Python Password Checker with zxcvbn Integration

Project Goals

- **Problem Statement:** People often use weak and easily guessable passwords, compromising the security of their online accounts. This tool aims to provide users with more accurate password strength assessments leveraging the zxcvbn library, helping them make informed choices about their passwords.
- **Target Audience:** Individuals concerned about online security, and potentially security educators who want to demonstrate password best practices.

Scope

- **Core Features:**
    - **User Input:** A mechanism for the user to enter a password.
    - **Strength Assessment using zxcvbn:** The zxcvbn library is used to estimate password strength realistically, providing a more nuanced evaluation than simple rule-based checkers.
    - **Feedback:** Clear visual or textual feedback to the user, indicating the password's strength (e.g., "Weak", "Moderate", "Strong") and suggestions for improvement (e.g., lengthen password, add character variety).
- **Potential Libraries:**
    - **zxcvbn:** A library for estimating password strength based on patterns found in real-world password leaks.

Outcomes and Deliverables

- **The Tool:** Well-structured and commented Python code implementing the password checker, particularly highlighting the integration of the zxcvbn library.
- **Demonstration:** A use-case with sample passwords showcasing the tool's output and its strength assessment capabilities.
- **Write-up:** A brief report on the value of using zxcvbn for password strength estimations, and how the tool's design decisions reflect that.

Timeline

- **Week 1-2:**

    - **Research:** Best practices in password strength evaluation, familiarization with the zxcvbn library.
    - **Core Algorithm Design:** Outline the logic for integrating zxcvbn assessments into the tool.
- **Week 3-4**
    - **Coding:** Develop the core password checking functionality and the integration of zxcvbn.
    - **Testing:** Use various input scenarios to ensure the tool works as intended and accurately reflects zxcvbn's capabilities.
- **Week 5-6**
    - **Interface:** (If applicable) Build a simple user interface (command-line or web-based).
    - **Documentation:** Write clear comments in your code and prepare the brief report.

Code Repository

- The Python Password Checker tool developed for this project can be found on GitHub at the following link: https://github.com/Bruce-Matrix/password-checker


File Actions Edit View Help
(root㉿kali4life)-[/home/forat/Desktop]
python3 password-checker.py
Choose an option:
1. Check the strength of a password
2. Generate a strong password
3. Quit
Enter your choice (1/2/3): 1
Enter your password: sdfgd
Password Strength: Weak
Suggestions for improvement:
- Password should be at least 8 characters long.
Choose an option:
1. Check the strength of a password
2. Generate a strong password
3. Quit
Enter your choice (1/2/3): 1
Enter your password: Tvhj$ebs5
Your password is MODERATE.
Choose an option:
1. Check the strength of a password
2. Generate a strong password
3. Quit
Enter your choice (1/2/3): 2
Enter the length of the password you want to generate: 10
Generated strong password: dSKc17l^1o
This is a strong password. Make sure not to share it with anyone!
(root㉿kali4life)-[/home/forat/Desktop]
python3 password-checker.py
Choose an option:
1. Check the strength of a password
2. Generate a strong password
3. Quit
Enter your choice (1/2/3): 1
Enter your password: dZ_Bv}qgb^p\Q8%
Your password is STRONG already. You don't need me.
(root㉿kali4life)-[/home/forat/Desktop]
#