
10,987
Action this item to copy this number to the clipboard

No.   Time          Source
 5969 60.504949     123.134.156.178

› Frame 5969: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 123.134.156.178, Dst: 12.34.56.78
› Transmission Control Protocol, Src Port: 1111, Dst Port: 9876, Seq: 0, Len: 0

0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 28 00 01 00 00 40 06  1e 27 7b 86 9c b2 0c 22   ·(····@· ·'{····"
0020  38 4e 04 57 26 94 00 00  00 00 00 00 00 00 50 02   8N·W&··· ······P·
0030  20 00 08 4f 00 00                                   ··O··

1.pcap                                   Packets: 6873 · Displayed: 1 (0.0%)      Profile: Default
```

Filtering with respect to the port value 10987.


1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

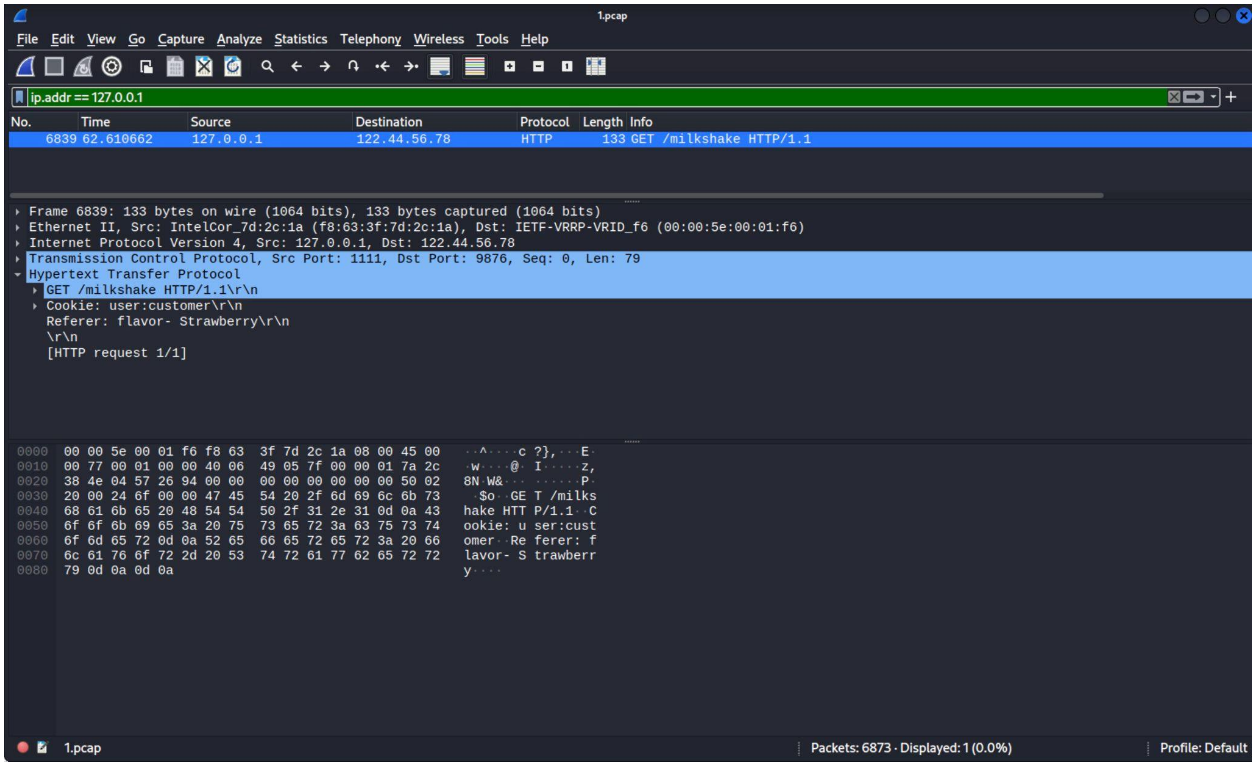
```
tcp.port == 10987

No.   Time        Source          Destination     Protocol Length Info
 5971 60.566185   123.128.56.78   12.128.128.78   TCP      107 10987 → 443 [SYN] Seq=0 Win=8192 Len=53

› Frame 5971: 107 bytes on wire (856 bits), 107 bytes captured (856 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 123.128.56.78, Dst: 12.128.128.78
⌄ Transmission Control Protocol, Src Port: 10987, Dst Port: 443, Seq: 0, Len: 53
    Source Port: 10987
    Destination Port: 443
    [Stream index: 167]
    [Conversation completeness: Incomplete (9)]
    [TCP Segment Len: 53]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 0
    [Next Sequence Number: 54    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0

0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 5d 00 01 00 00 40 06  39 fe 7b 80 38 4e 0c 80   ·]····@· 9·{·8N··
0020  80 4e 2a eb 01 bb 00 00  00 00 00 00 00 00 50 02   ·N*····· ······P·
0030  20 00 98 4e 00 00 54 68  65 20 70 65 72 73 6f 6e    ··N··Th e person
0040  20 79 6f 75 20 61 72 65  20 6c 6f 6f 6b 69 6e 67    you are  looking
0050  20 66 6f 72 20 69 73 20  52 61 62 69 6e 64 72 61    for is  Rabindra
0060  6e 61 74 68 20 54 61 67  6f 72 65                  nath Tag ore

1.pcap          Packets: 6873 · Displayed: 1 (0.0%)          Profile: Default
```

We just checked the conversation tab and followed the stream to just get a better understanding.


1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Conversations · 1.pcap
Ethernet · 1
IPv4 · 1
IPv6
TCP · 1
UDP
Wireshark · Follow TCP Stream (tcp.stream eq 167) · 1.pcap
Name resolution
Limit to
Find Next
Filter Out This Stream
Print
Save as…
Back
Close
Help
Graph…
Conversation Types

```
Address A      Port A  Address B      Port B  Packets  Bytes  Packets A → B  Bytes A → B  Packets B → A  Bytes B → A  Rel Start  Duration  Bits/s A → B  Bits/s B → A
123.128.56.78  10987   12.128.128.78  443     1        107    1              107          0              0            60.566185  0.0000    —             —

The person you are looking for is Rabindranath Tagore

1 client pkt, 0 server pkts, 0 turns.
Entire conversation (53 bytes)      Show data as  ASCII      Stream  167
Find:

1.pcap          Packets: 6873 · Displayed: 1 (0.0%)          Profile: Default
```

5. **Strawberry**


1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 127.0.0.1
```

```
No.   Time        Source      Destination   Protocol Length Info
 6839 62.610662   127.0.0.1   122.44.56.78  HTTP      133 GET /milkshake HTTP/1.1
```

```
› Frame 6839: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
› Ethernet II, Src: IntelCor_7d:2c:1a (f8:63:3f:7d:2c:1a), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
› Internet Protocol Version 4, Src: 127.0.0.1, Dst: 122.44.56.78
› Transmission Control Protocol, Src Port: 1111, Dst Port: 9876, Seq: 0, Len: 79
⌄ Hypertext Transfer Protocol
  › GET /milkshake HTTP/1.1\r\n
  › Cookie: user:customer\r\n
    Referer: flavor- Strawberry\r\n
    \r\n
    [HTTP request 1/1]
```

```
0000  00 00 5e 00 01 f6 f8 63  3f 7d 2c 1a 08 00 45 00   ··^····c ?},···E·
0010  00 77 00 01 00 00 40 06  49 05 7f 00 00 01 7a 2c   ·w····@· I·····z,
0020  38 4e 04 57 26 94 00 00  00 00 00 00 00 00 50 02   8N·W&··· ······P·
0030  20 00 24 6f 00 00 47 45  54 20 2f 6d 69 6c 6b 73    ·$o··GE T /milks
0040  68 61 6b 65 20 48 54 54  50 2f 31 2e 31 0d 0a 43   hake HTT P/1.1··C
0050  6f 6f 6b 69 65 3a 20 75  73 65 72 3a 63 75 73 74   ookie: u ser:cust
0060  6f 6d 65 72 0d 0a 52 65  66 65 72 65 72 3a 20 66   omer··Re ferer: f
0070  6c 61 76 6f 72 2d 20 53  74 72 61 77 62 65 72 72   lavor- S trawberr
0080  79 0d 0a 0d 0a                                    y····
```

```
1.pcap        Packets: 6873 · Displayed: 1 (0.0%)        Profile: Default
```

Local Host has an IP address: 127.0.0.1, so filtering with respect to that we get the flag.