COMPUTER NETWORK'S
ASSIGNMENT- 7

Apoorv Gupta
AIML A1
21070126018

Aim -

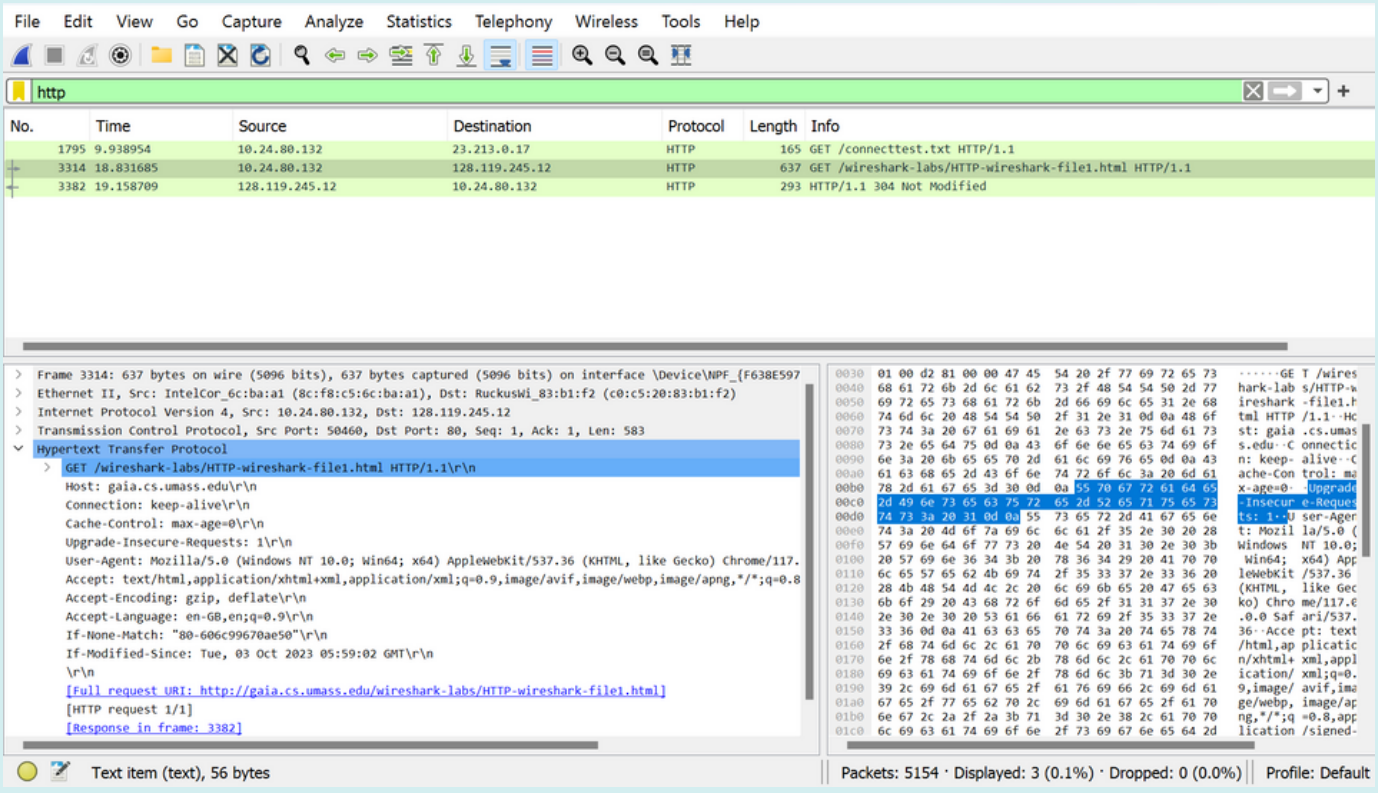
Packet Capturing and Analysis with Wireshark

Theory -:

Wireshark is a widely used network protocol analyzer that allows users to capture and inspect data packets on a computer network. It is a powerful tool for network administrators, security professionals, and developers to analyze and troubleshoot network issues, monitor network performance, and investigate security incidents. When it comes to understanding Wireshark and the HTTP protocol, here's a brief overview:

1. **Wireshark Basics:**
   - **Wireshark captures and displays the data traveling back and forth on a network in real-time.**
   - **It supports a wide range of network protocols, making it a versatile tool for analyzing various types of traffic.**
2. **HTTP Protocol:**
   - **HTTP (Hypertext Transfer Protocol) is an application layer protocol used for transferring data, typically in the form of web pages, between a client (usually a web browser) and a server (usually a web server).**
   - **It operates over TCP (Transmission Control Protocol) on port 80 for unencrypted connections and port 443 for encrypted connections (HTTPS).**
   - **HTTP follows a client-server model, where the client sends requests to the server, and the server responds with the requested data.**
3. **Using Wireshark to Analyze HTTP Traffic:**
   - **Wireshark can capture HTTP traffic, allowing you to inspect the details of HTTP requests and responses.**
   - **When capturing HTTP traffic in Wireshark, you can apply various filters to focus on specific HTTP packets, such as filtering by source/destination IP addresses, ports, or specific HTTP methods (e.g., GET, POST).**
   - **Wireshark can dissect HTTP packets to reveal information like headers, URL paths, status codes, and payload data.**

4. **Common Use Cases:**
    - **Troubleshooting: Wireshark helps identify issues with HTTP requests or responses, such as errors, slow performance, or missing resources.**
    - **Security Analysis: It can be used to detect potential security threats like malicious traffic, unauthorized access, or data leaks.**
    - **Performance Monitoring: Wireshark can track the time taken for HTTP requests and responses, helping diagnose slow-loading web pages.**
    - **Protocol Debugging: Developers use Wireshark to debug and optimize applications that rely on HTTP communication.**
5. **HTTPS Traffic:**
    - **While Wireshark can capture HTTPS traffic, it cannot decrypt the encrypted content because it relies on TLS/SSL encryption. Decrypting HTTPS traffic requires access to the server's private key or the use of other specialized tools.**



Self Assessment:

Q1) Please note down the IP address of your machine and the destination machine (gaia.cs.umass).

```
Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : sitin.sitpune.edu.in
   Link-local IPv6 Address . . . . . : fe80::4631:a447:acd8:6d22%17
   IPv4 Address. . . . . . . . . . . : 10.24.80.132
   Subnet Mask . . . . . . . . . . . : 255.255.224.0
   Default Gateway . . . . . . . . . : 10.24.64.1
```

No.	Time	Source	Destination	Protocol	Length	Info
1795	9.938954	10.24.80.132	23.213.0.17	HTTP	165	GET /connecttest.txt HTTP/1.1
3314	18.831685	10.24.80.132	128.119.245.12	HTTP	637	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
3382	19.158709	128.119.245.12	10.24.80.132	HTTP	293	HTTP/1.1 304 Not Modified

- **SOURCE: 10.24.80.132**
- **DESTINATION: 128.119.245.12**

Q2) What do you observe in the HTTP request message.

1. **Time**
2. **Source**
3. **Destination**
4. **Protocol**
5. **Length**
6. **Info**

Time	Source	Destination	Protocol	Length	Info

Q3) Write down the details of the HTTP response message such as status code, content length and file modified last time.

```
> Frame 1212: 637 bytes on wire (5096 bits), 637 bytes captured (5096 bits) on interface \Device\NPF_{F638E597-007B-41B1-8AD9-4912952D699C}, id 0
> Ethernet II, Src: IntelCor_6c:ba:a1 (8c:f8:c5:6c:ba:a1), Dst: RuckusWi_83:b1:f2 (c0:c5:20:83:b1:f2)
> Internet Protocol Version 4, Src: 10.24.80.132, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 50818, Dst Port: 80, Seq: 2, Ack: 1, Len: 583
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-GB,en;q=0.9\r\n
    If-None-Match: "80-606c99670ae50"\r\n
    If-Modified-Since: Tue, 03 Oct 2023 05:59:02 GMT\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
    [HTTP request 1/1]
    [Response in frame: 1244]
```

1. Content Length: 637

Length
637

2. Modified last time :

```
If-Modified-Since: Fri, 06 Oct 2023 05:59:02 GMT
\r\n
```

3. status code:

```
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
```

```
01 01 5e 9e 00 00 47 45  54 20 2f 77 69 72 65 73   ··^···GE T /wires
68 61 72 6b 2d 6c 61 62  73 2f 48 54 54 50 2d 77   hark-lab s/HTTP-w
69 72 65 73 68 61 72 6b  2d 66 69 6c 65 31 2e 68   ireshark -file1.h
74 6d 6c 20 48 54 54 50  2f 31 2e 31 0d 0a 48 6f   tml HTTP /1.1··Ho
```

```
[Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n]
Request Method: GET
Request URI: /wireshark-labs/HTTP-wireshark-file1.html
Request Version: HTTP/1.1
t: gaia.cs.umass.edu\r\n
```

Q4) Write down your interesting observations for the GET request and response messages.

- HTTP Method: The request uses the HTTP method "GET," indicating that the client wants to retrieve the specified resource from the server.
- Resource Path: The path "/wireshark-labs/HTTP-wireshark-file1.html" is the specific resource being requested. This path is relative to the server's root directory or the current context.
- HTTP Version: The request specifies the HTTP version as "HTTP/1.1." This version of the HTTP protocol is commonly used for web communication.

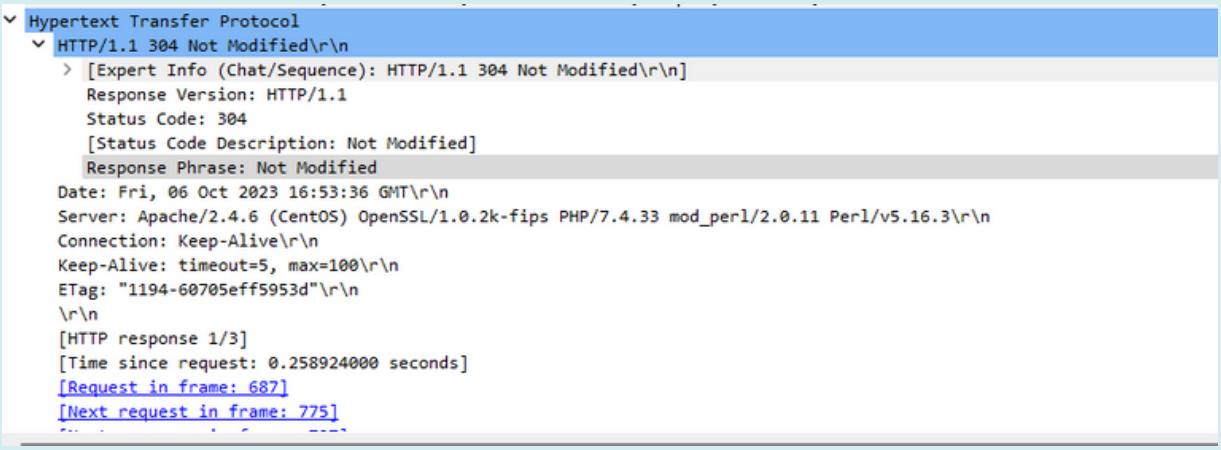
Q5) As you are retrieving long document, how many request packets are sent from the client to the server.

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Packet Lengths	174	223.76	42	3113	0.0143	100%	0.2400	3.085
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	43	64.07	42	75	0.0035	24.71%	0.0800	4.291
80-159	87	102.05	83	157	0.0072	50.00%	0.1300	3.085
160-319	21	229.10	167	319	0.0017	12.07%	0.0300	4.533
320-639	8	467.62	342	639	0.0007	4.60%	0.0300	4.291
640-1279	7	888.57	692	1228	0.0006	4.02%	0.0300	3.085
1280-2559	7	1345.14	1292	1664	0.0006	4.02%	0.0200	3.809
2560-5119	1	3113.00	3113	3113	0.0001	0.57%	0.0100	4.110
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

Q6) Write down your understanding on how the HTTP long file is supported by underlying TCP.

1. **TCP Reliability:** HTTP relies on TCP to ensure the reliable delivery of data during long file transfers. TCP's ability to retransmit lost or corrupted packets guarantees data integrity.
2. **Efficient Data Handling:** TCP segments large files into manageable packets and reassembles them at the destination. It also manages flow control to prevent congestion, allowing for efficient long file transfers.
3. **Summary:** TCP, the underlying protocol for HTTP, establishes reliable connections, maintains order, and manages congestion, making it suitable for long file transfers. HTTP defines how these TCP mechanisms are used to efficiently transfer large files between clients and servers while ensuring data integrity and network stability.

Q7) Inspect the packet which contains the status code and phrase of the response message.



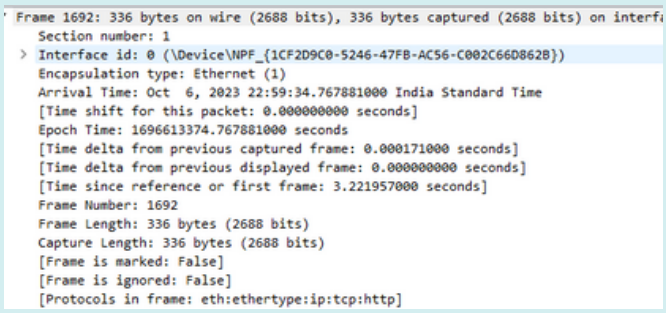
- In packet number 37037, I came across an interesting network communication. This packet is associated with the "application/pkix-crl" content type, which typically holds Certificate Revocation List (CRL) or related certificate information. The protocol in use here is Transmission Control Protocol (TCP), with the source port being 80 and the destination port being 64451. These port numbers help identify the source and destination applications or services.
- <u>As I delved deeper into the packet, I noticed some key details:</u>
  - The sequence number (Seq) is 1, indicating the initial sequence number for this TCP segment. It's essential for packet ordering and data reassembly.
  - The acknowledgment number (Ack) is 228, showing that the sender acknowledges the receipt of data up to sequence number 228. This ensures data transmission confirmation.
  - The length (Len) of this packet is 263 bytes, signaling the size of the data it holds.
- In terms of the HTTP response message, this packet typically includes the following components:
- The HTTP Response Status Line, which is the first line of the HTTP response message, contains the status code and phrase. For instance:
  - The "200" status code signifies a successful response.
  - The "OK" status phrase provides a concise description of the status.

```
HTTP/1.1 200 OK
```

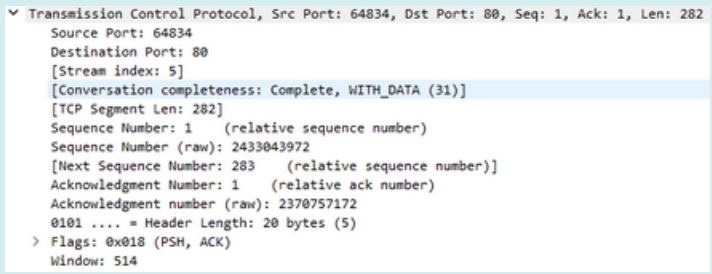
- Additional Response Headers, following the status line, may include headers like "Content-Type," "Content-Length," and more, depending on the specific response. These headers offer supplementary information about the response.
- The Response Body, located after the headers, contains the actual data or content sent by the server. In this particular case, the "Len: 263" suggests that the response body is 263 bytes long. The content within the response body varies based on the application's nature and the response's purpose.

Q8) Write down your interesting observations for the request and response messages while performing this task.

1692	3.221957	192.168.1.5	209.197.3.8	HTTP	336 GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?355e0d16ff952201 HTTP/1.1
1730	3.288932	209.197.3.8	192.168.1.5	HTTP	245 HTTP/1.1 304 Not Modified
23420	49.599448	192.168.1.5	128.119.245.12	HTTP	553 GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
23589	49.857880	128.119.245.12	192.168.1.5	HTTP	771 HTTP/1.1 401 Unauthorized (text/html)
26984	58.633628	192.168.1.5	128.119.245.12	HTTP	638 GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
27095	58.895323	128.119.245.12	192.168.1.5	HTTP	771 HTTP/1.1 401 Unauthorized (text/html)
29793	66.530420	192.168.1.5	128.119.245.12	HTTP	638 GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1
29945	66.836017	128.119.245.12	192.168.1.5	HTTP	583 HTTP/1.1 404 Not Found (text/html)
29966	66.895733	192.168.1.5	128.119.245.12	HTTP	499 GET /favicon.ico HTTP/1.1



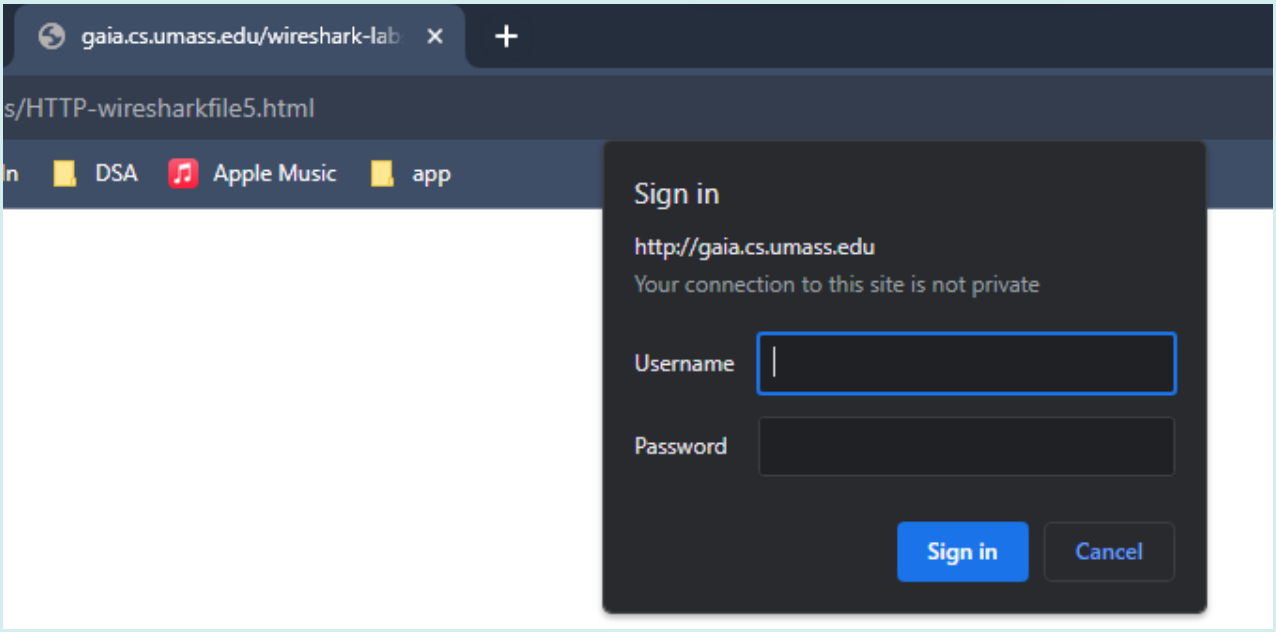
```
Frame 1692: 336 bytes on wire (2688 bits), 336 bytes captured (2688 bits) on interfa
    Section number: 1
  > Interface id: 0 (\Device\NPF_{1CF2D9C0-5246-47FB-AC56-C002C66D862B})
    Encapsulation type: Ethernet (1)
    Arrival Time: Oct  6, 2023 22:59:34.767881000 India Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1696613374.767881000 seconds
    [Time delta from previous captured frame: 0.000171000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 3.221957000 seconds]
    Frame Number: 1692
    Frame Length: 336 bytes (2688 bits)
    Capture Length: 336 bytes (2688 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
```



```
Transmission Control Protocol, Src Port: 64834, Dst Port: 80, Seq: 1, Ack: 1, Len: 282
    Source Port: 64834
    Destination Port: 80
    [Stream index: 5]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 282]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2433043972
    [Next Sequence Number: 283    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2370757172
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 514
```

- The response message "Not Found" and the accompanying message "The requested URL /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html was not found on this server" provide valuable insights and observations regarding the issue with the requested resource:
- HTTP Status Code: The "404 Not Found" status code is a standard HTTP response indicating that the server was unable to locate the requested resource. This is a common status code used to signify that the URL or resource specified in the request does not exist on the server. It's a clear indicator of a resource retrieval problem.
- Error Message Details: The additional message "The requested URL..." offers a more detailed explanation of the problem. It precisely identifies the URL that couldn't be found on the server. This detailed message assists both the client and any individuals analyzing the response in pinpointing the source of the issue.

- **Resource Path: The provided URL, "/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html," reveals the path to the resource that the client was attempting to access. This URL structure suggests that the client was trying to retrieve a file named "HTTP-wiresharkfile5.html" from a directory called "protected_pages" under the "wireshark-labs" directory. The fact that the resource was not found implies either an incorrect URL or the absence of the specified file in the indicated location on the server.**
- **User Authentication: There's a possibility that the error occurred after some form of user authentication or authorization check. If the user did not possess the necessary permissions or credentials to access the resource, it could result in a "Not Found" error. In such cases, resolving the issue might involve verifying user permissions and authentication credentials.**
- **Overall, this response message serves as a valuable diagnostic tool for troubleshooting. It strongly suggests that there is an issue with the URL or the availability of the requested resource on the server. To address this, further investigation should include verifying the correctness of the URL, confirming the existence and location of the requested file, and ensuring that any required authentication and authorization mechanisms are properly configured to grant access to the resource.**



Conclusion:-

- In my experience with Wireshark, the capability to locate and dissect packets has been incredibly valuable for comprehending how network communication works. The ability to spot HTTP response status codes has been particularly helpful in determining whether requests were successful or encountered issues, which proved essential for troubleshooting.
- Additionally, scrutinizing response messages has been illuminating when it comes to understanding the server's behavior. These messages have provided vital insights for diagnosing problems, optimizing network performance, and ensuring the efficient transfer of data.
- As a network professional, I've found Wireshark to be an indispensable tool. It offers a comprehensive view of network traffic, empowering me to make informed decisions and take the necessary actions to maintain network reliability and efficiency.

End of Report