

Project Report: Simple Ransomware Simulator

Abstract:

With the rising incidents of ransomware attacks across the globe, it is essential for cybersecurity students and professionals to understand how ransomware behaves. This project aims to simulate a basic ransomware mechanism in a safe, controlled environment. It showcases file encryption and decryption using symmetric key cryptography, mimicking real-world ransomware tactics. This simulator is developed purely for educational purposes and demonstrates how critical encryption is in both offense and defense strategies within cybersecurity.

Objective:

To develop a simple ransomware simulator that encrypts files in a specified folder and allows them to be decrypted only with the correct key. The objective is to educate learners about the working mechanism of ransomware and the importance of secure key management and backup strategies.

Tools & Technologies Used:

- **Programming Language:** Python 3.x
- **Encryption Library:** cryptography (Fernet symmetric encryption)
- **Operating System:** Windows/Linux
- **IDE:** VS Code / PyCharm
- **Folder/File Access:** Python os module for recursive directory handling

Problem Statement:

Ransomware is a major cybersecurity threat that locks users out of their own data by encrypting it and demanding ransom for access. Understanding its inner workings can help cybersecurity professionals design effective detection and mitigation strategies.

System Requirements:

- Python 3 installed
- cryptography library (pip install cryptography)
- A directory with sample files to simulate the attack

Project Structure:

ransomware_simulator/

```
|— encryptor.py    # Encrypts all files in target folder
|— decryptor.py    # Decrypts all files using saved key
|— key.key         # Secret key generated for encryption/decryption
|— sample_files/   # Folder with files to be encrypted/decrypted
|   |— file1.txt
|   |— file2.txt
|— README.txt      # Instructions and disclaimer
```

Working Mechanism / Steps Involved:

1. Encryption Phase (encryptor.py)

- A new key is generated using `Fernet.generate_key()` and saved as `key.key`.
- All files inside the `sample_files/` directory are read and encrypted using the key.
- The original contents of each file are replaced with their encrypted versions.

2. Decryption Phase (decryptor.py)

- The previously saved key (`key.key`) is loaded.
- Each encrypted file is read, decrypted using the key, and restored to its original content.

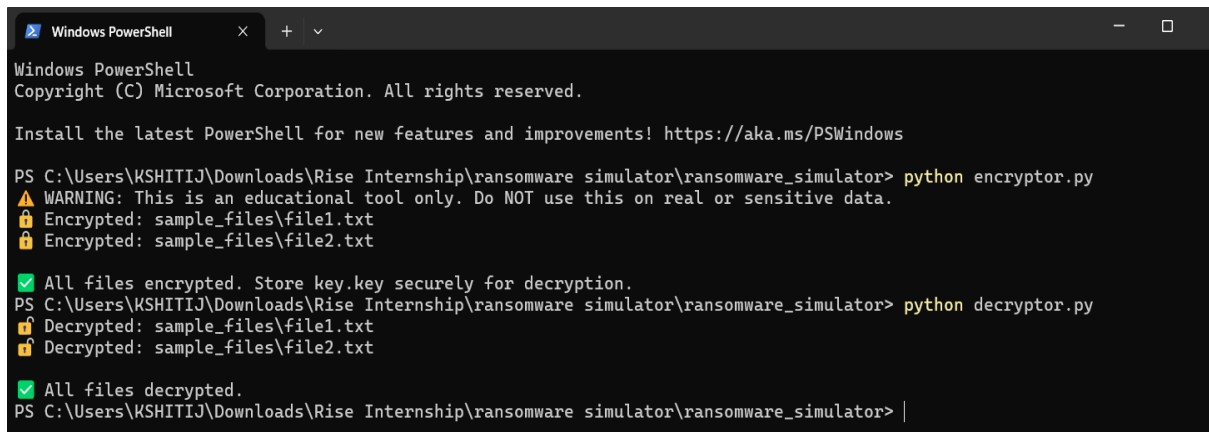
3. Safety Measures

- A warning message is displayed before encryption to ensure the user understands it's for learning only.
- Sample files are used instead of real or sensitive data.

Expected Outcome:

- All files in the specified folder will be encrypted and unreadable.
- Only by using the correct key stored in `key.key`, the original files can be decrypted and restored.
- Demonstrates how ransomware affects files and why backup and security controls are essential.

Screenshots:



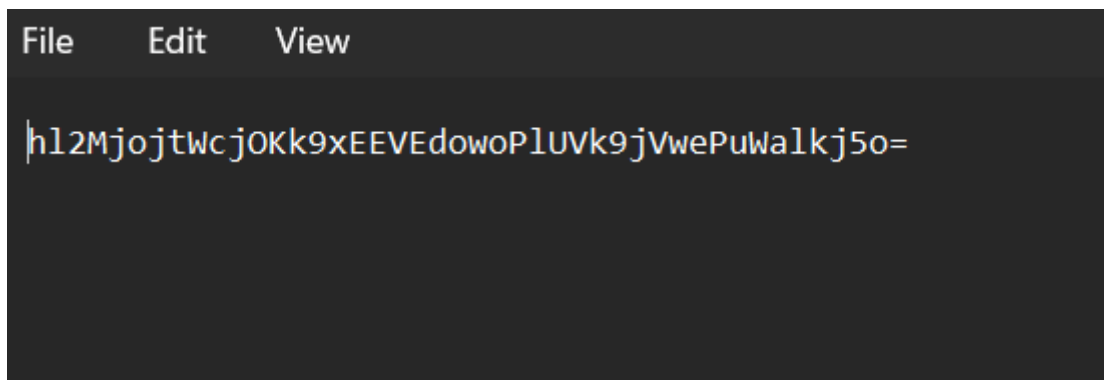
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\KSHITIJ\Downloads\Rise Internship\ransomware simulator\ransomware_simulator> python encryptor.py
⚠ WARNING: This is an educational tool only. Do NOT use this on real or sensitive data.
🔒 Encrypted: sample_files\file1.txt
🔒 Encrypted: sample_files\file2.txt

✅ All files encrypted. Store key.key securely for decryption.
PS C:\Users\KSHITIJ\Downloads\Rise Internship\ransomware simulator\ransomware_simulator> python decryptor.py
🔓 Decrypted: sample_files\file1.txt
🔓 Decrypted: sample_files\file2.txt

✅ All files decrypted.
PS C:\Users\KSHITIJ\Downloads\Rise Internship\ransomware simulator\ransomware_simulator> |
```



Conclusion:

The Simple Ransomware Simulator successfully demonstrates a basic form of ransomware behavior using Python and symmetric encryption. It provides valuable insights into the functioning of ransomware, making it a useful tool for cybersecurity education. By simulating encryption and decryption, this project emphasizes the importance of security practices such as data backups, key management, and user awareness in mitigating ransomware threats.