

Introduction to Hashcat

A comprehensive introduction to the world's most popular open-source password recovery tool — from basic concepts to your first hash crack.

Learning Objectives

- Understand **what Hashcat is** and why it exists
 - Learn the **history and evolution** of the tool
 - Explore the **main capabilities** and supported attack types
 - Install Hashcat and run **basic commands**
 - Perform a **test dictionary attack** on a known MD5 hash
-

What Is Hashcat?

Hashcat is an **open-source password recovery tool** that uses modern hash function recognition methods to recover lost passwords or verify the security of existing ones.

Key Characteristics

- Free and open-source (MIT license)
- Extremely fast — GPU-accelerated
- Supports 350+ hash types
- Cross-platform (Windows, Linux, macOS)

Who Uses It?

- **Security researchers** — studying password security
- **Penetration testers** — auditing organizations
- **System administrators** — verifying password policies
- **Forensic analysts** — recovering evidence

Notes: Hashcat's primary goal is to recover lost passwords or to verify the security of existing password hashes. It should only be used on systems you own or have written authorization to test.

History and Development

Hashcat was created to improve and accelerate the password recovery process. Over the years, it has grown from a simple CPU-based cracker into the most powerful hash recovery tool available.

Key Milestones

- **Early versions** — CPU-only password recovery tool with limited hash support
- **oclHashcat era** — Introduction of GPU acceleration using OpenCL, dramatically increasing speed
- **Hashcat 3.0+ (2016)** — CPU and GPU versions merged into a single unified tool
- **Ongoing development** — Constant updates adding new hash types and optimizing recognition processes

Open-Source Milestone

In 2015, Hashcat was made fully open-source, allowing the community to contribute hash algorithms, performance optimizations, and bug fixes. This accelerated development

significantly.

Main Capabilities

Hashcat stands out due to its extensive feature set and broad compatibility across platforms and hardware.

Platform Support

- Windows
- Linux
- macOS

Hardware Acceleration

- CPU (all platforms)
- NVIDIA GPU (CUDA)
- AMD GPU (OpenCL)
- Apple Metal (macOS)

Core Features

- **350+ hash types** — MD5, SHA-1, SHA-256, bcrypt, NTLM, WPA, and many more
- **Multiple attack modes** — dictionary, mask, combinator, rule-based, and hybrid attacks
- **Session management** — pause, resume, and restore cracking sessions

- **Distributed cracking** — split workloads across multiple machines
-

Attack Types Overview

Hashcat supports several attack modes, each suited for different scenarios.

Mode	Flag	Description
Dictionary Attack	<code>-a 0</code>	Tries every word from a wordlist file
Combinator Attack	<code>-a 1</code>	Combines words from two dictionaries
Brute-Force / Mask	<code>-a 3</code>	Tries all combinations matching a pattern (mask)
Hybrid (Dict + Mask)	<code>-a 6</code>	Appends mask patterns to dictionary words
Hybrid (Mask + Dict)	<code>-a 7</code>	Prepends mask patterns to dictionary words

Notes: The dictionary attack (-a 0) is the most commonly used starting point. Rule-based attacks add transformations (capitalize, append digits, etc.) to dictionary words, making them extremely powerful.

Quick Demo: Your First Crack

Let's walk through cracking a known MD5 hash using a dictionary attack.

Step 1: Create a hash file

```
# The MD5 hash of "password"
echo "5f4dcc3b5aa765d61d8327deb882cf99" > hashes.txt
```

Step 2: Run the attack

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

Step 3: Check results

```
cat cracked.txt
# Output: 5f4dcc3b5aa765d61d8327deb882cf99:password
```

Command Breakdown

- `-m 0` — Hash type: MD5
 - `-a 0` — Attack mode: Dictionary
 - `-o cracked.txt` — Output file for recovered passwords
 - `hashes.txt` — File containing target hashes
 - `example.dict` — Wordlist / dictionary file
-

Essential Commands

Before diving deeper, familiarize yourself with these fundamental Hashcat commands.

System Information

```
# Show detected compute devices (GPUs, CPUs)
hashcat -I

# Display full help with all options
hashcat --help
```

Common Flags Reference

Flag	Purpose	Example
<code>-m</code>	Hash type	<code>-m 0</code> (MD5)
<code>-a</code>	Attack mode	<code>-a 0</code> (dictionary)
<code>-o</code>	Output file	<code>-o results.txt</code>

Flag	Purpose	Example
<code>--show</code>	Show already cracked	<code>hashcat -m 0 hash.txt --show</code>
<code>-w</code>	Workload profile	<code>-w 3</code> (high performance)

Lab & Homework

Lab Exercises

1. **Install Hashcat** on your operating system and verify the installation
2. **Run basic commands** — `hashcat -I` and `hashcat --help` to explore options
3. **Crack an MD5 hash** — perform a dictionary attack on a known hash and verify the output

Homework Tasks

1. Research and list 5 hash types supported by Hashcat with their `-m` codes
2. Explain the difference between dictionary and mask attacks in your own words
3. Create and crack your own MD5 hash using a custom wordlist

Reference

Official Hashcat FAQ: [hashcat.net/wiki — FAQ](https://hashcat.net/wiki/FAQ)

