Project Report: A Dual-Function Cybersecurity Tool

Password Strength Analyzer & Custom Wordlist Generator

Prepared for: Elevate Labs & Ministry of MSME, Government of India

Date: July 28, 2025

1. Introduction

Strong passwords are vital in protecting digital systems. Many people use weak passwords, exposing themselves to cyber threats. This project, developed during an Elevate Labs internship, provides both a password strength analyzer and a wordlist generator to demonstrate secure practices and common attack strategies.

The Python-based tool educates users about creating secure passwords and shows how attackers exploit personal data to guess them. It aims to enhance user awareness and password hygiene.

2. Abstract

This dual-purpose application has two modules: a password analyzer and a wordlist generator. The analyzer evaluates a password using zxcvbn to detect patterns and estimate crack time. It scores from 0 to 4 with feedback.

The generator simulates attacks by using personal data to generate possible passwords. It applies techniques like leetspeak, capitalization, and appending symbols. The result is a comprehensive command-line tool demonstrating both defense and attack perspectives in cybersecurity.

3. Tools Used

- Python 3: Chosen for ease of use and strong library support.
- zxcvbn-python: Estimates password strength using real-world data.
- itertools: Generates permutations and combinations efficiently.
- datetime: Parses dates for mutation in password generation.

Project Report: A Dual-Function Cybersecurity Tool

4. Steps Involved in Building the Project

1. Conceptualization: Selected the project idea and planned module development.
2. Setup: Installed Python and dependencies like zxcvbn.
3. Analyzer Module: Collected input, processed with zxcvbn, and presented user-friendly results.
4. Wordlist Generator:
   - Gathered personal info from user.
   - Sanitized inputs and generated keywords.
   - Applied variations (leetspeak, caps, suffixes) with itertools.
5. Integration: Built a menu interface linking both modules.
6. Documentation: Cleaned code, wrote README, generated requirements.txt, and prepared this report.

5. Conclusion

The tool meets its goal of educating users about password security. It offers a hands-on look at weak password risks and how attackers exploit them. This project improved skills in Python, libraries, and command-line app structure.

Future work could include a GUI, breach data integration, and more advanced password mutation rules. The tool is a practical demonstration of cybersecurity fundamentals.