Random Password Generator/Final Project

CYSE 250

[Sebastian Stovall, Reed Wilhelm, William Van Opstal]

[Fall 2022]

[Dec 5, 2022]

Old Dominion University

Problem Statement

In today's age of close to all important information being stored online, an estimated 81% of all data breaches are due to poor password security. (21 must know password statistics). In an idea world, users would always create complex and secure passwords, however many users often can't even be bothered to come up with a complex password and as such, often use predictable or easy to guess passwords. It is evident that simplifying the process of creating a complex password would lead to more people using better, more secure passwords. To help with that issue, our group has created a random password generator that can be used by anyone to create a complex password, using letters, or pin, using numbers, that can be of any length.

Details

The program was created using PyCharm version 2022.2, free downloadable Python programming software. The program was created on a 2022 Apple MacBook Air, M2 chip model. The MacBook features an 8-core CPU and GPU, along with a 16-core neural engine.(MacBook Air with M2 chip-tech specs).

Discussion of Results

Once the script is run, the program prompts the user to choose whether the user want to create a password or pin. The user is then asked how long they need the password or pin to be. The machine will then generate a random collection of letters or numbers to serve as a new password. The user is then told to type 'Yes' if they would like to generate a new password or pin. If they type 'yes', the loop starts over, if the user types anything else, this will break the loop.

Conclusion

Overall, the need for more users to create complex passwords is very high. With the high percentage of data leaks coming from low password strength, it is pivotal that this process be made as easy as possible, being as a lot of companies still don't require strong passwords to make sign up processes easier on users. As such the creation of the random password generator

will help to make it easier for users to create complex passwords to better their cybersecurity hygiene

Appendix

Final Program

Final addition

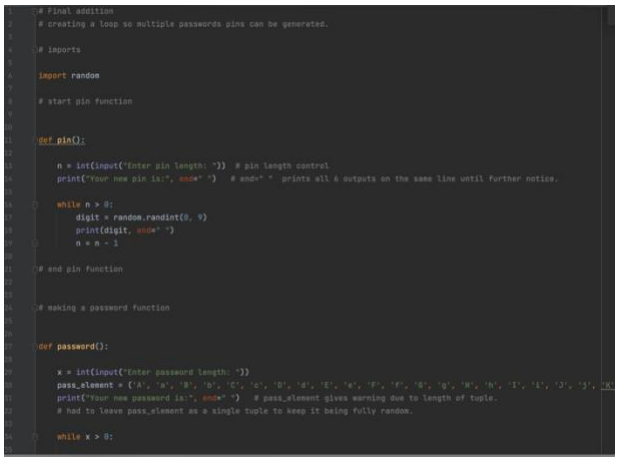
```
# creating a loop so multiple passwords pins can be generated.

# imports

import random

# start pin function



def pin():

    n = int(input("Enter pin length: "))  # pin length control
    print("Your new pin is:", end=" ")   # end=" "  prints all 6 outputs on the same line until further notice.

    while n > 0:
        digit = random.randint(0, 9)
        print(digit, end=" ")
        n = n - 1

# end pin function
```

```
# making a password function


def password():

    x = int(input("Enter password length: "))
    pass_element = ('A', 'a', 'B', 'b', 'C', 'c', 'D', 'd', 'E', 'e', 'F', 'f', 'G', 'g', 'H', 'h', 'I', 'i',
'J', 'j', 'K', 'k', 'L', 'l', 'M', 'm', 'N', 'n', 'O', 'o', 'P', 'p', 'Q', 'q', 'R', 'r', 'S', 's', 'T', 't', 'U',
'u', 'V', 'v', 'W', 'w', 'X', 'x', 'Y', 'y', 'Z', 'z')
    print("Your new password is:", end=" ")   # pass_element gives warning due to
length of tuple.
    # had to leave pass_element as a single tuple to keep it being fully random.

    while x > 0:

        control = random.randint(0, 51)
        print(pass_element[control], end="")
        x = x - 1

# end password function


# start of loop

user_response = "yes"   # Control to start program
```

```
# Start of loop

while user_response == 'yes':

    print("Type 'pin' for a new pin, or 'password' for a new password.")
    pass_type = input(str(":"))

    if pass_type == 'pin':   # Calling pin function.
        pin()

    if pass_type == 'password':   # Calling password function.
        password()

    print("\nIf you would like to generate another password or pin type 'yes'")   #
Option to run program again.
    user_response = input(str(":"))
# end of while loop

if pass_type == 'pin':   # pass_type will show as a warning here because its
undefined until the code runs
    print("Enjoy your new pin")

if pass_type == 'password':
    print("Enjoy your new password")
```

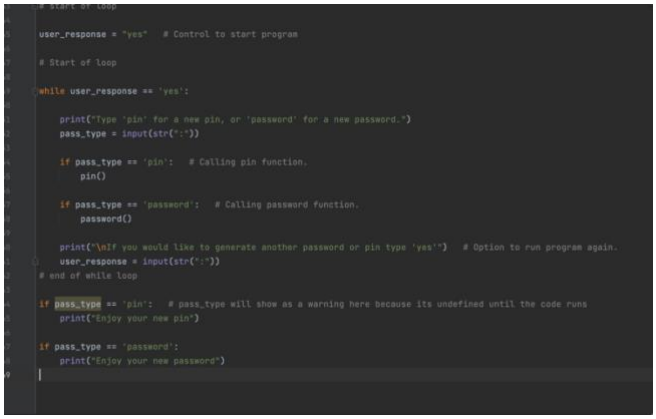
Images

Script



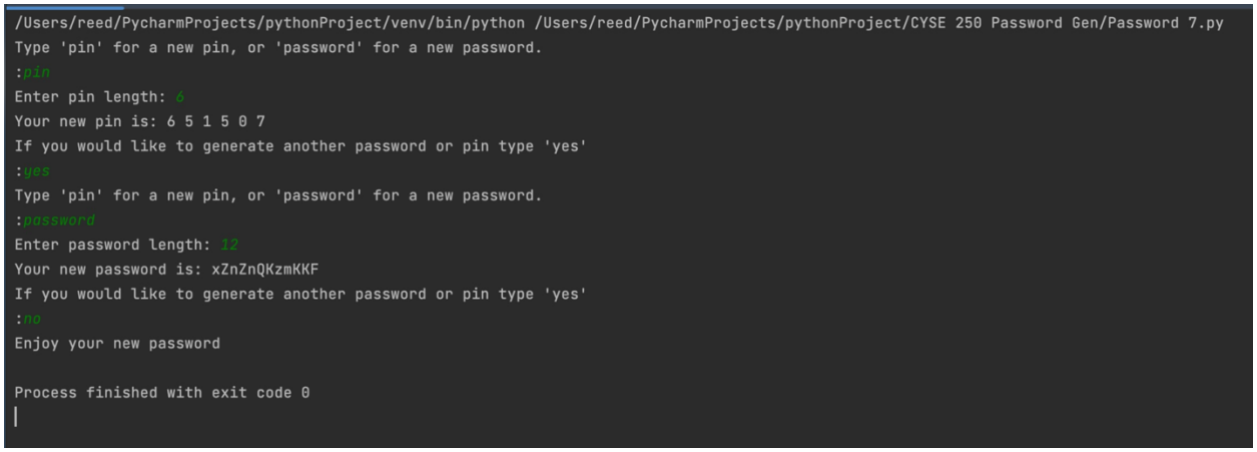
```
# Final addition
# creating a loop so multiple passwords pins can be generated.

# imports

import random

# start pin function


def pin():

    n = int(input("Enter pin length: "))  # pin length control
    print("Your new pin is:", end=" ")  # end=" " prints all 6 outputs on the same line until further notice.

    while n > 0:
        digit = random.randint(0, 9)
        print(digit, end=" ")
        n = n - 1

# end pin function


# making a password function


def password():

    x = int(input("Enter password length: "))
    pass_element = ['A', 'a', 'B', 'b', 'C', 'c', 'D', 'd', 'E', 'e', 'F', 'f', 'G', 'g', 'H', 'h', 'I', 'i', 'J', 'j', 'K
    print("Your new password is:", end=" ")  # pass_element gives warning due to length of tuple.
    # had to leave pass_element as a single tuple to keep it being fully random.

    while x > 0:
```

```
def password():

    x = int(input("Enter password length: "))
    pass_element = ['A', 'a', 'B', 'b', 'C', 'c', 'D', 'd', 'E', 'e', 'F', 'f', 'G', 'g', 'H', 'h', 'I', 'i', 'J', 'j',
    print("Your new password is:", end=" ")  # pass_element gives warning due to length of tuple.
    # had to leave pass_element as a single tuple to keep it being fully random.

    while x > 0:

        control = random.randint(0, 51)
        print(pass_element[control], end="")
        x = x - 1

# end password function


# start of loop

user_response = "yes"  # Control to start program

# Start of loop

while user_response == 'yes':

    print("Type 'pin' for a new pin, or 'password' for a new password.")
    pass_type = input(str(":"))

    if pass_type == 'pin':  # Calling pin function.
        pin()
```



```
# start of loop

user_response = "yes"  # Control to start program

# Start of loop

while user_response == 'yes':

    print("Type 'pin' for a new pin, or 'password' for a new password.")
    pass_type = input(str(":"))

    if pass_type == 'pin':  # Calling pin function.
        pin()

    if pass_type == 'password':  # Calling password function.
        password()

    print("\nIf you would like to generate another password or pin type 'yes'")  # Option to run program again.
    user_response = input(str(":"))
# end of while loop

if pass_type == 'pin':  # pass_type will show as a warning here because its undefined until the code runs
    print("Enjoy your new pin")

if pass_type == 'password':
    print("Enjoy your new password")
```

Output



References

21 must-know weak password statistics for Utmost Security. KommandoTech. (n.d.). Retrieved December 8, 2022, from https://kommandotech.com/statistics/weak-password-statistics/#:~:text=An%20estimated%2081%25%20of%20data,due%20to%20poor%20password%20security.

MacBook Air with M2 chip - tech specs. Apple. (n.d.). Retrieved December 8, 2022, from https://www.apple.com/macbook-air-m2/specs/