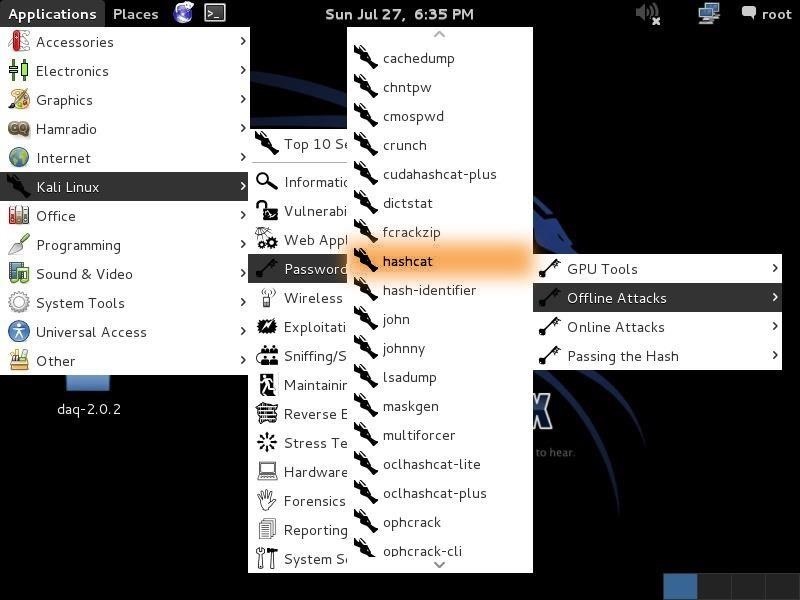
I want to introduce you to one of the newest and best designed password crackers out there – hashcat. The beauty of hashcat is in its design, which focuses on speed and versatility. It enables us to crack multiple types of hashes, in multiple ways, very fast.

Passwords are often stored in a one-way encryption called hashes. There are multiple ways of obtaining these hashes. Once we can grab the hash, the next step becomes one of finding an effective and efficient way of cracking it. There are numerous tools, but hashcat is unique in its design and versatility, so let's take a look at how it works.

**Step 1: Fire Up Kali & Open Hashcat**

Let's start by firing up Kali and opening hashcat. Go to Applications -> Kali Linux -> Password Attacks -> Offline Attacks -> **hashcat**, as seen below.

[](https://img.wonderhowto.com/img/original/57/82/63542242877265/0/635422428772655782.jpg)

When we click on the hashcat menu item, it opens the help screen.

[](https://img.wonderhowto.com/img/original/02/67/63542243020251/0/635422430202510267.jpg)

At the top of the screen, you can see the basic hashcat syntax:

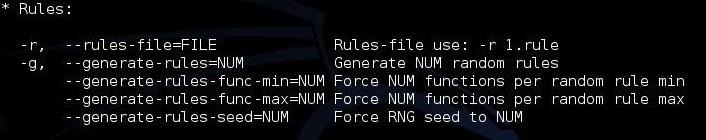
**kali > hashcat options hashfile mask|wordfiles|directories**

We can see some of the options for hashcat displayed below the basic syntax. Some of the most important of these are **-m** (the hashtype) and **-a** (attack mode). In general, we will need to use both of these options in most password cracking attempts with hashcat.

**Step 2: More Extensive Options**

If we scan a bit further down this hashcat help screen, we can see more options. The first two below are some of the key options that hashcat enables.

First, hashcat enables **rules** that allow us to apply specifically designed rules to use on our wordlist file. These rules can take our wordlist file and apply capitalization rules, special characters, word combinations, appended and prepended numbers, and so on. Each of these will help us to break passwords that have been made more complex to avoid dictionary attacks.

[](https://img.wonderhowto.com/img/original/10/68/63542245220156/0/635422452201561068.jpg)

The next stanza shows us **custom character sets**. This enables us to set the character set that we want to use to crack the passwords. If we know the company's or institution's password policy, we can choose a subset of all characters to meet their policy and speed up our cracking. For instance, if a company allows an all-numeric character set, choose to crack the hashes with just numbers. These types of passwords are VERY easy to crack.

[](https://img.wonderhowto.com/img/original/84/39/63542245229718/0/635422452297188439.jpg)

The next screen includes some of the more obscure options, including the **output file type**, the **debug mode** and the **built-in character sets**.

[](https://img.wonderhowto.com/img/original/67/23/63542245080484/0/635422450804846723.jpg)

Finally, we have to chose the type of hash we are trying to crack. Hashcat gives us numerous options. When we get ready to crack the hash, we need to designate in our command what type of hash we are working with by giving hashcat the number associated with the hash type. Here we can see a list of some of the hash types hashcat can work with.

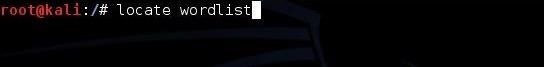
[](https://img.wonderhowto.com/img/original/30/70/63542245737250/0/635422457372503070.jpg)

**Step 3: Choose Your Wordlist**

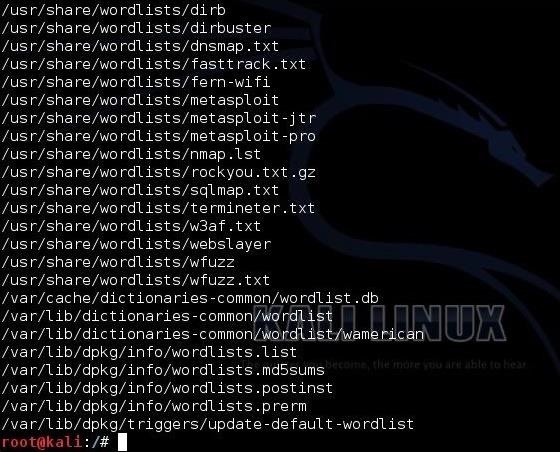
In this tutorial, we will be using a simple dictionary attack on some Linux hashes. To do so, we need a wordlist to work from. There are literally thousands of wordlists available on the web, but Kali has numerous wordlists built right in, so let's try using one of those.

To find the built in wordlists in Kali, we can type:

**kali > locate wordlist**

[](https://img.wonderhowto.com/img/original/57/54/63542246161515/0/635422461615155754.jpg)

When we do, we can see that there are dozens of wordlists available.

[](https://img.wonderhowto.com/img/original/56/61/63542246234765/0/635422462347655661.jpg)

I will be using the wordlist built for sqlmap, which has over one million words and hybrid words.

**Step 4: Grab the Hashes**

In the next step, we need to grab the hashes on our Kali system. If we are logged in as root, we can see and grab the hashes. In Linux, the hashes are stored in the */etc/shadow* file, so if we type:

**kali > tail /etc/shadow**

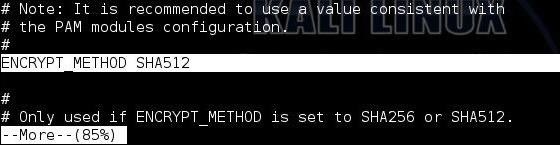
We can see the shadow file with the hashes, as below.

[](https://img.wonderhowto.com/img/original/23/31/63542246344484/0/635422463444842331.jpg)

Next, we need to know what type of hashing the system is using. In Linux, we go to the */etc/login.defs* to view what encryption type the system is using. We open that file by typing:

**kali > more /etc/login.defs**

When we navigate about 85% down the file, we can see that Kali is using SHA512 encryption. This is important, as we will need to tell hashcat this information when we are ready to crack the hashes.

[](https://img.wonderhowto.com/img/original/88/12/63542246444499/0/635422464444998812.jpg)

**Step 5: Crack the Hashes!**

Now, that we know the basics of hashcat, where the hashes are located and the type of encryption, we are ready to begin cracking the hashes.

Let's first put those hashes into a separate file we will name *hash.lst*.

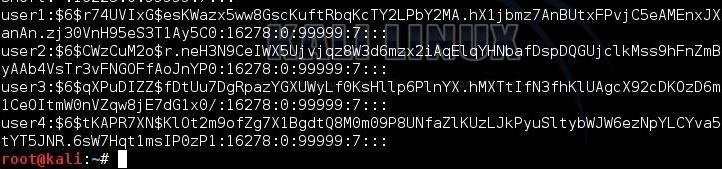
**kali > cp /etc/shadow hash.lst**

To make sure that they were copied over, let's check by typing:

**more hash.lst**

[](https://img.wonderhowto.com/img/original/88/82/63542246512796/0/635422465127968882.jpg)

As we can see, the hashes have been copied over to the hash.lst file.

[](https://img.wonderhowto.com/img/original/85/05/63542246573343/0/635422465733438505.jpg)

To prepare this file for cracking, we need to remove all of the information in this file, except the hashes. The */etc/shadow* file includes the username, then the salted hash, and then information about the applicable user policy. We need to remove all that information leaving just the hash.

We can see that this file starts with the username, i.e., "user1", "user2", etc. Open this file in your favorite text editor (vim, vi, leafpad) and delete the username and the following colon. Then, go to the end of the line and remove the information after the hash that starts with a colon (:). Now we will have a file with just the hashes and nothing else.

In the final step, we can now start cracking the hashes. Here's the command I used.

**kali > hashcat -m 1800 -a 0 -o cracked.txt --remove hash.lst /usr/share/sqlmap/txt/wordlist.txt**

* **-m 1800** designates the type of hash we are cracking (SHA-512)
* **-a 0** designates a dictionary attack
* **-o cracked.txt** is the output file for the cracked passwords
* **--remove** tells hashcat to remove the hash after it has been cracked
* **hash.lst** is our input file of hashes
* **/usr/share/sqlmap/txt/wordlist.txt** is the absolute path to our wordlist for this dictionary attack

[](https://img.wonderhowto.com/img/original/61/10/63542246802015/0/635422468020156110.jpg)

Once the cracking process starts, we can hit <enter> to get an update on the process. When hashcat has completed its work, you will see a screen like below where hashcat announces that it has recovered all my hashes after 9 :47:16 of work.

[](https://img.wonderhowto.com/img/original/33/06/63542246859517/0/635422468595173306.jpg)

Now, we only need to open the cracked.txt file to view our cracked passwords!

Hashcat may be the world's best password cracking tool right now, so take some time to get to know it. It has many more features that we have not yet touched on, and a version that uses your GPU (**oclhashcat**) that can crack passwords many times faster than your CPU can!

https://null-byte.wonderhowto.com/how-to/hack-like-pro-crack-passwords-part-3-using-hashcat-0156543/