


# Hashing - Crypto 101

example\_hashes [hashcat wiki]

If you get a "line length exception" error in hashcat, it is often because the hash mode that you have requested does not match the hash. To verify, you can test your commands against example hashes.

 [https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

## Task - 1:

### key Terms

**Plaintext** - data before encryption or hashing, often text but not always as it could be a photograph or other file insted.

**Encoding** - this is not a form of encryption, just a form of data representation like base64 or hexadecimal. immediately reversible.

**Hash:** A hash is the output of a hash function, hashing can also be used as a verb, "to hash", meaning to produce the hash value of some data.

**Brute force:** Attacking cryptography by trying every different password or every different key

**Cryptanalysis:** Attacking cryptography by finding a weakness in the underlying maths.

---

## Task - 2

### what is a hash function

hash functions are quite different from encryption, there is no key, and its meant to be impossible to go from the output back to the input.

a hash function takes some input data of any size, and creates a summary or digest of that data. the output is a fixed size, its hard to predict what the output will be from any input and vice versa. good hahing algorithms will be fast to compute, and slow to reverse, any small change in the input data should cause a large change in the output.

hashing is used very often in cyber security, when you logged into tryhackme, that used hashing to verify your password.

when you logged into your computer, that also used hashing verify your password.

### hash collision

a hash collision is when 2 different inputs give the same output. hash function are designed to avoid this as best as they can especially being able to engineer a collision.

Due to the pigeonhole effect, collisions are not avoidable. the pigeonhole effect is basically there are a set number of different output values for the hash function, but you can give it any size input, as there are more inputs than outputs, some of the inputs must give the same output, if you have 128 pigeons and 96 pigeonholes, some of the pigeons are going to have to share.

MD5 and SHA1 have been attacked and made technically insecure due to engineering hash collisions. however, no attack has yet given a collision in both algorithms at the same time so if you use the MD5 hash and the SHA1 hash to compare, you will see they're different,

**1. What is the output size in bytes of the MD5 hash function?**

ans: **16 (bytes)**

**2. Can you avoid hash collisions?**

ans: **no**

**3. If you have an 8 bit hash output, how many possible hashes are there?**

ans: **256**

---

**Task - 3**

## uses for hashing

CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc.

Crackstation is the most effective hash cracking service. We crack: MD5, SHA1, SHA2, WPA, and much more...

 <https://crackstation.net/>

hashing is used for 2 main purposes in cyber security, to verify integrity of data, or for verifying passwords.

Adobe had a notable data breach

LinkedIn also had a data breach. LinkedIn used SHA1 for password verification

**1. Crack the hash "d0199f51d2728db6011945145a1b607a" using the rainbow table manually.**

ans: **basketball**

Enter up to 20 non-salted hashes, one per line:

d0199f51d2728db6011945145a1b607a



I'm not a robot



Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
d0199f51d2728db6011945145a1b607a	md5	basketball

## 2. Crack the hash "5b31f93c09ad1d065c0491b764d04933" using online tools

ans: [tryhackme](#)

## 3. Should you encrypt passwords?

ans: [nay](#)

## Task - 4

## Recognising password hashes

Automated hash recognition tools such as <https://pypi.org/project/hashID/> exist, but they are unreliable for many formats

**If you found the hash in a web application database, it's more likely to be md5 than NTLM.**

Unix style password hashes are very easy to recognise, as they have a prefix. The prefix tells you the hashing algorithm used to generate the hash. The standard format is `$format$rounds$salt$hash`.

**Windows passwords are hashed using NTLM**, which is a variant of md4. They're visually identical to md4 and md5 hashes, so it's very important to use context to work out the hash type

**On Linux, password hashes are stored in /etc/shadow.**

**On Windows, password hashes are stored in the SAM.** Windows tries to prevent normal users from dumping them, but tools like

## mimikatz

exist for this.

Here's a quick table of the most Unix style password prefixes that you'll see.

Prefix	Algorithm
\$1\$	md5crypt, used in Cisco stuff and older Linux/Unix systems
\$2\$, \$2a\$, \$2b\$, \$2x\$, \$2y\$	Bcrypt (Popular for web applications)
\$6\$	sha512crypt (Default for most Linux/Unix systems)

A great place to find more hash formats and password prefixes is the hashcat example page, available here: [https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes).

1. **How many rounds does sha512crypt (\$6\$) use by default?**

ans: 5000

2. **What's the hashcat example hash (from the website) for Citrix Netscaler hashes?**

ans: 1765058016a22f1b4e076dccd1c3df4e8e5c0839ccded98ea

3. **How long is a Windows NTLM hash, in characters?**

ans: 32

---

### Task - 5

## Password Cracking

We've already mentioned rainbow tables as a method to crack hashes that don't have a salt, but what if there's a salt involved?

You have to crack the hashes by hashing a large number of different inputs (often rockyou, these are the possible passwords), potentially adding the salt if there is one and comparing it to the target hash. Once it matches, you know what the password was. Tools like Hashcat and John the Ripper are normally used for this.

### Why crack on GPUs?

Graphics cards have thousands of cores. Although they can't do the same sort of work that a CPU can, they are very good at some of the maths involved in hash functions. This means you can use a graphics card to crack most hash types much more quickly. Some hashing algorithms, notably bcrypt, are designed so that hashing on a GPU is about the same speed as hashing on a CPU which helps them resist cracking.

**NEVER (I repeat, NEVER!) use --force for hashcat.** It can lead to false positives (wrong passwords being given to you) and false negatives (skips over the correct hash).

1. **Crack this hash: \$2a\$06\$7yoU3Ng8dHTXphAg913cyO6Bjs3K5lBnwq5FJyA6d01pMSrddr1ZG**

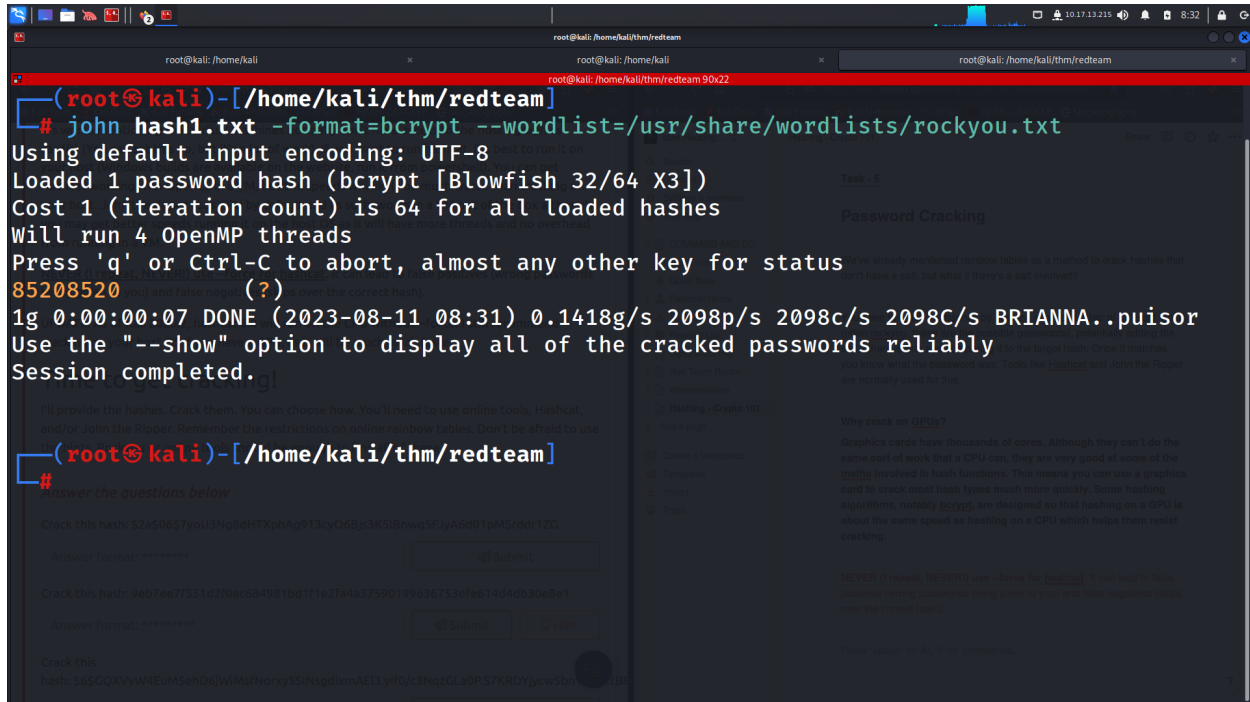
**process:**

add this hash to the text file in my case it was hash1.txt

then after using of john the ripper will help us to crack the password

command:

```
$ john hash1.txt -format=bcrypt --wordlist=/usr/share/wordlist/rockyou.txt
```



```
(root@kali)-[/home/kali/thm/redteam]
# john hash1.txt -format=bcrypt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 64 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
85208520 (?)
1g 0:00:00:07 DONE (2023-08-11 08:31) 0.1418g/s 2098p/s 2098c/s 2098C/s BRIANNA..puisor
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

I'll provide the hashes. Crack them. You can choose how. You'll need to use online tools. Hashcat,
and/or, into the browser. Remember the restrictions on online rainbow tables. Don't be afraid to use

Answer the questions below

Crack this hash: 9eb7ee7f551d2f0ac684981bd1f1e2fa4a37590199636753efe614d4db30e8e1
Answer format: plaintext
Submit

Crack this hash: 9eb7ee7f551d2f0ac684981bd1f1e2fa4a37590199636753efe614d4db30e8e1
Answer format: plaintext
Submit

Crack this
hash: SASGQXVwW4buh5hD10WMeVnreyCSHupl0mAE13yF0/C3HupCLa0R5TKR0Vpce50K...
```

ans: 85208520

2. Crack this hash: 9eb7ee7f551d2f0ac684981bd1f1e2fa4a37590199636753efe614d4db30e8e1

copying the hash in the text file, in mycase it was hash2.txt

command:

```
$ john -format=raw-sha256 --wordlist=/usr/share/wordlist/rockyou.txt
```

```
(root@kali)~/home/kali/thm/redteam
# cat hash2.txt
9eb7ee7f551d2f0ac684981bd1f1e2fa4a37590199636753efe614d4db30e8e1

(root@kali)~/home/kali/thm/redteam
# john hash2.txt -format=raw-sha256 --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
halloween format=raw (?) --wordlist=/usr/share/wordlists/rockyou.txt
1g 0:00:00:00 DONE (2023-08-11 08:41) 100.0g/s 6553Kp/s 6553Kc/s 6553KC/s 123456..sabrina7
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.

Crack this hash:
$ cat hash3.txt
$ john hash3.txt --format=sha512crypt --wordlist=/usr/share/wordlists/rockyou.txt

This is of the hash format sha512crypt (see previous task). The command is
therefore:

john hash3.txt --format=sha512crypt --wordlist=/usr/share/wordlists/rockyou.txt
```

ans: **halloween**

### 3. Crack this

hash: \$6\$GQXVvW4EuM\$ehD6jWiMsfNorxy5SINsgdLxmAEI3.yif0/c3NqzGLa0P.S7KRDYjycw5bnYkF5ZtB8wQy8KnskuWV

copying the password and paste it a file, in my case the file was hash3.txt

then using of john

command:

```
$ john hash3.txt --wordlist=/usr/share/wordlist/rockyou.txt
```

```
(root@kali)-[/home/kali/thm/redteam]
# vim hash3.txt

(root@kali)-[/home/kali/thm/redteam]
# john hash3.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spaceman (?)
1g 0:00:00:03 DONE (2023-08-11 08:47) 0.2832g/s 5366p/s 5366c/s 5366C/s sweetgurl..playas
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(root@kali)-[/home/kali/thm/redteam]
# [ ]
```

ans: spaceman

4. Bored of this yet? Crack this hash: b6b0d451bbf6fed658659a9e7e5598fe

we gonna use online tool for this that is

CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc.  
Crackstation is the most effective hash cracking service. We crack: MD5, SHA1, SHA2, WPA, and much more...  
<https://crackstation.net/>

Enter up to 20 non-salted hashes, one per line:

b6b0d451bbf6fed658659a9e7e5598fe

I'm not a robot

reCAPTCHA

Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
b6b0d451bbf6fed658659a9e7e5598fe	md5	funforyou

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

ans: funforyou

## Task - 6

### Hashing for integrity checking

#### integrity checking

hashing can be used to check that files havent been changed, if you put the same data in, you always get the sam data out. if even a single bit changes, the hash will change a log, this means you can use it to check that files havent been modified or to make sure that they have downloaded correctly , you can also use hashing to find duplicate files, if two pictures have the same hash then they are the same picture.

#### HMACs

HMAC is a method of using a cryptographic hashing function to verify the authenticity and integrity of data. the tryhackme VPN uses HMAC-SHA512 for message authentication, which you can see in the terminal output. A HMAC can be used to ensure that the person who created the HMAC is who they say they are, and that the message hasnt been modified or corrupted, they use a secret key, and a hashing algorithm in order to produce a hash


1. What's the SHA1 sum for the amd64 Kali 2019.4 ISO? <http://old.kali.org/kali-images/kali-2019.4/>

ans: **186c5227e24ceb60deb711f1bdc34ad9f4718ff9**

2. What's the hashcat mode number for HMAC-SHA512 (key = \$pass)?

example\_hashes [hashcat wiki]

If you get a "line length exception" error in hashcat, it is often because the hash mode that you have requested does not match the hash. To verify, you can test your commands against example hashes.

 [https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

1720	sha512(\$salt.\$pass)	976b451818634a1e2acba682da3fd6ef
1730	sha512(utf16le(\$pass).\$salt)	13070359002b6fbb3d28e50fba55efcf3
1740	sha512(\$salt.utf16le(\$pass))	bae3a3358b3459c761a3ed40d34022f0
1750	HMAC-SHA512 (key = \$pass)	94cb9e31137913665dbea7b058e10be
1760	HMAC-SHA512 (key = \$salt)	7cce966f5503e292a51381f238d07197
1770	sha512(utf16le(\$pass))	79bba09eb9354412d0f2c037c22a777b
1800	sha512crypt \$6\$, SHA512 (Unix) <sup>2</sup>	\$6\$52450745\$k5ka2p8bFuSmoVT1tzC
2000	STDOUT	n/a
2100	Domain Cached Credentials 2 (DCC2), MS Cache 2	\$DCC2\$10240#tom#e4e938d12fe5974

ans: **1750**