Password Strength Analyzer With Custom Wordlist Generator

Project Report

Sandra Sree Babu

Cyber Security Intern

Elevate Labs

October 2025

Contents

1 Abstract 2

2 Introduction 2

3 Tools Used 2

4 Steps Involved in the Project 3

5 Conclusion 3

1 Abstract

This project develops a Python-based tool that analyzes password strength and generates custom wordlists for authorized password-cracking simulations. It uses entropy calculations and the `zxcvbn` library to measure complexity and estimate crack times. Users can create targeted wordlists from personal data like names or dates, enhanced with leetspeak substitutions and year variations, and export them in `.txt` format for tools such as John the Ripper or Hashcat. The project promotes awareness of password weaknesses and supports cybersecurity professionals in building effective, customized password dictionaries.

2 Introduction

Passwords are one of the most common authentication methods, yet weak or predictable ones often lead to security breaches. Attackers exploit password reuse and personal information to gain access. The Password Strength Analyzer with Custom Wordlist Generator helps users assess password strength and enables cybersecurity professionals to create personalized wordlists for authorized testing. It combines strength analysis with intelligent wordlist generation using human patterns, entropy metrics, and linguistic variations, and includes both command-line and graphical interfaces for ease of use.

3 Tools Used

- **Python 3.x** – Core programming language used for implementation.
- **argparse** – For handling command-line arguments and user inputs.
- **Tkinter** – For building the graphical user interface (GUI).
- **zxcvbn** – Library used to evaluate password strength and crack times.
- **NLTK (Natural Language Toolkit)** – For optional word expansions and synonym generation.
- **OS & pathlib** – For file handling and exporting custom wordlists.
- **math & itertools** – For entropy calculations and generating pattern permutations.

4 Steps Involved in the Project

Requirement Gathering and Environment Setup

Defined objectives (password analysis, wordlist generation) and installed/verified required Python libraries (`zxcvbn-python`, `nltk`, etc.).

Password Strength Analysis Module

Used `zxcvbn` for scoring, entropy and crack-time estimates; added an entropy-based fallback if `zxcvbn` is unavailable.

Custom Wordlist Generation Module

Took personal inputs (names, dates, pet names), generated permutations, capitalization and year variants, applied leetspeak (e.g., `a → @`, `o → 0`), and optional `NLTK` synonym expansion.

Export and Output Management

Merged variants, removed duplicates, and exported the final wordlist as a `.txt` file for tools like John the Ripper or Hashcat.

Interface Development (CLI & GUI)

Provided an `argparse` CLI for advanced users and a `Tkinter` GUI for non-technical users.

Testing and Validation

Tested various passwords and personal-data combinations to confirm correct generation and expected variations.

5 Conclusion

The Password Strength Analyzer with Custom Wordlist Generator integrates password evaluation and intelligent wordlist creation in a single Python tool. It offers insights into password strength and supports ethical hacking through customized wordlists for authorized testing. Future enhancements may include multiprocessing for faster generation, cloud-based datasets, and a web-based interface for wider accessibility.