INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX Y LAMPIAO

Informe Técnico de Hacking Ético utilizando Kali Linux y Lampiao

Universidad Tecnológica del Chocó “Diego Luis Córdoba”

Docente

ING Rafael Sandoval Morales

Asignatura

Seguridad y Auditoría en Redes

Estudiante

Yeiber A. Mena Robledo

Quibdó/Chocó

24/09/2025

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX Y LAMPIAO

TABLA DE CONTENIDO

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX Y LAMPIAO

TABLA DE ILUSTRACIONES

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX Y LAMPIAO

1. INSTRODUCCION

El presente informe técnico describe el desarrollo de un laboratorio práctico de *hacking ético* enfocado en la identificación de vulnerabilidades y la evaluación de la seguridad de un sistema Linux. Para ello se emplearon las herramientas Kali Linux y un servidor objetivo denominado Lampiao, configurado para simular un entorno real. El ejercicio permitió aplicar técnicas de recolección de información, escaneo de puertos, pruebas de autenticación y escalada de privilegios, demostrando cómo un atacante podría comprometer un sistema si no se implementan controles de seguridad adecuados. Este proceso busca fortalecer las competencias en auditoría de redes, sensibilizar sobre los riesgos existentes y destacar la importancia de implementar medidas preventivas que garanticen la integridad de los sistemas informáticos.

2. OBEJETIVOS

Objetivo General

Realizar un ejercicio de *hacking ético* utilizando Kali Linux para identificar, explotar y documentar vulnerabilidades en un servidor Lampiao, con el fin de reforzar el conocimiento en seguridad y auditoría de redes.

Objetivos Específicos

- Emplear herramientas de reconocimiento como Nmap y CeWL para la recolección de información del sistema objetivo.
- Analizar los servicios y puertos abiertos para detectar posibles puntos de acceso no autorizados.
- Ejecutar pruebas de autenticación controladas con Hydra para evaluar la fortaleza de las credenciales.
- Implementar técnicas de escalada de privilegios para obtener permisos de administrador.
- Acceder y manipular bases de datos internas para evidenciar el impacto de una intrusión exitosa.
- Documentar cada etapa del proceso.

3. CAPITULO#1 DESARROLLO DEL LABORATORIO

3.1 Escaneo de Puertos

Se realizó un escaneo de puertos en el host **178.90.90.14** utilizando la herramienta **Nmap** con el rango de puertos de la 1 a la 2000. El escaneo permitió identificar que el sistema objetivo se encuentra activo.

Del total de puertos analizados, se evidenciaron **tres servicios en estado abierto**:

- **Puerto 22 (TCP – SSH):** Servicio de conexión remota segura.

  Puerto 80 (TCP – HTTP): Servicio web en ejecución.

- **Puerto 1898 (TCP – cymtec-port):** Servicio no común, posiblemente relacionado con aplicaciones específicas o configuraciones del sistema.

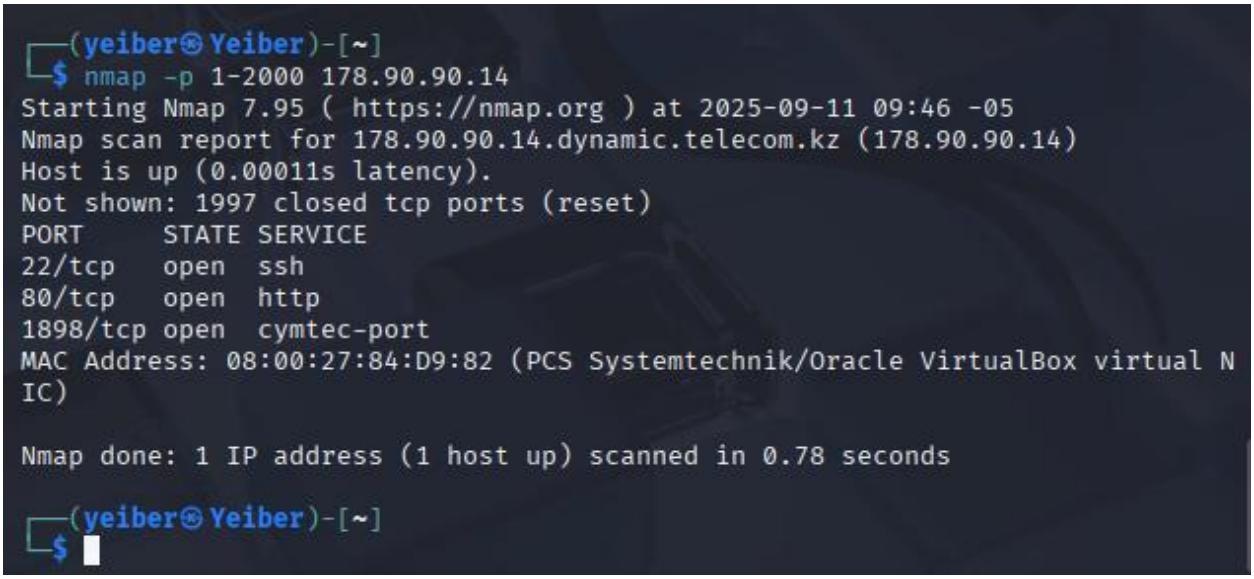

Ilustración 1-Escaneo de Puertos

3.2 Evidencia del Servicio Web

Al revisar el puerto **1898/TCP**, se encontró que este abre una página web titulada *"Lampião, herói ou vilão do Sertão?"*.

La página muestra un sistema en el que los usuarios pueden **iniciar sesión** con un nombre de usuario y una contraseña, además de opciones para crear una nueva cuenta o recuperar la clave. Esto indica que no se trata de un puerto sin uso, sino que está vinculado a un sitio web que funciona de manera activa e interactiva.

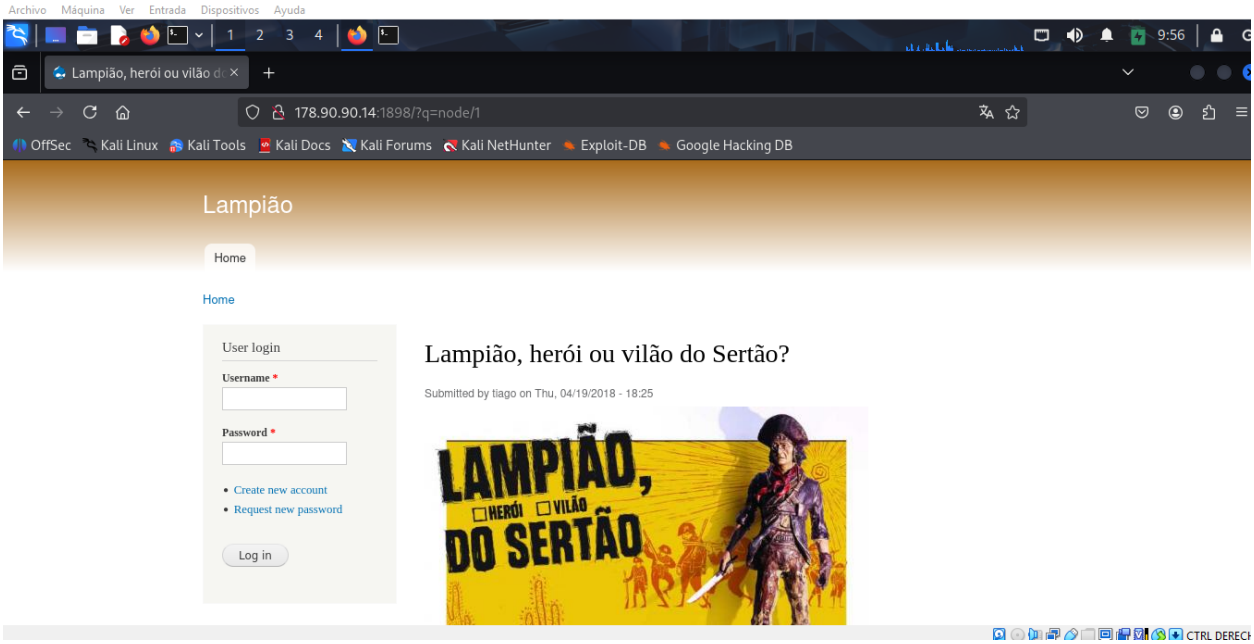

Ilustración 2-Evidencia del Servicio Web

3.3 Escaneo Detallado del Puerto 22

Se realizó un análisis más profundo sobre el puerto **22/TCP**, el cual está asociado al servicio **SSH**. Este servicio normalmente se utiliza para la administración remota de sistemas.

El escaneo permitió identificar que el servidor está ejecutando **Ubuntu Linux** con una versión del servicio **OpenSSH 6.6.1p1**.

```
└─$ nmap -p 22 178.90.90.14 -A -sC
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-11 09:58 -05
Nmap scan report for 178.90.90.14.dynamic.telecom.kz (178.90.90.14)
Host is up (0.00097s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.7 (Ubuntu Linux; protoco
l 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: 08:00:27:84:D9:82 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
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   0.97 ms 178.90.90.14.dynamic.telecom.kz (178.90.90.14)
```

Ilustración 3-Escaneo Detallado del Puerto 22

3.4 Recolección de Información y Evidencia de Archivos

Se utilizó la herramienta **CeWL** para analizar el sitio web alojado en el puerto **1898**. Esta utilidad permitió extraer palabras del contenido de la página y almacenarlas en un archivo denominado **Diclampiao.txt**. Dicho archivo puede servir posteriormente como base para generar listas de contraseñas o realizar pruebas relacionadas con autenticación.

Posteriormente, se comprobó la existencia del archivo en el directorio del usuario mediante el comando **ls -la**, donde se observa claramente que el archivo **Diclampiao.txt** fue creado con éxito, junto con otros documentos y configuraciones propias del sistema.

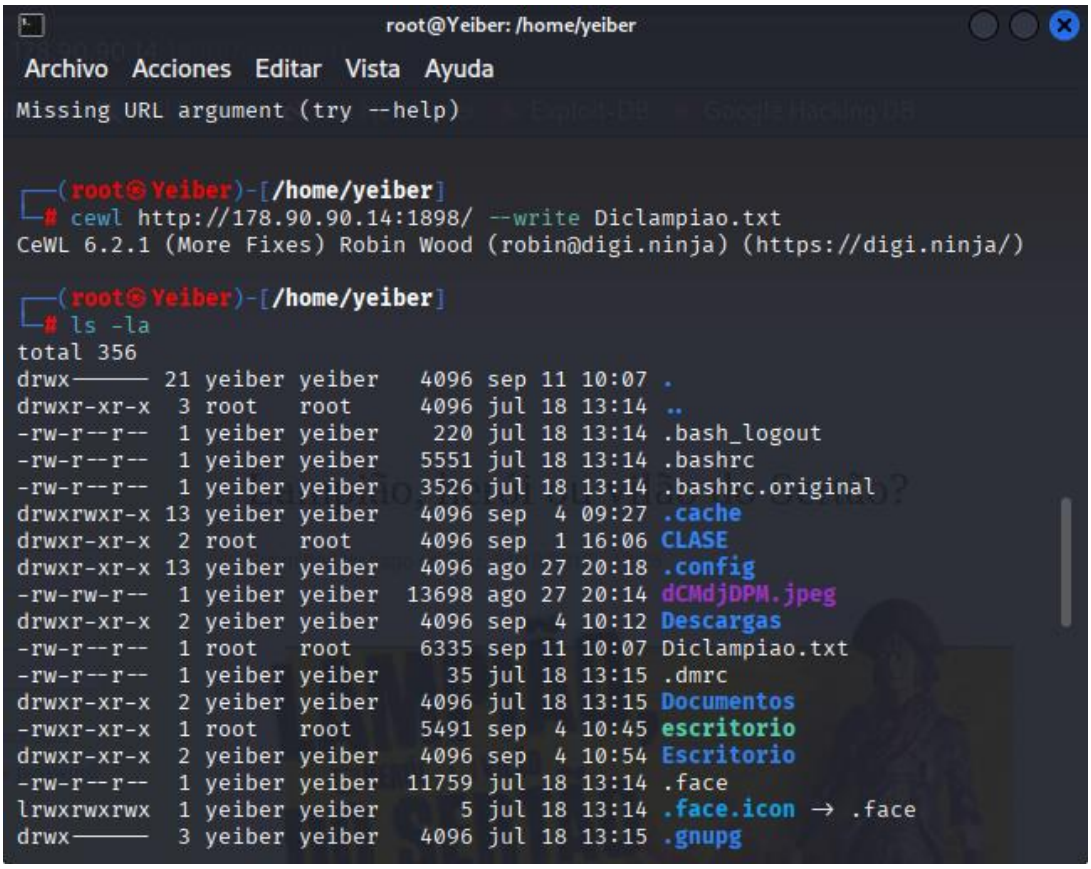

Ilustración 4-Recolección de Información y Evidencia de Archivos

3.5 Análisis del Archivo Generado

Una vez creado el archivo **Diclampiao.txt**, se procedió a verificar su contenido utilizando el comando **wc -l**, que permite contar la cantidad de líneas. El resultado mostró que el archivo contiene **844 palabras registradas**, extraídas directamente del sitio web previamente identificado.

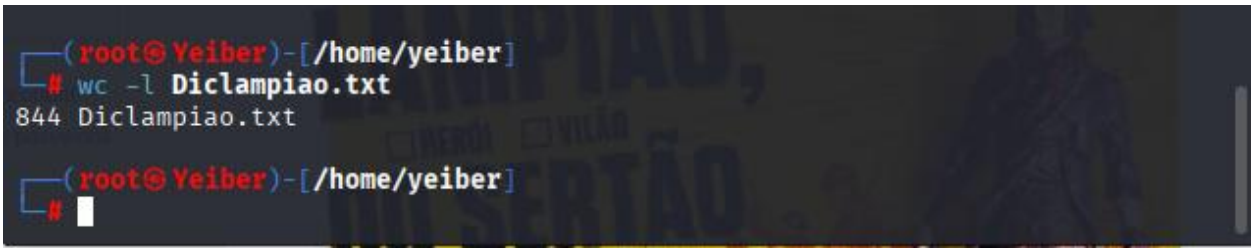

Ilustración 5-Análisis del Archivo Generado

3.6 Contenido del Archivo Generado

Finalmente, se revisó el contenido del archivo **Diclampiao.txt** utilizando el comando **cat**, lo que permitió visualizar directamente las palabras almacenadas. Entre ellas se encuentran términos relacionados tanto con la temática del portal web como con funciones del propio sitio, tales como *account*, *User*, *page*, *password*, entre otros.

La presencia de estas palabras confirma que la recolección de información fue exitosa, ya que el archivo contiene vocablos útiles derivados del análisis de la aplicación web.

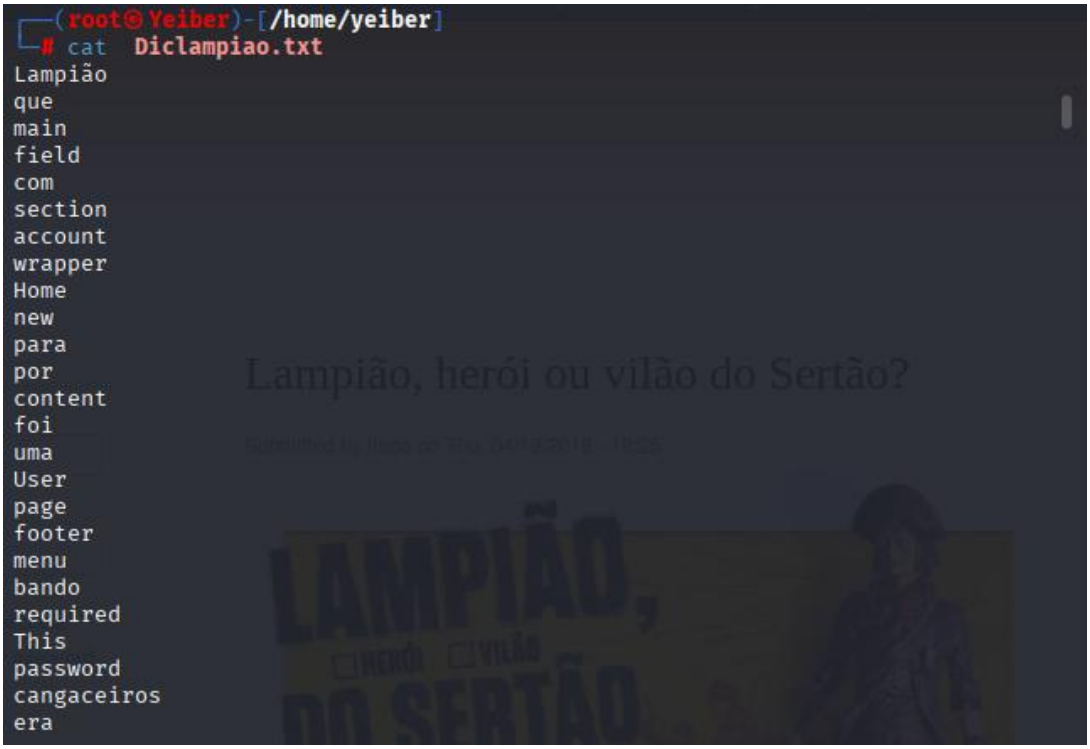

Ilustración 6-Contenido del Archivo Generado

3.7 Preparación de Pruebas con Hydra

En esta etapa se utilizó la herramienta **Hydra**, la cual está diseñada para realizar pruebas de acceso sobre servicios que requieren autenticación. En la imagen se observa la ejecución inicial del programa y el despliegue de sus diferentes opciones de uso.

La finalidad de esta acción no fue realizar un ataque directo, sino **mostrar las capacidades de la herramienta** y dejar en evidencia que existen mecanismos que, en manos de un investigador, pueden servir para evaluar la fortaleza de las contraseñas y la seguridad de los accesos.

Ilustración 7-Preparación de Pruebas con Hydra

3.8 Pruebas de Autenticación (Hydra)

Se realizó una prueba controlada sobre el servicio **SSH (puerto 22)** empleando la herramienta **Hydra**, indicando el usuario objetivo **tiago** y una lista de posibles contraseñas recolectadas previamente (archivo *Diclampiao.txt*). La ejecución intentó automáticamente las credenciales del archivo contra el servidor, pero **no se encontró ninguna contraseña válida**; el proceso finalizó mostrando *"0 valid password found"*.

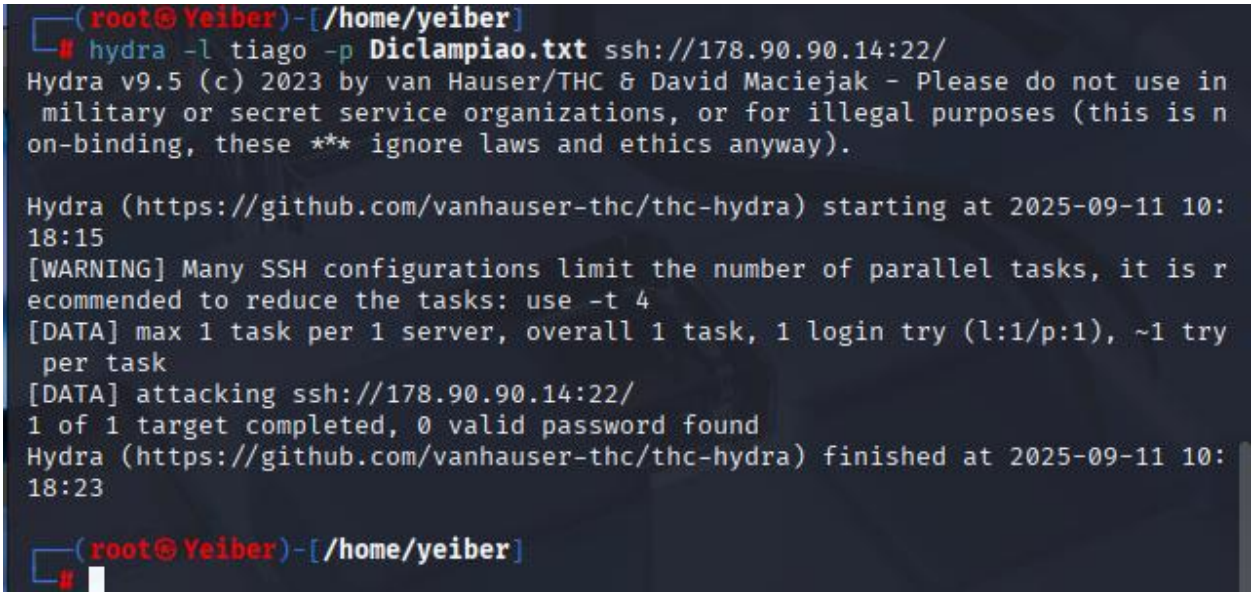

Ilustración 8-Pruebas de Autenticación (Hydra)

3.9 Acceso al Servidor

Tras varios intentos de conexión al servicio **SSH**, se consiguió ingresar de manera exitosa con el usuario **tiago**. Una vez validada la contraseña correcta, el sistema permitió el acceso, mostrando que se trataba de un servidor con **Ubuntu 14.04.5 LTS**.

En la terminal se observa también información del sistema, incluyendo la versión instalada y un mensaje sobre la disponibilidad de una actualización hacia **Ubuntu 16.04.7 LTS**. Así mismo, se evidencia la fecha y origen de la última conexión registrada al servidor.

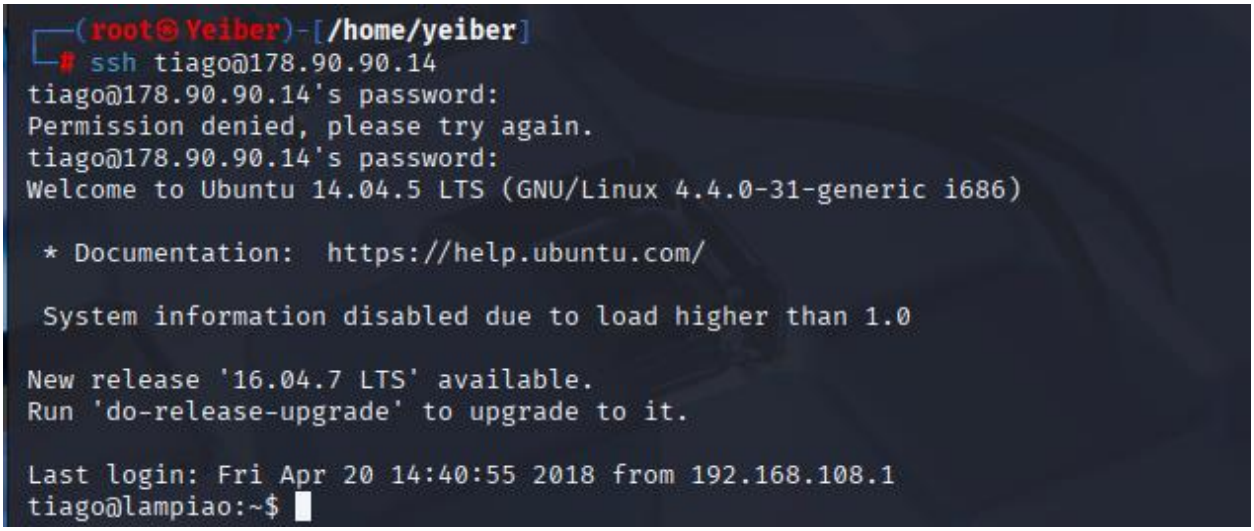

Ilustración 9-Acceso al Servidor

3.10 Ejecución de Herramientas para Escalada de Privilegios

Después de obtener acceso al servidor como usuario **tiago**, se procedió a clonar la herramienta **LinEnum** desde GitHub. Esta utilidad se usa comúnmente para **recolectar información del sistema**, detectar configuraciones débiles y posibles vías para escalar privilegios dentro de la máquina comprometida.

El comando ejecutado fue:

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

La clonación se completó exitosamente, descargando todos los archivos necesarios (234 objetos en total).

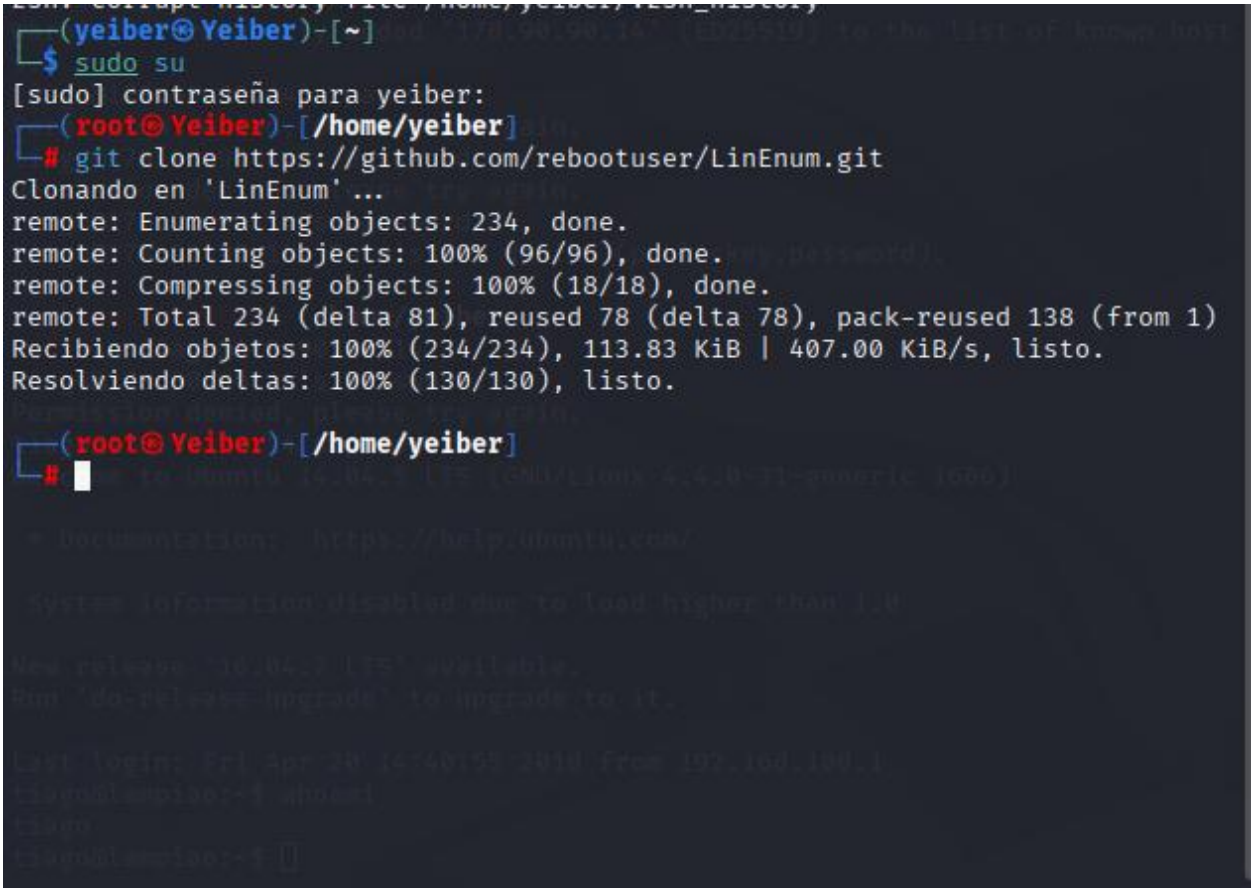

Ilustración 10-Ejecución de Herramientas para Escalada de Privilegios

3.11 Exploración del Sistema y Preparación para la Escalada de Privilegios

Después de clonar la herramienta **LinEnum**, se accedió al directorio que contiene sus archivos. La ejecución del comando **ls** confirmó que la descarga se realizó de manera exitosa, ya que se visualizan los archivos que componen la herramienta, incluyendo el script principal LinEnum.sh. Esta acción es un paso clave en la fase de **post-explotación**, donde el objetivo es recopilar información detallada del sistema y buscar posibles vulnerabilidades que permitan elevar los privilegios del usuario tiago a un nivel de administrador, o root.

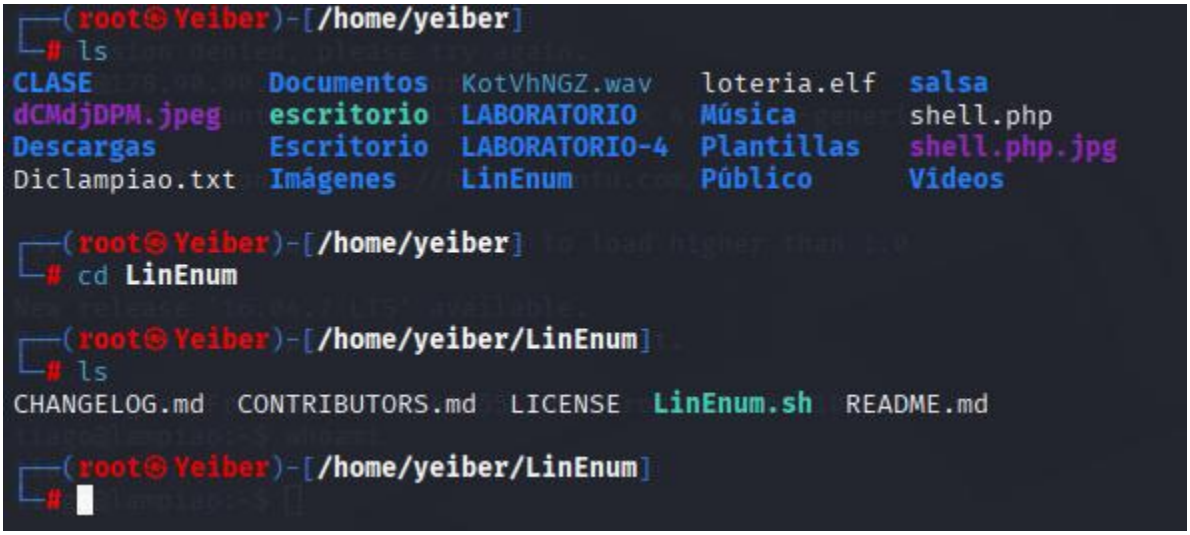

Ilustración 11-Exploración del Sistema y Preparación para la Escalada de Privilegios

3.12 Verificación del entorno de usuario

Al ingresar al servidor, el primer paso fue verificar los archivos y directorios del usuario **tiago**. Se utilizaron los comandos ls y ls -la para listar el contenido del directorio. La salida de estos comandos muestra que:

- El usuario tiago tiene permisos de lectura, escritura y ejecución sobre su propio directorio.
- Se encontraron archivos y carpetas relacionados con el entorno del usuario, como su historial de comandos (.bash_history), configuraciones de sesión (.bash_profile, .bashrc) y caché.
- Se confirma que la herramienta **LinEnum** se clonó exitosamente en la máquina, lo que se evidencia en el directorio LinEnum/.

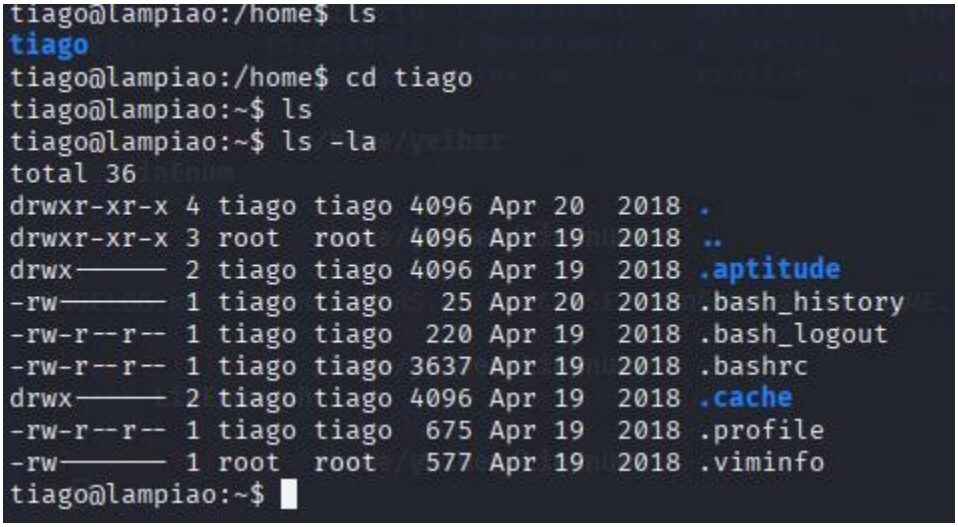

Ilustración 12-Verificación del entorno de usuario

3.13 Análisis del archivo /etc/passwd

Para buscar posibles vulnerabilidades, se accedió al archivo /etc/passwd. Este archivo es clave en los sistemas Linux, ya que contiene una lista de todos los usuarios del sistema, junto con información importante como:

- **Nombre de usuario:** Por ejemplo, tiago, root, mysql.
- **UID (User ID) y GID (Group ID):** Identificadores numéricos que el sistema usa para reconocer a los usuarios y sus grupos.
- **Directorio de inicio:** La ubicación donde el usuario se ubica al iniciar sesión.
- **Shell de inicio de sesión:** El intérprete de comandos que se ejecuta para ese usuario.

Al revisar el contenido de este archivo, se puede buscar si existen usuarios con permisos especiales o configuraciones inusuales que puedan ser explotadas. En la captura, se observa una

lista de usuarios del sistema, incluyendo el usuario tiago (con UID 1000) y el usuario root (con UID 0). La presencia de usuarios de servicio como mysql o lp también es común.

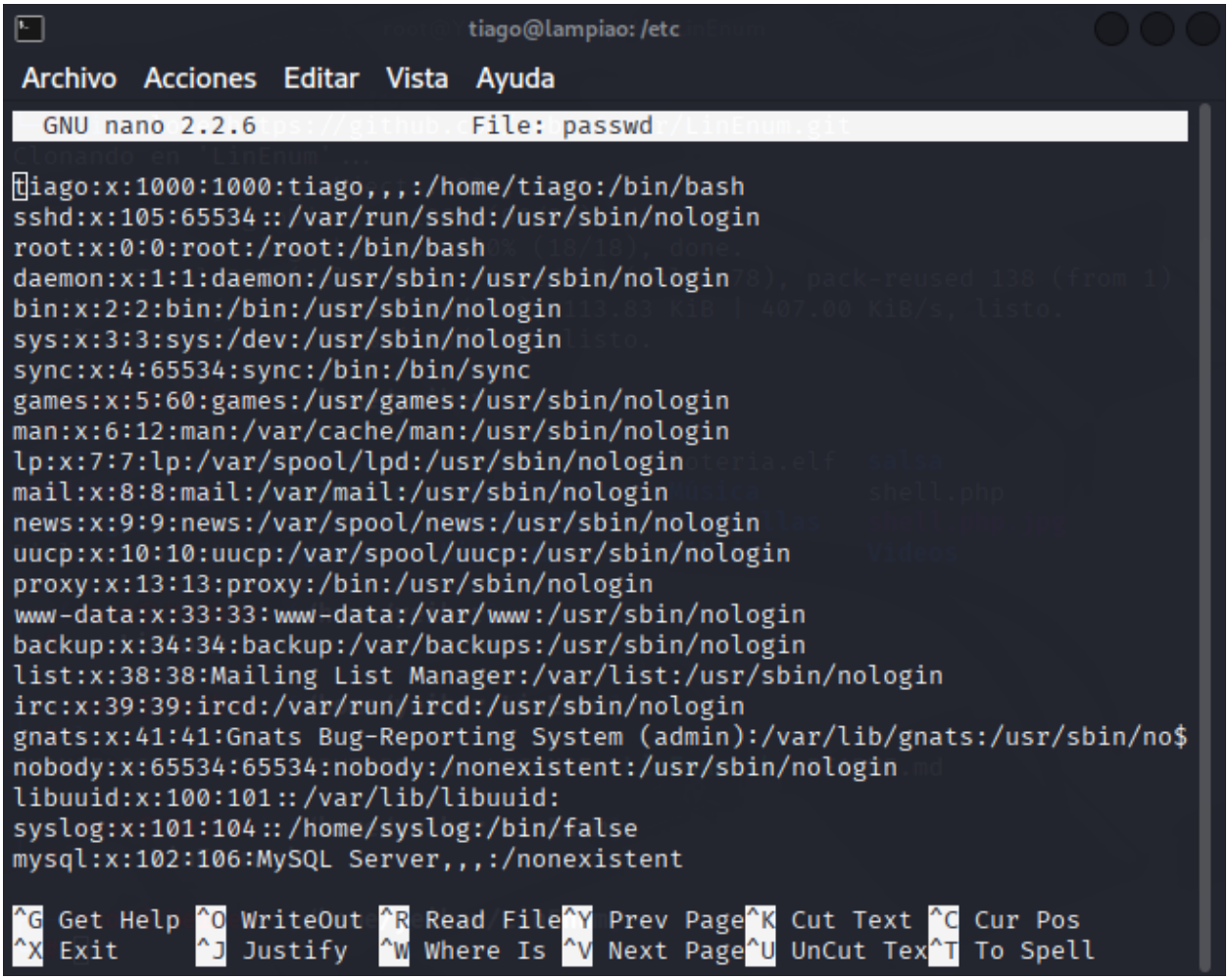

Ilustración 13-Análisis del archivo /etc/passwd

3.14 Transferencia de herramientas para escalada de privilegios

Tras analizar el sistema operativo y el software que corre en el servidor, se procedió a buscar una vulnerabilidad específica que pudiera permitir una escalada de privilegios. En este caso, se

encontró un *exploit* conocido, que es un fragmento de código diseñado para aprovechar un fallo de seguridad.

Los pasos que se realizaron fueron los siguientes:

1. **Descarga del exploit:** Se utilizó el comando wget para descargar el *exploit* desde una base de datos pública (*Exploit-DB*). El archivo descargado se llama **40847.cpp**.
2. **Transferencia del archivo:** Una vez descargado en la maquina, se empleó el comando scp (Secure Copy Protocol) para transferir el archivo **40847.cpp** al directorio del usuario **tiago** en el servidor objetivo.

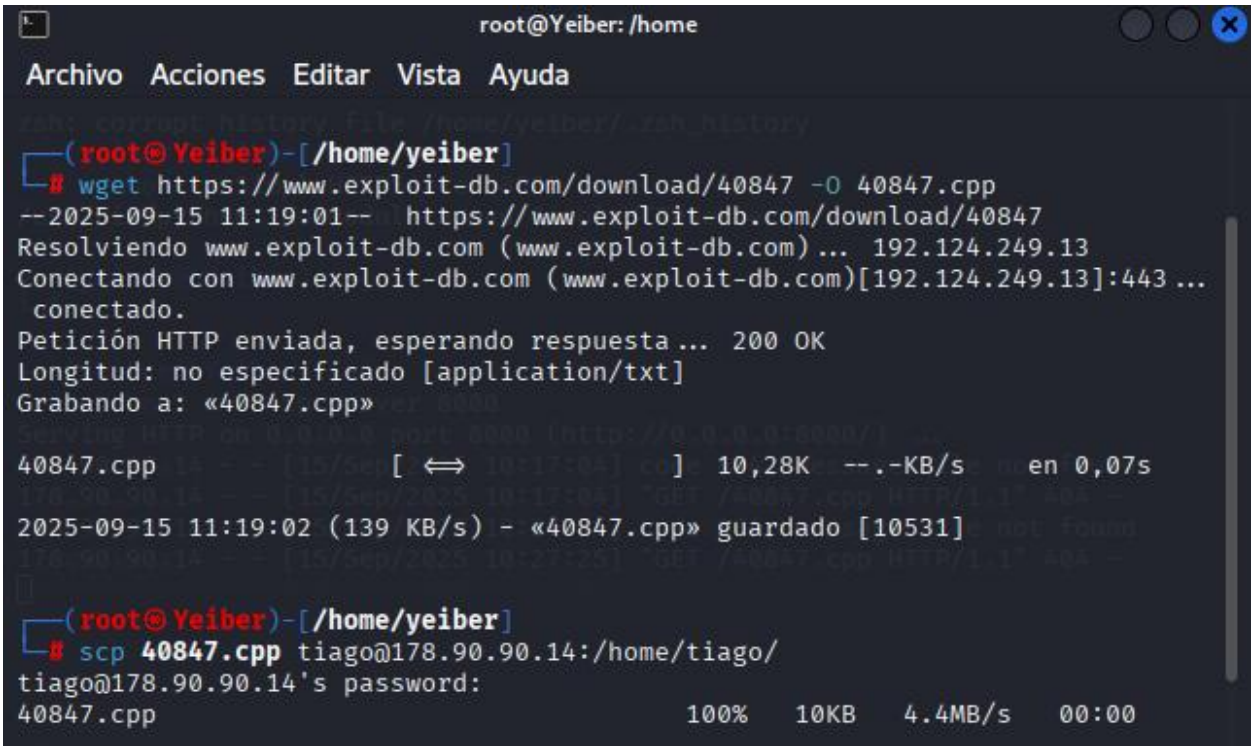

Ilustración 14-Transferencia de herramientas para escalada de privilegios

3.15 Escalada de Privilegios y Obtención de Acceso Root

Una vez transferido el *exploit* al servidor, se procedió a compilar y ejecutar el código para aprovechar la vulnerabilidad. En este caso, el *exploit* **40847.cpp** está relacionado con una vulnerabilidad conocida como **Dirty Cow** (CVE-2016-5195), que permite a un usuario sin privilegios escribir en archivos a los que normalmente no tiene acceso.

Los pasos que se realizaron fueron los siguientes:

1. **Compilación del exploit:** Se utilizó el comando g++ para compilar el código del *exploit*, generando un archivo ejecutable llamado dcow.
2. **Ejecución del exploit:** Se ejecutó el programa dcow, el cual explotó la vulnerabilidad para sobrescribir la contraseña de un usuario del sistema, dándo una forma de obtener privilegios de **root**.
3. **Acceso con privilegios de administrador:** Finalmente, se ejecutó el comando whoami, y el resultado root confirma que se logró una **escalada de privilegios** exitosa. Esto significa que el usuario tiago ahora tiene acceso total al sistema, con los mismos permisos que el administrador.

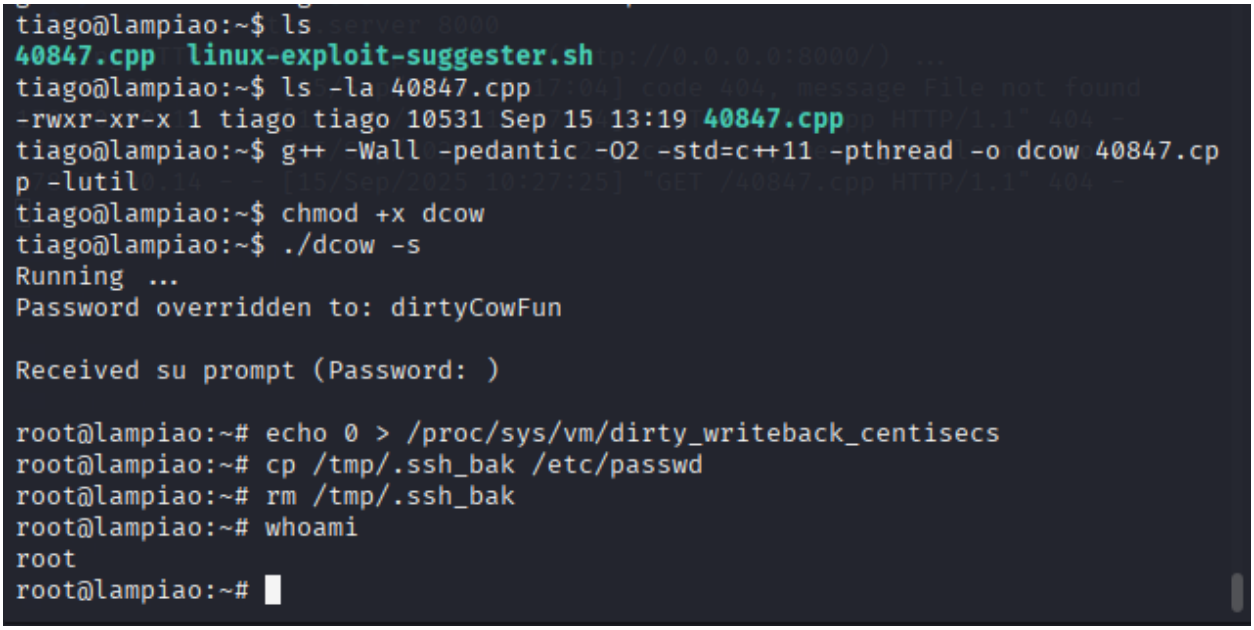

Ilustración 15-Escalada de Privilegios y Obtención de Acceso Root

3.16 Creación de Usuario con Privilegios Elevados

Después de haber escalado privilegios a root (el usuario con control total del sistema), se procedió a crear un nuevo usuario con los mismos permisos. Esto se hace a menudo para asegurar la **persistencia** dentro de un sistema comprometido, ya que, aunque el exploit original sea parchado, el nuevo usuario con privilegios de administrador permitirá mantener el acceso.

Los pasos que se realizaron fueron los siguientes:

1. **Creación de un nuevo usuario:** Se utilizó el comando adduser usuario1. Este comando crea el usuario, su grupo y su directorio de inicio.
2. **Asignación de privilegios de administrador:** Se utilizaron los comandos usermod -aG sudo usuario1 y echo "usuario1 ALL=(ALL:ALL) ALL" >> /etc/sudoers. Estas dos

acciones garantizan que el nuevo usuario, **usuario1**, tenga los mismos permisos que el usuario **root** y que pueda ejecutar comandos de administrador sin necesidad de la contraseña de root.

```
root@lampiao:~# adduser usuario1
Adding user `usuario1' ...
Adding new group `usuario1' (1001) ...
Adding new user `usuario1' (1001) with group `usuario1' ...
Creating home directory `/home/usuario1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
Sorry, passwords do not match
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for usuario1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@lampiao:~# usermod -aG sudo usuario1
root@lampiao:~# usermod -aG root usuario1
root@lampiao:~# echo "usuario1 ALL=(ALL:ALL) ALL" >> /etc/sudoers
root@lampiao:~#
```

Ilustración 16-Creación de Usuario con Privilegios Elevados

Una vez creado el usuario1, repetimos la misma secuencia de comandos para crear el usuario2 y asignarle los mismos privilegios como administrador

```
root@lampiao:~# adduser usuario2
Adding user `usuario2' ...
Adding new group `usuario2' (1002) ...
Adding new user `usuario2' (1002) with group `usuario2' ...
Creating home directory `/home/usuario2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for usuario2
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@lampiao:~# usermod -aG sudo usuario2
root@lampiao:~# usermod -aG root usuario2
root@lampiao:~# echo "usuario2 ALL=(ALL:ALL) ALL" >> /etc/sudoers
root@lampiao:~#
```

Ilustración 17-Craeción de usuario2

3.17 Verificación de Usuarios con Privilegios Elevados

Para validar que las acciones de persistencia se realizaron con éxito, se procedió a verificar la creación y los privilegios de los nuevos usuarios. Se utilizaron los siguientes comandos:

- **cat /etc/passwd | grep usuario**: Este comando buscó en el archivo de usuarios del sistema a las cuentas usuario1 y usuario2. La salida confirmó su existencia y que tienen un *shell* de inicio de sesión activo (/bin/bash), lo que les permite acceder al sistema de forma remota.
- **grep sudo /etc/group**: Este comando buscó el grupo **sudo**, que es el que otorga permisos de administrador. La salida mostró que tanto **usuario1** como **usuario2** son miembros de este grupo.

Esta verificación confirma que el control total sobre el servidor ha sido consolidado de manera efectiva, ya que se crearon cuentas con permisos de administrador que pueden ser utilizadas para mantener el acceso al sistema

```
root@lampiao:~#
root@lampiao:~# cat /etc/passwd | grep usuario
usuario1:x:1001:1001:,,,:/home/usuario1:/bin/bash
usuario2:x:1002:1002:,,,:/home/usuario2:/bin/bash
root@lampiao:~# grep sudo /etc/group
sudo:x:27:usuario1,usuario2
root@lampiao:~#
```

Ilustración 18-Verificación de Usuarios con Privilegios Elevados

3.18 Ajuste de Permisos de Directorio

Como parte del proceso de consolidación, se ejecutaron los comandos **chmod 755 /home/usuario1** y **chmod 755 /home/usuario2**.

La finalidad de esta acción es asegurar que los directorios de los nuevos usuarios creados (usuario1 y usuario2) tengan permisos de acceso correctos. Específicamente, el número 755 significa que el dueño del directorio (en este caso, cada usuario) tiene control total (7 o rwx), mientras que los miembros del mismo grupo y el resto de los usuarios solo pueden ver el contenido y entrar (5 o r-x).

```
root@lampiao:~#
root@lampiao:~# chmod 755 /home/usuario1
root@lampiao:~# chmod 755 /home/usuario2
root@lampiao:~#
```

Ilustración 19-Ajuste de Permisos de Directorio

3.19 Prueba de acceso con las nuevas credenciales

Una vez que se crearon los nuevos usuarios con privilegios de administrador, se verificó si también podían iniciar sesión en el sitio web alojado en el puerto 1898. El intento de acceso con el usuario **usuario2** no fue exitoso.

Este resultado, aunque parece un fallo, es un hallazgo importante. Demuestra que el sistema de autenticación del sitio web (el CMS) y la gestión de usuarios del servidor (Linux) no están vinculados. Los usuarios creados a nivel del sistema operativo no se replican automáticamente en la base de datos de la página web. Esto subraya la importancia de analizar por separado cada servicio, ya que sus vulnerabilidades y configuraciones son independientes.

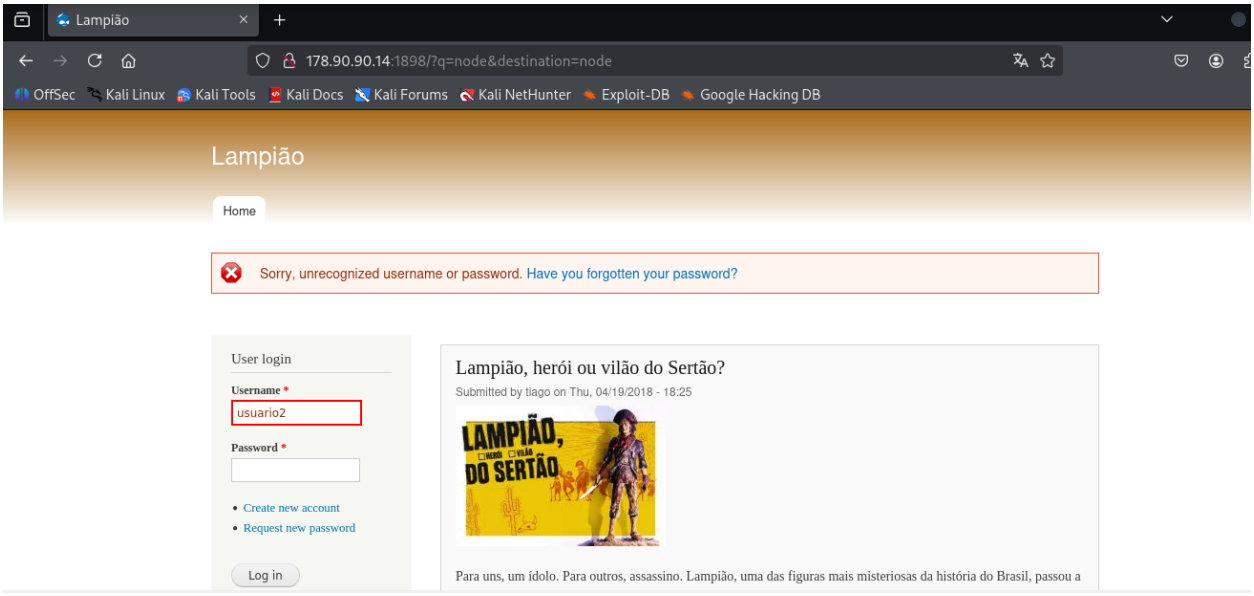

Ilustración 20-Prueba de acceso con las nuevas credenciales

3.20 Acceso y Modificación de la Base de Datos

Luego de obtener control total del sistema, se logró acceder a la base de datos MySQL que gestiona el sitio web "Lampião". Esta acción permite manipular directamente los datos del portal, incluyendo la tabla de usuarios.

En la captura se evidencian dos acciones clave:

1. **Verificación de la base de datos:** Se ejecutó un comando SQL para confirmar que la base de datos del sitio web estaba activa.
2. **Creación de usuarios a nivel del sitio web:** Se insertaron nuevos usuarios, usuario1 y usuario2, directamente en la tabla de usuarios de la base de datos. Esto es diferente a la creación de usuarios a nivel del sistema operativo. Al hacer esto, los nuevos usuarios tienen credenciales válidas para iniciar sesión en la página web, tal como si se hubieran registrado de forma normal.

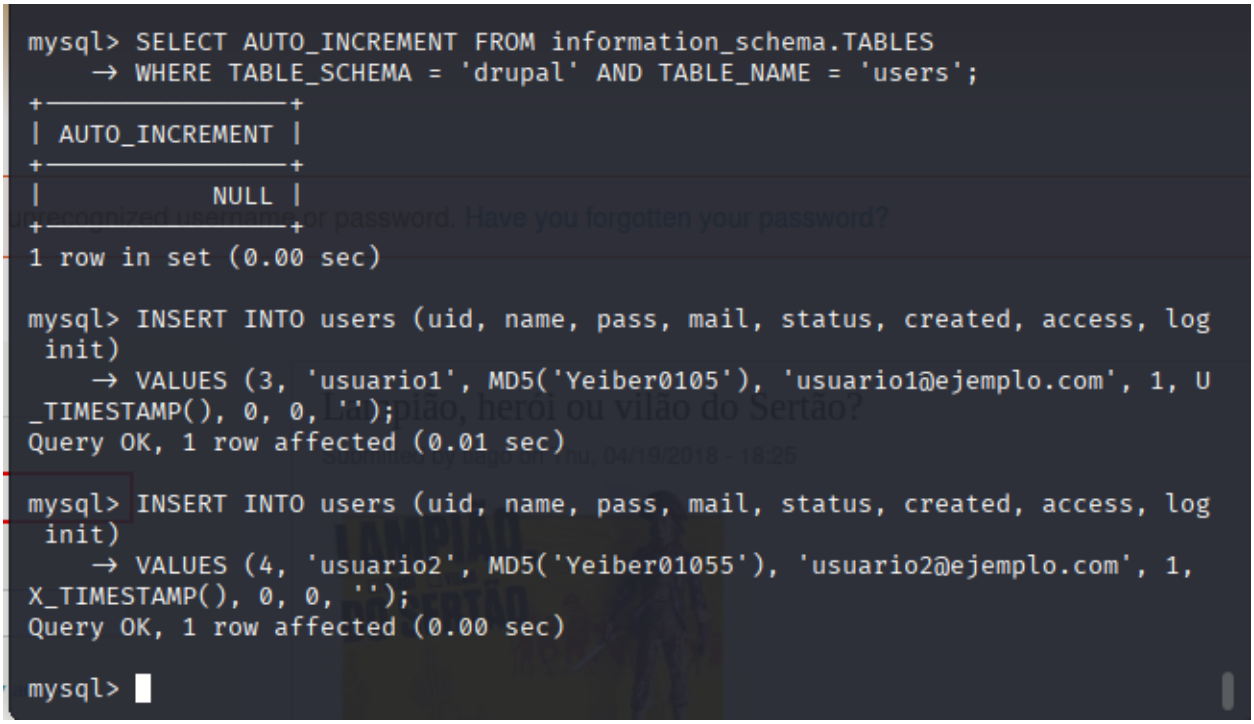

Ilustración 21-Acceso y Modificación de la Base de Datos

3.21 Verificación Final de la Base de Datos

Para verificar que la inyección de usuarios en la base de datos fue exitosa, se ejecutó un comando SQL para mostrar la tabla de usuarios del sitio web. La consulta SELECT uid, name, mail, status FROM users ORDER BY uid; reveló lo siguiente:

- **Usuarios existentes:** La tabla contiene al usuario original **tiago** y otros, además de los dos usuarios que se insertaron manualmente: **usuario1** y **usuario2**.
- **Confirmación de los datos:** Se puede ver claramente que los nuevos usuarios tienen un ID (uid), un nombre de usuario (name) y un correo electrónico (mail), lo que confirma que las entradas se crearon correctamente en la base de datos de Drupal.

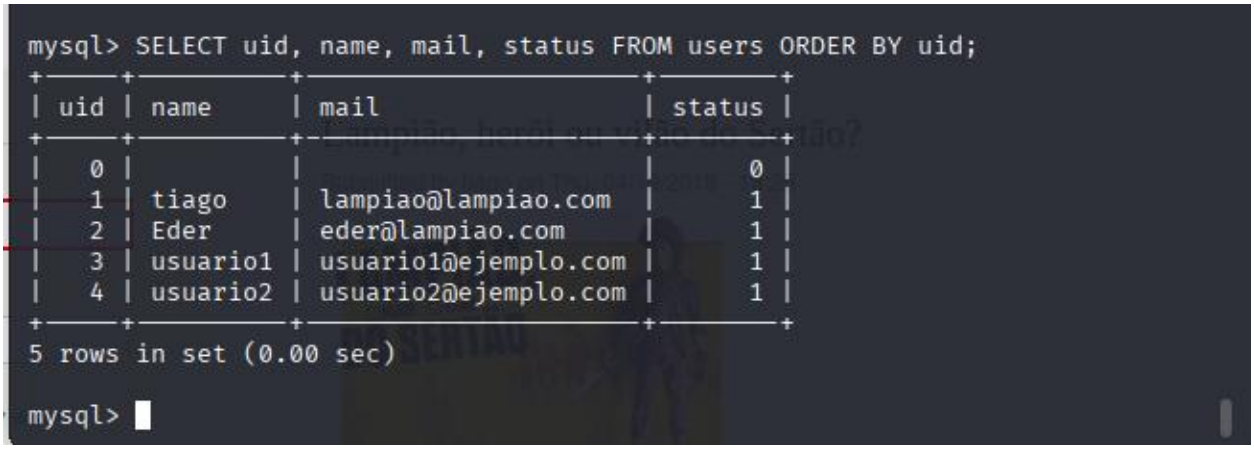

Ilustración 22-Verificación Final de la Base de Datos

3.22 Asignación de Roles de Administrador en la Base de Datos

Para confirmar que los nuevos usuarios creados (usuario1 y usuario2) tuvieran los máximos privilegios en el sitio web, se modificó la base de datos para asignarles el rol de **administrador**.

En la captura, se ejecutaron dos comandos SQL:

1. **INSERT INTO users_roles**: Se insertaron los ID de los usuarios (3 para usuario1 y 4 para usuario2) y el ID del rol de administrador (3) en la tabla users_roles, que es la que vincula a los usuarios con sus privilegios.

2. **SELECT u.uid, u.name, r.name as role_name**: Esta consulta final verificó que la acción anterior se realizó correctamente. La tabla de resultados muestra a usuario1 y usuario2 con el rol **administrator**, lo que significa que tienen control total sobre el sitio web, incluyendo la capacidad de crear contenido, gestionar usuarios y cambiar la configuración.

Este es el último eslabón de la cadena de análisis. Se demostró que al obtener acceso al servidor, se puede manipular la base de datos para obtener privilegios de administrador no solo en el sistema operativo, sino también en la aplicación web que el servidor aloja.

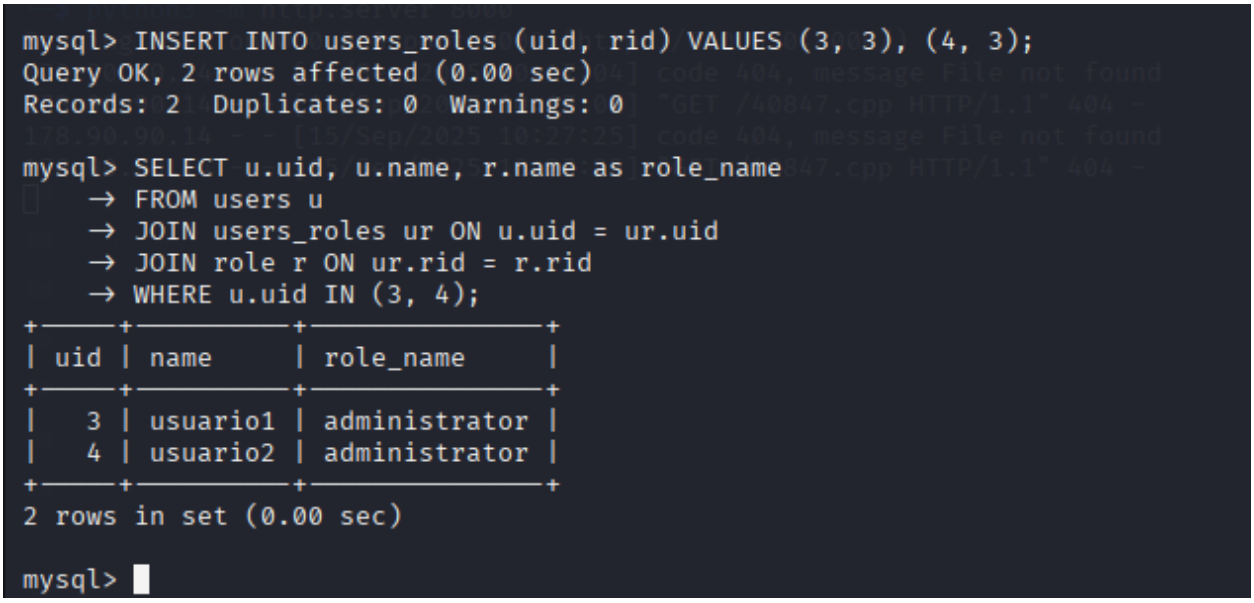

Ilustración 23-Asignación de Roles de Administrador en la Base de Datos

3.23 Verificación de Acceso con el Nuevo Usuario

Como último paso, se probó la conexión al servidor SSH utilizando la cuenta **usuario2** creada durante la fase de post-explotación.

- La captura muestra un inicio de sesión exitoso. El mensaje "Welcome to Ubuntu 14.04.5 LTS" confirma que el usuario se ha conectado al servidor.
- Esto valida la capacidad del investigador para mantener acceso al sistema con una cuenta diferente a la original, una prueba clave para asegurar la persistencia.

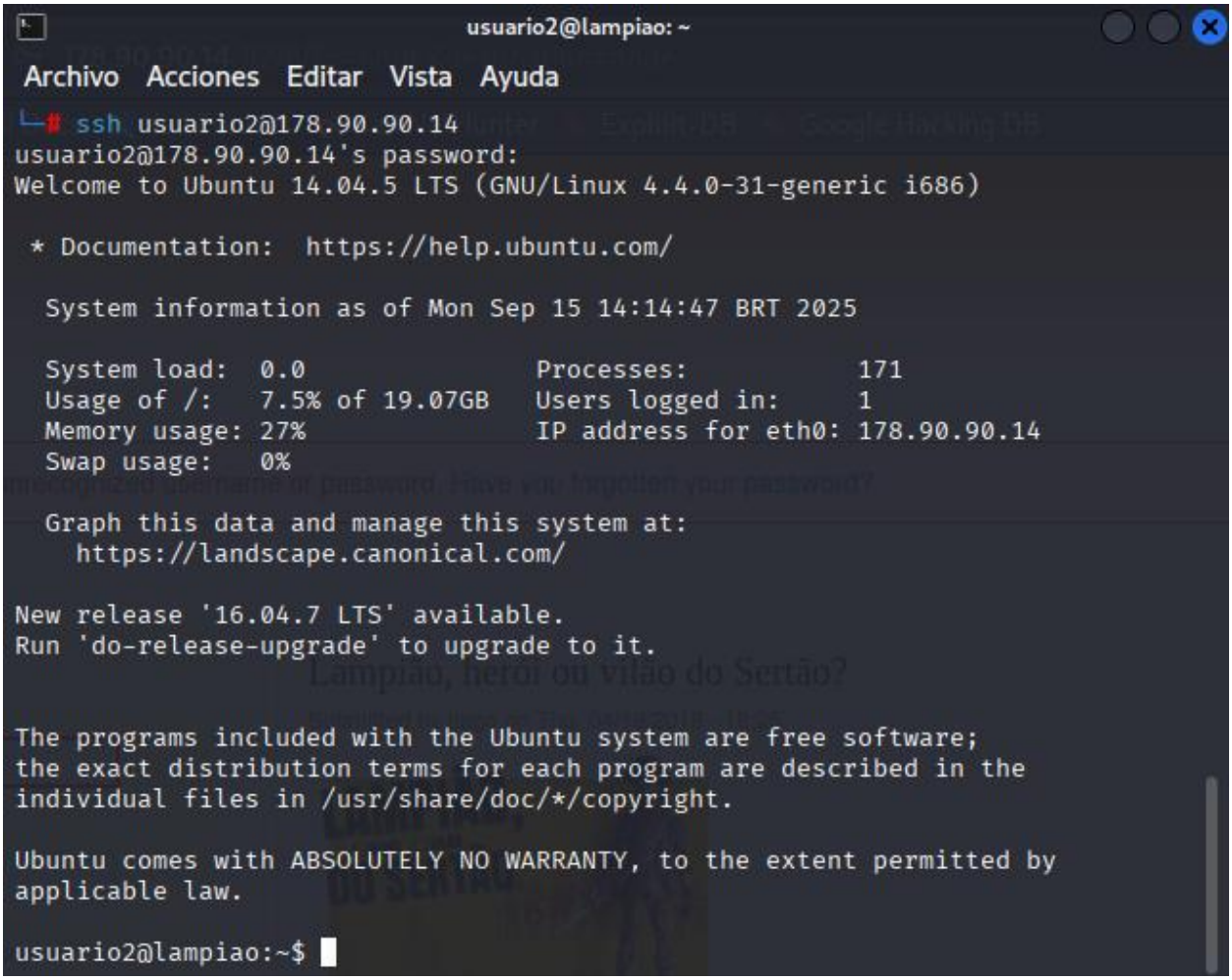

Ilustración 24-Verificación de Acceso con el Nuevo Usuario

4. CONCLUSIÓN

La realización de este laboratorio permitió comprender de forma práctica cómo un sistema informático puede ser analizado y puesto a prueba para detectar posibles fallas de seguridad. A través de cada etapa se evidenció la importancia de proteger los servidores, mantenerlos actualizados y utilizar contraseñas seguras para evitar accesos no autorizados. Esta experiencia demuestra que el *hacking ético* no busca dañar, sino aprender y prevenir, ayudando a que las organizaciones refuercen sus medidas de protección y garanticen la seguridad de su información.

5. REFERENCIAS

- CVE-2016-5195 – Dirty Cow Vulnerability. *MITRE Corporation*. Disponible en: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-5195
- Nmap Network Scanning. *Nmap Official Documentation*. Disponible en: https://nmap.org/book/
- Hydra – Password Cracking Tool. *Kali Linux Tools*. Disponible en: https://tools.kali.org/password-attacks/hydra
- LinEnum – Linux Privilege Escalation Script. *GitHub Repository*. Disponible en: https://github.com/rebootuser/LinEnum