

How To Use Mask Attack With Hashcat – A complete guide

 infosecscout.com/use-mask-attack-with-hashcat

Patrick Fromaget



Hashcat is a free and fast password cracker available on any platform (Linux, Windows, macOS). I talk a lot about this tool on this website, and today we'll focus on one of the most popular feature you can use with Hashcat: the mask attack.

Hashcat can use several attacks. Mask attacks is the number 3 (-a 3) and should be followed with the key space of the password you are looking to crack (ex: ?l?l?l for a three characters password in lowercase). The full command looks like: *hashcat <hashes> -m 0 -a 3 <key space>*.

Don't worry if you are new to this vocabulary, I'll explain everything in this article. Just [make sure you have Hashcat installed on your computer](#), and read the following to know everything about mask attacks with Hashcat.

Master Linux Commands

Your essential Linux handbook

Want to level up your Linux skills? Here is the perfect solution to become efficient on Linux. **20% off today!**

[Download now](#)

What is a mask attack in Hashcat

Mask attack is a built-in feature in Hashcat, allowing to try all combinations in a specific key space (a set of possible passwords formats). Most people are using similar passwords formats, making it a faster solution than brute force on completely random characters.

For example, if you know that most users have an 8 characters password (because of a minimum number of character set in the system), and most of them will end their password with their birth year, you can guess that many passwords might look like "mike1990" or "jordan85".

Hide your IP address and location with a free VPN:

[Try it for free now](#), with advanced security features.

2900+ servers in 65 countries. It's free. Forever.

In this case, we can try these two formats (4 letters + 4 numbers, 6 letters + 2 numbers) before using brute force on random formats. Hashcat can do this for you. You can even specify if you are testing all letters or just the lowercase ones.

The less possibility you use in your mask, the faster it will be to find the corresponding matches.

But instead of being too long on the theory, let's start directly the practice. I'll explain how this strategy works with Hashcat and give you a few examples.

How to use mask attack with Hashcat

Theory

Before using mask attack with Hashcat, there are a few concepts you need to understand.

First, the hashcat command syntax looks like this:

```
hashcat <options> <hashes> <mask>
```

```
C:\hashcat\hashcat-6.2.5>hashcat
Usage: hashcat [options]... hash[hashfile|hccapxfile] [dictionary|mask|directory]...
```

The main options include the algorithm you are testing (0 is MD5 for example), and the attack you want to try (mask attack is 3). So, your command will start with something like:

```
hashcat -m 0 -a 3 <hashes> <mask>
```

The <hashes> parameter can either be one specific MD5 hash (5f4dcc3b5aa765d61d8327deb882cf99 is "password" for example) or a file containing different hashes.

The<mask> parameter is the format you want to try (the key space), or a file containing a list of key spaces (I will talk about this at the end). The key space tells the length of the password you are trying to crack, and the charsets used in the plain text password (lowercase, uppercase, numbers, special characters, etc.).

Charsets

Hashcat include a number of built-in charsets that you can use directly in the mask attack command:

- ?l = abcdefghijklmnopqrstuvwxyz
- ?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
- ?d = 0123456789
- ?h = 0123456789abcdef
- ?H = 0123456789ABCDEF
- ?s = «space»!''#\$%&'()*+,-./:;<=>?@[]^_`{|}~
- ?a = ?l?u?d?s
- ?b = 0x00 – 0xff

Knowing this, a first basic example for mask attack in Hashcat can look like this:

```
hashcat -m 0 -a 3 5f4dcc3b5aa765d61d8327deb882cf99 ?l?l?l?l?d?d?d?d
```

In this case, I'm testing all the possibilities for a password starting with 4 letters in lowercase, followed by 4 digits. This won't find a match, as my MD5 hash is the encrypted version of the word "password". But you get the idea.

Custom charsets

It is possible to create custom charsets with Hashcat, in case the built-in charsets don't fit your needs. You can use 4 custom charsets by using 4 shortcuts (-1, -2, -3 and -4).

Here is an example:

```
hashcat -m 0 -a 3 -1 abcdefghijklmnopqrstuvwxyz0123456789 <hash> ?l?l?l?l
```

In this example, I create a new charset including lowercase letters and digits, and try to crack the password by testing 4 characters words using this charset.

Examples

You can then combine these different charsets to do exactly what you want, here are a few examples:

- **Passwords containing 8 digits:**

```
hashcat -m 0 -a 3 <hash> ?d?d?d?d?d?d?d?d
```
- **Passwords in lowercase letters, ending with two digits:**

```
hashcat -m 0 -a 3 <hash> ?l?l?l?l?l?l?l?d?d
```

- **Passwords composed of 5 letters (lowercase and uppercase):**

```
hashcat -m 0 -a 3 -1 ?l?u <hash> ?1?1?1?1?1
```

They are random examples, but I hope you get the idea. You can always refer to the official documentation now that you know the basics of mask attack with Hashcat.

And if you want to try to crack the word “password” I give you letter, you’ll need to try an 8 characters with lowercase only, so something like:

```
hashcat -m 0 -a 3 5f4dcc3b5aa765d61d8327deb882cf99 ?1?1?1?1?1?1?1?1
```

Giving you this result after a few seconds:

```
Host memory required for this attack: 1457 MB

5f4dcc3b5aa765d61d8327deb882cf99:password

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 0 (MD5)
Hash.Target.....: 5f4dcc3b5aa765d61d8327deb882cf99
Time.Started.....: Tue Jul 26 06:03:51 2022, (0 secs)
Time.Estimated...: Tue Jul 26 06:03:51 2022, (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Mask.....: ?1?1?1?1?1?1?1?1 [8]
```

Each matching password cracked will be shown in the command output. The process will stop once all hashes have been cracked (you can replace the hash in the command line by the name of a file containing all hashes, one per line):

```
hashcat -m 0 -a 3 hashes.txt ?1?1?1?1?1?1?1?1
```

Using mask files with Hashcat

In general, you won’t know the exact mask for a set of hashes you want to crack. But you can guess what the best chances are. Most people are using the same kind of passwords formats.

As I explained previously, you have more chances to find short passwords starting in lowercase and ending with a few digits, maybe one special character, than 20 characters long passwords with 10 special characters in them.

Hide your IP address and location with a free VPN:

[Try it for free now](#), with advanced security features.

2900+ servers in 65 countries. It's free. Forever.

Use built-in mask files

This PC > OS (C:) > hashcat > hashcat-6.2.5

Name	Date modified	Type	Size
charsets	11/21/2021 4:43 PM	File folder	
docs	11/21/2021 4:43 PM	File folder	
extra	11/21/2021 4:43 PM	File folder	
kernels	7/26/2022 6:03 AM	File folder	
layouts	11/21/2021 4:43 PM	File folder	
masks	11/21/2021 4:43 PM	File folder	
modules	11/21/2021 4:43 PM	File folder	
OpenCL	11/21/2021 4:43 PM	File folder	
rules	11/21/2021 4:43 PM	File folder	

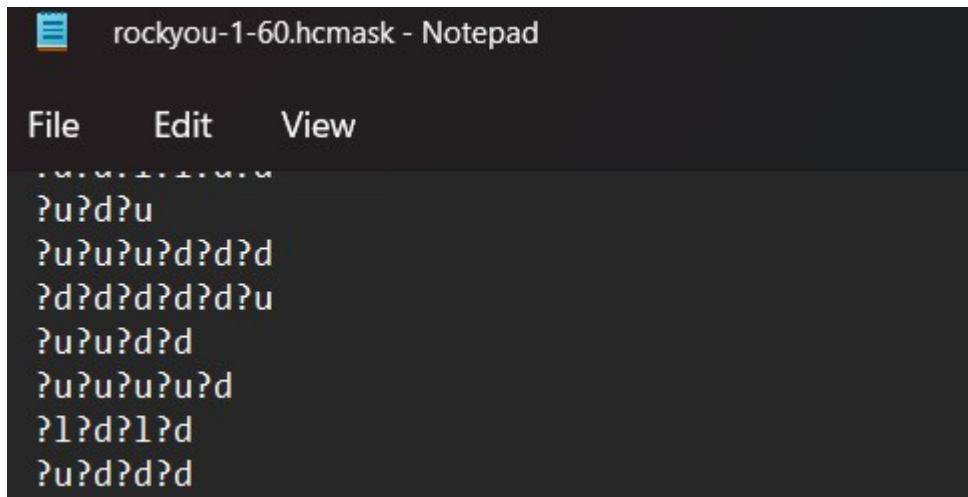
When you install Hashcat, a folder named “masks” is created in the main directory, including a few mask files you can use for mask attacks. These files are created with the most common passwords formats found in major leaks.

Name	Date modified	Type	Size
8char-1l-1u-1d-1s-compliant.hcmask	11/21/2021 4:43 PM	HCMASK File	718 KB
8char-1l-1u-1d-1s-noncompliant.hcmask	11/21/2021 4:43 PM	HCMASK File	435 KB
hashcat-default.hcmask	11/21/2021 4:43 PM	HCMASK File	1 KB
rockyou-1-60.hcmask	11/21/2021 4:43 PM	HCMASK File	11 KB
rockyou-2-1800.hcmask	11/21/2021 4:43 PM	HCMASK File	42 KB
rockyou-3-3600.hcmask	11/21/2021 4:43 PM	HCMASK File	58 KB

You can for example try the file rockyou-1-60.hcmask. Rock You is probably one of the major password leak, and several files have been created for you in this folder. You can use this file with the command:

```
hashcat -m 0 -a 3 hashes.txt masks\rockyou-1-60.hcmask
```

It will test all the masks contained in this file one by one and print the corresponding matches found in your hashes.txt file:



```
rockyou-1-60.hcmask - Notepad
File Edit View
?u?d?u
?u?u?u?d?d?d
?d?d?d?d?d?u
?u?u?d?d
?u?u?u?u?d
?l?d?l?d
?u?d?d?d
```

Create your own files

An alternative is to create your own files, following the same format (one key space per line). This is often the fastest way to crack passwords when you have a few insights about the passwords formats.

Create the file under the masks folder (or anywhere else really), put a few key spaces you want to try in it, and run the same command, just changing the file name.

Default files are good, but won't necessarily fit the info you already have. For example, RockYou try a lot of short passwords. If you have a list of passwords that you know are at least 10 characters long, you'll lose a ton of time testing shorter ones, while there is no need to.

Want to know more about hashcat? Read my other articles on the topic:

- [How to Install and Use Hashcat to Decrypt MD5? \(Tutorial\)](#)
- [How To Install Hashcat on Windows](#)
- [How To Install Hashcat On Ubuntu](#)

Whenever you're ready for more security, here are things you should think about:

- **Break free from Gmail**: You should be able to choose what happens to your data. With Proton, only you can read your emails. [Get private email](#).
- **Protect yourself online**: Use a high-speed Swiss VPN that safeguards your privacy. Open-source, no activity logs. [Get Proton VPN risk-free](#).
- **Master Linux commands**: A sure method to learn (and remember) Linux commands. Useful ones only, one at a time, with clear explanations. [Download the e-book](#).