Universidad Tecnológica del Chocó
Diego Luis Córdoba

PROGRAMA DE INGENIERÍA DE TELECOMUNICACIONES E INFORMÁTICA.

SEGURIDAD Y AUDITORIA DE REDES

LABORATORIO #4

LUIS JADER CUESTA MOYA
Estudiante**.**

SEMESTRE:
VIII

QUIBDÓ-CHOCÓ

Fecha: 09/09/2025

CONTENIDO

INTRODUCCIÓN

La seguridad informática es uno de los pilares fundamentales en el desarrollo y administración de sistemas, dado que la creciente interconexión digital expone constantemente a organizaciones e individuos a múltiples amenazas cibernéticas. En este contexto, el laboratorio realizado con Kali Linux y la máquina virtual FristiLeaks se convierte en una práctica esencial para comprender cómo se identifican, explotan y mitigan vulnerabilidades en entornos controlados. A través de la simulación de ataques, como el escaneo de directorios, la decodificación de información oculta y la explotación de formularios de subida de archivos, se puede evidenciar la importancia de fortalecer la infraestructura tecnológica frente a ciberataques. Este trabajo no solo permite afianzar conocimientos técnicos en auditoría de redes y seguridad informática, sino que también resalta el papel crítico de la ética profesional en el manejo de dichas herramientas.

OBJETIVOS

- **Objetivo general:**

Analizar y aplicar técnicas de auditoría de seguridad mediante la explotación de vulnerabilidades en la máquina virtual FristiLeaks utilizando Kali Linux.

- **Objetivos específicos:**

Configurar adecuadamente el entorno virtual con VirtualBox y establecer la conectividad de red necesaria para la práctica.

Identificar directorios, rutas ocultas y posibles fallos de seguridad a través de escaneos de fuerza bruta y análisis de archivos.

Ejecutar pruebas de subida de archivos maliciosos con el fin de obtener acceso al servidor víctima.

Evaluar los resultados de la intrusión, validando el impacto que tendría una vulnerabilidad de este tipo en un entorno real.

Reflexionar sobre las medidas de prevención y buenas prácticas que deberían implementarse para mitigar los riesgos detectados.

TABLA DE ILUSTRACIONES

1. EXPLOTACIÓN DE VULNERABILIDADES EN FRISTILEAKS MEDIANTE KALI LINUX

Para este laboratorio necesitaba el sistema operativo **fristileaks** y lo pude conseguir en el salón de clases gracias a que lo suministro el docente quien es nuestro tutor en lo que respecta de la asignatura, ya que contamos con el virtualbox instalado lo siguiente es ejecutar el archivo **.ova** (ejecutable), con el virtualbox que es como yo lo ejecuto inicialmente.

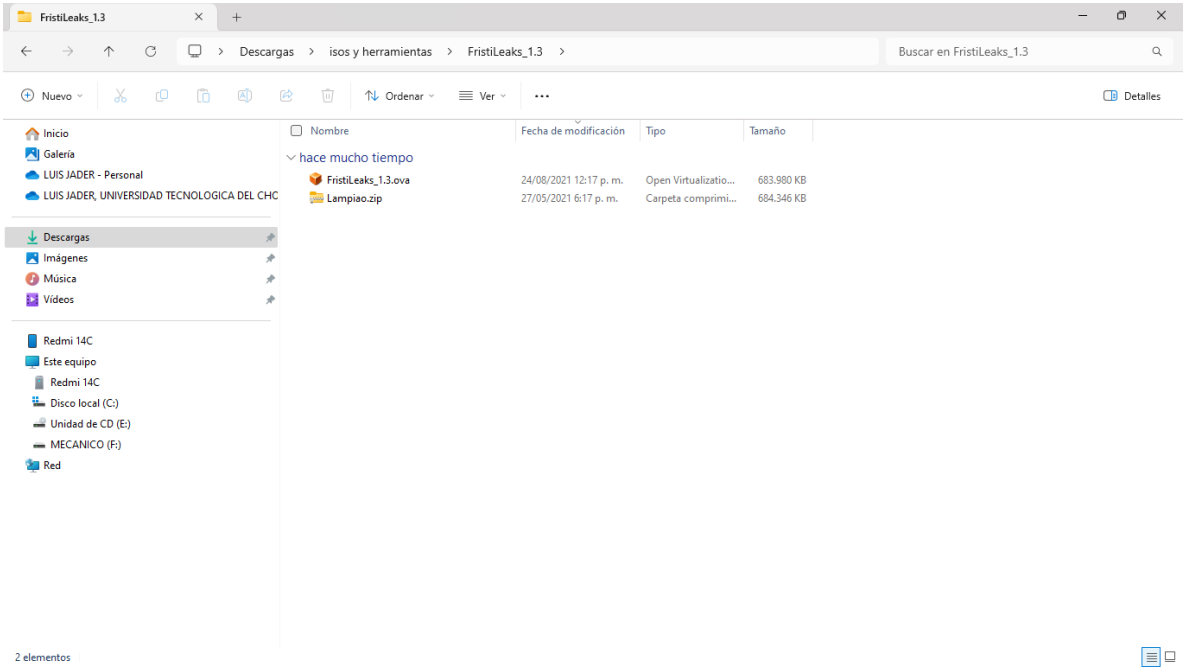


Ilustración 1 archivo ejecutable fristileaks

Nos resulta mucho más fácil montar estas máquinas cuando les indicamos como queremos que se ejecute.

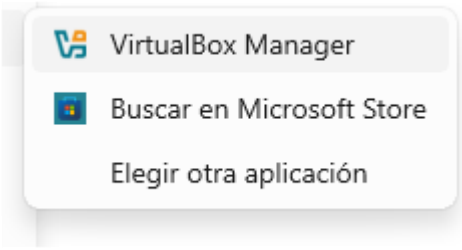


Ilustración 2 primer paso para la ejecucion

Luego de iniciar a ejecutar la instalación en nuestra máquina virtual nos aparecerá una ventana que la podemos llamar el comando de instalación inicial donde le vamos a dar algunos ajustes a nuestro sistema en la máquina virtual, también

podríamos cambiar la ubicación del archivo desde donde se tomara el archivo raíz para la instalación.

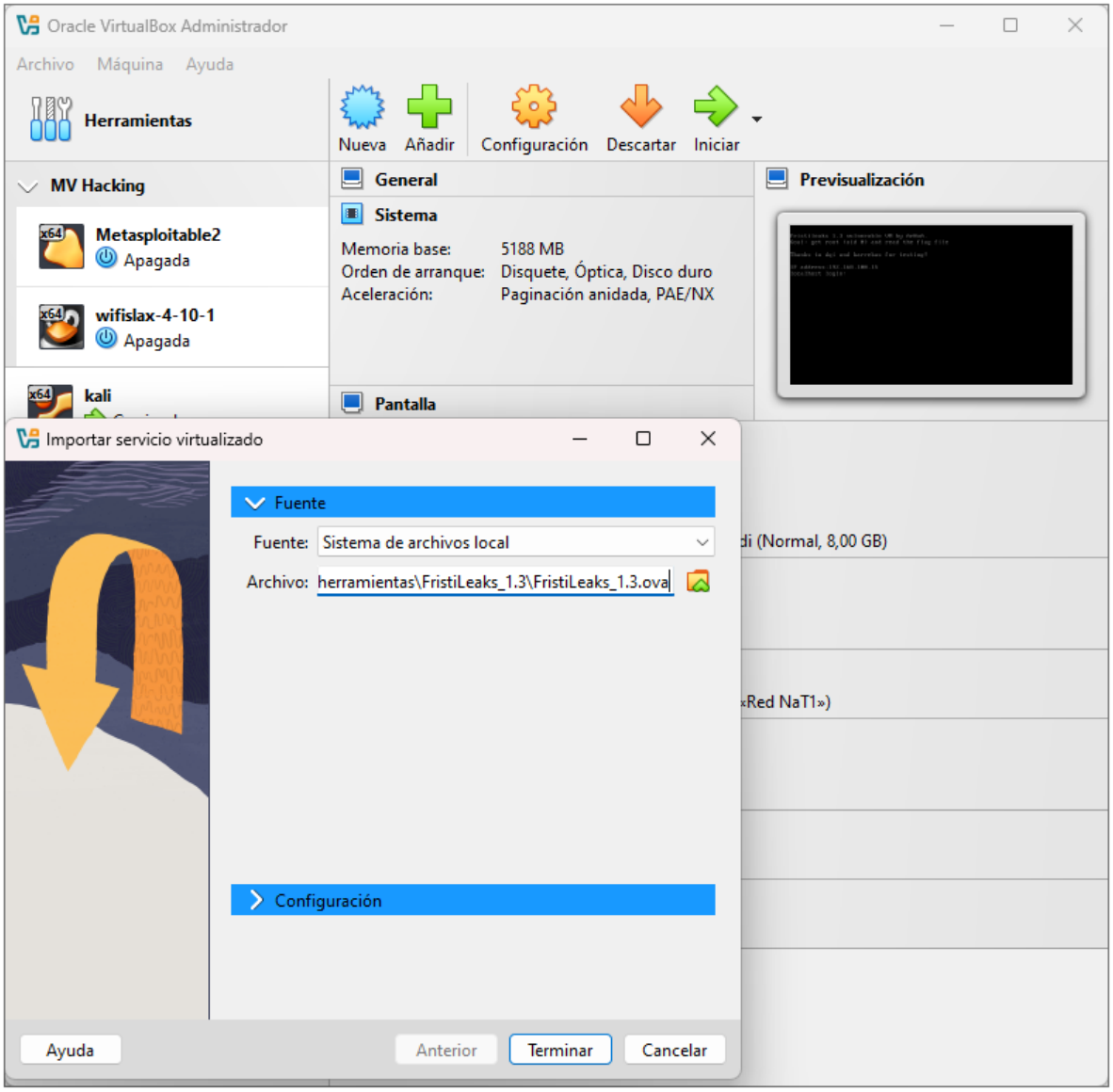


Ilustración 3 administrador de instalación.

Este vendría siendo el administrador de configuración que se puede ver luego de la instalación luego de que ya el sistema esté en la máquina virtual central.

Aquí podríamos hacer parte de las configuraciones que hicimos a la hora de instalar como quien dice modificar la mayoría de sus configuraciones, como el nombre, darle bidireccionalidad para poder interactuar más fácil entre las diferentes maquinas!

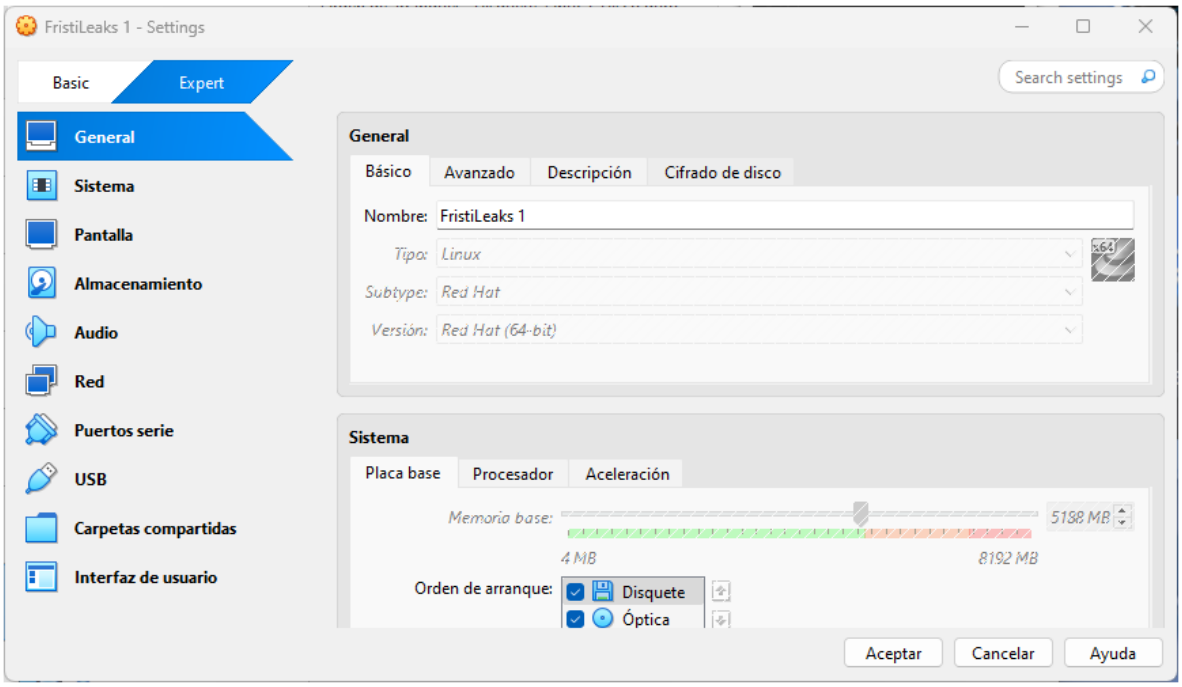


Ilustración 4 panel de configuración

Otra de las configuraciones que podemos hacer es la dirección MAC: esta vez utilizamos una dirección estática para este sistema y corresponde a la dirección **080027A5A676** como lo podemos apreciar el anexo.

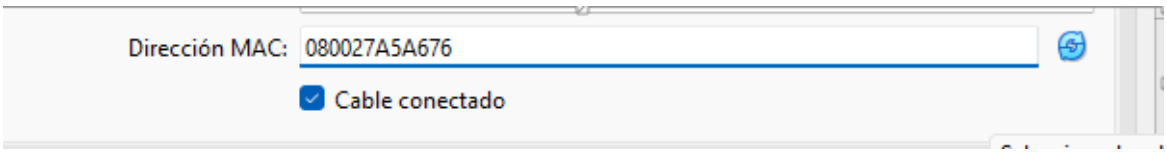
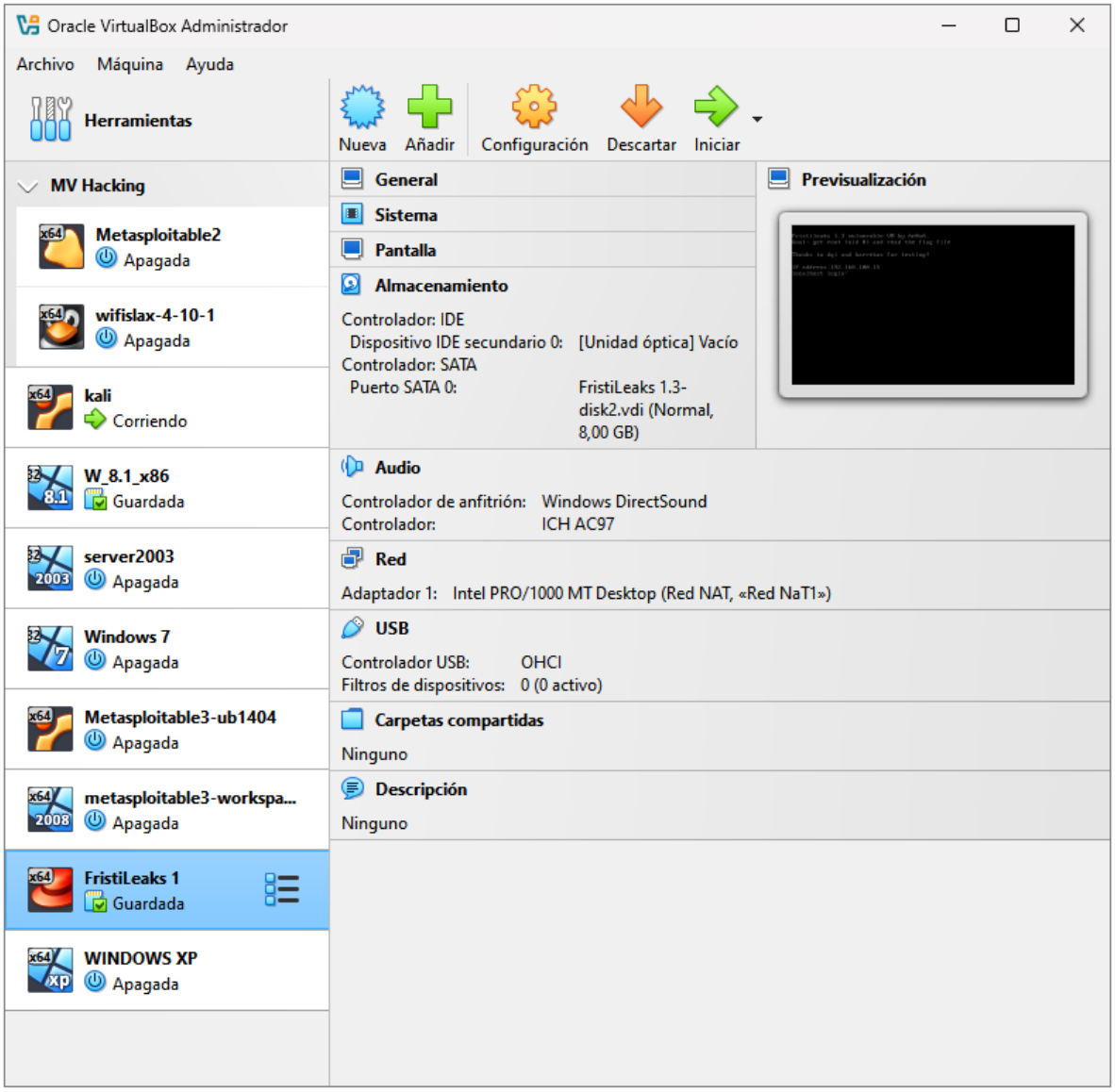


Ilustración 5 configuracion MAC.

Estas son las maquinas o parte de las cuales estamos utilizando o vamos a utilizar en todo el desarrollo de esta asignatura y cómo podemos ver ya existe nuestra máquina virtual **FristiLeaks,** que en esta oportunidad jugara un papel como maquina victima.



Esta es la configuración de la red Nat. La cual está creada y configurada en nuestro virtualbox para posteriormente ser configurada en cada una de las máquinas virtuales si así lo requiere la temática del docente.

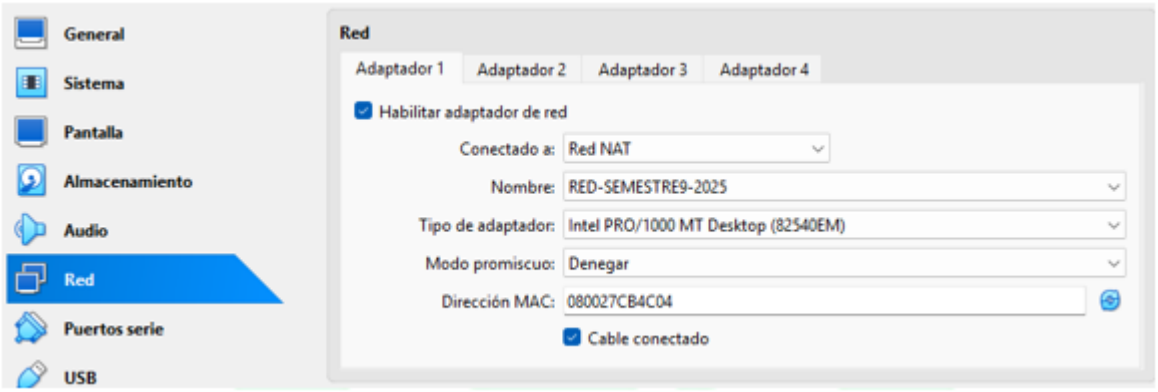


Ilustración 6 Configuracion Red Nat en fristileaks.

La principal configuración para la red Nat nace aquí en esta interfaz donde se coloca la dirección Ip central o puerta de enlace que en este caso vendría siendo **192.168.100.0/24** con mascara 24.

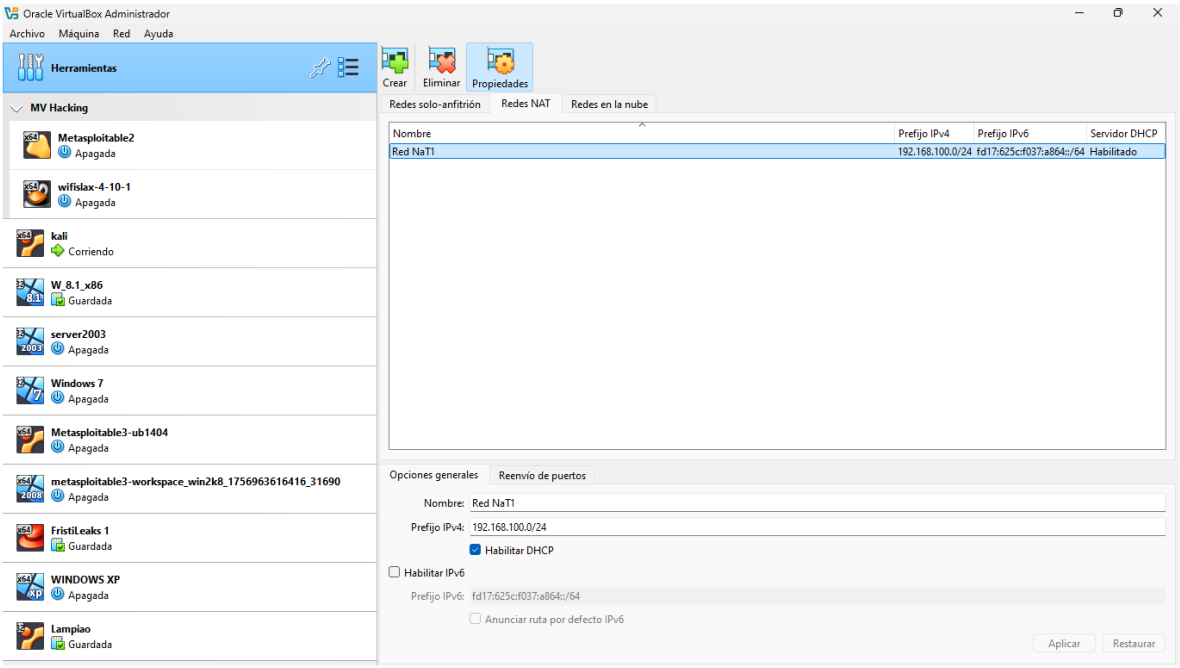


Ilustración 7 Creación de la red Nat en virtualbox.

Antes de iniciar a hacer ataques lo primero que hacemos es saber la dirección que estaba rentando la maquina víctima, debido a que siempre debemos tener algo para poder iniciar en este casa la dirección es la 192.168.100.15.

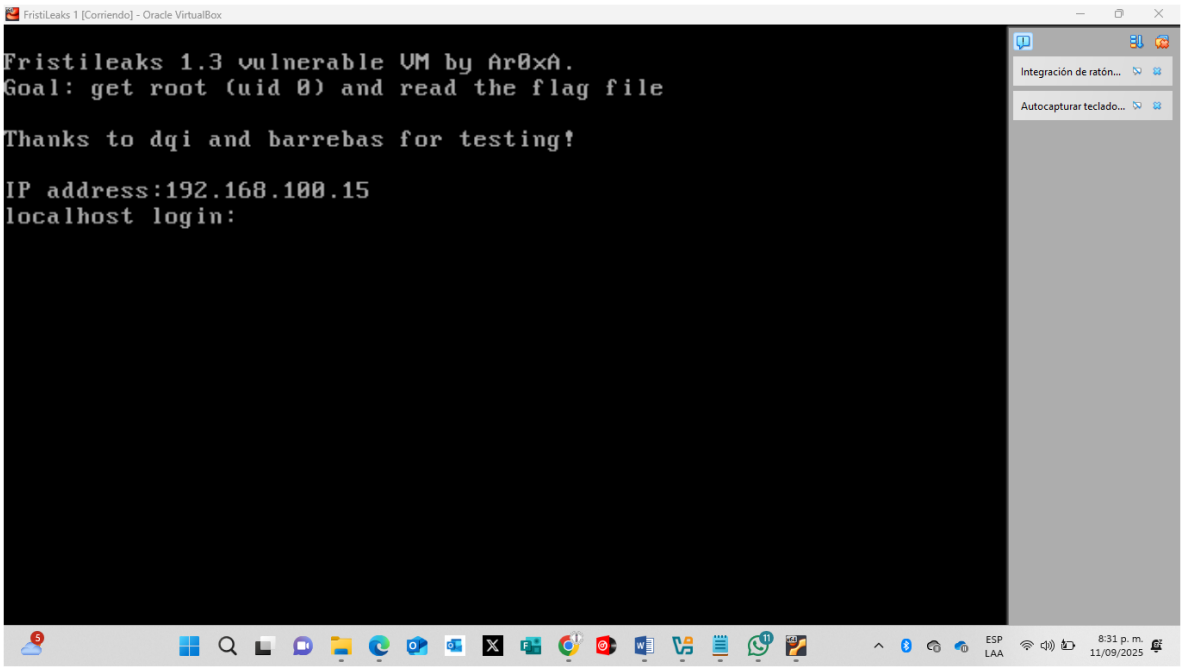


Ilustración 8 interfaz inicial fristileaks

Cuando iniciamos el laboratorio lo segundo que hicimos fue un escaneo de fuerza bruta con el comando dirb http://192.168.100.15 con la dirección de la maquina victima para ver que directorios o directorios que no estén enlazados en la pagina principal.

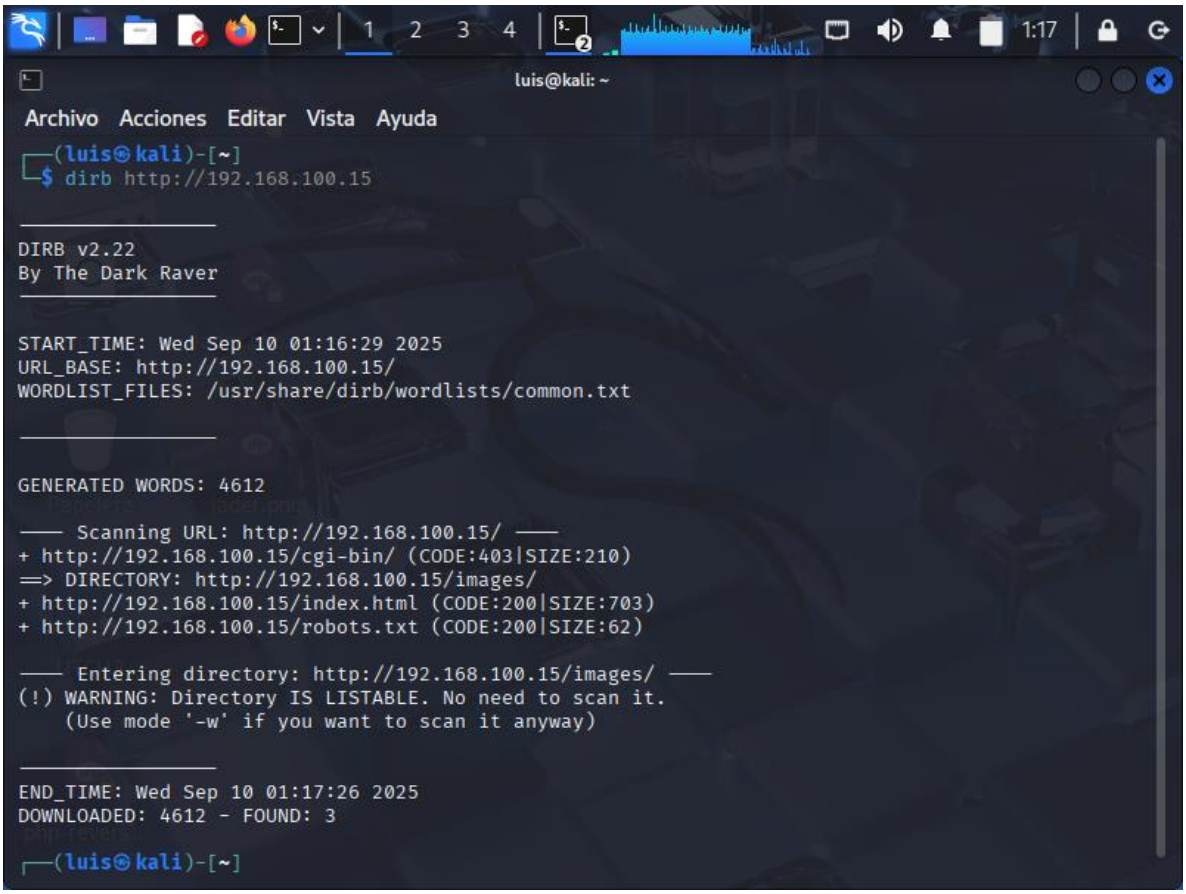


Ilustración 9 escaneo de fuerza bruta.

Se utiliza este comando http://192.168.100.15/images/ para mostrar el contenido de este sitio en el servidor debe existir alguna información que nos sirva como pie para iniciar y atacar los puntos de entrada.
Vemos que el servidor web tiene habilitado el listado de directorios en otras palabras podríamos decir que no sería tan necesario utilizar fuerza bruta

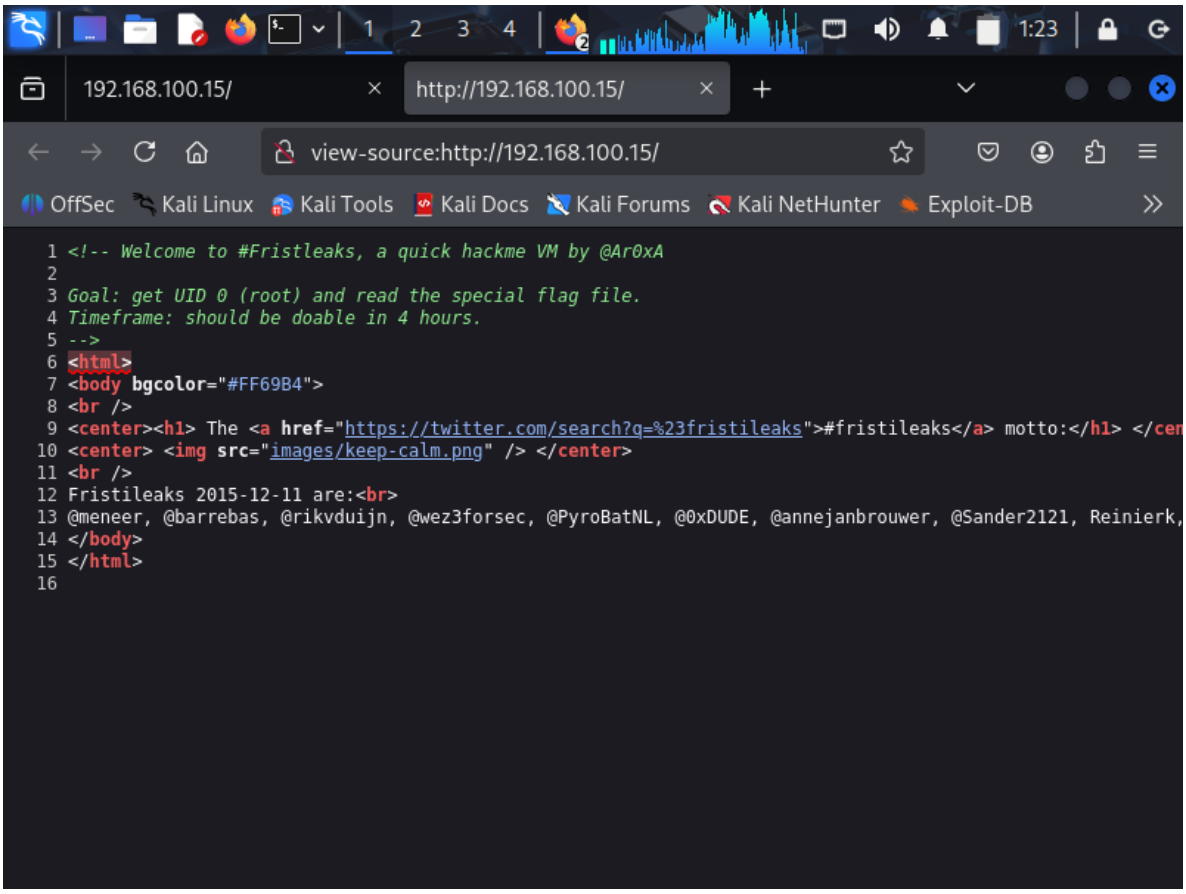


Ilustración 10 buscar Listado de directorios

Aquí se muestran los diccionarios disponibles estos son los diccionarios que se pueden usar en las explotaciones más avanzadas.
Y procedemos a cambiar de directorio hacia la carpeta donde se encuentran los diccionarios con el comando /usr/share/dirb/wordlists/

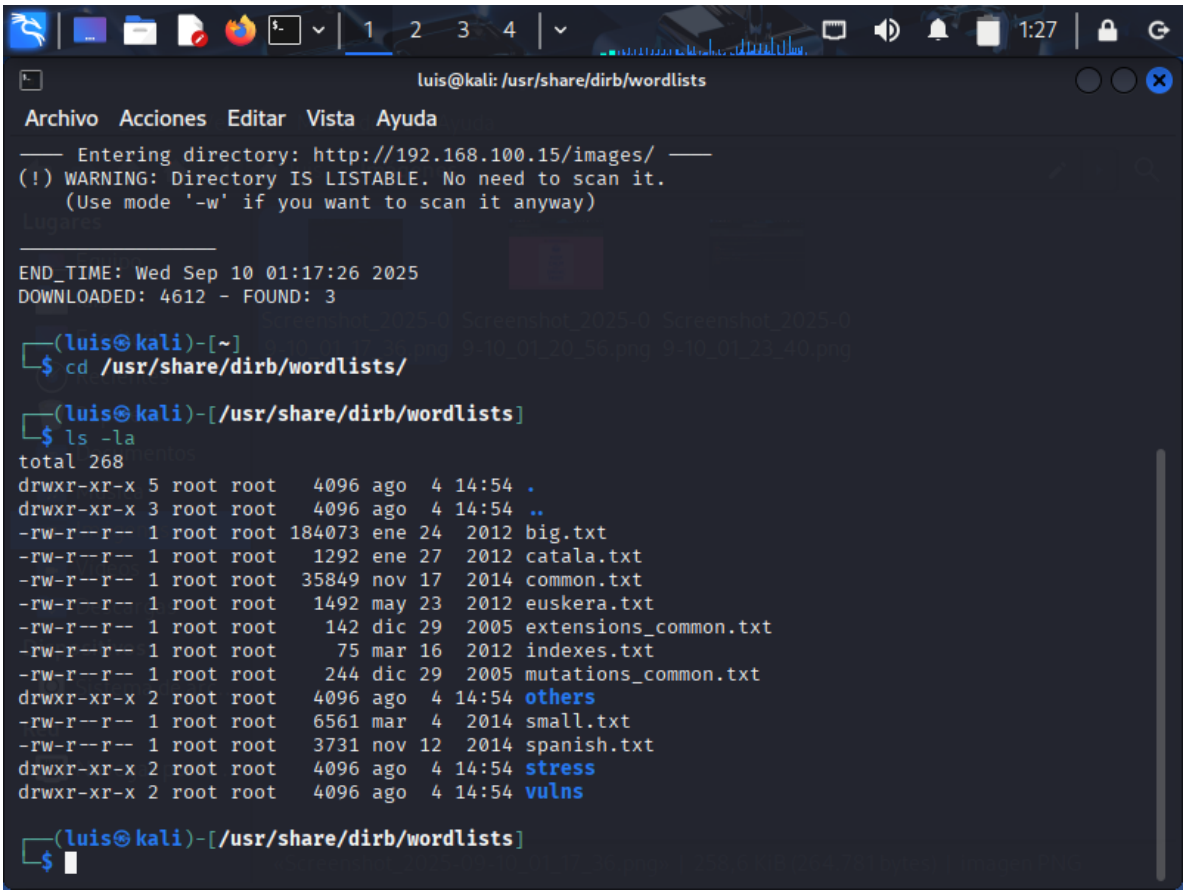


Ilustración 11 lista de diccionarios

Escaneamos la URL que rento nuestra maquina victima para usar el diccionario small.txt que contiene cerca de 1000 palabras por tanto es ligero y menos pesado

para hacer un escaneo rápido. El directorio **cgi-bin** suele usarse para scripts ejecutables, ósea que podrían haber vulnerabilidades

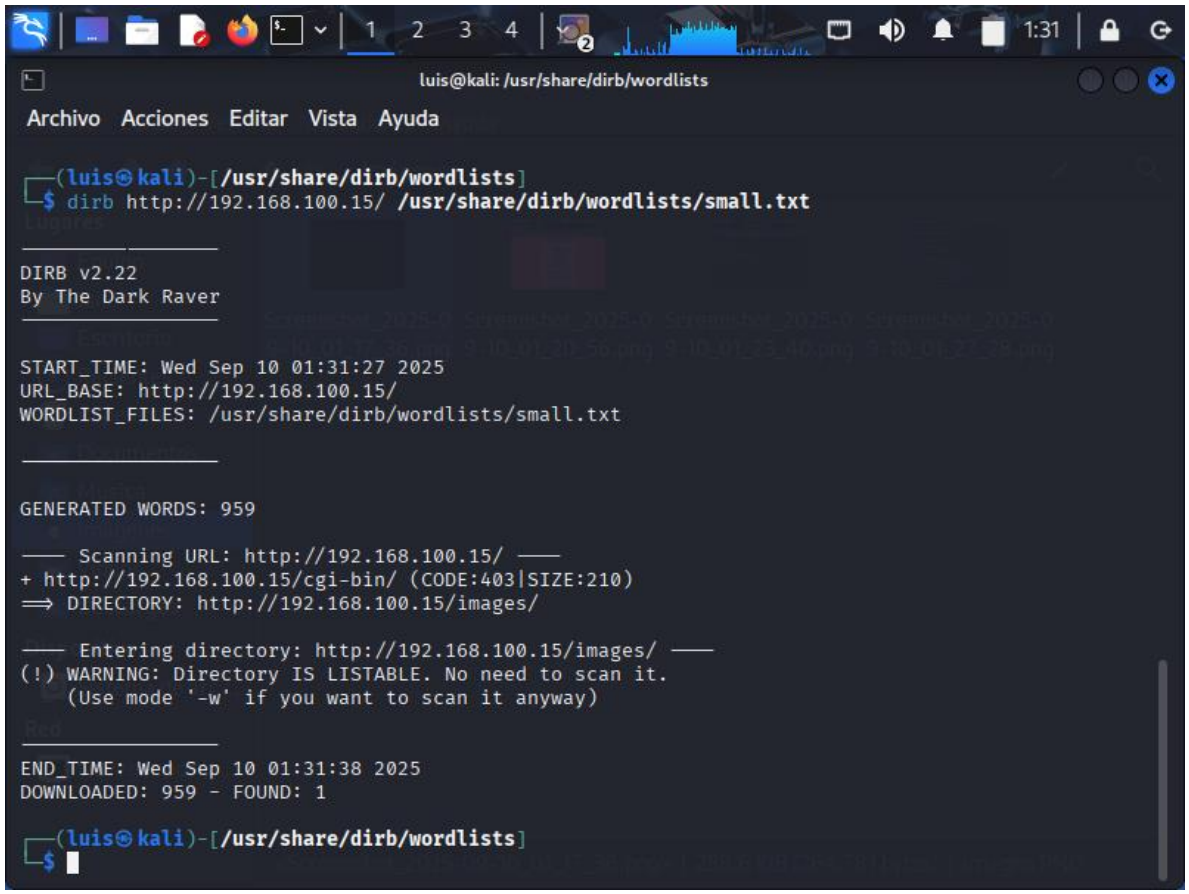


Ilustración 12 Escaneo de directorios

Podemos apreciar que el administrador no protegió estas rutas, lo que nos permite tener más facilidad de tener pistas positivas ya que **robots.txt** revelo rutas sensibles enesta ocacion.

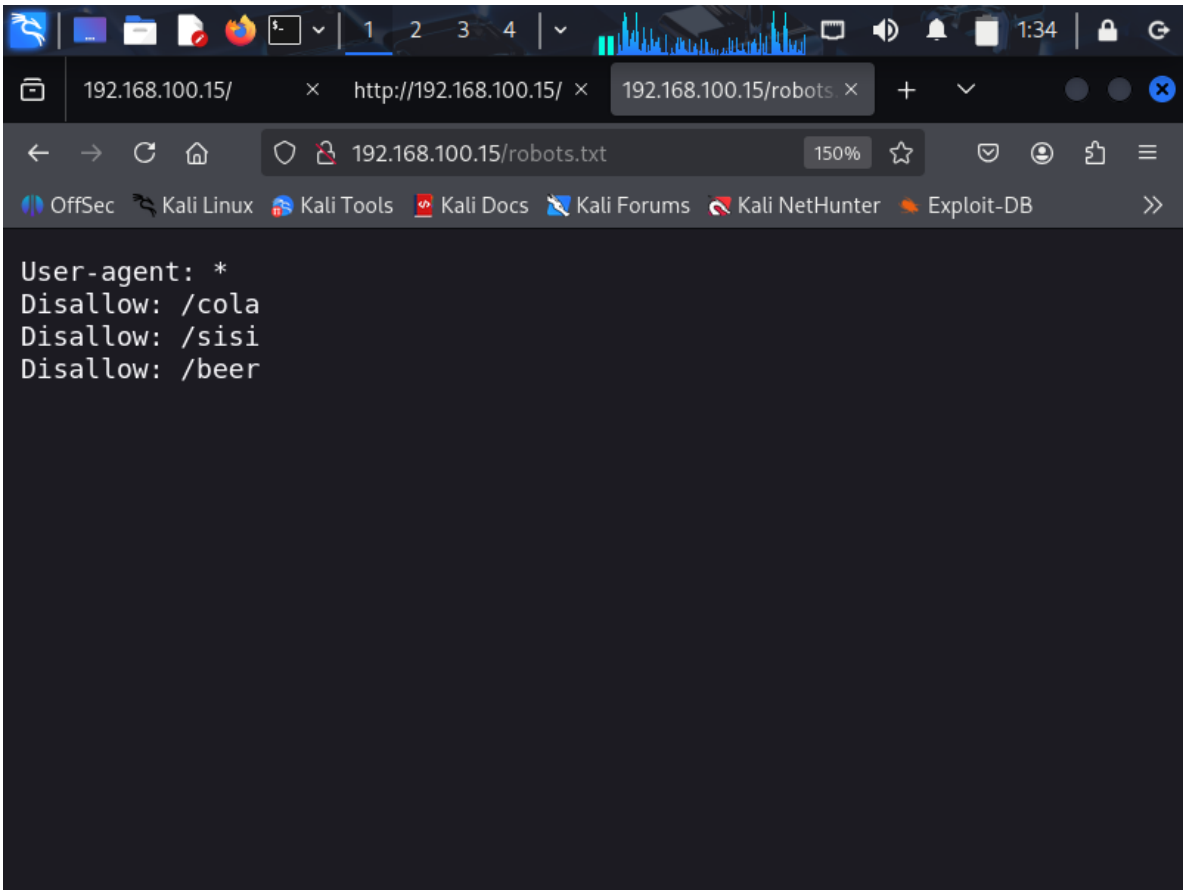


Ilustración 13 buscador de rutas robots.txt

Esto es la página **fristi** de la arrojada con la dirección ip base de **Fristileaks**, y muestra un portal de administración con formulario de login, Un portal así es un objetivo clásico en auditoría ya que puede contener credenciales por defecto, vulnerabilidades de inyección **SQL**, fallos de autenticación, gestión pobre de sesiones, o formularios que permiten subir y ejecutar archivos.

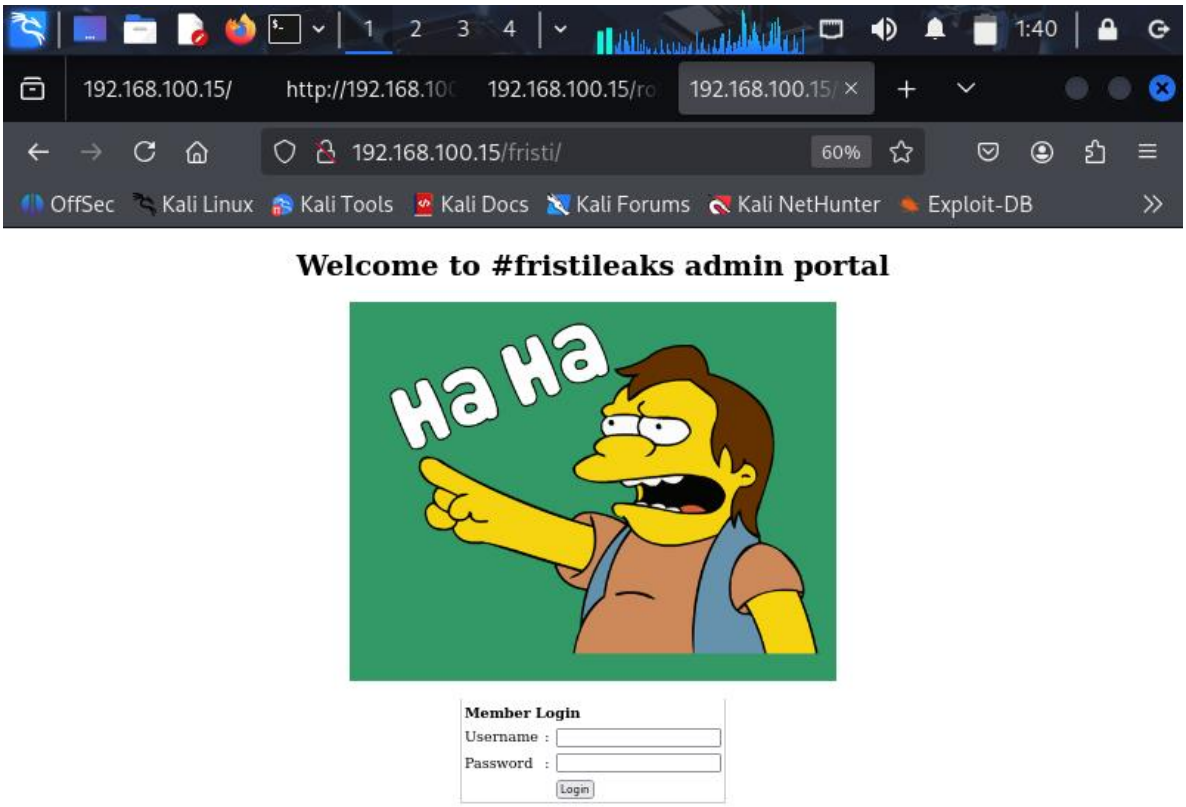


Ilustración 14 Esto es la página fristi

El servidor rechazó las credenciales debido a un fallo que se me presento con la red en la maquina atacante **kali Linux.** Lo que género que el backend procesara y respondiera con el mensaje de este fallo de autenticación

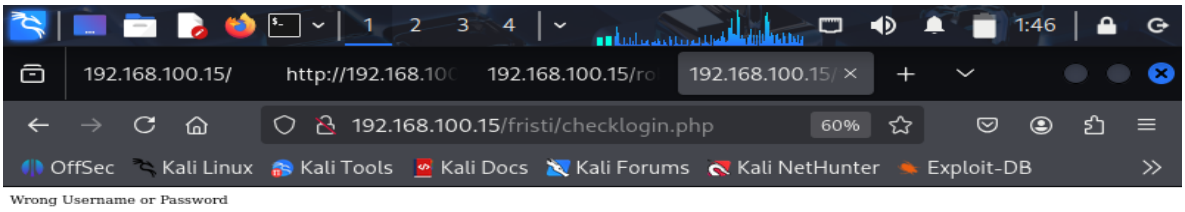


Ilustración 15 fallo de auntenticacion

En el view-source aparece un <**img src="data:image/png;base64**">. Eso significa que la imagen está embebida en el HTML como Base64, no cargada desde /images/.

Ya que este autor dejo antas pistas al parecer intencionalmente pensamos que pueden haber muchas más credenciales, comentarios e inputs ocultos o que la misma imagen puede contener información oculta.

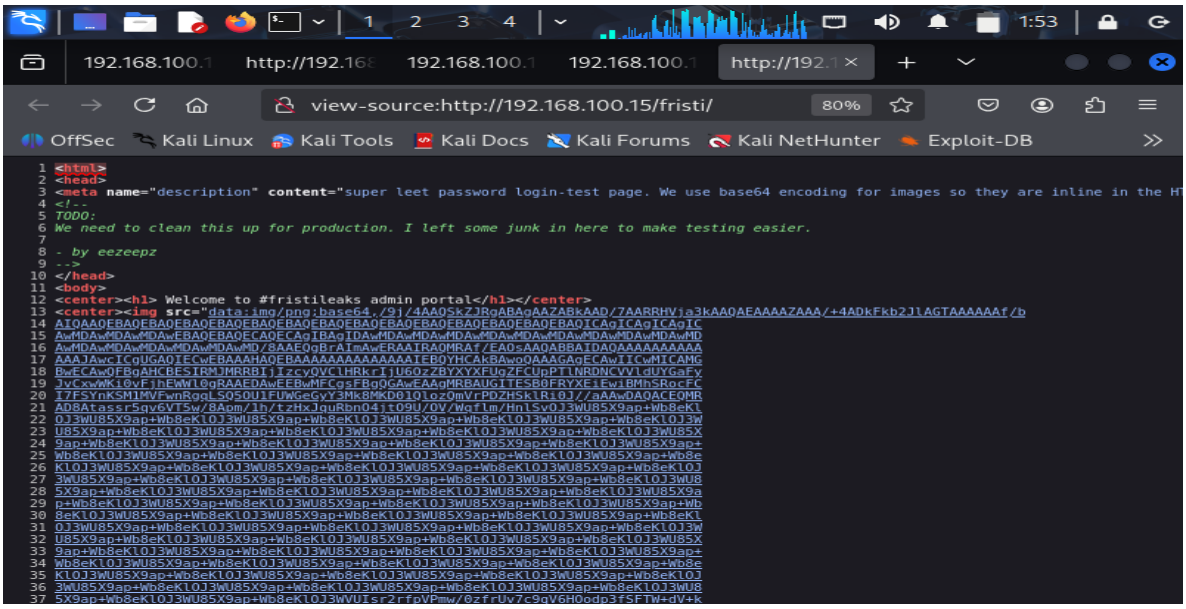


Ilustración 16 imagen oculta en HTML parte 1

Podemos extraer la imagen embebida en el HTML y ver qué información oculta contiene, además de confirmar la acción del formulario.

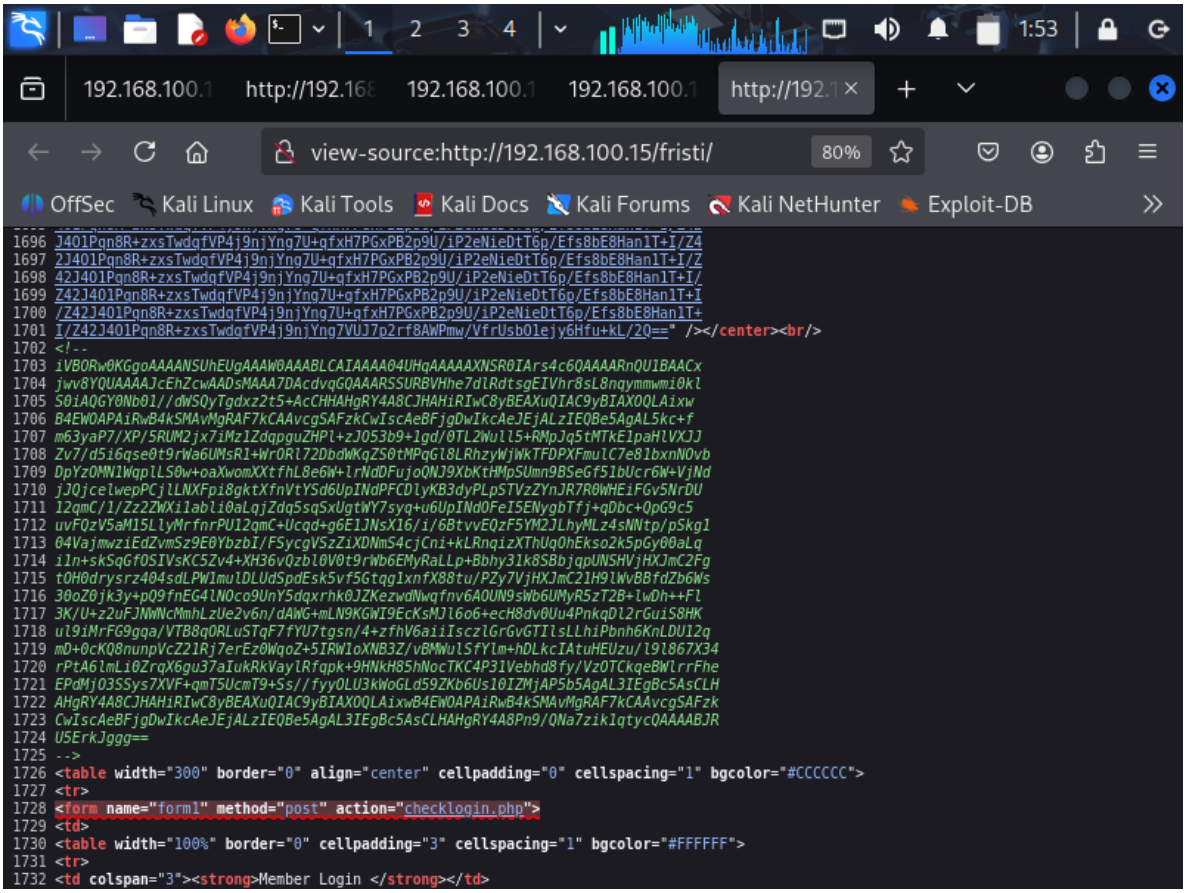


Ilustración 17 imagen oculta en HTML parte2

La conversión del comentario que habíamos llevado al descifrar en base64 nos arrojó otro código en binario que nos llevaría a una imagen **png.**

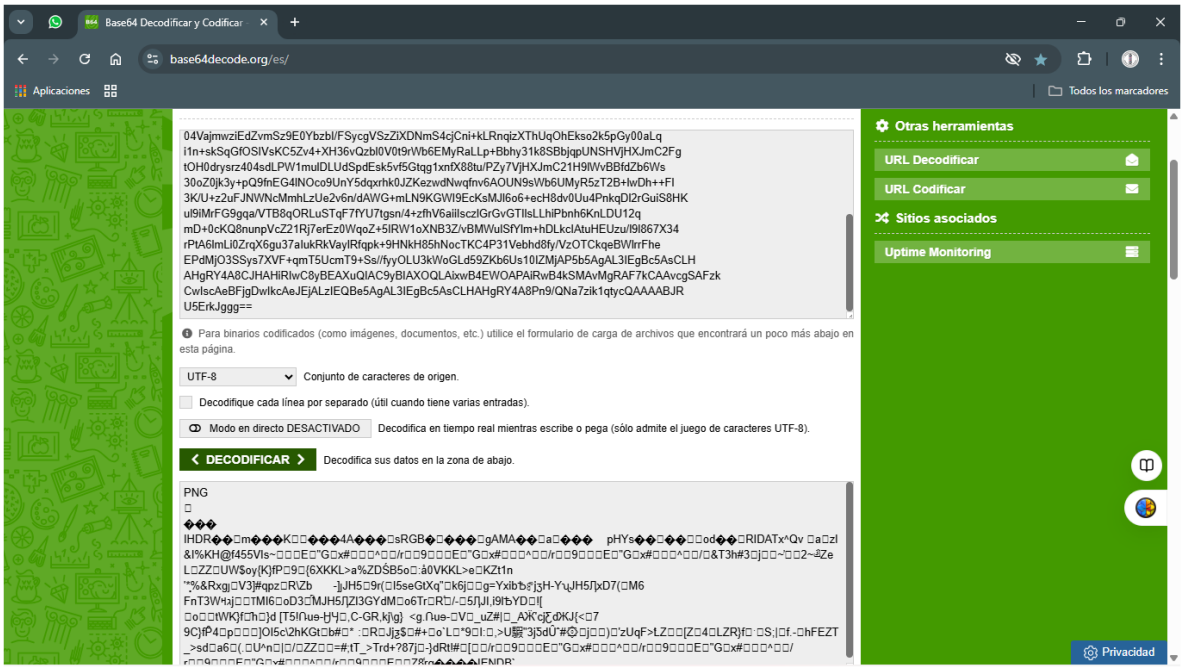


Ilustración 18 decodificador base64

Tras llevar a cabo la conversión, obtuve una imagen PNG que mostraba la inscripción "keKkeKKeKkEkkEk". Si bien a primera impresión parecía un enredo, dentro del contexto del reto lo interpreté como un indicio o código que tal vez me serviría más adelante para tratar de acceder al panel de gestión que había hallado con anterioridad.

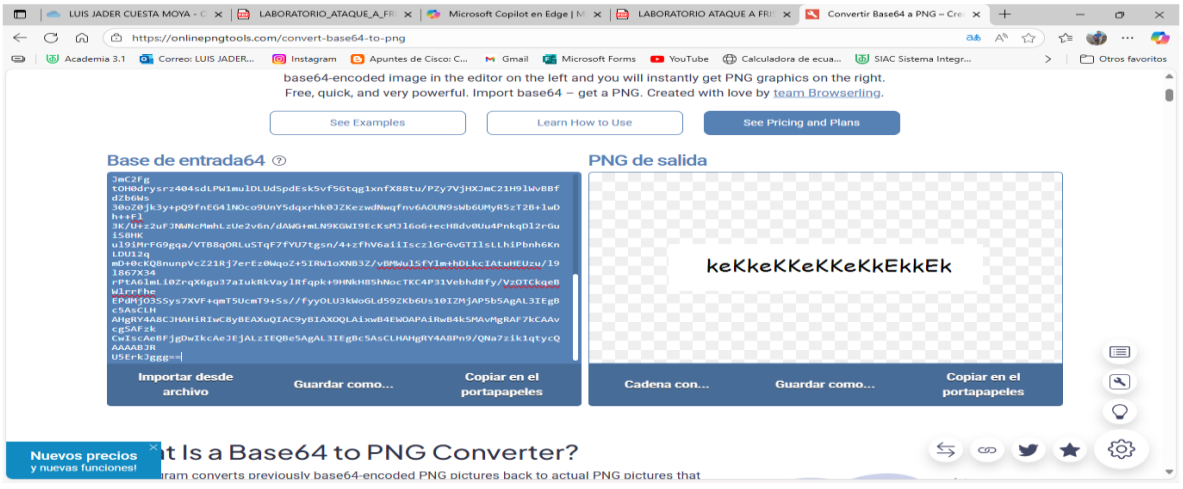


Ilustración 19 convertidor de base64 a png

Se accedió al formulario de carga de imágenes, Este formulario permite seleccionar y subir un archivo al servidor mediante el botón **"Upload image"**. Si el servidor no restringe adecuadamente los tipos de archivos, es posible subir scripts ejecutables como .**php.**
Si los archivos subidos se almacenan en una carpeta accesible y ejecutable /**uploads**/, se podría ejecutar código remoto.
Si el archivo mantiene su nombre original, puede predecirse o abusarse.

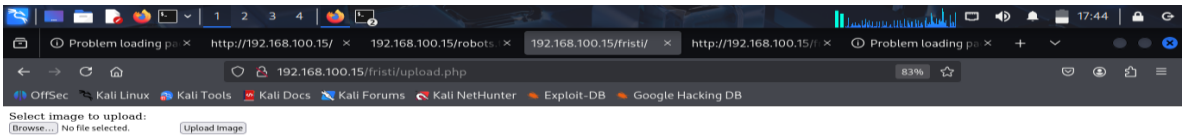


Ilustración 20 vulnerabilidad subida de archivos.

Aquí pude crear los ficheros dañinos. Mi objetivo era explotar la funcionalidad de subida de archivos que había detectado en el sistema. Para lograr esto, me dirigí a la ubicación /usr/share/webshells/php en Kali Linux, donde se almacenan diversos ejemplos de webshells listos para usar.

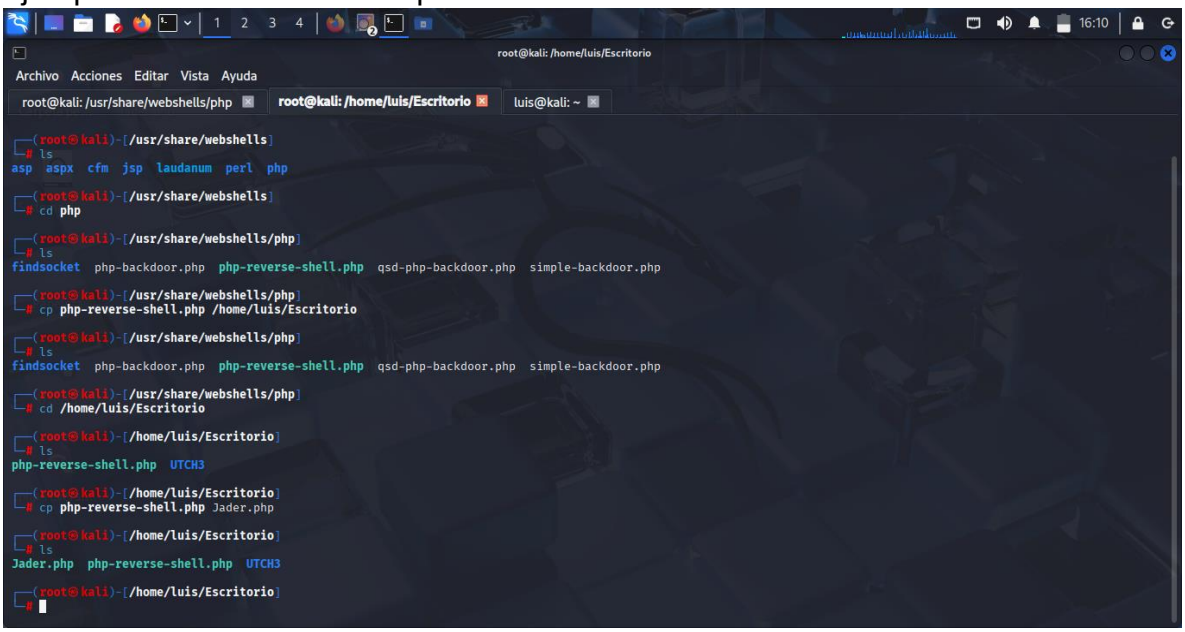


Ilustración 211 Creación de archivos maliciosos

Dentro de dicho directorio, di con el archivo php-reverse-shell. php, un script diseñado para originar una conexión inversa entre el equipo comprometido y mi propio sistema atacante. A continuación, trasladé este archivo a mi escritorio y lo renombré como **Jader.php**, con el fin de personalizarlo y, de esta manera, minimizar la posibilidad de alertar al sistema al cargarlo al servidor.

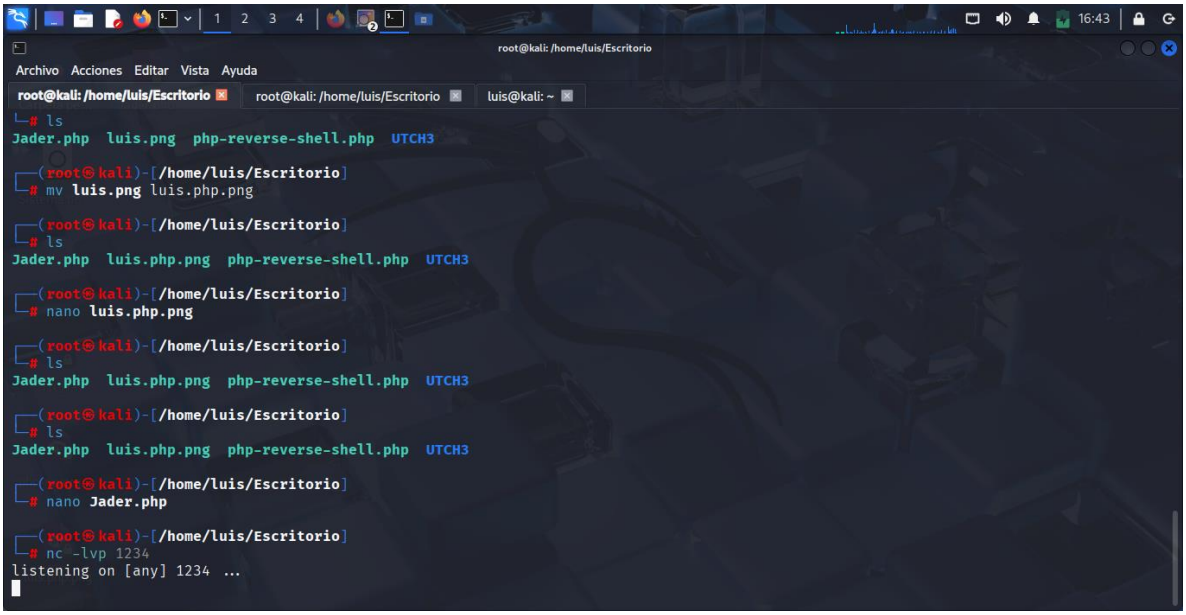


Ilustración 222 Creación de archivos maliciosos parte2

Se observa claramente el contenido del archivo **Jader.php.png**, que se está editando en **Kali Linux** utilizando el editor nano.

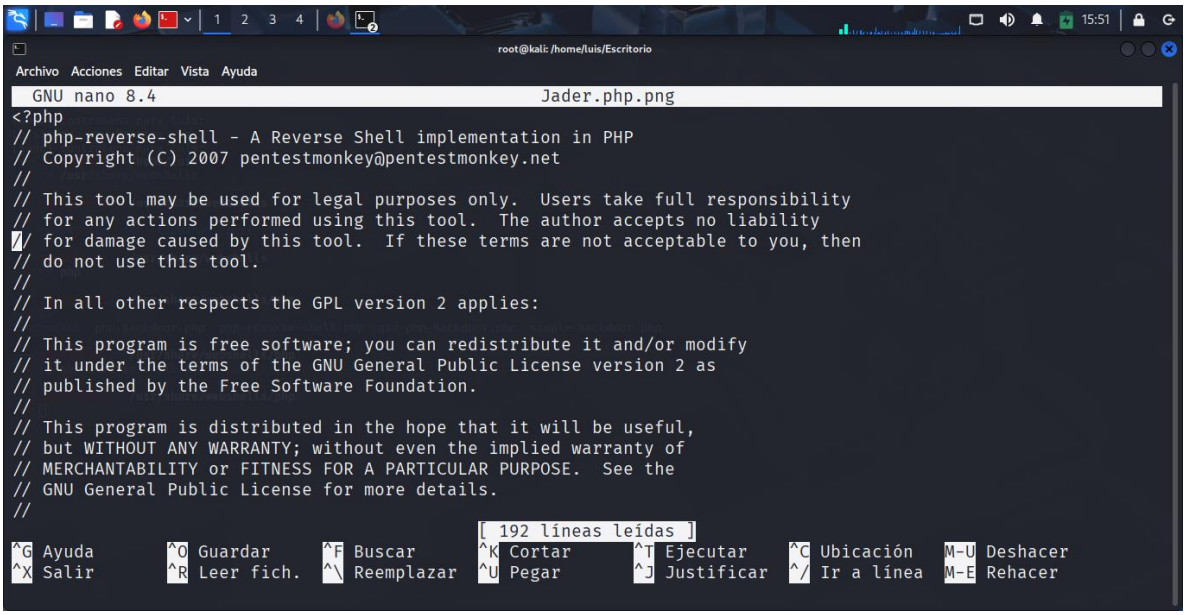


Ilustración 23 Editor nano parte1

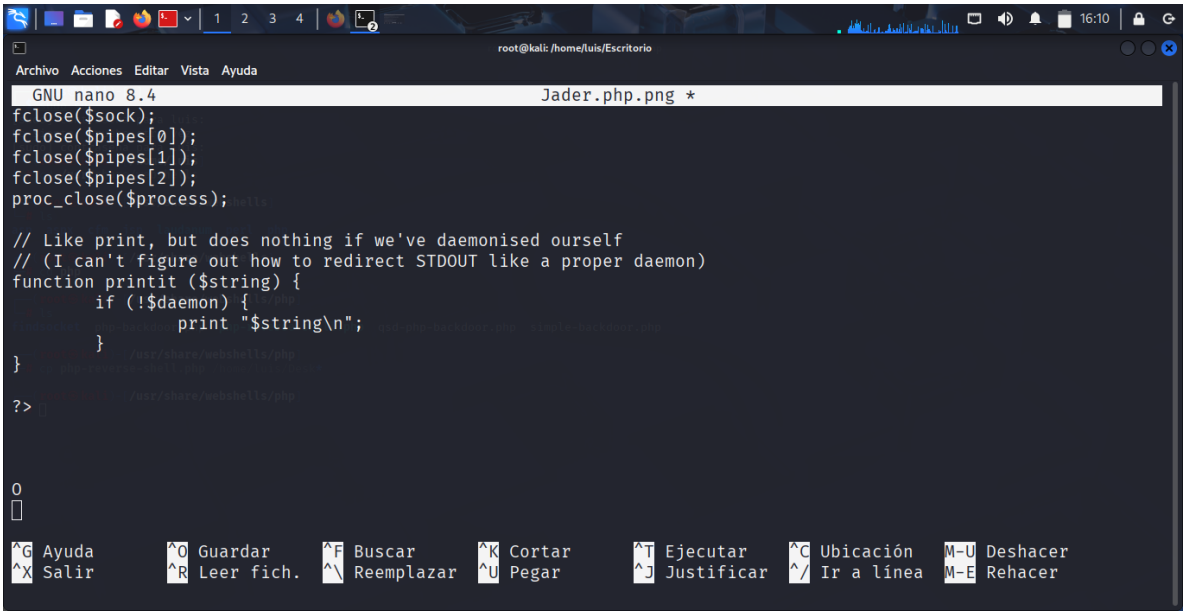


Ilustración 24 Editor nano parte2.

Proceso de subir un archivo al formulario de **upload.php** vemos como se está eligiendo entre varios archivos desde mi escritorio en mi maquina virtual Kali Linux.

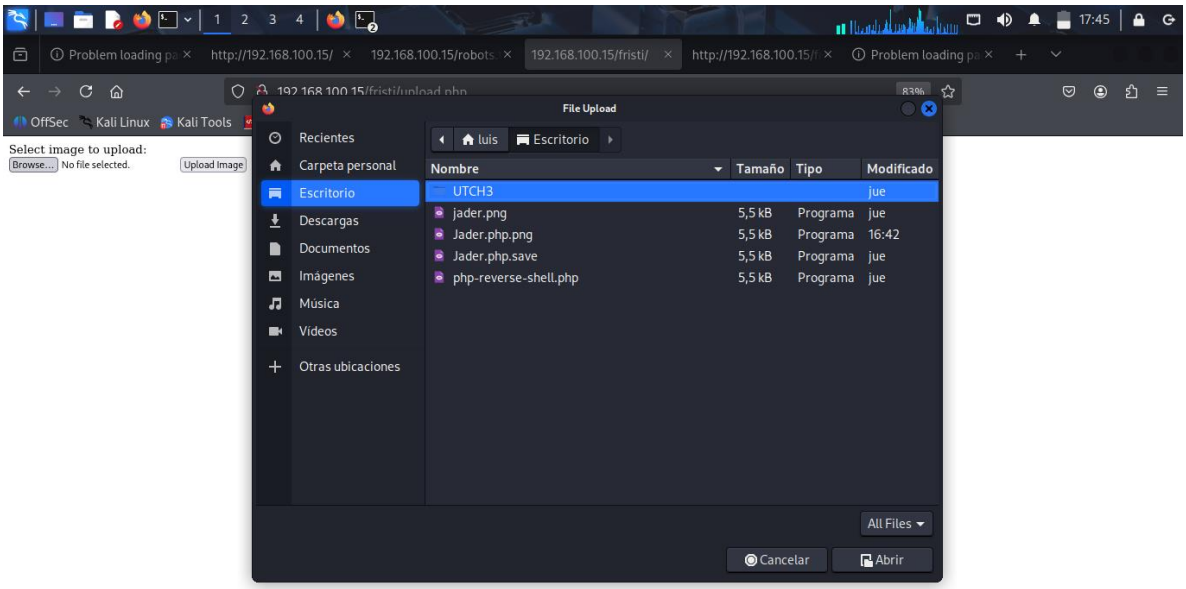


Ilustración 25 Cargar archivo malisioso.

La URL solicitada /fristi/uploads/Luis.php.png no se encontró en este servidor.
El navegador está buscando un archivo llamado Luis.php.png en el directorio /fristi/uploads/, pero el servidor responde con un **No** lo que indica que el archivo no existe o no está en esa ruta o se encontró en el servidor.

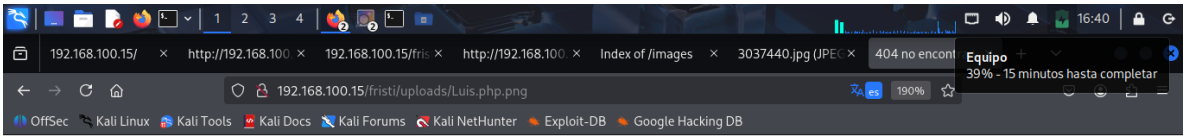


Ilustración 26 Luis.php.png no se encontró

Finalmente pude concretar el establecimiento de la conexión inversa con el sistema objetivo. Inicialmente, en mi máquina Kali, inicié la escucha con el comando "nc -lvp 1234", lo que significaba que estaría esperando conexiones entrantes en el puerto 1234. Luego, una vez que preparé y subí el archivo "Jader.php", al ejecutarse en el servidor, este generó la shell inversa hacia mi equipo.
En la parte baja de la terminal se observa que la conexión se realizó con éxito, mostrando datos del sistema de la máquina víctima (un sistema Linux, kernel 3.2.0, arquitectura x86_64) e indicando que la sesión se abrió bajo el usuario apache, lo cual valida que obtuve control inicial con privilegios acotados.
Una vez obtenido el acceso, ejecuté algunos comandos básicos como "whoami" y envié un ping a mi propia dirección (192.168.100.15) para verificar la comunicación, y todo funcionó correctamente.
Este fue un avance clave en la explotación, ya que a partir de ese instante contaba con acceso remoto al servidor, lo que me posibilitaría continuar con la escalada de privilegios y obtener un control más amplio del sistema.

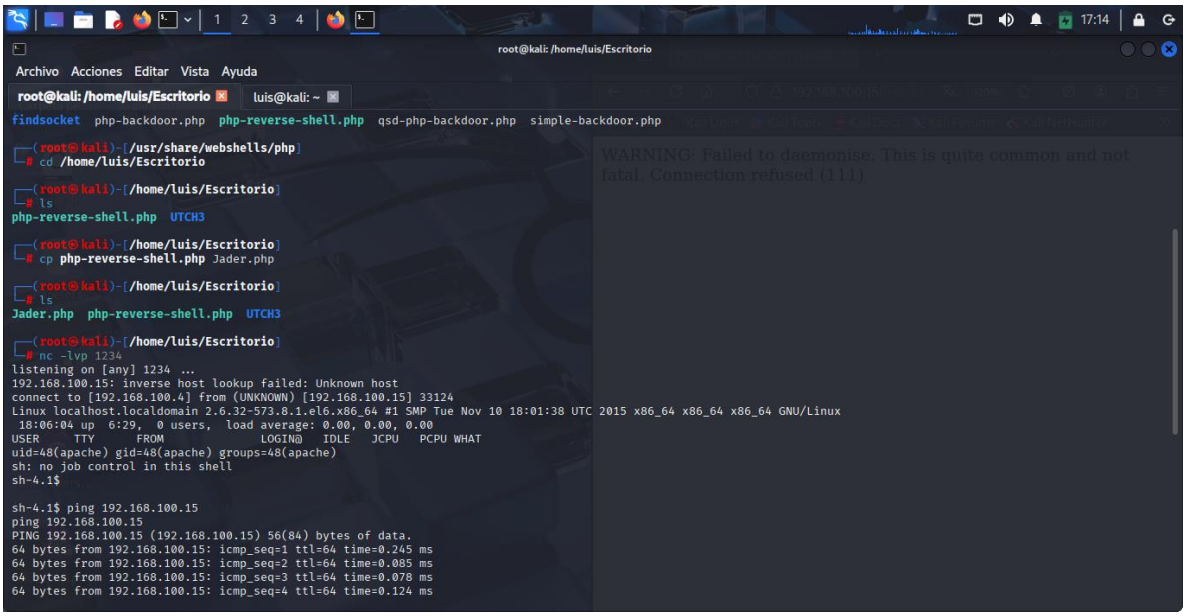
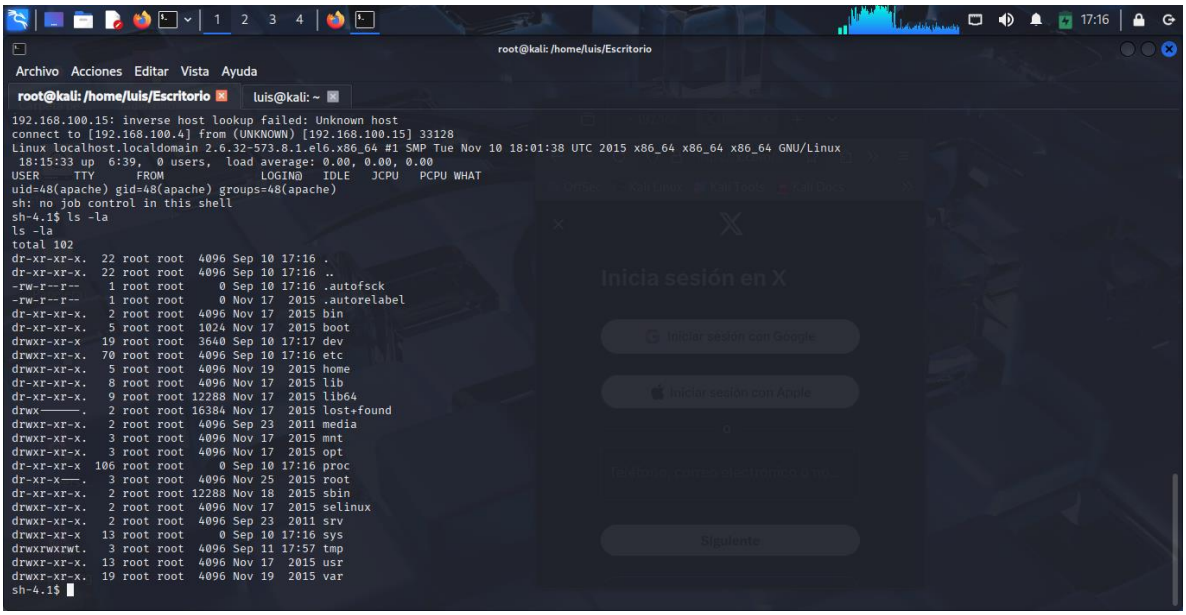


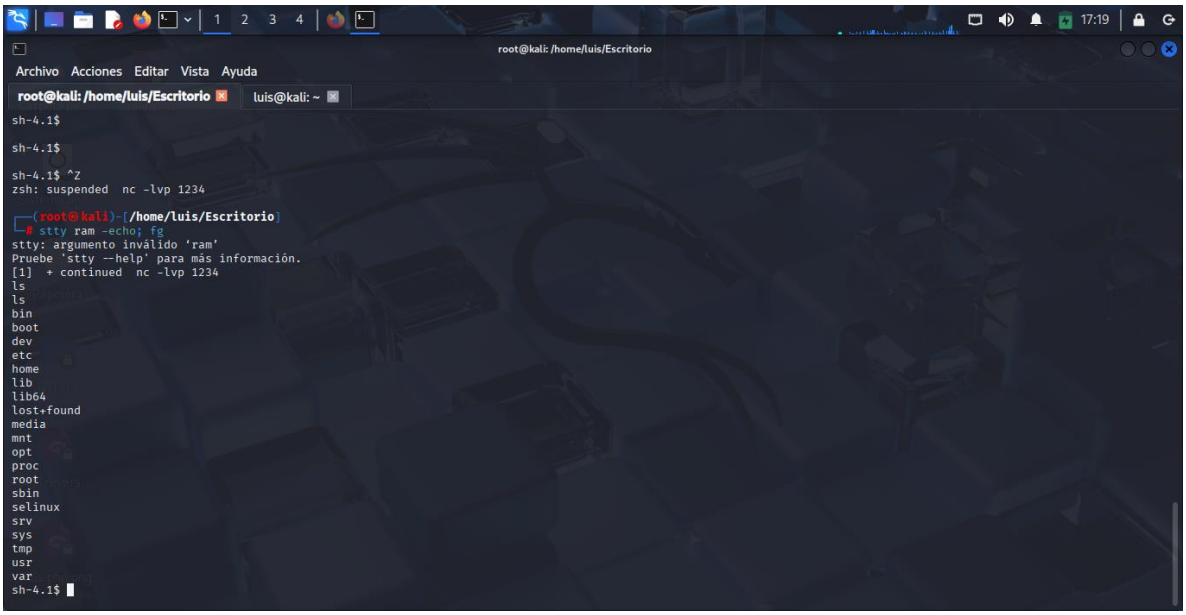
Ilustración 27 Explotación maliciosa en ejecución

Durante esta parte del proceso, noté que había conseguido comunicarme con el dispositivo elegido mediante un puerto específico (1234). Después de establecer esta comunicación, la interfaz me muestra información del sistema afectado, tal como el nombre del servidor, la versión del núcleo y los usuarios actualmente activos.
Una vez dentro, conseguí acceso a una consola interactiva, lo cual me facilitó la tarea de empezar a introducir instrucciones directamente en el sistema. Por ejemplo, usé la instrucción ls -la para visualizar los directorios y ficheros principales de la máquina. En ese momento se aprecia la estructura común de un sistema Linux, con

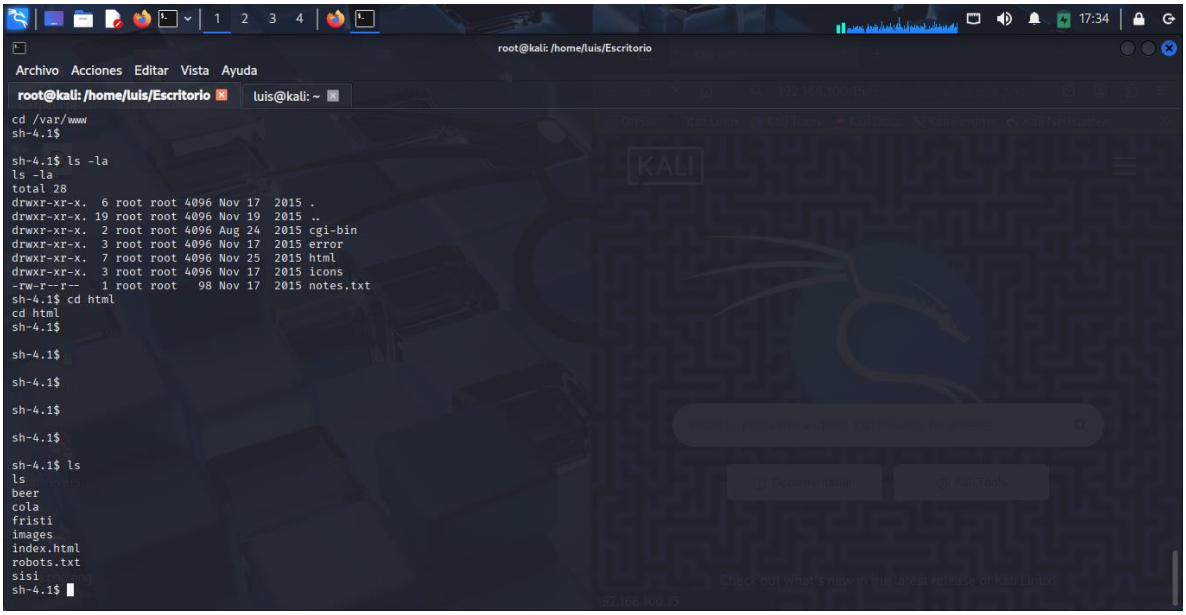
carpetas como /boot, /etc, /home, /lib y otras que ratifican que tengo acceso al entorno interno del dispositivo en peligro.
Esta acción señala que la debilidad fue explotada de forma exitosa, ya que pasé de intentar una comunicación a obtener acceso remoto al sistema y a tener la capacidad de interactuar con sus ficheros.



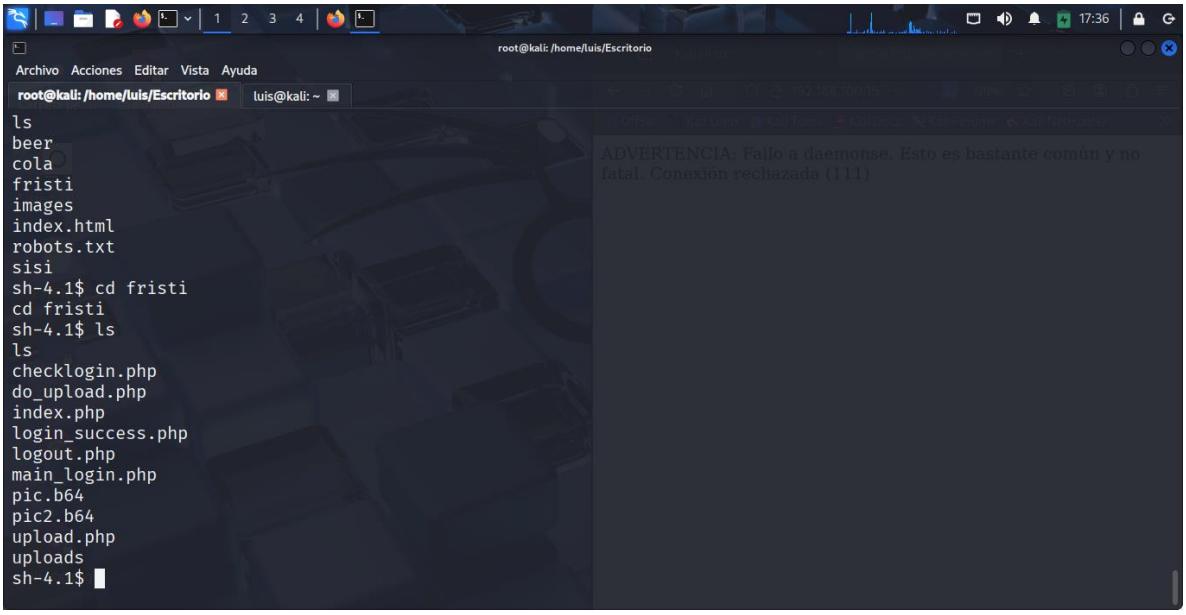
En esta etapa del procedimiento, observé que pude interactuar con el equipo deseado a través de un puerto determinado (1234). Una vez establecida la conexión, la consola me mostró detalles del sistema comprometido, tales como el nombre del servidor, la versión del kernel y los usuarios en línea.
Una vez dentro, conseguí acceso a una línea de comandos interactiva, lo cual me brindó la opción de empezar a ejecutar órdenes directamente en el equipo remoto. A modo de ejemplo, utilicé el comando ls -la para ver los directorios y archivos principales del ordenador. Allí se aprecia la estructura típica de un sistema Linux, con carpetas como /boot, /etc, /home, /lib y otras que indican que tengo acceso al entorno interno del equipo víctima.
Esta fase demuestra que la vulnerabilidad fue explotada de forma exitosa, ya que pasé de intentar establecer una conexión a tener acceso remoto al sistema y la capacidad de interactuar con sus archivos.

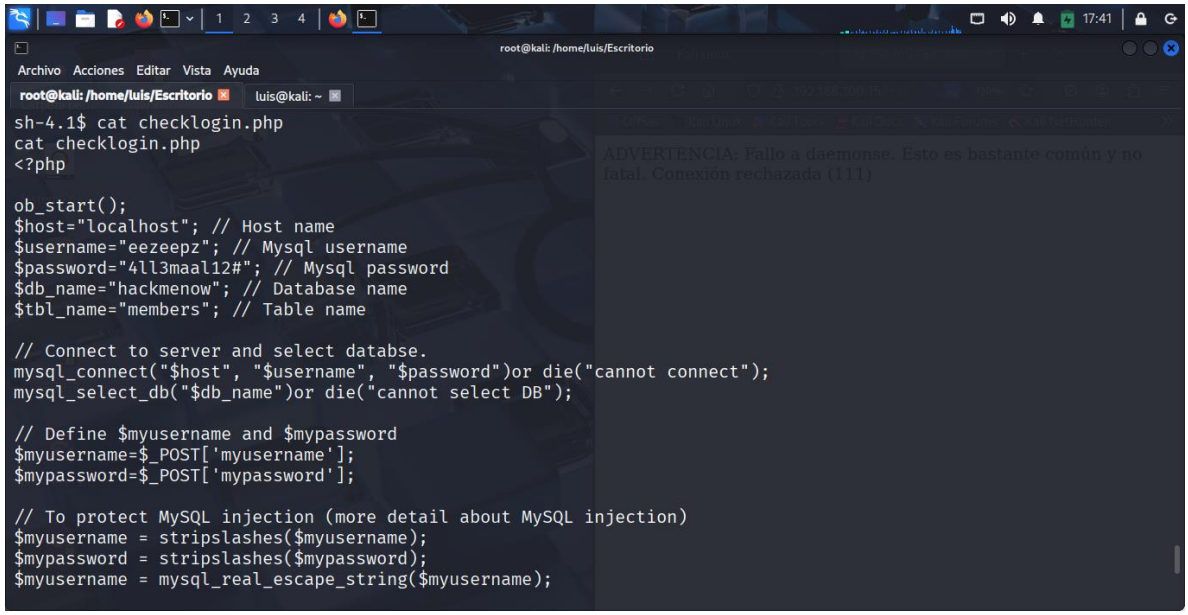


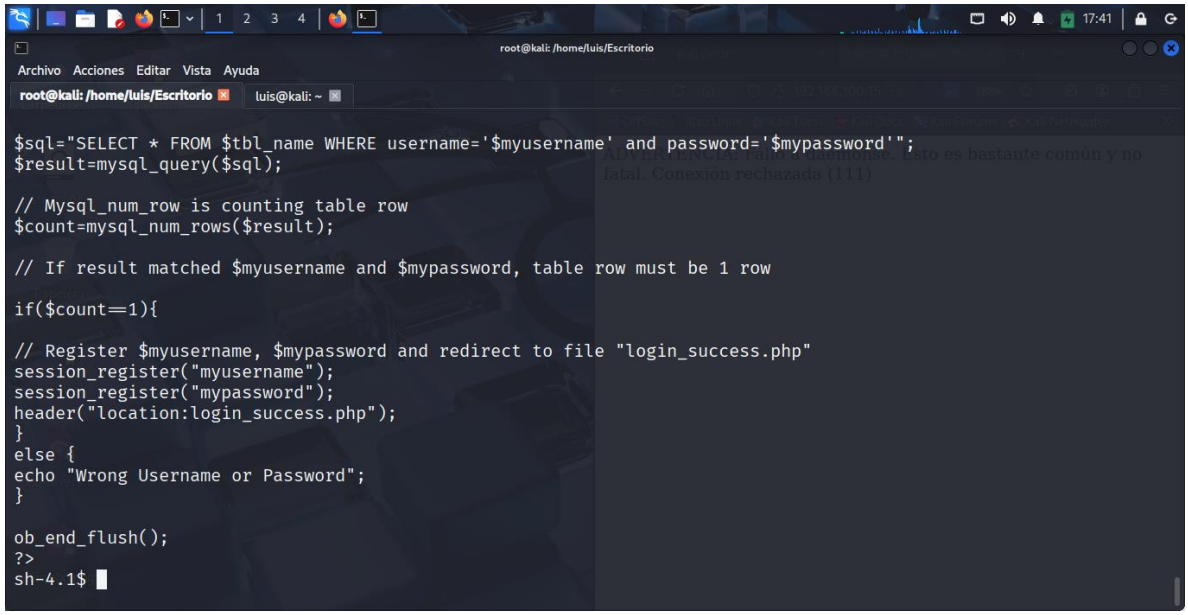
Dentro del análisis del sistema comprometido, decidí enfocarme en el directorio /var/www, un lugar habitual para los archivos del servidor web. Al usar el comando ls -la, vi varias carpetas comunes en un servidor, como cgi-bin, html, error e icons, que suelen guardar scripts, páginas y ajustes del sitio. Además, encontré un archivo llamado notes. txt, algo importante porque estos archivos a veces tienen datos valiosos, como notas, configuraciones o hasta contraseñas que podrían ayudar a obtener más permisos. Este hallazgo muestra que no solo tengo acceso completo al sistema operativo, sino que también puedo explorar directamente los archivos del servidor web.

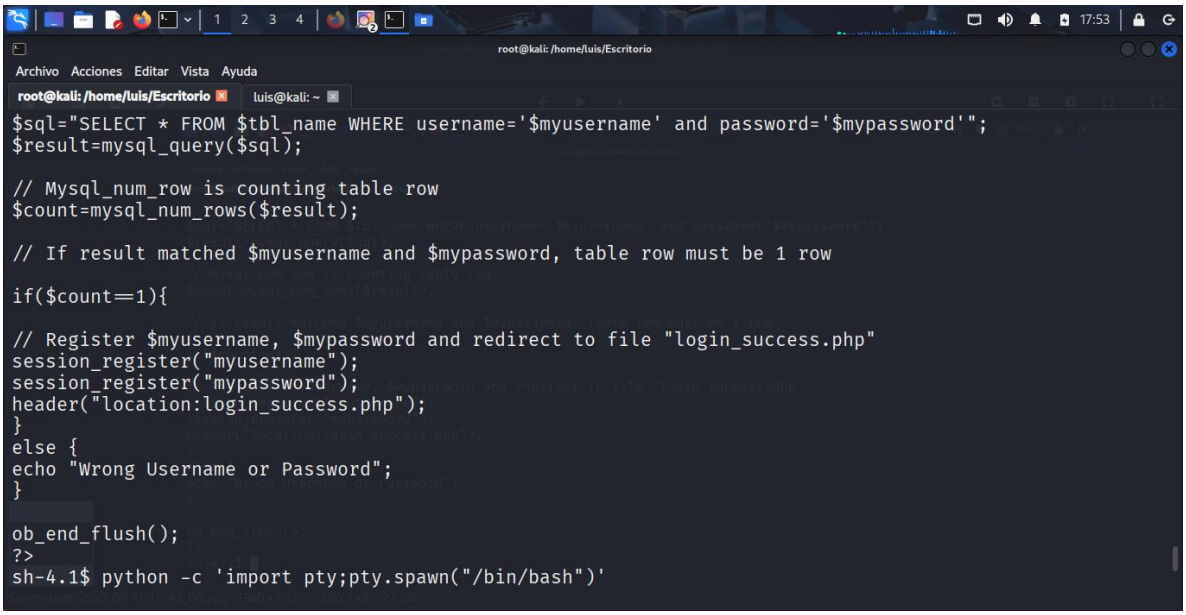


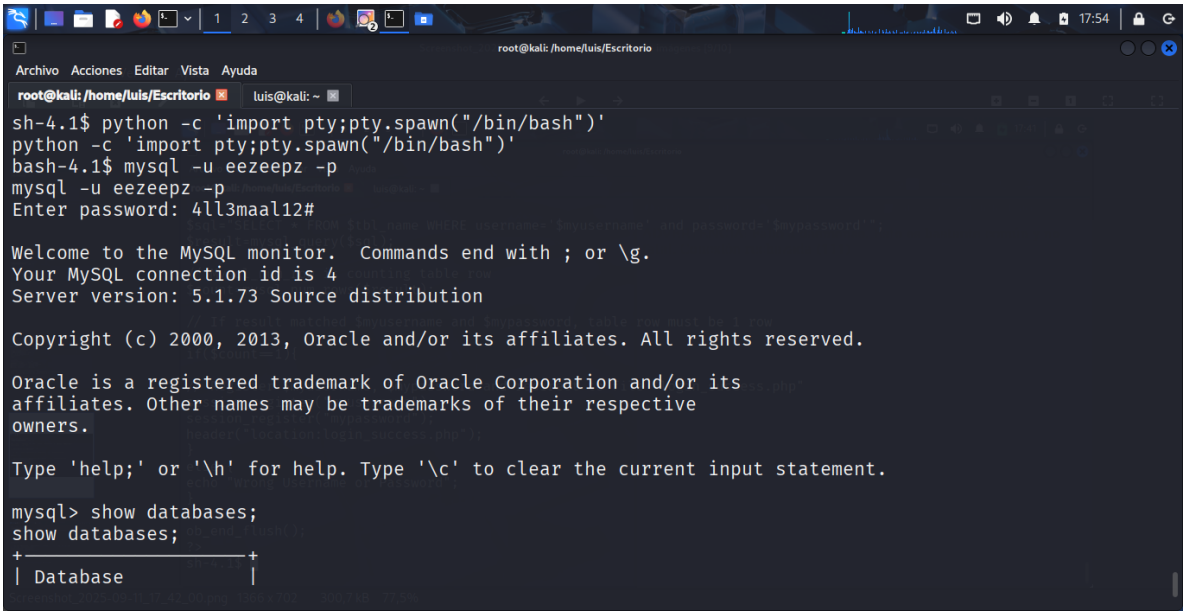
En directorio /var/www/html, que es donde suelen residir los archivos esenciales de un sitio web. Al usar el comando ls para examinar su contenido, descubrí varias carpetas y archivos dignos de mención. Entre ellos, resaltaban beer, cola, fristi y sisi, que aparentan ser carpetas o apartados extra del sitio, quizá empleados para organizar recursos o funcionalidades concretas. También se aprecian ficheros clave como index. html, que comúnmente representa la página de inicio del sitio, y robots. txt, que habitualmente incluye indicaciones para los buscadores, aunque a veces puede desvelar rutas ocultas o delicadas en el servidor. Igualmente, está presente la carpeta images, que frecuentemente es el sitio para almacenar elementos gráficos, pero que en ciertas ocasiones puede albergar archivos importantes si no se administra correctamente.

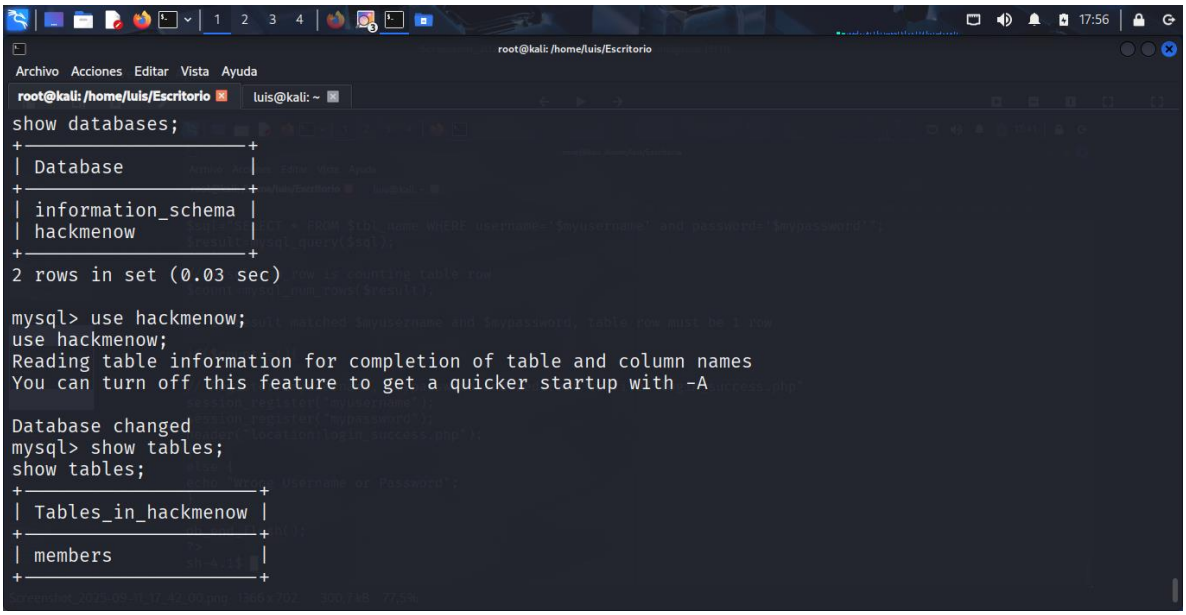


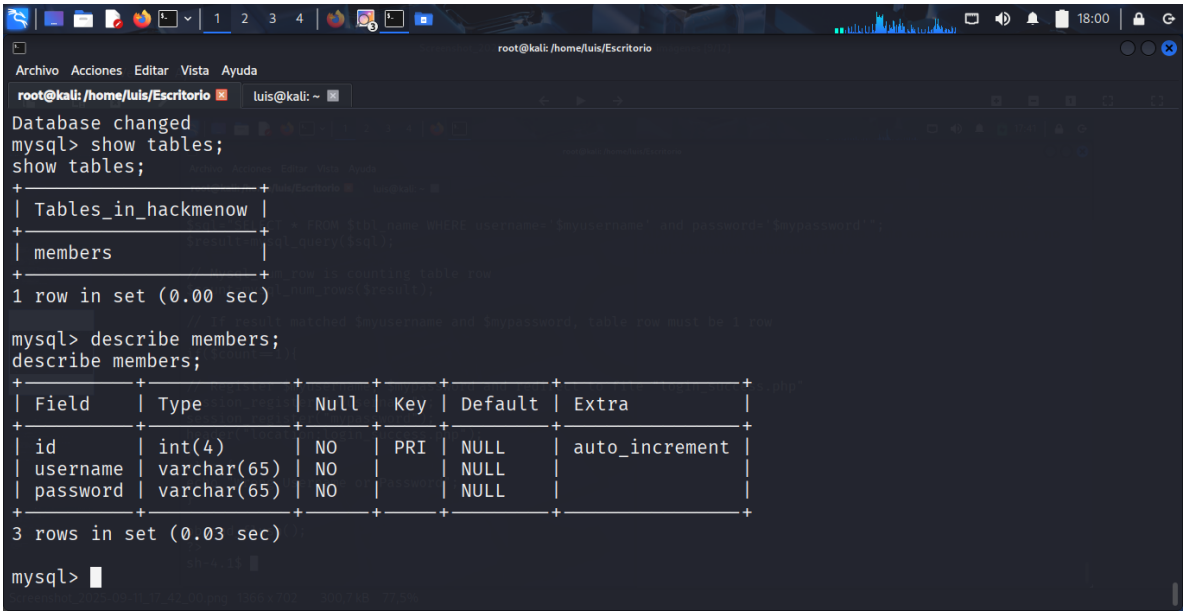

17:41
root@kali: /home/luis/Escritorio
Archivo Acciones Editar Vista Ayuda
root@kali: /home/luis/Escritorio
luis@kali: ~
sh-4.1$ cat checklogin.php
cat checklogin.php
<?php
ob_start();
$host="localhost"; // Host name
$username="eezeepz"; // Mysql username
$password="4ll3maal12#"; // Mysql password
$db_name="hackmenow"; // Database name
$tbl_name="members"; // Table name
// Connect to server and select databse.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");
// Define $myusername and $mypassword
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
// To protect MySQL injection (more detail about MySQL injection)
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);

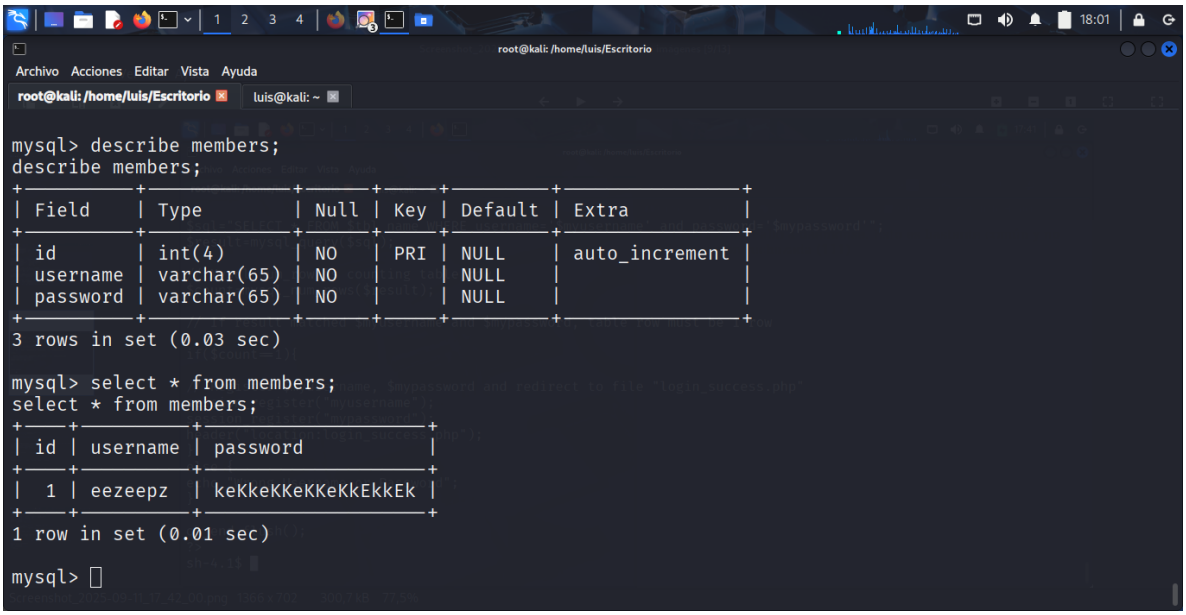

17:41
root@kali: /home/luis/Escritorio
Archivo Acciones Editar Vista Ayuda
root@kali: /home/luis/Escritorio
luis@kali: ~
$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and password='$mypassword'";
$result=mysql_query($sql);
// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){
// Register $myusername, $mypassword and redirect to file "login_success.php"
session_register("myusername");
session_register("mypassword");
header("location:login_success.php");
}
else {
echo "Wrong Username or Password";
}
ob_end_flush();
?>
sh-4.1$


17:53
root@kali: /home/luis/Escritorio
Archivo Acciones Editar Vista Ayuda
root@kali: /home/luis/Escritorio
luis@kali: ~
$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and password='$mypassword'";
$result=mysql_query($sql);
// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){
// Register $myusername, $mypassword and redirect to file "login_success.php"
session_register("myusername");
session_register("mypassword");
header("location:login_success.php");
}
else {
echo "Wrong Username or Password";
}
ob_end_flush();
?>
sh-4.1$ python -c 'import pty;pty.spawn("/bin/bash")'


17:54
root@kali: /home/luis/Escritorio
Archivo Acciones Editar Vista Ayuda
root@kali: /home/luis/Escritorio
luis@kali: ~
sh-4.1$ python -c 'import pty;pty.spawn("/bin/bash")'
python -c 'import pty;pty.spawn("/bin/bash")'
bash-4.1$ mysql -u eezeepz -p
mysql -u eezeepz -p
Enter password: 4ll3maal12#
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.1.73 Source distribution
Copyright (c) 2000, 2013, 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> show databases;
show databases;
+--------------------+
| Database |


root@kali: /home/luis/Escritorio
Archivo Acciones Editar Vista Ayuda
root@kali: /home/luis/Escritorio
luis@kali: ~
show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| hackmenow |
+--------------------+
2 rows in set (0.03 sec)
mysql> use hackmenow;
use hackmenow;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> show tables;
show tables;
+----------------------+
| Tables_in_hackmenow |
+----------------------+
| members |
+----------------------+


root@kali: /home/luis/Escritorio
Archivo Acciones Editar Vista Ayuda
root@kali: /home/luis/Escritorio
luis@kali: ~
Database changed
mysql> show tables;
show tables;
+----------------------+
| Tables_in_hackmenow |
+----------------------+
| members |
+----------------------+
1 row in set (0.00 sec)
mysql> describe members;
describe members;
+----------+-------------+------+-----+---------+----------------+
| Field | Type | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+----------------+
id	int(4)	NO	PRI	NULL	auto_increment
username	varchar(65)	NO		NULL	
password	varchar(65)	NO		NULL	
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.03 sec)
mysql>



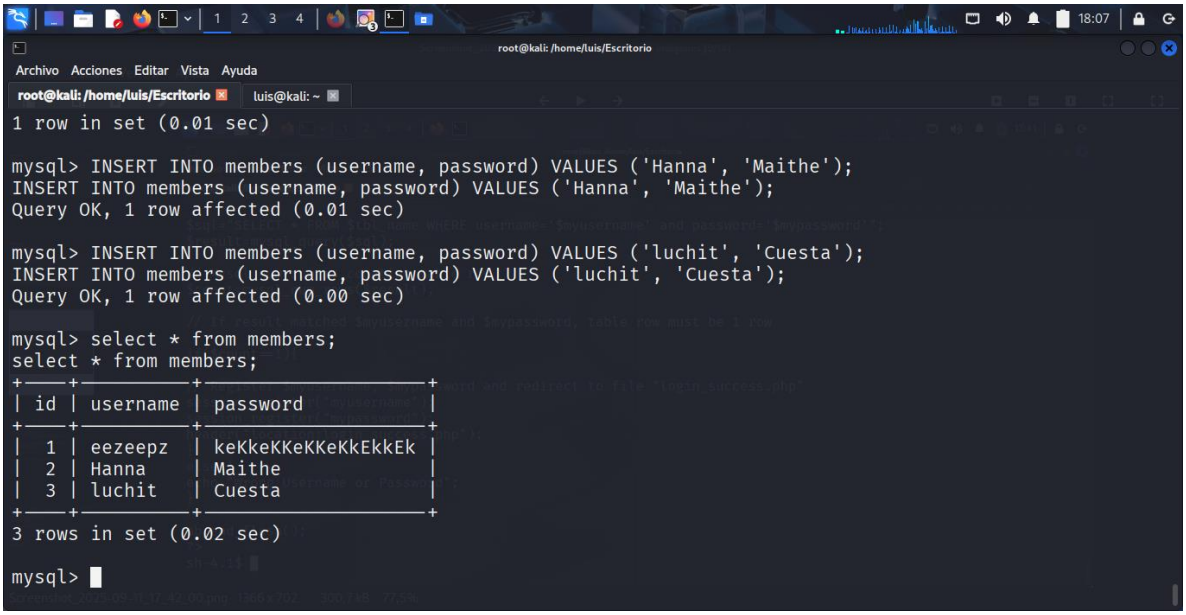
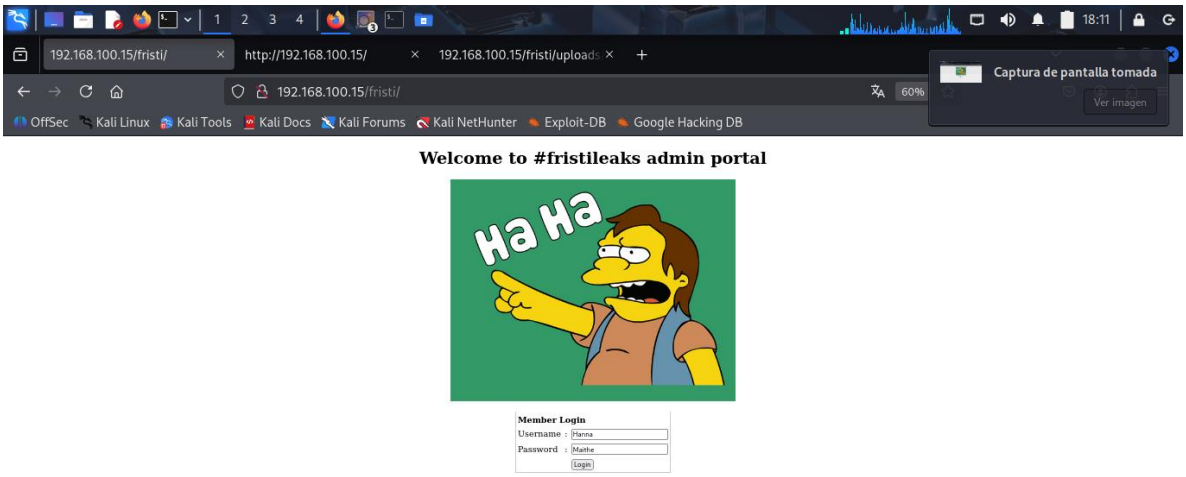
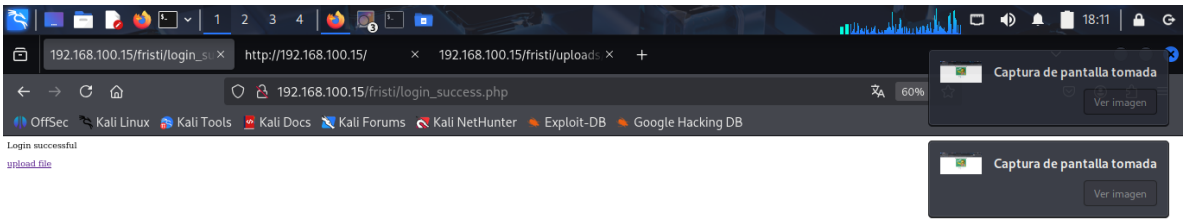
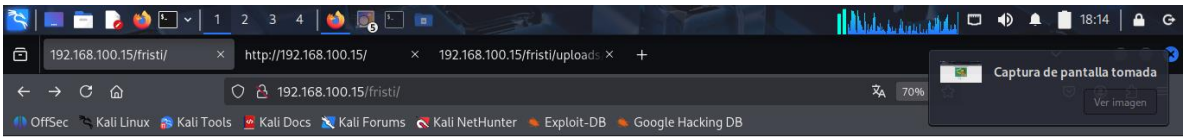


Ilustración 28 Creacion de los usuarios.



Anteriormente en la ilustración 15 pudimos apreciar un mensaje de fallo de autenticación, en este caso se ha Conseguido acceso al panel, ahora hay una funcionalidad de subida.





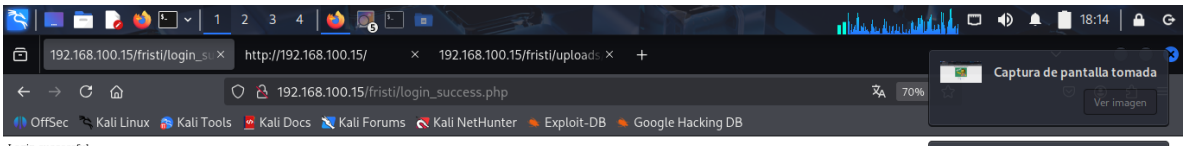
Welcome to #fristileaks admin portal

Member Login

Username : Luchit

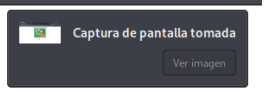
Password : Cuesta

Login



Login successful

upload file



JUSTIFICACIÓN

La realización de este laboratorio se justifica en la necesidad de formar profesionales competentes en seguridad informática, capaces de identificar riesgos antes de que sean aprovechados por atacantes malintencionados. El uso de FristiLeaks como entorno controlado permite experimentar con vulnerabilidades reales sin comprometer sistemas productivos, lo cual enriquece la experiencia de aprendizaje y fortalece las competencias prácticas del estudiante. Asimismo, el ejercicio contribuye al entendimiento del ciclo de la ciber seguridad, que inicia con la detección de debilidades y culmina con la implementación de medidas de protección. En este sentido, el trabajo no solo es un entrenamiento técnico, sino también una herramienta pedagógica que promueve la conciencia sobre la importancia de proteger la información y los recursos digitales en un mundo cada vez más interconectado.

CONCLUSIÓN

El desarrollo de la práctica permitió evidenciar que una configuración deficiente y la ausencia de controles de seguridad adecuados pueden facilitar la explotación de un sistema. A través de Kali Linux se logró establecer acceso a la máquina víctima, demostrando cómo vulnerabilidades comunes como la falta de protección en directorios, el uso de formularios inseguros y la gestión inadecuada de archivos pueden comprometer seriamente la integridad de un servidor.

De igual manera, se concluye que la seguridad informática no debe entenderse únicamente como un componente técnico, sino como una responsabilidad estratégica, donde la prevención, la actualización constante y la auditoría proactiva son claves para salvaguardar los sistemas de información. Finalmente, esta experiencia práctica constituye un aporte valioso en la formación académica, al proporcionar un acercamiento realista a las dinámicas de ataque y defensa en el campo de la ciberseguridad.