**Lab – Advanced Password Hacking with Metasploit**

**Overview**

Post-exploitation refers to the actions taken after a session is opened between the attacker and the target. A session is an open shell from a successful exploit or bruteforce attack. A shell can be a standard shell or Meterpreter. In this lab, you will learn how to capture usernames and passwords from a target machine using Metasploit.

No post-exploitation of a Microsoft Windows machine would be complete without acquiring as much information as possible about the Security Account Manager (SAM) database. The Security Account Manager (SAM) is a database present on all computers running Windows operating systems that stores user accounts and security descriptors for users on the local computer.

**Lab Requirements**

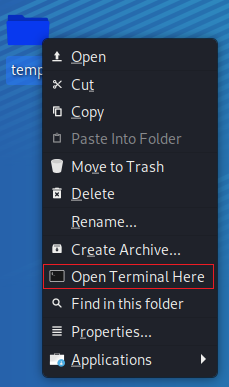
* One virtual install of Kali Linux.
* One virtual Install of Windows 7 Pro or Enterprise.
* An established Meterpreter session with your Windows 7 target.

**Begin the lab!**

Create a meterpreter session between your Kali machine and your Windows 7 Pro target.

From your Kali desktop, right-click on your working folder, and from the context menu,

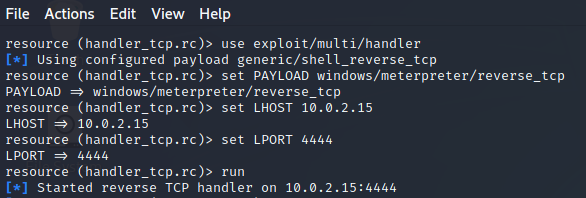
select **Open Terminal Here**.



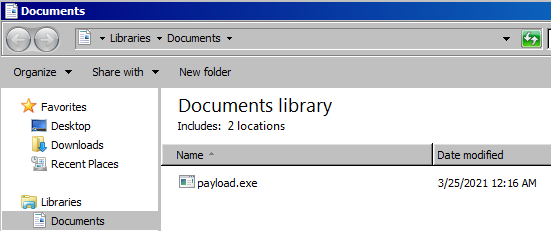
Use your meterpreter script to create a listener. At the terminal prompt, type:

msfconsole -r handler\_tcp.rc

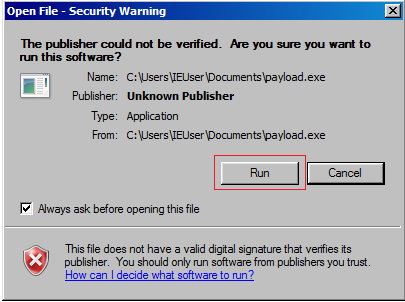
If the script completes successfully, your kali should be standing by for communication from your Windows 7 Pro machine when you launch the payload.exe.



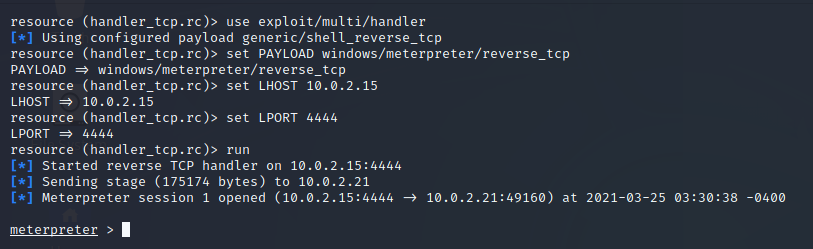
Return to your Windows 7 Pro machine. Open the Documents folder and 2X click the payload.exe file.



When prompt, click the Run button.

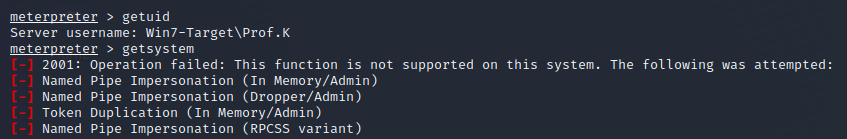


Return to your Kali terminal, and you should see a Meterpreter prompt.



At the Meterpreter prompt, type, **getuid**

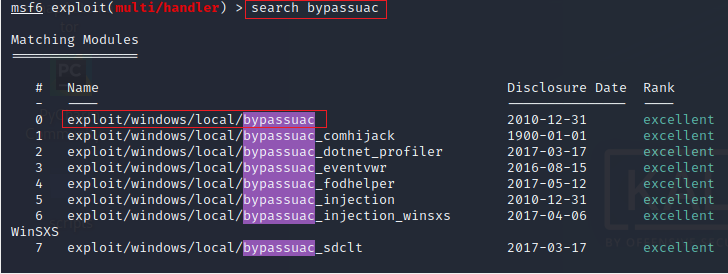
The getuid function returns the real user ID of the calling process. We can try and escalate our privileges using the **getsystem** command, but this operation fails as the command is not supported.

****

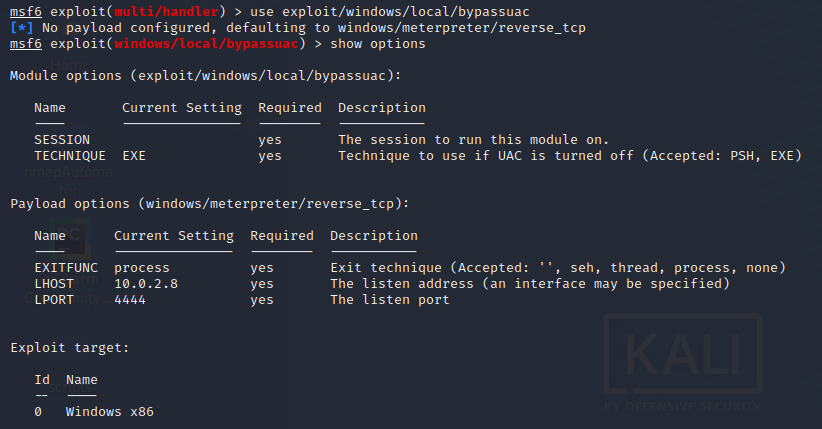
We need to bypass the UAC to get escalated privileges. To do this, we first need to background our current Meterpreter session. We do this by typing **background** at the prompt. Once the session has been background, we need to search for a UAC bypass exploit.



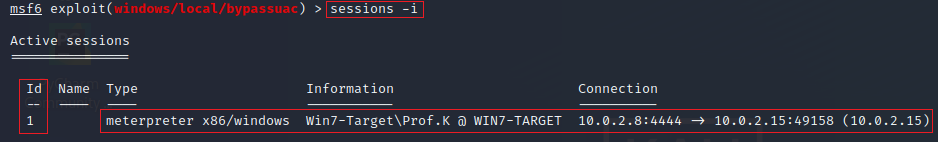
At the prompt, type **search bypassuac**.



At the prompt type, **use** **exploit/windows/local/bypassuac**

****

The missing parameter is the session ID. We can list all meterpreter sessions running using the **sessions -i** command.

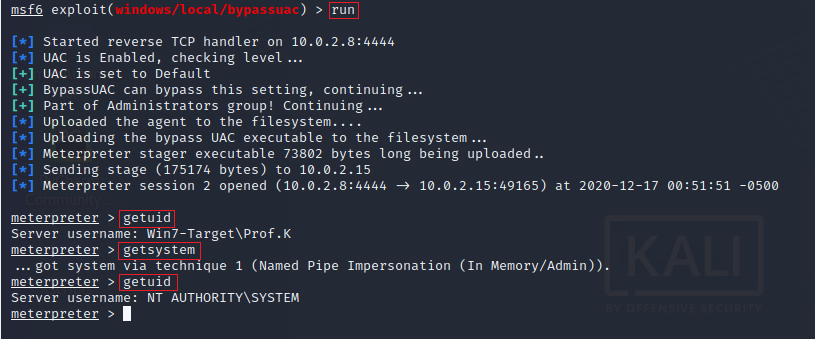


From the results, we know that our Metrepreter session is using the session ID of 1.

We next need to set the SESSION parameter to 1. At the prompt type, **set session 1**.



At the prompt, type **run**.



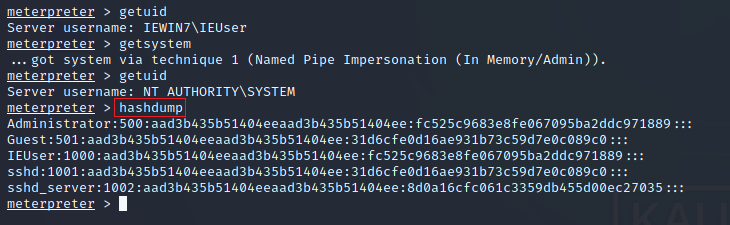
We check the real user ID of the calling process. Now that we have bypassed the UAC, we can escalate our privileges using the **getsystem** command, and we are currently running as NT AUTHORITY\SYSTEM.

**Begin the Lab!**

Because we have some level of administrative running as NT AUTHORITY\SYSTEM on the target machine, we can easily see the contents of the SAM database using the **hashdump** command.

The "hashdump" command is an in-memory version of the pwdump tool, but instead of loading a DLL into LSASS.exe, it runs inside a memory process injecting raw assembly code. Hashdump runs using the CreateRemoteThread and then reads the captured hashes back out of memory.

At the meterpreter prompt, type **hashdump.**

****

**Post Exploitation in Metasploit**

We can also dump the local user accounts from the SAM database using the registry. To do this, we can use the **post/windows/gather/hashdump** exploit.

Background your meterpreter session. (Note the session id number!)

At the prompt type, **use post/windows/gather/hashdump**

****

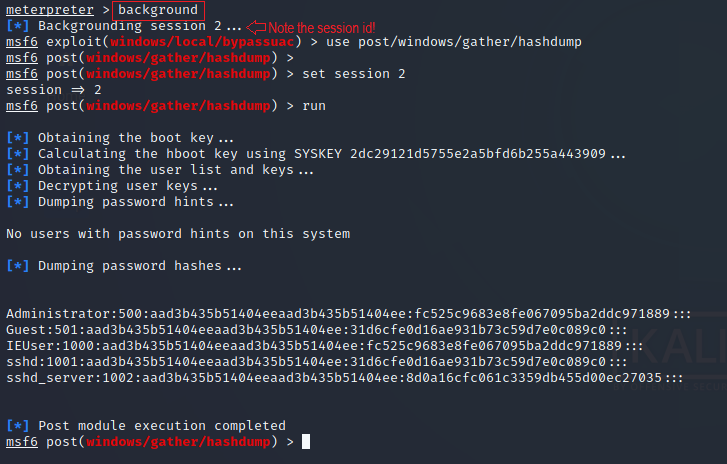
Press enter.

Set the session id to 2.



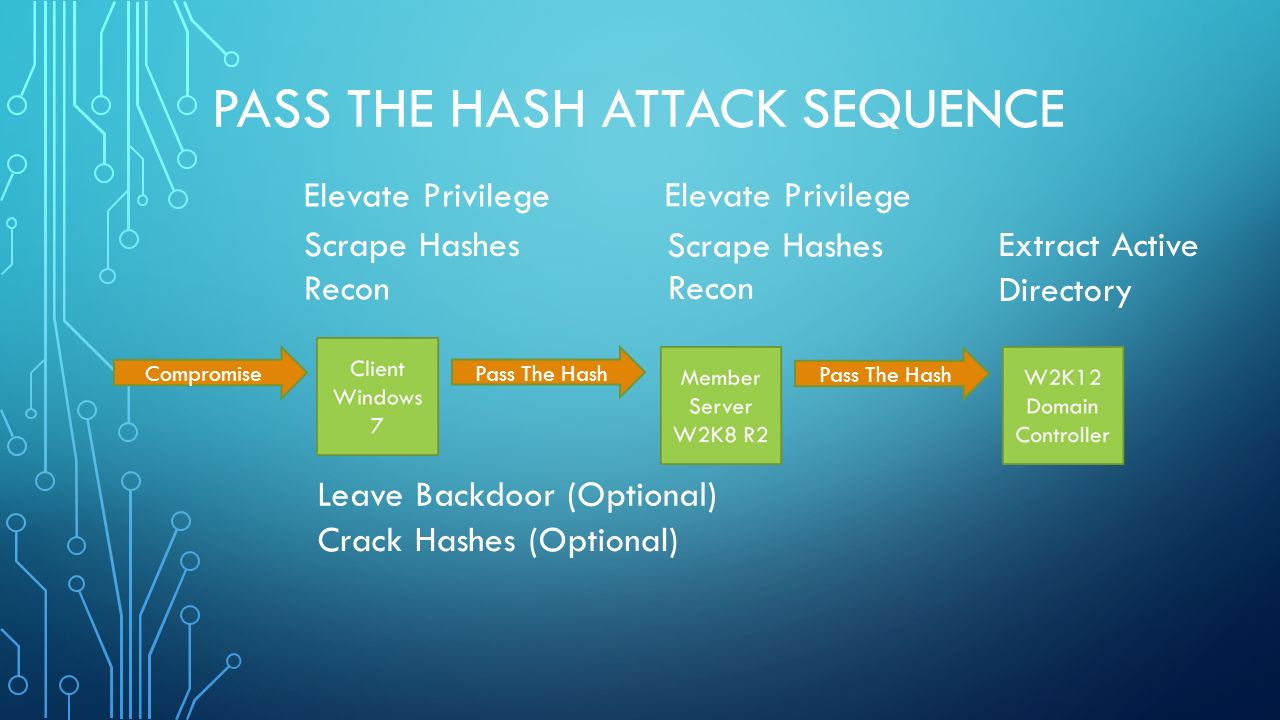
Type in the run command.



****

Note that this exploit also looks for any password hints.

At this point, we could perform a Pass the Hash using the administrator account and the revealed hash. The idea behind the Pass the Hash attack is that all the servers on the network use the same password for the administrator account. Regardless if it’s a file server joined to the domain or a domain controller, if we can get the hash for the administrator account of the file server, we can use the same hash to exploit the domain controller.



The lab environment to perform a Pass the HASH attack may be prohibitive for most students. Using a Server 2019 domain with one domain controller, one file server, and one Windows 10 client joined to the domain, I will demonstrate the attack. Check the section on Post Exploitation of Microsoft Windows for the video demonstration.

Moving on….

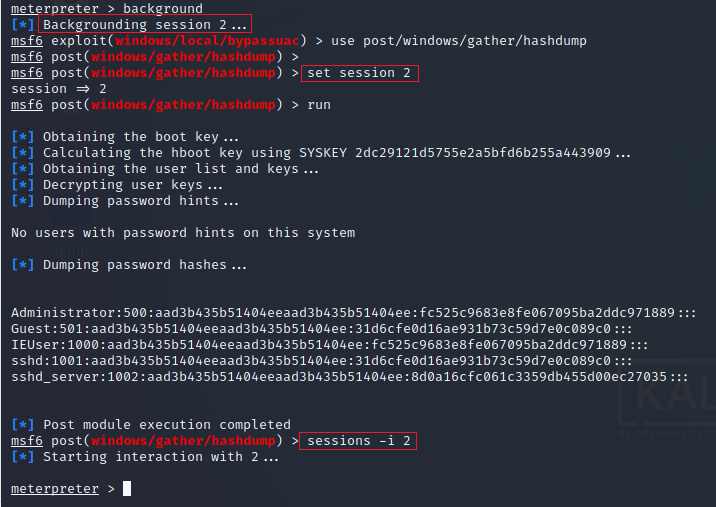
**Incognito attacks with Meterpreter**

In Metasploit, we can use an extension called **incognito** to allows us to perform token stealing and manipulation activities. In the privilege escalation stage of a penetration test, if we can steal the token of an administrator, we can perform higher privilege operations on the target.

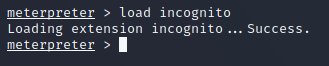
In this example, let’s imagine we have successfully exploited a remote system, and we have a meterpreter session. We first need to load the Meterpreter extension, incognito.

To return to your Meterpreter prompt, type **sessions -i 2.** Your session number may differ but use whatever session number was assigned when you last backgrounded Meterpreter.



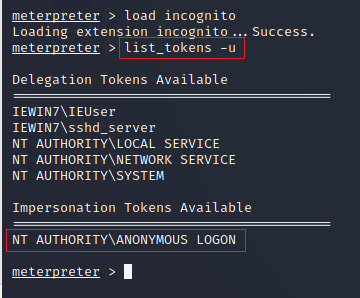


At your meterpreter prompt, type **load incognito**.



**List all Valid Tokens**

We can identify the valid tokens on the target machine using the **list\_tokens -u** the **-u** list tokens by a unique username.

****

**Caveat**

Ensure any user token you choose to impersonate is equal to or greater than the one you are currently using. For instance, if we decide to impersonate a regular user account, incognito will allow us to do so but, when you attempt to elevate your privileges back to NT AUTHORITY\SYSTEM, you will lack the privileges to do so. Using incognito to try and impersonate NT AUTHORITY\SYSTEM, you will be denied for lack of privileges.

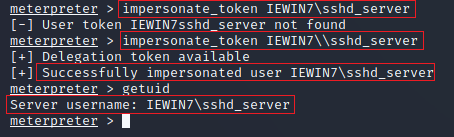
To have access to all the available tokens, you must be running with system privileges (NT AUTHORITY\SYSTEM). Not even administrators have access to all the tokens. For best results, escalate your privileges before using incognito (we did so earlier by bypassing the UAC and then running the **getsystem** command).

By compromising a domain controller with system privileges, we can wait for a domain administrator to login to the target machine and then use Incognito to impersonate the admins’ token and acquire their administrative privileges.

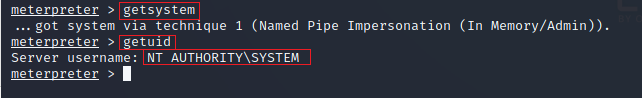
In this example, **sshd\_server** is a member of the Administrators group. We use the **impersonate\_token** followed by the name of the token you want to impersonate.

At the prompt, I typed, **impersonate\_token IEWIN7\\sshd\_server**

There is a known bug in Meterpreter that requires the name of the domain and the user’s token to be separated by a double backslash (\\).



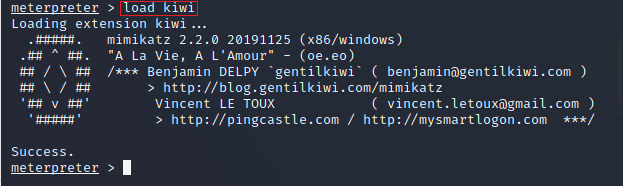
Once I corrected syntax, the impersonation was successful, and we confirmed this by using the **getuid** command. I was then able to use the administrator account to elevate by permission back to a system account.



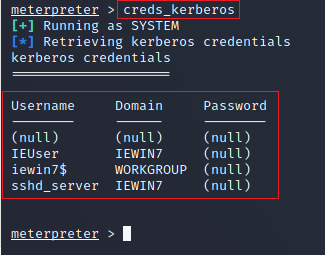
**Using Mimikatz**

Metasploit has two versions of Mimikatz available as Meterpreter extensions: version 1.0 by loading the mimikatz extension, and the newer version 2.x by loading the kiwi extension. In this lab, we are using the more recent version.

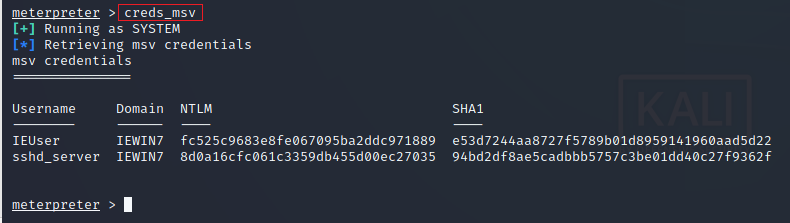
At the meterpreter prompt, type **load kiwi**



We will start by retrieving any Kerberos credentials from our target machine. To do this, we can use the **creds\_kerberos** command:

.

We next use the **creds\_msv** command to retrieve the LM/NTLM hashes using the MSV authentication package.



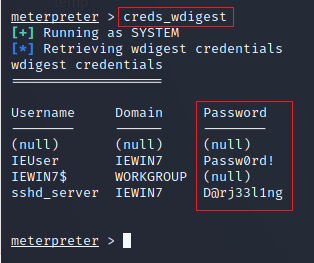
One of the features that have helped Mimikatz become an effective attack tool is the ability to retrieve cleartext passwords. After a user logs on, credentials are stored in memory by the Local Security Authority Subsystem Service (LSASS) process.

Using Mimikatz, we can retrieve the passwords as cleartext credentials.

**Caveat**

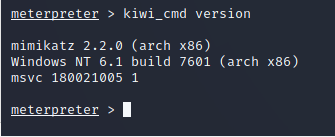
Starting with Windows 8.1 and Windows Server 2012 R2, cleartext credentials are no longer stored in memory.

To retrieve any stored passwords in cleartext, at the meterpreter prompt type, **creds\_wdigest**



Meterpreter comes with just a subset of commands of the most Mimikatz features, but we can get access to all the commands and features by using the **kiwi\_cmd version**

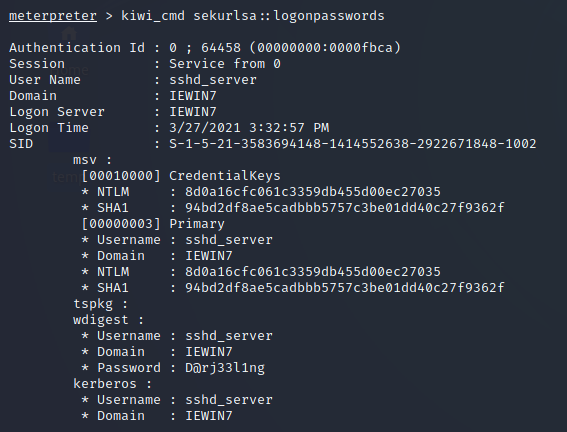
At your meterpreter prompt, type, **kiwi\_cmd version**

****

We can use the kiwi\_cmd command to list all the available credentials of the provider with the sekurlsa module:

At the meterpreter prompt, type the following command:

**kiwi\_cmd sekurlsa::logonpasswords**

****

**Summary –**

In this short lab, we looked at how we could extract usernames and passwords from our target machine using Metasploit running a Meterpreter shell. This is just one of several post-exploration tasks that should be performed once a target has been compromised.

End of the lab!