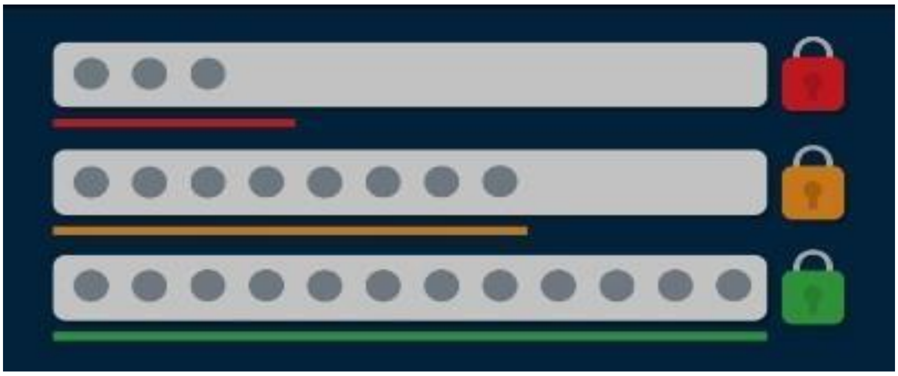
grow
intern

Internship report

Cybersecurity

Task1 : Test password strength



Produced by :

ASMAE EZZINE

11/07/2024

Table des matières

Introduction :

In today's digital age, ensuring the security of online accounts and sensitive information is of paramount importance. A key element in this security is the use of strong passwords. Weak passwords can be easily guessed or cracked, leading to unauthorized access and potential data breaches. Therefore, it is crucial for users to create passwords that are not only unique but also strong enough to withstand various types of attacks.

This project, the Password Strength Checker, aims to provide a user-friendly tool to help users assess the strength of their passwords. By evaluating key criteria such as length, the inclusion of uppercase and lowercase letters, digits, and special characters, the tool categorizes passwords as weak, medium, or strong. This real-time feedback encourages users to create stronger, more secure passwords.

I. The tools used :

To achieve this, the project leverages several tools and technologies:

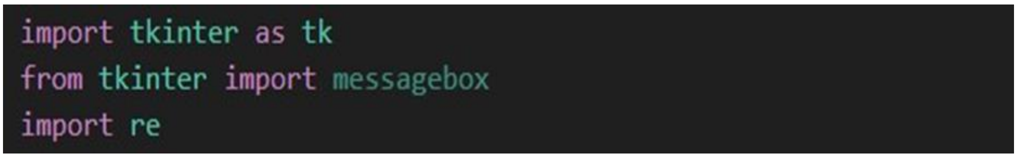
- **<u>Tkinter:</u>** This Python library is used to create the graphical user interface (GUI). It provides a simple yet effective way to build desktop applications with input fields, labels, and buttons.

- **<u>Regular Expressions (re module):</u>** Regular expressions are employed to check for the presence of different types of characters in the password. This ensures that the password meets the complexity requirements.

- **<u>Messagebox:</u>** Part of the Tkinter library, messagebox is used to display messages and alerts to the user, enhancing the interactivity of the application.

II. Code explanation :

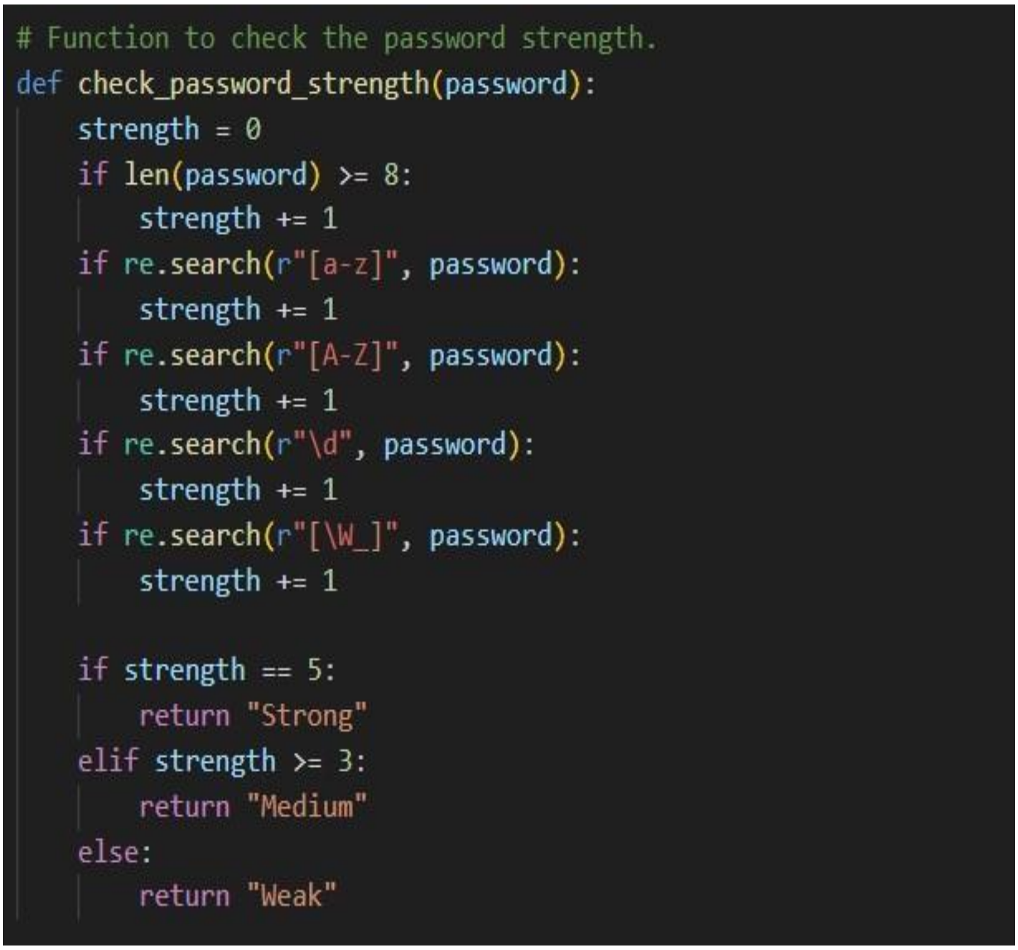
a. Importing Libraries :



```
import tkinter as tk
from tkinter import messagebox
import re
```

- re: Imported for regular expression operations to validate password criteria.
- tkinter: Imported to create the GUI.
- messagebox: Imported from Tkinter to show messages in dialog boxes.

b. Functions to Check Password Strength :



```
# Function to check the password strength.
def check_password_strength(password):
    strength = 0
    if len(password) >= 8:
        strength += 1
    if re.search(r"[a-z]", password):
        strength += 1
    if re.search(r"[A-Z]", password):
        strength += 1
    if re.search(r"\d", password):
        strength += 1
    if re.search(r"[\W_]", password):
        strength += 1

    if strength == 5:
        return "Strong"
    elif strength >= 3:
        return "Medium"
    else:
        return "Weak"
```

check_password_strength(password): This function checks the following criteria for a given password:

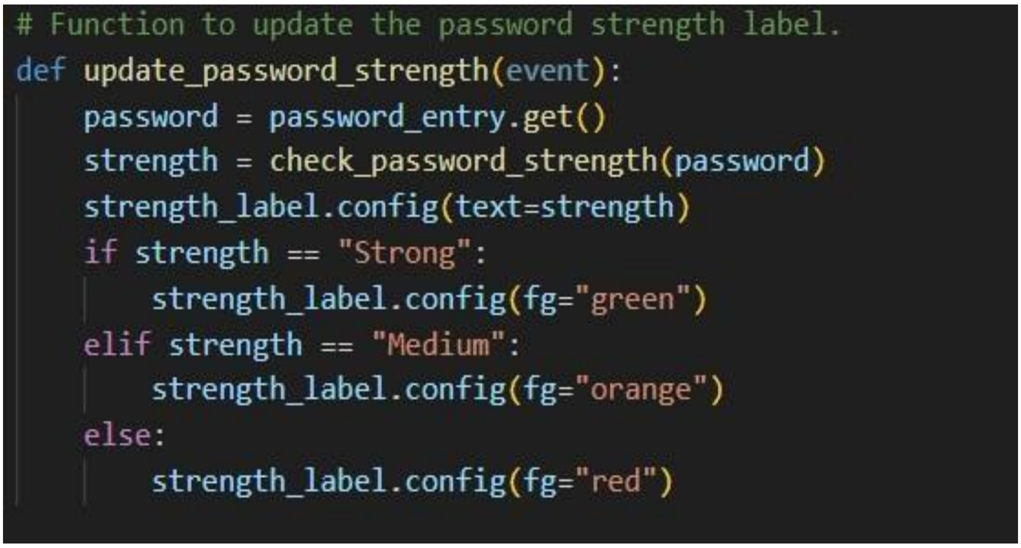
- **Length**: Must be at least 8 characters.
- **Lowercase Letter:** Must contain at least one lowercase letter.
- **Uppercase Letter:** Must contain at least one uppercase letter.
- **Digit:** Must contain at least one numeric digit.

- **Special Character:** Must contain at least one special character (!, @, #, $, etc.).

The function assigns a score based on the number of criteria met:

- 5 criteria met: "Strong"
- 3 or 4 criteria met: "Medium"
- Fewer than 3 criteria met: "Weak"

c. Function to Update Password Strength Label :



```python
# Function to update the password strength label.
def update_password_strength(event):
    password = password_entry.get()
    strength = check_password_strength(password)
    strength_label.config(text=strength)
    if strength == "Strong":
        strength_label.config(fg="green")
    elif strength == "Medium":
        strength_label.config(fg="orange")
    else:
        strength_label.config(fg="red")
```

update_password_strength(event): This function retrieves the password from the entry field, evaluates its strength using check_password_strength(password), and updates the strength_label accordingly:

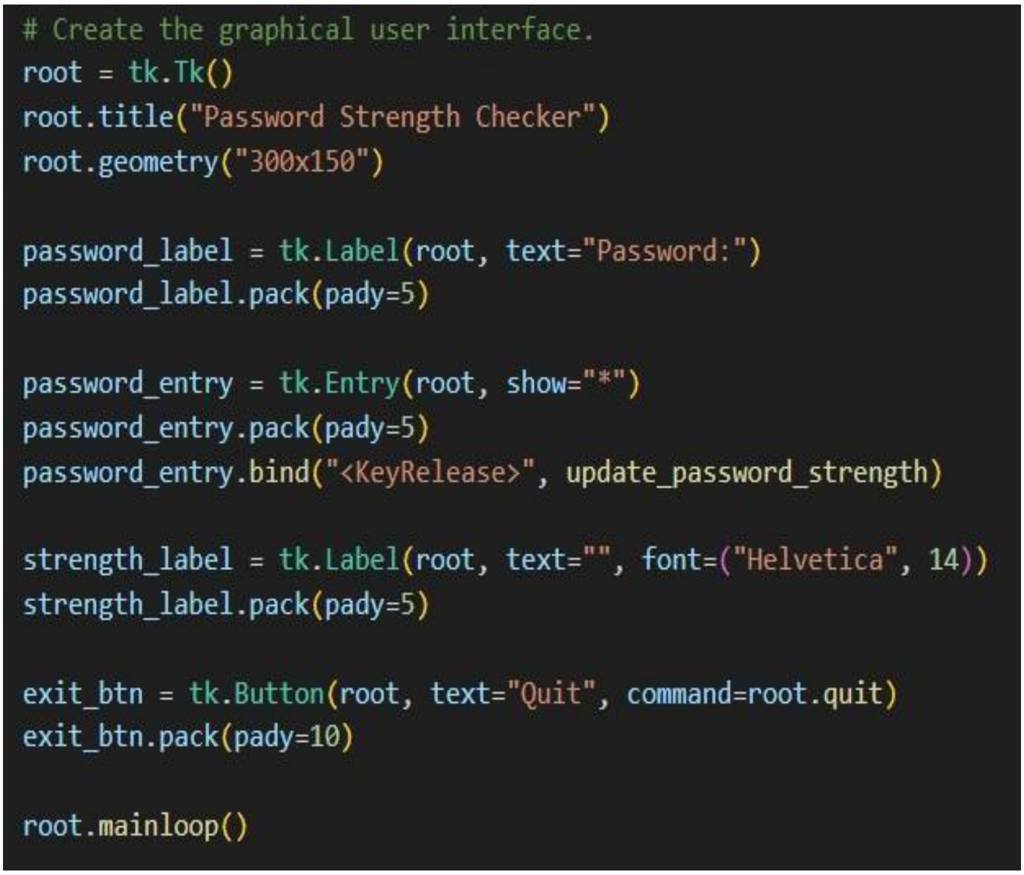
- **Password Retrieval:** Gets the password entered by the user.
- **Strength Evaluation:** Calls check_password_strength(password) to determine the strength of the password.
- **Label Update:** Updates the text and color of the strength_label based on the password strength:

➢ "Strong" passwords are shown in green.

➢ "Medium" passwords are shown in orange.

➢ "Weak" passwords are shown in red.

d. Creating the Graphical User Interface :

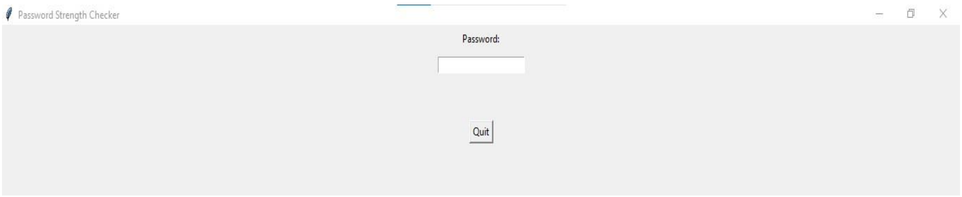


```python
# Create the graphical user interface.
root = tk.Tk()
root.title("Password Strength Checker")
root.geometry("300x150")

password_label = tk.Label(root, text="Password:")
password_label.pack(pady=5)

password_entry = tk.Entry(root, show="*")
password_entry.pack(pady=5)
password_entry.bind("<KeyRelease>", update_password_strength)

strength_label = tk.Label(root, text="", font=("Helvetica", 14))
strength_label.pack(pady=5)

exit_btn = tk.Button(root, text="Quit", command=root.quit)
exit_btn.pack(pady=10)

root.mainloop()
```

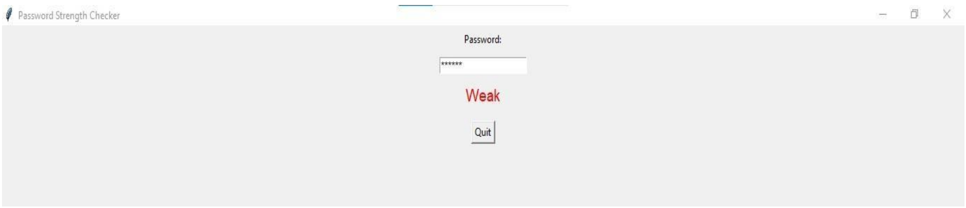
- **root = tk.Tk():** Creates the main window for the application.
- **root.title("Password Strength Checker"):** Sets the title of the main window.
- **root.geometry("300x150"):** Sets the size of the main window.
- **tk.Label():** Creates labels for the password entry and the strength message.
- **tk.Entry():** Creates an entry field where users can input their passwords. The show="*" parameter masks the password for privacy.
- **password_entry.bind("<KeyRelease>", update_password_strength):** Binds the key release event to the update_password_strength function, so the strength is updated as the user types.
- **strength_label:** Displays the strength of the password.
- **tk.Button():** Creates a button to quit the application.
- **root.mainloop():** Starts the Tkinter event loop to run the application.

III. Demonstration :

Users are prompted to enter their password into the text field provided :

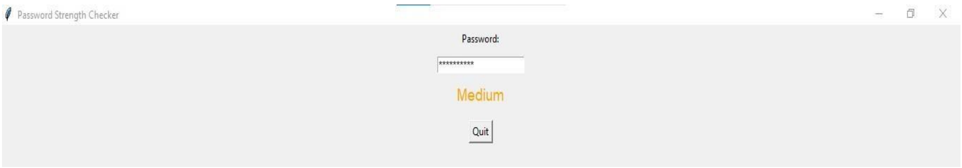


After entering a password, the application evaluates it and displays the strength as "Weak" in red text. This indicates that the password does not meet most of the security criteria :



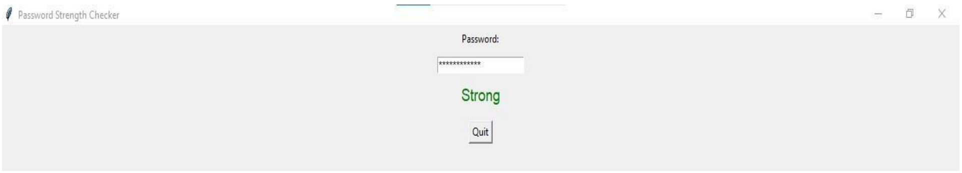
Weak

In this instance, the entered password is evaluated and deemed "Medium" strength, shown in orange text. This indicates that the password meets some, but not all, of the security requirements :



Medium

Here, the entered password is evaluated as "Strong," with the result displayed in green text. This indicates that the password meets all the essential security criteria, ensuring better protection :



Strong

Conclusion :

This Password Strength Checker project effectively demonstrates how to create a user-friendly tool to help users generate strong passwords. By utilizing the Tkinter library for the graphical user interface and regular expressions for password validation, the application provides immediate feedback on the strength of a password. This project showcases the integration of various tools and techniques to enhance password security, encouraging users to adopt better password practices.