Title:

Password Strength Analyzer with Custom Wordlist Generator

Introduction

Passwords play a key role in securing our online accounts, yet many people still rely on weak or predictable ones. This project addresses that issue by building a tool that evaluates how strong a password is and generates a personalized wordlist using information like a user's name, pet name, or birth year. The goal is to raise awareness about how easily personal data can be turned into potential passwords and encourage users to create stronger, more secure credentials.

Abstract

The project is divided into two main parts: analyzing password strength and creating a custom wordlist. The password analyzer checks for important factors such as

minimum length, a mix of characters (uppercase, lowercase, numbers, and special characters), and whether the password is commonly used. It also calculates the password's entropy to estimate how resistant it is to brute-force attacks.

The second part of the project generates a wordlist using user-provided details. It combines these inputs with common patterns like leetspeak (e.g., replacing "a" with "@") or appending years and symbols. The wordlist is saved in .txt format and can be used in educational or ethical hacking contexts to demonstrate how targeted password guessing works.

Tools Used

- Python – core language used for scripting
- Regex – to check character patterns in passwords
- Math library – for entropy calculation

- File Handling – to write the wordlist to a .txt file
  (*Optional: Tkinter for GUI, argparse for command-line interface*)

Steps Involved

1. Password Analysis

- The tool first ensures the password is at least 8 characters long.
- It checks if the password contains:
  - Uppercase and lowercase letters
  - Numbers

- Special characters

- It flags commonly used passwords like “123456” or “qwerty.”
- Then it calculates entropy (in bits) based on the variety and length of characters.
- Based on entropy, the tool labels the password as Weak, Moderate, or Strong.

2. Wordlist Generator

- The user is prompted to enter basic personal information like their name, pet’s name, or favorite number.
- The script uses this data to generate likely password combinations by:
  -

Adding suffixes like 123, @2023, or !

- Swapping characters (e.g., a → @, o → 0)
- Creating different casing variation.

Conclusion

This tool is a simple yet effective way to understand what makes a password strong — and how attackers might try to guess weak ones using personal information. It's educational, beginner-friendly, and demonstrates key cybersecurity concepts like brute-force vulnerability, entropy, and social engineering. With minor additions like a GUI or CLI, it can be further extended into a practical security training application.