Introduction

This tool helps you assess password strength by trying to "crack" their hashed values. A hash is a one-way encoded representation of a password. Security professionals and ethical hackers use such tools to confirm their systems are secure or to recover passwords they have permission to access.

Important Notice:
Use this tool only on systems and passwords for which you have explicit legal permission. Unauthorized use may be illegal.

What the Tool Does

- **Hash Cracking:** Given a password hash, the tool attempts multiple passwords—either from a dictionary file or by brute force. If a match is found, it displays the recovered password.
- **Algorithms Supported:**
  - *Unsalted:* MD5, SHA1, SHA256, SHA512, SHA3-256, Scrypt
  - *Salted:* Bcrypt, Argon2id
- **Cracking Methods:**
  1. **Dictionary-Based Attacks:** Uses known or commonly used passwords from a list.
  2. **Brute Force Attacks:** Tries every possible combination of characters up to a specified length.
- **Resource Management:**
  The tool can run multiple threads in parallel, speeding up the process. You can choose from preset options (Low, Medium, High) or enter a custom number of threads.

Prerequisites

1. **Python 3.x:**
   Install from https://www.python.org/downloads/ if needed.
2. **Required Python Libraries:**
   - bcrypt (for bcrypt hashes)
   - argon2-cffi (for argon2id hashes)

   Install them with:
   pip install bcrypt argon2-cffi

   Other required libraries (hashlib, itertools, string, sys, os, time, threading, concurrent.futures) are included in Python's standard library.

Step-by-Step Usage Guide

1. **Running the Tool:**
   Start the tool.
2. **Initial Notice:**
   The tool will display a disclaimer reminding you to use it ethically and legally.
3. **Choosing Resource Usage (Threads):**
   You'll be asked how many threads to use:
   - Option 1: Low (1 thread)
   - Option 2: Medium (4 threads)
   - Option 3: High (8 threads)
   - Option 4: Custom (enter a number between 1 and 1000)

   Tip: Start with Low or Medium if you're new.

4. **Main Menu:**
   - Option 1: Crack Password
   - Option 2: Exit

   Choose 1 to begin the cracking process.

5. **Selecting the Hash Algorithm:**
   Type the algorithm you're working with (e.g., md5, sha1, sha256, sha512, sha3_256, bcrypt, argon2id, scrypt).
6. **Entering the Hash Value:**
   Paste or type the hash when prompted. If the hash format is incorrect, you'll be asked to try again.
7. **Choosing the Cracking Method:**
   - **Dictionary-Based:** Provide one or more dictionary files. The tool removes duplicates and tries each password in turn.
   - **Brute Force:** Specify the password length, and the tool will try all combinations of letters and digits at that length.

   Note: Brute force can be slow for longer, more complex passwords.

8. **Running the Cracking Process:**
   The tool will show progress as it tries each password.
   - If it finds the correct password, it will display it.
   - If it doesn't find a match, it will let you know.
9. **Timing Information:**
   After the attempt, the tool shows how long it took, helping you understand the impact of chosen settings and the complexity of the hash.

Example Usage

- **Dictionary Attack (MD5):**
  1. Run the tool and select resource usage.
  2. Choose 1 (Crack Password).
  3. Algorithm: md5
  4. Enter the MD5 hash.
  5. Choose dictionary-based cracking, specify the number of dictionaries, and provide their paths.
     The tool then tries each password from the dictionary until it finds a match.
- **Brute Force Attack (SHA1):**
  1. Run the tool and select resource usage.
  2. Choose 1 (Crack Password).
  3. Algorithm: sha1
  4. Enter the SHA1 hash.
  5. Choose brute force.
  6. Enter the password length (e.g., 4).
     The tool will attempt all possible 4-character passwords made of letters and digits until it finds a match or exhausts all possibilities.

FAQ

- **Q:** Can I crack any hash I find?

  A: No. Only use this tool on hashes you're authorized to test. Unauthorized use is illegal.

- **Q:** Why is it slow?

  A: Some algorithms are intentionally slow. Try more threads or a more powerful machine, but understand that strong algorithms like argon2id or scrypt are designed to resist quick cracking.

- **Q:** Hash not recognized?

  A: Ensure the hash matches the chosen algorithm and that it's entered correctly.

- **Q:** Can I add more algorithms?

  A: It's possible with code modifications, but not recommended for beginners.

Legal and Ethical Disclaimer
This tool is meant for educational, authorized security testing only. Always act lawfully and ethically. You are responsible for ensuring compliance with all applicable laws and regulations.

By following these steps, even those with minimal technical experience can use this tool responsibly to test password strength and recover permitted passwords.