Brute-force Password Guessing Web Application

Done by: Janavi Vilas Niwate

Sonali pawar

PTVA Sathaye College

Abstract

This project is a demonstration of a brute-force password guessing system developed using Python and Flask. The tool is designed for educational purposes to illustrate how easily short and simple passwords can be compromised using brute-force techniques. The goal is to raise awareness about the importance of strong password policies and basic cybersecurity hygiene.

Problem Statement & Objective

Weak passwords are a major cybersecurity vulnerability. This project aims to simulate brute-force attacks to showcase how quickly simple passwords can be guessed, highlighting the need for stronger authentication mechanisms.

Literature Review

Studies have shown that a significant number of breaches are due to weak or reused passwords. Brute-force techniques are among the oldest forms of attack but remain effective against poorly designed systems. Various password-cracking tools like Hydra, John the Ripper, and Hashcat support brute-force modules. However, awareness and education remain the most effective countermeasures.

Research Methodology

The project uses the Flask framework to build a web interface. A brute-force algorithm iterates through all possible combinations of lowercase letters and digits up to four characters long. Time taken and number of guesses are recorded for each input.

Tool Implementation

The main script, app.py, initializes a Flask server with routes for the index and result pages. Users

input a password, and the server responds with the brute-force guess results. The frontend uses basic HTML and CSS for interaction.

Results & Observations

The tool successfully demonstrates brute-force password cracking for passwords up to 4 characters. On average, the system can guess such passwords within seconds, depending on the password complexity and system performance.

Ethical Impact & Market Relevance

This project is intended for ethical and educational use only. It serves as a proof of concept to emphasize the importance of choosing strong, complex passwords and to support cybersecurity training programs.

Future Scope

Future versions of the tool may incorporate hash cracking, support for longer passwords, implementation of rainbow tables, and better visualization tools to make demonstrations more effective for educational purposes.

References

1. OWASP. Password Storage Cheat Sheet. https://owasp.org

2. Bonneau et al., The Quest to Replace Passwords, IEEE Security & Privacy, 2012.

3. NIST Digital Identity Guidelines, 2020.

4. Anderson, R. (2020). Security Engineering.

5. John the Ripper - https://www.openwall.com/john/

6. Hashcat - https://hashcat.net/

7. Bruce Schneier, Applied Cryptography, 1996.

8. TryHackMe - Password Cracking Rooms.

9. SANS Institute, Password Auditing and Cracking Tools.

10. MITRE ATT&CK Framework - Credential Access Techniques.