

Task 4: The Key Under the Mat - Basic Password Cracking

Name :- Rushikesh Sanjay Kumavat

INTERN ID:- SN1000726

DOMAIN:- Cyber Security

Aim :-

To understand password security by cracking a hashed password using John the Ripper and a dictionary attack.

OBJECTIVE :-

- Generate a password hash
- Store it in a file
- Use John the Ripper with rockyou wordlist
- Crack the password
- Analyze the weakness of weak passwords

Tools Used :-

- Kali Linux
- Metasploitable 2
- John the Ripper

Procedure :-

Step 1: Generate Password Hash

Command used:

➔ `openssl passwd -1 password123`

Output:

➔ `1oR3nThsQ$/bmUQic3Q8BNFzJbeRYki.`

Step 2: Save Hash in File

Command:

→ nano hash.txt

Paste the hash and save.

Kali Linux 2.0.0 - Kali 2020.2 (64-bit) - 2020-02-12 22:42:43 UTC

File Actions Edit View Help

```
[kali㉿kali] ~
$ openstl passwd -l password123
$ $ORIGIN/htsql?http://10.10.10.10:8080/htpasswd?&password=password123
[kali㉿kali] ~
$ nano hash.txt
[kali㉿kali] ~
$ grip -o /usr/share/wordlists/rockyou.txt.gz
gzip: /usr/share/wordlists/rockyou.txt.gz: No such file or directory
[kali㉿kali] ~
$ ls /usr/share/wordlists/
amass dirb dirbuster dnsmap.txt fasttrack.txt fern-wifi John.lst Legion metasploit nmap.1st rockyou.txt sqlmap.txt wfuzz wifite.txt
[kali㉿kali] ~
$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Warning: detected hash type "md5Crypt", but the string is also recognized as "md5crypt-long".
Use the "--format=md5crypt-long" option to force loading these as that type instead.
Using default character encoding: UTF-8
Loaded 1 password hash in 0.002s (320K/s)
John 1 password hash(s) md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4x3]
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[1g] 0% 00:00:00 DONE (2020-02-12 22:42:42) 50.00g/s 76800g/s 76800c/s teacher..mexicali
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

[kali㉿kali] ~
$ john --show hash.txt
[1g] 0% 00:00:00 DONE (2020-02-12 22:42:42) 50.00g/s 76800g/s 76800c/s teacher..mexicali
You can manually add items by right-clicking on them, hit Esc to exit and viewing files etc.
You can also add selected items by double-clicking on them.

[kali㉿kali] ~
$ ./findfuzz.py Found:
Meeting-Ans-Configuring-Header
Server uses a connection-based identifier
Delayed header insertion alternative via server
HTTP/2 Connection Headers (read, writing)
```

Step 3: Verify Wordlist Location

Command:

→ ls /usr/share/wordlists/

Confirmed:

→ rockyou.txt

Step 4: Perform Dictionary Attack

Command:

```
→ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

John loaded hash type:

→ md5crypt

Password Cracked:

→ password123

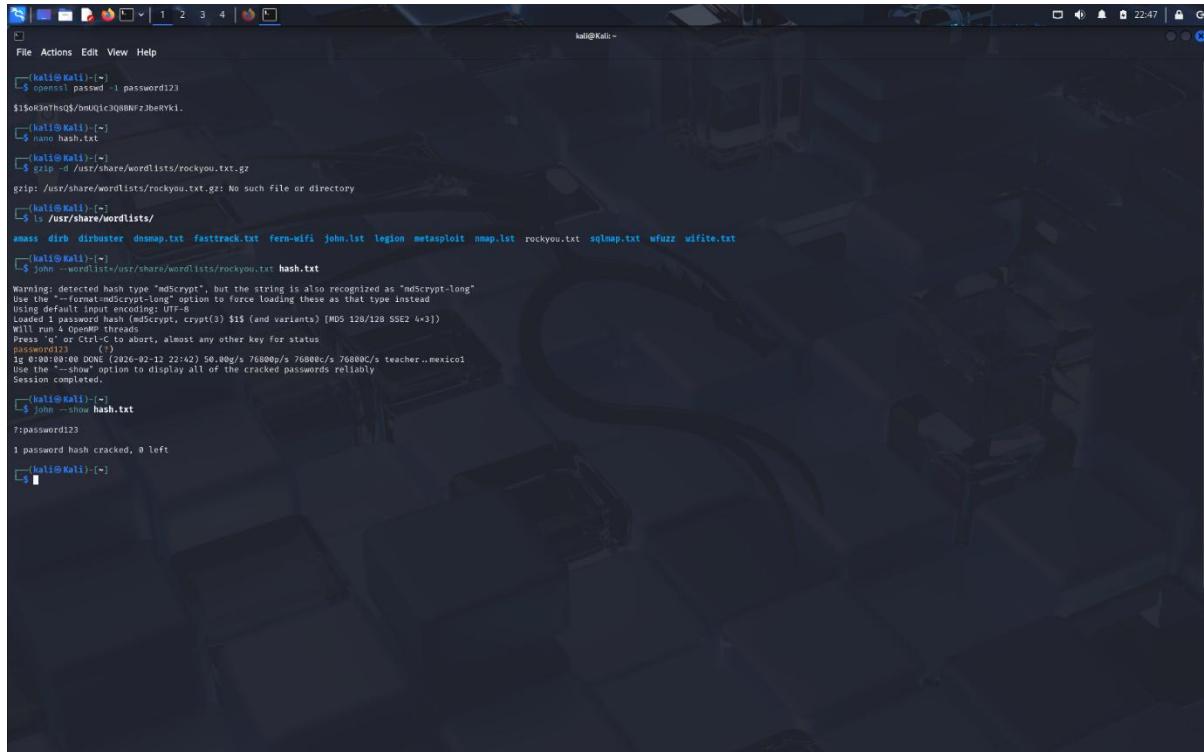
Step 5: Display Cracked Password

Command:

→ john --show hash.txt

Output:

→ password123



```
(kali㉿Kali)-[~]
$ ./john --wordlist=/usr/share/wordlists/rockyou.txt.gz
john: warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4x3])
Will attempt to guess the key until success (or Ctrl-C to abort, almost any other key for status)
Press 'q' or Ctrl-C to abort, almost any other key for status
password123
[...]
Session completed.

(kali㉿Kali)-[~]
$ ./john --show hash.txt
?password123
1 password hash cracked, 0 left
(kali㉿Kali)-[~]
```

RESULT :-

The password hash was successfully cracked using a dictionary attack.
The original password was:

→ password123

CONCLUSION :-

This task demonstrates:

- Weak passwords can be cracked quickly
- Dictionary attacks are very effective
- Proper password policies are necessary
- Hashing alone is not enough without strong passwords