Task 5: Network Traffic Capture & Analysis Report

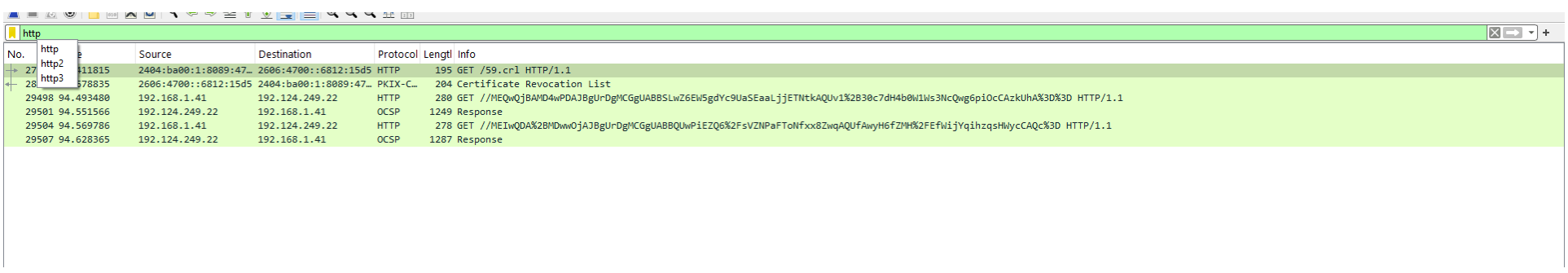
Tool Used: Wireshark
Websites Visited: YouTube, GitHub, Dailymotion
File Submitted: .pcap file + Screenshots

Objective:

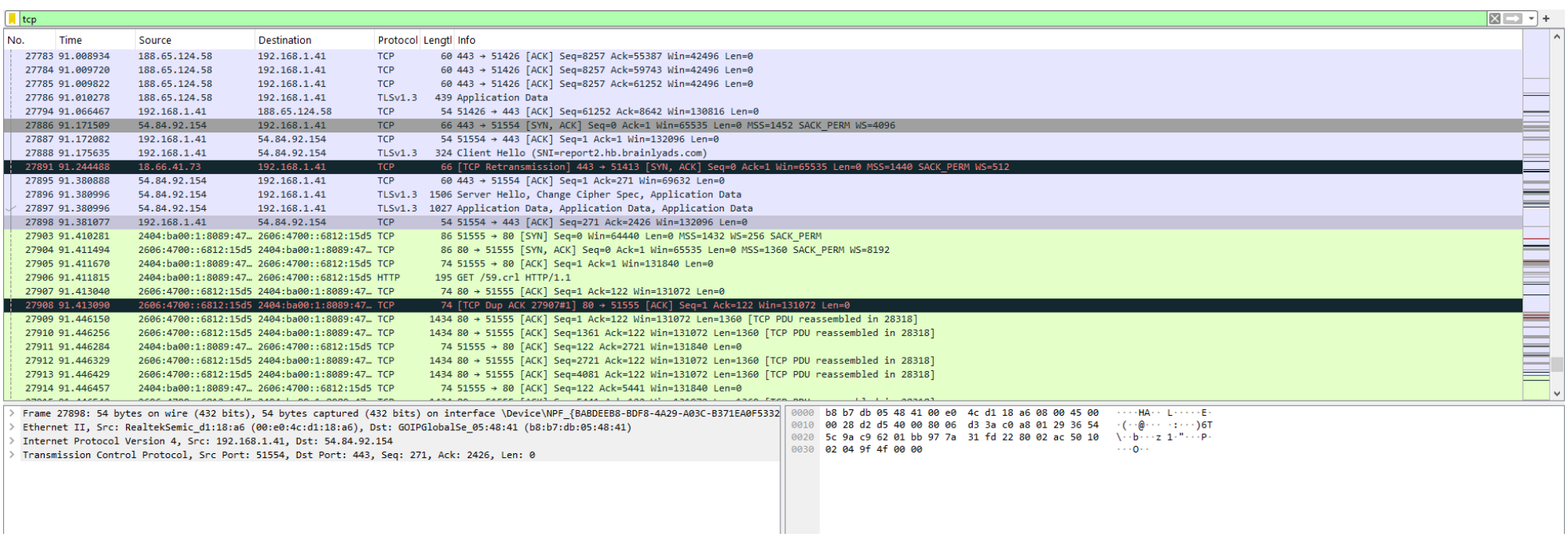
To capture live network traffic using Wireshark and analyze packets based on protocol types. This helps understand how different protocols behave and identify them during traffic inspection.

Steps taken:

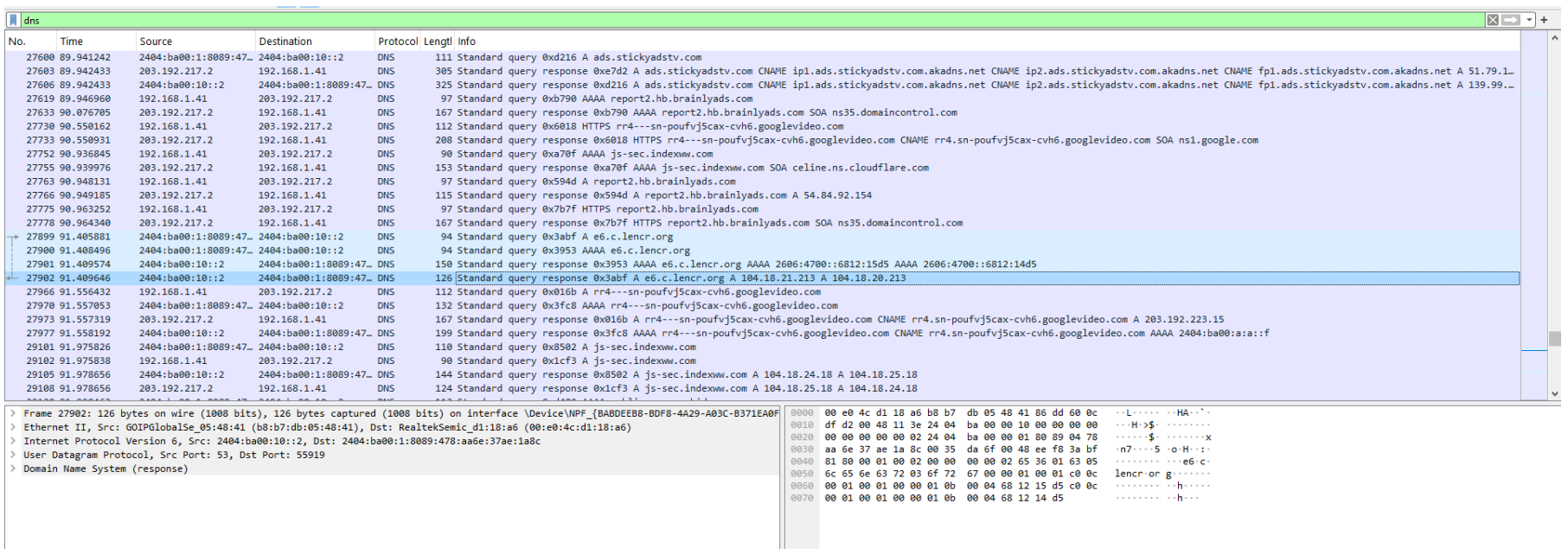
1. **Started Wireshark** and selected my active network interface.
2. **Began live capture** and opened 3 websites in browser:
    - **YouTube** (streaming/media-heavy site)
    - **GitHub** (HTTPS-encrypted developer platform)
    - **Dailymotion** (another video/media-based site)
3. **Stopped the capture** after ~1 minute.
4. **Applied filters** one by one:
    - http to view HTTP traffic (non-encrypted web data)



- tcp to view TCP-level connections for all visited sites



- dns to view DNS query/response packets (domain resolution)



5. **Saved the capture as .pcap file** and took protocol-specific screenshots.
6. **Analyzed individual packets** for source/destination IPs, flags, and info.

Protocols Identified:

Protocol	Where It Was Found	Purpose
DNS	On all websites	Used to resolve domain names (e.g., youtube.com to IP)
TCP	On all connections	Underlying transport protocol, ensures delivery
HTTP	Seen on GitHub/Dailymotion (some content)	Transmits web content (unencrypted)
HTTPS	Seen for YouTube & GitHub	Secure version of HTTP (was visible as TCP + encrypted)

Observations:

- **YouTube and Dailymotion** triggered lots of TCP packets for video loading (streaming behavior).
- **GitHub** showed secure https traffic but its **DNS queries** and **TCP handshakes** were visible.
- DNS packets clearly showed the domain names I accessed.
- The packet info (source/destination, flags) helped track how devices communicated.

Conclusion:

Wireshark provided real-time insight into how data flows when visiting different websites. I could see how each website uses a mix of DNS, TCP, and HTTP(S) to load content. This task gave me practical experience in filtering, identifying protocols, and understanding how even encrypted traffic leaves metadata behind.