Password Strength Analyzer with Custom Wordlist Generator

Internship Project Report – June 2025

1. Introduction

Passwords are a common target for attackers using brute-force, dictionary, and social engineering techniques. This project combines a password strength analyzer with a custom wordlist generator to demonstrate password weaknesses and simulate how attackers might generate passwords using personal information.

2. Abstract

This tool was built using Python and focuses on two key components:
1. A password strength analyzer using the zxcvbn library, which provides a score from 0 to 4 along with suggestions and warnings.
2. A custom wordlist generator that takes user-specific inputs (like name, birth year, pet name, etc.) and creates a wordlist using common password patterns such as leetspeak, appended numbers, and combined strings.

The entire tool is command-line based and designed to simulate real-world password hygiene checks and password-based attack vectors.

3. Tools Used

- Language: Python 3
- Libraries:
  - zxcvbn – Password strength analysis
  - argparse – Command-line interface
  - itertools – Wordlist combinations
- Editor: Visual Studio Code
- Platform: Windows 10

4. Steps Involved in Building the Project

- Step 1: Set up the Python project with required packages (zxcvbn, argparse, etc.)
- Step 2: Implemented password strength analysis logic using zxcvbn and printed score, warnings, and suggestions.
- Step 3: Created a wordlist generator that takes name, pet name, birth year, and a favorite word and generates combinations, including leetspeak variations.
- Step 4: Combined both tools into a single Python script (main.py) using argparse for command-line options.
- Step 5: Added log saving to password_log.txt and wordlist export to custom_wordlist.txt.
- Step 6: Final testing with multiple inputs and password variations, validating output files.

5. Conclusion

This project helped me understand password entropy, common password flaws, and how custom wordlists are used in password cracking attempts. It also gave hands-on experience with Python scripting, CLI design, and string manipulation techniques used in ethical hacking.

In the future, this tool could be extended with a GUI, or integrated with password cracking tools like John the Ripper.