Password Manager

Theodore Contes and Connor Waters

**Introduction**

Good passwords are a critical part of keeping information secure in the current day, but maintaining good passwords is becoming more difficult every day. Not only do you need long, complex passwords to defeat standard attacks, such as dictionary attacks or brute force methods, but you need multiple such passwords. Password reuse leaves oneself open to having multiple accounts and services vulnerable if any password is compromised. But the alternative is to memorize many passwords, and then memorize new ones when you need to change those passwords. For many, this is not a practical solution, and so they instead leave themselves open to attacks through weak passwords and password reuse.

This project aims to provide a solution to the problem, by allowing a user to memorize a single strong password, and then store the rest of their password information behind this password. The password manager will create and store passwords in encrypted files, allowing the user to access them but not others.

Password managers do have some weaknesses; inherently they rely on a single password to protect all of the other passwords, thus introducing a single point of failure. If the single password is compromised then all of the passwords and accounts stored within the manager are at risk. Alternatively, if the access to the password manager is lost, then the user has just lost access to all of their accounts. This can be mitigated through backups to an extent, though storing the master password does expose it to risk. Finally, password managers cannot be used for all passwords, if only because they are stored on computers: to access the password manager you must first be able to access the computer it is on. Thus, while they greatly reduce the number of passwords needed, users are still required to know more than 1 strong password for good protection (one for the password manager, one per computer used).

**Related Work**

The concept of a password manager is not new. There are many such programs offered, and they come with a vast array of useful features if you are willing to pay the price. As many of these services offer data backup, these costs are generally subscription based rather than one time.

We are using the PyCrypto library, which implements the hash functions an AES encryption used for this project.

**Design** (optional)

The architecture design of the project (if applicable).

**Project Description**

This project will allow the user to store passwords in an encrypted file. This file is encrypted using AES 256 bit encryption. The key for this file is a password the user selects which is then hashed using SHA256 to create the key. As usual, the rules for strong password selection are important for this: using password or other weak passwords means that the encryption will be easily circumvented. The specific AES encryption uses Cipher-Block Chaining to prevent the encrypted file from cryptanalysis. Thus, the encryption of each block depends on all previous blocks.

The program allows the user to

Describe the major functionalities of your program.

Guide the users on how to use your program to perform certain tasks.