Password Strength Analyzer with Custom Wordlist Generator

SHAIKH AQSA Z. A.

Date: 26-06-2025

—

🔹 Introduction

The very basics of digital security fall into password security. Weak, predictable passwords are usually the most frequent reasons behind data breach incidents. The study is focused on the development of a Python-based tool that can itself test the strength of any particular password and generate a custom wordlist by using personal information and some common patterns. The tool is intended to teach the user so-called password 'hygiene' and demonstrate to users that a targeted list of possible passwords can be crafted using even the most rudimentary information about a person.

—

🔹 Abstract

The Password Strength Analyzer and Custom Wordlist Generator is a very simple desktop utility meant to help users gauge the strength of passwords and to generate realistically crafted attack wordlists for teaching and ethical hacking purposes. The tool rates the strength of passwords using the zxcvbn library, and it employs tkinter to generate a user-friendly GUI. It takes as an input, for example, a name, date of birth, pet name, and so on, then converts it into leetspeak, uppercase or lowercase, and commonly used number suffixes, such as years. It outputs the wordlist into a .txt format. This type of project is good to add to one's knowledge base while learning cyber-security and GUIs made in Python.

—

🔹 Tools Used

- Python 3.x (Programming Language)
- zxcvbn (Password strength analysis)
- tkinter (Graphical User Interface)
- Thonny IDE (Beginner-friendly development environment)

—

- Steps Involved in Building the Project

1. Installed Python and required libraries (zxcvbn and tkinter).
2. Created a function to analyze password strength using zxcvbn.
3. Designed the GUI layout using tkinter with input fields and buttons.
4. Collected user data (name, date of birth, pet name) through the GUI.
5. Generated variations of the input data using:
    a. Lowercase, uppercase
    b. Leetspeak substitutions (e.g., a → @, s → $)
    c. Common suffixes (e.g., 2023, 123)
6. Removed duplicates and exported the wordlist to a .txt file using the Save As dialog.
7. Tested the application and refined the user interface.

—

- Conclusion

The project successfully shows the importance of strong-password creation and how attackers utilize very simple personal data in brute-force attempts. The tool is easy to operate, informative, and provides a realistic insight into how passwords are actually assessed and how weak credentials can be targeted. It caters to both a learner's environment and acts as a real tool for early-stage cybersecurity practice.