

Lab Exercise 1 – Introduction to Password Cracking

Due Date: January 28, 2022 11:59pm
Points Possible: 7 points

Name: Joseph Bannon (jb9war)

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 notice 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)

The hash type for student is `$6` which translates to SHA-512. This can be found by looking in `/etc/shadow`, finding the student user, then looking at the 2nd : delimited entry and the first `$` delimited entry.

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

Two other hash ids that can be found are `$1` which is MD5 and `$2a` which is Blowfish.

Source: <https://www.cyberciti.biz/faq/understanding-etcshadow-file/>

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

Password salting is adding random characters to a password then hashing it. This prevents an attacker from being able to recognize a common passwords hash and makes it harder to guess the password from the hash. For example, the password “Apple” will have the same hash for every user, but adding salt changes the hash for every user.

Source: <https://cyberhoot.com/cybrary/password-salting/#:~:text=Password%20Salting%20is%20a%20technique,password%20and%20then%20hashing%20it>

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)

```
root@kali:/# sudo useradd user1_easy
root@kali:/# sudo passwd user1_easy
New password:
Retype new password:
passwd: password updated successfully
root@kali:/# sudo useradd user2_hard
root@kali:/# sudo passwd user2_hard
New password:
Retype new password:
passwd: password updated successfully
root@kali:/#
```

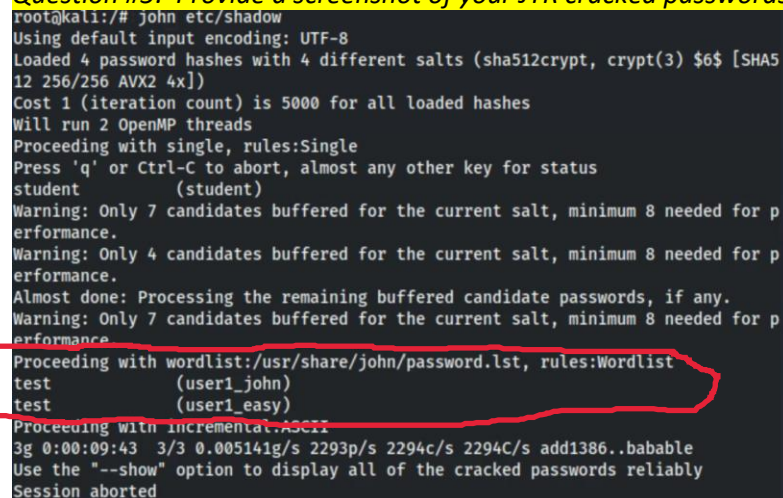
Password for user1_easy is test
Password for user2_hard is ls63vs

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:~# john etc/shadow
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA5
12 256/256 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 p
erformance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for p
erformance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for p
erformance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
test (user1_john)
test (user1_easy)
Proceeding with incremental.ASCI11
3g 0:00:09:43 3/3 0.005141g/s 2293p/s 2294c/s 2294C/s add1386..babable
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

User1_easy took about 5 seconds, user2_hard did not finish in 2 minutes.

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

Dictionary attack is a type of offline attack that enters every single word in the dictionary as password. It can also be a collection of phrases or words with letter number replacement. Most passwords are derivatives of words so dictionary attack takes advantage by limiting its attempts to just words.

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

A brute force attack is a type of offline attack that tries every single character combination until it is successful. A brute force attack takes a lot of time especially for complicated passwords but will eventually be able to crack every password.

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.

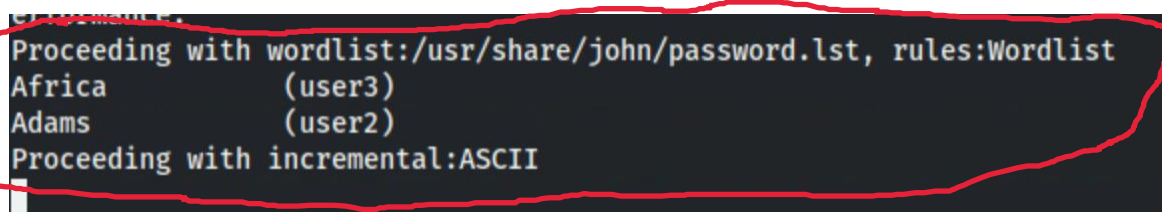
John the Ripper's default dictionary is a short list of common passwords. Sometimes a standard English dictionary is a better option. In this exercise we will 1) download a Linux shadow file that contains a set of user accounts and hashed passwords, 2) download a different dictionary, and then 3) attempt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



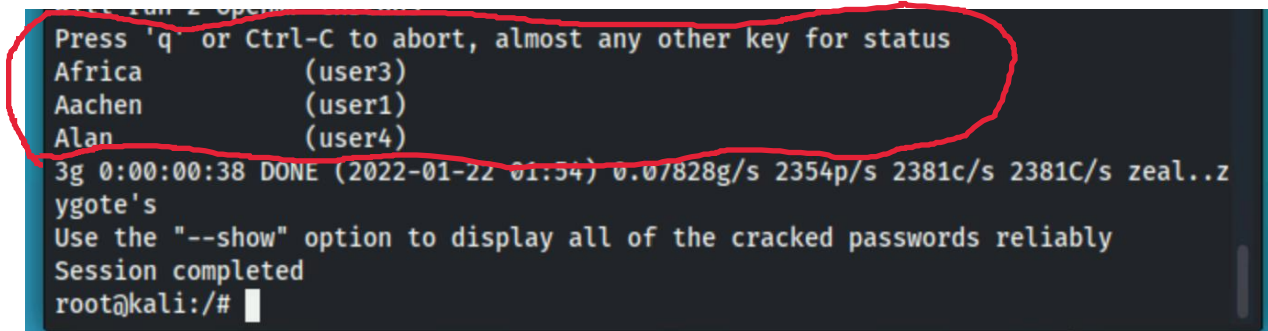
```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Africa      (user3)
Adams       (user2)
Proceeding with incremental:ASCII
```

3. Install a new dictionary using the following command:

```
# 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**)

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



```
Press 'q' or Ctrl-C to abort, almost any other key for status
Africa      (user3)
Aachen      (user1)
Alan        (user4)
3g 0:00:00:38 DONE (2022-01-22 01:54) 0.07828g/s 2354p/s 2381c/s 2381C/s zeal..z
ygote's
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:/#
```

Question #10: What is the difference between the two dictionaries that made one attempt more effective than the other? (1 point)

The second attempt had more words that the dictionary attempted before moving to a brute force approach. Therefore it was able to get the passwords within the dictionary much quicker than it would with a brute force method.

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

One method is to have longer passwords. These passwords take exponentially longer to crack on a brute force method. Another method is to not reuse passwords. If an attacker is able to crack one of the passwords, then they know can run that password on all your accounts, which takes much less time than a brute force attack.

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

