

Lab Guide

Time estimate: 30–45 minutes · Goals: Install Hashcat, learn basic commands and parameters, run a test attack on a known hash

Lab Goals

- Learn basic Hashcat functions and interface
- Install Hashcat on your operating system
- Learn basic commands and parameters
- Run a test dictionary attack on a known MD5 hash

Task 1: Install Hashcat

Installation Instructions

Choose the installation method for your operating system:

macOS

```
# Using Homebrew
brew install hashcat

# Verify installation
hashcat --version
```

Linux (Debian/Ubuntu)

```
# Using apt
sudo apt update
sudo apt install hashcat

# Verify installation
hashcat --version
```

Linux (Arch)

```
# Using pacman
sudo pacman -S hashcat

# Verify installation
hashcat --version
```

Windows

1. Download the latest release from <https://hashcat.net/hashcat/>
2. Extract the archive to a folder (e.g., C:\hashcat)
3. Open Command Prompt or PowerShell
4. Navigate to the folder: `cd C:\hashcat`
5. Run: `hashcat.exe --version`

✓ **Check:** Running `hashcat --version` prints a version number (e.g., `v6.2.6`).

Task 2: Explore Basic Commands

Step 2.1 — Check Available Devices

The `-I` flag displays information about compute devices detected by Hashcat (CPUs, GPUs, etc.):

```
hashcat -I
```

You should see output listing your available devices. Example:

```
hashcat (v6.2.6) starting in backend information mode
```

```
OpenCL Info:
```

```
=====
```

```
OpenCL Platform ID #1
```

```
Vendor...: Apple
```

```
Name.....: Apple
```

```
Version..: OpenCL 1.2
```

```
Backend Device ID #1
Type.....: GPU
Vendor.ID.....: 32
Vendor.....: Apple
Name.....: Apple M1
...
```

✓ **Check:** At least one compute device is listed. If no GPU is detected, Hashcat will fall back to CPU mode.

Step 2.2 — View Help

The `--help` flag displays the full list of options, attack modes, and hash types:

```
hashcat --help
```

This produces extensive output. Take note of these key sections:

- **Attack Modes** — the different `-a` options (0 = dictionary, 3 = mask, etc.)
- **Hash Types** — the full list of `-m` values and their corresponding hash algorithms
- **Output Options** — flags for controlling where and how results are saved

✓ **Check:** You can identify the `-m` value for MD5 (it is `0`).

Task 3: Crack a Known MD5 Hash

Goal

Perform a dictionary attack on a known MD5 hash to verify that your Hashcat installation works correctly.

Step 3.1 — Create the Hash File

Create a file called `hashes.txt` containing the MD5 hash of the word "password":

```
echo "5f4dcc3b5aa765d61d8327deb882cf99" > hashes.txt
```

About This Hash

The string `5f4dcc3b5aa765d61d8327deb882cf99` is the MD5 hash of the plaintext word "password". This is one of the most commonly found hashes in leaked databases, because "password" is one of the most commonly used passwords worldwide.

Step 3.2 — Create a Simple Wordlist

Create a small dictionary file called `example.dict` with a few common passwords:

```
cat <<EOF > example.dict
123456
password
12345678
qwerty
abc123
monkey
master
dragon
letmein
football
EOF
```

Step 3.3 — Run the Dictionary Attack

Execute the following command to crack the hash:

```
hashcat -m 0 -a 0 -o cracked.txt hashes.txt example.dict
```

Let's break down each part of this command:

Part	Meaning
<code>hashcat</code>	The Hashcat executable
<code>-m 0</code>	Hash type: MD5 (mode 0)
<code>-a 0</code>	Attack mode: Dictionary (mode 0)

Part	Meaning
<code>-o cracked.txt</code>	Write recovered passwords to this file
<code>hashes.txt</code>	Input file containing target hashes
<code>example.dict</code>	Wordlist file to use for the attack

Step 3.4 — Verify the Results

Check the output file to see the cracked password:

```
cat cracked.txt
```

Expected output:

```
5f4dcc3b5aa765d61d8327deb882cf99:password
```

The format is `hash:plaintext`, showing that the hash `5f4dcc3b...` corresponds to the password "password".

✓ **Check:** The file `cracked.txt` contains `5f4dcc3b5aa765d61d8327deb882cf99:password`.

Troubleshooting

- **"Hash not found" or "Already cracked"** — Hashcat caches results in a potfile. If you've already cracked this hash, use `--potfile-disable` to ignore the cache, or use `hashcat -m 0 hashes.txt --show` to display previously cracked hashes.
- **"No devices found"** — Ensure you have appropriate GPU drivers installed. On macOS, Metal is used automatically. On Linux/Windows, you need OpenCL or CUDA drivers.
- **"Token length exception"** — The hash format doesn't match the specified `-m` mode. Double-check that your hash file contains only the hash with no extra spaces or characters.

Finishing Up

Once you have completed all three tasks, verify the following:

Self-Check

- ✓ Hashcat is installed and `hashcat --version` works
- ✓ `hashcat -I` lists at least one compute device
- ✓ You can locate MD5 in the `hashcat --help` output and know its mode number
- ✓ You successfully cracked the MD5 hash and `cracked.txt` contains the correct result
- ✓ You can explain what `-m 0`, `-a 0`, and `-o` mean