**Lab #4**

**CECS 378 – Spring 2021 Cappel**

**Due:** Monday, March 8th prior to lab (11:59 PM)

**Lab #4 is focused on Linux/Windows password cracking and password policies.**

**We will be using the following VM’s for this lab: KaliLinux (Task 1) & Win10 (Tasks 2 & 3). There are five screenshots & five questions all worth 10 points each. Also 4 optional extra credit.**

**Task 1 – Cracking Linux Passwords**

The following task will demonstrate how to obtain passwords on a Linux device once you have access to an account with root privileges. If the user passwords on the system can be obtained and cracked, an attacker can use them to pivot to other machines if the login credentials are the same across systems. There are two tried-and-true password cracking tools that can accomplish the task of cracking a Linux password: **John the Ripper and Hashcat**. We will demonstrate them both in this task.

Perform the following steps on the Kali Linux virtual machine within Oracle VirtualBox:

1. Turn on your KaliLinux virtual machine in Oracle VirtualBox.

**Note**: You will only require the KaliLinux virtual machine in this particular task.

1. Open the Terminator (command line) program.

**Note**: You may need to append sudo to the commands if you are not logged in as root (2020 images).

1. Create three new Linux accounts using the following commands:

*useradd -m user1 -G sudo -s /bin/bash*

*useradd -m user2 -G sudo -s /bin/bash*

*useradd -m user3 -G sudo -s /bin/bash*

1. Set the passwords (shown below) on the three new accounts using the following commands:

*passwd user1* **Note**: We will set the password = ***admin*** on the user1 account.

*passwd user2* **Note**: We will set the password = ***12345678*** on the user2 account.

*passwd user3* **Note**: We will set the password = ***password1*** on the user3 account.

A couple files of interest on Linux systems are the **/etc/passwd** and **/etc/shadow** files. The /etc/passwd file contains basic information about each user account on the system, including the root user which has full administrative rights, system service accounts, and actual users. The /etc/shadow file contains the encrypted (hashed) passwords of users on the system. While the /etc/passwd file is typically world-readable, the /etc/shadow file is only readable by the root account. The shadow file also contains other information such as password expiration dates.

The first thing we need to do is copy the contents of the passwd and shadow files into their own text files on our local machine; let's call them **passwd.txt** and **shadow.txt**, respectfully.

1. Copy the two files from their source locations into two text files using the following commands:

*cp /etc/passwd /etc/passwd.txt*

*cp /etc/shadow /etc/shadow.txt*

1. View the password text file using the following command:

*cat /etc/passwd.txt*

There are seven fields in each line of the /etc/passwd file. A typical line looks something like this:

msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash

The **first field** is the user's login name. The **second field** traditionally contained an encrypted password, but nowadays (unless you get extremely lucky) it merely contains the letter "x," to denote that a password has been assigned. If this field is blank, the user does not need to supply a password to log in. The **third field** is the user ID, a unique number assigned to the user, followed by the group ID in the **fourth field**. The **fifth field** is typically the full name of the user, although this can also be left blank. The **sixth field** is the user's home directory, and finally, the **seventh field** is the default shell, usually set to /bin/bash.

1. View the shadow file using the following command:

*cat /etc/shadow.txt*

There are eight fields in the /etc/shadow file and a typical line will look like this:

msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::

The **first field** is the username. The **second field** is the password hashing details plus the hashed password. $1 denotes MD5 with 22 characters, $5 denotes SHA-256 with 43 characters, and $6 denotes SHA-512 with 86 characters. $XN10Zj2c$ = Salt and separators. The salt is a small string of characters to mix into the hashing function. Its goal is making it more difficult to perform certain password based attacks on the hashed password. *Long string of characters* = hashed password. When the password field has a ! or \*, then the account is locked. A double ! (!!) indicates a password has never been set. The **third field** is the last changed field. This number indicates when the password was last changed. The number does indicate the day number, starting from epoch date (1 January 1970). The **fourth field** is the number of days before the password can be changed. Zero means it can be changed now. The **fifth field** is the number of days until the password will be required to be changed. Another pretty self-explanatory field, stating how long is left (in days), before a password change is required. A great option to force password changes. The **sixth field** is the warning threshold in days. In line with previous field it describes the number of days until a warning will be giving. In our example it is a week. The **seventh field** is the expire date. Also stored in days, describing when the account was expired (from epoch date). The **eighth field** is the reserved field. Usually not used by Linux distributions.

Since we have root-level access on our device, we can use these files to uncover passwords of other users in the hopes of pivoting to other systems and being able to perform furthering exploitation.

**Cracking the passwords with John the Ripper.** John the Ripper is a popular password cracking tool that supports many common hash types as well as a useful autodetect feature. It has been around for a while now, and as such, it continues to be one of the strongest and easiest to use crackers available. Before we can feed the hashes we obtained into John, we need to use a utility called **unshadow** to combine the passwd and shadow files into a format that John can read.

1. Run the following command to merge the data into a new text file called filetocrack.txt:

*unshadow /etc/passwd.txt /etc/shadow.txt > /etc/filetocrack.txt*

John can run on its own by just typing **john** plus whatever file you are using for input, but it's often much more useful to supply a wordlist. There are some wordlists available for use under the **/usr/share/wordlists** directory, but for now, we'll use the wordlist file **password.lst** that can be found in the /usr/share/john directory since it is quite a nice list. We will use the **--wordlist** flag to specify the list to use and pass in our input file.

1. Run the following command to crack the Linux passwords:

*john --format=sha512crypt --wordlist=/usr/share/john/password.lst /etc/filetocrack.txt*

1. Run the following command to show the passwords that were successfully cracked:

*john --show /etc/filetocrack.txt*

After the username in the **first field**, we can now see the cleartext password in the **second field**. It tells us that a certain number of password hashes were cracked; Depending on the hardware being used, the wordlist that's supplied, and the length and complexity of the passwords, various levels of success will be achieved using John the Ripper.

**Screen Shot 1)** Take a screen shot of your John the Ripper results showing the passwords that were cracked.

**Cracking the passwords with Hashcat.** The next tool that we will look at is Hashcat. Before leveraging Hashcat, we will install the gedit text editor on our Kali Linux virtual machine to more easily edit files.

1. Run the following command to install the gedit tool on the Kali Linux virtual machine:

*apt-get install gedit*

1. Run the following command to copy the filetocrack.txt file to a new hashes.txt file:

*cp /etc/filetocrack.txt /etc/hashes.txt*

We will now edit the hashes.txt file using the gedit command so that the hashes file only includes the password hash information for the three user accounts created earlier.

1. Run the following command to edit the hashes.txt file:

gedit /etc/hashes.txt

**Note**: You will need to modify each line for the 3 user accounts and only keep the hash portion.

You will need to keep only the characters *between* the first “:” and the second “:” in each line.

**Note**: Here is a link to the standard hash types (the KaliLinux password hashes are **hash-mode 1800**).

<https://hashcat.net/wiki/doku.php?id=example_hashes>

**Note**: The hashes in your hashes.txt file should look like the hash-mode 1800 hashes!

1. Run the following command to examine the extensive options for the hashcat command:

*hashcat --help*

1. Run the following command to crack the Linux passwords:

*hashcat -m 1800 -a 0 -o /etc/cracked.txt /etc/hashes.txt /usr/share/john/password.lst –force*

**Note**: We are leveraging hash-mode 1800 and straight attack mode.

1. Run the following command to show the passwords that were successfully cracked:

*gedit /etc/cracked.txt*

**Screen Shot 2)** Take a screen shot of your Hashcat results showing the passwords that were cracked.

Using more powerful rules and wordlists and allowing for more time, more difficult passwords can also be cracked using these two tools. However, by enforcing various password complexity rules when users are creating their passwords, you can make it much more difficult for rookies to use these tools and be successful in cracking the hashes. For professional hackers with dedicated brute-force machines and clusters of superior GPU’s, an attacker can easily crack any hash derived from a wider keyspace. Nonetheless, we will demonstrate how to enforce a more complex password policy in later tasks in this lab.

1. You can now shutdown the KaliLinux virtual machine as it will not be needed in the next task below.

**Task 2 – Cracking Windows Passwords**

The following task will demonstrate how to obtain passwords on a Windows device once you have access to an account with administrator privileges. If the password hashes on the system can be obtained and cracked, an attacker can use them to pivot to other machines if the login credentials are the same across systems.

Windows 10 passwords stored as NTLM hashes can be dumped and exfiltrated to an attacker's system in seconds. The hashes can be very easily brute-forced and cracked to reveal the passwords in plaintext using a combination of tools, including Mimikatz, ProcDump, John the Ripper, and Hashcat. We will demonstrate the use of ProcDump, Mimikatz and John the Ripper in this task since you have already seen the Hashcat utility crack the Linux password hashes previously.

The Local Security Authority Subsystem Service (LSASS) is a process in all Microsoft Windows operating systems that is responsible for enforcing the security policy on the system. It verifies users logging on to a Windows computer or server, handles password changes, and creates access tokens. It also writes to the Windows Security Log.

We will use the ProcDump utility provided by Microsoft to analyze dump files for various processes running in Windows to obtain the dump file of the lsass.exe process. We will then leverage Mimikatz to obtain the password hashes stored in the lsass dump file. We will then use John the Ripper to crack the password hash. First, we will need to download & install ProcDump, Mimikatz, and John the Ripper for our Windows device.

Perform the following steps on the Windows 10 virtual machine within Oracle VirtualBox:

1. Turn on your Windows 10 virtual machine in Oracle VirtualBox.

**Note**: You will only require the Windows 10 virtual machine in this particular task.

1. Login to the Windows 10 virtual machine with an account that has administrator privilege.
2. Access the ProcDump v9.0 utility from the following location:

<https://docs.microsoft.com/en-us/sysinternals/downloads/procdump>

1. Select the Download ProcDump link.
2. When prompted, select the Save option which will save the zip file in the downloads directory of the logged in user and then close the prompt Window.

The ProcDump is a utility provided by Microsoft to perform normal crash dump analysis and is even signed by Microsoft. Using ProcDump will not set off any alarms in normal intrusion detection software or be flagged by the antivirus software. However, we will next turn off the virus protection running on the Windows 10 virtual machine as it will not be happy when we try to install the well-known Mimikatz utility. It will detect that program as malware and prevent you from downloading it unless we disable antivirus. Remember, if we were an attacker we would more than likely be grabbing the lsass dump file on the target device and then executing the Mimikatz and John the Ripper tools from a separate dedicated machine used for cracking passwords as opposed to performing all the work on the target device where monitoring software would easily detect us. The bad guys like to fly under the radar!

Turn off Windows Defender by performing the following steps on the Windows 10 virtual machine:

1. Click the start menu (Window icon) in the lower-left corner and type the word “virus”.
2. Select the Virus & Threat Protection link.
3. At the Virus & Threat Protection screen, scroll down to the virus and threat protection settings and Select Manage Settings.
4. Turn OFF real-time protection. When prompted if you want to allow the action, Select Yes.
5. Close the Virus & Threat Protection Window.

**Note**: You will need to turn off virus protection when running Mimikatz if the Windows device is ever rebooted or restarted as the virus protection will be re-enabled.

1. Access the Mimikatz v2.2 utility from the following location:

<https://sourceforge.net/projects/mimikatz.mirror/>

1. Select the Released /2.20-20200308/mimikatz\_trunk.zip link.
2. When prompted, select the Save option which will save the zip file in the downloads directory and then close the prompt Window.
3. Access the John the Ripper v1.9 utility from the following location:

<https://www.techspot.com/downloads/6970-john-the-ripper.html>

1. Scroll down the page and Select Windows Binaries on the left side.
2. When prompted, select the Save option which will save the zip file in the downloads directory and then close the prompt Window.

**Note**: The zip files will be stored in the C:\Users\*YourLoginID*\Downloads folder. In this lab, whenever you see “*YourLoginID*” you need to replace that text with the login ID you are using on this device.

1. Extract the ProcDump, Mimikatz and John the Ripper zip files by right-clicking on each zip file in Windows Explorer and selecting Extract All and extracting them to the default locations.
2. You should now have three folders in your Downloads directory containing the binaries for the ProcDump, Mimikatz, and John the Ripper utilities.

**Note**: Save the original zip files you downloaded from the internet as you may need them if the antivirus software removes the binaries due to them being malicious.

We are now ready to leverage these utilities to crack the Windows password of the currently logged in account. In order to make the task go quickly, I **highly recommend** you change the password on the login account you have been using to something simple to crack. Otherwise, the cracking steps below could take hours to complete. Adversaries would not mind waiting but I figure you want to get on with the lab and your life! Set your Windows password to something easy to crack like **password1**.

**Dumping the LSASS data with ProcDump.** The first step in cracking a Windows password is to create a dump file containing the LSASS data discussed above. We will leverage the ProcDump tool to do that.

1. Open a command prompt on the Windows machine by Selecting the Start Button and typing CMD.
2. Right-click on the Command Prompt application and Select Run as Administrator (Then Select Yes).
3. Change directories to the ProcDump directory by typing the following:

cd \users\*YourLoginID*\Downloads\ProcDump

1. List all the options you can leverage for this command by typing the following:

procdump /?

**Note**: This utility can generate several types of dump files for any process running in memory.

1. Dump the lsass data by executing the following command:

procdump lsass.exe -ma lsass.dmp

**Note**: This will generate the lsass.dmp file and place it in the ProcDump folder.

**Acquiring the NTLM hash with Mimikatz.** The second step in cracking a Windows password is to acquire the NTLM hash from the dump file generated above. We will leverage the Mimikatz tool to do that.

1. Open a command prompt on the Windows machine by Selecting the Start Button and typing CMD.
2. Right-click on the Command Prompt application and Select Run as Administrator (Then Select Yes).
3. Change directories to the Mimikatz directory by typing the following:

cd \users\*YourLoginID*\Downloads\Mimikatz\_trunk\x64

1. List all the options you can leverage for this command by typing the following:

mimikatz --help

**Note**: The mimikatz developers have a sense of humor – check out the “coffee” and “answer” options.

1. Acquire the NTLM hash of the currently logged in account by executing the following commands:

sekurlsa::minidump c:\users\*YourLoginID*\Downloads\ProcDump\lsass.dmp

sekurlsa::logonpasswords

**Note**: If you scroll up in the results that were generated on the screen you should find the NTLM hash of the password associated with *YourLoginID*. The NTLM hash will be 32 characters in length.

1. Copy the NTLM hash and paste it into a new text file called hashes.txt stored in the ProcDump folder.

**Note**: We will now leverage the John the Ripper tool to crack the hash.

**Cracking the NTLM hash with John the Ripper.** The last step in generating the cleartext Windows password is to crack the NTLM hash from the hash file generated above. We will leverage the John the Ripper tool to do that.

1. Open a command prompt on the Windows machine by Selecting the Start Button and typing CMD.
2. Right-click on the Command Prompt application and Select Run as Administrator (Then Select Yes).
3. Change directories to the John the Ripper directory by typing the following:

cd \users\*YourLoginID*\Downloads\john-1.9.0-jumbo-1-win64\run

1. List all the options you can leverage for this command by typing the following:

john --help

1. Lets now crack the NTLM hash by executing the following commands:

john --format=NT c:\users\*YourLoginID*\Downloads\ProcDump\hashes.txt

john --show --format c:\users\*YourLoginID*\Downloads\ProcDump\hashes.txt

**Note**: You should see the password listed in cleartext if you were successful.

**Screen Shot 3)** Take a screen shot of your John the Ripper results showing the password that was cracked.

**Cracking the NTLM hash with a rainbow table.** One other way to generate the cleartext Windows password is to crack the NTLM hash by using a rainbow table. If you recall when we studied Chapter 3, a rainbow table is a quick way to crack a password by finding the hash for that password in a giant table of pre-computed hashes and passwords. We can leverage a website to do this.

1. Open a web browser on the Windows machine by selecting the Internet Explorer icon on the taskbar.
2. Navigate to the following website on your Windows virtual machine:

[https:\\crackstation.net](https://crackstation.net)

**Note**: This website contains billions of hashes for passwords – read the section at the bottom of the page to see how it actually works.

1. Let’s now try cracking the NTLM hash by using this rainbow table. Copy and Paste the hash from the hashes.txt file into the website and attempt to crack it.

**Note**: You should see the password listed in cleartext if you were successful.

**Screen Shot 4)** Take a screen shot showing the hash, type, and result from the web page.

**Task 3 – Enforcing Password Policies**

As you can see from the previous tasks in this lab, you must enforce password policies in order to prevent password cracking tools from being able to easily acquire the cleartext passwords. In this task we will demonstrate how to enforce password policies on both Windows and Linux devices.

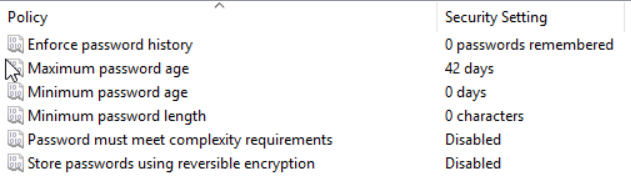
**Enforcing password policies on Windows.** One way of setting password policies on Windows devices is to leverage the local security policy console. We will demonstrate how to do that in the following steps.

1. Click the start menu (Window icon) in the lower-left corner and type the word “local”.
2. Select the Local Security Policy application.

**Note**: Another way is to enter secpol.msc in the search box in the lower left corner of the desktop.

1. Expand the Account Policies folder in the left pane of the window.
2. Select (Click on) the Password Policies folder.

**Note**: You should see the following default Windows password policies in the right pane of the window.



**Question 1)** Do you feel these default password policies are strong enough? Why or why not?

The default password policies are NOT strong enough. The password length requirement is way too short, it does not enforce password reuse, and it does not enforce complexity rules.

1. Change the password policies to the following security settings:

Enforce password history: 20 passwords remembered

Maximum password age: 30 days

Minimum password age: 5 days

Minimum password length: 14 characters

Password must meet complexity requirements: Enabled

Store passwords using reversible encryption: Disabled

**Screen Shot 5)** Take a screen shot showing the stricter password policies you have set.

**Question 2)** Since the password on your current account is probably set with a password that does not meet these new policies, what do you think will happen if you logout and try to log back in? No issues.

Try it… Why were you successful? It only enforces the complexity when new passwords are generated.

**Question 3)** Try changing your existing password to something like Password1. Were you successful? Why or why not? Now change it to a password that meets the new, enforced requirements. Were you successful?

No. It did not meet the complexity requirements. Yes. It met the complexity requirements.

In the real world, Windows devices (servers and desktops) are a part of a domain and the system administrator can set password policies on the domain controllers and push these policies to ALL Windows devices (and there can be thousands) in the domain. This is done by leveraging group policy which is a very powerful tool and a system administrators’ best friend!

**Enforcing password policies on Linux.** As is usually the case, performing password policy enforcement on Linux is not as easy as it is within Windows. I am going to let you research how this can be done and I’ll ask you a couple simple questions below.

**Question 4)** What is one way of enforcing password policy on a Linux operating system? You can choose which Linux distribution you would like to research (i.e., Red Hat, Suse, Ubuntu, Debian, Fedora, etc.).

**Note:** You will need to do some research on your own for this one.

**Question 5)** Last question….Did you like this lab? Why or why not?

Loved it!!

**Extra Credit)** Crack the following hashes shown below using any method you learned in this lab or online.

One Extra credit point for each correct answer…

**Linux Hashes:**

$6$Tb/MEhvAuMczADJc$u3A8gCyn4p3qwAJxghxHMct41cS9Y2G7RYfAyO8bNSfy3uvU3uGlgqwhPMR/kmNhGmWerhzzcN8dKOdKnr3/K. 🡺 password1

**Hint:** This is a not a strong password and it does not meet the enforced complexity requirements.

$6$BMBrsE1pFCV96k9b$o/9NDLUlDp7q1kfUsGOegrad/3STPJ5qCI6onj1cEcAWSnAXioOiowOxIZ96Gnj2aWlnWJ7qdnv1z.f8Es1.k. 🡺 Diploma2020!!!

**Hint:** This is a strong password that meets the enforced complexity requirements.

**Windows NTLM Hashes:**

9075168608b7aba2428c8387bfeb9aee 🡺 Hacker123

**Hint:** This not a very strong password that does not meet the enforced complexity requirements.

603d0577c4928244c02a1b7b2a9a62da 🡺 Graduate2020!!

**Hint:** This is a strong password that meets the enforced complexity requirements.