Introduction to Computer Security
CSCE 5550
Phase - 2

Group -8
Team members:-
Pooja Peechara
Rachana Reddy Sunki
Mounika Nuchu
Lagadapati Kondarao
Yenugu Ruthiksha Reddy

Overview:- In this phase we implemented and described the encryption and decryption as part of the project. We are encrypting the lab exercises which we have worked on, and the user will not be able to access the files until we give the key to the encrypted files.

Encryption:- We have encrypted the files by using Python. Fernet module, which is one of the secure ways to encrypt and decrypt files.

Working:- The encrypt_files function takes two args i.e. the path to the file directory to encrypt and the encryption key.
2. It navigates through the file path mentioned which will encrypt each file that does not have an .enc extension. Here .enc is the encrypted files type.
3. The original file is replaced with its encrypted version, and the .enc extension is appended to the file's name.
4. If an encryption key is not provided, the code will automatically generate a new encryption key and feed it to the console.

Command line prompts:- To encrypt files in a directory, we ran the following command in the terminal:
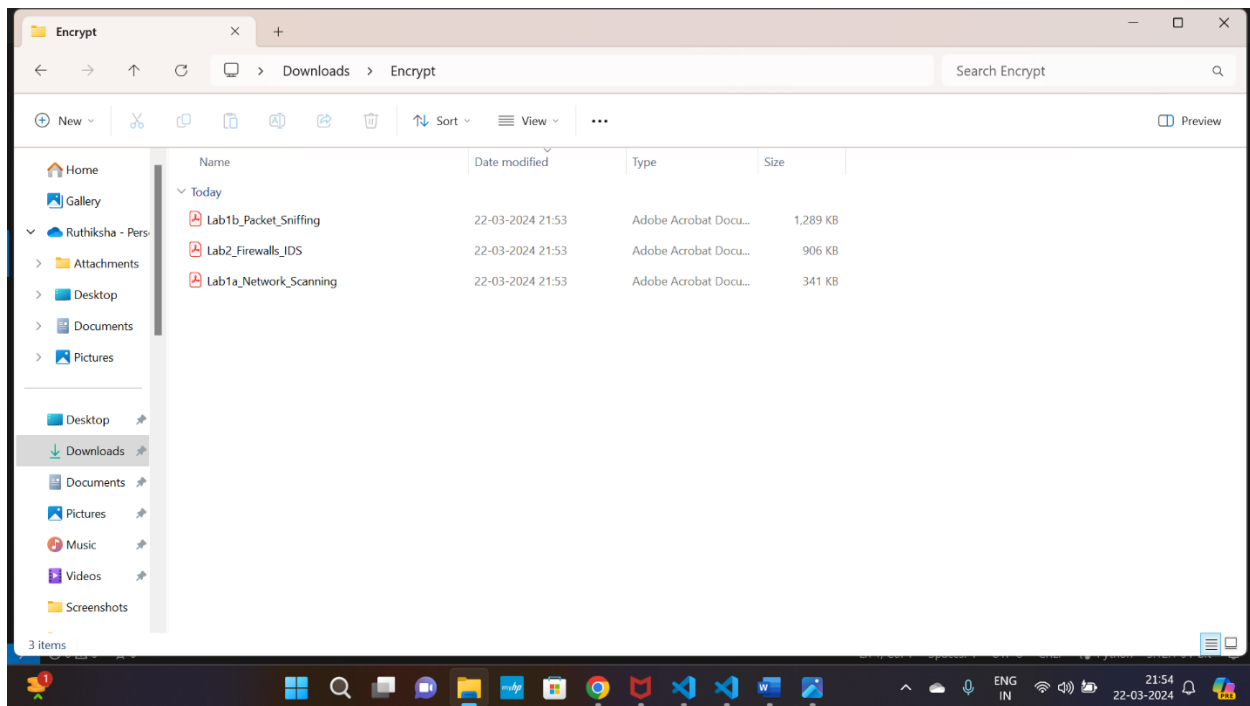
python enc.py encrypt "<path_to_directory>" --key "<encryption_key>"

If you do not have an encryption key, ignore the --key argument, and the code will Automatically generate it for you.

Encrypted Files:-
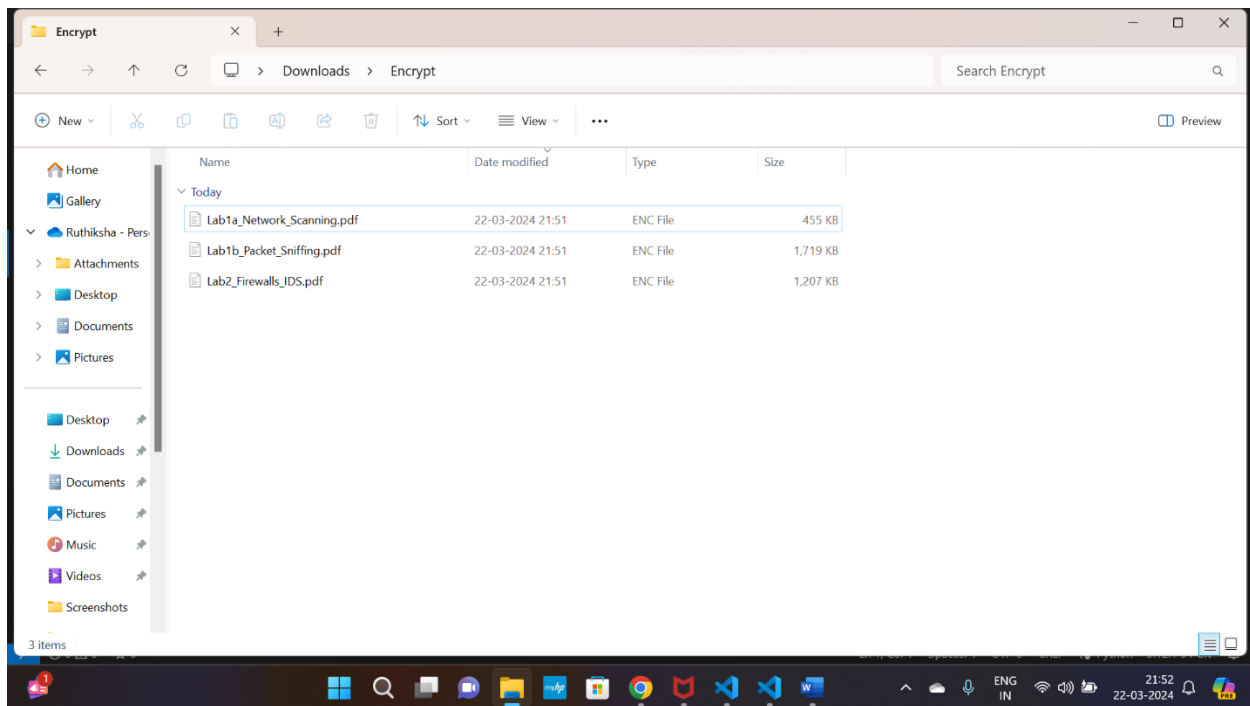
Actual Files/decrypted files:-

Decryption:-

The decryption code will now trace back to the process carried out by the encryption methods used to encrypt the files.

Working:- The decrypt_files function takes the same arguments as encrypt_files.
2. Now, The code to decrypt the files will search for files with a .enc extension in the given path and it will decrypt them using the key used to encrypt the files.
3. The decryption to the files are performed and the .enc extension is removed. Now, the user or the organization will have access to the files again.

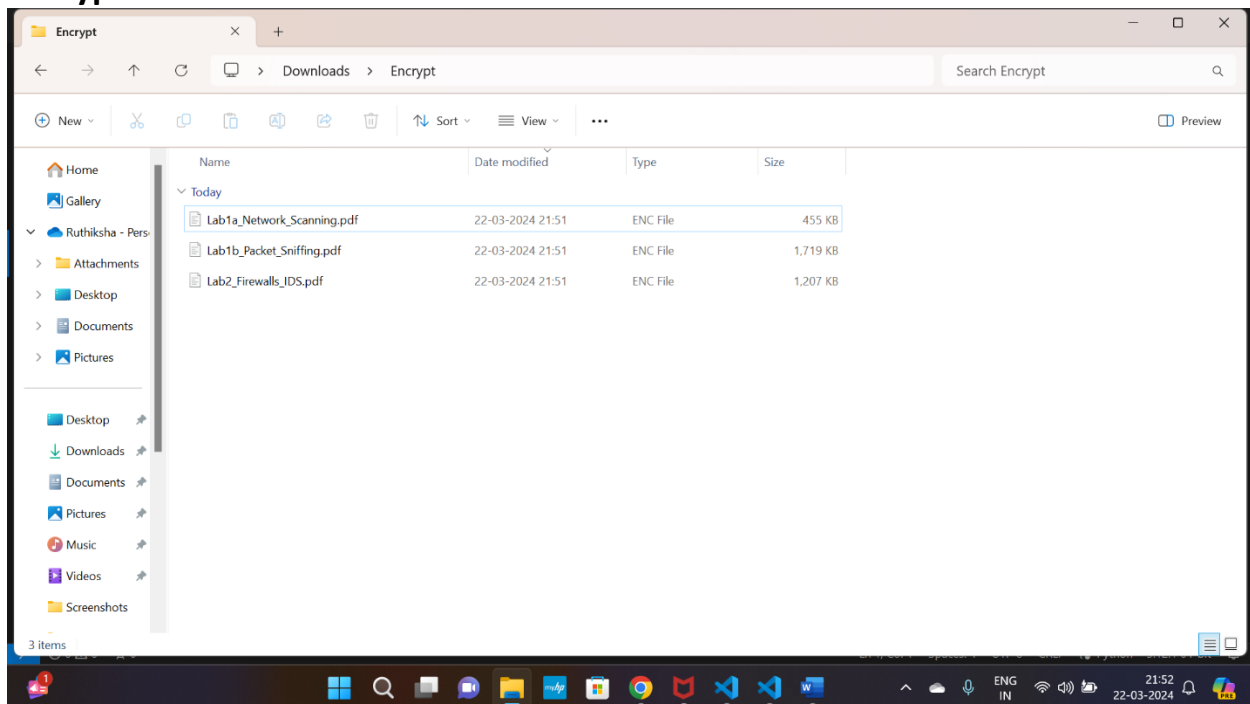Usage:- To decrypt the files in the directory, run the following command:

 [python enc.py decrypt "<path_to_directory>" --key "<encryption_key>"]

It is crucial and important to use the same encryption key to decrypt the files. Otherwise, you won't be able to decrypt them without the possible key.
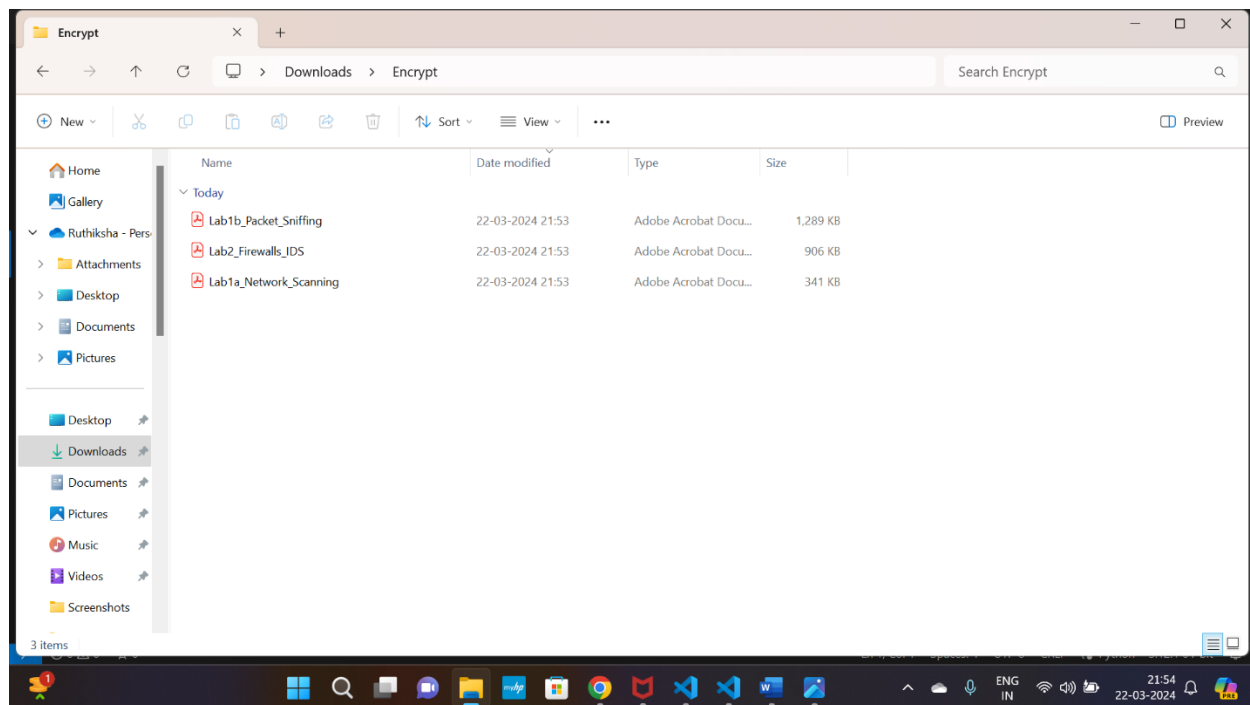
Screenshot:-

Encryption:-

Decryption:-

References:-

A.Usha, M. (2016, December 24). Performance Study of Key Developer Data Encryption and Decryption Algorithm (KDDEDA) with AES, DES and BLOWFISH. *International Journal of Engineering and Computer Science*. https://doi.org/10.18535/ijecs/v5i12.62

"All Your Files Have Been Encrypted!"—Ransomware Attack at Keystone Insurance. (2021, May 11). *Journal of Applied Business and Economics*, *23*(2). https://doi.org/10.33423/jabe.v23i2.4100

Brown, Alan. "Understanding Fernet: A Simple, Secure, Symmetric Encryption Algorithm." Medium, 2017.

Kumar, Ajay, and Ankit Kumar. 2016. File Encryption System Based on Symmetric Key Cryptography . International Journal Of Engineering And Computer Science.