# **FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING**

**Department of Computer Engineering**

# **Course, Subject & Experiment Details**

|  |  |
| --- | --- |
| **Practical No:** |  |
| **Title:** | **Hashcat** |
| **Name of the Student:** | **Warren Fernandes** |
| **Roll No:** | **8940** |
| **Date of Performance:** | **21-03-2022** |
| **Date of Submission:** | **04-04-2022** |

Evaluation:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Rubric** | **Grade** |
| **1** | **On time submission/completion (2)** |  |
| **2** | **Preparedness (2)** |  |
| **3** | **Skill (4)** |  |
| **4** | **Output (2)** |  |

# **Signature of the Teacher**

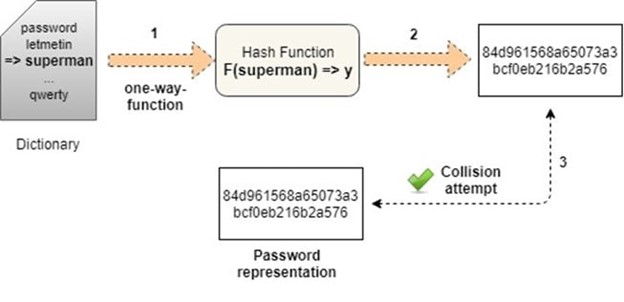
**Hashcat:**

Hashcat is a popular password cracker and designed to break even the most complex passwords representation. To do this, it enables the cracking of a specific password in multiple ways, combined with versatility and speed. Hashcat turns readable data into a garbled state (this is a random string of fixed-length size). Hashes do not allow someone to decrypt data with a specific key, as standard encryption protocols allow. Hashcat uses precomputed dictionaries, rainbow tables and even brute-force approaches to find an effective and efficient way to crack passwords.

#### How to crack hashes:

The simplest way to crack a hash is to try first to guess the password. Each attempt is hashed and then is compared to the actual hashed value to see if they are the same, but the process can take a long time

Dictionary and brute-force attacks are the most common ways of guessing passwords. These techniques make use of a file that contains words, phrases, common passwords and other strings that are likely to be used as a viable password.



It should be noted that there is no guaranteed way to prevent dictionary attacks or brute-force attacks.

Other approaches used to crack passwords:

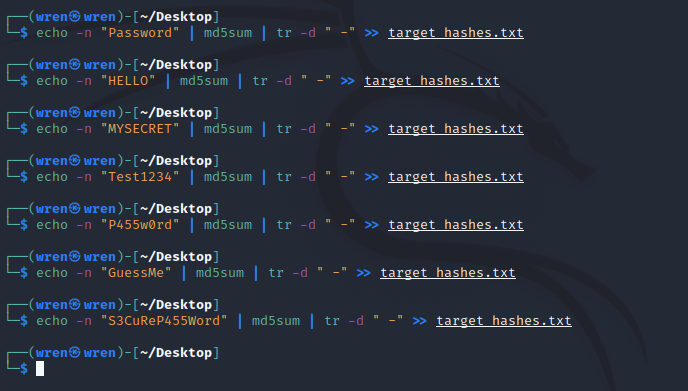
1. Lookup tables: Hashes are pre-computed from a dictionary and then stored with their corresponding password into a lookup table structure.
2. Reverse lookup tables: This attack allows for a cyber attacker to apply a dictionary or brute-force attack to many hashes at the same time without having to pre-compute a lookup table.
3. Rainbow tables: Rainbow tables are a time-memory technique. They are similar to lookup tables, except that they sacrifice hash cracking speed to make the lookup tables smaller.
4. Hashing with salt: With this technique, the hashes are randomized by appending or prepending a random string, called a “salt.” This is applied to the password before hashing.

#### Implementation:

1. Create a dictionary with MBD5 hashes

To start this demonstration, we will create multiple hash entries containing several passwords.

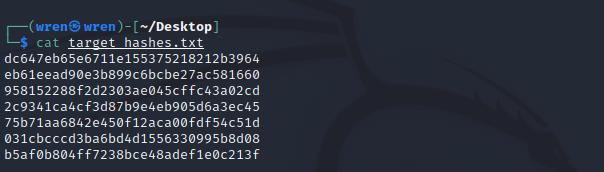
In detail, they will then be outputted to a file called “target\_hashes.” Each command should be executed in the terminal, as demonstrated below:



The -n option removes the new line added to the end of “Password.” This is important as we don’t want the new line characters to be hashed with our password. The part “tr –d ‘ -‘ “ removes any characters that are a space or hyphen from the output.

2. Check password hashes

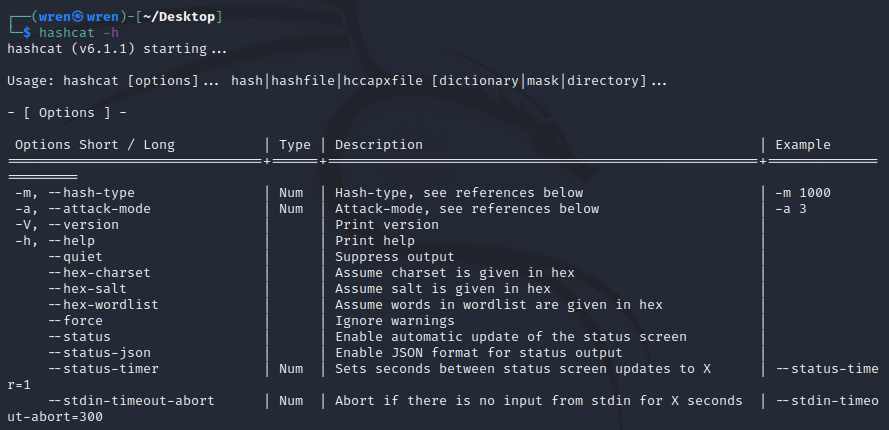
To do this, we need to type the following command line in the terminal:



3. Start Hashcat in Kali Linux

Hashcat can be started on the Kali console with the following command line: hashcat -h.

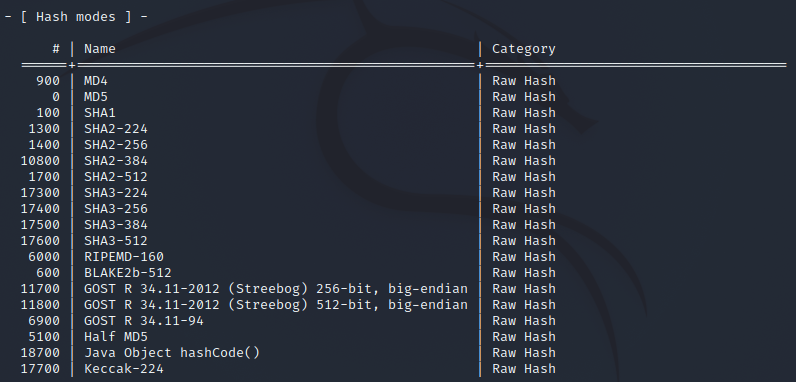
This is illustrated in the screenshot below:



Some of the most important hashcat options are -m (the hashtype) and -a (attack mode). In general, we need to use both options in most password-cracking attempts when using Hashcat.

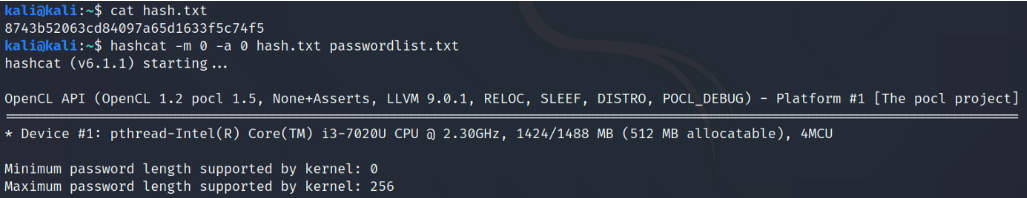
Hashcat also has specifically designed rules to use on a wordlist file. The character list can be customized to crack the password(s).

Finally, Hashcat provides numerous options for password hashes that can be cracked. This can be seen in the screenshot below:



4. Cracking the hashes

In the final step, we can now start cracking the hashes contained in the target\_hashes.txt file. We will use the following command line, as illustrated below:



-m 0 designates the type of hash we are cracking (MD5)

-a 0 designates a dictionary attack

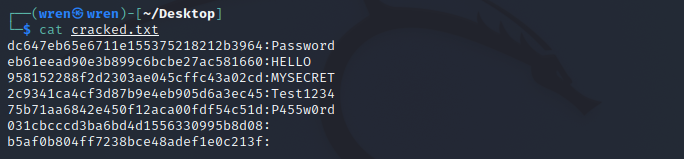
-o cracked.txt is the output file for the cracked passwords

target\_hashes.txt is our input file of hashes

/usr/share/wordlists/rockyou.txt is the absolute path to the wordlist file for this dictionary attack

6. Results

Finally, we have cracked five out of seven target hashes that were initially proposed. These can be seen below:



#### Postlab:

1. Describe about Hashcat and John the ripper?

## Hashat is a particularly fast, efficient, and versatile hacking tool that assists brute-force attacks by conducting them with hash values of passwords that the tool is guessing or applying. When used for benign purposes, such as in penetration testing one’s own infrastructure, it can reveal compromised or easy to guess credentials

## Hashcat is, however, better known for being used for nefarious purposes. Hackers use Hashcat, readily available for download on all major operating systems, to automate attacks against passwords and other shared secrets. It gives the user the ability to brute-force credential stores using known hashes, to conduct dictionary attacks and rainbow tables, and to reverse engineer readable information on user behavior into hashed-password combination attacks.

John the Ripper is a free password cracking software tool. It was designed to test password strength, brute-force encrypted (hashed) passwords, and crack passwords via dictionary attacks. John the Ripper is a part of the Rapid7 family of penetration testing/ hacking tools. Also, John is already installed on Kali Linux.