CYBER SECURITY

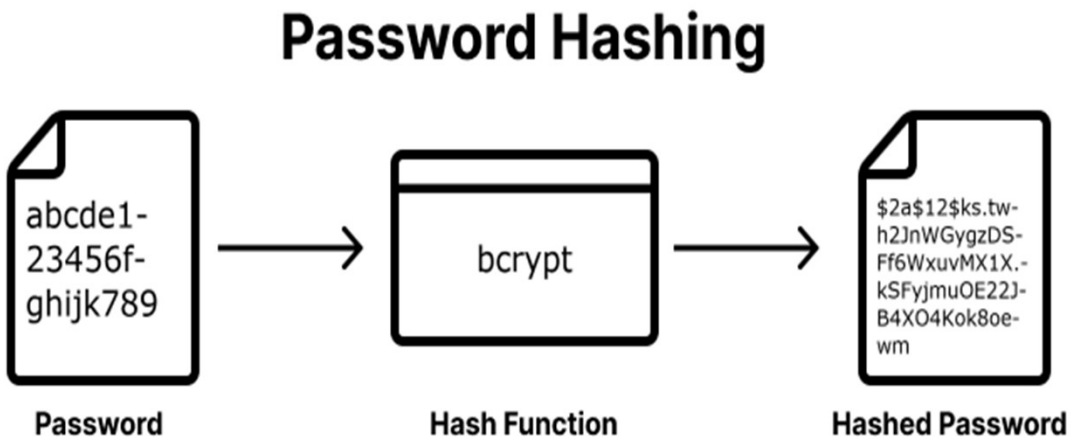
ASSIGNMENT

AMRUTHA SIVARATHRI
160122737021
IT-1

JOHN THE RIPPER
Password Cracking Tool

Password Cracking

Password cracking involves retrieving the original password from its hash, which is generated using algorithms like MD5, SHA-1, SHA-256, or bcrypt. These algorithms convert passwords into irreversible, fixed-length hashes. Cracking techniques such as brute force, dictionary attacks, or rainbow tables are used to guess the original password. Stronger hashing algorithms provide better protection against unauthorized access.



ALGORITHMS USED TO HASH

- MD5 and SHA-1 are outdated and insecure for password hashing.
- SHA-256 is secure for general hashing purposes but not ideal for passwords.
- bcrypt is currently one of the best choices for password hashing due to its resistance to brute-force attacks and adaptability.

TECHNIQUES TO CRACK

- Brute-Force Attack: Attempts every possible character combination to guess the password.
- Dictionary Attack: Uses common words or phrases to quickly guess likely passwords.
- Rainbow Tables: Uses precomputed hash tables to match hashed passwords, bypassed by salting.

Hashing Algorithms Overview

Different hash algorithms offer varying levels of security for stored passwords. MD5 and SHA-1 were once widely used due to their speed, but they are now considered insecure because they are vulnerable to collision attacks and can be brute-forced rapidly with modern hardware. SHA-256 improves security by producing longer hashes, but it is still relatively fast and not ideal for protecting passwords. In contrast, bcrypt is specifically designed for password hashing. It incorporates a computational cost factor that can be adjusted to make brute-force attacks more time-consuming. Additionally, bcrypt automatically incorporates a salt to defend against rainbow table attacks, further enhancing security.

What is John the Ripper?

- John the Ripper is an open-source password cracking tool used for testing password strength and recovering lost passwords.
- It supports a wide range of hash algorithms like MD5, SHA-1, NTLM, bcrypt, and more.
- Utilizes dictionary attacks, brute-force, and hybrid methods for password cracking.
- Can leverage multiple CPU cores or GPUs to speed up the cracking process.
- Known for its versatility and efficiency in handling different hash types.
- Popular among penetration testers and security professionals.

Architecture of John the Ripper

John the Ripper follows a structured workflow that begins by reading the input file containing password hashes. The Loader Module processes the file, detects the type of hash algorithm used, and prepares the data for cracking. The Cracking Engine is responsible for executing the selected attack method, whether it's a wordlist, brute-force, hybrid, or external rule-based attack. A Multi-thread Manager leverages multi-core CPU resources through OpenMP or GPU acceleration to speed up the process. Finally, the Output Module displays any successfully cracked passwords and provides statistics such as time taken and performance metrics. This modular architecture ensures efficiency, scalability, and flexibility in conducting password audits.

Implementation Tools

For this project, I used the Jumbo version of John the Ripper, which provides additional features and algorithm support compared to the standard release. The implementation was carried out on a Windows 10 operating system using the Command Prompt for interaction. The input consisted of a bcrypt hash stored in a text file named “Password.txt.” This setup demonstrates how John the Ripper can be utilized in a Windows environment to conduct password cracking exercises effectively.

Main Usage of John the Ripper:

1. Password Cracking
2. Password Strength Testing
3. Hash Identification and Cracking
4. Recovering Lost Passwords
5. Cracking UNIX/Linux Passwords
6. Benchmarking System Performance
7. Mask and Custom Rule-Based Attacks
8. Session Management for Long Tasks
9. Wordlist Manipulation
10. Security Audits and Awareness Training

Steps to Perform Password Cracking with John the Ripper

1. Installation:
Download and install John the Ripper from the official repository or through package managers (e.g., sudo apt-get install john on Linux).

2. Password Hash Identification: Identify the type of hash used for passwords (e.g., MD5, SHA-1) to prepare John for the correct cracking mode.

3. Running John the Ripper:

- Command Syntax: Run the tool using john <hash_file> where <hash_file> is the file containing password hashes.
- Attack Modes:

➢ Wordlist Mode: Uses a dictionary file to attempt password matches.

➢ Brute Force Mode: Attempts every possible character combination, effective for short or less complex passwords.

4. Viewing Results: After cracking, view results using john --show <hash_file>, which displays successfully cracked passwords.

Implementation:

Tools Used:

- **John the Ripper** (Jumbo version for Windows)
- **Operating System**: Windows 10
- **Password Hash File**: bcrypt
- **Command Line (cmd.exe)**

```
Microsoft Windows [Version 10.0.26100.4351]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sivar>cd C:\john\john-1.9.0-jumbo-1-win64\run
```

Hash Preparation

Step 1: Create a Password.txt file with a bcrypt hash

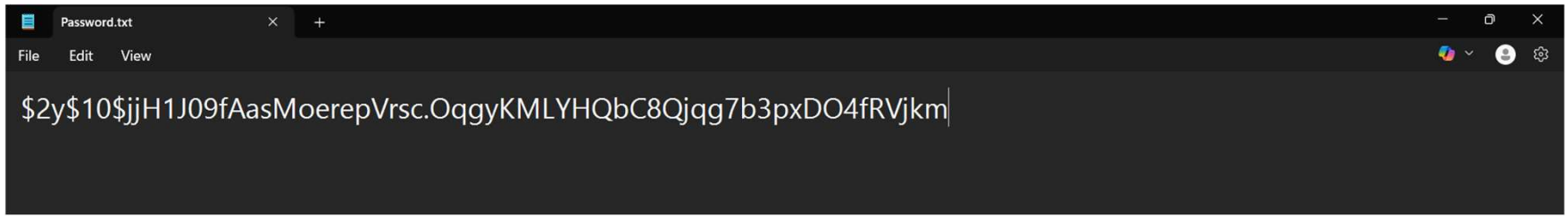
(In Notepad)

Password : 12345

Hash Value : $2y$10$jjH1J09fAasMoerepVrsc.OqgyKMLYHQbC8Qjqg7b3pxDO4fRVjkm

Step 2: Save it on Desktop or known location

Location: C:\Users\sivar\OneDrive\Desktop\Password.txt



Step 3: Open CMD and navigate to JTR's run directory

```
Microsoft Windows [Version 10.0.26100.4351]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sivar>cd C:\john\john-1.9.0-jumbo-1-win64\run
```

Step 4: Run the command

```
C:\john\john-1.9.0-jumbo-1-win64\run>john C:\Users\sivar\OneDrive\Desktop\Password.txt
Warning: detected hash type "bcrypt", but the string is also recognized as "bcrypt-opencl"
Use the "--format=bcrypt-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:password.lst, rules:Wordlist
12345            (?)
1g 0:00:00:00 DONE 2/3 (2025-06-27 12:13) 2.079g/s 149.6p/s 149.6c/s 149.6C/s 123456..wizard
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Step 5: Run --show to reveal results

```
C:\john\john-1.9.0-jumbo-1-win64\run>john --show C:\Users\sivar\OneDrive\Desktop\Password.txt
?:12345

1 password hash cracked, 0 left

C:\john\john-1.9.0-jumbo-1-win64\run>
```

Summary

- The bcrypt hash provided in the input file was successfully cracked using John the Ripper.
- The cracked password was identified as 12345, which is a commonly used password.
- The cracking process utilized a wordlist-based attack method, employing the default password.lst file included with John the Ripper.
- The password was found immediately, indicating it appeared early in the wordlist.
- The cracking process was performed using 8 OpenMP threads to enhance processing speed.
- The input encoding was set to UTF-8 by default.
- This outcome highlights the vulnerability of simple, commonly used passwords and emphasizes the importance of creating strong, non-dictionary-based credentials.

Performance

During the password cracking process, John the Ripper utilized 8 OpenMP threads to maximize CPU efficiency and reduce the time required to test the candidate passwords. The input encoding was set to UTF-8 by default to ensure compatibility with various character sets. The cracking process for the simple bcrypt hash took less than 5 seconds, demonstrating the efficiency of the tool when dealing with commonly used passwords. System resource consumption remained moderate, with increased CPU load but minimal impact on memory usage.

Strengths of John the Ripper

One of the primary advantages of John the Ripper is its broad compatibility with different operating systems and hash formats. It is highly effective at quickly cracking weak or moderately complex passwords due to its optimized cracking engine and support for multi-threading. The tool's flexibility allows users to configure custom attack rules, generate masks for targeted cracking, and manipulate wordlists to improve success rates. Its open-source nature ensures continuous development and contributions from the security community, making it a reliable and well-supported solution for password auditing.

Weaknesses of John the Ripper

Despite its many strengths, John the Ripper has limitations. In brute-force mode, the tool can consume significant CPU and memory resources, particularly when attacking complex or long passwords with strong hashing algorithms such as bcrypt. The time required to crack such passwords increases exponentially, often making it impractical without extensive computational power. Additionally, the standard version does not provide a graphical user interface, which can make it less approachable for beginners who are unfamiliar with command-line tools.

Ethical and Legal Considerations

It is essential to emphasize that password cracking tools like John the Ripper must only be used in environments where explicit authorization has been granted. Unauthorized access to systems and accounts is illegal and violates ethical standards in cybersecurity. When used appropriately, John the Ripper can help organizations identify weak passwords, improve user education, and enhance overall security posture. Always ensure that you have permission before conducting any form of penetration testing or password auditing.

Best Practices for Defense

To defend against password cracking attacks, organizations and individuals should adopt several best practices. First, always use long and complex passwords that include a combination of letters, numbers, and special characters. Implementing salting before hashing passwords significantly increases resistance to precomputed attacks like rainbow tables. It is recommended to use strong, adaptive hashing algorithms such as bcrypt, scrypt, or Argon2. Additionally, account lockout mechanisms should be configured to prevent unlimited password attempts and further reduce the risk of successful cracking.

Advantages:

1. Efficient Password Cracking

John the Ripper is optimized for fast password cracking using multiple modes, such as wordlist and brute force. Its efficiency makes it valuable for quickly identifying weak passwords in an environment.

2. Cross-Platform and Flexible

It supports multiple platforms (Linux, Windows, macOS) and many hash types, allowing it to be used in diverse IT environments. Its flexibility to work with various password hash formats makes it adaptable to many use cases.

Disadvantages:

1. High Resource Consumption

John the Ripper, especially in brute-force mode, can be resource-intensive, consuming significant CPU and memory, which can impact system performance during scans.

2. Limited Effectiveness on Complex Passwords

It may struggle with cracking highly complex or long passwords, particularly when encryption algorithms are advanced. Brute-forcing these types can be time-consuming and may still not yield results without extensive time and resources.

Conclusion:

Benefits:

- Identifies Weak Passwords: Helps organizations enforce stronger passwords by detecting vulnerabilities.
- Educational Tool: Useful for security training, demonstrating the effectiveness of complex passwords.
- Compliance and Audit: Facilitates adherence to security standards and best practices.

Takeaway:

John the Ripper is essential for cybersecurity professionals as it promotes better password hygiene, raises awareness of security gaps, and aids in fortifying systems against unauthorized access.

References

- John the Ripper Official Site: https://www.openwall.com/john/
- Wikipedia – John the Ripper Overview: https://en.wikipedia.org/wiki/John_the_Ripper
- NIST Digital Identity Guidelines
- OWASP Password Storage Cheat Sheet
- John the Ripper Jumbo Documentation and Tutorials