A

LAB REPORT

ON

Network Security

By

Rajesh Kumar Yadav

Exam Roll No: 7629020

Registration No.9-2-29-630-2019

Submitted to:

Sujan Poudel

Mahendra Ratna Campus

In partial fulfillment of the requirements for the Course

Network Security

Tahachal, Kathmandu

March, 2024

Table Of Contents

Lab 1: Viewing SSL Certificates with Wireshark

Introduction

Secure Sockets Layer and its successor, Transport Layer Security are cryptographic protocols that ensure secure communication over a network. They establish encrypted connections between a client and a server to protect sensitive data transmission. This lab report details the steps to view SSL certificates using Wireshark, a popular network traffic analyzer.

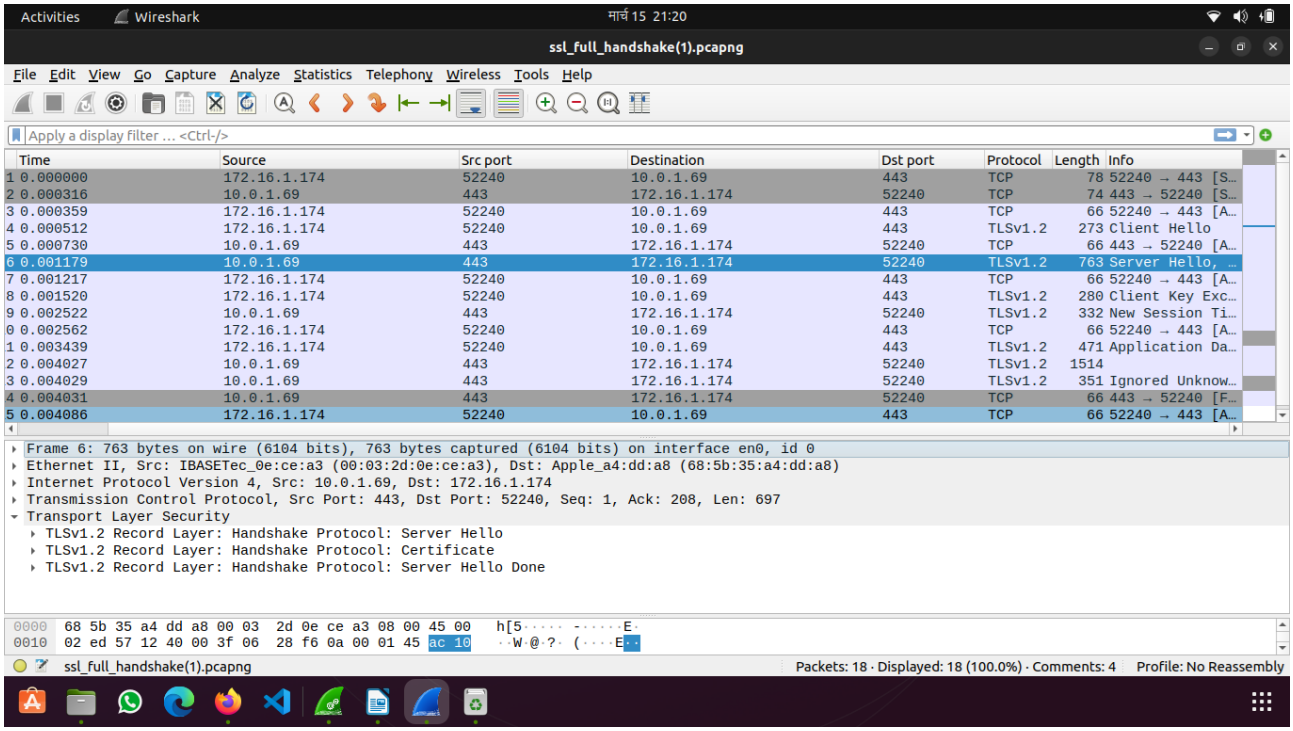
Objectives

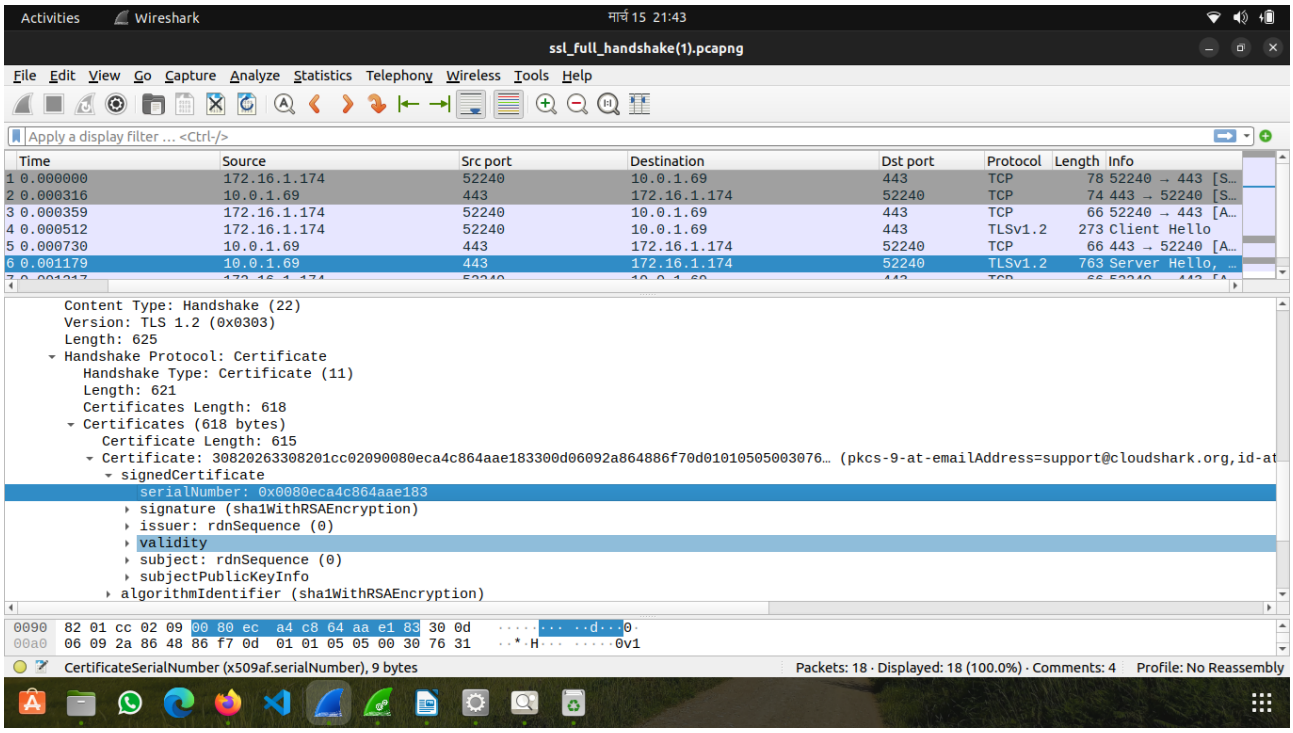
- Understand the role of SSL certificates in secure communication.
- Learn how to capture network traffic with Wireshark.
- Identify the steps to view the details of an SSL certificate within a captured session.

Methodology

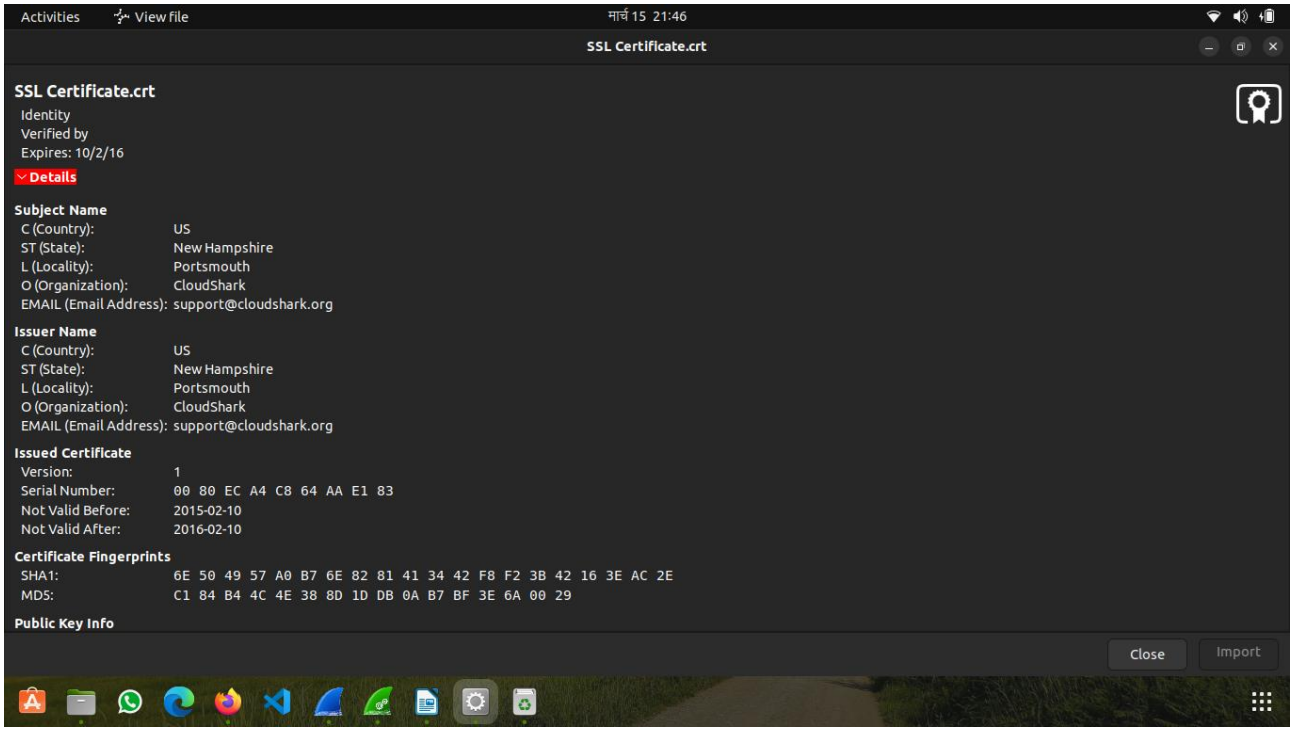
This experiment involved using Wireshark to capture network traffic while visiting a website secured with HTTPS. The captured traffic was then analyzed to extract the SSL certificate information.

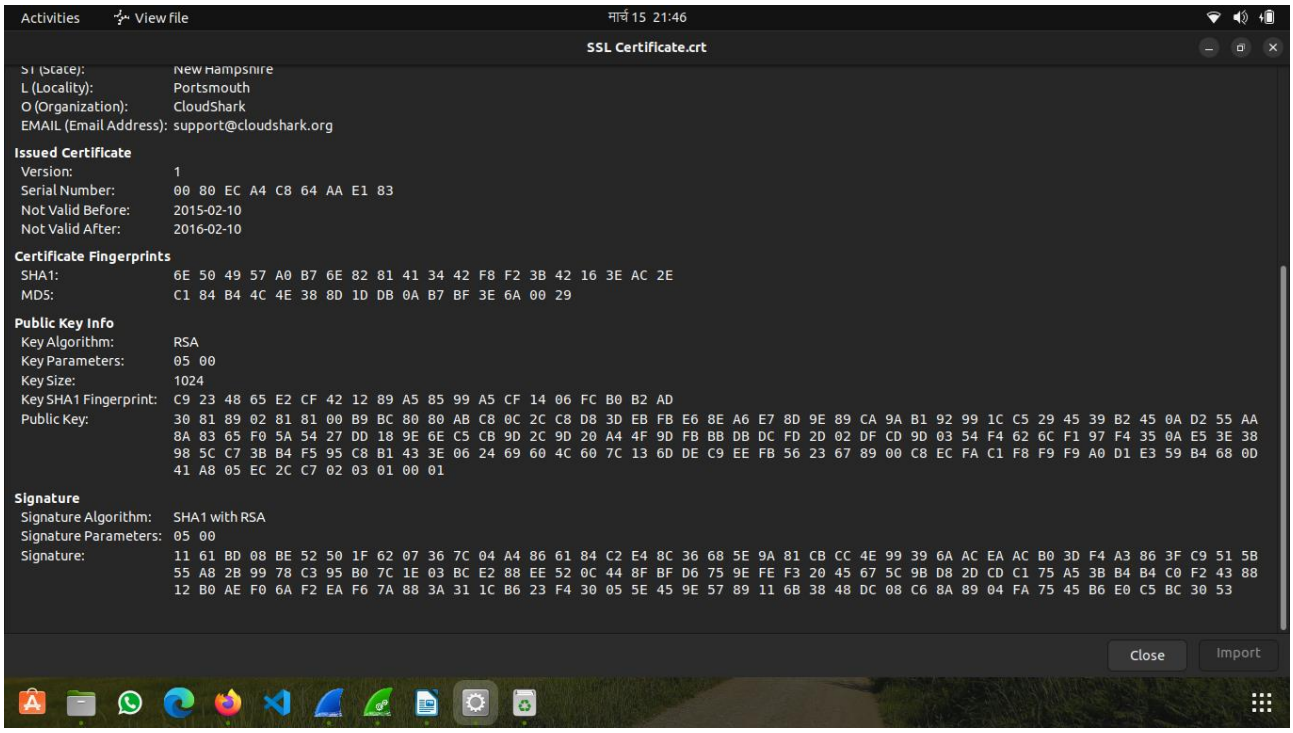
ScreenShot of Process





Certificate





Analysis

The SSL certificate details extracted from the "Server Hello" packet might include:

- **Issuer:** The entity that issued the certificate
- **Subject:** The website domain name for which the certificate is valid.
- **Validity Period:** The start and end dates of the certificate's validity.

⑩ **Public Key:** The server's public key used for encryption in the secure connection.

Lab2: Use encryption/decryption systems

Introduction

In today's digital world, protecting sensitive information is critical. Encryption and decryption systems play a vital role in safeguarding data confidentiality and integrity. This lab report explores the fundamentals of these systems, their functionalities, and practical applications.

Objectives

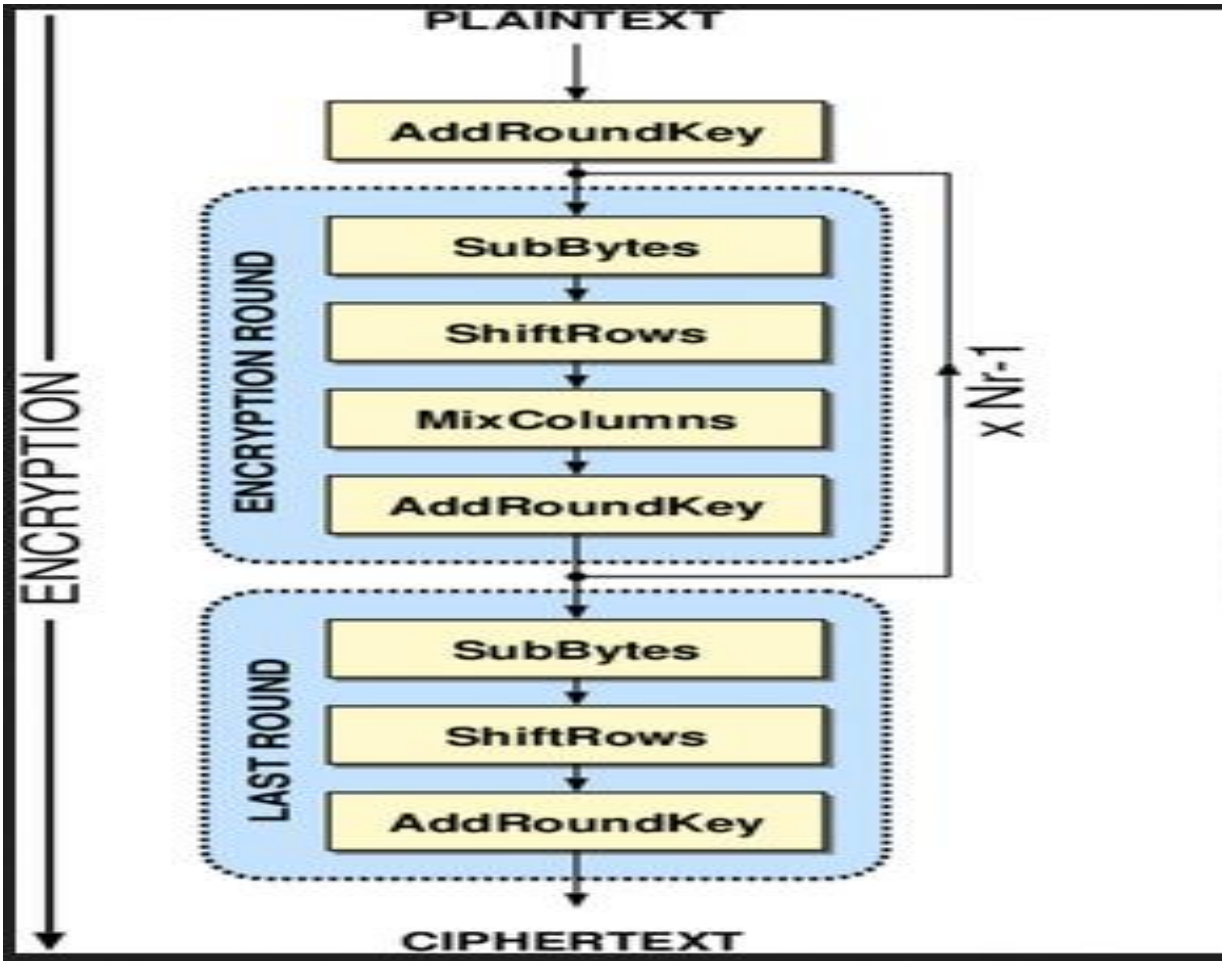
- Understand the core concepts of encryption and decryption.
- Analyze different types of encryption algorithms used for data security.
- Gain hands-on experience with encryption and decryption tools.

Methodology

This experiment involved exploring various encryption and decryption algorithms through software tools and online resources. Sample data was encrypted using different algorithms like AES (Advanced Encryption Standard) and RSA (Rivest–Shamir–Adleman). The encrypted data was then decrypted to verify its functionality.

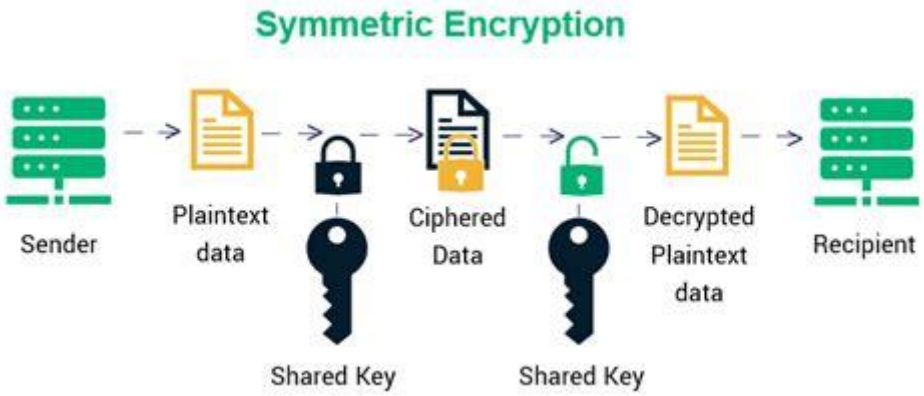
Encryption

In cryptography, encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext. Ideally, only authorized parties can decipher a ciphertext back to plaintext and access the original information.



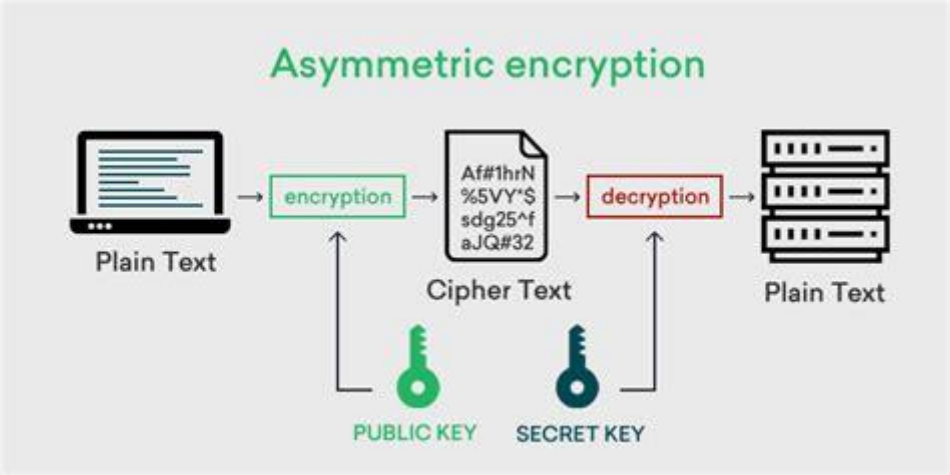
Types of Encryption Algorithms

- **Symmetric Encryption:**
  - Uses a single secret key for both encryption and decryption. Popular algorithms include AES and DES This method is efficient but requires secure key distribution.



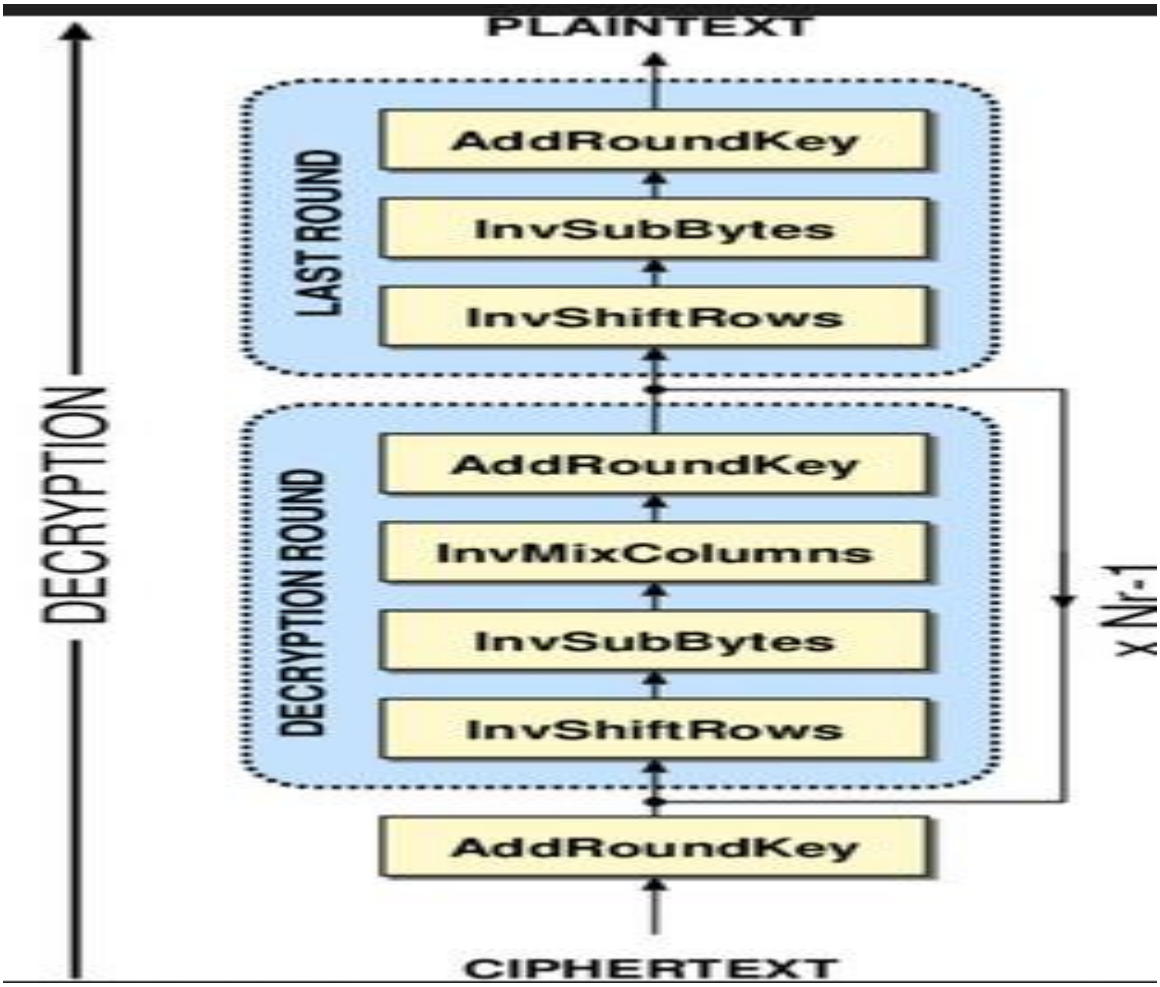
- **Asymmetric Encryption:**

⑩ Utilizes a public-key pair – a public key for encryption and a private key for decryption. This eliminates the need for secure key exchange but is computationally more expensive than symmetric encryption. RSA is a widely used asymmetric algorithm.



Decryption

Decryption is the process of converting meaningless message (Ciphertext) into its original form (Plaintext).



Practical Applications

- **Securing Data at Rest:** Encryption protects sensitive data stored on hard drives, USB drives, and cloud storage.
- **Secure Communication:** Encryption safeguards data transmission during online activities like email communication and web browsing.
- **Financial Transactions:** Encryption secures financial transactions on e-commerce platforms and online banking systems.

Observations

- Different encryption algorithms offer varying levels of security and performance.
- Symmetric encryption is faster but requires secure key management.
- Asymmetric encryption offers enhanced security but is computationally slower.

Conclusion

Encryption and decryption systems are fundamental tools for data security. Understanding these systems and their implementations is crucial for protecting sensitive information in a digital environment.

Lab 3: Digital Certificates and Assigning Digital Signatures

Introduction

In today's digital world, ensuring the authenticity and integrity of electronic documents is crucial. Digital certificates and digital signatures play a vital role in achieving this objective. This lab report explores these concepts and demonstrates how to assign a digital signature to a file.

Objectives

- Understand the concept of digital certificates and their role in digital signatures.
- Learn the process of digitally signing a document.
- Explore software tools used for digital signatures.

Digital Certificates

A digital certificate, also known as an electronic certificate, is a digital document that electronically binds an entity to a public key. It acts as a trusted credential in the digital world, similar to how a physical ID card verifies a person's identity.

Components of a Digital Certificate

- **Subject:** The entity to which the certificate is issued .
- **Issuer:** The trusted Certificate Authority that verifies the subject's identity and issues the certificate.
- **Public Key:** The subject's public key used for encryption in digital signatures.
- **Validity Period:** The time frame during which the certificate is considered valid.
- **Digital Signature:** A digital signature created by the CA using its private key, verifying the authenticity of the certificate itself.

Digital Signatures

A digital signature is a cryptographic technique used to ensure the authenticity and integrity of a digital document. It involves:

1. **Hashing:** The document is passed through a hashing algorithm, generating a unique digest (fingerprint) representing the document's content.
2. **Signing with Private Key:** The signer uses their private key to digitally sign the hash digest.
3. **Verification with Public Key:** The recipient uses the signer's public key to verify the signature. If the verification succeeds, it confirms that the document originated from the signer and has not been tampered with since signing.

Assigning a Digital Signature:

The process of assigning a digital signature to a file can vary depending on the software used. However, the general steps involve:

1. **Opening the Signing Tool:** Launch the digital signature software provided by your chosen Certificate Authority (CA).
2. **Selecting the File:** Browse and select the file you want to sign digitally.
3. **Choosing the Certificate:** Select the digital certificate you want to use for signing (associated with your private key).
4. **Customizing Options (Optional):** Depending on the software, you might be able to set additional options like time stamping or specifying the signing reason.
5. **Signing the Document:** Initiate the signing process. The software will use your private key to sign the document's hash digest.
6. **Saving the Signed File:** The software will save the original file along with the attached digital signature.

Lab 4: Exploring Wireshark -A Network Traffic Analyzer

Introduction:

Wireshark is a powerful and widely-used open-source network traffic analyzer. It allows users to capture network packets flowing across a network interface, enabling detailed analysis of network communication. This lab report explores the functionalities of Wireshark and its key features.

Objectives:

- Understand the purpose and capabilities of Wireshark.
- Learn how to capture and analyze network traffic using Wireshark.
- Identify some of Wireshark's valuable features for network troubleshooting and security analysis.

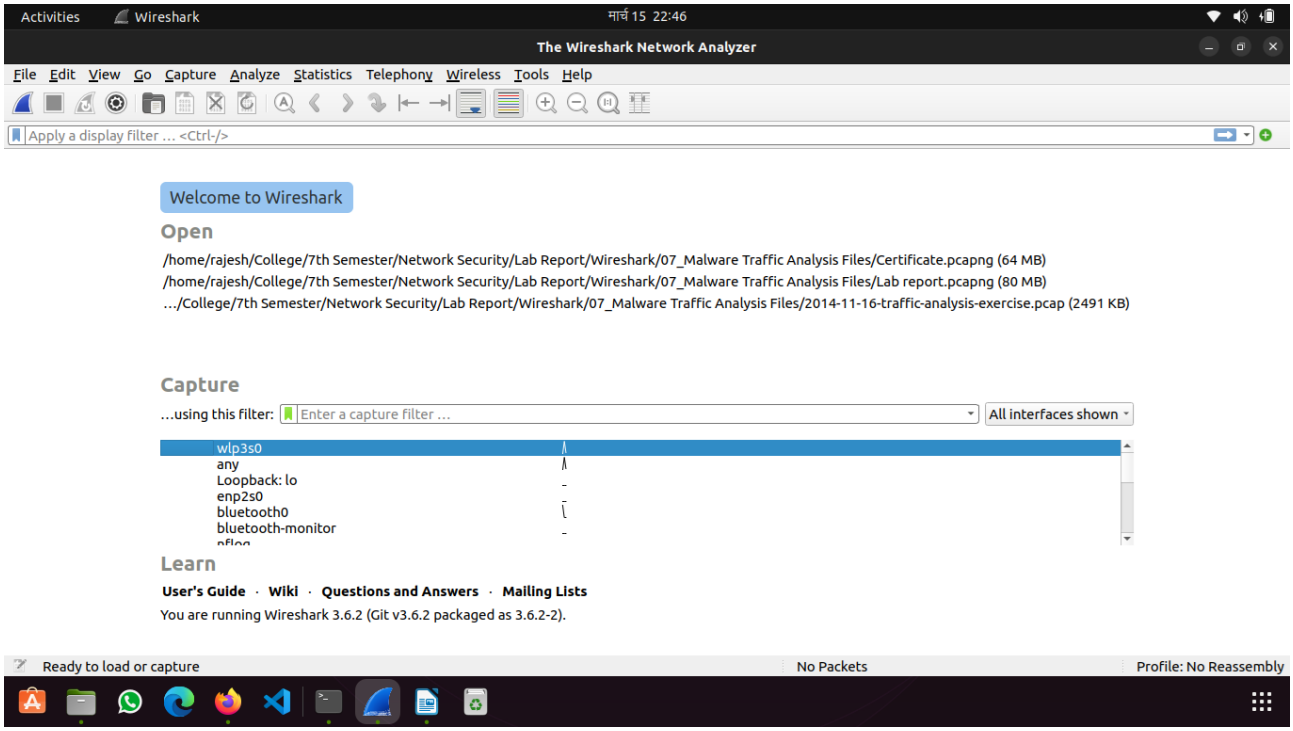
Wireshark Functionality:

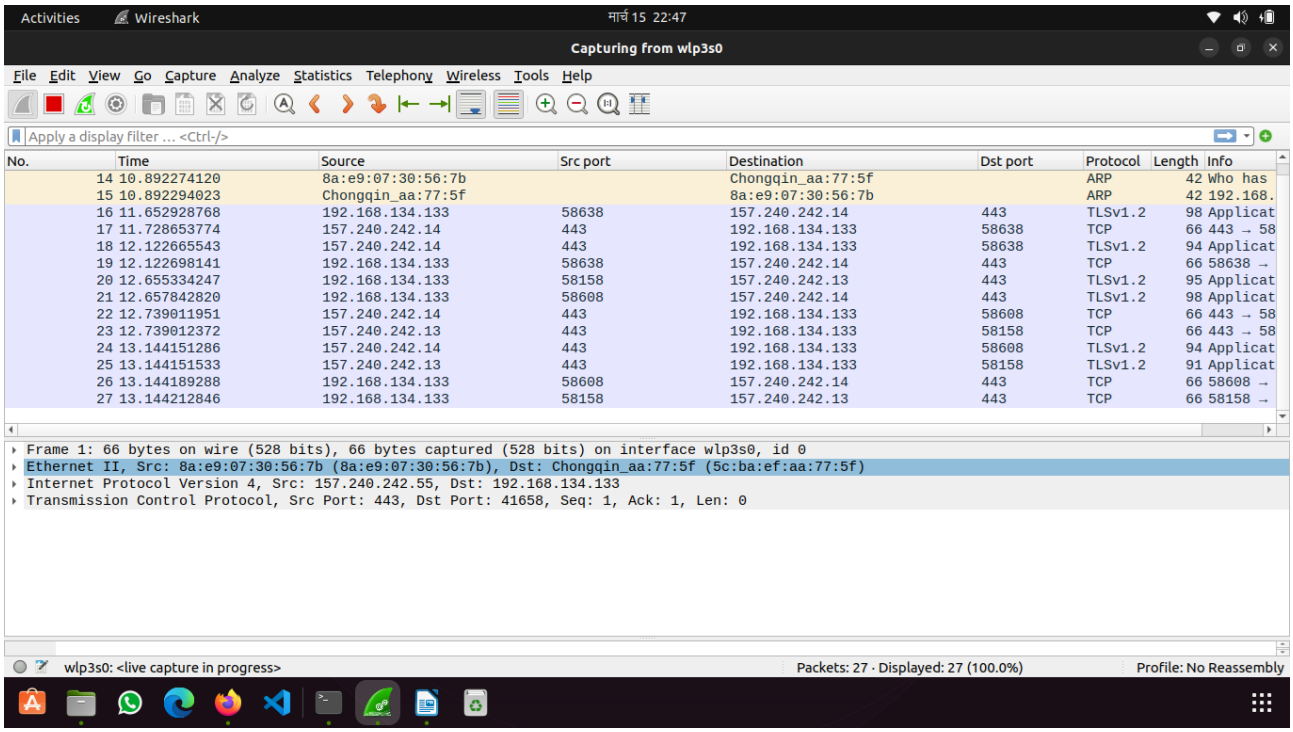
Wireshark operates at the network layer of the OSI model. It captures packets transmitted and received by your computer's network interface card . These packets contain data exchanged between devices on a network, including protocols like TCP, UDP, and IP. Wireshark dissects these packets, presenting them in a user-friendly interface for analysis.

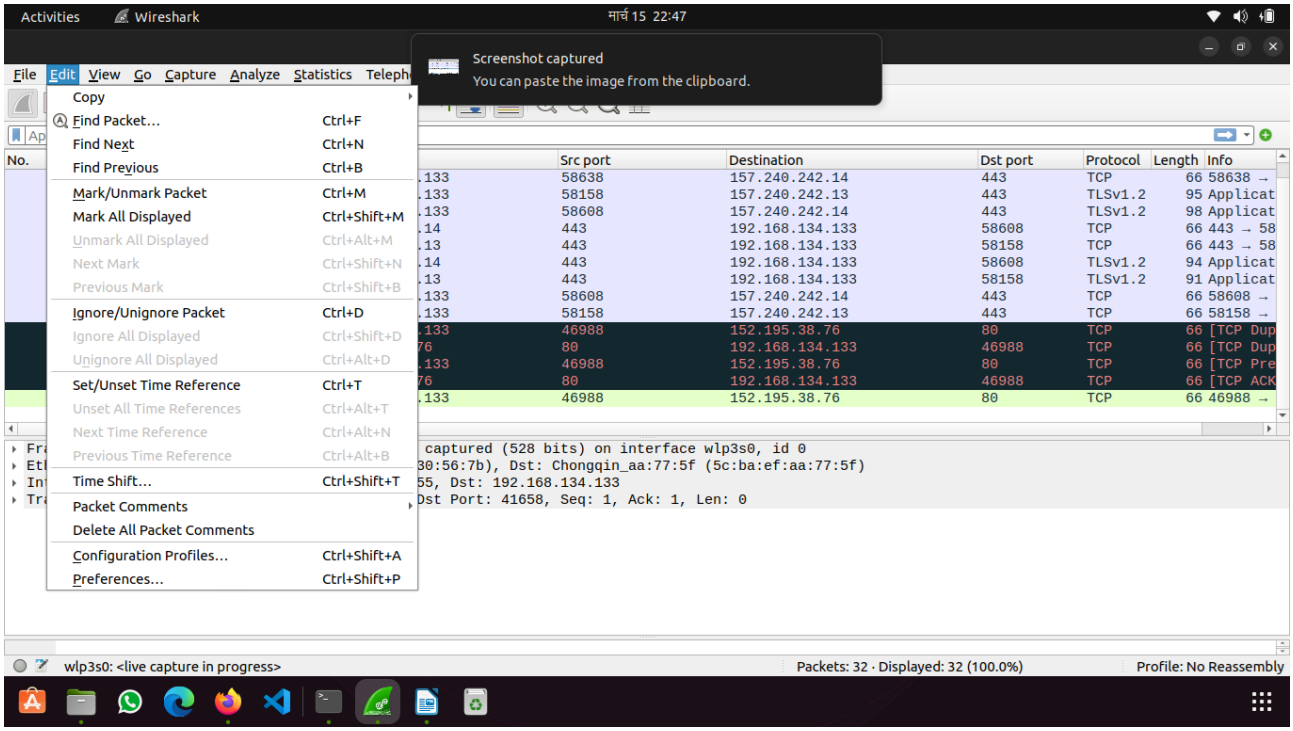
Capturing Network Traffic:

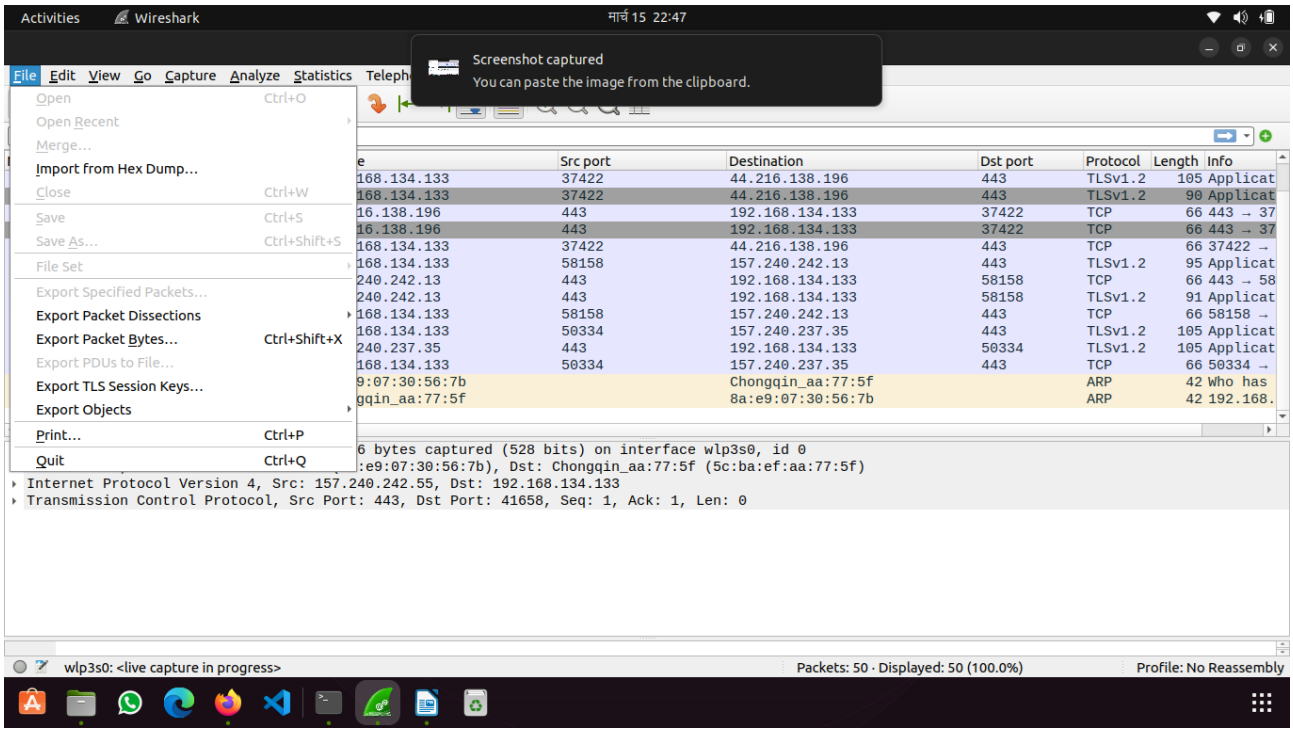
1. **Interface Selection:** Choose the network interface you want to capture traffic from (e.g., Wi-Fi, Ethernet).
2. **Capture Start/Stop:** Initiate capture by clicking the "Capture" button and stop it when desired.
3. **Filtering:** Wireshark allows filtering captured packets based on various criteria (e.g., protocol, IP address, port number).

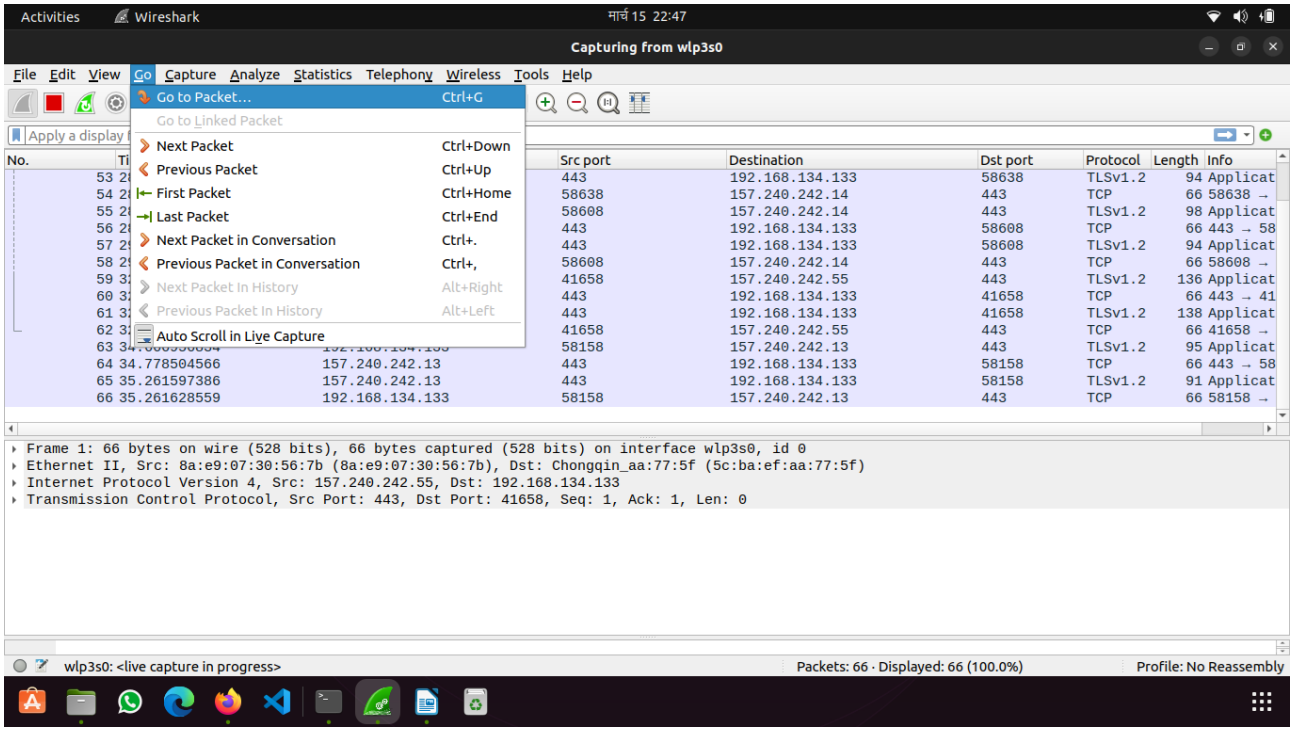
Screenshot Of Wireshark

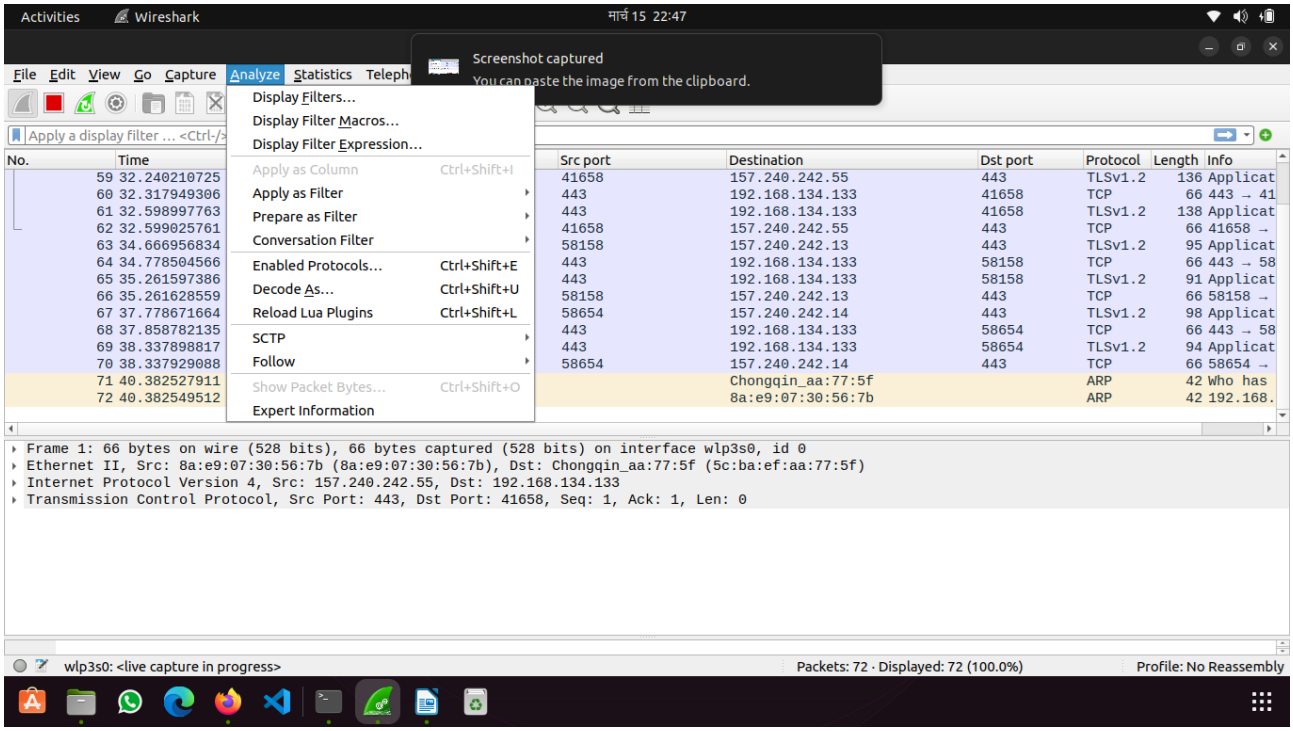

Activities
Wireshark
मार्च 15 22:46
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark
Open
/home/rajesh/College/7th Semester/Network Security/Lab Report/Wireshark/07_Malware Traffic Analysis Files/Certificate.pcapng (64 MB)
/home/rajesh/College/7th Semester/Network Security/Lab Report/Wireshark/07_Malware Traffic Analysis Files/Lab report.pcapng (80 MB)
…/College/7th Semester/Network Security/Lab Report/Wireshark/07_Malware Traffic Analysis Files/2014-11-16-traffic-analysis-exercise.pcap (2491 KB)
Capture
…using this filter: Enter a capture filter …
All interfaces shown
wlp3s0
any
Loopback: lo
enp2s0
bluetooth0
bluetooth-monitor
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 3.6.2 (Git v3.6.2 packaged as 3.6.2-2).
Ready to load or capture
No Packets
Profile: No Reassembly

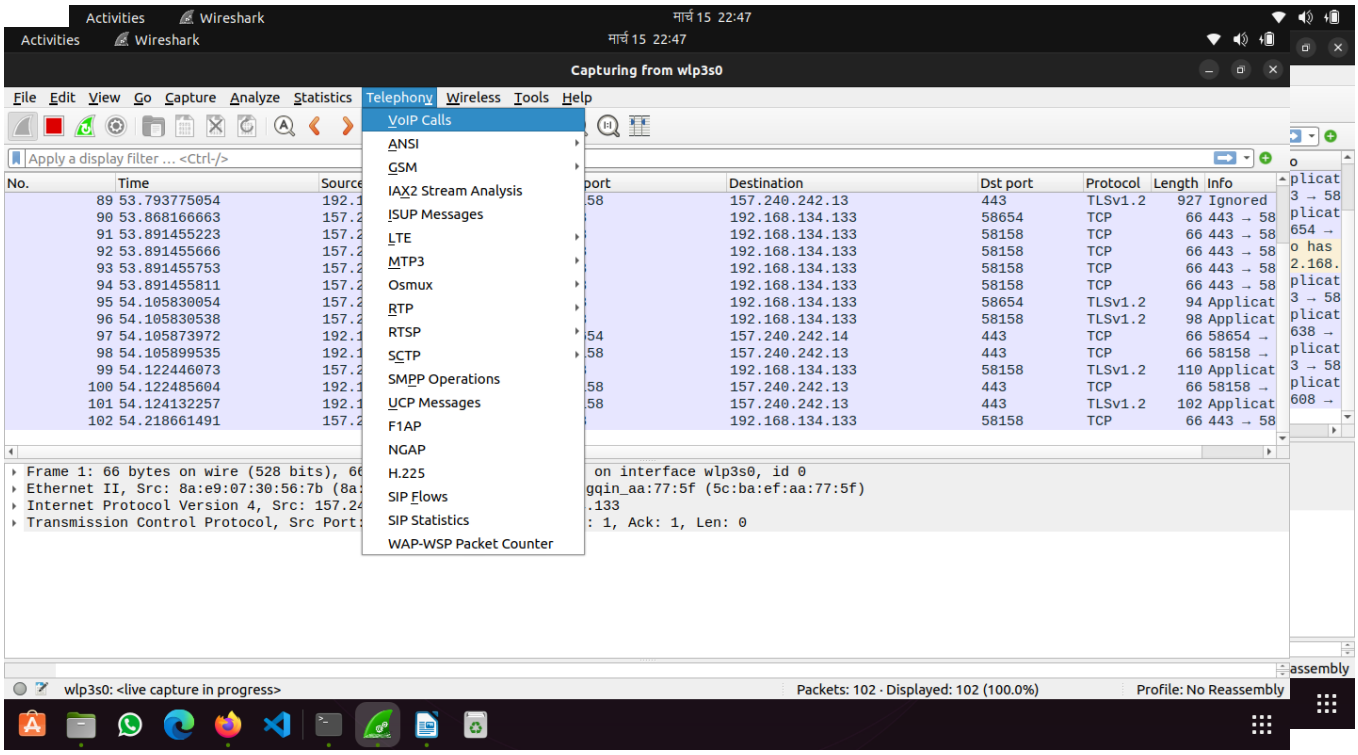

Activities
Wireshark
मार्च 15 22:47
Capturing from wlp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Src port Destination Dst port Protocol Length Info
14 10.892274120 8a:e9:07:30:56:7b Chongqin_aa:77:5f ARP 42 Who has
15 10.892294023 Chongqin_aa:77:5f 8a:e9:07:30:56:7b ARP 42 192.168.
16 11.652928768 192.168.134.133 58638 157.240.242.14 443 TLSv1.2 98 Applicat
17 11.728653774 157.240.242.14 443 192.168.134.133 58638 TCP 66 443 → 58
18 12.122665543 157.240.242.14 443 192.168.134.133 58638 TLSv1.2 94 Applicat
19 12.122698141 192.168.134.133 58638 157.240.242.14 443 TCP 66 58638 →
20 12.655334247 192.168.134.133 58158 157.240.242.13 443 TLSv1.2 95 Applicat
21 12.657842820 192.168.134.133 58608 157.240.242.14 443 TLSv1.2 98 Applicat
22 12.739011951 157.240.242.14 443 192.168.134.133 58608 TCP 66 443 → 58
23 12.739012372 157.240.242.13 443 192.168.134.133 58158 TCP 66 443 → 58
24 13.144151286 157.240.242.14 443 192.168.134.133 58608 TLSv1.2 94 Applicat
25 13.144151533 157.240.242.13 443 192.168.134.133 58158 TLSv1.2 91 Applicat
26 13.144189288 192.168.134.133 58608 157.240.242.14 443 TCP 66 58608 →
27 13.144212846 192.168.134.133 58158 157.240.242.13 443 TCP 66 58158 →
Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface wlp3s0, id 0
Ethernet II, Src: 8a:e9:07:30:56:7b (8a:e9:07:30:56:7b), Dst: Chongqin_aa:77:5f (5c:ba:ef:aa:77:5f)
Internet Protocol Version 4, Src: 157.240.242.55, Dst: 192.168.134.133
Transmission Control Protocol, Src Port: 443, Dst Port: 41658, Seq: 1, Ack: 1, Len: 0
wlp3s0: <live capture in progress>
Packets: 27 · Displayed: 27 (100.0%)
Profile: No Reassembly

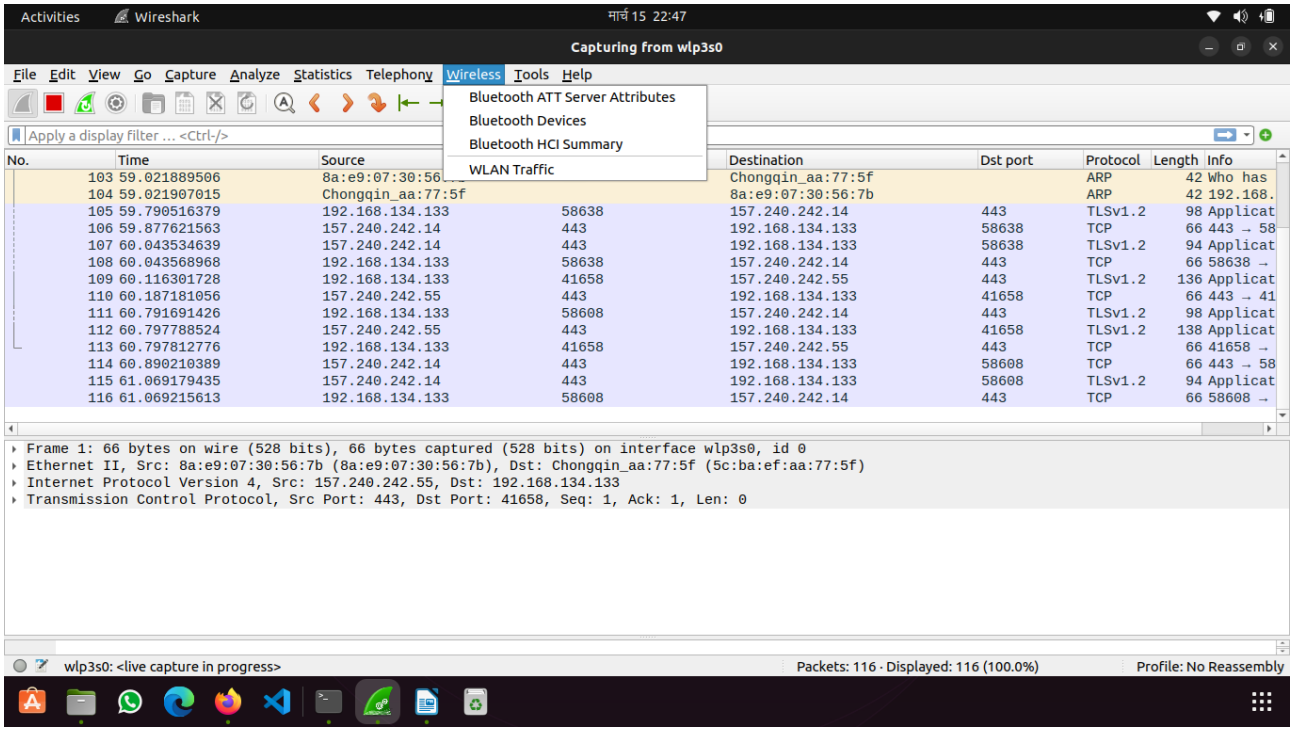

Activities
Wireshark
मार्च 15 22:47
Screenshot captured
You can paste the image from the clipboard.
File Edit View Go Capture Analyze Statistics Telephony
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift... Ctrl+Shift+T
Packet Comments
Delete All Packet Comments
Configuration Profiles... Ctrl+Shift+A
Preferences... Ctrl+Shift+P
No. Src port Destination Dst port Protocol Length Info
66 58638 →
95 Applicat
98 Applicat
66 443 → 58
66 443 → 58
94 Applicat
91 Applicat
66 58608 →
66 58158 →
66 [TCP Dup
66 [TCP Dup
66 [TCP Pre
66 [TCP ACK
66 46988 →
captured (528 bits) on interface wlp3s0, id 0
30:56:7b), Dst: Chongqin_aa:77:5f (5c:ba:ef:aa:77:5f)
55, Dst: 192.168.134.133
Dst Port: 41658, Seq: 1, Ack: 1, Len: 0
wlp3s0: <live capture in progress>
Packets: 32 · Displayed: 32 (100.0%)
Profile: No Reassembly

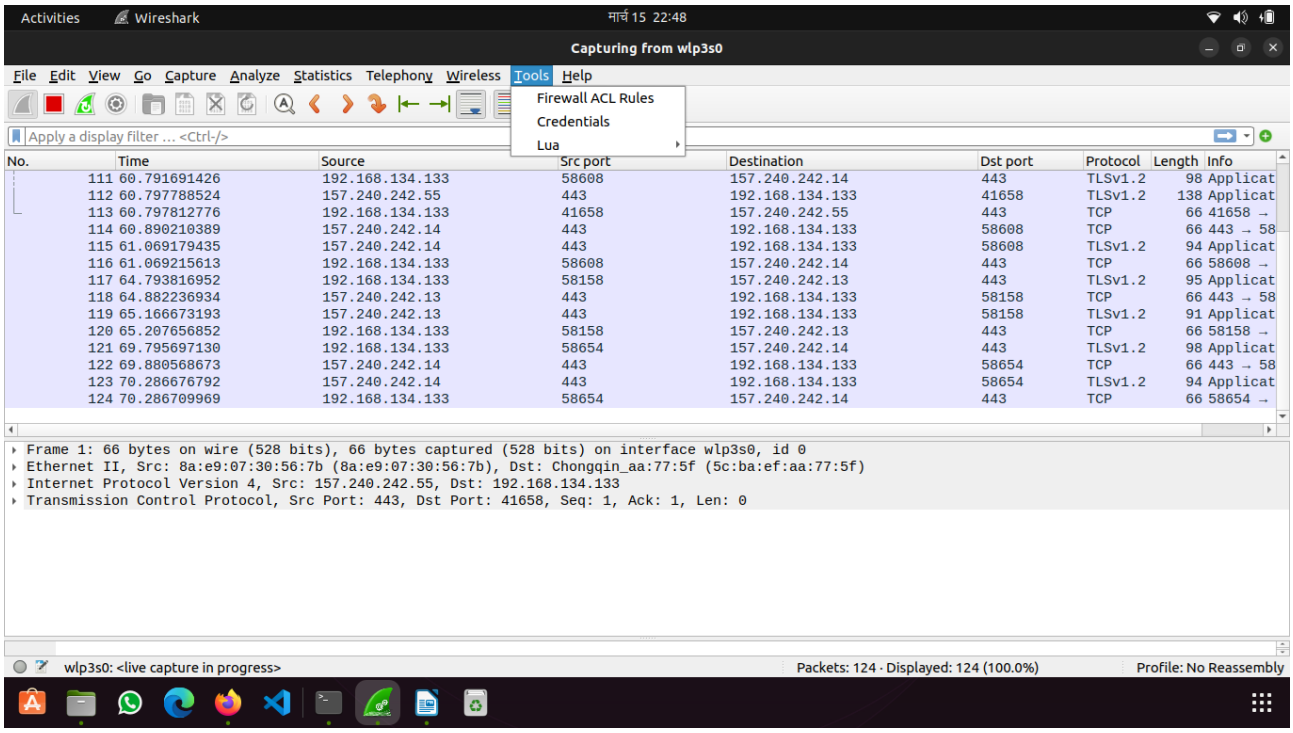

Activities
Wireshark
मार्च 15 22:47
Screenshot captured
You can paste the image from the clipboard.
File Edit View Go Capture Analyze Statistics Telephony
Open Ctrl+O
Open Recent
Merge...
Import from Hex Dump...
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Packet Bytes... Ctrl+Shift+X
Export PDUs to File...
Export TLS Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
Src port Destination Dst port Protocol Length Info
37422 44.216.138.196 443 TLSv1.2 105 Applicat
37422 44.216.138.196 443 TLSv1.2 90 Applicat
443 192.168.134.133 37422 TCP 66 443 → 37
443 192.168.134.133 37422 TCP 66 443 → 37
37422 44.216.138.196 443 TCP 66 37422 →
58158 157.240.242.13 443 TLSv1.2 95 Applicat
443 192.168.134.133 58158 TCP 66 443 → 58
443 192.168.134.133 58158 TLSv1.2 91 Applicat
58158 157.240.242.13 443 TCP 66 58158 →
50334 157.240.237.35 443 TLSv1.2 105 Applicat
443 192.168.134.133 50334 TLSv1.2 105 Applicat
50334 157.240.237.35 443 TCP 66 50334 →
Chongqin_aa:77:5f ARP 42 Who has
8a:e9:07:30:56:7b ARP 42 192.168.
6 bytes captured (528 bits) on interface wlp3s0, id 0
:e9:07:30:56:7b), Dst: Chongqin_aa:77:5f (5c:ba:ef:aa:77:5f)
Internet Protocol Version 4, Src: 157.240.242.55, Dst: 192.168.134.133
Transmission Control Protocol, Src Port: 443, Dst Port: 41658, Seq: 1, Ack: 1, Len: 0
wlp3s0: <live capture in progress>
Packets: 50 · Displayed: 50 (100.0%)
Profile: No Reassembly


Activities
Wireshark
मार्च 15 22:47
Capturing from wlp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Go to Packet... Ctrl+G
Go to Linked Packet
Next Packet Ctrl+Down
Previous Packet Ctrl+Up
First Packet Ctrl+Home
Last Packet Ctrl+End
Next Packet in Conversation Ctrl+.
Previous Packet in Conversation Ctrl+,
Next Packet In History Alt+Right
Previous Packet In History Alt+Left
Auto Scroll in Live Capture
Apply a display
No. Ti Src port Destination Dst port Protocol Length Info
53 443 192.168.134.133 58638 TLSv1.2 94 Applicat
54 58638 157.240.242.14 443 TCP 66 58638 →
55 58608 157.240.242.14 443 TLSv1.2 98 Applicat
56 443 192.168.134.133 58608 TCP 66 443 → 58
57 443 192.168.134.133 58608 TLSv1.2 94 Applicat
58 58608 157.240.242.14 443 TCP 66 58608 →
59 41658 157.240.242.55 443 TLSv1.2 136 Applicat
60 443 192.168.134.133 41658 TCP 66 443 → 41
61 443 192.168.134.133 41658 TLSv1.2 138 Applicat
62 41658 157.240.242.55 443 TCP 66 41658 →
63 58158 157.240.242.13 443 TLSv1.2 95 Applicat
64 34.778504566 157.240.242.13 443 192.168.134.133 58158 TCP 66 443 → 58
65 35.261597386 157.240.242.13 443 192.168.134.133 58158 TLSv1.2 91 Applicat
66 35.261628559 192.168.134.133 58158 157.240.242.13 443 TCP 66 58158 →
Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface wlp3s0, id 0
Ethernet II, Src: 8a:e9:07:30:56:7b (8a:e9:07:30:56:7b), Dst: Chongqin_aa:77:5f (5c:ba:ef:aa:77:5f)
Internet Protocol Version 4, Src: 157.240.242.55, Dst: 192.168.134.133
Transmission Control Protocol, Src Port: 443, Dst Port: 41658, Seq: 1, Ack: 1, Len: 0
wlp3s0: <live capture in progress>
Packets: 66 · Displayed: 66 (100.0%)
Profile: No Reassembly


Activities
Wireshark
मार्च 15 22:47
Screenshot captured
You can paste the image from the clipboard.
File Edit View Go Capture Analyze Statistics Teleph
Display Filters…
Display Filter Macros…
Display Filter Expression…
Apply as Column Ctrl+Shift+I
Apply as Filter
Prepare as Filter
Conversation Filter
Enabled Protocols… Ctrl+Shift+E
Decode As… Ctrl+Shift+U
Reload Lua Plugins Ctrl+Shift+L
SCTP
Follow
Show Packet Bytes… Ctrl+Shift+O
Expert Information
Apply a display filter … <Ctrl-/>
No. Time Src port Destination Dst port Protocol Length Info
59 32.240210725 41658 157.240.242.55 443 TLSv1.2 136 Applicat
60 32.317949306 443 192.168.134.133 41658 TCP 66 443 → 41
61 32.598997763 443 192.168.134.133 41658 TLSv1.2 138 Applicat
62 32.599025761 41658 157.240.242.55 443 TCP 66 41658 →
63 34.666956834 58158 157.240.242.13 443 TLSv1.2 95 Applicat
64 34.778504566 443 192.168.134.133 58158 TCP 66 443 → 58
65 35.261597386 443 192.168.134.133 58158 TLSv1.2 91 Applicat
66 35.261628559 58158 157.240.242.13 443 TCP 66 58158 →
67 37.778671664 58654 157.240.242.14 443 TLSv1.2 98 Applicat
68 37.858782135 443 192.168.134.133 58654 TCP 66 443 → 58
69 38.337898817 443 192.168.134.133 58654 TLSv1.2 94 Applicat
70 38.337929088 58654 157.240.242.14 443 TCP 66 58654 →
71 40.382527911 Chongqin_aa:77:5f ARP 42 Who has
72 40.382549512 8a:e9:07:30:56:7b ARP 42 192.168.
Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface wlp3s0, id 0
Ethernet II, Src: 8a:e9:07:30:56:7b (8a:e9:07:30:56:7b), Dst: Chongqin_aa:77:5f (5c:ba:ef:aa:77:5f)
Internet Protocol Version 4, Src: 157.240.242.55, Dst: 192.168.134.133
Transmission Control Protocol, Src Port: 443, Dst Port: 41658, Seq: 1, Ack: 1, Len: 0
wlp3s0: <live capture in progress>
Packets: 72 · Displayed: 72 (100.0%)
Profile: No Reassembly


Activities
Wireshark
मार्च 15 22:47
Capturing from wlp3s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
VoIP Calls
ANSI
GSM
IAX2 Stream Analysis
ISUP Messages
LTE
MTP3
Osmux
RTP
RTSP
SCTP
SMPP Operations
UCP Messages
F1AP
NGAP
H.225
SIP Flows
SIP Statistics
WAP-WSP Packet Counter
Apply a display filter … <Ctrl-/>
No. Time Source Destination Dst port Protocol Length Info
89 53.793775054 157.240.242.13 443 TLSv1.2 927 Ignored
90 53.868166663 192.168.134.133 58654 TCP 66 443 → 58
91 53.891455223 192.168.134.133 58158 TCP 66 443 → 58
92 53.891455666 192.168.134.133 58158 TCP 66 443 → 58
93 53.891455753 192.168.134.133 58158 TCP 66 443 → 58
94 53.891455811 192.168.134.133 58158 TCP 66 443 → 58
95 54.105830054 192.168.134.133 58654 TLSv1.2 94 Applicat
96 54.105830538 192.168.134.133 58158 TLSv1.2 98 Applicat
97 54.105873972 157.240.242.14 443 TCP 66 58654 →
98 54.105899535 157.240.242.13 443 TCP 66 58158 →
99 54.122446073 192.168.134.133 58158 TLSv1.2 110 Applicat
100 54.122485604 157.240.242.13 443 TCP 66 58158 →
101 54.124132257 157.240.242.13 443 TLSv1.2 102 Applicat
102 54.218661491 192.168.134.133 58158 TCP 66 443 → 58
wlp3s0: <live capture in progress>
Packets: 102 · Displayed: 102 (100.0%)
Profile: No Reassembly





Analyzing Captured Traffic:

- **Packet List:** Displays a list of captured packets with timestamps, source and destination information, protocol used, and packet size.
- **Packet Details:** Provides detailed information about a selected packet, including headers and data payload (often displayed in hexadecimal format).

- **Dissection Pane:** Decodes the different layers (protocols) within a packet, allowing deeper inspection of the communication details.
- **Conversations:** Groups related packets together based on conversation flows between devices.

Key Features of Wireshark:

- **Protocol Decoding:** Supports a wide range of network protocols, allowing analysis of various communication types.
- **Filtering and Search:** Powerful filtering capabilities help isolate specific packets of interest.
- **Packet Coloring Rules:** Visually distinguish different types of traffic based on user-defined rules.
- **VoIP Analysis:** Analyzes Voice over IP (VoIP) calls, providing insights into call quality and performance.
- **Statistics:** Generates network traffic statistics for troubleshooting and performance assessments.
- **Exporting Data:** Allows exporting captured traffic data in various formats for further analysis with other tools.

Benefits of Using Wireshark:

- **Network Troubleshooting:** Helps diagnose network connectivity issues by identifying errors or unexpected behavior in network traffic.
- **Security Analysis:** Enables the detection of suspicious activity on a network, aiding in malware and intrusion detection.
- **Protocol Understanding:** Provides a valuable tool for learning and understanding how different network protocols function.
- **Performance Monitoring:** Allows monitoring network performance by analyzing traffic patterns and identifying bottlenecks.

Conclusion:

Wireshark is a versatile and feature-rich network traffic analyzer that empowers users to gain deep insights into network communication. Its capabilities benefit network professionals, security analysts, and anyone interested in understanding how data flows across a network.

Lab 5: Exploring Hash Functions

Introduction

This lab report investigates Hash functions, a cryptographic primitive crucial for data integrity verification but distinct from encryption. We'll explore their functionalities, analyze their properties, and observe their behavior through practical exercises.

Encryption scrambles data to ensure confidentiality, while Hash functions generate a fixed-size digest (fingerprint) from an arbitrary input. This digest helps verify if the original data has been altered.

Objectives

- Understand the concept of Hash functions and their role in data integrity.
- Analyze key properties of Hash functions like collision resistance and avalanche effect.
- Experiment with different Hashing algorithms (e.g., MD5, SHA-256) and observe their output.

Hash Functions Explained

A Hash function takes an input of arbitrary length and generates a fixed-size output, called a hash value or message digest. This process is mathematically irreversible, meaning it's infeasible to retrieve the original data from the hash value alone.

Key Properties of Hash Functions

- **Collision Resistance:** It's extremely difficult to find two different inputs that produce the same hash value (collision).
- **Preimage Resistance:** Given a hash value, it's computationally hard to find the original input that generated it.
- **Second Preimage Resistance:** Given an input, it's challenging to find another different input that generates the same hash value.
- **Avalanche Effect:** Small changes in the input data should result in significant changes in the hash value.

Experimentation with Hashing Algorithms

This lab can involve using online tools or code libraries to experiment with different Hashing algorithms:

1. **Choose Hashing Algorithm:** Select a popular Hashing algorithm like MD5, SHA-256, or SHA-3.
2. **Input Data:** Enter some data as input for the Hash function.
3. **Generate Hash Value:** Calculate the hash value using the chosen algorithm.
4. **Modify Input Data:** Make a slight change to the original input data.
5. **Generate New Hash Value:** Calculate the hash value for the modified data.

Observations:

- The same input data will always generate the same hash value using the same algorithm.
- Even a minor change in the input data will result in a significantly different hash value.
- Finding collisions is computationally expensive for good Hash functions.

Conclusion

Hash functions are a valuable cryptographic tool for ensuring data integrity. They offer efficient verification of whether data has been tampered with during transmission or storage. While not a substitute for encryption, Hash functions play a vital role in securing data integrity in various applications.

Lab 6: Changing the MySQL Superuser Username and Password In Wireshark

Objective

The objective of this lab is to demonstrate how to change the MySQL superuser's username and password securely and to analyze the network traffic using Wireshark to understand the security implications of the process.

Equipment

1. MySQL server
2. Wireshark
3. Network connection between the MySQL server and the Wireshark machine

Procedure

1. **Prepare MySQL Server:**
    - ⓾ Make sure your MySQL server is running and accessible on the network.
    - ⓾ Identify the current superuser (root) credentials to be changed.
2. **Prepare Wireshark:**
    - ⓾ Launch Wireshark on a separate machine connected to the same network as the MySQL server.
    - ⓾ Select the appropriate network interface to capture traffic.
3. **Start Capturing Traffic:**
    - ⓾ Start the packet capture in Wireshark to capture network traffic.
    - ⓾ Apply a filter to capture only MySQL-related traffic. You can use a filter like `mysql` or `port 3306` (the default MySQL port).
4. **Change MySQL Superuser Username and Password:**
    - ⓾ Connect to the MySQL server using a MySQL client such as MySQL Workbench, command-line client, or any other tool.
    - ⓾ Execute SQL commands to change the superuser's username and password. For example:

```
sql
```

```
    ⓾ ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

Replace `'new_password'` with your desired new password.

2. **Analyze Wireshark Capture:**
    - ⓾ Stop the packet capture in Wireshark after the changes are made in MySQL.

- ⓾ Analyze the captured packets related to MySQL communication.
- ⓾ Look for packets containing authentication-related information (such as username, password, authentication method).
- ⓾ Use Wireshark's packet details and follow TCP streams if necessary to inspect the MySQL protocol exchanges.

3. **Interpretation and Security Considerations:**
   - ⓾ Identify packets containing the changed username and password. Ensure that sensitive information like passwords is not transmitted in plain text.
   - ⓾ Note the authentication method used (e.g., MySQL native password, SSL/TLS).
   - ⓾ Consider the security implications of transmitting credentials over the network and ensure that encryption mechanisms like SSL/TLS are used to secure sensitive data.
   - ⓾ Review the packet capture to understand how MySQL authentication works and the potential risks associated with unencrypted traffic.

Conclusion

In this lab, we successfully changed the MySQL superuser's username and password while capturing and analyzing network traffic using Wireshark. We observed the authentication process and highlighted the importance of using encryption for securing sensitive information during database interactions over the network. Understanding network protocols and security best practices is crucial for maintaining the integrity and confidentiality of data in networked environments.

Lab 7: Nessus Essential – Monitoring Tool

Objective

The objective of this lab is to explore Nessus Essential as a monitoring tool for vulnerability scanning. We will cover the basic monitoring capabilities, installation procedure, and perform a vulnerability scan to identify potential security issues.

Procedure

1. **Introduction to Nessus:**
    - Nessus is a widely-used vulnerability scanning tool developed by Tenable.
    - It helps in identifying vulnerabilities, mis-configurations, and potential security threats in network devices, systems, and applications.
2. **Installation of Nessus:**
    - Download Nessus Essential from the Tenable website (https://www.tenable.com/downloads/nessus) based on your operating system (e.g., Windows, Linux).
    - Follow the installation instructions provided by Tenable to install Nessus on your system.
3. **Starting Nessus:**
    - Once installed, start the Nessus service on your system. This usually involves starting the Nessus daemon or service, which listens on a specific port (default is 8834).
4. **Accessing Nessus Web Interface:**
    - Open a web browser and navigate to https://localhost:8834 (replace "localhost" with the IP address or hostname of your Nessus server if accessing remotely).
    - You will be prompted to log in with the default credentials (username: `admin`, password: initially set during installation).
5. **Basic Monitoring with Nessus:**
    - In the Nessus web interface, explore the dashboard and various tabs (e.g., Scans, Policies, Reports) to familiarize yourself with the interface.
    - Create a new scan policy based on your requirements (e.g., scan target IP ranges, scan frequency, types of vulnerabilities to check).
    - Initiate a vulnerability scan using the created policy to start monitoring and identifying potential vulnerabilities in the specified targets.
6. **Analyzing Vulnerabilities:**

- Once the scan is completed, review the scan results and identified vulnerabilities.
- Nessus categorizes vulnerabilities based on severity (e.g., Critical, High, Medium, Low) and provides detailed information about each vulnerability, including description, impact, and remediation steps.

7. **Interpretation of Results:**
   - Analyze the vulnerabilities found by Nessus and prioritize them based on severity and potential impact on your system or network.
   - Take necessary actions to remediate identified vulnerabilities, such as applying patches, updating configurations, or implementing security measures.
8. **Conclusion and Security Recommendations:**
   - Nessus Essential provides valuable insights into the security posture of your systems and networks by continuously monitoring for vulnerabilities.
   - Regular vulnerability scanning and proactive remediation efforts are crucial for maintaining a secure environment and reducing the risk of security breaches.
   - Implement best practices such as regular scanning, timely patching, and following security guidelines to enhance overall security posture.

Security Considerations

- Ensure that Nessus is used in compliance with legal and ethical guidelines. Obtain necessary permissions before scanning systems or networks.
- Keep Nessus updated with the latest plugins and software versions to detect newly discovered vulnerabilities.
- Secure access to the Nessus web interface using strong passwords and encryption (HTTPS).
- Regularly review and act upon Nessus scan results to mitigate identified vulnerabilities effectively.