

## Wireshark Practical

### Practical no 10

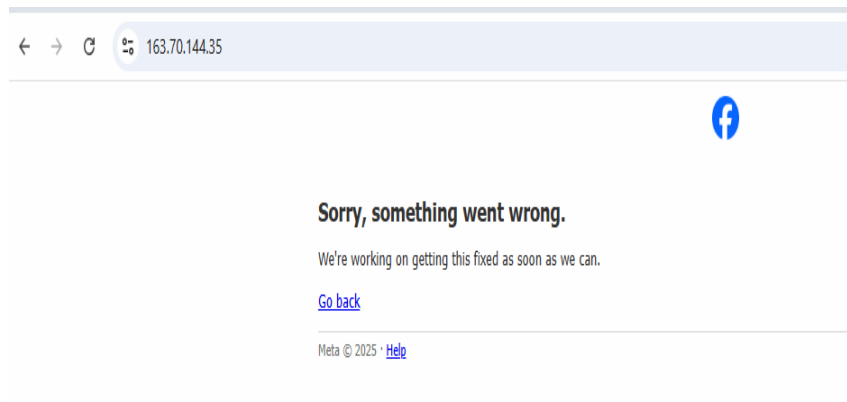
A. Capture all TCP traffic to/from Facebook, during the time when you log in to your Facebook account

MUST Create your own Facebook account

1. Open Wireshark on your system.
2. Select the active network interface (Ethernet or Wi-Fi).
3. Once selected, Wireshark will start capturing packets.
4. Log in to your Facebook account only after Wireshark is running.
5. Open the Command Prompt and run:

**ping facebook.com this will give you Facebook address  
(eg for my machine 163.70.144.35 / 157.240.237.35)**

6. In Wireshark, apply a filter to capture only Facebook packets:  
**tcp && ip.addr == 163.70.144.35 (Facebook id)**
7. To verify, open your browser and enter the same IP address. If it opens Facebook, you know the captured packets belong to Facebook.



B. Capture all HTTP traffic to/from Facebook (other website), when you log in to your Facebook account

1. 1<sup>st</sup> you need to open http website like techpanda.org
2. you have to enter any sample email & enter password and click on submit

## Login | Personal Contacts Manager v1.0

Email\*

abc@gmail.com

Password\*

...

☐ Remember me

Submit

3. You will get

## Dashboard | Personal Contacts Manager v1.0

Add New Contact

Log Out

ID	First Name	Last Name	Mobile No	Email	Actions
1	mynams	jenefry	9898989898	admin@gmail.com	
99213	asfe	liihl	13230842384	admin@example.com	<a href="#">Edit</a>
99214	shreya	shinde	8999235684	shrreyanb@gmail.com	<a href="#">Edit</a>
99215	Bhupiiindra	JOGIIIII	828282828282	bhupjaaaaa2@jaa2.com	<a href="#">Edit</a>
99216	Sandesh	Pawar	9999999999	sandesh@snapchat.com	<a href="#">Edit</a>
99217	Yash	Pawar	9658741230	yash16@gmail.com	<a href="#">Edit</a>
99218	protagonist	netent	946987582315	admin@google.com	<a href="#">Edit</a>
99219	anurag	Satav	0000000000	admin@google.com	<a href="#">Edit</a>
99220	j	s	123	j@gmail.com	<a href="#">Edit</a>
99221	jy	sh	456	js@gmail.com	<a href="#">Edit</a>
99222	Jack	Sparrow	0987654321	Captain@gmail.com	<a href="#">Edit</a>
99223	ksjbdck	sdvcS	65846345435	abc@gmail.com	<a href="#">Edit</a>
99224	Jaaxfgnb rmtdgsncv wswjodw1	aq	zsGTRY	hiray@info.com	<a href="#">Edit</a>
99225	tgfse	Patel	73839903	sheikhnamra42@gmail.com	<a href="#">Edit</a>
99226	soooo	hammmm	5465891265	admin@google.com	<a href="#">Edit</a>
99227	<a href="#">Dark</a>	lewis	1234567	admin@xyz.com	<a href="#">Edit</a>

Total Records Count: 16

3. Open cmd and give command and ping techpanda.org then you will get connectivity with

5. `http && ip.addr==10.30.74.132 (Machine IP) && ip.addr==271.174.153.52 (techpanda.org ip) (imp command )`

The image shows a Wireshark packet capture window titled "Capturing from Ethernet". The filter bar at the top shows the filter: `http && ip.addr==10.30.74.132 && ip.addr==10.30.74.132`. The packet list on the left shows three packets:

No.	Time	Source	Destination	Protocol	Length	Info
2085	42.135562	10.30.74.132	142.251.228.35	HTTP	289	GET /wr2/HPiWUDBOMewSJA3BgUrDgKCGuABRRtQSE18EXKz8bYUTXd8%2ByhD3s1zQU3hse7XKvID43JPHhu%2Bw8Qw1CsJACEQC3vQda5m%2BKaq0Lc1ab8Qq HTTP/1.1
2087	42.139420	142.251.228.35	10.30.74.132	OCSP	1169	Response
11782	259.813218	10.30.74.132	142.251.228.35	HTTP	291	GET /wr2/HPewTzBNPEswSTA3BgUrDgKCGuABRRtQSE18EXKz8bYUTXd8%2ByhD3s1zQU3hse7XKvID43JPHhu%2Bw8Qw1CsJACEAGarvyxh%2F11EwtRvkToUg%3D HTTP/1.1
11787	259.821116	142.251.228.35	10.30.74.132	OCSP	1167	Response

The packet details pane on the left shows the selected packet (No. 2085) with the following details:

- Frame 2085: 289 bytes on wire (2312 bits), 289 bytes captured (2312 bits) on interface \Device\NPF\_{6E1080F6-7...}
- Ethernet II, Src: Dell\_2a:54:f2 (74:86:e2:2a:54:f2), Dst: JuniperNetwo\_0d:6b:c0 (78:50:7c:0d:6b:c0)
- Internet Protocol Version 4, Src: 10.30.74.132, Dst: 142.251.228.35
- Transmission Control Protocol, Src Port: 54400, Dst Port: 80, Seq: 1, Ack: 1, Len: 235
- Hypertext Transfer Protocol

The packet bytes pane on the right shows the raw data of the selected packet, including the HTTP request line and headers.

3 & 4. 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.

tcp.flags.syn==1

The screenshot shows the Wireshark interface with the packet capture filter `tcp.flags.syn==1` applied. The packet list displays 20 captured packets, all of which are TCP SYN packets. The packet details pane shows the structure of a TCP packet, including the Ethernet II header, Internet Protocol Version 4 header, and Transmission Control Protocol header. The packet bytes pane shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
27893	192.738736	10.30.73.147	10.30.75.79	TCP	66	13111 → 65002 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
27895	192.731628	10.30.73.147	10.30.75.79	TCP	66	13111 → 65003 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
27101	192.740135	10.30.75.79	10.30.73.147	TCP	66	65004 → 13111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
27102	192.740928	10.30.75.79	10.30.73.147	TCP	66	65005 → 13111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
27103	192.740937	10.30.73.147	10.30.75.79	TCP	66	13111 → 65004 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
27105	192.742967	10.30.73.147	10.30.75.79	TCP	66	13111 → 65005 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
27339	193.578225	10.30.75.79	23.193.114.33	TCP	66	65006 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
27341	193.582429	23.193.114.33	10.30.75.79	TCP	66	443 → 65006 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1384 SACK_PERM WS=128
27520	195.347376	10.30.75.79	142.251.42.46	TCP	66	65007 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
27524	195.372378	142.251.42.46	10.30.75.79	TCP	66	443 → 65007 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
28009	201.999064	10.30.75.79	10.10.128.78	TCP	66	65008 → 13000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
28010	201.999487	10.10.128.78	10.30.75.79	TCP	66	13000 → 65008 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
28798	203.061705	10.30.75.79	142.251.222.67	TCP	66	65009 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
28799	203.067766	142.251.222.67	10.30.75.79	TCP	66	443 → 65009 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
33283	237.056247	10.30.75.79	23.206.173.50	TCP	66	65010 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
33285	237.060826	23.206.173.50	10.30.75.79	TCP	66	443 → 65010 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1384 SACK_PERM WS=128
34004	248.031278	10.30.75.79	10.30.73.147	TCP	66	65011 → 13111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
34006	248.035137	10.30.73.147	10.30.75.79	TCP	66	13111 → 65011 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
37597	262.157665	10.30.75.79	10.10.128.78	TCP	66	65012 → 13000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
37598	262.158045	10.10.128.78	10.30.75.79	TCP	66	13000 → 65012 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

tcp.flags.push == 1

Wireshark packet capture showing TCP flags push == 1. The packet list shows several application data packets with the push flag set. The packet details pane shows the structure of a TCP segment.

No.	Time	Source	Destination	Protocol	Length	Info
1176	18.898922	142.250.183.78	10.30.75.79	TLSv1.2	127	Application Data
1181	18.957413	157.240.237.60	10.30.75.79	TLSv1.2	278	Application Data
1182	18.981544	10.30.75.79	157.240.237.60	TLSv1.2	127	Application Data
1239	20.477933	142.250.183.78	10.30.75.79	TLSv1.2	127	Application Data
1245	20.579404	142.251.42.238	10.30.75.79	TLSv1.2	127	Application Data
1262	20.996408	142.250.70.46	10.30.75.79	TLSv1.2	336	Application Data
1263	20.996408	142.250.70.46	10.30.75.79	TLSv1.2	93	Application Data
1265	21.001885	10.30.75.79	142.250.70.46	TLSv1.2	89	Application Data
1266	21.001978	10.30.75.79	142.250.70.46	TLSv1.2	93	Application Data
1267	21.005488	10.30.75.79	142.250.70.46	TLSv1.2	4673	Application Data
1273	21.024414	10.30.75.79	10.30.73.147	TCP	432	64910 → 13111 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=378
1278	21.103943	10.30.73.147	10.30.75.79	TCP	189	13111 → 64910 [PSH, ACK] Seq=1 Ack=379 Win=65824 Len=135
1287	21.313379	142.250.70.46	10.30.75.79	TLSv1.2	120	Application Data
1288	21.313379	142.250.70.46	10.30.75.79	TLSv1.2	122	Application Data
1292	21.402823	157.240.237.60	10.30.75.79	TLSv1.2	277	Application Data
1293	21.406434	10.30.75.79	157.240.237.60	TLSv1.2	127	Application Data
1302	21.567908	10.30.75.79	10.10.128.78	TLSv1.2	444	Client Hello (SNI=sdcc-pc-06.unitech.local)
1306	21.572452	10.10.128.78	10.30.75.79	TLSv1.2	616	Certificate, Server Key Exchange, Certificate Request, Server Hello Done
1308	21.576508	10.30.75.79	10.10.128.78	TLSv1.2	192	Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1309	21.577559	10.10.128.78	10.30.75.79	TLSv1.2	208	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

tcp.flags.reset == 1

Wireshark packet capture showing TCP flags reset == 1. The packet list shows several RST packets. The packet details pane shows the structure of a TCP segment.

No.	Time	Source	Destination	Protocol	Length	Info
3480	53.486436	23.206.173.42	10.30.75.79	TCP	60	443 → 64864 [RST] Seq=1 Win=0 Len=0
5258	63.872129	150.171.69.254	10.30.75.79	TCP	60	443 → 64845 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5839	72.775038	150.171.27.10	10.30.75.79	TCP	60	443 → 64857 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
6119	78.121569	172.217.174.228	10.30.75.79	TCP	60	443 → 64913 [RST] Seq=4102 Win=0 Len=0
6206	78.810400	150.171.28.11	10.30.75.79	TCP	60	443 → 64868 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
6230	78.918870	40.126.17.135	10.30.75.79	TCP	60	443 → 64880 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
6375	80.642136	20.190.146.33	10.30.75.79	TCP	60	443 → 64881 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
12289	88.389053	204.79.197.222	10.30.75.79	TCP	60	443 → 64843 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
15711	107.852728	142.250.192.35	10.30.75.79	TCP	60	88 → 64811 [RST] Seq=2 Win=0 Len=0
15735	108.617971	150.171.27.11	10.30.75.79	TCP	60	443 → 64904 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
17696	116.210655	150.171.27.11	10.30.75.79	TCP	60	443 → 64889 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
18079	120.600529	79.133.169.96	10.30.75.79	TCP	60	443 → 64905 [RST, ACK] Seq=3 Ack=4 Win=64256 Len=0
18112	120.726024	10.30.75.79	79.133.170.48	TCP	54	64964 → 443 [RST, ACK] Seq=4 Ack=3 Win=0 Len=0
18133	120.940860	79.133.170.48	10.30.75.79	TCP	60	443 → 64964 [RST, ACK] Seq=3 Ack=4 Win=64256 Len=0
18562	121.307104	79.133.169.96	10.30.75.79	TCP	60	443 → 64971 [RST, ACK] Seq=3 Ack=4 Win=64237 Len=0
18591	121.722457	79.133.170.48	10.30.75.79	TCP	60	443 → 64970 [RST, ACK] Seq=3 Ack=4 Win=64236 Len=0
24782	163.318139	10.30.75.79	185.201.2.39	TCP	54	64487 → 443 [RST, ACK] Seq=27 Ack=1 Win=0 Len=0
27757	190.120780	10.30.75.79	82.202.184.184	TCP	54	64486 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
40429	294.821247	10.30.75.79	157.240.237.2	TCP	54	65018 → 443 [RST, ACK] Seq=1772 Ack=4230 Win=0 Len=0
40701	297.111032	10.30.75.79	23.206.173.50	TCP	54	65010 → 443 [RST, ACK] Seq=598 Ack=4716 Win=0 Len=0