Auditing Passwords with a Password Cracking Utility in Linux

Scenario

You are auditing password quality to better teach your fellow employees the importance of strong passwords. First, you send out an employee survey, asking seemingly harmless questions. The results are in the following table. Next, you will add the results to a wordlist to be used as a source for password cracking utilities such as John the Ripper. Finally, you will crack the passwords to demonstrate whether they expose the organization to authentication vulnerabilities.

Objectives

This activity is designed to test your understanding of and ability to apply content examples in the following CompTIA Security+ objectives:

- 1.2 Given a scenario, analyze potential indicators to determine the type of attack.

Lab

- Kali VM
- pfSense VM

Survey results

Here is an excerpt from the email message:

```
Please fill out the following survey so that we can get to know you
better.

•  What is your birthday?
•  What is your spouse’s name?
•  What is your anniversary?
•  What is your pet’s name?
```

You have documented the survey results in the table below:

Name	Birthday	Spouse	Anniversary	Pet name
user01	06101988	Mary	05232010	Max
user02	10141976	Tim	06011989	no pet
user03	09081998	Rick	07032018	Duke
user04	02081980	George	06142004	Rover
user05	03121985	Shawna	12132010	Spot
user06	no response	no response	no response	no response

Table 1.1 – Survey Results table.

Task 1

Create the necessary accounts and passwords

You will create six user accounts with passwords related to the above survey.

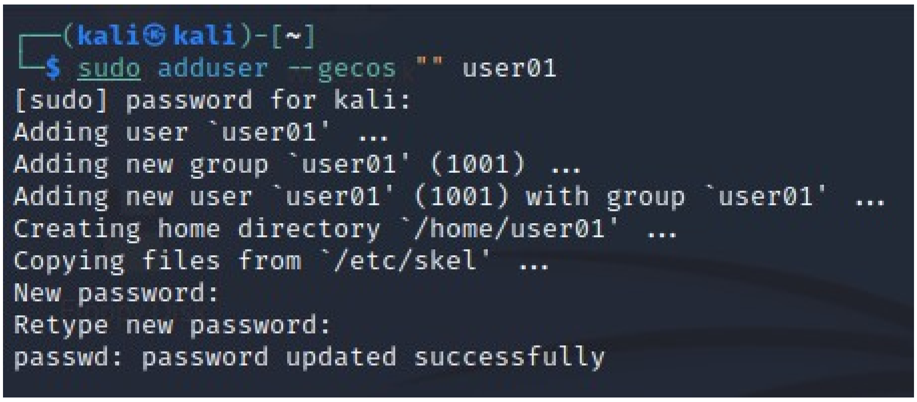
1. Sign in to the **Kali** VM as **kali** using **Pa$$w0rd** as the password.

2. Launch the **Terminal** application from the toolbar on the top of the Kali desktop.

3. Run the following command to create the first user:

```
sudo adduser --gecos “” user01
```

When prompted, set 06101988 as the password (you’ll type it twice).



```
┌──(kali㉿kali)-[~]
└─$ sudo adduser --gecos "" user01
[sudo] password for kali:
Adding user `user01' ...
Adding new group `user01' (1001) ...
Adding new user `user01' (1001) with group `user01' ...
Creating home directory `/home/user01' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
```

Figure 1.1 – Adding users with the adduser command.

NOTE: As a Debian-based Linux distribution, Kali Linux prefers the **adduser** command to create users. Red Hat-derived distributions, such as CentOS, typically use **useradd**.

4. Create the following additional accounts by using the **adduser** command, and set the specified passwords when prompted:

Username	**Passwords**
user02	Password
user03	Duke
user04	george
user05	$p0T
user06	G00dPa$$w0rd

TIP: Don’t forget about using Bash’s history feature. After creating **user01**, press the **UP ARROW** key one time, backspace over the **1** character and then enter the **2** character. This should allow you to create the accounts quickly.

TIP: Recall that Linux does not display any indication on the screen that the password is being entered. It is accepting your keystrokes, however.

Task 2

Add probable passwords to the word list file

John the Ripper uses word list files as the basis for its password cracking attempts. The employee survey results above include many probable passwords. You will extract the compresses wordlist file, and then add the probable passwords to the list.

1. Run the following command to extract the **/usr/share/wordlists/rockyou.txt.gz** word list file:

```
sudo gunzip usrshare/wordlists/rockyou.txt.gz
```

NOTE: This word list is used as the source for the password cracking attempt.

NOTE: The rockyou.txt wordlist contains entries with language that some may find offensive. If you are offended by bad language, please do not examine the contents.

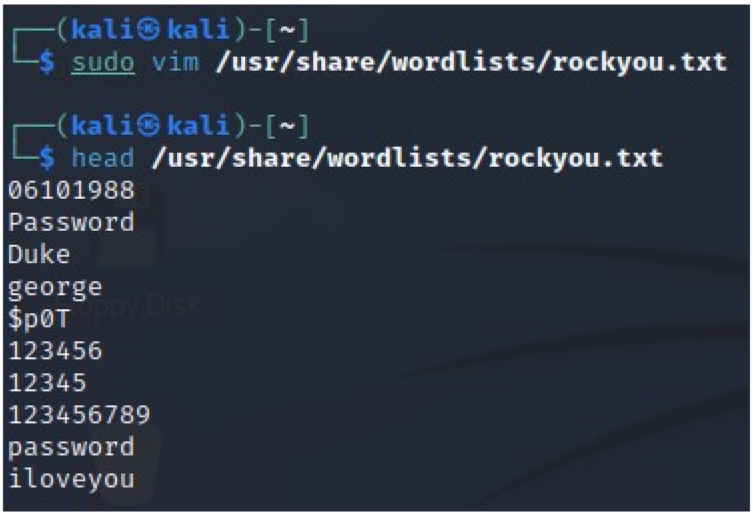
2. Enter the following command to open the **rockyou.txt** wordlist file for editing:

```
sudo vim /usr/share/wordlists/rockyou.txt
```

3. Select the **i** key to enter Vim's insert mode, and then add the following passwords, each on a separate line, at the top of the file:

```
06101988
Password
Duke
george
Sp0T
```

4. In Vim, select **Esc** key, and then enter **:wq** to save your changes and exit the file editor.



```
┌──(kali㉿kali)-[~]
└─$ sudo vim /usr/share/wordlists/rockyou.txt

┌──(kali㉿kali)-[~]
└─$ head /usr/share/wordlists/rockyou.txt
06101988
Password
Duke
george
$p0T
123456
12345
123456789
password
iloveyou
```

Figure 2.1 – Updated password list.

TIP: You would normally enter every survey response for each user (birthday, spouse name, anniversary, favorite color, favorite band, pet name). In order to better manage time, you will only enter the passwords John the Ripper needs to guess. Note that you cannot enter a possible password for user06, because that user declined to fill out the survey.

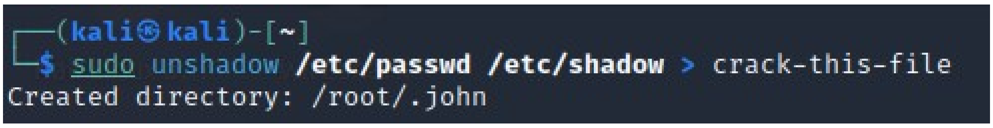
Task 3

Run John to crack passwords

You need to combine the /**etc/passwd** and /**etc/shadow** files, and then use John the Ripper to audit the employee passwords.

1. Run the following command to create a text file of usernames and password hashes:

```
sudo unshadow /etc/passwd /etc/shadow > crack-this-file
```

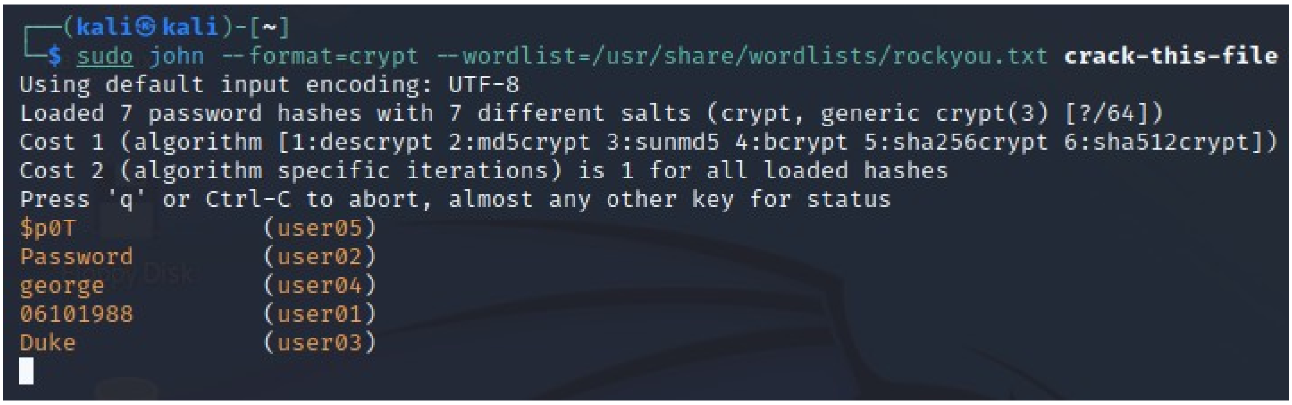


```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow /etc/passwd /etc/shadow > crack-this-file
Created directory: /root/.john
```

Figure 3.1 – Creating a file of username and password hashes.

2. Run the following command to crack passwords:

```
sudo john --format=crypt --wordlist=/usr/share/wordlists/rockyou.txt
crack-this-file
```



```
┌──(kali㉿kali)-[~]
└─$ sudo john --format=crypt --wordlist=/usr/share/wordlists/rockyou.txt crack-this-file
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt])
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
$p0T             (user05)
Password         (user02)
george           (user04)
06101988         (user01)
Duke             (user03)
```

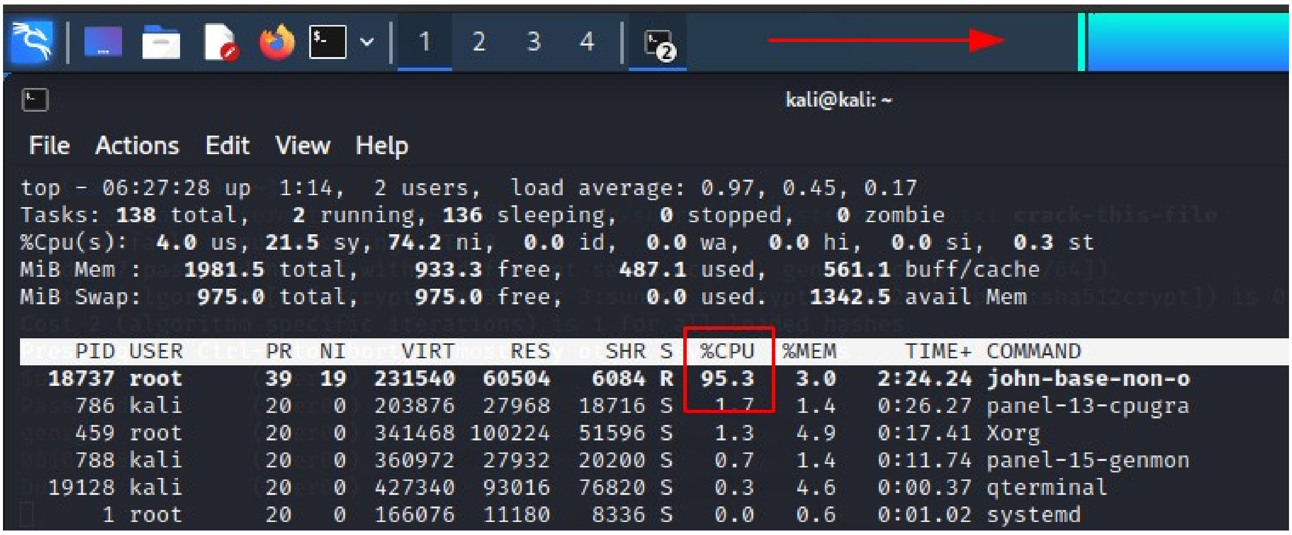
Figure 3.2 – John cracking passwords.

3. Open a second **Terminal**, and then run the following command to view the status of the audit:

```
sudo john --show crack-this-file
```

4. Type **top** to display system utilization information. Observe that John the Ripper is consuming most of the system’s processing power.

TIP: You can see the CPU consumption on the top bar graph in Kali Linux.



```
top - 06:27:28 up  1:14,  2 users,  load average: 0.97, 0.45, 0.17
Tasks: 138 total,   2 running, 136 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.0 us, 21.5 sy, 74.2 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.3 st
MiB Mem :  1981.5 total,   933.3 free,   487.1 used,   561.1 buff/cache
MiB Swap:   975.0 total,   975.0 free,     0.0 used.  1342.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  18737 root      39  19  231540  60504   6084 R  95.3   3.0   2:24.24 john-base-non-o
    786 kali      20   0  203876  27968  18716 S   1.7   1.4   0:26.27 panel-13-cpugra
    459 root      20   0  341468 100224  51596 S   1.3   4.9   0:17.41 Xorg
    788 kali      20   0  360972  27932  20200 S   0.7   1.4   0:11.74 panel-15-genmon
  19128 kali      20   0  427340  93016  76820 S   0.3   4.6   0:00.37 qterminal
      1 root      20   0  166076  11180   8336 S   0.0   0.6   0:01.02 systemd
```

Figure 3.3 – Checking system utilization with top.

5. Select **q** to exit **top**.

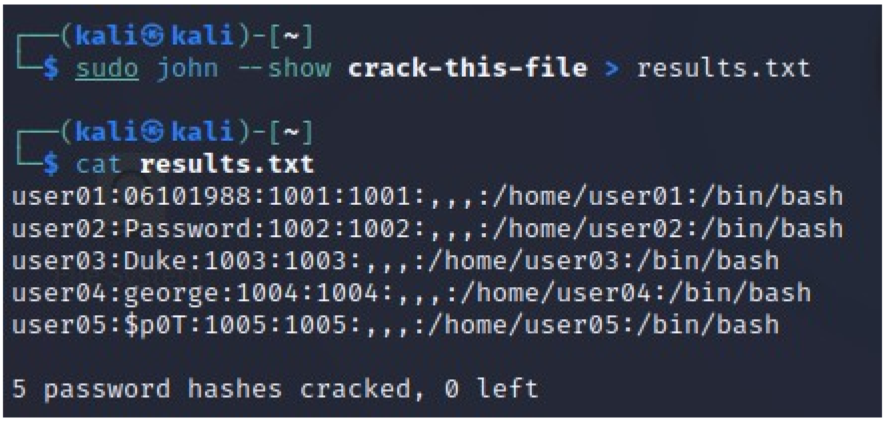
TIP: John the Ripper will eventually break the kali user password Pa$$w0rd, because it is in the word list too. It may take as much as 10 minutes.

6. Switch to the **Terminal** where John the Ripper is running, and then type **q** to interrupt the cracking attempt.

7. Redirect the results of the **john --show crack-this-file** to a text file:

```
sudo john --show crack-this-file > results.txt
```

8. Display the results.txt file contents by using the **cat** command.



```
┌──(kali㉿kali)-[~]
└─$ sudo john --show crack-this-file > results.txt

┌──(kali㉿kali)-[~]
└─$ cat results.txt
user01:06101988:1001:1001:,,,:/home/user01:/bin/bash
user02:Password:1002:1002:,,,:/home/user02:/bin/bash
user03:Duke:1003:1003:,,,:/home/user03:/bin/bash
user04:george:1004:1004:,,,:/home/user04:/bin/bash
user05:$p0T:1005:1005:,,,:/home/user05:/bin/bash

5 password hashes cracked, 0 left
```

Figure 3.4 – The results.txt file contents.