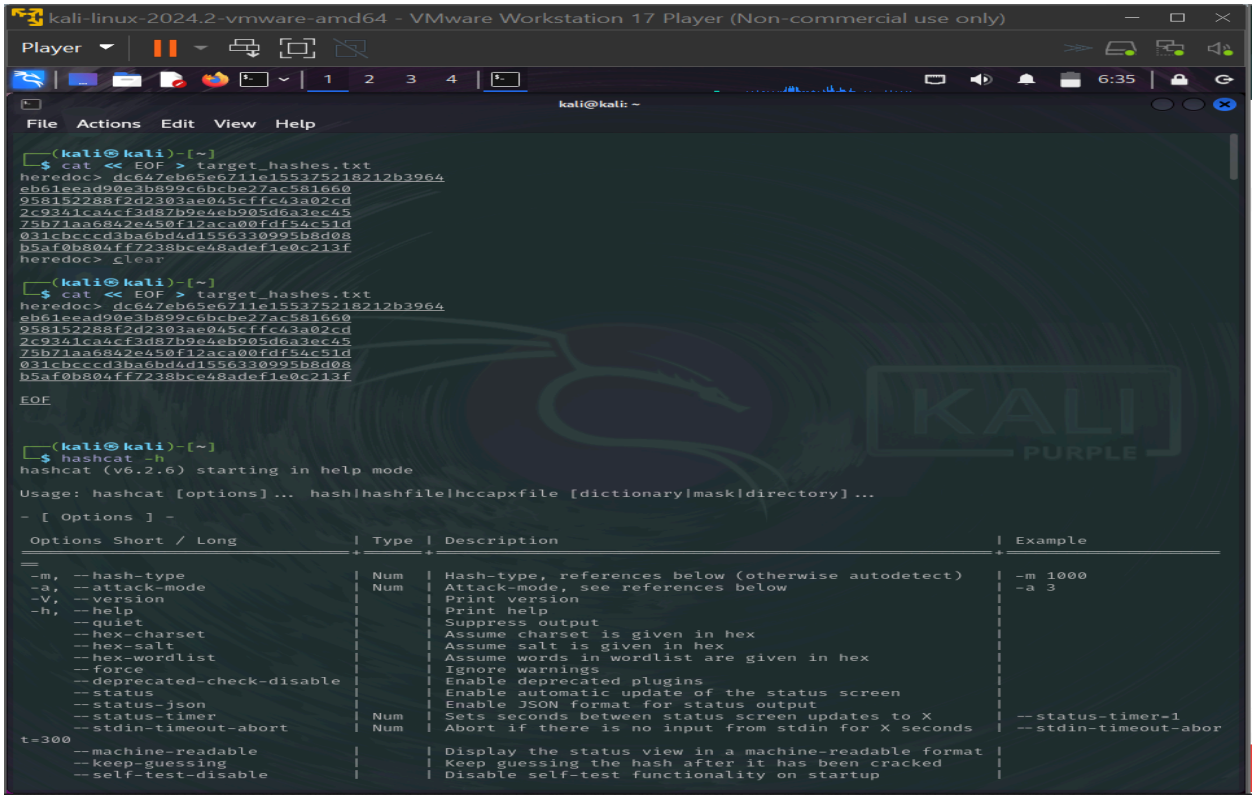
CRACKING A PASSWORD WITH HASHCAT

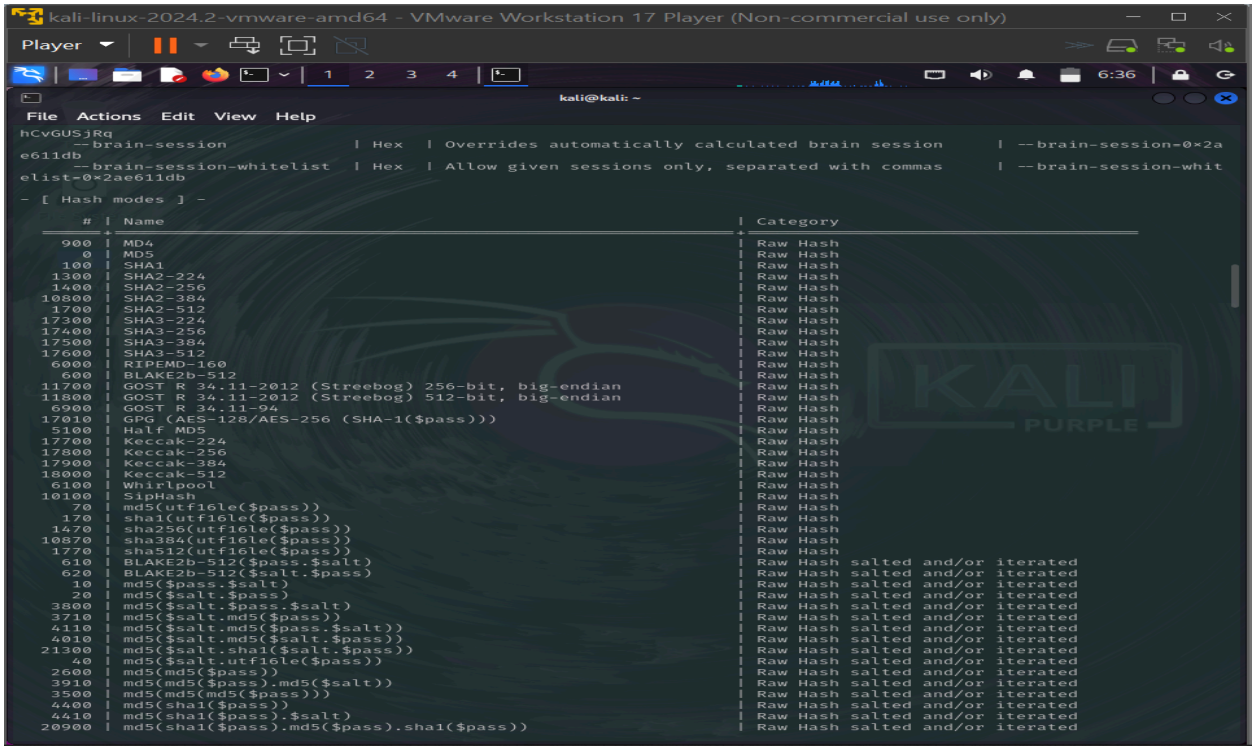
Tools : KALI LINUX

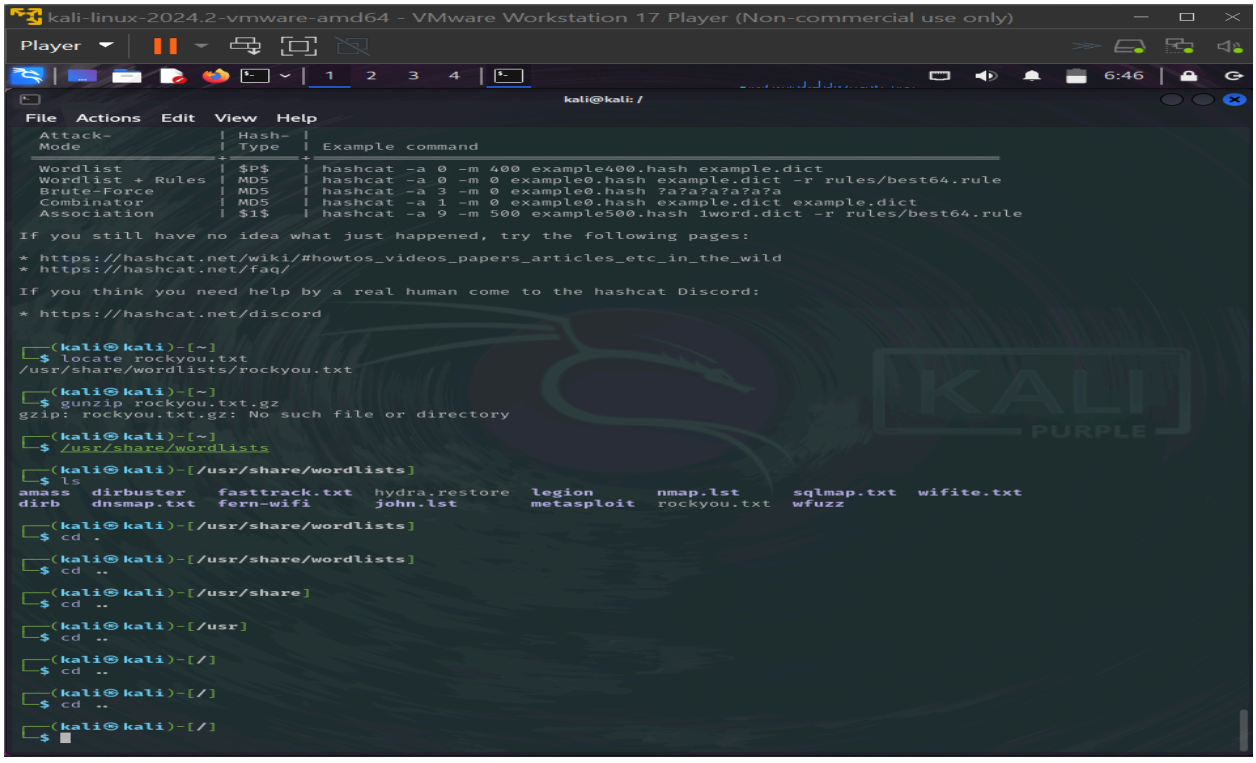
Hashcat is a popular password cracking tool used for cracking various types of password hashes. It supports multiple attack modes, including dictionary attacks, brute force, and rule-based attacks, making it very versatile. Hashcat is known for its speed and efficiency, often leveraging the power of GPUs to accelerate cracking processes.

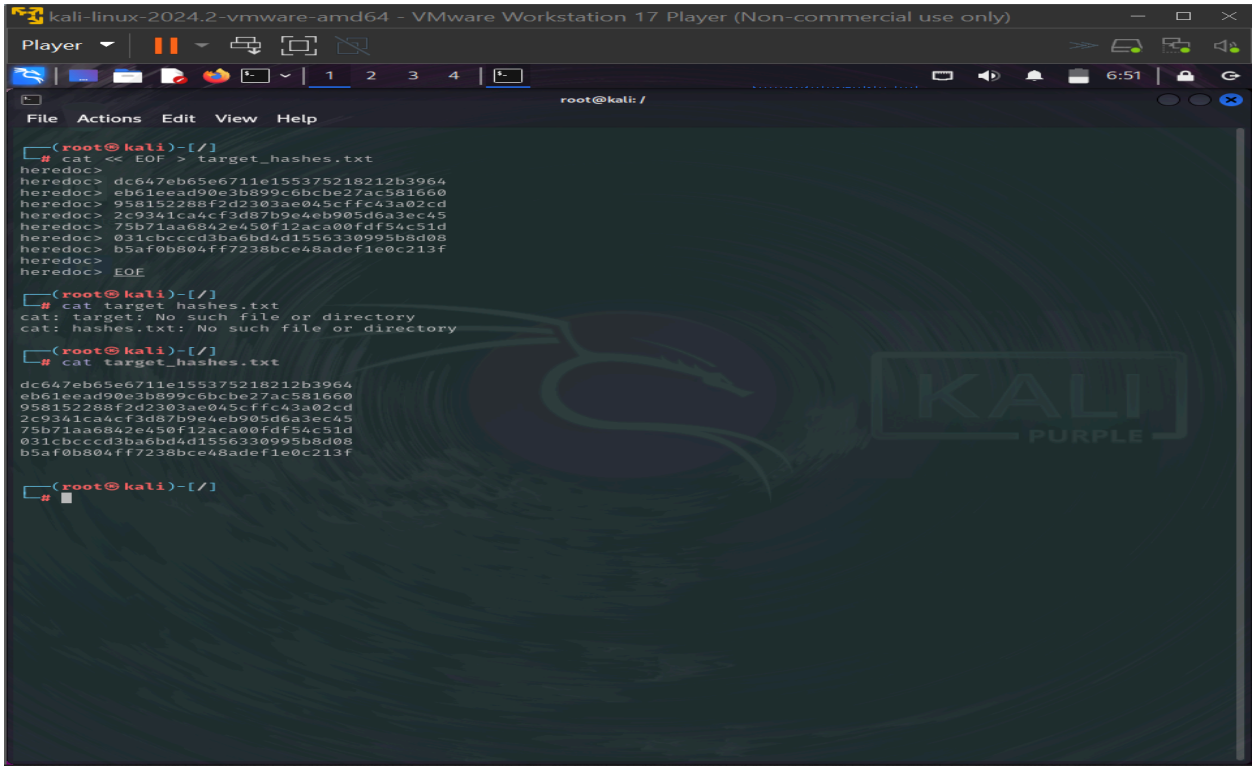
It supports a wide variety of hash algorithms like MD5, SHA-1, SHA-256, and more, and is commonly used for security testing, penetration testing, and recovering lost passwords. It can be used on multiple platforms, including Windows, Linux, and macOS.

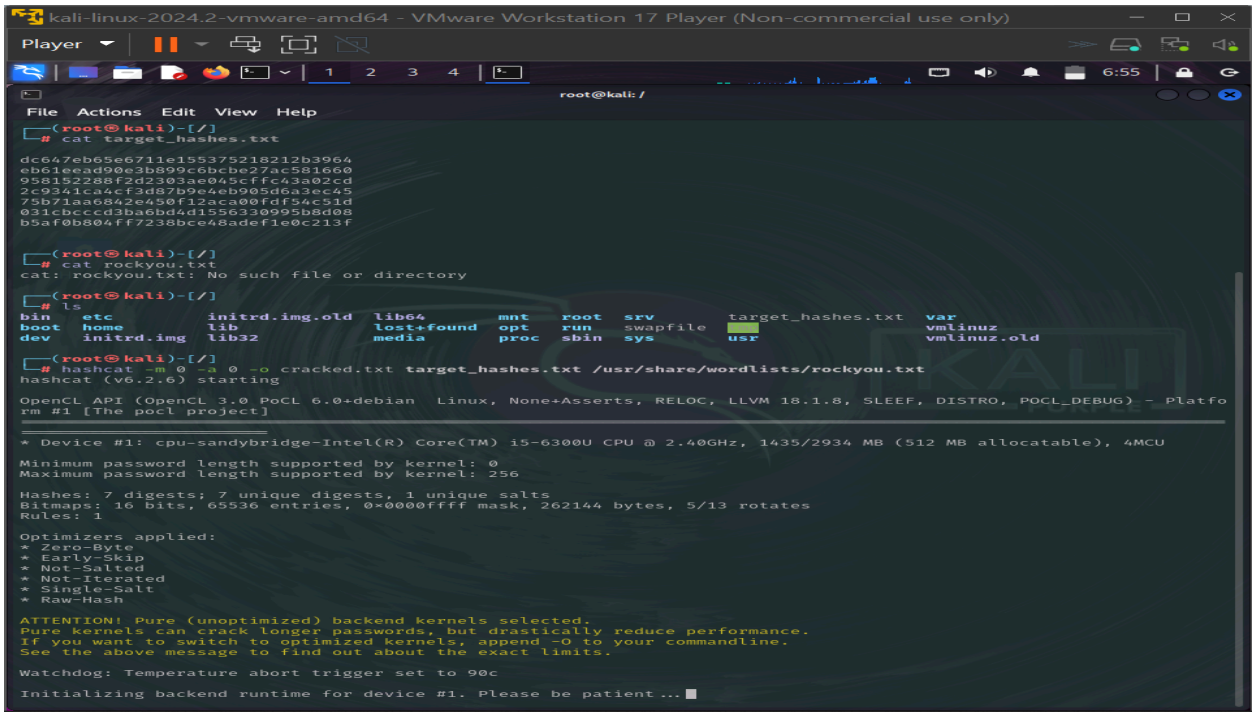
Input from kali :

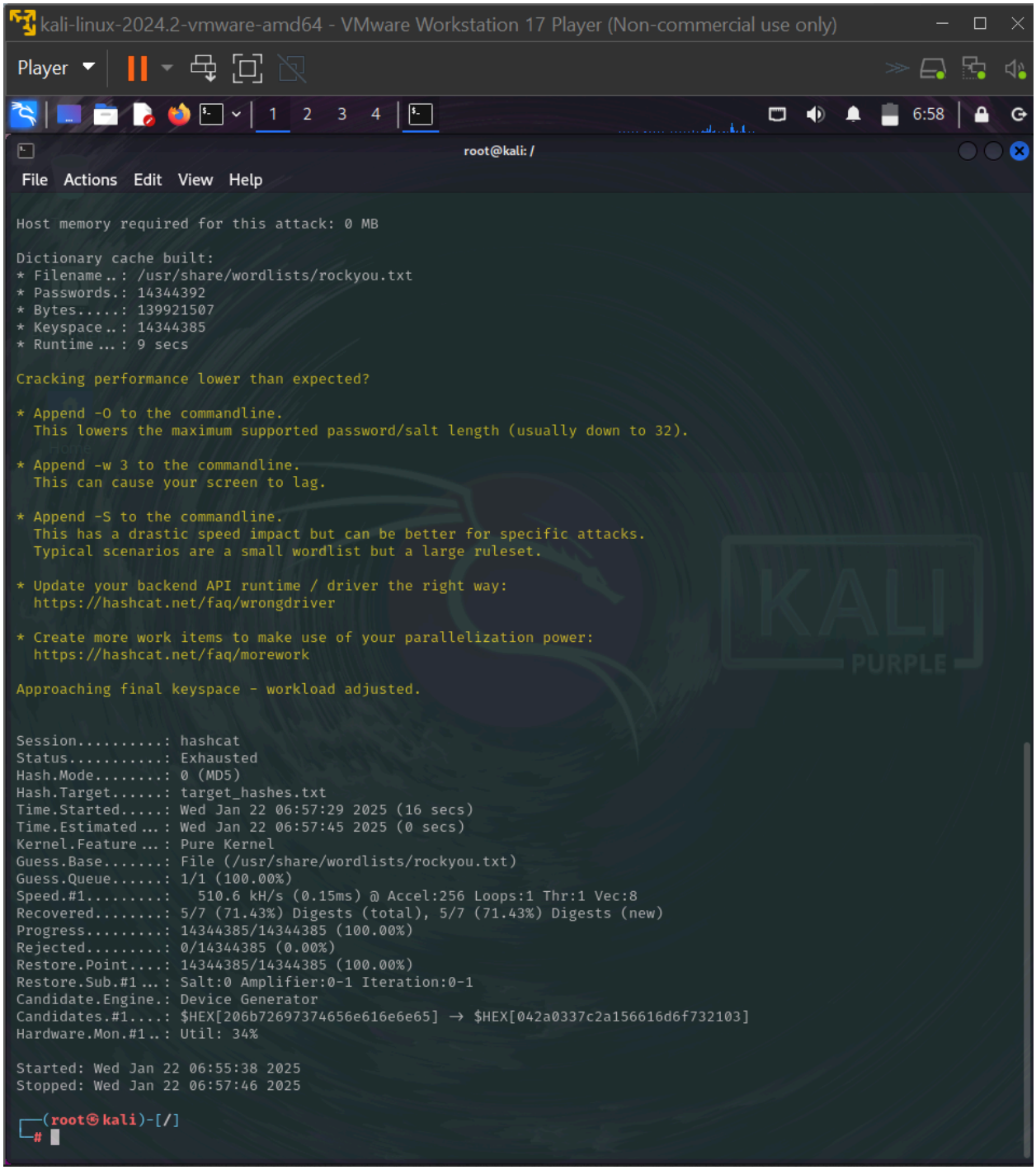



kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
6:36
kali@kali: ~
File Actions Edit View Help
hCvGUSjRq
--brain-session | Hex | Overrides automatically calculated brain session | --brain-session-0×2a
e611db
--brain-session-whitelist | Hex | Allow given sessions only, separated with commas | --brain-session-whit
elist-0×2ae611db
- [Hash modes] -
| Name | Category
900 | MD4 | Raw Hash
0 | MD5 | Raw Hash
100 | SHA1 | Raw Hash
1300 | SHA2-224 | Raw Hash
1400 | SHA2-256 | Raw Hash
10800 | SHA2-384 | Raw Hash
1700 | SHA2-512 | Raw Hash
17300 | SHA3-224 | Raw Hash
17400 | SHA3-256 | Raw Hash
17500 | SHA3-384 | Raw Hash
17600 | SHA3-512 | Raw Hash
6000 | RIPEMD-160 | Raw Hash
600 | BLAKE2b-512 | Raw Hash
11700 | GOST R 34.11-2012 (Streebog) 256-bit, big-endian | Raw Hash
11800 | GOST R 34.11-2012 (Streebog) 512-bit, big-endian | Raw Hash
6900 | GOST R 34.11-94 | Raw Hash
17010 | GPG (AES-128/AES-256 (SHA-1($pass))) | Raw Hash
5100 | Half MD5 | Raw Hash
17700 | Keccak-224 | Raw Hash
17800 | Keccak-256 | Raw Hash
17900 | Keccak-384 | Raw Hash
18000 | Keccak-512 | Raw Hash
6100 | Whirlpool | Raw Hash
10100 | SipHash | Raw Hash
70 | md5(utf16le($pass)) | Raw Hash
170 | sha1(utf16le($pass)) | Raw Hash
1470 | sha256(utf16le($pass)) | Raw Hash
10870 | sha384(utf16le($pass)) | Raw Hash
1770 | sha512(utf16le($pass)) | Raw Hash
610 | BLAKE2b-512($pass.$salt) | Raw Hash salted and/or iterated
620 | BLAKE2b-512($salt.$pass) | Raw Hash salted and/or iterated
10 | md5($pass.$salt) | Raw Hash salted and/or iterated
20 | md5($salt.$pass) | Raw Hash salted and/or iterated
3800 | md5($salt.$pass.$salt) | Raw Hash salted and/or iterated
3710 | md5($salt.md5($pass)) | Raw Hash salted and/or iterated
4110 | md5($salt.md5($pass.$salt)) | Raw Hash salted and/or iterated
4010 | md5($salt.md5($salt.$pass)) | Raw Hash salted and/or iterated
21300 | md5($salt.sha1($salt.$pass)) | Raw Hash salted and/or iterated
40 | md5($salt.utf16le($pass)) | Raw Hash salted and/or iterated
2600 | md5(md5($pass)) | Raw Hash salted and/or iterated
3910 | md5(md5($pass).md5($salt)) | Raw Hash salted and/or iterated
3500 | md5(md5(md5($pass))) | Raw Hash salted and/or iterated
4400 | md5(sha1($pass)) | Raw Hash salted and/or iterated
4410 | md5(sha1($pass).$salt) | Raw Hash salted and/or iterated
20900 | md5(sha1($pass).md5($pass).sha1($pass)) | Raw Hash salted and/or iterated


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
6:46
kali@kali: /
File Actions Edit View Help
Attack-Mode | Hash-Type | Example command
Wordlist | P | hashcat -a 0 -m 400 example400.hash example.dict
Wordlist + Rules | MD5 | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force | MD5 | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
Combinator | MD5 | hashcat -a 1 -m 0 example0.hash example.dict example.dict
Association | 1 | hashcat -a 9 -m 500 example500.hash 1word.dict -r rules/best64.rule
If you still have no idea what just happened, try the following pages:
* https://hashcat.net/wiki/#howtos_videos_papers_articles_etc_in_the_wild
* https://hashcat.net/faq/
If you think you need help by a real human come to the hashcat Discord:
* https://hashcat.net/discord
┌──(kali㉿kali)-[~]
└─$ locate rockyou.txt
/usr/share/wordlists/rockyou.txt
┌──(kali㉿kali)-[~]
└─$ gunzip rockyou.txt.gz
gzip: rockyou.txt.gz: No such file or directory
┌──(kali㉿kali)-[~]
└─$ /usr/share/wordlists
┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ ls
amass dirbuster fasttrack.txt hydra.restore legion nmap.lst sqlmap.txt wifite.txt
dirb dnsmap.txt fern-wifi john.lst metasploit rockyou.txt wfuzz
┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ cd .
┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ cd ..
┌──(kali㉿kali)-[/usr/share]
└─$ cd ..
┌──(kali㉿kali)-[/usr]
└─$ cd ..
┌──(kali㉿kali)-[/]
└─$ cd ..
┌──(kali㉿kali)-[/]
└─$ cd ..
┌──(kali㉿kali)-[/]
└─$


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
root@kali: /
File Actions Edit View Help
┌──(root㉿kali)-[/]
└─# cat << EOF > target_hashes.txt
heredoc>
heredoc> dc647eb65e6711e155375218212b3964
heredoc> eb61eead90e3b899c6bcbe27ac581660
heredoc> 958152288f2d2303ae045cffc43a02cd
heredoc> 2c9341ca4cf3d87b9e4eb905d6a3ec45
heredoc> 75b71aa6842e450f12aca00fdf54c51d
heredoc> 031cbcccd3ba6bd4d1556330995b8d08
heredoc> b5af0b804ff7238bce48adef1e0c213f
heredoc>
heredoc> EOF
┌──(root㉿kali)-[/]
└─# cat target hashes.txt
cat: target: No such file or directory
cat: hashes.txt: No such file or directory
┌──(root㉿kali)-[/]
└─# cat target_hashes.txt
dc647eb65e6711e155375218212b3964
eb61eead90e3b899c6bcbe27ac581660
958152288f2d2303ae045cffc43a02cd
2c9341ca4cf3d87b9e4eb905d6a3ec45
75b71aa6842e450f12aca00fdf54c51d
031cbcccd3ba6bd4d1556330995b8d08
b5af0b804ff7238bce48adef1e0c213f
┌──(root㉿kali)-[/]
└─#


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
root@kali: /
File Actions Edit View Help
┌──(root㉿kali)-[/]
└─# cat target_hashes.txt
dc647eb65e6711e155375218212b3964
eb61eead90e3b899c6bcbe27ac581660
958152288f2d2303ae045cffc43a02cd
2c9341ca4cf3d87b9e4eb905d6a3ec45
75b71aa6842e450f12aca00fdf54c51d
031cbcccd3ba6bd4d1556330995b8d08
b5af0b804ff7238bce48adef1e0c213f
┌──(root㉿kali)-[/]
└─# cat rockyou.txt
cat: rockyou.txt: No such file or directory
┌──(root㉿kali)-[/]
└─# ls
bin boot dev etc home initrd.img initrd.img.old lib lib32 lib64 lost+found media mnt opt proc root run sbin srv swapfile sys target_hashes.txt usr var vmlinuz vmlinuz.old
┌──(root㉿kali)-[/]
└─# hashcat -m 0 -a 0 -o cracked.txt target_hashes.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz, 1435/2934 MB (512 MB allocatable), 4MCU
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Hashes: 7 digests; 7 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash
ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.
Watchdog: Temperature abort trigger set to 90c
Initializing backend runtime for device #1. Please be patient ...


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player
1 2 3 4
6:58
root@kali: /
File Actions Edit View Help
Host memory required for this attack: 0 MB
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 9 secs
Cracking performance lower than expected?
* Append -O to the commandline.
This lowers the maximum supported password/salt length (usually down to 32).
* Append -w 3 to the commandline.
This can cause your screen to lag.
* Append -S to the commandline.
This has a drastic speed impact but can be better for specific attacks.
Typical scenarios are a small wordlist but a large ruleset.
* Update your backend API runtime / driver the right way:
https://hashcat.net/faq/wrongdriver
* Create more work items to make use of your parallelization power:
https://hashcat.net/faq/morework
Approaching final keyspace - workload adjusted.
Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 0 (MD5)
Hash.Target......: target_hashes.txt
Time.Started.....: Wed Jan 22 06:57:29 2025 (16 secs)
Time.Estimated...: Wed Jan 22 06:57:45 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 510.6 kH/s (0.15ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 5/7 (71.43%) Digests (total), 5/7 (71.43%) Digests (new)
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: $HEX[206b72697374656e616e6e65] → $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#1..: Util: 34%
Started: Wed Jan 22 06:55:38 2025
Stopped: Wed Jan 22 06:57:46 2025
(root㉿kali)-[/]
#

Here we would be using HASHCAT to crack some hashed password.
A hash is a one-way function that takes a word or string of words and turns them into a fixed length of random characters. This is a much more secure method of storing passwords rather than storing them in plain text. It is not reversible.Hashcat attempts to crack these passwords by guessing a password, hashing it, and then comparing the resulting hash to the one it's trying to crack.

Create the Hashes

To generate a set of hashes and save them to a text file, run the following command:

```
cat << EOF > target_hashes.txt

Dc647eb65e6711e155375218212b3964
Eb61eead90e3b899c6bcbe27ac581660
958152288f2d2303ae045cffc43a02cd
2c9341ca4cf3d87b9e4eb905d6a3ec45
75b71aa6842e450f12aca00fdf54c51d
031cbcccd3ba6bd4d1556330995b8d08
b5af0b804ff7238bce48adef1e0c213f
EOF
```

This creates a `target_hashes.txt` file containing the hashes to crack.

2. View Hashcat Help

To explore Hashcat's options, view the help screen:

```
hashcat -h | more
```

Press the **Space** key to scroll through pages and **Ctrl + C** to exit.

3. Choose Hash Type and Attack Mode

The two key options are:

- **Hash Type (`-m`)**: Identifies the hash format (e.g., MD5).
- **Attack Mode (`-a`)**: Specifies the method for attacking the hash (e.g., dictionary attack).

For this case, we are using:

- **MD5** (Hash Type `-m 0`)
- **Dictionary Attack** (Attack Mode `-a 0`)

4. Locate and Unzip Wordlist

To use the `rockyou.txt` wordlist, first locate it:

```
locate rockyou.txt
```

If the file is compressed (`rockyou.txt.gz`), unzip it with:

```
gunzip /path/to/rockyou.txt.gz
```

Navigate back to your home directory if necessary:

```
cd
```

5. Crack the Password Hashes

Now you're ready to start cracking the hashes. Run the following command:

```
hashcat -m 0 -a 0 -o cracked.txt target_hashes.txt
/usr/share/wordlists/rockyou.txt
```

Here’s what each option does:

- **`-m 0`**: Specifies MD5 hash type.
- **`-a 0`**: Uses a dictionary attack.
- **`-o cracked.txt`**: Outputs the cracked passwords to `cracked.txt`.
- **`target_hashes.txt`**: The file containing the hashes to crack.
- **`/usr/share/wordlists/rockyou.txt`**: The wordlist used for the attack.

6. Review Cracked Passwords

After the attack completes, check the cracked passwords in the `cracked.txt` file:

```
cat cracked.txt
```