Password Strength Analyzer with Custom Wordlist Generator

Cybersecurity Tool with CLI and GUI

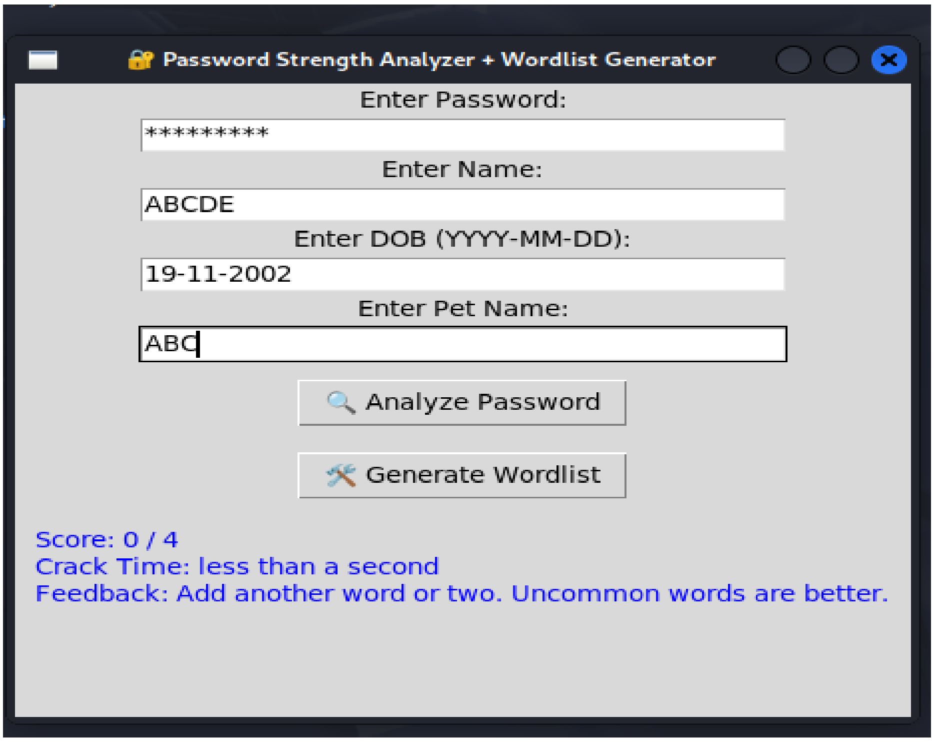
Submitted by:

Vidyasagar KM

Elevate Labs - Internship Program

This report presents a comprehensive overview of the design, implementation, and capabilities of a cybersecurity tool developed to assess the robustness of passwords and create tailored wordlists. These wordlists are specifically crafted to enhance the effectiveness of penetration testing activities by simulating realistic password attack scenarios

GUI Demonstration



Password Strength Analyzer + Wordlist Generator

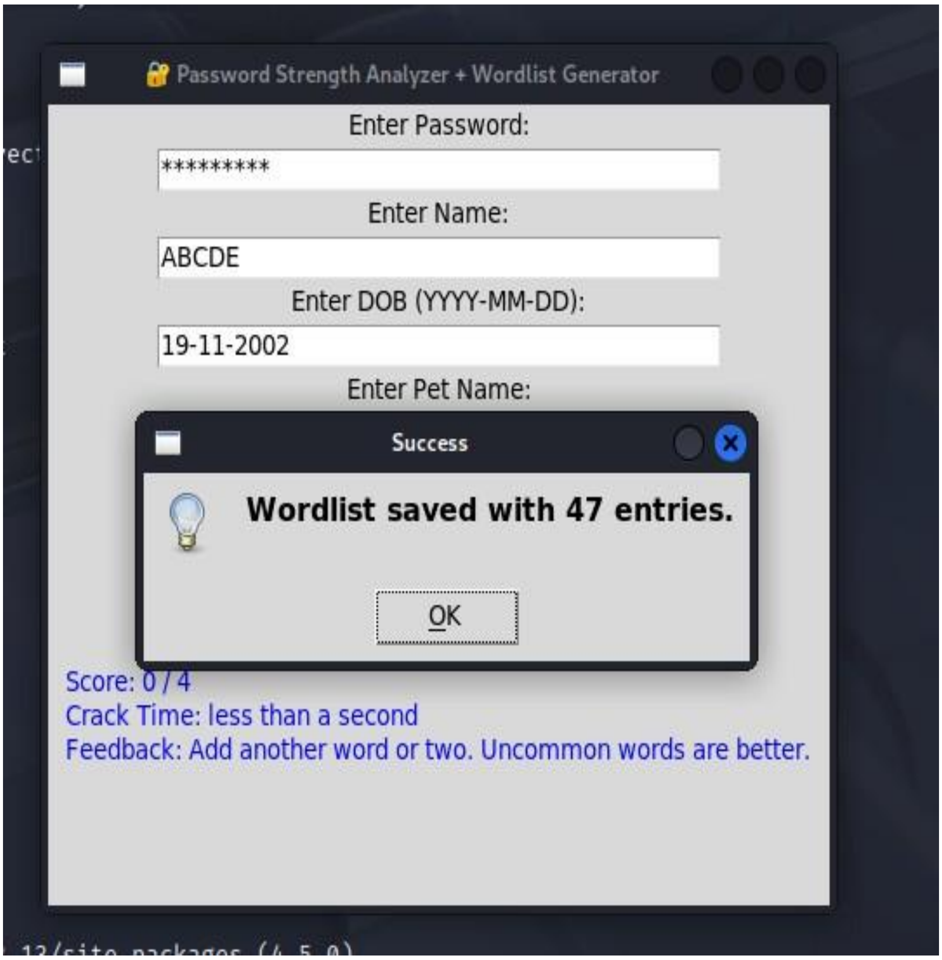
Enter Password:
****

Enter Name:
ABCDE

Enter DOB (YYYY-MM-DD):
19-11-2002

Enter Pet Name:
ABC

Score: 0 / 4
Crack Time: less than a second
Feedback: Add another word or two. Uncommon words are better.



Password Strength Analyzer + Wordlist Generator

Enter Password:
****

Enter Name:
ABCDE

Enter DOB (YYYY-MM-DD):
19-11-2002

Enter Pet Name:

Success

Wordlist saved with 47 entries.

Score: 0 / 4
Crack Time: less than a second
Feedback: Add another word or two. Uncommon words are better.

Password Strength Analyzer with Custom Wordlist Generator

Introduction

The Password Strength Analyzer with Custom Wordlist Generator is a Python-based tool designed to help users assess the strength of their passwords and generate targeted wordlists based on personal inputs. The tool is developed with a focus on cybersecurity, penetration testing, and password auditing.

Abstract

This project uses the zxcvbn algorithm to evaluate password strength and applies leetspeak, date patterns, and personal details like name and pet name to generate custom wordlists. The user can interact with the tool via a Command-Line Interface (CLI) or a Graphical User Interface (GUI) built using tkinter.

Tools Used

- Python 3.13
- zxcvbn-python (for password strength estimation)
- tkinter (for GUI)
- argparse (for CLI argument parsing)
- Custom functions for wordlist generation and mutations

Steps Involved in Building the Project

1. Setup Python virtual environment and install required libraries.
2. Build modular files: analyzer.py, generator.py, cli.py, gui.py.
3. Use zxcvbn to analyze password and display crack time & feedback.
4. Collect personal inputs (name, DOB, pet) and generate leetspeak & year-based wordlist.
5. Export results in .txt format for use in password cracking tools.
6. GUI built using tkinter for user-friendly interaction.

Conclusion

This project provided hands-on experience with password security concepts and Python scripting. By combining entropy-based analysis and targeted wordlist generation, the tool can support ethical hackers and forensic professionals during penetration testing or auditing. The dual interface (CLI + GUI) ensures flexibility for diverse users.