Password Strength Analyzer & Wordlist Generator

Abstract

This project implements a **Password Strength Analyzer and Wordlist Generator** using Python. The tool evaluates the strength of user-provided passwords by measuring length, entropy, and character composition. It also highlights common weaknesses such as short length, predictable sequences, and repeated characters. Along with analysis, the tool provides actionable **suggestions** for creating stronger passwords. Additionally, a **custom wordlist generator** is included to assist in security testing and auditing. This project demonstrates the importance of password security and provides practical exposure to cybersecurity practices.

Introduction

Passwords remain the most common method of authentication, yet weak passwords are one of the leading causes of security breaches. Attackers often exploit simple, predictable, or reused passwords to gain unauthorized access. This project addresses the challenge by developing a tool that can analyze password strength, highlight weaknesses, and suggest improvements. It also generates custom wordlists that can be used for penetration testing. The tool thus bridges the gap between **awareness** (teaching users to create strong passwords) and **testing** (assisting ethical hackers in password audits).

Tools Used

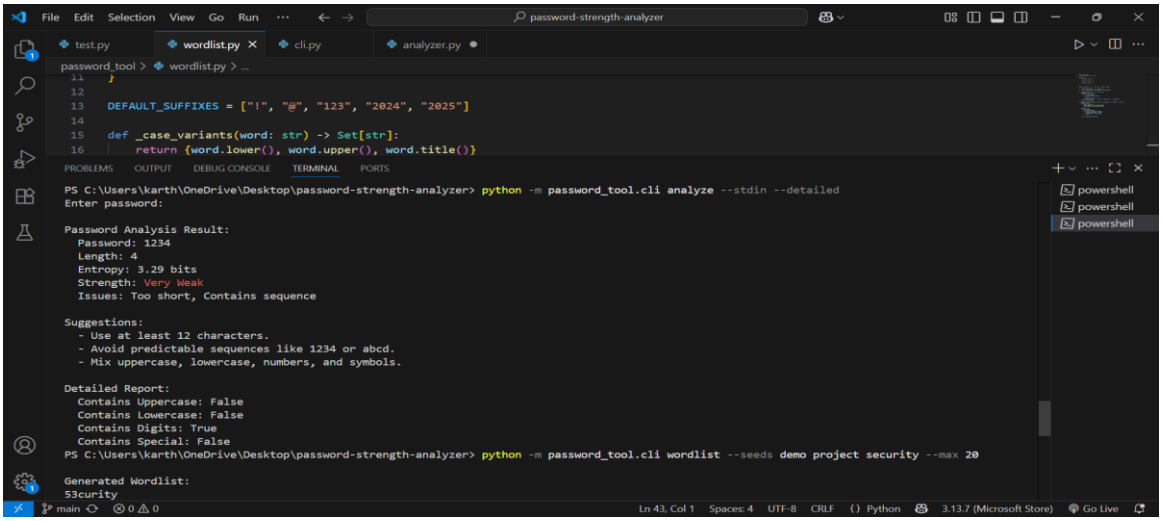
- **Python 3.10+** – Programming language
- **argparse** – Command-line interface handling
- **getpass** – Secure password input
- **colorama** – Enhanced terminal output with colors
- **GitHub** – Version control and repository hosting

Steps Involved in Building the Project

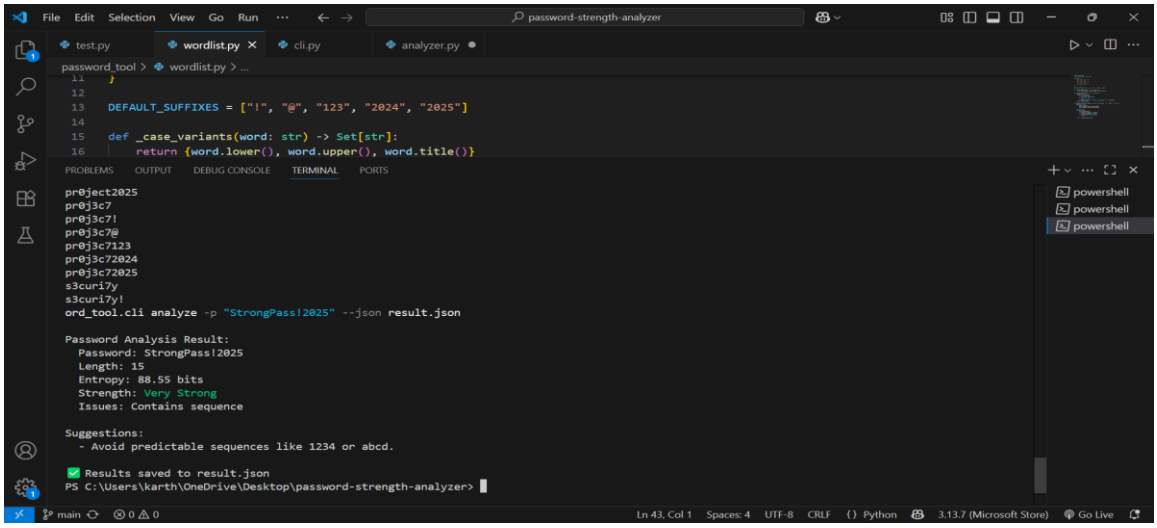
1. **Project Setup** – Created a virtual environment and project structure (`password_tool/`).
2. **Analyzer Module** – Implemented `analyze_password` function to calculate entropy, classify strength (Very Weak → Very Strong), and detect weaknesses.
3. **Suggestions Engine** – Added logic to suggest improvements such as longer passwords, use of mixed character sets, and avoiding predictable sequences.
4. **Detailed Report** – Enhanced analysis to show presence of uppercase, lowercase, digits, and special characters.
5. **Wordlist Generator** – Implemented `generate_wordlist` to create targeted wordlists from seed words, with options to save to a file.
6. **Command-Line Interface (CLI)** – Built `cli.py` to allow users to analyze passwords, view detailed reports, and generate wordlists through simple commands.
7. **Testing & Validation** – Ran multiple test cases to verify classification accuracy and ensured proper file outputs (JSON and wordlist files).

Output

🔎 Password Analysis



📝 Wordlist Generation



Conclusion

The **Password Strength Analyzer & Wordlist Generator** successfully analyzes password strength, provides detailed reports, and suggests improvements for stronger security. It also generates useful wordlists for penetration testing, making it valuable for both users and cybersecurity learners. The project highlights the importance of password hygiene and demonstrates practical skills in Python programming and security testing. Future improvements may include integration with breach databases (e.g., HaveIBeenPwned), a GUI version for non-technical users, and batch password analysis.