Password Strength Analyzer with Custom Wordlist Generator Project Report

◆ INTRODUCTION:

Cybersecurity is a cornerstone of modern digital defense mechanisms. With the increasing reliance on online services, cloud platforms, and digital storage, safeguarding sensitive information has become critical. Among the many security risks, weak or reused passwords remain one of the most exploited vulnerabilities in both personal and corporate environments.

Attackers often use automated scripts that run through billions of password combinations to compromise accounts. These scripts are often powered by pre-built wordlists that contain common patterns, personal data, and predictable formats. This project focuses on addressing this weakness by creating a tool that can both analyse the strength of a password and demonstrate how attackers might generate wordlists using simple user information.

◆ ABSTRACT:

The **Password Strength Analyzer with Custom Wordlist Generator** is a Python-based tool that accomplishes two main objectives:

1. **Password Strength Analysis**: It uses the zxcvbn algorithm developed by Dropbox to evaluate passwords. This includes providing a score from 0 (weak) to 4 (strong), estimating crack time using various attack methods, and offering actionable feedback for stronger passwords.
2. **Custom Wordlist Generation**: The tool accepts user inputs like name, date of birth, and pet name to create personalized wordlists. These inputs are processed using leetspeak substitutions (e.g., a → @, s → $) and common suffixes (123, @, 2024, etc.) to simulate how attackers form password guesses.

The goal is to educate users about the importance of strong, unpredictable passwords while also providing hands-on experience with a key cybersecurity concept: dictionary-based password attacks.

◆ Tools Used

- **Python 3**: The core programming language used for the entire project.
- **zxcvbn-python**: A Python implementation of Dropbox’s password strength estimation algorithm.
- **itertools**: A built-in Python library used for combinations and text variations.
- **datetime**: For managing and formatting date-based data entries.
- **File I/O**: Used to read user input and save the generated wordlist in .txt format.

These tools were selected for their simplicity, wide support, and effectiveness in security-related tasks.

🔹 Steps Involved in Building the Project

1. **Project Setup**: Python 3 was installed and dependencies like zxcvbn-python were added using pip.
2. **Password Analysis Module**: A function was created using zxcvbn to evaluate passwords. This module provides crack time estimates, entropy scores, and improvement suggestions.
3. **User Input**: The script prompts users to enter their name, date of birth, and pet name or favorite item.
4. **Wordlist Generation**:
   - Each user input is processed.
   - Variants are created using leetspeak, number suffixes (123, 2024, etc.), and special characters (!, @).
   - The final list is cleaned of duplicates and saved as a .txt file.
5. **Execution and Output**: Results from both the password analysis and wordlist generation are printed to the console. The wordlist can be used for educational testing with password cracking tools.
6. **Modularity**: The code is structured with functions and main() so it's reusable and easy to expand (e.g., adding a GUI later).

🔹 Conclusion

This project demonstrates the ease with which weak or personally predictable passwords can be cracked using automated tools. By allowing users to input their own data and see the kinds of wordlists attackers might generate, the tool effectively teaches one of the core lessons in password hygiene.

Moreover, by combining theoretical understanding with practical coding, this project becomes both a learning tool and a potential resume booster. It highlights essential cybersecurity skills including entropy evaluation, dictionary attacks, and secure software design principles.

This project is particularly beneficial for students and beginners who wish to understand the attacker's mindset and develop better security practices in real-world scenarios. It's a simple, effective, and educational tool that fits perfectly into any cybersecurity learning path or internship submission.