Networking Concepts

You can refer here to gain a basic understanding of networking concepts.

Network packet analyzer tools:

Wireshark- Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.

You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).

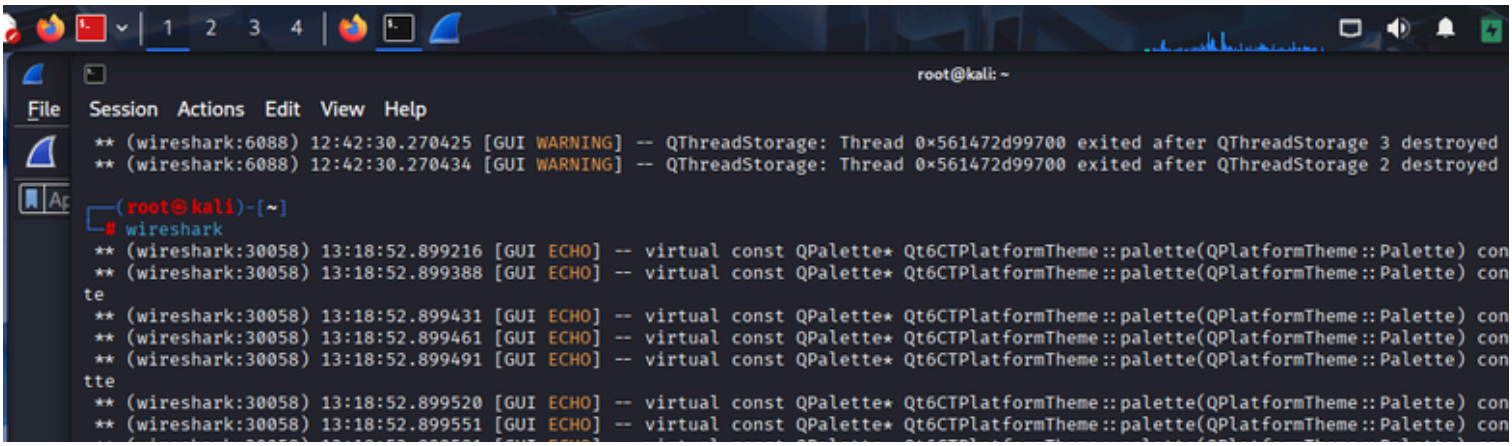
tcpdump- A powerful, command-line packet analyzer used primarily on Unix-like operating systems (Linux, macOS, BSD) to capture and inspect network traffic in real-time or save it to a file (.pcap or .pcapng format) for later analysis. It interacts with the user through a Command Line Interface (CLI).

Capturing Live traffic through Wireshark

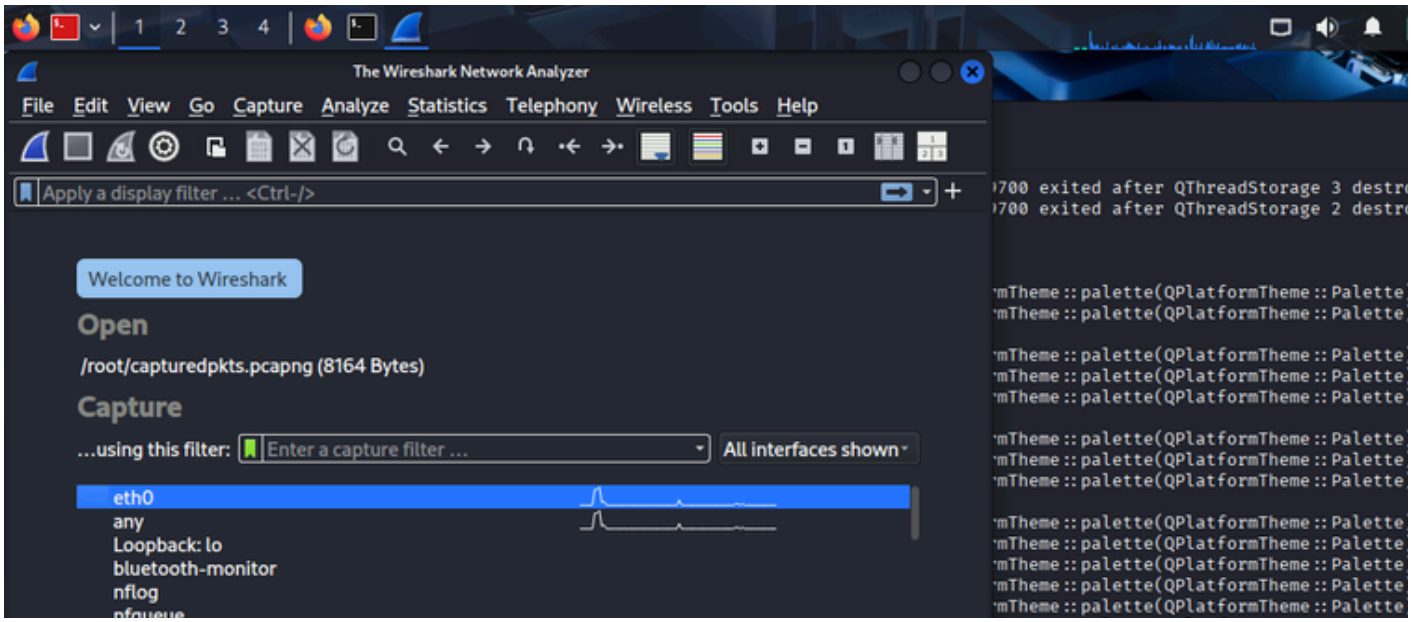
- Open kali Linux VM on your desktop.

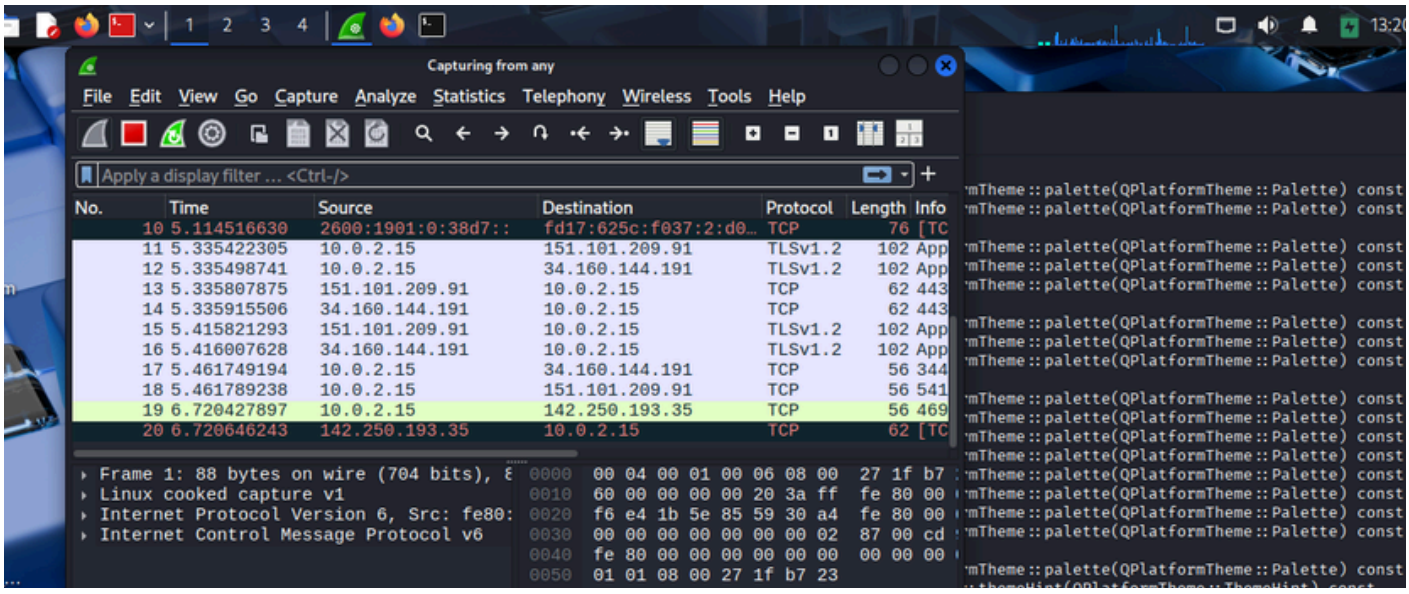
- Open the Root terminal-enter the password to authenticate.

- Type wireshark and press enter to open wireshark GUI.

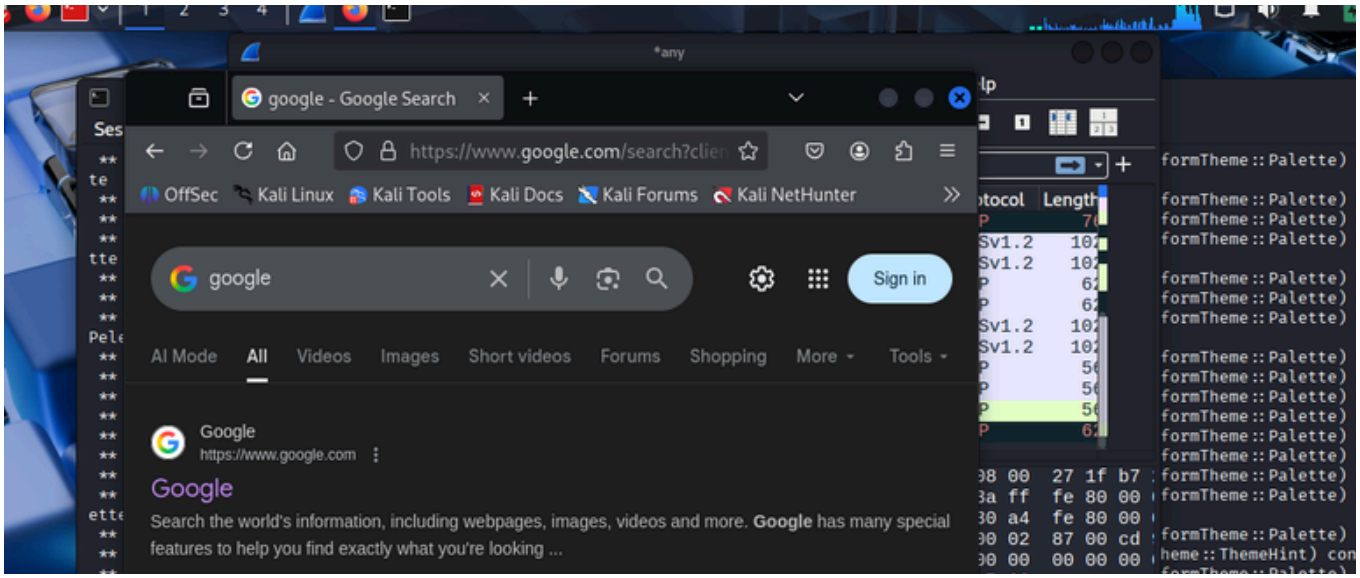


- Choose the required interface for capturing traffic. Double click to start the capture.

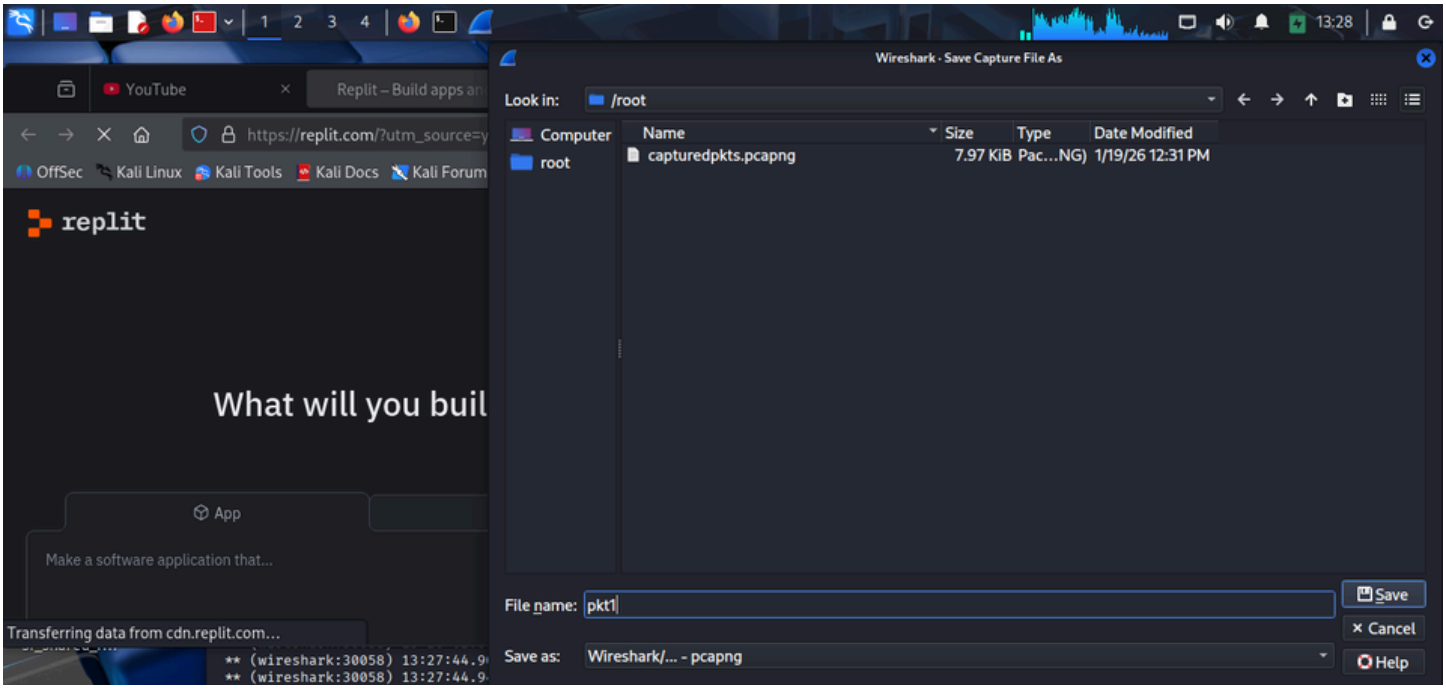




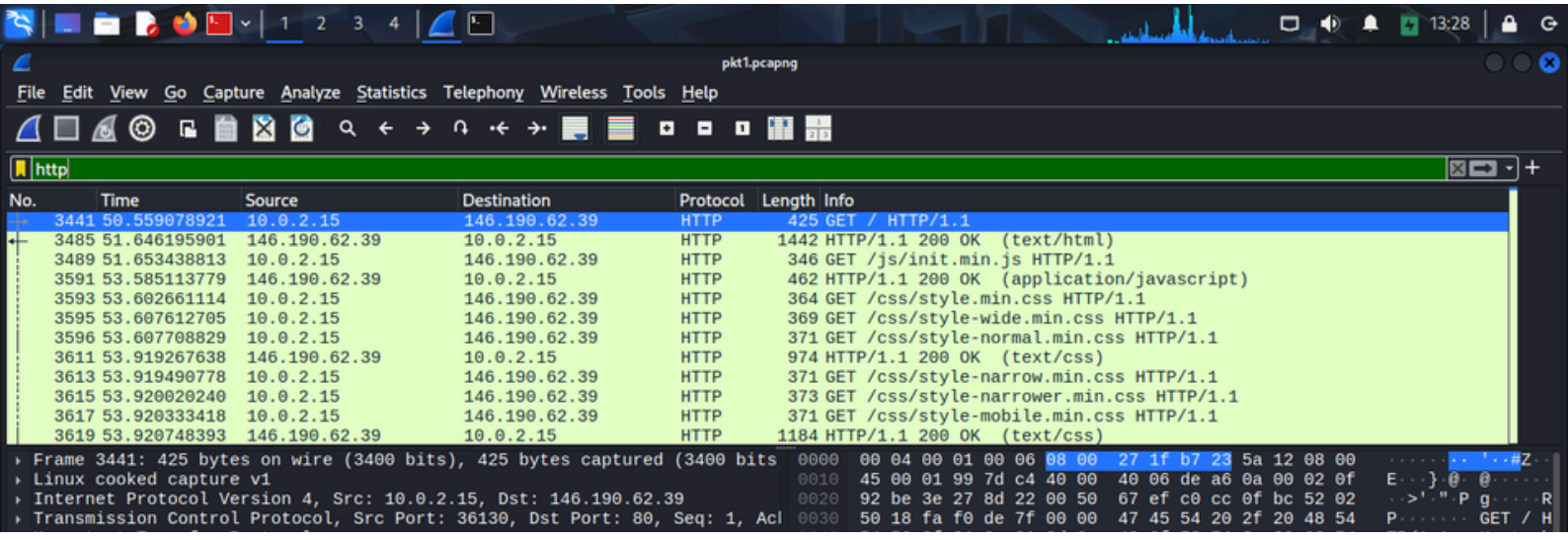
- Open firefox browser in your kali machine and search and go to different websites to capture different traffic which uses http, dns, tcp protocols.

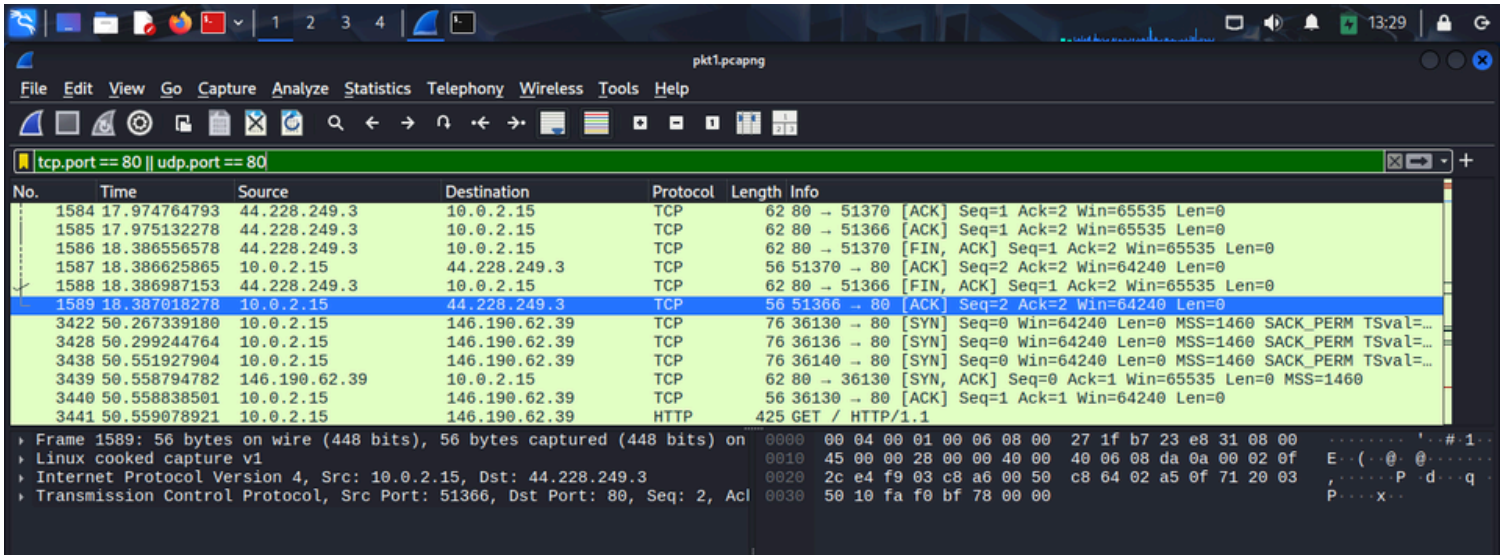


- Save the captured traffic in a .pcap file to save and analyse the network traffic.



- Filter it by applying display filter for filtering different traffic, e.g.- http, tcp, dns, etc.





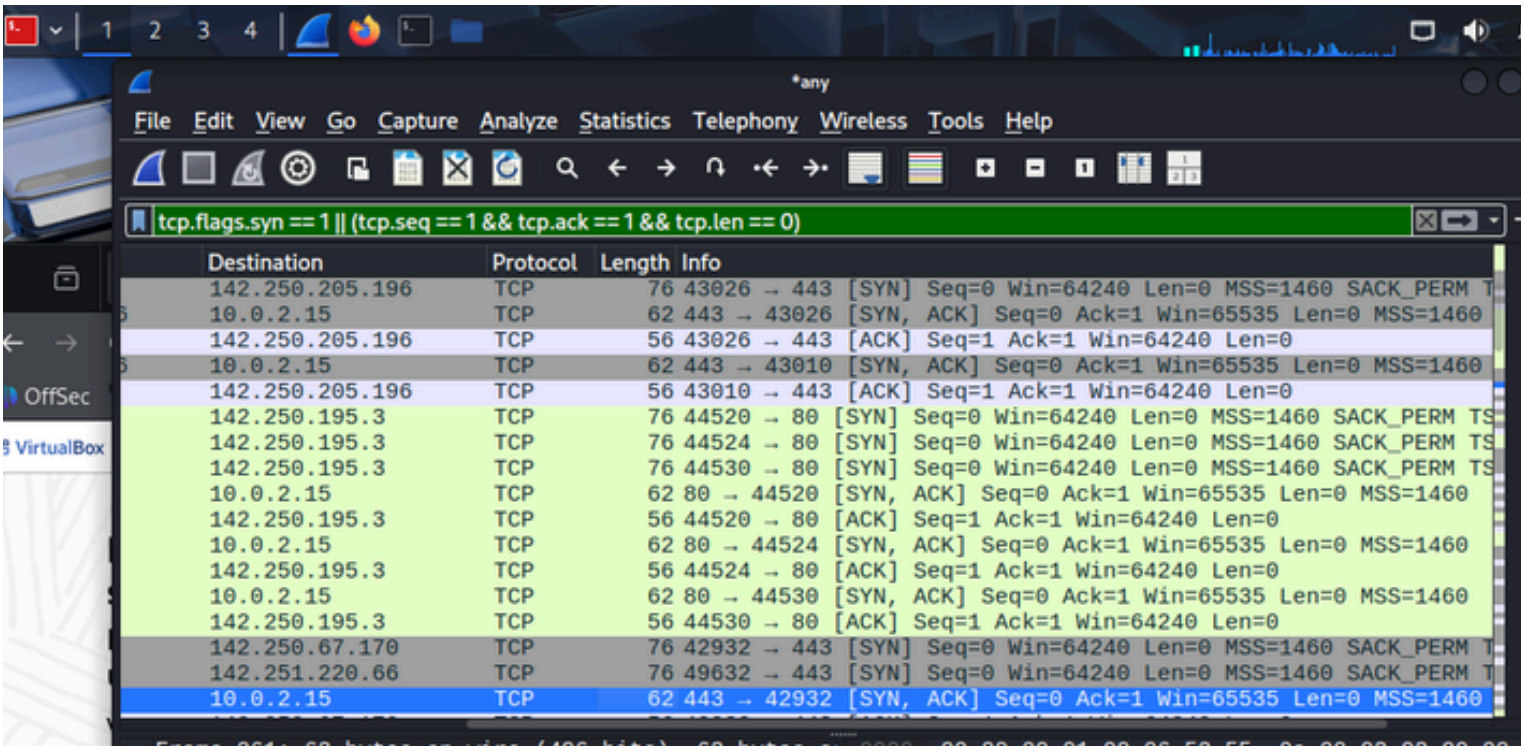
Three-Way TCP Handshake

The **three way handshake**, also known as the **TCP handshake**, is a fundamental process in **networking** that establishes a reliable connection between a client and a server before data transmission begins. This handshake ensures that both devices are synchronized and ready for communication, providing the backbone for **TCP (Transmission Control Protocol)** reliability.

Observing in wireshark

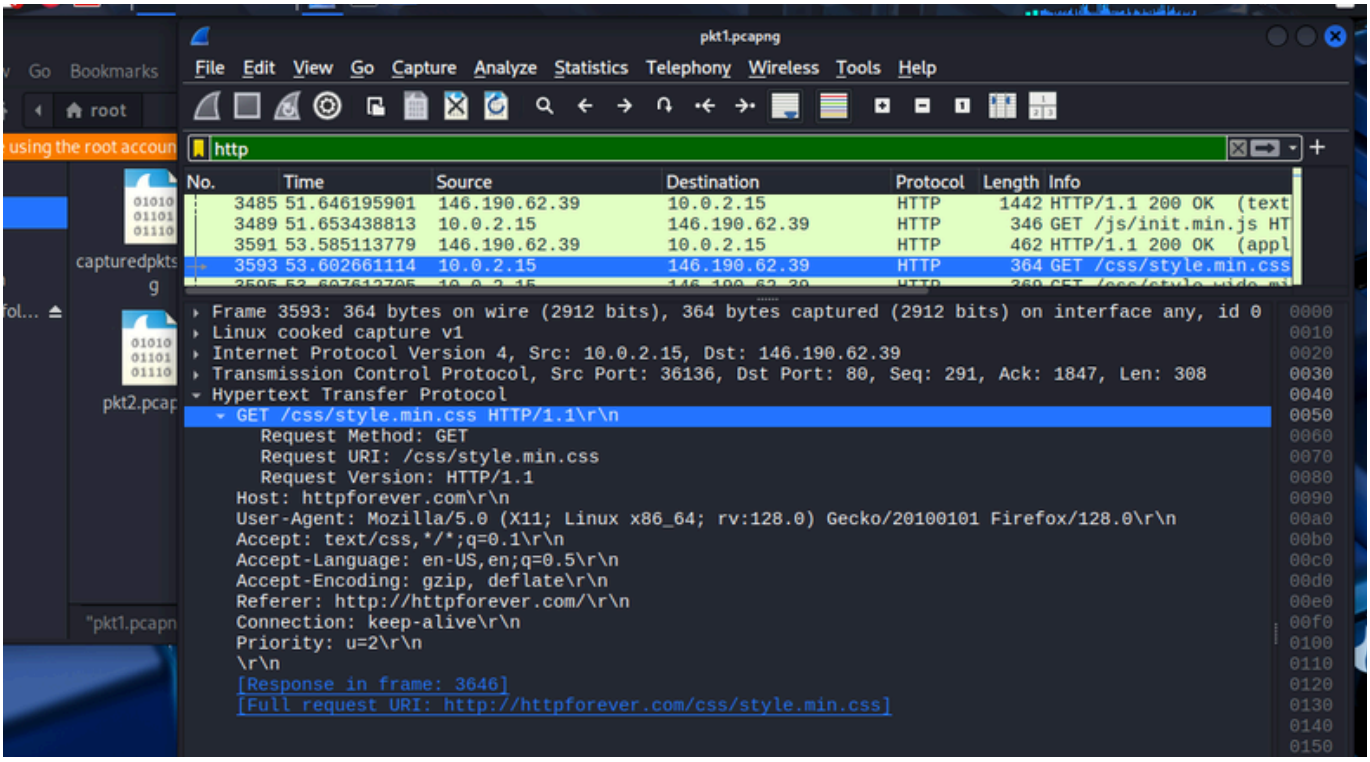
To see the "handshake" that establishes a connection:

1. Filter for tcp.flags.syn == 1.
2. Find a packet, right-click it, and select **Follow -> TCP Stream**.
3. Look for the sequence:
   - **SYN** (Synchronize)
   - **SYN, ACK** (Synchronize-Acknowledge)
   - **ACK** (Acknowledge)

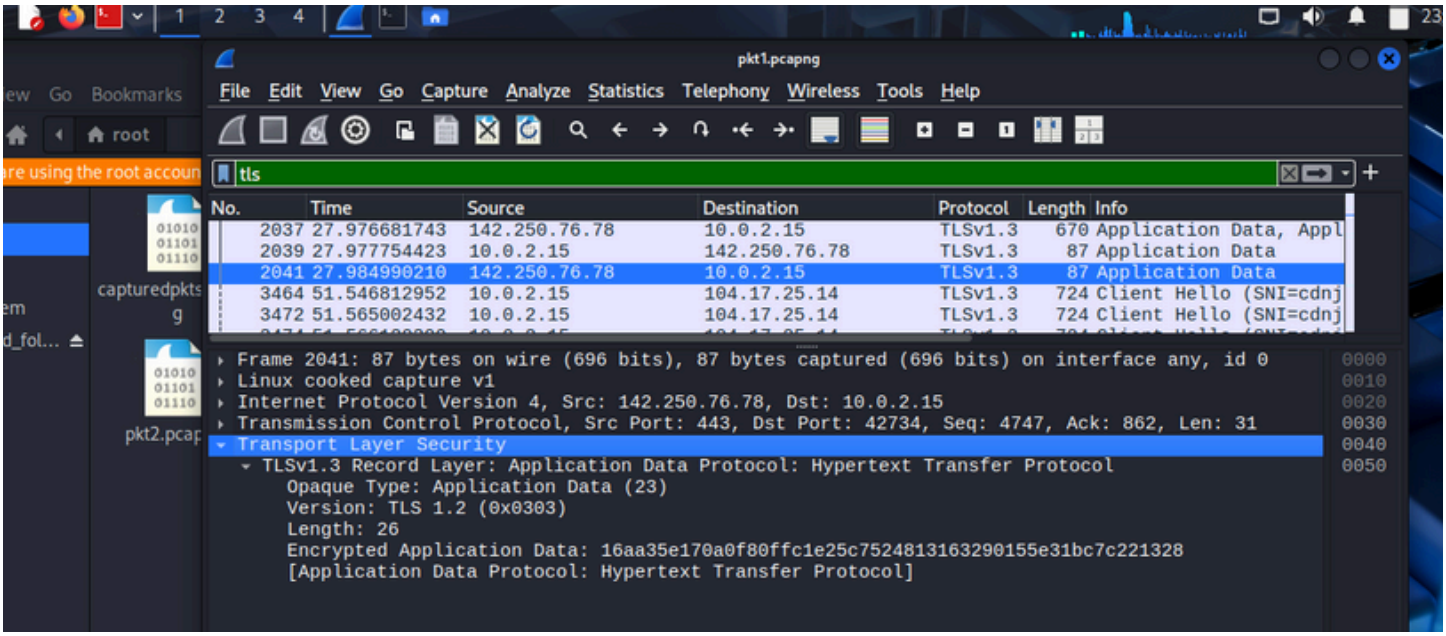


Plain-Text vs. Encrypted Traffic

- **Plain-Text (HTTP):** Filter for http. Find a "GET" or "POST" request. Click into the "Packet Details" pane. You will likely be able to read HTML code, form data, or headers in clear text.

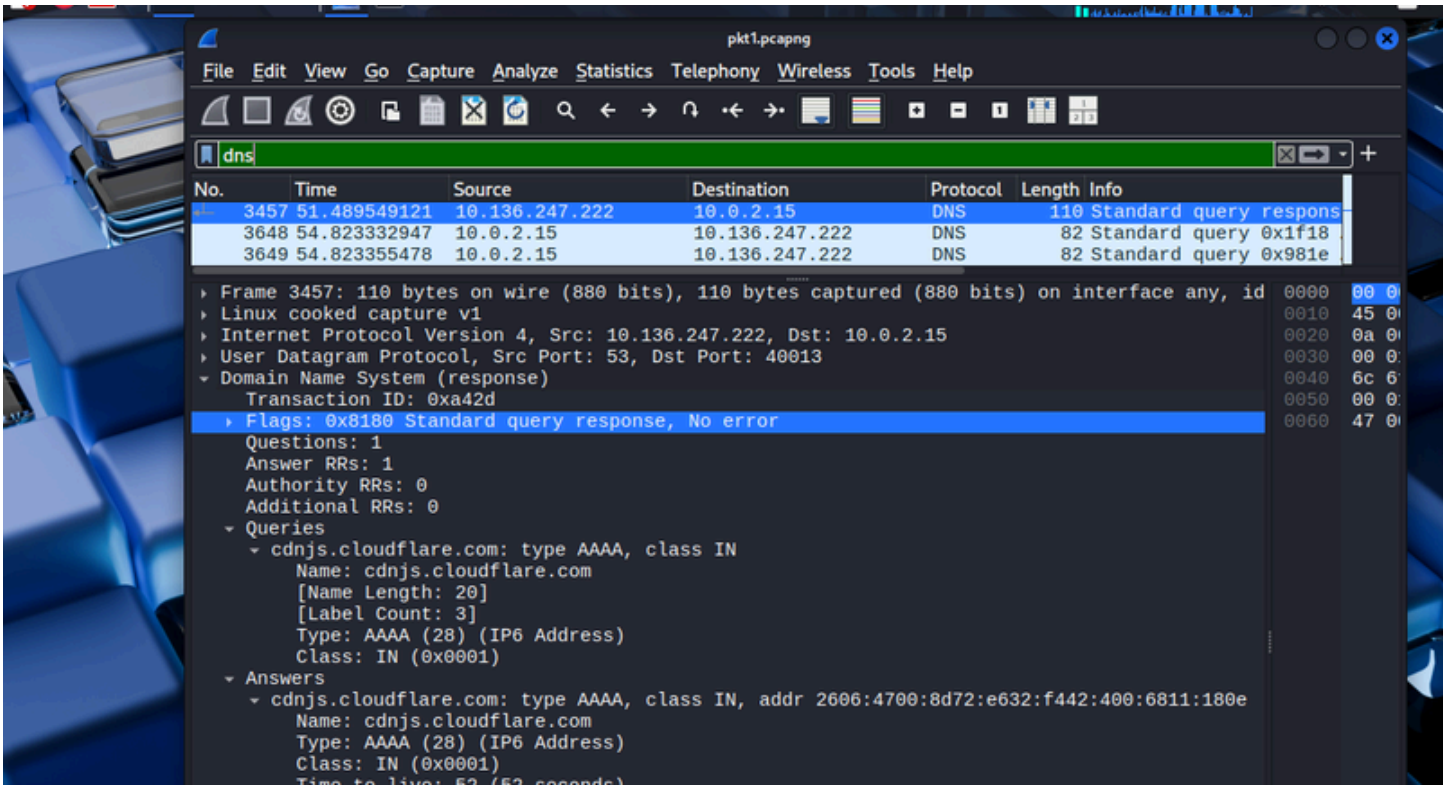


- **Encrypted (HTTPS/TLS):** Filter for tls. You will see "Application Data" packets. If you look at the raw data, it will look like random gibberish (hexadecimal), which is the encrypted payload.



Analyze DNS Queries

1. Filter for dns.
2. Look for a "Standard query" (Your PC asking "Where is https://www.google.com/url?sa=E&source=gmail&q=google.com?") and a "Standard query response" (The server answering with an IP address).
3. Expand the **Queries** section in the packet details to see exactly what domain was requested.



Analysis Report

Project Overview Captured live traffic on interface eth0 and interface any. Analysed DNS, HTTP, TLS, TCP.

Protocol Observations are listed above with screenshots I captured traffic in different files to analyse different type of traffic.

Analysed packet with HTTP protocol -found plain text headers. TLS traffic-contained enceypted headers/data.

Analysed DNS queries- Listed standard query and answers containing domain names in response.

Filtered for TCP- observed TCP 3 way handshake by observing sequence- SYN, SYN-ACK, ACK packages.

Security Findings

- **Plain-text risk:** I observed that traffic to [Site Name] was unencrypted, meaning a hacker could see my data.
- **Encryption:** Most modern web traffic was hidden behind TLS 1.3, ensuring privacy.