Lab - Using Password Tools

Objectives

In this lab, you will complete the following objectives:

- Part 1: Investigate Password Attacks
- Part 2: Crack Hashes with Hashcat Dictionary Attacks
- Part 3: Crack Hashes with John the Ripper Using Dictionary and Brute Force Attacks
- Part 4: Crack Hashes using RainbowCrack and Rainbow Tables

Background / Scenario

Passwords are vulnerable to attack. Passwords are usually stored as encrypted hashes. An attacker can capture the hashes sent over the network using sniffing tools or can gain access to the files containing password hashes on vulnerable systems. When the attacker has the hashes, they can then apply dictionary, rainbow table, and brute force attacks against them offline to crack the hash to recover the plaintext passwords. There are many password attack tools included with Kali Linux. This lab will look at three popular tools; Hashcat, John the Ripper, and RainbowCrack.

Required Resources

- Kali VM customized for the Ethical Hacker course
- Internet access

Instructions

Part 1: Investigate Password Attacks

Step 1: Log into Kali Linux and verify the environment.

a. Log into Kali using **kali** as the username and password.
b. Select **Applications > 05 – Password Attacks**.
In the Kali Password Attacks menu, which four subcategories of password attack tools are available?

Answer Area

<!----><!---->

Step 2: Examine the available password attack tools.

a. Click each attack subcategory and review the available attack tools.
b. Hover the cursor over each tool. Note that some tools have a popup text box containing a brief description of the tool. You can also search for the tools in the Kali Tools page to learn more

about them and what they do.

Which tool is a Microsoft password cracker that uses rainbow tables? Which subcategory contains this tool?

Answer Area

<!----><!---->

Part 2: Crack Hashes with Hashcat Dictionary Attacks

Step 1: Create a file that contains MD5 hashes to be cracked.

First, some MD5 hashes of passwords are needed. In an actual exploit, an attacker will have already compromised a vulnerable system to obtain a password file containing stored password hashes to be cracked offline. In this step you simulate this by creating a password file that contains the hashes you will crack in an upcoming step.

a. In a terminal window, create five target hashes by entering the following commands at the prompt:

```
echo -n 'Password' | md5sum | awk '{ print $1 }' > my_pw_hashes.txt

echo -n 'Password123' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt

echo -n 'Letmein!' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt

echo -n 'ilovedogs' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt

echo -n '1234abcd' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt
```

Note that the passwords vary in complexity.

The hashes generated are written to the **my_pw_hashes.txt** file.

b. Next, check the password hashes that you just created by entering the **cat** command.

The output should look similar to that below:

```
┌──(kali㉿Kali)-[~]
└─$ cat my_pw_hashes.txt
dc647eb65e6711e155375218212b3964
42f749ade7f9e195bf475f37a44cafcb
e85a3b267e94f3721117fc7ac54fbeba
33830b8b7fd414b12c208c4de5055464
ef73781effc5774100f87fe2f437a435
```

Step 2: Start Hashcat in Kali.

a. Open a new Kali console and enter the command: **man hashcat**.

This opens the Hashcat manual.

b. Review the options available in the first man page.
What is specified with the **-m** and **-a** options?

Answer Area

<!----><!---->

c. Scroll through the man page output to find the values that can be supplied to each of these options.

You will use these options soon in upcoming steps.

Using the hashcat man pages, which hash type and attack mode would you use to crack the password hashes in the my_pw_hashes.txt file? Explain.

Answer Area

<!----><!---->

Step 3: View available wordlists.

Kali comes with several wordlists built in. Hashcat needs to use a wordlist to crack the hashes.

a. To view the built-in wordlists, enter the command: **ls -lh /usr/share/wordlists/**.

```
┌──(kali㉿Kali)-[~]
└─$ ls -lh /usr/share/wordlists/
```

This lists the wordlists that are distributed with Kali. We will use the **rockyou.txt** word list. The rockyou.txt wordlist is a password dictionary that contains more than 14 million passwords.

What needs to be done to the rockyou.txt.gz file before the wordlist text file can be used?

Answer Area

<!----><!---->

b. Change the directory to **/usr/share/wordlists** by entering the command:

```
┌──(kali㉿Kali)-[~]
└─$ cd /usr/share/wordlists
```

c. Extract the rockyou.txt.gz file using the **gzip** command:

```
┌──(kali㉿Kali)-[/usr/share/wordlists]
└─$ sudo gzip -d rockyou.txt.gz
```

d. List the contents of the directory as was done previously using the **ls** command. Verify that the **rockyou.txt** file is now unzipped.

```
┌──(kali㉿Kali)-[/usr/share/wordlists]
└─$ ls
```

e. Use the **more** command, followed by the file name, to view the contents of the file to see some of the passwords that hashcat will use to crack your hashes.

```
┌──(kali㉿Kali)-[/usr/share/wordlists]
└─$ more rockyou.txt
```

Wordlists for cracking hashes or brute forcing logins are often collected from password dumps that publicly disclose stolen user account information. Scroll through the output to get a sense of the file contents.

What seems to be a popular type of password? How could this trend be useful to a penetration tester?

Answer Area

<!----><!---->

f. Press **q** or **Ctrl-z** to exit the file contents.

g. Return to the home directory.

```
┌──(kali㉿Kali)-[/usr/share/wordlists]
└─$ cd /home/kali
```

Step 4: Crack hashes with Hashcat.

a. To crack the hashes contained in the **my_pw_hashes.txt** file use the following command:

```
┌──(kali㉿Kali)-[~]
└─$ sudo hashcat -m 0 -a 0 -o cracked.txt my_pw_hashes.txt /usr/share/wordlists/rockyou.txt
```

This command outputs the cracked passwords in the new **cracked.txt** file.

b. To view the contents of the cracked.txt file and the plaintext password enter the command:

```
┌──(kali㉿Kali)-[~]
└─$ sudo cat cracked.txt
```

How many passwords were cracked?

Answer Area

<!----><!---->

Part 3: Crack Hashes with John the Ripper Using Dictionary and Brute Force Attacks

Step 1: View the John the Ripper help file.

In a terminal window, enter the command: **john -h** to view the John the Ripper help file.

```
┌──(kali㉿Kali)-[~]
└─$ john -h
```

Step 2: Crack hashes with John the Ripper.

Use the following command to crack the hashes in the **my_pw_hashes** file. This may take some time.

```
┌──(kali㉿Kali)-[~]
└─$ john --format=raw-md5 my_pw_hashes.txt
```

John shows the cracked passwords in orange. Which passwords were cracked with the password wordlist?

Answer Area

<!----><!---->

In this instance John uses a minimal password wordlist by default to quickly crack common passwords.

What does John the Ripper do if there are hashes it cannot crack with its wordlists?

Answer Area

<!----><!---->

If you let John continue to run long enough it will eventually crack the remaining passwords. (Note this may take 10-20 minutes) Press **Ctrl-C** or **q** to abort at any time after a few passwords have been cracked if desired.

Step 3: Use larger wordlists.

The default wordlist for John the Ripper is fairly small. John can use other wordlists, such as the rockyou.txt wordlist. It is also possible to download additional wordlists from the internet.

Use the following command to instruct John the Ripper to use the rockyou.txt wordlist.

```
┌──(kali㉿Kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 my_pw_hashes.txt
```

Step 4: Use brute force.

To instruct John the Ripper to use only brute force cracking use the following command:

```
┌──(kali㉿Kali)-[~]
└─$ john --incremental my_pw_hashes.txt
```

Note, using brute force can take a very long time to crack password hashes. A powerful GPU can take many hours to crack a complex 8-character password.

MD5 is considered too weak to use. However, notice how long it takes to crack even one MD5 hash using brute force. Abort the process with **Ctrl-C** or **q**.

Step 5: Show your cracked passwords.

In this example, if you interrupted the password cracking process using john and the raw-md5 format, you can still review the cracked passwords using the **--show** option.

```
┌──(kali㉿Kali)-[~]
└─$ john --show --format=raw-md5 my_pw_hashes.txt
```

Step 6: Experiment

If you have time, try some complex passwords of varying lengths of 4 to 8 characters. Attempt to brute force crack the passwords with John the Ripper in incremental mode. Create a file containing password hashes and then run the tool.

Part 4: Crack Hashes using RainbowCrack and Rainbow Tables

Note: RainbowCrack is not available in the VM using ARM CPUs (Apple M1/M2).

Step 1: Install RainbowCrack.

The RainbowCrack utility may need to be installed. Rainbow crack differs from hash cracking utilities that use brute force algorithms in that it uses rainbow tables to crack password hashes.

To install RainbowCrack enter the following command:

```
┌──(kali㉿Kali)-[~]
└─$ sudo apt install rainbowcrack
```

Step 2: Creating rainbow tables with rtgen.

Rainbow tables are ordinary files and can be created with RainbowCrack, or they can be downloaded from the internet. Creating a rainbow table can take a considerable amount of time and storage space as they are very large, ranging in size from 20GB to more than a terabyte.

a. Create a small simple rainbow table that will crack MD5 passwords of up to 3 characters with only lowercase letters.

   The **rtgen** program is used to generate rainbow tables based on user specified parameters.

   1. Enter the **rtgen -h** command and review the options.

      The example rainbow tables are given at the bottom of the output.

   2. Create a rainbow table by entering:

      ```
      ┌──(kali㉿Kali)-[~]
      └─$ sudo rtgen md5 loweralpha 1 3 0 1000 1000 0
      ```

This command creates a rainbow table that can crack passwords that are three characters long and only consist of lower-case letters. The application created a fille with 1000 entries. Creating more complex rainbow tables can take significant time and use significant resources.

b. Verify the rainbow table is created. Display the contents of the rainbowcrack directory by entering the command:

```
┌──(kali㉿Kali)-[~]
└─$ cd /usr/share/rainbowcrack
┌──(kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ ls
```

The newly created rainbow table should be in the directory as an **.rt** file.

Step 3: Sort the rainbow table.

a. Next, the rainbow table must be sorted. Entering the command**: sudo rtsort .** at the prompt. (**Note**: be sure to include the space and the period after **rtsort** as part of the command)

```
┌──(kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ sudo rtsort .
```

b. Generate a hash for a simple 3-character password which can then be cracked. Enter the command: **echo -n 'dog' | md5sum | awk '{print $1}'**.

```
┌──(kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ echo -n 'dog' | md5sum | awk '{print $1}'
06d80eb0c50b49a509b49f2424e8c805
```

c. Crack the hash with the rainbow table with RainbowCrack. At the prompt, enter the **rcrack . -h 06d80eb0c50b49a509b492424e8c805** command.

```
┌──(kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ rcrack . -h 06d80eb0c50b49a509b492424e8c805
```

Within milliseconds RainbowCrack should crack the hash and reveal the password **dog**.

d. You can also crack hashes contained in a .txt file as was done in Part 1 of the lab. To create a .txt file with some hashes, enter the following commands at the prompt:

```
echo -n 'fox' | md5sum | awk '{print $1}' > ~/my_rainbow_hashes.txt
echo -n 'boo' | md5sum | awk '{print $1}' >> ~/my_rainbow_hashes.txt
echo -n 'pop' | md5sum | awk '{print $1}' >> ~/my_rainbow_hashes.txt
```

To crack the hashes in the file, enter the **rcrack . -l ~/my_rainbow_hashes.txt** command at the prompt. The **-l** option tells rcrack to use a hash list file as input.

```
┌──(kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ rcrack . -l ~/my_rainbow_hashes.txt
```

Step 4: Explore resources to download rainbow tables.

In addition to generating Rainbow Tables with the **rtgen** command, there are many resources on the internet for downloading rainbow tables.

Open a web browser and search for **rainbow table download**.

Reflection Questions

1. Why is complexity and length so important with creating passwords?

Answer Area

<!----><!---->

2. In addition to complexity and length, what other measures can be taken to protect passwords?

Answer Area

<!----><!---->