CYSE 270: Linux System for Cybersecurity

The goal of this lab is to test the strength of different passwords.

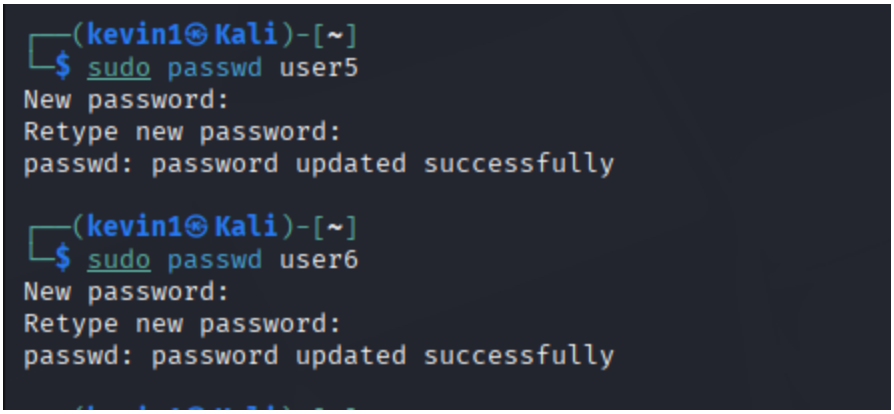
Task A – Password Cracking

1. Create **6 users** in your Linux Terminal, then set the password for each user that meets the following complexity requirement respectively. You should list the passwords created for each user. **[6 * 5 = 30 points]**.
    1. For user1, the password should be a simple dictionary word (all lowercase)
    2. For user2, the password should consist of 4 digits.
    3. For user3, the password should consist of a simple dictionary word of any length characters (all lowercase) + digits.
    4. For user4, the password should consist of a simple dictionary word characters (all lowercase) + digits + symbols.
    5. For user5, the password should consist of a simple dictionary word (all lowercase) + digits.
    6. For user6, the password should consist of a simple dictionary word (with a combination of lower and upper case) + digits + symbols.

1.

```
┌──(kevin1㉿Kali)-[~]
└─$ sudo useradd -m -d /home/user1 user1
[sudo] password for kevin1:

┌──(kevin1㉿Kali)-[~]
└─$ sudo useradd -m -d /home/user2 user2

┌──(kevin1㉿Kali)-[~]
└─$ sudo useradd -m -d /home/user3 user3

┌──(kevin1㉿Kali)-[~]
└─$ sudo useradd -m -d /home/user4 user4

┌──(kevin1㉿Kali)-[~]
└─$ sudo useradd -m -d /home/user5 user5

┌──(kevin1㉿Kali)-[~]
└─$ sudo useradd -m -d /home/user6 user6

┌──(kevin1㉿Kali)-[~]
└─$
```

```
┌──(kevin1㉿Kali)-[~]
└─$ sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

┌──(kevin1㉿Kali)-[~]
└─$ sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully

┌──(kevin1㉿Kali)-[~]
└─$ sudo passwd user3
New password:
Retype new password:
passwd: password updated successfully

┌──(kevin1㉿Kali)-[~]
└─$ sudo passwd user4
New password:
Retype new password:
passwd: password updated successfully
```



user1= world | 2. user2= 1234 | 3. user3= laundry79 | 4. user4= strong57$$ | 5. user5= weak88 | 6. user6= FAst45*$

Remember, do not use the passwords for your real-world accounts.

2. Export above users' hashes into a file named **xxx.hash** **(replace xxx with your MIDAS name)** and use John the Ripper tool to crack their passwords in wordlist mode (use rockyou.txt). **[40 points]**

```
┌──(kevin1㉿Kali)-[~]
└─$ gunzip rockyou.txt.gz

┌──(kevin1㉿Kali)-[~]
└─$ ls
copyright_cyse270  data  Desktop  Documents  Downloads  kevin1  kloar001.hash  Music  Pictures  Public  rockyou.txt  Templates  Videos
```

```
┌──(kevin1㉿Kali)-[~]
└─$ sudo john --format=crypt kloar001.hash --wordlist=rockyou.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with 6 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
1234             (user2)
world            (user1)
2g 0:00:07:52 0.21% (ETA: 2025-10-02 18:53) 0.004235g/s 77.46p/s 353.1c/s 353.1C/s titis..puppy3
2g 0:00:11:57 0.34% (ETA: 2025-10-02 15:54) 0.002788g/s 81.24p/s 353.4c/s 353.4C/s manger..krystin
2g 0:00:12:14 0.35% (ETA: 2025-10-02 15:50) 0.002724g/s 81.33p/s 353.4c/s 353.4C/s shawn23..ratrat
Use the "--show" option to display all of the cracked passwords reliably
Session aborted

┌──(kevin1㉿Kali)-[~]
└─$ sudo john -show kloar001.hash
user1:world:20361:0:99999:7:::
user2:1234:20361:0:99999:7:::

2 password hashes cracked, 0 left
```

After using the touch command to create the hash text file, I used the “sudo tail –6 /etc/shadow > kloar001.hash” command to copy the last 6 lines in that file to then put the info to the hash file. Afterwards, I copied the rockyou.txt.gz to my home directory and then used the “gunzip” command to unzip the file. Lastly the final command would use the john the ripper application to decrypt the information in the kloar001.hash file using the rockyou.txt file. Then to check it, I did the “john –show” to show two of the password hashes cracked.

3. Keep your john the ripper cracking for 10 minutes. How many passwords have been successfully cracked? **[30 points]**

```
┌──(kevin1㉿Kali)-[~]
└─$ sudo john -show kloar001.hash
user1:world:20361:0:99999:7:::
user2:1234:20361:0:99999:7:::

2 password hashes cracked, 0 left

┌──(kevin1㉿Kali)-[~]
└─$
```

Once 10 minutes had passed, it cracked two of them, which were easy passwords.

CYSE 270: Linux System for Cybersecurity

Extra credit (10 points):

1. Find and use the proper format in John the ripper to crack the following **MD5 hash**. Show your steps and results.
    a. 5f4dcc3b5aa765d61d8327deb882cf99
    b. 63a9f0ea7bb98050796b649e85481845

```
┌──(kevin1㉿Kali)-[~]
└─$ vim EC.txt

┌──(kevin1㉿Kali)-[~]
└─$ ls EC.txt
EC.txt

┌──(kevin1㉿Kali)-[~]
└─$ cat EC.txt
5f4dcc3b5aa765d61d8327deb882cf99
63a9f0ea7bb98050796b649e85481845

┌──(kevin1㉿Kali)-[~]
└─$ sudo john --format=Raw-MD5 EC.txt
[sudo] password for kevin1:
Sorry, try again.
[sudo] password for kevin1:
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=8
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password         (?)
Proceeding with incremental:ASCII
root             (?)
2g 0:00:00:01 DONE 3/3 (2025-09-30 05:27) 1.941g/s 5464Kp/s 5464Kc/s 5465KC/s rome..rams
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(kevin1㉿Kali)-[~]
└─$
```

After creating the file with vim and adding the encrypted hash in there, I ran a similar command while changing the format to “Raw-MD5” and then the text file alone to start it. It would proceed to give the words password for the 5f4… hash and root for the other one.

Reflection

In this lab, I used the program, John the ripper, to be able to crack different kinds of passwords. The lab taught an important lesson on implementing variation in password creation as there are applications (such as John) that could crack the password within time. I did have trouble at one point when completing the lab when I forgot to unzip the "rockyou.txt" file for the application (John) to use.