CS 3710 Introduction to Cybersecurity
Term: Spring 2023

Lab Exercise 1 – Introduction to Password Cracking

Due Date: January 27, 2023 11:59pm
Points Possible: 7 points

Name: Sidhardh Burre

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)

We know that at the beginning of hashed passwords stored in the /etc/shadow file we can find a dollar sign followed by a number. This corresponds to the type of hashing used to hash the password. In our case we have:

student:$6$DqVA5JXkXEqT.Z1B$uzi1tTzdhd37kncHRSKkboGf.dimrPDZYiwmCIEGXDQtZJPQ6p1el286cwb2.EJf9Rc9v0mnH4zIvxmqJdqnw/:19374:0:99999:7:::

And using the "$6" we can find that we are using the SHA-512 hash.

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

Within the /etc/shadow file, no other users (except for student) are enabled so in place of their passwords, we have “!”s and “*”s.

But other hash IDs that we could see include:

- **1: MD5**
- **$2a$: Blowfish**
- **$2y$: EKSBlowfish**
- **5: SHA-256**

(from: https://cybersophia.net/linux/etc-shadow-file-in-linux/)

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

Password salting is taking the original password and appending a random string to it prior to the hash. This way even if a hacker attempts a bruteforce attack with pre-computed hashed dictionary words, the salt would make it difficult to find a match.

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)*

I used “useradd” twice to create two new users (user1, user2). Switched to student (from root) via “su student”. And then performed “sudo passwd user1” to set user1’s password to “1234” and likewise for user2 to set user2’s password to “w}EK!uC3m~T,dz#:”.

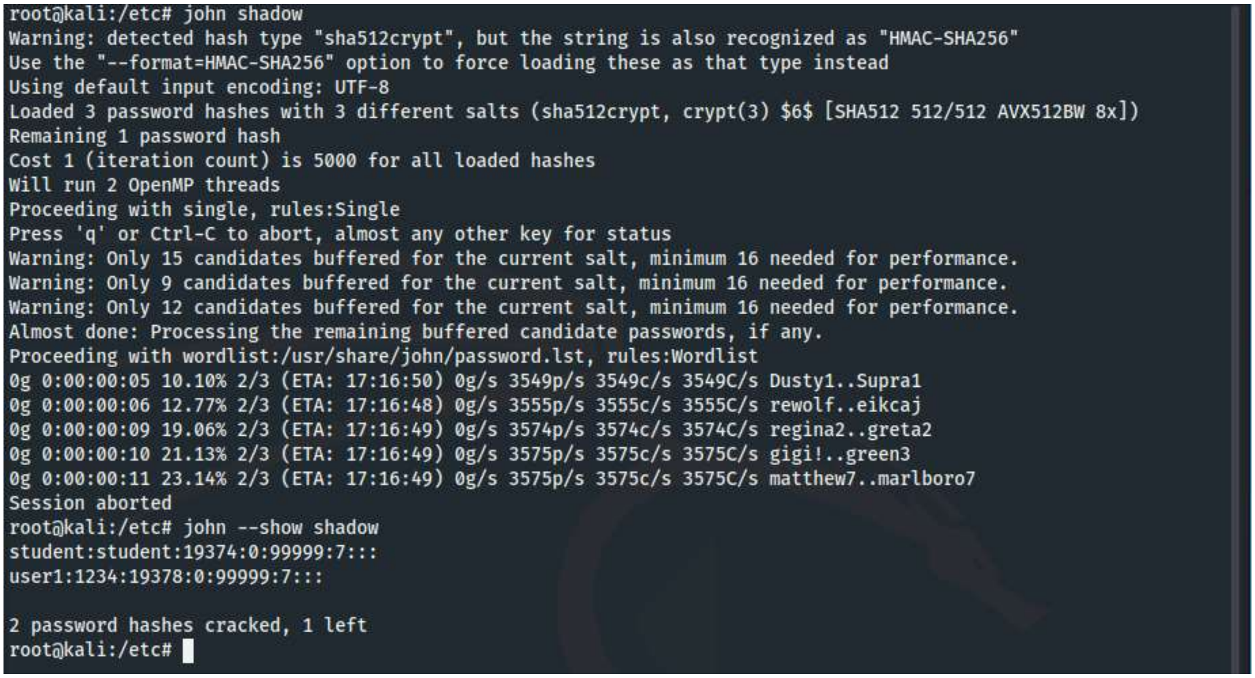
2. Now let’s see which ones we can crack. Run john against the /etc/shadow file.

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)*

Although we were able to crack student's and user1's passwords, user2's password was not able to be cracked.



```
root@kali:/etc# john shadow
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 512/512 AVX512BW 8x])
Remaining 1 password hash
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 15 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 9 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 12 candidates buffered for the current salt, minimum 16 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
0g 0:00:00:05 10.10% 2/3 (ETA: 17:16:50) 0g/s 3549p/s 3549c/s 3549C/s Dusty1..Supra1
0g 0:00:00:06 12.77% 2/3 (ETA: 17:16:48) 0g/s 3555p/s 3555c/s 3555C/s rewolf..eikcaj
0g 0:00:00:09 19.06% 2/3 (ETA: 17:16:49) 0g/s 3574p/s 3574c/s 3574C/s regina2..greta2
0g 0:00:00:10 21.13% 2/3 (ETA: 17:16:49) 0g/s 3575p/s 3575c/s 3575C/s gigi!..green3
0g 0:00:00:11 23.14% 2/3 (ETA: 17:16:49) 0g/s 3575p/s 3575c/s 3575C/s matthew7..marlboro7
Session aborted
root@kali:/etc# john --show shadow
student:student:19374:0:99999:7:::
user1:1234:19378:0:99999:7:::

2 password hashes cracked, 1 left
root@kali:/etc# █
```

Question #6: Briefly describe how a dictionary based password attack works. *(.75 point)*

For a dictionary based attack, hackers iterate through words in a dictionary and then guess those iterations as passwords to an account. This can be advantageous if the user is more likely to have a select word or phrase as part of their password.

Question #7: Briefly describe how a brute force password attack works. *(.75 point)*

Hackers iterate over all characters in the keyspace and use each iteration as a guess. This approach is guaranteed to find the password but takes a exponential time and can be thwarted by rate limiting.

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.

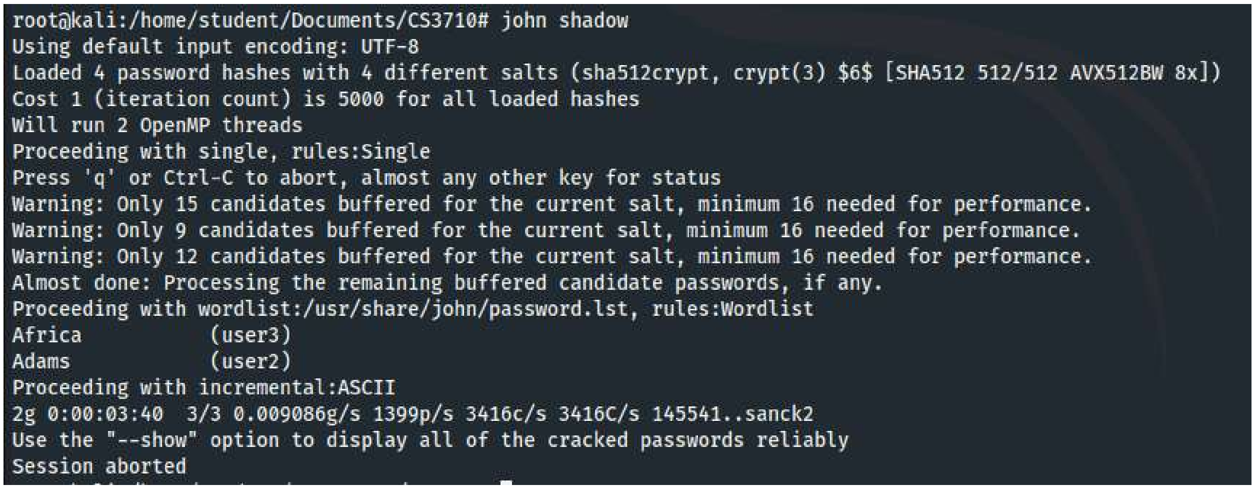
John the Ripper's default dictionary is a short list of common passwords. Sometimes a standard English dictionary is a better option. In this exercise we will 1) download a new Linux shadow file that contains a set of user accounts and hashed passwords, 2) download a different dictionary, and then 3)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

1. Download the following new shadow file using the wget command:

   `artifacts.virginiacyberrange.net/gencyber/shadow`

2. Run John against the newly downloaded shadow file. It should be in your current working directory, it is not /etc/shadow. 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/Documents/CS3710# john shadow
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 512/512 AVX512BW 8x])
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 15 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 9 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 12 candidates buffered for the current salt, minimum 16 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Africa           (user3)
Adams            (user2)
Proceeding with incremental:ASCII
2g 0:00:03:40  3/3 0.009086g/s 1399p/s 3416c/s 3416C/s 145541..sanck2
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

3. Download a new American English dictionary by opening the web browser in Kali Linux and going to this Google drive link:

   https://drive.google.com/file/d/1uvCYVKxh7ErGGXqhnPi7HNYcgku_4ScP/view?usp=sharing

   This new dictionary file will be located in /home/student/Downloads.

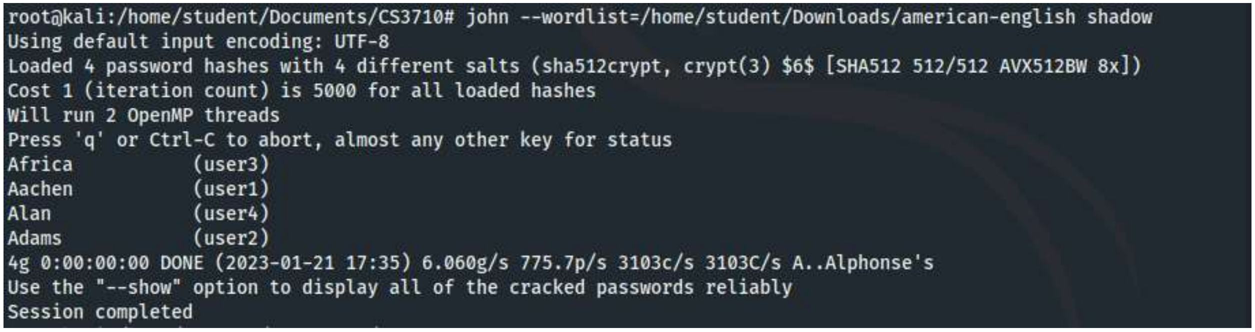
4. Clear the John cache from the previous run by deleting the **/root/.john/john.pot** file.

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

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

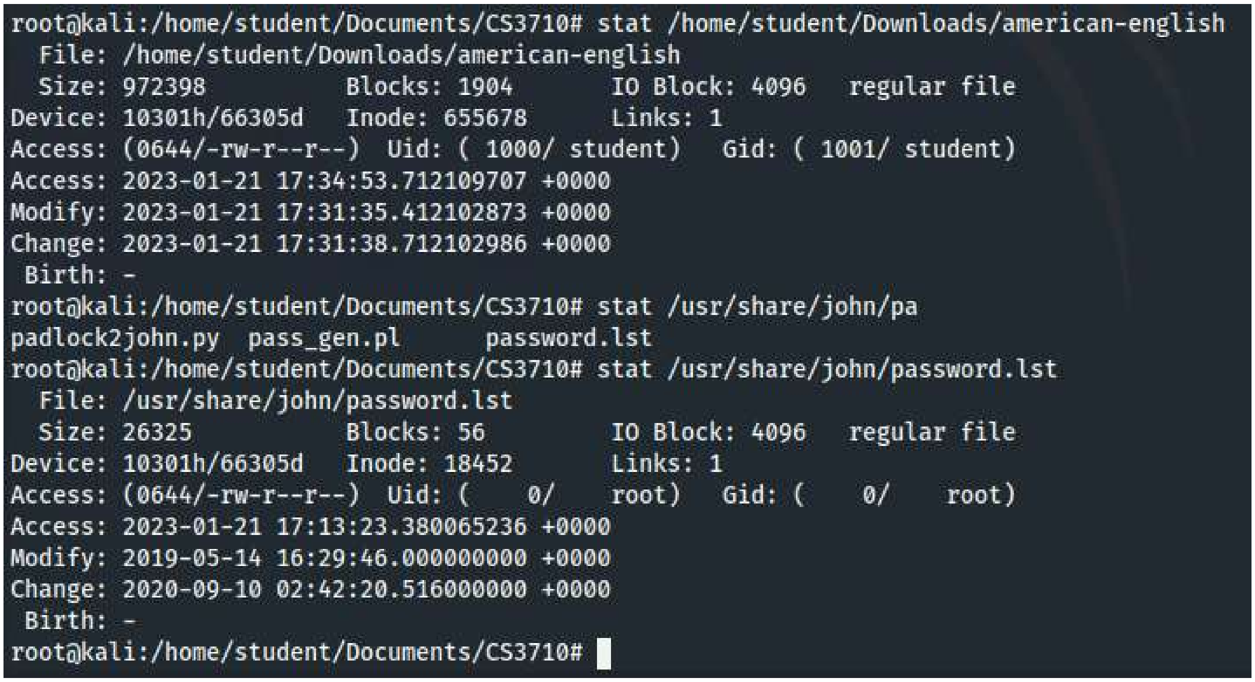
Question #9: Which passwords were revealed this time? (cut and paste or screen capture) (.5 point)

Now we got all four passwords!



```
root@kali:/home/student/Documents/CS3710# john --wordlist=/home/student/Downloads/american-english shadow
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 512/512 AVX512BW 8x])
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)
Alan             (user4)
Adams            (user2)
4g 0:00:00:00 DONE (2023-01-21 17:35) 6.060g/s 775.7p/s 3103c/s 3103C/s A..Alphonse's
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

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 to compare them.) (1 point)



```
root@kali:/home/student/Documents/CS3710# stat /home/student/Downloads/american-english
  File: /home/student/Downloads/american-english
  Size: 972398          Blocks: 1904       IO Block: 4096   regular file
Device: 10301h/66305d   Inode: 655678      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ student)   Gid: ( 1001/ student)
Access: 2023-01-21 17:34:53.712109707 +0000
Modify: 2023-01-21 17:31:35.412102873 +0000
Change: 2023-01-21 17:31:38.712102986 +0000
 Birth: -
root@kali:/home/student/Documents/CS3710# stat /usr/share/john/pa
padlock2john.py  pass_gen.pl      password.lst
root@kali:/home/student/Documents/CS3710# stat /usr/share/john/password.lst
  File: /usr/share/john/password.lst
  Size: 26325           Blocks: 56         IO Block: 4096   regular file
Device: 10301h/66305d   Inode: 18452       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2023-01-21 17:13:23.380065236 +0000
Modify: 2019-05-14 16:29:46.000000000 +0000
Change: 2020-09-10 02:42:20.516000000 +0000
 Birth: -
root@kali:/home/student/Documents/CS3710#
```

The american-english dictionary is a lot larger (~40x larger) so it is more likely to contain the words that are used for the passwords by the users.

Question #11: What are two methods that will help provide more secure authentication and protect against password cracking? (Something different than just making your password longer or more complex.) (1 point)

Two methods to provide more secure authentication and protection are:

1. **Use uncommon words or avoid using words at all and definitely don't use key phrases such as your college, street name, etc.**
2. **Avoid reusing passwords whenever possible, ideally have a unique password for every service.**

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