Computer Networks: Assignment 1

Nimitt

Sawale Sumeet Shivaji

GitHub Repository

- Link to GitHub Repository: https://github.com/nimittnim/snifox

Part I

Implementation

- The objective of this part is to build a packet sniffer that can access and analyse network packets.
- We implemented the sniffer using **socket and struct** modules to parse and analyse the packets that analyses the packets live until stopped.
- It keeps track of packet in three dictionaries that store source-destination address and then computes overall traffic metrics like mean, max and min packet size.
- We use **tcpreplay** to replay the network packets as in file 6.pcap.

Link to code

- Main Sniffer
- Running tcpreplay and sniffer

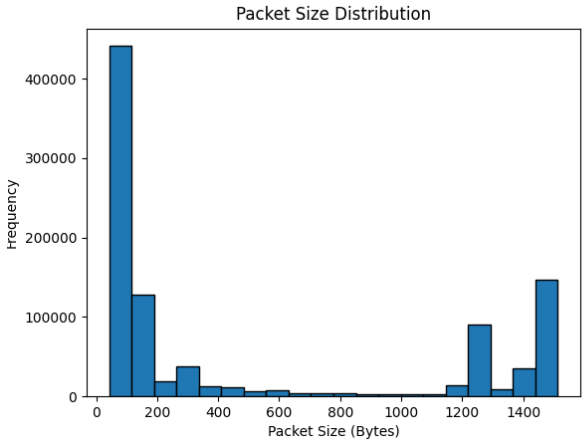
The following block shows the typical output from the sniffer:

```
--------------------------------------------------
Sniffer Started
--------------------------------------------------
Sniffer Closed
Sniffing Stats:
Total Data: 515832534 bytes
Total Packets: 982137
Min Size: 42, Max Size: 1514, Avg Size: 525.21
Top Data Transfer Pair: ('180.149.61.76', '10.240.0.41'): 52695453 bytes
Distribution of Packet Sizes Histogram Stored at results/packet_size_distribution.png
Flow data with IPs stored at results
Thank You!
--------------------------------------------------
```

Results

Traffic Stats

1. Total Data Transferred = **515462763** bytes
2. Total number of Packets transferred = **981722**
3. Distribution of Packet Size:



We can notice packets of size **40 - 200 bytes** and **1300 - 1500 bytes** are most frequently transferred.

4. Minimum Packet Size = 42 bytes
5. Maximum Packet Size = 1512 bytes
6. Mean Packet Size = 525 bytes

Unique Source-Destination Pairs

The linked file displays each of the pairs and amount of data shared between them.

1. Total Unique Source-Destination Pairs = **9450**

This may vary from the actual number as the sniffer also captured the default packets as noise.

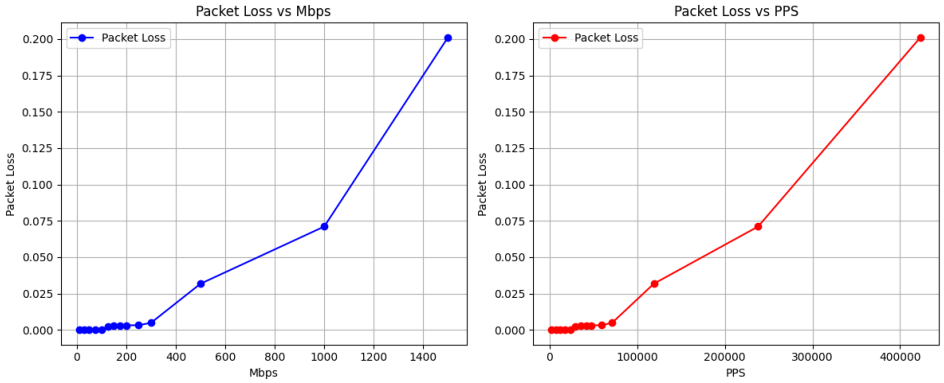
Flow Statistics

1. The linked file displays the sources with their IP address and corresponding total flow.
2. The linked file displays the destinations with their IP address and corresponding total flow.
3. The (**180.149.61.76 — 10.240.0.41)** pair transferred the most data: **52686409** bytes

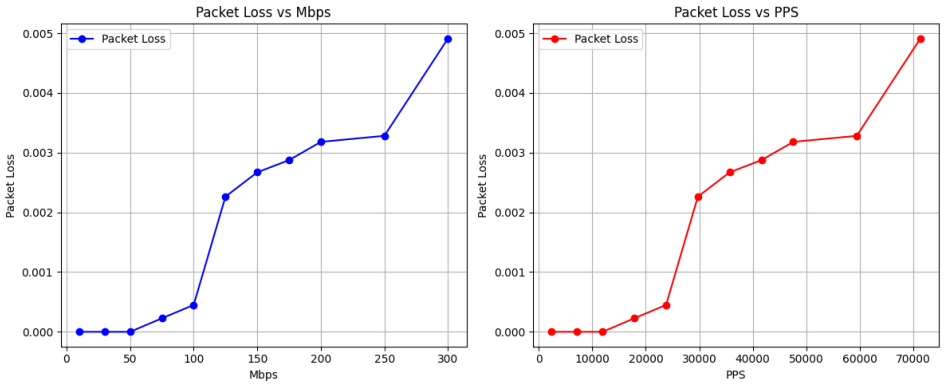
Packet Loss Statistics

1. **Running Sniffer and Traffic replay on same VM :**

Firstly, we ran sniffer and replay on the same machine: **Ubuntu 20.04 QEMU 9.1 ARM Virtual Machine.** The following plots describe the packet loss as we configure replay speed:



Looking closely,



We can note that around speed of **50 MBPS** and **12000 PPS, t**he sniffer starts to miss some transferred packets.

2. **Running Sniffer and Traffic replay on two different VMs:**

Then, we ran sniffer and replay on two different Virtual Machines: **Ubuntu 20.04 QEMU 9.1 ARM VMs** connected using bridged network over WiFi.

This setup also gave similar results as packet loss increased as replay was configured at higher speed. We noted that around speed of **30 MBPS** and **70000 PPS,** the sniffer starts to miss some transferred packets.

Part II

Implementation:

1. **Finding IP address of the attacker's phishing page**:
   - The attacker uses DNS spoofing to redirect secure-bank.com to his private ip.
   - We need to check for DNS response with "secure-bank.com" in the query name.
   - The IP we get should also be private.
2. **Username of the victim:**
   - We look for a packet with POST request that is directed towards "secure-bank.com", i.e. the phishing IP.
   - We look for strings like "username" and "password" in the url encoded format.
3. **Getting information about the attacker:**
   - We know that the attacker sent an email using their own machine in plain-text.
   - We look for packets with source ip equal to the phishing ip.
   - We check for SMTP traffic with Raw data.
   - We find an email from the attacker.

Results:

1. IP address of the phishing page: **192.168.1.100**
2. Victim's information :
   a. username = **alexwell**
   b. password = **bemymaxwell**
3. Information about the attacker:
   a. Name of the attacker: **Chris Martin**
   b. Email address of the attacker: **ab@iitgn.ac.in**

c. Email subject: CS331 - **Enrollment**

d. Email body: **Can I get enrolled into CS331? I need it for my graduation.**

Part III

1. 5 Application Layer Protocols not discussed in class

1. **SNMP (Simple Network Management Protocol)**
   - **Operation/Usage**: Monitors and manages network devices (routers, switches) by collecting data and configuring parameters.
   - **Layer**: Application layer.
   - **RFC**: RFC 1157 (SNMPv1), updated by later RFCs for newer versions.
2. **SIP (Session Initiation Protocol)**
   - **Operation/Usage**: Establishes, modifies, and terminates multimedia sessions (e.g., VoIP calls, video conferencing).
   - **Layer**: Application layer.
   - **RFC**: RFC 3261.
3. **NTP (Network Time Protocol)**
   - **Operation/Usage**: Synchronizes clocks across networked devices to ensure accurate timekeeping.
   - **Layer**: Application layer.
   - **RFC**: RFC 5905.
4. **LDAP (Lightweight Directory Access Protocol)**
   - **Operation/Usage**: Accesses and maintains distributed directory information (e.g., user accounts in an organization).
   - **Layer**: Application layer.
   - **RFC**: RFC 4511.
5. **MQTT (Message Queuing Telemetry Transport)**
   - **Operation/Usage**: Enables lightweight publish-subscribe messaging for IoT devices in low-bandwidth environments.
   - **Layer**: Application layer.
   - **RFC**: RFC 7252.

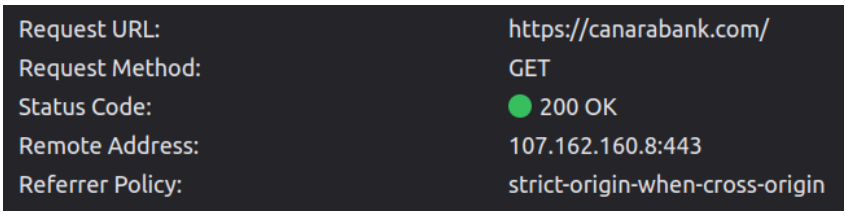
2. Analysing websites: Browser - Brave

Part a. Request Headers : Results

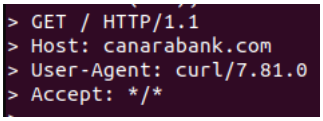
Website	Request Line	Version of Application layer protocol	IP address	Is the connection persistent
canarabank.com	GET	HTTP/1.1	107.162.160.8	Persistent (keep-alive)
github.com	GET	HTTP/2	20.207.73.82	Persistent (Default)
netflix.com	GET	HTTP/2	44.242.60.85	Persistent (Default)

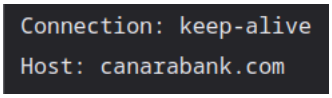
Proof / Source of the data above

1. **Canara Bank**



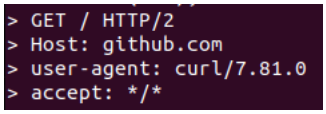
After running **“curl -v https://canarabank.com”**

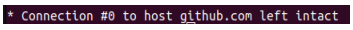




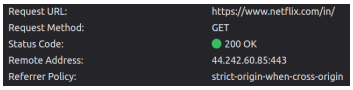
2. **GitHub**

After running **“curl -v https://github.com”**

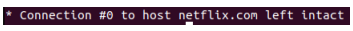




3. **Netflix**



After running **“curl -v https://netflix.com”**



Part b. Header Fields and HTTP error codes

This is for github.com

Request Header

Header Filed	Value
:authority	github.com
:scheme	https
cache-control	max-age=0

Response Header

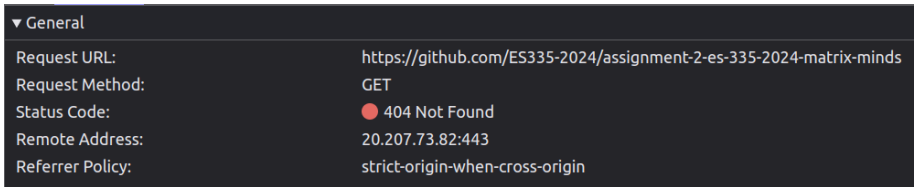
Header Filed	Value
cache-control	max-age=0, private, must-revalidate
content-encoding	gzip

HTTP Error Codes

Header Filed	Value
server	GitHub.com

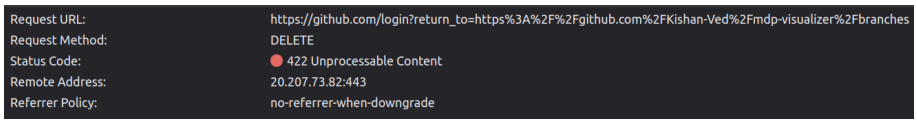
1. 404 Not Found:

   I Tried opening a private repo in github without signing in.



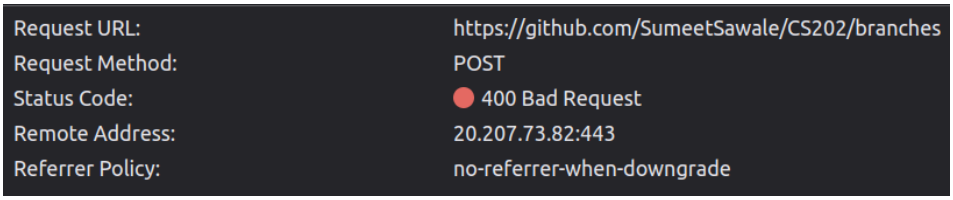
2. 422 Unprocessable Content:

   I tried deleting a branch of a repository without having the permission.

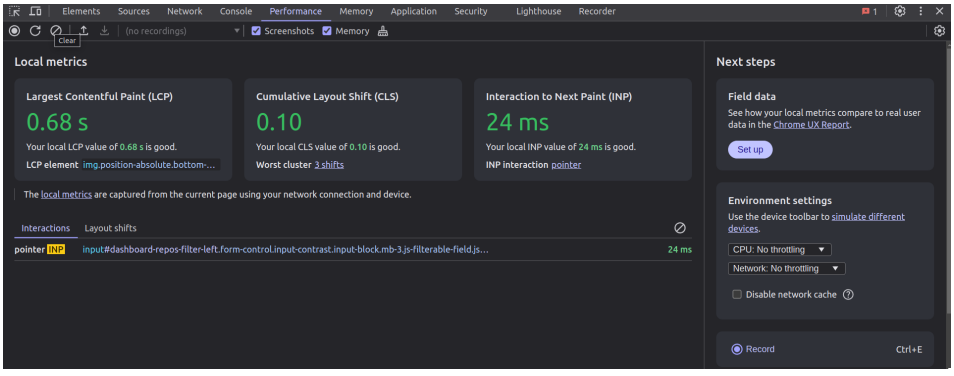


3. 400 Bad Request:

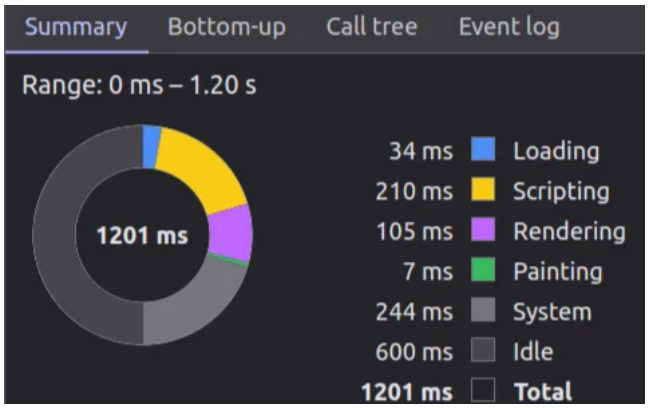
   I tried creating a branch with no name.



Part c. Performance Metrics for “github.com”:



- LCP = 0.6 s
- CLS = 0.1 s
- INP = 24 ms



List of Cookies for "github.com":

1. **Response Header**
   - Set-Cookie :

   _gh_sess=XQKp1BhDHgdiKSG6AR82ObOgejHxQtbl9zBpNZgbAA4GuXlTUhcxubKWgk50yVUzkkxrORj2Z4mA4kan3ZuBF
   -X0U%2BJ0x2F1DlrAa2--XacbWFWVM6cJfHfHzNWrJw%3D%3D; path=/; secure; HttpOnly; SameSite=Lax
2. **Request Header**
   - Cookie : _octo=GH1.1.895787376.1723207687; _device_id=9f52446374bf74ca30cf2cb5183cf3d8; saved_user_sessions=117972328%3Aib83prvEiHTKNA9XxiIvKqPltPcrTfe1Bbw6b658git7bBlg; user_session=ib83prvEiHTKNA9XxiIvKqPltPcrTfe1Bbw6b658git7bBlg; __Host-user_session_same_site=ib83prvEiHTKNA9XxiIvKqPltPcrTfe1Bbw6b658git7bBlg; logged_in=yes; dotcom_user=SumeetSawale; color_mode=%7B%22color_mode%22%3A%22auto%22%2C%22light_theme%22%3A%7B%22name%22%3A%22 cpu_bucket=lg; preferred_color_mode=dark; tz=Asia%2FCalcutta; _gh_sess=JPGRj2eVLhPRmYoi2QRwm0flJPKfA8HwWGUB%2F1fpzkPVfvv2XXKDcVEsCujW78B6w3T%2BvpgzkHpb -3XZZd%2B%2BtVIXM29M%2F--8WCd1ijyiKiwczd%2BenvSEA%3D%3D

Acknowledgement

We would like to express our sincere gratitude to Prof. Sameer Kulkarni for providing us with the opportunity to work on this assignment. We also appreciate the valuable assistance and support from the teaching assistants.