Project Title: Password Strength Analyzer with Custom Wordlist Generator

Your Name: Abhyas Kathuria

Internship: Elevate Labs Cybersecurity Internship

Project Number: 4

Introduction This project involves building a Python-based tool that evaluates password strength and generates custom wordlists based on user-provided personal information. The objective is to demonstrate common password vulnerabilities, highlight the risks of weak and predictable passwords, and promote awareness of secure password practices. The tool is developed purely for educational purposes and aligns with ethical cybersecurity learning.

.

Abstract

The application performs two main functions:

1. **Password Strength Analysis –** Uses the zxcvbn library (used by Dropbox) to provide realistic strength scoring (0–4), estimated crack time, entropy calculation, warnings, and improvement suggestions.
2. **Custom Wordlist Generation** – Takes personal inputs (name, birth year, pet name, city, etc.), applies transformations such as leetspeak, capitalization, year/number appending, reversals, and pairwise combinations, and optionally includes common English words from NLTK. The generated wordlist is exported as a .txt file. The tool supports both Command Line Interface (CLI) and a user-friendly Tkinter Graphical User Interface (GUI).

Tools Used

- Python 3.x
- zxcvbn (for realistic strength scoring)
- NLTK (for common English words integration)
- Tkinter (for GUI)
- argparse, datetime, math (built-in)

Steps Involved in Building the Project

1. Set up Python environment and installed required libraries (zxcvbn, NLTK).
2. Implemented password analysis: Used zxcvbn for score/crack time/feedback + custom entropy formula.
3. Built wordlist generator: Added leetspeak variations, year/number appending, capitalization, reversals, and pairwise combinations.
4. Integrated optional NLTK common words for more realistic lists.
5. Added file export functionality and CLI with argparse.

6. Created Tkinter GUI for easy interaction (analysis + generation with file save dialog).
7. Tested thoroughly with weak/strong passwords and various inputs; verified outputs.

Conclusion

This project provided hands-on experience with real-world password security concepts, including dictionary attacks, pattern-based cracking, and entropy measurement. I gained deep understanding of why weak passwords are easily compromised and the importance of unpredictability. The tool successfully meets all deliverables: it evaluates password strength accurately and generates attack-specific wordlists. Limitations: Designed only for educational use and awareness – not for actual malicious activities. Future enhancements could include integration with larger public datasets or hashing demonstration. This project has significantly boosted my ability to explain password security topics confidently in technical interviews.