UNIX PROJECT TUTORIAL: KALI LINUX - HYDRA AND HASHCAT
GIDEON ELEBODA
DAN LULKIN

Tutorial: Cracking Password Hashes Using Hashcat

Introduction to Hashing and Hashcat

1. **What is Hashing?**
    - Hashing is the process of converting an alphanumeric string into a fixed-length string using a hash function.
    - Hash lengths are constant and irreversible, making them useful for verifying data integrity and securing passwords.
    - Example: When you create a password on a website, the password is hashed and stored. When you log in, your input is hashed and compared to the stored hash.
2. **What is Hashcat?**
    - Hashcat is a powerful and fast password recovery tool that leverages GPU acceleration to crack hashes.
    - It supports a variety of hash algorithms (e.g., MD5, SHA1, SHA256) and attack modes.

Step-by-Step Guide to Using Hashcat

1. Generate Hashes

To generate sample hashes for testing, you can use an online tool like Browserling.
Example: Generate the MD5 hash of "Password123".

2. Prepare Files

Create a file to store your hashes:
echo "42f749ade7f9e195bf475f37a44cafcb" > md5.txt

3. Access Wordlists

Hashcat requires wordlists to test password guesses against hashes.
Common wordlists like rockyou.txt are included in Kali Linux and can be found in /usr/share/wordlists/.

Unzip the file if it is compressed:
sudo gunzip /usr/share/wordlists/rockyou.txt.gz

4. Run Hashcat

Basic syntax for Hashcat:
hashcat -m <hash_type> -a <attack_mode> <hash_file> <wordlist>

- -m: Specifies the hash type.
- -a: Specifies the attack mode (0 = Dictionary Attack, 1 = Combinator Attack, etc.).
- <hash_file>: File containing the hashes (e.g., md5.txt).
- <wordlist>: Path to the wordlist (e.g., /usr/share/wordlists/rockyou.txt).

Example: Cracking an MD5 hash with a dictionary attack:
hashcat -m 0 -a 0 md5.txt /usr/share/wordlists/rockyou.txt

---

5. Advanced Attack Modes

Combinator Attack combines words from two dictionaries:
hashcat -m <hash_type> -a 1 <hash_file> <dict1> <dict2>

- Create two files with 4 words each.
- Generate a hash for the chosen combination (e.g., file1 word1 + file2 word3).
- Store the hash in a file and run the command.

If you encounter an error, use the --force flag:
hashcat --force -m <hash_type> -a 1 <hash_file> <dict1> <dict2>

---

6. Optimize GPU Utilization

If the wordlist is too small, performance may suffer due to insufficient workload for the GPU. Use modifiers like rules, masks, or hybrid attacks to amplify password candidates.

---

7. Cleanup

After cracking the hashes, delete sensitive files to avoid misuse:

rm md5.txt sha1.txt