CYBERSECURITY FOUNDATIONS & NETWORKING LAB NOTES

Prepared by: Chennari Mohammed Idris

Date: February 21, 2026

Project: Task 1 - Foundation & Environment Setup

1. The Core: CIA Triad

The foundation of all security policies I implemented in this lab is the **CIA Triad**:

- **Confidentiality:** Ensuring data is only accessible to authorized users.
    - *Applied:* Used a **Host-Only Adapter** to keep lab traffic invisible to the outside world.
- **Integrity:** Guaranteeing that data has not been altered or tampered with.
    - *Applied:* Verified connectivity using ping and monitored packet health via **Wireshark**.
- **Availability:** Ensuring systems and data are accessible when needed.
    - *Applied:* Configured static environments to prevent downtime during testing.

2. Threat Landscape & Attack Vectors

Understanding what we are defending against is crucial.

- **Common Threats:** * *Phishing/Social Engineering:* Deceiving users to gain access.
    - *Malware/Ransomware:* Malicious software designed to disrupt or lock systems.
    - *DDoS:* Overwhelming a service to break its **Availability**.
- **Vectors:** I learned that an **Insider Threat** (someone already inside the network) can be just as dangerous as an external hacker using a **Wireless Attack**.

3. Networking & The OSI Model

- To analyze the "Ping" test in Wireshark, I mapped the data flow across the **OSI Model**:

Layer	Name	Function	Protocol Example
7	Application	User Interface	HTTP, DNS, SMTP
4	Transport	End-to-End Connection	**TCP** (Reliable), **UDP** (Fast)

Layer	Name	Function	Protocol Example
3	Network	Routing & Logical Addressing	**IP**, **ICMP** (Ping)
2	Data Link	Physical Addressing	MAC Address, Ethernet

4. Cryptography Basics

I explored how we protect data at rest and in transit:

- **Encryption:** * *Symmetric:* Same key for lock/unlock (Fast).
    - *Asymmetric:* Public key to lock, Private key to unlock (Secure).
- **Hashing:** A "one-way" fingerprint (MD5/SHA256) used to verify **Integrity**. Unlike encryption, a hash cannot be reversed.
- **Digital Certificates:** Used in **SSL/TLS** to prove a website's identity (the "S" in HTTPS).

HANDS-ON “ENCRYPT AND DECRYPT USING OPENSSL”:

Objective: To demonstrate the practical application of symmetric encryption and decryption to ensure data Confidentiality.

1. File Creation

I created a plaintext file containing sensitive information to be protected.

- **Command:** echo "this is a secret message" > secret.txt

2. Symmetric Encryption (AES-256)

I used the Advanced Encryption Standard (AES) with a 256-bit key to encrypt the file. I applied the PBKDF2 (Password-Based Key Derivation Function 2) to ensure the password was securely hashed before being used as a key.

- **Command:** openssl enc -aes-256-cbc -pbkdf2 -iter 100000 -in secret.txt -out secret.enc
- **Observation:** After encryption, attempting to read the file using cat secret.enc resulted in unreadable ciphertext (gibberish). This confirms that the data is protected.

3. Decryption and Verification

To regain access to the data, I used the decryption flag with the original password.

- **Command:** openssl enc -aes-256-cbc -pbkdf2 -iter 100000 -d -in secret.enc -out recovered.txt
- **Result:** The command cat recovered.txt displayed the original message perfectly, confirming the integrity of the encryption/decryption process.

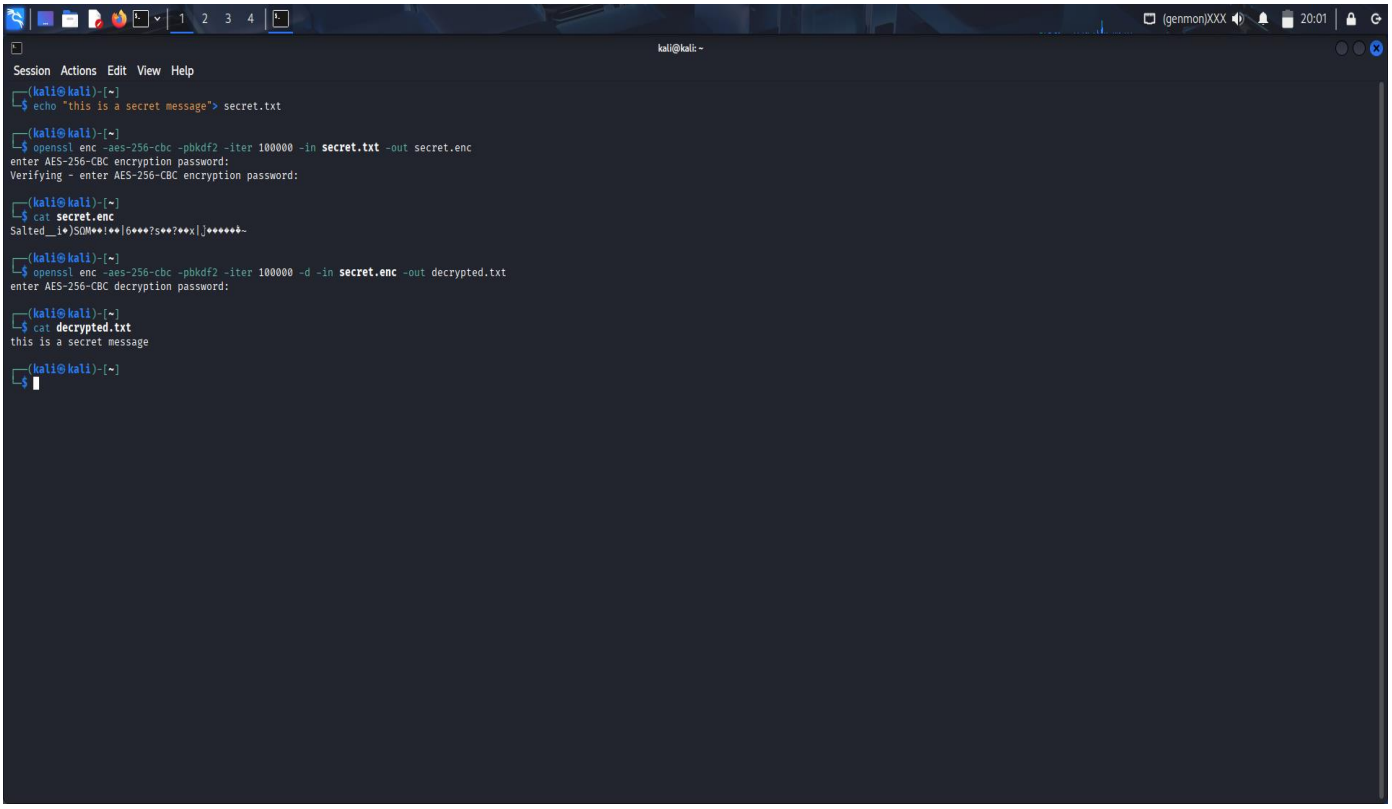

Image : Cryptographic operations using Openssl