CS 3710 Introduction to Cybersecurity
Term: Fall 2022

Lab Exercise 1 – Introduction to Password Cracking

Due Date: September 2, 2022 11:59pm
Points Possible: 7 points

Name: Jingtao Scott Hong jh4ctf

By submitting this assignment you are digitally signing the honor code, "On my honor, I pledge that I have neither given nor received help on this assignment."

1. Overview

This lab exercise will provide some hands-on experience with password strength analysis using command-line tools in Linux.

2. Resources required

This exercise requires a Kali Linux VM running in the Virginia Cyber Range.

3. Initial Setup

From your Virginia Cyber Range course, select the **Cyber Basics** environment. Click "start" to start your environment and "join" to get to your Linux desktop login.

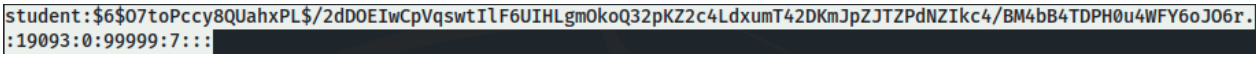
4. Tasks

Task 1: Introduction to password auditing.

On Linux systems, user accounts are stored in the **/etc/passwd** file (world-readable text file) and passwords are hashed and stored in **/etc/shadow** (a text file only readable by root). Click on the Terminal Emulator to open a command prompt. You will need to become an administrator on the system to see the shadow file. Type "**sudo su -**" and hit enter. You will noticed your command prompt changed from a **$** to a **#** and your user changed from student to root. Go ahead and "cat" those two password files to see what they look like.

*Question #1: What hash type is used by your Cyber Range version of Linux? How can you determine that by looking at the hashed passwords in /etc/shadow**?*** *(.5 point)*

It is SHA 512



```
student:$6$O7toPccy8QUahxPL$/2dDOEIwCpVqswtIlF6UIHLgmOkoQ32pKZ2c4LdxumT42DKmJpZJTZPdNZIkc4/BM4bB4TDPH0u4WFY6oJO6r.
:19093:0:99999:7:::
```

It is listed after the first dollar sign which is $6$ and it is the type SHA 512

*Question #2: What are two other hash IDs and their types that you may see in /etc/shadow**?*** *(.5 point)*

$5 SHA 256 and $1 MD5

*Question #3: What is password salting and why is it important**?*** *(.5 point)*

It is a way of protecting the password by further adding a type of random data such as string and further hashing them. It is important that it adds another layer of protection.

We'll use a password auditing tool called John the Ripper (JTR), a very effective and widely known password cracker. JTR is available from www.openwall.com/john. JTR is already installed in the virtual environment so you won't need to download it.

Task 2: **Crack Linux passwords.**

1. Create 2 new accounts, one with an easy to guess password (such as 1234) and one with a difficult to guess password.

Question #4: Cut and paste or screen capture the commands you used to create the accounts and set the passwords. *(.5 point)*

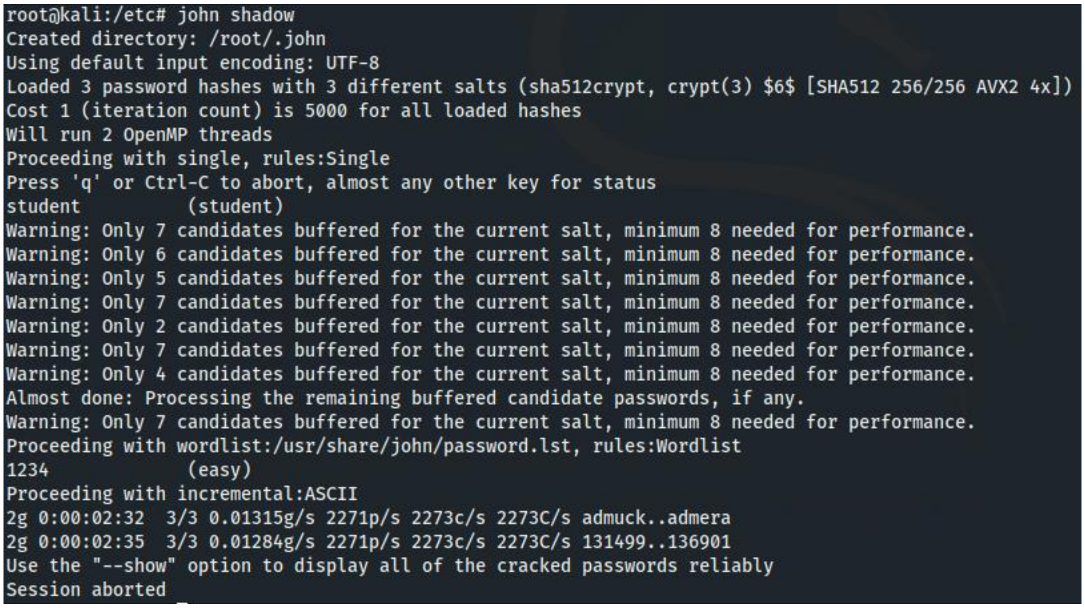
```
Sudo useradd easy
Sudo passwd easy
1234
1234
Sudp useradd hard
Sudp passwd hard
Scotthong0601
Scotthong0601
```

2. Now let's see which ones we can crack. Run john against the /etc/shadow file.

JTR will attempt to crack the passwords and display any that it 'cracks' as it goes along. It starts in "single crack" mode, mangling username and other account information. It then moves on to a dictionary attack using a default dictionary, then with a hybrid attack, then brute force where it will try every possibly combination of characters (letters, numbers, and special characters) until it cracks them all. You may see several warnings about candidates buffered for the current salt and that is ok. You can ignore those warnings.

The account with the easy to guess password should be cracked rather quickly. Wait for a little bit for it to crack the difficult password, but don't wait too long as it could take months or years to complete if your password is really strong! Press [CTRL]-[C] to stop execution if it doesn't automatically complete and return to the command prompt.

Question #5: Provide a screenshot of your JTR cracked passwords (.5 point)



```
root@kali:/etc# john shadow
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
student          (student)
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
1234             (easy)
Proceeding with incremental:ASCII
2g 0:00:02:32  3/3 0.01315g/s 2271p/s 2273c/s 2273C/s admuck..admera
2g 0:00:02:35  3/3 0.01284g/s 2271p/s 2273c/s 2273C/s 131499..136901
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

Question #6: Briefly describe how a dictionary based password attack works. *(.75 point)*
This method is provided with dictionary of pretrained words and phrases or other commonly occurred combination to better attack. Those words or phrases includes 1234, asdfgh, jackson and so on.

Question #7: Briefly describe how a brute force password attack works. *(.75 point)*
This method attacks by trying out different keys to keys value and try to attack simply using combination of all possible keys.

John uses the following files to manage execution. Most are all stored in the **`/usr/share/john`** folder on your Kali virtual machine (john.pot is stored elsewhere as indicated):

- **`password.lst`** is john's default dictionary. You can **cat** this file to look at it. You can specify another wordlist on the command line using the **`--wordlist=`** directive (for example **`# john --wordlist=/usr/share/dict/american-english /etc/shadow`**
- **`john.conf`** is read when JTR starts up and has rules for dictionary mangling for the hybrid crack attempt
- **`john.rec`** is used to record the status of the current password cracking attempt. If john crashes, it will start where it left off instead of starting again from the beginning of the dictionary.
- **/root/.john/`john.pot`** lists passwords that have already been cracked. If you run john again on the same shadow file, it won't show these cracked passwords unless you delete this file first using **rm /root/.john/john.pot.**

Task 3. More password audit.

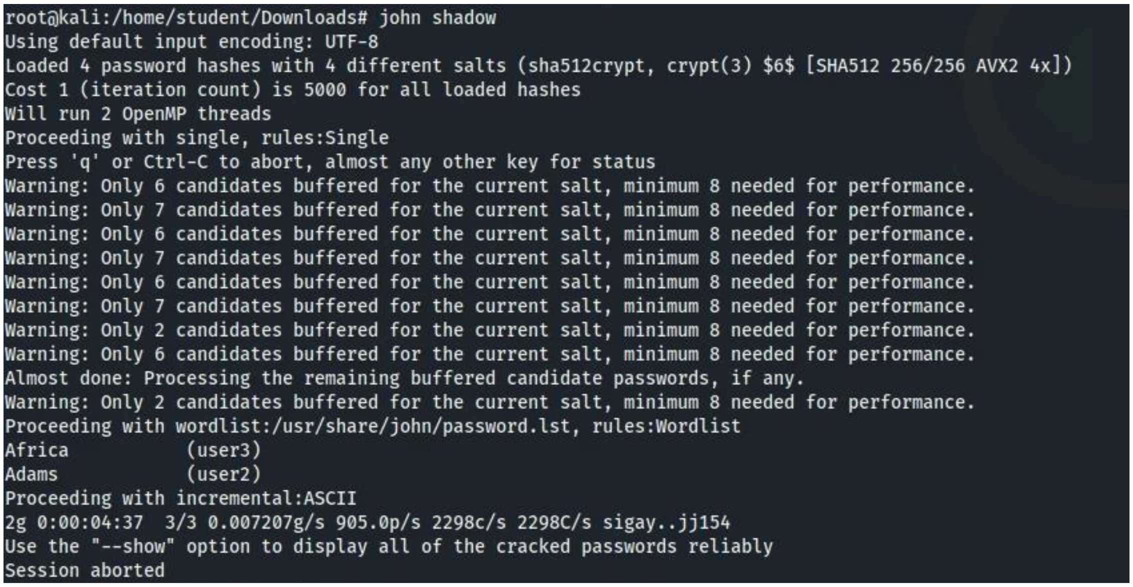
John the Ripper's default dictionary is a short list of common passwords. Sometimes a standard English dictionary is a better option. In this exercise we will 1) download a Linux shadow file that contains a set of user accounts and hashed passwords, 2) download a different dictionary, and then 3) attempt to determine the passwords using the default dictionary and the new dictionary.

1. Download the following file using the wget command:

   `artifacts.virginiacyberrange.net/gencyber/shadow`

2. Run John against the newly downloaded shadow file. Let John run for a few minutes, then stop with [CTRL]-[C].

Question #8: Which passwords are revealed? *(cut and paste or screen capture) (.5 point)*



```
root@kali:/home/student/Downloads# john shadow
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Africa           (user3)
Adams            (user2)
Proceeding with incremental:ASCII
2g 0:00:04:37  3/3 0.007207g/s 905.0p/s 2298c/s 2298C/s sigay..jj154
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

User3 Africa
User2 Adams

3. Install a new dictionary using the following command:

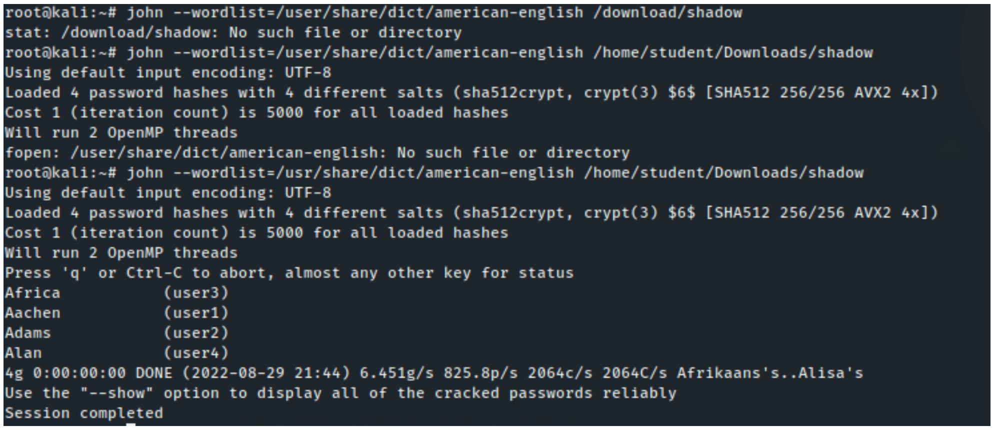
   ```
   # apt-get update
   # apt-get install wamerican
   ```

4. Clear the John cache from the previous run by deleting the **/root/.john/john.pot** file.

5. Next run John against the downloaded shadow file again but this time using the newly downloaded dictionary by invoking the --wordlist directive at the command line with the location of the new dictionary (**`--wordlist=/usr/share/dict/american-english`**)

   Note: If you get an error about a locked /root/.john/john.rec file, you can delete that file.

Question #9: Which passwords were revealed this time? *(cut and paste or screen capture) (.5 point)*



```
root@kali:~# john --wordlist=/user/share/dict/american-english /download/shadow
stat: /download/shadow: No such file or directory
root@kali:~# john --wordlist=/user/share/dict/american-english /home/student/Downloads/shadow
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
fopen: /user/share/dict/american-english: No such file or directory
root@kali:~# john --wordlist=/usr/share/dict/american-english /home/student/Downloads/shadow
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Africa           (user3)
Aachen           (user1)
Adams            (user2)
Alan             (user4)
4g 0:00:00:00 DONE (2022-08-29 21:44) 6.451g/s 825.8p/s 2064c/s 2064C/s Afrikaans's..Alisa's
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

User3 Africa
User1 Aachen
User2 Adams
User4 Alan

Question #10: What is the difference between the two dictionaries that made one attempt more effective than the other? You may want to take a look at each of the dictionaries or metadata about the dictionaries. (1 point)

The dictionary we have downloaded utilized all American English words and prioritize using these combinations. The original dictionary that John the Ripper uses contains common password combinations.

Question #11: What are two methods that will help provide more secure authentication and protect against password cracking? (1 point)

1. We can use uncommonly used complicated long password combinations
2. We prevent the uses of common language such as American English words to be the part of the password.

To close the exercise, just click the X on the terminal window to close it and click on the Log Out icon in the upper right hand corner of the screen to log out.

By submitting this assignment you are digitally signing the honor code, "I pledge that I have neither given nor received help on this assignment".

END OF EXERCISE

References

- John the Ripper (JTR): www.openwall.com/john