Password Strength Analyzer with Custom Wordlist Generator

This project was developed as part of a Cyber Security Internship. The main goal of this project is to analyze the strength of passwords and educate users about secure password practices. Weak passwords are one of the most common reasons for security breaches, making password strength analysis an important aspect of cybersecurity.

Project objectives :

The objectives of this project are as follows:

- To evaluate password strength using basic rule-based checks.
- To perform advanced password analysis using the **zxcvbn** library.
- To estimate password crack time and strength score.
- To generate a custom wordlist based on user-provided information.
- To support command-line execution for automation.

Features :

1.Checks password length and character types (uppercase, lowercase, digits, special characters)

2.Uses zxcvbn to estimate password strength and crack time

3.Displays clear strength results (Weak / Medium / Strong)

4.Generates a custom wordlist using name, pet name, and year

5.Supports command-line arguments using argparse

6.Allows exporting wordlists to a .txt file

Tools and Technologies Used :

1.Python 3

2.zxcvbn library

3.argparse module

4.Command Line Interface (CLI)

Implementation Details :

The password strength analyzer accepts input either through the command line or interactive mode. It performs basic checks such as password length, presence of uppercase letters, lowercase letters, digits, and special characters. Advanced analysis is performed using the zxcvbn library, which provides a strength score and estimated crack time.

The project also includes a custom wordlist generator. Users can provide optional inputs such as name, pet name, and significant year. Based on these inputs, the tool generates realistic password combinations using techniques like leetspeak substitutions and year appending. The generated wordlist is exported as a text file for security testing and learning purposes.

Project structure :

Password strength Analyzer

- main.py
- wordlist.py
- requirements.txt
- README.md
- wordlist.txt

Installation & setup :

1.Clone the repository or download the project folder

2.stall required dependencies:

```
pip install -r requirements.txt
```

How to run the project :

1.Analyze password using command line

```
python main.py --password "YourPassword123!"
```

2.Generate wordlist and export it

```
python main.py --export
```

3.Run in interactive mode

```
python main.py
```

Sample output :

- Password length and character checks
- Strength score (0–4)
- Estimated crack time
- Final strength label (e.g., STRONG)
- Wordlist generated and saved as wordlist.txt

Results and Output :

The tool successfully evaluates password strength and categorizes it as weak, medium, or strong. It displays estimated crack time and strength score to help users understand password security. When enabled, the wordlist generator creates and exports a customized wordlist containing multiple password combinations.

Learning outcomes :

- Learned how password strength is evaluated
- Gained hands-on experience with Python modules and libraries
- Understood argument parsing using argparse
- Improved understanding of cybersecurity basics
- Built a real-world CLI-based Python project

Conclusion:

This project provided hands-on experience in developing a cybersecurity tool using Python. It helped in understanding password vulnerabilities and the importance of strong password policies. The project fulfills all internship requirements and demonstrates practical application of cybersecurity concepts.

AUTHOR

Sowmya S

Intern – Cybersecurity