Wireshark and TShark TCP Commands

1. Basic TCP Commands in Wireshark:

- Show all TCP packets:

```
tcp
```

- Show TCP packets to/from a specific port:

```
tcp.port == 80  # For HTTP traffic (port 80)
```

- Show packets with specific source IP and TCP port:

```
ip.src == 192.168.1.1 && tcp.port == 443  # For HTTPS traffic (port 443)
```

- Show TCP packets with specific flags:

  - SYN (used for the connection handshake):

    ```
    tcp.flags.syn == 1 && tcp.flags.ack == 0
    ```

  - ACK (used for acknowledgment):

    ```
    tcp.flags.ack == 1
    ```

  - FIN (used for connection termination):

    ```
    tcp.flags.fin == 1
    ```

  - RST (reset the connection):

    ```
    tcp.flags.reset == 1
    ```

- Show TCP packets with a specific sequence number:

```
tcp.seq == 12345  # Replace with your sequence number
```

- Show TCP packets with a specific acknowledgment number:

```
tcp.ack == 67890  # Replace with your acknowledgment number
```

- Show TCP stream (e.g., for HTTP requests/responses or any conversation over TCP):

```
tcp.stream eq 0  # Filter packets belonging to the first TCP stream
```

- Show TCP retransmissions:

```
tcp.analysis.retransmission
```

- Show TCP out-of-order packets:

```
tcp.analysis.out_of_order
```

- Show TCP segments with duplicate ACKs:

```
tcp.analysis.duplicate_ack
```

2. TShark TCP Command-Line Filters:

- Capture TCP traffic:

```
tshark -i eth0 -f "tcp" -w capture_output.pcap
```

- Capture traffic on a specific TCP port (e.g., port 80 for HTTP):

```
tshark -i eth0 -f "tcp port 80" -w http_traffic.pcap
```

- Capture TCP traffic for a specific source IP:

```
  tshark -i eth0 -f "tcp and src host 192.168.1.1" -w src_ip_traffic.pcap
```

- Capture packets with specific flags (e.g., SYN flag):

```
  tshark -i eth0 -f "tcp[13] & 2 != 0" -w syn_packets.pcap
```

- Display TCP stream analysis (e.g., display summary of TCP streams):

```
  tshark -r capture_output.pcap -z tcp,streams
```

- Show TCP retransmissions (using TShark's -Y filter):

```
  tshark -r capture_output.pcap -Y "tcp.analysis.retransmission"
```

- Display TCP statistics (e.g., a summary of all TCP connections):

```
  tshark -r capture_output.pcap -z io,stat,0
```

- Extract TCP packet details (e.g., source and destination IP, and port):

```
  tshark -r capture_output.pcap -T fields -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport
```

- Filter for TCP streams:

```
  tshark -r capture_output.pcap -Y "tcp.stream eq 0"
```

3. Advanced TCP Analysis with Wireshark:

- TCP Window Size: To show packets with specific TCP window size values (useful for diagnosing congestion or flow control issues):

```
  tcp.window_size >= 1024
```

- TCP Packet Length: To display TCP packets of a certain length:

```
  tcp.len == 1500  # To show TCP packets of exactly 1500 bytes
```

- Analyze TCP Handshake: To focus on the three-way handshake (SYN, SYN-ACK, ACK) of TCP connections:

```
  tcp.flags.syn == 1 && tcp.flags.ack == 0  # SYN packet
  tcp.flags.syn == 1 && tcp.flags.ack == 1  # SYN-ACK packet
  tcp.flags.ack == 1 && tcp.flags.syn == 0  # ACK packet (final)
```

- Show HTTP over TCP: Combine TCP filter with HTTP to show HTTP traffic:

```
  tcp.port == 80 && http
```

- TCP Resets (RST): To show TCP connections that have been reset (useful for troubleshooting):

```
  tcp.flags.reset == 1
```

4. TShark Command Examples:

- Capture TCP traffic with SYN flag:

```
  tshark -i eth0 -f "tcp[13] & 2 != 0" -w capture_syn.pcap
```

- Show statistics of all TCP streams:

```
  tshark -r capture_output.pcap -z tcp,streams
```

- Capture packets from a specific TCP stream:

```
  tshark -r capture_output.pcap -Y "tcp.stream eq 0"
```