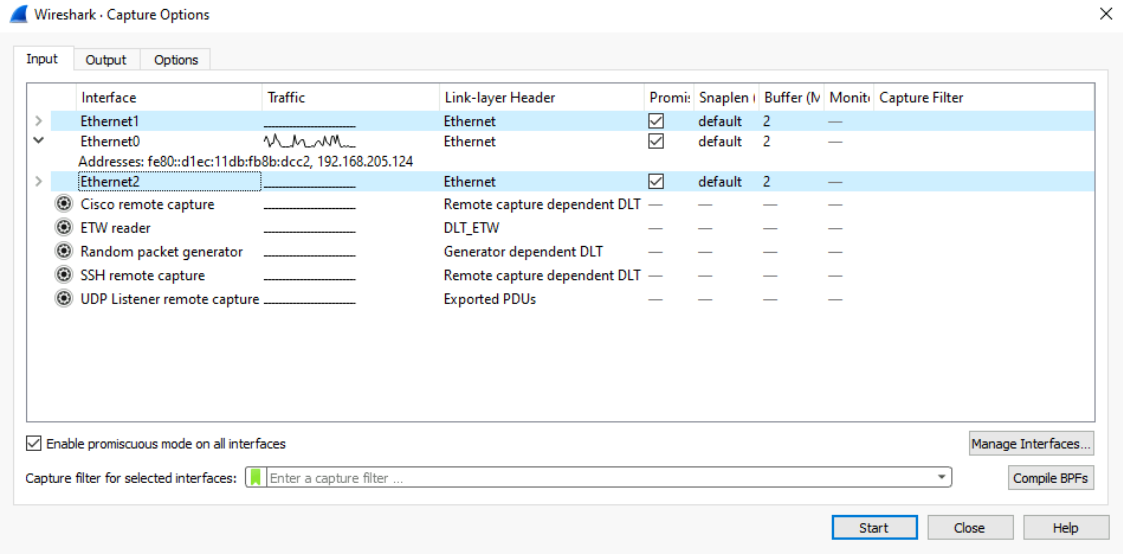
Packet Capture and Analysis Report

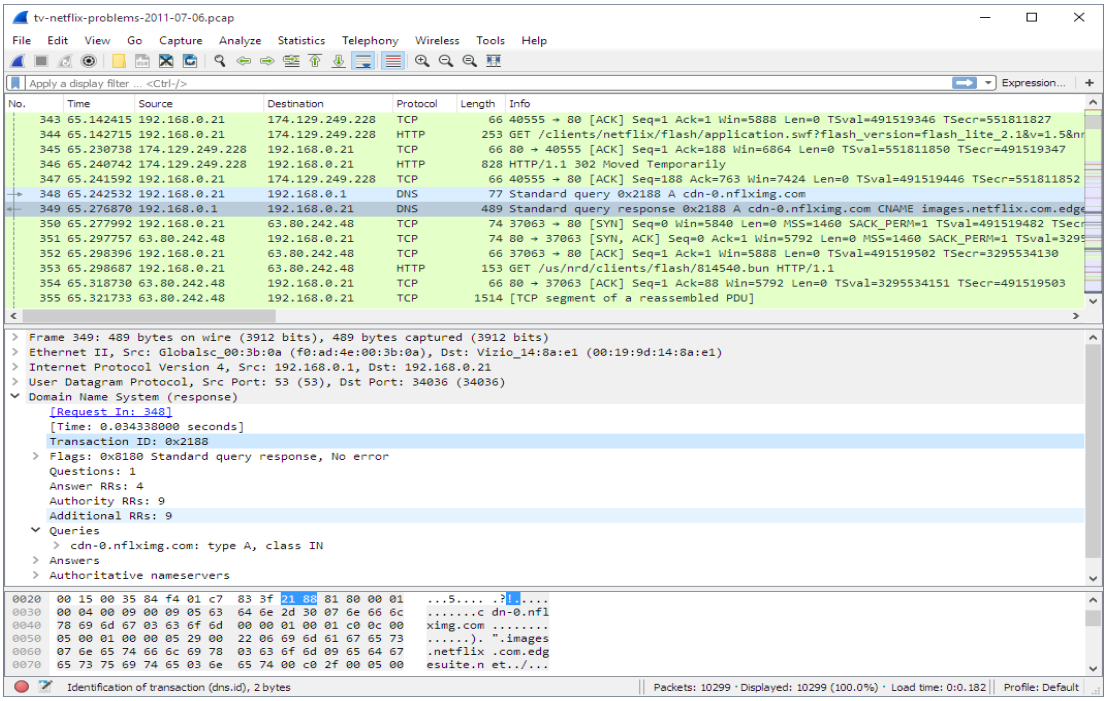
Packet Capture File

- **Tool Used:** Wireshark
- **Capture Format:** `.pcapng`
- **Network Interface:** Active internet interface (Wi-Fi/Ethernet)
- **Capture Type:** Live network traffic

Packet Capture Process

I opened Wireshark and selected the active network interface connected to the internet. After starting the capture, packets were displayed continuously, showing real-time communication between my system and different servers.
The capture included various protocols such as DNS, TCP, HTTP, and HTTPS.





Packet Analysis Report

1. Basic Packet Observation

Each packet captured contained information such as:

- Source IP address
- Destination IP address
- Protocol type
- Packet length

By expanding packet details, I could view different protocol layers including Ethernet, IP, TCP/UDP, and application-layer protocols.

2. Protocol Filtering and Analysis

To analyze specific traffic, I used display filters:

- `dns` → To view DNS packets
- `tcp` → To view TCP communication
- `http` → To view HTTP traffic

Using filters reduced unnecessary packets and made analysis easier.

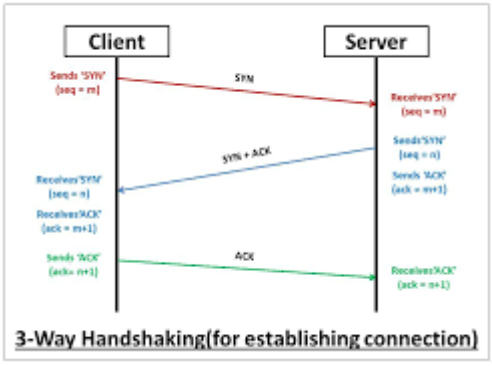
---

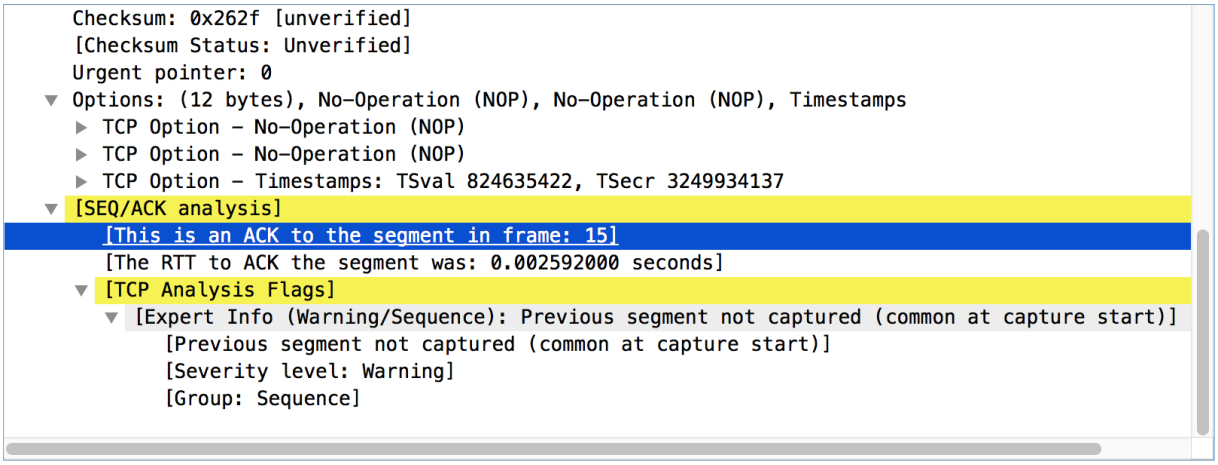
3. TCP Connection Analysis

While analyzing TCP packets, I observed the TCP three-way handshake.
The connection was established using the following sequence:

- SYN packet from the client
- SYN-ACK packet from the server
- ACK packet from the client

This confirmed that TCP uses a connection-oriented method for reliable data transmission.





4. Plain-Text vs Encrypted Traffic

I compared HTTP and HTTPS traffic during the capture.

- **HTTP traffic:**
  The data was visible and readable, including request headers and URLs.
- **HTTPS traffic:**
  The data was encrypted and appeared unreadable, showing secure communication using encryption.

This comparison clearly showed the importance of encrypted traffic for security.

5. DNS Query Analysis

Using the DNS filter, I observed DNS queries generated while opening websites.
The DNS packets showed:

- Requested domain names
- Corresponding IP addresses returned in responses

This demonstrated how domain names are resolved into IP addresses before establishing connections.

6. Saving the Packet Capture File

After completing the capture and analysis, I stopped the capture and saved the file in `.pcapng` format.
The saved capture file can be reopened later for further inspection or reference.

Conclusion

The packet capture and analysis helped me understand real-time network communication.
By analyzing DNS, TCP, HTTP, and HTTPS traffic, I gained practical insight into how connections are established, how data is transmitted, and how encryption protects information.
This analysis strengthened my understanding of basic networking and packet-level communication.