CS 3710 Introduction to Cybersecurity
Term: Spring 2022

Lab Exercise 1 – Introduction to Password Cracking

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

Name: Hanzhang Zhao

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

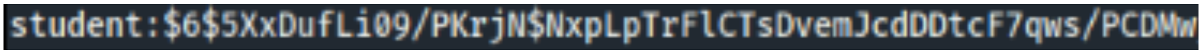
From your Virginia Cyber Range course, select the **Cyber Basics** environment. Click "start" to start your environment and "join" to get to your Linux desktop login.

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)*
SHA512,



```
student:$6$5XxDufLi09/PKrjN$NxpLpTrFlCTsDvemJcdDDtcF7qws/PCDMw
```

You can see that my password's hash ID is 6, which is typically encrypted with SHA512.
*Question #2: What are two other hash IDs and their types that you may see in /etc/shadow**?** (.5 point)*
5: SHA256
1: MD5

*Question #3: What is password salting and why is it important**?** (.5 point)*
Salting makes the original hashed password unrecognizable and unique. It is important because it can greatly enhance the security of passwords.

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.

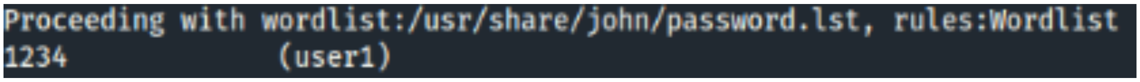
Question #4: Cut and paste or screen capture the commands you used to create the accounts and set the passwords. *(.5 point)*
sudo adduser user1
sudo passwd user1
sudo adduser user2
sudo passwd user2

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)



```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
1234             (user1)
```

Question #6: Briefly describe how a dictionary based password attack works. (.75 point)
A dictionary attack is attempt to guess passwords by using well-known words or phrases.

Question #7: Briefly describe how a brute force password attack works. (.75 point)
A brute force attack relies on guessing all possible combinations of a targeted password until the correct password is discovered.

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)*

```
Africa       (user3)
Adams        (user2)
```

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)*

```
Africa       (user3)
Aachen       (user1)
Adams        (user2)
Alan         (user4)
```

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

Words contained in the two dictionaries are different. Words contained in American-english dictionary are closer to the passwords for users in shadow, so it is more effective than the default dictionary.

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

1. Use longer passwords
2. Use different passwords for different accounts

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