**Lab Exercise 1 – Introduction to Password Cracking**

Due Date: January 24, 2025 11:59pm

Points Possible: 7 points

Name: Abhinav Pappu

*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.”*

**Generative AI assistance is NOT permitted 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 My Environment” to start your environment and once it is ready click “Join My Environment” to open your Linux desktop.

**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 store’s password for the student credential starts with $y$, which is the prefix that is used for the script. I determine this by identifying the prefix and searching for the hash type associated with the $y$ prefix.

*Question #2: What are two other hash IDs and their types that you may see in an /etc/shadow file? (The ID is the numbers/letters that identify the hash and the type is the name of the hash) (.5 point)*

Sha-512 (identified by prefix $6$) and MD5 (identified by prefix $1$) are 2 other hash types that are you may see in a Linux/etc/shadow/ file.

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

Password Salting is when a random string is concatenated to the password before applying the hashing algorithm. Salting makes the hashed password more secure and difficult to decipher.

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](http://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.

Difficult Password: ILoveThomasJefferson123

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

A computer screen shot of a computer code

AI-generated content may be incorrect. A computer screen with white text

AI-generated content may be incorrect.

2. Now let’s see which ones we can crack. Run john against the /etc/shadow file. You will need to use the **-format:crypt** command line option to crack this particular hash method.

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)*

***A computer screen shot of a program

AI-generated content may be incorrect.***

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

A dictionary based attack uses a predetermine “dictionary” of words and phrases. For example, this dictionary could be ‘1234’, ‘password’ ‘qwerty’ or other fairly common and predictable passwords.

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

Repetitively trying multiple permutations of passwords in the hope that one of them will eventually be correct.

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 cracking.**

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**

1. Take a look at the newly downloaded shadow file. It should be in your current working directory, it is not /etc/shadow. Notice that it uses a different hash ID. You won’t need to use the -format:crypt command line option, let’s see if John can automatically figure out the hash. 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)*

Africa

A computer screen shot of a computer program

AI-generated content may be incorrect.

1. Next we will run John again with a different dictionary. First, we will download an American English dictionary on to our Kali Linux system. (If the dictionary is already there, it will be updated.)

**# sudo apt update; sudo apt install -y wamerican**

1. Clear the John cache from the previous run by deleting the **/root/.john/john.pot** file.
2. 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=/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)*

*A computer screen shot of a computer code

AI-generated content may be incorrect.*

Aachen, Adams, Africa, Alan

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

The American dictionary contains all the words in the American English dictionary which would obviously make it easier for guessing passwords that are word that are contained in the dictionary like the passwords above. On the other hand, the password.lst in the john file contains some of the most common passwords which includes words, phrases, and other combinations such as :”hal9000”, “1960”, “22”, “1qw23e” which are typically not found in the dictionary.

*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)*

1. You can use something similar to DuoPush, an RSASecurID, or a similar authenticator app that requires you to possess omething on top of knowing your password in order to login.
2. Security Question. A security Question that only you would know the answer would make it harder for brute force attacks to get through.

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](http://www.openwall.com/john)