AIR UNIVERSITY

GROUP MEMBERS

- NOOR UL HASSAN
- MUHAMMAD-ASIM
- FARAN KHAN

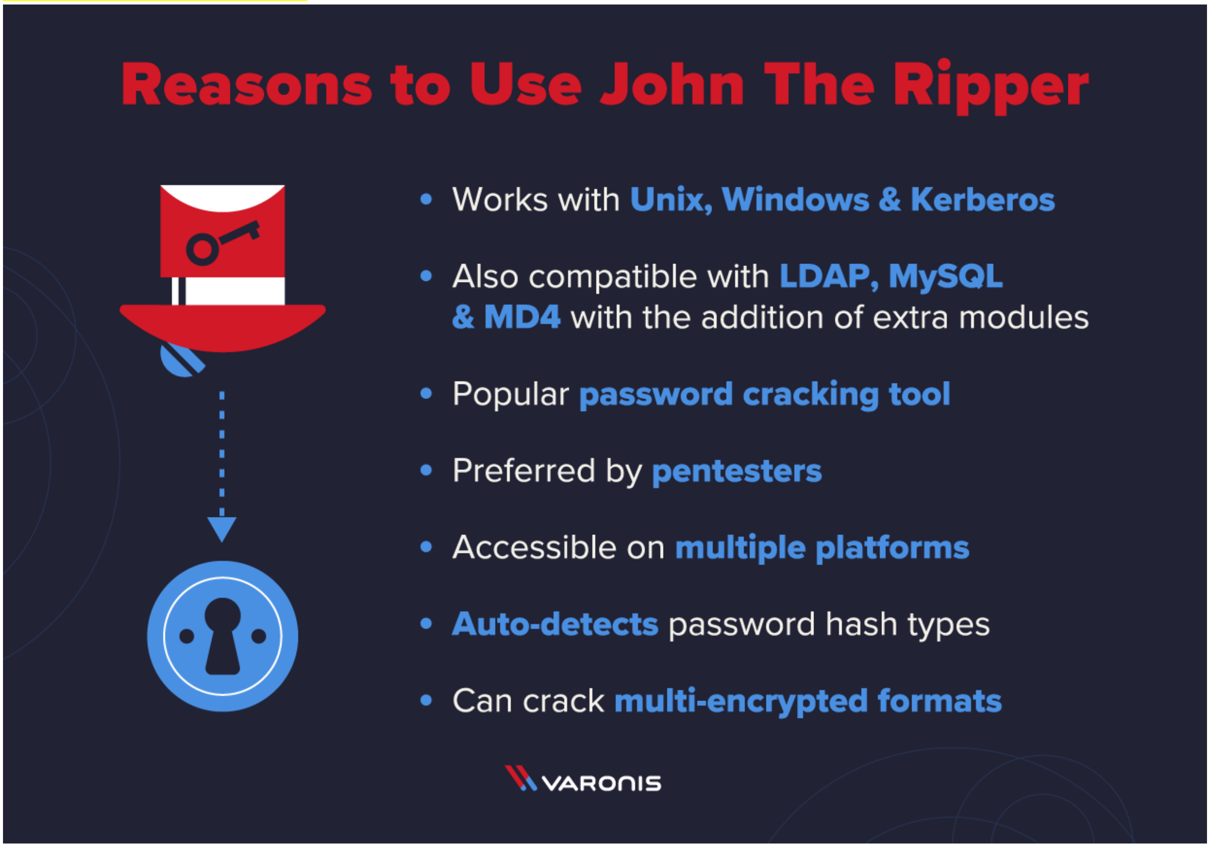
Contents

John the Ripper: Password Cracking Tool

1. Introduction:

John the Ripper is an open-source password cracking tool commonly utilized in information security. Its development targeted detecting weak passwords by combining variety of techniques under one strong package. It initially created for Unix-based systems, but with time it is now available for most systems and file formats.

2. Domain of Use:



Reasons to Use John The Ripper

- Works with **Unix, Windows & Kerberos**
- Also compatible with **LDAP, MySQL & MD4** with the addition of extra modules
- Popular **password cracking tool**
- Preferred by **pentesters**
- Accessible on **multiple platforms**
- **Auto-detects** password hash types
- Can crack **multi-encrypted formats**

John the Ripper primarily finds application in the domain of information security. This is due to the following reasons:

John the Ripper is a potent, open-source password cracker widely used in information security for several critical reasons, including:

1. Penetration Testing:

John the Ripper is a vital component in penetration testing, where security professionals assess the robustness of password policies and user credentials in organizations. By simulating an attacker's approach, it helps identify weak passwords that might compromise systems. This tool can perform dictionary attacks, brute force attacks, and smarter methods like rule-based attacks, allowing security teams to evaluate whether users adhere to strong password practices.

2. Digital Forensics:

In digital forensics, John the Ripper helps investigators recover lost or encrypted files that may prove to be critical in cases or during incident response. In case passwords have been applied to sensitive data, decryption of those files becomes a significant aspect of forensic analysis. By using its capabilities, forensic analysts can reveal information that would otherwise be impossible to access, thereby facilitating the examination of devices and systems involved in cyber incidents.

3. IT Security Audits:

Organizations carry out IT security audits as part of internal and external compliance with security policies. John the Ripper is a tool for auditing password management practices by checking how strong their password policies are. It indicates the number of time units required to crack a given password; it informs how weak a user selection is and the amount of change required in policies. This process not only helps in strengthening security measures but also raises awareness among employees about the importance of using strong passwords.

4. Data Breach Analysis:

After a data breach, organizations can use John the Ripper to verify whether the passwords that might have been compromised are actually strong. Security teams can scan the leaked password hashes against a database of known weak passwords to determine the risk the breach poses and take proactive steps to avoid the same in the future.

5. User Training and Awareness:

The organization can use John the Ripper to show how weak passwords can be cracked and make users understand the need to use complex passwords and change them periodically. This is an excellent hands-on way to demonstrate password vulnerabilities and encourages security awareness in the workplace.

6. Customization and Flexibility:

John the Ripper supports a number of hash algorithms and can be configured according to specific cracking needs. Its flexibility makes it an ideal tool for any given application, ranging from testing local system passwords to more complicated scenarios involving networked environments or cloud services. Its extension through plugins means that it can be easily tailored to help solve particular security problems that an organization may encounter.

3. John the Ripper in Real-World Applications Scenarios:

John the Ripper is widely deployed for many real-world applications of different industries and sectors. Some real-world applications scenarios our group had thought are as follows:

1. Corporate Security Assessments:

Scenario: A company has a mid-size number of employees and is afraid about employees using password practices. The vulnerability is assessed by running the IT security team against the user accounts with John the Ripper.

Application: The tool identifies weak passwords like "12345" or "password," making the organization implement stronger policies of passwords, such as increasing the length and strength of the password. Then comes a visible reduction of access attempts by unauthorized access.

2. Ethical Hacking Engagements:

Scenario: The scenario is that an information security firm has been engaged to perform a penetration test for a financial bank on its online banking portal for identifying potential vulnerabilities of that system.

Application: The ethical hackers utilize John the Ripper to test the strength of customers' passwords. Through simulated attacks, they discover weak credential vulnerabilities that malicious actors could exploit, and the bank can strengthen its security measures before launching.

3. Incident Response for Data Breaches:

Scenario: A healthcare provider suffers a data breach, and encrypted patient records are inaccessible.

Application: Incident response teams make use of John the Ripper to decrypt the credentials encrusted in these files. With the recovered lost files, the team assists an organization to determine the scale of the breach and therefore corrective measures can be deployed fast.

4. Colleges Universities

Scenario: A university runs a cybersecurity class teaching students how to cover and manage passwords.

Application: During lab sessions, students use John the Ripper to check the strength of test passwords and see how attackers use weak credentials. This will enable them to have a better understanding and be prepared for work in cybersecurity.

5. Adherence to Regulatory Requirements:

Case Study: A financial firm must comply with PCI-DSS standards regarding sensitive data protection.

Application: Internal auditors use John the Ripper to audit password security in networks. By detecting and correcting weak passwords, the organization will be compliant and avoid fines from regulatory bodies.

6. Password Recovery in Digital Forensics:

Scenario: A law enforcement agency is investigating a case involving encrypted communications on suspect devices.

Application: Forensic analysts use John the Ripper to recover passwords for many encrypted files, which helps to gather evidence for the case. Successful decryption of the credentials may lead to obtaining very important information related to the case.

7. Quality Assurance in Software Development:

Situation: A software company is developing a new application with user authentication features.

Application: During the QA phase, developers use John the Ripper to evaluate how well the application handles password security. This includes testing various password scenarios to ensure that the application can withstand typical password cracking methods.

8. Research and Development:

Scenario: Security researchers are studying the effectiveness of different hashing algorithms in protecting passwords.

Application: Researchers test the strength of different hash functions by using John the Ripper against them, thus contributing insights to the password security domain and helping the industry perfect its standards.

4. Technical Details:

John the Ripper works on multiple modes to crack passwords, including:

- **Dictionary Attacks:** Using predefined word lists to test possible passwords.
- **Brute-Force Attacks:** Systematically trying all combinations of characters.
- **Rule-Based Attacks:** Customizing attack patterns based on specific criteria.

5. Key Features:

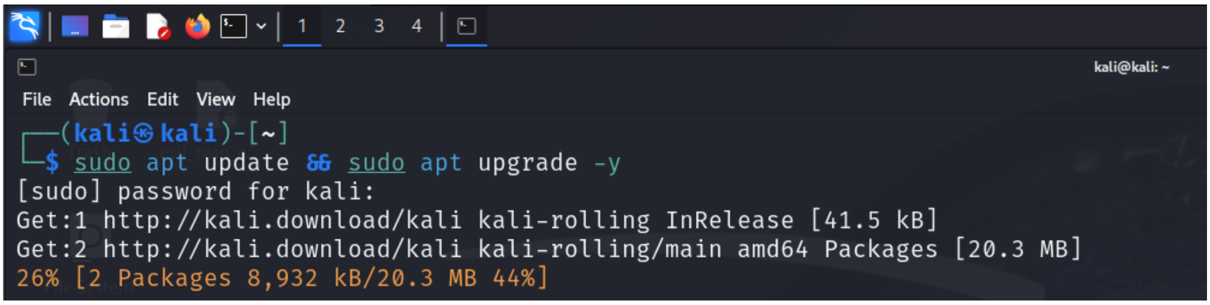
- **Extensive Format Support:** Can crack passwords from various file formats (e.g., ZIP, PDF, Shadow files).
- **Adaptable:** Users can create or adapt rules for specific requirements
- **Optimize Performance:** Utilizes all CPU and GPU power to give outcomes faster

6. Setting up the tool:

➢ On Kali Linux:

1. Update Your System:

```
sudo apt update && sudo apt upgrade -y
```

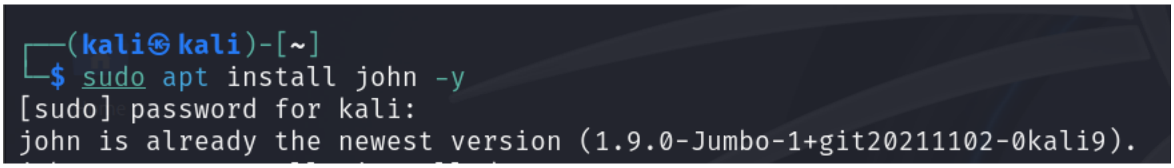


```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt upgrade -y
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.3 MB]
26% [2 Packages 8,932 kB/20.3 MB 44%]
```

2. Install John the Ripper:

Use the package manager to install the tool:

sudo apt install john -y



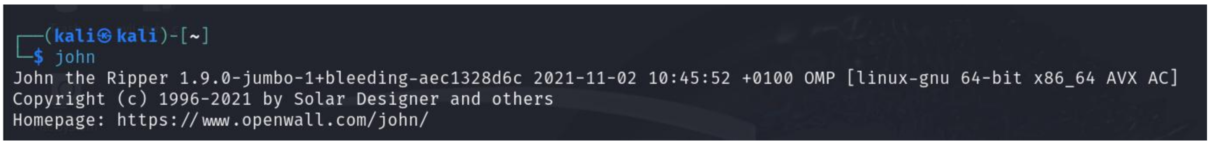
```
┌──(kali㉿kali)-[~]
└─$ sudo apt install john -y
[sudo] password for kali:
john is already the newest version (1.9.0-Jumbo-1+git20211102-0kali9).
```

Usually it is already installed in Kali.

3. Verify Installation:

Verify the version installed:

John



```
┌──(kali㉿kali)-[~]
└─$ john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 AVX AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/
```

4. Optional: Install the Jumbo Version:

Clone the Jumbo repository for extra functionality:

git clone https://github.com/openwall/john.git

cd john/src

./configure

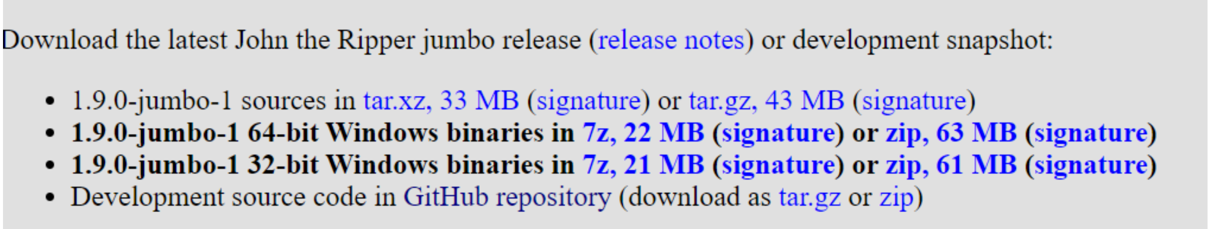
make -s clean && make -sj4

Run john from the run/ directory.

➢ On Windows

1. Download the Tool:

Go to the official download page (https://www.openwall.com/john/) and download the latest Windows binaries.

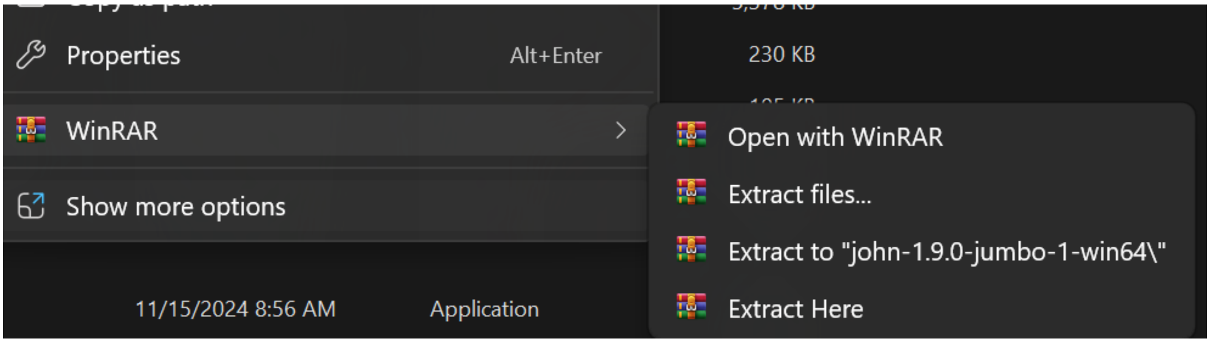


Download the latest John the Ripper jumbo release (release notes) or development snapshot:

- 1.9.0-jumbo-1 sources in tar.xz, 33 MB (signature) or tar.gz, 43 MB (signature)
- **1.9.0-jumbo-1 64-bit Windows binaries in 7z, 22 MB (signature) or zip, 63 MB (signature)**
- **1.9.0-jumbo-1 32-bit Windows binaries in 7z, 21 MB (signature) or zip, 61 MB (signature)**
- Development source code in GitHub repository (download as tar.gz or zip)

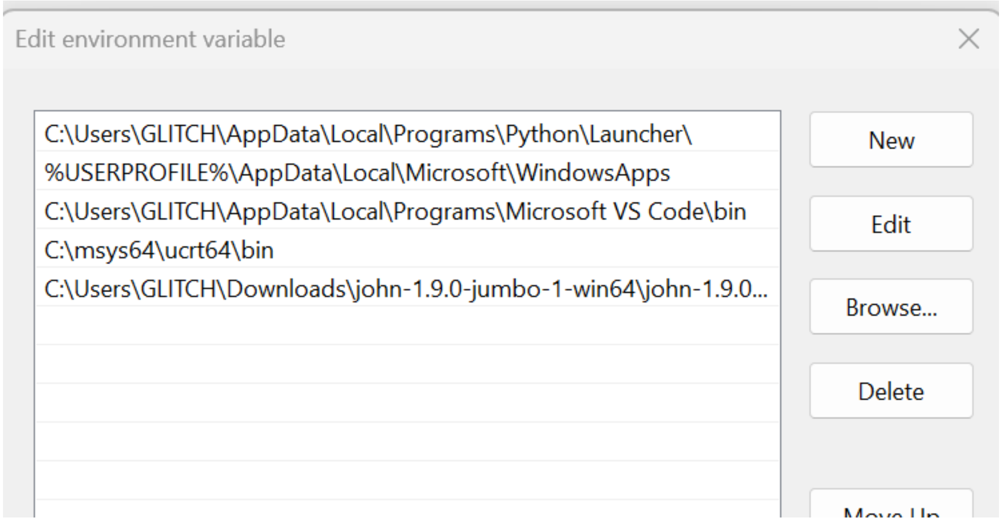
2. Extract the Files:

Unzip the downloaded file into a folder (e.g., C:\\JohnTheRipper).



3. Set Up the Environment:

Add the run directory to your system's PATH variable for easier access.



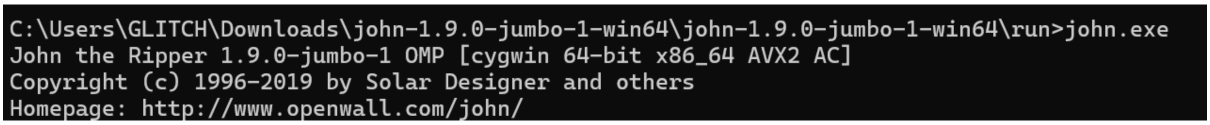
```
C:\Users\GLITCH\AppData\Local\Programs\Python\Launcher\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\GLITCH\AppData\Local\Programs\Microsoft VS Code\bin
C:\msys64\ucrt64\bin
C:\Users\GLITCH\Downloads\john-1.9.0-jumbo-1-win64\john-1.9.0...
```

4. Verify Installation:

Open Command Prompt and navigate to the run directory:

cd <Directory to the run folder>

john.exe



```
C:\Users\GLITCH\Downloads\john-1.9.0-jumbo-1-win64\john-1.9.0-jumbo-1-win64\run>john.exe
John the Ripper 1.9.0-jumbo-1 OMP [cygwin 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2019 by Solar Designer and others
Homepage: http://www.openwall.com/john/
```

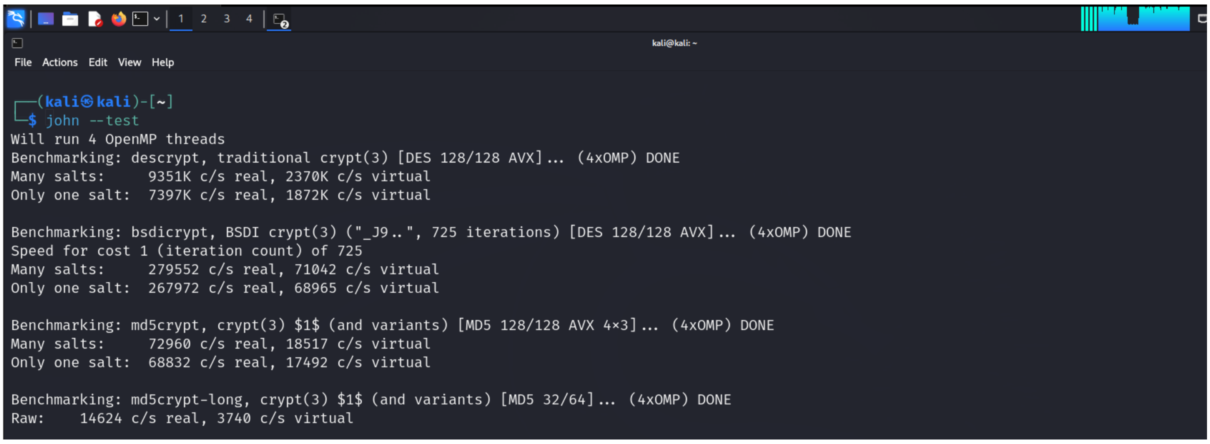
And the Tool is successfully installed.

➢ Common Post-Setup Steps

1. Test the Installation:

Run the sample hashes found in the run directory:

john –test



```
┌──(kali㉿kali)-[~]
└─$ john --test
Will run 4 OpenMP threads
Benchmarking: descrypt, traditional crypt(3) [DES 128/128 AVX]... (4xOMP) DONE
Many salts:     9351K c/s real, 2370K c/s virtual
Only one salt:  7397K c/s real, 1872K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 128/128 AVX]... (4xOMP) DONE
Speed for cost 1 (iteration count) of 725
Many salts:     279552 c/s real, 71042 c/s virtual
Only one salt:  267972 c/s real, 68965 c/s virtual

Benchmarking: md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4×3]... (4xOMP) DONE
Many salts:     72960 c/s real, 18517 c/s virtual
Only one salt:  68832 c/s real, 17492 c/s virtual

Benchmarking: md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64]... (4xOMP) DONE
Raw:    14624 c/s real, 3740 c/s virtual
```

2. Prepare Your Target File:

Make sure your password file is in a supported format. Use tools like unshadow (on Linux) to combine system files if necessary:

unshadow /etc/passwd /etc/shadow > combined.txt

3. Ready to Crack:

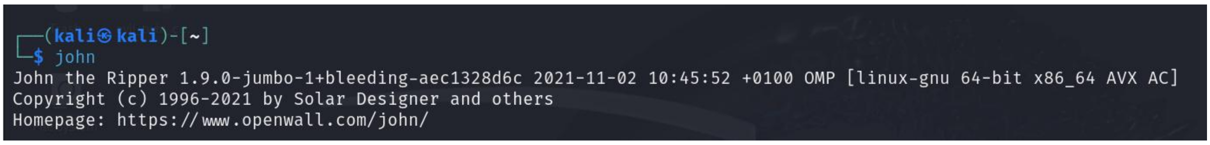
Begin to use John the Ripper with the chosen attack type (for example, dictionary or brute-force).

7. Running the tool:

1. Preparing the Environment:

Before running John the Ripper, make sure you have the following ready:

- John the Ripper installed (refer to the setup guide).
- A hash file containing the passwords to crack (e.g., hashes.txt).
- A wordlist file for dictionary attacks (e.g., rockyou.txt).



```
┌──(kali㉿kali)-[~]
└─$ john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 AVX AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/
```

2. Understanding Hash Files

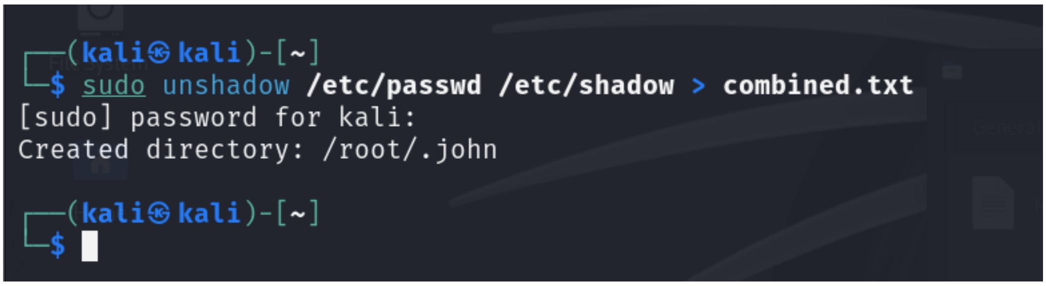
John the Ripper works with various hash formats. Common sources include:

- **/etc/shadow** and **/etc/passwd** files on Linux.
- Extracted hashes from password-protected files **(e.g., ZIP, PDF).**

Combining Shadow and Passwd Files:

On Linux systems:

unshadow /etc/passwd /etc/shadow > combined.txt



```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow /etc/passwd /etc/shadow > combined.txt
[sudo] password for kali:
Created directory: /root/.john

┌──(kali㉿kali)-[~]
└─$ █
```

The logic in using the "unshadow" command to combine **"/etc/passwd"** and **"/etc/shadow"** files is based on how user credentials are stored on Linux systems:

1. Separation of User Data:

- The "/etc/passwd" file stores user account information, such as usernames and user IDs, but not passwords.
- The "/etc/shadow" file stores the actual password hashes, but it is readable only by privileged users (e.g., root).

2. Unifying Data for Cracking:

- For tools like John the Ripper to work, the password hashes must be paired with the corresponding usernames and IDs to perform a meaningful crack.
- The "unshadow" command merges these two files into a single file ("combined.txt"), making it easier for the tool to process and test password guesses against hashes.

How It Works:

Input:

- "/etc/passwd": Offers user names, user IDs, and group IDs.
- "/etc/shadow": Offers the encrypted password hashes.

Output:

There should be one file: with every line containing information on one user, including the hash.

Why?

John needs a username plus the associated hash to

- Know which is being cracked.
- Output results in a human-readable format, such as username and cracked password.

3.1 Running John the Ripper:

1. Run the tool with a wordlist file (like rockyou.txt) to crack the hash:

```
john --wordlist=rockyou.txt hashes.txt
```

Explanation:

- --wordlist=rockyou.txt: This is the path to the wordlist file.
- hashes.txt: File containing hashes to crack.

3.2 Brute-Force Attack

If no wordlist is available, use an incremental mode for brute-force cracking:

```
john –incremental=Rule hashes.txt
```

Explanation:

- --incremental: Runs John the Ripper's built-in brute-force algorithm.
- Rule is optional

3.3 Resume Cracking

To resume a cracked session that was previously stopped:

```
john --restore
```

3.4 Listing Supported Hash Formats

To list all supported hash formats:

```
john --list=formats
```

3.5. Cracking Specific Hash Formats

If the hash format is known, use the --format option to identify it:

```
john --format=sha256 hashes.txt
```

Explanation:

- --format=sha256: Specifies the hash format.

4. Viewing Cracked Passwords

After John the Ripper finishes, print out the cracked passwords:

```
john –show –format=Raw-MD5 hashes.txt
```

Explanation:

- --show: Prints out the cracked password.

5. Generate Custom Wordlist

The john.conf rules can create custom wordlists in John the Ripper using the command:

```
john --wordlist=custom.txt –rules=ShiftToggle hashes.txt
```

6. Good Practices

Keep John the Ripper to those who have official permissions for authorized environment.

- First, test it on small datasets to understand its behavior.
- Optimize the performance using GPUs if available.

Conclusion:

John the Ripper is a powerful and versatile tool for password security analysis. Its ability to handle various file formats and implement diverse attack strategies makes it invaluable in cybersecurity. By using this tool, organizations and professionals can identify vulnerabilities, enhance security measures, and contribute to a safer digital environment.