Introduction

Passwords remain one of the most widely used methods for securing digital accounts, yet weak or reused passwords continue to be a major cause of data breaches. This project focuses on building a user-friendly, Python-based tool that not only analyzes the strength of a password but also generates custom wordlists based on personal information. The goal is to raise awareness of how attackers can exploit weak or predictable passwords, and how user-specific data can be weaponized in password-cracking attempts. The application features a simple graphical interface to make it accessible to users with any technical background.

Abstract

This tool helps users understand the vulnerability of their passwords by measuring strength using the `zxcvbn` library—offering real-time feedback and crack time estimates. Additionally, it simulates how attackers might build targeted password guesses by generating a custom wordlist from personal details like name, date of birth, or pet names. The graphical user interface, built using Python's `tkinter` library, allows anyone to use the tool without needing command-line skills or coding experience.

Tools Used

- **Programming Language**: Python
- **Libraries**:
  - `zxcvbn` – For password strength analysis
  - `tkinter` – For building the GUI
- **Environment**: Terminal/Text Editor (Linux)
- **Output**: `custom_wordlist.txt` – Contains the generated wordlist

Steps Involved

1. **Password Strength Analyzer**

- Integrated the `zxcvbn` library to analyze password entropy, estimated crack time, and assign a strength score (0–4).
- Users input their password through the GUI, and strength feedback is displayed instantly, including suggestions for improvement.

2. **Custom Wordlist Generator**

- Collected basic personal information from the user: name, date of birth, and pet name.
- Applied techniques like:

  - Leetspeak substitutions (e.g., `a` → `@`, `e` → `3`)
  - Word reversals
  - Adding common numeric patterns like birth years
- All variations are saved to a file named `custom_wordlist.txt`, simulating how attackers build customized password dictionaries.

3. Graphical User Interface (GUI)

- Developed a clean, two-part interface using `tkinter`:
  - One section for analyzing password strength
  - Another for generating wordlists
- Used layout tools like frames, labels, input fields, and buttons for intuitive navigation and interaction.

Conclusion

This project offered hands-on experience in multiple areas: password security, attack simulation, and GUI development. It reinforced the reality that weak or predictable passwords—especially those based on personal information—can pose a serious security risk. By giving users an interactive way to test their own passwords and understand how attackers operate, this tool can be a powerful resource for both learning and awareness. The simple interface also makes it suitable for demonstrations or educating non-technical users on the importance of strong password practices.