

Experiment no.: 08

Title: Filter packets according to protocol using any packet filtering tool.

Course Outcome: Build systems that are more secure against attacks.

Theory:

Packet filtering is a fundamental concept in **computer networking and network security**. It involves monitoring and controlling network data packets based on predefined rules or protocols such as **TCP, UDP, ICMP, or HTTP**. Each packet contains header information, including source and destination addresses, ports, and the communication protocol used.

When filtering packets according to protocol, a **firewall or network analyzer** inspects each packet and allows or blocks it depending on the specified protocol type. For example, a rule may allow only **TCP packets** and block **UDP or ICMP packets**. This process helps in managing network traffic, enhancing security, and preventing unauthorized access or attacks.

Tool used: Wireshark

Wireshark is a popular open-source **network protocol analyzer** used to capture and examine data packets traveling across a network. It helps in understanding how network protocols work, identifying network problems, and ensuring secure communication.

By using **packet filtering** in Wireshark, users can view packets of a **specific protocol** (like TCP, UDP, ICMP, or HTTP), which makes analysis easier and more focused.

Concept

Every data packet transmitted over a network contains header information, which includes the **protocol type** used for communication.

Wireshark allows users to apply **display filters** to show only packets matching a specific protocol or condition.

For example:

- `tcp` → Displays only TCP packets
- `udp` → Displays only UDP packets
- `icmp` → Displays only ping (Internet Control Message Protocol) packets
- `http` → Displays only web traffic packets

This helps in analyzing how specific protocols behave and interact within the network.

Applications

1. **Network Troubleshooting** – Helps identify network issues like delay or packet loss.
2. **Security Monitoring** – Detects suspicious or unauthorized network activities.
3. **Performance Analysis** – Analyzes traffic patterns and data flow efficiency.
4. **Educational Use** – Demonstrates how protocols like TCP, UDP, and HTTP work.

Advantages

1. **Real-Time Monitoring** – Captures live network traffic instantly.
2. **Detailed Packet Information** – Displays in-depth protocol-level data.
3. **Protocol-Specific Filtering** – Simplifies analysis using filter expressions.
4. **Free and Open Source** – Available for all major operating systems.

Limitations

1. **Complex for Beginners** – Requires understanding of network protocols.
2. **High Data Volume** – Capturing large traffic can slow down the system.
3. **Limited Access on Encrypted Traffic** – Cannot read encrypted HTTPS data.
4. **Administrative Privileges Required** – Needs elevated permissions to capture packets.

Steps:

1. **Install Wireshark** – Download and install Wireshark along with WinPcap/Npcap.
2. **Open Wireshark** – Launch the application to access the main interface.
3. **Select Network Interface** – Choose the network (Wi-Fi or Ethernet) to capture packets.
4. **Start Packet Capture** – Begin capturing live network traffic.
5. **Apply Protocol Filter** – Enter the desired protocol (e.g., TCP, UDP, ICMP) in the filter bar.

- 6. Analyze the Packets** – View details like source, destination, ports, and data of filtered packets.
- 7. Stop the Capture** – End the packet capture after collecting sufficient data.
- 8. Save or Export Results** – Save the captured or filtered packets for future analysis.

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ... All interfaces shown ▾

Wi-Fi

- Local Area Connection* 10
- Local Area Connection* 9
- Local Area Connection* 8
- Bluetooth Network Connection
- Local Area Connection* 2
- Local Area Connection* 1
- Adapter for loopback traffic capture
- Local Area Connection
- Ethernet 2
- Ethernet
- Event Tracing for Windows (ETW) reader: etwdump

Capturing from Wi-Fi

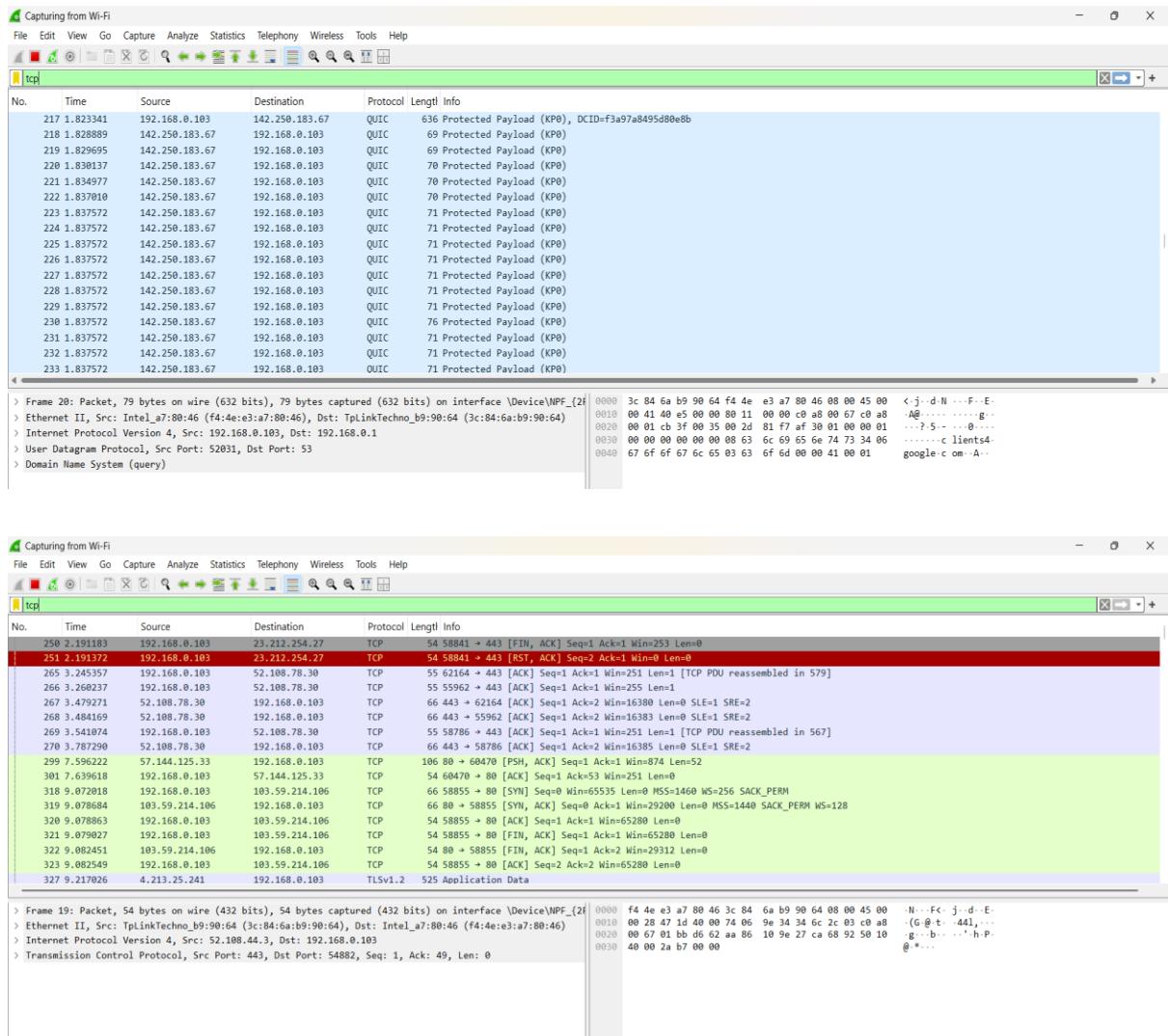
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
250	2.191183	192.168.0.103	23.212.254.27	TCP	54	58841 → 443 [FIN, ACK] Seq=1 Ack=1 Win=253 Len=0
251	2.191372	192.168.0.103	23.212.254.27	TCP	54	58841 → 443 [RSN, ACK] Seq=2 Ack=1 Win=0 Len=0
252	2.217674	192.168.0.103	142.250.183.67	UDP	75	59470 → 443 Len=33
253	2.219139	142.250.183.67	192.168.0.103	UDP	70	443 → 59470 Len=24
254	2.219523	192.168.0.103	142.250.183.67	UDP	75	59470 → 443 Len=33
255	2.224333	142.250.183.67	192.168.0.103	UDP	71	443 → 59470 Len=25
256	2.264226	192.168.0.103	142.250.183.67	UDP	76	59470 → 443 Len=34
257	2.268150	142.250.183.67	192.168.0.103	UDP	71	443 → 59470 Len=25
258	2.466858	192.168.0.103	142.250.183.67	UDP	76	59470 → 443 Len=34
259	2.470503	142.250.183.67	192.168.0.103	UDP	538	443 → 59470 Len=492
260	2.471803	142.250.183.67	192.168.0.103	UDP	71	443 → 59470 Len=25
261	2.477675	142.250.183.67	192.168.0.103	UDP	217	443 → 59470 Len=171
262	2.481951	192.168.0.103	142.250.183.67	UDP	75	59470 → 443 Len=33
263	2.483991	192.168.0.103	142.250.183.67	UDP	76	59470 → 443 Len=34
264	2.487874	142.250.183.67	192.168.0.103	UDP	71	443 → 59470 Len=25
265	3.245357	192.168.0.103	52.108.78.30	TCP	55	62164 → 443 [ACK] Seq=1 Ack=1 Win=251 Len=1

```

> Frame 1: Packet, 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface \Device\NPF_{2F5...
> Ethernet II, Src: TplinkTechno_b9:90:64 (3c:84:6a:b9:90:64), Dst: Intel_a7:80:46 (f4:4e:e3:a7:80:46)
> Internet Protocol Version 4, Src: 20.189.173.18, Dst: 192.168.0.103
> Transmission Control Protocol, Src Port: 443, Dst Port: 58853, Seq: 1, Ack: 1, Len: 42
> Transport Layer Security
0000  f4 4e e3 a7 80 46 3c 84 6a b9 90 64 08 00 45 00  N-- F<- j - d - E-
0010  00 52 f0 af 40 00 6c 06 9b 17 14 bd ad 12 c9 a8  R - @ - l - . . .
0020  00 67 01 b6 e5 d3 d6 73 99 25 e6 8b f1 50 18  g - . . . s % - P -
0030  40 01 e9 df 00 00 17 03 03 00 25 e6 00 00 00 00 00  @ - . . . %
0040  00 00 04 82 9f 97 39 a8 35 1f ef df 22 40 b4 26  . . . 9 - 5 - " @ &
0050  47 8e c5 59 7c 34 d9 5f 9e ec c1 52 e6 e6 60 2c  G - Y | 4 - . . . R - ,
```



Conclusion:

I have successfully learned packet filtering using Wireshark.