Password Strength Analyzer and Custom Wordlist Generator Project Report

1. Introduction

In the current landscape of cyber threats, weak passwords remain the single most exploited vulnerability, often leading to data breaches through techniques like Brute-force and Dictionary Attacks. This project, the Password Strength Analyzer and Custom Wordlist Generator, addresses this critical security gap. The tool is designed with two primary objectives: first, to enhance user awareness by providing objective analysis of password strength; and second, to demonstrate the dangers of using personal information in passwords by generating targeted, attack-specific wordlists.

2. Abstract

This tool is developed using Python 3.x and integrated with a Tkinter GUI for user-friendly interaction. The analysis module utilizes the sophisticated zxcvbn library, which employs entropy calculations based on various attack models (dictionaries, common patterns, and crack time estimation) to assign a strength score from 0 (weakest) to 4 (strongest). The core innovation lies in the Custom Wordlist Generator, which takes personal data (Name, Pet Name, Year) and generates variations using common Leet-speak substitutions (e.g., 'A' -> '4') and appends common suffixes (e.g., '123', '2024') to create a focused list of probable passwords, thereby simulating a highly effective password attack. The project serves as a practical demonstration of security vulnerabilities and defense strategies.

3. Tools Used

Category	Tool / Library	Purpose
Programming Language	Python 3.x	Core development language for building the tool.
Analysis Engine	zxcvbn	Advanced password entropy calculation, scoring(0-4), and cracking time estimation.
UI Framework	tkinter	Creation of the Graphics User Interface(GUI) with two tabbed sections.
Wordlist Logic	itertools	Efficient generation of leet-speak and general character combinations for the wordlist.

4. Steps Involved in Building the Project

- Environment and Library Setup: Installed Python 3.x and the necessary packages, primarily zxcvbn, itertools, and ensuring tkinter compatibility.

- Core Analysis Logic Development: Coded the analyze_password_strength function to accept a password, process it through zxcvbn, and return the score, crack time, and suggestions.
- Wordlist Generation Logic: Developed the generate_wordlist function. This involved defining Leet-speak rules, creating common appending lists (years, symbols), and using combination logic to efficiently produce thousands of probable passwords from minimal personal input.
- GUI Integration (Tkinter): Designed a robust GUI structure featuring two tabs:
- The Analyzer Tab handles password input and displays results (score, time, feedback).
- The Generator Tab provides input fields for personal data (Name, Pet, Year) and handles file saving.
- Error Handling and Refinement: Implemented basic validation checks (e.g., required input fields) and improved the overall user experience.

5. Conclusion

The Password Strength Analyzer and Custom Wordlist Generator is a successful implementation of practical cybersecurity principles. It effectively uses zxcvbn to educate users on strong password criteria and, more importantly, uses the Wordlist Generator to powerfully illustrate the danger of creating predictable passwords based on publicly available personal information. This project serves as a valuable educational resource, demonstrating that awareness of password attack vectors is the first step toward effective digital self-defense.