Password Strength Analyzer with Custom Wordlist Generator

Project Report

Prepared by: Bhanu Prakash Guda
Date: 27/10/2025

Objective: To develop a tool that analyzes password strength and generates custom wordlists for cybersecurity learning and password auditing.

1. Abstract

This project focuses on building a Password Strength Analyzer and Custom Wordlist Generator that helps users assess password robustness using entropy and the zxcvbn library. It generates customized wordlists based on user inputs like names, dates, and common patterns. The system aids cybersecurity learners and professionals in understanding password vulnerabilities and encourages the creation of strong, secure passwords.

2. Introduction

In the digital age, passwords remain the most common form of authentication, yet weak passwords lead to a majority of security breaches. The purpose of this project is to provide users with a simple but effective way to evaluate their password strength and understand common weaknesses. It also enables ethical penetration testers to generate realistic wordlists for password auditing under authorized scenarios.

3. Tools Used

1. Python – Core programming language used for developing the tool.
2. zxcvbn – Library for advanced password strength analysis.
3. argparse – For command-line interface implementation.
4. tkinter – Used to create the graphical user interface (GUI).
5. NLTK – For natural language processing of input data.
6. ReportLab – To generate project reports in PDF format.

4. Steps Involved in Building the Project

1. Step 1: Initialized the Python environment and installed required libraries such as zxcvbn, argparse, and tkinter.
2. Step 2: Developed the password strength analyzer using zxcvbn and added entropy fallback for unsupported cases.

3 Step 3: Collected user data inputs (e.g., name, pet, birthdate) for personalized wordlist generation.
4 Step 4: Implemented transformation logic including case variants, leetspeak substitutions, and appended years.
5 Step 5: Enabled exporting of generated wordlists in .txt format for use with password-cracking tools.
6 Step 6: Integrated both CLI and GUI interfaces for versatile user experience.
7 Step 7: Conducted comprehensive testing to ensure reliability and performance of the tool.

5. Conclusion

The Password Strength Analyzer and Custom Wordlist Generator effectively combine analysis and automation to educate users about password security. By offering both strength evaluation and targeted wordlist generation, the project bridges the gap between user awareness and ethical hacking practices. This tool can be extended further for enterprise security testing, awareness training, and academic research in cybersecurity.

End of Report