Jawuan Hart and Joseph DiGennaro
Professor Shobba Vatsa
CYSE 250 Programming
05 December 2021

Password Protection

An Overview of the Client:

This client sells many different types of products to customers at a discounted price, as they sell products wholesale to their customers. They have eight employees at the office. They have a simple LAN network that they use. Typically, they manage their website, and send out email blasts to customers about their products, and current sales and promotions.

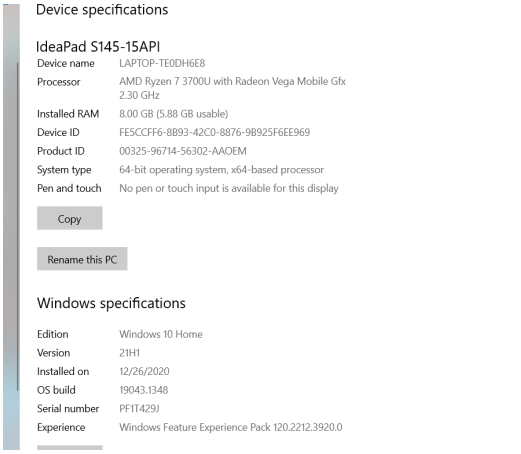
The Problem:

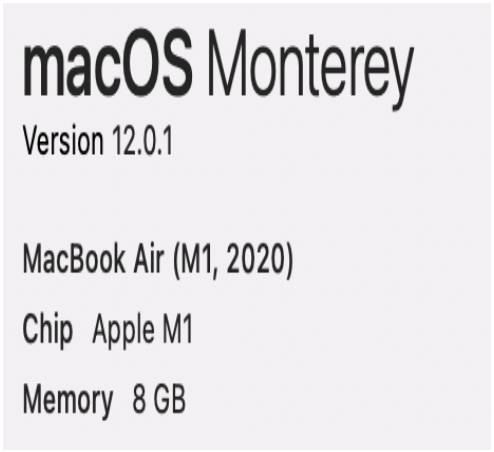
The client has experienced frequent breaches in their employees email and related work accounts. This led to many spam emails being sent out by the company while they were breached by hackers. The hackers sent out emails while inside the company's email servers, after they had acquired sensitive customer information. The emails that were sent out to the client's customers were sent as a phishing email that offered a really good price on one of the client's products, that when bought by the customer via the email link, the customer's credit card and address information was captured by the hackers. This led the client to contact a cybersecurity specialist who could help their employees close the open door of poor password management and construction.

Constructing a Solution:

The company wanted a very simple solution to allow their employees to create a stronger password without such a task being a burden to them. We eventually decided on a simple program that would evaluate password strength, through a series of if/then statements to filter out the weak passwords, and help employees make stronger passwords. The password needed to have at least one uppercase letter, an uppercase letter, a special character or symbol, and the overall length should be at least 10 characters long for a strong password. Previously, employees were making very simple passwords, as they had to frequently sign back into their accounts after minimal activity for security purposes. The simple passwords that they made seemed easier, but they actually opened the door to hackers.

Hardware and Software:





Jawuan's Setup

Joseph's Setup

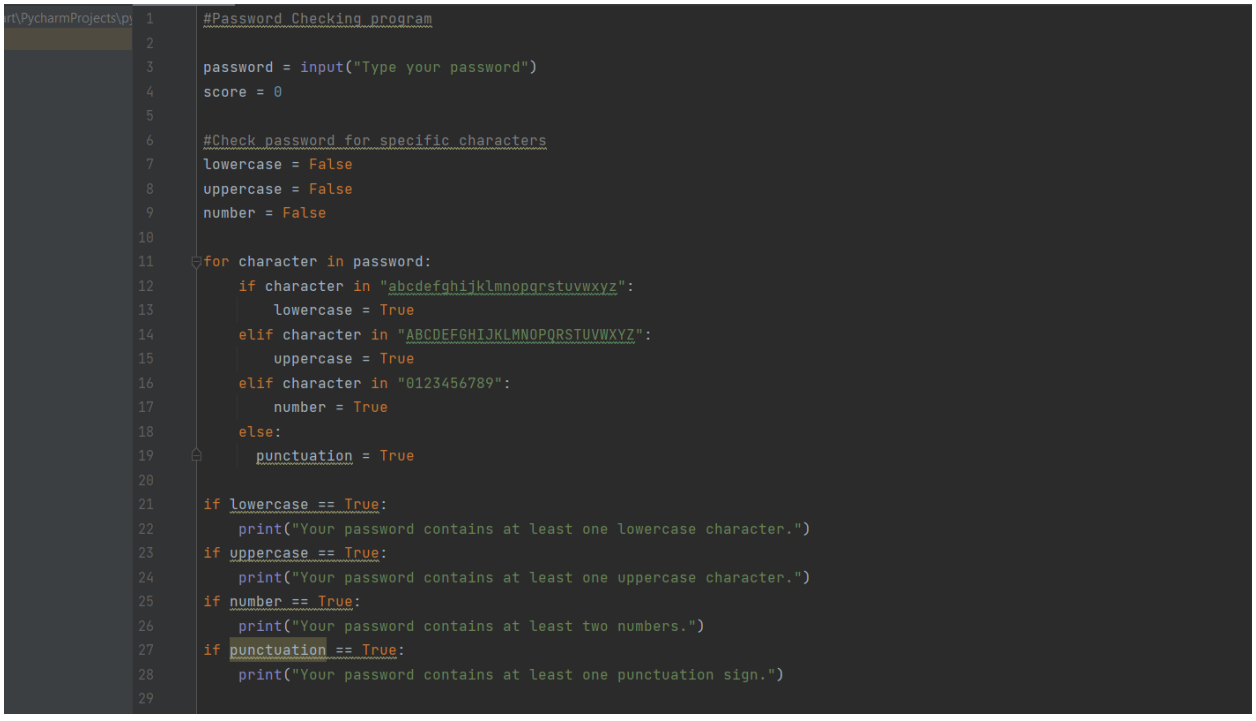
Having properly working computers is essential to have when in the cybersecurity field. It would be foolish to use antiquated equipment, as this could create another open door for cybersecurity issues, especially when working on the systems of other companies. Jawuan is currently using a Microsoft Windows operating system. The computer is a Lenovo Ideapad that has the AMD Ryzen 3700U processor, paired with eight gigabytes of memory for quick data processing. It is also important to note that a laptop is very convenient for this type of field work, since it is essentially a mobile workstation. The version it is operating on is 21H1 for the windows based system. Joseph is also using a laptop. It is the Apple Macbook Air 2020 model. The current version for Apple Mac OSx is 12.0.1 for the Macbook Air. Having an up to date computer is very important, as it can close doors as a result of installing patching and eliminating security vulnerabilities found in older versions of the software. The laptop has the Apple M1 processor with eight gigabytes of ram for efficient productivity. This hardware and software from both computers allowed for collaboration, and minimal lag, as the computers are up to date, and ready for work.

Constructing a Plan:

For the construction of the password program, python was the language used. We needed to use an editor that was cross platform, meaning that it would work with Jawuan's windows PC, and Joseph's Apple laptop. PyCharm allows for multi-user collaboration, and it can also be installed on many different types of operating systems. This editor also allows for customization of code and has many features that makes coding easier to work with. The program needed a way to rate the user's password to see how it would stand up in the real world, where hackers could easily steal a poorly made password. The program will tell the user how strong their password is based upon how many requirements they meet. Positive points are given to the user depending on how strong their password is. For example, if the user's password has a lower and/or uppercase letter present in their password, they get an increase in points to show that their password is stronger than if it did not have those elements. Further increases in password strength can be achieved through adding a punctuation sign. Also, having an overall password length of 10 characters or more can also strengthen the password, therefore, the user gets a higher score on their password. The higher the score, the stronger the password is. The stronger the password, the more protected the client is from security breaches, which protects customers from their sensitive information being accessed, and closing the door of vulnerabilities.

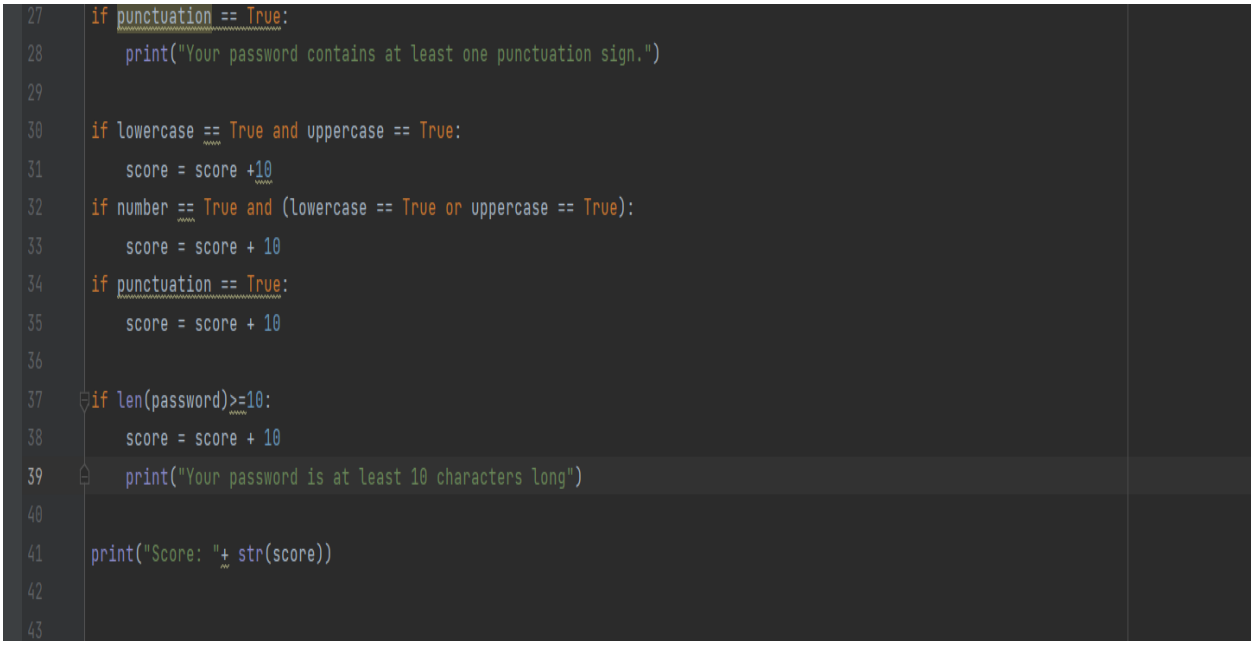
Writing the Code:

Coding the program with simplicity in mind can eliminate repeated strings, and make it easier for the user to use. When first using the program, the user is prompted to enter the password that they want to use for their workstation, or cloud services. Keeping simplicity in mind, setting the user password score to zero allows for a clean foundation. Then, declaring the values and the if statements set up the algorithm that will determine how strong the password is.



```
#Password Checking program

password = input("Type your password")
score = 0

#Check password for specific characters
lowercase = False
uppercase = False
number = False

for character in password:
    if character in "abcdefghijklmnopqrstuvwxyz":
        lowercase = True
    elif character in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
        uppercase = True
    elif character in "0123456789":
        number = True
    else:
      punctuation = True

if lowercase == True:
    print("Your password contains at least one lowercase character.")
if uppercase == True:
    print("Your password contains at least one uppercase character.")
if number == True:
    print("Your password contains at least two numbers.")
if punctuation == True:
    print("Your password contains at least one punctuation sign.")
```

As the password filters through the if and elif statements, a higher score can be added to the password to show its strength. As seen in the image below, the user is presented with the score of the password. The client can then determine what score they want used for their applications. However, we always recommend using the strongest password possible to close any possible doors.

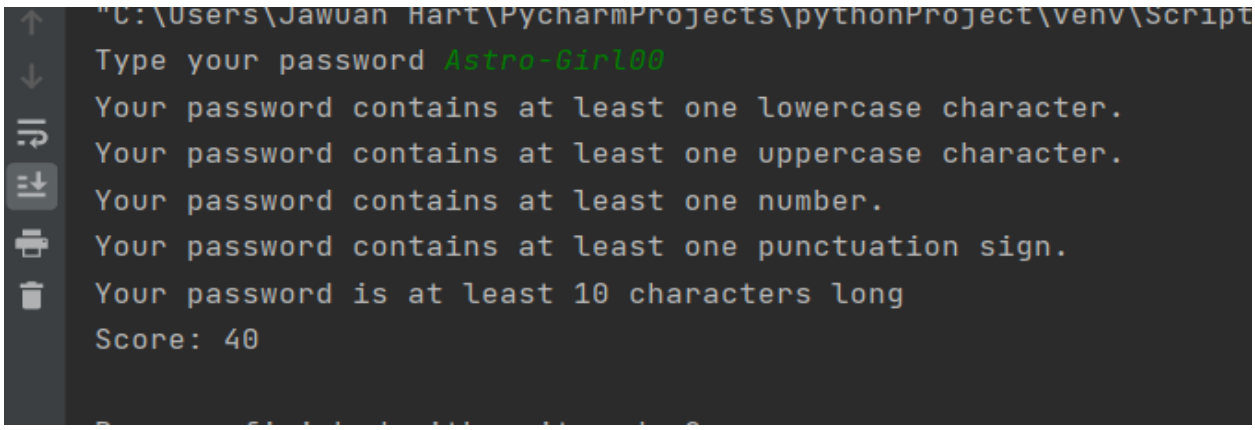


```
if punctuation == True:
    print("Your password contains at least one punctuation sign.")

if lowercase == True and uppercase == True:
    score = score +10
if number == True and (lowercase == True or uppercase == True):
    score = score + 10
if punctuation == True:
    score = score + 10

if len(password)>=10:
    score = score + 10
    print("Your password is at least 10 characters long")

print("Score: "+ str(score))
```

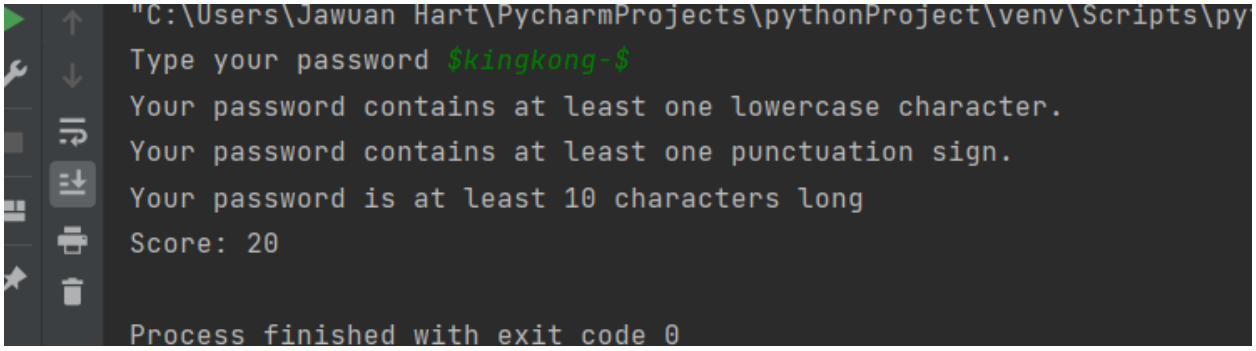
The Output and Conclusion:

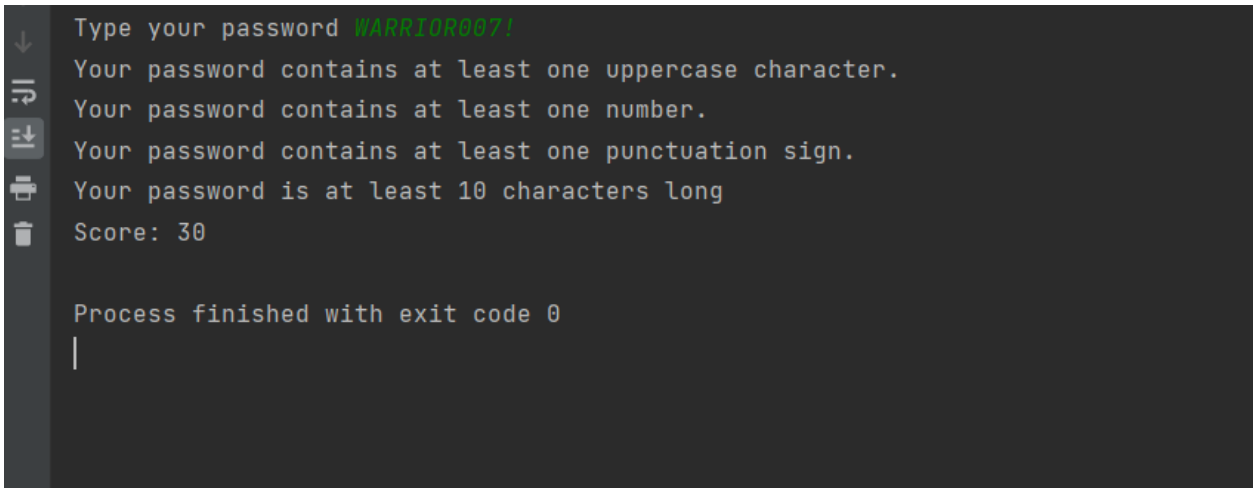
With this new program, the client should be able to stop using broken passwords that can cause open doors. With this program, these open doors are now closed; protecting customers and the client's reputation.

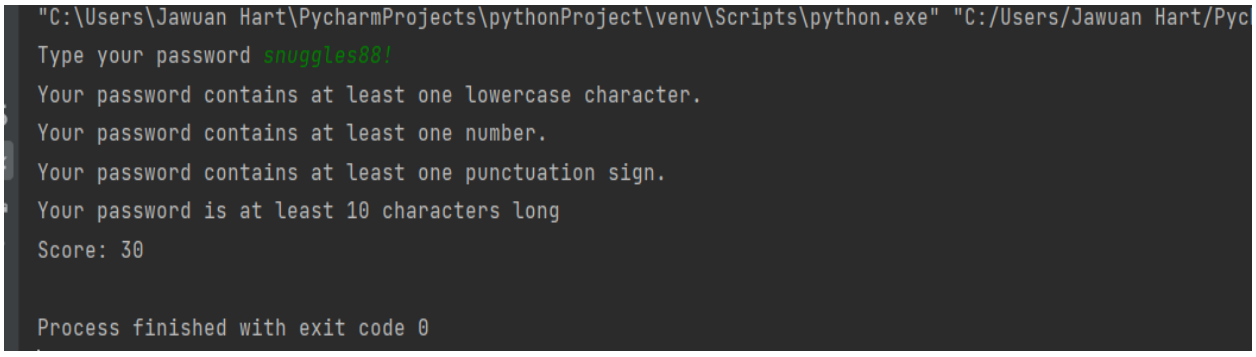
Examples of Code Outputs:



```
"C:\Users\Jawuan Hart\PycharmProjects\pythonProject\venv\Script
Type your password Astro-Girl00
Your password contains at least one lowercase character.
Your password contains at least one uppercase character.
Your password contains at least one number.
Your password contains at least one punctuation sign.
Your password is at least 10 characters long
Score: 40
```


"C:\Users\Jawuan Hart\PycharmProjects\pythonProject\venv\Scripts\py
Type your password $kingkong-$
Your password contains at least one lowercase character.
Your password contains at least one punctuation sign.
Your password is at least 10 characters long
Score: 20
Process finished with exit code 0


Type your password WARRIOR007!
Your password contains at least one uppercase character.
Your password contains at least one number.
Your password contains at least one punctuation sign.
Your password is at least 10 characters long
Score: 30
Process finished with exit code 0


"C:\Users\Jawuan Hart\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:/Users/Jawuan Hart/Pyc
Type your password snuggles88!
Your password contains at least one lowercase character.
Your password contains at least one number.
Your password contains at least one punctuation sign.
Your password is at least 10 characters long
Score: 30
Process finished with exit code 0

Python Code:

```
#Password Checking program

password = input("Type your password")
score = 0

#Check password for specific characters
lowercase = False
uppercase = False
number = False

for character in password:
    if character in "abcdefghijklmnopqrstuvwxyz":
        lowercase = True
    elif character in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
        uppercase = True
    elif character in "0123456789":
        number = True
    else:
      punctuation = True

if lowercase == True:
    print("Your password contains at least one lowercase character.")
if uppercase == True:
    print("Your password contains at least one uppercase character.")
if number == True:
    print("Your password contains at least two numbers.")
if punctuation == True:
    print("Your password contains at least one punctuation sign.")

if lowercase == True and uppercase == True:
    score = score +10
if number == True and (lowercase == True or uppercase == True):
    score = score + 10
if punctuation == True:
    score = score + 10

if len(password)>=10:
    score = score + 10
    print("Your password is at least 10 characters long")

print("Score: "+ str(score))
```

Image Used (Cover Page):

https://lh3.googleusercontent.com/proxy/RqhcIKWdaDbOjirrksyMJxFeOSN7SMgew0Xbld3CJdHNVMVpRH8kXwydKaRBOTxdC9wAXI0FsSWZwj9wJ0PS-Y0HkPNi5pgHpsaygCjWrA646GfY7JEI2Gw9R2VJopL_7uHgstX74il8zFUIZQ=w2880-h1438