Password Strength Analyzer with Custom Wordlist Generator

Introduction

In today's cybersecurity landscape, password hygiene remains a critical defense against unauthorized access. This project aims to build a Python-based tool that analyzes password strength and generates targeted wordlists for security testing and educational purposes. It supports both CLI and GUI interfaces, making it accessible for penetration testers, educators, and learners alike.

Abstract

The tool evaluates password strength using the zxcvbn library, which estimates entropy and crack times based on real-world attack patterns. It then generates a custom wordlist derived from the password itself, incorporating leetspeak variants and common suffixes like years. This dual functionality helps users understand password vulnerabilities while producing realistic wordlists for ethical hacking and training simulations.

Tools Used

- **Python 3.x** – Core programming language

- **zxcvbn –** Password strength estimation

- **nltk** – Optional for future linguistic enhancements

- **Tkinter** – GUI interface

- **argparse –** CLI argument parsing

- **Standard File I/O –** For exporting .txt wordlists

Steps Involved in Building the Project

1. Password Analysis Module

- Integrated zxcvbn to evaluate password strength
- Extracted score, feedback, and crack time estimates

2. Wordlist Generator

- Parsed the password to generate base variants
- Applied leetspeak substitutions and appended common years
- Ensured uniqueness and sorted output

3. Export Functionality

- Saved generated wordlist in .txt format
- Compatible with password cracking tools like John the Ripper or Hashcat

4. User Interfaces

- CLI: Accepts password input and outputs analysis + wordlist
- GUI: Built with Tkinter for interactive use and file saving

5. Testing & Validation

- Verified crack time outputs across multiple password types
- Ensured wordlist generation logic was reproducible and extensible

Conclusion

This project bridges the gap between password education and practical cybersecurity tooling. By combining analysis and generation, it empowers users to understand password weaknesses and simulate realistic attack scenarios. Future enhancements may include entropy visualization, multilingual support, and integration with hash cracking workflows.