📄 Project Submission: Password Cracker (Brute Forsce using Wordlists)

For: Internship Application – Penetration Tester
Submitted to: Deltaware Solutions Pvt. Ltd.
Submitted by: JAYADEV PANDA
Date: 04/08/2025

🔐 Project Title:

Password Hash Cracker using Brute Force and Wordlists

📝 Objective:

To build a password hash cracking tool that uses a brute-force approach with a wordlist to test the strength of hashed passwords. This tool is designed for ethical hacking and penetration testing environments to help organizations identify weak password implementations.

□ Project Overview:

This tool simulates the method attackers use to crack passwords by comparing hashed values of potential passwords (from a wordlist) to a given hash. It supports multiple hashing algorithms like **MD5**, **SHA-1**, and **SHA-256**, making it suitable for real-world penetration testing scenarios.

⚙ Technology Stack:

- **Programming Language:** Python 3
- **Libraries Used:** hashlib, argparse, time
- **Platform:** CLI (Command Line Interface)
- **Hash Algorithms Supported:** MD5, SHA-1, SHA-256

📁 Project Files:

- cracker.py — Main Python script for cracking
- wordlist.txt — Dictionary of possible passwords
- README.md — Project documentation and instructions
- sample_hashes.txt — Sample hashes for demonstration
- Optional: Screenshots or demo video

□ How It Works:

1. User inputs the target hash and selects the hashing algorithm.
2. The script reads passwords from the wordlist.
3. Each word is hashed and compared to the target hash.
4. If a match is found, the password is displayed.

📌 **Sample Output:**

less

```
[+] Trying: 123456
[+] Trying: password
[✓] Match found: password
[⏱] Time taken: 0.18 seconds
```

□ **Code Snippet:**

python

```
import hashlib

def crack(hash_input, wordlist_path, algo='md5'):
    with open(wordlist_path, 'r') as f:
        for word in f:
            word = word.strip()
            if algo == 'md5':
                hashed = hashlib.md5(word.encode()).hexdigest()
            elif algo == 'sha1':
                hashed = hashlib.sha1(word.encode()).hexdigest()
            elif algo == 'sha256':
                hashed = hashlib.sha256(word.encode()).hexdigest()
            if hashed == hash_input:
                return f"Password found: {word}"
    return "Password not found."
```

```
# Example

print(crack('5f4dcc3b5aa765d61d8327deb882cf99', 'wordlist.txt', 'md5')) # "password"
```

□ Use Cases:

- Evaluate password strength during penetration tests
- Demonstrate importance of strong password policies
- Ethical hacking practice in a controlled lab environment

⚠ Ethical Disclaimer:

This tool is intended **strictly for ethical and educational purposes**. Unauthorized use against systems without permission is illegal and unethical. Always ensure you have written permission before using this tool in real-world environments.

🚀 Future Improvements:

- Multi-threading or multiprocessing for faster cracking
- GUI version for non-technical users
- Salted hash support
- Integration with online hash databases

📞 Contact Info (Optional):

- Name: JAYADEV PANDA
- Email: jayadev19198@gmail.com
- Phone: 8249868119
- LinkedIn: https://www.linkedin.com/in/jayadev-panda-622a84318?

✅ Declaration:

I declare that this project was developed by me for the sole purpose of educational and ethical penetration testing practices. I understand the legal and ethical implications of cybersecurity tools and confirm that I will use this tool responsibly.
