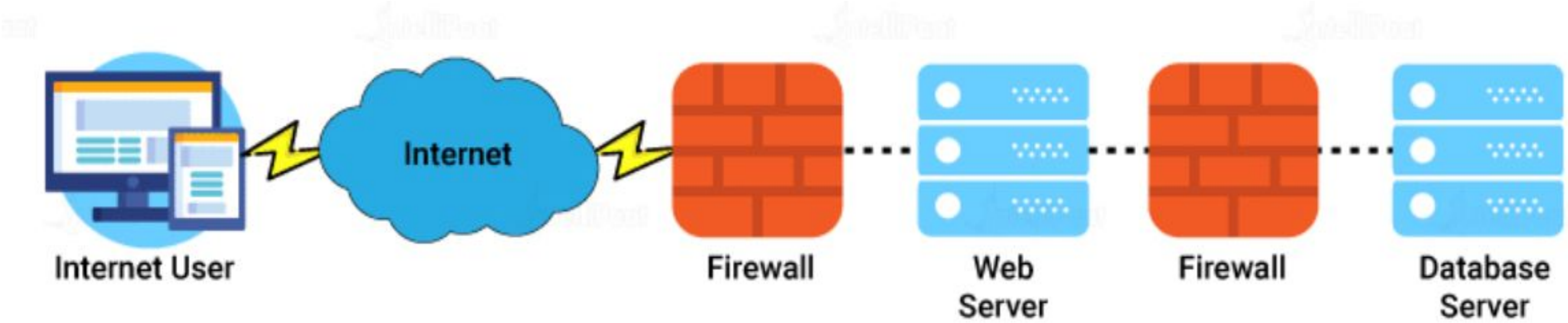
Network Security

Network Malware Analysis and Forensics



Introduction

The Internet is a network of networks run by different organizations. These organizations in turn, maintain a complex network themselves. This span to multiple locations or departmental responsibilities.

Network security, is an important part of cyber security that helps in protecting your network and data stored in network connected devices from network breaches, software and hardware intrusions and so on …

Communication , session and protocols in a network

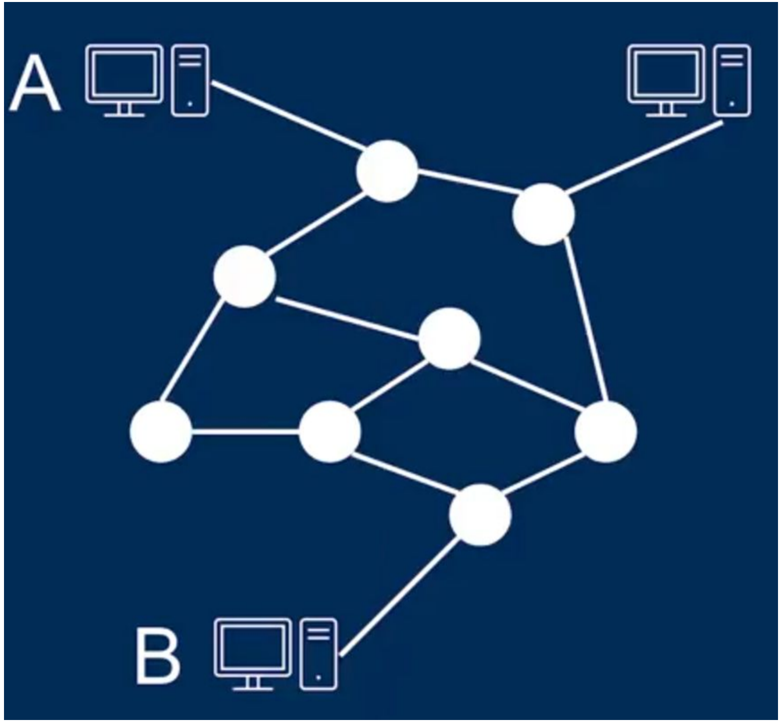
Communication is defined as the exchange of information between entities and **session** is a logical connection between two or more devices for the purpose of exchanging data, managed by the session layer of the OSI model. It involves the establishment, maintenance, and termination of the connection. **Protocols:** Technical communication in networks follows protocols, which are defined sets of rules that guide how to interact. These protocols involve complex security threats, particularly regarding confidentiality, integrity, and availability.

Addressing and Routing

Where vs How to send data.

Addressing refers to the process of identifying devices or entities within a network to ensure that data can be sent to the correct destination. **Routing** is the process of determining the best path for data to travel from the source device to the destination device, especially when there’s no direct connection between them. Routers examine the destination IP address of a data packet and decide the best path for the data to travel through the network.

Based on current understanding, what threats you foresee? Think and please write them down.



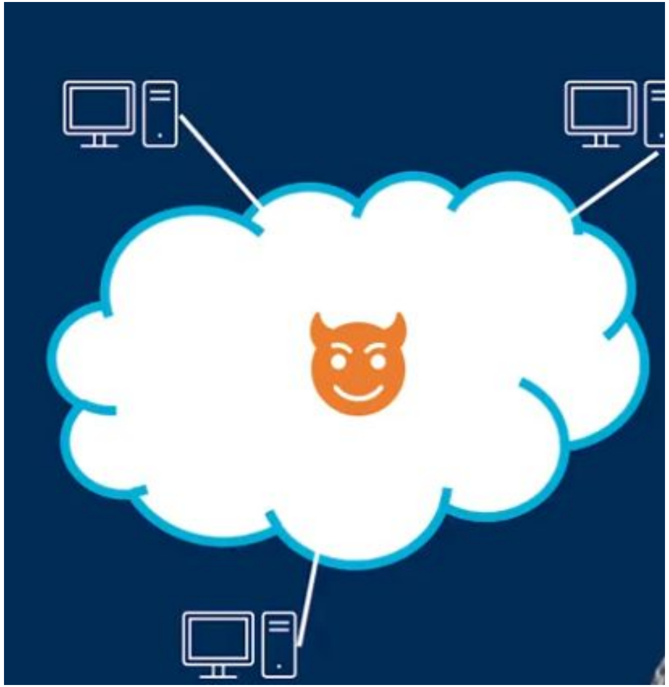
Adversaries in a network

How can we achieve CIA triad in the presence of an adversary in a network

A **powerful adversary** may control the entire network, allowing them to block, intercept, or spoof communication. Strong **cryptography** (like **TLS**) is used to maintain security in such environments.

A **weaker adversary** may be **passive**, eavesdropping on data without blocking communication. In this case, availability is preserved but confidentiality may be compromised.

Some adversaries may change their behavior or location. For example, a machine compromised through vulnerabilities may become a new attacker.



Think like an adversary

Think of PUCIT and Punjab CM Free WIFI networks. Assume that you are an adversary. Imagine one or two good scenarios for attacks on each of them considering the following points:

- What are the possible targets for an adversary in this network?
- What kind of adversary could try to attack these targets?

Network Layers

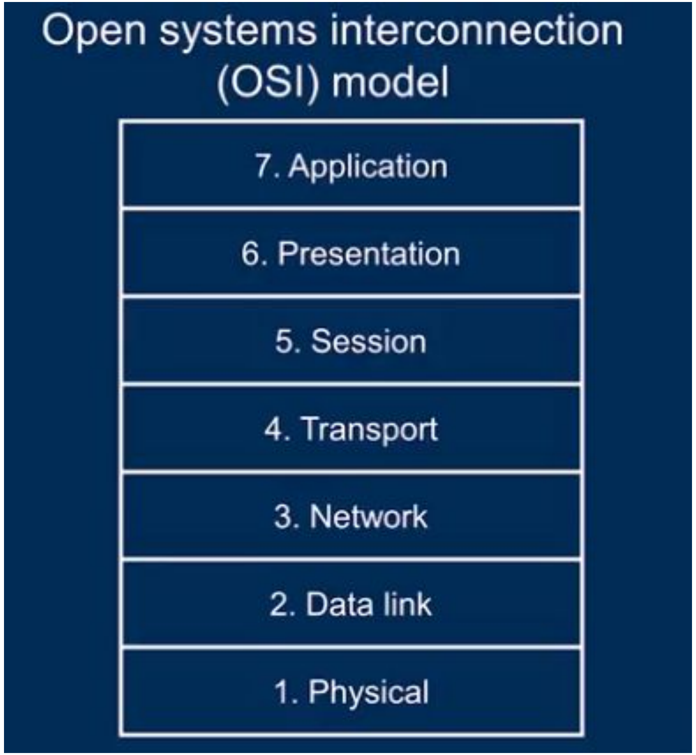
Network communication relies on a layered approach, from the physical transmission of data to the application logic. The **OSI model** provides a structured framework, but modern network design often simplifies this model for practical use, emphasizing key layers such as **transport** and **application**. Networks use a **layering strategy** to address one problem at a time, making systems modular and easier to understand.

- To connect multiple local networks globally, the **Internet layer** (or **network layer**) is used, with the **Internet Protocol (IP)** enabling global addressing and routing of messages.
- The **transport layer** manages the communication between applications using **TCP (Transmission Control Protocol)** and **UDP (User Datagram Protocol)**. These protocols introduce the concept of **ports** for addressing applications and their **sockets**.
- **TCP** creates a **bidirectional channel** for continuous data transfer, splitting data into packets, ensuring packets arrive in order, and retransmitting lost packets.
- The **application layer** enables communication between applications, and different protocols are used based on the type of communication. Examples include:
    - **HTTP (Hypertext Transfer Protocol)**: A request-response protocol.
    - **SOAP (Simple Object Access Protocol)**: Built on HTTP, used for remote procedure calls and complex data transfers.

OSI Model

The **OSI Model** (Open Systems Interconnection) has **seven layers**: Physical, Data Link, Network, Transport, Session, Presentation, and Application.

Some layers, like **Session** and **Presentation**, are less relevant in modern networks and are often considered part of the **application layer**.



Open systems interconnection (OSI) model

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data link
1. Physical

OSI Layer Attacks

	Layer	Attack
7	Application	Buffer overflow, XSS, DDoS
6	Presentation	Unicode vulnerability, SSL strip
5	Session	Session hijacking, DNS poisoning
4	Transport	SYN flood, invalid TCP flags, UDP flood
3	Network	ICMP flood, OS fingerprinting, IP address spoofing, routing table poisoning
2	Data link	Sniffing, ARP cache poisoning, macof attack
1	Physical	Cutting cables, jamming, keystroke logging

Network attack examples

Eavesdropping and Packet Sniffing: Attackers may intercept and monitor network traffic to capture sensitive information, such as login credentials or confidential data, by "sniffing" packets as they traverse the network.

DNS Spoofing and Cache Poisoning: These attacks manipulate the Domain Name System (DNS) to redirect users to malicious websites or intercept their traffic. This can lead to unauthorized access or data theft.

Intrusion Detection

Involves monitoring a network or a system for malicious activities, policy violations, or unauthorized access and then generating alerts or taking action based on the findings. The primary purpose of intrusion detection is to identify potential threats and security incidents in real-time or near real-time, allowing security personnel to respond quickly and mitigate risks.

Network-based Intrusion Detection System: NIDS monitors network traffic in real-time and looks for patterns or signatures that indicate known attack methods, anomalies, or suspicious behavior.

Host-based Intrusion Detection System: It is installed on individual host systems (such as servers or workstations) and monitors activities on those hosts. It examines system logs, file changes, and system calls for signs of unauthorized access, malware, or other security issues specific to that host.

Sniffing Network Traffic

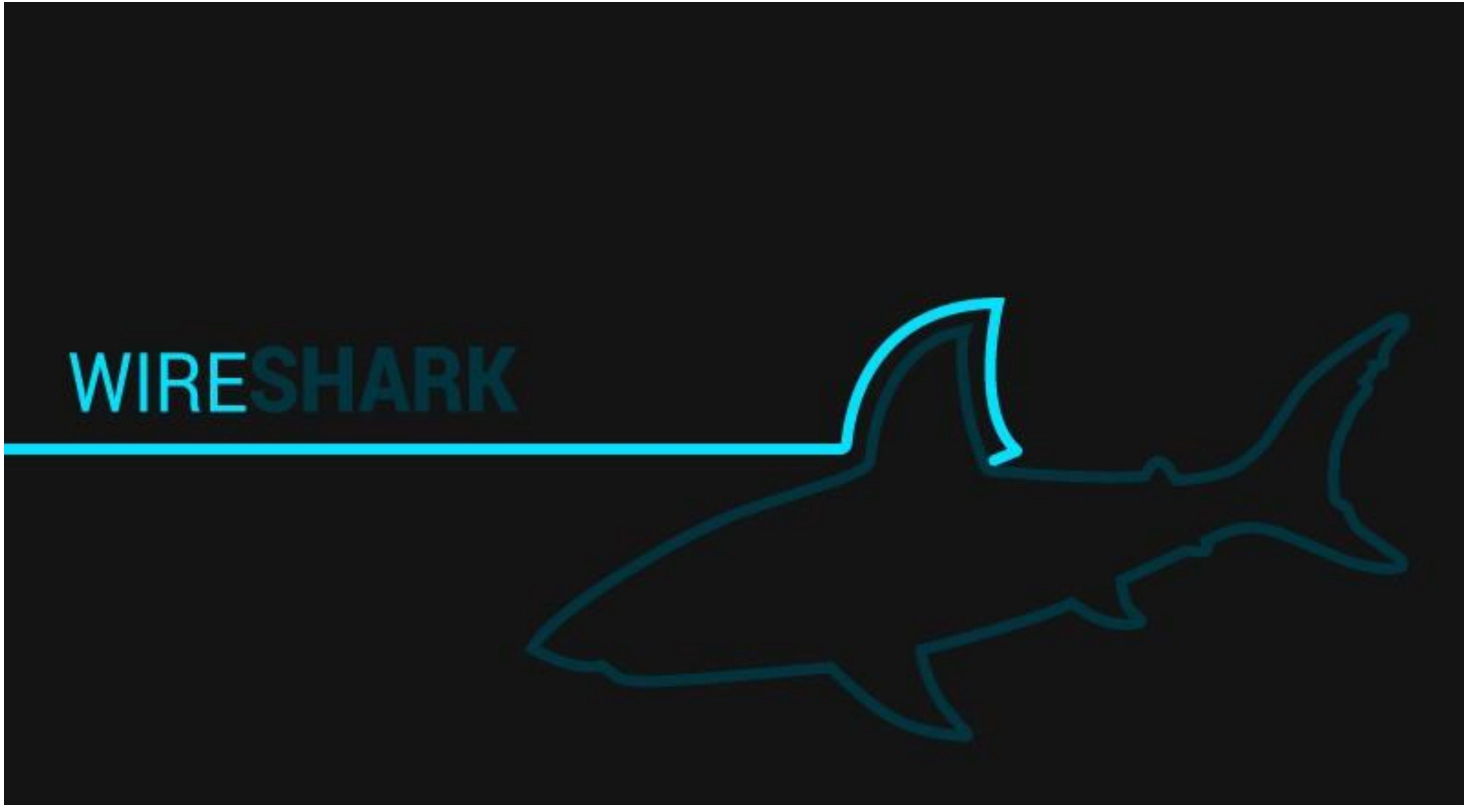
Sniffing network traffic is an important skill, as today's threats may have slipped by threat management systems and have found a home within network.

We shall learn to use **Wireshark** for packet sniffing.

https://www.wireshark.org/docs/wsug_html_chunked/ChCustCommandLine.html

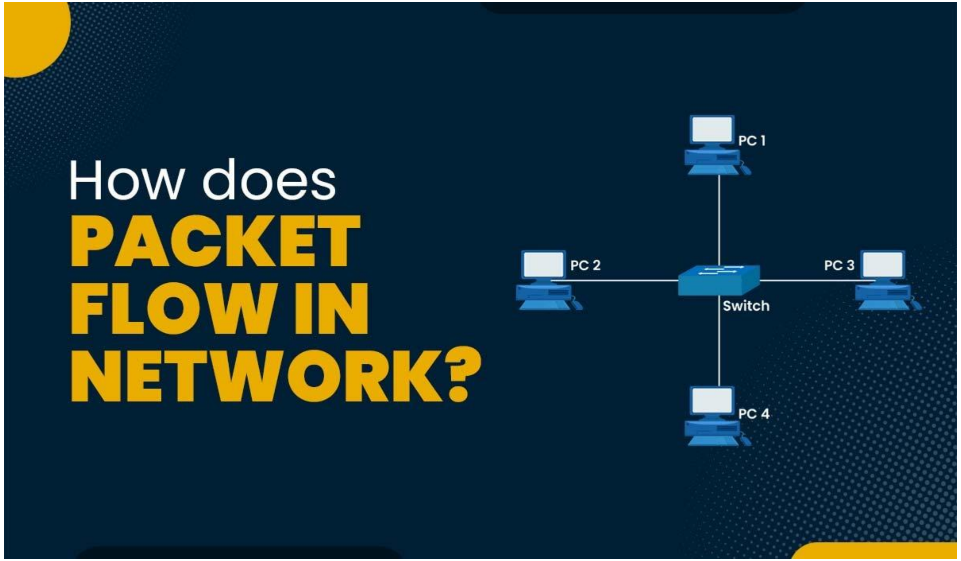
Please learn important commands from above link.

Packet analysis using Wireshark



Packet analysis using Wireshark

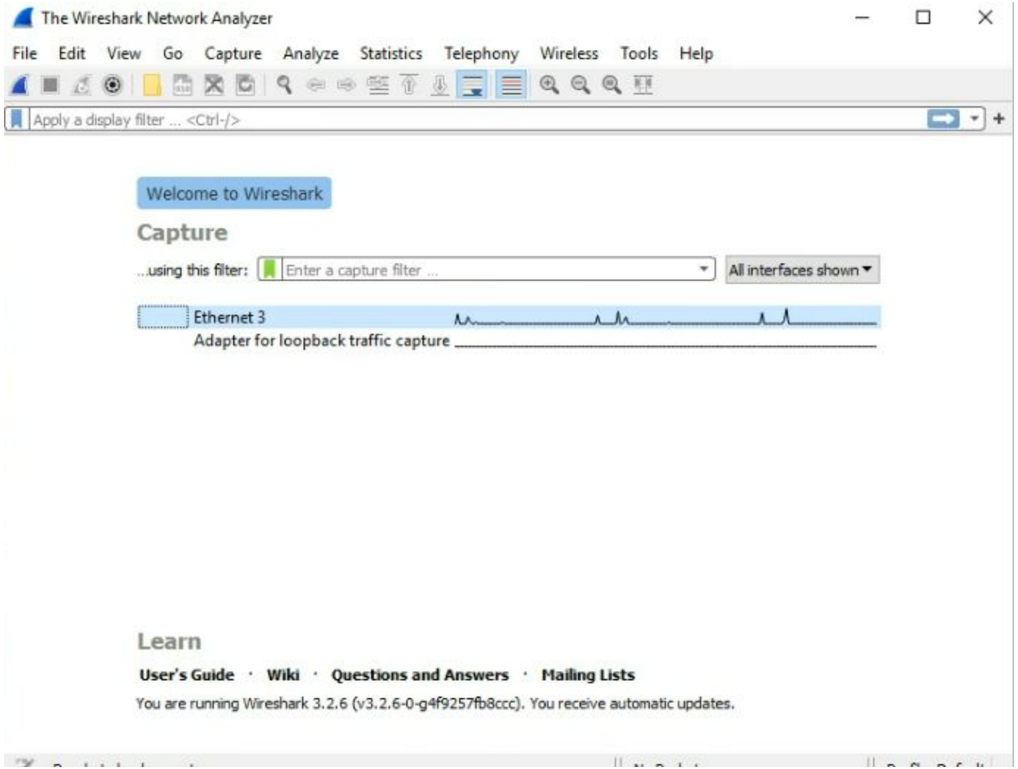
- What is packet sniffing and why it is important for CyberSecurity?
- How can we capture and analyze packets in real world?



Packet analysis using Wireshark

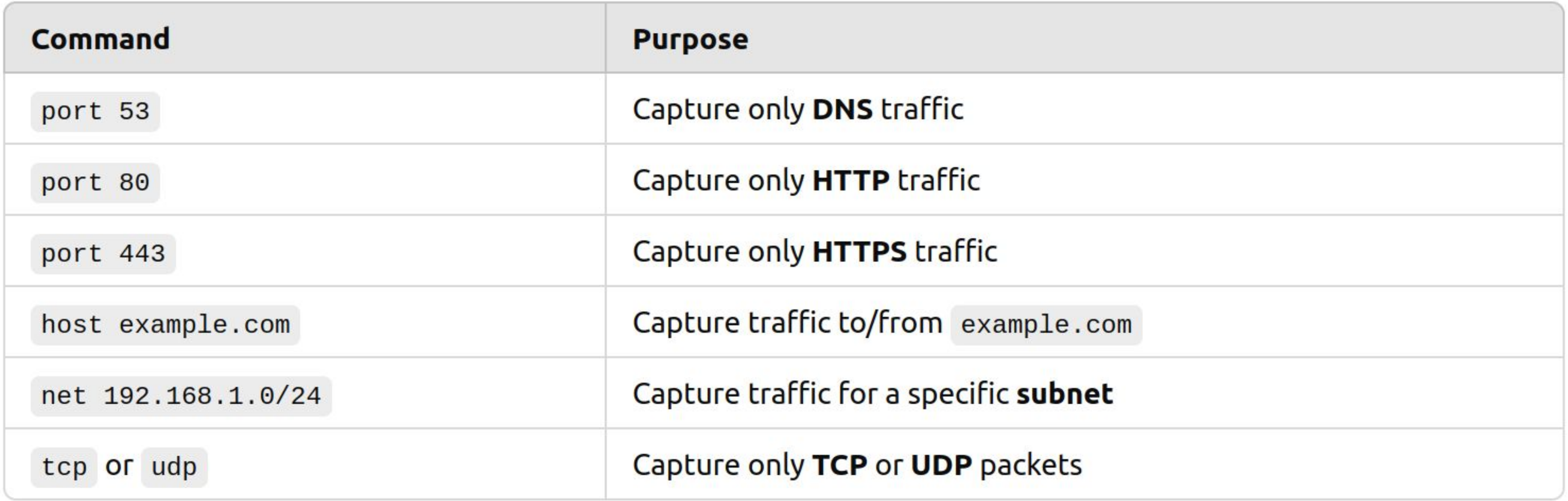
- Identify the interface
- Apply Capture Filters
- Look for required Display Filters
- What happens when you visit a website using a browser
- What happens when you visit http://example.com

➔ Important commands in next slides


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
Ethernet 3
Adapter for loopback traffic capture
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.2.6 (v3.2.6-0-g4f9257fb8ccc). You receive automatic updates.

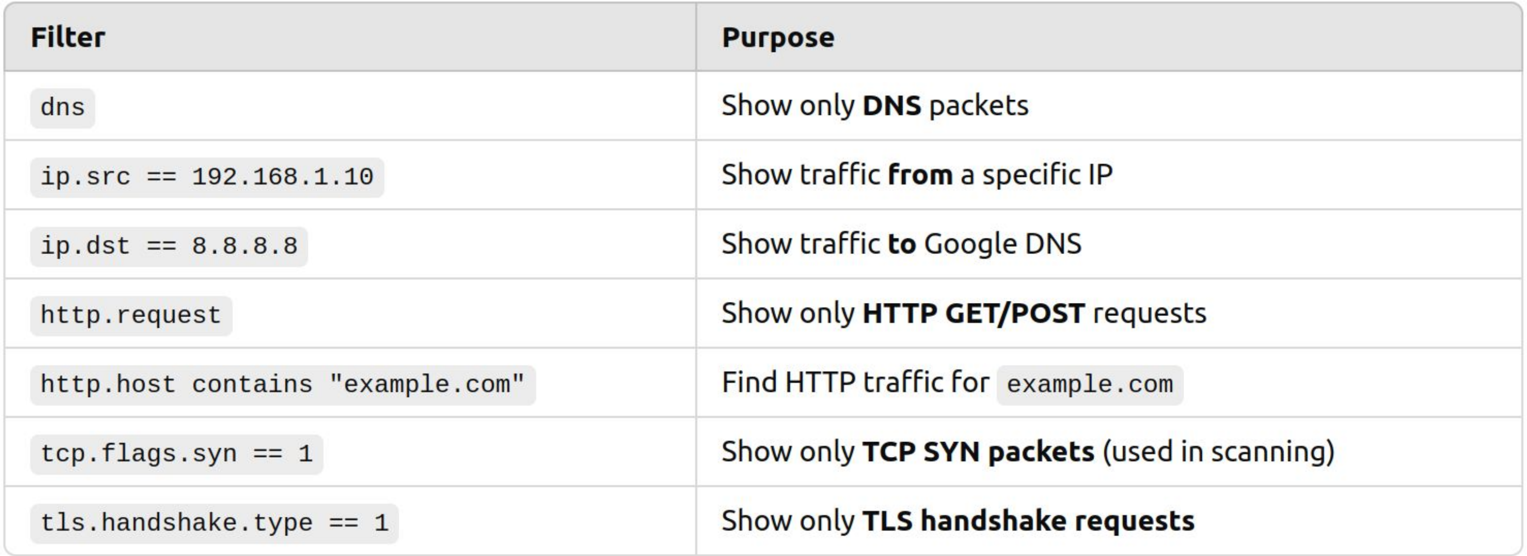
Capture Filters



Command	Purpose
`port 53`	Capture only **DNS** traffic
`port 80`	Capture only **HTTP** traffic
`port 443`	Capture only **HTTPS** traffic
`host example.com`	Capture traffic to/from `example.com`
`net 192.168.1.0/24`	Capture traffic for a specific **subnet**
`tcp` or `udp`	Capture only **TCP** or **UDP** packets

Capture filters help reduce unnecessary traffic and focus on specific packets.

Display Filters



Filter	Purpose
`dns`	Show only **DNS** packets
`ip.src == 192.168.1.10`	Show traffic **from** a specific IP
`ip.dst == 8.8.8.8`	Show traffic **to** Google DNS
`http.request`	Show only **HTTP GET/POST** requests
`http.host contains "example.com"`	Find HTTP traffic for `example.com`
`tcp.flags.syn == 1`	Show only **TCP SYN packets** (used in scanning)
`tls.handshake.type == 1`	Show only **TLS handshake requests**

Display filters refine traffic after capturing, making analysis easier.

Activity # 1

1. Open Wireshark and select your network interface.
2. Click Start Capture.
3. Open a browser and visit http://example.com.
4. Stop capture after page fully loaded, see packets in wireshark.
5. Identify source/destination IPs and protocols used.

Activity # 2

1. Filter for DNS Traffic: dns
2. Filter for HTTP Requests: http.request
3. Filter for a Specific IP: ip.addr == 192.168.8.4
4. Filter for HTTP Response: http.response

Activity # 3

1. Please capture HTTP traffic using Wireshark. You can visit http://example.com
2. Put filter for http-only traffic. Right-click an HTTP GET request → Follow TCP Stream.
3. Can you see raw HTTP communication? What else do you see regarding TCP?
4. Why HTTPS is crucial?

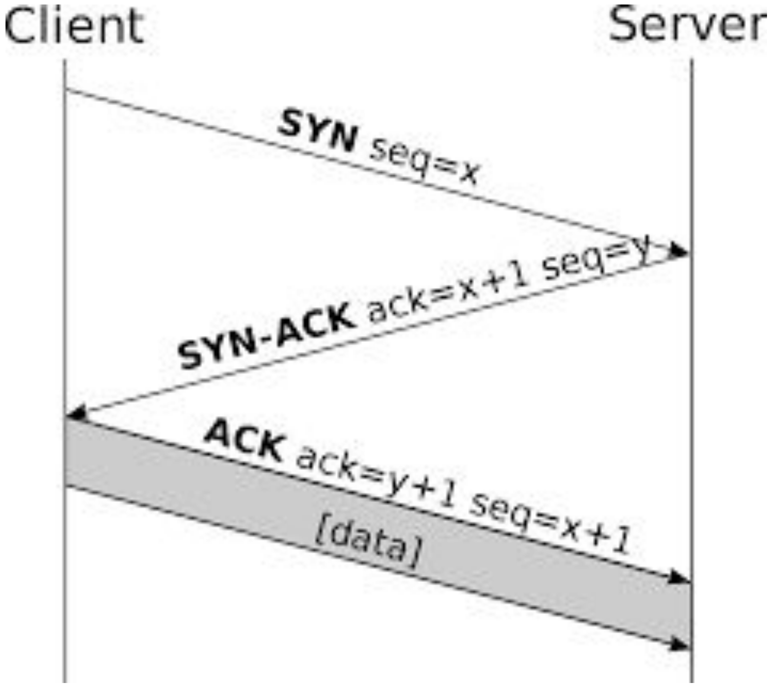
Activity # 4

1. Visit www.pucit.edu.pk
2. In Wireshark, filter DNS traffic (dns).
3. Identify the query and response (IP address assigned to pucit.edu.pk)
4. Can you find IP address of www.pu.edu.pk?

Note: You may need to clear your DNS cache.

TCP Handshake

The TCP (Transmission Control Protocol) handshake, also known as the three-way handshake, is a crucial process in establishing a reliable connection between two devices over a network. It is the foundation of TCP's reliability, error-checking, and sequencing mechanisms. It consists of three steps, please read more in this handout, ***tcp handshake***.



Activity # 4

1. Capture and analyze local TCP Client and Server traffic.
2. Set up a TCP Server using Python on local host and port (12345)
3. Set up a TCP client
4. Open Wireshark and select the Loopback interface (lo on Linux and localhost on Windows).
5. Start packet capture.
6. Run the TCP server first, then execute the TCP client.
7. Stop capture in Wireshark.
8. Use the following Wireshark filters to analyze traffic:
    a. Filter for TCP traffic: tcp
    b. Filter for specific port: tcp.port == 12345
    c. Filter for localhost traffic: ip.addr == 127.0.0.1
9. Identify TCP 3-way handshake (SYN, SYN-ACK, ACK).
10. Locate data packets exchanged between client and server.

Activity #5

1. https://www.cloudshark.org/captures/5cc921a5df00 , Can you find and export and image from this http traffic pcap?
2. **Can you capture a message of another class fellow from Activity #4?**
3. **Can you do a ping flood attack on a network device?**
4. **Think of a DNS Spoof attack?**

Activity #6

1. Wireshark Activity uploaded in google classroom

https://docs.google.com/document/d/1-VSsJ4A_HjXTO0EesNX8GYvVgJAvFPuJovOLV_hx5H4/edit?usp=sharing

2. Try Hack Me Activity

https://tryhackme.com/room/introtonetworking

TBD in network security later in semester….

- Access Control Lists
- ARP Storm attack
- Firewall rules creation