Target Machine NOOB 1 VM

Machine Information

```
Author: Shantanu Ramdas Kakade
```

- Platform: VulnHub
- Machine Name: **NOOB:1**
- Difficulty: Beginner
- Goal: Gain **root access** and capture the flag.

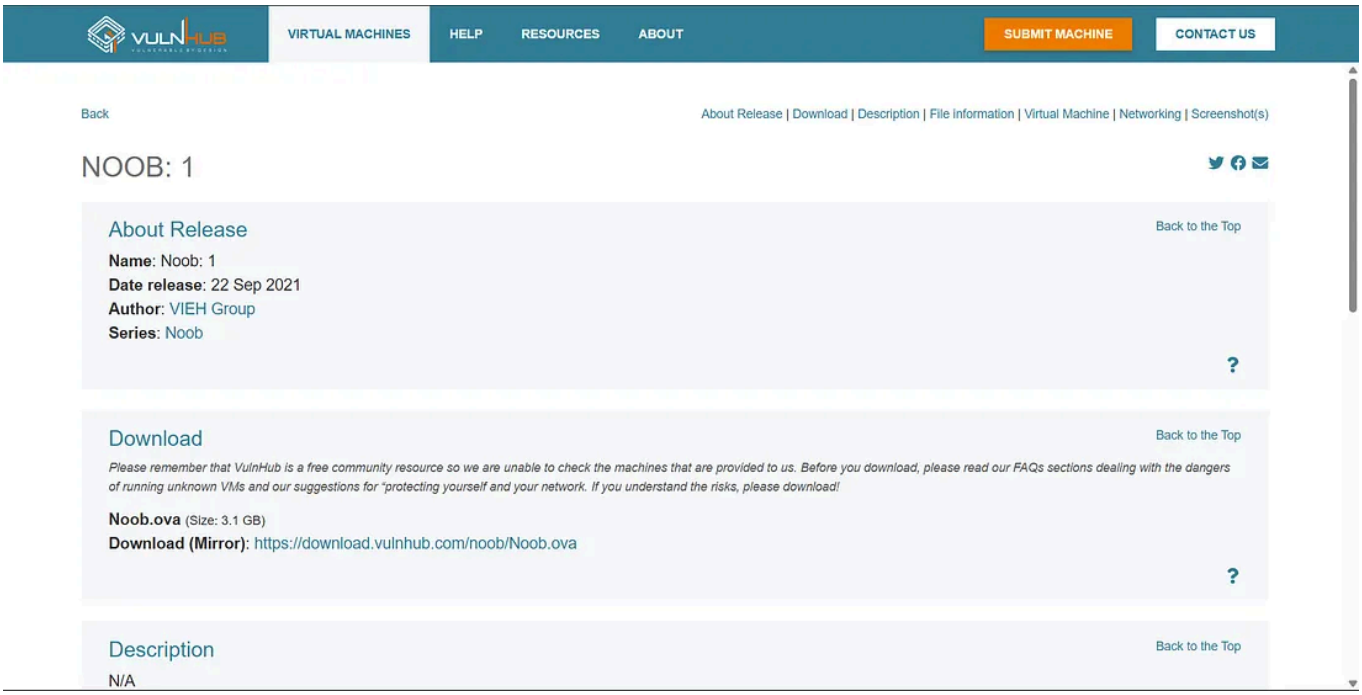
Lab Setup

Attacker Machine

- Kali Linux

Target Machine

- NOOB:1 VM

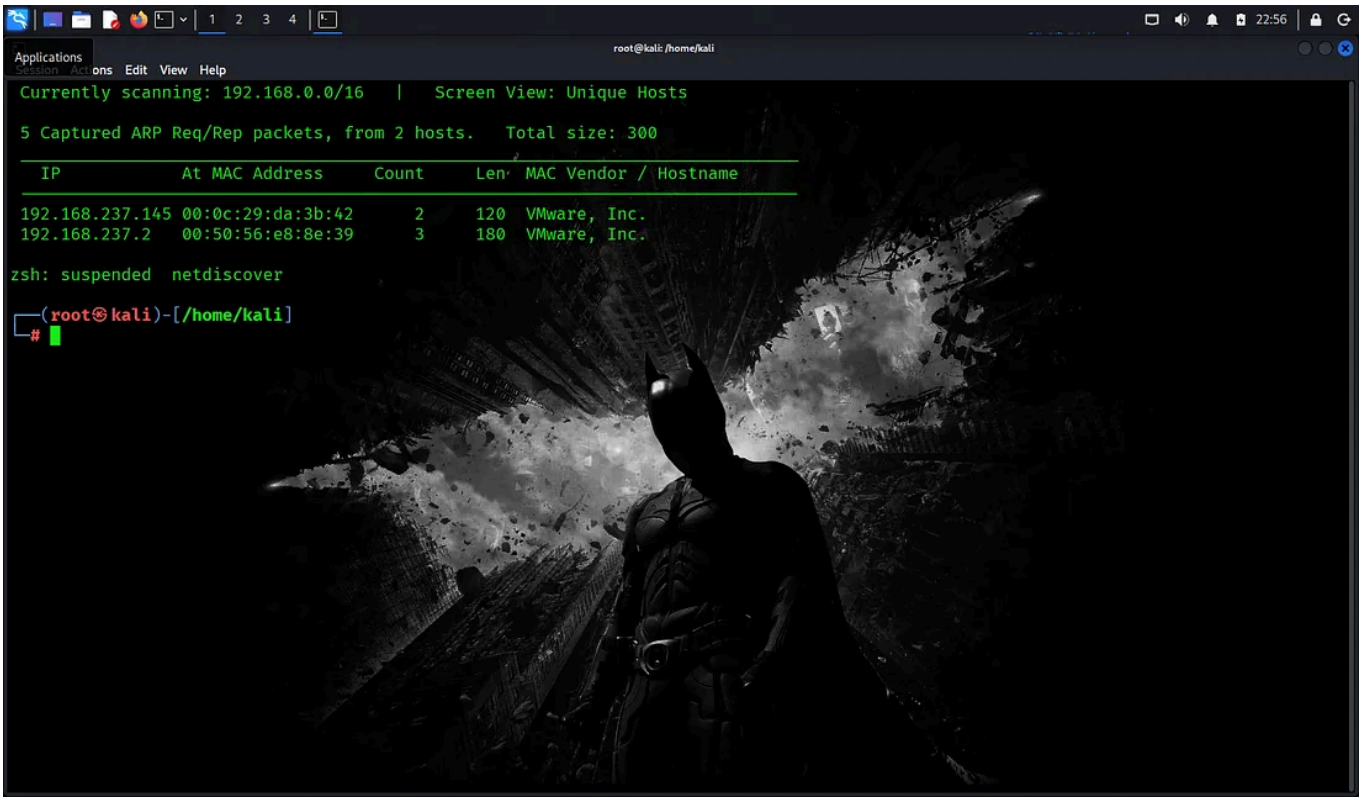


Both machines connected on the **same network (NAT or Host-Only)**.

Network Discovery

First, identify the target IP address

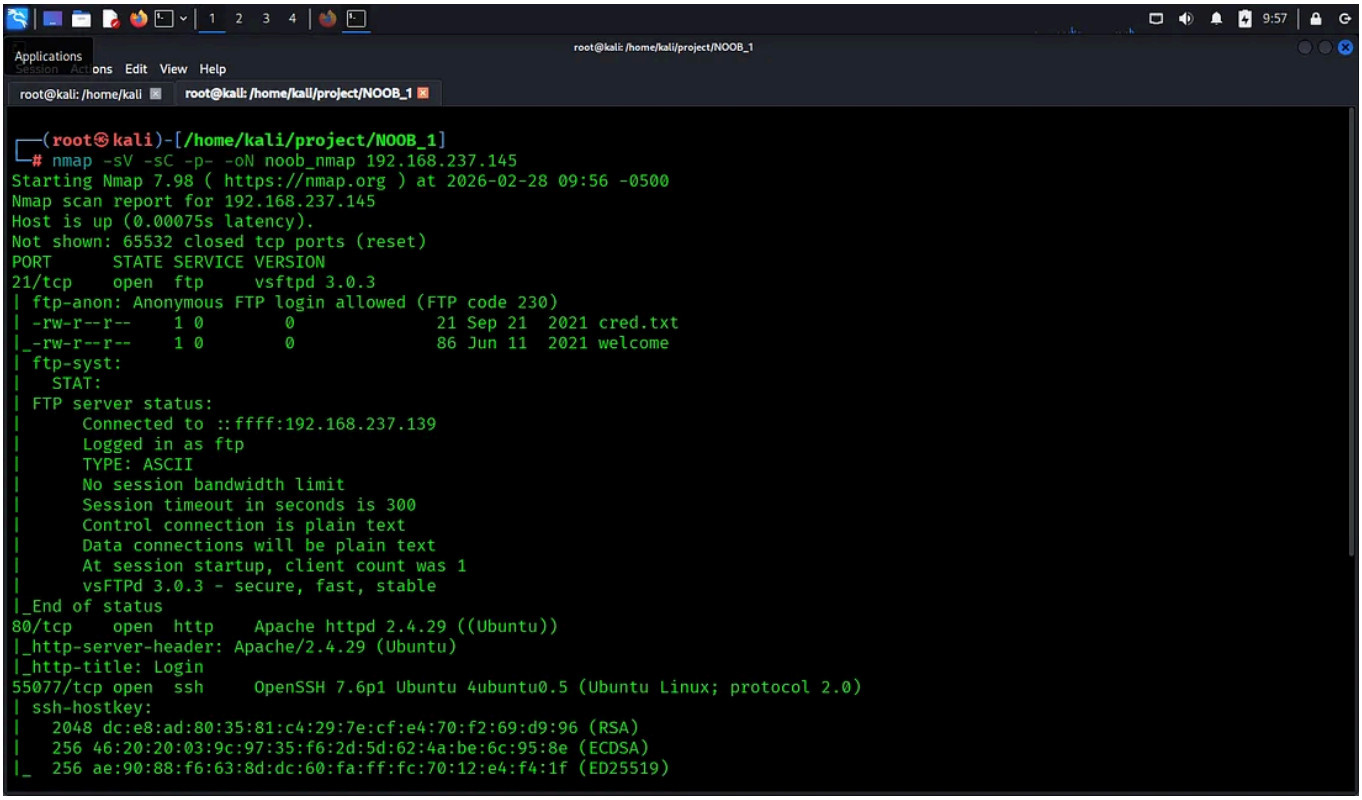
```
netdiscover
```



Port Scanning and Service Enumeration

After identifying the target IP (**192.168.237.145**), I performed a detailed scan using **Nmap** to discover open ports and running services.

```
nmap -sV -sC -p- -oN noob_nmap 192.168.237.145
```



FTP Enumeration

After discovering that **anonymous FTP login was allowed** during the Nmap scan, I attempted to connect to the FTP service. The FTP service was running **vsftpd 3.0.3** on port **21**.

Connecting to FTP

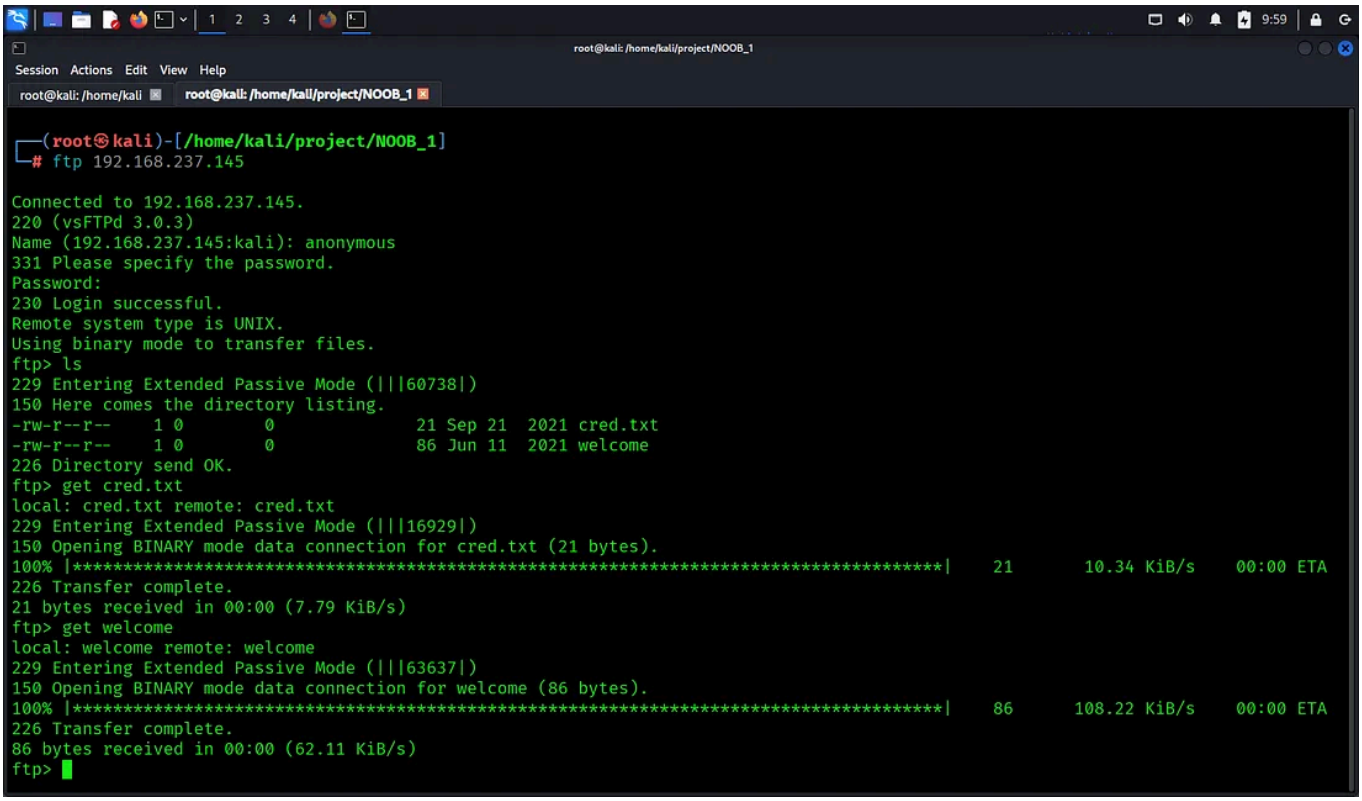
I used the following command to connect to the FTP server:

```
ftp 192.168.237.145
```

I entered **anonymous** as the username and **anonymous** as the password and successfully logged into the FTP server.

```
Username: anonymous
Password: anonymous
```

The login was successful, confirming that the FTP server allowed **unauthenticated access**. After logging into the FTP server, I used the `ls` command to list the available files. I found two files: **cred.txt** and **welcome**. I then used the `get` command to download these files to my local machine.



Analyzing cred.txt

After downloading the files from the FTP server, I examined the **cred.txt** file to check if it contained any useful information.

To read the file, I used the following command:

```
cat cred.txt
```

The file contained the following encoded string:

```
Y2hhbXA6cGFzc3dvcmQ=
```

The format suggested that the string was **Base64 encoded**, so I decoded it using the following command:

```
echo Y2hhbXA6cGFzc3dvcmQ= | base64 -d
```

The decoded result was:

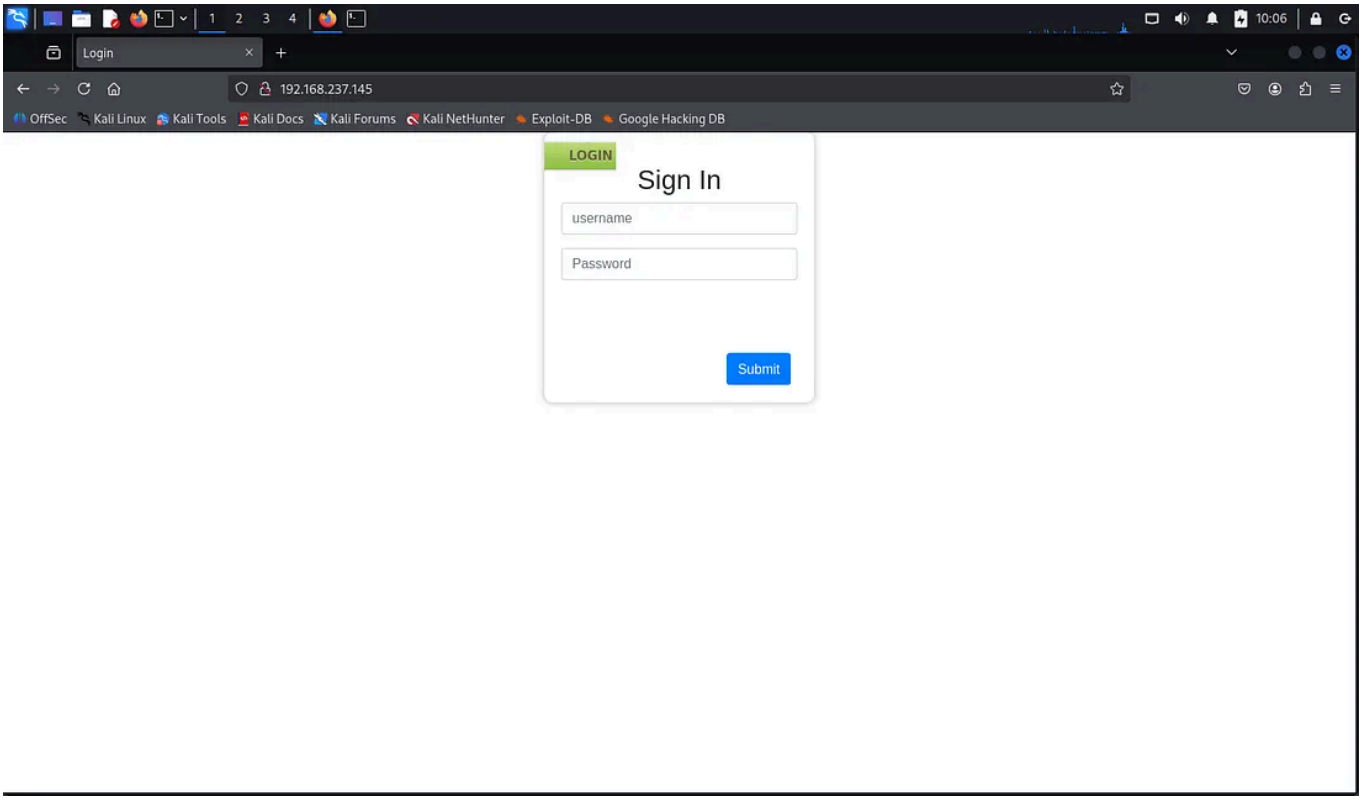
```
champ:password
```

Web Login Page

During enumeration, I visited the web service running on port **80**.

```
http://192.168.237.145
```

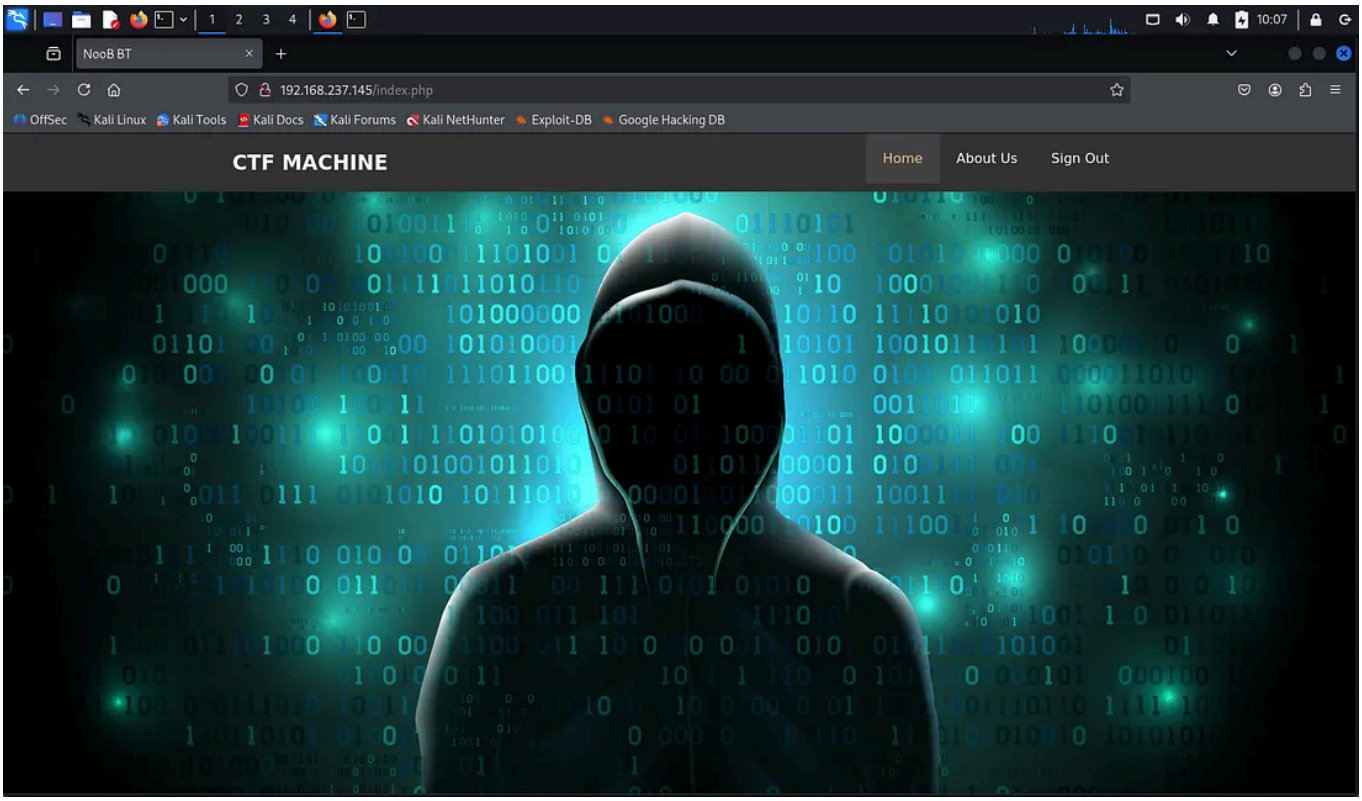
The page displayed a **login interface** with fields for a username and password. This indicates that the target machine hosts a **web application with an authentication system**. Since credentials were discovered earlier in the **cred.txt** file, they can be tested on this login page.



Credentials Found Earlier

```
Username champ
Password password
```

Using that credentials I successfully logged into the web application.



After logging into the web application, I clicked the **About Us** section. A file named **downloads.rar** was automatically downloaded. This file was then examined for further information.

Extracting `downloads.rar`

After downloading the **downloads.rar** file, I extracted it to examine its contents.

```
unrar x downloads.rar
```

After extracting the archive, I found two image files:

```
funny.bmp
funny.jpg
```

Since images in CTF challenges often contain hidden data, I used **Steghide** to check if any information was embedded inside the images.

Extracting Hidden Data

I attempted to extract hidden data from **funny.jpg** using the following command:

```
steghide extract -sf funny.jpg
```

Analyzing the Extracted File

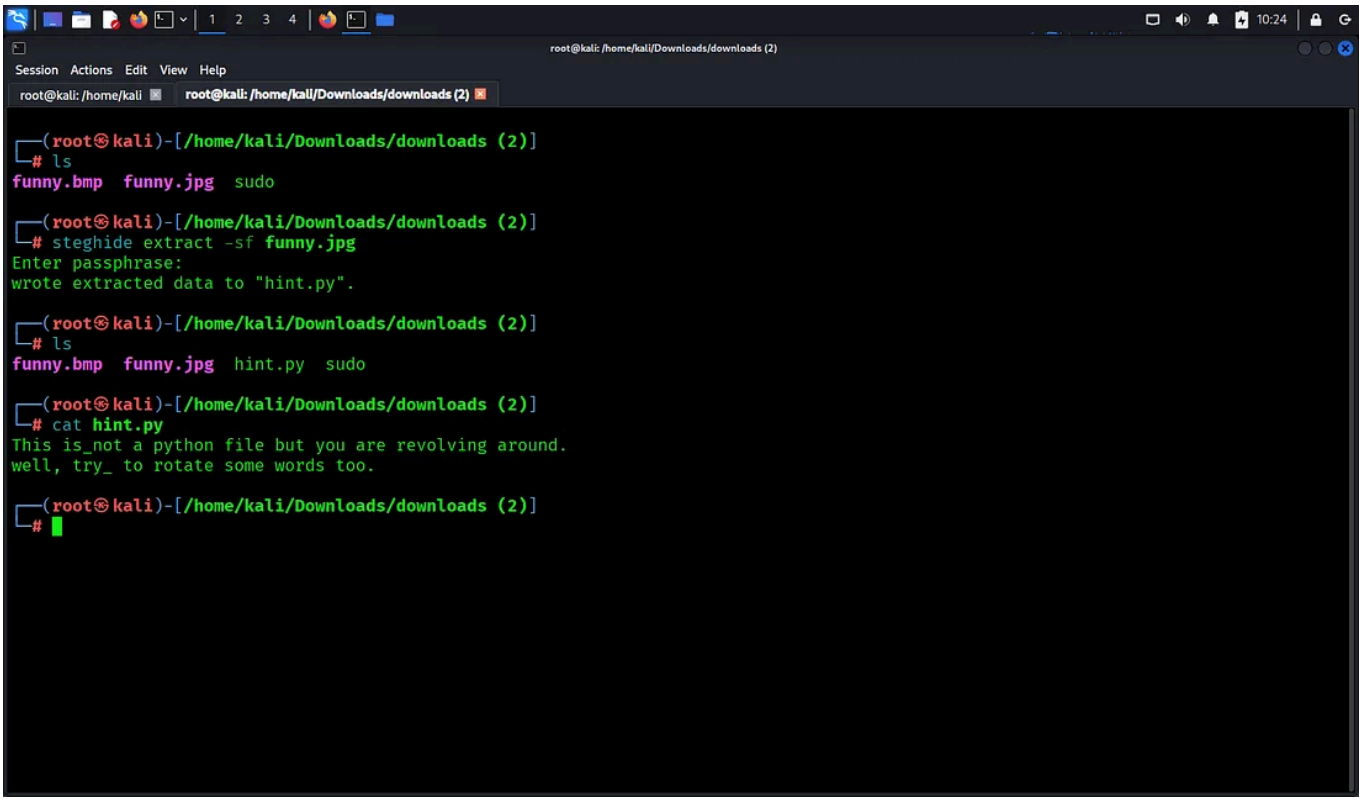
I then examined the contents of the file:

```
cat hint.py
```

The file contained the following message:

```
This is not a python file but you are revolving around.
well, try to rotate some words too.
```

I used `steghide` to extract hidden data from **funny.jpg**. This produced a file named **hint.py**. After reading the file, it suggested trying a **rotation-based cipher** to decode the hidden message.



Extracting Data from `funny.bmp`

Since the previous hint suggested trying rotations, I checked the second image file **funny.bmp** for hidden data using **Steghide**.

```
steghide extract -sf funny.bmp
```

After running the command, hidden data was successfully extracted and saved to a file named:

```
user.txt
```

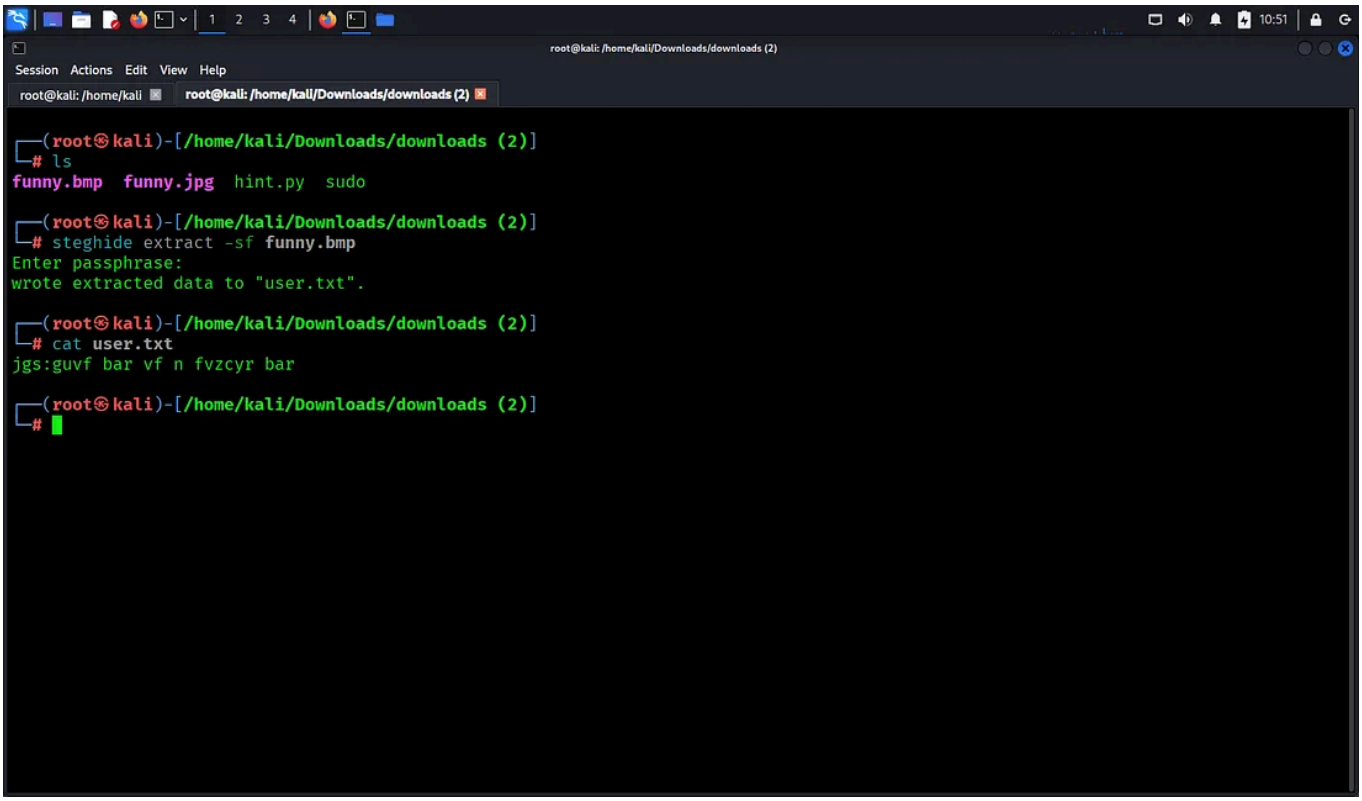
Reading the Extracted File

I then viewed the contents of the file using:

```
cat user.txt
```

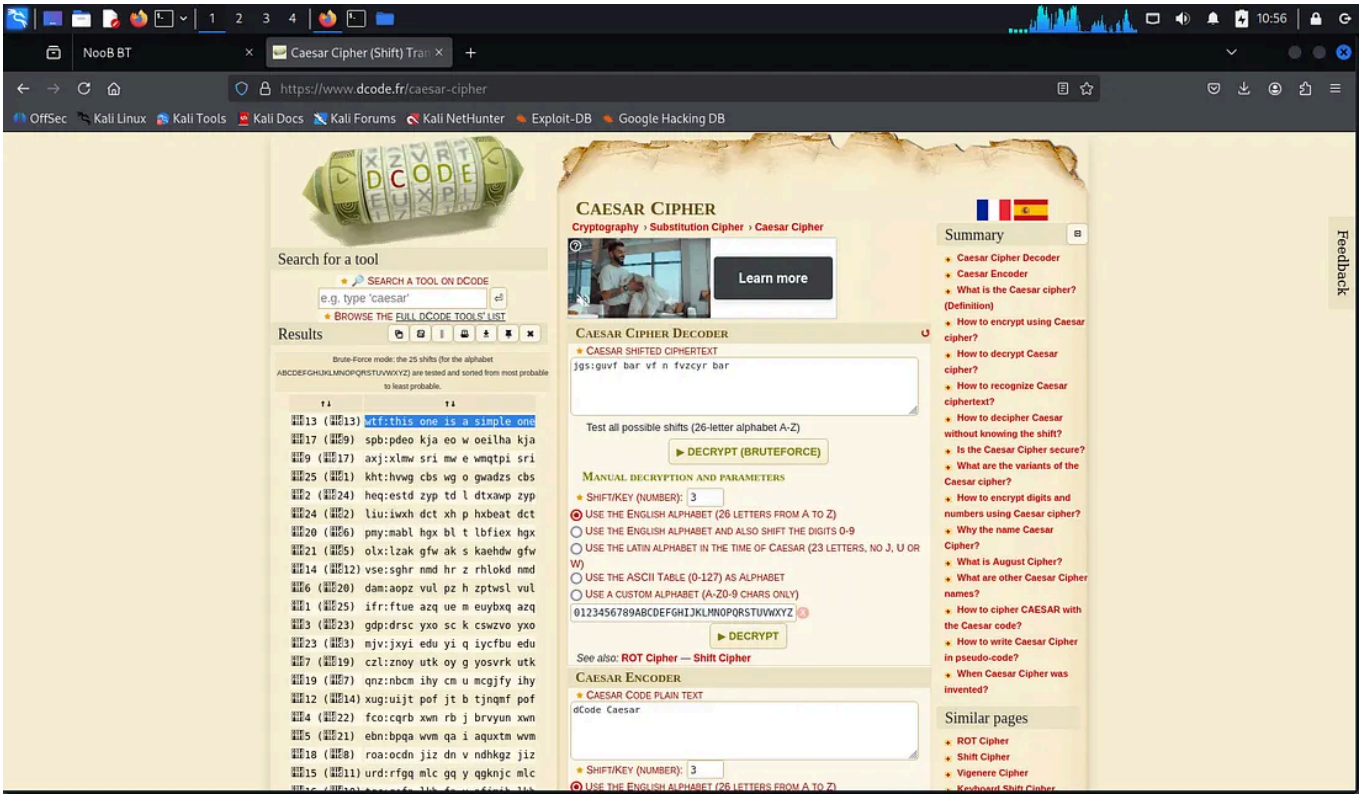
The file contained the following encoded message:

```
jgs:guvf bar vf n fvzcyr bar
```



Based on the earlier hint suggesting **word rotation**, I suspected that the message was encoded using a **Caesar cipher (ROT13)**

To decode the message, I used an online Caesar cipher decoder:



After decoding the text using **ROT13**, the message revealed

```
wtf : this one is a simple one
```

The message in **user.txt** appeared to be encoded using a ROT cipher. After decoding it using ROT13, the result was **“wtf:this one is a simple one.”**

SSH Login

After decoding the message, I obtained the credentials:

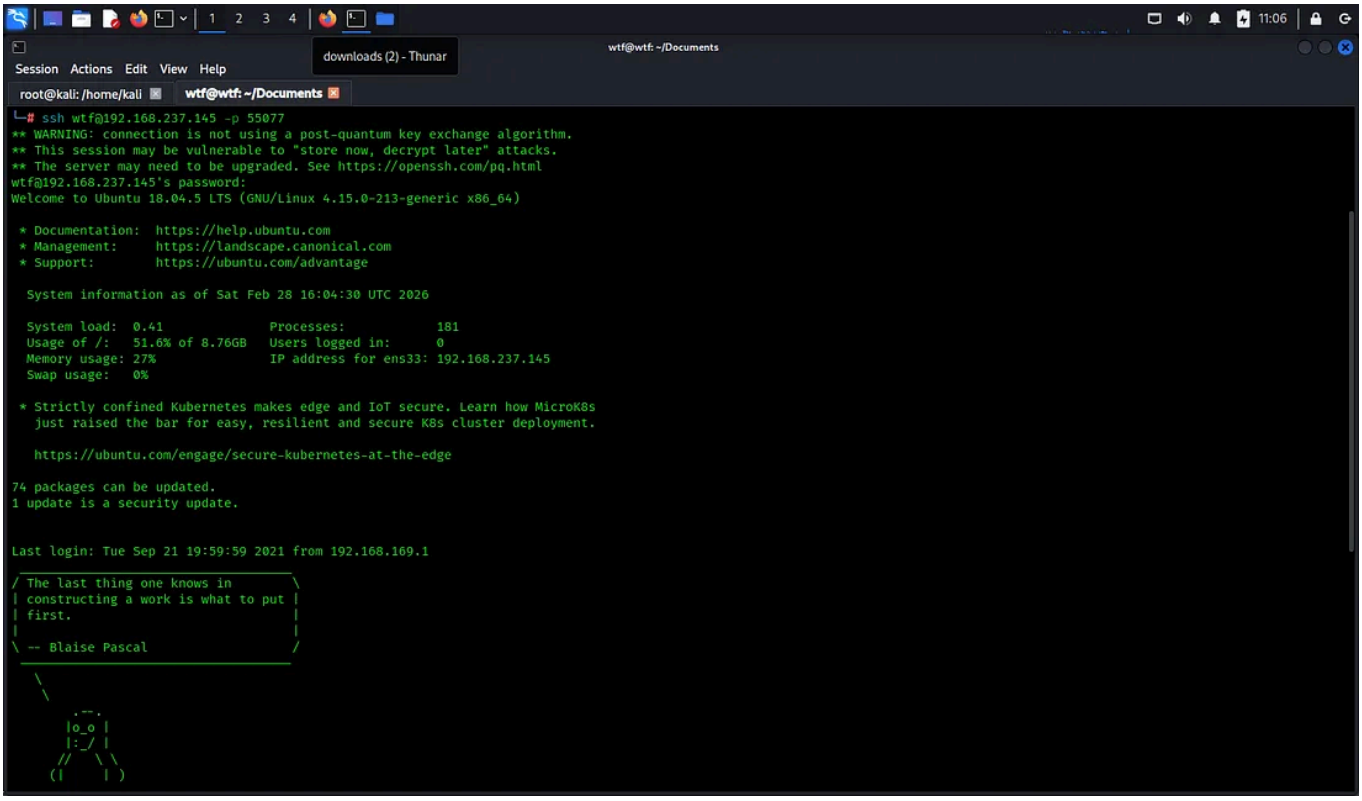
```
Username : wtf
Password : this one is a simple one
```

Since the **OpenSSH** service was running on port **55077**, I attempted to log in using these credentials.

Command Used

```
ssh wtf@192.168.237.145 -p 55077
```

Using the discovered credentials, I connected to the machine via SSH on port **55077** and successfully obtained a user shell.



Privilege Escalation

Step 1 — Check sudo permissions

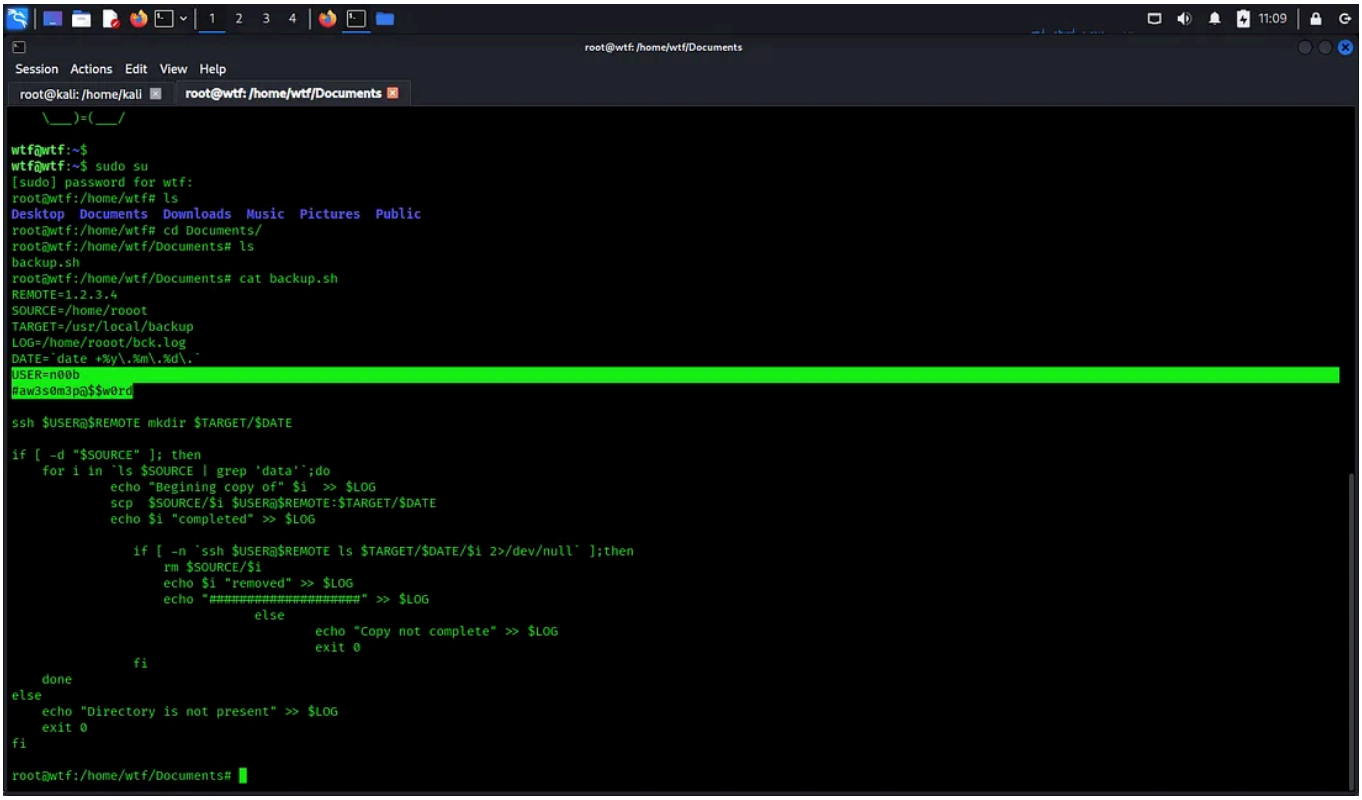
After logging in through SSH as the **wtf** user, I checked the sudo permissions.The output showed that the user could run **any command with sudo privileges**, which means privilege escalation is possible.

Step 2 — Switch to Root

Since sudo access was allowed, I switched to the root user.

```
sudo su
```

After entering the password, I successfully obtained **root access**.



Post Exploitation and Flag Retrieval

Step 1 — Check User Directories

After gaining **root privileges**, I started exploring the user directories.

```
cd /home/wtf
ls
```

The directory contained several folders:

```
Desktop  Documents  Downloads  Music  Pictures  Public
```

Step 2 — Locate the First Flag

I navigated to the **Downloads** directory.

```
cd Downloads
ls
```

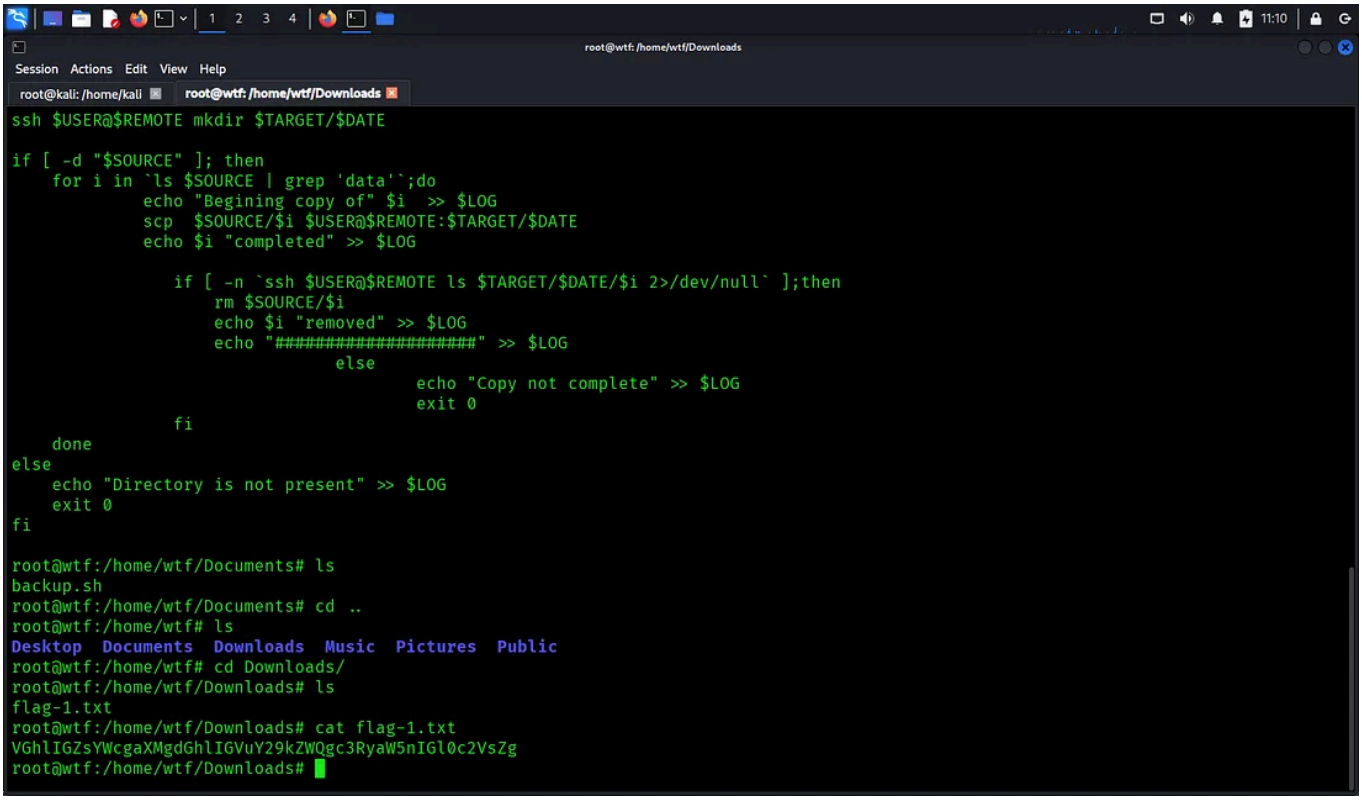
A flag file was found:

```
flag-1.txt
```

Output:

```
VGhlIGZsYWcgaXMgZGhIIGVuY29kZWQgR3RyaW5nIGI0c2VsZg
```

This is the **first flag of the machine**.



Step 3 — Search for Additional Clues

Next, I explored another directory:

```
cd /home/roooot
ls
```

I found another file:

```
flag.txt
```

Reading the file:

```
cat flag.txt
```

Output

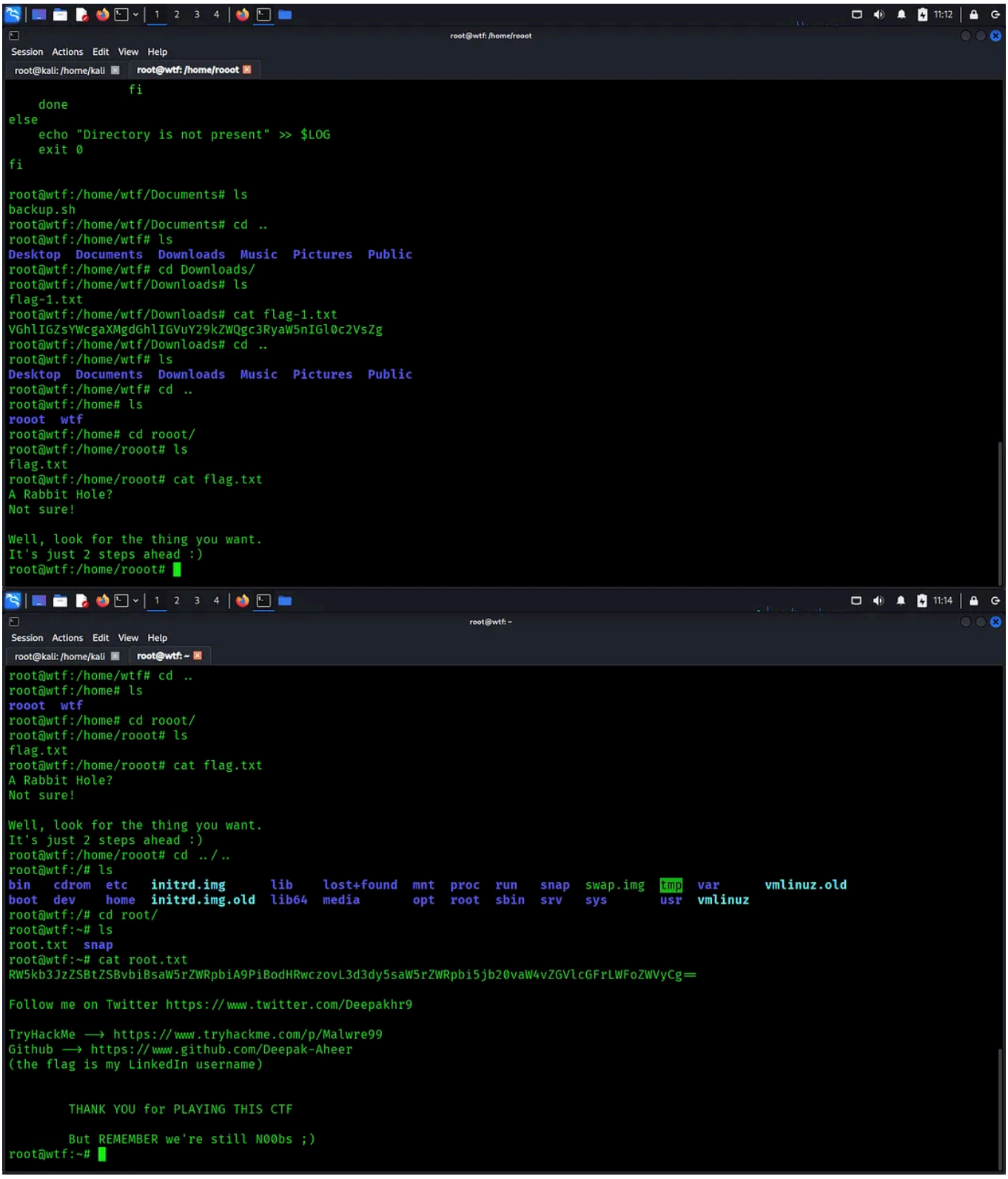
```
A Rabbit Hole?
Not sure!

Well, look for the thing you want.
It's just 2 steps ahead :)
```

After retrieving the first flag, I continued exploring the system to find more clues.

To move two directories back, I used:

```
cd ../..
```

This command takes the current directory **two levels up** in the directory structure. After moving back, I listed the directories to see what else was available.



Then I navigated to the root user's directory.

```
cd /root
ls
```

Here I found the final flag file:

```
root.txt
```

Conclusion

This machine was compromised using the following steps:

1. Network discovery
2. Port scanning and service enumeration
3. Anonymous FTP access
4. Steganography analysis
5. Cipher decoding
6. SSH login
7. Privilege escalation using sudo
8. Retrieving user and root flags

NOOB:1 machine successfully completed.