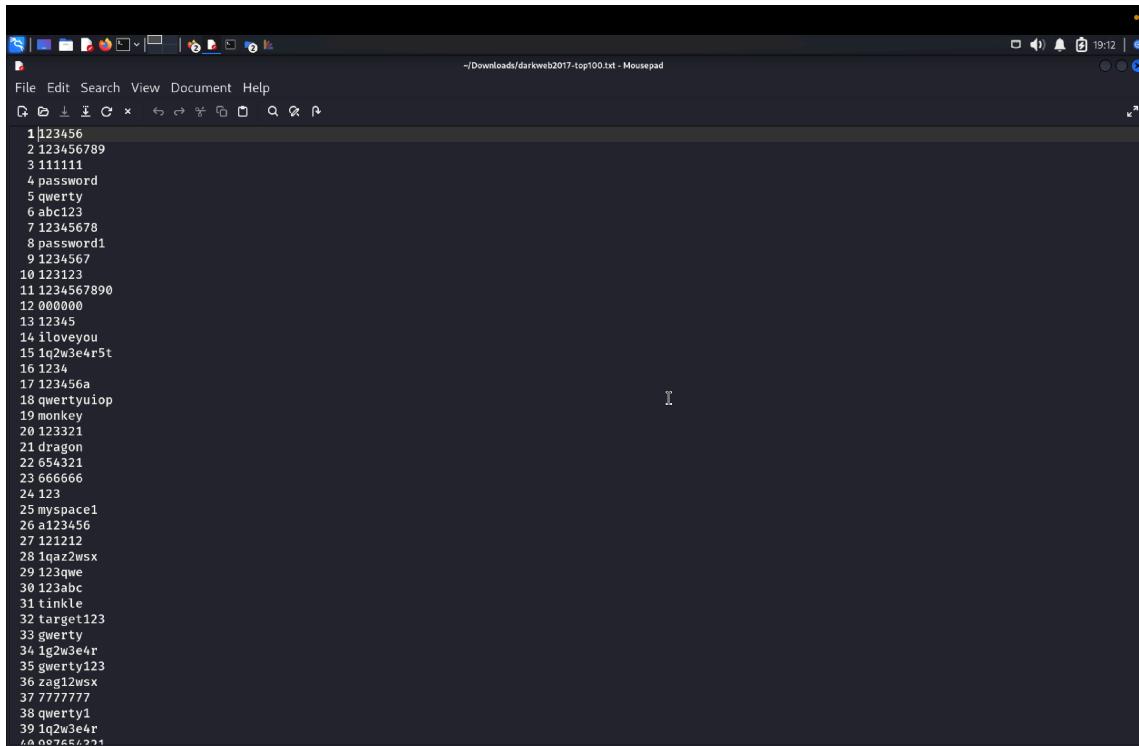
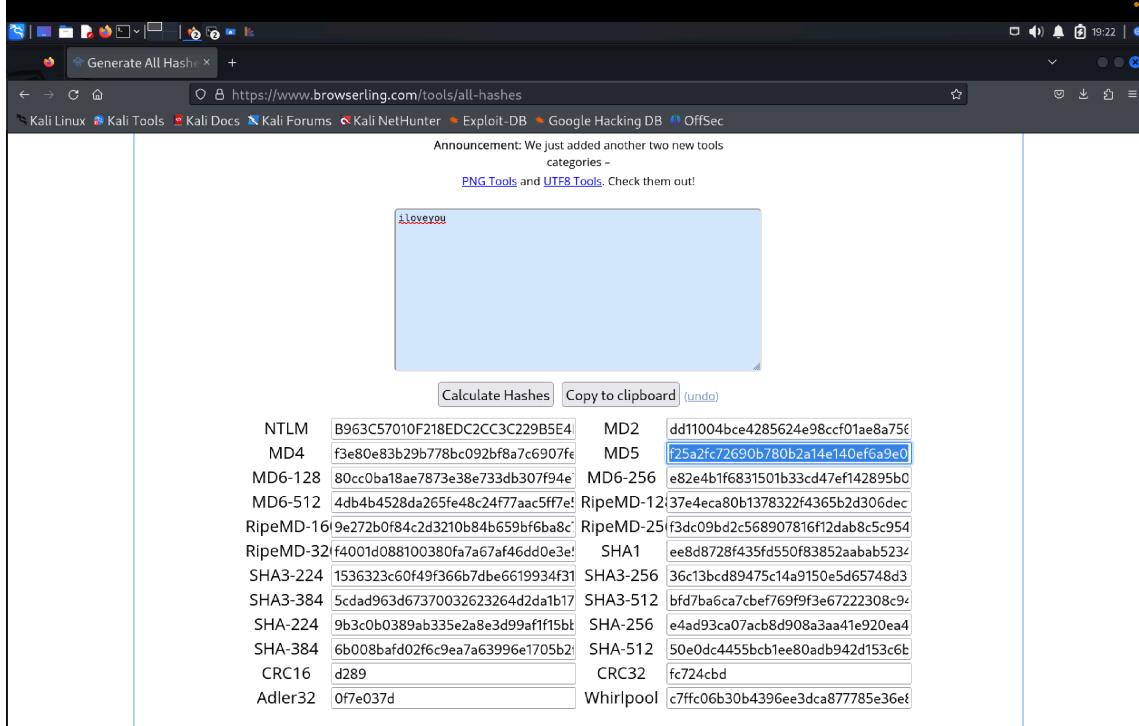


USE HASHCAT TO CRACK A PASSWORD HASH



```
1 123456
2 1234567890
3 111111
4 password
5 qwerty
6 abc123
7 12345678
8 password1
9 1234567
10 123123
11 1234567890
12 000000
13 12345
14 iloveyou
15 1q2w3e4r5t
16 1234
17 123456a
18 qwertyuiop
19 monkey
20 123321
21 dragon
22 654321
23 666666
24 123
25 myspace1
26 a123456
27 121212
28 1qaz2wsx
29 123qwe
30 123abc
31 tinkle
32 target123
33 qwerty
34 1g2w3e4r
35 qwerty123
36 zag12wsx
37 777777
38 qwerty1
39 1q2w3e4r
40 0987654321
```

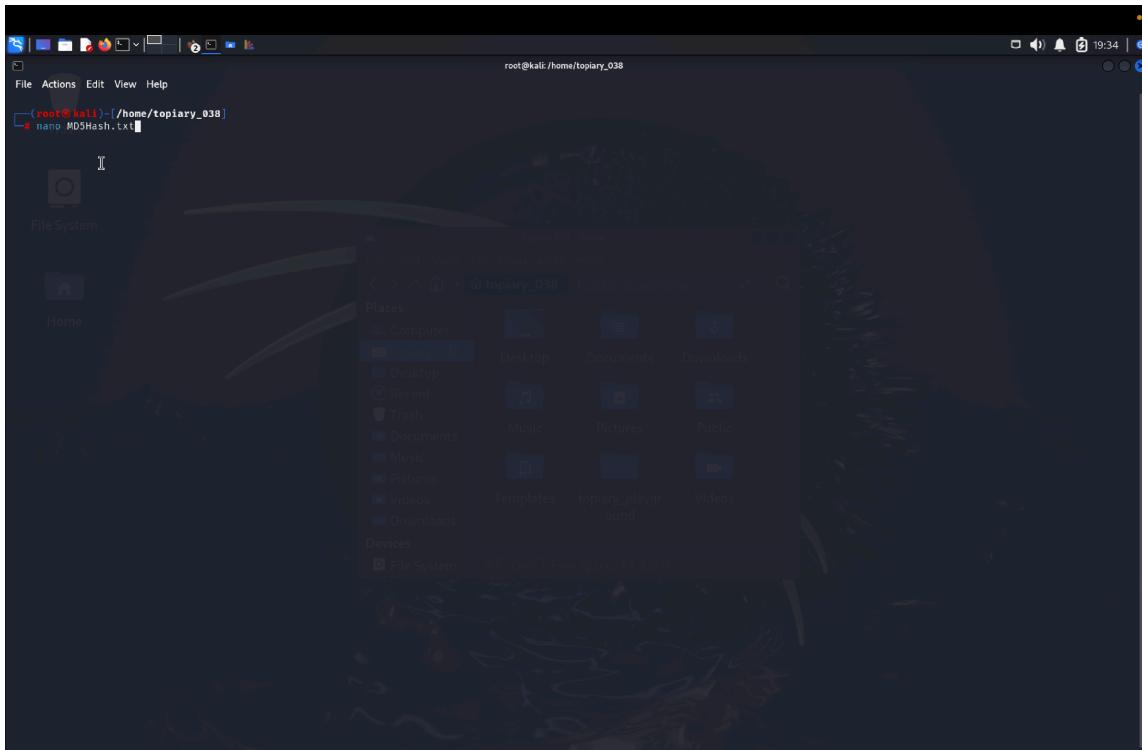
1. First we create a word list/password list or download it for generating the password . Here the wordlist file name is darkweb2017-top100.txt . Now we are going to select a random word/password from this wordlist/password list to generate the hash



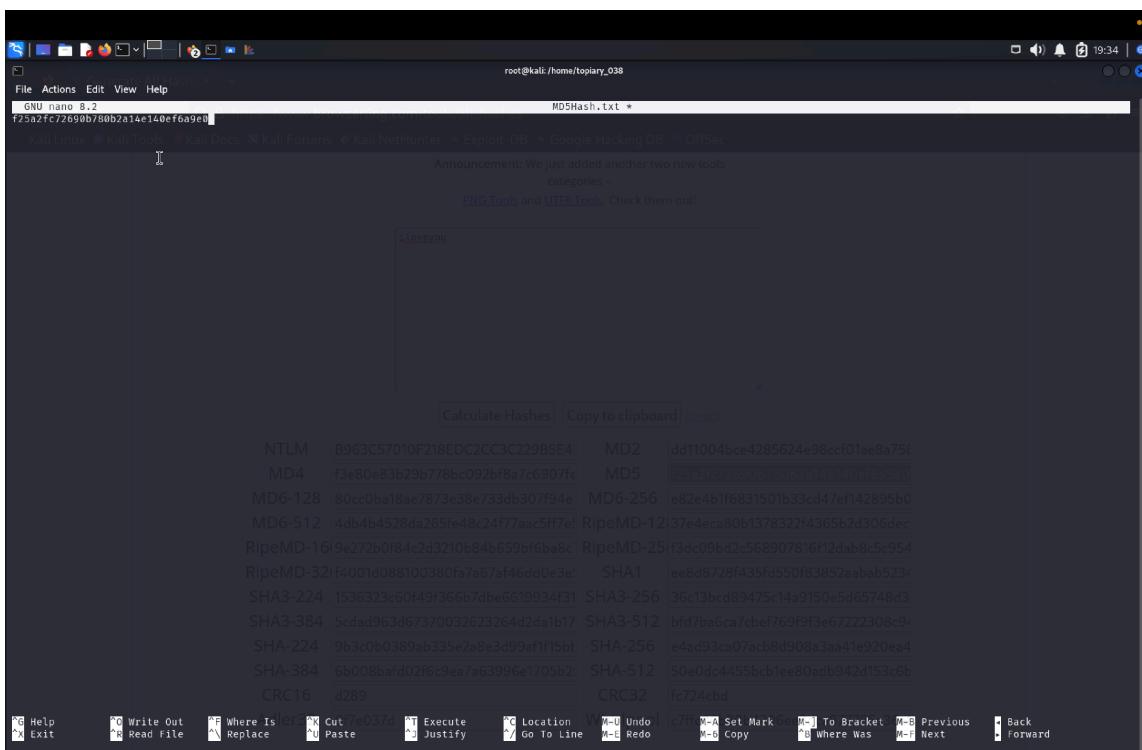
NTLM	B963C57010F218EDC2CC3C229B5E4	MD2	dd11004bce4285624e98ccf01ae8a75e
MD4	f3e80e83b29b778bc092bf8a7c6907fe	MD5	f25a2fc72690b780b2a14e140ef6a9e0
MD6-128	80cc0ba18ae7873e38e733db307f94e	MD6-256	e82e4b1f6831501b33cd47ef142895b0
MD6-512	4db4b4528da265fe48c24f7aac5ff7e	RipeMD-128	37e4eca80b1378322f4365b2d306dec
RipeMD-160	9e272b0f84c2d3210b84b659bf6ba8c	RipeMD-256	f3dc09bd2c568907816f12dab8c5c954
RipeMD-32	f4001d088100380f67a67af46dd0e3e	SHA1	ee8d8728f435fd550fb3852aabab5234
SHA3-224	1536323c60f49f366b7dbe6619934f31	SHA3-256	36c13bcd89475c14a9150e5d65748d3
SHA3-384	5cdad963d67370032623264d2da1b17	SHA3-512	bfd7ba6ca7cbe769f9f3e67222308c94
SHA-224	9b3c0b0389ab335e2a8e3d99af1f15bt	SHA-256	e4ad93ca07acb8d908a3aa41e920ea4
SHA-384	6b008baf02f6c9ea7a63996e1705b2	SHA-512	50e0dc4455bcb1ee80adb942d153c6b
CRC16	d289	CRC32	fc724cbd
Adler32	0f7e037d	Whirlpool	c7ffc06b30b4396ee3dca877785e36e8

2. To generate the hash from that particular word/password we are going to use any third party website here it is www.browserling.com it converts the word to all kinds of hash

But here we are going to work with MD5 hash



3. here we are going to create a hash file named MD5Hash.txt



4. In order to create the hash file, inside the hash file we are just going to write the MD5 hash of the selected word/password from the wordlist/password list

```
[+] Possible Hashes:
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
Least Possible Hashes:
[+] RAdmin v2.x
[+] NTLM
[+] MySQLsystem
[+] MD5
[+] MD5(HMAC)
[+] MD4(HMAC)
[+] MD2(HMAC)
[+] MD5(HMAC(wordpress))
[+] Haval-128
[+] Haval-128(HMAC)
[+] RipeMD-128
[+] RipeMD-128(HMAC)
[+] SHA1
[+] SHA1MD5
[+] SNEFRU-128(HMAC)
[+] Tiger-128
[+] Tiger-128(HMAC)
[+] md5($pass.$salt)
[+] md5($salt.$pass)
[+] md5($salt.$pass.$salt)
[+] md5($salt.$pass.$username)
[+] md5($salt.md5($pass))
[+] md5(md5($salt.$pass))
[+] md5($salt.md5($salt.$salt))
[+] md5($salt.md5($pass.$salt))
[+] md5($salt.md5($salt.$pass))
[+] md5($salt.md5(md5($salt.$salt)))
[+] md5($username.O.$pass)
[+] md5($username.LF.$pass)
[+] md5($username.md5($pass))
[+] md5(md5($username.$pass))
[+] md5(md5(md5($pass.$salt)))
[+] md5(md5($pass.$salt))
[+] md5(md5($salt).$pass)
[+] md5(md5($salt).md5($pass))
[+] md5(md5($username.$pass).$salt)
[+] md5(md5(md5($pass)))
[+] md5(md5(md5(md5($pass))))
[+] md5(md5(md5(md5(md5($pass)))))

[+] md5($hai($pass))
[+] md5($hai(md5($pass)))
[+] md5($hai(md5($hai($pass))))
[+] md5($stroupper(md5($pass)))

HASH: f25a7fc72690b780b2a14e140efca9e0
```

5. Now we are going to use hash-identifier to identify the hash present in the hash file MD5Hash.txt , here it says that the possible hash is MD5 , so we now know the type of hash , we will further move on to decode the hash to get the original word/password.

```
(root㉿kali)-[~/home/topiary_038]
# hashcat --help | grep MD5
      0 | Raw Hash
  51000 | Raw Hash
  50 | HMAC-MD5 (key = $pass)
  60 | HMAC-MD5 (key = $salt)
119000 | PBKDF2-HMAC-MD5
114000 | SIP digest authentication (MD5)
  5300 | IKE-PSK MD5
251000 | SNMPv3 HMAC-MD5-96
250000 | SNMPv3 HMAC-MD5-96/HMAC-SHA1-96
102000 | CRAM-MD5
48000 | LDAPv3 GSSAPI authentication, MD5 (CHAP)
300000 | QNX /etc/shadow (MD5)
  2410 | Cisco-ASA MD5
  2400 | Cisco-PIX MD5
  500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)
111000 | PostgreSQL CRAM (MD5)
  164000 | CRAM-MD5 Dovecot
  249000 | Dahua Authentication MD5
  16000 | Apache $apr1$ MD5, md5Apr1, MD5 (APR)
  97000 | MS Office < 2003 $0/$1, MD5 + RC4
  9710 | MS Office < 2003 $0/$1, MD5 + RC4, collider #1
  9720 | MS Office < 2003 $0/$1, MD5 + RC4, collider #2
  380000 | Python Werkzeug MD5 (HMAC-MD5 (key = $salt))
  225000 | Multibit Classic .key (MD5)
Wordlist + Rules | MD5 | hashcat -a 0 -m 0 example0.hash example.dic -r rules/best6.rule
Brute-Force | MD5 | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?
Combinator | MD5 | hashcat -a 1 -m 0 example0.hash example.dic example.dic

(root㉿kali)-[~/home/topiary_038]
```

6. Now we use the grep command to get the attack mode for the MD5 hash from – help documentation which will be further useful to write the syntax.

In this case the attack mode for MD5 is 0

```

root@kali:~/home/topiary_038/topiary_playground
# hashcat -m 0 MD5Hash.txt darkweb2017-top100.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEP, POCL_DEBUG) - Platform #1 [The pool project]
* Device #1: cpu—0x000, 1437/2939 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 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-Hash
* 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: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename.: darkweb2017-top100.txt
* Passwords.: 99
* Bytes....: 802
* Keystpace.: 99
* Runtime ...: 0 secs

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keystpace - workload adjusted.

f25a2fc72690b780b2a14e140ef6a9e0:iloveyou

Session.....: hashcat

```

7.now we are going to write the syntax to perform the Hashcat operation to get the final word/password from the hash , the syntax is -

`hashcat -m [hash_type] -a [attack_mode] [hash_file] [wordlist]`

In this case it is- `hashcat -m 0 MD5Hash.txt darkweb2017-top100.txt`

```

root@kali:~/home/topiary_038/topiary_playground
# hashcat -m 0 MD5Hash.txt darkweb2017-top100.txt --show
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEP, POCL_DEBUG) - Platform #1 [The pool project]
* Device #1: cpu—0x000, 1437/2939 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 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-Hash
* 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: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename.: darkweb2017-top100.txt
* Passwords.: 99
* Bytes....: 802
* Keystpace.: 99
* Runtime ...: 0 secs

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keystpace - workload adjusted.

f25a2fc72690b780b2a14e140ef6a9e0:iloveyou

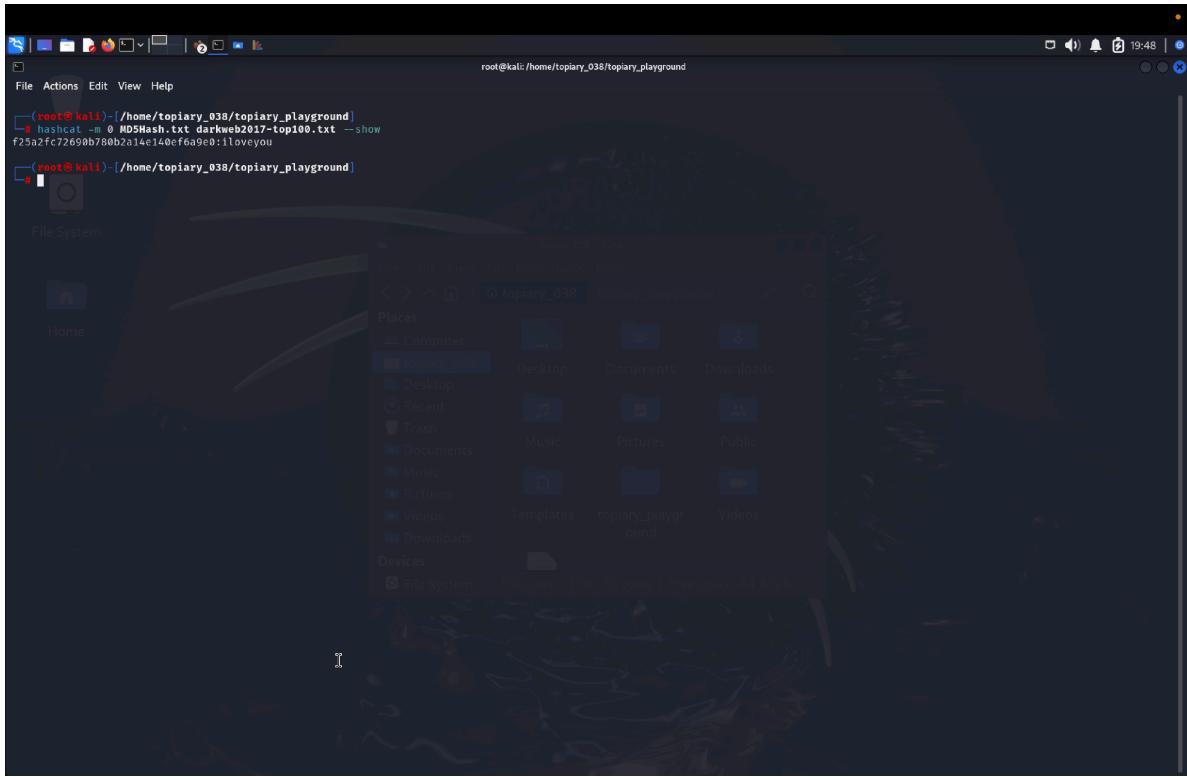
Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 0 (MD5)
Hash.Target....: f25a2fc72690b780b2a14e140ef6a9e0
Time.Started....: Sun Oct 13 19:47:30 2024 (0 secs)
Time.Estimated...: Sun Oct 13 19:47:30 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (darkweb2017-top100.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.MF....: 4219 M/s (0.01ms) @ Accel:256 Loops:1 Thr:1 Vec:4
Recovered....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 99/99 (100.00%)
Rejected.....: 0/99 (0.00%)
Restore.Point...: 0/99 (0.00%)
Restore.Sub.#...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#...: 123450 → hābygirl1

Started: Sun Oct 13 19:47:17 2024
Stopped: Sun Oct 13 19:47:31 2024
[...]

```

8. It shows status cracked which means the MD5 hash has been cracked
We can get the specific word/password from the hash by the syntax-

`hashcat -m 0 MD5Hash.txt darkweb2017-top100.txt —show`



9. Finally we get the word from the MD5 hash , the word is iloveyou

Thus we have successfully used hashcat to crack a MD5 hash to get the desired word/password