Task Objective

To perform a live network packet capture using Wireshark, identify multiple network protocols in use, and analyze key packets to understand client-server interactions.

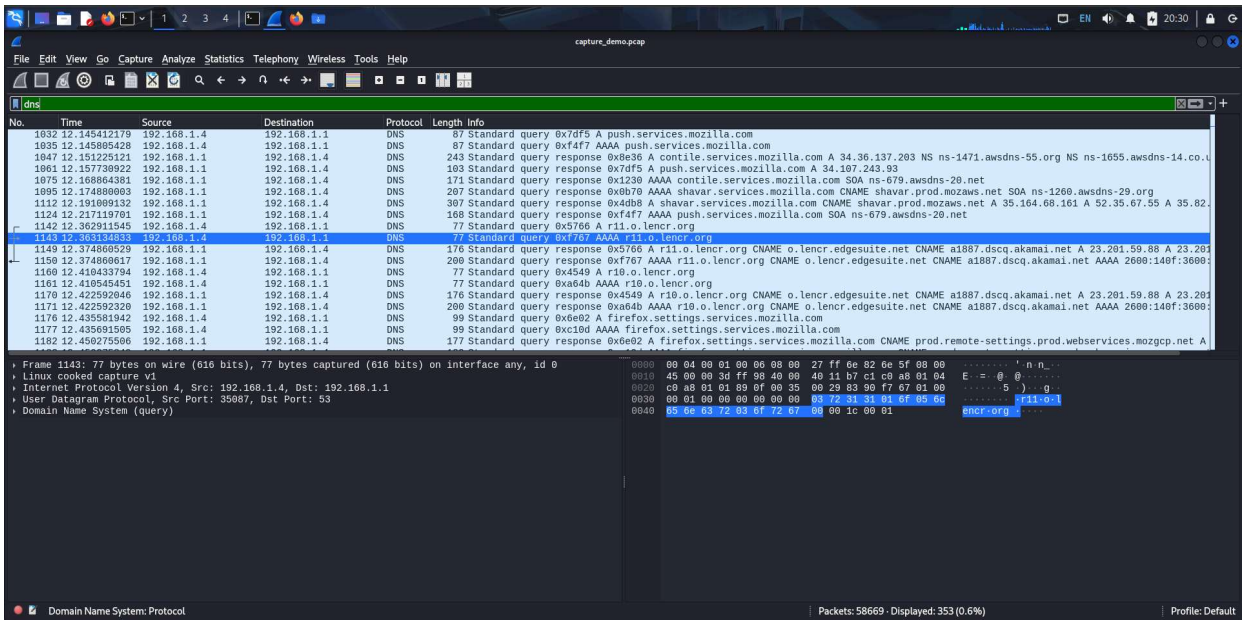
Capture Details

- **Interface Used:** any
- **Duration of Capture:** ~1 minute
- **Total Packets Captured:** 58669
- **Storage Format:** `.pcap` (Portable Capture Format)
- **Traffic Type:** Web browsing and ping-generated traffic

Protocols Identified

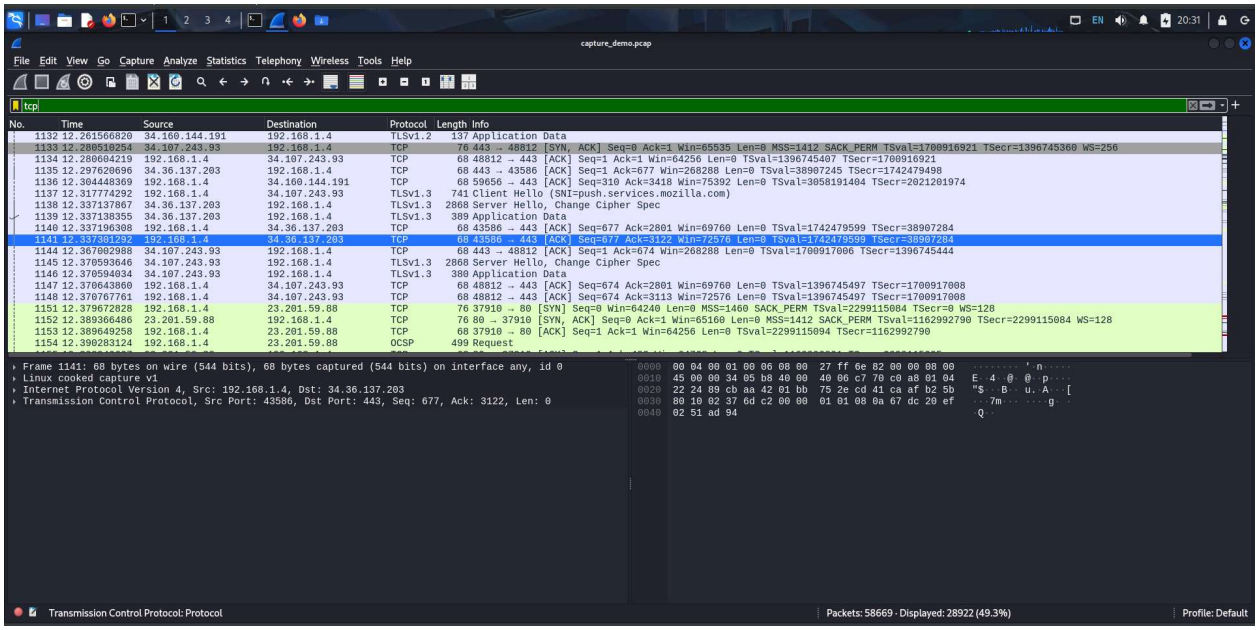
1. **DNS (Domain Name System)**

   Responsible for resolving domain names (e.g., `www.google.com`) into IP addresses. Queries were observed being sent to public DNS servers like Google's `8.8.8.8`.



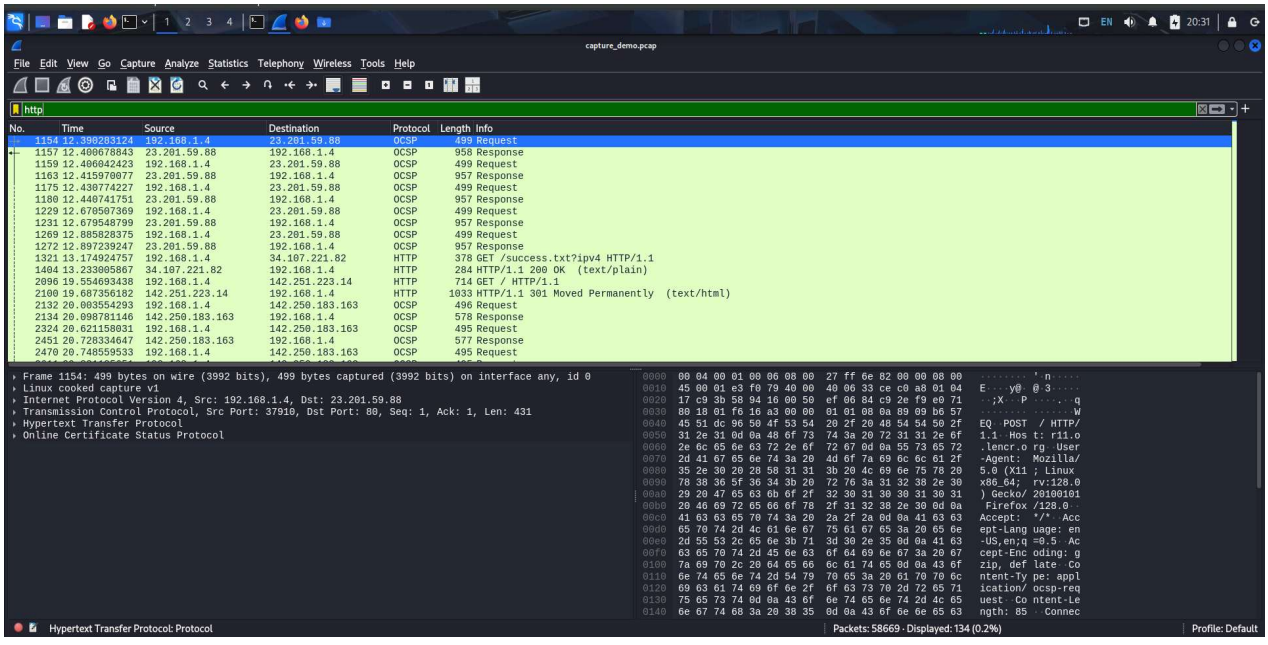
2. **TCP (Transmission Control Protocol)**

   Handled reliable transport-layer communication. Three-way handshakes and session establishments were visible between the client system and multiple web servers.



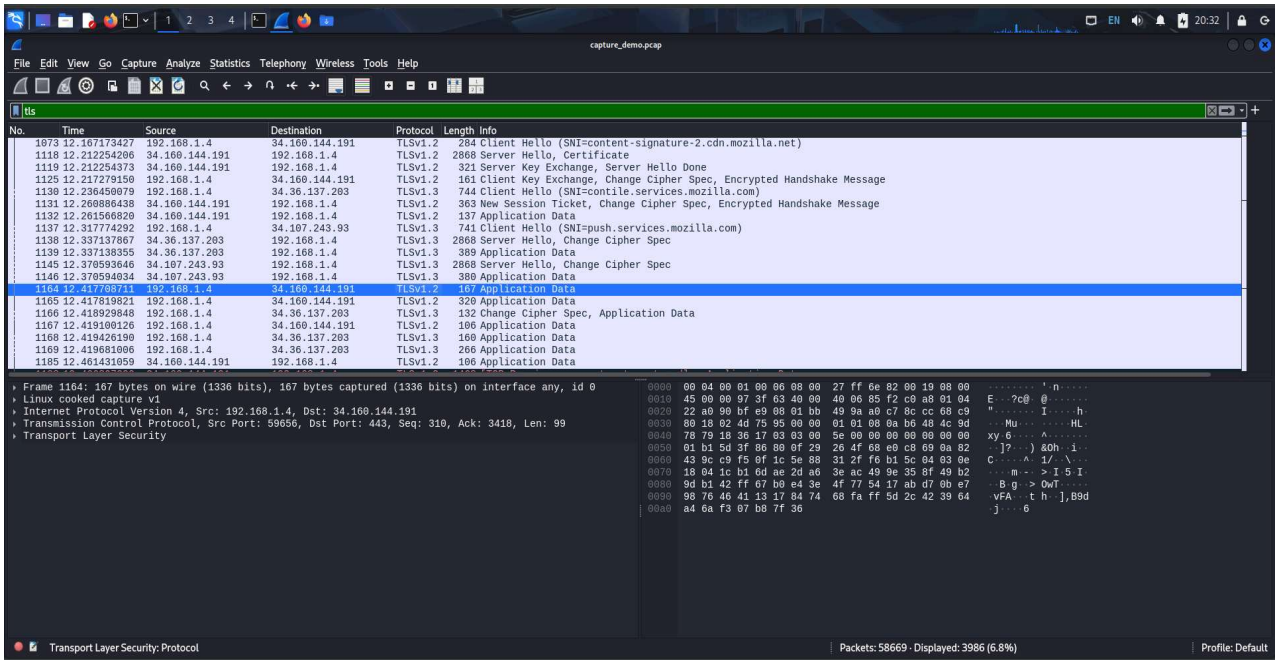
3. **HTTP (HyperText Transfer Protocol)**

   Represented unencrypted communication with web servers. Multiple GET requests and responses were captured during website browsing.



4. **TLS (Transport Layer Security) v1.1/v1.2/v1.3**

   Secure session establishment was observed when accessing HTTPS-enabled websites. TLS Client Hello and Server Hello packets were clearly visible, showing the handshake process and cryptographic negotiation.



Key Observations

- The captured traffic confirmed a standard layered model of communication, progressing from DNS resolution to TCP-based connections and secure TLS-encrypted data transfer.
- The presence of multiple TLS versions indicates interactions with various secure web servers supporting different encryption standards.
- Packet inspection validated the ability of Wireshark to dissect protocol headers and payloads, which is crucial for traffic analysis, troubleshooting, and forensic investigation in cybersecurity.

Conclusion

This Wireshark capture demonstrated real-time web traffic analysis, highlighting multiple protocol layers such as DNS, TCP, HTTP, and encrypted TLS sessions. The exercise provided hands-on insight into how network protocols function together during typical web activity, and how tools like Wireshark can be effectively used in cybersecurity for traffic inspection, protocol analysis, and troubleshooting.