Usage:

To run: (Python 2 will give error, use Python 3)

python3 pwManager.py www.google.com

To reset:

rm passwords

Our project uses the provided python example code and adapts it to use an AES cipher to encrypt and decrypt the dictionary of passwords for the user.

* A master password is provided when first ran. This master password is used as the input for a key created using PBKDF2 along with the use of a salted string value.
* The program checks for an existing passwords file, if it does not exist, one is created, as well as an empty dictionary.
* There is then a check in place to ensure the user specified a website while running the program, which the program will either return the password for, or prompt the user for a new password if the website does not exist in the passwords file yet.
* The encrypt and decrypt functions both use AES cipher encryption with the key created at the start, as well as the AES mode AEX, and no nonce to be able to convert to and from plaintext. These cryptographic functions return as tag as well, which is stored in the password file, and used to validate if the data converted from the file is accurate or not.

Addition comments are provided in the source code, which is tested and running on the UNT CSCE machines.

Examples of the code being used are included on page two of this report.

(**NOTE**: The dictionary being printed after the “The message is authentic” message was only used to demonstrate previous passwords were being saved for debugging and has been removed in the current version.)

