Wireshark Sniffing & Packet Analysis

Unveiling Network Traffic Patterns

A comprehensive guide for networking students to master live traffic capture and protocol identification.

Agenda

Setting Up Wireshark

Configuring for optimal packet capture.

Live Traffic Capture

Initiating and managing network sniffing sessions.

HTTP vs. HTTPS

Identifying and differentiating web traffic.

Packet Breakdown & Analysis

Deep diving into captured data with display filters.

Why Packet Analysis Matters

Troubleshooting

Diagnose network issues, identify bottlenecks, and resolve connectivity problems efficiently.

Security Auditing

Detect suspicious activities, unauthorized access attempts, and potential vulnerabilities in real-time.

Performance Optimization

Analyze traffic patterns to optimize bandwidth usage and application response times.

Preparing for Capture

Before you begin, ensure Wireshark is correctly installed and your network interface is ready. Select the appropriate interface (e.g., Ethernet or Wi-Fi) that is actively transmitting traffic.

Pro Tip: Close unnecessary applications to minimize irrelevant traffic and simplify your analysis.

Install Wireshark

Download the latest version from wireshark.org and complete the installation.

Identify Network Interface

Open Wireshark and choose the active interface (e.g., your wired or wireless connection).

Check Promiscuous Mode

Ensure your adapter is in promiscuous mode to capture all traffic, not just your own.

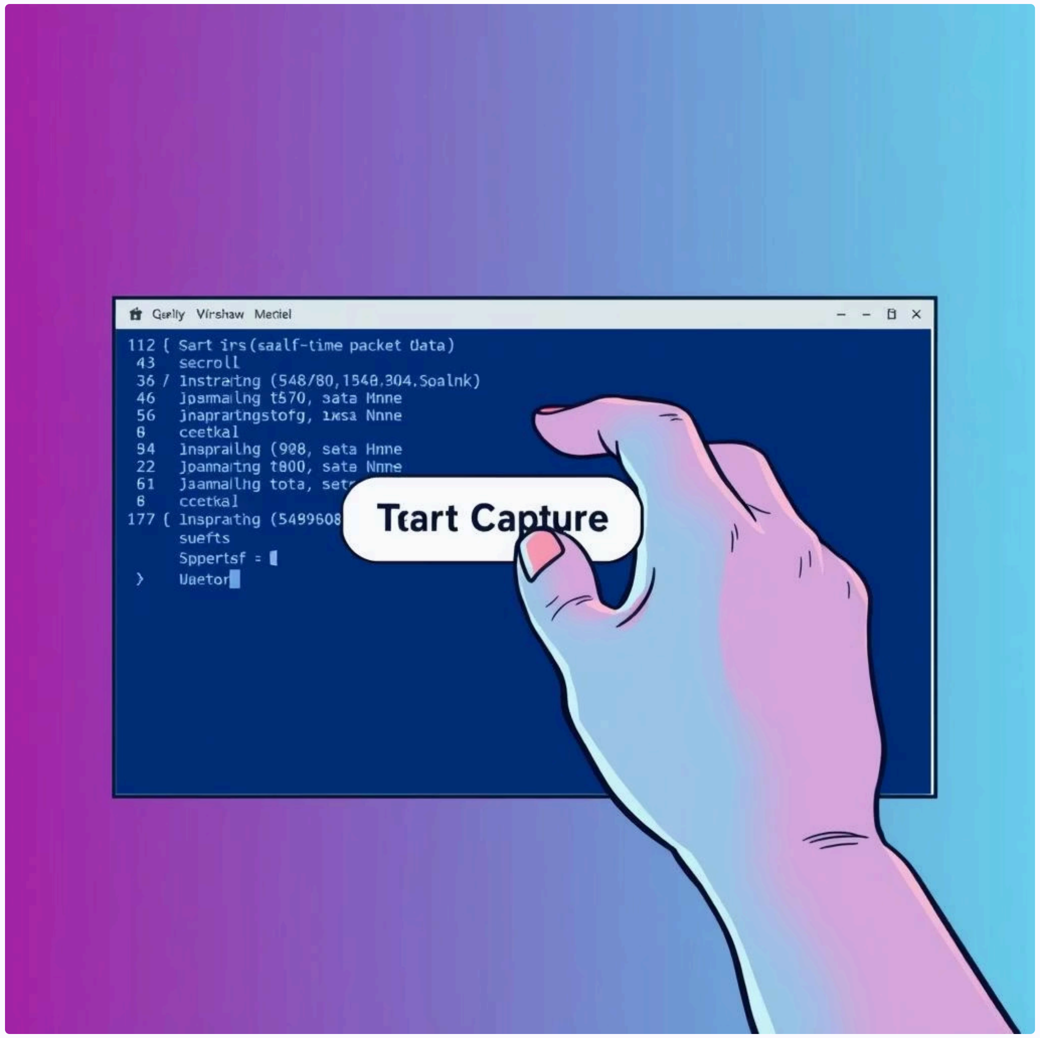
Assignment Task

Capture Live Network Traffic

Start a capture session in Wireshark. During the capture, perform common web activities:

- Browse a regular HTTP website (e.g., a very old, unencrypted site).
- Browse a secure HTTPS website (e.g., google.com, banking site).

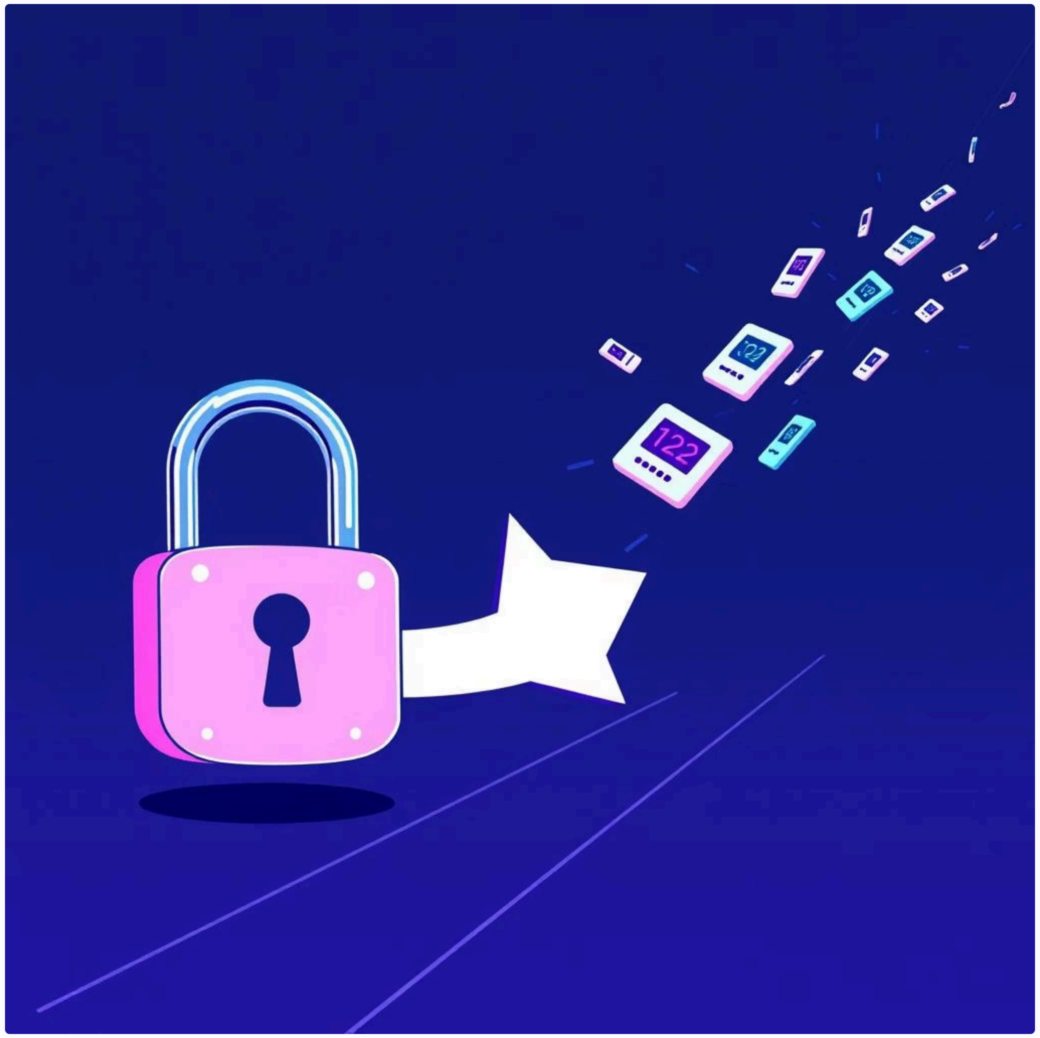
Aim for about 30-60 seconds of diverse traffic to ensure you capture both types of transmissions.


Gerlly Virshaw Meciel
112 { Sart irs(saalf-time packet Uata)
43 secroll
36 / Instraitng (548/80,1540.304.Soalnk)
46 Jpsmmailng t570, sata Mnne
56 jnaprartngstofg, 1xsa Nnne
8 ceetkal
94 Inspraillhg (908, sata Hnne
22 Jpannaitng t800, sata Nnne
61 Jaamnallhg tota, set
8 ccetkal
177 { Inspraithg (5499608
suefts
Sppertsf =
› Uaetor
Tcart Capture

☐ Remember to save your capture file (.pcap) after completing the capture. This is your primary submission.

HTTP vs. HTTPS: The Key Difference

HTTP (Hypertext Transfer Protocol)



- **Unencrypted:** Data sent in plain text, visible to anyone intercepting the traffic.
- **Port 80:** Default port for web communication.
- **Vulnerable:** Susceptible to eavesdropping and man-in-the-middle attacks.

HTTPS (Hypertext Transfer Protocol Secure)

- **Encrypted:** Data is encrypted using SSL/TLS, ensuring privacy and integrity.
- **Port 443:** Default secure port for web communication.
- **Secure:** Protects sensitive information like login credentials and financial data.

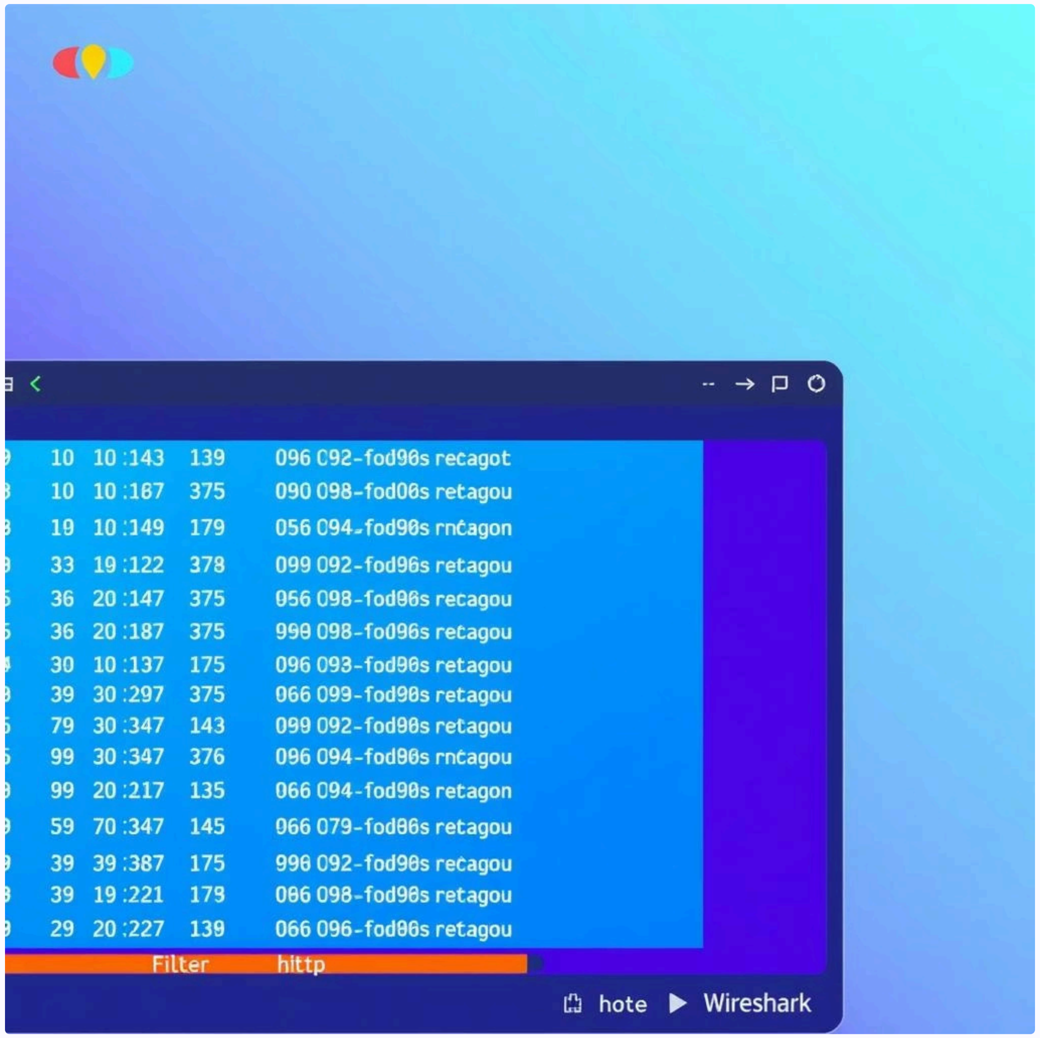
Identifying HTTP & HTTPS Traffic

Once you've captured your traffic, use Wireshark's display filters to isolate and analyze HTTP and HTTPS packets.

HTTP Filter

```
http
```

This filter shows all packets identified as HTTP traffic. You'll observe GET/POST requests and responses, often revealing the URLs and data in plain text.



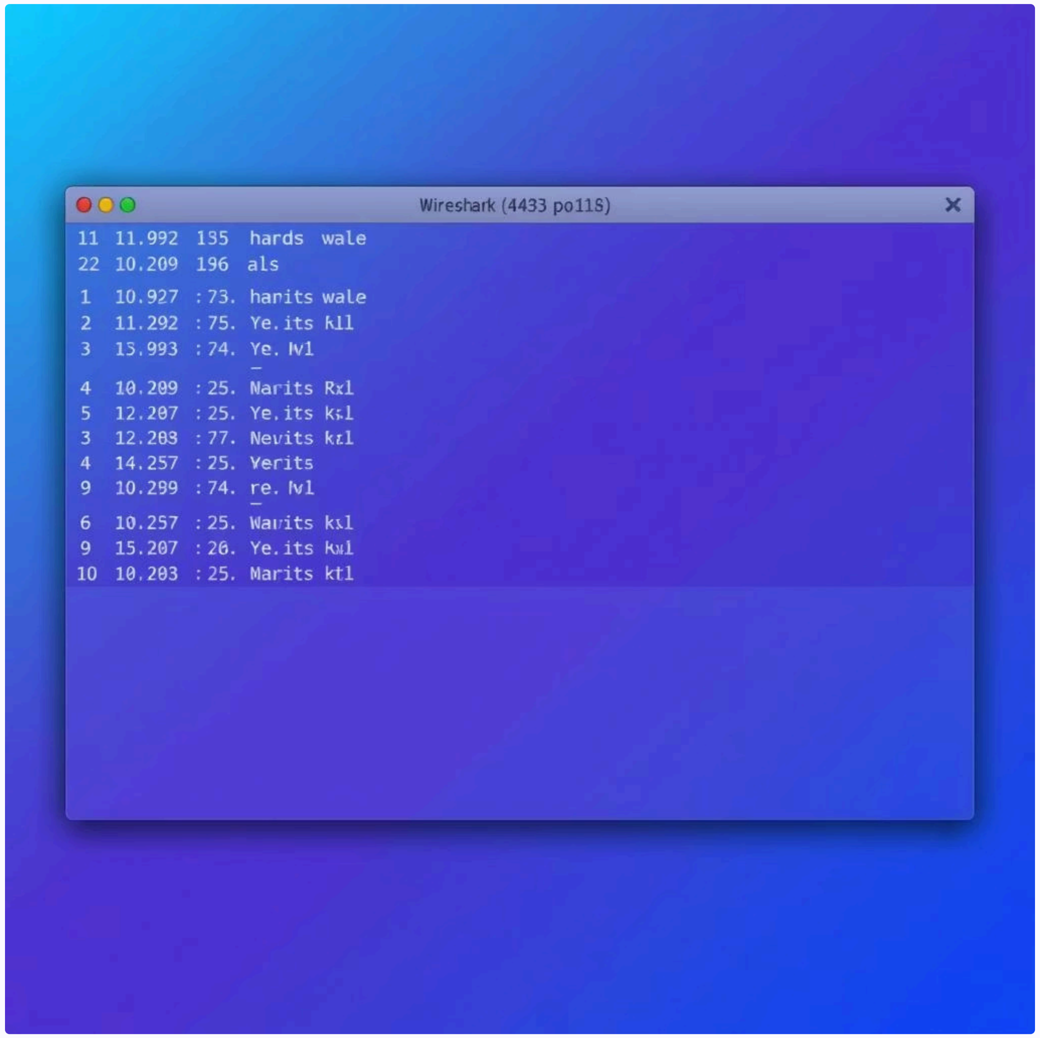
```
10  10:143  139   096 C92-fod90s recagot
10  10:167  375   090 098-fod06s retagou
19  10:149  179   056 094-fod90s rnćagon
33  19:122  378   099 092-fod96s retagou
36  20:147  375   056 098-fod06s recagou
36  20:187  375   990 098-fod96s retagou
30  10:137  175   096 093-fod90s retagou
39  30:297  375   066 099-fod90s retagou
79  30:347  143   099 092-fod90s retagou
99  30:347  376   096 094-fod90s rnćagou
99  20:217  135   066 094-fod90s retagon
59  70:347  145   066 079-fod06s retagou
39  39:387  175   996 092-fod90s recagou
39  19:221  179   086 098-fod90s retagou
29  20:227  139   066 096-fod06s retagou
```

Filter   http

HTTPS Filter (TCP Port 443)

```
tcp.port == 443
```

Since HTTPS is encrypted, you won't see clear HTTP requests. Instead, you'll observe encrypted TCP traffic over port 443, often with "Client Hello" and "Server Hello" messages for TLS/SSL handshake.



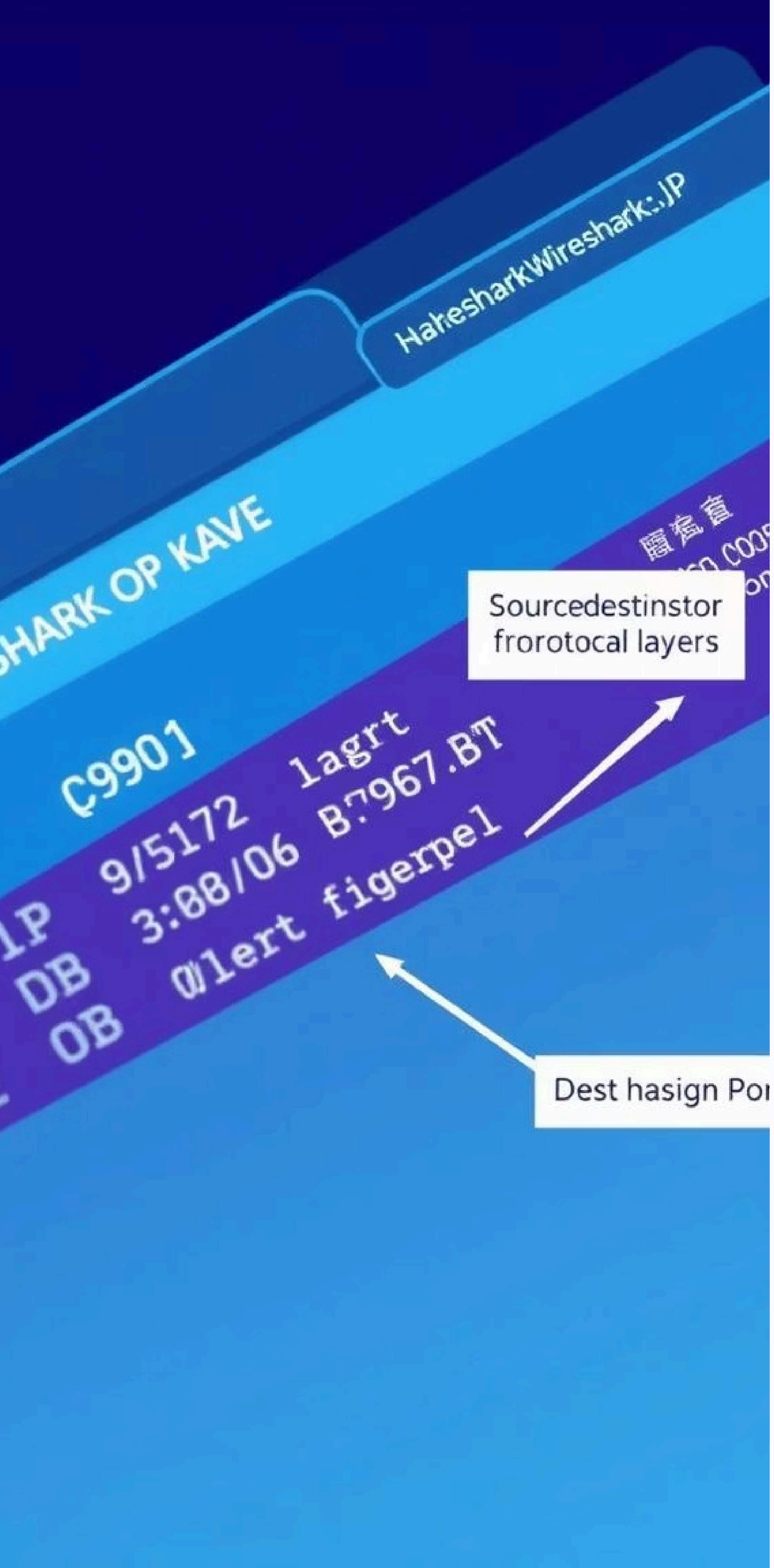
Wireshark (4433 po118)

```
11  11.992  135  hards  wale
22  10.209  196  als

1   10.927  :73.  harits wale
2   11.292  :75.  Ye.its kll
3   13.993  :74.  Ye. Nl

4   10.209  :25.  Narits Rxl
5   12.207  :25.  Ye.its ksl
3   12.203  :77.  Nevits kzl
4   14.257  :25.  Yerits
9   10.299  :74.  re. Nl

6   10.257  :25.  Warits ksl
9   15.207  :26.  Ye.its kwl
10  10.203  :25.  Marits ktl
```



Packet Breakdown & Analysis

For your submission, select a few representative packets (at least one HTTP and one HTTPS) and break them down.

1

Identify Protocol

Confirm if it's HTTP (Application Layer) or TCP (Transport Layer, for HTTPS).

2

Source & Destination

Note the IP addresses and port numbers involved in the communication.

3

Packet Details

For HTTP, identify the URI, host, and user-agent. For HTTPS, observe TLS/SSL handshake messages.

4

Interpretation

Explain what the packet reveals about the web communication (e.g., a request for a webpage, an encrypted data exchange).

Submission Requirements

✓ Your complete submission should include two main components:

- **PCAP File:** The saved Wireshark capture file (.pcap) from your live traffic capture session.
- **Packet Breakdown Document:** A text document or PDF detailing your analysis.

In your packet breakdown, ensure you explicitly state the display filters used (`http` and `tcp.port == 443`) and provide your detailed analysis for at least one HTTP and one HTTPS packet.