PASSWORD STRENGTH ANALYZER WITH CUSTOM WORDLIST GENERATOR

1. Introduction

Passwords serve as the primary layer of protection for digital systems and user accounts. However, many users create weak passwords using predictable personal information such as names, dates of birth, or pet names. Such patterns make passwords vulnerable to brute-force and dictionary attacks. Attackers often exploit common combinations like appending years (2024, 123) or applying leetspeak substitutions (@, 3, 1). This project aims to analyze password strength and demonstrate how attackers generate targeted wordlists using personal data patterns.

2. Abstract

The Password Strength Analyzer with Custom Wordlist Generator is a Command Line Interface (CLI)-based security tool developed using Python. The system evaluates password strength using the zxcvbn library, which estimates crack time and provides entropy-based feedback. Additionally, the tool generates customized wordlists based on user inputs such as name, date of birth, and pet name. It applies common password attack strategies including year appending and leetspeak substitutions to simulate real-world password cracking techniques. The generated passwords are exported into a .txt file for educational and security testing purposes. This project highlights password security awareness and practical attack simulation concepts in cybersecurity.

3. Tools Used

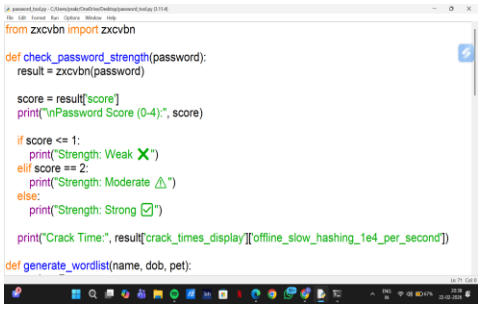
- Python 3.11
- zxcvbn Library
- NLTK (Optional)
- Command Line Interface (CLI)
- IDLE / Command Prompt

4. Steps Involved

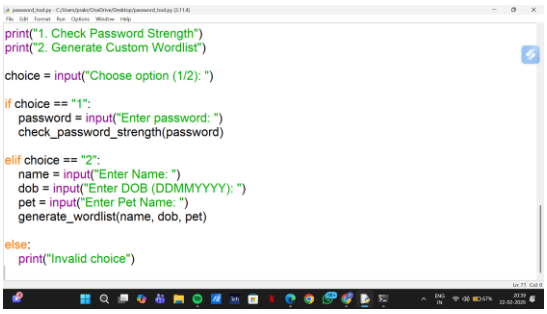
1. Installed Python and required libraries using pip (zxcvbn and nltk).
2. Implemented password strength analysis using zxcvbn to calculate score (0–4 scale) and estimate offline crack time.
3. Developed a custom wordlist generator using user inputs (Name, DOB, Pet Name).
4. Applied password attack techniques such as year appending and leetspeak substitutions.
5. Removed duplicate entries and exported results to wordlist.txt.
6. Designed a structured CLI menu with loop control and modular functions.

5. Conclusion

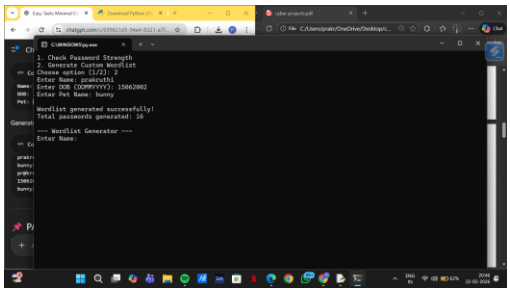
This project demonstrates how weak and predictable password patterns can be exploited using common attack techniques. By combining password strength analysis with targeted wordlist generation, the tool emphasizes the importance of creating strong and unpredictable passwords. The structured CLI implementation reflects practical software development standards and enhances understanding of password security and real-world attack methodologies.



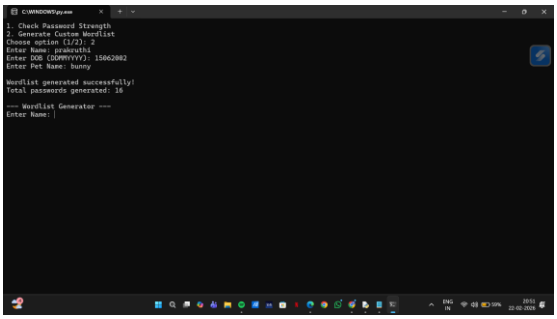
```
from zxcvbn import zxcvbn

def check_password_strength(password):
    result = zxcvbn(password)

    score = result['score']
    print("\nPassword Score (0-4):", score)

    if score <= 1:
        print("Strength: Weak ❌")
    elif score == 2:
        print("Strength: Moderate ⚠️")
    else:
        print("Strength: Strong ☑️")

    print("Crack Time:", result['crack_times_display']['offline_slow_hashing_1e4_per_second'])

def generate_wordlist(name, dob, pet):
```



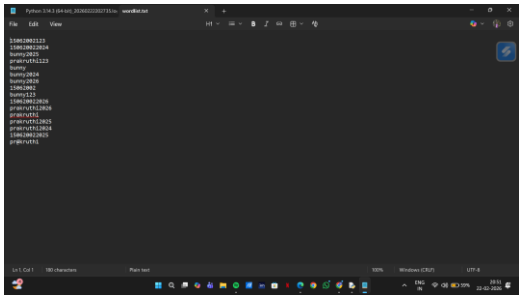
```
print("1. Check Password Strength")
print("2. Generate Custom Wordlist")

choice = input("Choose option (1/2): ")

if choice == "1":
    password = input("Enter password: ")
    check_password_strength(password)

elif choice == "2":
    name = input("Enter Name: ")
    dob = input("Enter DOB (DDMMYYYY): ")
    pet = input("Enter Pet Name: ")
    generate_wordlist(name, dob, pet)

else:
    print("Invalid choice")
```



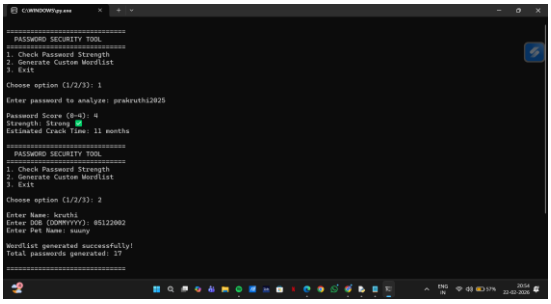
```
1. Check Password Strength
2. Generate Custom Wordlist
Choose option (1/2): 2
  Enter Name: prakruthi
Enter DOB (DDMMYYYY): 15062002
Enter Pet Name: bunny

Wordlist generated successfully!
Total passwords generated: 16

  --- Wordlist Generator ---
  Enter Name:
```



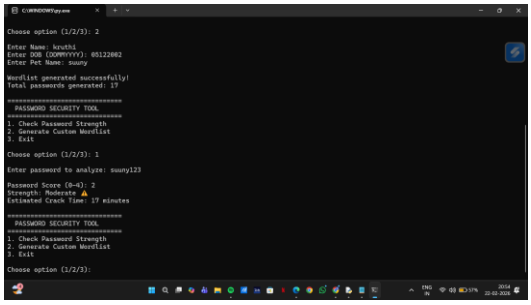
```
1. Check Password Strength
2. Generate Custom Wordlist
Choose option (1/2): 2
Enter Name: prakruthi
Enter DOB (DDMMYYYY): 15062002
Enter Pet Name: bunny

Wordlist generated successfully!
Total passwords generated: 16

--- Wordlist Generator ---
Enter Name:
```



```
15062002123
15062002024
bunny2025
prakruthi123
bunny
bunny2024
bunny2026
15062002
bunny123
15062002026
prakruthi2026
prakruthi
prakruthi2025
prakruthi2024
15062002025
prakruthi
```



```
==============================
  PASSWORD SECURITY TOOL
==============================
1. Check Password Strength
2. Generate Custom Wordlist
3. Exit

Choose option (1/2/3): 1

Enter password to analyze: prakruthi2025

Password Score (0-4): 4
Strength: Strong 🟩
Estimated Crack Time: 11 months

==============================
  PASSWORD SECURITY TOOL
==============================
1. Check Password Strength
2. Generate Custom Wordlist
3. Exit

Choose option (1/2/3): 2

Enter Name: kruthi
Enter DOB (DDMMYYYY): 05122002
Enter Pet Name: suuny

Wordlist generated successfully!
Total passwords generated: 17

==============================
```



```
Choose option (1/2/3): 2

Enter Name: kruthi
Enter DOB (DDMMYYYY): 05122002
Enter Pet Name: suuny

Wordlist generated successfully!
Total passwords generated: 17

==============================
  PASSWORD SECURITY TOOL
==============================
1. Check Password Strength
2. Generate Custom Wordlist
3. Exit

Choose option (1/2/3): 1

Enter password to analyze: suuny123

Password Score (0-4): 2
Strength: Moderate ⚠️
Estimated Crack Time: 17 minutes

==============================
  PASSWORD SECURITY TOOL
==============================
1. Check Password Strength
2. Generate Custom Wordlist
3. Exit

Choose option (1/2/3):
```