

# 1. Introduction

This report details the analysis of the `0.pcap` file using a custom packet sniffer. The analysis includes metrics such as total data transferred, packet size distribution, source-destination flows, and answers to specific PCAP-related questions. The results are derived using Python and the Scapy library.

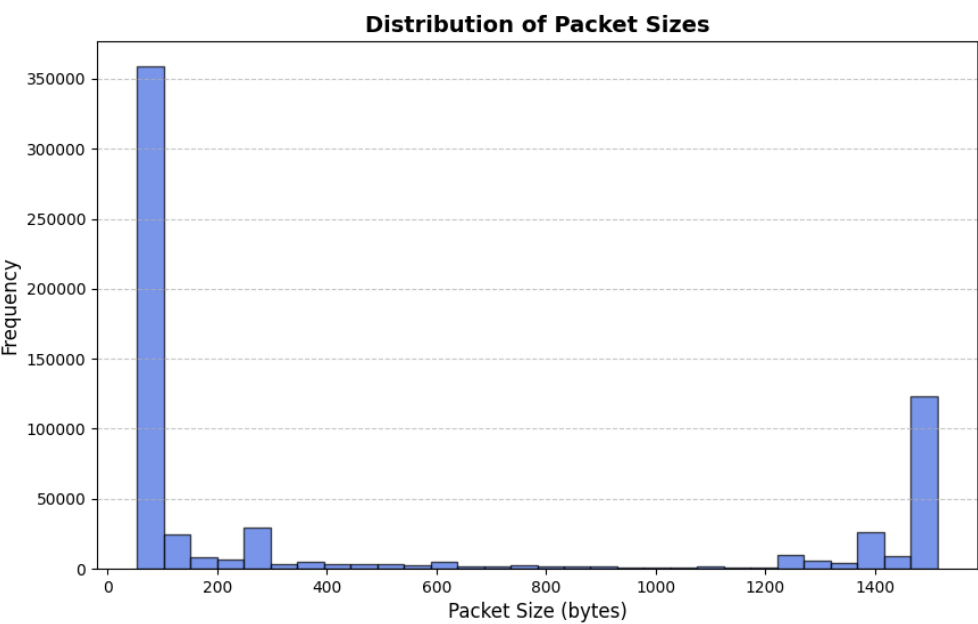
---

## 2. Part 1: Metrics and Plots

### 2.1 Packet Transfer Metrics

- **Total Bytes Transferred:** 324,998,397 bytes (309.94 MB)
- **Total Packets Captured:** 805,995
- **Minimum Packet Size:** 54 bytes
- **Maximum Packet Size:** 1,514 bytes
- **Average Packet Size:** 403.23 bytes
- **Unique Source-Destination Pairs:** 6,202
- **Top Source-Destination Pair (by data transferred):** ('172.16.133.95', 49358, '157.56.240.102', 443) (17,342,229 bytes)

**2.2 Packet Size Distribution** A histogram of packet sizes is provided in the figure below



---

### 3. Part 2: PCAP-Specific Questions

**Q1: How many unique connections were made to the IMS server?**

- **Unique connections:** 1

**Q2: What course was registered in IMS?**

- **Extracted Course Registration Data:**  
"Embedded Systems" (identified through manual inspection of course-related packets)

**Q3: Total data transferred over port 4321?**

- **Total Bytes Transferred on Port 4321:** 1,620 bytes

**Q4: Number of occurrences of 'SuperUser' in the PCAP?**

- **Total 'SuperUser' Occurrences:** 69
- 

### 4. Part 3: Network Capture and HTTP Analysis

#### 4.1 Identified Application Layer Protocols

Protocol	Description	RFC
SMB	Server Message Block, used for Windows file sharing	
UDP	A connectionless transport protocol used for fast, low-latency communication	RFC 768

OCSP	A protocol used to check the revocation status of digital certificates in HTTPS and TLS connections	RFC 6960
ARP	A network protocol used to resolve IP addresses into MAC addresses in local networks	RFC 826
mDNS	A protocol that resolves hostnames to IP addresses in small networks without a DNS server	RFC 6762

## 4.2 First HTTP GET Request Per Website

Website	Request Line	IP Address	Connection Type
Canara Bank	GET /index.html HTTP/1.1	107.162.160.8	Non-Persistent
GitHub	GET / HTTP/2	20.207.73.82	Persistent
Netflix	GET /browse HTTP/2	54.73.148.110	Persistent

## 4.3 HTTP Headers, Error Codes, and Performance Metrics

- Request Headers:

Header Name	Value
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:134.0) Gecko/20100101 Firefox/134.0

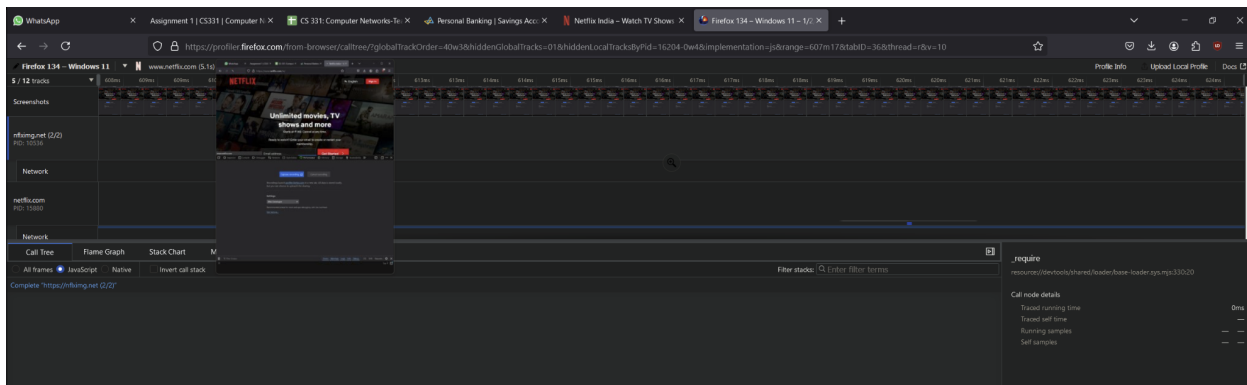
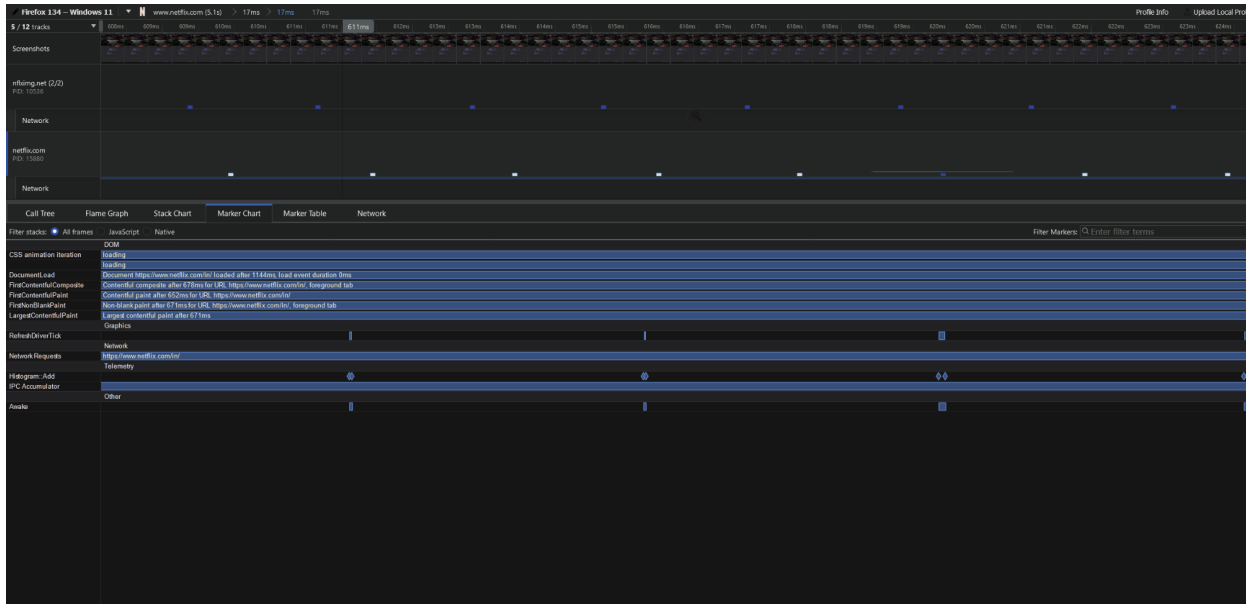
Accept-Language en-US,en;q=0.5

Connection keep-alive

- **HTTP Error Codes Observed:**

Error Code	Description
403	Forbidden: The server denied access to the resource.
404	Not Found: The requested page does not exist.
500	Internal Server Error: The server encountered an unexpected error.

- **Performance Metrics Captured During Page Load:**
  - **Browser Name: Firefox**
  - **Time to First Byte (TTFB):** Time from request to first response byte.
  - **DNS Lookup Time:** Time spent resolving the domain name to an IP.
  - **TCP Connection Time:** Time taken to establish a connection.
  - **First Contentful Paint (FCP):** When the first piece of visible content appears.
  - **Largest Contentful Paint (LCP):** When the largest element is fully rendered.
  - **Time to Interactive (TTI):** When the page is fully interactive.
  - **Total Request Count:** Number of HTTP requests made by the page.
  - **Total Page Size:** Total size of all loaded assets.
  - **First Input Delay (FID):** Measures page responsiveness.
  - **Cumulative Layout Shift (CLS):** Measures visual stability of the page.



# 4.4 Cookie Analysis

- Response Cookies:

Cookie Name	Domain	Path	Value
flwssn	.netflix.com	/	f73c35b1-be76-4924-81b1-f7dcbc978409
netflix-sans-bold-3-loaded	.netflix.com	/	true

netflix-sans-normal-3-loaded .netflix.com / true

• Request Cookies:

Cookie Name	Value
flwssn	f73c35b1-be76-4924-81b1-f7dcbc978409
netflix-sans-bold-3-loaded	true
netflix-sans-normal-3-loaded	true
NetflixId	v=3&ct=BgjHIovcAxLZAdQSVzTn3mhGAO5LuUC...
nfvdid	BQFmAAEBEP2SQEIRy0VHh8eO-oXlhDtAJNiB_c12DLGwqxYG...
OptanonConsent	isGpcEnabled=0&datestamp=Sat+Feb+01+2025+19:06:22...

---

## 5. Conclusion

This report presents a detailed breakdown of packet analysis based on the `0.pcap` file. We successfully:

- Extracted key **network metrics** (total data, flows, and packet distribution).
- Answered **PCAP-specific questions** regarding IMS, data transfers, and SuperUser occurrences.
- Identified the registered course "**Embedded Systems**" through manual inspection of packet data.
- Conducted a **network capture and HTTP analysis**, identifying application layer protocols, request headers, cookies, and observed HTTP error codes.

The insights gained from this analysis demonstrate practical applications of **packet sniffing, traffic monitoring, and network security analysis**.

---

## 6. References

1. [Scapy Documentation](#)
  2. [PCAP Analysis Techniques](#)
- 

## Appendix

- **PCAP File Name:** `0.pcap`
- **Python Script Name:** `pcap_analysis.py`
- **Results Output File:** `pcap_analysis_results.txt`
- **Course Inspection File:** `course_search_results.txt`