Password Strength Analyzer with Custom Wordlist Generator

Project Report

Name: Pradeepsha

Introduction

In today's digital world, passwords act as the first line of defence against unauthorized access. Weak or reused passwords are a common cause of security breaches, making it essential to help users create stronger credentials. This project focuses on building a **Password Strength Analyzer with Custom Wordlist Generator**, a tool that not only evaluates password security but also generates custom wordlists for penetration testing and security research.

Abstract

The Password Strength Analyzer is a lightweight tool designed using **Python, HTML, CSS, and JavaScript**. It allows users to:

- **Analyze password strength** using both **rule-based checks** and the **zxcvbn library**.
- **Generate custom wordlists** based on user-provided information (e.g., names, birthdays, or keywords).
- Provide **feedback** on weak passwords along with recommendations for improvement.

Tools Used

1. **Python** – for backend password analysis and wordlist generation.
2. **zxcvbn** – open-source library for advanced password strength estimation.
3. **Argparse** – for handling command-line arguments in the Python tool.
4. **HTML5 & CSS3** – for designing a simple and responsive user interface.
5. **JavaScript (ES6)** – for client-side password strength evaluation.
6. **Git & GitHub** – version control and repository hosting.
7. **GitHub Pages** – for free hosting of the static website.

Steps Involved in Building the Project

1. **Planning & Research** – Studied password security practices and identified key requirements.
2. **Backend Development** – Implemented a Python tool using argparse and zxcvbn for password analysis and wordlist generation.

3. **Frontend Development** – Designed a browser-based interface with an input field, strength meter, and feedback section.
4. **Integration of zxcvbn.js** – Used JavaScript to evaluate password entropy and estimate crack times.
5. **Testing** – Verified with weak and strong passwords, checking the accuracy of feedback and wordlist output.
6. **Deployment** – Published the static web version using GitHub Pages for global accessibility.

Conclusion

The Password Strength Analyzer with Custom Wordlist Generator successfully demonstrates how programming and web technologies can be applied to improve security awareness. By combining **password analysis** with **custom wordlist generation**, the project serves dual purposes: educating users on strong password creation and assisting security professionals in penetration testing.

Future improvements may include:

- Adding a **password generator** with strong random suggestions.
- Building a **database integration** for analyzing common password leaks.