Documentation

Group 101 members: Anna Kafrune; Jessica Vo; Kevin Johansen; Joseph Candella

Explanation of the purpose of each file in the repository:

- 100000_regularly_used_passwords_breached.txt → This is a text file that consists of the 100,000 regularly used passwords, which is used when the program compares the suggestion that the user inputs and this text file and lets the user know if the suggestion will be used or not. If the suggestion is on this text file, then the suggestion will be denied.
- password_generator.py → This is the main script for the program. This file will generate a random password based on the users suggestion. The user will also be able to store the newly created password in a document and reset their password if needed.
- test_password_generator.py → This is the pytest test script for the password_generator.py file. This tests three functions that were created in the password_generator.py file.

Instructions on how to run your program from the command line and on how to use the program and/or interpret the output of the program, as applicable:

1. In the command line, enter: python password_generator X or python3 password_generator X
    a. Note: X is the user's suggestion of a sequence of characters that he/she wishes to be in the generated password.
    b. Example of input 1: python password_generator genpass --- this suggestion will not be denied since it is not in the breached passwords file
    c. Example of input 2: python password_generator password --- this suggestion will be denied since it is in the breached passwords file
    d. Note: The suggestion given by the user will be included into the password and a message saying "Your suggestion passed. It will be used when generating your password" will be displayed if the suggestion is not found within the list of 100,000 most commonly hacked passwords. If it is found, a message saying "Exact password found in the regularly used password file" or "This is part of a regularly used password" is going to show up, refusing the suggestion for either being the whole or part of a password that is regularly used.
2. A menu with a series of options of requirements will appear and one at a time, the user needs to pick which ones he/she wants for the password. These choices allow the user to select how they want their password to be.
    2.1.1 Entering the number 1 will bring another question about the full length desired for the password (a number is needed to be entered here). It is recommended

the user starts with the password length requirement (1) and is encouraged to choose a password length longer than their suggestion.

a. Note: If the length inputted is less than the sum of the number of requirements with the length of the suggestion, your password will not be generated, for a greater length would be needed to fulfill all of the user's requests.

2.1.2 Not entering the number 1 at all will result in the password length be the length of the suggestion, and no more requirements will be able to be part of the password.

a. Note: Not choosing 1 and choosing other requirement numbers will result in your password not being generated, for a greater length would be needed to fulfill all of the user's requests.

2.2 Entering the number 2 will include numbers in the password.

2.3 Entering the number 3 will include letters in the password.

2.4 Entering the number 4 will include number, letters, and symbols in the password.

2.5 Entering the number 5 will leave the requirements section and move on to the next step.

3. The user will be prompted once again, with this prompt being for 3 different levels of password personalization. Once this is complete, the generated password will be returned to the user.

3.1 Entering the number 1 will have the user's suggestion stay unbroken within the password.

3.2 Entering the number 2 will have the user's suggestion be broken apart into groups of random length.

3.3 Entering the number 3 will have the user's suggestion be broken apart into groups containing a single character.

4. The user will be asked if they would like to save their password. If the user wants to save their password to keep for their records, they will input Y for yes, otherwise N for no. If Y is selected, the user is asked to input the username or email they want the password to be saved with. Once this is done, the password and username/email will be saved to a text file titled pwdmanager.txt in the same directory as the program file.

5. The user will then be prompted if they would like to see their other passwords, they will input Y for yes, otherwise N for no. It is recommended that they put Y for yes.

5.1 If Y is selected, then the user will be prompted to input a username and password (keep in mind that the first username and password created and saved will be the ones that should be entered here to access other accounts' information saved). If the username and password is correct, the program will print "Access permitted". Then, the user will be asked to input the account name, then the program will print the account's username and

password for that account.

5.2 If either the username or password is incorrect, the program will print a message saying

"The username or password is incorrect. Please try again." and will once again prompt the

user for a username and password. Once the username or password is continuously

inputted incorrectly 3 times, it will print "Access denied. Too many wrong attempts."

Annotated bibliography and explanation of how each source was used:

- https://www.geeksforgeeks.org/python-string-ascii_lowercase/

  Used to learn how to use the string library in the generation of letters and symbols. The information learned was implemented in the functions let() and num_let_sym().

- https://www.ncsc.gov.uk/news/most-hacked-passwords-revealed-as-uk-cyber-survey-exposes-gaps-in-online-security

  Used as a source of data to create the text file 100000_regularly_used_passwords_breached.txt with the 100,000 most regularly used passwords.