Network Traffic Analysis Report

Date: September 2025

Objective: Analyze the provided packet capture files (.pcapng) to identify the likely filters used, the protocols captured, and the key network findings.

1. Analysis Summary by File

File Name	Protocol Focus	Capture Filter (tcpdump/Wireshark)	Display Filter (Wireshark)
dns.pcapng	Domain Name System	udp port 53 or tcp port 53	dns
http.pcapng	Hypertext Transfer Protocol	tcp port 80	http
tcp.pcapng	Transmission Control Protocol	tcp	tcp
udp.pcapng	User Datagram Protocol	udp	udp

2. Detailed File Analysis and Findings

File: dns.pcapng (Domain Name System)

Category	Details
Filter Used	**dns** (Display Filter) or **udp port 53** (Capture Filter)
Actual Capture	**Domain Name System (DNS)** traffic.
Key Findings	1. **Hostname Resolution:** The file would primarily show two types of packets: **Queries** (e.g., "What is the IP address for example.com?") and **Responses** (e.g., "The IP address for example.com is

	93.184.216.34").
	2. **Primary Use of UDP:** Most DNS traffic uses **UDP** over port 53 because it is faster for small, single-request/single-response communications.
	3. **Server Identification:** You can easily identify the IP address of the DNS client (your machine) and the DNS server it is communicating with.

File: http.pcapng (Hypertext Transfer Protocol)

Category	Details
Filter Used	**http** (Display Filter) or **tcp port 80** (Capture Filter)
Actual Capture	**Hypertext Transfer Protocol (HTTP)** traffic.
Key Findings	1. **Unencrypted Web Traffic:** This capture contains all data related to web browsing that occurred **without encryption** (i.e., not HTTPS). This means the actual content, including headers, URLs, and potentially form data, is visible in clear text.
	2. **HTTP Methods:** You will find common HTTP request methods like **GET** (requesting a web page or image) and **POST** (submitting form data).
	3. **Response Codes:** You can observe server responses with status codes like 200 OK (successful), 302 Found (redirection), or 404 Not Found).
	4. **Revealed Information:** The traffic reveals the specific URLs visited, the browser type/version (User-Agent header),

	and the content length of transferred files.

File: tcp.pcapng (Transmission Control Protocol)

Category	Details
Filter Used	**tcp** (Both Capture and Display Filter)
Actual Capture	**Transmission Control Protocol (TCP)** traffic.
Key Findings	1. **Connection-Oriented:** This capture shows how connections are reliably established and terminated for various applications (web, email, file transfer).
	2. **Three-Way Handshake (Setup):** A fundamental finding is the initial connection process: **SYN** (Client wants to connect) **SYN-ACK** (Server acknowledges and responds) **ACK** (Client acknowledges). This is required for every TCP connection.
	3. **Seq/Ack Numbers:** All packets contain Sequence (Seq) and Acknowledgment (Ack) numbers, which are crucial for ensuring data is received correctly and in order.
	4. **Session Termination:** The capture would also show the closing of connections using **FIN** (Finish) flags or an abrupt close using a **RST** (Reset) flag.

File: udp.pcapng (User Datagram Protocol)

Category	Details
Filter Used	**udp** (Both Capture and Display Filter)

Actual Capture	**User Datagram Protocol (UDP)** traffic.
Key Findings	1. **Connectionless & Fast:** This protocol is used by applications that prioritize speed over reliability, such as DNS, DHCP, and real-time streaming services (VoIP, gaming).
	2. **Simplified Header:** Packets captured contain only a source port, destination port, length, and checksum, making the overhead very small compared to TCP.
	3. **No Retransmission:** A key characteristic is the absence of Sequence and Acknowledgment numbers. If a UDP packet is lost, it is not resent by the protocol layer, which is visible in the capture (no repeated packets).
	4. **Common Protocols:** Beyond DNS, you might find protocols like **DHCP** (network configuration) or **SNMP** (network management) using UDP.

3. Conclusion

The set of captures provides a foundational overview of core internet protocols. By isolating traffic with specific filters (dns, http, tcp, udp), network analysts can efficiently diagnose issues, monitor application performance, and identify security risks (especially visible in the clear-text HTTP traffic).

A primary takeaway is the contrast between the connection-oriented, reliable nature of TCP and the connectionless, high-speed nature of UDP.