# CNSL Assignment 10

## Facebook Packet Analysis

### Aim

Capture packets using Wireshark, write the exact packet capture filter expressions to accomplish the following and save the output in file:

1. Capture all TCP traffic to/from Facebook, during the time when you log in to your Facebook account
2. Capture all HTTP traffic to/from Facebook, when you log in to your Facebook account
3. Write a DISPLAY filter expression to count all TCP packets (captured under item #1) that have the flags SYN, PSH, and RST set. Show the fraction of packets that had each flag set.
4. Count how many TCP packets you received from / sent to Face book, and how many of each were also HTTP packets.

### Software / Hardware Requirements

Software: Wireshark (latest version)

Hardware: Computer with active internet connection

Operating System: Windows/Linux/MacOS

### Theory

Wireshark is a widely used network protocol analyzer that captures and inspects packets in real time. It allows users to apply capture filters (to restrict packets being recorded) and display filters (to analyze specific traffic after capture).

### Key concepts:

Capture Filter: Applied before capturing packets, based on Berkeley Packet Filter (BPF) syntax.

Display Filter: Applied after capturing, using Wireshark’s own syntax.

TCP Flags: Control flags in TCP header such as SYN (synchronize), PSH (push), RST (reset).

### Steps with sample output

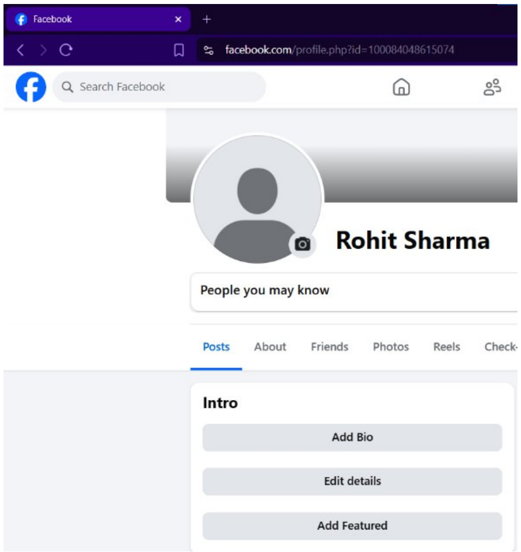
1. Capture all TCP traffic to/from Facebook, during the time when you log in to your Facebook account:

   Open Wireshark protocol analyzer in ensp mode.

   Login to facebook.com and log out immediately once the home page appears.

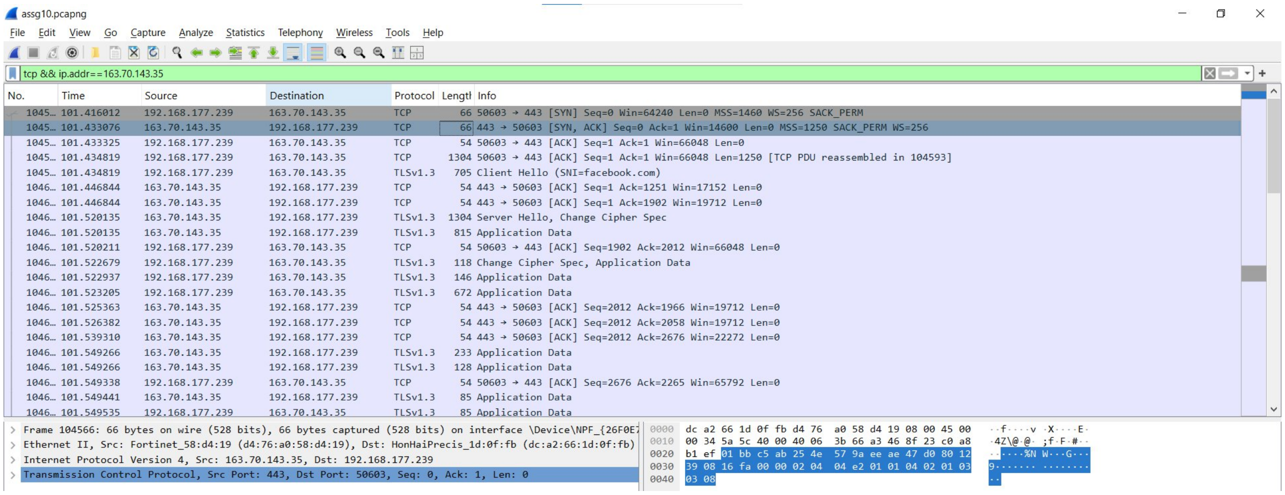
   Switch back to Wireshark protocol analyzer and press stop. Save the pcapng file.

   Now, the analyzation begins. Firstly, apply the filter of Facebook IP address and tcp. The output is shown in below figure.



facebook.com/profile.php?id=100084048615074

Fig 1: Facebook login page

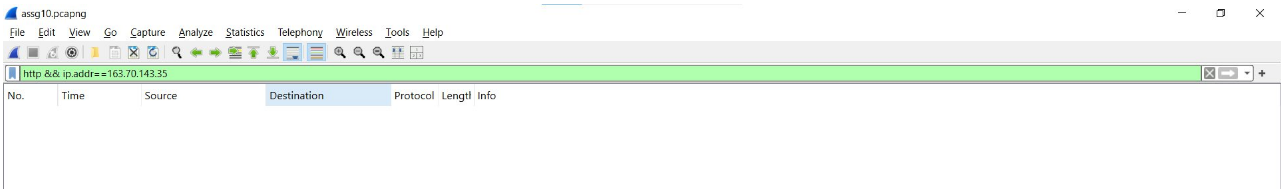

assg10.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp && ip.addr==163.70.143.35

No.     Time        Source           Destination      Protocol Lengtl Info
1045… 101.416012  192.168.177.239  163.70.143.35    TCP        66 50603 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1045… 101.433076  163.70.143.35    192.168.177.239  TCP        66 443 → 50603 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=256
1045… 101.433325  192.168.177.239  163.70.143.35    TCP        54 50603 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0
1045… 101.434819  192.168.177.239  163.70.143.35    TCP      1304 50603 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=1250 [TCP PDU reassembled in 104593]
1045… 101.434819  192.168.177.239  163.70.143.35    TLSv1.3   705 Client Hello (SNI=facebook.com)
1046… 101.446844  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=1 Ack=1251 Win=17152 Len=0
1046… 101.446844  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=1 Ack=1902 Win=19712 Len=0
1046… 101.520135  163.70.143.35    192.168.177.239  TLSv1.3  1304 Server Hello, Change Cipher Spec
1046… 101.520135  163.70.143.35    192.168.177.239  TLSv1.3   815 Application Data
1046… 101.520211  192.168.177.239  163.70.143.35    TCP        54 50603 → 443 [ACK] Seq=1902 Ack=2012 Win=66048 Len=0
1046… 101.522679  192.168.177.239  163.70.143.35    TLSv1.3   118 Change Cipher Spec, Application Data
1046… 101.522937  192.168.177.239  163.70.143.35    TLSv1.3   146 Application Data
1046… 101.523205  192.168.177.239  163.70.143.35    TLSv1.3   672 Application Data
1046… 101.525363  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=2012 Ack=1966 Win=19712 Len=0
1046… 101.526382  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=2012 Ack=2058 Win=19712 Len=0
1046… 101.539310  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=2012 Ack=2676 Win=22272 Len=0
1046… 101.549266  163.70.143.35    192.168.177.239  TLSv1.3   233 Application Data
1046… 101.549266  163.70.143.35    192.168.177.239  TLSv1.3   128 Application Data
1046… 101.549338  192.168.177.239  163.70.143.35    TCP        54 50603 → 443 [ACK] Seq=2676 Ack=2265 Win=65792 Len=0
1046… 101.549441  163.70.143.35    192.168.177.239  TLSv1.3    85 Application Data
1046… 101.549535  192.168.177.239  163.70.143.35    TLSv1.3    85 Application Data

> Frame 104566: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{26F0E7
> Ethernet II, Src: Fortinet_58:d4:19 (d4:76:a0:58:d4:19), Dst: HonHaiPrecis_1d:0f:fb (dc:a2:66:1d:0f:fb)
> Internet Protocol Version 4, Src: 163.70.143.35, Dst: 192.168.177.239
> Transmission Control Protocol, Src Port: 443, Dst Port: 50603, Seq: 0, Ack: 1, Len: 0

0000  dc a2 66 1d 0f fb d4 76  a0 58 d4 19 08 00 45 00   ··f····v ·X····E·
0010  00 34 5a 5c 40 00 40 06  3b 66 a3 46 8f 23 c0 a8   ·4Z\@·@· ;f·F·#··
0020  b1 ef 01 bb c5 ab 25 4e  57 9a ee ae 47 d0 80 12   ······%N W···G···
0030  39 08 16 fa 00 00 02 04  04 e2 01 01 04 02 01 03   9······· ········
0040  03 08                                              ··
```

Fig 2: Facebook ip & tcp filter

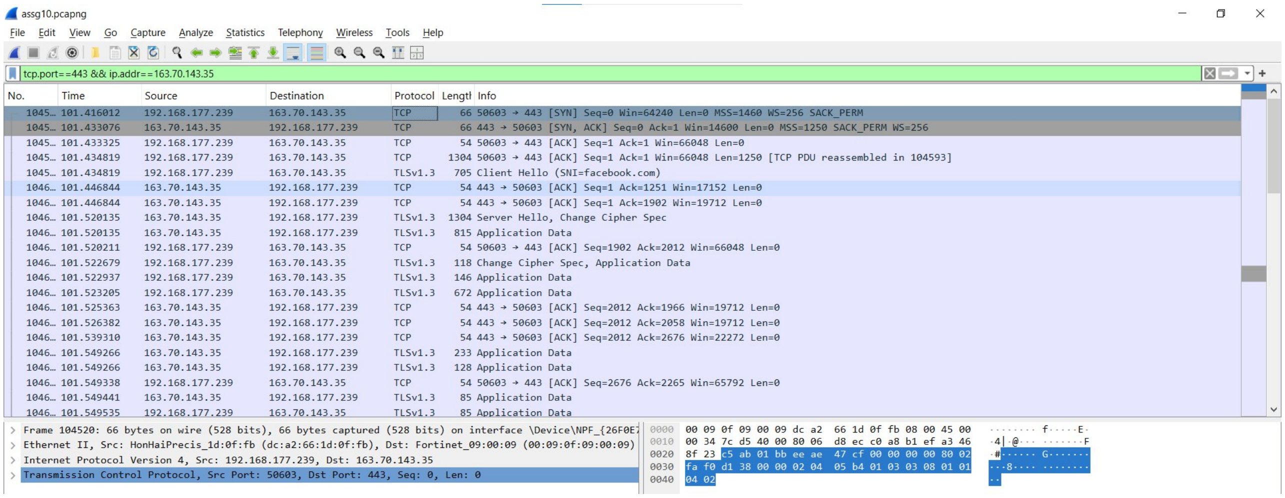
2. Capture all HTTP/HTTPS traffic to/from Facebook, when you log in to your Facebook account.

Now, apply the Facebook IP and http/https filter. The output of the following is shown below figure.


assg10.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http && ip.addr==163.70.143.35

No.     Time        Source           Destination      Protocol Lengtl Info
```

Fig 3: packets with http filter.

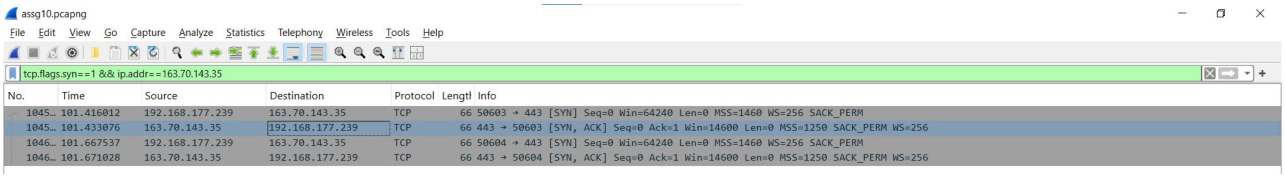

assg10.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.port==443 && ip.addr==163.70.143.35

No.     Time        Source           Destination      Protocol Lengtl Info
1045… 101.416012  192.168.177.239  163.70.143.35    TCP        66 50603 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1045… 101.433076  163.70.143.35    192.168.177.239  TCP        66 443 → 50603 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=256
1045… 101.433325  192.168.177.239  163.70.143.35    TCP        54 50603 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0
1045… 101.434819  192.168.177.239  163.70.143.35    TCP      1304 50603 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=1250 [TCP PDU reassembled in 104593]
1045… 101.434819  192.168.177.239  163.70.143.35    TLSv1.3   705 Client Hello (SNI=facebook.com)
1046… 101.446844  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=1 Ack=1251 Win=17152 Len=0
1046… 101.446844  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=1 Ack=1902 Win=19712 Len=0
1046… 101.520135  163.70.143.35    192.168.177.239  TLSv1.3  1304 Server Hello, Change Cipher Spec
1046… 101.520135  163.70.143.35    192.168.177.239  TLSv1.3   815 Application Data
1046… 101.520211  192.168.177.239  163.70.143.35    TCP        54 50603 → 443 [ACK] Seq=1902 Ack=2012 Win=66048 Len=0
1046… 101.522679  192.168.177.239  163.70.143.35    TLSv1.3   118 Change Cipher Spec, Application Data
1046… 101.522937  192.168.177.239  163.70.143.35    TLSv1.3   146 Application Data
1046… 101.523205  192.168.177.239  163.70.143.35    TLSv1.3   672 Application Data
1046… 101.525363  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=2012 Ack=1966 Win=19712 Len=0
1046… 101.526382  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=2012 Ack=2058 Win=19712 Len=0
1046… 101.539310  163.70.143.35    192.168.177.239  TCP        54 443 → 50603 [ACK] Seq=2012 Ack=2676 Win=22272 Len=0
1046… 101.549266  163.70.143.35    192.168.177.239  TLSv1.3   233 Application Data
1046… 101.549266  163.70.143.35    192.168.177.239  TLSv1.3   128 Application Data
1046… 101.549338  192.168.177.239  163.70.143.35    TCP        54 50603 → 443 [ACK] Seq=2676 Ack=2265 Win=65792 Len=0
1046… 101.549441  163.70.143.35    192.168.177.239  TLSv1.3    85 Application Data
1046… 101.549535  192.168.177.239  163.70.143.35    TLSv1.3    85 Application Data

> Frame 104520: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{26F0E7
> Ethernet II, Src: HonHaiPrecis_1d:0f:fb (dc:a2:66:1d:0f:fb), Dst: Fortinet_09:00:09 (00:09:0f:09:00:09)
> Internet Protocol Version 4, Src: 192.168.177.239, Dst: 163.70.143.35
> Transmission Control Protocol, Src Port: 50603, Dst Port: 443, Seq: 0, Len: 0

0000  00 09 0f 09 00 09 dc a2  66 1d 0f fb 08 00 45 00   ········ f·····E·
0010  00 34 7c d5 40 00 80 06  d8 ec c0 a8 b1 ef a3 46   ·4|·@··· ·······F
0020  8f 23 c5 ab 01 bb ee ae  47 cf 00 00 00 00 80 02   ·#······ G·······
0030  fa f0 d1 38 00 00 02 04  05 b4 01 03 03 08 01 01   ···8···· ········
0040  04 02                                              ··
```

Fig 4: packets with https filter.

3. Write a DISPLAY filter expression to count all TCP packets that have the flags SYN, PSH, and RST set. Show the fraction of packets that had each flag set.

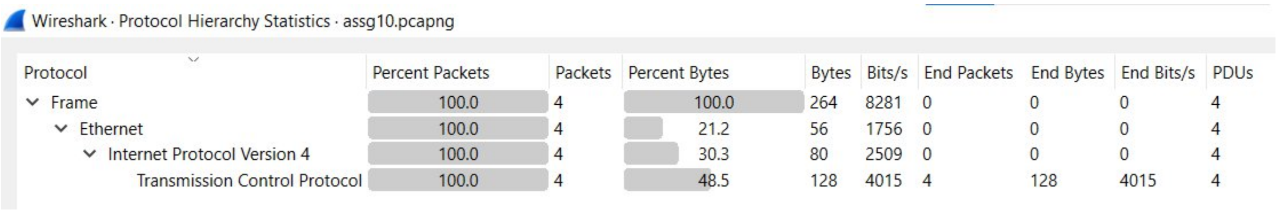
Now, apply the filter of Facebook IP and tcp.flag.syn == 1. The output is shown in the following figure.


assg10.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.flags.syn==1 && ip.addr==163.70.143.35

No.     Time        Source           Destination      Protocol Lengtl Info
1045… 101.416012  192.168.177.239  163.70.143.35    TCP        66 50603 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1045… 101.433076  163.70.143.35    192.168.177.239  TCP        66 443 → 50603 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=256
1046… 101.667537  192.168.177.239  163.70.143.35    TCP        66 50604 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1046… 101.671028  163.70.143.35    192.168.177.239  TCP        66 443 → 50604 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=256
```

Fig 5: packets with SYN flag filter.

Now, on the header menu, click the statistics options and select the option saying protocol hierarchy. This will result in the stats about the protocol. The output is shown in the below figure.

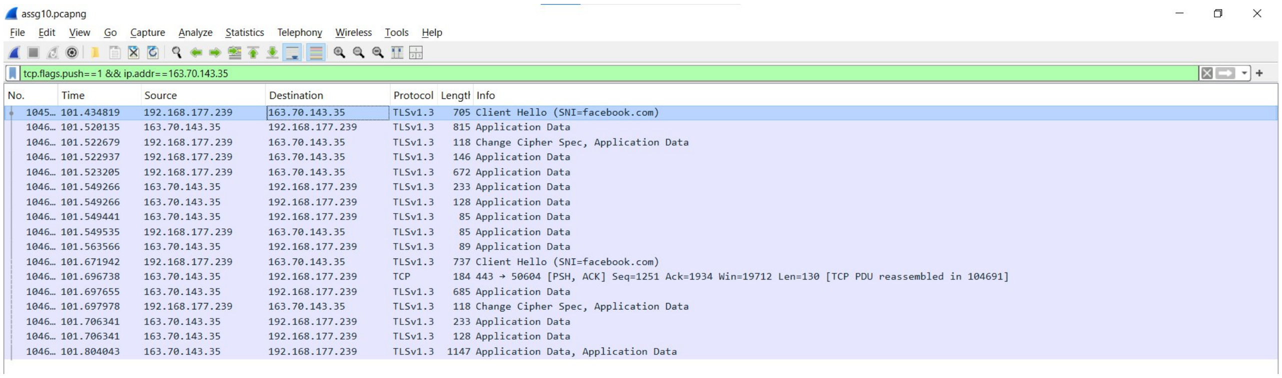


Wireshark · Protocol Hierarchy Statistics · assg10.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s | PDUs |
|---|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 4 | 100.0 | 264 | 8281 | 0 | 0 | 0 | 4 |
| Ethernet | 100.0 | 4 | 21.2 | 56 | 1756 | 0 | 0 | 0 | 4 |
| Internet Protocol Version 4 | 100.0 | 4 | 30.3 | 80 | 2509 | 0 | 0 | 0 | 4 |
| Transmission Control Protocol | 100.0 | 4 | 48.5 | 128 | 4015 | 4 | 128 | 4015 | 4 |

Fig 6: SYN flag protocol hierarchy.

Now, apply the filter of Facebook IP with push flag tcp.flags.psh == 1. The output of the following is shown in the following figure.



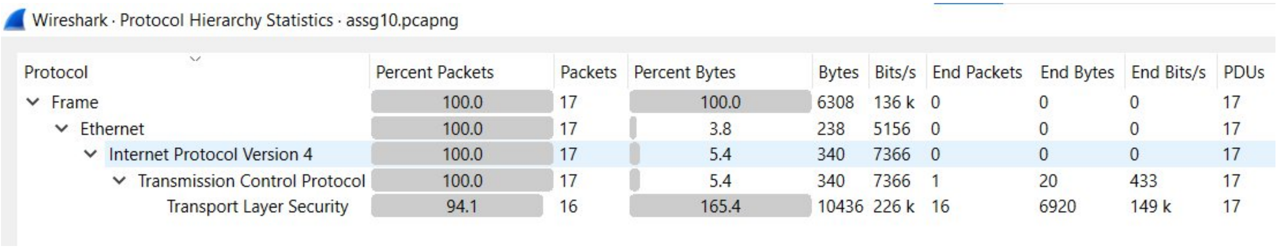
assg10.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.push==1 && ip.addr==163.70.143.35

```
No.     Time        Source            Destination       Protocol Lengtl Info
1045… 101.434819  192.168.177.239   163.70.143.35     TLSv1.3   705 Client Hello (SNI=facebook.com)
1046… 101.520135  163.70.143.35     192.168.177.239   TLSv1.3   815 Application Data
1046… 101.522679  192.168.177.239   163.70.143.35     TLSv1.3   118 Change Cipher Spec, Application Data
1046… 101.522937  192.168.177.239   163.70.143.35     TLSv1.3   146 Application Data
1046… 101.523205  192.168.177.239   163.70.143.35     TLSv1.3   672 Application Data
1046… 101.549266  163.70.143.35     192.168.177.239   TLSv1.3   233 Application Data
1046… 101.549266  163.70.143.35     192.168.177.239   TLSv1.3   128 Application Data
1046… 101.549441  163.70.143.35     192.168.177.239   TLSv1.3    85 Application Data
1046… 101.549535  192.168.177.239   163.70.143.35     TLSv1.3    85 Application Data
1046… 101.563566  163.70.143.35     192.168.177.239   TLSv1.3    89 Application Data
1046… 101.671942  192.168.177.239   163.70.143.35     TLSv1.3   737 Client Hello (SNI=facebook.com)
1046… 101.696738  163.70.143.35     192.168.177.239   TCP       184 443 → 50604 [PSH, ACK] Seq=1251 Ack=1934 Win=19712 Len=130 [TCP PDU reassembled in 104691]
1046… 101.697655  163.70.143.35     192.168.177.239   TLSv1.3   685 Application Data
1046… 101.697978  192.168.177.239   163.70.143.35     TLSv1.3   118 Change Cipher Spec, Application Data
1046… 101.706341  163.70.143.35     192.168.177.239   TLSv1.3   233 Application Data
1046… 101.706341  163.70.143.35     192.168.177.239   TLSv1.3   128 Application Data
1046… 101.804043  163.70.143.35     192.168.177.239   TLSv1.3  1147 Application Data, Application Data
```

Fig 7: PUSH flag filter.

Now, generate the protocol hierarchy of the same.

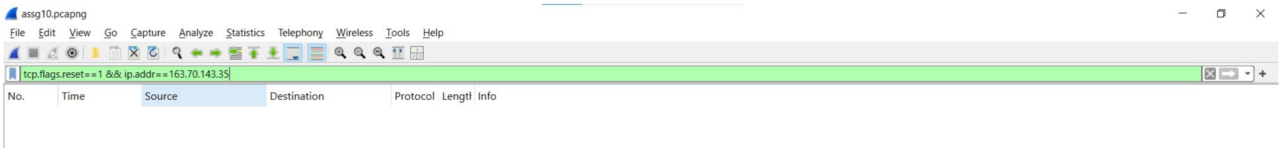


Wireshark · Protocol Hierarchy Statistics · assg10.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s | PDUs |
|---|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 17 | 100.0 | 6308 | 136 k | 0 | 0 | 0 | 17 |
| Ethernet | 100.0 | 17 | 3.8 | 238 | 5156 | 0 | 0 | 0 | 17 |
| Internet Protocol Version 4 | 100.0 | 17 | 5.4 | 340 | 7366 | 0 | 0 | 0 | 17 |
| Transmission Control Protocol | 100.0 | 17 | 5.4 | 340 | 7366 | 1 | 20 | 433 | 17 |
| Transport Layer Security | 94.1 | 16 | 165.4 | 10436 | 226 k | 16 | 6920 | 149 k | 17 |

Fig 8: PUSH flag protocol hierarchy.

Repeat the above steps with reset flag filter also. The output is shown in the following figure.

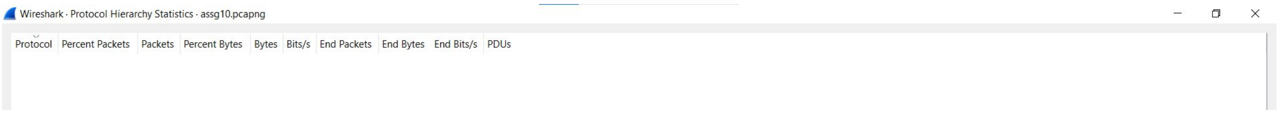


assg10.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.reset==1 && ip.addr==163.70.143.35

| No. | Time | Source | Destination | Protocol | Lengtl | Info |
|---|---|---|---|---|---|---|

Fig 9: Reset flag filter.



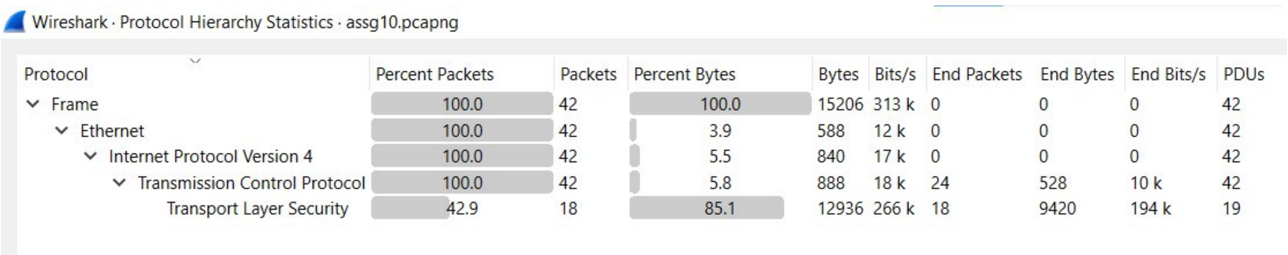
Wireshark · Protocol Hierarchy Statistics · assg10.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s | PDUs |
|---|---|---|---|---|---|---|---|---|---|

Fig 10: RESET flag protocol hierarchy.

4. Count how many TCP packets you received from / sent to Facebook, and how many of each were also HTTP packets.

Now, simply apply the Facebook IP and tcp filter and check the protocol hierarchy status. This will tell the number of packets sent and received from Facebook. The output is shown in the below figure.

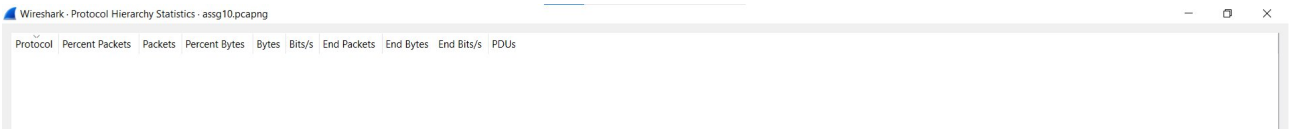


Wireshark · Protocol Hierarchy Statistics · assg10.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s | PDUs |
|---|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 42 | 100.0 | 15206 | 313 k | 0 | 0 | 0 | 42 |
| Ethernet | 100.0 | 42 | 3.9 | 588 | 12 k | 0 | 0 | 0 | 42 |
| Internet Protocol Version 4 | 100.0 | 42 | 5.5 | 840 | 17 k | 0 | 0 | 0 | 42 |
| Transmission Control Protocol | 100.0 | 42 | 5.8 | 888 | 18 k | 24 | 528 | 10 k | 42 |
| Transport Layer Security | 42.9 | 18 | 85.1 | 12936 | 266 k | 18 | 9420 | 194 k | 19 |

Fig 11 : TCP Packet Count

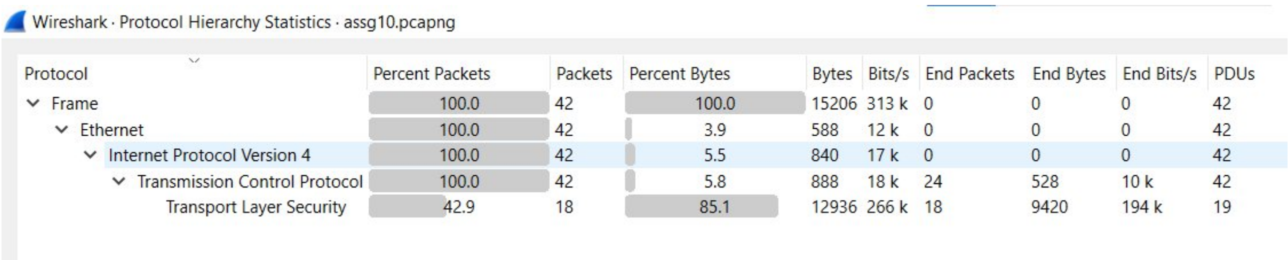
For the count of http packets apply the Facebook IP and http filter and check the protocol hierarchy status. This will tell the number of packets sent and received from Facebook. The output is shown in the below figure.



Wireshark · Protocol Hierarchy Statistics · assg10.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s | PDUs |
|---|---|---|---|---|---|---|---|---|---|

Fig 12: HTTP packet count.



Wireshark · Protocol Hierarchy Statistics · assg10.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s | PDUs |
|---|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 42 | 100.0 | 15206 | 313 k | 0 | 0 | 0 | 42 |
| Ethernet | 100.0 | 42 | 3.9 | 588 | 12 k | 0 | 0 | 0 | 42 |
| Internet Protocol Version 4 | 100.0 | 42 | 5.5 | 840 | 17 k | 0 | 0 | 0 | 42 |
| Transmission Control Protocol | 100.0 | 42 | 5.8 | 888 | 18 k | 24 | 528 | 10 k | 42 |
| Transport Layer Security | 42.9 | 18 | 85.1 | 12936 | 266 k | 18 | 9420 | 194 k | 19 |

Fig 13: HTTPS packet count.

## Conclusion

Wireshark successfully captured and filtered Facebook login traffic. Using filters, we analyzed TCP packets with SYN, PSH, RST flags and compared the number of TCP vs HTTP packets exchanged.