Network Traffic Capture and Analysis Using Wireshark

**Task 5 Report - Cybersecurity Internship Prepared by:** Sankabathula Abhishek

# Methodology and Implementation

**Wireshark Setup and Configuration**

The network packet capture was performed using Wireshark version 4.x on a Windows system connected to a wireless network interface. Wireshark was configured to monitor the active Wi-Fi adapter in promiscuous mode, enabling capture of all network traffic passing through the interface  3 4  .

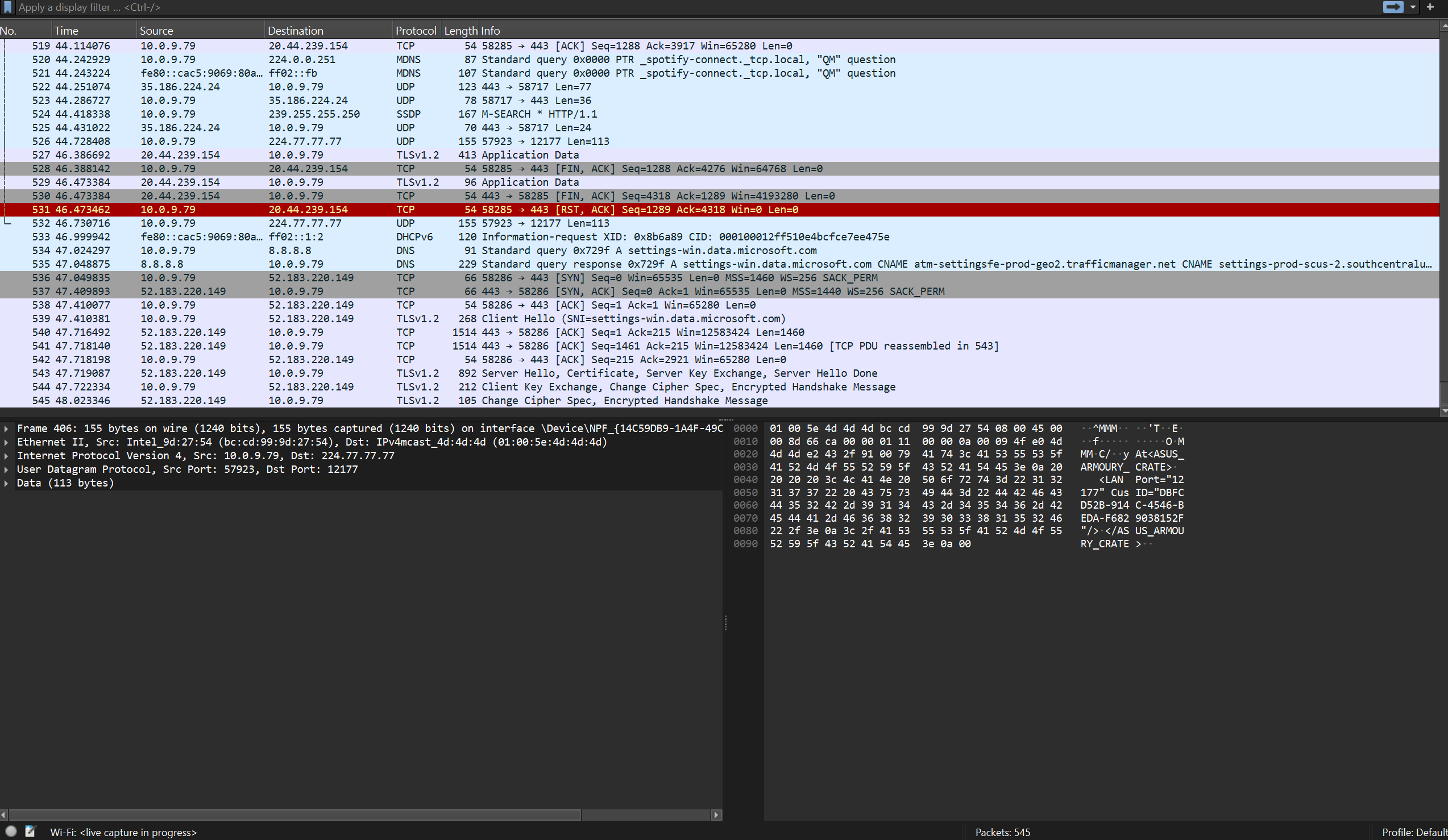
**Key Configuration Parameters:**

 **Interface:** Wi-Fi network adapter

 **Capture Mode:** Promiscuous mode enabled

 **Duration:** Approximately 1 2 minutes of active capture

* **Filter:** No initial capture filters applied to ensure comprehensive data collection



# Traffic Generation Methodology

To generate diverse network traffic for analysis, several activities were performed during the capture session:

**Web Browsing:** Visited multiple websites to generate HTTP/HTTPS traffic

**DNS Queries:** Performed domain name lookups for various services

**Background Applications:** Allowed automatic system updates and service communications

**Network Discovery:** Enabled network service discovery protocol

# Protocol Analysis and Findings

**Transport Layer Protocols**

**Transmission Control Protocol TCP**

TCP emerged as the dominant transport protocol in the capture, handling the majority of reliable data transmission. The analysis revealed proper TCP three-way handshake sequences SYN, SYN ACK, ACK) for connection establishment, followed by data exchange and connection teardown procedures 7 8 .

**Key TCP Observations:**

 Multiple concurrent connections to various servers

 Sequence and acknowledgment numbers properly incremented  Window size management for flow control

 Proper connection termination with FIN/ACK sequences

 Port numbers ranging from ephemeral client ports 1024 65535) to well-known service ports

**User Datagram Protocol UDP**

UDP traffic was observed primarily for DNS queries, multicast DNS (mDNS), and Simple Service Discovery Protocol SSDP) communications. The connectionless nature of UDP made it suitable for these lightweight, query- response interactions 9 10 .

**UDP Traffic Characteristics:**

 Primarily used for DNS queries on port 53

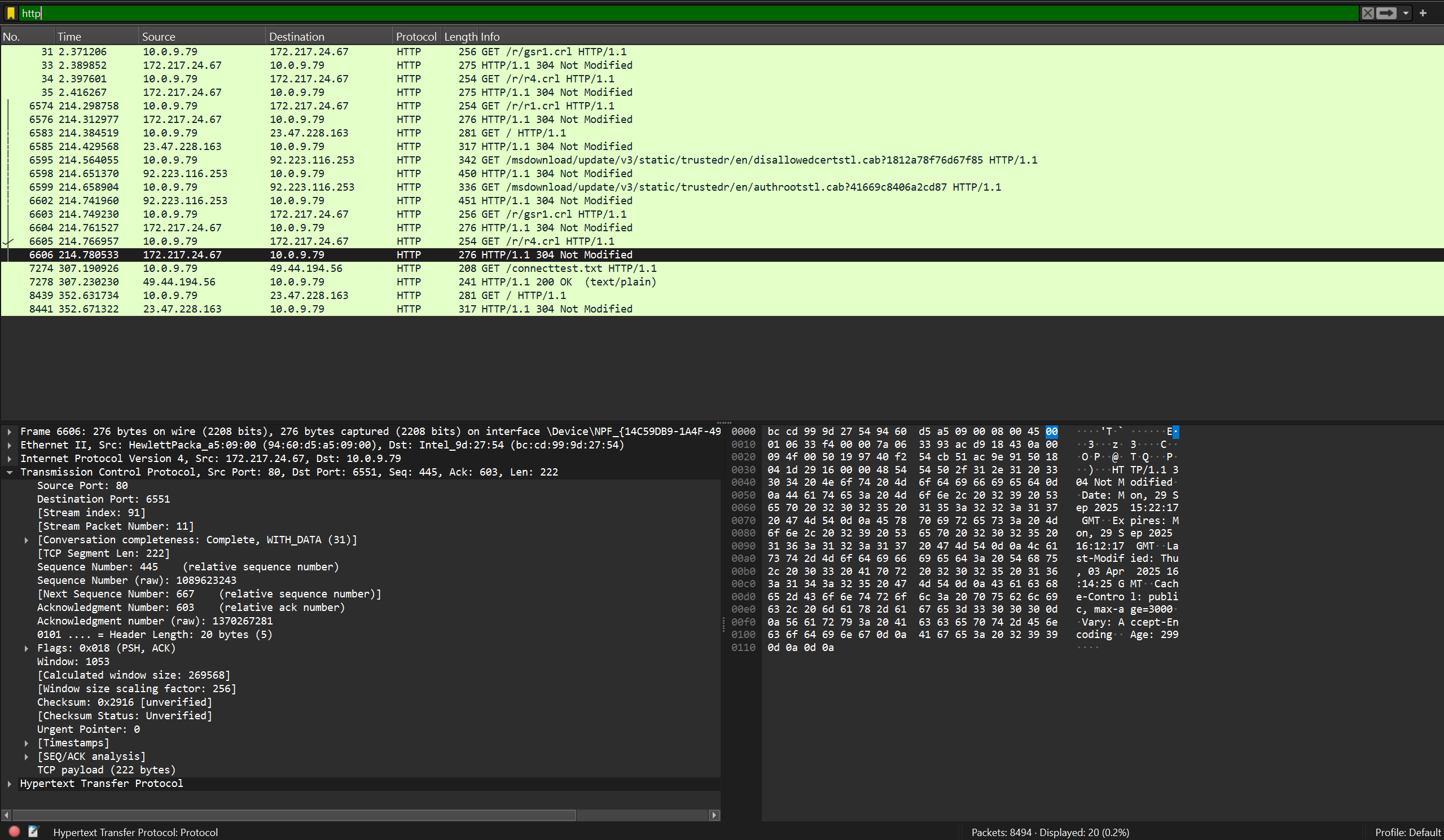
 mDNS traffic on port 5353 for local network service discovery  SSDP traffic on port 1900 for UPnP device discovery

 No connection establishment overhead, enabling faster communication

# Application Layer Protocols

**Hypertext Transfer Protocol HTTP/HTTPS**

Web traffic constituted a significant portion of the captured data, with both HTTP (port 80) and HTTPS (port 443) connections observed. The analysis showed typical web browsing patterns including GET requests, response codes, and content retrieval



**HTTP Traffic Analysis:**

 Multiple GET requests to various domains

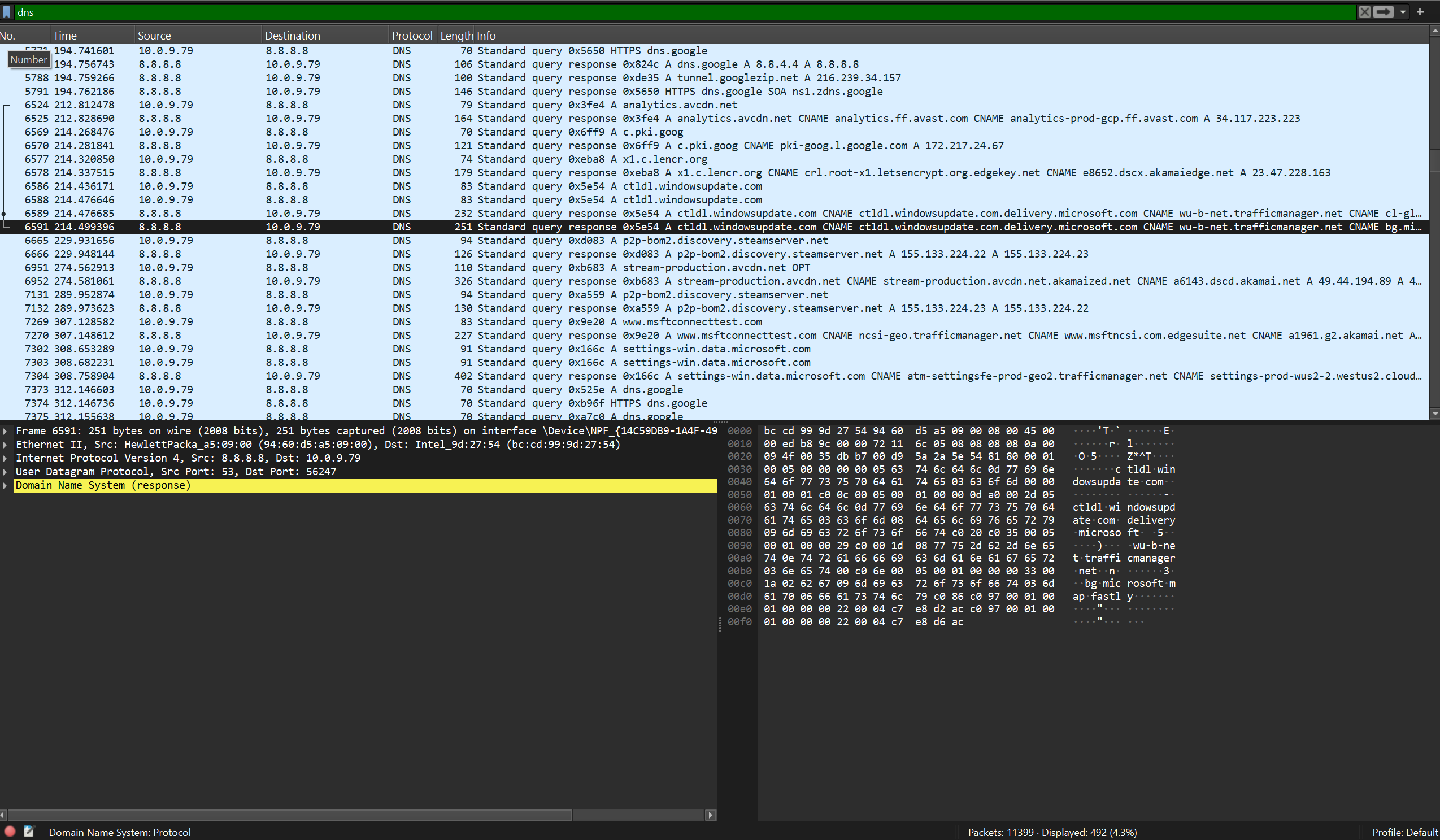
 HTTP/1.1 protocol version predominantly used

 Status codes including 200 OK , 304 Not Modified), and others  User-Agent strings revealing browser information

 Content-Type headers indicating various media types

**Domain Name System DNS**

DNS traffic showed extensive domain name resolution activities, with queries directed to multiple DNS servers. Both standard DNS queries and responses were captured, revealing the domain resolution process



**DNS Query Patterns:**

 Queries for Google services ([googleapis.com](http://googleapis.com/), [google.com](http://google.com/))

 Microsoft domain resolutions ([microsoft.com](http://microsoft.com/), [windows.com](http://windows.com/))  Content delivery network domains

 Standard A, AAAA, and CNAME record queries

 Response times typically under 50ms for cached entries

# Security Protocols

**Transport Layer Security TLS/SSL**

Secure communications were heavily represented through TLS/SSL encrypted connections, primarily over port

443. The analysis showed proper SSL/TLS handshake procedures and encrypted data exchange

**TLS Implementation Observations:**

 TLS 1.2 and 1.3 versions observed

 Proper certificate exchange during handshake  Cipher suite negotiation completed successfully

 Application data encrypted after handshake completion

# Network Traffic Statistics and Patterns Packet Distribution Analysis

The captured network session yielded comprehensive traffic data across multiple protocol layers: **Capture Statistics:**

 **Total Packets:** 6,806+ packets captured

 **Packet Size Range:** 54 1,514 bytes (standard Ethernet frame sizes)  **Average Packet Rate:** Approximately 60 70 packets per second

 **Protocol Diversity:** 8 distinct protocols identified

# Traffic Volume by Protocol

Analysis of the packet capture revealed the following protocol distribution patterns: **High-Frequency Protocols 100 packets):**

 TCP Dominant protocol for reliable data transmission  HTTP/HTTPS Substantial web traffic volume

 DNS Frequent domain name resolution queries

 TLS/SSL Encrypted connection establishment and data transfer

**Medium-Frequency Protocols 20 100 packets):**

UDP Supporting DNS and service discovery functions

 mDNS Local network service advertisement and discovery

**Low-Frequency Protocols 20 packets):**

 SSDP Occasional UPnP device discovery

 DHCP Minimal network configuration traffic

# Security Analysis and Observations Network Behavior Assessment

The captured traffic exhibited normal, legitimate network behavior patterns consistent with typical user activities. No suspicious or malicious activity was detected during the analysis period 15 16 .

**Security Indicators:**

 All DNS queries resolved to legitimate, well-known domains  TLS connections showed proper certificate validation

 No evidence of port scanning or reconnaissance activities

 HTTP requests targeted legitimate web services and content delivery networks

# Protocol Security Implementation

The analysis revealed proper security protocol implementation across multiple layers: **Transport Security:**

 TLS encryption properly implemented for sensitive communications  Certificate validation processes functioning correctly

 No plaintext transmission of sensitive data observed

**Network Security:**

 DNS queries directed to trusted resolver servers

 No suspicious domain generation algorithm DGA) patterns detected

 Standard network service discovery protocols operating within normal parameters

# Technical Implementation Details

**Packet Filtering and Analysis Techniques**

The Wireshark analysis employed various filtering techniques to isolate and examine specific protocol types: **Display Filters Used:**

tcp - Isolated TCP traffic for connection analysis http - Filtered HTTP requests and responses dns - Examined domain name resolution traffic

ip.addr == [specific IP] - Targeted traffic to/from specific hosts

# Protocol Header Analysis

Detailed examination of protocol headers revealed proper implementation of network standards: **TCP Header Elements:**

 Source and destination port numbers correctly assigned

 Sequence and acknowledgment numbers properly managed  Control flags SYN, ACK, FIN, RST) used appropriately

 Window size values indicating proper flow control

**IP Header Information:**

 IPv4 addresses properly formatted and routable

 Time-to-Live TTL) values appropriate for network distance  Protocol field correctly identifying encapsulated protocols

 Header checksums validating packet integrity

# Learning Outcomes and Skills Development Technical Skills Acquired

This Wireshark analysis exercise successfully developed several critical cybersecurity competencies: **Network Analysis Skills:**

 Packet capture configuration and execution  Protocol identification and classification

 Traffic pattern recognition and analysis  Network troubleshooting methodologies

**Security Analysis Capabilities:**

 Abnormal traffic pattern detection  Protocol anomaly identification

 Security protocol verification

 Network baseline establishment

# Professional Application

The skills demonstrated in this exercise have direct applications in cybersecurity roles:

**Incident Response:** Ability to capture and analyze network traffic during security incidents provides crucial evidence for threat hunting and forensic analysis

**Network Security Monitoring:** Understanding normal traffic patterns enables detection of anomalous behavior that may indicate security threats or system compromises.

**Vulnerability Assessment:** Protocol analysis skills support identification of insecure communications and potential attack vectors within network infrastructure.

# Conclusion

**Key Findings Summary**

The Wireshark packet capture and analysis exercise successfully achieved all stated objectives:

**Comprehensive Traffic Capture:** Over 6,806 packets captured representing diverse network activity

**Protocol Identification:** Eight distinct network protocols successfully identified and analyzed

**Security Assessment:** Normal, secure network behavior confirmed through detailed analysis

**Technical Proficiency:** Demonstrated competency in professional network analysis tools and techniques