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

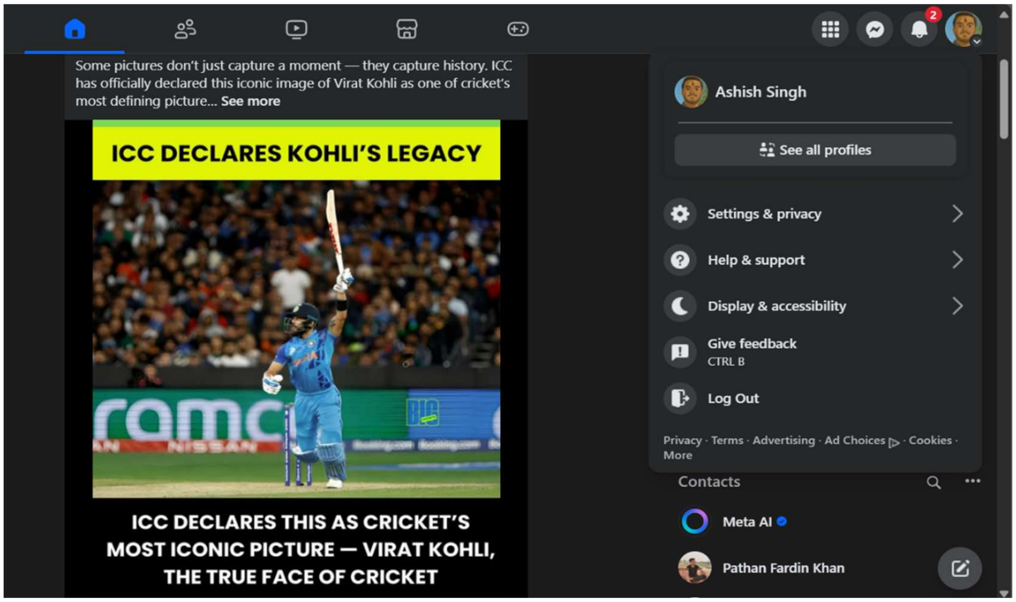
Wireshark is a widely used network protocol analyzer that captures and inspects packets in real time. It allows users to apply **capture filters** (to restrict packets being recorded) and **display filters** (to analyze specific traffic after capture).

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.

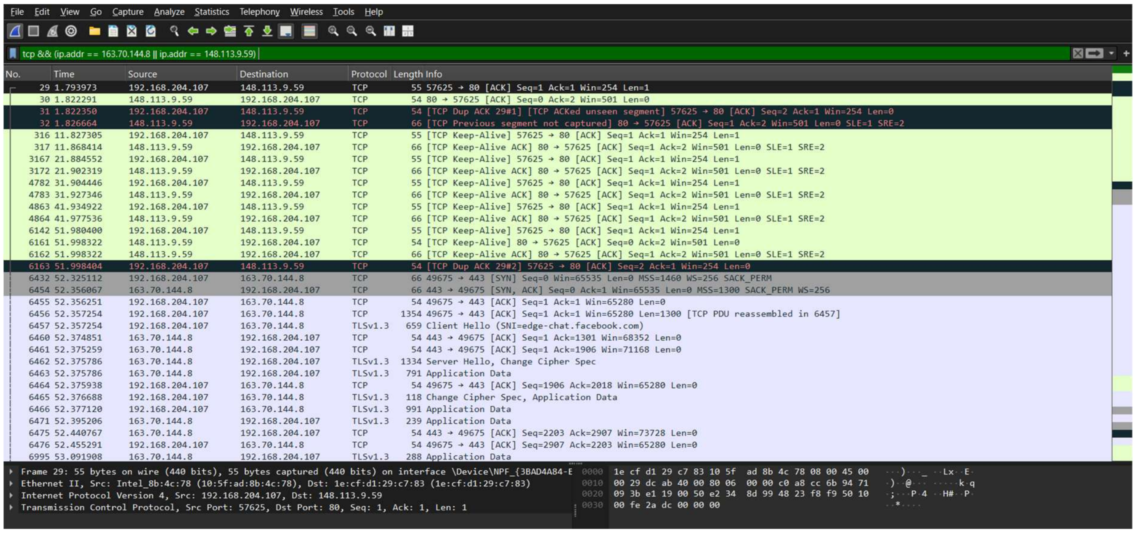


Some pictures don't just capture a moment — they capture history. ICC has officially declared this iconic image of Virat Kohli as one of cricket's most defining picture... See more

ICC DECLARES KOHLI'S LEGACY

ICC DECLARES THIS AS CRICKET'S MOST ICONIC PICTURE — VIRAT KOHLI, THE TRUE FACE OF CRICKET

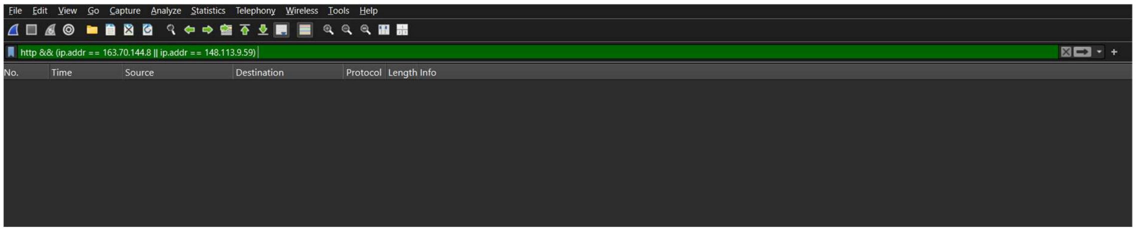
Fig 1: Facebook login page



```
tcp && (ip.addr == 163.70.144.8 || ip.addr == 148.113.9.59)

No.   Time        Source            Destination      Protocol Length Info
   29 1.793973    192.168.204.107   148.113.9.59     TCP        55 57625 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
   30 1.822291    148.113.9.59      192.168.204.107  TCP        54 80 → 57625 [ACK] Seq=0 Ack=2 Win=501 Len=0
   31 1.822350    192.168.204.107   148.113.9.59     TCP        54 [TCP Dup ACK 29#1] [TCP ACKed unseen segment] 57625 → 80 [ACK] Seq=2 Ack=1 Win=254 Len=0
   32 1.826664    148.113.9.59      192.168.204.107  TCP        66 [TCP Previous segment not captured] 80 → 57625 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
  316 11.827305   192.168.204.107   148.113.9.59     TCP        55 [TCP Keep-Alive] 57625 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
  317 11.868414   148.113.9.59      192.168.204.107  TCP        66 [TCP Keep-Alive ACK] 80 → 57625 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
 3167 21.884552   192.168.204.107   148.113.9.59     TCP        55 [TCP Keep-Alive] 57625 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
 3172 21.902319   148.113.9.59      192.168.204.107  TCP        66 [TCP Keep-Alive ACK] 80 → 57625 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
 4782 31.904446   192.168.204.107   148.113.9.59     TCP        55 [TCP Keep-Alive] 57625 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
 4783 31.927346   148.113.9.59      192.168.204.107  TCP        66 [TCP Keep-Alive ACK] 80 → 57625 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
 4863 41.934922   192.168.204.107   148.113.9.59     TCP        55 [TCP Keep-Alive] 57625 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
 4864 41.977536   148.113.9.59      192.168.204.107  TCP        66 [TCP Keep-Alive ACK] 80 → 57625 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
 6142 51.980400   192.168.204.107   148.113.9.59     TCP        55 [TCP Keep-Alive] 57625 → 80 [ACK] Seq=1 Ack=1 Win=254 Len=1
 6161 51.998322   148.113.9.59      192.168.204.107  TCP        54 [TCP Keep-Alive] 80 → 57625 [ACK] Seq=0 Ack=2 Win=501 Len=0
 6162 51.998322   148.113.9.59      192.168.204.107  TCP        66 [TCP Keep-Alive ACK] 80 → 57625 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
 6163 51.998404   192.168.204.107   148.113.9.59     TCP        54 [TCP Dup ACK 29#2] 57625 → 80 [ACK] Seq=2 Ack=1 Win=254 Len=0
 6432 52.325112   192.168.204.107   163.70.144.8     TCP        66 49675 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
 6454 52.356067   163.70.144.8      192.168.204.107  TCP        66 443 → 49675 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 SACK_PERM WS=256
 6455 52.356251   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
 6456 52.357254   192.168.204.107   163.70.144.8     TCP      1354 49675 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=1300 [TCP PDU reassembled in 6457]
 6457 52.357254   192.168.204.107   163.70.144.8     TLSv1.3   659 Client Hello (SNI=edge-chat.facebook.com)
 6460 52.374851   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=1 Ack=1301 Win=68352 Len=0
 6461 52.375259   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=1 Ack=1906 Win=71168 Len=0
 6462 52.375786   163.70.144.8      192.168.204.107  TLSv1.3  1334 Server Hello, Change Cipher Spec
 6463 52.375786   163.70.144.8      192.168.204.107  TLSv1.3   791 Application Data
 6464 52.375938   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=1906 Ack=2018 Win=65280 Len=0
 6465 52.376688   192.168.204.107   163.70.144.8     TLSv1.3   118 Change Cipher Spec, Application Data
 6466 52.377120   192.168.204.107   163.70.144.8     TLSv1.3   991 Application Data
 6471 52.395206   163.70.144.8      192.168.204.107  TLSv1.3   239 Application Data
 6475 52.440767   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=2203 Ack=2907 Win=73728 Len=0
 6476 52.455291   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=2907 Ack=2203 Win=65280 Len=0
 6995 53.091908   163.70.144.8      192.168.204.107  TLSv1.3   288 Application Data

› Frame 29: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{3BAD4A84-E
› Ethernet II, Src: Intel_8b:4c:78 (10:5f:ad:8b:4c:78), Dst: 1e:cf:d1:29:c7:83 (1e:cf:d1:29:c7:83)
› Internet Protocol Version 4, Src: 192.168.204.107, Dst: 148.113.9.59
› Transmission Control Protocol, Src Port: 57625, Dst Port: 80, Seq: 1, Ack: 1, Len: 1

0000  1e cf d1 29 c7 83 10 5f  ad 8b 4c 78 08 00 45 00   ···)···_ ··Lx··E·
0010  00 29 dc ab 40 00 80 06  00 00 c0 a8 cc 6b 94 71   ·)··@··· ·····k·q
0020  09 3b e1 19 00 50 e2 34  8d 99 48 23 f8 f9 50 10   ·;···P·4 ··H#··P·
0030  00 fe 2a dc 00 00 00                               ··*····
```

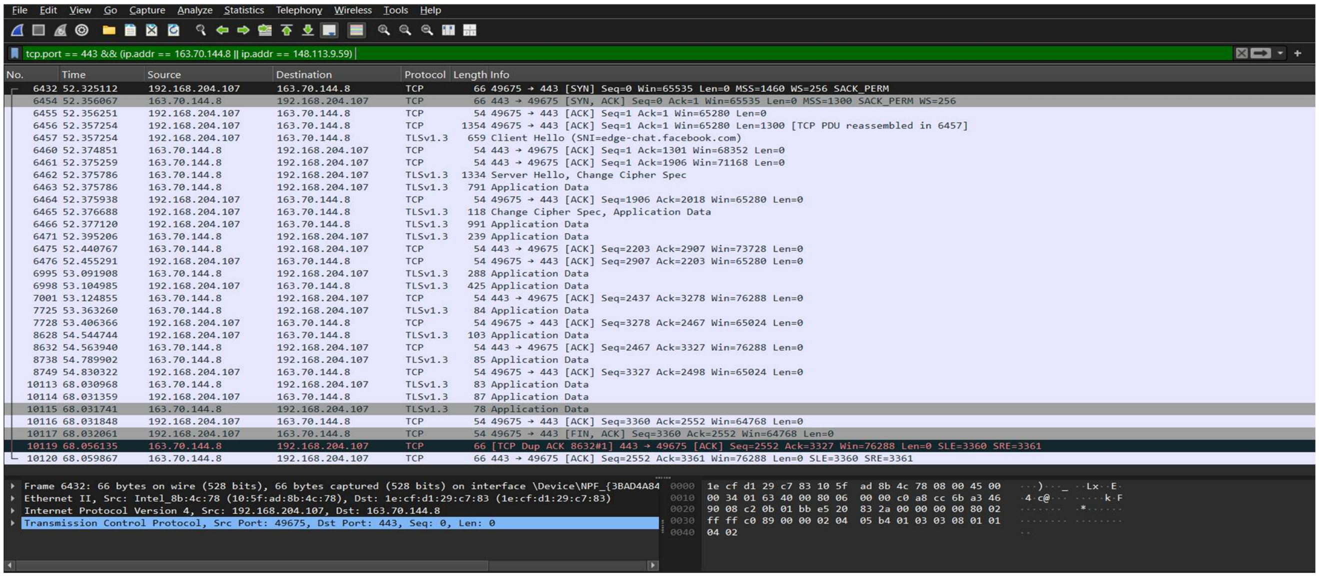
Fig 2: Facebook ip & tcp filter

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.



```
http && (ip.addr == 163.70.144.8 || ip.addr == 148.113.9.59)

No.   Time        Source            Destination      Protocol Length Info
```

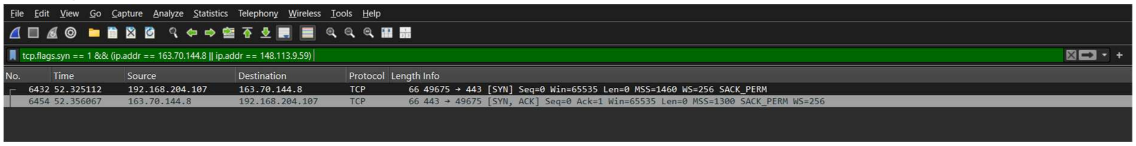
Fig 3: packets with http filter.



```
tcp.port == 443 && (ip.addr == 163.70.144.8 || ip.addr == 148.113.9.59)

No.    Time        Source            Destination      Protocol Length Info
  6432 52.325112   192.168.204.107   163.70.144.8     TCP        66 49675 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  6454 52.356067   163.70.144.8      192.168.204.107  TCP        66 443 → 49675 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 SACK_PERM WS=256
  6455 52.356251   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
  6456 52.357254   192.168.204.107   163.70.144.8     TCP      1354 49675 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=1300 [TCP PDU reassembled in 6457]
  6457 52.357254   192.168.204.107   163.70.144.8     TLSv1.3   659 Client Hello (SNI=edge-chat.facebook.com)
  6460 52.374851   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=1 Ack=1301 Win=68352 Len=0
  6461 52.375259   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=1 Ack=1906 Win=71168 Len=0
  6462 52.375786   163.70.144.8      192.168.204.107  TLSv1.3  1334 Server Hello, Change Cipher Spec
  6463 52.375786   163.70.144.8      192.168.204.107  TLSv1.3   791 Application Data
  6464 52.375938   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=1906 Ack=2018 Win=65280 Len=0
  6465 52.376688   192.168.204.107   163.70.144.8     TLSv1.3   118 Change Cipher Spec, Application Data
  6466 52.377120   192.168.204.107   163.70.144.8     TLSv1.3   991 Application Data
  6471 52.395206   163.70.144.8      192.168.204.107  TLSv1.3   239 Application Data
  6475 52.440767   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=2203 Ack=2907 Win=73728 Len=0
  6476 52.455291   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=2907 Ack=2203 Win=65280 Len=0
  6995 53.091908   163.70.144.8      192.168.204.107  TLSv1.3   288 Application Data
  6998 53.104985   192.168.204.107   163.70.144.8     TLSv1.3   425 Application Data
  7001 53.124855   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=2437 Ack=3278 Win=76288 Len=0
  7725 53.363260   163.70.144.8      192.168.204.107  TLSv1.3    84 Application Data
  7728 53.406366   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=3278 Ack=2467 Win=65024 Len=0
  8628 54.544744   192.168.204.107   163.70.144.8     TLSv1.3   103 Application Data
  8632 54.563940   163.70.144.8      192.168.204.107  TCP        54 443 → 49675 [ACK] Seq=2467 Ack=3327 Win=76288 Len=0
  8738 54.789902   163.70.144.8      192.168.204.107  TLSv1.3    85 Application Data
  8749 54.830322   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=3327 Ack=2498 Win=65024 Len=0
 10113 68.030968   163.70.144.8      192.168.204.107  TLSv1.3    83 Application Data
 10114 68.031359   192.168.204.107   163.70.144.8     TLSv1.3    87 Application Data
 10115 68.031741   163.70.144.8      192.168.204.107  TLSv1.3    78 Application Data
 10116 68.031848   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [ACK] Seq=3360 Ack=2552 Win=64768 Len=0
 10117 68.032061   192.168.204.107   163.70.144.8     TCP        54 49675 → 443 [FIN, ACK] Seq=3360 Ack=2552 Win=64768 Len=0
 10119 68.056135   163.70.144.8      192.168.204.107  TCP        66 [TCP Dup ACK 8632#1] 443 → 49675 [ACK] Seq=2552 Ack=3327 Win=76288 Len=0 SLE=3360 SRE=3361
 10120 68.059867   163.70.144.8      192.168.204.107  TCP        66 443 → 49675 [ACK] Seq=2552 Ack=3361 Win=76288 Len=0 SLE=3360 SRE=3361

› Frame 6432: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{3BAD4A84
› Ethernet II, Src: Intel_8b:4c:78 (10:5f:ad:8b:4c:78), Dst: 1e:cf:d1:29:c7:83 (1e:cf:d1:29:c7:83)
› Internet Protocol Version 4, Src: 192.168.204.107, Dst: 163.70.144.8
› Transmission Control Protocol, Src Port: 49675, Dst Port: 443, Seq: 0, Len: 0

0000  1e cf d1 29 c7 83 10 5f  ad 8b 4c 78 08 00 45 00   ···)···_ ··Lx··E·
0010  00 34 01 63 40 00 80 06  00 00 c0 a8 cc 6b a3 46   ·4·c@··· ·····k·F
0020  90 08 c2 0b 01 bb e5 20  83 2a 00 00 00 00 80 02   ········ ·*······
0030  ff ff c0 89 00 00 02 04  05 b4 01 03 03 08 01 01   ········ ········
0040  04 02                                              ··
```

Fig 4: packets with https filter.

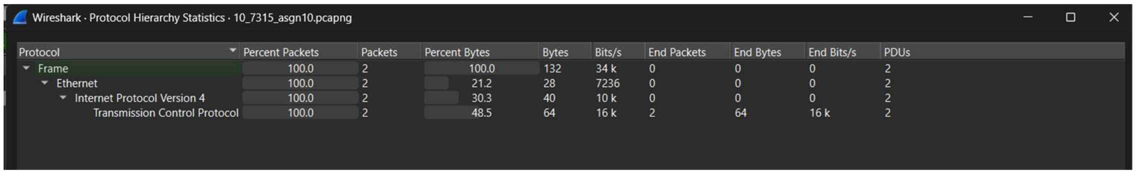
3. **Write a DISPLAY filter expression to count all TCP packets that have the flags SYN, PSH, and RST set. Show the fraction of packets that had each flag set.**
    - Now, apply the filter of Facebook IP and tcp.flag.syn == 1. The output is shown in the following figure.



```
tcp.flags.syn == 1 && (ip.addr == 163.70.144.8 || ip.addr == 148.113.9.59)

No.   Time        Source            Destination      Protocol Length Info
 6432 52.325112   192.168.204.107   163.70.144.8     TCP        66 49675 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
 6454 52.356067   163.70.144.8      192.168.204.107  TCP        66 443 → 49675 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 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.

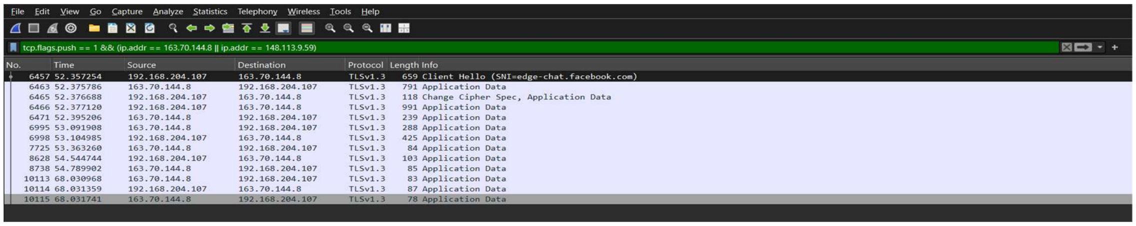


Wireshark · Protocol Hierarchy Statistics · 10_7315_asgn10.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	2	100.0	132	34 k	0	0	0	2
Ethernet	100.0	2	21.2	28	7236	0	0	0	2
Internet Protocol Version 4	100.0	2	30.3	40	10 k	0	0	0	2
Transmission Control Protocol	100.0	2	48.5	64	16 k	2	64	16 k	2

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.

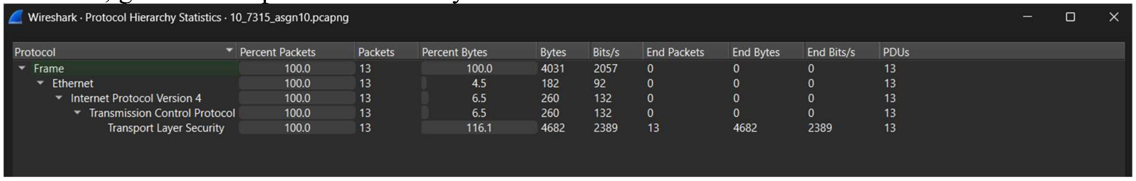


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.flags.push == 1 && (ip.addr == 163.70.144.8 || ip.addr == 148.113.9.59)

No.    Time       Source           Destination      Protocol Length Info
 6457 52.357254  192.168.204.107  163.70.144.8     TLSv1.3   659 Client Hello (SNI=edge-chat.facebook.com)
 6463 52.375786  163.70.144.8     192.168.204.107  TLSv1.3   791 Application Data
 6465 52.376688  192.168.204.107  163.70.144.8     TLSv1.3   118 Change Cipher Spec, Application Data
 6466 52.377120  192.168.204.107  163.70.144.8     TLSv1.3   991 Application Data
 6471 52.395206  163.70.144.8     192.168.204.107  TLSv1.3   239 Application Data
 6995 53.091908  163.70.144.8     192.168.204.107  TLSv1.3   288 Application Data
 6998 53.104985  192.168.204.107  163.70.144.8     TLSv1.3   425 Application Data
 7725 53.363260  163.70.144.8     192.168.204.107  TLSv1.3    84 Application Data
 8628 54.544744  192.168.204.107  163.70.144.8     TLSv1.3   103 Application Data
 8738 54.789902  163.70.144.8     192.168.204.107  TLSv1.3    85 Application Data
10113 68.030968  163.70.144.8     192.168.204.107  TLSv1.3    83 Application Data
10114 68.031359  192.168.204.107  163.70.144.8     TLSv1.3    87 Application Data
10115 68.031741  163.70.144.8     192.168.204.107  TLSv1.3    78 Application Data
```

Fig 7: PUSH flag filter.

- Now, generate the protocol hierarchy of the same.

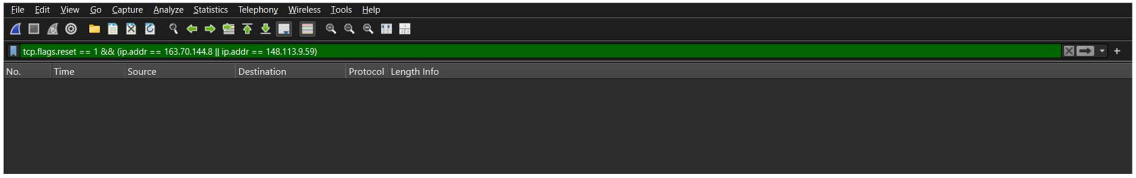


Wireshark · Protocol Hierarchy Statistics · 10_7315_asgn10.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	13	100.0	4031	2057	0	0	0	13
Ethernet	100.0	13	4.5	182	92	0	0	0	13
Internet Protocol Version 4	100.0	13	6.5	260	132	0	0	0	13
Transmission Control Protocol	100.0	13	6.5	260	132	0	0	0	13
Transport Layer Security	100.0	13	116.1	4682	2389	13	4682	2389	13

Fig 8: PUSH flag protocol hierarchy.

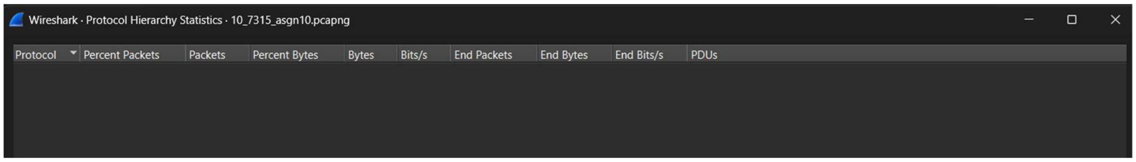
- Repeat the above steps with reset flag filter also. The output is shown in the following figure.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.flags.reset == 1 && (ip.addr == 163.70.144.8 || ip.addr == 148.113.9.59)

No.    Time       Source           Destination      Protocol Length Info
```

Fig 9: Reset flag filter.

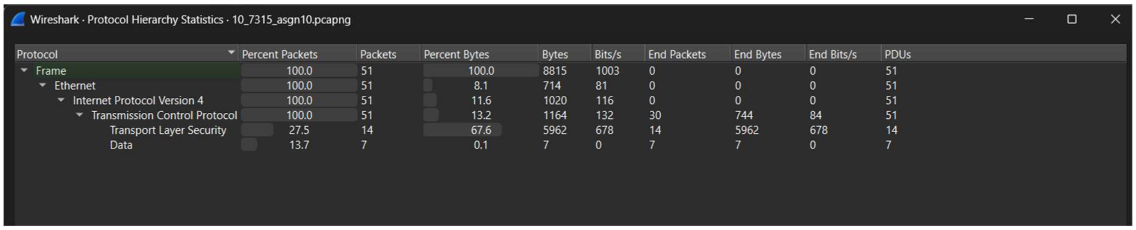


Wireshark · Protocol Hierarchy Statistics · 10_7315_asgn10.pcapng

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.

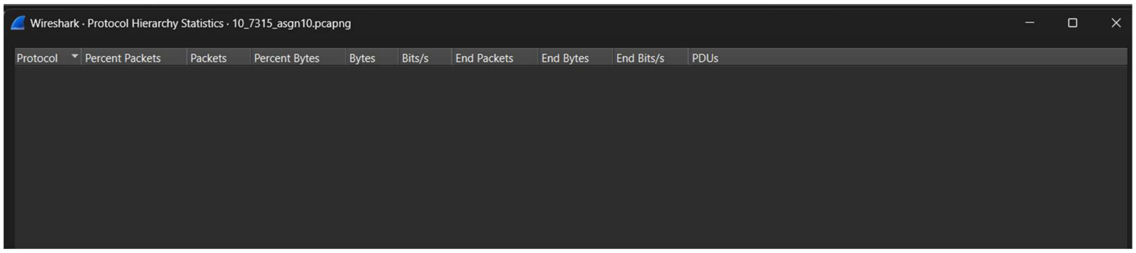


Wireshark · Protocol Hierarchy Statistics · 10_7315_asgn10.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	51	100.0	8815	1003	0	0	0	51
Ethernet	100.0	51	8.1	714	81	0	0	0	51
Internet Protocol Version 4	100.0	51	11.6	1020	116	0	0	0	51
Transmission Control Protocol	100.0	51	13.2	1164	132	30	744	84	51
Transport Layer Security	27.5	14	67.6	5962	678	14	5962	678	14
Data	13.7	7	0.1	7	0	7	7	0	7

Fig 11: TCP packet count.

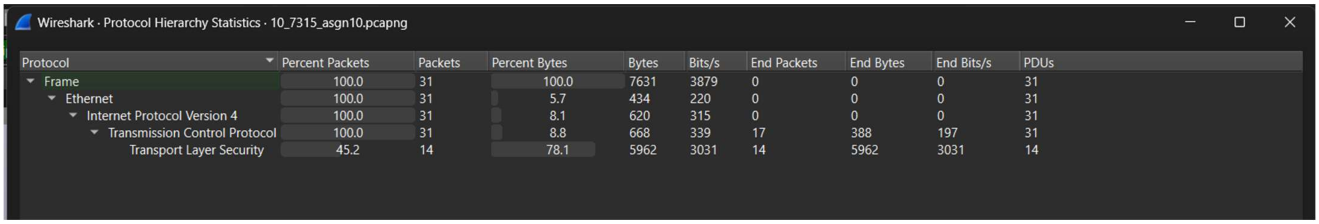
For the count of http packets apply the Facebook IP and http filter and check the protocol hierarchy status. This will tell the number of packets sent and received from Facebook. The output is shown in the below figure.



Wireshark · Protocol Hierarchy Statistics · 10_7315_asgn10.pcapng

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

Fig 10: HTTP packet count.



Wireshark · Protocol Hierarchy Statistics · 10_7315_asgn10.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	31	100.0	7631	3879	0	0	0	31
Ethernet	100.0	31	5.7	434	220	0	0	0	31
Internet Protocol Version 4	100.0	31	8.1	620	315	0	0	0	31
Transmission Control Protocol	100.0	31	8.8	668	339	17	388	197	31
Transport Layer Security	45.2	14	78.1	5962	3031	14	5962	3031	14

Fig 11: 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.