Date: 11/02/2025

Lab Practical 10:

Perform password cracking concept using brute force tools- L0phtCrack and John the ripper.

➢ John the ripper:

John the Ripper (JtR) is an open-source password cracking tool designed for security auditing and penetration testing. It is used to test password strength by attempting to crack hashed or encrypted passwords using various attack methods such as dictionary attacks, brute force attacks, and hybrid attacks.

Developed by Openwall, John the Ripper supports multiple operating systems (Windows, Linux, macOS) and works with numerous password hash types (e.g., MD5, SHA, NTLM, bcrypt). It is widely used by cybersecurity professionals, ethical hackers, and penetration testers to identify weak passwords and enhance system security.

➢ It specializes in cracking password hashes using various attack methods, such as:

- Dictionary Attacks: Tries a list of common passwords and variations.
- Brute Force Attacks: Systematically tries all possible password combinations.
- Hybrid Attacks: Combines dictionary and brute force methods.
- Rainbow Table Attacks: Uses precomputed hash values to speed up cracking.

• Creaking MD5 password:

- Create an MD5 Hash File
  - **echo -n "Mohit" | md5sum**
  - using above command create multiple hashes and store it into on file.ex-**md5_hash.txt**

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ echo -n "Mohit" | md5sum
17893f3471ba48b1e9fb3abebf05c921  -
```

```
17893f3471ba48b1e9fb3abebf05c921
06a8314efaefacd775554831b29bfa8c
8036c7ec6415d8ae9c17fe319adcce0d
```

- Crack MD5 hashes stored in the file md5_hash.txt using the raw MD5 format using below command:
  - **john --format=raw-md5 md5_hash.txt**
    - **john :** Runs John the Ripper.

Date: 11/02/2025

- **--format=raw-md5 :** Specifies that the hashes in md5_hash.txt are in raw MD5 format (i.e., just plain MD5 hashes, without salt).
- **md5_hash.txt :** The file that contains the MD5 hashes to crack.

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ john --format=raw-md5 md5_Hash.txt

Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
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:/usr/share/john/password.lst
Proceeding with incremental:ASCII
Mohit            (?)
```

- **Creaking SHA256 password:**
- Create an SHA256 Hash File
  - **echo -n "abcde" | sha256sum**
  - using above command create multiple hashes and store it into on file.ex-**sha256_Hash.txt**

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ echo -n "abcde" | sha256sum
36bbe50ed96841d10443bcb670d6554f0a34b761be67ec9c4a8ad2c0c44ca42c  -
```

```
36bbe50ed96841d10443bcb670d6554f0a34b761be67ec9c4a8ad2c0c44ca42c
```

- Crack SHA256 hashes stored in the file sha256_Hash.txt using the raw SHA256 format using below command:
  - **john --format=raw-sha256 sha256_Hash.txt**
    - **john :** Runs John the Ripper.
    - **--format=raw-sha256 :** Specifies that the hashes in sha256_Hash.txt are unsalted SHA-256 hashes.
    - **sha256_Hash.txt :** The file that contains the SHA256 hashes to crack.

Date: 11/02/2025

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ john --format=raw-sha256 sha256_Hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 SSE2 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 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:/usr/share/john/password.lst
abcde            (?)
1g 0:00:00:00 DONE 2/3 (2025-02-16 11:27) 16.66g/s 546133p/s 546133c/s 546133C/s 123456..skyline!
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

- **Modes:**
- **Single Mode:**
  - The Single Crack Mode in John the Ripper is an intelligent password-cracking method. It uses information like the username, variations of the hash, and common patterns to guess the password.
  - Create a Hash File
  - Assume we have a user `john` with this **MD5 hash** of **`"Bhadra123"`**:
  - Now, save it in a file **with the username**:
  - **Run Single Mode Attack using below command:**
    - **john –single a1.txt –format=Raw-md5**
      - **john : Runs John the Ripper.**
      - **--single :** Uses Single Crack Mode (tries usernames and common variations).
      - **a1.txt :** The file containing MD5 hashes (ideally in username:hash format).
      - **--format=raw-md5 :** Tells John that the hashes are unsalted MD5.
        - John tries smart guesses like:
        - Bhadra
        - Bhadra1999
        - Bhadra@
        - Bhadra'sPassword

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ john --single FileA.txt --format=Raw-md5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 3 candidates buffered for the current salt, minimum 12 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE (2025-02-16 11:30) 0g/s 29766p/s 29766c/s 29766C/s bhadra1904..bhadra1900
Session completed.
```

Date: 11/02/2025

➢ Incremental Mode:

- The Incremental Mode in John the Ripper performs a brute force attack, trying all possible character combinations to crack a password. It is the most powerful but slowest attack mode, as it systematically generates and tests passwords of increasing length and complexity.
- Create a Hash File
- Let's assume we have an **SHA512 hash** of **"Bhadra@123 "**
- Save it to **incre_hash.txt**

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ echo -n "Bhadra@123" | sha256sum
d52965abf52d4c55ce56cd83d8191f5d0a6f2f7914189daa303616563l5961c6  -

┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ cat incre_hash.txt
d52965abf52d4c55ce56cd83d8191f5d0a6f2f7914189daa303616563l5961c6
```

➔ **Run Incremental Mode Attack using below command:**

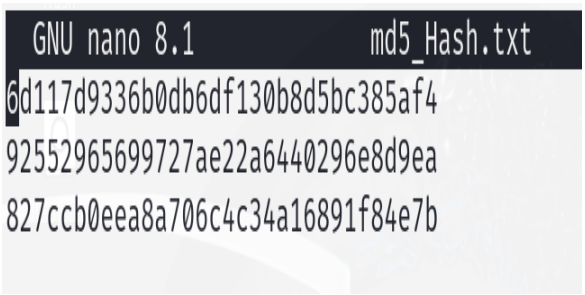
- **john –incremental:alpha incre_hash.txt –format=Raw-sha512**
- This command runs John the Ripper (JtR) in Incremental Mode, specifically targeting only alphabetic characters (a-z, A-Z), to crack raw SHA-512 hashes stored in incre_hash.txt.
  - **john :** Runs John the Ripper.
  - **--incremental:alpha :** Enables Incremental Mode but restricts it to only letters (no numbers or symbols).
  - **incre_hash.txt :** The file containing SHA-512 hashes to crack.
  - **--format=raw-sha512 :** Tells John that the hashes are unsalted SHA-512.
    - Tries all possible letter combinations in increasing length.
    - Starts with short, simple passwords, then progresses to longer ones.
    - Only letters (a-z, A-Z) are tested (no numbers or symbols).
    - Runs until the password is found or manually stopped.

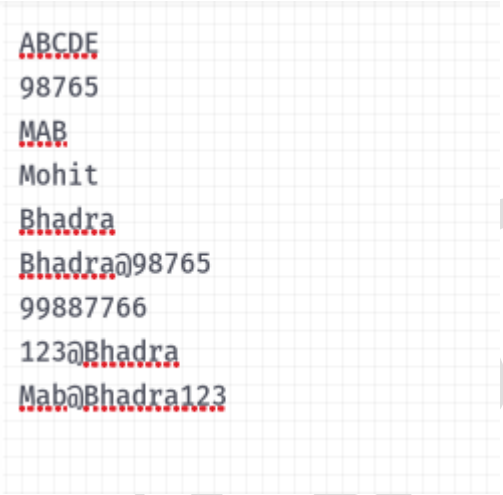
```
└─$ john --incremental:alpha incre_hash.txt --format=raw-sha512
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA512 [SHA512 128/128 SSE2 2x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
abc              (?)
1g 0:00:00:00 DONE (2025-02-15 02:36) 2.702g/s 55351p/s 55351c/s 55351C/s ami
sss..meliame
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Date: 11/02/2025

➢ Cracking MD5 password file using wordlist:

- First have you one file that contains multiple **MD5** hash.ex-**md5_hash.txt**
- Now create one **wordlist file** that contains possible password of hash.ex-**word.txt**





- **Run John the Ripper with the Wordlist using below command:**
  - **john --wordlist=word.txt --format=raw-md5 md5_hash.txt**
  - **john :** Runs John the Ripper.
  - **--wordlist=wordlist.txt :** Uses the specified wordlist (wordlist.txt) instead of brute **force.**
  - **--format=raw-md5 :** Specifies that the hashes are raw (unsalted) MD5.
  - **md5_hash.txt :** The file containing the MD5 hashes.

joh

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ john --wordlist=word.txt md5_Hash.txt --format=Raw-md5
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Remaining 2 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 9 candidates left, minimum 12 needed for performance.
0g 0:00:00:00 DONE (2025-02-16 12:48) 0g/s 450.0p/s 450.0c/s 900.0C/s ABCDE..Mab@Bhadra123
Session completed.
```

➢ Rules:

- **Rules** in **John the Ripper** allow you to modify and enhance wordlists dynamically, making password cracking more effective. These rules help **transform basic words** from a wordlist into more complex variations, mimicking real-world password habits.
- **Rule=ShiftToggle:**

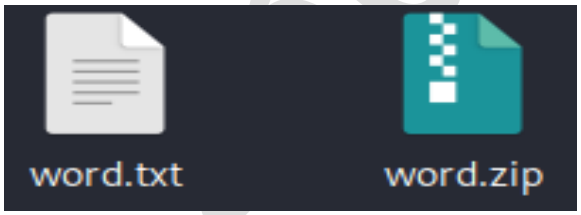
Date: 11/02/2025

- **ShiftToggle** is a **rule in John the Ripper** that **modifies capitalization** in a wordlist. It creates **different case variations** of words to improve password cracking efficiency.
- **john –rules=ShiftToggle –wordlist=word.txt –stdout | more**
  - **john :** Runs John the Ripper.
  - **--rules=ShiftToggle :** Uses the ShiftToggle rule (modifies words based on uppercase/lowercase shifts).
  - **--wordlist=word.txt :** Loads words from word.txt.
  - **--stdout :** Outputs the transformed words instead of using them for cracking.
  - **| more :** Paginate output (useful for long lists).

```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ john --rules=ShiftToggle --wordlist=word.txt --stdout |more
Using default input encoding: UTF-8
Press 'q' or Ctrl-C to abort, almost any other key for status
ABCDE
98765
MAB
Mohit
Bhadra
Bhadra@98765
99887766
123@Bhadra
Mab@Bhadra123
```

➢ Creaking Zip file password:

- John the Ripper can crack ZIP file passwords by using a brute-force attack or a wordlist attack. However, it first requires extracting the hash from the ZIP file before running the attack.
- Create one zip file with password for secuity.ex-**word.zip.**




```
┌──(mohit㉿kali)-[~/Documents/JohnTheRipper]
└─$ ls
FileA.txt  hash.txt  incre_hash.txt  md5_Hash.txt  sha216_Hash.txt  sha256_Hash.txt  word.txt  word.zip
```

Date: 11/02/2025

- **Extract ZIP Hash**
  - Before cracking, you need to extract the hash from the ZIP file using zip2john.

```
└─$ zip2john word.zip > zipHash.txt
```

- **Cracking zip file password using below command:**
  - **john zipHash.txt –wordlist=/usr/share/wordlists/rockyou.txt**
    - **john :** Runs John the Ripper.
    - **zipHash.txt :** The file that contains the extracted ZIP hash (from **zip2john).**
    - **--wordlist=/usr/share/wordlists/rockyou.txt :** Uses RockYou.txt as a password list.

```
└─$ john zipHash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 128/128 SSE2 4x])
Cost 1 (HMAC size) is 56 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
12345            (word.zip/word.txt)
1g 0:00:00:00 DONE (2025-02-15 03:56) 4.347g/s 35617p/s 35617c/s 35617C
/s 123456..whitetiger
Use the "--show" option to display all of the cracked passwords reliabl
y
Session completed.
```