

Forum:[Hack Forums](#)

Thread: [What is and how to work with Hashcat \[Ethical Hacking\]](#)

JoseMar - January 17, 2025, 10:04 am

What is and how to work with Hashcat [Ethical Hacking]

Hashcat is a powerful password cracker for ethical hacking and cybersecurity. This is a guide to its features, usage and examples so you can use it in your lab.

Hashcat is called the “world’s fastest password cracker”. It’s an open source software that uses CPU and GPU to crack password hashes. Ethical hackers and security professionals use Hashcat to test password implementations and make sure systems are secure from unauthorized access.

Features

- 1: Cross Platform: Windows, Linux and macOS
- 2: Hardware Acceleration: GPU and CPU cracking for speed and flexibility.
- 3: Hash Support: 300+ hashing algorithms, including MD5, SHA1, bcrypt, NTLM and more.
- 4: Modular Attack Modes: Dictionary Attack: Hashes against a wordlist.

Brute Force Attack: Tries all possible combinations.

Mask Attack: Targets specific patterns in passwords.

Hybrid Attack: Dictionary and brute force.

Rule-Based Attack: Modifies dictionary words with predefined rules.

- 5: Community-Supported: Open-source with updates and community.

Installation and Setup

For Linux:

```
Code sudo apt update  
sudo apt install hashcat
```

For Windows:

Download the executable from Hashcat's Official Website.

Extract the ZIP file.

Open a command prompt in the extracted folder.

Test Environment Preparation:

Create a file, hashes.txt, containing hashed passwords. For example:

MD5 hash for "password123": 482c811da5d5b4bc6d497ffa98491e38

Use wordlists like rockyou.txt, available from ethical sources or download from github: Wordlist 2024 update

Online Tools Websites for discover or Crack Hash

Code <https://crackstation.net>
https://hashes.com/en/tools/hash_identifier

hash-identifier Online:
<https://www.tunnelsup.com/hash-analyzer/>
... most used are this.

Examples of Hashcat Usage

Example 1: Basic Dictionary Attack

Code `hashcat -m 0 -a 0 hashes.txt rockyou.txt-m 0`: MD5 hash type.

`-a 0`: Dictionary attack mode.

hashes.txt: File containing hashed passwords.

rockyou.txt: Wordlist file.

Example 2: Brute Force Attack

Try all possible combinations of 8-character alphanumeric passwords:

Code `hashcat -m 0 -a 3 hashes.txt ?a?a?a?a?a?a?a?a`: Represents all possible characters.

Adjust the number of `?a` to match the length of the target password.

Example 3: Mask Attack for Specific Patterns

Crack passwords like `abc####` (where `#` is a digit):

Code `hashcat -m 0 -a 3 hashes.txt abc?d?d?d?d?d`: Represents digits (0-9).

Example 4: Rule-Based Attack

Apply rules to modify dictionary words. For example, add numbers to the end of each word:

Code `hashcat -m 0 -a 0 hashes.txt rockyou.txt -r rules/best64.rule-r`: Specifies the rule file.

Example 5: Hybrid Attack (Dictionary + Brute Force)

Append two digits to each word in a dictionary:

Code `hashcat -m 0 -a 6 hashes.txt rockyou.txt ?d?d-a 6`: Hybrid attack mode with dictionary and brute force suffix.

Example 6: Cracking NTLM Hashes

Target Windows NTLM hashes:

Code `hashcat -m 1000 -a 0 ntlm_hashes.txt rockyou.txt-m 1000`: NTLM hash type.

Example 7: Testing SHA256 Hashes

Code `hashcat -m 1400 -a 0 sha256_hashes.txt rockyou.txt-m 1400`: SHA256 hash type.

Example 8: Case-Sensitive Mask Attack

Passwords starting with an uppercase letter followed by lowercase and digits:

Code `hashcat -m 0 -a 3 hashes.txt ?u?l?l?l?d?d?d?u`: Uppercase letters.

?l: Lowercase letters.

?d: Digits.

Example 9: Multi-Hash Cracking

Crack multiple hashes from a file:

Code `hashcat -m 0 -a 0 multiple_hashes.txt rockyou.txt` Supports simultaneous cracking of multiple

hashes.

Example 10: Restore a Session

If interrupted, restore your session:

Code `hashcat --session session_name --restoreUse --session` to name and manage sessions.

More Practical Hash crack with proofs:

Possible algorithms: sha512crypt \$6\$, SHA512 (Unix)

Code `hashcat -m 1800 hash.txt /usr/share/wordlists/rockyou.txt` This is the hash:

Code

`$6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPMAXi4bJMI9be.cfi3/qxlf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02`. We use `-m 1800`

Hash: bcrypt (We need to wait longer)

Code `hashcat -m 3200 hash.txt /usr/share/wordlists/rockyou.txt`

Hash MD5 or MD4 -> hash-identifier told me that this is md5 but hashcat is not working with md5 command so I changed to MD4 `-m 900`

Code `hashcat -m 900 hash.txt /usr/share/wordlists/rockyou.txt`

Ethical Hacking Best Practices

Always Ask: Get explicit permission before testing anything.

Use Isolation: Test in virtual labs.

Log Everything: Document your steps, results and recommendations.

Educate on Passwords: Share password best practices and secure habits.

Stay Current: Update Hashcat and learn new attack vectors.

Conclusion

Hashcat is an awesome tool for ethical hacking and password testing. Practice with these examples and you'll get hands on experience in finding vulnerabilities and promoting better security. Always use Hashcat responsibly, respect privacy and legal boundaries.

NullSector - January 17, 2025, 10:16 am

(Jan 17, 2025 11:04 AM)JoseMar Wrote: [\[CITE\]](#) What is and how to work with Hashcat [Ethical Hacking]

Hashcat is a powerful password cracker for ethical hacking and cybersecurity. This is a guide to its features, usage and examples so you can use it in your lab.

Hashcat is called the "world's fastest password cracker". It's an open source software that uses CPU and GPU to crack password hashes. Ethical hackers and security professionals use Hashcat to test password implementations and make sure systems are secure from unauthorized access.

Features

- 1: Cross Platform: Windows, Linux and macOS
- 2: Hardware Acceleration: GPU and CPU cracking for speed and flexibility.
- 3: Hash Support: 300+ hashing algorithms, including MD5, SHA1, bcrypt, NTLM and more.
- 4: Modular Attack Modes: Dictionary Attack: Hashes against a wordlist.

Brute Force Attack: Tries all possible combinations.

Mask Attack: Targets specific patterns in passwords.

Hybrid Attack: Dictionary and brute force.

Rule-Based Attack: Modifies dictionary words with predefined rules.

- 5: Community-Supported: Open-source with updates and community.

Installation and Setup

For Linux:

```
Code sudo apt update  
sudo apt install hashcat
```

For Windows:

Download the executable from Hashcat's Official Website.

Extract the ZIP file.

Open a command prompt in the extracted folder.

Test Environment Preparation:

Create a file, hashes.txt, containing hashed passwords. For example:

MD5 hash for "password123": 482c811da5d5b4bc6d497ffa98491e38

Use wordlists like rockyou.txt, available from ethical sources or download from github: Wordlist 2024 update

Online Tools Websites for discover or Crack Hash

Code <https://crackstation.net>

https://hashes.com/en/tools/hash_identifier

hash-identifier Online:

<https://www.tunnelsup.com/hash-analyzer/>

... most used are this.

Examples of Hashcat Usage

Example 1: Basic Dictionary Attack

Code `hashcat -m 0 -a 0 hashes.txt rockyou.txt-m 0`: MD5 hash type.

`-a 0`: Dictionary attack mode.

hashes.txt: File containing hashed passwords.

rockyou.txt: Wordlist file.

Example 2: Brute Force Attack

Try all possible combinations of 8-character alphanumeric passwords:

Code `hashcat -m 0 -a 3 hashes.txt ?a?a?a?a?a?a?a?a`: Represents all possible characters.

Adjust the number of `?a` to match the length of the target password.

Example 3: Mask Attack for Specific Patterns

Crack passwords like `abc####` (where `#` is a digit):

Code `hashcat -m 0 -a 3 hashes.txt abc?d?d?d?d?d`: Represents digits (0-9).

Example 4: Rule-Based Attack

Apply rules to modify dictionary words. For example, add numbers to the end of each word:

Code `hashcat -m 0 -a 0 hashes.txt rockyou.txt -r rules/best64.rule-r`: Specifies the rule file.

Example 5: Hybrid Attack (Dictionary + Brute Force)

Append two digits to each word in a dictionary:

Code `hashcat -m 0 -a 6 hashes.txt rockyou.txt ?d?d-a 6`: Hybrid attack mode with dictionary and brute force suffix.

Example 6: Cracking NTLM Hashes

Target Windows NTLM hashes:

Code `hashcat -m 1000 -a 0 ntlm_hashes.txt rockyou.txt-m 1000`: NTLM hash type.

Example 7: Testing SHA256 Hashes

Code `hashcat -m 1400 -a 0 sha256_hashes.txt rockyou.txt-m 1400`: SHA256 hash type.

Example 8: Case-Sensitive Mask Attack

Passwords starting with an uppercase letter followed by lowercase and digits:

Code `hashcat -m 0 -a 3 hashes.txt ?u?!?!?!?d?d?d?u`: Uppercase letters.

?!: Lowercase letters.

?d: Digits.

Example 9: Multi-Hash Cracking

Crack multiple hashes from a file:

Code `hashcat -m 0 -a 0 multiple_hashes.txt rockyou.txt` Supports simultaneous cracking of multiple hashes.

Example 10: Restore a Session

If interrupted, restore your session:

Code `hashcat --session session_name --restoreUse --session to name and manage sessions.`

More Practical Hash crack with proofs:

Possible algorithms: sha512crypt \$6\$, SHA512 (Unix)

Code `hashcat -m 1800 hash.txt /usr/share/wordlists/rockyou.txt` This is the hash:

Code

`$6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPMAXi4bJMI9be.cfi3/qxlf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02`. We use `-m 1800`

Hash: bcrypt (We need to wait longer)

Code `hashcat -m 3200 hash.txt /usr/share/wordlists/rockyou.txt`

Hash MD5 or MD4 -> hash-identifier told me that this is md5 but hashcat is not working with md5 command so I changed to MD4 `-m 900`

Code `hashcat -m 900 hash.txt /usr/share/wordlists/rockyou.txt`

Ethical Hacking Best Practices

Always Ask: Get explicit permission before testing anything.

Use Isolation: Test in virtual labs.

Log Everything: Document your steps, results and recommendations.

Educate on Passwords: Share password best practices and secure habits.

Stay Current: Update Hashcat and learn new attack vectors.

Conclusion

Hashcat is an awesome tool for ethical hacking and password testing. Practice with these examples and you'll get hands on experience in finding vulnerabilities and promoting better security. Always use Hashcat responsibly, respect privacy and legal boundaries.

6a52e854adab78abfe1781cb02eb03386f70ae39994d0e1870d46e42a39ecca7