Project 1: Password Cracking

Objective: The objective of this project is to learn and understand the process of password cracking using tools like John the Ripper or Hashcat to crack password from hashed files.

Steps to Perform the Project

1. **Understand Password Hashing:** Before diving into password cracking. It’s crucial to understand how password are stored in systems. Password are not stored in plain text but are hashed using cryptographic algorithms like MD5, SHA-1, Sha-256, etc. Hashing converts the password into a fixed-length string of characters, making it computationally difficult to reverse the process and obtain the original password.
2. **Select a Password Cracking Tool:** Choose a password cracking tool such as John the Ripper or Hashcat. These tools support various hash algorithms and have capabilities for dictionary attacks, brute-force attacks and rule-based attacks.
3. **Acquire Hashed Passwords:** Obtain a set of hashed passwords for testing purposes. You can use sample hashed password file from online resources or generate your own hashed passwords using tools like openssl, hashlib in python or online hash generators.
4. **Prepare Wordlists:** Create or obtain wordlists (lists of potential passwords) for use in dictionary-based attacks. Wordlists can contain common passwords, dictionary words, permutations, or patterns relevant to the target users.
5. **Perform Dictionary Based Attack:**
    - A. Use the password cracking tool (e.g John The Ripper) with the hashed password file and a wordlist to perform a dictionary attack.
    - B. Common examples for John the Ripper: **john –-format=FROMATE hash-file.txt wordlist.txt**

Replace FORMATE with the appropriate hash format (e.g. MD5, SHA-1) and provide the paths to hashed password file (hash_file.txt) and wordlist (wordlist.txt)

Perform Brute-Force Attack:

a) Use the password cracking tool with specific parameters to perform a brute-force attack by trying all possible combinations of characters.
b) Command example for Hashcat: **hachcat -m 0 hash_file.txt rockyou.txt**

Replace -m 0 with hash mode appropriate for the hash algorithm used in your hashed password (e.g., -m 0 for MD5) and provide the path to the hashed password file (hash_file.txt) and a wordlist (rockyou.txt is a common wordlist).

1. **Analyze Results:** Once the password cracking process complete, analyze the results to identify successfully cracked passwords. The tool will display cracked password if found in the chosen wordlist or through brute-force.
2. **Evalute Security Implications:** Reflection on the implication of successful password cracking. Understand the importance of using strong, unique passwords, implementing salted hashes, and employing additional security measures such as multi-factor authentication (MFA) to enhance password security.

3. **Document Findings:** Document the process, tools used, result obtained and any insights gained during the password cracking exercise. This documentation helps in understanding password security concepts and can be used for educational or security assessment purposes.
4. **Further Learning:** Explore advanced password cracking techniques, password policy enforcement, password salting, and security best practices related to password management in systems and applications.

By following these steps, you'll gain hands-on experience in password cracking techniques, understand the importance of password security, and learn about common vulnerabilities associated with weak or poorly managed passwords in systems.

Let's dive deeper into each step of the Password Cracking Project

1. Understand Password hashing

- Password are not stored as plain text in modern system due to security concerns. Instead, they are converted into a hashed format using cryptographic hash functions.
- Hashing algorithms like MD5, SHA-1, SHA-256 are commonly used to convert password into fixed-length hash values.
- Hashing is a one-way process, meaning it's designed to be computationally difficult to reverse the hashed value back to the original password.

2. Select a password Cracking Tool

- John the Ripper: it's a versatile password cracking tool that supports various hash types and attack modes such as dictionary attacks, brute-force attacks, and hybrid attacks.
- Hashcat: Known for its speed and GPU acceleration, Hashcat supports a wide range of hash algorithms and attack modes, making it suitable for craking complex passwords.

3. Acquire Hashed Passwords:

- You can obtain hashed passwords from various sources such as leaked password databases, security challenges, or by generating your own hashed passwords for testing purposes.

4. Prepare Wordlists:

- Wordlists contain a collection of potential password that the password cracking tool will use during dictionary attacks.
- Wordlists can be generic (containing common passwords) or customized based on specific patterns, rules, or known information about the target users (e.g., common words related to a company's culture).

5. Perform Dictionary Attack:

- In a dictionary attack, the password cracker compares each hashed password in the file against the entries in the wordlist.
- If a hashed password matches an entry in the wordlist, it means the corresponding plain-text password has been found.

6. Perform Brute-Force Attack:

- Brute-force attack involve trying every possible combination of characters until the correct password is discovered.
- Depending on the Complexity of passwords and the computing power available, brute-force attacks can be time-consuming but effective against week passwords.

7. Analyze Results:

- After the password cracking process completes, review the output provided by the password cracking tool.
- The tool will indicate which passwords were successfully cracked and display them along with their corresponding hashed values.

8. Evaluate Security Implications:

- Successful password cracking highlights the importance of strong, unique passwords and proper password strong mechanisms.
- Organizations should enforce password policies that encourage users to create complex passwords and regularly update them. Additionally, implementing salted hashes adds an extra layer of security by making each password hash unique.

9. Document Finding:

- Documenting the password cracking process, tools used, and results obtained helps in knowledge sharing, security training, and improving overall security practices within organizations.
- Note down any insights gained during the exercise, such as common password patterns, password reuse issues, or weeknesses in password security implementaions.

10. Further learning:

- Explore advanced password cracking techniques such as rainbow table attacks, GPU acceleration for faster cracking speeds, and password spraying attacks against authentication systems.
- Stay updated with password security best practices, industry standard (e.g., NIST guidelines), and evolving password cracking methodologies to better defend against password-based attacks.