Cyber Security Project Report

National University of Computer and Emerging Sciences

Password Manager

Batch: 2023

Section: CY-3A

Group Members

Muhammad Hammad (23K-2005)

Muhammad Sami Ashfaq (23K-2031)

Professor

Fahad Samad

Department of Cyber Security

National University of Computer and Emerging Sciences

– FAST Karachi Campus

Acknowledgement

We would like to express our sincere gratitude and appreciation to our supervisor Mr. Fahad Samad for their inspiration, expert guidance and supervision during the research work on completing the dissertation in time.

Abstract

This project presents a password manager that securely stores, manages, and analyzes passwords using MD5 and RSA encryption and decryption. The system features user registration, credential management, password analysis and generation.

Problem Statement

In today's digitally connected world, passwords have become the primary defense mechanism for protecting our online identities. However, with the ever-increasing number of online accounts, managing passwords securely has become a daunting task. Weak passwords, passwords reuse, and inadequate password storage can lead to devastating consequences, including identity theft, financial loss, and compromised personal data.

Project Overview

The Secured Password Manager project aims to provide a comprehensive solution for managing passwords securely. The project involves designing and developing a software application that enables users to store, organize and analyze their passwords in a secure and efficient manner. The system will provide a robust and reliable solution for password management, addressing the growing need for secure password practices.

System Architecture

- Frontend: A user-friendly interface designed using a Tkinter (python library) for user registration, login and credential management.
- Backend: Secure password storage using advanced MD5 and RSA encryption and decryption algorithms implemented in Python.
- Database: JSON files are used as databases for storing user data and credentials.

Proposed Solution

To combat these threats, this project introduces a comprehensive Secure Password Manager designed to efficiently store, manage and analyze passwords. Leveraging advanced encryption techniques, including MD5 and RSA encryption/decryption algorithms, this system ensures the confidentiality and integrity of stored passwords. The primary objective of this project is to provide a robust and reliable solution for password management, mitigating the risks associated with password-related security breaches and promoting a safe online environment.

Workflow

1. **User Registration:** The user registers for the password manager system by providing a username and password.
2. **Password Hashing and Storage:** The user’s password is hashed using a secure hashing algorithm that is MD5 and stored in database.
3. **Password Encryption:** When the user adds a new password to the system, the password is encrypted using a secure encryption algorithm that is RSA.

4. **Password Storage:** The encrypted password is stored in the database, along with the user's username and other relevant information.
5. **Password Retrieval:** When the user requests to receive a stored password, the system decrypts the password using the secure decryption algorithm that is RSA.
6. **Password Analysis:** The system analyzes the user's passwords to identify potential security risks like weak passwords.
7. **Password Generation:** The system generates a strong, unique password for the user based on their preferred length.
8. **User Authentication:** The user is authenticated using their username and password, to ensure that only authorized users can access the system.

Sources

- **os**: Python standard library, used for interacting with the operating system (e.g., file and directory operations).
Documentation: os module

- **re**: Python standard library, used for regular expression matching and text processing.
Documentation: re module

- **struct**: Python standard library, used for handling binary data and packing/unpacking data into bytes.
Documentation: struct module

- **json**: Python standard library, used for encoding and decoding JSON data.
Documentation: json module

- **tkinter**: Python standard library, used for creating graphical user interfaces.
Documentation: tkinter module

- **math**: Python standard library, `gcd` function used for calculating the greatest common divisor of two numbers.
Documentation: math module

- **pyperclip**: Third-party library, used for accessing the system clipboard (e.g., copying and pasting text).
Source: pyperclip on PyPI

- **string**: Python standard library, used for common string operations (e.g., generating random passwords).
Documentation: string module

- **random**: Python standard library, used for generating random numbers and selecting random elements.
Documentation: random module

- **tkinter.simpledialog**: Python standard library, provides dialogs like `askstring` for input retrieval.
Documentation: simpledialog